

SERWONAPĘDY I SILNIKI KROKOWE

www.ep.com.pl

ELEKTRONIKA PRAKTYCZNA + FTP

● Międzynarodowy magazyn elektroników konstruktorów ● Listopad ● 11/2019 ●



TYLKO DLA
PRENUMERATORÓW

Niezbędnik elektronika na DVD

inspirujące, użyteczne projekty

Bezprzewodowa EKO-stacja do pomiaru jakości powietrza

- Raspipool – system automatyki do przydomowego basenu
- Sensor pracy mięśni z wizualizacją w sieci Web i na diodach LED RGB
- MIDIswitch – przełącznik MIDI do wzmacniacza lampowego
- Sztuczne obciążenie w.cz. o impedancji 50 Ohm
- Przenośny rejestrator temperatury z zapisem na pendrive i oprogramowaniem do wizualizacji
- Rysino – płytką ewaluacyjną z FPGA Intel MAX10
- Miernik odległości na bazie czujnika GP2Y0A02YK0F
- Uniwersalny układ czasowy 230 V
- Regulowany bezpiecznik
- Półprzewodnikowa cewka Tesli o podwójnym rezonansie

podzespoły, sprzęt, aplikacje

Nowe przekaźniki firmy Relpol • Zasilacze LED do zadań specjalnych • AudioDSP zestaw z procesorem Sigma DSP ADAU1701

tutoriale

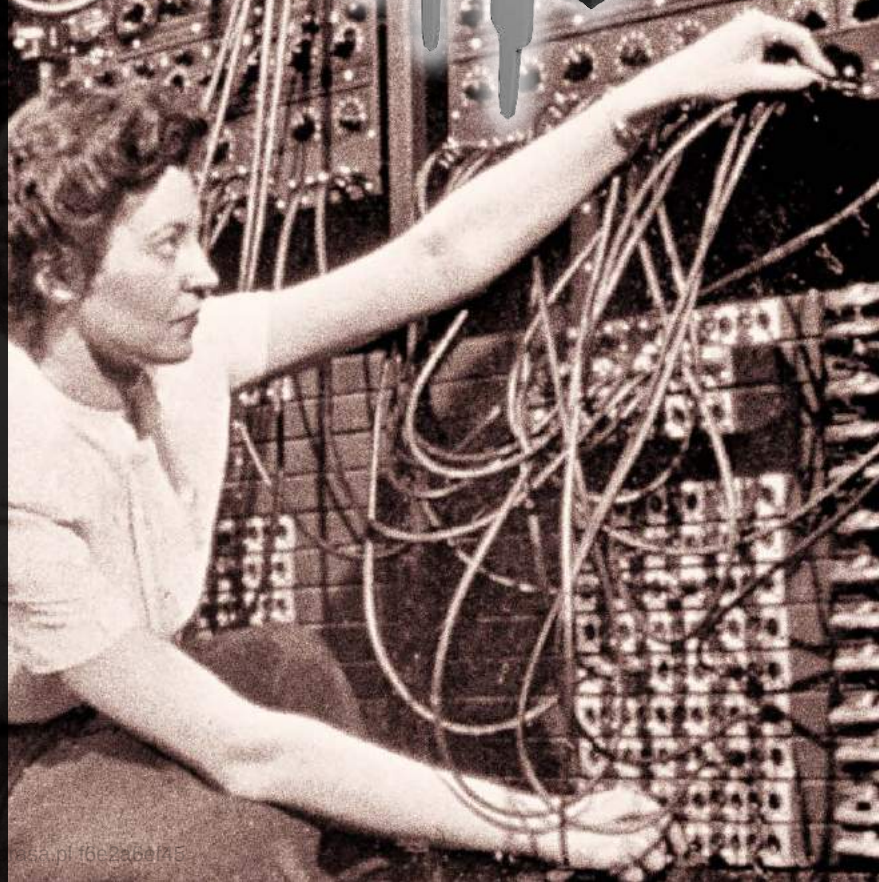
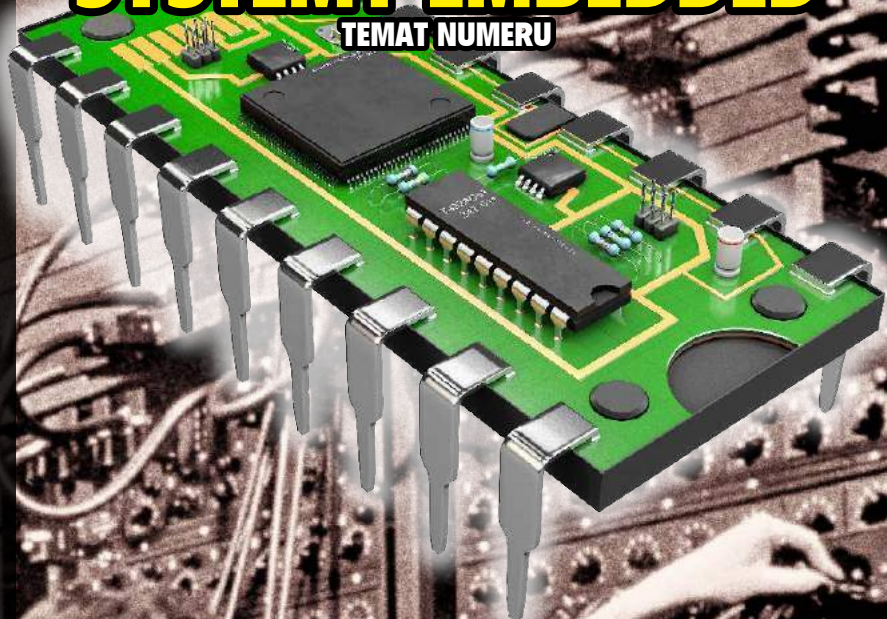
Dioda elektroluminescencyjna, czy fotodioda? A może jedno i drugie...? • Sterowanie diodami RGB z Raspberry Pi • Symulacja i pomiar czyli LTspice i Analog Discovery 2 w rękach konstruktora. Pomiar pojemności mostkiem Wheatstone'a

kursy

Systemy dla Internetu Rzeczy. Nordic Thingy:91 – platforma do prototypowania dla mobilnego IoT • ISIXRTOS v3 mini system operacyjny dla mikrokontrolerów rodziny M0/M3/M4/M7. Panel dotykowy • RISC-V – budujemy własny mikrokontroler

SYSTEMY EMBEDDED

TEMAT NUMERU



16 zł (w tym 8% VAT) • PRICE: 8 EUR • Nakład 27000 egz.

ISSN 1230-3526 Indeks 357677



9 771230 352191

11 >

NOWE WYDANIE SPECJALNE!

Poznaj najnowsze wydanie specjalne magazynu Digital Camera Polska. Tym razem w całości poświęcone jest pracy z ruchomym obrazem. To obszerne kompendium wiedzy, które w przystępny sposób wprowadza początkujących i średniozaawansowanych twórców wideo w arkaana sztuki filmowej.

WYDANIE SPECJALNE DIGITAL CAMERA POLSKA | NOWOŚĆ NA POLSKIM RYNKU!

Digital Video Book

Kompletny przewodnik dla kreatywnego twórcy wideo

186
STRON
PORAD
I TRIKÓW



WYBIERAMY SPRZĘT!

- ☒ Aparaty do filmowania
- ☒ Gimbałe
- ☒ Oświetlenie
- ☒ Backup
- ☒ Workflow

Jak kręcić profesjonalnie:
Światło | Dźwięk | Postprodukcja

Przejrzyj i zamów na www.UlubionyKiosk.pl
(przesyłka GRATIS)



STM32MP1 microprocessor Your next STM32 success

► discover how Linux meets STM32



Nowa rodzina heterogenicznych mikroprocesorów aplikacyjnych

Rodzina STM32MP1 w skrócie

- ★ **2 x Cortex-A7 @650MHz**
- ★ **1 x Cortex-M4 @209MHz**
- ★ **32-b interfejs pamięci DDR3/3L**
- ★ **GPU 3D/OpenGL ES2.0**
- ★ **MAC Ethernet 1Gb/s**
- ★ **interfejs MIPI-DSI**
- ★ **koprocesor kryptograficzny**
- ★ **ekosystem bazujący na STM32CUBE**
- ★ **peryferia znane z mikrokontrolerów STM32**

Szeroki wybór i łatwość użycia

- ★ **8 modeli MPU różniących się wyposażeniem**
- ★ **obudowy BGA od 257 do 448 pinów**
- ★ **możliwość implementacji systemu na PCB 4l**
- ★ **wsparcie dla Linuxa i Androida**



★ dystrybucja ★ szkolenia ★ wsparcie techniczne ★
eprasa.pl f6e2a6ef45



life.augmented

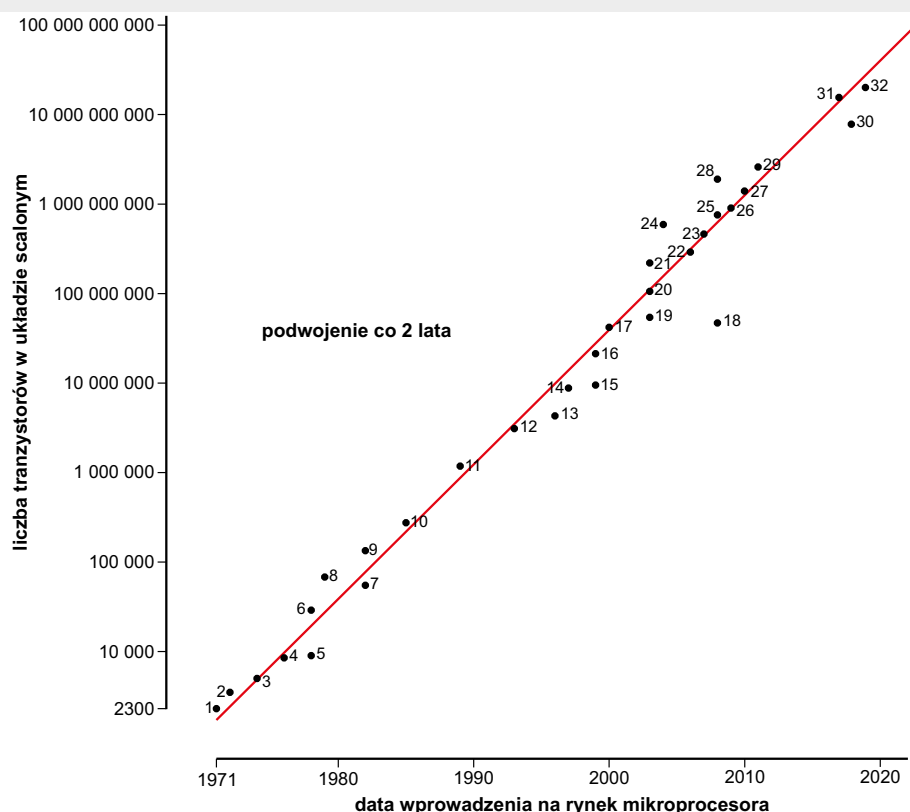
Koniec i co dalej

Czy wyczerpują się możliwości rozwoju mikroelektroniki opartej na technologii krzemowej? A jeśli tak, to co dalej?

Dotychczas przez 60 lat od powstania pierwszych układów scalonych skala integracji, tj. liczba tranzystorów w pojedynczym czipie, podwajała się co 2 lata. Już w połowie lat sześćdziesiątych zauważył tę prawidłowość Gordon Moore, przyszły szef Intela. Jak widać na **rysunku 1**, tzw. prawo Moore'a sprawdza się do dziś. Jak długo jeszcze? Oto jest pytanie. Liczba tranzystorów w mikroprocesorach lub pamięciach sięga już dziesiątków miliardów. I co dalej?

Wiadomo, że zwiększanie liczby tranzystorów w układzie scalonym odbywa się dzięki miniaturyzacji tranzystorów. Zatem pytamy, czy miniaturyzacja tranzystorów w układach scalonych zbliża się do swego kresu? Od lat zadawano sobie pytania o fizyczne i technologiczne ograniczenia miniaturyzacji tranzystorów. Wiele prognoz się nie sprawdziło. Kolejne bariery zostały sforsowane. Jednak tym razem to już koniec. Długość kanału tranzystora MOS, szerokości ścieżek i grubości warstw wynoszą już kilka nanometrów, a więc mieści się w nich kilkanaście atomów.

Kolejna redukcja wymiarów o rząd wielkości oznaczałaby, że w długości kanału, szerokości ścieżki lub grubości warstwy mieści się 1 atom, a to nie jest możliwe. Popatrzmy na **rysunek 2**, jaką drogę przeszła miniaturyzacja przez 50 lat, od roku 1970 do 2019. Wymiar charakterystyczny w pierwszych układach scalonych MOSLSI wynosił 10 μm , by osiągnąć 7 nm w roku 2019. Mówi się, że za rok-dwa możliwa jest jeszcze jedna redukcja rozmiarów do 3 nm, co w Si lub SiO_2 odpowiada dziesięciu



Rysunek 1. Prawo Moore'a ciągle obowiązuje

atomom. I na tym zapewne koniec. Czy to oznacza koniec historii rozwoju mikroelektroniki?

Nie sprawdziły się zapowiedzi końca historii rozwoju społeczno-politycznego (Hegel, Fukuyama) i pewnie błędne byłoby oczekiwanie, że za 2 lata mikroelektronika przestanie się rozwijać. Żeby jednak szukać odpowiedzi na pytanie o przyszłość, trzeba dobrze rozumieć przeszłość. Poświęćmy jej trochę miejsca. Zaczniemy od pytania, dlaczego miniaturyzacja jest utożsamiana z postępem. Po co jest miniaturyzacja? Początek tej historii sięga lat pięćdziesiątych, gdy zaczęły powstawać komputery montowane najpierw z lamp, a potem z tranzystorów. Wtedy pojawiło się pojęcie *tyrania liczb*. Montaż tysięcy tranzystorów był niezwykle pracochłonny i wystarczyło, by jeden z tranzystorów okazał się wadliwy, aby cała praca poszła na marne. Stąd zrodziła się idea, by wytwarzać układ złożony z wielu elementów w jednym procesie technologicznym. W roku 1958 Jack Kilby z Texas Instruments pokazał, że jest to możliwe, demonstrując pierwszy układ scalony wykonany w germanie (był to oscylator składający się z jednego tranzystora ze sprzężeniem zwrotnym RC – **fotografia 3**). Technologia Jacka Kilby'ego nie odegrała

żadnej istotnej roli, bo łączył on złotym drutem poszczególne elementy ulokowane na oddzielnych wypachach, co niewiele różniło się od montażu elementów dyskretnych.

Pół roku po Kilby'ym Robert Noyce w firmie Fairchild zademonstrował układ scalony w technologii planarnej z połączeniami elementów ścieżkami naparowanymi aluminium. Był to początek ery technologii planarnej stosowanej do dziś z różnymi modyfikacjami. Był to też początek drogi w redukowaniu kosztów.

Lp.	Nazwa	Liczba tranzystorów	Data wprowadzenia
1	Intel 4004 (4-bit, 16-pin)	2 250	1971
2	Intel 8008 (8-bit, 18-pin)	3 500	1972
3	Intel 8080 (8-bit, 40-pin)	6 000	1974
4	Zilog Z80 (8-bit, 4-bit ALU, 40-pin)	8 500	1976
5	Motorola 6809 (8-bit z niektórymi funkcjami 16-bit, 40-pin)	9 000	1978
6	Intel 8086 (16-bit, 40-pin)	29 000	1978
7	Intel 80186 (16-bit, 68-pin)	55 000	1982
8	Motorola 68000 (16/32-bit, rejestry 32-bit, 16-bit ALU)	68 000	1979
9	Intel 80286 (16-bit, 68-pin)	134 000	1982
10	Intel 80386 (32-bit, 132-pin; bez pamięci podręcznej)	275 000	1985
11	Intel 80486 (32-bit, 4 KB pamięci podręcznej)	1 180 235	1989
12	Pentium (32-bit, 16 KB pamięci podręcznej)	3 100 000	1993

Lp.	Nazwa	Liczba tranzystorów	Data wprowadzenia
13	AMD K5 (32-bit, pamięć podręczna)	4 300 000	1996
14	AMD K6 (32-bit, pamięć podręczna)	8 800 000	1997
15	Pentium III Katmai (32-bit, 128-bit SIMD, pamięć podręczna)	9 500 000	1999
16	AMD K6-III (32-bit, pamięć podręczna)	21 300 000	1999
17	Pentium 4 Willamette (32-bit, duża pamięć podręczna)	42 000 000	2000
18	Atom (32-bit, duża pamięć podręczna)	47 000 000	2008
19	Barton (32-bit, duża pamięć podręczna)	54 300 000	2003
20	AMD K8 (64-bit, duża pamięć podręczna)	105 900 000	2003
21	Itanium 2 McKinley (64-bit, duża pamięć podręczna)	220 000 000	2002
22	2-rdzeniowy Core 2 Duo Conroe (64-bit, duża pamięć podręczna)	291 000 000	2006
23	4-rdzeniowy AMD K10 2M L3 (64-bit, duża pamięć podręczna)	463 000 000	2007

Lp.	Nazwa	Liczba tranzystorów	Data wprowadzenia
24	Itanium 2 (64-bit, 9 MB pamięci podręcznej)	592 000 000	2004
25	4-rdzeniowy Core i7 (64-bit, SIMD, duża pamięć podręczna)	731 000 000	2008
26	6-rdzeniowy Xeon 2400 (64-bit, SIMD, duża pamięć podręczna)	904 000 000	2009
27	6-rdzeniowy Xeon 7400 (64-bit, SIMD, duża pamięć podręczna)	1 900 000 000	2008
28	8-rdzeniowy Xeon Nehalem-EX (64-bit, SIMD, duża pamięć podręczna)	2 300 000 000	2010
29	10-rdzeniowy Xeon Westmere-EX (64-bit, SIMD, duża pamięć podręczna)	2 600 000 000	2011
30	Tegra Xavier SoC (64/32-bit)	9 000 000 000	2018
31	32-rdzeniowy AMD Epyc (64-bit, SIMD, pamięć podręczna)	19 200 000 000	2017
32	AMD Epyc Rome (64-bit, SIMD, pamięć podręczna)	32 000 000 000	2019

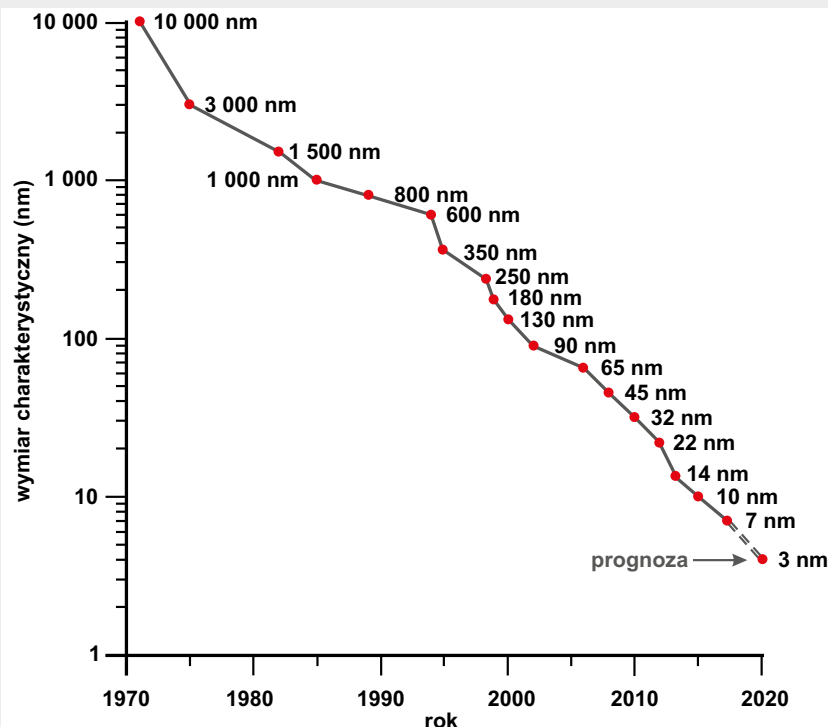
Tak jest – podstawowym czynnikiem napędowym miniaturyzacji jest **dążenie do redukcji kosztów**. Pierwszy rynkowy układ scalony wypuszczony przez Texas Instruments w 1960 roku (opracowany przez Jacka Kilby'ego multiwibrator na dwóch tranzystorach, dwóch diodach i ośmiu elementach biernych) kosztował 450 dolarów. Dzisiaj za 1 dolara można kupić układ scalony zawierający ponad miliard tranzystorów. Drugim czynnikiem napędowym miniaturyzacji jest **dążenie do zwiększania szybkości działania układów**. Im krótszy jest kanał tran-

zystora, tym mniejszy jest czas przelotu nośników od źródła do drenu, a więc większa częstotliwość działania układu. Zatem kierunek i cel rozwoju mikroelektroniki jest oczywisty – coraz mniejsze rozmiary tranzystorów to coraz niższy koszt produkcji liczonego na pojedynczy tranzystor i coraz większa szybkość działania układów scalających coraz większe liczby tranzystorów.

A co nam przeszkadza w realizacji tego celu? Przeshkód jest wiele, a ich działanie ograniczające rozwój ujawniało się z różną siłą na poszczególnych etapach rozwoju. Zmniejszanie rozmiarów od 10 μm do ok. 100 nm, czyli w latach 1970...2000, postępowało dość gładko w wyniku ulepszania procesów technologicznych, głównie litografii, przy niewielkich zmianach konstrukcji tranzystora poddawanego procedurze przeskalowania (scaling down). W kolejnych latach, przy zmniejszaniu rozmiarów od 100 nm do 10 nm, trzeba było pokonać kilka ograniczeń fizycznych.

Po pierwsze, wzrost szybkości działania układów scalonych zaczęły ograniczać obwody RC tworzone przez pojemności pasożytnicze i rezystancje kontaktów, a więc dalsze zmniejszanie długości kanału nie dawało oczekiwanego przyrostu szybkości działania.

Po drugie, w miarę zwiększania gęstości upakowania tranzystorów i liczby tranzystorów w układzie, rośnie ilość wydzielanego ciepła. Ma to też związek z pojemnościami pasożytniczymi, które zwiększają pobór prądu w stanie dynamicznym, przy przełączaniu inwertera CMOS. Aby zmniejszyć ilość wydzielanego ciepła, trzeba było sukcesywnie zmniejszać napięcie zasilania inwerterów CMOS. Było to też niezbędne dla uniknięcia przebicia skrośnego dren-źródło oraz prądu tunelowego źródło-dren dla zatkanego tranzystora. Dla długości kanału 10 nm stosuje



Rysunek 2. Jak się zmniejszał wymiar charakterystyczny w miarę rozwoju technologii

się napięcie $U_{DD}=0,8\text{ V}$, a dla kanału 5 nm $U_{DD}=0,65\text{ V}$. Tak niskie napięcie U_{DD} wymaga niezwykle powtarzalności i stabilności napięcia progowego U_T . Dla pokonania tych trudności trzeba było wprowadzić istotne modyfikacje w technologii planarnej.

Trzeba było wreszcie sięgnąć po trzeci wymiar. Tak powstała technologia FinFET, w której kanał tranzystora MOS ma kształt pionowej ściany pokrytej bramką, która oddziałuje polem z dwóch boków i z góry. Kanał przypomina swoim kształtem płetwę rekina wystającą nad powierzchnię wody, stąd nazwa FinFET. Z tak zmodyfikowaną

technologią planarną udało się „dojechać” do rozmiaru 7 nm i prawdopodobnie uda się jeszcze osiągnąć 3 nm, ale na tym będzie koniec rozwoju mikroelektroniki krzemowej. Czy na pewno? Otóż niezupełnie.

Pojawia się możliwość wytwarzania układów wielowarstwowych. Są już pamięci flash 3DNAND składające się z 96 warstw. Na rok 2021 planowane jest osiągnięcie 256 warstw. Samsung pracuje nad

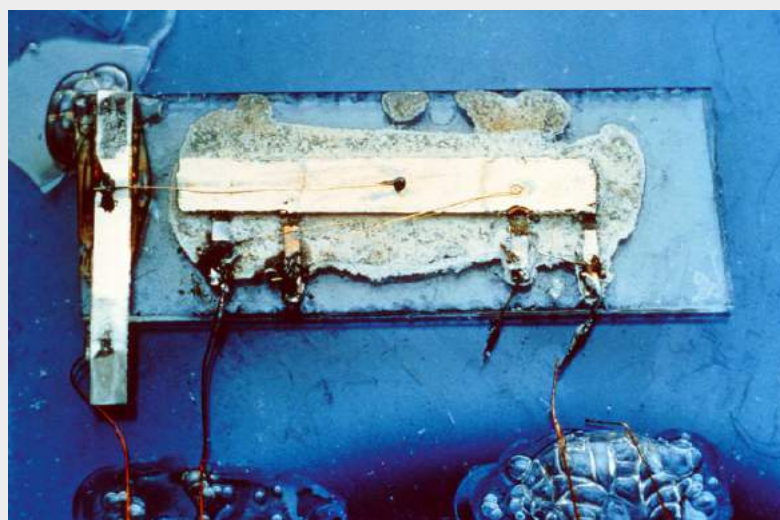
układami 4 Tbit (terabit, czyli 10^{12}) z 410 warstwami. Przewiduje się, że ta koncepcja pozwoli wytwarzać układy z 1024 warstwami. Jednak wygląda na to, że to ląbodzi śpiew technologii krzemowej. I co dalej? Od dość dawna pewne nadzieje wiąże się z wprowadzeniem do gry nowych (starych) materiałów, tj. Ge i GaAs.

Na horyzoncie widać też nowe idee fizyczne. Eksperymenty z zastąpieniem w układach scalonych połączeń miedzianych fotonową transmisją sygnałów mogą doprowadzić do pod-

wyższenia szybkości działania układów. Od wielu lat wielkie nadzieje łączy się z grafenem i jeszcze większe z tzw. komputerem kwantowym (quantum computing), czyli z nową elektroniką, nazywaną również spintroniką. Historia uczy, że od nowych idei do praktycznych zastosowań na dużą skalę, o ile wszystko idzie dobrze, trzeba ok. 30 lat. Te nowe idee świetnie się zapowiadają już od ok. 10 lat, a więc pozostało jeszcze ok. 20 lat. Przez najbliższe 20 lat będziemy więc mieli ciągle elektronikę krzemową opartą na transporcie ładunków. Czy będzie to okres stagnacji, gdy prawo Moore'a nie będzie już działać?

Aż nie chce się wierzyć.

Niestan Kurenski



Fotografia 3. Pierwszy układ scalony wykonany w germanie

Nie przeocz

Nowe podzespoły.....	8
Koktajl niusów.....	15
Arrow Multisolution Day 2019.....	19
Dodaj do obserwowanych.....	21
Niezbędnik elektronika.....	105

Projekty soft

Bezprzewodowa EKO-stacja do pomiaru jakości powietrza.....	24
Raspipool – system automatyki do przydomowego basenu.....	28
Sensor pracy mięśni z wizualizacją w sieci Web i na diodach LED RGB.....	32

Projekty

MIDISwitch – przełącznik MIDI do wzmacniacza lampowego.....	36
Sztuczne obciążenie w.c. o impedancji 50 Ohm.....	44
Przenośny rejestrator temperatury z zapisem na pendrive i oprogramowaniem do wizualizacji.....	48
Rysino – płytka ewaluacyjna z FPGA Intel MAX10.....	54

Moduły w aplikacjach

Miernik odległości na bazie czujnika GP2Y0A02YK0F.....	57
--	----

Miniprojekty

Uniwersalny układ czasowy 230 V.....	60
Regulowany bezpiecznik.....	61

Projekty czytelników

Półprzewodnikowa cewka Tesli o podwójnym rezonansie.....	65
---	----

Temat numeru: Systemy wbudowane

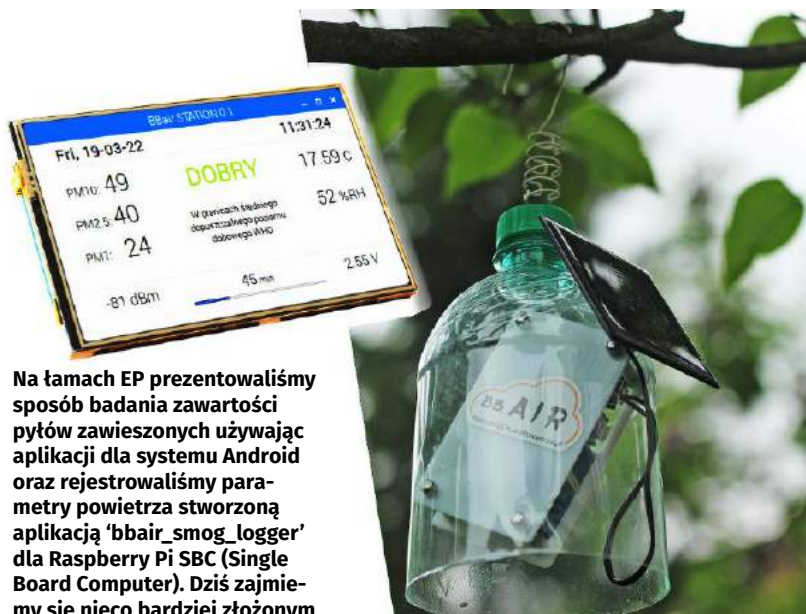
Systemy embedded. µCyfrowa ewolucja.....	72
--	----

Prezentacje

Nowe przekązniki firmy Relpol.....	84
Zasilacze LED do zadań specjalnych.....	106

Notatnik konstruktora

Dioda elektroluminescencyjna czy fotodioda? A może jedno i drugie...?.....	86
Sterowanie diodami RGB z Raspberry Pi.....	90
Symulacja i pomiar czyli LTspice i Analog Discovery 2 w rękach konstruktora (4). Pomiar pojemności mostkiem Wheatstone'a.....	98



Na łamach EP prezentowaliśmy sposób badania zawartości pyłów zawieszonych używając aplikacji dla systemu Android oraz rejestrowaliśmy parametry powietrza stworzoną aplikacją 'bbair_slog_logger' dla Raspberry Pi SBC (Single Board Computer). Dziś zajmemy się nieco bardziej złożonym tematem – budową stacji pomiarowej zasilanej energią słoneczną na bazie bezprzewodowego, laserowego sensora pyłów zawieszonych BBair z Bluetooth LE.

Podzespoły

AudioDSP zestaw z procesorem Sigma DSP ADAU1701.....	94
--	----

Wybór konstruktora:

Silniki krokowe i serwonapędy.....	108
WObit.....	117
OEM Automatic.....	118
Eldar.....	119

Kursy

Systemy dla Internetu Rzeczy (31). Nordic Thingy:91 – platforma do prototypowania dla mobilnego IoT.....	121
ISIXRTOS v3 mini system operacyjny dla mikrokontrolerów rodziny M0/M3/M4/M7 (8). Panel dotykowy.....	128
RISC-V – budujemy własny mikrokontroler (3).....	131

Od wydawcy.....	4
Konkurs Microchip.....	18
Prenumerata.....	120
Hity następnego numeru.....	138
Kramik i rynek.....	139



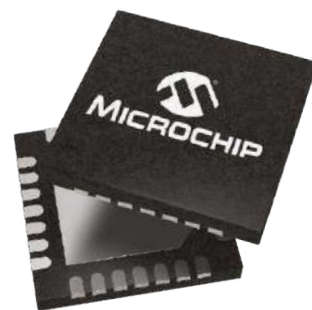
Wysokiej klasy układy analogowe od lidera rynku mikrokontrolerów?

Jeśli postrzegasz firmę Microchip tylko jako dostawcę MCU, mamy zamiar zmienić twoje myślenie!

Ocena pozycji firmy Microchip nie byłaby kompletna bez uwzględnienia naszych rozwiązań analogowych. Nasza historia jako wiodącego dostawcy rozwiązań zapewniającego wszechstronne wsparcie projektowe i szeroką gamę produktów nie obejmuje wyłącznie mikrokontrolerów.

Oferujemy również wydajne, łatwe w implementacji produkty analogowe, rozwiązania dla sygnałów mieszanych, zarządzania energią, a także układy interfejsowe I do zarządzania ciepłem. Razem szeroka oferta Microchip może być wykorzystywana w wielu aplikacjach o różnych wymaganiach w zakresie parametrów i wydajności.

Zapewniamy wydajność, elastyczność i pewność optymalnego wyboru rozwiązania dla swojego projektu, niezależnie od istniejących ograniczeń projektowych. Skorzystaj z naszego doświadczenia i kompletnych rozwiązań systemowych, aby zaoszczędzić czas i uprościć pracę nad projektem.



Zaplanuj własny sukces na
www.microchip.com/Real-Analog



nowe podzespóły

Z kilkuset nowości wybraliśmy te, których nie wolno przeoczyć. Bieżące nowości można śledzić na www.elektronikaB2B.pl



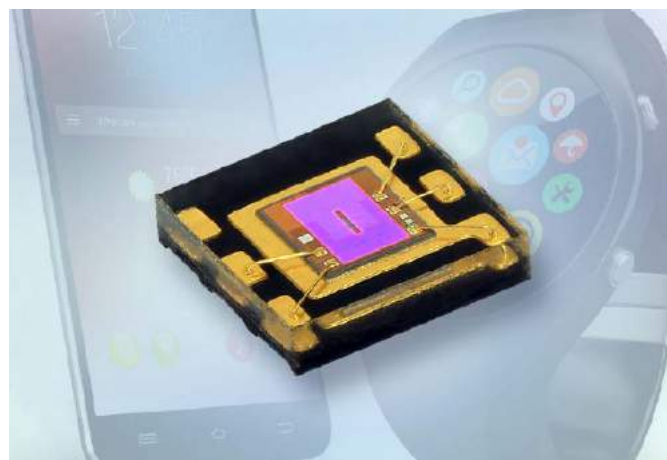
Dławiki SMD skompensowane prądowo o współczynniku tłumienia większym od 40 dB

Firma Coilcraft wprowadziła do oferty nową rodzinę dławików skompensowanych prądowo, przeznaczonych do tłumienia zaburzeń w.cz. na liniach zasilających. Dławiki rodziny Cx występują w 16 rozmiarach/konfiguracjach. Zapewniają tłumienie zaburzeń sumacyjnych w paśmie do 100 MHz oraz współczynnik tłumienia lepszy od 40 dB. Są zamykane w obudowach SMD o powierzchni od 13×13 mm z ekranowaniem magnetycznym. Mogą przewodzić prądy o natężeniu do 10 A rms. Izolacja między uzwojeniami wynosi w zależności od wersji 1000 lub 1500 V.

<http://bit.ly/2lr0so>

Czujnik światła o dużej czułości i wymiarach 2×2×0,4 mm

VEML6035 to nowy czujnik natężenia światła w ofercie firmy Vishay, wyróżniający się dużą czułością i małymi gabarytami (2×2×0,4 mm), pozwalającymi na zastosowania w smartfonach i innych urządzeniach



przeznaczonych. Funkcja przerywania generowanego po przekroczeniu ustawionej przez użytkownika jasności progowej zmniejsza obciążenie współpracującego mikroprocesora host. Układ komunikuje się za pomocą prostych komend I²C. Może pracować również w trybie przełącznika optycznego w urządzeniach konsumenckich i przemysłowych. Zastosowanie technologii Filtron zapewnia charakterystykę widmową zbliżoną do charakterystyki ludzkiego oka.

VEML6035 charakteryzuje się liniową charakterystyką w zakresie od 0,004 do 6710 lx i rozdzielczością od 0,0004 lx/ct, dzięki czemu może być umieszczany m.in. pod warstwą przyciemnianego szkła. Zapewnia tłumienie tętnień napięcia zasilającego (100/120 Hz) oraz kompensację temperaturową. Jest zasilany napięciem z zakresu od 1,7 do 3,6 V. Pobiera 170 A prądu w stanie aktywnym i 0,5 A w trybie shutdown.

<http://bit.ly/2kulw4r>



REKLAMA

BORNICO

Teraz większe MOŻLIWOŚCI

bornico.com.pl

- montaż kontraktowy elektroniki
- projektowanie urządzeń i systemów

Zakład Elektroniczny BORNICO

ul. Matczyńska 25
26-600 Radom
tel. +48 48 365 58 22
bornico@bornico.com.pl



Trójkolorowe wyświetlacze e-paper w wersjach o przekątnej 7,4" i 12,2"

Po wprowadzonych wcześniej na rynek wersjach małogabarytowych, do oferty firmy Pervasive Displays wchodzi dwa nowe wyświetlacze e-paper o większych przekątnych: 7,4" (ozn. E2741GS081) i 12,2"

(E2B98GS082). Są to wyświetlacze 3-barwne (żółty, biały, czarny), przeznaczone do zastosowań głównie w handlu i systemach Digital Signage. Ich przewagą nad tradycyjnymi wyświetlaczami jest bardzo mały pobór mocy. Nie wymagają one w ogóle zasilania w stanie aktywnym, a jedynie do zmiany wyświetlanego obrazu, dzięki czemu mogą pracować przez wiele miesięcy, a nawet lat na pojedynczej baterii. Do ich zalet należą również szerokie kąty widzenia oraz czytelny obraz w warunkach silnego oświetlenia zewnętrznego.

<http://bit.ly/31J1y1Y>



Pierwsze na rynku pamięci NOR Flash z wbudowanym adresem MAC

Microchip wprowadza do oferty pierwsze na rynku pamięci NOR Flash z wbudowanym adresem MAC (EUI-48 i EUI-64). 3-woltowe pamięci Serial Quad I/O (SQI) serii SST26VF, mogące znaleźć zastosowanie w aplikacjach komunikujących się w standardach Ethernet, Bluetooth, Wi-Fi, IEEE 802.15.4 i FireWire, są oferowane bez minimalnych wielkości zamówienia, pozwalając zaoszczędzić czas i koszty oraz skrócić czas wprowadzania produktów na rynek w przypadku urządzeń produkowanych w małych i średnich seriach. Występują w wersjach o pojemności 16, 32 i 64 Mb.

Szybki rozwój IoT powoduje, że projektanci systemów embedded muszą obecnie pobierać i programować adresy MAC od IEEE-Registration Authority (RA) do wszystkich produktów pracujących w sieci. Adresy są sprzedawane z IEEE-RA tylko w blokach od 4000 do 16 milionów, przez co małe i średnie firmy są zmuszone do płacenia większych kosztów za usługi. Pamięci SST26VF pozwalają wyeliminować konieczność samodzielnego zamawiania przez firmy i zarządzania tymi unikalnymi identyfikatorami z IEEE-RA.



Zaprogramowane adresy węzłów EUI-48 i EUI-64 składają się z unikalnego identyfikatora organizacji (OUI – Organisational Unique Identifier) i identyfikatora rozszerzenia (EI – Extension Identifier). Podczas gdy identyfikatory OUI są nabywane od IEEE przez poszczególne firmy, identyfikatory EI są przydzielane i zarządzane przez każdą firmę indywidualnie, tworząc unikatowy globalnie adres do każdego produktu.

Pamięci SST26VF064BEUI, SST26VF032BEUI i SST26VF016BEUI zawierają sześcioprzewodowy, 4-bitowy interfejs I/O, pozwalający na pracę z małym poborem mocy i dużą szybkością transmisji przy małej liczbie wyprowadzeń. Są produkowane w obudowach SOIC-8 i WSON-8. Oferują funkcję zabezpieczenia przed zapisem, eliminującą ryzyko przypadkowego nadpisania lub usunięcia kodu. Wykorzystują opracowaną przez Microchip technologię SuperFlash zapewniającą mały pobór prądu i dużą niezawodność. Ich zakres zastosowań obejmuje przemysłowe i domowe aplikacje wymagające łączności z chmurą obliczeniową, np. inteligentne urządzenia domowe, systemy automatyki budynków, automatykę przemysłową, instalacje alarmowe itp.

<http://bit.ly/2LKmsBN>

High-performance MEMS IMU for AR/VR and tracking applications



3-osiowy czujnik inercyjny MEMS o zakresie pomiaru prędkości obrotowej do 4000 dps

LSM6DSR to czujnik inercyjny wykonany w technologii MEMS, zawierający 3-osiowy żyroskop i 3-osiowy czujnik przyspieszenia. Został zaprojektowany do zastosowań w aplikacjach o dużej dynamice, m.in. sportowych, a także w robotyce, dronach i systemach monitorowania wibracji. Oferuje szeroki zakres pomiaru prędkości obrotowej żyroskopu do 4000 dps oraz zapewnia bardzo dobrą stabilność w funkcji temperatury i czasu. Jest kompatybilny z najpopularniejszymi systemami operacyjnymi dla urządzeń



REKLAMA



Obudowy odlewane: standardowe i malowane

Ponad 5000 różnych stylów obudowy:
hammfg.com/electronics/small-case

+44 1256 812812
sales@hammondmfg.eu



mobilnych i obsługuje standard synchronizacji S4S (Synchronization for Sensors). Oferuje funkcje detektora/licznika kroków, detektora ruchu i detektora nachylenia.

Pozostałe cechy:

- zakresy pomiarowe akcelerometru: $\pm 2/\pm 4/\pm 8/\pm 16$ g,
- zakresy pomiarowe żyroskopu: $\pm 125/\pm 250/\pm 500/\pm 1000/\pm 2000/\pm 4000$ dps,
- zakres napięć zasilania: 1,71...3,6 V,
- interfejs: szeregowy SPI/I²C i MIPI I3C,
- wbudowany czujnik temperatury,
- do 9 KB pamięci Smart FIFO.

LSM6DSR może pracować w przemysłowym zakresie temperatur od -40 do +85°C. Jest zamykany w obudowie LGA-14L o wymiarach 3,0×2,5×0,83 mm. Jego cena hurtowa wynosi 3,00 USD przy zamówieniach 1000 sztuk.

<http://bit.ly/2qGiv3B>



Najmniejszy na rynku konwerter DC-DC step-down 600 mA o wymiarach 0,96×0,88×0,33 mm

XC9281/82 to najmniejszy obecnie na rynku konwerter DC-DC step-down o wydajności prądowej 600 mA, zamykany w obudowach LGA-6B01 i WLP-6-05 (0,96×0,88×0,33 mm). Wraz z elementami współpracującymi (dwa kondensatory i cewka) całkowita powierzchnia konieczna do implementacji konwertera wynosi zaledwie 3,52 mm².

XC9281/82 akceptuje napięcia wejściowe z zakresu od 2,5 do 5,5 V, a jego napięcie wyjściowe jest ustalane fabrycznie i może wynosić od 0,7 do 3,6 V. Ze względu na małą długość połączeń zapewnia on małą emisję elektromagnetyczną. Dostępne są dwie wersje układu różniące się prądem spoczynkowym (590 µA dla XC9281, 11 µA dla XC9282) i trybem sterowania (odpowiednio PWM i PWM/PFM). Do standardowego wyposażenia należy układ miękkiego startu, ogranicznik prądu wyjściowego oraz zabezpieczenie zwarciove, podnapięciowe i termiczne.

<http://bit.ly/2PfOp1c>

Seria oscylatorów OCXO o zakresie częstotliwości wyjściowych od 200 MHz do 6 GHz

Firma MtronPTI wprowadziła na rynek serię oscylatorów OCXO XO9095 o dużej stabilności, małych szumach fazowych i małej wrażliwości na przyspieszenie (maks. 0,7 ppb/g). Są one dostępne w wersjach standardowych z wyjściem sinusoidalnym 1 GHz, 1,28 GHz,



2 GHz, 4 GHz i 6 GHz oraz w wersjach na zamówienie, których częstotliwość wyjściowa może wynosić od 200 MHz do 6 GHz. Mogą znaleźć zastosowanie w radarach, odbiornikach GPS, aparaturze pomiarowej, komunikacji satelitarnej i w systemach wojny elektronicznej. Zawierają wejście dostrajania częstotliwości sterowane napięciem 0...4,5 V, umożliwiające regulację w zakresie 2,5 ppm. Ich stabilność temperaturowa wynosi 100 ppb w zakresie -40...+85°C, a niestaość długoterminowa 0,5 ppm po upływie roku i 2,0 ppm po 20 latach.

Pozostałe parametry:

- szumy fazowe:
 - 76 dBc/Hz (offset 10 Hz),
 - 135 dBc/Hz (offset 1 kHz),
 - 154 dBc/Hz (offset 100 kHz),
- napięcie zasilania: 4,75...5,25 VDC,
- pobór mocy: maks. 2 W (maks. 3 W w fazie nagrzewania od -30°C),
- moc sygnału wyjściowego: +10 dBm @ 50,
- czas nagrzewania: typ. 3 min (maks. 5 min),
- wymiary obudowy: 36,5×27,5×17,5 mm.

<http://bit.ly/33Zn6fD>



Teseo-LIV3R ROM Module GNSS solution for accurate tracking



Tani moduł nawigacyjny GNSS o poborze prądu 15 µA w trybie standby

Do oferty firmy STMicroelectronics trafił nowy moduł nawigacyjny GNSS, oznaczony symbolem Teseo-LIV3R, zapewniający obsługę konstelacji satelitów GPS, Glonass, Beidou i QZSS oraz zgodność z systemami wspomagania S-BAS i RTCM. Dzięki łatwej implementacji, małym wymiarom i niskiej cenie może on znaleźć szeroki zakres zastosowań w systemach śledzenia obiektów, automatycznego naliczania opłat, antykradzieżowych, lokalizacji ludzi i zwierząt, zarządzania flotą pojazdów, powiadamiania alarmowego, a także w dronach i w transporcie publicznym. Zawiera wbudowaną funkcję licznika przebiegu z alarmem osiągniętego dystansu oraz funkcję GeoFence umożliwiającą wysyłanie alarmów, gdy urządzenie GPS wyszło poza określony obszar.

Teseo-LIV3R zapewnia dokładność pozycjonowania 1,5 m (50% CEP), czułość -163 dBm i czas TTFF nieprzekraczający 1 s. Jest modulem łatwym w obsłudze, obsługującym komendy NMEA. Pobiera zaledwie 15 µA prądu w trybie standby z podtrzymaniem zegara RTC oraz oferuje tryby pracy *continuous-fix*, *periodic-fix* i *fix-on-demand* o różnicowanym poborze mocy, co czyni go idealnym do zastosowań w urządzeniach baterijnych.

Teseo-LIV3R jest produkowany w obudowie LCC18 o wymiarach 10,1×9,7 mm. Jego ceny hurtowe zaczynają się od 7,3 USD przy zamówieniach 1000 sztuk.

<http://bit.ly/2qGiv3B>





Miniaturowe przetworniki LVDT do monitorowania parametrów lotu w dronach

Drony i pojazdy UAV wymagają zastosowania wielu czujników do monitorowania krytycznych parametrów decydujących o zapewnieniu stabilnego lotu. Miniaturowe czujniki przemieszczenia liniowego produkowane przez firmę NewTek Sensor Solutions realizują sprzężenie zwrotne przy monitorowaniu położenia stateczników i sterów oraz kąta nachylenia wirnika, zapewniając krótki czas odpowiedzi, niezbędny do szybkiego wprowadzania korekt zapewniających precyzyjną kontrolę lotu. Innym potencjalnym zastosowaniem jest kontrola kąta natarcia łopatek śmigła w maszynach jednowirnikowych oraz pomiar masy i siły inercji ładunku w dronach wielowirnikowych przeznaczonych do transportu przedmiotów.

Zarówno w pojazdach sterowanych przez człowieka, jak i UAV masa podzespołów jest ważnym parametrem, decydującym o poborze mocy i ładowności. Przetworniki przemieszczenia liniowego LVDT firmy NewTech charakteryzują się małymi wymiarami i masą, bardzo



dobłą dynamiką oraz odpornością na udary i wibracje do odpowiednio 1000 g (do 11 ms) i 20 g (do 2 kHz). Są zamykane w obudowach ze stali nierdzewnej, odpornych na narażenia środowiskowe. Obecnie do oferty NewTech wchodzi dwie nowe serie: M-375 o średnicy 3/8" i M-188 o średnicy 0,188". Są one produkowane na zakresy pomiarowe od 0,01" do 1,0" (0,254...25,4 mm). Charakteryzują się błędem nieliniowości już od 0,15% zakresu. Pracują w zakresie temperatur otoczenia od -54 do +135°C (do +204°C w wykonaniach specjalnych). Poza dronami i pojazdami UAV mogą też znaleźć zastosowanie w robotyce, medycynie, hydraulice i laboratoriach badawczych.

<http://bit.ly/2MFAEqO>



Sterowniki diod LED o małych wymiarach i dużym stopniu integracji

Dwa najnowsze sterowniki diod LED, MAX25610A i MAX25610B z oferty firmy Maxim, ułatwiają realizację systemów zasilania diod o wysokiej jasności w elektronice samochodowej, przemyśle



REKLAMA

COMPUTER CONTROLS

Autoryzowany dystrybutor Altium w Polsce



- Zarządzanie biblioteką komponentów i danymi projektowymi
- Aktualne i standaryzowane dane komponentów
- Możliwość śledzenia użycia komponentów w projektach
- Dostęp w czasie rzeczywistym do informacji od dostawców
- Dwukierunkowa wymiana danych ECAD-MCAD

Computer Controls Sp. z o.o.
Bielsko-Biała, ul. Budowlanych 1

tel.: +48 (33) 485 94 90

e-mail: info@ccontrols.pl
www.ccontrols.pl

i komercyjnych systemach oświetleniowych. Mogą one pracować w konfiguracji buck, buck-boost lub boost, wybieranej optymalnie do rodzaju zastosowanych diod i topologii obwodu zasilania. Zapewniają dużą sprawność energetyczną, sięgającą 90% w konfiguracji buck-boost wobec 85% uzyskiwanych w typowych sterownikach dostępnych obecnie na rynku. Ponadto, dzięki zastosowaniu modulacji spread spectrum, układy te spełniają wymogi specyfikacji CISPR 25 w zakresie poziomu generowanych zaburzeń elektromagnetycznych, co eliminuje problemy z uzupełnianiem obwodu o dodatkowe komponenty niezbędne do obniżenia poziomu emisji EMI do wartości dopuszczalnej.

Kolejną zaletą są małe wymiary obudów. MAX25610A i MAX25610B, zawierające wbudowane tranzystory MOSFET high-side/low-side i czujnik prądu wyjściowego, są zamykane w obudowach TQFN o wymiarach zaledwie 5×5 mm. Umożliwiają sterowanie do 8 diod LED wysokiej jasności bezpośrednio z akumulatora samochodowego. Ich duży stopień integracji pozwala do minimum obniżyć liczbę elementów współpracujących, a duża częstotliwość taktowania (2,2 MHz w przypadku wersji MAX25610B) pozwala dodatkowo ograniczyć ich wymiary.

Ceny hurtowe MAX25610A/B zaczynają się od 2,10 USD przy zamówieniach 1000 sztuk. Producent oferuje do tych układów zestaw ewaluacyjny MAX25610EVKIT# w cenie 100 USD.

Pozostałe cechy:

- zakres napięć wejściowych: od 5 do 36 V,
- maksymalny prąd wyjściowy: 2,85 A,
- dokładność regulacji prądu wyjściowego: 3% dla MAX25610A i 6% dla MAX25610B,
- częstotliwość taktowania: 400 kHz dla MAX25610A i 2,2 MHz dla MAX25610B,
- regulacja jasności: sygnał analogowy i PWM,
- zakres temperatur pracy: od -40°C do +125°C,
- zabezpieczenie: zwarciove, nadnapięciowe i termiczne,
- kwalifikacja AEC-Q100.

<http://bit.ly/2Pgfsj6>



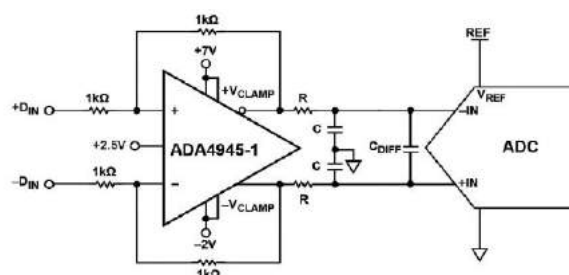
400-watowe konwertery DC-DC 1/8th Brick z wyjściem 12 VDC i interfejsem PMBus

Murata informuje o wprowadzeniu do oferty nowej serii 400-watowych, izolowanych konwerterów DC-DC formatu 1/8th Brick z pojedynczym wyjściem 12 VDC/33 A. Są to konwertery pracujące z napięciem wejściowym z zakresu od 36 do 75 V, których sprawność wynosi typowo 95,5%. Umożliwiają równoległe łączenie wyjść kilku modułów w aplikacjach wymagających większej mocy wyjściowej. Zapewniają izolację do 2250 VDC między wejściem i wyjściem. Zostały wyposażone w zabezpieczenie nadprądowe, nadnapięciowe i termiczne. Spełniają wymogi norm UL/EN/IEC 60950-1 i CAN/CSA-C22.2 w zakresie bezpieczeństwa użytkowania oraz EN55022/CISPR22 w zakresie kompatybilności EMC.

Konwertery serii DSE korzystają z zaawansowanej technologii konwersji mocy opartej na 32-bitowym mikroprocesorze ARM kontrolującym krytyczne funkcje konwersji oraz oferującym interfejs cyfrowy PMBus do wykorzystania przez aplikację systemową. Interfejs ten może być wykorzystany do programowania konfiguracji i monitorowania parametrów pracy. Dostępne są różne konfiguracje wyprowadzeń.

Konwertery serii DSE, zgodne z cyfrowym standardem DOSA 1/8th Brick, zawierają linie sense, trim i PMBus. Konwertery serii DAE, zgodne z analogowym standardem DOSA 1/8th Brick, zawierają linie sense i trim (bez PMBus). Konwertery serii DCE, zgodne z analogowym standardem DOSA 1/8th Brick, są produkowane w tradycyjnych obudowach 5-pinowych bez linii sense i trim oraz bez interfejsu PMBus.

<http://bit.ly/2MJM9NK>



Szybki i precyzyjny wzmacniacz różnicowy do sterowania wejściami przetworników A/C

ADA4945-1 to niskoszumowy wzmacniacz różnicowy o bardzo małych zniekształceniach, zaprojektowany do sterowania wejściami

REKLAMA

ZAJRZYSZ NA TE STRONY

All In One

- Projektowanie i wykonywanie
 - modeli karkasów i obudów na drukarce 3D
 - transformatorów i induktorów
 - prototypów PCB
- Modelowanie 3D modułów i urządzeń
- Projektowanie urządzeń zasilających



Feryster - producent elementów EMC

FERYSTER

www.feryster.pl

www.piekarz.pl
części elektroniczne
sprzedaz@piekarz.pl tel. 22 599 49 70

RENEX **RENEX** **RENEX**
www.renec.com.pl www.ipctraining.pl www.renecline.pl

GAMMA **www.gamma.pl**
info@gamma.pl **PODZESPOŁY ELEKTRONICZNE**

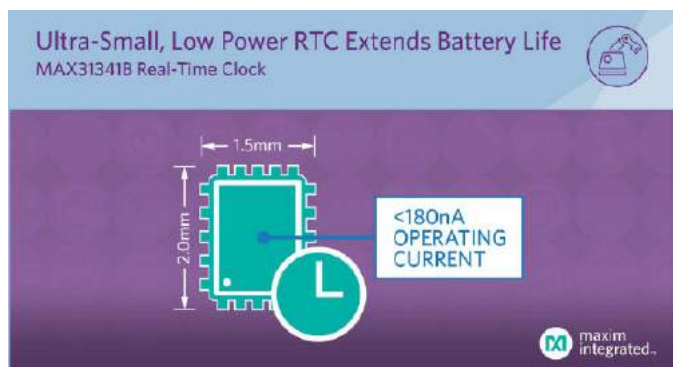
przetworników A/C. Wyróżnia się małym offsetem DC i małym dryftem przy bardzo dobrych parametrach dynamicznych. Może pracować z napięciem zasilania z szerokiego zakresu od 3 do 10 V. Został wyprodukowany w technologii SiGe, co pozwoliło osiągnąć bardzo małe zniekształcenia i wejściowe napięcie szumów wynoszące zaledwie 1,8 nV/√Hz. ADA4945-1 jest szczególnie polecany do sterowania wejściami przetworników A/C SAR i delta-sigma o wysokiej rozdzielczości, stosowanych w elektronice medycznej i układach sterowania procesami przemysłowymi. Regulowany zakres wejściowych napięć sumacyjnych pozwala na dopasowanie układu do zakresu napięć wejściowych konkretnego typu przetwornika A/C. Różnicowe wzmocnienie układu jest programowane czterema zewnętrznymi rezystorami. Dwa dostępne tryby pracy przełączane przez użytkownika: full power i low power o poborze prądu w stanie spoczynkowym odpowiednio 4 mA i 1,4 mA, pozwalają balansować pomiędzy poborem mocy i parametrami dynamicznymi, stosownie do wymagań konkretnego typu współpracującego przetwornika.

Ważniejsze parametry:

- zakres wejściowych napięć sumacyjnych: od VS– do VS+ – 1,3 V,
- tryby pracy:
 - full-power (prąd spoczynkowy 4 mA, pasmo 145 MHz),
 - low-power (prąd spoczynkowy 1,4 mA, pasmo 80 MHz),
- zawartość harmonicznnych (tryb full-power),
 - 133 dBc HD2 i –140 dBc HD3 @ 1 kHz,
 - 133 dBc HD2 i –116 dBc HD3 @ 100 kHz,
- czas stabilizacji napięcia wyjściowego (tryb full-power):
 - 18 bitów: 100 ns,
 - 16 bitów: 50 ns,
- wejściowe napięcie szumów (tryb full-power): 1,8 nV/√Hz @ f=100 kHz,
- napięcie offsetu: maks. ±115 μV w zakresie temperatur otoczenia od –40°C do +125°C,
- wyjście Rail-to-Rail.

ADA4945-1 jest zamykany w obudowie LFCSP-16 o powierzchni 3×3 mm. Pracuje w zakresie temperatur otoczenia od –40 do +125°C.

<http://bit.ly/2kFCHzZ>



Najmniejszy na rynku zegar RTC w obudowie o wymiarach 2,0×1,5 mm

Maxim powiększa ofertę zegarów RTC o najmniejszy tego typu układ dostępny obecnie na rynku. MAX31341B nanoPower jest zamykany w obudowie WLP o powierzchni 2,0×1,5 mm, mniejszej o ponad 35% od wcześniejszych odpowiedników. Pozwala obniżyć pobór mocy systemu, eliminując konieczność pozostawiania mikroprocesora w stanie aktywnym tylko w celu utrzymywania aktywności zegara. Pobiera prąd nieprzekraczający 180 nA, mniejszy o prawie 20% od najbliższego odpowiednika.

MAX31341B może współpracować z sygnałem referencyjnym GPS 1PPS oraz z szeroką gamą zewnętrznych rezonatorów kwarcowych o pojemności do 6 pF i rezystancji ESR do 100 kΩ. Komunikuje się



Klub Aplikantów Próbek



to inicjatywa redakcji „Elektroniki Praktycznej”. W kontaktach z firmami redakcja często otrzymuje do przetestowania próbki podzespołów, modułów, a nawet całych urządzeń elektronicznych. Są to zwykle najnowsze typy/modeli produktów na rynku. Z chęci podzielenia się z Czytelnikami tymi próbkami zrodziła się inicjatywa pod nazwą Klub Aplikantów Próbek. Członkiem KAP staje się każdy, kto zgłosi chęć przetestowania próbki. Wykaz i krótki opis próbek, którymi dysponuje redakcja EP, można znaleźć na stronie <http://bit.ly/339j0Sb>. Wystarczy wybrać rodzaj próbek i zwrócić się majllem na adres:

Szef Pracowni Konstrukcyjnej
grzegorz.becker@ep.com.pl

z prośbą o przestanie bezpłatnych próbek, podając ich nazwę i adres wysyłki. Warto dopisać jaki jest plan zastosowania tych próbek. Nie jest to konieczne, ale może mieć znaczenie przy podziale próbek w przypadku większej liczby zgłoszeń. Mile widziane, choć nieobowiązkowe, jest też przystanie do redakcji EP opisu wykonanej aplikacji próbek, oczywiście po jej wykonaniu z zastosowaniem otrzymanej próbki. Autorom przysłanych opisów przyznamy punkty, które będą im dawały pierwszeństwo przy ubieganiu się o kolejne próbki. Najciekawsze opisy aplikacji opublikujemy na forum ep.com.pl lub na łamach „Elektroniki Praktycznej”.

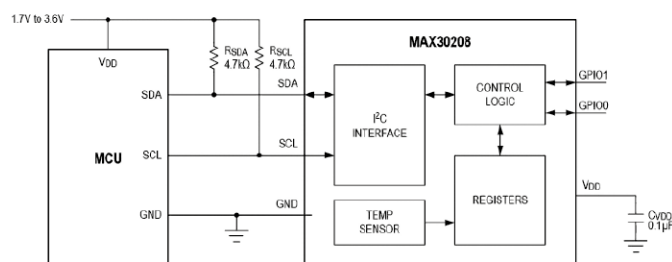
Dla pełnej jasności jeszcze raz podkreślamy, że próbki przekazujemy bezpłatnie i nie trzeba ich zwracać do redakcji. Z uwagi na ograniczoną liczbę dostępnych próbek i niemałe zainteresowanie nimi, prosimy o opisanie swojego pomysłu na projekt na naszym forum internetowym, w dziale poświęconym Klubowi Aplikantów Próbek <http://bit.ly/2qeN28e>. Ponadto, by zwiększyć swoje szanse na bycie wybranym do realizacji projektu w oparciu o nasze próbki, należy polubić fanpage Elektroniki Praktycznej na Facebooku (<http://bit.ly/2WygF09>) oraz udostępnić post, w którym opisujemy rozdawane próbki. W przypadku podobnie interesujących pomysłów na projekty, będziemy uwzględniać to jako dodatkowe kryterium wyboru.

www.ep.com.pl/kap

z mikroprocesorem za pośrednictwem interfejsu I²C. Zawiera programowalne wyjście cyfrowe do monitorowania zegara, 64 bajty pamięci RAM na dane użytkownika oraz wejście cyfrowe z bramką Schmitta i wejście analogowe o programowalnym napięciu progowym, służące do generowania przerwań (po zarejestrowaniu zbocza narastającego lub opadającego na wejściu cyfrowym lub przekroczenia poziomu progowego na wejściu analogowym). Wbudowana funkcja Power-on Reset inicjuje zawartość wewnętrznych rejestrów po włączeniu zasilania. Układ zawiera wyjście do doładowywania zewnętrznego superkondensatora lub akumulatora. Może pracować z napięciem zasilania od 1,6 do 3,6 V z pełną funkcjonalnością oraz >1,0 V do podtrzymywania zegara.

Cena hurtowa MAX31341B wynosi 0,70 USD przy zamówieniach 1000 sztuk. W ofercie firmy Maxim jest też dostępny zestaw ewaluacyjny MAX31341BEVKIT# w cenie 100 USD.

<http://bit.ly/2Pgfsj6>



Miniaturowy termometr cyfrowy o dokładności klinicznej 0,1°C w zakresie +30...+50°C

MAX30208 to miniaturowy termometr cyfrowy o dokładności klasy klinicznej (0,1°C w zakresie pomiarowym od +30 do +50°C), krótkim czasie odpowiedzi i dwukrotnie mniejszym poborze prądu od układów konkurencyjnych. Może znaleźć zastosowanie w smartwatchach i „inteligentnych” plastrach. Jest łatwiejszy w implementacji od czujników konkurencyjnych z uwagi na brak efektu samonagrzewania. Możliwość wyboru jednego z 4 dostępnych adresów I²C pozwala na współpracę 4 czujników podłączonych do wspólnej szyny. MAX30208 może być montowany zarówno na sztywnej, jak i elastycznej (FPC) płytce drukowanej. Charakteryzuje się dwukrotnie mniejszym poborem prądu (67 A vs. 135 A) od najbliższego odpowiednika. Jest produkowany w obudowie LGA-10 o wymiarach 2×2×0,75 mm. Cena hurtowa układu wynosi 1,25 USD przy zamówieniach 1000 sztuk. Dodatkowo producent oferuje zestaw ewaluacyjny MAX30208EVSYS# w cenie 56 USD.

Ważniejsze cechy:

- dokładność: ±0,15°C w zakresie 0...+70°C; ±0,1°C w podzakresie +30...+50°C,
- rozdzielczość: 16-bitowa (0,005°C),
- napięcie zasilania: 1,7...3,6 V,
- pobór prądu: 67 µA w stanie aktywnym; 0,5 µA w trybie Standby,
- czas pomiaru: 15 ms,
- alarmy przekroczenia temperatury (dolnej i górnej),
- pamięć FIFO (32 słowa),

- interfejs I²C z możliwością wyboru 4 różnych adresów,
- konfigurowalne wyjście przerwania,
- identyfikator chipa (ROM ID) do zadań kalibracyjnych.

<http://bit.ly/2Pgfsj6>



650-woltowe tranzystory SiC FET do pracy w układach hard-switching

Firma UnitedSiC dodaje do oferty dwa nowe 650-woltowe tranzystory SiC FET, stanowiące rozszerzenie serii UF3C FAST. Są to tranzystory przeznaczone do pracy w układach hard-switching, zamykane w obudowach TO220-3L. Mogą znaleźć szeroki zakres zastosowań w zasilaczach impulsowych, układach korekcji PFC, grzejnikach indukcyjnych, układach napędowych, ładowarkach akumulatorów i falownikach instalacji fotowoltaicznych. Różnią się rezystancją RDS(on) i dopuszczalnym prądem drenu, wynoszącymi 30 mΩ/85 A dla UF3C065030T3S oraz 80 mΩ/31 A dla UF3C065080T3S.



Oba tranzystory zrealizowano w konfiguracji kaskady z normalnie włączonym tranzystorem SiC JFET i krzemowym tranzystorem MOSFET, tworzącymi normalnie wyłączony tranzystor SiC FET. W istniejących projektach mogą one być stosowane jako zamienniki krzemowych tranzystorów IGBT, FET i superzłączowych oraz tranzystorów SiC MOSFET, pozwalające ograniczyć straty na przewodzenie i przełączanie oraz poprawić właściwości termiczne. Zawierają zabezpieczenie bramki przed wyładowaniami ESD.

W przypadku nowych projektów tranzystory UF3C FAST pozwalają zwiększyć częstotliwość taktowania, a w konsekwencji zwiększyć sprawność energetyczną oraz zmniejszyć wymiary i koszt zewnętrznych cewek i kondensatorów. Ponadto do zalet tranzystorów serii FAST należy bardzo mały ładunek bramki i łagodna charakterystyka regeneracji. Nadają się one doskonale do przełączania obciążeń o charakterze indukcyjnym, pod warunkiem zastosowania rekomendowanych filtrów RC.

Oferta tranzystorów SiC UF3C FAST firmy UnitedSiC obejmuje obecnie 14 wersji zamykanych w obudowach TO247-3L, TO247-4L, TO220-3L i D2PAK7-3L. Są one produkowane na napięcia znamionowe 650 i 1200 V. Ceny hurtowe UF3C065080T3S i UF3C065030T3S wynoszą odpowiednio 5,18 USD i 13,79 USD przy zamówieniach 1000 sztuk.

<http://bit.ly/32KJCSy>

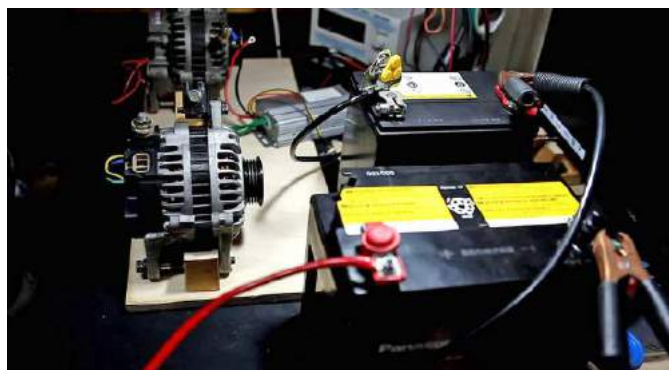
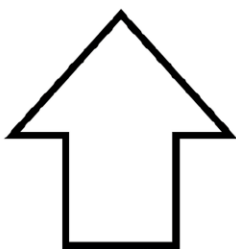
REKLAMA

www.ep.com.pl

koktajl niusów

Klawisz Shift – etymologia

Obecny w klawiaturach komputerowych klawisz Shift nie jest przypadkowy – wywodzi się z maszyny do pisania, w której służy do przesunięcia czcionki zawierającej jeden znak nad drugim, którym jest zwykle mała i duża litera. Jego słowem wciśnięcie powoduje zmianę położenia elementu w pionie, aby na papierze odbijała się druga, podczas gdy dla pierwszej jest to czynione bez jego użycia. Zmiana ta to przejście do górnej części czcionki, z ang. „shift up”, które dalej skrócono do dzisiejszego „shift” upowszechnionego dzięki komputerom, przez przeniesienie wspomnianego klawisza do ich klawiatur, aby zapewnić łatwą przesiadkę osobom dotychczas piszącym na maszynach. I choć nie ma już nic wspólnego z czcionkami w maszynie, to dalej pozwala na realizację pierwotnego zamysłu, nie mówiąc o użyciu go w kombinacjach klawiszy wywołujących określoną akcję w komputerach i jego wciśnięciu na dany czas, które również do niej prowadzi. Niemniej pamiątką po dawnych czasach jest jego oficjalny symbol, przedstawiony na rysunku, zdefiniowany normą ISO/IEC-9995-7-001, który pokazuje prawdziwe oblicze klawisza!



12 V – samochodowa historia prawdziwa

Skąd pomysł na napięcie zasilania 12 V w samochodach? Otóż z amerykańskiej firmy Ford, która wprowadziła je jako pierwsza do swoich samochodów w 1955 roku, w miejsce dotychczasowych 6 V. Stało się tak w odpowiedzi na rosnące zapotrzebowanie na moc wynikające z coraz prężniejszych silników i wzrastającej ilości udogodnień takich jak: klimatyzacja, ogrzewanie, regulowane szyby, elektryczny szyberdach itp. Czemu jednak napięcie, a nie prąd? Wystarczy skorzystać z prawa Ohma. Grubość kabli musi być większa dla prądów o większej wartości, a tym samym koszty, których nie chce ponosić żadna firma. Niemcy w XIX wieku wprowadziły swoje napięcie sieciowe 220 V, w miejscu dotąd stosowanych 110 V, aby także pozostać przy istniejącym okablowaniu. Przypadek (zwłaszcza z podwojeniem wartości)? Nie sądzę. Niemniej szczęściu trzeba dopomóc, w tym 12 V, którego sukces nie byłby możliwy bez taniej ropy naftowej i wynikającej stąd popularności motoryzacji, dzięki której wspomniana z początku firma sprzedawała bardzo dużą liczbę aut z tym napięciem, co sprawiło, że i pozostali gracze z branży je wdrożyli, aby nie wypaść z rynku.



I piorun jest odwrotny

Zazwyczaj piorun kojarzy się z wyładowaniem od chmury do ziemi bądź z wyładowaniem między chmurami lub w ich wnętrzu. Jednak to również wyładowanie od ziemi do chmury lub dokładnie: od obiektu o dużej wysokości, na szczycie którego jest obecne pole elektryczne o dużej wartości natężenia. Mówiąc inaczej, zdarza się, że to on odznacza się większym potencjałem niż chmura, w wyniku czego zmienia się kierunek przepływu ładunków. W efekcie jest to zjawisko takie jak na fotografii, choć jest rzadkie i zdarza się zwykle w przypadku wieżowców z iglicami lub masztów nadawczych. Niemniej, tak jak w zwykłych piorunach, częściej zdarza się przepływ ładunków ujemnych niż dodatnich, przez co i one to pioruny ujemne. Natomiast wysokość wynika z faktu, że łatwiej jest przebić izolator o mniejszej, aniżeli większej grubości (co dotyczy i powietrza), w myśl powiedzenia, że prąd zawsze idzie po linii najmniejszego oporu.



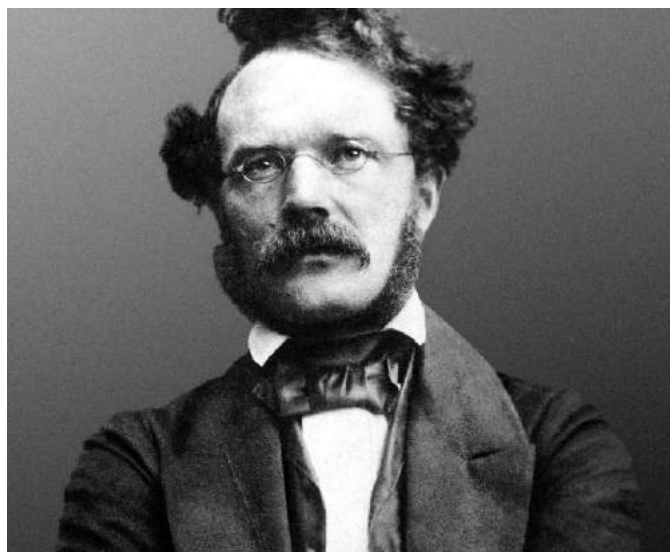
NIC, co komputera, nie jest „kartą sieciową”

Choć to tylko parafraza słów Terencjusza, to jednak i ona jest prawdziwa jak jego słowa. Jednak nie w polskim, a w angielskim znaczeniu, które stało się powodem do jej napisania. Dlaczego? Ponieważ NIC to nie tylko Network Interface Card – to również Network Interface Controller, które obejmuje i inne rozwiązania niż karty, takie jak np. pojedyncze układy scalone obsługujące Ethernet. Jest to zatem pojęcie ogólniejsze, w dodatku coraz bardziej domyślnie rozumiane pod wspomnianym skrótem niż to wcześniejsze. Jednak, lepsze to niż „nic” w naszym rozumieniu!



Więcej niż dwie karty SIM to...

No właśnie, co to może być? Nie, nie jest to lama ani tym bardziej *Familiada* i jej kolejne pytanie. Za to jest to telefon Triple SIM i Quad SIM, czyli urządzenie z trzema lub czterema kartami SIM, które pozwala na trzy, a nawet cztery rozmowy GSM jednocześnie! Jednak jaki jest tego sens, skoro szybciej się rozładuje bateria, a do tego wspomniane karty stają się już przeżytkiem na rzecz eSIM? Cóż – jak nie wiadomo, o co chodzi, to chodzi o pieniądze, a zwłaszcza o ich zarobienie na czymś czego nikt jeszcze dotąd nie zrobił. A to, że klient nie jest technicznie zorientowany i nie wie nawet o większym zużyciu baterii – nie szkodzi, gdyż liczy się po prostu sprzedaż, która nie dokona się, jeżeli zacznie mu się wyjaśniać, jak to działa, albo jeżeli już sam o tym wie. Niemniej nie chodzi tu o krytykę wolnego rynku, a o pokazanie, że często prawda nie leży w rozwiązaniach technicznych, a w ekonomii, która czasem skutkuje tak dziwnymi produktami jak przytoczone telefony, niczym gospodarka planowa PRL-u, która zaskakiwała ciągłymi brakami szczególnie w latach 80. XX wieku. Niezależnie od tego, przykłady takich telefonów to: Acer Liquid E700 Trio i Coolpad Mega, z trzema kartami SIM, oraz Green Berry W1 Plus Quad i Servo H8 z czterema, wszystkie raczej dostępne za granicą niż w Polsce, co jednak nie powinno dziwić, gdyż u nas są po prostu nieznanne (choć być może to się zmieni po tej ciekawostce).



Simens przewodnością – ale czy tylko?

Jak wiadomo, przewodność to wielkość wyrażana w siemensach [S] – jednostce nadanej na cześć niemieckiego konstruktora

elektrotechniki Wernera von Siemensa. Jednak nie od początku tak było – pierwotnie, w latach 1860–1881, stosowano ją do określania rezystancji (zanim [Ω] stał się jednostką oficjalną) i była zdefiniowana wzorcem fizycznym w postaci rtęci wypełniającej przewód o długości 1 metra, którego przekrój poprzeczny wynosi 1 mm², a temperatura na całej długości jest równa 0 stopni Celsjusza. Tym samym był to tzw. siemens rtęciowy (ang. Siemens mercury unit), równy 0,953 Ω, który niczym się nie różnił od masy, ponieważ tak jak ona, był z tym wzorcem porównywany. Było to kłopotliwe, ponieważ nie tylko nie używano w tym celu jednego wzorca, ale również były to wzorce nieidealne, szczególnie wrażliwe na temperaturę, której już niewielkie wahania powodowały istotną zmianę rezystancji rtęci. Nie wspominając o tym, że w XIX wieku nie istniały jeszcze metody uzyskiwania temperatury na bardzo stabilnym poziomie. Z tego względu szybko z jednostki zrezygnowano i przyjęto prostszą definicję rezystancji, nieangażującą temperatury, zgodną z tym, co odkrył Ohm w przewodnikach. Tym samym zaczęto ją określać omem na jego cześć, a na pocieszenie siemensa przydzielono przewodności. Co jednak stało się znacznie później (bowiem dopiero w 1971 roku) – do tego czasu wyrażano ją w tzw. moach, czyli słownej odwrotności omów, oznaczonych odwróconą dużą literą omega, tj. Ω, które zaproponował lord Kelvin – ten sam od temperatury absolutnej (może warto do nich powrócić, szczególnie że łatwo je zapamiętać na zasadzie skojarzenia z omami?)

Photon Energy

The energy of a photon is given by this equation:

$$E = h\nu$$

where $h = 6.6262 \times 10^{-34} \text{ J}\cdot\text{s}$
 ν = frequency (Hz)

Częstość, a częstotliwość – co lepsze?

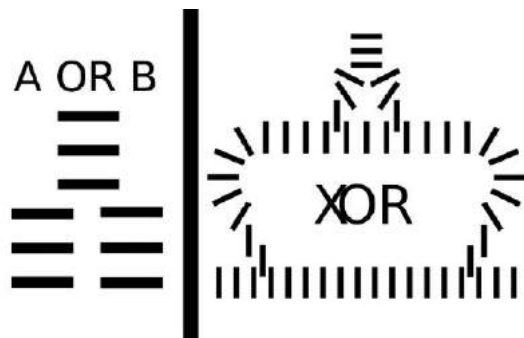
Spójrzmy na przedstawiony na rysunku wzór reprezentujący energię fotonu w funkcji częstości fali, po czym zrobmy to jeszcze raz, aby zobaczyć wzór wyrażający tę energię jako jej zależność od częstotliwości tej fali. Pytanie brzmi: gdzie jest błąd? Odpowiedź brzmi: w słowach na „c” przed falą, gdyż tylko jedno jest właściwe. Które? Otóż drugie. Wydaje się to błahostką, ale wbrew pozorom jest to ważne, ponieważ wystarczy spojrzeć, co oznacza słowo częsty, a co częstotliwy, z którego wywodzi się wspomniana częstotliwość. Zatem co one oznaczają? Jak się okazuje, pierwsze to wykonywany wiele razy, natomiast drugie to wykonywany raz po raz. Skoro tak, to lepsze jest drugie, gdyż zmiany fali są nie tyle wiele razy wykonywane, co jedno po drugim, czyli inaczej raz po raz, a tym samym należy podkreślić ten fakt, a nie pierwszy nieprecyzyjny – co bowiem znaczy wykonywanie czegoś wielokrotnie w fali? Niemniej, prawdą jest, że początkowo mówiono częstość zamiast częstotliwości, lecz było to w początkach XX wieku, kiedy jeszcze nie w pełni rozumiano naturę fal elektromagnetycznych, w tym nie wiadano, że jej parametry zmieniają się raz po raz. W związku z tym in. częj niż dzisiaj, wystarczało podejście matematyczne, traktujące częstotliwość jako parametr częstościowy. Słowem, skoro nie wiadano o następnych zmianach w fali, to nie było potrzeby utrudniać sobie życia i na siłę wskazywać tę kolejność, która w takim wypadku brzmi jak nieintuicyjny, nic niewnoszący twór abstrakcyjny (do czasu!). Patrząc przez pryzmat życia: co jest niepotrzebne to jest niestosowane, i na odwrót – co jest potrzebne, to znajduje użytek!

„Tylne” kolory komputera

Obserwując tył komputera stacjonarnego, można zauważyć, że jego gniazda są tam oznaczone konkretnymi kolorami. I nie, nie jest to pomyłka – jest to bowiem tzw. standard PC99 opracowany przez firmy

Color	Function	Connector on PC
User input		
Green	PS/2 mouse / pointing device	6-pin mini-DIN female
Purple	PS/2 keyboard	6-pin mini-DIN female
Gold	Game port / MIDI	15-pin D female
General input/output		
White	USB 1 low speed and full speed	USB Type A receptacle
Black	USB 2.0 high speed	USB Type A receptacle
Sky blue	USB 3.0 SuperSpeed	USB Type A SuperSpeed receptacle
Yellow	USB 3.0 SuperSpeed Always On even if the PC is turned off	USB Type A SuperSpeed receptacle
Red	High-power USB (more than 500 mA current available)	USB Type A receptacle
Grey	IEEE 1394 (FireWire)	6-pin FireWire 400 or 8-pin FireWire 800
Burgundy	Parallel port	24-pin D female
Teal or turquoise	Serial port	9-pin D male
Video		
Blue	Analog monitor	15-pin VGA female
White	Digital monitor	DVI female
Yellow	S-Video	4-pin mini-DIN
Yellow	Composite video	RCA jack
Black	Digital audio/video	HDMI or DisplayPort female
Audio		
Pink	Analog microphone audio input (mono or stereo)	3.5 mm TRS
Light blue	Analog line level audio input	3.5 mm TRS
Light green	Analog line level audio output, front stereo (speakers or headphones)	3.5 mm TRS
Black	Analog line level audio output, rear stereo (surround speakers)	3.5 mm TRS
Orange	Analog line level audio output, center and subwoofer (surround speakers)	3.5 mm TRS
Silver	Analog line level audio output, side stereo (surround speakers)	3.5 mm TRS
Orange	Digital audio output (coaxial S/PDIF)	RCA connector
Black	Digital audio output (optical S/PDIF)	FOXS/PAC

Microsoft i Intel w 1999 roku jako zbiór zaleceń dla producentów komputerów, mający na celu nie tylko sprawną identyfikację gniazd bez ich wcześniejszej znajomości, ale również łatwiejszą ich obsługę z poziomu programowania (w znaczeniu interfejsu graficznego). Jednak jest on obecnie stosowany tylko do pierwszego (i nie zanoszą się na to, aby jeszcze przez długi czas to się zmieniło!) i jest dany tabelą z rysunku. Co ciekawe, pierwszą polską firmą, która zaczęła go używać, był Optimus.



Domino jako logika cyfrowa – czy to możliwe?

Tak, z domina można zrobić logikę cyfrową! Nie wierzyć? To spojrzcie na rysunek, który przedstawia dwuwęściową bramkę OR (po lewej) i dwuwęściową bramkę XOR (po prawej)! Mało? To spojrzcie jeszcze w dokument dostępny pod adresem: <http://bit.ly/2oIqDjh>.

Krótko mówiąc, nie ma co wyrzucać nieużywanych klocków domina na śmietnik – zamiast tego lepiej zbudować z nich wyjątkowe w swym rodzaju układy cyfrowe, i to jeszcze z dziećmi, które mogą stać się w ten sposób elektronikami! I kto powiedział, że nie można się uczyć przez zabawę?

$$Q = Kc \cdot Kq \int \left(\frac{T_H - 20}{60} \right)^{1.33} dt$$

Podzielnikowy zawrót głowy (lecz niepotrzebnie)

Jesień za oknem, a co za tym idzie coraz niższe temperatury. Skoro tak, to trzeba się ogrzać! Jednak nie wszyscy to czynią, zwłaszcza w blokach mieszkalnych. Dlaczego? Ponieważ podzielniki ciepła, których wiele osób się obawia z powodu rachunków, które stają się rzekomo wyższe! Jednak to nieprawda – wystarczy bowiem zrozumieć istotę ich działania! Jaka ona jest? Otóż taka, że są to urządzenia zawierające dwa czujniki – jeden tylny, służący do pomiaru temperatury

powierzchni grzejnika, oraz drugi przedni, służący do pomiaru temperatury pomieszczenia. Oczywiście są takie, które zawierają tylko czujnik tylny – jednak są one obecnie rzadkością i zostaną tu pominięte. To samo z tzw. podzielnikami wyparkowymi, które z elektroniką nie są związane. Niemniej chodzi o to, że podzielnik zaczyna naliczać jednostki wtedy, gdy po pierwsze temperatura powierzchni grzejnika jest równa minimum 20...25 stopni (co podzielnik, to inna wartość graniczna), a po drugie, gdy różnica temperatury grzejnika i pomieszczenia wynosi co najmniej 3...5 stopni (również zależy od podzielnika). Ale uwaga – obowiązuje to przez cały rok, nieważne czy to jest sezon grzewczy, czy sezon letni. Inaczej mówiąc, jeżeli latem z jakichś przyczyn kaloryfer się nagrzeje do co najmniej przytroczonego minimum, to rozpocznie się naliczanie jednostek! Prócz tego im większa temperatura grzejnika, tym szybciej są naliczane jednostki w czasie przez podzielnik. Tempo naliczania jest przy tym różne – przykładem niech będzie podzielnik SonoHCA firmy Danfoss o parametrach takich jak na rysunku (wliczając w to przytoczone warunki aktywujące naliczanie jednostek), a zwłaszcza zaznaczony na czerwono wzór, który choć przeraża całą, to jednak w rzeczywistości jest zależnością liniową. Słowem, należy przenieść ułamek z całki przed symbol całki, a jego samego i dt należy zastąpić przez t , co w efekcie daje konkretny wzór pozwalający określić, ile jednostek jest zliczanych w ciągu sekundy. Dzięki temu już na oko można przewidzieć ile ich zostanie docelowo osiągniętych (źródło rysunku: <http://bit.ly/31eF9g3>). Na koniec rada: wszelkich informacji o podzielnikach szukaj w niepolskojęzycznych źródłach (tak jak to załączone), ponieważ firmy z nimi związane celowo nie podają wszystkich informacji, gdyż wiedzą, że dużo osób w Polsce kombinuje i/lub oszukuje, i chcą ten proceder po swojemu ukrócić.



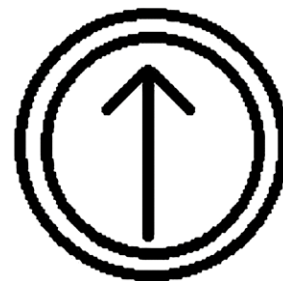
Scjentolodzy i ich „cudowny” przyrząd

Jeżeli dotąd nie słyszałeś o scjentologach, to pora się nimi zainteresować. Jednak nie ze względu na to, że to sekta, a ze względu na przedstawiony na fotografii przyrząd, który jest przez nich stosowany. O czym mowa? O tzw. e-metrze, który bada reakcję skórno galwaniczną, zwaną również reakcją elektrodermalną, czyli zmiany rezystancji skóry w czasie, w zależności od stopnia jej wilgotności. Krótko mówiąc, można nim sprawdzić to jak się mocno człowiek poci. Jednak nie w przypadku scjentologów, którzy używają tego przyrządu w czasie sesji odznaczających się psychomanipulacjami, zwanych audytami, mających na celu oczyszczenie osoby z negatywnych przeżyć poprzez wyparcie. W jaki sposób? Otóż w taki, że „badają” nim stan emocjonalny osoby, który uznają, że jest tym większy, im większe zachodzą zmiany na urządzeniu, oraz tym mniejszy im e-metr rzadziej reaguje na tę osobę. „Badanie” to jest czynione, aby ustalić, czy próba wyparcia przeżycia okazała się skuteczna – jeżeli tak, to sesja jest kończona, a jeżeli nie, to jest dalej kontynuowana. Inaczej mówiąc, według scjentologów brak pocenia się to brak reakcji na przeżycia. Jednak jest to brednia niedowiedzona naukowo, ponieważ

emocje obejmują całe ciało, a nie tylko skórę, i należy zbadać więcej rzeczy, w szczególności serce (EKG) i mózg (EEG), aby je obiektywnie opisać. Ponadto człowiek poci się zawsze, niezależnie od tego, czy coś w danej chwili przeżywa, czy nie. Niemniej nie ma co się temu dziwić – nie od dziś wiadomo, że wiele ruchów celowo odrzuca naukę, uważając ją za zło wcielone (patrz: płaskoziemcy). Gorzej, jeżeli jej zdobycze w tym im pomagają, tak jak elektronika w przypadku scjentologów! Niby nic, ale i w Polsce działa ten ruch, choć jest niewielki, nie wspominając o tym, ile osób ulega co roku manipulacjom, zbliżonych do tych z e-metrem – dość powiedzieć tu o tzw. cudownych urządzeniach leczniczych oferowanych w ramach naciąganych, a przy tym drogich pakietów medycznych, których działanie również nie jest poparte nauką, a nawet mogą przyczynić się do pogorszenia stanu zdrowia. Warto o tym pamiętać, w tym o przedstawionym przyrządzie!

Dioda jako źródło prądowe

To co jest przedstawione na rysunku, to nie tylko liniowe źródło prądowe – to również dioda. Jak to możliwe? Otóż nie jest powiedziane, że nie może to być element nieliniowy! Skoro tak, to nic nie stoi na przeszkodzie, aby stwierdzić, że jest to układ, którego prąd jest zależny od napięcia danego charakterystyką diody $I(U)$. Co więcej, jest to podstawa wszelkich modeli zastępczych diody, bez względu na to, czy mowa o jej modelu małosygnałowym, wielkosygnałowym, statycznym czy dynamicznym. W szczególności wiedza o tym studenci elektroniki niejednokrotnie zaliczający te schematy w ramach przeróżnych terminów (zasadniczych i poprawkowych).



REKLAMA

Wygraj zestaw Microchip z szybkim przetwornikiem A/C typu SAR



1 Msps SAR ADC Evaluation Kit (Part # ADM00873-BNDL)

Firma Microchip organizuje konkurs dla czytelników Elektroniki Praktycznej, w ramach którego można wygrać zestaw ewaluacyjny (model ADM00873-BNDL) z 16-bitowym przetwornikiem analogowo-cyfrowym MCP3313D-10, pracującym w trybie sukcesywnej aproksymacji z szybkością 1 Ms/s. W zestawie znalazła się płytka ewaluacyjna z 9-woltowym źródłem zasilania, wstępnie zaprogramowana płytka PIC32MZ EF Curiosity Board i kabel USB.

Przetworniki z rodziny MCP331x1(D)-xx zostały zaprojektowane by pracować w wysokich temperaturach oraz w środowisku o dużym natężeniu pola elektromagnetycznego. Są w stanie działać w tempie miliona próbek na sekundę, w pełni zgodnie z wymogami normy AEC-Q100, nawet z 16-bitową rozdzielczością. Dzięki temu dobrze spełniają wymogi aplikacji motoryzacyjnych i przemysłowych.

Aby wziąć udział w konkursie wystarczy zarejestrować się pod adresem <http://bit.ly/3341K0D>.



Arrow Multisolution Day 2019

Korporacja Arrow jest amerykańskim dostawcą elementów elektronicznych i usług powiązanych o zasięgu ogólnosiwiatowym. Według wielu rankingów zajmuje wśród dystrybutorów tego rodzaju towarów pierwsze (pod względem liczby klientów) lub drugie (pod względem sprzedaży) miejsce. Firma ta skupia w swojej ofercie produkty wielu wiodących na świecie marek.

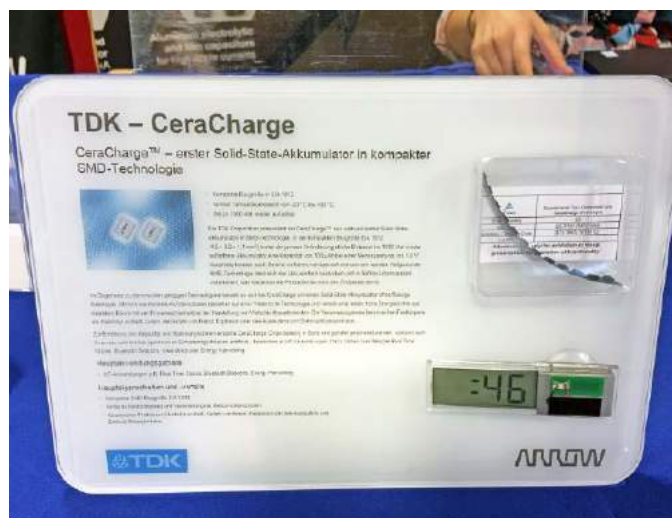
Na targach Arrow Multisolution Day, które odbyły się 1 października na Stadionie Narodowym w Warszawie, można było spotkać producentów elementów elektronicznych, których dystrybutorem jest firma Arrow. Wśród nich dało się znaleźć tak duże i rozpoznawalne



Fotografia 1. Moduł IoT z funkcją energy harvesting



Fotografia 2. Kondensator produkcji TDK o nietypowych parametrach



Fotografia 3. Ognia piezoelektryczne CeraCharge



Fotografia 4. Zestaw oświetlenia do uprawy roślin

firmy jak Analog Devices, Microchip czy TDK, oraz mniejszych dostawców o bardziej wyspecjalizowanej ofercie, takich jak Lantronix czy Fortinet. Wszystkie firmy przedstawiały swoje najnowsze układy i technologie, ale widocznych było kilka trendów, na które obecnie kładziony jest największy nacisk.

Hasłem, które występowało bardzo często, był Internet Rzeczy. Prezentowane były liczne moduły komunikacyjne wyposażone w takie interfejsy jak nowy **802.11ac** (i zapowiedzi **802.11ax**) oraz Bluetooth, w tym układy SoC (*system on a chip*) i SiP (*system in a package*) z wbudowanymi transceiverami BT. Najciekawszym układem tej grupy był proponowany przez ON Semiconductor SiP z wbudowanym modulem zbierania energii (**fotografia 1**). Takie rozwiązanie pozwala na budowę systemu zasilanego energią zbieraną z otoczenia.

Firma TDK, oprócz imponujących kondensatorów o pojemności 900 μF i napięciu 1100 V (**fotografia 2**), zaprezentowała także kompaktowe ognia piezoelektryczne CeraCharge. Są to pierwsze tego rodzaju akumulatory dostępne na rynku – przechowują one ładunek

w postaci naprężenia w strukturze piezoelektryka, z którego wykonany jest element. Ognio w obudowie SMD typu 1812 (**fotografia 3**) może mieć pojemność do 100 μAh przy napięciu pracy 1,4 V. Idealnie nadaje się do zasilania zegara czasu rzeczywistego czy transceivera Bluetooth o ultraniskim poborze prądu. Dzięki temu, że ognio nie zawiera ciekłego elektrolitu, może być bezpiecznie lutowane w piecu (reflow), usprawniając produkcję kompaktowych urządzeń elektronicznych.

Ciekawy pokaz przygotowała na swoim stanowisku firma OSRAM. Ten gigant z branży oświetleniowej zaprezentował zestaw energooszczędnych świateł RGB polecanych do uprawy roślin (**fotografia 4**). System pozwala na uprawę roślin w warunkach bez naturalnego oświetlenia. Technologia nie jest przeznaczona tylko dla futurystycznych upraw hydroponicznych na stacjach kosmicznych, ale także dla zwykłych wysokowydajnych upraw roślin na Ziemi czy też systemów doświetlania roślin za pomocą sztucznego światła w celu zwiększenia wydajności upraw.

REKLAMA

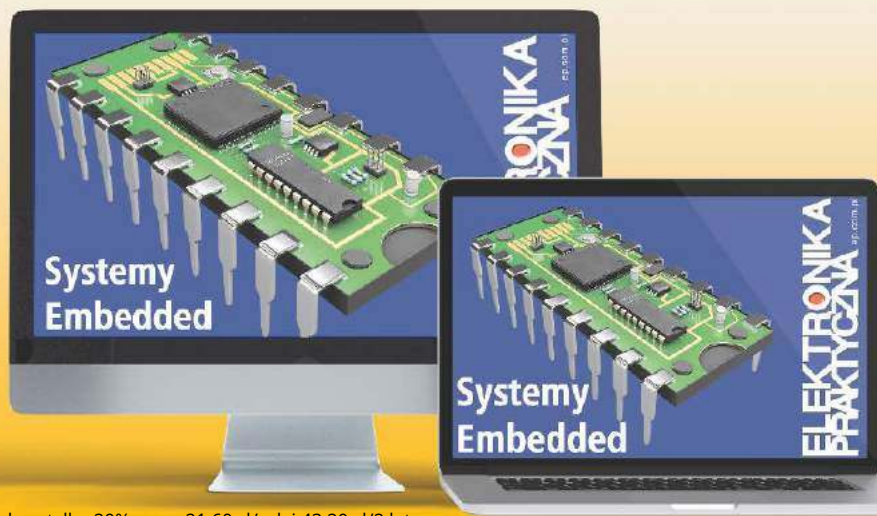
E-prenumerata to:

- najszybszy dostęp do nowego wydania magazynu
- wygodne archiwum na www.avt.pl
- hipertekstowy spis treści i wyszukiwarka
- wbudowane linki – klikasz i jesteś na odpowiedniej stronie WWW

Zamów e-prenumeratę (.pdf) na
www.avt.pl/prenumerata/elektroniczne

e-prenumerata roczna z rabatem
15% – 91,80 zł

e-prenumerata dwuletnia z rabatem
30% – 151,20 zł



Chcesz otrzymywać dodatek
Niezbędnik Elektronika?
Zamów prenumeratę
drukowaną na
www.avt.pl/prenumerata

dodaj do obserwowanych

Przedstawiamy redakcyjny wybór najciekawszych projektów spośród ostatnio anonsowanych w internecie. Są to projekty na różnych etapach realizacji. Warto się zapoznać z projektami zakończonymi i śledzić realizację projektów niegotowych, by czerpać z nich inspirację do własnych prac.



PyPPM: magnetometr protonowy DIY

Magnetometr Protonowy to urządzenie do pomiaru natężenia pola magnetycznego wykorzystujące zjawisko precesji protonów w polu magnetycznym. Jest to bardzo precyzyjny i stabilny sposób pomiaru pola magnetycznego dzięki temu, że częstotliwość precesji jest tylko liniowo związana z polem magnetycznym (związek ten opisany jest stałymi atomowymi).

Metoda pomiaru realizowana magnetometrem tego rodzaju jest bardzo prosta – prąd stały przepływający w cewce wytwarza silne pole magnetyczne wokół płynu bogatego w wodór (stosuje się naftę lub dekan, ale może to być nawet woda), powodując, że niektóre protony dopasowują się do tego pola. Następnie prąd zostaje wyłączony, a gdy protony te relaksują, dopasowując się do otaczającego je pola magnetycznego, występuje ich precesja z częstotliwością wprost proporcjonalną do pola magnetycznego. Powoduje to wytwarzanie słabego pola



magnetycznego, które jest wychwytywane przez cewkę, wzmacniane elektronicznie i podawane do częstotliciemierza.

„Kiedy ostatni raz obserwowałeś coś czysto kwantowo-mechanicznego, bez żadnego klasycznego analogu?”, pyta retorycznie autor konstrukcji, Bradley Worley. PyPPM to jednopłytkowy magnetometr protonowy, który pozwala na przeprowadzanie eksperymentów jądrowego rezonansu magnetycznego (NMR) z wykorzystaniem naturalnego pola magnetycznego Ziemi, bez potrzeby stosowania drogich magnesów nadprzewodzących.

Jest to pierwszy układ tego rodzaju zaprezentowany jako open source. Do zestawienia tego systemu nie jest wymagany żaden dodatkowy sprzęt etc. System ten łączy się bezpośrednio z komputerem PC z systemem Linux lub Mac przez USB i można nim sterować za pomocą prostego API dla języka Python. Na stronie projektu autor udostępnia kompletny opis całego systemu, a także poradnik, w jaki sposób wykonać uzwojenia obu cewek, niezbędnych do działania magnetometru.

<http://bit.ly/32A07aC>



Zynthian: otwarta platforma dla cyfrowego syntezyzatora audio

Zynthian to nowa klasa maszyny do tworzenia muzyki, rodzaj szwajcarskiego szczyrzyka do syntezy dźwięku. Jest wyposażony w wiele modułów, filtrów i efektów, jest całkowicie konfigurowalny i bez problemu można go aktualizować, dodając nowe funkcje i moduły. System bazuje na module



Raspberry Pi z systemem operacyjnym Linux. Zynthian to w pełni otwarta platforma do syntezy dźwięku. Cała specyfikacja sprzętu jest publiczna, a oprogramowanie jest w całości open source. Syntezator zaprojektowano jako urządzenie do samodzielnego wykonania.

Aby stać się posiadaczem Zynthiana, można samodzielnie zbudować urządzenie na podstawie udostępnionej dokumentacji lub wybrać jeden ze sprzedawanych przez twórców systemu zestawów, dostosowanych do różnych poziomów umiejętności.

<http://bit.ly/2N6nD8C>



Dzięki zaawansowanym algorytmom sterowania i mocnym silnikom, sterowanym równie mocnym driverem, robot jest bardzo dynamiczny. Wydolność prądowa drivera wynosi do 50 A na fazę, silniki mają moment do 40 Nm przy masie ok. 400 g. **MJBOT** może nie tylko chodzić, ale także biegać i skakać. Warto zajrzeć na stronę projektu, aby zobaczyć zamieszczone tam filmy, pokazujące, jak się porusza.

<http://bit.ly/2MD3p7z>



Wyświetlacz HUD DIY do naszego samochodu

Wyświetlacz HUD to specjalny, przezierny wyświetlacz, który pozwala na wygodną prezentację danych dla kierowcy. Zaletą tego rodzaju prezentacji danych, jest fakt, że wyświetlacz nie wymaga odrywania wzroku kierowcy od drogi, gdyż wyświetlacz znajduje się na linii jego wzroku. Przedstawiony projekt to właśnie taki wyświetlacz.

Prezentowana konstrukcja ma półprzepuszczalny ekran, który odbija część światła z ekranu telefonu komórkowego, który znajduje się poniżej. Jedyne, co trzeba zrobić, to zapewnić lustrzane odbicie treści na wyświetlaczu urządzenia, aby odbity od ekranu obraz był już prosty. Nie jest to takie łatwe, jak może się początkowo wydawać. Kilka aplikacji do nawigacji samochodowej na smartfony z systemem Android ma taką możliwość, ale nie da się zastosować tego do wszystkich aplikacji.

Autor przedstawionej konstrukcji, występujący pod pseudonimem thiagesh D, opracował prosty uchwyt ekranu wykonany w technologii druku 3D. Dzięki temu każdy może wykonać podobny ekran do wyświetlacza HUD. Dodatkowo, w ramach projektu, autor opracował plik ROM z systemem operacyjnym Android do odblokowanych smartfonów. Zmodyfikowany system nie różni się niczym od oryginalnego, oprócz tego, że ma możliwość lustrzanego odbicia całego ekranu, tak aby poprawnie prezentował się na wyświetlaczu HUD.

<http://bit.ly/2W3YQpX>



MJBOT: otwarty i niedrogi robot na czterech nogach

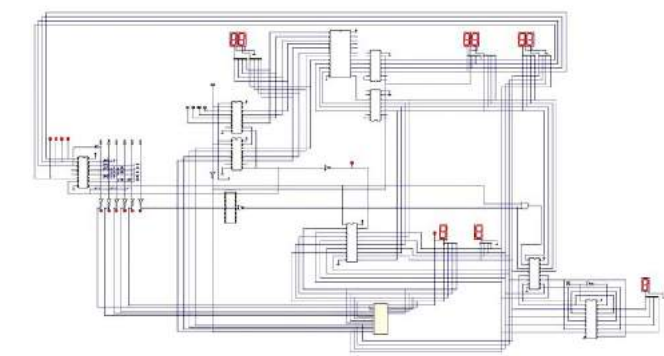
MJBOT quad A0 to niewielki, dynamiczny robot, poruszający się na czterech kończynach. Konstrukcja została opracowana przez Josha Piepera, który prezentuje ją na portalu Hackaday.io. Robot ma cztery, niezależnie sterowane nogi, podobnie jak wiele innych konstrukcji prezentowanych m.in. przez MIT czy Boston Dynamics. Konstrukcja pokazana tutaj wykonana została w technologii druku 3D, napędzana jest silnikami bezszczotkowymi Moteus M80, które sterowane są przez kontroler Moteus w wersji 3.1.



Konkurs multidyscyplinarny na Instructables

Portal Instructables otworzył kolejny ze swoich konkursów. Jego tematem jest... multidyscyplinarnizm. Jedynym wymogiem, jaki stawiają organizatorzy, jest to, by wykonany przedmiot, urządzenie czy instalacja powstały z zastosowaniem co najmniej dwóch kreatywnych dyscyplin. Co to oznacza w praktyce? Połączenie nawet dalekich od siebie umiejętności w jednym projekcie może zaowocować wygraniem głównej nagrody – bonu na 500 dolarów do Amazona. Dodatkowo, oferowana jest nagroda-tajemnica od jury za najbardziej kreatywne i niespotykane połączenie dziedzin, więc jeśli jesteś w stanie połączyć swoje umiejętności spawacza i hobby w postaci robótek na szydełku lub jeżeli lubisz piec ciasta i masz pomysł, jak przy tym użyć swoich umiejętności z zakresu elektroniki, ta nagroda może być właśnie dla Ciebie.

<http://bit.ly/31A7wFK>



16-bitowy procesor TTL – Suite-16

Suite-16 jest eksperymentalnym 16-bitowym procesorem TTL zaprojektowanym głównie w celach edukacyjnych. Jak pisze autor projektu, podpisujący się na Hackaday.io monsonite, konstrukcja powstała do „eksplorowania architektury procesorów oraz



badania interakcji między sprzętem, oprogramowaniem układowym i software, potrzebnym do stworzenia działającego komputera”.

Nazwa procesora Suite-16 to gra słów, nawiązuje do 16-bitowego procesora wirtualnego, stworzonego przez Steve'a Wozniaka – Sweet-16. Był to napisany w języku asemblera 6502 wirtualny, 16-bitowy procesor dla Apple II, który miał działać jako akcelerator do wspomagania komputera podczas wykonywania operacji 16-bitowych.

„Suite-16 powstał z ciekawości, jak działają komputery”, pisze autor. „Mam nadzieję, że poszerzę moje horyzonty, nauczę się nowych umiejętności i osiągnę cel – zbuduję działający system komputerowy TTL”.

Głównym rezultatem projektu ma być 16-bitowy system komputerowy zaprojektowany i zbudowany z układów TTL serii 7400. Z punktu widzenia sprzętowego obejmie to schematy obwodów, rysunki płytek drukowanych, diagramy czasowe i programy dla stanowisk testowych. Prawdopodobnie w projekcie powstanie również prototyp zaimplementowany w FPGA wraz z konstrukcjami pomocniczymi napisanymi w Verilogu.

Podsumowując projekt Suite-16, można stwierdzić, że będzie również zapisem eksploracji przestrzeni wiedzy związanej z elektroniką cyfrową, oprogramowaniem i projektowaniem FPGA. Warto śledzić postępy prac.

<http://bit.ly/35XRpoH>



System rzeczywistości rozszerzonej uGlass

uGlass to moduł rzeczywistości rozszerzonej (AR), który teoretycznie można przymocować do zwykłych okularów i wyświetlać obrazy bezpośrednio na nich. Prototyp wykonano z zastosowaniem szkiełek mikroskopowych i kleju na gorąco, a sam wyświetlacz to mały ekran OLED, ale zespół Ultimate Robotics



pracuje nad rozwojem tego projektu. „Większość dostępnych na rynku okularów AR jest nieporęczna, przeprojektowana i stanowi zamknięty system. Są zwykłymi okularami z elektroniką, dostępnymi tylko w kilku rozmiarach – nie pasują wszystkim”, tłumaczy zespół projektowy. „Ale co, jeśli byłby inny sposób?”, pytają retorycznie. Oczywiście innym sposobem jest uGlass.

Początkowo był tylko pomysł na urządzenie, które podłączasz do okularów i które wyświetla na nich obrazy „natychmiastowe AR”. Pierwsza wersja zbudowana była w oparciu na Arduino, komercyjnym wyświetlaczu OLED i elementach optycznych z recyklingu. Druga wersja ma o wiele większe możliwości. Może m.in. współpracować z urządzeniami opartymi na DWM1001, aby prezentować np. odległość między nimi lub pokazać ustawione punkty trasy.

Płytką jest bardzo prosta, zbudowana wokół mikrokontrolera nRF52832 i IMU BMI160 z ekranem OLED. Moduł wykorzystuje Bluetooth do komunikacji z innymi urządzeniami, przetwarza dane i pokazuje je w wygodnej formie. Optyka jest na razie na stałe przymocowana do okularów, ale w przyszłości autorzy chcą zaprojektować oprawkę z druku 3D i odpowiednią obudowę.

<http://bit.ly/2BChonU>



Kurs dotyczących dużych silników elektrycznych

Portal Instructables to nie tylko autorskie projekty twórców z całego świata. Znaleźć można tam także proste kursy, dotyczące m.in. elektroniki. Jednym z ciekawszych na tym portalu kursów jest ten poświęcony silnikom elektrycznym.



W ramach tego kursu można się dowiedzieć, jak sterować silnikami elektrycznymi odpowiednimi do budowy gokartów, skuterów elektrycznych, dużych robotów i wielu innych rzeczy. Wymaga to wyjątkowego zestawu wiedzy i umiejętności w porównaniu z mniejszymi silnikami elektrycznymi. Kurs jest przeznaczony dla osób z pewnym podstawowym doświadczeniem w pracy z silnikami i układami mechanicznymi. Można dowiedzieć się, jak wybrać odpowiedni zasilacz, zbudować obwód sterujący i stabilizować prędkość silnika



czy zmieniać kierunek jego obrotów. Sterowanie silnikiem jest demonstrowane za pomocą szeregu technik, takich jak stosowanie sterowników silników z analogowymi przepustnikami, a także zintegrowanych kontrolerów silników do Arduino. Warto prześledzić ten kurs.

<http://bit.ly/2pLd9Dy>



Na łamach EP prezentowaliśmy sposób badania zawartości pyłów zawieszonych przy użyciu aplikacji dla systemu Android oraz rejestrowaliśmy parametry powietrza stworzoną aplikacją 'bbair_smog_logger' dla Raspberry Pi SBC (Single Board Computer). Dziś zajmiemy się nieco bardziej złożonym tematem – budową stacji pomiarowej zasilanej energią słoneczną na bazie bezprzewodowego, laserowego sensora pyłów zawieszonych BBair z Bluetooth LE.



Bezprzewodowa EKO-stacja do pomiaru jakości powietrza



Idea użytkowa jest nieskomplikowana: powiesić na ścianie, postawić na biurku czy na półce taki mały ekranik. Wystarczy jedno spojrzenie, aby wiedzieć, co dzieje się na zewnątrz: jaki jest poziom stężenia pyłów zawieszonych poszczególnych frakcji czy jest ciepło, czy może zimno i czy powietrze jest przenikliwe wilgotne, czy może przyjemnie suche (**rysunek 1**). Gdy jednak czarodziejski ekranik jest poza zasięgiem wzroku (bo np. leżąc wygodnie na kanapie w sąsiednim pokoju, czytamy najnowszą EP ☺), równie łatwo można wszystko to sprawdzić za pomocą telefonu z aplikacją BBair. Szybkie połączenie z EKO-stacją pozwoli zobaczyć wskazania. Ten projekt to proste w użyciu i użyteczne narzędzie integrujące termometr, higrometr, zegar, kalendarz i czujnik jakości powietrza, tak bardzo pomocne w codziennym życiu.

Wartość niezależnych, lokalnych pomiarów

Istnieje wiele systemów pomiarów jakości powietrza. Dociekliwi zapytają: po co zatem budować kolejny – własny system? Na tak postawione pytanie można udzielić wyczerpującej dwuwymiarowej

odpowiedzi. Po pierwsze wartość edukacyjna i niesamowita satysfakcja z posiadania własnoręcznie zbudowanej stacji pomiarowej są nieocenione. Po drugie i dla wielu z pewnością dużo istotniejsze okażą się kwestie lokalności i niezależności pomiarów wykonywanych na żądanie. Posiadając własny system pomiarowy, mamy zapewnione:

- Niezależność i wiarygodność danych pomiarowych, które nie są w żaden sposób przetwarzane;
- Wykonywanie pomiarów na żądanie, czyli w dowolnym momencie, w którym chcemy poznać stan zaokienego powietrza;
- Lokalność, ponieważ po wykonaniu pomiaru za pomocą własnej zaokiennej stacji mamy pewność, że wskazanie dotyczy stanu powietrza z najbliższego otoczenia. Nie jest to szacunkowa, uśredniona wartość z kilku najbliższych, często oddalonych o kilometry stacji;
- Możliwość obserwowania zmian zanieczyszczenia powietrza w najbliższej okolicy, a dzięki temu wyszukiwania i śledzenia źródeł zanieczyszczeń.



Rysunek 1. Wyniki pomiarów zaprezentowane w atrakcyjnej formie na czterocalowym, graficznym wyświetlaczu LCD

Budowa

EKO-stacja składa się z dwóch modułów: sensora montowanego na zewnątrz i „prezentera” instalowanego wewnątrz. Główne składniki potrzebne do realizacji projektu to:

1. Mikrokomputer Raspberry Pi z chipem Bluetooth Smart. W zupełności wystarczy ekonomiczna wersja: Raspberry Pi zero W.
2. Dotykowy, czterocalowy ekran LCD z interfejsem SPI. (<http://bit.ly/35A3tMT>).
3. Dokładny, laserowy sensor parametrów powietrza z interfejsem Bluetooth Smart – BBair (<http://bit.ly/35uyxgN>).
4. Małe ogniwo słoneczne. W prototypie użyto panelu o wymiarach 6,5×6,5 cm, napięciu jałowym 5,5 V i maksymalnej mocy 0,6 W.

Konfiguracja Raspberry Pi

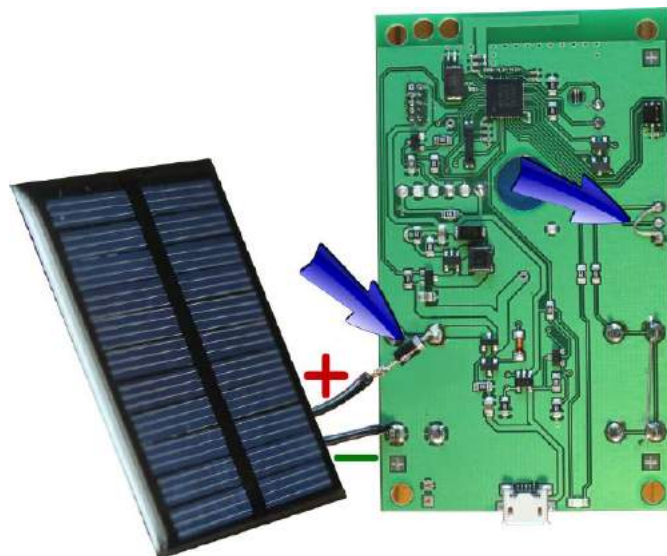
Po podłączeniu monitora, klawiatury i zasilacza do Raspberry Pi SBC jesteśmy gotowi do instalacji systemu operacyjnego. Nie zapomnijmy również o podłączeniu ekranu LCD. W Internecie dostępnych jest wiele instrukcji, jak zainstalować system operacyjny, dlatego nie będziemy powielać tych informacji. Założmy, że wychodzimy z punktu, gdzie na naszej malinice mamy już zainstalowany (np. za pomocą pliku ‘NOOBS_lite_v3_0’) świeży system Raspbian Lite. Jeśli nie uruchamia się w trybie graficznym, zmieńmy to, odpalając z konsoli malinowy konfigurator poleceniem:

```
sudo raspi-config
```

Gdy na ekranie pojawi się menu, przechodzimy do punktu 3.*Boot Options*, dalej do B1.*Desktop/CLI*, a następnie wybieramy B4.*Desktop Autologin* i naciskamy Enter. Wychodzimy z konfiguratora, zatwierdzając restart systemu, który ponownie uruchomi się już w wersji graficznej.

Przygotowanie sensora

Sensor BBair jest gotowy do pracy natychmiast po włączeniu zasilania. Korzystając z urządzenia mobilnego typu smartfon lub tablet z zainstalowaną aplikacją BBair, można w dowolnej chwili wykonać pomiar jakości powietrza. Aby użyć sensora w tym projekcie,



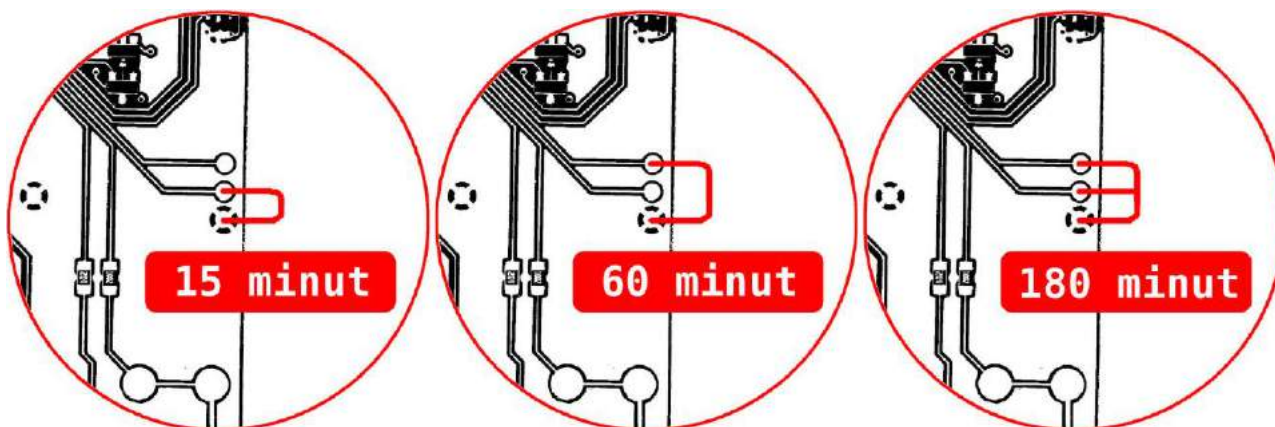
Rysunek 3. Sposób połączenia ogniwa fotowoltaicznego

konieczne jest dostosowanie go do nowych potrzeb. Będzie to polegało na:

1. włączeniu trybu pomiarów automatycznych,
2. wyposażeniu sensora w funkcję ładowania z ogniwa słonecznego,
3. oraz przystosowaniu go do pracy w warunkach atmosferycznych.

Do włączenia trybu automatycznych pomiarów wystarczy wykonanie połączenia padów konfiguracyjnych na płycie sensora. Do wyboru mamy trzy okresy, spośród których wybieramy jeden, łącząc pola według wskazówek z **rysunku 2**. Zwróćmy uwagę na to, że im częściej sensor będzie wykonywał pomiary, tym więcej energii będzie musiał dostarczyć panel słoneczny. Z kolei to, ile energii pozyska ogniwo fotowoltaiczne, zależy m.in. od: warunków nasłonecznienia panujących w miejscu zainstalowania sensora, sprawności układu ładowania oraz oczywiście mocy samego panelu. Sensownym kompromisem na początek wydaje się wybranie godzinowego okresu wykonywania pomiarów.

Dalej zajmiemy się kwestią zasilania. EKO-stacja, czerpiąc energię jedynie z akumulatorów, wymagałaby co jakiś czas (proporcjonalny do uprzednio wybranego okresu wykonywania automatycznych pomiarów) wymiany wyczerpanego zestawu na nowy. W ten sposób mając cztery akumulatory AAA, moglibyśmy zapewnić ciągłość zasilania: dwa ogniwa byłyby w stacji, a dwa po naładowaniu oczekiwałyby na odpowiedni moment podmienienia. Byłoby to jednak nieco uciążliwe, choć wielokrotnie rzadsze niż ładowanie urządzeń mobilnych. Istnieje jednak dość prosty sposób na wyeliminowanie konieczności cyklicznej obsługi. Wykorzystując fakt, że stacja zainstalowana będzie pod gołym niebem, możemy podłączyć ją do ogromnego źródła darmowej energii. **Rysunek 3** pokazuje najprostszy sposób połączenia ogniwa fotowoltaicznego poprzez diodę prostowniczą z sensorem



Rysunek 2. Sposób wyboru częstotliwości pomiarów



Rysunek 4. Gotowy sensor w pomysłowej obudowie

skonfigurowanym do wykonywania pomiarów co 60 minut. Dziwić może fakt pominięcia regulatora ładowania i zabezpieczenia użytych ogniw NiMH przed przeładowaniem. Wątpliwości szybko rozwiewa zestawienie parametrów zastosowanych elementów. W naszym klimacie użyte akumulatory o pojemności 1000 mAh nie zostaną uszkodzone panelem słonecznym o mocy szczytowej 0,6 W. Panel nigdy nie będzie pracował w punkcie swojej maksymalnej mocy, więc uzyskana ilość energii będzie niewielka. Zrobmy szybkie oszacowanie: prąd pobierany z baterii przez sensor podczas pomiaru wynosi średnio 120 mA. Pomiar trwa 1 minutę, zatem zużycie energii to 2 mAh ($120 \text{ mA} \times 1/60 \text{ h}$). Jeśli ustawiliśmy stację do powtarzania pomiarów co 1 godzinę, to w ciągu doby mamy 24 pomiary. Daje to ostatecznie zużycie energii równe 48 mAh na dobę. Prąd czuwania jest tak mały, że spokojnie możemy go pominąć w tych szacunkach bez negatywnego wpływu na wynik obliczeń. Gdy założymy, że panel dostarczałby energię tylko przez 5 godzin na dobę, wystarczyłby ciągły prąd ładowania o wartości 10 mA. Niewiele, ale należy wziąć pod uwagę również występowanie kilku dni pochmurnych z rzędu oraz ekspozycję ogniw na słońce w miejscu zainstalowania EKO-stacji. Dysponując ogniwem słonecznym o większej mocy, konieczne będzie zastosowanie jakiejś formy zabezpieczenia akumulatorów przed przeładowaniem.

Teraz uzasadnimy, dlaczego budowana stacja nosi przedrostek EKO. W dzisiejszych czasach obfitości, gdy coraz bardziej doskwiera nam problem zaśmiecenia świata, pożądaną cechą staje się umiejętność ponownego wykorzystania użytych już rzeczy. Znalazienie dla nich nowej funkcji lub nowego wymiaru użyteczności. W tym właśnie duchu wykonana została obudowa EKO-stacji chroniąca sensor przed letnim deszczem i zimowym śniegiem. W sumie bardzo proste w wykonaniu, bardzo tanie i bardzo skuteczne rozwiązanie. W cienkiej ściance obciętej butelki PET wykonano niewielki otwór, przez który przeprowadzony jest przewód od ogniw słonecznego do wnętrza. Otwór zabezpieczony został klejem, który jednocześnie mocuje panel fotowoltaiczny. Przez otwór w nakrętkę przewleczono twardy drut w otulinie izolacyjnej obustronnie zakończony uformowanymi haczykami: jeden z nich posłuży do podwieszenia sensora BBair, a drugi do zainstalowania całego zestawu EKO-stacji w docelowym miejscu zbierania danych pomiarowych. Instalując sensor wewnątrz butelkowej kopułki, pamiętajmy o tym, aby

otwory służące do zasysania badanego powietrza skierowane były w dół (rysunek 4).

Instalacja aplikacji BBair_station na Raspberry Pi

Część sprzętowa jest już gotowa. Pozostaje zainstalować oprogramowanie i świętować zakończenie projektu. Ponieważ proces instalacji niezbędnego oprogramowania składa się z kilku kroków, dla ułatwienia przygotowany został skrypt wykonujący w kolejności wszystkie niezbędne zadania. Logujemy się zatem do systemu i pobieramy go poleceniem: `wget http://bbmagic.net/download/install_BBair_station.sh`

Aby możliwe było jego uruchomienie, konieczna jest zmiana praw dostępu do pobranego pliku komendą:

```
sudo chmod +x install_BBair_station.sh
```

Wynik działania powyższych poleceń pokazuje zrzut ekranu z **rysunku 5**. Teraz możliwe jest skorzystanie ze skryptu, który pobierze aplikację *BBair_station* oraz sterownik wyświetlacza LCD. Uruchamiamy go poleceniem:

```
./install_BBair_station.sh
```

Po chwili ekran wypełnią komunikaty raportujące przebieg instalacji jak na zrzucie ekranu z **rysunku 6**. Aplikacja *BBair_station* zainstalowała się w systemie, a na pulpicie znajduje się ikona umożliwiająca jej uruchomienie. Z aplikacji można korzystać na dużym ekranie, mając podczas pracy bieżące informacje o stanie powietrza za oknem (**rysunek 7**). Skorzystanie z podłączonego wyświetlacza LCD umożliwi nam pobranie skryptu, uruchomimy go z parametrem „*lcd-4*”. Po wpisaniu komendy:

```
./install_BBair_station.sh lcd-4
```

```
pi@raspberrypi:~$ wget http://bbmagic.net/download/install_BBair_station.sh
--2019-04-24 11:03:44-- http://bbmagic.net/download/install_BBair_station.sh
Resolving bbmagic.net (bbmagic.net)... 95.17.23.23
Connecting to bbmagic.net (bbmagic.net)|95.17.23.23|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2516 (2.5K) application/x-sh
Saving to: 'install_BBair_station.sh'

install_BBair_station. 100%[=====] 2.46K --.-73/s in 0s
2019-04-24 11:03:44 (71.8 MB/s) - 'install_BBair_station.sh' saved [2516/2516]

pi@raspberrypi:~$ sudo chmod +x install_BBair_station.sh
pi@raspberrypi:~$
```

Rysunek 5. Pobranie skryptu i zmiana praw dostępu

```
pi@raspberrypi:~$ ./install_BBair_station.sh
-----
BBair Station App installer START
-----
Creating BBair station folder...OK
Installing App...OK
Downloading LCD driver...OK
Installing LCD driver...OK
-----
BBair station install completed !!
Click desktop icon to start BBair Station App.
'./install_BBair_station.sh lcd-4' - switch to 4-inch SPI LCD
'./install_BBair_station.sh lcd-4-180' - switch to 4-inch SPI LCD upside down
'./install_BBair_station.sh lcd-hdmi' - switch to HDMI display
See more on: www.bbmagic.net

pi@raspberrypi:~$
```

Rysunek 6. Efekt uruchomienia skryptu

Rysunek 7. Widok działającej aplikacji

Rysunek 8. Skrypt uruchamiający dołączony wyświetlacz

Rysunek 9. Skrypt umożliwiający ponowne przetłoczenie ekranów

Rysunek 10. Skrypt umożliwiający obrócenie wyświetlanej treści

`./install_BBair_station.sh lcd-4-180`

który spowoduje przełączenie na wyświetlacz LCD, z tym że wyświetlany obraz będzie obrócony o 180 stopni. Na pierwszy rzut oka nie widać zastosowania dla tej funkcji, ale po sprawdzeniu, w którym kierunku względem wyświetlacza wyprowadzone są przewody Raspberry Pi zero W, wszystko stanie się w pełni zrozumiałe. Obrócenie obrazu do góry nogami pozwala prowadzić przewody w dół, co podniesie estetykę magicznego ekranu o kilka rzędów wielkości.

Opis budowy EKO-stacji (zwłaszcza aspekt mechaniczny) w artykule został skrócony do niezbędnego minimum, ale w serwisie YouTube (po wyszukaniu frazy „projekt bbmagic”) dostępny jest film dokumentujący ten proces. Zainteresowani Czytelnicy na pewno znajdą tam wiele interesujących szczegółów, zgodnie z empiryczną zasadą, że jeden obraz wart jest więcej niż tysiąc słów. Pliki źródłowe aplikacji *BBair_station* są dostępne pod adresem <http://bit.ly/32g2Hm6>.

Mariusz Żądło

iram@poczta.onet.pl

100% ELEKTRONIKI NA AVT.PL/PRENUMERATA

[illegible]

**Prenumerujesz Elektronikę Praktyczną + Elektronikę dla Wszystkich?
Skorzystaj z promocji 1+1=3 i zamów bezpłatną prenumeratę Elektronika**

Raspipool

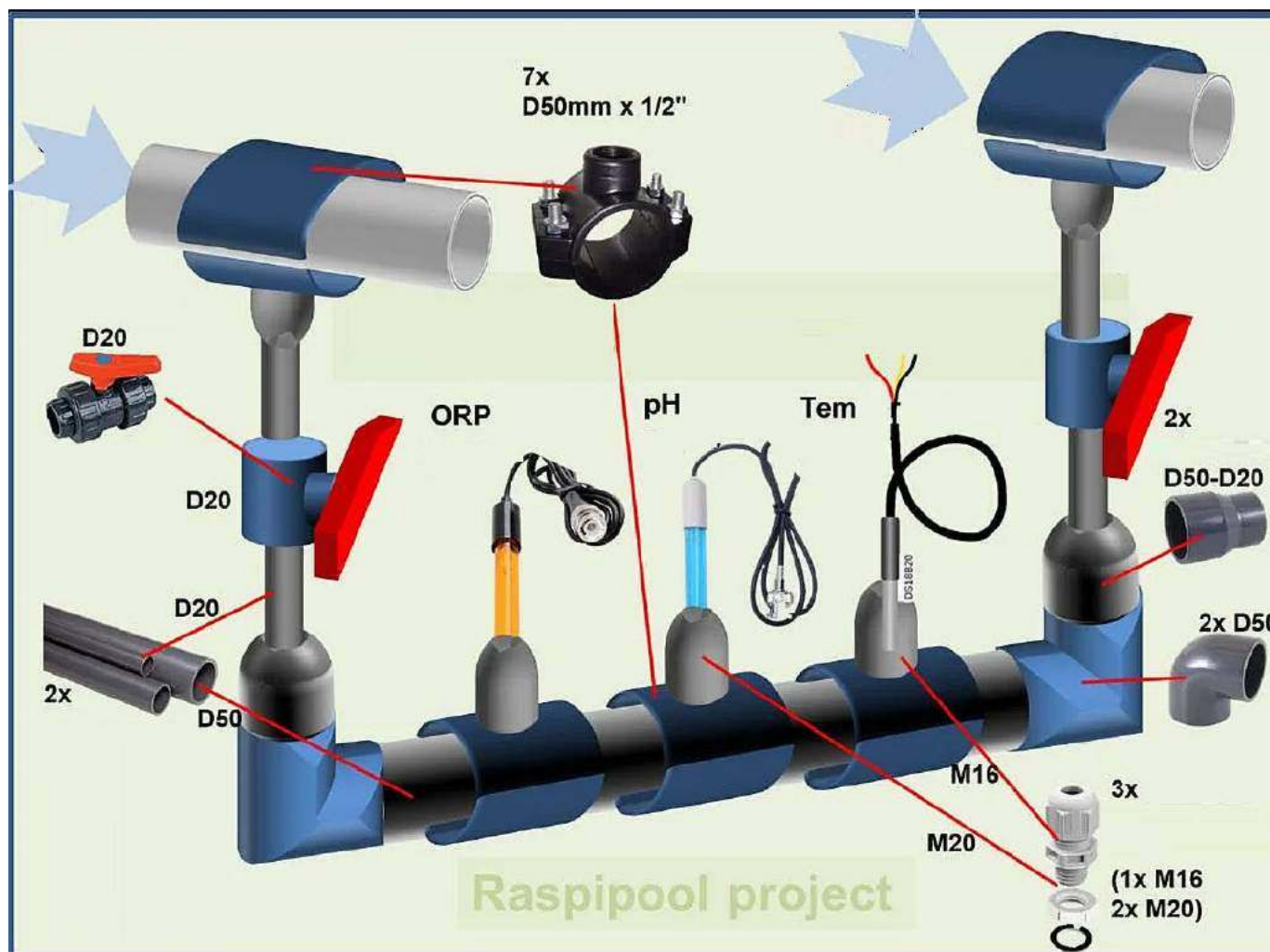
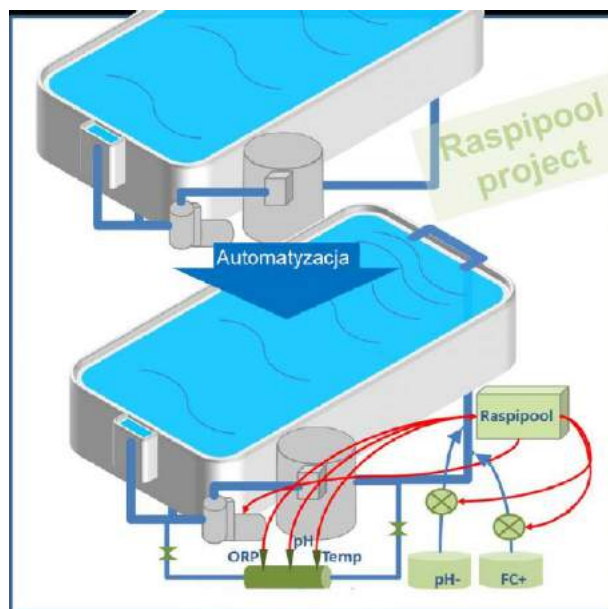


System automatyki do przydomowego basenu

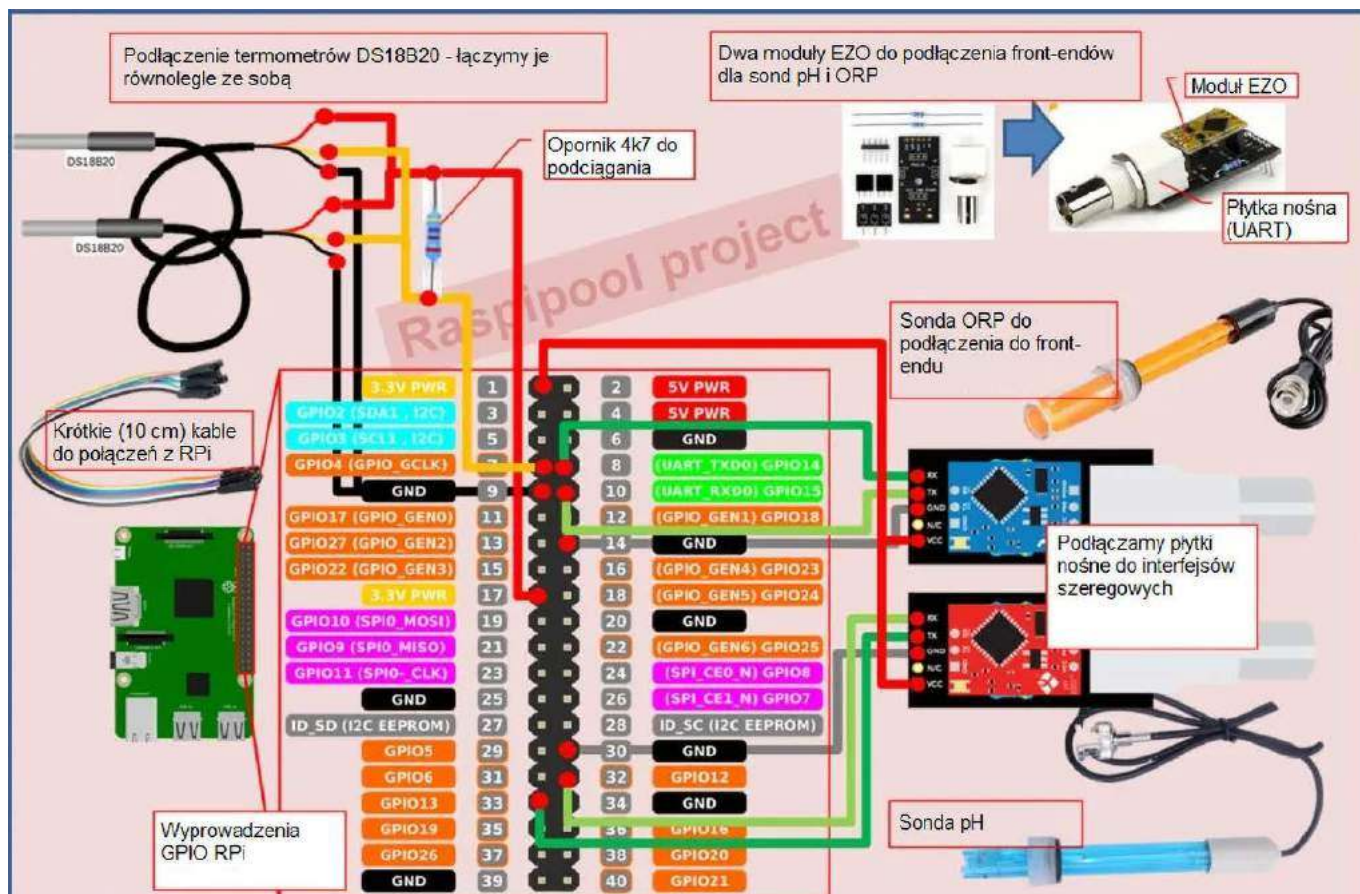
Zima jest doskonałym okresem, aby rozpocząć realizację projektu Raspipool. Jest to system automatyki do przydomowego basenu, który z pewnością przyda się późną wiosną i latem. Dzięki niemu korzystanie z basenu stanie się samą przyjemnością, a czynności konserwacyjne basenu odejdą w niepamięć.

Dotychczasowe systemy automatyki do obsługi basenów, szczególnie tych znajdujących się w naszych ogródkach, nie były ani łatwe do zastosowania, ani tanie. Teraz, dzięki rozwojowi elektroniki i dostępności różnorodnych modułów elektronicznych, systemy tego rodzaju przestają być skomplikowane w realizacji i drogie. W poniższym artykule zaprezentujemy taki system zbudowany na komputerze jednopłytkowym Raspberry Pi.

Zautomatyzowanie we wnętrzu domu staje się standardem. Jednak naprawdę nieporównywalne korzyści może przynieść automatyzacja takiego obiektu jak przydomowy basen, szczególnie w zakresie



Rysunek 1. Schemat dołączenia sensorów do układu wodnego basenu



Rysunek 2. Schemat podłączenia sensorów do pomiaru najważniejszych parametrów wody w basenie

koniecznych nakładów czasu, potrzebnych do jego utrzymania pieniędzy i obciążenia dla środowiska. Głównym celem prezentowanego poniżej projektu jest zbudowanie taniego i modułowego systemu, który umożliwia sterowanie urządzeniami związanymi z basenem. Do sterowania zastosowano komputer jednopłytkowy Raspberry Pi, oferujący dużą moc obliczeniową, co pozwala na implementację wielu zaawansowanych algorytmów sterowania.

Wstęp

Urządzenia do automatyzacji, stosowane w domach czy ogrodach, najczęściej kontrolują kilka parametrów i wykonują kilka zadań przy realizacji swego celu. Zwykle są to urządzenia autonomiczne, to znaczy, że zbierają jakieś informacje z otoczenia a następnie dobierają odpowiednie parametry pracy dla dołączonego osprzętu. Informacja o zebranych danych i o odpowiedzi urządzenia nie jest dostępna dla innych urządzeń. Problem pojawia się w momencie, gdy w systemie znajdują się dwa lub więcej urządzeń, które próbują regulować ten sam parametr. Przykładem mogą być systemy kontroli pH w basenie. Na ten parametr wpływa działanie wielu urządzeń w systemie, od systemów dozujących do wody w basenie solankę, poprzez specjalne pompy kontrolujące filtrowanie, aż do zwykłych pomp z kontrolą czasową, które pompują wodę do/z basenu. Taki sprzęt tylko częściowo rozwiązuje zadania automatyzacji i nie pozwala na żaden rodzaj rozbudowy lub ulepszenia.

Podejście prezentowane w opisywanym projekcie jest zupełnie inne. Autor proponuje, aby urządzenia peryferyjne były ekstremalnie proste (więc także tanie) i realizowały tylko swoje główne zadanie – pompa: filtruje, roztwór soli: wytwarza chlor etc. Natomiast centralny sterownik odpowiada za całą automatyzację. Właśnie w ten sposób konstruowane są profesjonalne systemy sterowania, ale te mają dwie podstawowe wady: są bardzo drogie oraz najczęściej zmuszają do zakupu wszystkich komponentów tej samej marki.

Opis systemu

System korzysta z zespołu sensorów umieszczonych w rurze, będącej obejściem pompy, która przepompowuje wodę z basenu przez filtr. W pojedynczej rurze z PCV należy umieścić sensory: ORP (potencjał oksydacyjny), pH oraz temperatury. Oczywiście, projekt można uzupełnić dodatkowymi sensorami, według własnych potrzeb.

Elementami wykonawczymi są: pompa filtrująca wodę oraz dwie pompy perystaltyczne, przeznaczone do podawania do wody w basenie roztworów do zakwaszania wody (redukcji pH) oraz chlorku sodu, którego zadaniem jest sterylizowanie wody. System można uzupełnić o sterowane grzałki do wody czy elektrozawory podające wodę z sieci wodociągowej dla uzupełnienia jej braku.

Wszystkimi systemami steruje pojedynczy komputer jednopłytkowy Raspberry Pi z zainstalowanym systemem operacyjnym Hassbian. Jest to dystrybucja Raspbiana, na którym domyślnie zainstalowany jest pakiet oprogramowania Home Assistant – otwarta platforma automatyki domowej.

Potrzebne komponenty

Do zbudowania systemu automatyzacji basenu potrzebne będą następujące elementy:

- Komputer jednopłytkowy Raspberry Pi, najlepiej w wersji 3 lub 4, wraz z kartą pamięci (rekomendowana jest karta o pojemności co najmniej 32 GB i klasie prędkości A1) oraz zasilaczem USB o wydajności prądowej nie mniejszej niż 2 A.
- Sensor pH (EZO pH od Atlas Scientific),
- Sensor ORP (EZO ORP od Atlas Scientific),
- Płytki nośne od Atlas Scientific do podłączenia modułów do systemu,
- Termometr DS18B20 w wodoodpornej obudowie wraz z opornikiem 4,7 kΩ, który podłącza się bezpośrednio do GPIO Raspberry Pi,
- Przełączniki i styczniki do sterowania urządzeniami elektrycznymi: pompami do filtrów, pompami perystaltycznymi, grzałkami

etc. Pamiętajmy, aby dobrać elementy sterujące do posiadanych przez nas urządzeń; prąd i napięcie pracy styków przekaźników musi być nie mniejsze niż napięcie zasilające urządzenie i prąd przez nie pobierany. Część z przekaźników stosowana jest również do kontroli silników, jeżeli np. mają one dwa biegi z różnymi prędkościami obrotowymi.

- Opcjonalnie – monitor zużycia energii elektrycznej, który można podłączyć do Raspberry Pi. Na rynku jest wiele tego rodzaju sensorów. Pozwoli on nam zmierzyć, ile prądu pobiera nasz system basenowy, co umożliwi dostrojenie algorytmu kontroli tak, by system był możliwie energooszczędny.

Obejście systemu – pętla dla sensorów

Na **rysunku 1** pokazano rurę, w której umieszczono sensory. Jest ona instalowana jako obejście głównej pompy z filtrami. Umieszczamy w obejściu wszystkie sondy (pH, ORP i temperatury, a także inne, według uznania), jak zaprezentowano na rysunku. Dołączenie takiego układu do systemu wodnego w basenie jest proste i nie powinno być dla nikogo problemem.

Do zbudowania obejścia potrzebnych jest kilka elementów – trochę rur PCV, klej do PCV oraz podstawowe narzędzia, takie jak wiertarka czy śrubokręt. Całe obejście nie powinno kosztować więcej niż 200 zł plus koszt sond, jakie w nim umieścimy. Podczas budowy tego elementu trzeba zwrócić uwagę na trzy rzeczy:

1. budując obejście, dobrze jest zmontować i spasować je ze sobą i z istniejącymi już rurami, zanim wszystko sklejimy i połączymy na stałe. Możemy robić to wielokrotnie, aby dopasować do siebie otwory, pozycje kolanek etc.,
2. rura, w której znajdują się sondy (D50 – średnica 50 mm), powinna być ułożona możliwie nisko i poziomo w systemie, tak aby zawsze była pełna wodą. Zaleca się, aby odległość pomiędzy

sondami a komputerem z modułami pomiarowymi nie była większa niż 3 m.

3. krawędzie rur, które ze sobą łączymy, należy lekko zaokrąglić pilnikiem bądź elektryczną szlifierką. Dobrze jest przeciwdziałać wykonywaniu takiej czynności na krawędzi rur na niepotrzebnym, odciętym fragmencie.

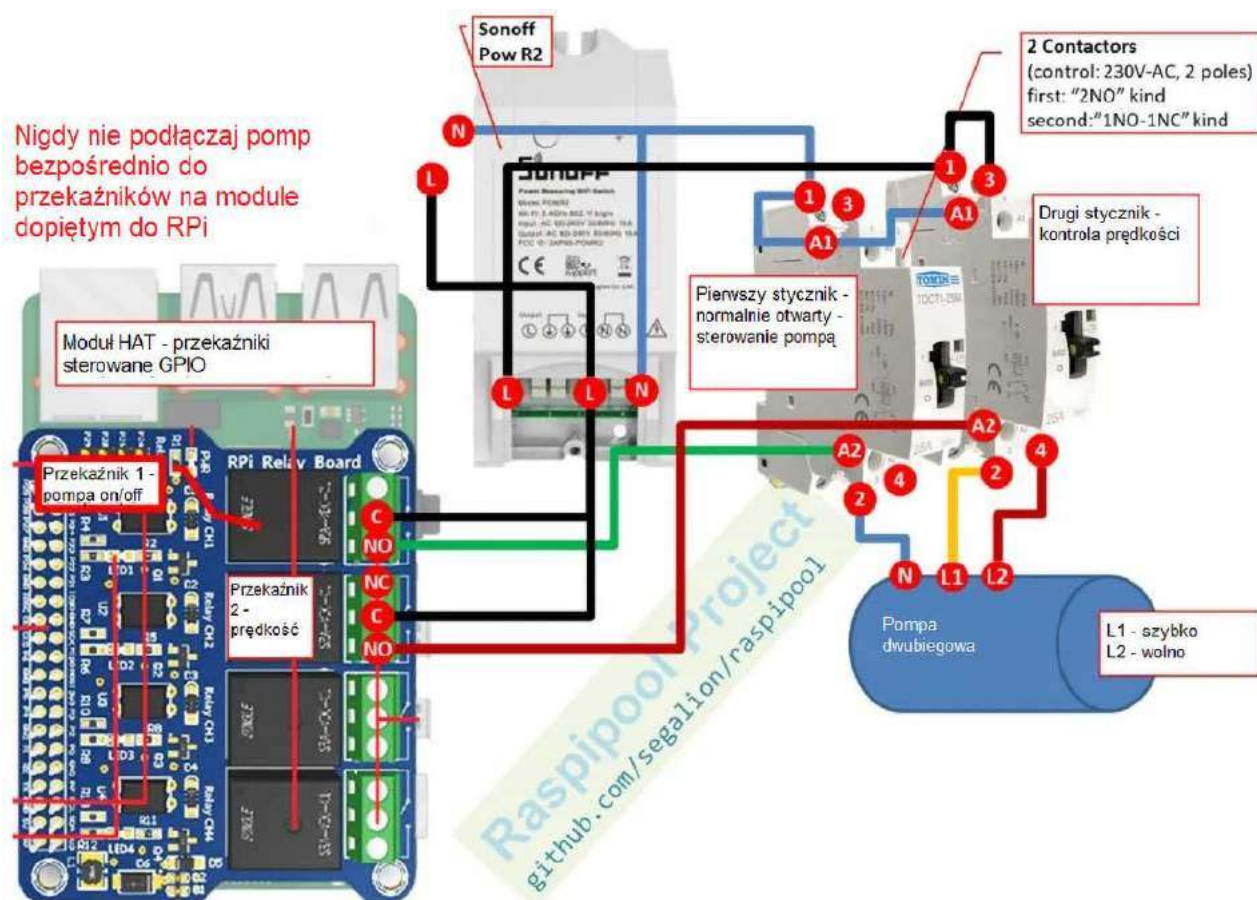
Podstawowe oprogramowanie na Raspberry Pi

Aby oprogramowanie działało na naszym Raspberry Pi, trzeba zainstalować na nim Hassbiana – jest to dystrybucja Raspbiana wyposażona w oprogramowanie Home Assistant. Zaawansowani użytkownicy Linuksa, mogą zainstalować Home Assistant niezależnie, na istniejącej już instalacji systemu operacyjnego Raspbian lub innego systemu dla tego komputera jednopłytkowego.

Warto zapewnić łączność z Internetem naszemu systemowi. Dzięki temu nie tylko instalacja oprogramowania czy aktualizacji będzie prostsza, ale będzie możliwa także integracja systemu z popularnymi systemami automatyki domowej, takimi jak Sonos, Kodi czy ekosystemami Alexa, Google Home itp. Dodatkowo połączenie z Internetem pozwala na korzystanie z usług takich jak np. *If This Then That* (IFTTT) do tworzenia własnych algorytmów automatyki.

Jeśli to Wasze pierwsze zetknięcie z Home Assistantem, możecie być przytłoczeni ogromną liczbą zintegrowanych narzędzi dostępnych w tej aplikacji. Niniejszy projekt jest dobrą metodą na rozpoczęcie korzystania z tego pakietu, szczególnie że wszystkie pliki do korzystania z poszczególnych sond i pliki konfiguracyjne są dostępne do pobrania.

Na repozytorium w GitHubie projektu (<http://bit.ly/2NbuBt3>) znajdujemy foldery *custom_components* i *packages*, które musimy skopiować na nasz minikomputer.



Rysunek 3. Schemat podłączenia styczników do sterowania pompą głównego obiegu wody

Listing 1. Modyfikacja pliku *configuration.yaml*

```
homeassistant:
  name: Raspipool
  # Dane lokalizacyjne potrzebne są do obliczenia godziny wschodu
  # i zachodu słońca
  latitude: !secret latitude
  longitude: !secret longitude
  unit_system: metric
  customize: !include customize.yaml
  packages: !include_dir_named packages

default_config:
# Wyłączenie automatycznego wykrywania urządzeń
# discovery:

tts:
- platform: google_translate

logger:
  default: critical
  logs:
    homeassistant.components.sensor.atlas_scientific: debug
    homeassistant.components.switch.command_line: debug

group: !include groups.yaml
# Wyłączenie automatyzacji
# automation: !include automations.yaml

script: !include scripts.yaml

lovelace:
  mode: yaml
```

Tak samo musimy postąpić z plikiem konfiguracyjnym *ui-lovelace.yaml*, który opisuje wygląd przedniego panelu Raspipool. Plik ten umieszczamy w folderze konfiguracyjnym Home Assistant (zazwyczaj */home/homeassistant/.homeassistant/*). Trzeba ponadto zmodyfikować plik *configuration.yaml*, tak jak pokazano na **listingu 1**. Wpisujemy tam dane dotyczące lokalizacji czy inne informacje dla różnych API, z których korzystać może system.

Podłączanie sensorów do minikomputera

Możemy teraz podłączyć poszczególne sensory do minikomputera Raspberry Pi. Na **rysunku 2** zaprezentowano schemat podłączenia poszczególnych sensorów do Raspberry Pi. Termometry DS18B20 podłączone są do jednego pinu Raspberry Pi (można podłączyć ich wiele do jednego pinu). Moduły EZO z sensorami podłączone są do interfejsów UART minikomputera. Aby interfejs szeregowy działał, musimy zmienić wpis w pliku konfiguracyjnym *config.txt*. W tym celu w linii komend wpisujemy:

```
sudo nano /boot/config.txt
```

W pliku konfiguracyjnym musimy zmienić lub dodać wpisy, by wyglądały one tak, jak poniżej:

```
force_turbo=1
enable_uart=1
dtoverlay=pi3-disable-bt
# second UART1 /dev/ttyS0
dtoverlay=uart1,txd1_pin=32,rx1_pin=33
```

Podłączenie przełączników pomp

W kolejnym kroku możemy do Raspberry Pi podłączyć układy wykonawcze – przełączniki i styczniki, które pozwolą m.in. na kontrolę głównej pompy. Na **rysunku 3** zaprezentowano sposób podłączenia przełączników (na module HAT – zamontowanym na Raspberry Pi) do styczników, które sterują pompą. Jeden stycznik odpowiedzialny jest za włączanie i wyłączanie pompy, a drugi z nich za kontrolę jej prędkości obrotowej, jeśli w naszym systemie mamy pompę dwubiegową.

Do sterownika dołączony jest także moduł Sonoff Pow R2 do pomiaru zużycia energii elektrycznej przez pompę oraz wszystkie inne elementy infrastruktury basenowej, które dołączymy przez ten moduł do zasilania.

W module HAT z przełącznikami, jaki dołączono do Raspberry Pi, pozostają jeszcze dwa kanały. Można je wykorzystać np. do kontroli pomp perystaltycznych zasilanych napięciem stałym 12 V. Pompy te odpowiedzialne są za podawanie np. kwasu, do regulacji pH wody oraz substancji dezynfekującej basen, której zadawanie regulowane jest przez oprogramowanie na podstawie pomiaru ORP.

Konfiguracja Raspipool

Po zmontowaniu i uruchomieniu systemu można przejść do jego konfiguracji. Całe sterowanie systemem i parametrami pracy odbywa się poprzez webowy interfejs. Stronę konfiguracyjną można wyświetlić, korzystając z dowolnej przeglądarki, z poziomu komputera lub smartfona. W menu konfiguracyjnym ustawiamy wszystkie parametry Raspipool, jakie są potrzebne – podajemy:

- wielkość (pojemność) naszego basenu,
- zadany cykl filtrowania (jego czas trwania),
- prędkość podawania środków z pomp perystaltycznych,
- stężenie podawanych przez pompy perystaltyczne środków,
- jakie chcemy otrzymywać powiadomienia,
- i wiele innych parametrów.

Dodawać można też oczywiście własne funkcjonalności do otwartego środowiska zapewnianego przez Home Assistant.

Podsumowanie

Home Assistant i Raspipool tworzą doskonałe, otwarte środowisko do stworzenia systemu automatyzacji obsługi przydomowego basenu. Dzięki użyciu otwartego oprogramowania system można modyfikować, aby dostosować go do własnych potrzeb. Do komputera jednopłytkowego Raspberry Pi można podłączyć nie tylko opisane powyżej urządzenia, ale także wiele innych, które zapewnią nam komfort korzystania z naszego basenu. Jeśli do realizacji projektu zabierzemy się już teraz, to od pierwszych ciepłych miesięcy przyszłego roku będziemy mogli cieszyć się zautomatyzowanym basenem obok naszego domu.

Nikodem Czechowski, EP

REKLAMA

**ELEKTRONIKA
PRAKTYCZNA**

**MATERIAŁY
DODATKOWE**



MEDIA

Aby skorzystać z materiałów dodatkowych dołączonych do numeru, należy:

1. Wejść na stronę **www.media.avt.pl**
2. Zarejestrować się lub zalogować
3. Wybrać wydanie „Elektroniki Praktycznej”, które ma trafić
4. Odpowiedzieć na proste pytanie dotyczące bieżącego numeru
5. Pobrać pliki do biblioteki osobistej



Sensor pracy mięśni z wizualizacją w sieci Web i na diodach LED RGB

Nazwa tego projektu początkowo może wydawać się nieco dziwna, ale poniższy system właśnie tak działa – steruje diodami LED RGB za pomocą impulsów elektrycznych, które docierają do naszych mięśni przez układ nerwowy. Działanie takiego urządzenia jest możliwe dzięki temu, że nasz „system sterowania”, podobnie jak mikrokontrolery, używa elektryczności.

Opisany poniżej układ bazuje na module do elektromiografii **MyoWare**. Sensor ten został zaprojektowany do pomiaru niewielkich sygnałów elektrycznych, jakie są obecne w naszych mięśniach, gdy to się poruszają. Na wyjściu sensora powstaje sygnał analogowy, który można mierzyć za pomocą dowolnego mikrokontrolera.

Elektromiografia (EMG) mierzy odpowiedź mięśni lub aktywność elektryczną w odpowiedzi na stymulację mięśnia przez układ nerwowy. Badanie to służy zazwyczaj do wykrywania nieprawidłowości nerwowo-mięśniowych. Podczas testu jedną lub więcej małych igieł (zwanymi również elektrodami) wprowadza się przez skórę do mięśnia. W przypadku opisanego modułu nie ma potrzeby wprowadzać elektrod do mięśnia, wystarczy je nakleić na skórę.

Potrzebne elementy

Do budowy opisywanego urządzenia będą potrzebne następujące elementy:

- Moduł sensora mięśniowego MyoWare,
- Elektrody do EMG,
- Moduł NodeMCU (z mikrokontrolerem ESP8266),

- Pasek LED-owy Neopixel,
- Kable do podłączenia paska LED-owego i kabel microUSB.

Dodatkowo, na komputerze, na którym będziemy kompilować kod programu w Arduino IDE do tego urządzenia, oprócz samego środowiska programistycznego dla Arduino musimy zainstalować następujące biblioteki:

- Adafruit io Arduino (w wersji co najmniej 2.3.0),
- Adafruit Neopixel (wersja 1.2.5 i wyżej),
- Adafruit MQTT (1.0.3),
- Arduino HttpClient (0.4.0).

Jeśli nie posiadasz w swoim IDE zainstalowanych takich bibliotek, można je z łatwością doinstalować z poziomu Arduino IDE, klikając po kolei *Szkic* → *Dołącz bibliotekę* → *Zarządzaj bibliotekami*. Tam wystarczy wpisać nazwę biblioteki, aby móc ją odnaleźć i zainstalować.

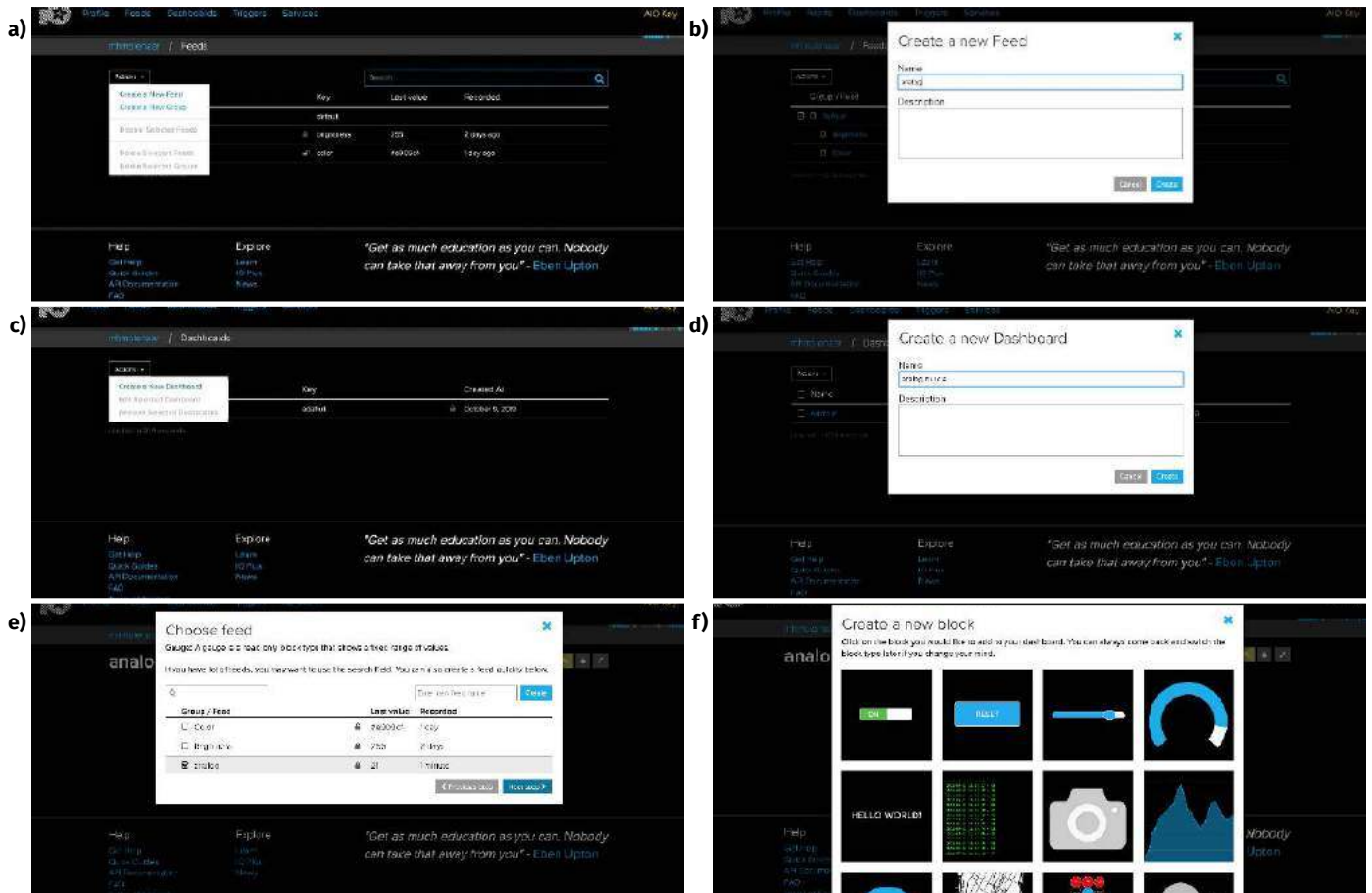
Krok 1: Podłączenie sensora MyoWare do NodeMCU

Jeśli nie jesteś zaznajomiony z sensorem MyoWare, realizację tego projektu warto rozpocząć od zrozumienia, w jaki sposób sensor jest podłączony do Arduino. Sposób przygotowania opisany został na stronie Adafruit <http://bit.ly/31Pcwpp>.



Do podłączenia sensora potrzebne są zaledwie trzy przewody – dwa do zasilania (VCC i masa) oraz jeden dla sygnału wyjściowego. Powinny to być dłuższe kable zakończone złączami do goldpinów modułu NodeMCU. Dodatkowo musimy przylutować krótki kabelek zakończony elektrodą.

W celu przetestowania modułu podłączamy wyjście sygnału EMG do wejścia analogowego A0 mikrokontrolera. Z menu w Arduino IDE



Rysunek 1. Konfiguracja Adafruit IO

należy wybrać przykład *01.Basics* → *AnalogReadSerial*. Szkic ten odczytuje wartość analogową z pinu A0 i przesyła ją poprzez port szeregowy. Jeśli po załadowaniu szkicu zobaczymy na ekranie monitora portu szeregowego wartości zmieniające się wraz z ruchem mięśni, do których podłączona jest elektroda MyoWare, to znaczy, że wszystko poprawnie podłączyliśmy.

Krok 2: Konfiguracja Adafruit IO

Pierwszym krokiem jest zalogowanie się lub utworzenie konta na portalu Adafruit IO. Wchodzimy na stronę www.io.adafruit.com i tam logujemy się lub tworzymy nowe konto użytkownika, którego będziemy używali w dalszej części projektu. Po zalogowaniu się do naszego konta możemy utworzyć nowe źródło danych (kanał) – *Feed* – tak jak pokazano na **rysunku 1a** oraz **1b**. *Feed* nazywamy *Analog*, ponieważ będzie obsługiwał wejście analogowe. Jeśli chcemy, możemy dodać do niego także opis, objaśniający, co to za wejście.

Jeśli chcemy prezentować dane z wejścia analogowego w Adafruit IO, musimy dodać i skonfigurować odpowiedni panel, gdzie będziemy je pokazywali. W tym celu, jak pokazano na **rysunku 1c** i **1d**, tworzymy nowy *Dashboard*, który nazywamy dowolnie. Na tym panelu możemy dodać dowolne wizualizacje tych danych, pamiętając tylko, by jako źródło danych ustawić skonfigurowany wcześniej *feed* o nazwie *Analog* (patrz **rysunek 1e**). Wybierzmy wskaźnik o nazwie *Gauge* i skonfigurujemy zakres danych od 0 (minimum) do 1024 (maksimum) – takie wartości zwraca nasz sensor. W katalogu *Adafruit IO* znajdziemy wiele różnych metod wizualizacji danych, część z nich została pokazana na **rysunku 1f**.

Krok 3: Wejście analogowe w Arduino

W kroku pierwszym podłączyliśmy i przetestowaliśmy sensor pracy mięśni MyoWare, teraz przejdziemy do oprogramowania go na mikrokontrolerze, z użyciem biblioteki i dashboarda, który wygenerowaliśmy w kroku 2. Nasz program będzie łączył się z siecią

i z API Adafruit IO, gdzie będzie wysyłał zbierane dane. Pozwoli to na wizualizowanie ruchu mięśni i prezentację efektów w sieci. Po upewnieniu się, że mamy zainstalowane wszystkie wymienione powyżej biblioteki, możemy uruchomić przykładowy szkic, na którym zbudujemy nasz program. W tym celu z menu z przykładami wybieramy: *Adafruit IO Arduino* → *adafruitio_08_analog_in*, jak pokazano na **rysunku 2a**. Musimy teraz skonfigurować w tym przykładowym szkicu szereg rzeczy, aby działał z naszym kontem Adafruit IO.

Aby skonfigurować ustawienia sieciowe, kliknij kartę *config.h* na szkicu. Musimy wpisać tam swoją nazwę użytkownika Adafruit IO w definicji *IO_USERNAME*, a kluczem do IO Adafruit w definicji *IO_KEY*. Nazwę użytkownika i klucz znaleźć można w prawym górnym rogu ekranu na stronie Arduino IO (**rysunek 2b**):

```
#define IO_USERNAME „your_username”
#define IO_KEY „your_key”
```

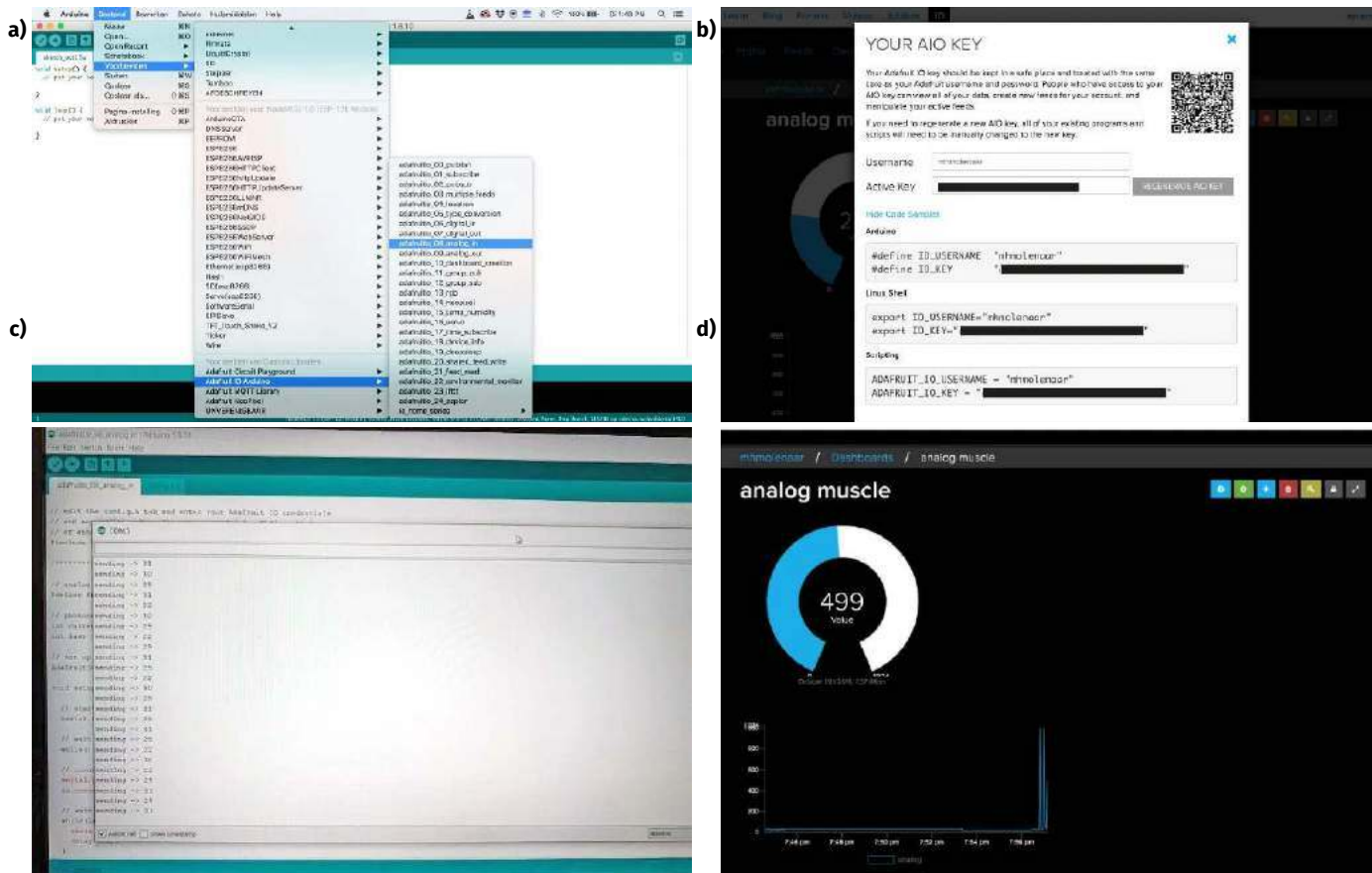
Następnie możemy skonfigurować połączenie NodeMCU z naszą siecią Wi-Fi. W tym celu musimy uzupełnić w pliku *config.h* parametry *WIFI_SSID* oraz *WIFI_PASS*, opisujące, odpowiednio, nazwę naszej sieci domowej oraz hasło, jakiego NodeMCU ma używać podczas podłączania się do Wi-Fi.

```
#define WIFI_SSID „your_ssid”
#define WIFI_PASS „your_pass”
```

Po wprowadzeniu powyższych zmian możemy przetestować system. Jeśli podłączyliśmy sygnał z sensora do wejścia analogowego A0, to nie musimy nic zmieniać. Możemy zmienić nazwę zmiennej, przechowującej numer pinu, by zwiększyć czytelność kodu. Zmieniamy *PHOTOCELL_PIN* na *MUSCLESENSOR_PIN*. Robimy tak w dwóch miejscach, gdzie występuje odwołanie do tej wartości:

```
// analog pin 0
#define PHOTOCELL_PIN A0
```

W ostatniej linijce kodu znajduje się opóźnienie – jest ono ustawione domyślnie na 1000 milisekund, ale zmieniamy na 2000 milisekund,



Rysunek 2. Szkic obsługujący Adafuit IO oraz dane potrzebne do zalogowania do systemu

z uwagi na ograniczenie przepustowości Adafuit IO w wersji podstawowej, z jakiej korzystamy (szybsze przetwarzanie danych wymaga stosownej licencji).

```
// wait one second (1000 milliseconds == 1 second)
delay(2000);
```

Po wprowadzeniu tych zmian, możemy przetestować szkic. Wystarczy go skompilować i wgrać do naszego NodeMCU. Po uruchomieniu mikrokontrolera z nowym oprogramowaniem, system powinien połączyć się z Wi-Fi, a następnie nawiązać połączenie z Adafuit IO. Możemy teraz napiąć nasze mięśnie i sprawdzić, jak zachowuje się system. Z jednej strony zwraca on wartość pomiaru poprzez port szeregowy (prędkość transmisji 115200 bodów), jak pokazano na **rysunku 2c**. Taka sama wartość powinna być zaprezentowana w sieci na **dashboardzie** Adafuit IO, jak widzimy na **rysunku 2d**.

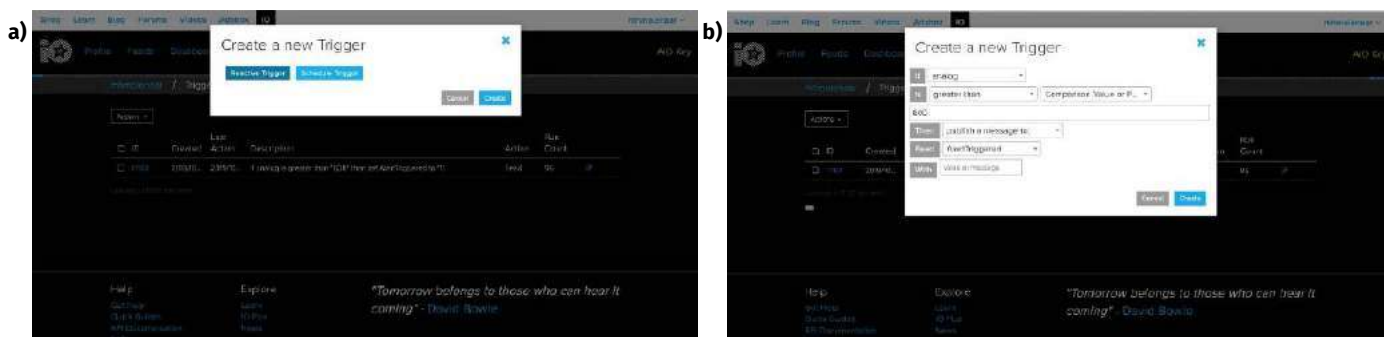
Krok 4: Konfiguracja wyzwalacza w Adafuit IO

Do tej pory tylko wprowadzaliśmy informacje do systemu Adafuit IO. Teraz zaczniemy także z systemu czytać. Skonfigurujemy system tak, że po wykryciu przekroczenia wartości powyżej ustalonego progu czujnika MyoWare, pasek ledowy RGB wykona predefiniowaną

akcję. W tym celu musimy skonfigurować wyzwalacz (*trigger*) w Adafuit IO.

W pierwszej kolejności konfigurujemy dodatkowy kanał w Adafuit IO. Musimy utworzyć inny kanał, aby móc uzyskać dane wyjściowe. W tym celu tworzymy (analogicznie jak poprzednio) kanał o nazwie *AlertTriggered*. Teraz można utworzyć wyzwalacz. W menu Adafuit IO, obok nagłówek *Feed* i *Dashboard*, znajdziemy przycisk „Trigger”, skorzystajmy z niego. Pozwoli to utworzyć nowy wyzwalacz. W otwartym w ten sposób menu (**rysunek 3a** oraz **3b**) najpierw wybieramy, czy ma być to wyzwalacz reaktywny, czy okresowy. Wybieramy reaktywny i podajemy wszystkie dane, jakie są wymagane (patrz rysunek 3b). W menu wyzwalacza musimy skonfigurować szereg parametrów:

- wyberamy kanał, w którym znajdują się dane analogowe z naszego sensora,
- jako wyzwalacz musimy wybrać komparator i uruchamiać go, gdy wartość przekroczy ustawiony próg,
- wprowadzamy bezpieczną wartość progu, na przykład 600,
- aby odebrać wiadomość w Arduino, musimy ustawić „publish a message to”,
- musimy wskazać kanał, w którym znajdzie się wyzwalacz, wybieramy utworzony wcześniej *AlertTriggered*.



Rysunek 3. Konfiguracja wyzwalacza w Adafuit IO

Teraz możemy nacisnąć „utwórz”, aby wygenerować wyzwalacz.

Krok 5: Sterowanie diodami LED RGB

Aby sterować diodami Neopixel, musimy nasłuchiwać kanału *AlertTriggered*, który przed chwilą utworzyliśmy. NodeMCU będzie sterowało paskiem LED-owym, który potrzebuje jednej linii cyfrowej i zasilania 5 V, w związku z czym podłączamy go w następujący sposób:

1. Lewe pole paska LED do zasilania, najlepiej 5 V, ale powinny one działać z zasilaniem 3,3 V z NodeMCU.
2. Środkowe pole – DIN (wejście danych) podłączamy do portu D5 NodeMCU.
3. Prawe pole paska LED podłączamy do masy.

Do sterowania systemem zmodyfikowano szkic z przykładu. Pełny kod programu pokazany został na **listingu 1**. Kod ten współpracuje z plikiem nagłówkowym *config.h*, w którym znajdują się dane konfiguracyjne dla programu. Po skompilowaniu i załadowaniu szkicu do NodeMCU możemy uruchomić system. Połączy się z Adafruit i zacznie pracę. Po pięciokrotnym przekroczeniu progu pomiaru z sensora (ustaliliśmy wartość 600) uruchomi się alarm sygnalizowany przez diody na pasku LED-owym Neopixel.

Nikodem Czechowski, EP

Źródło:

1. <http://bit.ly/32Psh1m>

Listing 1.

```

/***** Konfiguracja *****/
#include „config.h”
#include „Adafruit_NeoPixel.h”

/***** Kod programu zaczyna się tutaj *****/

//definicje dla paska LED
#define PIXEL_PIN          D5
#define PIXEL_COUNT        10
#define PIXEL_TYPE          NEO_GRB + NEO_KHZ800

//pin analogowy dla sensora
#define MUSCLESSENSOR_PIN A0

//Uruchamia alert po X przekroczeniu wartości w wyzwalaczu
#define ALERT_AFTER_TIMES 6

//Stan sensora MyoWare
int current = 0;
int last = -1;

//Stan alarmu
bool alertActive = false;
int alertLevel = 0;
int secondsPassed = 0;

// Neopixel
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(
  PIXEL_COUNT, PIXEL_PIN, PIXEL_TYPE);

//Podłączenie do kanału pod nazwą „analog”
AdafruitIO_Feed *analog = io.feed(„analog”);

//Podłączenie do wyzwalacza, który wysyła wartość 1,
//gdzy zostanie uruchomiony
AdafruitIO_Feed *alertTriggered = io.feed(„AlertTriggered”);

void setup() {
  //Uruchomienie konsoli szeregowej
  Serial.begin(115200);

  //oczekiwanie, aż konsola uruchomi się
  while(! Serial);

  //połączenie z io.adafruit.com
  Serial.print(„Connecting to Adafruit IO”);
  io.connect();

  // obsługa wyzwalacza przychodzącego
  alertTriggered->onMessage(handleTrigger);

  //oczekiwanie na połączenie z Adafruit IO
  while(io.status() < AIO_CONNECTED) {
    Serial.print(„.”);
    delay(500);
  }

  //połączenie
  Serial.println();
  Serial.println(io.statusText());

  //inicjalizacja Neopixel
  pixels.begin();
  setToWhite();
}

void loop() {
  //io.run(); jest wymagane przez wszystkie szkice
  //powinno być zawsze na starcie głównej pętli programu
  //służy do połączenia z io.adafruit.com.
  io.run();

  //aktualny stan sensora
  current = analogRead(MUSCLESSENSOR_PIN);

  //wyjście, jeśli wartość się nie zmieniła
  if(current == last)
    return;

  //zapisanie obecnej wartości zmierzonej
  //z sensora do kanału analogowego
  Serial.print(„sending -> ”);

  Serial.println(current);
  analog->save(current);

  // zapisujemy wartość sensora
  last = current;

  // poczekaj dwie sekundy
  delay(2000);

  testIfShouldReset();

  // obsługa wyzwalacza
  void handleTrigger(AdafruitIO_Data *data) {
    if(data->toInt() == 1) {
      alertLevel++;
      Serial.println(„Alert level up:”);
      Serial.println(alertLevel);

      //Uruchomienie alarmu w momencie, gdy wyzwalacza
      //zgłosił się więcej niż ALERT_AFTER_TIMES
      if(alertLevel >= ALERT_AFTER_TIMES) {
        alertOn();
      }
    }
  }

  //Uruchom alarm, wysyłając informacje do diod LED
  void alertOn() {
    alertActive = true;
    setToRed();
    Serial.println(„Alert on”);
  }

  // Wyłącz alarm
  void alertOff() {
    //ściemnianie diod LED poprzez obniżanie jasności,
    //jaka jest ustawiona
    for(int brightness = 255; brightness > 0; brightness -= 5) {
      pixels.setBrightness(brightness);
      pixels.show();
      delay(100);
    }

    //Reset alert variables
    alertLevel = 0;
    alertActive = false;
    Serial.println(„Alert off”);

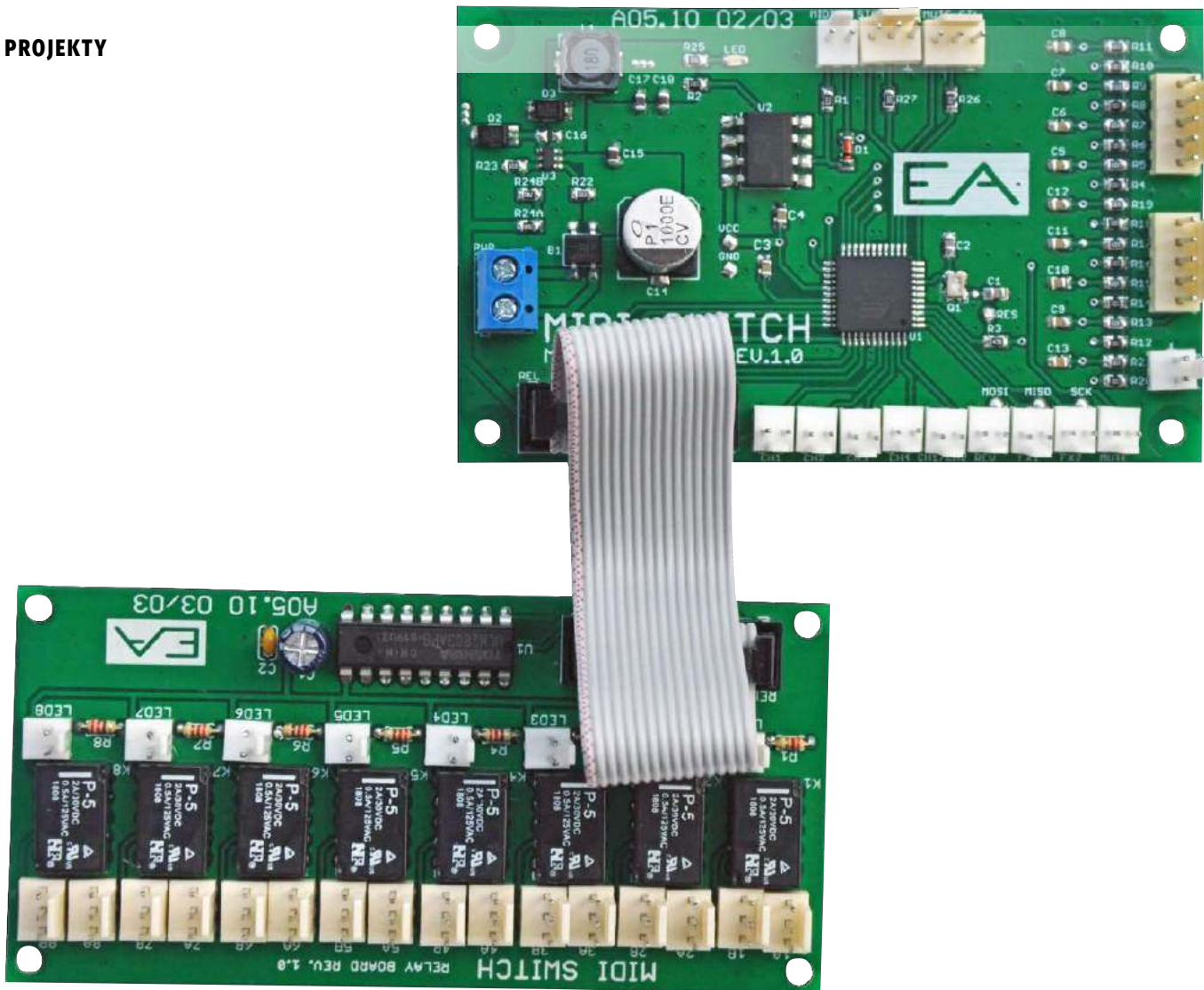
    setToWhite();
  }

  //Zapalenie diod na biało
  void setToWhite() {
    for (int i = 0; i < PIXEL_COUNT; ++i) {
      pixels.setPixelColor(i, 255, 255, 255);
    }
    pixels.setBrightness(255);
    pixels.show();
  }

  //Zapalenie diod na czerwono
  void setToRed() {
    for (int i = 0; i < PIXEL_COUNT; ++i) {
      pixels.setPixelColor(i, 255, 0, 0);
    }
    pixels.setBrightness(255);
    pixels.show();
  }

  //Wyłączenie alarmu po jednej minucie
  void testIfShouldReset() {
    if(alertActive) {
      secondsPassed += 2;
      // 60 to liczba sekund, możemy to zmienić
      if(secondsPassed >= 60) {
        alertOff();
      }
      //dane o liczbie sekund, jaka upłynęła od alarmu,
      //przesyłana jest przez port szeregowy
      Serial.println(„Seconds passed:”);
      Serial.println(secondsPassed);
    } else {
      secondsPassed = 0;
    }
  }
}

```



MIDIswitch

– przełącznik MIDI do wzmacniacza lampowego

W praktyce każdego elektronika konstruktora są takie projekty, gdzie mariaż starego i nowego daje zupełnie nową jakość. Dotyczy to zwłaszcza tych urządzeń, w których implementację zaangażowanych jest kilka osób posiadających doświadczenie na różnych płaszczyznach sprzętowych. Tak właśnie jest za każdym razem, gdy do współpracy zapraszam mojego kolegę, który ma duże doświadczenie przy naprawach i projektowaniu układów z szeroko pojętej elektroniki estradowej. Tym razem postanowiliśmy wstrzyknąć sporą dawkę świeżości we wzmacniacze lampowe. Z jednej strony może to nieco dziwić, gdyż czasy największej popularności urządzeń lampowych już dawno minęły, z drugiej strony jest kilka grup urządzeń, w których układy lampowe są nadal bardzo popularne.

Wzmacniacze lampowe są bardzo popularne jako wzmacniacze gitarowe. Większość dobrych gitarzystów (zwłaszcza bluesowych i rockowych) używa wzmacniaczy tego typu.

Powód jest prosty – przesterowane lampy dają znacznie przyjemniejszy dźwięk niż układy tranzystorowe. Z tego powodu nadal powstaje wiele wzmacniaczy lampowych,

Dodatkowe materiały do pobrania ze strony www.media.avt.pl

W ofercie AVT* AVT-5724

Podstawowe parametry:

- cztery przełączniki działające na zasadzie „jeden z czterech”,
- cztery przełączniki działające na zasadzie „dowolny z czterech”,
- automatyczne wyciszenie przy zmianie kanału wejściowego,
- sterowanie poprzez interfejs MIDI,
- praca jako sterownik wzmacniacza lub jako tzw. looper,
- zasilanie z przedziału 8...20 V, max. 200 mA.

Projekty pokrewne na www.media.avt.pl:

AVT-5692	MidiCtrl – prosty sterownik MIDI (EP 8/2019)
AVT-5594	Konwerter MIDI-SPI (EP 7/2017)
AVT-960	Płyta ewaluacyjna i klawiatura muzyczna MIDI (EP 12/2006)
AVT-842	Asynchroniczny konwerter RS232-MIDI (EP 12/1999)

Uwagi: Elektroniczne zestawy do samodzielnego montażu.

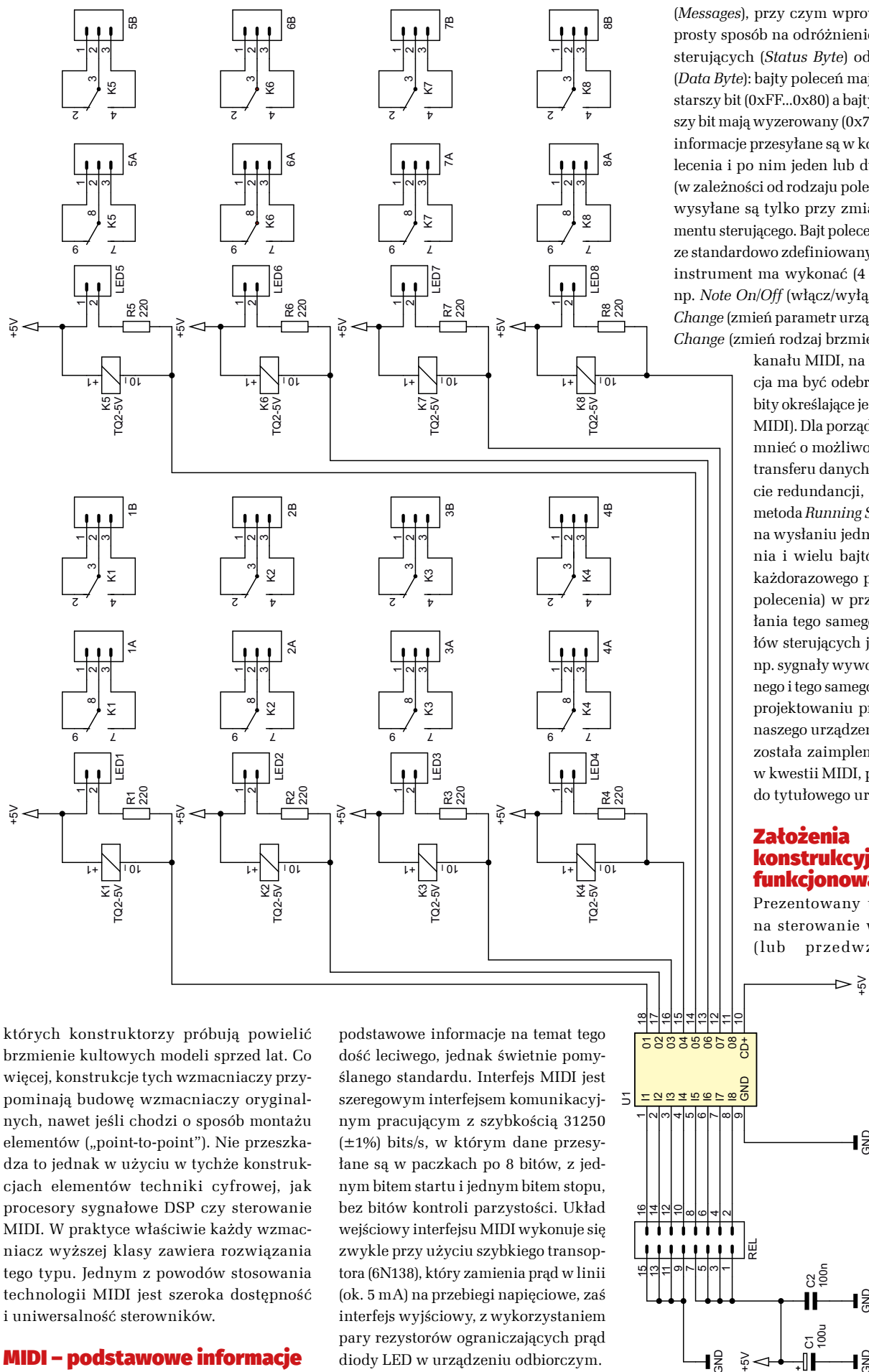
Wymagana umiejętność lutowania!

Podstawową wersją zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wzlutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu.

Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wzlutowane w płytkę PCB)
- wersja [A] – płytkę drukowaną bez elementów i dokumentacji Kity w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- wersja [A+] – płytkę drukowaną [A] + zaprogramowany układ [UK] i dokumentacja
- wersja [UK] – zaprogramowany układ

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! <http://sklep.avt.pl>. W przypadku braku dostępności na <http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.



(Messages), przy czym wprowadzono bardzo prosty sposób na odróżnienie bajtów poleceń sterujących (*Status Byte*) od bajtów danych (*Data Byte*): bajty poleceń mają ustawiony najstarszy bit (0xFF...0x80) a bajty danych najstarszy bit mają wyzerowany (0x7F...0x00). Zwykle informacje przesyłane są w kolejności: bajt polecenia i po nim jeden lub dwa bajty danych (w zależności od rodzaju polecenia). Polecenia wysyłane są tylko przy zmianie danego elementu sterującego. Bajt polecenia określa jedną ze standardowo zdefiniowanych funkcji, którą instrument ma wykonać (4 najstarsze bity), np. *Note On/Off* (włącz/wyłącz nutę), *Control Change* (zmień parametr urządzenia), *Program Change* (zmień rodzaj brzmienia) oraz numer kanału MIDI, na którym informacja ma być odebrana (pozostałe 4 bity określające jeden z 16 kanałów MIDI). Dla porządku należy wspomnieć o możliwości ograniczenia transferu danych poprzez usunięcie redundancji, z której korzysta metoda *Running Status*. Polega ona na wysłaniu jednego bajta polecenia i wielu bajtów danych (bez każdorazowego ponawiania bajta polecenia) w przypadku przesyłania tego samego rodzaju sygnałów sterujących jeden za drugim, np. sygnały wywołane zmianą jednego i tego samego regulatora. Przy projektowaniu programu obsługi naszego urządzenia metoda ta nie została zaimplementowana. Tyle w kwestii MIDI, powróćmy zatem do tytułowego urządzenia.

Założenia konstrukcyjne i sposób funkcjonowania

Prezentowany układ pozwala
na sterowanie wzmacniaczem
(lub przedwzmacniaczem)

kórych konstruktorzy próbują powielić brzmienie kultowych modeli sprzed lat. Co więcej, konstrukcje tych wzmacniaczy przypominają budowę wzmacniaczy oryginalnych, nawet jeśli chodzi o sposób montażu elementów („point-to-point”). Nie przeszkadza to jednak w użyciu w tychże konstrukcjach elementów techniki cyfrowej, jak procesory sygnałowe DSP czy sterowanie MIDI. W praktyce właściwie każdy wzmacniacz wyższej klasy zawiera rozwiązania tego typu. Jednym z powodów stosowania technologii MIDI jest szeroka dostępność i uniwersalność sterowników.

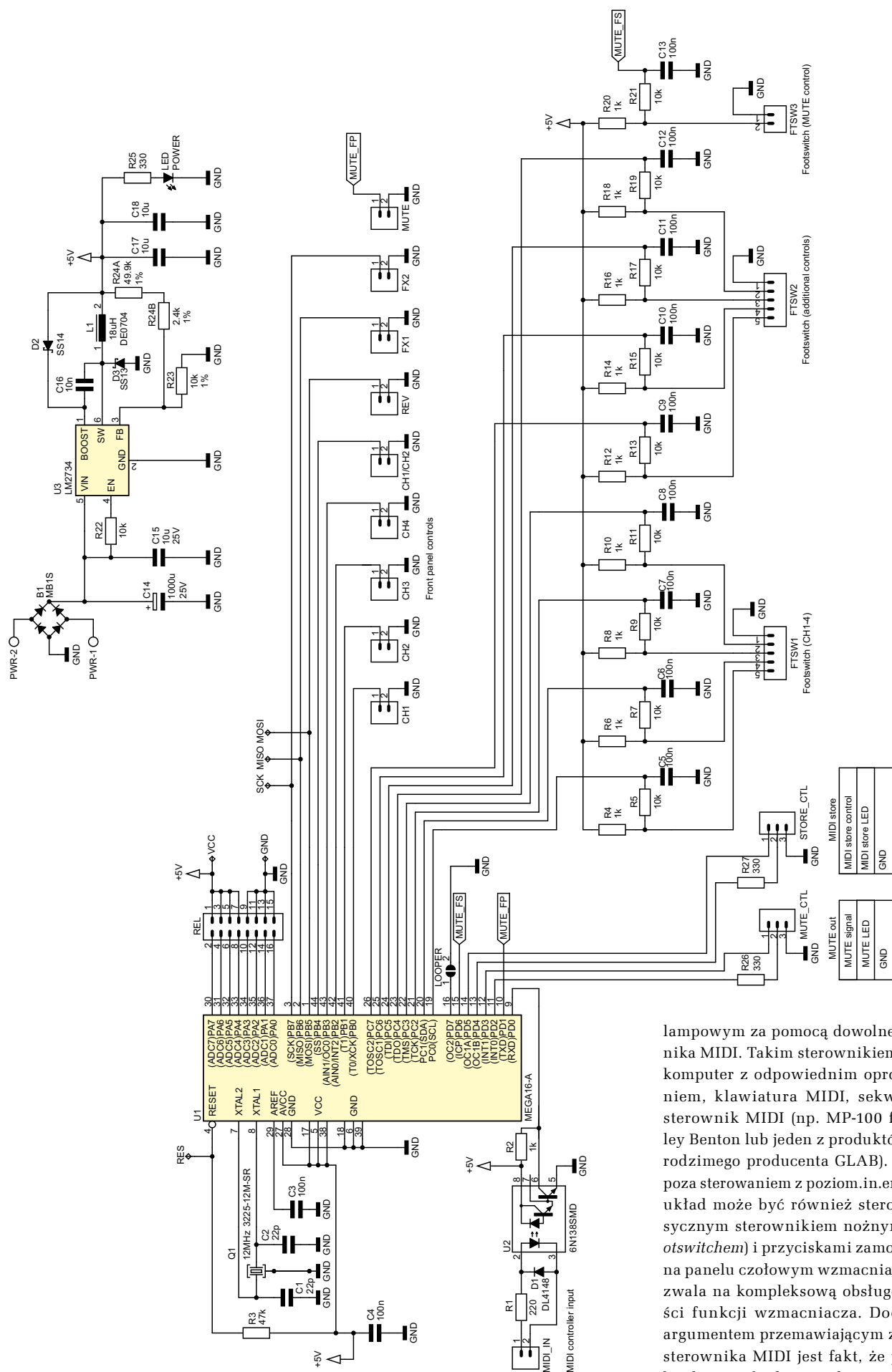
MIDI – podstawowe informacje

Zanim przejdziemy do opisu prezentowanego urządzenia, należy przypomnieć sobie

podstawowe informacje na temat tego dość leciwego, jednak świetnie pomyślanego standardu. Interfejs MIDI jest szeregowym interfejsem komunikacyjnym pracującym z szybkością 31250 ($\pm 1\%$) bits/s, w którym dane przesyłane są w paczkach po 8 bitów, z jednym bitem startu i jednym bitem stopu, bez bitów kontroli parzystości. Układ wejściowy interfejsu MIDI wykonuje się zwykle przy użyciu szybkiego transportora (6N138), który zamienia prąd w linii (ok. 5 mA) na przebiegi napięciowe, zaś interfejs wyjściowy, z wykorzystaniem pary rezystorów ograniczających prąd diody LED w urządzeniu odbiorczym.

W standardzie MIDI dane przesyłane są grupowo w formie tzw. komunikatów

Rysunek 1. Schemat ideowy części wykonawczej



Rysunek 2. Schemat ideowy części sterującej

lampowym za pomocą dowolnego sterownika MIDI. Takim sterownikiem może być komputer z odpowiednim oprogramowaniem, klawiatura MIDI, sekwencer lub sterownik MIDI (np. MP-100 firmy Harley Benton lub jeden z produktów naszego rodzimego producenta GLAB). Co więcej, poza sterowaniem z poziomu.in.erfejsu MIDI układ może być również sterowany klasycznym sterownikiem nożnym (tzw. *footswitchem*) i przyciskami zamocowanymi na panelu czołowym wzmacniacza. To pozwala na kompleksową obsługę większości funkcji wzmacniacza. Dodatkowym argumentem przemawiającym za użyciem sterownika MIDI jest fakt, że przy coraz bardziej rozbudowanych wzmacniaczach przełączniki nożne muszą być odpowiednie

Wykaz elementów:**Układ wykonawczy**

Rezystory: (obudowa miniaturowa 1/8 W)
R1...R8: 220 Ω

Kondensatory:

C1: elektrolityczny 100 µF/10 V (raster 0,1")
C2: ceramiczny 100 nF (raster 0,1")

Półprzewodniki:

U1: ULN2803 (obudowa DIL18)

Inne:

K1...K8: przełącznik miniaturowy TQ2-5 V
LED1...LED8: gniazdo męskie proste 2 piny
typu NS25-W2P (raster 0,1")
SL1...SL16: gniazdo męskie proste 3 piny
typu NS25-W3P (raster 0,1")
REL: gniazdo IDC Z-WS16 proste do druku
2x8 pinów (raster 0,1")

Układ sterujący

Rezystory: (obudowy SMD 0805)

R1: 220 Ω
R2, R4, R6, R8, R10, R12, R14, R16, R18,
R20: 1 kΩ
R3: 47 kΩ
R5, R7, R9, R11, R13, R15, R17, R19,
R21...R23: 10 kΩ
R24A: 49,9 kΩ 1%
R24B: 2,4 kΩ 1%
R25...R27: 330 Ω

Kondensatory: (obudowy SMD 0805,

jeśli nie zaznaczono inaczej)

C1, C2: ceramiczny 22 pF
C3...C13: ceramiczny X7R 100 nF
C14: elektr. niskoimpedancyjny SMD
1000 µF/25 V
C15, C17, C18: ceramiczny X7R 10 µF
C16: ceramiczny X7R 10 nF

Półprzewodniki:

U1: ATmega16A (TQFP44)
U2: 6N138S-L (DIL8-SMD)
U3: LM2734 (SOT23-6)
D1: dioda DL4148 (MINIMELF)
D2: dioda SS14 (DO-214AC)
D3: dioda SS13 (DO-214AC)
B1: mostek prostowniczy SMD typu MB1S
(S0IC-4)
LED: dioda LED SMD czerwona (SMD 0805)

Inne:

Q1: rezonator kwarcowy SMD 12 MHz typu
(obudowa 3,2x2,5 mm)
L1: dławik mocy SMD 18 µH typu FERROCORE
DE0704-18
PWR: złącze śrubowe AK500/2 (raster
0,1")
CH1...CH4, CH1/CH2, FX1, FX2, REV, MUTE,
MIDI_IN, FTSW3 – gniazdo męskie proste
2 piny typu NS25-W2P (raster 0,1")
MUTE_CTL, STORE_CTL: gniazdo męskie
proste 3 piny typu NS25-W3P (raster
0,1")
FTSW1, FTSW2: gniazdo męskie proste
5 pinów typu NS25-W5P (raster 0,1")
REL: gniazdo IDC Z-WS16 proste do druku
2x8 pinów (raster 0,1")

dla danego modelu przez co zwiększa się koszt całego systemu. Sterownik MIDI pozwala niejako rozwiązać ten problem. Jakby tego było mało, programy typu DAW (producentów takich jak Ableton, Avid, Cakewalk, Steinberg) pozwalają w sesji nagraniowej utworzyć ścieżkę z poleceniami MIDI, a to z kolei pozwala zautomatyzować również przełączanie kanałów i efektów we wzmacniaczu użytym w nagraniu. To oznacza, że ustawienia kanałów i efektów można zapisać w sesji nagraniowej. W przypadku konieczności powtórzenia nagrania może być to bardzo przydatne.

Opisywane urządzenie z założenia ma być uniwersalne, więc jego projekt powinien uwzględniać możliwość implementacji w typowych wzmacniaczach lampowych. Jednocześnie powinien pozwalać na zastosowanie w najbardziej rozbudowanych konstrukcjach. Takiego

rodzaju wzmacniacze mają do czterech kanałów, które można dowolnie przełączać i jednocześnie mogą mieć pogłos (sprężynowy lub elektroniczny), jedną lub dwie pętle efektów (z których każda może być oddzielnie włączana lub wyłączana), efekty typu chorus, distorion, itp. Stosownie do tych wymagań założenia projektowe przy konstruowaniu niniejszego sterownika zostały zdefiniowane następująco:

- możliwość zamontowania sterownika w istniejącym wzmacniaczu przy minimalnych modyfikacjach oryginalnego schematu urządzenia,
- cztery przełączniki działające na zasadzie „jeden z czterech” (K1 do K4) pozwalające na wybranie jednego z czterech kanałów,
- cztery przełączniki działające na zasadzie „dowolny z czterech” (K5 do K8) pozwalające na niezależne sterowanie dodatkowymi możliwościami wzmacniacza takimi, jak: pogłos, pętle efektów, przester, chorus, itp.,
- możliwość wyciszania wzmacniacza (np. aby gitarzysta mógł nastroić gitarę) przyciskiem na panelu czołowym lub *footswitchem*.
- automatyczne wyciszanie wzmacniacza w momencie przełączania kanałów. Pozwala to uniknąć niepożądanych efektów dźwiękowych w głośniku wzmacniacza w chwili przełączania kanałów,
- możliwość zapisania bieżącej konfiguracji pracy wzmacniacza w nieulotnej pamięci sterownika, np. w celu jej późniejszego przywołania jednym rozkazem MIDI.

Układ jest bardzo elastyczny – pozwala na sterowanie bardzo skomplikowanymi wzmacniaczami, ale równocześnie można go użyć do kontroli bardzo prostego urządzenia składającego się jedynie z dwóch kanałów. Aby to zrobić, można zastosować pierwsze dwa przełączniki z pierwszej grupy lub pojedynczy przełącznik z drugiej grupy. Zaletą tego drugiego rozwiązania jest możliwość przełączania dwóch kanałów za pomocą wyłącznie jednego przycisku. Co więcej, implementując nasz sterownik w prostym wzmacniaczu, można zwyczajnie nie montować części elementów, które nie będą użyte, co zmniejszy sumaryczny koszt systemu. Dodatkowo nasz układ można zastosować do budowy zupełnie innego urządzenia, niż pierwotnie zakładano. Można z niego zbudować tzw. *looper* – czyli przełącznik pętli efektów. Jedyną wymaganą zmianą w takim wypadku jest umieszczenie przełącznika w oddzielnej obudowie, z zewnętrznym zasilaczem i dodatkowymi gniazdami podłączeniowymi. Pozwoli to na wybranie jednej z 4 pętli (przełączniki K1 do K4)

i dowolnej kombinacji z pozostałych pętli (przełączniki K5 do K8). Oprogramowanie sterownika pozwala ponadto zmienić zasadę działania urządzenia, a mianowicie daje możliwość wybierania dowolnej kombinacji przełączników (przełączniki K1 do K8). Funkcjonalność tą wybieramy, zwierając pole lutownicze *looper* na płytce głównej sterownika (warstwa BOTTOM). W przypadku pracy przełącznika w trybie *looper* warto przyrzeć się istniejącym konstrukcjom, chociażby FXL8 Pro firmy Harley Benton (przeszukując witryny sklepów takich jak www.thomman.de, znajdziemy wiele więcej urządzeń tego typu).

Warto podkreślić, że nazwy symboliczne przypisane do wejść z drugiej grupy przycisków na panelu przednim (sterujące przełącznikami od K5 do K8) są umowne i sugerują wyłącznie, do czego dany przełącznik może zostać użyty. Nie ma przecież powodu, aby nie użyć wejścia FX2 do sterowania włączaniem dowolnej, innej funkcji wzmacniacza, jak np. efekt chorus (jeśli wzmacniacz takowy ma, a nie ma drugiej pętli efektów). Tyle, jeśli chodzi o założenia konstrukcyjne i sposób funkcjonowania naszego sterownika.

Budowa

Sterownik MIDISwitch składa się z dwóch płytek: wykonawczej, z przełącznikami i sterującej, z procesorem odbierającym i interpretującym polecenia MIDI. Zaczniemy od części wykonawczej, której schemat ideowy pokazano na **rysunku 1**. Układ wykonawczy jest bardzo prosty i składa się w zasadzie wyłącznie z bufora sterującego przełącznikami i samych przełączników. Dodatkowo załączenie każdego z przełączników może być sygnalizowane zaświeceniem diody LED, którą można umieścić na przednim panelu modyfikowanego wzmacniacza. Jest to zalecane rozwiązanie, gdyż użytkownik musi wiedzieć, który kanał jest aktualnie wybrany.

Schemat ideowy części sterującej pokazano na **rysunku 2**. Zaprojektowano bardzo prosty system mikroprocesorowy z użyciem

REKLAMA

Specjalistyczne szkolenia
dla elektroników
i automatyków



techdays@techdays.pl
TECHDAYS.PL

certyfikowany
PARTNER
SZKOLENIOWY

CERTYFIKOWANY
PARTNER
SZKOLENIOWY

popularnego mikrokontrolera firmy Microchip (dawniej Atmel) o oznaczeniu ATmega16, który taktowany rezonatorem kwarcowym o częstotliwości 12 MHz odpowiada za obsługę interfejsu MIDI (obwód wejściowy z wykorzystaniem transoptora 6N138), obsługę przycisków na panelu czołowym urządzenia (gniazda: CH1...CH4, CH1/CH2, REV, FX1, FX2, MUTE), obsługę przycisków nożnych tzw. *footswitchy* (gniazda: FTSW1...FTSW3), obsługę wyjść przełączników (złącze REL) oraz obsługę dodatkowych wejść/wyjść MUTE_CTL, STORE_CTL. Obsługa wszelkich wejść odbywa się z użyciem programowego mechanizmu eliminacji drgań styków, co jest wyjątkowo istotne dla wejść *footswitchy*, gdzie zintegrowano dodatkowe rozwiązanie sprzętowe w postaci obwodów RC (należy pamiętać, że kabel łączący przełącznik nożny z gniazdem może mieć długość nawet kilkunastu metrów). Wybór tego konkretnego modelu mikrokontrolera nie był podyktowany komplikacją programu obsługi aplikacji, bo ten jest dość prosty, a wynikał wyłącznie z konieczności obsługi wielu sygnałów wejściowych i wyjściowych a zatem koniecznych portów I/O. Zastosowanie rezonatora kwarcowego, jako źródła taktowania mikrokontrolera wynikało z kolei z potrzeby zapewnienia dużej dokładności prędkości transmisji MIDI.

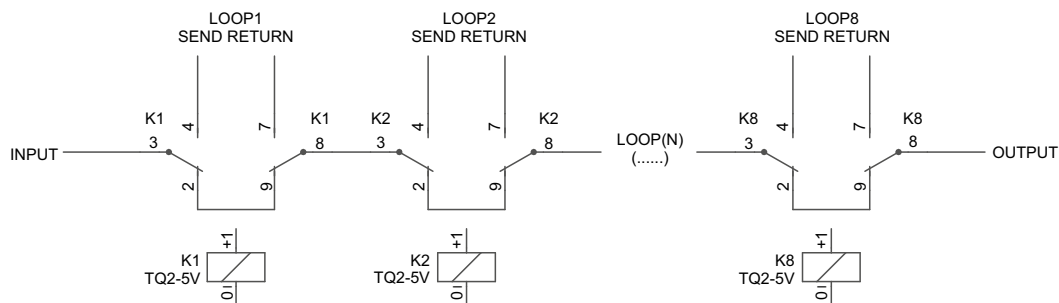
W zakresie części sterującej urządzeniem, zaprojektowano również kompletny układ zasilacza z wykorzystaniem scalonej przetwornicy LM2734 w konfiguracji step-down. Zintegrowany zasilacz jest istotną częścią niniejszego sterownika. Nie można przecież oczekiwać, że we wzmacniaczu dostępne będzie napięcie +5 VDC do zasilania sterownika, więc przyjęto założenie, że układ zasilany będzie z uzwojenia żarzenia lamp (6,3 V lub 12,6 V napięcia zmiennego). Szacowany pobór prądu sterownika jest rzędu 100...200 mA (może zależeć od użytych diod LED), więc uzwojenie żarzenia powinno być w stanie dostarczyć prąd o takiej wartości. **Co szczególnie istotne, masa sterownika nie może być połączona z masą wzmacniacza, gdyż powodowałoby to zwarcie uzwojenia żarzenia wzmacniacza.** Jeśli sterownik nasz chcielibyśmy zasilac z zasilacza należy pamiętać, że zakres napięć wejściowych pokrywa

przedział od 8 V do 20 V, zaś sama przetwornica teoretycznie może dostarczyć prąd rzędu 1 A, ale jak wspomniano, potrzebny prąd jest znacznie mniejszy. Wracając zaś do samego sterownika, należy pamiętać, że aby wzmacniacz można było sterować z dowolnej grupy wejść, przełączniki, zarówno te użyte w przełączniku nożnym, jak i na przednim panelu wzmacniacza, powinny być chwilowe.

Funkcje dodatkowe

Urządzenie dysponuje dodatkowymi funkcjami, które wymagają kilku słów komentarza.

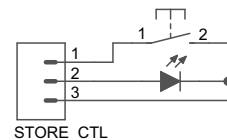
Pierwsza z nich pozwala na zapisanie bieżącej konfiguracji urządze-



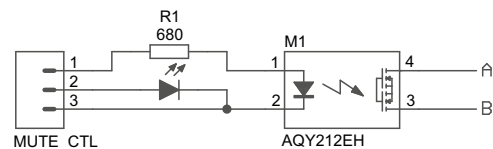
Rysunek 5. Sposób podłączenia sterownika w wersji looper

nia (stanu przełączników), by można ją było przywołać z pomocą polecenia MIDI. Funkcjonalność tę uzyskujemy dzięki implementacji wejścia STORE_CTL, do którego podłączamy chwilowy przycisk sterujący oraz diodę sygnalizacyjną LED. Naciśnięcie wspomnianego przycisku powoduje zapisanie bieżącej konfiguracji urządzenia i sygnalizację tego stanu, przy pomocy diody LED podłączonej do tego wejścia. Sposób podłączenia przycisku sterującego i diody LED do gniazda STORE_CTL pokazano na **rysunku 3**.

Kolejną, przydatną funkcjonalnością jest możliwość wyciszenia wzmacniacza (np. aby pozwolić gitarzyście nastroić gitarę). Dostęp do niej uzyskujemy dzięki wykorzystaniu wyjścia MUTE_CTL, które integruje w sobie dwa sygnały: sygnał sterujący wyciszaniem (MUTE-signal) i wyjście sygnalizacyjne diody LED (MUTE LED). Działanie tej funkcjonalności przedstawia się następująco: pojawienie się niskiego poziomu sygnału na wejściu MUTE (panel przedni) lub FTSW3 (przełącznik nożny) powoduje pojawienie się wysokiego stanu na wyjściu MUTE_CTL i jednocześnie włączenie diody LED podłączonej do tego wyjścia. Wysoki stan na wyjściu MUTE_CTL pozwala na użycie przełącznika typu MOSFET (np. AQY212EH mającego bardzo małą oporność w stanie włączonym) pozwalającego na wyciszenie wzmacniacza (poprzez zwarcie sygnału do masy). Przełącznik taki powinien być zamontowany jak najbliżej miejsca we



Rysunek 3. Sposób podłączenia przycisku sterującego i diody LED do gniazda STORE_CTL



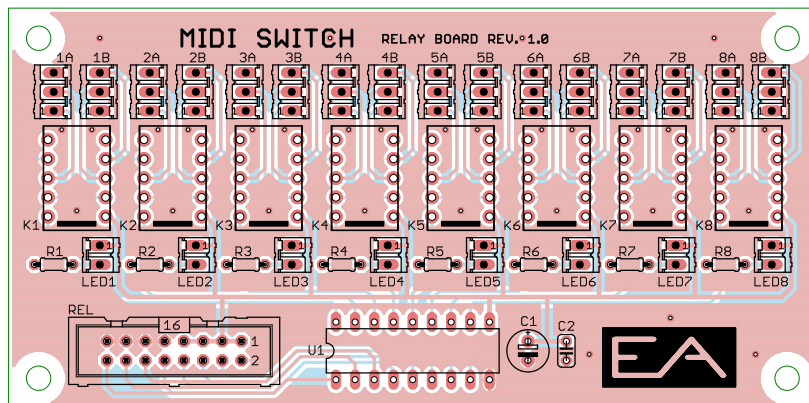
Rysunek 4. Sposób podłączenia do gniazda MUTE_CTL

wzmacniaczu, gdzie zwarcie sygnału spowoduje jego wyciszenie. Właśnie z tego powodu nie jest on zamontowany na obwodzie drukowanym naszego sterownika, a powinien być raczej zamontowany na małej płytce uniwersalnej i umieszczony np. blisko wejścia końcówki mocy (w przypadku wzmacniacza) lub wyjścia przedwzmacniacza (w przypadku, gdy sterownika użyto do sterowania przedwzmacniaczem lampowym). Sposób podłączenia gniazda MUTE_CTL pokazano na **rysunku 4**. W przypadku urządzenia lampowego przełącznik taki powinien zwracać do masy siatkę lampy. Może to być lampka w przedwzmacniaczu, we wzmacniaczu pogłosu lub na wejściu końcówki mocy. Ponieważ wspomniany przełącznik ma bardzo małą oporność w stanie ON, może on również zwracać wyjście przedwzmacniacza lub stopień wzmocnienia pętli efektów. Jednocześnie możliwe jest zastosowanie kilku takich układów (wejścia połączone równolegle), tak aby uzyskać efekt wyciszenia w kilku miejscach przedwzmacniacza. Należy wtedy dobrać odpowiednio opornik ustalający prąd sterujący przełącznikiem (tak by dla jednego przełącznika był to prąd rzędu 5 mA i aby nie przekroczyć wydajności prądowej portu procesora).

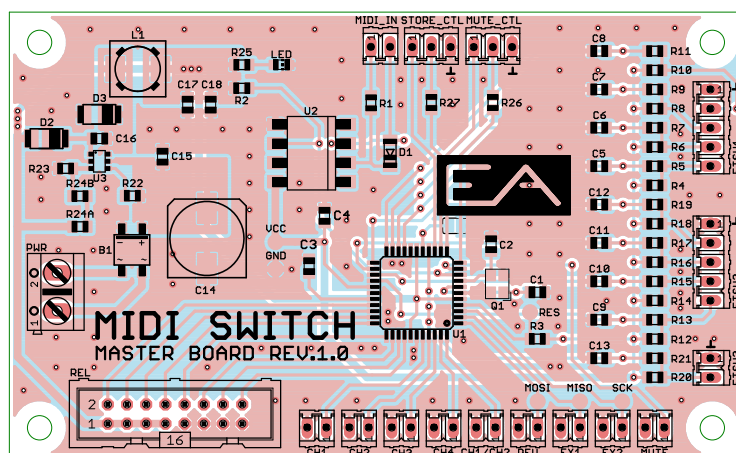
Ostatnią, dodatkową funkcjonalnością sterownika MIDIswitch jest możliwość budowy zupełnie innego urządzenia niż sterownik wzmacniacza lampowego, a mianowicie można z niego zbudować tzw. *looper*

Ustawienia Fusebitów (ważniejszych):

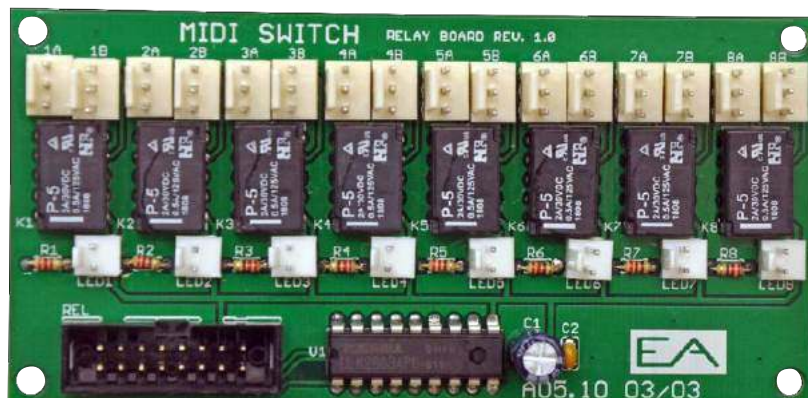
```
CKSEL3..0: 1111
SUT1..0: 11
CKOPT: 0
OCDEN: 1
JTAGEN: 1
```



Rysunek 6. Schemat montażowy części wykonawczej



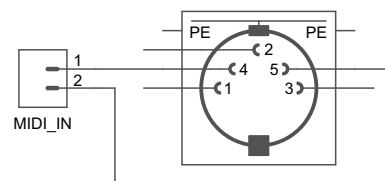
Rysunek 7. Schemat montażowy części sterującej



Fotografia 9. Widok zmontowanego obwodu drukowanego modułu wykonawczego



Fotografia 10. Widok zmontowanego obwodu drukowanego modułu sterującego



Rysunek 8. Sposób podłączenia gniazda połączeniowego MIDI_IN z gniazdem MIDI

– czyli przełącznik efektów, często używany przez gitarzystów. W takim przypadku należy zamontować przełącznik w oddzielnej obudowie, podłączyć do niego zewnętrzny zasilacz (można użyć typowego zasilacza 9 V stosowanego do efektów). Obudowę należy wyposażyć w gniazda typu Jack dla wejścia i wyjścia instrumentu oraz do podłączenia pętli efektów (2 gniazda dla jednej pętli). Jak napisano wcześniej, standardowe oprogramowanie przełącznika pozwala na wybranie jednej z 4 pętli (przełączniki K1 do K4) i dowolnej kombinacji z pozostałych pętli (przełączniki K5 do K8). Opcjonalnie, po zwarceniu punktów lutowniczych oznaczonych jako *looper*, oprogramowanie sterownika pozwala również zmienić zasadę działania urządzenia i wybierać dowolną kombinację pętli (przełączniki K1 do K8). Sposób podłączenia sterownika w wersji *looper* pokazano na rysunku 5.

Montaż

Montaż urządzenia rozpoczynamy od obwodu drukowanego części wykonawczej, który pokazano na rysunku 6. Montaż części wykonawczej sterownika rozpoczynamy od określenia jego funkcjonalności, czyli liczby montowanych przełączników i gniazd. Należy również zdecydować, czy montujemy gniazda wyjściowe (pozwalające na wygodne odłączanie wejść-wyjść i diod LED) lub czy zdejemy się na połączenia lutowane (przewody połączeniowe lutowane bezpośrednio do płytki sterownika). Ostatnie rozwiązanie jest mniej elastyczne, jednak pewniejsze w przypadku drgań wzmacniacza (wzmacniacz postawiony na kolumnie głośnikowej). Po zamontowaniu wybranych

REKLAMA

Specjalistyczne szkolenia dla elektroników i automatyków

STM32

TECHDAYS

techdays@techdays.pl
TECHDAYS.PL

CERTYFIKOWANY PARTNER SZKOLENIOWY

przekazników i gniazd połączeniowych montujemy układ scalony sterownika przekazników, następnie montujemy elementy bierne, zaś na samym końcu gniazdo połączeniowe REL.

Schemat montażowy części sterującej naszego urządzenia pokazano na **rysunku 7**. Zaprojektowano dość zwarty obwód drukowany ze zdecydowaną przewagą elementów SMD. Montaż rozpoczynamy od przylutowania mikrokontrolera, następnie lutujemy pozostałe półprzewodniki, dalej elementy pasywne a na samym końcu montujemy elementy przewlekane, czyli wszelkiego rodzaju gniazda przyłączeniowe (jeśli, podobnie jak dla części wykonawczej, zdecydujemy się na korzystanie z gniazd, a nie połączeń lutowanych – nie dotyczy gniazda REL). W kolejnym kroku należy połączyć układ sterujący z układem wykonawczym, korzystając z odpowiednich gniazd i 16-żyłowej taśmy połączeniowej oraz sprawdzić działanie całości. Poprawnie zmontowane urządzenie nie wymaga żadnych regulacji i powinno działać tuż po włączeniu zasilania.

Jeśli wszystko działa poprawnie, przystępujemy do montażu obu płytek we wzmacniaczu poddawanemu modyfikacji. Płytką sterującą powinna być zamontowana blisko gniazda wejściowego MIDI, które prawdopodobnie będzie zamontowane na tylnym panelu wzmacniacza. Płytką wykonawczą powinna być zamontowana tak, by przewody podłączone do przekazników były jak najkrótsze. Właściwym miejscem są okolice przedwzmacniacza. Oczywiście, w przypadku przełączania sygnału audio zalecane są przewody ekranowane. Masy przewodów (ekrany) powinny być podłączone do masy modyfikowanego wzmacniacza. Sposób podłączenia gniazda połączeniowego MIDI_IN z gniazdem MIDI pokazano na **rysunku 8**.

Widok zmontowanego obwodu drukowanego modułu wykonawczego pokazano na **fotografii 9**, zaś na **fotografii 10** pokazano wygląd zmontowanego obwodu drukowanego modułu sterującego.

Specyfikacja MIDI

Implementując program obsługi aplikacji, chciałem, by sterowanie naszym urządzeniem było jak najprostsze. Zgodnie z tym założeniem domyślnie nasz sterownik odbiera komunikaty MIDI na kanale 1. Oczywiście, ustawienie to może być zmodyfikowane dzięki przesłaniu odpowiedniego komunikatu MIDI i zapamiętaniu w nieulotnej pamięci EEPROM sterownika. Co ważne, wszystkie obsługiwane przez układ MIDIswitch komunikaty MIDI to komunikaty typu *Control Change*, zatem oprócz konieczności przesłania numeru kontrolera Control Change zachodzi potrzeba wysłania również jego wartości. W takim przypadku komunikat MIDI składa się z 3 bajtów:

Tabela 1. Opis kontrolerów MIDI obsługiwanych przez urządzenie MIDIswitch

Rodzaj polecenia	Numer kontrolera	Wartość kontrolera
Wybór kanału CH1 (1 z 4) ⁽¹⁾	81	≥64=ON ⁽¹⁾
Wybór kanału CH2 (1 z 4) ⁽¹⁾	82	
Wybór kanału CH3 (1 z 4) ⁽¹⁾	83	
Wybór kanału CH4 (1 z 4) ⁽¹⁾	84	
Wybór kanału CH1/CH2	85	≥64=ON, <64=OFF
Wybór kanału REV	86	
Wybór kanału FX1	87	
Wybór kanału FX2	88	
Zmiana na kanale CH1/CH2	89	Bez znaczenia. Zmiana stanu na przeciwny (Toggle)
Zmiana na kanale REV	90	
Zmiana na kanale FX1	91	
Zmiana na kanale FX2	92	
Wybór funkcji MUTE	93	≥64=ON, <64=OFF
Zmiana funkcji MUTE	94	Bez znaczenia. Zmiana stanu na przeciwny (Toggle)
Przywrócenie zapisanej konfiguracji	101	Bez znaczenia
Wyczyszczenie zapisanej konfiguracji	102	≥64
Ustawienie aktywnego kanału MIDI	103	0...15
Zapis konfiguracji	104	Bez znaczenia

⁽¹⁾ W przypadku aktywnej funkcji *looper* kontrolery o numerach 81...84 działają w identyczny sposób, jak kontrolery o numerach 89...92, tyle że odnoszą się do kanałów CH1...CH4

Listing 1. Funkcja obsługi przerwania interfejsu USART odpowiedzialna za odbiór komunikatów MIDI (wraz z deklaracją niezbędnych zmiennych)

```
//Definicje zmiennych bufora odbiorczego MIDI
volatile char MIDIBuffer[MIDI_RX_BUF_SIZE];
// indeks oznaczający „głową węża” dla bufora MIDI
volatile uint8_t MIDIBufferHead;
// indeks oznaczający „ogon węża” dla bufora MIDI
volatile uint8_t MIDIBufferTail;
//Liczba odebranych danych MIDI będąca jednocześnie flagą odebrania danych
volatile uint8_t MIDIbytes;
//Definicje bufora odbiorczego MIDI
#define MIDI_RX_BUF_SIZE 64
#define MIDI_RX_BUF_MASK (MIDI_RX_BUF_SIZE - 1)

ISR(USART_RXC_vect){
    uint8_t newHead, Data;
    //Pobieramy bajt danych z bufora sprzętowego USART
    Data = UDR;

    //Obliczamy nowy indeks „głowy węża”
    newHead = (MIDIBufferHead + 1) & MIDI_RX_BUF_MASK;

    //Sprawdzamy, czy wąż nie zacznie zjadać własnego ogona
    if (newHead == MIDIBufferTail){
        //Przepełnienie bufora - można obsłużyć ten błąd
    } else {
        //Zapamiętujemy nowy indeks „głowy węża”
        MIDIBufferHead = newHead;
        //Wpisujemy odebrany bajt do bufora FIFO MIDI
        MIDIBuffer[newHead] = Data;
        ++MIDIbytes;
    }
}
```

Listing 2. Funkcja pobierająca bajt z bufora kołowego MIDI

```
uint8_t MIDIGetByte(void){
    uint8_t Byte;
    ATOMIC_BLOCK(ATOMIC_RESTORESTATE){
        //Obliczamy i zapamiętujemy nowy indeks „ogona węża” bufora MIDI
        MIDIBufferTail = (MIDIBufferTail + 1) & MIDI_RX_BUF_MASK;
        //Zwracamy bajt pobrany z bufora jako rezultat wykonania funkcji
        Byte = MIDIBuffer[MIDIBufferTail];
        --MIDIbytes;
    }
    return Byte;
}
```

identyfikatora komunikatu typu Control Change (0xB0), bajta zawierającego numer kontrolera i bajta zawierającego wartość danego kontrolera (nawet wtedy, gdy wartość ta jest bez znaczenia, jeśli chodzi o funkcjonowanie sterownika). Co warto podkreślić, niektóre z poleceń można wydawać

na dwa sposoby: w zależności od przekazanej wartości można włączyć lub wyłączyć dany przekaznik (kontrolery 85...88) lub innym poleceniem zmienić stan przekaznika na przeciwny (kontrolery 89...92). W ostatnim przypadku przekazywana wartość nie jest brana pod uwagę. Opis wszystkich

Listing 3. Funkcja inicjalizująca interfejs USART do użycia jako sprzęg MIDI

```
//Definicja prędkości interfejsu MIDI
#define MIDI_BAUD 31250
#define CALCULATED_UBRR F_CPU/16/MIDI_BAUD-1

void MIDIinit(void) {
    //Ustawienie prędkości interfejsu MIDI
    UBRRH = (CALCULATED_UBRR)>>8;
    UBRRL = CALCULATED_UBRR;
    //Załączenie odbiornika oraz
    //aktywacja przerwania od odbioru danych
    UCSRB = (1<<RXEN)|(1<<RXCIE);
    //Ustawienie formatu ramki:
    //8bitów danych, 1 bit stopu
    UCSRC = (3<<UCSZ0);
}
```

Listing 4. Funkcja analizująca nadchodzące komunikaty MIDI

```
void MIDHandling(void){
    uint8_t receivedByte;
    static uint8_t CCnumber, CCvalue;
    static uint8_t MIDstatus = WAIT_FOR_CTRL_CHANGE;

    //Sprawdzamy czy odebrano nowe dane MIDI
    if(MIDIIsReady()){
        receivedByte = MIDIGetByte(); //Odczytujemy znak z bufora MIDI

        switch(MIDstatus){
            //Sprawdzamy odebranie CC na aktywnym kanale MIDI
            case WAIT_FOR_CTRL_CHANGE:
                if(((receivedByte & 0xF0) == MIDI_CTRL_CHANGE)
                    && ((receivedByte & 0x0F) == eeprom_read_byte(&activeMIDchannelEE)))
                    MIDstatus = WAIT_FOR_CTRL_NUMBER;
                break;

            //Sprawdzamy, czy nadeszły dane - numer kontrolera MIDI
            // (gdy dane to bit7 wyzerowany)
            case WAIT_FOR_CTRL_NUMBER:
                if(!(receivedByte & 0x80)){
                    CCnumber = receivedByte;
                    MIDstatus = WAIT_FOR_CTRL_VALUE;
                } else {
                    MIDstatus = WAIT_FOR_CTRL_CHANGE;
                }
                break;

            //Sprawdzamy, czy nadeszły dane - wartość kontrolera MIDI
            // (gdy dane to bit7 wyzerowany)
            case WAIT_FOR_CTRL_VALUE:
                if(!(receivedByte & 0x80)){
                    CCvalue = receivedByte;
                    //W zależności od numeru kontrolera CC i jego wartości
                    //podejmujemy stosowne akcje
                    switch(CCnumber){
                        //Obsługa kolejnych kontrolerów MIDI...
                    }
                    MIDstatus = WAIT_FOR_CTRL_CHANGE;
                }
                break;
        }
    }
}
```

kontrolerów MIDI wraz ze znaczeniem ich wartości dla funkcjonowania urządzenia pokazano w tabeli 1.

Program sterujący

Na koniec nieco szczegółów implementacyjnych, ponieważ MIDI to ciekawe, a mimo wszystko rzadko prezentowane zagadnienie. Wszystkie komunikaty MIDI przesyłane do naszego sterownika odbierane są w procedurze obsługi przerwania interfejsu USART (USART_RXC_vect) i ładowane do specjalnego bufora kołowego. W ten sposób mamy pewność, że żaden taki komunikat nie zostanie pominięty. Całość procedury obsługi przerwania odpowiedzialnej za odbiór komunikatów MIDI wraz z deklaracją zmiennych pokazano na listingu 1. Całość prostej funkcji pobierającej bajt z bufora danych pokazano na listingu 2. Funkcja inicjalizująca interfejs USART, by można go było użyć jako sprzęg MIDI, została pokazana na listingu 3. Znając niezbędne funkcje narzędziowe związane z MIDI, możemy przystąpić do implementacji funkcji analizującej nadchodzące dane i sterującej zasobami sprzętowymi urządzenia. Kod takiej funkcji przedstawiono na listingu 4. Funkcja wywoływana jest każdorazowo w pętli głównej programu aplikacji i przez swoje działanie powoduje stosowną zmianę na portach wyjściowych mikrokontrolera, do których podłączono przekaźniki K1...K8 czy też na wyjściu MUTE. Innymi słowy, wspomniana funkcja realizuje obsługę wszystkich komunikatów MIDI, jakie przedstawiono wcześniej w tabeli 1.

Robert Wołgajew, EP

REKLAMA

Najlepszy
adres w sieci
<http://ep.com.pl>



Sztuczne obciążenie w.cz. o impedancji 50 Ohm (1)

W artykule opisano projekt sztucznego obciążenia zmiennoprądowego w.cz. o dopuszczalnej mocy strat rzędu 5...10 W (QRP), przeznaczonego do pracy w pasmie KF. Jest to pierwsza część podwójnego projektu sztucznych obciążeń zmiennoprądowych małej mocy i zarazem miękkie wprowadzenie do świata projektów urządzeń nadawczo-odbiorczych. Takim bowiem konstruktorom autor tego podwójnego projektu zamierza poświęcić więcej uwagi.

Opisany w tym artykule przyrząd to tzw. sztuczne obciążenie (*dummy load*) zmiennoprądowe, przeznaczone przede wszystkim do pracy z wysokimi częstotliwościami radiowymi (w.cz.) z przedziału co najmniej od 100 kHz do 30 MHz. Jest to urządzenie absolutnie niezbędne w warsztacie każdego radioamatora – konstruktora. Jego główne przeznaczenie to pełnienie funkcji wzorcowego obciążenia wyjścia wzmacniacza mocy w.cz. o dopasowanej rezystancji $R_{load}=50\ \Omega$. Takie obciążenie jest niezwykle przydatne w trakcie wszelkich prób technicznych oraz napraw, regulacji, a także strojenia

wzmacniaczy, bowiem rzeczywiste obciążenie antenowe może nie być akurat dostępne lub – co jeszcze bardziej prawdopodobne – nie mieć odpowiednio stabilnych i neutralnych parametrów (jak np. stała i rzeczywista, dopasowana impedancja, równa dokładnie $50\ \Omega$) w całym wymaganym pasmie oddziaływania testowanego źródła energii w.cz.

Zasada działania i właściwości urządzenia

Rysunek 1 pokazuje schemat tego niezwykle prostego, ale przydatnego przyrządu. Sygnał mocy w.cz. jest podawany na wejście

Dodatkowe materiały do pobrania ze strony www.media.avt.pl

W ofercie AVT* AVT-3210

Podstawowe parametry:

- przeznaczone do pracy z częstotliwościami od 100 kHz do 30 MHz,
- impedancja równa dokładnie $50\ \Omega$ w całym pasmie,
- dopuszczalna ciągła moc strat 5 W (przez krótki czas do 10 W),
- wyposażony w prosty detektor szczytowy.

Projekty pokrewne na www.media.avt.pl:

- AVT-5586 Programowalne sztuczne obciążenie DC (EP 11/2018)
- AVT-5510 Sztuczne obciążenie (EP 8/2015)
- AVT-1797 Sztuczne obciążenie wysokonapięciowe (EP 4/2014)

Uwaga! Elektroniczne zestawy do samodzielnego montażu. wymagana umiejętność lutowania!

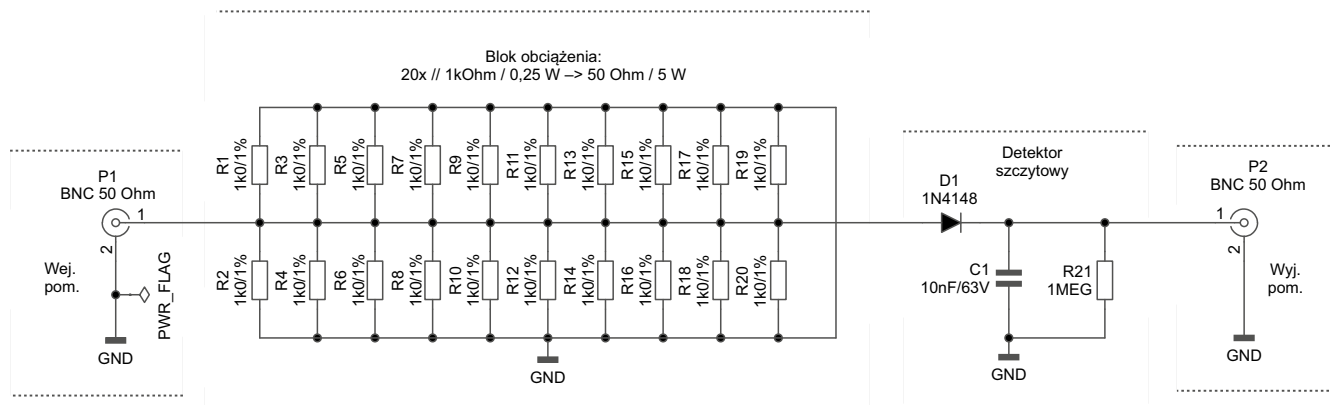
Podstawową wersją zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wzlutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu.

Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

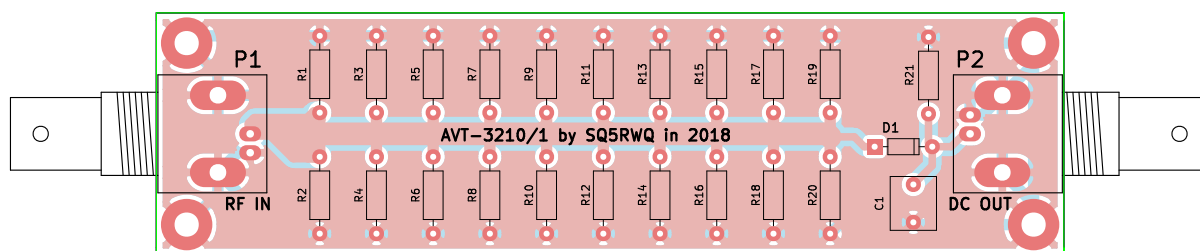
- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wzlutowane w płytkę PCB)
- wersja [A] – płytkę drukowaną bez elementów i dokumentacji Kity w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
 - wersja [A*] – płytkę drukowaną [A] + zaprogramowany układ [UK] i dokumentacja
 - wersja [UK] – zaprogramowany układ

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz!

na <http://sklep.avt.pl>. W przypadku braku dostępności na <http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.



Rysunek 1. Schemat elektryczny urządzenia



Rysunek 2. Schemat płytki PCB wraz z rozmieszczeniem elementów

P1, zrealizowane w postaci koncentrycznego gniazda (męskiego) typu BNC. Zasadniczo w urządzeniach radiokomunikacyjnych do przenoszenia znacznych mocy w.cz. stosuje się złącza typu UC-1, jednak w zakresie mocy do 20 W włącznie akceptowalne jest stosowanie połączeń typu BNC.

Wejściowy sygnał mocy jest wprowadzany na zestaw 20 sztuk połączonych równolegle rezystorów R1...R20 o opornościach równych 1 kΩ, tolerancji 1% i nominalnych mocach strat termicznych równych 0,25 W. Zastosowanie popularnych rezystorów o tolerancji 5% mogłoby pogorszyć nieco własności impedancyjne (a więc i dopasowanie do źródła energii przyrządu), bowiem rozrzuty wartości elementów, pochodzących z tej samej serii produkcyjnej, mogłyby okazać się podobne i – w rezultacie – nie skompensować się w ramach oczekiwanej wypadkowej impedancji 50 Ω.

Opisywana konstrukcja jest w stanie bezproblemowo odprowadzać 20×0,25 W=5 W mocy ciągłej. Jakkolwiek eksperymenty praktyczne (pokazane także na fotografii tytułowej) wykazały, że przy odpowiednio starannej konstrukcji przyrząd bez problemu wytrzymuje obciążenie mocą nawet do 10 W przez kilka minut – bez nagrzewania się do temperatur, które mogłyby go uszkodzić. Oczywiście, zastosowanie oporników o mocy zwiększonej z 0,25 W do 0,5 W zapewne podwoiłoby tę moc, jednak mogłoby nieco pogorszyć właściwości impedancyjne przyrządu.

Należy podkreślić, że konstrukcja oparta na zestawie połączonych równolegle niewielkich rezystorów o małej dopuszczalnej mocy strat, w miejsce użycia mniejszej liczby rezystorów (lub nawet pojedynczego opornika)

większej mocy, nie jest błędem konstruktorskim i ma dwa główne cele. Pierwszym jest zmniejszenie wypadkowej pasywności indukcyjności szeregowej Ls obciążenia, która w przypadku połączenia równoległego oporników ulega redukcji w stopniu równym ich ilości. Drugi cel, to poprawa warunków oddawania ciepła (poprzez zastosowanie zestawu rozproszonych, oddalonych od siebie rezystorów), która obniża efektywną temperaturę pracy aktywnych elementów (oporników). To z kolei zjawisko znacznie ogranicza termiczne zmiany wartości ich rezystancji a – w konsekwencji – stabilizuje dopasowaną wartość rzeczywistej impedancji, równej 50 Ω, w szerokim zakresie częstotliwości radiowych.

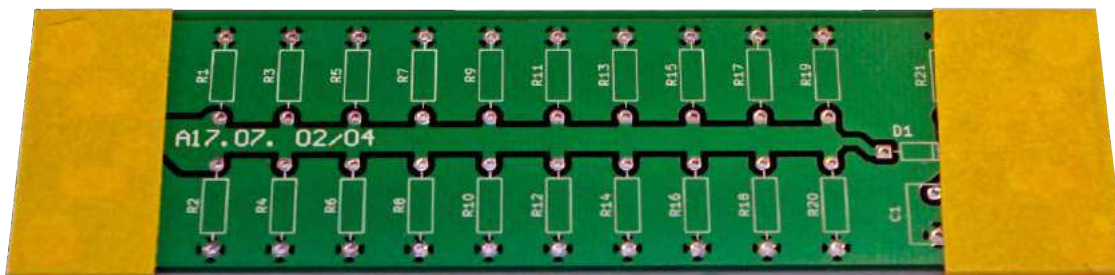
Równoległe do zestawu oporników dołączony jest prosty detektor szczytowy, oparty na topologii jednopółkowego prostownika szeregowego z diodą D1 (1N4148). W tym układzie kondensator C1 (10 nF) pełni funkcję filtra wygładzającego szczyty wyprostowanych połówek mierzonego sygnału sinusoidalnego. Natomiast rezystancja R21 (1 MΩ) ma za zadanie rozładowywać pojemność C1 w sytuacji, gdy spadnie poziom mierzonego napięcia wejściowego U_{in}(P1) lub zaniknie ono całkowicie. Stała czasowa rozładowania tego równoległego dwójnika RC (w przypadku nieobciążonego wyjścia pomiarowego) wynosi 10 ms i pozwala prawidłowo pracować z częstotliwościami mierzonych sygnałów nawet znacznie poniżej 100 kHz. Zmierzone stałe (lub wolnozmiennne) napięcie jest podawane na port wyjściowy P2, zrealizowany tutaj także w formie gniazda BNC. Takie rozwiązanie nie ma na celu zapewnienia stałego

dopasowania do impedancji 50 Ω w całym torze wyjściowym, ale jedynie umożliwienie wyprowadzenia sygnału mierzonego z zastosowaniem koncentrycznego kabla ekranowanego, chroniącego skutecznie małe napięcia mierzone przed zewnętrznymi silnymi zakłóceniami elektromagnetycznymi, które mogą pochodzić m.in. z badanych wzmacniaczy mocy w.cz.

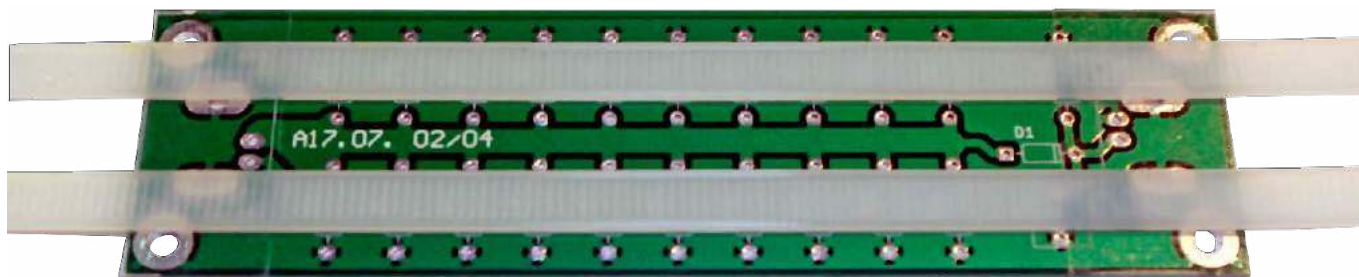
Montaż przyrządu

Na rysunku 2 pokazano layout płytki drukowanej sztucznego obciążenia wraz z rozmieszczeniem na niej elementów. Pokrycie strefą masy występuje po obu stronach PCB, co ma na celu przede wszystkim poprawę emisji wytwarzanego w układzie ciepła. Montaż przyrządu należy rozpocząć od przygotowania rezystorów R1...R20 poprzez zagięcie ich wyprowadzeń w kształcie litery „U”, odpowiednie do rozstawu otworów montażowych na PCB. Następnie

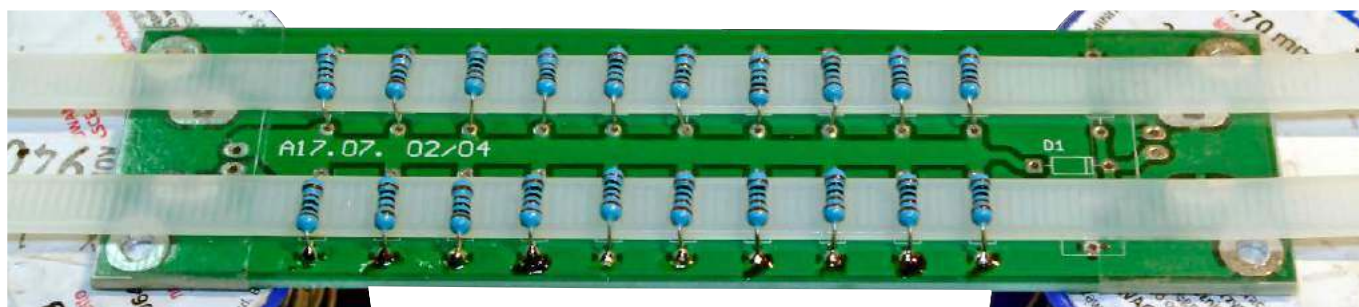
REKLAMA



Fotografia 3. Montaż rezystorów – przygotowanie taśmy dwustronnej



Fotografia 4. Montaż rezystorów – przygotowanie dystansów



Fotografia 5. Zamontowane rezystory

należy starannie przygotować płytkę drukowaną do montażu tych elementów. **Fotografie 3, 4 i 5** przybliżają kolejność oraz szczegóły prawidłowego montażu przyrządu. Widać na nich, w jaki sposób zastosowano dwustronną taśmę samoprzylepną do mocowania dwóch tymczasowych elementów podpierających, wykonanych z opasek samozaciskowych z tworzywa sztucznego o odpowiedniej szerokości.

Po solidnym zalutowaniu rezystorów R1...R20 (najlepiej po obu stronach PCB) opaski samozaciskowe należy delikatnie odkleić od PCB i wysunąć spod rezystorów, a następnie usunąć odcinki taśmy samoprzylepnej dwustronnej oraz zmyć (np. denaturatem) resztki pozostałego po niej kleju.

Montaż pozostałych elementów nie wymaga szczególnego komentarza. Układ nie wymaga żadnego uruchomienia, natomiast warto jest zastosować obudowę metalową,

która zapewni funkcje: dobrego ekranowania oraz odprowadzania ciepła z przyrządu. Taka metalowa obudowa powinna być starannie i wielopunktowo połączona galwanicznie z masą PCB przyrządu, a jej zewnętrzna powierzchnia dla bezpieczeństwa eksploatacji powinna zostać pokryta lakierem o dobrych właściwościach izolacyjnych.

Fotografia tytułowa pokazuje przykładową realizację przyrządu z obudową wykonaną z kawałków laminatu, dwustronnie pokrytego miedzią i metalizowanego stopem niskotemperaturowym o wysokiej przewodności.

Eksploracja przyrządu

Prawidłowa i bezpieczna eksploatacja przyrządu sprowadza się do podłączenia do wejścia P1 (gniazda BNC) źródła energii w.c.z. o impedancji wyjściowej jak najbardziej zbliżonej do 50 Ω (niedopasowanie może wywołać niebezpieczne przepięcia, wynikające z powstawania fali stojącej w torze przesyłu mocy w.c.z.) i mocy dysponowanej nieprzekraczającej 5...10 W. W egzemplarzu modelowym sprawdzono bezpieczne rozpraszanie mocy o wartości 10 W nawet przez kilka minut, jednak może to zależeć od sposobu oraz staranności wykonania obudowy przyrządu i zasadniczo obciążanie przyrządu mocami znacznie przekraczającymi 5 W przez dłuższy okres nie jest

zalecane. Wartość mocy oddawanej do obciążenia możemy odczytać z **tabeli 1** lub z wykresu na **rysunku 6**, które odwzorowują zależność na wartość szczytową U_p [V]

Tabela 1. Wartości szczytowe napięcia odpowiadające wartościom mocy oddawanej do obciążenia

Pin(P1) [W]	Up(P2) [V]
0,1	2,7
0,2	4,1
0,3	5,1
0,4	5,9
0,5	6,7
0,6	7,3
0,7	7,9
0,8	8,5
0,9	9,1
1	9,6
2	13,7
3	16,9
4	19,6
5	21,9
6	24,1
7	26,0
8	27,9
9	29,6
10	31,2

Wykaz elementów:

Rezystory:

R1...R20: 1 k Ω /0,25 W/1%
R21: 1 M Ω /5%

Kondensatory:

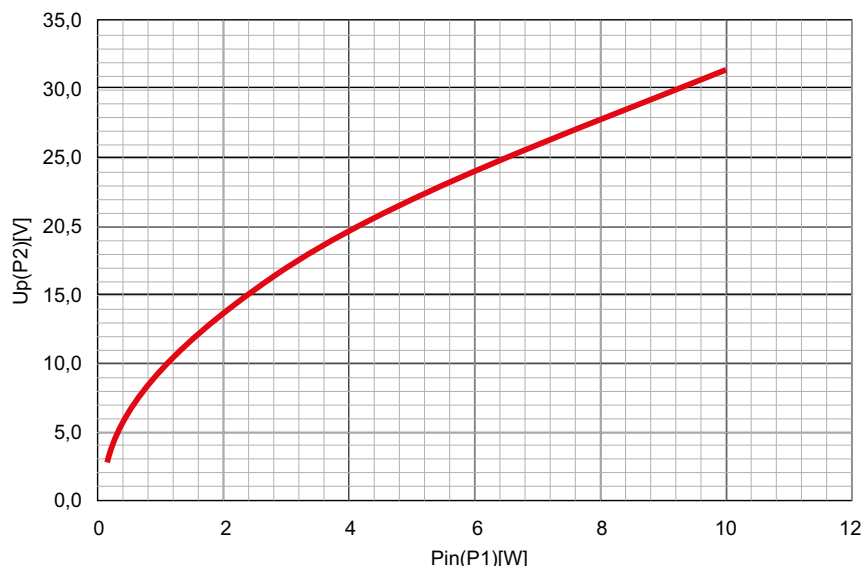
C1: 10 nF/63 V

Półprzewodniki:

D1: 1N4148/D0-35

Inne:

P1, P2: gniazda BNC 50 Ω (kątowe, lutowane do druku)
Obudowa: wg opisu w tekście



Rysunek 6. Wykres wartości szczytowej napięcia w funkcji mocy oddawanej do obciążenia

napięcia sinusoidalnego przy mocy chwilowej P_{in} [W], podanej na port P1 obciążenia $R_{load}=50\ \Omega$, pomniejszoną o uśredniony spadek napięcia na diodzie detekcyjnej D1, obciążonej rezystorem R21 i pojemnością C1:

$$U_p(P2) = \sqrt{2 \cdot R_{load} \cdot P_{in}(P1)} - 0,42\ [V]$$

Spadek napięcia na diodzie detekcyjnej D1 został wyznaczony obliczeniowo za pomocą modelu symulatora SPICE, podanego przez jednego z producentów diody 1N4148

jako właściwy dla typowego jej egzemplarza, oraz z użyciem podstawowych wzorów, obowiązujących dla tego przyrządu półprzewodnikowego. Dotyczy on obciążenia mocą $P_{in}(P1)=5\ W$, natomiast dokładne wartości dla zakresu mocy $P_{in}(P1)=0,1...10\ W$ mieściły się pomiędzy 0,32 V a 0,44 V. Jakkolwiek w typowych przypadkach wartości te są nieporównywalnie mniejsze od mierzonego napięcia wyjściowego i dlatego przyjęte uśrednienie można uznać za bardzo

mało znaczące dla dokładności prowadzonego pomiaru.

Podsumowanie projektu

W artykule opisano projekt tzw. sztucznego obciążenia, przeznaczonego do pracy z częstotliwościami radiowymi (w.cz.) z przedziału od 100 kHz do 30 MHz z dopasowaną rezystancją $R_{load}=50\ \Omega$ co najmniej w całym podanym zakresie częstotliwości. Jest to urządzenie bardzo przydatne do napraw, regulacji i strojenia wzmacniaczy w.cz., ponieważ pełni funkcję rzeczywistego obciążenia antenowego z gwarantowanymi parametrami. To urządzenie, przy odpowiednio starannym wykonaniu, jest w stanie skutecznie konkurować zarówno cenowo, jak i jakościowo z gotowymi urządzeniami, dostępnymi ze sklepowej półki.

Więcej informacji o dalszych jego losach można będzie znaleźć na stronie www autora pod adresem: <http://sq5rww.pl/?p=1869>. W trakcie przygotowania jest także wersja przyrządu o dwukrotnie większej mocy rozpraszanej oraz z wbudowanym wielostopniowym wskaźnikiem poziomów mocy, opartym na diodach LED i rozwiązania układowe, odporne na występujące w układzie pomiarowym zakłócenia impulsowe.

Adam Sobczyk SQ5RWW

sq5rww@gmail.com

<http://sq5rww.pl>

REKLAMA

Wszystko, co lubisz,
w jednym miejscu



UlubionyKiosk.pl

Oferuje papierowe i elektroniczne wydania czasopism z najważniejszych segmentów rynku:

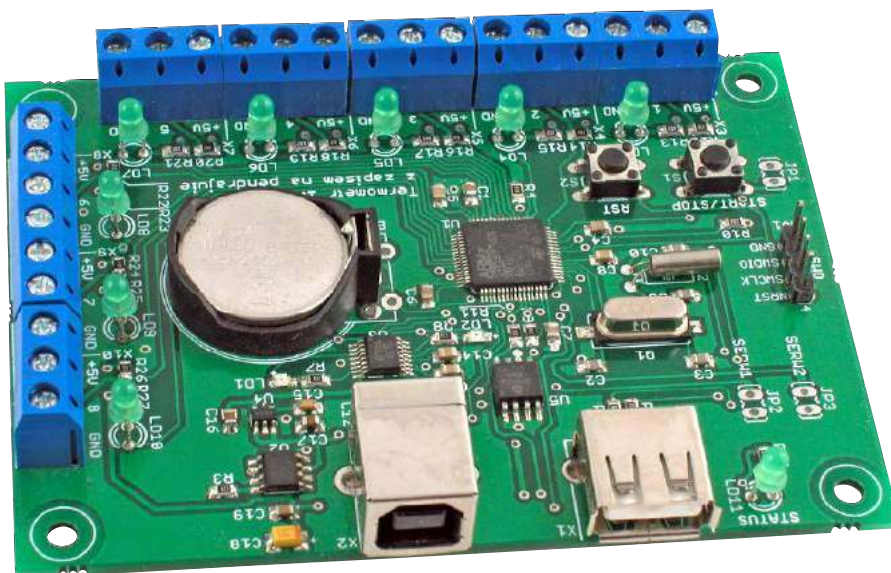
budownictwo i wnętrza, muzyka i dźwięk, elektronika i automatyka, edukacja i hi-tech, rodzina.

Przesyłka
GRATIS

Przenośny rejestrator temperatury

z zapisem na pendrivie i oprogramowaniem do wizualizacji

Artykuł prezentuje projekt urządzenia przeznaczonego do długotrwałych pomiarów i rejestracji temperatury. Dane przechowywane są w postaci plików tekstowych na wymiennym, łatwo dostępnym nośniku. Program komputerowy ułatwia ustawianie parametrów, kontrolę przebiegu cyklu pomiarowego i wstępną wizualizację zarejestrowanych pomiarów.



Dodatkowe materiały do pobrania ze strony www.media.avt.pl

W ofercie AVT* AVT-5725

Podstawowe parametry:

- 8 niezależnych kanałów pomiarowych do podłączenia czujników temperatury typu 18B20 lub 18S20,
- czas pomiędzy kolejnymi pomiarami ustawiany przez użytkownika w zakresie od 2 sekund do 23 godzin 59 minut 59 sekund,
- wyzwalanie nowego cyklu pomiarowego z programu sterującego lub po naciśnięciu przycisku na rejestratorze,
- zapis pomiarów w formie pliku tekstowego na wymiennym nośniku danych: pendrive,
- możliwość częściowego odczytu danych podczas aktywnego cyklu pomiarów,
- zasilanie napięciem stałym +5 V z portu USB lub zasilacza z gniazdem USB,
- pobór prądu zależny od użytego nośnika, zazwyczaj od 50 mA do 100 mA.

Projekty pokrewne na www.media.avt.pl:

AVT-5573	Nieskomplikowany termometr-rejestrator (EP 11/2016)
AVT-5420	Wielopunktowy termometr z rejestracją (EP 10/2013)
AVT-5373	Tlogger - rejestrator temperatury (EP 12/2012)
AVT-5352	Dobowy rejestrator temperatury (EP 7/2012)
AVT-5230	Rejestrator temperatury z interfejsem USB (EP 4/2010)
AVT-1830	Sygnalizator temperatury do pieca CO (EP 8/1994)

Uwaga! Elektroniczne zestawy do samodzielnego montażu.

Wymagana umiejętność lutowania!

Podstawową wersją zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] - jeśli występuje w projekcie), które należy samodzielnie wzlutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu. Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] - zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wzlutowane w płytkę PCB)
- wersja [A] - płytkę drukowaną bez elementów i dokumentacji. Wersja [A] ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz <http://sklep.avt.pl>. W przypadku braku dostępności na <http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.
- wersja [UK] - zaprogramowany układ

Dokładny i czytelny sposób działania rejestratora opisują poniższe zasady:

1. Rejestrator może znajdować się w jednym z dwu stanów: gotowości lub w cyklu pomiarów. Przejście do cyklu pomiarów lub jego zakończenie następuje po wysłaniu komendy z programu sterującego uruchomionego na komputerze lub, jeśli taka opcja jest ustawiona, po 3-sekundowym naciśnięciu przycisku S1.
2. Gdy jest aktywny cykl pomiarów z zaprogramowaną częstotliwością, testowane są wszystkie kanały i po wykryciu dołączonych czujników odczytywana jest temperatura. Jeżeli odczyt z czujnika zakończy się sukcesem, zapala się dioda sygnalizacyjna skojarzona z kanałem. W przypadku błędu odczytu lub jeśli do kanału nie jest podłączony czujnik, dioda jest gaszona. Kolejne dane pomiarowe dodawane są do utworzonego pliku tekstowego w wymiennej pamięci masowej, czyli na pendrivie. Nazwa pliku składa się z liter „MT” i numeru cyklu np.: MT_001.txt. W przypadku kolejnego np. i następnego cyklu pomiarów, numeracja rozpoczyna się ponownie od 1, a dane starego pliku o takiej samej nazwie zostają nadpisane. Wszystkie pliki pomiarów zapisywane są w utworzonym podkatalogu o nazwie „Pomiary Temperatury”.

3. W trybie gotowości można ustawiać parametry rejestratora, w tym: czas pomiędzy kolejnymi pomiarami, włączać lub wyłączać aktywność przycisku S1 służącego do ręcznego wyzwalania cyklu pomiarowego, programowo wyzwoić cykl pomiarowy.
4. Dane on-line można importować z pendrive'u korzystając z gniazda USB transmisji oznaczonego na Rysunku 1 jako X2. Można to robić zarówno gdy rejestrator jest w trybie gotowości, jak i gdy aktywny jest cykl pomiarów.
5. Po włączeniu zasilania rejestrator odtwarza swój stan w momencie sprzed odłączenia. Gdy był aktywny, cykl pomiarów urządzenie kontynuuje pomiary tak, aby wznowione rejestracje były zgodne z zaprogramowanym okresem pomiarów.
6. W czasie cyklu pomiarów dostępne są informacje o statusie: dacie i czasie startu, liczbie zarejestrowanych pomiarów, sygnalizacji utraty 1 lub więcej kolejnych pomiarów na skutek zaniku zasilania, wystąpienia błędów zapisu na pendrive.

Budowa rejestratora

Na płycie rejestratora umieszczono 8 gniazd śrubowych do podłączenia maksymalnie 8 czujników temperatury. Oprócz tego zamontowano gniazdo USB-A dla pendrive'u i USB-B

służące zarówno do podłączenia zasilania, jak i do komunikacji z oprogramowaniem sterującym uruchomianym na komputerze. Schemat ideowy rejestratora pokazano na **rysunku 1**. Ze względu na łatwość tworzenia szkieletu oprogramowania przez ogólnodostępne narzędzie wybrano STM32F401 w obudowie z 64 wyprowadzeniami. Wszystkie czujniki, sygnalizacje i gniazdo pendrive'u obsługiwane są bezpośrednio przez porty kontrolera. Mostek USB, oznaczony na schemacie jako U3, umożliwia połączenie linii USART1 kontrolera z portem USB komputera.

Układ U2 jest zintegrowanym przełącznikiem dostarczającym napięcie zasilania +5 V do pendrive'u i wszystkich dołączonych czujników temperatury. Dobierając wartość opornika R3, można ustawić górny limit prądu, jaki może być pobrany z przełącznika i tym samym zabezpieczyć przed przeciążeniem port USB, który zasilą całe urządzenie. Wartość oporu można obliczyć, korzystając z wzoru:

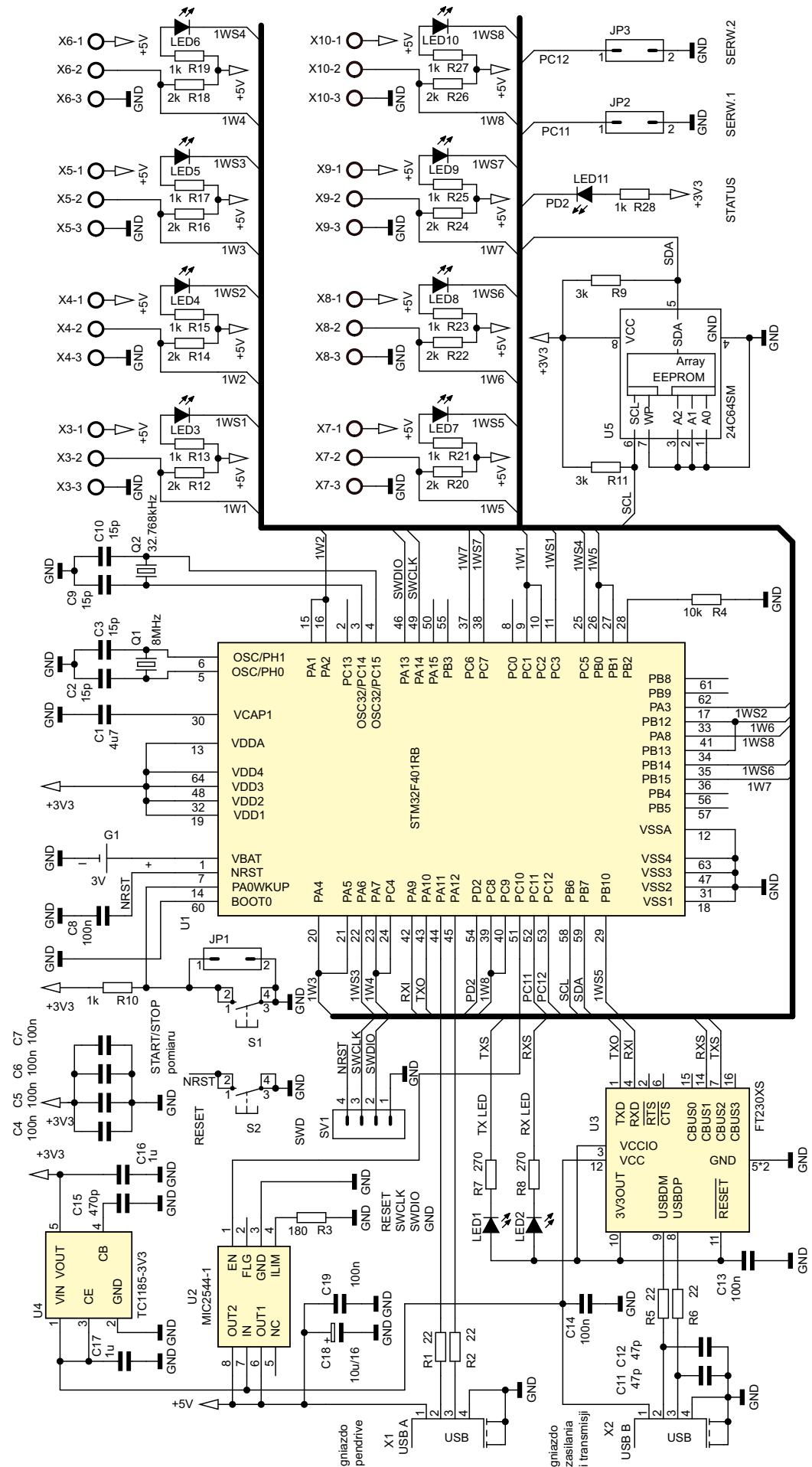
$$R_{set} \Omega = 230 \frac{V}{I_{limit} A}$$

przy dopuszczalnym zakresie $154 \Omega < R_{set} < 2,29 \text{ k}\Omega$.

Stabilizator U4 dostarcza zasilania 3,3 V do kontrolera i pamięci EEPROM U5,

w której przechowywane są ustawienia, które nie powinny zostać utracone w czasie odłączenia rejestratora od zasilacza. Bateria litowa G1 służy do podtrzymania działania wewnętrznych rejestrów zegara czasu rzeczywistego w kontrolerze.

Diody LED1 i LED2 sygnalizują przepływ danych portem USB podczas wymiany danych pomiędzy rejestratorem a programem sterującym. Świecenie diod LED3... LED10 sygnalizuje dołączenie do poszczególnych kanałów czujników temperatury i ich prawidłowy odczyt. Ciągłe świecenie lub miganie diody LED11 informuje o stanie, w jakim znajduje się rejestrator.



Rysunek 1. Schemat ideowy rejestratora

Dodatkowo na płytce zamontowane są dwa przyciski S1 i S2. Naciśnięcie S2 wymusza zerowanie kontrolera, natomiast dłuższe naciśnięcie S1 powoduje rozpoczęcie lub zakończenie cyklu pomiarów, o ile ustawienia rejestratora zezwalają na uaktywnienie tego przycisku.

Program sterujący Multi Rejestrator

Oprogramowanie Multi Rejestrator służy do sterowania rejestratorem, ustawiania parametrów pracy, kontrolowania przebiegu cyklu pomiarów, wizualizacji zgromadzonych pomiarów. W trybie off-line, kiedy rejestrator nie jest podłączony do komputera, program pozwala przeglądać zgromadzone na twardym dysku historyczne zapisy rejestracji. Gdy płytka rejestratora jest podłączona, oprogramowanie może na bieżąco kontrolować przebieg rejestracji.

Program Multi Rejestrator wyświetla się jako zestaw 4 zakładek: Tryb Tekstowy, Tryb Graficzny, Sterowanie, Import.

Zakładka Sterowanie

Po uruchomieniu programu należy przejść do zakładki Sterowanie. Początkowo żaden element na zakładce nie jest aktywny, dopóki nie zostanie nawiązane połączenie z płytką rejestratora. Służy do tego przycisk na pasku narzędziowym , automatyczne wyszukiwanie podłączonego urządzenia. Program przeszukuje wszystkie dostępne do otwarcia porty, wysyłając zapytanie. Gdy uzyska prawidłową odpowiedź identyfikującą dołączoną do portu płytkę, wyświetla informacje

Wykaz elementów:

Rezystory:

R4: 10 kΩ SMD 0805
R3: 180 Ω SMD 0805
R10, R13, R15, R17, R19, R21, R23, R25, R27, R28: 1 kΩ SMD 0805
R1, R2, R5, R6: 22 Ω SMD 0805
R7, R8: 270 Ω SMD 0805
R12, R14, R16, R18, R20, R22, R24, R26: 2 kΩ SMD 0805
R9, R11: 3 kΩ SMD 0805

Kondensatory:

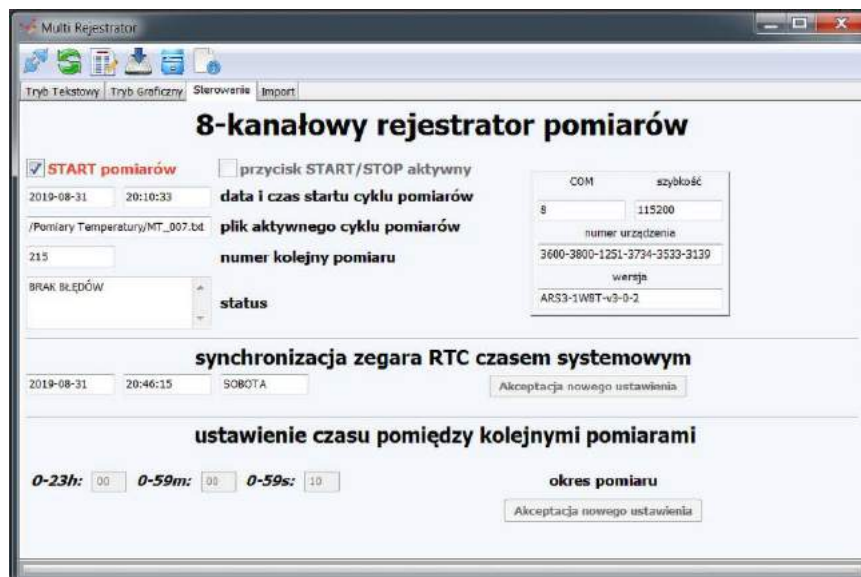
C4, C8, C13, C14, C19: 100 nF SMD 0805
C18: 10 μF/16 V SMD
C2, C3, C9, C10: 15 pF SMD 0805
C16, C17: 1 μF SMD 0805
C15: 470 pF SMD 0805
C11, C12: 47 pF SMD 0805
C1: 4,7 μF SMD 0805

Półprzewodniki:

LD3...LD11: LED przewlekane 3 mm
LD1, LD2: LED SMD 0805
U5: 24C64 EEPROM SO-08
U3: FT230XS SSOP16
U2: MIC2544-1 SO-08
U1: STM32F401RB TQFP64
U4: TC1185-3V3 SOT25

Inne:

S1, S2: mikroprzyciski
X3...X10: złącza śrubowe 3 pola raster 5 mm
JP1: goldpiny 2 styki
Q2: kwarc zegarkowy 32,768 kHz przewlekany
BAT1: koszyk baterii litowej CR2430 przewlekany
Q1: 8 MHz kwarc SMD SM49
X1: gniazdo USB-A przewlekane
X2: gniazdo USB-B przewlekane



Rysunek 2. Widok zakładki Sterowanie

o numerze płytki i wersji oprogramowania, tak jak to widać na **rysunku 2**. Po naciśnięciu na pasku narzędziowym przycisku , odświeżanie, na zakładce wyświetlają się dodatkowe informacje oraz zostaje odblokowany dostęp do pozostałych elementów.

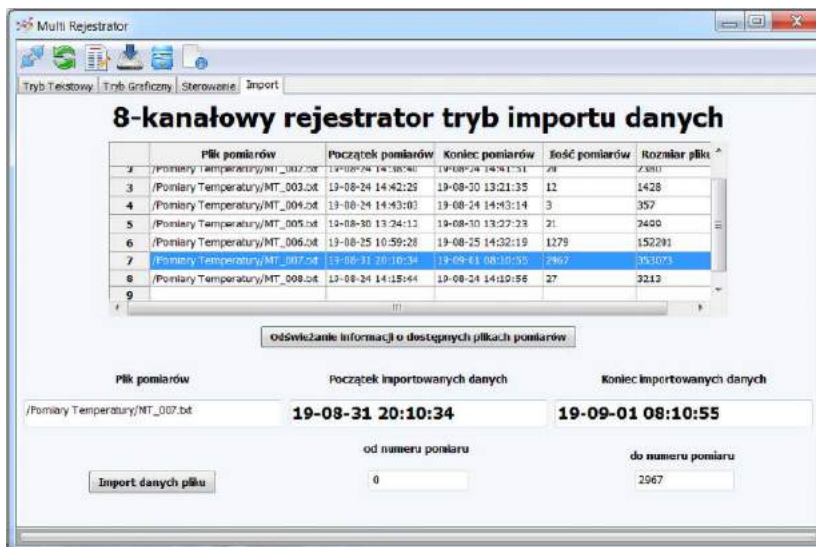
Gdy rejestrator jest w trybie gotowości, można przesłać do zegara RTC na płytce czas systemowy komputera. Można zaprogramować okres między kolejnymi pomiarami, wpisując wartości w polach godzin, minut i sekund, które to ustawienia po naciśnięciu klawisza Akceptacja nowego ustawienia, przesyłane są do płytki rejestratora i zapamiętywane w pamięci nieulotnej. Zaznaczając opcję przycisk START/STOP aktywny, można aktywować przycisk S1 ręcznego wyzwalania cyklu pomiarów na płytce. W końcu można z poziomu programu zainicjować tryb pomiarów, zaznaczając opcję START pomiarów lub naciskając na pasku narzędziowym ikonę przycisku  start/stop.

Gdy rejestrator jest w trybie aktywnego cyklu pomiarów, po każdorazowym naciśnięciu

klawisza Odświeżanie wyświetlone zostają informacje o dacie i czasie startu cyklu pomiarów, ścieżka dostępu do pliku z danymi aktualnych pomiarów oraz uaktualnione informacje o numerze pomiaru i statusie.

Zakładka Import

Służy do odczytu zawartości plików zapisanych na pendrivie umieszczonym w gnieździe X1 płytki rejestratora. Funkcje zakładki stają się aktywne po nawiązaniu połączenia z rejestratorem. Na **rysunku 3** pokazano przykładowy wygląd zakładki. Tabela zostaje zapełniona listą dostępnych do importu plików, po naciśnięciu ikony klawisza Odświeżanie informacji o dostępnych plikach pomiarów. Oprócz nazwy pliku wyświetlają się informacje o: dacie i czasie pierwszego oraz ostatniego pomiaru, liczbie pomiarów, rozmiarze pliku. Wskazanie pozycji tabeli kursorem i kliknięcie lewym przyciskiem myszki wyświetla zakres danych pliku, które można zaimportować. Zakres można zawęzić, korygując punkty czasowe odpowiadające



Rysunek 3. Widok zakładki Import

	data	czas	czujnik1	czujnik2	czujnik3	czujnik4	czujnik5	czujnik6	czujnik7	czujnik8
1	N:190828	Time:125921	T1= 27.4°C	T2= 27.4°C	T3= ...°C	T4= ...°C	T5= ...°C	T6= 25.7°C	T7= 25.5°C	T8= ...°C
2	N:190828	Time:125922	T1= 27.3°C	T2= 27.4°C	T3= ...°C	T4= ...°C	T5= ...°C	T6= 25.7°C	T7= 25.5°C	T8= ...°C
3	N:190828	Time:125932	T1= 27.4°C	T2= 27.3°C	T3= ...°C	T4= ...°C	T5= ...°C	T6= 25.7°C	T7= 25.5°C	T8= ...°C
4	N:190828	Time:125942	T1= 27.4°C	T2= 27.3°C	T3= ...°C	T4= ...°C	T5= ...°C	T6= 25.7°C	T7= 25.5°C	T8= ...°C
5	N:190828	Time:125952	T1= 27.4°C	T2= 27.3°C	T3= ...°C	T4= ...°C	T5= ...°C	T6= 25.7°C	T7= 25.5°C	T8= ...°C
6	N:190828	Time:130002	T1= 27.4°C	T2= 27.3°C	T3= ...°C	T4= ...°C	T5= ...°C	T6= 25.7°C	T7= 25.5°C	T8= ...°C
7	N:190828	Time:130012	T1= 27.4°C	T2= 27.4°C	T3= ...°C	T4= ...°C	T5= ...°C	T6= 25.8°C	T7= 25.5°C	T8= ...°C
8	N:190828	Time:130022	T1= 27.5°C	T2= 27.4°C	T3= ...°C	T4= ...°C	T5= ...°C	T6= 25.8°C	T7= 25.5°C	T8= ...°C
9	N:190828	Time:130032	T1= 27.5°C	T2= 27.5°C	T3= ...°C	T4= ...°C	T5= ...°C	T6= 25.8°C	T7= 25.5°C	T8= ...°C
10	N:190828	Time:130052	T1= 27.5°C	T2= 27.5°C	T3= ...°C	T4= ...°C	T5= ...°C	T6= 25.8°C	T7= 25.5°C	T8= ...°C
11	N:190828	Time:130052	T1= 27.5°C	T2= 27.5°C	T3= ...°C	T4= ...°C	T5= ...°C	T6= 25.8°C	T7= 25.5°C	T8= ...°C
12	N:190828	Time:130102	T1= 27.5°C	T2= 27.5°C	T3= ...°C	T4= ...°C	T5= ...°C	T6= 25.8°C	T7= 25.5°C	T8= ...°C
13	N:190828	Time:130112	T1= 27.5°C	T2= 27.5°C	T3= ...°C	T4= ...°C	T5= ...°C	T6= 25.8°C	T7= 25.5°C	T8= ...°C
14	N:190828	Time:130122	T1= 27.5°C	T2= 27.5°C	T3= ...°C	T4= ...°C	T5= ...°C	T6= 25.8°C	T7= 25.5°C	T8= ...°C
15	N:190828	Time:130132	T1= 27.5°C	T2= 27.5°C	T3= ...°C	T4= ...°C	T5= ...°C	T6= 25.8°C	T7= 25.5°C	T8= ...°C
16	N:190828	Time:130142	T1= 27.5°C	T2= 27.5°C	T3= ...°C	T4= ...°C	T5= ...°C	T6= 25.8°C	T7= 25.5°C	T8= ...°C
17	N:190828	Time:130152	T1= 27.5°C	T2= 27.5°C	T3= ...°C	T4= ...°C	T5= ...°C	T6= 25.8°C	T7= 25.5°C	T8= ...°C

Rysunek 4. Widok zakładki Tryb tekstowy

początkowi importowanych danych i Końcowi importowanych danych. Podany czas należy traktować orientacyjnie. Będzie on zgodny z rzeczywistymi zapamiętanymi w pliku danymi, o ile w trakcie rejestracji nie doszło do zaniku zasilania i utraty cykli pomiarowych. Po naciśnięciu ikony klawisza Import danych rozpocznie się przesyłanie danych. Zapelnienie paska postępu na dole karty informuje o ilości przesłanych danych.

Zakładki Tryb tekstowy i Tryb graficzny

Służą do prezentacji danych pomiarowych zaimportowanych do programu Multi Rejestrator. Na **rysunku 4** pokazano wygląd zakładki Tryb tekstowy z pomiarami wyświetlonymi w formie tabeli. Każda pozycja zawiera zebrane dane jednego pomiaru. W kolejnych polach podana jest data i czas pomiaru oraz informacje o temperaturach odczytanych ze wszystkich 8 kanałów. Jeżeli do kanału nie był podłączony czujnik albo wystąpił błąd odczytu, zamiast cyfr

temperatury wyświetlane są znaki podkreślenia. Wskazanie kursorem pola z pomiarem i kliknięcie lewym przyciskiem myszy spowoduje zaznaczenie pola na czerwono i pominięcie jego zawartości podczas prezentacji graficznej wykresów temperatur. Możliwe są także wykluczenia grupowe. Po wpisaniu np. wartości „37” w polu edycyjnym na dole zakładki Tryb tekstowy i kliknięciu ikony przycisku Zaznacz pomiar, wszystkie komórki tabeli zawierające odczyty z temperaturą 37°C zostaną zaznaczone na czerwono i pominięte przy sporządzaniu wykresu temperatur. Kliknięcie ikony przycisku Usuń zaznaczenie kasuje wszystkie zaznaczenia.

Pomiary zgromadzone w tabeli prezentowane są w formie wykresu zarejestrowanych temperatur na zakładce Tryb graficzny, której przykładowy wygląd pokazano na **rysunku 5**. Zakresy na osiach temperatury i liczby pomiarów dopasowują się automatycznie do prezentowanych wykresów temperatur. Ikony kolorowych przycisków z cyframi od 1 do 8 włączają lub wyłączają

wyświetlanie pomiarów kanału o odpowiadającym numerze. Jeżeli w trakcie rejestracji czujnik przez pewien czas nie był prawidłowo odczytywany, na wykresie temperatur kanału pojawi się przerwa.

Zapis i odczyt plików pomiarów

Pomiary zgromadzone w tabeli na zakładce Tryb tekstowy można zapisać jako plik historyczny na twardym dysku komputera. Po naciśnięciu przycisku na pasku narzędziowym Zapis pliku pomiarów na dysk, należy podać katalog i nazwę pliku, pod jaką pomiary z tabeli mają zostać zapisane. W podanej lokalizacji na dysku zostanie utworzony plik tekstowy z danymi.

Po naciśnięciu przycisku na pasku narzędziowym Odczyt z dysku pliku pomiarów można odczytać dane zapisanego wcześniej pliku pomiarów historycznych. Dane zostaną umieszczone w tabeli i można je będzie oglądać w formie wykresów temperatur.

Taki sam sposób odczytu można zastosować w przypadku plików zgromadzonych na pendrive, który po wyjęciu z gniazda rejestratora należy podłączyć do wolnego portu USB komputera. To szybszy sposób odczytu danych niż ich import, gdy pendrive znajduje się w gnieździe płytki rejestratora.

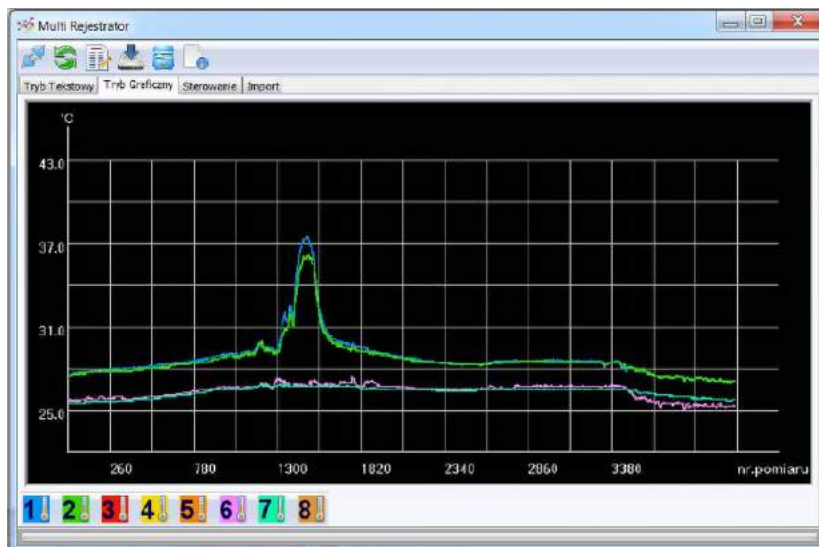
Funkcje zapisu i odczytu plików są aktywne nawet wtedy, gdy rejestrator nie zostanie podłączony do komputera. Objętość pojedynczego pliku pomiarów jest ograniczona do 1 MB.

Format pliku pomiarów

Plik pomiarów jest czystym plikiem tekstowym. Składają się na niego dane kolejnych pomiarów zapisywane ze stałą liczbą znaków:

N:190824;Time:143143;T1= ...°C;T2= ...°C;T3= 24.8°C;T4= 26.3°C;T5= 26.0°C;T6= ...°C;T7= ...°C;T8= 24.6°C; CR LF

Na początku znajduje się informacja o dacie pomiaru w formacie 2 cyfry roku, 2 cyfry miesiąca i 2 cyfry dnia. Dalej w podobny sposób zapisywana jest informacja o czasie



Rysunek 5. Widok zakładki Tryb graficzny

REKLAMA

Specjalistyczne szkolenia dla elektroników i automatyków

TECHDAYS

techdays@techdays.pl
TECHDAYS.PL

CERTYFIKOWANY PARTNER SZKOLENIOWY

Tabela 1. Rozkazy serwisowe

Rozkazy serwisowe	Odpowiedź rejestratora	Opis
+++	OK\r	Inicjacja trybu serwisowego
ATO\r	OK\r	Zakończenie trybu serwisowego
AT/N\r	AT/N<xxxx-xxxx-xxxx-xxxx-xxxx-xxxx\r	Odczyt numeru urządzenia, gdzie x to kolejne znaki HEX numeru
AT/V\r	AT/V<xyz\r	Odczyt wersji oprogramowania, gdzie xyz to znaki wersji
ATSS230=HH:MM:SS\r	OK\r	Przesłanie czasu do ustawienia zegara RTC
ATS230>\r	ATSS230<HH:MM:SS\r	Odczyt czasu zegara RTC
ATSS231=RRRR-MM-DD-T	OK\r	Przesłanie daty do ustawienia RTC rok-miesiąc-dzień-dzień tygodnia (1 – poniedziałek ... 7 – niedziela)
ATS231>\r	ATSS231=RRRR-MM-DD-T\r	Odczyt daty zegara RTC
ATS500= b=s b=s b=s b=s\r	OK\r	Zapis bitów sterujących, gdzie: b to numer bitu: 1 – START/STOP cyklu pomiarów, 4 – aktywacja przycisku na płytce s to stan bitu: 0 lub 1
ATS500>\r	ATSS500< b=s b=s b=s b=s b=s b=s b=s\r	Odczyt bitów statusu, gdzie b to numer bitu: 1 – trwa cykl pomiarów, 4 – aktywny przycisk na płytce, 5 – błąd zapisu do pamięci, 6 – nieobsługiwany typ czujnika, 7 – utrata co najmniej 1 cyklu pomiarów
ATS501=HH:MM:SS\r	OK\r	Ustawienie czasu cyklu pomiarów
ATS501>\r	ATSS501< HH:MM:SS\r	Odczyt ustawionego czasu cyklu pomiarów
ATS502>\r	ATSS502< RRRR-MM-DD\r	Odczyt daty startu ostatniego cyklu pomiarów
ATS503>\r	ATSS503< HH:MM:SS\r	Odczyt czasu startu ostatniego cyklu pomiarów
ATS504>\r	ATSS504< 123\r	Odczyt bieżącego numeru pomiaru
ATS505>\r	ATSS505< abc\r	Odczyt ścieżki dostępu do pliku bieżącego cyklu pomiarów

rejestracji: godzina, minuta, sekunda. Następnie w kolejności od 1 do 8 podawane są temperatury odczytane z kolejnych kanałów pomiarowych. Za etykietą kanału np. „T1=” występuje spacja lub znak minus w przypadku temperatur ujemnych. Trzy kolejne pozycje przeznaczone są na cyfry temperatury w stopniach. Po znaku kropki jedno miejsce jest zarezerwowane na cyfrę części dziesiętnej stopnia. W przypadku braku prawidłowego odczytu zamiast cyfr pojawiają się znaki podkreślenia. Za cyframi temperatury kanału występują dwa znaki: apostrof oraz „C”. Kolejne pozycje danych rozdzielone są znakiem średnika. Linia danych pomiarowych zakończona jest kodami powrotu karetki i nowej linii, w przykładzie zaznaczonymi jako CR i LF. Bezpośrednio za tymi kodami zapisywane są dane temperatur zarejestrowane w kolejnym cyklu pomiarowym.

Przeprowadzanie pomiarów

Typowy proces przeprowadzania pomiarów może wyglądać następująco:

1. Płytkę rejestratora podłączana jest do komputera, na którym został uruchomiony program Multi Rejestrator.
2. Po nawiązaniu połączenia między programem a płytką można sprawdzić i ewentualnie skorygować: ustawienie zegara RTC rejestratora i ustawienie okresu, jaki ma upłynąć pomiędzy kolejnymi pomiarami.
3. Zaznaczane jest pole opcji zezwalającej na wykorzystanie klawisza STARTU/STOPU cyklu rejestracji.
4. Płytkę rejestratora jest odłączana od komputera i przenoszona w docelowe miejsce.

5. Do zacisków gniazd wybranych kanałów podłączane są czujniki. W minimalnej konfiguracji należy podłączyć masę każdego z czujników i wyprowadzenie magistrali 1-Wire (na płytce są to środkowe zaciski gniazd, oznaczone cyfrą odpowiadającą numerowi kanału). Dodatkowo, jeśli są dostępne, można połączyć wyprowadzenia VCC czujników temperatury 18B20 z wyprowadzeniami +5 V na płytce.

6. Do gniazd USB płytki należy podłączyć pendrive i zasilanie. Jeżeli dioda STATUS-u LED11 migoce w sekwencji błysków: dłuższy i dwa krótsze, przycisk START/STOP S1 na płytce jest aktywny.
7. Po jego naciśnięciu i przytrzymaniu przez około 3 s migotanie diody STATUS-u powinno ustać. Rozpocznie się cykl pomiarowy a diody kanałów, do których są podłączone czujniki temperatury, powinny się świecić bez przerwy.

W czasie trwania cyklu pomiarów można na bieżąco kontrolować status rejestracji a także ściągać cząstkowe wyniki pomiarów. Wymaga to podłączenia do gniazda X2 kabla USB. Będzie on służył jednocześnie do zasilania i do komunikacji z komputerem, na którym jest uruchomiony program Multi Rejestrator. Rejestracja kończy się po ponownym naciśnięciu przez 3 sekundy klawisza S1 na płytce. Diody sygnalizacyjne kanałów powinny zgasnąć a dioda STATUS-u zacznie migotać.

Opis rozkazów serwisowych i poleceń sterujących

Do komunikacji i sterowania rejestratorem wykorzystywana jest transmisja szeregową obsługiwana portem UART1 kontrolera

U1. Parametry transmisji są stałe: szybkość 115200 bodów, 8 bitów danych, bez parzystości. Tą drogą przesyłane są rozkazy serwisowe i polecenia. Protokół opiera się na wysyłaniu przez program nadzorujący komend i oczekiwaniu na odpowiedź rejestratora.

Ponieważ rozkazy istotnie wpływają na funkcjonowanie urządzenia, są przez nie przyjmowane tylko w trybie serwisowym. Inicjacja trybu serwisowego następuje po wysłaniu 3 znaków „+++” poprzedzonych i zakończonych okresem ciszy równym co najmniej czasowi wysłania 2 znaków przy szybkości 115200. Rejestrator wchodzi w tryb serwisowy, potwierdzając swój stan komunikatem „OK\r” i trwałym zapaleniem diody STATUS-u. Jeżeli w ciągu 10 sekund nie zostaną przesłane kolejne prawidłowo zdekodowane rozkazy, tryb serwisowy automatycznie jest zamykany, o czym informuje zgaszenie diody LED STATUS-u. Każdy odebrany rozkaz automatycznie przedłuża czas trwania trybu serwisowego o kolejne 10 sekund. Wyjątkiem jest rozkaz „ATO\r”, który wymusza zakończenie trybu serwisowego.

Z wyjątkiem transmisji „+++” wszystkie rozkazy i potwierdzenia kończone są kodem powrotu karetki „\r” CR. W tabeli 1 zostały zebrane wszystkie rozkazy serwisowe, na które odpowiada rejestrator.

Polecenia sterujące nie wpływają na bieżące ustawienia rejestratora. Związane są z importem danych z plików na pendrive i są obsługiwane w trybie normalnej pracy, także gdy aktywny jest cykl pomiarowy. Wszystkie polecenia i potwierdzenia kończone są kodem powrotu karetki „\r” CR.

„PLIK_LIST\r” polecenie odczytu informacji o dostępnych plikach zapisanych w katalogu pomiarów na pendrive.

Zależnie od sytuacji rejestrator może odpowiedzieć następującym komunikatem:

path ST:RRMMDD,GGMMSS;SP:RRMMDD,GGMMSS;IE:xx;RR:xx\r

gdzie:

path – ścieżka do pliku pomiarów

ST: – data RRMMDD, czas GGMMSS startu pomiarów zapisanych w pliku

SP: – data RRMMDD, czas GGMMSS zakończenia pomiarów zapisanych w pliku

IE: – liczba zapisanych w pliku cykli pomiarów

RR: – rozmiar pliku w bajtach

Po odebraniu danych można do rejestratora wysłać komunikat „OKF\r” oznaczający oczekiwanie na przesłanie danych o kolejnym pliku.

Gdy przesłane zostaną informacje o wszystkich dostępnych plikach lub na pendrive nie ma żadnych plików pomiarów albo odczyt z pendrive'u jest niemożliwy, rejestrator przesyła w odpowiedzi komunikat „OK\r”.

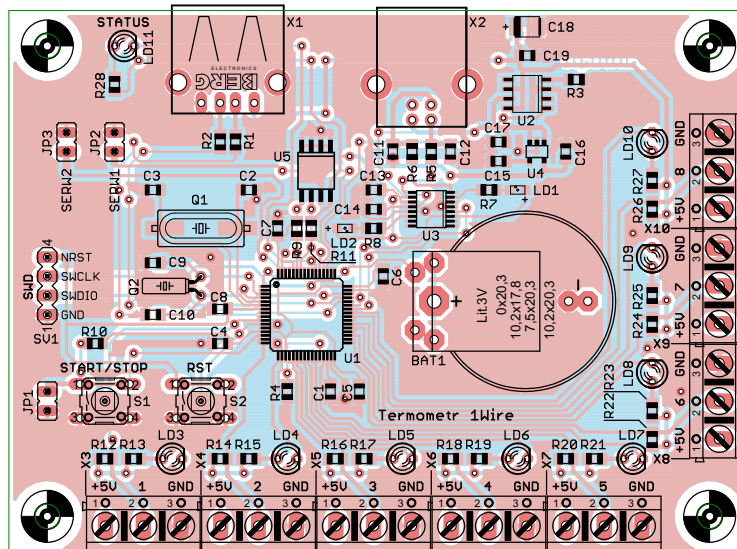
„PLIK_RETRY\r” polecenie przesłania pomiarów z ostatnio dopisywanego pliku lub

„PLIK_RETRY path\r” polecenie przesłania pomiarów z pliku wskazanego ścieżką dostępu path, lub

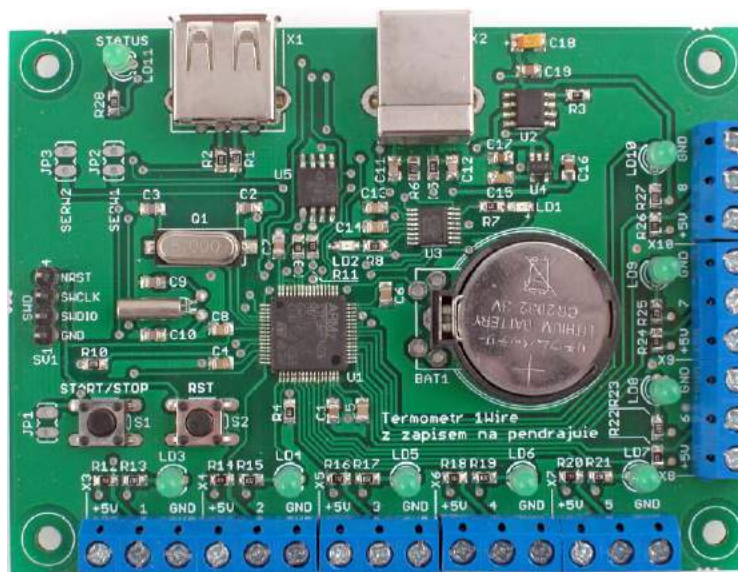
„PLIK_RETRY path;start;stop\r” polecenie przesłania pomiarów z pliku wskazanego ścieżką dostępu path od numeru pomiaru start do numeru pomiaru stop. W odpowiedzi rejestrator przesyła zapis kolejnego pojedynczego pomiaru. Po odebraniu danych można do rejestratora wysłać komunikat „OKF\r” oznaczający zachętę do przesłania kolejnej pozycji pomiarów w pliku. Jeżeli zostały przesłane wszystkie dane pomiarowe, rejestrator odpowiada komunikatem „OK\r”.

Montaż

Rejestrator mieści się na pojedynczej płytce drukowanej o wymiarach 98×72 mm, której



Rysunek 6.



Fotografia 7.

schemat wraz z rozmieszczeniem elementów pokazano na **rysunku 6**. Zbudowany jest głównie z elementów SMD o małym rozmiarze, więc montaż wymaga doświadczenia

i dokładności. Pomocna przy montażu będzie **fotografia 7**, na której pokazano zmontowaną płytkę.

Ryszard Szymaniak, EP

REKLAMA

bądźmy w kontakcie
na Facebook:

<http://bit.ly/320J9Be>

Rysino

plytka ewaluacyjna z FPGA Intel MAX10

Minimalistyczna płytka ewaluacyjna do prototypowania własnych projektów opartych na FPGA z rodziny Intel MAX10. Doskonale sprawdzi się jako baza do tworzenia własnych konstrukcji bazujących na tym ciekawym układzie.

Dodatkowe materiały do pobrania ze strony www.media.avt.pl

W ofercie AVT* AVT-5726

Podstawowe parametry:

- zastosowany układ FPGA to 10M04SAE144C8G z rodziny Intel MAX10,
- układ FPGA wyposażony w cztery tysiące elementów logicznych LE,
- pamięć RAM o rozmiarze 189 Kb pamięci (w 21 blokach po 9 Kb),
- 1248 Kb pamięci Flash dzielonej pomiędzy bitstream oraz aplikację użytkownika,
- programowanie poprzez interfejs JTAG przy użyciu, np. programatora USB Blaster,
- wbudowany konwerter USB-UART,
- zasilanie poprzez złącze miniUSB.

Wykaz elementów:

Rezystory:

R1...R17: 10 kΩ SMD0603

Kondensatory:

C1, C2: 10 μF SMD0405
C3...C10: 100 nF SMD0603

Półprzewodniki:

LED1...LED9: zielona SMD0603
IC1: 10M04SAE144C8G
IC2: FT230XS
IC3: LF33CDT
QG1: generator kwarcowy 8 MHz DIL08S

Pozostałe:

USB: złącze mikroUSB SMD
RST: uSwitch
S1: dipSwitch poczwórny
JTAG: wtyk kątowy goldpin 2x5
ARDUINO: gniazda goldpin 6, 6, 8, 10 pinów

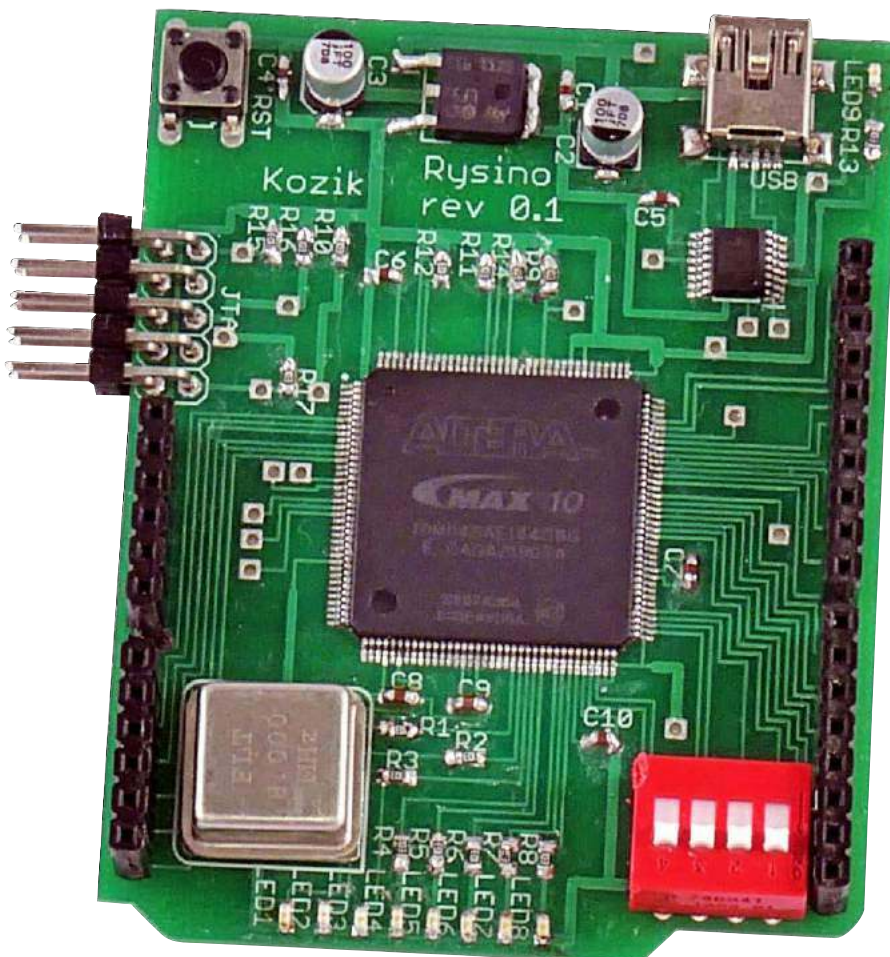
Projekty pokrewne na www.media.avt.pl:

- AVT-5574 Płytki ewaluacyjne dla STM32F2/F4/F7 do celów SDR i nie tylko (EP 2/2017)
- AVT-5529 Zestaw uruchomieniowy z mikrokontrolerem ATXmega256A3U (EP 2/2016)
- AVT-1875 Kieszonkowa płytka prototypowa (EP 8/2015)
- AVT-1675 STM32duino – kompatybilna z Arduino płytka z STM32F103C8T6 (EP 5/2012)
- AVT-1620 Cortexino – Kompatybilna z Arduino płytka z LPC1114 (EP 5/2011)
- AVT-5288 Zestaw ewal. dla FPGA (EP 4/2011)
- AVT-1610 Minimoduł z ATtiny 2313 (EP 3/2011)
- AVT-5275 ZEPIC – Zestaw ewaluacyjny dla mikrokontrolerów PIC (EP 2/2011)

Uwaga! Elektroniczne zestawy do samodzielnego montażu. Wymagana umiejętność lutowania!

Kytagawa wersja zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wstawić w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu. Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wstawione w płytkę PCB)
 - wersja [A] – płytka drukowana bez elementów i dokumentacji Kity w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
 - wersja [A+] – płytka drukowana [A] + zaprogramowany układ [UK] i dokumentacja
 - wersja [UK] – zaprogramowany układ
- Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! <http://sklep.avt.pl>. W przypadku braku dostępności na <http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.



Układy FPGA z rodziny MAX10 mają kilka ciekawych rozwiązań, rzadko występujących w innych podzespołach tego typu, które przekonują do stosowania tych układów we własnych projektach. Pierwszą zaletą jest zintegrowana nieulotna pamięć Flash. Pozwala nie tylko na przechowywanie konfiguracji, ale może być także dostępna dla aplikacji w czasie pracy układu. Drugim elementem ułatwiającym stosowanie układów MAX10 jest możliwość zasilania za pomocą tylko jednego napięcia 3,3 V. Równie istotną cechą jest dostępność podzespołów w łatwej do montażu obudowie EQFP. Dzięki temu zaprojektowanie i wykonanie PCB jest znacznie łatwiejsze niż w przypadku podobnych układów w obudowie BGA. Układy są dostępne w wersjach zawierających od 2 do 50 tysięcy elementów logicznych (LE – *logic element*), zatem umożliwiają realizację dość zaawansowanych projektów.

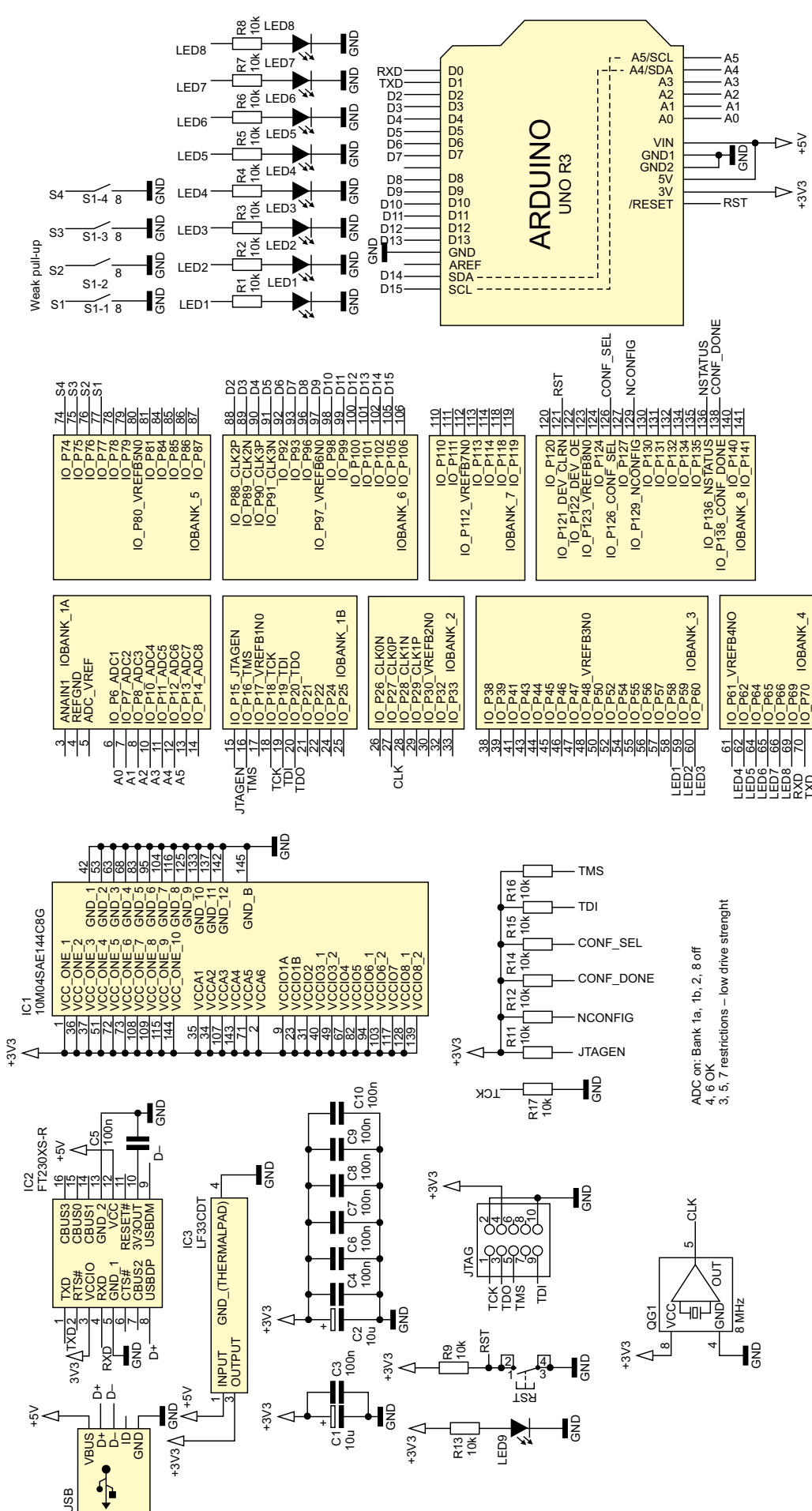
Budowa

Schemat układu pokazano na **rysunku 1**. Zasilanie jest dostarczone przez port USB.

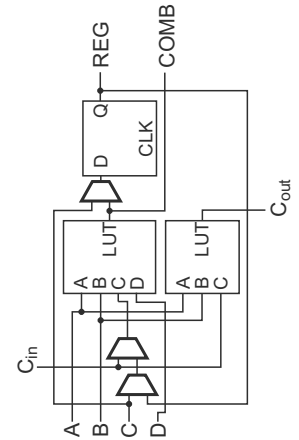
Napięcie jest obniżane do 3,3 V za pomocą stabilizatora LF33 (IC3). Dioda LED9 sygnalizuje podłączenie zasilania, a kondensatory C1...C10 odpowiadają za jego filtrację.

Najważniejszą częścią projektu jest 10M04SAE144C8G – tak oznaczony jest układ FPGA z rodziny Intel MAX10 (IC1). Zastosowana wersja wyposażona jest w cztery tysiące elementów logicznych. Schemat pojedynczego LE pokazano na **rysunku 2**. Składa się z dwóch tablic LUT oraz jednego przerzutnika D. Wartość z pierwszej, czterobitowej tablicy może być wyprowadzona na zewnątrz albo zatrzaśnięta w przerzutniku. Natomiast druga, trzybitowa pozwala na wyznaczenie przeniesienia, które za pomocą szybkiej ścieżki jest połączone z sąsiednim LE.

Wewnątrz układu zawarta jest pamięć RAM o rozmiarze 189 Kb pamięci (w 21 blokach po 9 Kb) oraz 1248 Kb pamięci Flash dzielonej pomiędzy *bitstream* oraz aplikację użytkownika. Dostępnych jest także 20 bloków mnożących 18-bitowe liczby oraz jeden przetwornik analogowo-cyfrowy.



Rysunek 1. Schemat ideowy układu

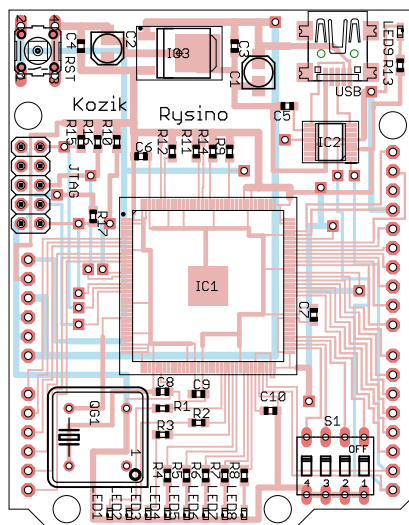


Rysunek 2. Uproszczony schemat pojedynczego elementu logicznego (LE) w układzie MAX10

Programowanie jest możliwe poprzez interfejs JTAG wyprowadzony na złączu goldpin. Do programowania można użyć programatora USB Blaster. Rezystory R10...R12 i R14...R17 ustalają odpowiednie stany na wejściach konfiguracyjnych. Więcej na ten temat można się dowiedzieć z dokumentacji [1], natomiast pełną bazę wiedzy znajdziemy na stronie [2].

Poza układem FPGA na płytce znajduje się kilka podstawowych peryferiów ułatwiających rozpoczęcie własnych eksperymentów. Generator kwarcowy QG1 dostarcza stabilny sygnał zegarowy o częstotliwości 8 MHz, który może być użyty do taktowania układu, bądź jako referencja dla wewnętrznej pętli PLL. Diody LED1...LED8 oraz przełącznik dip switch S1 są podłączone bezpośrednio do FPGA. Piny obsługujące włączniki muszą mieć włączony wewnętrzny rezystor podciągający do plusa zasilania. Część pinów została także wyprowadzona na złącze zgodne z Arduino. Należy jednak pamiętać, że maksymalne napięcie, jakie można na nie doprowadzić, to 3,3 V.

Złącza A0...A5 mogą służyć jako wejścia dla przetwornika ADC. Jego włączenie powoduje ograniczenia przy korzystaniu z niektórych portów wejścia/



Rysunek 3. Schemat montażowy

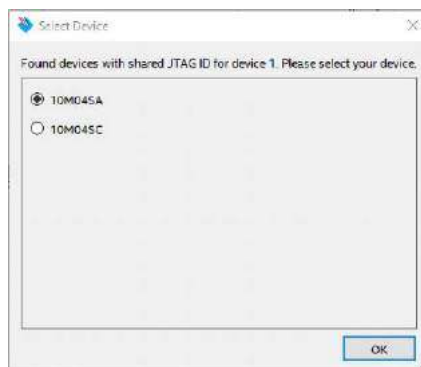
wyjścia. Bloki 1a, 1b, 2 i 8 są wtedy niedostępne, natomiast liczba pinów z bloków 3, 5 i 7 jest ograniczona. Konfiguracja ADC nie ma wpływu jedynie na bloki 4 i 6.

Przycisk RST jest podłączony do wejścia DEV_CLRn. Domyślnie pracuje ono jako normalny pin, jednak ma także specjalną opcję. Jeżeli włączymy tryb *device wide reset*, to stan niski na nim spowoduje reset wszystkich przerzutników. Dzięki układowi FT230 (IC2) port USB nie służy tylko do zasilania, ale umożliwia także komunikację poprzez port szeregowy. Linie Tx i Rx są podłączone bezpośrednio do MAX10.

Schemat płytki jest także dostępny na licencji BSD i można go pobrać z repozytorium [3].

Montaż i uruchomienie

Projekt dwustronnej płytki drukowanej pokazano na **rysunku 3**, a zdjęcie gotowego układu prezentuje **fotografia tytułowa**. Montaż warto rozpocząć od fragmentu odpowiedzialnego za zasilanie. Najpierw lutujemy złącze USB, stabilizator, kondensatory C1...C4 oraz diodę LED9 i opornik R13.

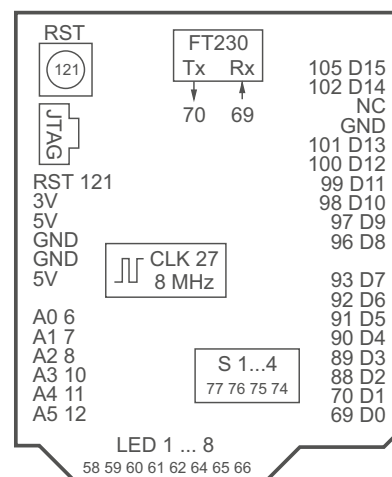


Rysunek 4. Wybór typu układu podczas wykrywania urządzenia

Po sprawdzeniu, czy nie występują zwarcia, podłączamy kabel USB. Powinna zaświecić się dioda. Woltomierzem sprawdzamy, czy napięcie na wyjściu stabilizatora jest równe 3,3 V. Warto także sprawdzić, czy światło diod LED nie jest zbyt intensywne – w początkowej wersji układu do ograniczenia prądu zastosowałem rezystory 330 Ω, jednak emitowane światło było zbyt jasne. Dopiero po zmianie rezystorów na 10 kΩ uzyskałem odpowiedni efekt.

Następnie przechodzimy do układu IC2. Po sprawdzeniu poprawności jego montażu podłączamy układ do komputera PC. System operacyjny powinien wykryć konwerter FT230. Teraz możemy przystąpić do lutowania układu FPGA i pozostałych komponentów. Po końcowym sprawdzeniu połączeń za pomocą testera zwarć podłączamy zasilanie.

Teraz potrzebny będzie programator USB Blaster, należy dołączyć go do płytki. Można go nabyć na popularnych serwisach aukcyjnych: zarówno polskich, jak i zagranicznych. Po wybraniu opcji *Auto Detect* w programie Quartus Prime Programmer układ powinien zostać wykryty. W oknie dialogowym (**rysunek 4**) wybieramy opcję 10M04SA. Do przetestowania płytki można wykorzystać przykładowy projekt znajdujący się także w repozytorium [3]. Wygenerowany wsad



Rysunek 5. Numery pinów układu FPGA podłączonych do poszczególnych peryferiów

znajduje się w pliku *board_test/board_test.sof*. W demonstracyjnym bitstreamie cztery diody są sterowane za pomocą przełączników. Pozostałe zostały podłączone do wyjść licznika binarnego. Linie Tx i Rx portu szeregowego zostały „zwarte” wewnątrz układu FPGA, powodując powstanie „echa”. Oznacza to, że jeśli otworzymy port com za pomocą programu PuTTY, w konsoli powinien pojawiać się dokładnie taki sam tekst, jaki wpisujemy na klawiaturze.

Układ jest gotowy do pracy. Aby stabilnie stał na stole, do spodniej strony PCB można dokleić gumowe nóżki.

Podczas uruchamiania własnych projektów pomocna może okazać się rozpiska numerów pinów podłączonych do różnych peryferiów pokazana na **rysunku 5**.

Rafał Kozik
rafkozik@gmail.com

Bibliografia

- [1] Intel® MAX® 10 FPGA Device Family Pin Connection Guidelines, <https://intel.ly/2NemCeS>
- [2] <https://intel.ly/2JoQqEw>
- [3] <http://bit.ly/33Xec2e>

REKLAMA

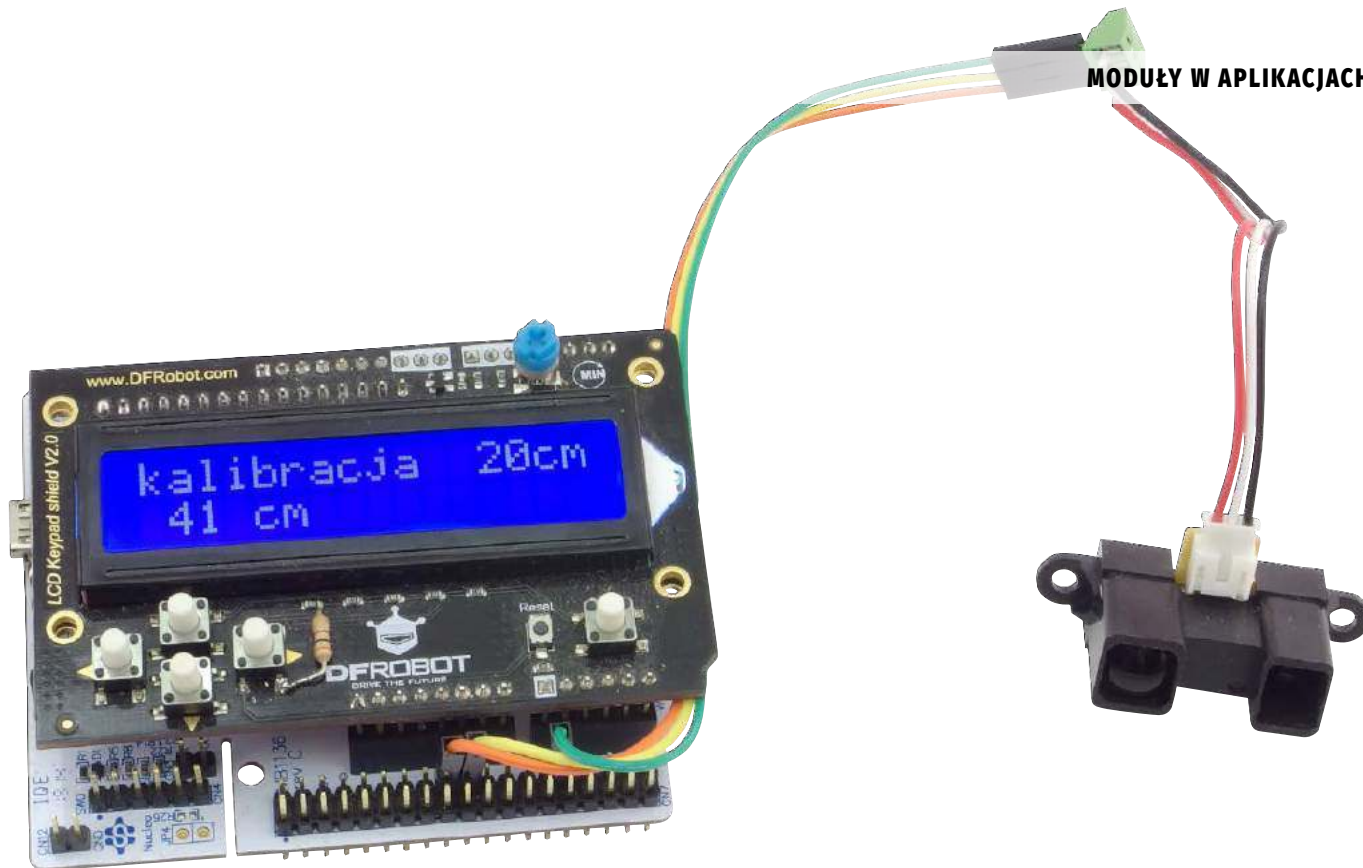
Wstęp do Klubu AVT Elektronika

będziesz miał prawo do korzystania z szeregu przywilejów:

- do 50% zniżki w Sklepie AVT
- darmowe prenumeraty Wydawnictwa AVT
- do 50% zniżki w Ulubionym Kiosku
- Zapraszamy do zapoznania się z zasadami Klubu!



<http://bit.ly/2GaDwtQ>



Miernik odległości na bazie czujnika GP2Y0A02YK0F

GP2Y0A02YK0F – tą kombinacją liter i cyfr oznaczono czujnik firmy Sharp przeznaczony do wykrywania obiektów w zasięgu od 20 do 150 cm. Element działa jak dalmierz i można go zastosować do pomiaru odległości do obiektu.

Czujnik emituje wiązkę podczerwieni. Na podstawie sygnału odbitego od obiektu, na wyjście czujnika podawane jest napięcie o poziomie proporcjonalnym do odległości do obiektu.

Podstawowe parametry czujnika są następujące:

- użyteczny zakres pomiarowy: od 20 cm do 150 cm,
- wyjście napięcie o poziomie od 2,5 V do 0,5 V, proporcjonalne do zmierzonej odległości,
- zasilanie od 4,5 V do 5,5 V,
- średni pobór prądu wynosi 33 mA (w szczycie impulsu wartość jest większa),
- wymiary 44,5×18,9×21,6 mm,
- złącze 3-stykowe typu JST PH.

Czujnik zamieszczono na **fotografii 1**, opis wyprowadzeń złącza JST PH prezentuje **rysunek 2**.

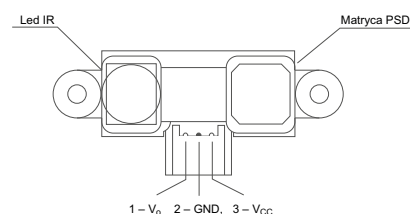
Budowa i działanie

Na **rysunku 3** pokazano schemat blokowy czujnika. LED oznacza diodę wytwarzającą skupioną wiązkę światła podczerwonego wysyłaną w kierunku obiektu. Po odbiciu światło pada na matrycę detektora PSD (*Position Sensing Device*). Moduł jest wyposażony w wewnętrzny oscylator i stabilizator, do którego podłączone jest wejście zasilania Vcc. Sygnał proporcjonalny do wyniku pomiaru wytwarzany jest w układzie pomiarowym (*Signal processing circuit*) i poprzez bufor wyprowadzany na wyjście V_o.

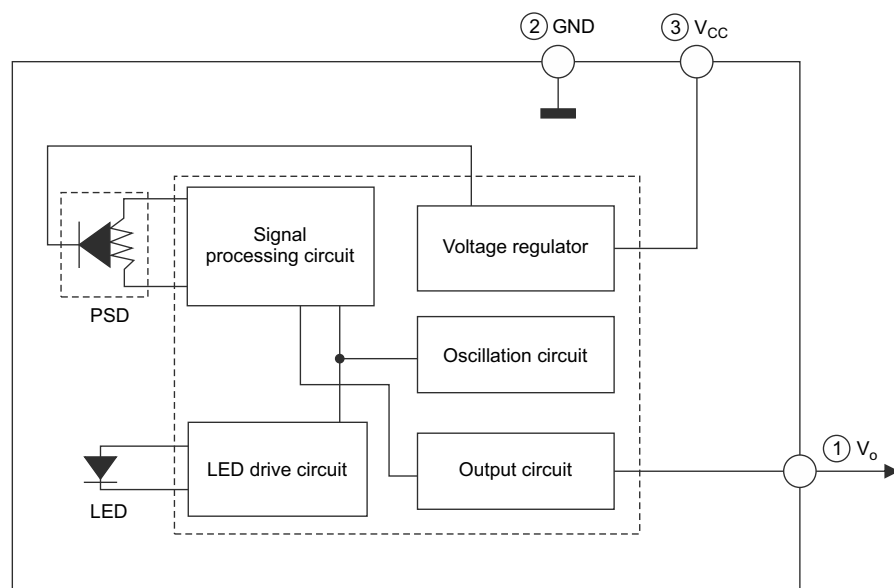
Do obliczenia dystansu pomiędzy czujnikiem a obiektem służy metoda triangulacji. Istotą metody i działanie czujnika objaśnia **rysunek 4**. Dioda Led IR wytwarza wiązkę światła podczerwonego skupionego przez soczewkę do rozmiarów niewielkiej plamki.



Fotografia 1. Wygląd czujnika



Rysunek 2. Rozmieszczenie i opis wyprowadzeń



Rysunek 3. Wewnętrzny schemat blokowy czujnika

Po odbiciu od przeszkody, plamka światła pada na matrycę PSD. W zależności od odległości, w jakiej znajduje się obiekt, odbite światło pada na matrycę pod innym kątem i w innym miejscu. To przesunięcie, proporcjonalne do odległości obiektu, zamieniane jest na odpowiedni poziom napięcia wyjściowego V_o . Wykres napięcia w funkcji zmierzonej odległości pokazany został na **rysunku 5**.

Ze względu na istotę stosowanej metody uzyskana krzywa ma przebieg logarytmiczny. Z kolei skończona rozdzielczość

użytej matrycy PSD sprawia, że wraz ze wzrostem mierzonej odległości spada nachylenie krzywej i pogarsza się precyzja i rozdzielczość pomiaru. W przypadku najkrótszych dystansów następuje gwałtowne przegięcie krzywej ograniczające możliwe do realnego zmierzenia odległości do 20 cm.

Podłączenie czujnika

Czujnik został podłączony do płytki Nucleo. Dla uzyskania większej mobilności do zasilania płytki wykorzystano zewnętrzny zasilacz dostarczający napięcie 7...12 V podawane na wyprowadzenie VIN. Zestawienie wszystkich koniecznych połączeń zebrano w **tabeli 1**. Ze względu na impulsowy pobór prądu przez czujnik producent zaleca podłączyć kondensator filtrujący 10 μ F jak najbliżej wyprowadzeń Vcc i GND modułu. Jego zastosowanie poprawiło nieco parametry sygnału wyjściowego, jednak w tej konfiguracji nie okazało się niezbędne. Do tej samej płytki Nucleo, do złącza typu arduino, podłączono moduł wyświetlacza LCD do prezentacji wyników prowadzonych pomiarów.

Procedury programowe

Czujnik GP2Y0A02YK0F wykonuje pomiary nieprzerwanie od momentu podłączenia zasilania. Co około 40 ms odświeżane jest ustawienie napięcia wyjściowego V_o .

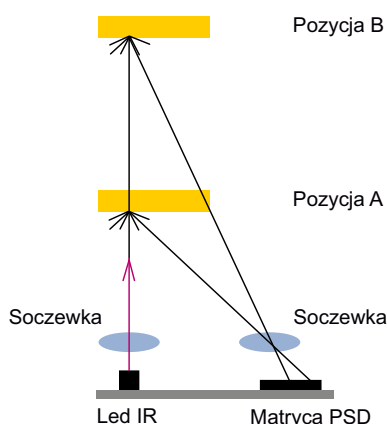
Na **rysunku 6** pokazano zależności czasowe cyklu kolejnych odczytów.

Aby odczytać zmierzoną przez czujnik odległość, należy napięcie z wyjścia V_o podać na wejście przetwornika ADC mikrokontrolera na płytce Nucleo, np. na port PA1. Przetwornik należy skonfigurować do pracy w trybie ciągłej konwersji napięcia podawanego na port PA1. Po zakończeniu kolejnego cyklu konwersji jej rezultat, jako wartość 16-bitowa, powinien być zapisywany przez mechanizm DMA w buforze. Zawartość bufora będzie pobierana przez procedury przeliczające odczytany poziom napięcia na odległość. Na **listingu 1** pokazano deklarację zmiennych oraz procedurę inicjującą. Tabela stałych `dist_volt_const_tab[]` zawiera wartości, które będą używane do przeliczeń napięcia na zmierzoną odległość. Wartości stałych odpowiadają zaznaczonym na wykresie punktom (**rysunek 5**).

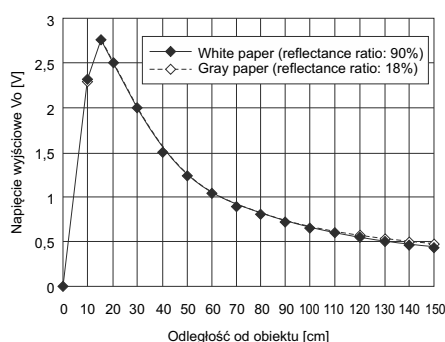
Procedura z **listingu 2** służy do odczytu i uśredniania wartości z przetwornika ADC. Uśrednianie polega na sumowaniu wartości kolejnych próbek. Gdy zostanie odczytana, określona przez zmienną `DALMIERZ_ILE_PROBEK`, liczba próbek, z otrzymanej sumy wyciągana jest średnia arytmetyczna.

Procedura `gp2y0a02_ADctoU()` z **listingu 3** przekształca otrzymaną w wyniku konwersji liczbę z przedziału 0...4095 na wartość napięcia V_o . Liczba 3,3 określa w woltach poziom napięcia referencyjnego przetwornika ADC na płytce Nucleo.

Listing 4 zawiera główną procedurę obsługi czujnika. Zastosowane są tam opisane wcześniej procedury do odczytu z przetwornika ADC i konwersji pomiaru. Następnie w pętli `for()` wyszukiwany jest przedział odległości, w jakim zawiera się uśredniony odczyt z czujnika. Wyszukiwanie polega na porównaniu uśrednionej wartości napięcia V_o z wartościami przepisanyymi z tabeli stałych `dist_volt_const_tab[]`. W tym momencie znana jest odległość z rozdzielczością 10 cm. Do dalszych obliczeń przyjmuje się uproszczone założenie, że krzywa z **rysunku 5** składa się z liniowych odcinków pomiędzy punktami wyznaczającymi kolejne 10-centymetrowe przedziały. Na zasadzie proporcji wyliczana jest odległość między czujnikiem a obiektem, z dokładnością do 1 cm.



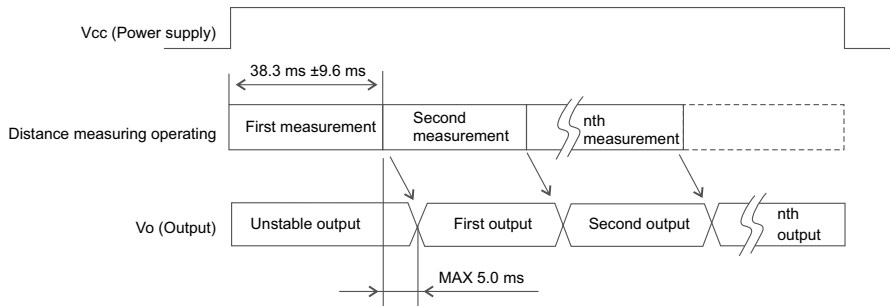
Rysunek 4. Pomiar odległości metodą triangulacji



Rysunek 5. Krzywa zależności napięcia V_o od odległości

Tabela 1. Zestawienie wszystkich koniecznych połączeń

Złącze płytki Nucleo	Podłączenie
CN7-30 port PA1	Wyprowadzenie V_o czujnika
CN7-18 +5 V	Wyprowadzenie Vcc czujnika
CN7-20 GND	Wyprowadzenie GND czujnika
CN7-25 VIN	Zasilanie płytki od +7 V do +12 V
CN7-22 GND	Zasilanie płytki GND
JP5	Zwora w pozycji E5V
JP1	Zwora rozwarta



Rysunek 6. Zależności czasowe cyklu kolejnych odczytów

Listing 1. Deklaracje oraz procedura inicjująca

```
uint32_t dalmierz_srednia;
uint16_t dalmierz_wynik;
uint8_t num_probki;
const double dist_volt_const_tab[]={
    //20 30 40 50 60 70 80 90 100 110 120 130 140 150
    2.5, 2.0, 1.5, 1.25, 1.05, 0.9, 0.82, 0.73, 0.65, 0.6, 0.55, 0.5, 0.45, 0.42
};
double dist_volt_tab[14];

#define DALMIERZ_POMIAR_BRAK -1
#define DALMIERZ_ILE_PROBEK 100
#define DALMIERZ_PRZEDZIAL_MIN 0
#define DALMIERZ_PRZEDZIAL_MAX 14

//inicjacja procedur dalmierza gp2y0a02
uint8_t gp2y0a02_Ini(void)
{
    for (uint8_t x=0; x<DALMIERZ_PRZEDZIAL_MAX; x++)
    {
        dist_volt_tab[x] =dist_volt_const_tab[x];
    }
    dalmierz_wynik =DALMIERZ_POMIAR_BRAK;
    dalmierz_srednia =0;
    num_probki =0;
    return TRUE;
}
```

Listing 2. Procedura odczytu i uśredniania wartości z przetwornika ADC

```
//odczyt uśrednionej wartości z przetwornika ADC
uint16_t gp2y0a02_OdczytSredniejADC(void)
{
    int16_t adc_odczyt =ADC_Odczyt(NumKanal_Dalmierz);
    if (adc_odczyt ==ADC_TRWA_KONWERSJA) return dalmierz_wynik;

    dalmierz_srednia =dalmierz_srednia +adc_odczyt;
    num_probki++;
    if (num_probki >=DALMIERZ_ILE_PROBEK)
    {
        dalmierz_srednia =dalmierz_srednia / num_probki;
        dalmierz_wynik =(uint16_t)dalmierz_srednia;
        dalmierz_srednia =0;
        num_probki =0;
    }
    return dalmierz_wynik;
}
```

Listing 3. Procedura przekształcająca wynik na wartość napięcia

```
//konwersja odczyt ADC -> napięcie
double gp2y0a02_ADCToU(uint16_t adc_odczyt)
{
    double napiecie =(3.3 * (double)adc_odczyt) / 4095;
    return napiecie;
}
```

Listing 4. Główna procedura obsługi czujnika

```
//odczyt zmierzonej odległości
double gp2y0a02_OdczytOdleglosci_cm(void)
{
    int8_t przedzial;
    double odleglosc_cm;

    uint16_t adc_odczyt =gp2y0a02_OdczytSredniejADC();
    double napiecie =gp2y0a02_ADCToU(adc_odczyt);

    for (przedzial=DALMIERZ_PRZEDZIAL_MIN; przedzial<DALMIERZ_PRZEDZIAL_MAX; przedzial++)
    {
        if (napiecie >=dist_volt_tab[przedzial]) break;
    }
    odleglosc_cm =(przedzial-1)*10+20;
    if (przedzial==DALMIERZ_PRZEDZIAL_MIN || przedzial==DALMIERZ_PRZEDZIAL_MAX)
    {
        //pomiar poza dopuszczalnym zakresem
        return odleglosc_cm;
    }
    double delta_v_przedzial =dist_volt_tab[przedzial-1]-dist_volt_tab[przedzial];
    if (delta_v_przedzial<0) return odleglosc_cm;
    double napiecie_d =napiecie -dist_volt_tab[przedzial];
    double prop =1 -(napiecie_d/delta_v_przedzial);
    odleglosc_cm =odleglosc_cm +(prop *10);

    return odleglosc_cm;
}
```

Rozrzut parametrów poszczególnych egzemplarzy czujnika GP2Y0A02YK0F powoduje, że każdy czujnik ma trochę inną charakterystykę pomiarową. Dla zwiększenia precyzji pomiaru można przewidzieć w programie dodatkową procedurę kalibracyjną. Powinna polegać na wykonaniu serii 14 pomiarów, zaczynając od 20 cm do 150 cm, co 10 cm. Odczyty wartości napięcia V_o każdego pomiaru powinny zostać przepisane do tabeli zmiennych `double dist_volt_tab[14]`. Możliwość przeprowadzenia kalibracji jest powodem, dla którego w obliczeniach używa się zmiennych z tej tabeli a nie stałych z tabeli `dist_volt_const_tab[]`.

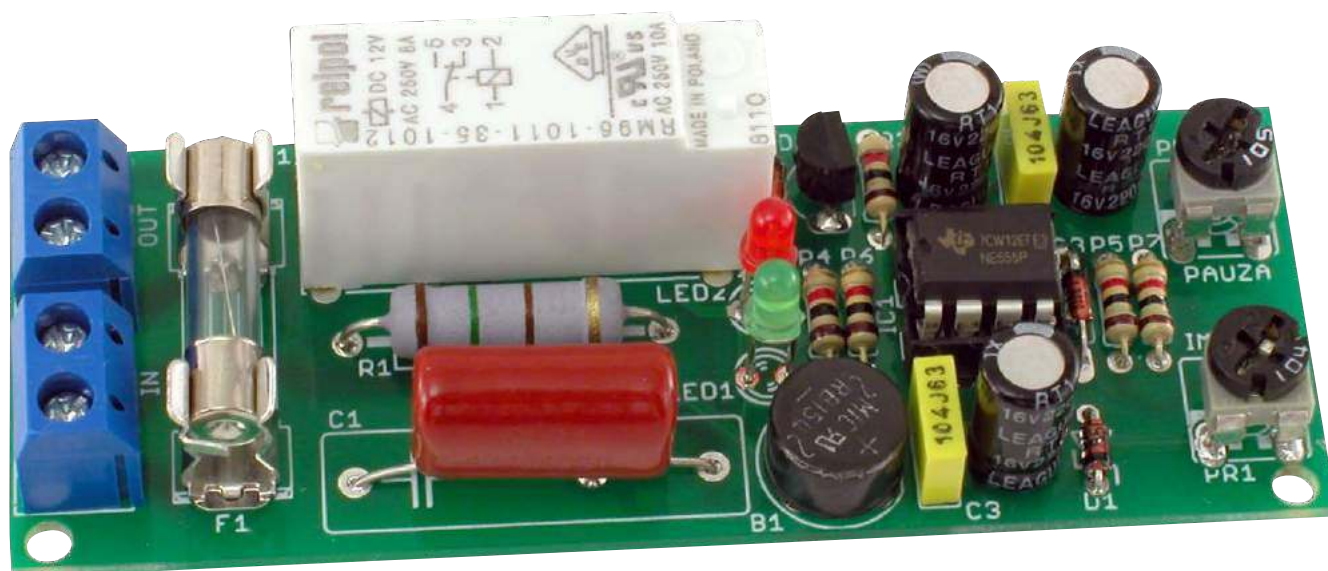
Końcowe wnioski

Czujnik GP2Y0A02YK0F, ze względu na wyjście napięciowe, jest prosty w obsłudze i może pracować z różnymi typami kontrolerów wyposażonych w przetwornik ADC. Cechuje się wąską wiązką podczerwieni i dosyć stabilnymi odczytami, mało wrażliwymi na poziom zewnętrznego oświetlenia. Wadą jest wyraźnie spadająca rozdzielczość przy pomiarach na dystansach dłuższych niż 1 metr.

Ryszard Szymaniak

REKLAMA

Na temat
modułów
w aplikacjach
dyskutuj na
forum.ep.com.pl

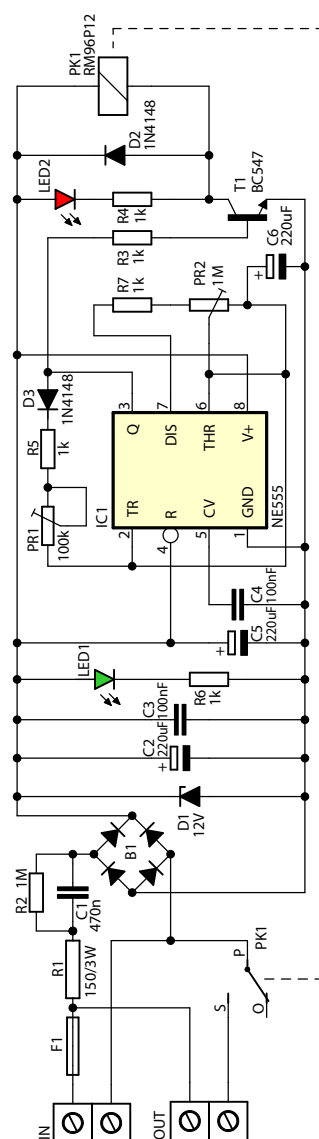


Uniwersalny układ czasowy 230 V

Niezwykle prosty, ale funkcjonalny układ czasowy, który pozwala sterować dowolnym urządzeniem zasilanym z sieci energetycznej 230 V AC w sposób cykliczny, przerywany. Bazuje na nieśmiertelnej i legendarnej kostce NE555 w typowym układzie astabilnym – generator wytwarza impulsy o stałej czasowej określonej przez kombinację kondensatora i rezystorów. Jako element sterujący zastosowano przełącznik.

UWAGA! W układzie występują napięcia niebezpieczne dla zdrowia i życia człowieka

Układ służy do cyklicznego włączania i wyłączania odbiornika dołączonego do wyjścia. Dzięki takiemu działaniu umożliwia np. ograniczenie poboru energii (poprzez zredukowanie wydajności) lub np. cykliczne dozowanie czegoś. Na rysunku 1 pokazano schemat elektryczny. Układ zasilany jest bezpośrednio z sieci 230 V AC za pośrednictwem zasilacza beztransformatorowego. Kondensator C1 ogranicza prąd, który urządzenie może pobrać z sieci energetycznej, a rezystor R1 zabezpiecza mostek B1 przed uszkodzeniem na skutek przeciążenia, które wystąpiłoby w chwili włączenia do sieci. Rezystor R2 służy do rozładowania kondensatora C1 po odłączeniu zasilacza. Mostek prostowniczy został dołączony do ogranicznika



Rysunek 1. Schemat elektryczny układu

napięcia w postaci diody Zenera 12 V, natomiast kondensatory C2 i C3 pełnią funkcję filtra zasilania. Dioda LED1 informuje o obecności napięcia zasilającego.

Głównym elementem układu jest wciąż nieśmiertelny timer NE555, pracujący w jednym z najczęściej stosowanych rozwiązań, czyli w układzie generatora astabilnego generującego impulsy o czasie trwania zależnym od pojemności kondensatora C6, rezystancji

Dodatkowe materiały do pobrania ze strony www.media.avt.pl

W ofercie AVT* AVT-5730

Podstawowe parametry:

- płynna regulacja czasu włączenia i wyłączenia,
- zakres czasu włączenia: 0,5...15 s,
- zakres czasu wyłączenia: 0,5...60 s,
- sygnalizacja pracy diodą LED,
- element wykonawczy: przełącznik,
- max. obciążenie wyjścia: 8 A, 230 V AC,
- zasilanie 230 V AC

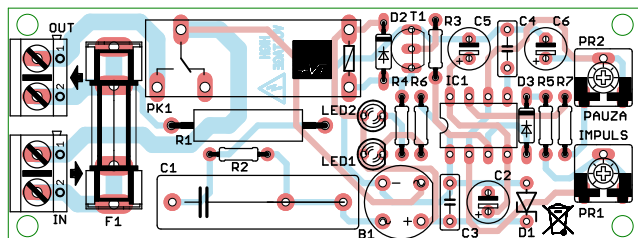
Wykaz elementów:

- R1: 150 Ω/3 W
- R2: 1 MΩ
- R3...R7: 1 kΩ
- C1: 470 nF/250 V
- C2, C5, C6: 220 μF
- C3, C4: 100 nF
- IC1: NE555
- T1: BC547
- D1: dioda Zenera 0,4 W, 12 V
- D2, D3: 1N4148
- LED1, LED2: dioda LED 3 mm
- PK1: przełącznik
- IN, OUT: złącza śrubowe

Uwaga! Elektroniczne zestawy do samodzielnego montażu. Wymagana umiędzynarodowiona! Podstawową wersją zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wylutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu. Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw
- wersja [B] (elementy wylutowane w płytkę PCB)
- wersja [A] – płytkę drukowaną bez elementów i dokumentacji
- wersja [A+] – płytkę drukowaną [A] + zaprogramowany układ
- wersja [UK] – zaprogramowany układ

 Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! <http://sklep.avt.pl> – w przypadku braku dostępności na <http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.



Rysunek 2. Schemat płytki PCB wraz z rozmieszczeniem elementów

rezystora R7 oraz ustawienia suwaka potencjometru PR2. Obwód złożony z kondensatora C5, rezystora R5 oraz potencjometru PR1 określa czas przerwy pomiędzy kolejnymi impulsami wyjściowymi. Dioda LED2 pełni funkcję sygnalizatora zadziałania przełącznika PK1. Dwa potencjometry PR1 i PR2

pozwalają niezależnie regulować czas włączenia (w przedziale ok. 0,5...15 sekund) i wyłączenia (w przedziale ok. 0,5...60 sekund).

Rozmieszczenie elementów na płycie drukowanej pokazuje rysunek 2. Montaż układu jest typowy i nie powinien przysporzyć problemów. **Zasilacz beztransformatorowy nie**

zapewnia separacji od sieci energetycznej, dlatego wszelkie prace nad układem należy wykonywać przy odłączonym zasilaniu.

Jako układ wykonawczy zastosowano przełącznik o obciążalności styków do 8 A/230 V AC. Przy sterowaniu obciążeniem o znacznej mocy należy zwrócić uwagę nie tylko na obciążenie styków przełącznika, ale także ścieżek płytki drukowanej. Aby poprawić ich obciążalność, można pocynować ścieżki lub ułożyć na nich i przylutować drut miedziany. Układ znajdzie wiele zastosowań, m.in. do uzyskania efektów świetlnych, w fotografii czy też jako prosta automatyka do różnych urządzeń.

EB

Regulowany bezpiecznik

Zwykły bezpiecznik topikowy jest mało praktyczny do zabezpieczenia przed nadmiernym poborem prądu, ponieważ wymaga wymiany po zadziałaniu. Prezentowany układ pełni taką samą funkcję – czyli odcina dopływ prądu po przekroczeniu zadanej wartości, ale umożliwia łatwe przywrócenie zasilania. Próg zadziałania można wybrać z zakresu 0,25 A...3 A.

Dodatkowe materiały do pobrania ze strony www.media.avt.pl

W ofercie AVT* AVT-5728

Podstawowe parametry:

- odłączanie zasilania po przekroczeniu ustalonego progu poboru prądu,
- możliwe do ustawienia wartości prądu maksymalnego: 0,25 A, 0,5 A, 0,75 A, 1 A, 1,5 A, 2 A, 2,5 A, 3 A,
- przywrócenie zasilania badanemu obwodowi po wciśnięciu przycisku,
- sygnalizacja stanu zasilania monitorowanego obwodu (załączone/wyłączone) na dwóch diodach LED,
- współpraca z obwodami prądu stałego, zasilanymi napięciem do 50 V,
- zasilanie stałym napięciem 12 V,
- pobór prądu ok. 30 mA.

* Uwaga! Elektroniczne zestawy do samodzielnego montażu wymagają umiejętności lutowania!

Podstawowa wersja zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wlutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu.

Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw
- wersja [B] – elementy wlutowane w płytkę PCB
- wersja [A] – płytkę drukowaną bez elementów i dokumentacji

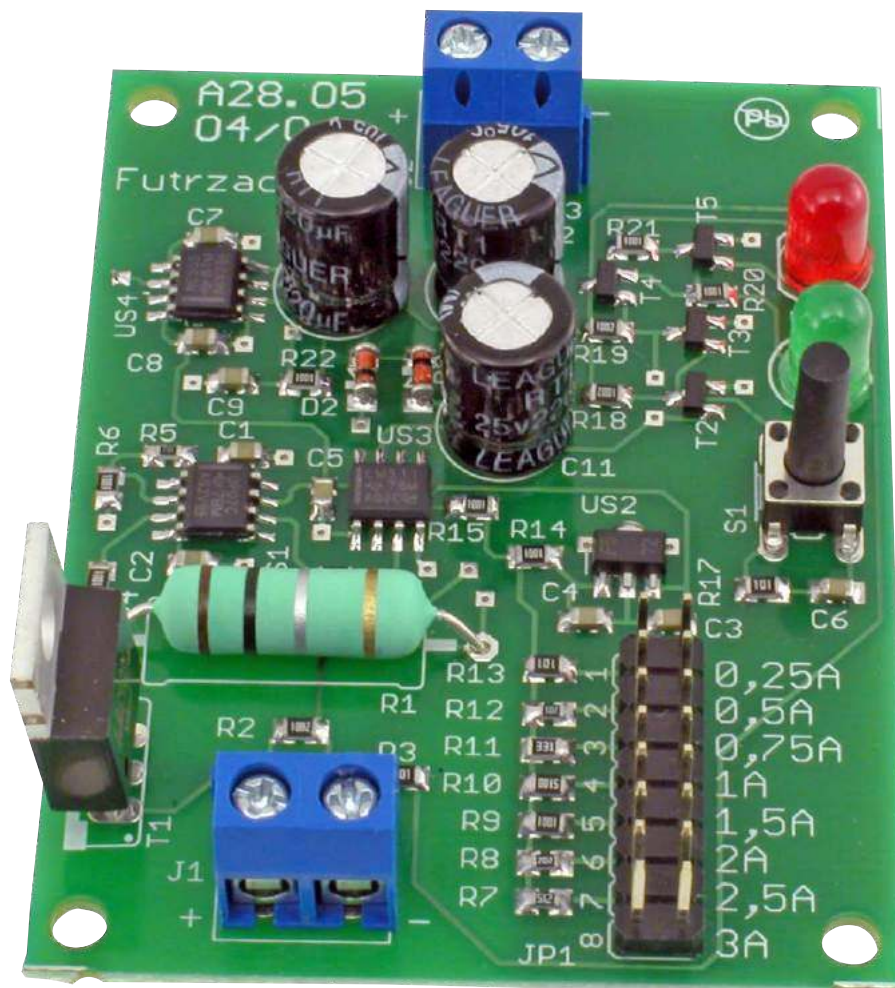
Kity w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:

- wersja [Av] – płytkę drukowaną [A] + zaprogramowany układ [UK] i dokumentacja
- wersja [UK] – zaprogramowany układ

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja na załączony ten sam plik pdf. Podczas składania zamówienia upewnij się, którą wersję zamawiasz!

<http://sklep.avt.pl>. W przypadku braku dostępności na <http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.

Przedstawiony układ działa jak wyłącznik nadmiarowoprądowy – po przekroczeniu zadanej wartości pobieranego prądu odcina dopływ zasilania i sygnalizuje to przełączeniem



diod LED. Przywrócenie zasilania jest bardzo proste, wystarczy pstryknąć przyciskiem. Układ doskonale nadaje się do monitorowania stanu pracy układów, które łatwo mogą zostać przeciążone, np. zawierających silniki prądu stałego lub do uchronienia testowanych prototypów przed bezpowrotnym zniszczeniem w razie awarii.

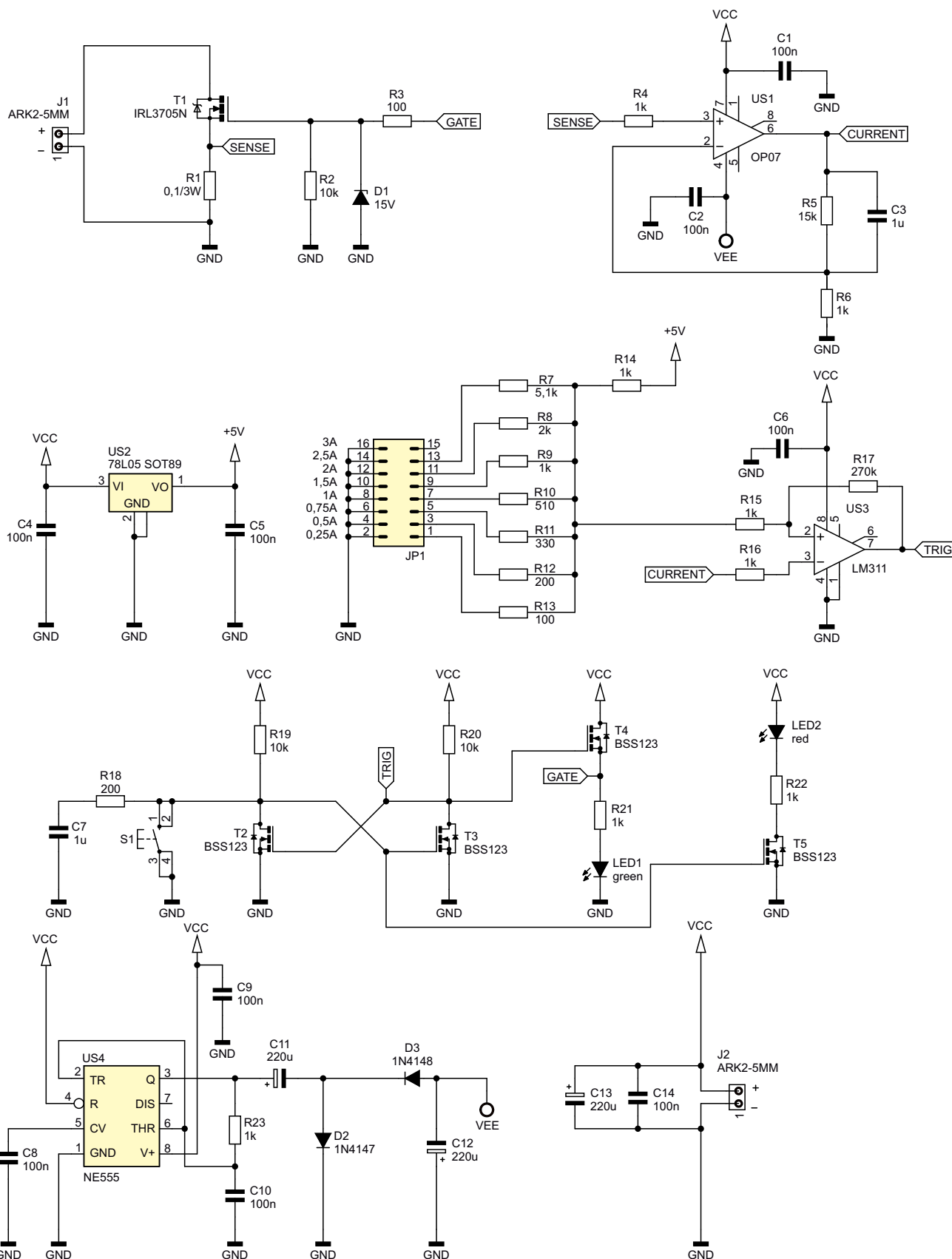
Budowa i działanie

Schemat ideowy układu pokazano na rysunku 1. Nie zawiera układów programowalnych, a działanie opiera się na kilku nieskomplikowanych i popularnych podzespołach, więc analiza jego działania będzie prosta nawet dla mniej zaawansowanych elektroników.

Przewód „masowy” nadzorowanego obciążenia należy przeciąć i przyłączyć do zacisków złącza J1. Prąd będzie wtedy płynął przez tranzystor T1 oraz rezystor R1, który służy do pomiaru natężenia prądu. Po zatankaniu tranzystora prąd przestanie płynąć.

Obwód bramki T1 jest zabezpieczony diodą Zenera D1, aby nie doszło do przypadkowego przebicia izolacji bramka-kanal tego tranzystora, dla którego maksymalna wartość napięcia bramka-źródło wynosi 16 V. Rezystor R2 rozładowuje pojemność bramka-źródło,

dlatego przypadkowo nagromadzone ładunki elektrostatyczne go nie otworzą. Rezystor R3 ogranicza prąd przepływający przez obwód sterujący bramką, gdyby takowy wystąpił – wskutek np. nieoczekiwanego wzrostu napięcia zasilającego powyżej 15 V.



Rysunek 1. Schemat ideowy układu bezpiecznika

Wybór tranzystora nie był przypadkowy. Rezystancja otwartego kanału wynosi zaledwie 10 mΩ. Oznacza to, że przy przepływie prądu o natężeniu 3 A (maksymalna możliwa do ustawienia wartość) na tranzystorze wydzielą się moc zaledwie 90 mW, co jest możliwe do rozproszenia bez jakiegokolwiek radiatora.

Wnikliwy czytelnik zauważy, że spadek napięcia na R1 zmniejsza efektywne napięcie bramka-źródło widziane przez T1 w stanie jego otwarcia. To prawda, lecz na R1 odłoży się nie więcej niż 0,3 V (3 A·0,1 Ω), co, wobec potencjału bramki wynoszącego ok. 10 V, jest wartością całkowicie pomijalną. Za to uprościł się pomiar prądu, który sprowadza się do zmierzenia spadku napięcia na rezystorze, podłączonego jednym wyprowadzeniem do masy. Napięcie z rezystora musi zostać wzmacnione, aby komparator mógł działać prawidłowo. Wzmacniacz powinien cechować się niską wartością napięcia niezrównoważenia, stąd wybór padł na popularny precyzyjny wzmacniacz operacyjny, jakim jest OP07. Niestety, w przeciwieństwie do układów typu LM358, nie potrafi on prawidłowo obsługiwać napięć bliskich potencjałom linii zasilających, stąd zaszła konieczność dodania prostego generatora napięcia ujemnego. Wzmocnienie tego członu wynosi 16 V/V. Rezystor R4 kompensuje wpływ prądów polaryzujących wejścia wzmacniacza, przez co nie powstaje dodatkowa, niepożądana składowa stała napięcia wyjściowego. Kondensator C3 ogranicza górną częstotliwość graniczną do, zaledwie, 10 Hz. To zaś przekłada się na maksymalny czas narastania rzędu 35 ms – krótkie impulsy o charakterze szpilkowym, powstające np. wskutek

Tabela 1. Wartości napięć podawanych na komparator przy określonych progach

Lp.	Ustawiony próg [A]	Rezystory tworzące dzielnik	Napięcie wychodzące z dzielnika [V]	Maksymalny spadek napięcia na R1 [mV]
1	0,25	R14+R13	0,45	28,1
2	0,5	R14+R12	0,83	51,9
3	0,75	R14+R11	1,24	77,5
4	1	R14+R10	1,69	106
5	1,5	R14+R9	2,50	156
6	2	R14+R8	3,33	208
7	2,5	R14+R7	4,18	261
8	3	R14 (brak podziału)	5,00	313

ładowania kondensatorów wewnątrz zasilanego układu, nie będą powodowały jego zadziałania.

Napięcie referencyjne, którym sterowany jest komparator, dostarcza stabilizator typu 78L05. Dokładność nie jest wprawdzie wysoka, ale wystarczająca na potrzeby tak prostego układu. Zaletą jego zastosowania jest niska cena oraz wysoka wydajność prądowa. Powstałe w ten sposób napięcie o wartości 5 V jest dzielone z użyciem dzielnika napięcia, który tworzy rezystor R14 i jeden z rezystorów R7...R13. Tabela 1 zawiera zestawienie uzyskiwanych w ten sposób napięć, podawanych na wejście nieodwracające oraz odpowiadających im wartości prądów wyzwajających. Na wejście odwracające wchodzi sygnał ze wzmacniacza, co oznacza, że przekroczenie dopuszczalnej wartości prądu spowoduje szybkie nasycenie jego tranzystora wyjściowego, czyli spadek potencjału wyjścia. Aby zminimalizować ryzyko powstawiania oscylacji na progu przełączenia, wprowadzono niewielką histerezę, którą realizuje rezystor R17.

Układ musi mieć jednobitową „pamięć”, która pozwoli utrzymać tranzystor wykonawczy w stanie pełnego otwarcia lub zatkania bez ograniczeń czasowych. Posłużył do tego nieco już dzisiaj zapomniany

przerzutnik Eccles-Jordana, wykonany na tranzystorach... MOSFET. Pozwoliło to uprościć układ poprzez pominięcie rezystorów ograniczających prądy baz. Użytkownik, poprzez zwarcie styków S1, może załączyć tranzystor T2, jednocześnie wyłączając T3. To samo robi również kondensator C7, który stanowi zwarcie przez chwilę od włączenia zasilania układu. Daje to gwarancję, że układ zawsze uruchomi się w tym samym stanie, tj. z załączonym obciążeniem. Rezystor R18 ogranicza prąd przeładowywania tego kondensatora.

Kiedy potencjał drenu T3 jest wysoki, niemal równy napięciu zasilającemu, wtedy tranzystor wyjściowy T1 przewodzi. Jego załączenie odbywa się za pośrednictwem wtórnika napięciowego, który realizuje tranzystor T4. Przyspiesza to proces załączania, a jednocześnie dren tranzystora T3 nie jest obciążany dodatkowym prądem. Potencjał bramki T1 będzie niższy od napięcia zasilającego o ok. 2 V, lecz to wystarczy, aby go w pełni otworzyć. Jednocześnie zostaje załączona dioda LED1 i wyłączona LED2.

Wykaz elementów:

Rezystory:

R1: 0,1 Ω 3 W
R2, R19, R20: 10 kΩ SMD0805
R3, R13: 100 Ω SMD0805
R4, R6, R9, R14...R16, R21...R23: 1 kΩ SMD0805
R5: 15 kΩ SMD0805
R7: 5,1 kΩ SMD0805
R8: 2 kΩ SMD0805
R10: 510 Ω SMD0805
R11: 330 Ω SMD0805
R12, R18: 200 Ω SMD0805
R17: 270 kΩ SMD0805

Kondensatory:

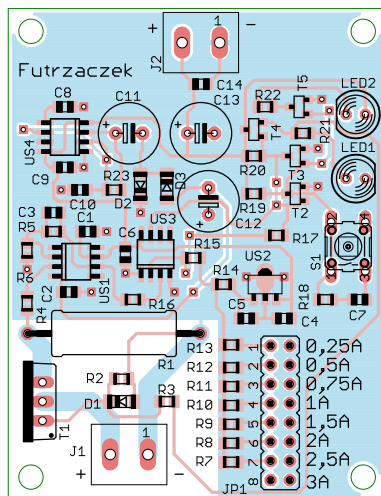
C1, C2, C4...C6, C8...C10, C14: 100 nF SMD0805
C3, C7: 1 μF SMD0805
C11...C13: 220 μF/25 V THT

Półprzewodniki:

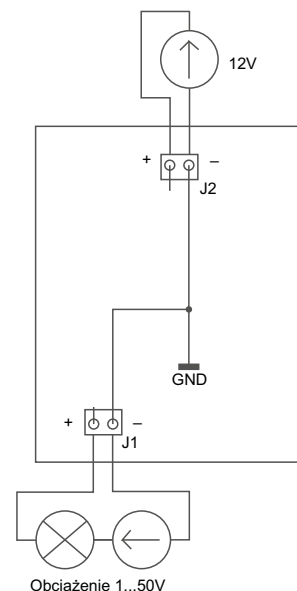
D1: Zener 15 V 0,4 W MiniMELF
D2, D3: 1N4148 MiniMELF
LED1: zielona 5 mm
LED2: czerwona 5 mm
T1: IRL3705N T0220
T2...T5: BSS123 SOT23
US1: OP07 S08
US2: 78L05 SOT89
US3: LM311 S08
US4: NE555 S08

Inne:

J1, J2: ARK2 5 mm
JP1: goldpin 2×8 pin męski 2,54 mm THT + zworka
S1: microswitch 6×6 mm, wysokość 13 mm



Rysunek 2. Schemat płytki bezpiecznika wraz z rozmieszczeniem elementów



Rysunek 3. Schemat podłączenia układu

Zmiana stanu tego przerzutnika na przeciwny jest możliwa wyłącznie przez nasycenie tranzystora wyjściowego w komparatorze US3, ponieważ zostaje wtedy zamknięty tranzystor T2 i otwarty T3. Potencjał drenu T2 wzrasta, ładując jednocześnie kondensator C7, co otwiera jednocześnie tranzystor T3. Wtórnik napięciowy ulega zatkanu: zatyka się również tranzystor T1, gaśnie dioda LED1 i załącza się LED2. Układ sygnalizuje, że wystąpił nadmierny pobór prądu.

Do wytwarzania napięcia ujemnego służy bardzo prosty obwód z układem 555 w roli głównej. Jego napięcie wyjściowe wynosi ok. -10 V i nie jest stabilizowane, ponieważ nie ma takiej potrzeby – parametr PSRR zasilanego układu (US1) wynosi jedynie 5 $\mu\text{V/V}$ i wpływ ewentualnych wahań tego napięcia będzie pomijalny.

Montaż i uruchomienie

Układ został zmontowany na dwustronnej płytce drukowanej o wymiarach 50×65 mm, której schemat montażowy oraz wzór ścieżek pokazuje rysunek 2. Otwory montażowe znajdują się w odległości 3 mm od krawędzi płytki.

Większość elementów jest montowana powierzchniowo, a wszystkie one znajdują się na wierzchniej stronie płytki (Top). Po tej samej stronie znajdują się również podzespoły przewlekane (THT). Montując

Tabela 2. Zmierzone wartości prądu powodującego reakcję układu

Żądane natężenie prądu maksymalnego [A]	Zmierzone natężenie prądu maksymalnego [A]
0,25	0,29
0,5	0,54
0,75	0,79
1	1,09
1,5	1,62
2	2,15
2,5	2,69
3	3,20

rezystor R1, warto przylutować go na nieco dłuższych wyprowadzeniach, aby poprawić jego chłodzenie. Tranzystor T1 został umieszczony przy krawędzi płytki, lecz nie powinien wymagać dodatkowego chłodzenia. Gdyby został zastosowany tranzystor innego typu, o większej rezystancji otwartego kanału, wówczas może zajść potrzeba jego chłodzenia.

Prawidłowo zmontowany układ nie wymaga jakichkolwiek czynności uruchomieniowych. Jest gotowy do działania po podłączeniu do źródła zasilania o napięciu ok. 12 V, z którego pobiera prąd o natężeniu ok. 30 mA. Należy również wybrać żadaną wartość prądu maksymalnego poprzez nałożenie zworki na odpowiednie wyprowadzenia JP1.

Eksplotacja

Układ regulowanego bezpiecznika należy włączyć do monitorowanego obwodu zgodnie z rysunkiem 3, pamiętając jednocześnie o tym, że masy obu zasilaczy są połączone na płytce. Maksymalne napięcie, jakim może być zasilany nadzorowany układ, wynosi 50 V i jest ograniczone wytrzymałością napięciową zastosowanego tranzystora T1.

Po włączeniu zasilania dla układu (podawanego na zaciski złącza J2), dioda LED1 załączy się, a LED2 pozostanie wygaszona. To oznacza, że między zaciskami złącza J1 może przepływać prąd, ponieważ tranzystor T1 przewodzi. Po przekroczeniu prądu maksymalnego dioda przełączając się, a zasilanie zostaje odcięte. Przywrócenie przewodzenia przez T1 jest możliwe po krótkotrwałym wciśnięciu przycisku S1.

Tolerancja użytych elementów, rozrzuty pozostałych parametrów oraz histereza komparatora powodują powstanie pewnej odchyłki między wartością prądu podaną na płytce a tą, przy której układ faktycznie zareaguje. W tabeli 2 zebrano wyniki tych pomiarów. Spadek napięcia na zaciskach złącza J1 uzależniony jest od natężenia płynącego przez nie prądu. Jego wartość maksymalna wynosi ok. 0,35 V (dla prądu 3 A).

Michał Kurzela, EP

REKLAMA



UWAGA KONKURS

Elektroniku mały i duży

KITYAVT ogłasza konkurs wiedzy o zestawach AVT.

Do zdobycia atrakcyjne nagrody i gadzety. Szczegóły i regulamin na profilu Facebook KITYAVT



nie przegap śledź profil już od dziś...

SKLEP.AVT.PL

Nie przegap! - interesujących materiałów w siostrzanym czasopiśmie



www.elportal.pl

A może masz pomysł na ciekawy artykuł lub projekt? Skonstruowałeś urządzenie, które jest godne zaprezentowania szerszej publiczności? Możesz napisać artykuł edukacyjny? Chcesz podzielić się doświadczeniem? W takim razie zapraszamy do współpracy na łamach Elektroniki dla Wszystkich. Kontakt: edw@elportal.pl

EdW możesz zamówić na www.ulubionykiosk.pl
Do kupienia również w Empikach
i wszystkich większych kioskach z prasą.

W listopadowym wydaniu Elektroniki dla Wszystkich między innymi:

- ESP8266 RC** – zdalne sterowanie modelem przez sieć Wi-Fi
Opis pojazdu zdalnie sterowanego drogą radiową za pośrednictwem smartfona i zainstalowanego na nim prostego programu to kolejny przykład wykorzystania bardzo popularnych od lat modułów ESP8266.
- Arduino Nano i oświetlenie choinkowe**
Fascynujący przykład wykorzystania Arduino do zaświecania różnokolorowych diod LED RGB w długim łańcuchu układów scalonych WS2811.
- MPPT. Maximum Power Point Tracking**, czyli śledzenie punktu mocy maksymalnej wytwarzanej przez panel słoneczny, to kluczowe zagadnienie, które dziś musi rozumieć każdy elektronik.
- Kurs Arduino – matrycowe wyświetlacze LED**
W kolejnym odcinku cyklu z powodzeniem wykorzystujemy punktowe, matrycowe wyświetlacze LED, a to dzięki zastosowaniu interesującej, bogatej biblioteki MD_MAX72xx.
- Kuchnia Konstruktora. Taki zwyczajny zasilacz...**
W tym odcinku zajmujemy się walką z samowzbudzeniem oraz przystosowujemy nasz eksperymentalny model do wykorzystania wzmacniaczy operacyjnych w wersji SMD.
- Ponadto w numerze:**
 - Elektroniczna „wielokostka”
 - Infinity – system automatyki domowej.
 - Pomiar wielkości fizycznych – pomiar temperatury
 - Miniaturowy termostat do rezonatora/generatora kwarcowego
 - Monostabilny przełącznik ON/OFF
 - Eksperymentalny ochronnik PV klasy B+C
 - Zdalne sterowanie ogrzewaniem budynku, część 1
 - Przetwornice indukcyjne, część 32
 - Szkoła Konstruktorów – wykorzystanie dowolnego gotowego, fabrycznego modułu.
 - Szkoła Konstruktorów – Zaproponuj sposób zasilania lub układ zasilany z pojedynczego akumulatora 1,2V
 - Baterijne odbiorniki lampowe, część 2

Dział „Projekty Czytelników” zawiera opisy projektów nadesłanych do redakcji EP przez Czytelników. Redakcja nie bierze odpowiedzialności za prawidłowe działanie opisywanych układów, gdyż nie testujemy ich laboratoryjnie, chociaż sprawdzamy poprawność konstrukcji. Prosimy o nadsyłanie własnych projektów z modelami (do zwrotu). Do artykułu należy dołączyć podpisane oświadczenie, że artykuł jest własnym opracowaniem autora i nie był dotychczas nigdzie publikowany. Honorarium za publikację w tym dziale wynosi 250,- zł (brutto) za 1 stronę w EP. Przesyłanych tekstów nie zwracamy. Redakcja zastrzega sobie prawo do dokonywania skrótów.

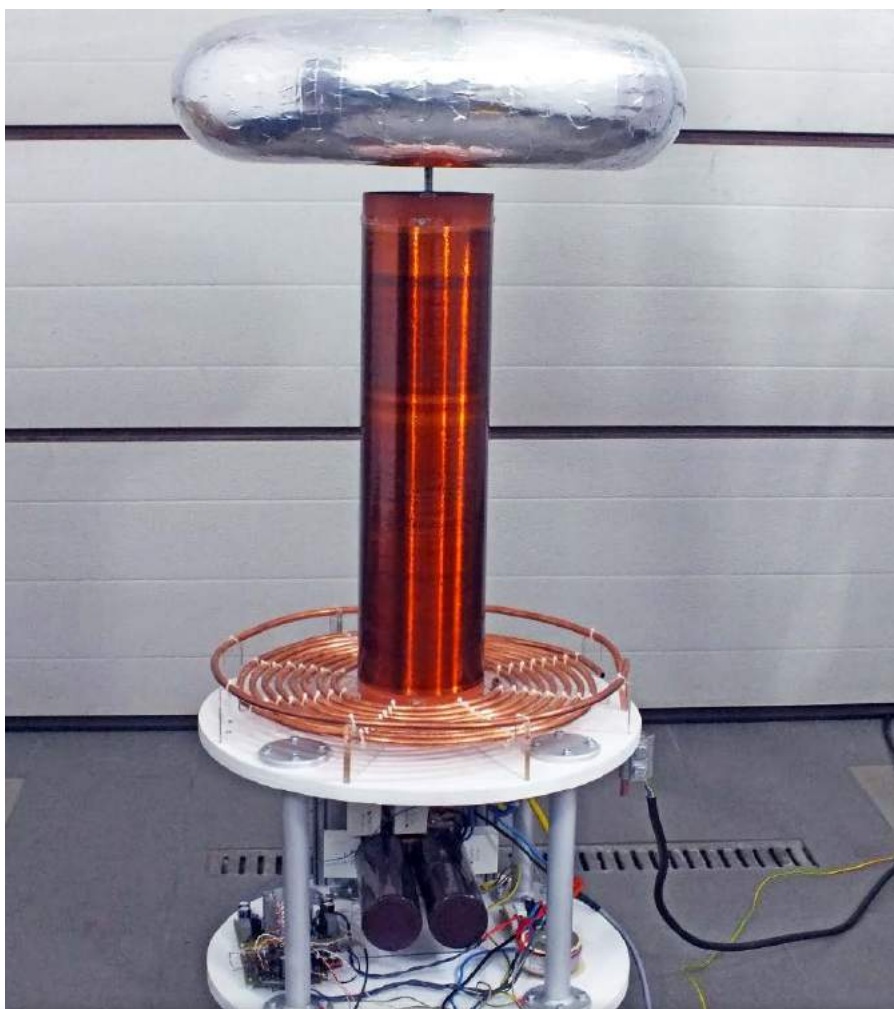
**PROJEKT
NR 247**

Półprzewodnikowa cewka Tesli o podwójnym rezonansie

Techniką interesowałem się od dzieciństwa, jednak najwyżej stawiałem elektronikę. Inspiracją do zbudowania własnej cewki Tesli był pokaz w Teatrze Wysokich Napięć odbywający się w Centrum Nauki Kopernik. Samodzielnie zaprojektowałem i wykonałem własną wersję cewki. Było to dla mnie duże wyzwanie, poświęciłem temu olbrzymią ilość czasu i energii, jednak podczas realizacji projektu nauczyłem się wielu nowych rzeczy. Konstrukcja różni się od większości cewek Tesli możliwością pełnej kontroli nad wyładowaniem dzięki zastosowaniu szeregu rozwiązań elektronicznych.

Konstrukтором transformatora Tesli jest pochodzący z Serbii Nikola Tesla. Wynalazek miał być używany do bezprzewodowego przesyłania energii. Jest to transformator o rdzeniu powietrznym, którego działanie opiera się na zjawisku rezonansu i generuje napięcie rzędu milionów woltów. Każde wyładowanie elektryczne powstające w powietrzu jest związane z powstaniem kanału zjonizowanego gazu. Znaczny wzrost temperatury i ciśnienia w kanale wyładowania są źródłem fali akustycznej. Objawia się to również efektownym zjawiskiem świetlnym.

Klasyczny transformator składa się z: zasilacza wysokiego napięcia, iskiernika, kondensatora, uzwojenia pierwotnego, uzwojenia wtórnego i toroidu. W półprzewodnikowej cewce Tesli o podwójnym rezonansie (DRSSTC *Dual Resonant Solid State Tesla Coil*) iskiernik został zastąpiony obwodem elektronicznym,



Podziękowania

Na szczególne podziękowanie zasługują: dr Paweł Sobczak (opiekun naukowy), Damian Pała (absolwent technologii chemicznej na Akademii Górniczo-Hutniczej im. Stanisława Staszica w Krakowie), Dominik Idziniak (student elektrotechniki na Politechnice Rzeszowskiej). Jestem wdzięczny tym osobom za cały poświęcony czas oraz trud włożony w poszukiwanie odpowiedzi na moje pytania. Panu mgr. inż. Andrzejowi Gajewskiemu dziękuję za pomoc merytoryczną w przygotowaniu do konkursu „Elektronik”.

Jakub Jędrzejewski



Fotografia 1. Uruchomiony transformator Tesli, długość wyładowań osiąga 50 cm

dzięki czemu do pracy nie jest wymagane wysokie napięcie. Właśnie to jest główną cechą, która odróżnia wykonaną przeze mnie cewkę od innych. Dzięki zastosowaniu falownika oraz specjalnego sterownika możliwa jest regulacja i kontrola parametrów pracy takich jak wartości prądu i napięcia w obwodzie, ilość pobieranej mocy czy częstotliwość i wypełnienie sygnału. Zmiana częstotliwości i czasu trwania wyładowań powoduje zmianę wysokości dźwięku generowanego przez wyładowanie, dzięki czemu możliwe jest odtwarzanie pojedynczych dźwięków o różnym tonie i głośności oraz złożonych z nich melodii. W niedalekiej przyszłości planuję przystosować układ do odtwarzania muzyki na cewce Tesli.

Półprzewodnikowa cewka Tesli o podwójnym rezonansie jest kolejnym wariantem transformatora. Jak sama nazwa wskazuje, mamy do czynienia z dwoma obwodami rezonansowymi. Pierwszym jest obwód LC sterowany modułami IGBT. Drugim obwodem jest klasyczne uzwojenie wtórne i torus. Warunek pracy DRS-STC jest taki, że oba obwody muszą pracować z taką samą lub zbliżoną częstotliwością rezonansową. Jeżeli ten warunek jest spełniony, następuje wydajny przekaz energii pomiędzy

obwodami. Mimo że prąd pobierany przez urządzenie z sieci jest niewielki, w obwodzie pierwotnym może sięgać nawet kilkuset amperów. Ograniczeniem jest wytrzymałość tranzystorów i kondensatorów. Prezentowana cewka **wytwarza napięcie dochodzące do 2 milionów woltów**, efekt działania pokazuje **fotografia 1**.

Budowa i działanie urządzenia

Schemat blokowy urządzenia prezentuje **rysunek 2**. Budowę można podzielić na 8 bloków.

1. Zasilanie

Sterownik wymaga zasilania napięciami stałymi 5 V, 15 V i -15 V. Do mostka pośredniego jest dostarczane napięcie 40 V. Wszystkie potrzebne napięcia uzyskałem dzięki zastosowaniu transformatora toroidalnego (**fotografia 3**). Obwód pierwotny jest zasilany wyprostowanym napięciem sieciowym. W fazie testów używałem autotransformatora, który umożliwiał stopniowe zwiększanie napięcia zasilania falownika. Pobór mocy przy wyładowaniu o długości ok. 50 cm nie przekraczał 1000 W.

2. Interrupter

Układem, który wyzwała pracę DRSSTC, jest interrupter. Ma regulację częstotliwości



Fotografia 3. Transformator zasilający układy sterujące

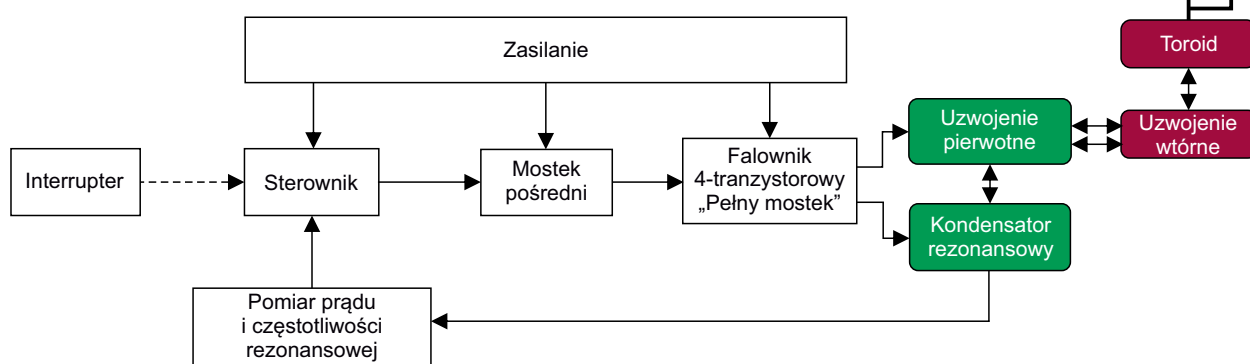


Fotografia 4. Interrupter – moduł kontrolujący parametry pracy cewki

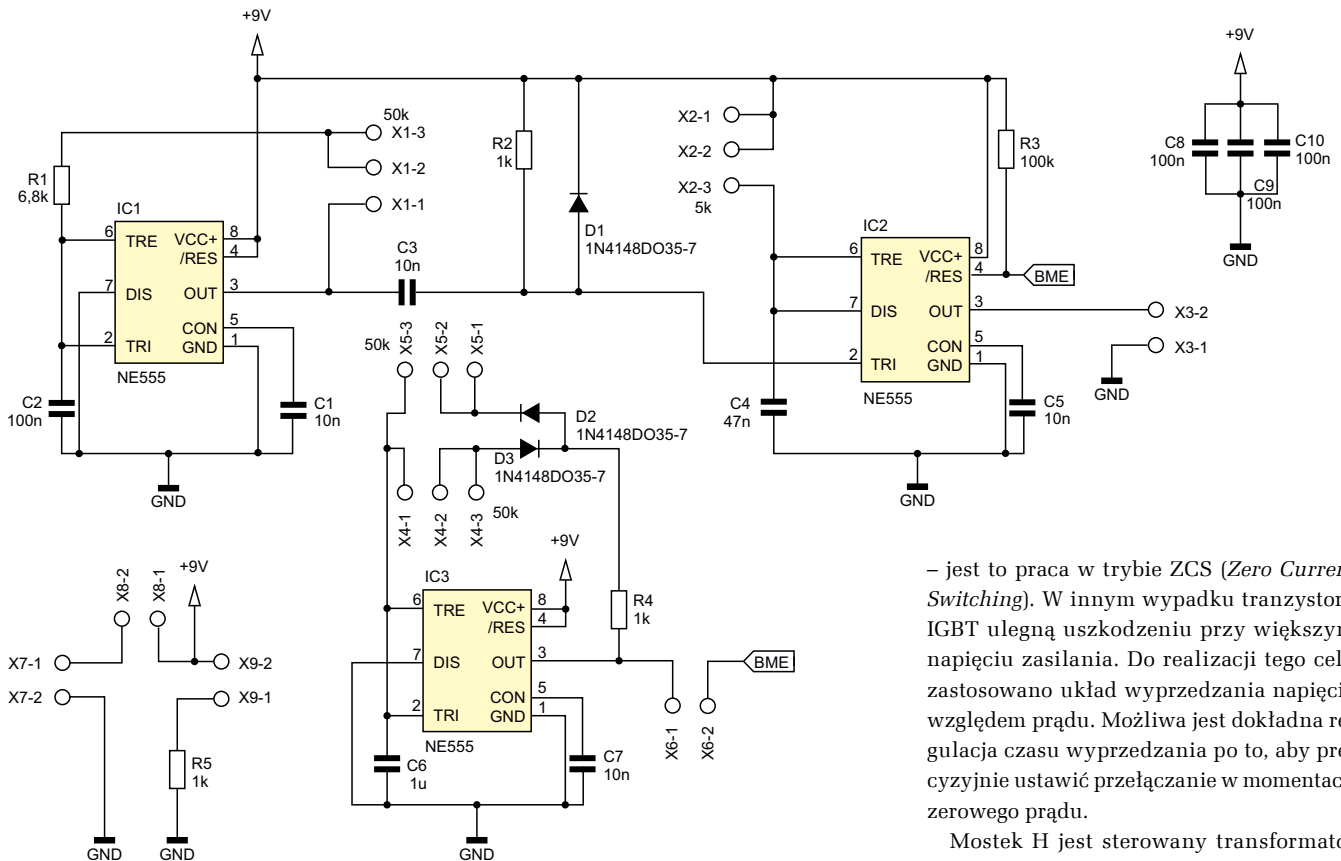
i wypełnienia sygnałów, którymi sterowana jest cewka. Jest zsynchronizowany z obwodem pierwotnym za pomocą specjalnego przerywnika. Interrupter to oddzielny element urządzenia (**fotografia 4**), pozwala sterować pracą cewki. Połączenie interruptera ze sterownikiem wykonałem przy użyciu kabla koncentrycznego, jednak zamierzam zastąpić go łączem światłowodowym. W planach mam także zbudowanie muzycznego interruptera, który będzie umożliwiał sterowanie transformatorem Tesli za pomocą muzyki. Jego podstawowym zadaniem będzie odczyt plików MIDI i generowanie odpowiedniego sygnału sterującego. Schemat interruptera pokazano na **rysunku 5**.

3. Sterownik

Pomysłodawcą i pierwszym autorem tego układu jest Finn Hammer. Zasada działania



Rysunek 2. Schemat blokowy urządzenia



Rysunek 5. Schemat elektryczny interruptera

sterownika polega na wykorzystaniu sygnału sprzężenia zwrotnego. Przekładnik prądowy o przekładni 1:100 nałożony na jeden z przewodów obwodu pierwotnego generuje sygnał, który następnie jest formowany do postaci impulsów i służy do sterowania modułami IGBT. Sterownik realizuje również funkcję ograniczenia prądowego z użyciem takiego samego przekładnika jak w przypadku sprzężenia zwrotnego. Tym razem sygnał jest

prostowany pełnookresowo i doprowadzony jest na wejście komparatora. Na drugie wejście komparatora doprowadzone jest napięcie odniesienia. Układ ten służy do blokowania pracy cewki w przypadku przekroczenia zadanej wartości prądu. Schemat sterownika pokazano na **rysunku 6**, a zmontowaną płytkę sterownika prezentuje **fotografia 7**.

Tranzystory IGBT muszą być przełączane przy zerowych wartościach prądu

– jest to praca w trybie ZCS (Zero Current Switching). W innym wypadku tranzystory IGBT ulegną uszkodzeniu przy większym napięciu zasilania. Do realizacji tego celu zastosowano układ wyprzedzania napięcia względem prądu. Możliwa jest dokładna regulacja czasu wyprzedzania po to, aby precyzyjnie ustawić przełączanie w momentach zerowego prądu.

Mostek H jest sterowany transformatorem sterującym GDT, a następnie mostkiem pośrednim, dzięki czemu jest zapewniona separacja galwaniczna sterownika od obwodu mocy. Transformator sterujący GDT oraz przekładniki prądowe zostały nawinięte na rdzeniach toroidalnych z materiału 3E25.

4. Mostek pośredni

Sygnał generowany przez sterownik i transformator sterujący GDT trafia do mostka pośredniego. Jest to typowy układ pełnego mostka z zastosowanymi tranzystorami MOSFET do maksymalnego skrócenia czasów narastania i opadania sygnału.

Wykaz elementów:

Interrupter

Rezystory:

R1: 6,8 kΩ
R2, R4, R5: 1 kΩ
R3: 100 kΩ

Kondensatory:

C1, C3, C5, C7: 10 nF
C2, C8, C9, C10: 100 nF
C4: 47 nF
C6: 1 μF

Półprzewodniki:

D1, D2, D3: 1N4148
IC1, IC2, IC3: NE555

Inne:

X1...X9: złącza typu ARK

Sterownik

Rezystory:

R1: potencjometr wieloobrotowy 100 Ω
R2, R3: 3,3 kΩ
R4, R10, R11, R14, R16, R20, R21: 1 kΩ
R5, R8, R12, R13: pot. wieloobrotowy 10 kΩ
R15: 220 kΩ
R17: 22 kΩ
R18: 10 kΩ
R9, R19: 220 Ω

Kondensatory:

C1 1 nF
C2: 1 nF
C3: 6,8 nF
C4: 4,7 nF
C5: 1 nF

C6, C7: 330 μF
C8...C10: 10 μF
C11...C21, C32...C34: 100 nF
C101: 1000 μF

Półprzewodniki:

D1, D2, D100...D103, D108: 1N4148
D104...D107, D109: 1N5819
D110: 1N5849
LED1, LED2: dioda LED
B1: mostek prostowniczy
T1: BC547
IC1, IC8: AD790
IC2: 7815
IC3: 7805
IC4: 7915
IC5, IC9: 74AC14N
IC6: 74AC109N
IC7: LM171
GD1, GD2: UCC37322/1

Inne:

X1...X6: złącza typu ARK

Mostek

Rezystory:

R1: 1 Ω
R2: 10 kΩ
R3: 220 Ω
R4, R9: 24 kΩ
R5, R6, R7, R8: dobrać przy użyciu oscyloskopu, wartość początkowa ok. 20 Ω

Kondensatory:

C1, C2: 470 μF

C3, C5, C9: 100 nF
C4, C6, C10, C11: 1000 μF
C7, C8: 1 μF

Półprzewodniki:

D1, D8, D13, D18: 1N5819
D2, D9, D14, D19: SB560
D3, D4, D6, D7, D11, D12, D16, D17: BZX85
D5, D10, D15, D20: MUR460
B1: mostek prostowniczy
Q1...Q4: IRFP460
T1: BD650
U1: LM317

Inne:

X1...X2: złącza typu ARK

Falownik

Rezystory:

R1...R4: dobrać przy użyciu oscyloskopu, wartość początkowa ok. 20 Ω

Kondensatory:

C1, C2: PMB 6800 μF 450 V
C3, C4, C6: 0,56 μF 4 kV
C5: 4x PMB 0,56 μF 4 kV

Półprzewodniki:

D1...D8: 1.5KE110
D9...D16: Zener 18 V
B1: mostek prostowniczy
Q1...Q4: SKM200GB128D

Inne:

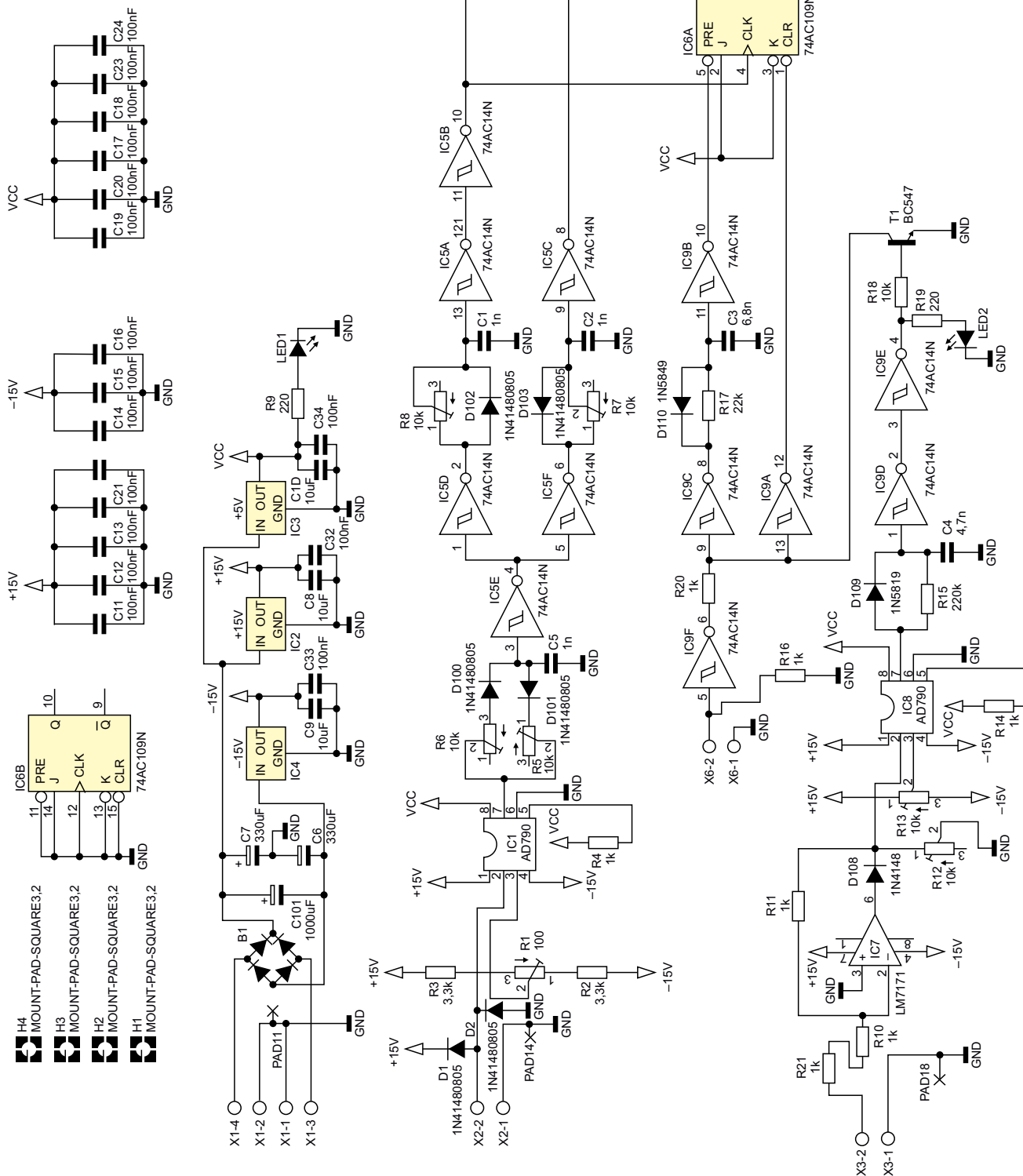
GDT: transformator separujący
X1...X2: złącza typu ARK

Zadaniem tej części jest wzmocnienie sygnału sterującego głównym falownikiem, a co za tym idzie odciążenie sterownika. Schemat mostka pośredniego pokazano na rysunku 8, a zmontowaną płytkę prezentuje fotografia 9.

5. Falownik

Schemat falownika pokazano na fotografii 10, jest to mostek H zbudowany z zastosowaniem tranzystorów IGBT SKM200GB128D, których maksymalne napięcie pracy wynosi 1200 V, a maksymalny prąd ciągle 300 A. W skład mostka wchodzi także kondensatory filtrujące – razem

13,6 mF 450 V. Podczas przełączania tranzystorów powstają spore przepięcia. Jest to skutek przełączania dużych prądów. Aby je zlikwidować, zastosowałem odpowiednie sterowanie oraz kondensatory typu snubber 2 μ F, które znacząco zmniejszają impulsy napięcia, oraz diody zabezpieczające TVS, których zadaniem jest zwieranie ujemnych impulsów napięcia. Na fotografii 11 pokazano oscylogram przebiegu prądu



Rysunek 6. Schemat elektryczny sterownika



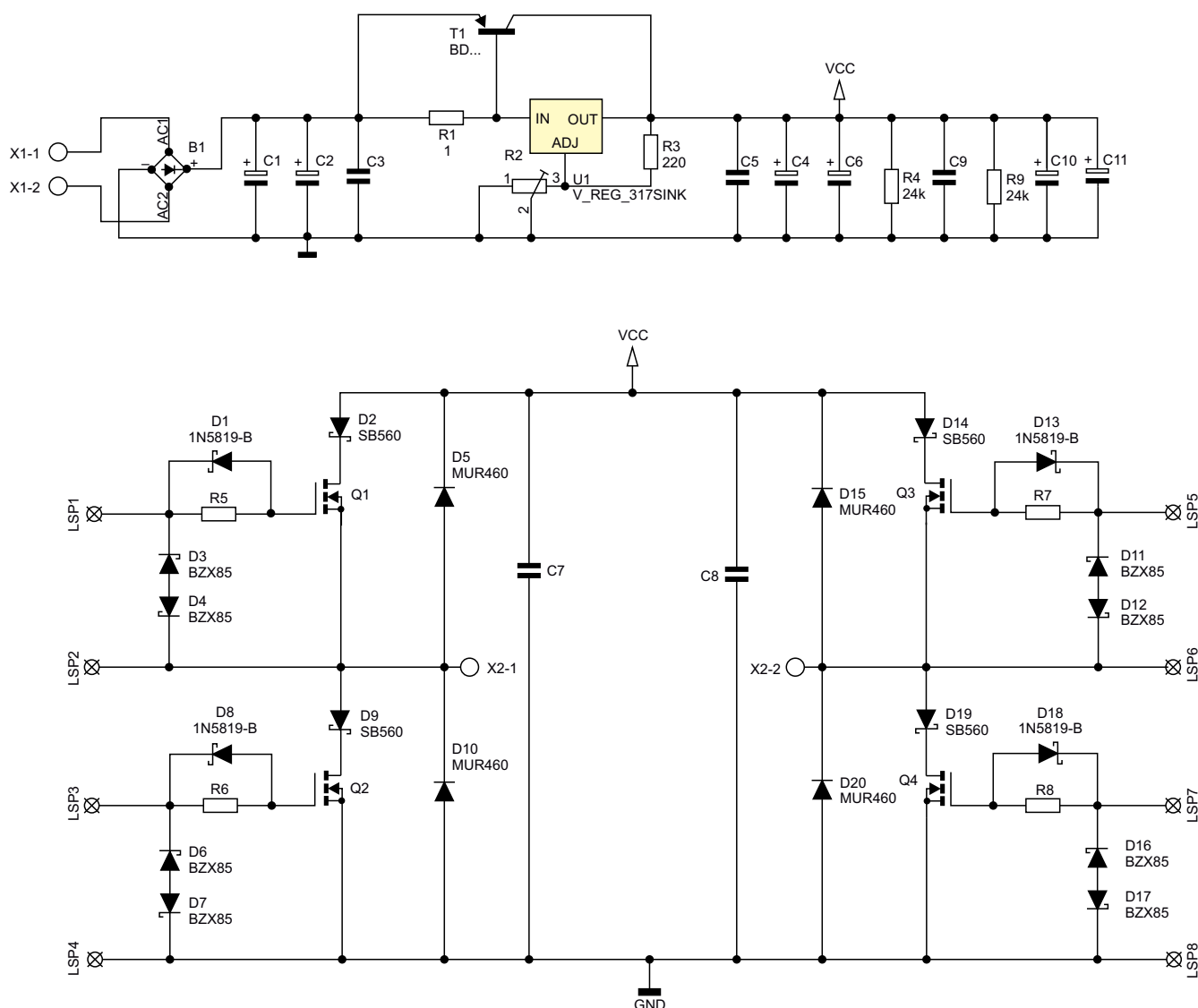
Fotografia 7. Wygląd zmontowanej płytki sterownika

w obwodzie pierwotnym (maksymalna wartość dochodzi do 300 A), natomiast na **fotografii 12** pokazano oscylogram przebiegu napięcia w tym obwodzie.

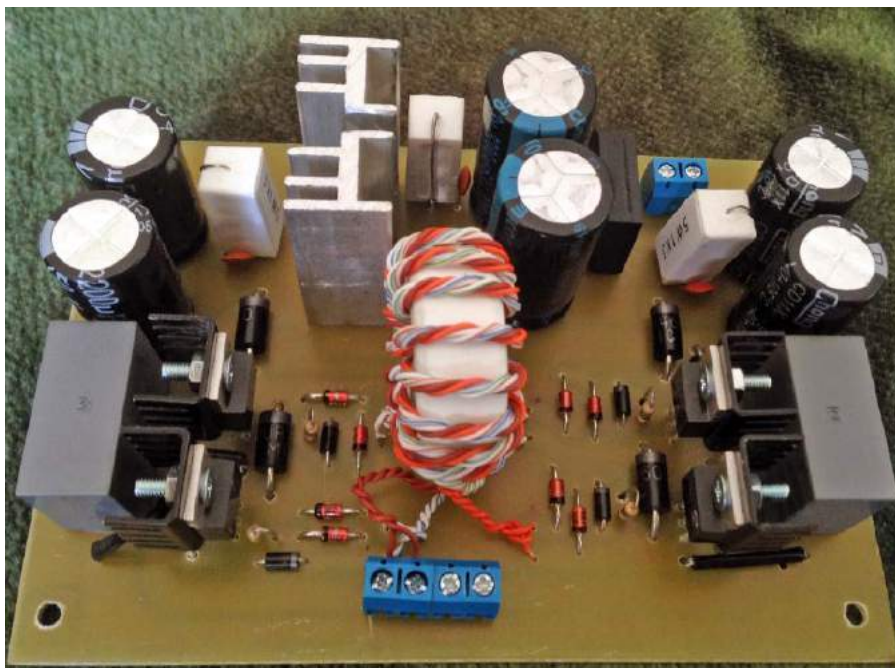
Połączenia między tranzystorami a kondensatorami filtrującymi powinny się charakteryzować jak najmniejszą indukcyjnością, ponieważ zwiększa ona energię impulsów napięcia. Aby temu zapobiec, do wykonania połączeń użyto kształtek z grubej blachy miedzianej. Ideą pracy falownika jest przełączanie tranzystorów dokładnie przy zerowych wartościach prądu (ZCS). Cały falownik został zmontowany na dużym radiatorze, jednak dzięki pracy w trybie ZCS pozostaje on zimny nawet po dłuższej pracy cewki. Zmontowany falownik prezentują **fotografie 13 i 14**.

6. Pierwotny obwód rezonansowy

Jako kondensator rezonansowy zastosowałem baterię 4 kondensatorów PMB 0,56 μF 4 kV. Zostały wybrane pod kątem wysokiej wytrzymałości prądowej i napięciowej. Uzwojenie pierwotne jest zbudowane z rurki miedzianej o średnicy 10 mm, nawiniętej



Rysunek 8. Schemat elektryczny mostka pośredniego



Fotografia 9. Wygląd zmontowanej płytki mostka pośredniego

na specjalnie wyciętych z pleksi podporach. Łączna długość rurki to około 10 m. Konstrukcję pierwotnego obwodu rezonansowego prezentują **fotografia 15 i 16**.

7. Wtórny obwód rezonansowy

Uzwojenie wtórne jest nawinięte na rurze kanalizacyjnej 160 mm drutem 0,3 mm na długości 56 cm, co daje około 1800 zwojów (**fotografia 17**). Do prawidłowego wykonania tej części niezbędna jest cierpliwość i duże zdolności manualne. Po nawinięciu pomalowałem je żywicą epoksydową, co uniemożliwia rozsypanie się drutu. Torus został zbudowany poprzez nawinięcie rury spiro na wytoczone dekle drewniane, wyrównanie powierzchni za pomocą gładzi szpachlowej oraz oklejenie całej powierzchni

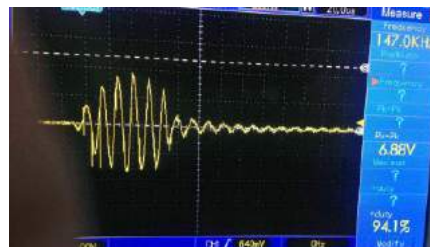
taśmą aluminiową. Częstotliwość rezonansowa uzwojenia wtórnego i torusa wynosi około 70 kHz.

8. Podstawa

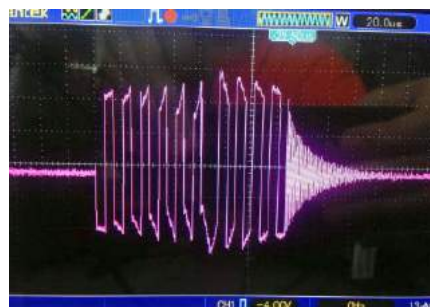
Baza całego urządzenia została wykonana z płyt spienionego PCV. Do ich połączenia użyłem nóżek aluminiowych (**fotografia 18**). Aby umożliwić mobilność całej konstrukcji, zostały zamontowane 4 kółka jezdne. Podstawa musiała być wykonana bardzo starannie ze względu na to, że przenosi spory ciężar wszystkich pozostałych elementów.

Podsumowanie

W trakcie budowy urządzenia napotkałem kilka problemów. Jednym z nich był zaskakująco duży wpływ ustawienia



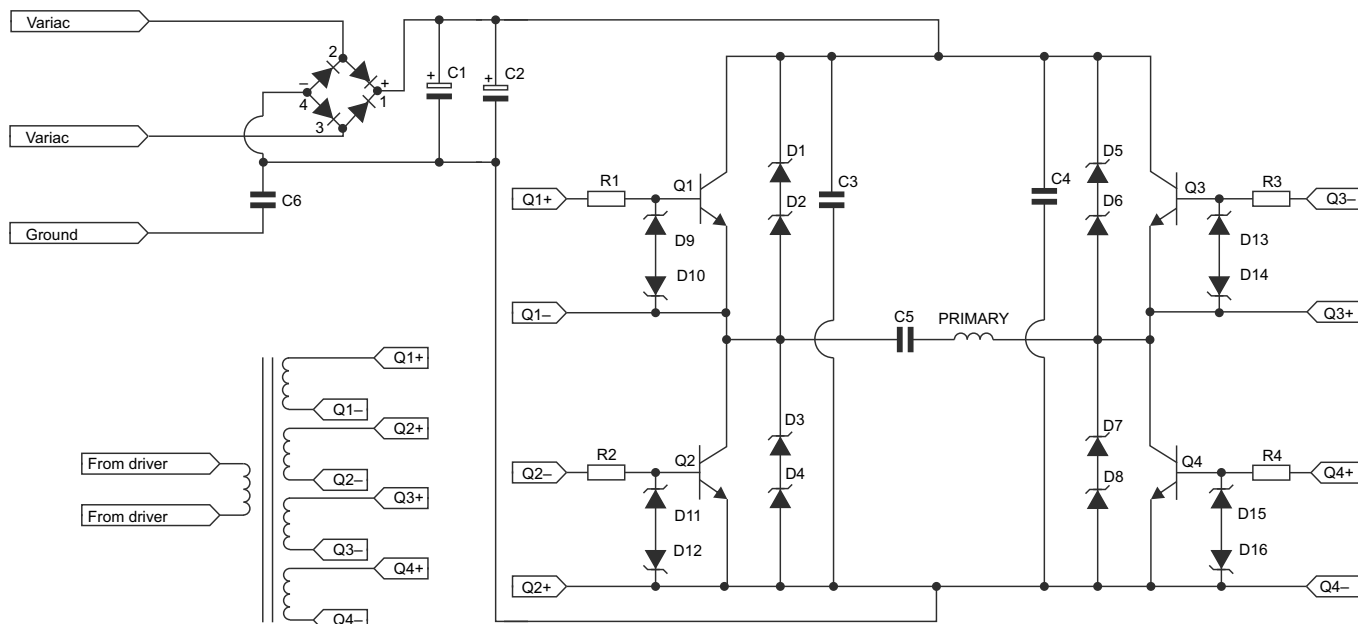
Fotografia 11. Oscylogram przebiegu prądu w obwodzie pierwotnym, prąd maksymalny wynosi około 300 A



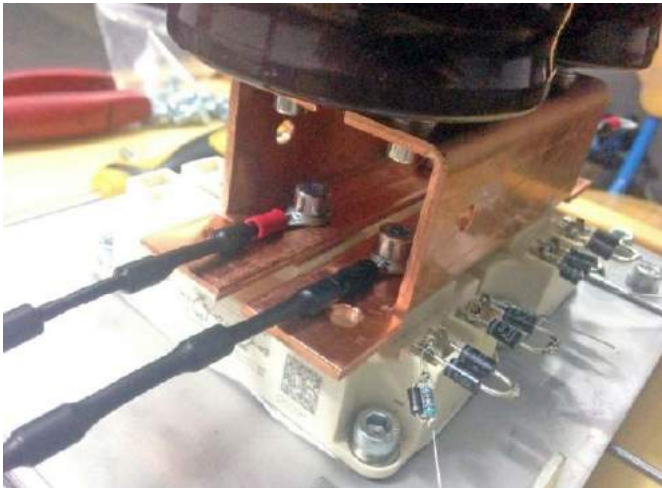
Fotografia 12. Oscylogram przebiegu napięcia w obwodzie pierwotnym cewki



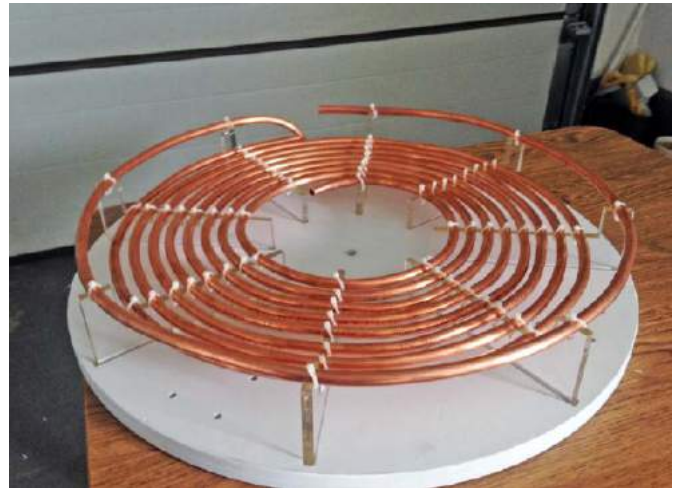
Fotografia 13. Wygląd zmontowanego falownika



Rysunek 10. Schemat elektryczny falownika



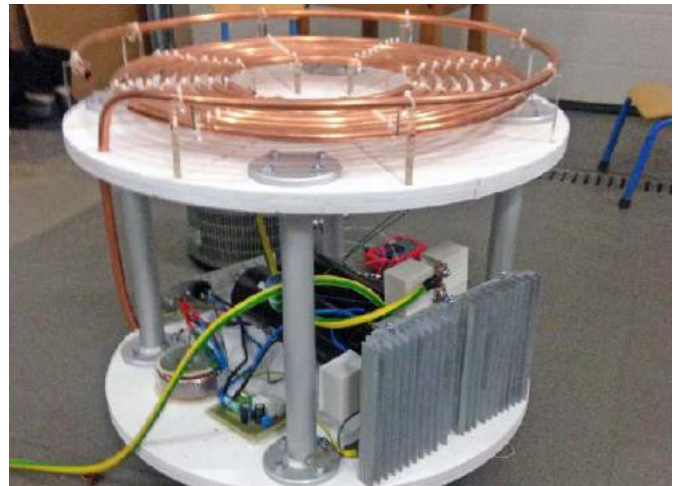
Fotografia 14. Sposób zamontowania miedzianych szyn



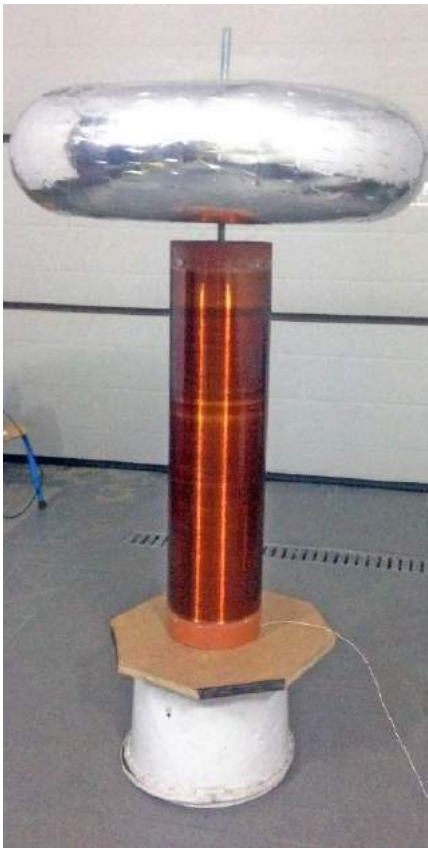
Fotografia 15. Konstrukcja uzwojenia pierwotnego



Fotografia 16. Elementy podstawy uzwojenia pierwotnego



Fotografia 18. Podstawa z zamontowanym osprzętem



Fotografia 17. Budowa obwodu wtórnego

przedmiotów w najbliższym otoczeniu transformatora na jego parametry. Każdorazowa zmiana powodowała bowiem rozstrojenie się układu. Kolejnym problemem była wytrzymałość prądowa kondensatora rezonansowego. W obwodzie pierwotnym przepływają bardzo duże prądy, rzędu setek amperów. Rozwiązałem to przez zastosowanie baterii kondensatorów, których parametry były uwzględnione z dużym marginesem.

Zbudowana przeze mnie cewka znajduje zastosowanie:

- Podczas pokazów wysokiego napięcia, koncertów, festynów, imprez okolicznościowych;
- W pracowni elektrotechnicznej i klasie fizycznej jako pomoc dydaktyczna do badania m.in. zjawiska rezonansu;
- Do bezprzewodowego przesyłania energii;
- Przy jonizacji gazów;
- Do sprawdzania wytrzymałości na wyładowania karoserii samochodów, samolotów, badań laboratoryjnych.

Bezpieczeństwo

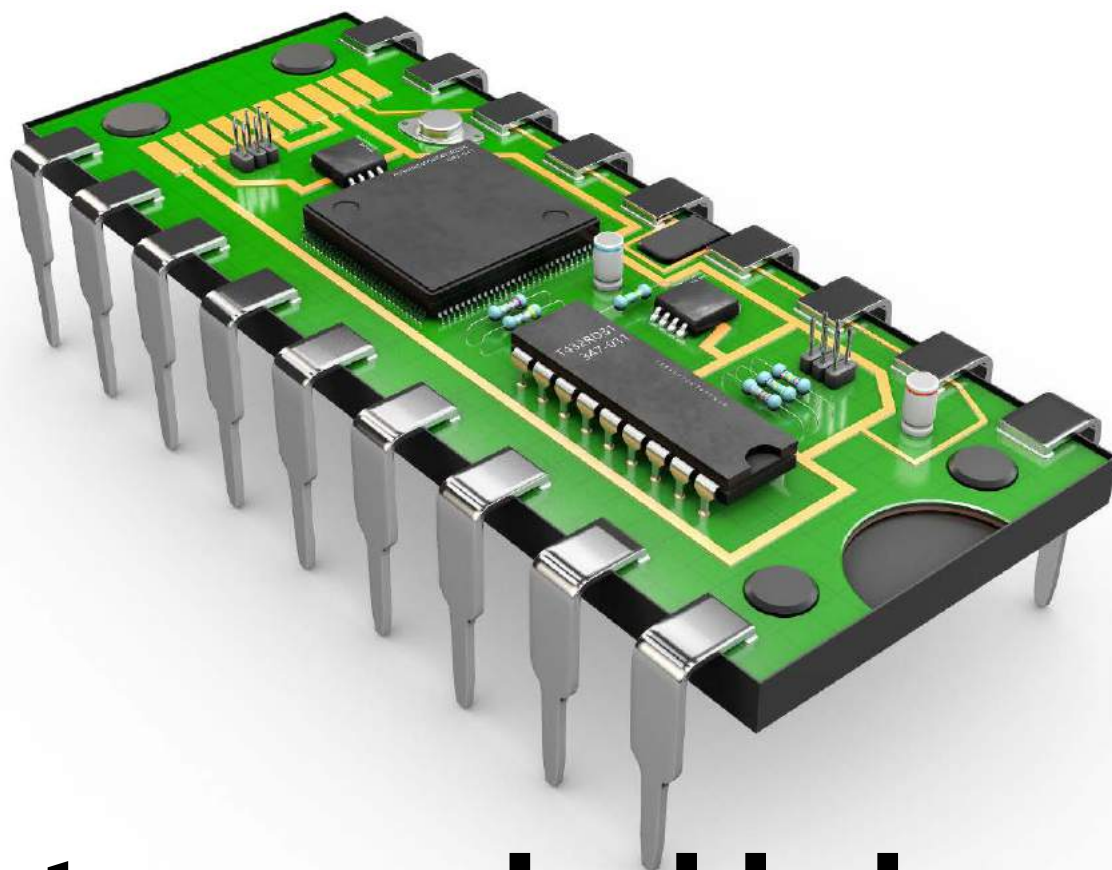
Podstawowym warunkiem bezpieczeństwa użytkowania transformatora Tesli

jest dobre uziemienie. Należy do niego podłączyć uziemienie elektroniki, wszystkie metalowe elementy, dolny koniec uzwojenia wtórnego oraz zwój zabezpieczający. Uziemienie zwoju zabezpieczającego gwarantuje ochronę obwodu pierwotnego przed wysokim napięciem. Osoba obsługująca cewkę musi znajdować się w bezpiecznej odległości (kilka metrów) od urządzenia. Samo urządzenie zostało zabezpieczone poprzez zastosowanie bezpieczników. Należy je również podłączać do instalacji elektrycznej z zastosowanym wyłącznikiem różnicowo-prądowym. Do uruchamiania cewki najlepiej używać autotransformatora, który pozwala na stopniowe podnoszenie napięcia zasilania falownika.

Jakub Jędrzejewski
kubajed@op.pl

Bibliografia:

1. <http://bit.ly/2JncXRR>
2. <http://bit.ly/2okvfMo>
3. <http://bit.ly/32RlvIC>
4. <http://bit.ly/340cLzY>
5. Film: <http://bit.ly/2MP0o4a>



Systemy embedded

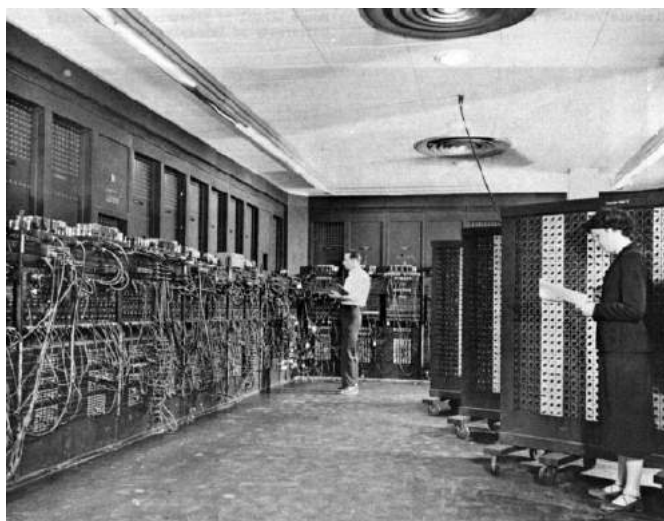
μCyfrowa ewolucja

Fascynującą cechą historii rozwoju systemów wbudowanych jest współlistnienie na rynku najnowocześniejszych układów SoC z legendarnymi 8-bitowymi mikrokontrolerami 8051, od których zaczęła się 40 lat temu era mikrokontrolerów. Panorama rozwiązań układowych i programistycznych, jakie pojawiły się przez te 40 lat i ciągle są stosowane, jest szersza niż Panorama Racławicka. Autor tego artykułu oprowadza nas po tej panoramie i jak dobry przewodnik rozjaśnia poszczególne jej fragmenty.

Redakcja

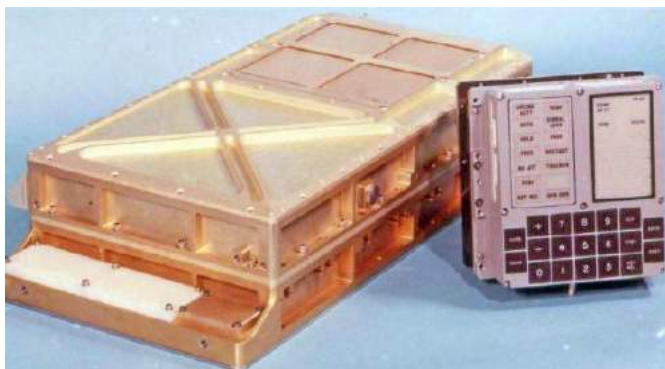
Pierwsze systemy komputerowe zajmowały całe pomieszczenia, ważyły kilkanaście ton i konsumowały niezmierzone ilości energii elektrycznej. Uważany za jeden z pierwszych komputerów ENIAC (fotografia 1) (choć tak naprawdę nie był pierwszy) zajmował powierzchnię 140 metrów kwadratowych, ważył 27 ton i pobierał energię o mocy 140 kW. Zbudowany był z ponad 18 tysięcy lamp elektronowych. Z uwagi na ograniczoną trwałość lamp oraz gigantyczny koszt komputery tego typu nie nadawały się do innych celów poza wykonywaniem specjalistycznych obliczeń.

W tamtych czasach nikt jeszcze nie myślał o używaniu komputerów do sterowania urządzeniami. Układy sterujące realizowane były głównie za pomocą prostej logiki zbudowanej na przekaźnikach oraz innych elementach elektromechanicznych. Pierwszy przełom nastąpił w 1948 roku wraz z wynalezieniem tranzystora, który rozpoczął



Fotografia 1. ENIAC – jeden z pierwszych komputerów

proces miniaturyzacji oraz stopniowego obniżania kosztów systemów komputerowych. W tym samym roku rozpoczęto również seryjną produkcję komputerów analogowych. Kolejnym ważnym krokiem było opracowanie na początku lat 60. pierwszych układów scalonych, co stanowiło kolejny krok w miniaturyzacji i wiązało się z obniżeniem kosztów i poboru mocy. Wówczas pojawiła się pierwsza idea zastosowania komputera do sterowania urządzeniami lub innymi obiektami. Na początku znalazła zastosowanie w urządzeniach wojskowych oraz w przemyśle kosmicznym.



Fotografia 2. Komputer używany w programie Apollo

Za jeden z pierwszych systemów wbudowanych, we współczesnym tego słowa znaczeniu, uważany jest komputer pokładowy zastosowany w programie Apollo, zbudowany w 1966 roku (fotografia 2). Używany był do sterowania systemem dowodzenia i modulem księżycowym. Komputer pracował z częstotliwością 2 MHz, miał 15-bitowe słowo maszynowe z jednym bitem parzystości, pamięć RAM o wielkości 2048 słów maszynowych oraz pamięć ROM o wielkości 36884 słów. Ważył 32 kg, a do jego zasilania wymagane było źródło o wydajności 55 W. Moc obliczeniowa była zbliżona do powstałego w 1981 roku 8-bitowego mikroprocesora 65c02. Jednak był to system zbudowany tylko w kilkunastu egzemplarzach. Pierwszym systemem komputerowym zastosowanym na szerszą skalę był komputer D17-B używany do sterowania pociskami międzykontynentalnymi „Minuteman I”. Został zbudowany z elementów dyskretnych i zawierał ponad 1500 tranzystorów oraz pod 6000 diod.

Wspomniane systemy były jednak bardzo drogie i jedynie wojsko oraz instytucje rządowe mogły sobie na nie pozwolić. Dalsze obniżenie kosztów i miniaturyzacja nastąpiły wraz z opracowaniem pierwszego 4-bitowego mikroprocesora 4004 a następnie produkowanego na masową skalę 8-bitowego mikroprocesora 8080. Do prawidłowego działania wymagał zewnętrznej pamięci RAM, pamięci ROM oraz kilkunastu dodatkowych układów scalonych, jak generatory sygnału zegarowego, układy czasowo-licznikowe czy układy wejścia-wyjścia. Systemy te były o rząd wielkości tańsze od wspomnianych wcześniej komputerów zbudowanych na układach scalonych małej skali integracji, jednak nadal były one stosunkowo drogie i stosowane tylko

w bardziej zaawansowanych urządzeniach. Opracowanie tańszego mikroprocesora 65c02 niewiele zmieniło, ponieważ do prawidłowego działania systemu komputerowego nadal potrzebne były dodatkowe układy zewnętrzne (fotografia 3).

Prawdziwy przełom nastąpił w 1976 roku wraz z opracowaniem układu 8048, który integrował w jednym układzie scalonym kompletny system: procesor, 1 kB pamięci ROM, 64 bajty pamięci RAM, układ czasowo-licznikowy oraz zewnętrzne porty GPIO. Integracja komputera w jednym układzie znacząco obniżała koszty systemu i z czasem okazało się, że taniej jest zbudować urządzenie w oparciu o mikrokontroler niż dedykowany układ. Mikrokontroler do prawidłowego działania potrzebował jedynie kilku elementów zewnętrznych. Dodatkowo okazało się, że wraz z rosnącą wydajnością obliczeniową układ sterujący z mikrokontrolerem może realizować szereg bardziej zaawansowanych funkcji.

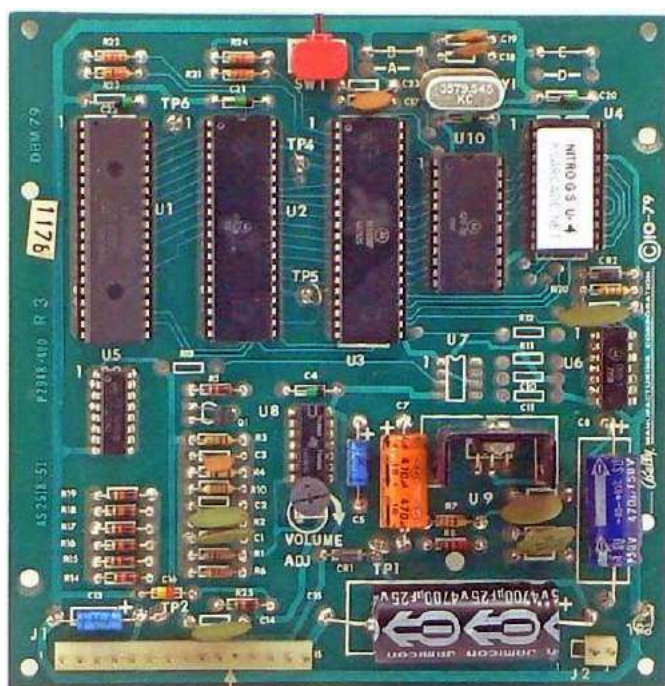
Definicja i podział

Zatem czym, od strony formalnej, jest system wbudowany? Jest to system komputerowy specjalnego przeznaczenia, który stanowi integralną część obsługiwanego przez niego sprzętu i służy do sterowania innymi obiektami. Trudno sobie dzisiaj wyobrazić nowoczesny sprzęt elektroniczny, zarówno ten konsumencki, jak i profesjonalny, bez możliwości swobodnego wyboru parametrów, konfiguracji i interakcji z użytkownikiem. Jest to możliwe dzięki zastosowaniu mikroprocesorów sterujących pracą tych urządzeń. Systemy wbudowane spotykamy w wielu miejscach, m.in. w zegarkach elektronicznych, telefonach komórkowych, telewizorach, odtwarzaczach, sprzęcie gospodarstwa domowego, samochodach, inteligentnych budynkach, przyrządach pomiarowych czy automatach. W zasadzie trudno obecnie znaleźć nawet najprostsze urządzenie, które byłoby zbudowane bez użycia mikroprocesora. Trend ten możemy wyraźnie dostrzec, obserwując publikacje w naszym czasopiśmie. Jeśli spojrzymy na opisy urządzeń opublikowanych na łamach EP w 1994 roku, projekty zbudowane w oparciu o mikrokontrolery stanowią rzadkość. Czasami pojawia się opis jednego lub dwóch urządzeń zbudowanych z użyciem mikrokontrolera, natomiast w niektórych numerach w ogóle ich nie ma. Znaczną większość urządzeń tamtego okresu stanowiły układy analogowe i cyfrowe zbudowane z wykorzystaniem elementów dyskretnych oraz układów scalonych małej skali integracji. Natomiast gdy spojrzymy na dowolny numer czasopisma z bieżącego roku, sytuacja wygląda odmiennie. Urządzenia i układy zrealizowane bez mikrokontrolerów lub układów SoC są w mniejszości, a urządzenia zrealizowane jedynie przy użyciu elementów dyskretnych stanowią zupełną rzadkość.

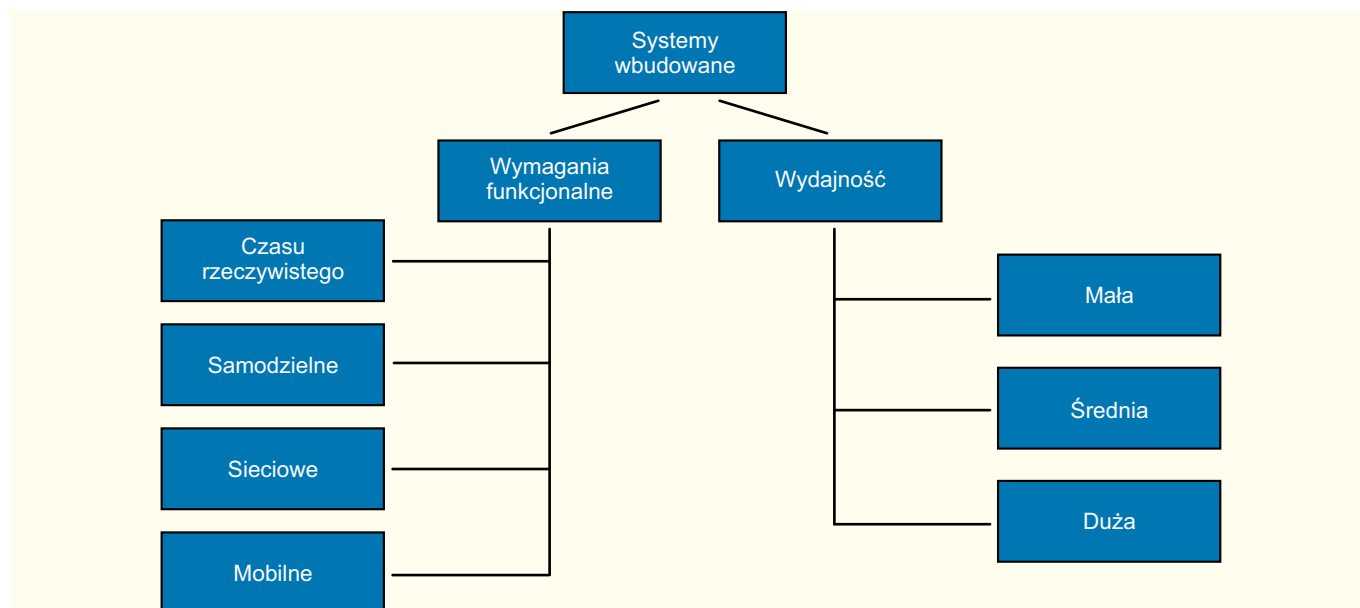
Cechy, które stanowią o tym, że system komputerowy jest systemem wbudowanym, a nie systemem komputerowym ogólnego przeznaczenia, to:

- Sterowanie odbywa się poprzez umieszczenie wewnątrz urządzenia (obiektu) określonego systemu komputerowego, trwale połączonego z danym obiektem, wyposażonego w specjalne oprogramowanie realizujące określone funkcje.
- System komputerowy spełnia określone wymagania zdefiniowane do zadań, które ma wykonywać.
- Jest oparty na mikroprocesorze (lub mikrokontrolerze) zaprogramowanym do wykonywania ograniczonej liczby zadań a nawet tylko do jednego.
- Zależnie od złożoności wykonywanych zadań, może zawierać oprogramowanie dedykowane wyłącznie temu urządzeniu (firmware) lub może to być system operacyjny ogólnego przeznaczenia wraz z zainstalowanym specyficznym oprogramowaniem.
- Podstawową cechą wyróżniającą systemy wbudowane na tle innych systemów komputerowych, oprócz dedykowanego charakteru, jest jakość oprogramowania oraz zastosowane komponenty systemu komputerowego.

Ogólna zasada jest taka, że im mniej złożone jest oprogramowanie i bardziej wyspecjalizowane, tym bardziej niezawodny jest system



Fotografia 3. System komputerowy z mikrokontrolerem 65c02



Rysunek 4. Podział systemów wbudowanych

i pozwala szybciej reagować na zdarzenia krytyczne. Niezawodność systemu może być zwiększona poprzez rozdzielenie zadań na mniejsze podsystemy a także przez redundancję. Może ona polegać na zastosowaniu dwóch identycznych urządzeń do jednego zadania, z których drugie przejmie zadania pierwszego w przypadku awarii jednego z nich. Jest to szczególnie istotnie w przypadku systemów odpowiedzialnych za nadzorowanie funkcji związanych z życiem i bezpieczeństwem (np. lotnictwo, medycyna). Systemom wbudowanym stawiane są różne wymagania, dlatego zastosowane oprogramowanie oraz systemy komputerowe mogą się znacząco różnić od siebie.

Systemy wbudowane możemy podzielić według dwóch kategorii. Pierwsza określa wymagania funkcjonalne, druga uwzględnia wymaganą moc obliczeniową (rysunek 4). Systemy o małej wydajności realizują proste czynności i współcześnie budowane są z zastosowaniem mikrokontrolerów 8-bitowych np. z rodziny PIC czy AVR lub najprostszych mikrokontrolerów 32-bitowych z rdzeniem Cortex-M0/M0+ np. STM32L0/F0. Projekt płytki PCB z reguły zawiera kilkanaście elementów zewnętrznych, a oprogramowanie pracuje samodzielnie bez nadzoru systemu operacyjnego. Systemy tego typu możemy zbudować z prostych układów skali LSI (dużej skali integracji, LSI – *large scale of integration*), jednak z uwagi na dużo niższe koszty i większą funkcjonalność produkowane są w postaci systemu wbudowanego. Interfejs użytkownika jest nieskomplikowany i najczęściej składa się z kilku przycisków i wyświetlacza alfanumerycznego lub 7-segmentowego LED (fotografia 5). W najprostszych przypadkach informacja o stanie systemu obrazowana jest za pomocą diod LED.

Systemy operacyjne średniej wydajności najczęściej realizowane są z zastosowaniem pojedynczego 32-bitowego mikrokontrolera, na przykład z rdzeniem MIPS (PIC32) lub ARM Cortex-M3/M4/M7 (STM32/LPC4xxx itp.) czy nawet na bazie prostego procesora DSP. Płytki PCB jest stosunkowo rozbudowana i może zawierać wiele elementów dodatkowych, jednak nie zawiera zewnętrznej pamięci FLASH czy RAM. Oprogramowanie również jest bardziej

rozbudowane i najczęściej zawiera system operacyjny czasu rzeczywistego oraz dodatkowe biblioteki służące do realizacji skomplikowanych zadań, takich jak: obsługa protokołu TCP/IP, obsługa stosu USB czy obsługa interfejsu graficznego. Interfejs użytkownika może być dość złożony i rozbudowany, wyposażony w wyświetlacz graficzny monochromatyczny lub kolorowy oraz panel dotykowy.

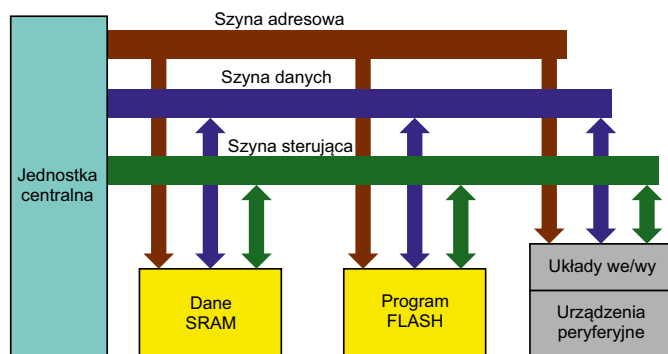
Duże systemy wbudowane mogą składać się z wielu procesorów, w tym DSP, układów programowalnych, czy rozbudowanych układów SoC wyposażonych w 64-bitowe wielordzeniowe CPU. Płytki PCB w tego typu systemach są bardzo rozbudowane, gdyż pamięć operacyjna, zewnętrzna pamięć FLASH oraz inne komponenty pracujące przy dużej częstotliwości wymagają zastosowania wielobitowych szybkich magistral. Oprogramowanie, w którym procesor aplikacyjny najczęściej pracuje pod kontrolą systemu Linux, jest również zaawansowane. Inne procesory mogą pracować pod kontrolą systemów czasu rzeczywistego realizujących zadania krytyczne czasowo. Systemy tego typu najczęściej dysponują dużymi ekranami dotykowymi lub zewnętrznymi monitorami dołączonymi np. poprzez port HDMI.

Bardzo istotnym kryterium podziału systemów wbudowanych jest funkcjonalność. Inne wymagania stawiamy systemom sterującym oświetleniem, gdzie czas reakcji nie ma większego znaczenia, a zupełnie inne systemom sterującym robotami przemysłowymi, gdzie konieczne jest zastosowanie systemu czasu rzeczywistego. Możemy wyszczególnić takie grupy:

- **Samodzielne systemy wbudowane** – systemy komputerowe zdolne działać niezależnie, bez konieczności współpracy z innymi urządzeniami np. komputerem nadrzędnym. Zbierają dane z wejść analogowych lub cyfrowych, wykonują obliczenia, a wyniki operacji przetwarzają na stany odpowiednich wyjść analogowych lub cyfrowych. Przykłady takich systemów wbudowanych to m.in: sterownik pralki automatycznej, kamera wideo, termostat lodówki, itp.
- **Systemy sieciowe** – charakteryzują się tym, że mają możliwość komunikowania się z innymi obiektami, np. za pomocą sieci komputerowych LAN, modemu GSM, modemu Wi-Fi, itp. Przykładami takich systemów są instalacje inteligentnych budynków czy kamery internetowe.
- **Systemy mobilne** – systemy przenośne, szczególnie nacisk kładziony jest na przenośność oraz minimalizację zużycia energii. z uwagi na to, że korzystają głównie z zasilania bateryjnego. Typowymi przedstawicielami takich systemów są: telefony komórkowe, odtwarzacze MP3, kamery, aparaty fotograficzne itp.



Fotografia 5. System wbudowany z mikrokontrolerem 32-bitowym



Rysunek 6. Uproszczony schemat wewnętrzny typowego mikrokontrolera

- **Systemy operacyjne czasu rzeczywistego** – charakteryzują się reakcją w ściśle zadanym reżimie czasowym. Systemy tego typu muszą odpowiedzieć na zadane wymuszenie w ściśle określonym czasie, ponieważ brak odpowiedzi może skutkować niepoprawnym działaniem, a nawet uszkodzeniem obiektu, którym system steruje. Takie sterowniki najczęściej wykonywane są z zastosowaniem mikrokontrolera lub mikroprocesora bez jednostki zarządzania pamięcią MMU (Memory Management Unit), ponieważ pamięć wirtualna nie gwarantuje odpowiedzi systemu w czasie deterministycznym. Do ochrony pamięci (jeśli jest potrzebna), stosuje się układ MPU (Memory Protection Unit), który umożliwia ustawienie zakresów ochronnych obszarów pamięci fizycznej, bez wirtualnej przestrzeni adresowej.

Przedstawione kategorie mogą się nawzajem przenikać, np. system wbudowany czasu rzeczywistego może być równocześnie systemem sieciowym oraz mobilnym.

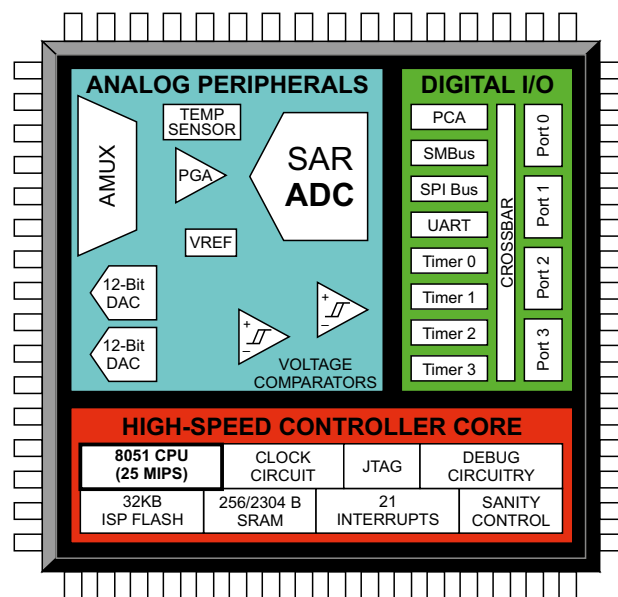
Sprzęt stosowany w systemach wbudowanych

W systemach wbudowanych najczęściej stosowany jest sprzęt oparty na mikrokontrolerach jednoukładowych lub układach mikroprocesorowych SoC (System on Chip), przy czym z reguły małe systemy wbudowane (np. myszki, klawiatury, sterowniki klimatyzacji itp.) z uwagi na cenę i niewielkie wymiary korzystają z mikrokontrolerów jednoukładowych. Bardziej skomplikowane średnie i duże systemy mogą korzystać z mikroprocesorowych układów SoC. Najbardziej zaawansowane układy mogą zawierać kilka mikrokontrolerów, układów SoC oraz układy programowalne. Zastosowanie układów programowalnych FPGA nie jest powszechnie spotykane, najczęściej z uwagi na wysoki koszt. Ma miejsce wówczas, gdy nie ma możliwości użycia innego tańszego rozwiązania.

Mikrokontrolery

Mikrokontrolery jednoukładowe stanowią główny wybór w przypadku małych i średnich systemów wbudowanych. Dzięki integracji w jednym układzie jednostki centralnej pamięci FLASH, RAM oraz wszystkich innych układów peryferyjnych (rysunek 6) umożliwiają znaczące obniżenie kosztów urządzenia. Gdy jeden układ stanowi kompletny system mikroprocesorowy, kompletne urządzenie otrzymujemy, dołączając do portów zewnętrznych kilka dodatkowych elementów. Płytkę PCB również jest stosunkowo prosta, ponieważ nie ma konieczności prowadzenia szerokich magistral do pamięci. Może być zrealizowana w technice dwuwarstwowej.

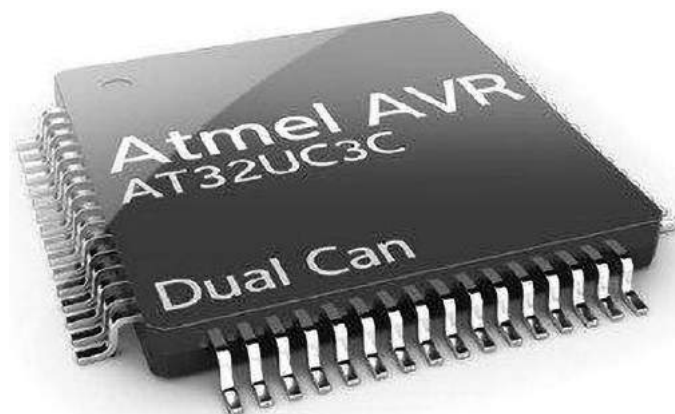
Na rynku istnieje bardzo wiele rodzajów mikrokontrolerów, od niewielkich 8-bitowych układów AVR, do bardzo rozbudowanych mikrokontrolerów z rdzeniem ARM, takich jak STM32. Pierwsze mikrokontrolery jednoukładowe miały niewielkie zasoby sprzętowe. Typowym przykładem jest popularny układ 8051 zaprojektowany przez firmę Intel. Zawiera jednostkę CPU o wydajności 1 DMIPS @12 MHz, 128 B pamięci RAM oraz 8 kB pamięci ROM lub FLASH. Mikrokontrolery tej rodziny były powszechnie spotykane w małych



Rysunek 7. Mikrokontroler rodziny C8051F000 firmy Silicon Labs

systemach do około 2000 roku, później zaczęły stopniowo wychodzić z użycia zastępowane przez lepsze konstrukcje 8-bitowe. Produkowane są do dziś, np. firma Microchip produkuje klasyczne mikrokontrolery z pamięcią FLASH, są to 80c51, 89C2051. Podobne układy znajdziemy także w ofertach takich firm jak: Infineon, Analog Devices czy Silicon Labs. Niektóre mają znacznie ulepszony rdzeń wykonujący instrukcje w 1 lub 2 cyklach zegarowych i występują w nowoczesnych obudowach SMD TQFP, więc mogą być stosowane we współczesnych produktach. Przykładem jest rodzina 8051F00/1/2/5/7 Silicon Labs (rysunek 7) wyposażona w znacznie ulepszony rdzeń 8051 z potokowym przetwarzaniem instrukcji, w większości wykonywanych w jednym lub dwóch cyklach zegarowych. Zawiera 2 kB pamięci XRAM, 256 B pamięci wewnętrznej oraz szeroki zestaw urządzeń peryferyjnych wraz z pamięcią ISP FLASH. Chociaż tworzenie nowych urządzeń w oparciu na mikrokontrolerze z rdzeniem 8051 jest nieco dyskusyjne, układy tego typu mogą znaleźć zastosowanie wszędzie tam, gdzie istnieje starszy kod napisany np. w assemblerze i gdzie nie opłaca się tworzyć nowego oprogramowania od podstaw.

Pomimo że inwazja 32-bitowych mikrokontrolerów ARM trwa już od jakiegoś czasu i możemy kupić mikrokontroler z rdzeniem Cortex-M0 za kilka centów, nieco nowocześniejsze 8-/16-bitowe mikrokontrolery nadal cieszą się dużą sympatią konstruktorów. Zapewne dlatego, że przygotowanie oprogramowania dla takiego mikrokontrolera jest znacznie prostsze niż w przypadku układów z rdzeniem ARM. Dla wielu peryferii wystarczy ustawić kilka bitów w rejestrach konfiguracyjnych i układ już działa. Obecnie najchętniej



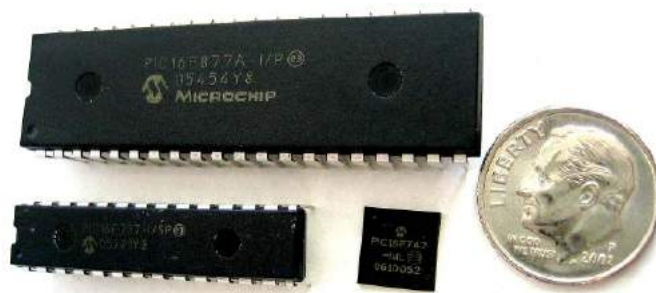
Rysunek 8. Mikrokontroler z rdzeniem AVR

stosowanymi układami 8-bitowymi są mikrokontrolery AVR i PIC, a wśród układów 16-bitowych – PIC i dsPIC.

Mikrokontrolery AVR (**rysunek 8**) swoją znaczącą popularność zyskały po 2002 roku, stopniowo wypierając bardzo popularne w tamtych czasach układy oparte na rdzeniu 8051. Dzięki unowocześnionej jednostce centralnej pracowały kilkanaście razy szybciej, pozwalając na znaczne zmniejszenie poboru energii. Szczególną sympatię zyskały w naszym kraju dzięki umiarkowanej cenie i dobrej polityce dystrybutorów. Do 2003 roku producent (Atmel) sprzedał ponad 500 milionów egzemplarzy mikrokontrolerów tego typu. Rdzeń AVR jest zbudowany w oparciu o architekturę harwardzką, z oddzielnymi instrukcjami dostępu do pamięci programu oraz do pamięci danych. Dzięki przetwarzaniu potokowemu większość 8-bitowych instrukcji jest wykonywana w 1 lub 2 cyklach zegarowych. Mikrokontrolery AVR zostały podzielone na kilka podtypów w zależności od przeznaczenia, dostępnej pamięci oraz układów peryferyjnych:

- **TinyAVR** – przeznaczone do najprostszych i najtańszych systemów wbudowanych. Rdzeń mikrokontrolera ma uproszczony zestaw instrukcji i pozbawiony jest np. operacji mnożenia. Zawiera również mocno ograniczony zestaw układów peryferyjnych oraz niewielką pamięć FLASH o rozmiarze od 0,5 kB do 32 kB. Układy tego typu dostępne są w niewielkich obudowach od 6 do 32 wyprowadzeń a ich cena jest na tyle atrakcyjna, że opłaca się je stosować jako zamiennik dla kilku układów cyfrowych małej skali integracji. Najmniejszym układem tego typu jest ATTINY4 w obudowie SOT23-6 mający pamięć FLASH o wielkości 512 B i pamięć RAM o wielkości 32 bajtów.
- **MegaAVR** – stanowi główną linię produktową, układy dysponują pamięcią FLASH z przedziału 4...256 kB oraz występują w wielu typach obudów od 28 do 100 wyprowadzeń. Mogą zawierać wiele, ciekawych układów peryferyjnych, a rdzeń dysponuje standardowym zestawem instrukcji, takich jak dostęp do pamięci o rozmiarze większym od 64 kB czy instrukcja mnożenia. Brakuje jednak instrukcji dzielenia DIV (której architektura AVR nie zawiera). Jednymi z najpopularniejszych w naszym kraju były mikrokontrolery ATMEGA8/48/88/16/32 z uwagi na dobry stosunek możliwości do ceny. Duża część projektów publikowanych na łamach EP wyposażona była w któryś ze wspomnianych układów.
- **XMEGA** – to najbardziej zaawansowana seria rodziny AVR. Układy zawierają zaawansowane bloki poprawiające wydajność obliczeniową, takie jak m.in. kontroler bezpośredniego dostępu do pamięci (DMA), system obsługi zdarzeń czy koprocessor kryptograficzny. Wyposażone są w pamięć programu o pojemności 16...384 kB i występują w obudowach do 100 wyprowadzeń. Również znacząco podniesiono częstotliwość taktowania rdzenia, która może wynosić 50 MHz. Układy tego typu, z uwagi na to, że powstały w okresie, gdy na rynku zdomowały się mikrokontrolery 32-bitowe, nie zyskały dużej popularności.

Kolejną rodziną, która nadal cieszy się stosunkowo dużą popularnością (choć w Europie nie tak wielką), jest rodzina mikrokontrolerów PIC (**fotografia 9**) opracowana przez firmę Microchip Technology. Mikrokontrolery 8-bitowe i 16-bitowe tej rodziny do dziś mają swoich zagorzałych sympatyków. Pierwsze, dostępne z pamięcią ROM, pojawiły się na rynku już w 1976 roku, a do 2013 roku firma sprzedawała ponad 12 miliardów układów tej rodziny. Podobnie jak poprzednik, rdzeń mikrokontrolera oparty jest na architekturze harwardzką i większość instrukcji wykonuje w 2 lub 4 cyklach zegarowych. W przeciwieństwie do nowoczesnych architektur posiada zestaw rejestrów dedykowanych określonym zadaniom np. WO akumulator czy sprzętowy stos systemowy,

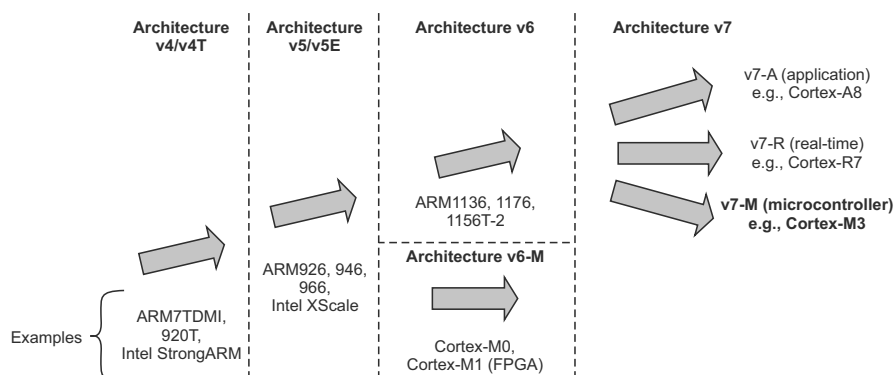


Fotografia 9. Mikrokontrolery PIC w różnych obudowach

na którym przechowywane się adresy powrotów z podprogramów. W starszych wersjach stos nie był adresowany programowo, co zostało zmienione dopiero od serii F18. Powyższe cechy architektury sprawiają, że nie jest dobrze dopasowany do zastosowania kompilatorów języków wyższego poziomu jak C czy implementacji systemów operacyjnych z wyłączeniem (niemniej jednak takie kompilatory istnieją). Układy zostały podzielone na podrodziny służące do segmentacji rynku – do dyspozycji mamy następujące podtypy:

- **PIC10/12** – występują w obudowach o wielkości od 6 do 40 wyprowadzeń. Jest to seria najprostszych układów przeznaczona do podstawowych rozwiązań. Ma niewielką przestrzeń adresową programu do 512 słów (12-bitowych), jednym z najbardziej znanych mikrokontrolerów tej serii jest PIC10f32x.
- **PIC16** – bardzo często spotykane w małych systemach. Wyposażone w pamięć danych o wielkości do 2048 słów 15-bitowych oraz o 8-poziomowym stosie sprzętowym.
- **PIC18** – wraz z nastaniem nowego tysiąclecia programiści stopniowo odchodzili od programowania systemów wbudowanych w assemblerze na rzecz języka C, co spowodowało, że konieczna stała się modyfikacja architektury PIC tak, aby była przyjazna dla kompilatorów. W ten sposób powstała rodzina **PIC18**, która uzyskała głębszy oraz programowo adresowany stos o głębokości 31 poziomów, dodatkowe instrukcje warunkowe oraz liniową adresowalną przestrzeń rejestrów specjalnego przeznaczenia.
- **PIC24 i dsPIC** – bazując na architekturze PIC18, w kolejnych latach opracowano 16-bitowe układy PIC24 i dsPIC przeznaczone do średnio zaawansowanych rozwiązań oraz obliczeń DSP. Mikrokontrolery mają rejestry o rozmiarze 16 bitów, zwiększony licznik rozkazów do 24 bitów, a rozmiar pamięci programu może wynosić 64 kB. Wersja dsPIC dodatkowo zawiera zestaw instrukcji przydatny podczas cyfrowego przetwarzania sygnałów, takich jak instrukcję mnożenia z akumulacją, odwracanie bitów, dzielenie sprzętowe itp.

Mikrokontrolery 8-bitowe, które jeszcze 15 lat temu pełniły główną funkcję w systemach wbudowanych, wraz z nastaniem „ARM-owej rewolucji” ustąpiły pola mikrokontrolerom 32-bitowym, głównie z rdzeniem ARM. Rewolucję rozpoczęła firma NXP w 2004 roku,



Rysunek 10. Evolucja rodziny ARM

wprowadzając konkurencyjne cenowo, dużo szybsze układy LPC2000 z 32-bitowym rdzeniem ARM7TDMI-S, co spowodowało masowe odchodzenie od rozwiązań 8-bitowych. Układy ARM swoje pierwotne korzenie mają w mikroprocesorach przeznaczonych do komputerów domowych oraz w większych układach SoC (**rysunek 10**). Pierwszym układem zmodyfikowanym z myślą o mikrokontrolerach był rdzeń ARM7TDMI-S, który wywodził się z mikroprocesorów ogólnego przeznaczenia, przez co nie do końca był dopasowany do systemów wbudowanych. Największe wady tego układu to brak jasno zdefiniowanych trybów oszczędzania energii oraz zewnętrzny kontroler przerwań VIC, który nie potrafił obsługiwać wyłączenia. Powyższe wady zmusiły firmę ARM do opracowania jednostek centralnych przeznaczonych do mikrokontrolerów i w ten sposób powstał rdzeń Cortex-M3, a następnie dostosowano do różnych segmentów rynku jego kolejne odmiany, takie jak Cortex-M0/M4/M7. Bardzo istotną cechą jest to, że firma ARM nie produkuje gotowych układów, a jedynie licencjonuje jednostki centralne innym producentom. Na rynku możemy nabyć setki różnych typów układów różnych producentów, z różnymi układami peryferyjnymi, ale mających taki sam zestaw instrukcji. Jest to duża zaleta, ponieważ możemy wykorzystać ten sam zestaw narzędzi, np. kompilator debugger, bez konieczności zakupu nowych narzędzi przy zmianie producenta. Nie bez znaczenia jest również dostęp do wszelakich narzędzi na licencji open source, co znacząco obniża koszty wdrożenia projektu. Współcześnie możemy powiedzieć, że mikrokontrolery z jednostką centralną ARM stanowią standard przemysłowy, tak jak 30 lat temu funkcję tego standardu pełniły układy oparte na jednostce 8051.

We współczesnych mikrokontrolerach z rdzeniem ARM stosowane są rdzenie Cortex występujące w czterech odmianach:

- **Cortex-M0/M0+** – przeznaczone do najprostszych układów, cenowo konkurujące z układami 8-bitowymi. Zbudowane w oparciu o architekturę **ARMv6m** z trzystopniowym przetwarzaniem potokowym, zintegrowanym kontrolerem przerwań obsługującym maksymalnie 32 przerwania i wydajnością wynoszącą 0,9 DMIPS/MHz. Produkowane przez wiele różnych firm, przy czym najpopularniejsze są układy produkcji NXP LPC1100/1200/800, STMicroelectronics STM32F0/L0, Toshiba TX00, Renesas S1 itp. Najczęściej mają pamięć programu o rozmiarze 2...32 kB oraz pamięć danych do 8 kB i dostępne są w obudowach o wielkości od 5 do 64 wyprowadzeń. Zestaw układów peryferyjnych jest stosunkowo ubogi. Najczęściej dostępne są podstawowe układy peryferyjne, takie jak porty szeregowy, kontroler SPI, I²C itp., a ich częstotliwość pracy wynosi maksymalnie kilkadziesiąt MHz.
- **Cortex-M3** – obecnie najpowszechniej używane układy z rdzeniem ARM z uwagi na stosunkowo prostą budowę oraz bardzo dobry stosunek możliwości do ceny. Zbudowane są w oparciu o architekturę **ARMv7m** z 3-stopniowym przetwarzaniem potokowym oraz kontrolerem przerwań obsługującym do 256 źródeł zewnętrznych. Opcjonalnie mogą zostać wyposażone w jednostkę FPU, realizującą sprzętową obsługę obliczeń na liczbach zmiennoprzecinkowych, co jednak nie jest praktykowane. Typowymi przedstawicielami serii są mikrokontrolery produkowane przez firmę NXP (LPC13xx, LPC17xx), STMicroelectronic (F1, F2). Charakteryzują się pamięcią programu o rozmiarze 32...256 kB oraz pamięcią danych od 6 do 128 kB. Najczęściej wyposażone są w bogaty zestaw układów peryferyjnych, takich jak kontroler bezpośredniego dostępu do pamięci (DMA), kontroler USB, kontroler Ethernet itp. Mogą one również pracować z większą częstotliwością sięgającą 120 MHz.
- **Cortex-M4** – dysponują znacznie większą wydajnością, są nieco droższe, jednak ich cena najczęściej mieści się w granicach 30 zł. Zbudowane są w oparciu o architekturę **ARMv7E-m** z 3-stopniowym przetwarzaniem potokowym wyposażonym w jednostkę przewidywania rozgałęzień, ponadto wyposażone są w zestaw instrukcji służących do realizacji algorytmów DSP oraz



Fotografia 11. Mikrokontroler PIC32

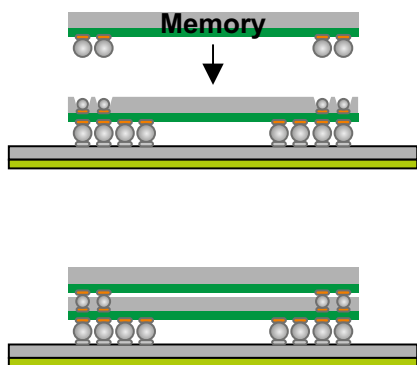
opcjonalną jednostkę zmiennoprzecinkową o pojedynczej precyzji. Typowymi przedstawicielami tej linii są mikrokontrolery NXP LPC40xx/53xx STM32F3/4 czy Texas Instruments (TM4C). Wyposażone są w pamięć o rozmiarze od 32 do 1024 kB, pamięć RAM 8...512 kB oraz bogaty zestaw układów peryferyjnych. Pracują z częstotliwością zbliżoną do 200 MHz i charakteryzują się wydajnością rzędu 1,25 DMIPS/MHz.

- **Cortex-M7** – najbardziej wydajne mikrokontrolery oraz najdroższe. W stosunku do poprzedników zostały znacznie rozbudowane, jednostka centralna bazuje na rdzeniu **ARMv7E-M** z usprawnionym 6-stopniowym przetwarzaniem potokowym i towarzyszy jej jednostka zmiennoprzecinkowa o podwójnej precyzji. Mogą być wyposażone w pamięć **CACHE** pierwszego poziomu dla instrukcji oraz danych do 64 KB. Mają pamięć programu o wielkości do 2 MB, pamięć RAM do 1 MB oraz zaawansowane układy peryferyjne. Pracują z częstotliwością ponad 200 MHz, osiągając wydajność rzędu 2,14 DMIPS/MHz. Typowymi przedstawicielami tej linii są układy: ATMEEL SAMS70, STMicroelectronic STM32F7 czy Freescale KV5x. Niestety z tak dużą wydajnością wiąże się odpowiednio wysoka cena, sięgająca kilkadziesiąt złotych za sztukę. Często w tego typu układach pamięć wewnętrzna bywa niewystarczająca, zwłaszcza gdy konieczne jest podłączenie kolorowego wyświetlacza TFT. Czyni to wybór tego typu mikrokontrolerów rozwiązaniem dyskusyjnym, zwłaszcza, że w obecnych czasach w porównywalnych cenach możemy dostać całe komputery zbudowane w oparciu o układy SoC, potrafiące uruchamiać pełnoprawne systemy operacyjne. Jeśli nie potrzebujemy pracować z systemami czasu rzeczywistego w twardym reżimie czasowym, układy SoC często okazują się lepszym wyborem.

Oprócz układów z rdzeniem ARM, które w znacznym stopniu zdominowały rynek, możemy się również spotkać z układami **PIC32** (**fotografia 11**). Firma Microchip jak zawsze lubi chadzać własnymi ścieżkami i nie poddała się „ARM-owej rewolucji”, stosując w swoich mikrokontrolerach architekturę MIPS32 o wydajności i parametrach zbliżonych do architektury ARM. Na bazie tej jednostki powstały cztery rodziny mikrokontrolerów oparte o różne rodzaje architektury MIPS32:

- **PIC32MX** – oparta o rdzeń M4K, wydajnością zbliżona do Cortex-M3;
- **PIC32MZ** – pracująca z częstotliwością maksymalną 250 MHz i wydajnością zbliżoną do układów zbudowanych w oparciu o Cortex-M7;
- **PIC32MM** – przeznaczona do zastosowań ekonomicznych oraz energooszczędnych, stanowiąca odpowiednik architektury Cortex-M0;
- **PIC32MK** – przeznaczona do zastosowań motoryzacyjnych.

Pomimo, że dostępne są darmowe środowiska programistyczne, układy nie zdobyły wielu zwolenników i stosowane są głównie w motoryzacji oraz przez konstruktorów przyzwyczajonych do firmy Microchip.



Rysunek 12. Budowa układu w technologii PoP

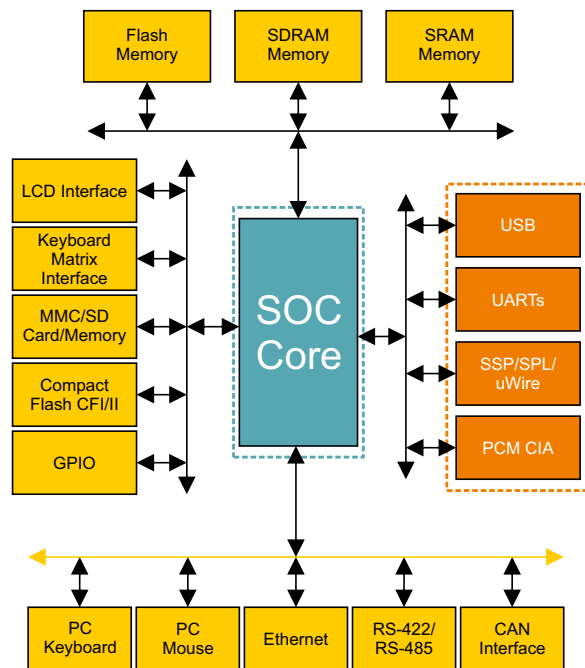
Układy SoC

Więszymi kuzynami mikrokontrolerów są układy SoC, czyli układy scalone zawierające kompletny system elektroniczny, w tym układy cyfrowe, analogowe (także radiowe) oraz cyfrowo-analogowe. Poszczególne moduły tego systemu, ze względu na swoją złożoność, pochodzą zazwyczaj od różnych dostawców. Jednostka centralna może pochodzić od jednego dostawcy np. ARM, a porty komunikacji szeregowej od innego dostawcy lub stanowić własne opracowanie. Typowym obszarem zastosowań SoC są systemy wbudowane, a najbardziej rozpowszechnione są systemy oparte na procesorach w architekturze ARM. W przypadku układów SoC najczęściej używa się rdzeni procesorów aplikacyjnych rodziny **ARM Cortex-A**. Mają one jednostkę zarządzania pamięcią MMU (*Memory Management Unit*) wraz z rozbudowanymi układami pamięci cache. Mogą również występować w odmianach wielordzeniowych, pracujących w układzie multiprocesorowości symetrycznej (**SMP**).

Jeżeli zintegrowanie wszystkich bloków SoC na jednym podłożu półprzewodnikowym jest niemożliwe, poszczególne moduły wykonuje się na oddzielnych strukturach a całość zamyka w jednej obudowie. Rozwiązanie takie określane jest mianem **SiP** (*System-in-package*), jednak jest mniej opłacalne ekonomicznie, zwłaszcza przy produkcji półprzewodników w dużych seriach.

Innym rozwiązaniem, stosowanym np. w telefonach komórkowych, gdy we wnętrzu nie ma wiele miejsca, jest technologia **PoP** (*Package-on-package*). Polega na zastosowaniu dwu układów scalonych, z których jeden jest montowany pod drugim. Zwykle układ znajdujący się pod spodem jest układem typu SoC, a na nim umieszczony jest układ pamięci RAM lub/i FLASH (**rysunek 12**).

Największą różnicą pomiędzy mikrokontrolerami a układami SoC jest ilość pamięci, jakiej wymagają do pracy. W przypadku mikrokontrolerów jest to zwykle poniżej 100 kB pamięci RAM oraz mniej niż 1 MB pamięci FLASH. Układy SoC potrzebują zewnętrznych układów pamięci RAM i FLASH, a określenie „*System on chip*” jest, głównie w przypadku większych systemów, tylko przenośnią, która określa ukierunkowanie prac projektowych niż rzeczywistą realizację. Ponadto SoC wyposażone są w CPU o stosunkowo dużej mocy obliczeniowej, które dzięki jednostce zarządzania pamięcią MMU pozwalają uruchamiać systemy operacyjne znane z dużych systemów komputerowych, takie jak Linux czy QNX, a także mogą obsługiwać bardziej wyspecjalizowane peryferia. Musimy jednak pamiętać, że tego typu rozwiązania nie nadają się do zastosowania w twardych systemach czasu rzeczywistego, gdzie potrzebny jest maksymalnie deterministyczny czas odpowiedzi. Pamięć wirtualna oraz jednostka zarządzania pamięcią MMU powoduje, że czas reakcji systemu staje się niedeterministyczny. Dzieje się tak, ponieważ translacja adresu wirtualnego (**VA**) na adres fizyczny (**PA**) zajmuje układowi MMU pewien czas potrzebny na znalezienie i przejście tablicy stron znajdujących się w pamięci operacyjnej. Aby skrócić ten czas, stosowany jest mechanizm **TLB**, zwany pamięcią skojarzeniową, umożliwiającą znaczące skrócenie czasu translacji adresu dla najczęściej używanych stron. Istnieje możliwość zablokowania niektórych adresów translacji

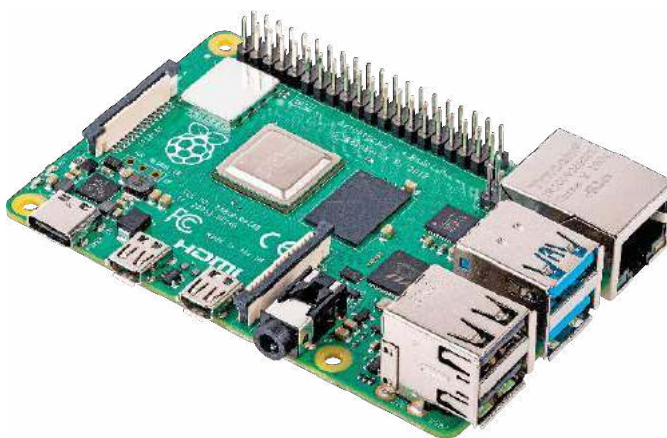


Rysunek 13. Uproszczony schemat wewnętrzny układu SoC

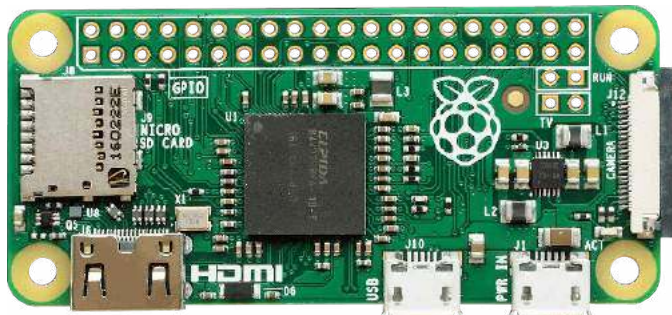
w TLB na stałe, co często stosuje się w systemach czasu rzeczywistego z miękkim reżimem czasowym. Aby połączyć moc obliczeniową układu SoC oraz szybką reakcję najczęściej integruje się w jednym układzie procesor aplikacyjny np. Cortex-A służący do obsługi zaawansowanych zadań oraz dodatkowy rdzeń Cortex-M, na którym pracuje system operacyjny czasu rzeczywistego.

Przykład typowego układu SoC z rdzeniem ARM pokazano na **rysunku 13**. We wnętrzu układu znajdują się klasyczne układy peryferyjne znane z mikrokontrolerów, jednak zwykle znacznie bardziej rozbudowane. Charakterystyczny jest brak wewnętrznych pamięci RAM oraz pamięci programu FLASH, które muszą być dołączone w postaci zewnętrznych układów. Z uwagi na to, że magistrale pamięci muszą pracować z dużą częstotliwością (zwłaszcza pamięć SDRAM), potrzebny jest skomplikowany projekt kilkuwarstwowej płytki PCB, co znacząco podwyższa koszty.

W ostatnim czasie na rynku mamy wysyp komputerów **SBC** (*Single Board Computer*), bazujących na układach SOC z rdzeniem ARM. Konstruktorzy bardzo chętnie sięgają po nie z uwagi na wydajność i niską cenę. Takie rozwiązanie zwalnia z potrzeby projektowania skomplikowanego systemu, a dostęp do dużej bazy otwartego oprogramowania pozwala szybko opracować bardzo zaawansowane konstrukcje, dzięki czemu w krótkim czasie powstaje gotowe urządzenie. Z tej właśnie przyczyny skupimy się na krótkim przeglądzie najpopularniejszych gotowych rozwiązań SBC, które możemy stosować w krótkich seriach prototypowych. Przegląd rozpoczniemy od doskonale



Fotografia 14. Komputer SBC Raspberry Pi 4



Fotografia 15. Komputer SBC Raspberry Pi Zero

znanego komputera **Raspberry Pi** (fotografia 14), który doczekał się już czwartej odsłony. Ma 4 GB pamięci operacyjnej, wyposażony jest w 4-rdzeniowy 64-bitowy procesor z wysoko wydajnymi rdzeniami **Cortex-A72**. Jako kanał komunikacyjny może posłużyć moduł Wi-Fi 2,4 GHz/5 GHz lub port Ethernet. Komputer dysponuje układem graficznym wspomagającym obsługę 3D czy dekodowanie strumieni wideo. Największą zaletą tego rozwiązania jest bardzo bogata dokumentacja techniczna oraz całe mnóstwo oprogramowania przy doskonałym wsparciu społeczności. Komputer znajduje zastosowanie przy opracowywaniu prototypów oraz do realizacji hobbyistycznych projektów. Z uwagi na umiejscowienie złączy oraz użyte komponenty nie jest przeznaczony do zastosowań przemysłowych. Dostępnych jest wiele akcesoriów oraz dodatkowych modułów przeznaczonych do współpracy z RPi, takich jak obudowy czy dodatkowe płytki dołączane do 40-pinowego złącza, które już pełni funkcję pewnego standardu.

Przy omawianiu Raspberry Pi warto wspomnieć również o skromniejszej wersji – **Raspberry Pi Zero** (fotografia 15). Ma jednordzeniowy układ SoC z nieco już leciwym rdzeniem **ARM11**. Wyposażony jest w 512 MB pamięci operacyjnej oraz procesor graficzny umożliwiający wyświetlanie obrazu. Jego największą zaletą jest cena wynosząca ok. 29 złotych brutto, co stawia pod znakiem zapytania sensowność używania większych mikrokontrolerów jednoukładowych, które niejednokrotnie kosztują więcej niż cały komputer. Należy jednak pamiętać, że do działania potrzebuje karty SD, co podnosi koszt rozwiązania. Istotną zaletą jest również złącze, które umożliwia wykorzystanie tego komputera jako bazy dla własnego urządzenia. Konstrukcja nie jest przeznaczona do zastosowania w warunkach przemysłowych.

Jeśli myślimy o poważniejszych zastosowaniach, dużo lepszym rozwiązaniem jest użycie płytki, która ma zintegrowaną pamięć **NAND Flash** lub **eMMC**. Dobrym wyborem wydają się płytki wyposażone w mikrokontroler **Sitara AM335x** firmy Texas Instruments.



Fotografia 16. Komputer SBC BeagleBone Black



Fotografia 17. Komputer SBC Nvidia Jetson Nano

Przykładem jest komputer **BeagleBone Black** (fotografia 16) taktowany z częstotliwością 1 GHz, wyposażony w 512 MB pamięci RAM oraz 4 GB pamięci Flash. Komputer ma szereg interfejsów komunikacyjnych: USB, HDMI oraz Ethernet. Ciekawym rozwiązaniem w cenie około 100 zł jest płytka NanoPi Neo 2, dysponująca 64-bitowym 4-rdzeniowym procesorem **Cortex-A53** taktowanym zegarem 1 GHz. Zawiera ona 512 MB pamięci operacyjnej DDR3, pamięć eMMC o wielkości 8 GB (na której możemy zainstalować system operacyjny oraz oprogramowanie) oraz gigabitowe złącze Ethernet i dwa porty USB. Jeżeli potrzebujemy bardzo dużej mocy obliczeniowej, np. do przetwarzania obrazów czy wirtualnej rzeczywistości, bardzo ciekawą propozycją jest płytka **Nvidia Jetson Nano** (fotografia 17) wyposażona w 4-rdzeniową jednostkę centralną z rdzeniami Cortex-A53 oraz procesor graficzny **Nvidia Maxell** ze 120 rdzeniami **CUDA**. Całość taktowana jest z częstotliwością 1,43 GHz.

Ostatnim rozwiązaniem zwykle stosowanym w najbardziej wymagających systemach wbudowanych jest zastosowanie układu programowalnego **FPGA** oraz zaimplementowanie procesora w postaci „SoftCore”. Jest to rozwiązanie stosunkowo rzadko stosowane w systemach profesjonalnych, głównie z uwagi na znaczne koszty, ponadto do pracy wymagają dodatkowego osprzętu takiego jak konfigurator oraz sekwensery zasilania. Sytuacja ulega zmianie dzięki takim układom jak **Lattice MachO2** czy **MAX10** firmy Altera, które integrują gotowy do działania układ FPGA w pojedynczej strukturze, co stanowi odpowiednik mikrokontrolera jednoukładowego w świecie układów programowalnych.

Układ SoC czy mikrokontroler jednoukładowy?

Konstruktor urządzeń wbudowanych dysponuje zarówno układami opartymi o mikrokontrolery jednoukładowe, jak i zaawansowanymi układami SoC. Dobór odpowiedniego rozwiązania przy szerokiej ofercie rynkowej nie jest łatwy.

Najprostsze urządzenia wbudowane, takie jak sterowniki pieca CO, sterowniki klimatyzacji czy termostaty zwykle budowane są w oparciu o mikrokontrolery, głównie ze względu na koszty oraz z uwagi na to, że do sterowania takimi procesami wystarczą nawet najprostsze rozwiązania. W bardziej zaawansowanych urządzeniach, gdzie potrzebna jest obsługa sieci, kolorowych wyświetlaczy LCD itp., wybór nie jest już taki oczywisty. Cena bardziej zaawansowanych mikrokontrolerów, potrafiących podjąć wyżej podany zadaniem, może nawet przewyższać ceny prostszych rozwiązań opartych o układy SoC. Dodatkową zaletą zastosowania układu SoC jest możliwość użycia systemu operacyjnego Linux i jego bogatych zasobów. W przypadku mikrokontrolera mamy do dyspozycji systemy operacyjne czasu rzeczywistego a większość oprogramowania musimy przygotować samodzielnie.

Innym kryterium wyboru jest określenie, czy urządzenie musi sterować procesami czasu rzeczywistego. W przypadku gdy nie ma takich wymagań lub wymagany jest jedynie system typu „soft real-time”, możemy zastosować jądro Linux z dodatkowymi łatkami czasu rzeczywistego. Natomiast gdy mamy do czynienia z systemem typu „hard realtime”, zmuszeni jesteśmy do zastosowania mikrokontrolera jednoukładowego oraz systemu czasu rzeczywistego. W przypadku bardziej zaawansowanych urządzeń czasu rzeczywistego możemy rozważyć scenariusz z dwoma układami, gdzie mikrokontroler steruje procesem czasu rzeczywistego, natomiast inne zadania, jak obsługa wyświetlacza graficznego, obsługa użytkownika czy komunikacja z systemem zdalnym, oddelegowana jest do układu SoC.

Oprogramowanie

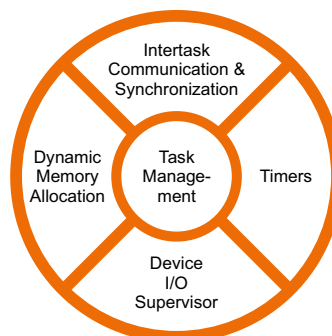
Istotną kwestią związaną z systemami wbudowanymi jest oprogramowanie, zarówno to działające w systemie docelowym, jak i wspomagające tworzenie oprogramowania dla tych systemów, takie jak: kompilatory, debuggery, zintegrowane środowiska programistyczne itp. W pierwszej kolejności omówimy oprogramowanie stosowane w systemach budowanych w oparciu o mikrokontrolery. W drugiej części omówimy oprogramowanie stosowane w rozwiązaniach z układami SOC, po czym w ostatniej części zajmiemy się oprogramowaniem wspomagającym.

Oprogramowanie dedykowane mikrokontrolerom jednoukładowym

Oprogramowanie dla mikrokontrolerów obecnie tworzymy głównie w językach ANSI – C oraz C++. W przypadku oprogramowania dla najmniejszych mikrokontrolerów 8-/16-bitowych z uwagi na mocno ograniczone zasoby stosowany jest niemal wyłącznie ANSI – C. Popularny niegdyś assembler znajduje zastosowanie głównie w postaci wstawek np. przy pisaniu systemów operacyjnych oraz w najmniejszych mikrokontrolerach, pozbawionych stosu sprzętowego lub dysponującego kilkudziesięcioma bajtami pamięci operacyjnej.

Jeśli system wbudowany zawiera bardziej zaawansowany mikrokontroler 32-bitowy z rdzeniem ARM, dysponujący większą pamięcią flash i RAM, możemy użyć języka C++ w najnowszym standardzie ISO/IEC 14882:2017 i jest to rozwiązanie zalecane. Umiejętne stosowanie języka C++ zwiększa bezpieczeństwo kodu oraz skraca czas przygotowania oprogramowania, nie podnosząc znacząco wymagań sprzętowych.

Oprogramowanie na najmniejsze systemy wbudowane tworzone jest bez użycia systemu operacyjnego. W takim przypadku program działa w trybie *foreground-background*, bazując na maszynach stanów oraz pętlach obsługi zdarzeń. Duża część kodu również wykonuje się bezpośrednio w procedurach obsługi przerwań. O ile napisanie tego typu oprogramowania dla najprostszych projektów nie nastrocza wielu problemów, tak w przypadku bardziej zaawansowanych systemów brak mechanizmów wielozadaniowości staje się problematyczny. W takich okolicznościach z pomocą przychodzą nam systemy operacyjne czasu rzeczywistego uruchamiane na małych mikrokontrolerach. Wymagania stawiane przed współczesnymi systemami, takie jak obsługa sieci, kolorowe wyświetlacze dotykowe, nowoczesne i skomplikowane w obsłudze magistrale typu USB, powodują, że napisanie tego typu oprogramowania bez systemu jest niewykonalne w rozsądnym przedziale czasowym. Najnowsze 32-bitowe układy wyposażone są w takie ilości pamięci operacyjnej, że uruchomienie systemu na układzie kosztującym kilka złotych nie stanowi wielkiego wyzwania. Na rynku istnieje wiele systemów operacyjnych przeznaczonych dla małych mikrokontrolerów, przy czym znakomitą większość stanowią rozwiązania otwartoźródłowe. W znakomitej większości przypadków systemy dla mikrokontrolerów nie są systemami operacyjnymi w pełnym tego słowa znaczeniu, zawierającymi mechanizmy sterowników urządzeń, zarządzanie pamięcią, zarządzanie użytkownikami czy obsługę sieci. Należy je



Rysunek 18. API typowego systemu operacyjnego dla małych mikrokontrolerów

traktować bardziej jako bibliotekę zapewniającą obsługę wątków niż pełnoprawny system operacyjny. Typowe API takich systemów pokazano na **rysunku 18**. Serce stanowi mechanizm obsługi wątków/zadań oraz mechanizm komunikacji i synchronizacji międzyprocesowej. Dodatkowe funkcje, jakie może pełnić system, to zarządzanie pamięcią czy sterowniki urządzeń, jednak nie są to elementy obowiązkowe. W przypadku układów pozbawionych jednostki MMU, sterowniki urządzeń nie są konieczne, ponieważ system oraz aplikacje pracują we wspólnej przestrzeni i mają dostęp do wszystkich zasobów, w tym do systemu przerwań oraz przestrzeni adresowej urządzeń peryferyjnych.

Jednym z najstarszych oraz powszechnie używanych systemów jest **FreeRTOS**, który możemy uruchomić na całej gamie mikrokontrolerów, począwszy od 8-bitowych AVR, skończywszy na 32-bitowych ARM. Został napisany w języku C i jest nastawiony głównie na prostotę oraz możliwość szybkiego przenoszenia na różne architektury. Obecnie może działać na ponad 30 różnych architekturach sprzętowych i w swojej podstawowej postaci zawiera obsługę mechanizmów takich jak tworzenie zadań, elementy synchronizacji i komunikacji międzyprocesowej oraz podstawową obsługę zarządzania pamięcią dynamiczną. Jego główną wadą jest to, że jest napisany w języku ANSI C z dużą liczbą makr i różnych współcześnie niezalecanych technik programistycznych.

Interesującym systemem jest **ChibiOS** dostępny na licencji GPL, przeznaczony dla mikrokontrolerów 8-/16-/32-bitowych. Jest dostępny na mniejszą liczbę architektur sprzętowych i w jego podstawowej wersji może zajmować od 1,2 kB do 5,5 kB pamięci FLASH. Zawiera bardziej rozbudowane mechanizmy synchronizacyjne a kod źródłowy jest dużo nowocześniejszy od wspomnianego wcześniej systemu. Oprócz samej biblioteki obsługi wątków jego opcjonalną część może stanowić mechanizm sterowników urządzeń oraz biblioteka graficzna **uGFX** przeznaczona do obsługi różnych wyświetlaczy graficznych.

Innym systemem doskonale znanym czytelnikom jest **ISIX** zapewniający podobną funkcjonalność do przedstawionego wcześniej systemu **ChibiOS**. Jego jądro zostało napisane w języku ANSI-C, natomiast pozostała część powstała z użyciem standardu **C++17**.

Jeżeli do dyspozycji mamy większą ilość pamięci, możemy użyć systemu czasu rzeczywistego **NuttX** przeznaczonego dla 8- i 32-bitowych mikrokontrolerów. Jego głównym założeniem jest zgodność ze standardem **ANSI** oraz **POSIX**, co umożliwia skompilowanie oraz uruchomienie dużej części bibliotek przeznaczonych np. dla systemu operacyjnego Linux. Jest jednym z najbardziej rozbudowanych systemów tego typu, ponieważ oprócz API standardu POSIX, zawiera podsystem sterowników urządzeń, wirtualny system plików i obsługę sieci. Obsługuje również Wi-Fi, NTPS, karty sieciowe, USB zarówno host, jak i device. Niestety duża zgodność ze wspomnianymi systemami pociąga za sobą stosunkowo duże jak na mikrokontrolery wymagania pamięciowe oraz sprzętowe.

Duża popularność technologii **IoT** spowodowała, że systemami dla mikrokontrolerów zainteresowały się duże koncerny, przykładem czego jest system **LiteOS**, którego autorem jest firma **Huawei**.

Dostępny jest na liberalnej licencji **BSD** i możemy go uruchomić na najważniejszych architekturach sprzętowych: ARM, x86, **RI-SC-V**. Charakteryzuje się bardzo lekką budową w podstawowej wersji zajmując mniej niż 10 kB pamięci FLASH. Jego głównym przeznaczeniem jest Internet Rzeczy, zatem szczególny nacisk położono na systemy łączności, krótki czas odpowiedzi oraz niski pobór energii. Obsługuje takie systemy łączności jak: Wi-Fi, Ethernet, **BLE**, **ZigBee** oraz wiele innych popularnych protokołów IoT. Systemem o podobnym przeznaczeniu jest **ZephyrOS** rozwijany pod skrzydłami Linux Foundation a w jego rozwój włączeni są m.in. Intel, NXP, Linaro czy Texas Instruments.

Kilkanaście lat temu jedynym oprogramowaniem dostarczanym przez producentów mikrokontrolerów jednoukładowych były zazwyczaj pliki nagłówkowe z opisem rejestrów. Reszta spoczywała na barkach programisty. Obecnie sytuacja wygląda inaczej, producenci prześcigają się nawzajem, dostarczając kompletne biblioteki, które za pomocą wywołania kilku funkcji umożliwiają wygodne uruchomienie układów peryferyjnych znajdujących się w mikrokontrolerze. Co prawda w starszych mikrokontrolerach układy peryferyjne były znacznie prostsze, wystarczyło zazwyczaj przestawić kilka bitów w rejestrach i układ był gotowy do pracy. Współczesne układy peryferyjne są coraz bardziej złożone i dysponują szeregiem zaawansowanych opcji, a ich obsługa jest dużo bardziej skomplikowana. Czas, jaki konstruktorzy mają na opracowanie urządzenia, uległ znacznemu skróceniu. W związku z tym koniecznością stało się dostarczanie bibliotek układów peryferyjnych, aby uprościć i przyspieszyć czas przygotowania projektu. Spójrzmy np. jak wygląda obsługa diody LED dołączonej do portu GPIO w przypadku 89C2051 i STM32F103 bezpośrednio z wykorzystaniem rejestrów (**listing 1**). Jak łatwo możemy zauważyć, kod sterujący diodą LED w przypadku mikrokontrolera 8051 jest bardzo prosty i wymaga w zasadzie jedynie zmiany stanu odpowiedniej linii portu. W przypadku mikrokontrolera STM32 uruchomienie portu GPIO w kierunku wyjścia wymaga wykonania szeregu czynności, takich jak włączenie sygnałów zegarowych oraz konfiguracji portu w kierunku wyjścia. Ponadto w kodzie pominięto inicjalizację i konfigurację początkową, której wymaga STM32, a w przypadku 8051 po włączeniu zasilania układ jest od razu gotowy do pracy. Oczywiście im bardziej wyszukana funkcja pełni układ peryferyjny, tym jego obsługa jest bardziej skomplikowana. Dziś na szczęście większość znaczących producentów układów dostarcza gotowe biblioteki, tak więc nie będziemy musieli operować na rejestrach jak jeszcze kilkanaście lat temu. W przypadku mikrokontrolerów STM32 jest to biblioteka **STM32Cube** integrująca się z oprogramowaniem narzędziowym dostarczoną przez producenta. Zupełnie oddzielną kwestią jest jakość kodów źródłowych dostarczanych przez producentów, która często bywa kiepska.

Oprogramowanie przeznaczone dla rozwiązań SoC/SBC

Układy SoC dzięki zastosowaniu jednostki centralnej, będącej pełnoprawnym mikroprocesorem z obsługą pamięci wirtualnej, otwierają dostęp do bogatej bazy oprogramowania przeznaczonego dla standardowych systemów komputerowych. Możliwość uruchomienia pełnoprawnego systemu operacyjnego ogólnego przeznaczenia daje dostęp do bogatej bazy oprogramowania oraz bibliotek, które rozwinęły się w ciągu ostatnich 10 lat. W dobie coraz większych wymagań stawianych systemom wbudowanym, takich jak obsługa wyświetlacza graficznego czy skomplikowanego interfejsu, jedyną możliwością przygotowania oprogramowania w krótkim czasie jest użycie pełnoprawnego systemu wraz z szeroką bazą oprogramowania. Wtedy możliwe jest skupienie się na przygotowaniu głównej funkcjonalności urządzenia bez potrzeby wnikania w niskopoziomowe szczegóły. Znacząco obniżamy koszt opracowania oraz skracamy czas dostarczenia urządzenia do klienta. Szczególną popularność zyskują tutaj systemy operacyjne rozwijane w oparciu o model otwartego oprogramowania. Rozwijane są przez społeczność

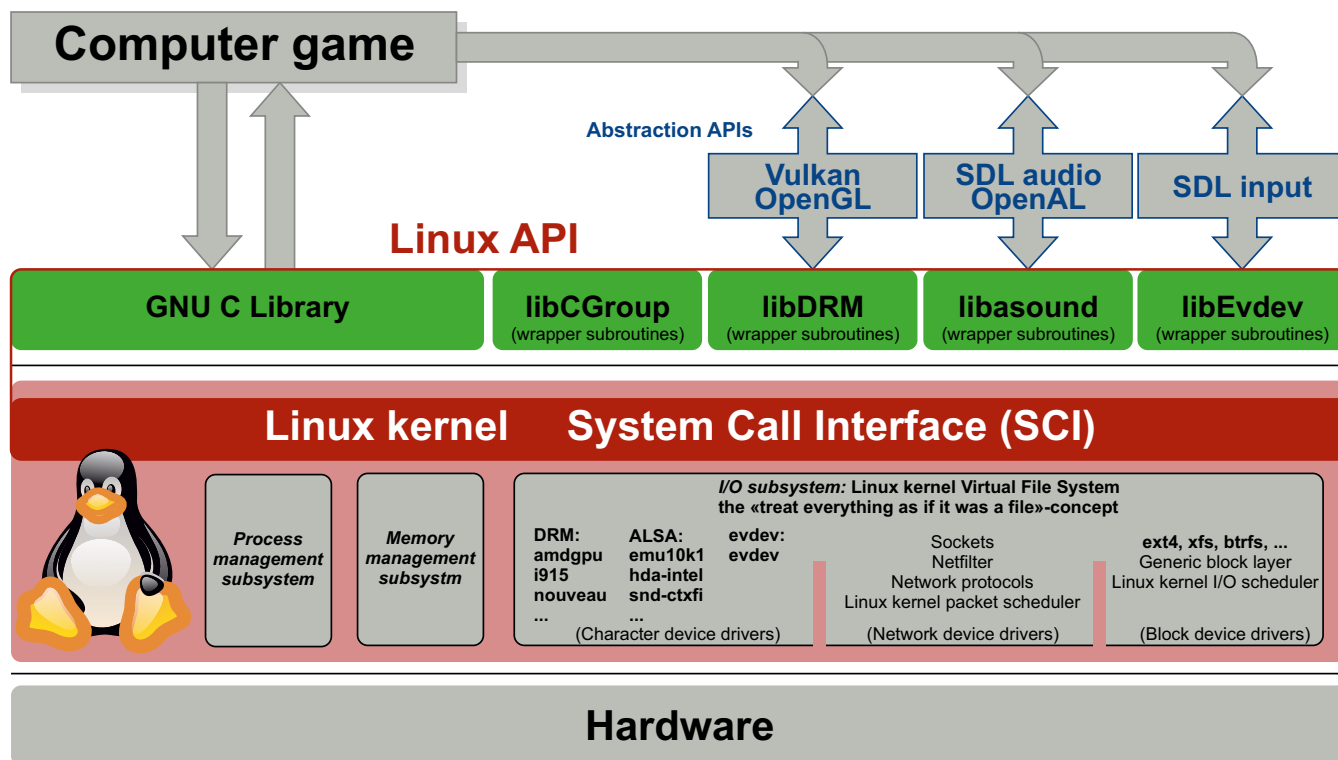
```
Listing 1. Program do sterowania diodą LED dołączoną do portu
GPIO w przypadku 89C2051 i STM32F103
//89C2051 (SDCC)
int main(void){
    for(;;) {
        P2_1 = 1;
        for(int i=0; i<10000; ++i);
        P2_1 = 0;
        for(int i=0; i<10000; ++i);
    }
}

//STM32F103 (GCC)
#include <stm32f30x.h>
int main(void){
    RCC -> AHBENR |= 1 << 21;
    GPIOE -> MODER |= 1 << 26;
    GPIOE -> ODR |= 1 << 13;
    while(1){
        for (int i = 0; i < 1000000; ++i) ;
        GPIOE -> ODR |= 1 << 13;
        for (int i = 0; i < 1000000; ++i) ;
        GPIOE -> ODR &= ~(1 << 13);
    }
}
```

(głównie duże firmy), co zapewnia znaczące oszczędności zarówno czasu, jak i pieniędzy potrzebnych do rozwoju systemu.

Najpopularniejszym systemem operacyjnym mającym zastosowanie w systemach wbudowanych jest Linux, który działa praktycznie na wszystkich obecnie istniejących układach SoC. Jądro Linux jest w świecie urządzeń wbudowanych standardem, więc każdy znaczący producent układu zapewnia wsparcie dla tego systemu. Używając słowa Linux, mamy na myśli samo jądro systemu operacyjnego. Natomiast cały system oprócz samego jądra stanowią biblioteki oraz aplikacje. Budowę typowego stosu programowego systemu Linux pokazano na **rysunku 19**. Na najniższym poziomie mamy jądro systemu, które jest odpowiedzialne za zarządzanie pamięcią, zarządzanie procesami oraz obsługę urządzeń wejścia/wyjścia za pośrednictwem sterowników urządzeń. Jądro dostarcza standardowe API wywołań systemowych, jednak najczęściej aplikacje użytkownika nie odwołują się bezpośrednio do niego, tylko korzystają z bibliotek pośrednich. Najważniejszą biblioteką zapewniającą podstawowe API dla aplikacji jest **libc**. Biblioteka zapewnia aplikacjom standardowy interfejs wymagany przez aplikacje pisane w języku C oraz dostęp do najbardziej niskopoziomowych funkcji związanych z systemem, takich jak: zarządzanie procesami, zarządzanie pamięcią, komunikacja międzyprocesowa itp. Kolejnymi istotnymi bibliotekami systemowymi są: **libDRM**, odpowiedzialna za obsługę systemu graficznego, **libasound** odpowiedzialna za obsługę dźwięku oraz **libevdev**, odpowiedzialna za obsługę urządzeń wejściowych, takich jak myszki, klawiatury, ekrany dotykowe itp. Oczywiście takich bibliotek jest znacznie więcej. Z punktu widzenia systemów wbudowanych istotną biblioteką jest **libusb**, umożliwiająca tworzenie oprogramowania dla urządzeń USB w postaci aplikacji, bez konieczności bezpośredniego tworzenia sterowników pracujących w przestrzeni jądra.

Istnieje wiele odmian systemu Linux zwanych dystrybucjami przygotowanymi przez różnych producentów oraz społeczność. O ile przygotowanie systemu dla komputerów PC nie stanowi problemu, z uwagi na to, że architektura PC jest ściśle określona, o tyle w przypadku systemów wbudowanych, które występują w wielu odmianach, przygotowanie dystrybucji jest znacznie bardziej skomplikowane. Po pierwsze mamy wiele architektur i konfiguracji sprzętowych, a po drugie przed systemami wbudowanymi stawiamy zupełnie różne wymagania. Oprogramowanie routera będzie wymagać zupełnie innego zestawu bibliotek i aplikacji niż oprogramowanie służące do sterowania maszyną CNC. Istnieją co prawda generyczne dystrybucje przeznaczone dla systemów wbudowanych, jednak najczęściej używane są jedynie na etapie prototypowania lub amatorskich projektów. Największym problemem z gotowymi dystrybucjami jest to, że zawierają one całe mnóstwo oprogramowania niekoniecznie dostosowanego i zoptymalizowanego dla danego rozwiązania. Tworząc własną dystrybucję od podstaw, możemy uszyć system na miarę dostosowany do specyfiki projektu. Najpopularniejszymi gotowymi dystrybucjami



Rysunek 19. Linux architektura systemu

przeznaczonymi dla architektury ARM są: **Debian**, **Ubuntu for the Internet of Things**, **Arch Linux ARM**. Są to dystrybucje zbliżone do tych znanych z komputerów PC, przy czym musimy pamiętać, że instalacja na komputerku SBC bywa bardziej skomplikowana. Korzystając z gotowej dystrybucji, musimy się liczyć z tym, że często będzie konieczne dostarczenie własnego *bootloadera* czy jądra z opcjami specyficznymi dla własnej płytki SBC.

Jeśli planujemy dostarczyć urządzenie, które nie jest prototypem, dużo lepszym rozwiązaniem jest zbudowanie systemu Linux ze źródeł dostosowanych do naszego urządzenia. Przygotowując obraz dysku w ten sposób, mamy kontrolę nad całym procesem kompilacji włącznie z dedykowaną optymalizacją dla naszej architektury oraz dostosowaniem poszczególnych komponentów dokładnie do naszych wymagań. Takie podejście pozwala na zbudowanie bezpiecznego i wysoce zoptymalizowanego systemu przeznaczonego do konkretnych potrzeb.

Kompilacja całego systemu krok po kroku jest zadaniem dosyć skomplikowanym i czasochłonnym, co już dawno temu zostało dostrzeżone przez społeczność, która przygotowała odpowiednie narzędzia automatyzujące przygotowanie własnej dystrybucji. Obecnie istnieją dwa konkurencyjne rozwiązania, prostszy **Buildroot** oraz dużo bardziej wyszukany **Yocto Project**.

Buildroot jest zbiorem zestawu plików **Makefile** oraz zestawem łatek automatyzujących proces przygotowania obrazu systemu Linux. Umożliwia wybór poszczególnych komponentów systemu oraz opcji konfiguracyjnych za pomocą menu KCONFIG. Do prawidłowego działania potrzebujemy jedynie narzędzia **make** oraz kompilatora potrafiącego zbudować kod dla architektury hostującej. Pozostałe elementy włącznie z kompilatorem skrośnym na docelową architekturę są kompilowane bezpośrednio z kodów źródłowych. Buildroot wspiera wiele architektur sprzętowych: x86, ARM, MIPS, PPC. Pozwala również na wybór jednej z kilku dostępnych bibliotek języka C. Umożliwia przygotowanie obrazu do wgrania na płytkę docelową oraz budowanie poszczególnych komponentów z wykorzystaniem menedżera pakietów co może być przydatne, jeśli planujemy zapewnić możliwość zdalnej aktualizacji oprogramowania.

Obecnie najbardziej zalecanym narzędziem do budowy własnych dystrybucji jest **Yocto Project** wspierane i rozwijane przez Linux Foundation. Przeznaczone jest do tworzenia własnych dystrybucji Linuksa

dla urządzeń IoT i wspiera ponad 22 różne architektury sprzętowe. Bazuje na podejściu obiektowym oraz warstwach oprogramowania, które są na siebie nakładane. Jego największą zaletą jest całe mnóstwo plików konfiguracyjnych opisujących sposób budowy oraz kompilacji poszczególnych bibliotek, programów czy innych komponentów. Pozwala to na przygotowanie własnej dystrybucji Linuksa dostosowanej do urządzenia i zawierającej jedynie niezbędne pakiety.

Oprócz systemu Linux, który króluje w systemach wbudowanych (podobnie jak Windows na desktopach), pewną popularność mają również inne systemy operacyjne przeznaczone do tego celu. Jednym z takich systemów jest **Android Things** zaprezentowany przez firmę Google w 2015 roku. Jego głównym przeznaczeniem są urządzenia wbudowane o niewielkiej ilości pamięci RAM rzędu 32 MB. System wspiera komunikację z wykorzystaniem protokołów BT oraz Wi-Fi. Podobnie jak w klasycznym Androidzie zawiera on maszynę wirtualną **JVM**, a więc możemy go wygodnie programować w języku **Java**. Do dyspozycji mamy bogaty zestaw bibliotek w tym obsługę tak niskopoziomowych interfejsów jak: I²C, SPI czy obsługę portów GPIO bezpośrednio z poziomu Javy.

Innym systemem mającym pewną popularność, jest **Windows 10 IoT** firmy Microsoft. Jest systemem płatnym z zamkniętym kodem źródłowym przeznaczonym dla urządzeń dysponujących relatywnie niewielką ilością pamięci RAM. Bazuje na jądrze **NT** i umożliwia pisanie aplikacji w technologii **.NET**. Umożliwia również korzystanie z **API Win32** znanego z pełnej wersji systemu Windows. Może pracować zarówno na procesorach o architekturze x86, jak i ARM, jednak z uwagi na to, że konieczne jest wnoszenie opłat licencyjnych, nie jest on zbyt popularny.

Oprogramowanie wspomagające tworzenie oprogramowania dla systemów wbudowanych

Odpowiednie narzędzia programistyczne pozwalają znacząco przyspieszyć projektowanie oprogramowania oraz skrócić czas wprowadzenia produktu na rynek. Na początku rozwoju systemów wbudowanych, gdy królowały układy 8-bitowe, oprogramowanie najczęściej było pisanie w assemblerze. W późniejszym czasie na rynku zaczęły pojawiać się kompilatory języka C, pozwalające znacznie ułatwić przygotowanie oprogramowania. W międzyczasie okazało się,

że wczesne architektury, takie jak 8051 czy PIC, nie do końca były dostosowane do wykonywania kodu wygenerowanego za pomocą kompilatora, chociażby z powodu braku adresowalnego stosu (PIC) czy różnych sposobów adresowania pamięci (8051). Producenci mikrokontrolerów dostrzegli ten problem i zaczęli opracowywać lub modyfikować rdzenie mikrokontrolerów, tak aby lepiej współpracowały z kompilatorami języka C. W tym samym czasie na rynku pojawiły się zintegrowane środowiska programistyczne, które oprócz samego kompilatora zawierały zintegrowany edytor, debugger czy kreator projektów umożliwiające przygotowanie bazy nowego projektu za pomocą kilku kliknięć. Na początku były to rozwiązania komercyjne, które kosztowały bardzo duże pieniądze i były poza zasięgiem małych firm. Z czasem społeczność zaczęła rozwijać kompilator GCC dla architektury AVR oraz ARM, równolegle dla architektury PIC i 8051 rozwijany był kompilator SDCC. Wczesnym wersjom kompilatorów z tamtych czasów dużo brakowało do rozwiązań komercyjnych zwłaszcza pod kątem wielkości i szybkości otrzymywanego kodu maszynowego. Producenci dostrzegli, że narzędzia oferowane po wysokich cenach nie są dobrym rozwiązaniem, ponieważ znacząco zmniejszają sprzedaż. W związku z powyższym firmy zaczęły się interesować rozwojem kompilatorów na darmowych licencjach. W międzyczasie na rynku pojawiło się również środowisko **Eclipse** umożliwiające tworzenie własnego IDE. Bazując na Eclipse oraz GCC, producenci zaczęli dostarczać darmowe zintegrowane środowiska programistyczne. Obecnie każdy szanujący się producent mikrokontrolerów dostarcza tego typu środowiska nieodpłatnie. Pomimo istnienia środowisk IDE, zdaniem autora nadal lepszym rozwiązaniem jest przygotowanie i tworzenie oprogramowania dla mikrokontrolerów, bazując bezpośrednio na narzędziach takich jak **make** czy **waf**, ponieważ do prawidłowego zbudowania projektu potrzebny będzie jedynie kompilator oraz mamy możliwość zbudowania całego projektu za pomocą automatycznych skryptów. Istotnym aspektem jest to, że mamy pełną kontrolę nad projektem i jesteśmy zupełnie niezależni od producenta.

Jeśli budujemy aplikację dla mikrokontrolerów jednocukłowych, jedyną potrzebną rzeczą do napisania i uruchamiania oprogramowania będzie zintegrowane środowisko IDE oraz ewentualnie interfejs **JTAG**. W przypadku JTAG sytuacja wygląda podobnie jak w przypadku kompilatorów i środowisk IDE, gdzie kiedyś narzędzia

te kosztowały tysiące dolarów a obecnie możemy je nabyć w cenie nieprzekraczającej kilkudziesięciu złotych.

W przypadku oprogramowania dla systemów opartych o układy SoC z systemem Linux sytuacja wygląda inaczej i zależy od tego, czy będziemy zajmować się oprogramowaniem bootloadera, sterownikami dla jądra systemu Linux, czy pisaniem oprogramowania w przestrzeni użytkownika. Przy tym w tekście skupimy się głównie na ostatnim przypadku, ponieważ większość układów SoC ma wsparcie dla bootloadera oraz jądra systemu Linux dostarczone przez producenta. Zestaw oprogramowania potrzebny do tworzenia aplikacji dla przestrzeni użytkownika niczym nie różni się od zestawu przeznaczonego do tworzenia oprogramowania dla komputera PC. Potrzebować będziemy: kompilatora skrośnego, debugera gdb pracującego na komputerze PC oraz gdb serwera pracującego na płycie docelowej. Wszystkie te narzędzia są automatycznie budowane, jeśli do tworzenia własnej dystrybucji wykorzystujemy narzędzia Yocto lub Buildroot. Dodatkowo, jeśli jesteśmy przyzwyczajeni do zintegrowanych środowisk graficznych, możemy skorzystać z darmowych edytorów, takich jak „Eclipse” lub „Visual Studio Code”. Pisanie oprogramowania w Linuksie jest zdecydowanie bardziej wygodne niż tworzenie aplikacji dla mikrokontrolerów, ponieważ do dyspozycji mamy dużo więcej pamięci oraz jednostkę MMU, która w przypadku rażących błędów aplikacji zapobiega awarii całego systemu.

Podsumowanie

Na przestrzeni lat zmieniały się systemy wbudowane oraz ich zastosowania. Dawniej można było spotkać je jedynie w najbardziej zaawansowanych urządzeniach a narzędzia przeznaczone do tworzenia oprogramowania były bardzo drogie. Z czasem stawały się coraz bardziej dostępne a ich cena spadała poniżej kosztu rozwiązań realizowanych bez udziału mikroprocesorów. Do wzrostu popularności przyczyniła się również dostępność darmowych narzędzi. Aktualnie wraz z rozwojem interfejsów graficznych coraz bardziej zauważalny jest trend odchodzenia od mikrokontrolerów jednokukłowych na rzecz systemów zbudowanych w oparciu o układy SoC. Myślę, że z czasem, jeśli producentom uda się zintegrować w jednym układzie SoC pamięć SDRAM, rozwiązania tego typu będą wypierały mikrokontrolery jednokukłowe.

Lucjan Bryndza, EP
Lucjan.bryndza@boff.pl

REKLAMA

nowa strona EP już wkrótce
zeglądaj na www.ep.com.pl
poczuj zmianę

The screenshot displays the homepage of 'ELEKTRONIKA PRAKTYCZNA'. At the top, there's a navigation bar with links like 'MENU', 'Kontakt', 'Forum', 'Newsletter', and 'Sklep AVT'. Below this, a banner for 'Informator Ryzykowny Automatyki IRA 2020' is visible. The main content area features several sections: 'EP W TOKU' with articles like 'Dwudziestowiec dsPIC33CH w praktyce', 'Środowisko projektowe STM32CubeIDE', and 'Płytki drukowane - wybór narzędzi projektowych, technologii wykonania...'; 'POLECAMY W NAJNOWSZYM NUMERZE: Kwiecień 2019' with articles like 'Aktualne standardy komunikacji radiowej' and 'Niezwodna sieć bezprzewodowa Wi-Fi bazująca na urządzeniach Moxa'; and 'SZCZEGÓLNI POLECAMY' featuring an image of a person working on a circuit board. The website has a clean, professional layout with a mix of text and images.



Nowe przekaźniki firmy Relpol

Więcej informacji:

RELPOŁ S.A.
68-200 Żary, ul. 11 Listopada 37
tel. 684790800, faks 683743866
sprzedaz@relpol.com.pl
www.relpol.pl

W ofercie firmy Relpol, krajowego producenta przekaźników, pojawiły się nowe przekaźniki czasowe oraz przekaźnik instalacyjny o wysokiej odporności na prąd udarowy. Wersje czasowe pozwalają na natychmiastowe uruchomienie wybranej funkcji bez konieczności ponownego włączenia zasilania. Są to RPC-1MC-UNI i RPC-1MD-UNI zawierające 1 zestyk przełączny oraz RPC-3MD-UNI wyposażony w 3 zestyki przełączne, umożliwiające sterowanie aż trzema niezależnymi urządzeniami w instalacjach niskiego napięcia. Zapraszamy do zapoznania się z ich właściwościami.

RPC-1MC-UNI

Przekaźniki czasowe RPC-1MC-UNI realizują aż 14 możliwych funkcji, a ich zmiana odbywa się bez konieczności ponownego włączenia przekaźnika. Niektóre z nich zostały rozszerzone o funkcję Stop (S) i funkcję Reset (R) do wyboru w zależności od wymagań procesu i aplikacji.

Dostępne funkcje Stop:

- E(S) – opóźnione załączenie z zatrzymaniem odmierzania czasu zestykiem S (Stop),

- Wu(S) – załączanie na nastawiony czas z zatrzymaniem odmierzania czasu przez zamknięcie zestyku S (Stop),
- Bi(S) – symetryczna praca cykliczna rozpoczynająca się od zadziałania z zatrzymaniem odmierzania czasu T na czas załączenia zestyku S (Stop),
- Bp(S) – symetryczna praca cykliczna rozpoczynająca się od przerwy z zatrzymaniem odmierzania czasu T na czas załączenia zestyku S (Stop).

W każdej z nich, gdy zestyk sterujący S zostanie zamknięty, wstrzymane zostaje odmierzanie czasu aż do momentu, gdy zestyk sterujący zostanie otwarty. Otwarcie zestyku S uruchamia dalszą realizację funkcji. Jeżeli zestyk sterujący S zostanie zamknięty podczas realizacji funkcji, to działanie zostanie wstrzymane. Po otwarciu zestyku S funkcja realizowana jest od nowa. Funkcje Reset to z kolei:

- E(R) – opóźnione załączenie z funkcją Reset,



Fot. 1. RPC-1MC-UNI

- Wu(R) – załączenie na nastawiony czas z funkcją Reset,
- Esa(R) – opóźnione załączenie i wyłączenie sterowane załączeniem i wyłączeniem zestyku S z funkcją Reset.

RPC-1MD-UNI, RPC-3MD-UNI

Przełączniki czasowe RPC-1MD-UNI, RPC-3MD-UNI to rozszerzenie serii RPC, będącej zamiennikiem serii MT-T. Dostępne są wykonania z 1 i 3 zestykami przełącznymi. Realizują one 10 możliwych funkcji, a ich zmiana odbywa się bez konieczności ponownego włączenia przełącznika. Są to:

- E – opóźnione załączenie,
- Wu – załączenie na nastawiony czas,
- Bp – symetryczna praca cykliczna rozpoczynająca się od przerwy,
- Bi – symetryczna praca cykliczna rozpoczynająca się od załączenia,
- R – opóźnione wyłączenie sterowane zestykiem S,
- Ws – jednokrotne załączenie na nastawiony czas, wyzwalane zamknięciem zestyku sterującego S,
- Wa – załączenie na nastawiony czas, wyzwalane otwarciem zestyku sterującego S,
- Esa – opóźnione załączenie i wyłączenie sterowane zestykiem S,
- B – praca cykliczna sterowana zamykaniem zestyku sterującego S,
- T – generowanie impulsu 0,5 s po upływie czasu T.

RPI-1ZI-U24A

RPI-1ZI-U24A to nowy przełącznik instalacyjny, który wytrzymuje maksymalny prąd załączania 120 A w czasie 20 ms. Jest on przeznaczony do załączania obwodów o dużym prądzie początkowym (udarowym), w szczególności do obwodów oświetleniowych. Dobrze sprawdza się w pracy z małymi, jednofazowymi silnikami do maksymalnej mocy 650 W. Maksymalna moc łączeniowa przy obciążeniu lampami wynosi:

- dla jarzeńówek 800 W,
- dla halogenów 2500 W,
- dla oświetlenia LED
- – 500 W.

Przełącznik jest przeznaczony do pracy w systemach automatyki budynkowej do załączania obwodów oświetleniowych we współpracy z zegarami sterującymi, wyłącznikami, przyciskami sterowniczymi. Sprawdza się też w automatyce przemysłowej, w tym może mieć szerokie zastosowanie w rozdzielniach aparatury modułowej, szczególnie do załączania obwodów o wysokim prądzie początkowym.

Relpol
www.repol.pl



Fot. 2. RPC-1MD-UNI



Fot. 3. RPC-3MD-UNI



Fot. 4. RPI-1ZI-U24A

Nowa seria

Przełączniki nadzorcze RPN

nadzór
temperatury

nadzór
napięcia



nadzór
prądu

repol® S.A.

11 różnych wykonan RPN – sprawdź

Zapraszamy

Energetics – hala C, stoisko 34
Warsaw Industry Week – hala E, stoisko 209

Biuro Obsługi Klienta
tel. 68 47 90 822, 850, sprzedaz@repol.com.pl
www.repol.com.pl



Dioda elektroluminescencyjna czy fotodioda? A może jedno i drugie...?

Diody to bardzo różnorodna rodzina elementów elektronicznych. Mimo tej różnorodności, pewne zjawiska fizyczne, które stoją u podstaw ich działania, są wspólne dla wszystkich tych elementów półprzewodnikowych. Jednym z nich jest interakcja półprzewodnika ze światłem.

Wszyscy elektronicy znają różne rodzaje diod półprzewodnikowych. W zasilaczach korzystamy z diod prostowniczych, diod Schottky'ego, czasami także innych, takich jak transile. W systemach pomiarowych często, jako detektor lub miernik światła, znaleźć możemy fotodiodę. Oczywiście każdy zna też diody elektroluminescencyjne – popularne LED-y; często używane są one jako kontrolki w urządzeniach elektronicznych, a od kilku lat także jako źródła światła w naszych domach, gdzie wyparły żarówki. Z uwagi na tak ogromną różnorodność tych elementów często zapomina się, że tak naprawdę każda z nich to niemalże taka sama struktura półprzewodnikowa. Drobne detale powodują, że różnią się parametrami

czy wykazują unikatowe własności, jednakże u podstaw ich działania stoją te same fizyczne zasady.

Każda dioda półprzewodnikowa, jak każdy element półprzewodnikowy, jest czuła na światło. Jest to dosyć nieoczywiste zjawisko, jednakże – jak czytamy w RAQ numer 45 [1] – może ono sporo napuścić w czułych układach analogowych. Przekonamy się, że zjawisko to może również zostać wykorzystane w ciekawych aplikacjach diod elektroluminescencyjnych.

Zadziwiające źródło przydźwięku sieciowego

Do Jamesa Bryanta, inżyniera aplikacyjnego pracującego w Analog Devices, zwrócił się projektant z problemem przydźwięku sieciowego w swoim układzie. Jest to typowy problem, pojawiający się w układach analogowych, gdy do sygnału przenika składowa sieci, bądź jej harmoniczne po wyprostowaniu (w Europie będzie to, odpowiednio, 50 Hz i 100 Hz; w Ameryce z kolei 60 Hz i 120 Hz).

W sytuacji gdy liniowy zasilacz sieciowy nie filtruje częstotliwości sieci od napięcia wyjściowego, składowa ta przenika do sygnału (z jaką amplitudą, to zależne jest od parametru PSRR – odrzucenie



James Bryant pracował w firmie Analog Devices jako menedżer ds. aplikacji na Europę od 1982 roku do przejścia na emeryturę w 2019 roku. Mimo odejścia ze struktur AD, nadal pisze i konsultuje dla firmy, tworząc ciekawe treści, takie jak te, które wykorzystano do napisania tego artykułu. Oprócz miłości do elektroniki jest on także krótkofalowcem, jego znak wywoławczy to G4CLF.

wpływu zasilania – poszczególnych elementów aktywnych w torze sygnału). Zazwyczaj problemy te da się rozwiązać, zwiększając pojemność filtrującą na wyjściu prostownika lub dodając stopień stabilizacji napięcia przed czułymi układami. Zjawisko to w zasadzie nie występuje w zasilaczach impulsowych – tam problemem są zakłócenia o wyższej częstotliwości, jednakże to zjawisko nie ma nic wspólnego z diodami i jest poza zakresem tego artykułu.

Innym niż zasilacz liniowy źródłem przydźwięku sieciowego jest sprzęganie się częstotliwości sieci z liniami sygnałowymi urządzenia lub poprzez masę (mitologizowane „pętle masy”, których obawiają się elektronicy). W obu tych sytuacjach zakłócenia o niskiej częstotliwości przenikać mogą do układu, gdzie najczęściej są wzmacniane wraz z sygnałem użytecznym i dostają się na jego wyjście. Tego problemu można uniknąć, dodając do wejść urządzenia filtry dolnozaporowe, które odcinają zakłócenia o niskiej częstotliwości od użytecznego sygnału.

W opisywanej w RAQ #45 [1] sytuacji ani filtrowanie zasilania, ani nawet zasilanie z baterii ze stabilizatorem liniowym nie pomogło w eliminacji przydźwięku sieciowego. Dodatkowo wspomniany projektant przetestował swój zasilacz – zamiast układu dołączony

zostaje opornik obciążający go, tak aby pobór prądu był podobny do tego, jaki pobiera urządzenie. W takim układzie można bezproblemowo przeanalizować napięcie zasilające np. oscyloskopem, aby spróbować znaleźć źródło zakłóceń, przenikających do układu. Jeśli powyższe sposoby zawiodły, trzeba nad problemem pochylić się bardziej i zlokalizować miejsce w układzie, przez które zakłócenia sieciowe przenikają do sygnału analogowego. Jak wskazuje James Bryant w swoim artykule, **zadziwiająco częstym miejscem przenikania zakłóceń są diody krzemowe w szklanych obudowach**. Elementy takie często stosuje się np. jako zabezpieczenie wejść układu.

Mechanizm przenikania zakłóceń przez diody

Każda dioda półprzewodnikowa będzie się zachowywała jak fotodioda, jeśli zostanie wystawiona na światło. Taka (foto)dioda działa najczęściej w trybie fotoprzewodzenia. Efekt ujawnia się w momencie, gdy dioda spolaryzowana jest zaporowo – światło zmienia prąd wsteczny (prąd upływu) diody. Fotony o odpowiedniej energii, padające na strukturę półprzewodnikową, powodują wzbudzenie w niej nośników. Od ich koncentracji zależy m.in. prąd wsteczny. W przypadku krzemu energia ta musi być wyższa niż 1,14 eV (co odpowiada światłu o długości fali równej lub mniejszej niż 1087 nm). Co oznacza, że całe spektrum światła widzialnego może oddziaływać na taki element. Gdy dioda znajduje się w ciemności, przez diodę spolaryzowaną zaporowo płynie tylko tzw. prąd ciemny, gdy zostaje ona oświetlona, prąd ten wzrasta proporcjonalnie do natężenia padającego światła. Jeżeli światło, które oświetla strukturę półprzewodnikową, jest modulowane to i prąd wsteczny diody zostanie zmodulowany. W większości warsztatów elektronicznych o źródło modulowanego światła nie jest trudno – często znajduje się ono nad głowami elektroników;

REKLAMA

ADELS
contact



AC 166

- system złącz i przewodów do szybkiej budowy instalacji
- do zastosowania w biurach, handlu i wystawiennictwie

AC 164

- „mniejszy brat” inspirowany AC 166®
- 16 A w znacznie mniejszych wymiarach
- idealny do zastosowań w ograniczonej przestrzeni
- podłączenie przewodów od 0,5 do 1,5 mm²

*LCS 75 – NOWOŚĆ

- ochrona IP66 – bezpieczeństwo pierwszej klasy
- prąd znamionowy 16 A przy napięciu 450 V
- przycisk zwalniający do szybkiego podłączenia przewodów elastycznych
- przewody zewnętrzne: 0,5 do 2,5 mm² (druć lub linka)
- montaż w standardowych otworach M20

AC 162 LED

- złącza i przewody do systemów niskonapięciowych

SMDflat345/SMDflat545

- złącza do listew LED
- SMDflat345 – wysokość tylko 3,45 mm, do 6 A
- SMDflat545 – do 2,5 mm², do 16 A

LK980

- rodzina złącz do opraw oświetleniowych
- połączenie samozaciskające
- dźwignia zwalniająca



www.contrans.pl



wejdź i kup on-line

CONTRANS TI Sp. z o.o. ul. Polanowicka 66, 51-180 Wrocław

✉ adels@contrans.pl ☎ +48 71 325 26 21...24 📠 +48 71 325 44 39

CONTRANS TI

są to świetlówki – mrugają one z podwojoną częstotliwością sieci – 100 Hz. W ten sposób modulacja sieciowa przenika do układu.

Dioda LED to też fotodioda

W RAQ numer 108 [2] ten sam James Bryant wskazuje na bardzo ciekawe wykorzystanie diod elektroluminescencyjnych – jako detektorów. Pozwala to wykorzystać opisane powyżej zjawisko w korzystny dla nas sposób. **Typowe diody półprzewodnikowe, takie jak klasyczna 1N4184, są dostatecznie czułe na światło, by wprowadzić przydźwięk sieciowy do naszego sygnału, ale nie na tyle, by pracować jako fotokomórka.** Dodatkowo najczulsze są w podczerwieni, dlatego też najczęściej przydźwięku w systemy elektroniczne w takim przypadku wprowadzają klasyczne żarówki a następnie świetlówki. Diody LED niemalże nie, z uwagi na fakt, że mają dosyć ograniczone pasmo. Jak czytamy w RAQ #108 [2], bezpośrednie światło słoneczne generuje w tych diodach fotoprąd na poziomie 10 nA. Latarka z klasyczną żarówką około 20...30 nA. Z kolei 60 W żarówka, świecąca, powoduje modulację fotoprądu na poziomie około 7% przy 100 Hz.

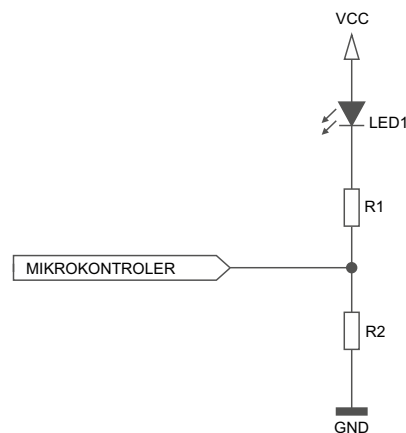
Diody elektroluminescencyjne (LED), jakkolwiek kosztujące więcej – ok. pięć razy tyle, co zwykłe diody krzemowe w szklanych obudowach – charakteryzują się o wiele większą czułością. Na przykład czerwona dioda LED 5 mm (1000 mCd przy 20 mA) generuje w pełnym świetle słonecznym fotoprąd na poziomie 20 μ A – w słonecznych warunkach, np. w tropikach, to wystarczająca czułość, by utrzymać przy życiu na przykład zegar czasu rzeczywistego. Być może nie sprawdzają się one jako zamienniki systemów fotowoltaicznych, ale LED-y są nawet do dziesięciu razy tańsze niż fotodiody, zaprojektowane dokładnie do pracy jako detektory światła.

Czułość spektralna diod LED, używanych jako fotodetektory, zależy od ich koloru emisji. Barwa, a ściślej mówiąc, długość emitowanej przez diodę fali elektromagnetycznej, zależy od przerwy energetycznej półprzewodnika, z jakiego wykonano strukturę. Zasadniczo, tak jak w przypadku diody krzemowej, tak w przypadku innych diod, elementy te absorbują światło o energii fotonu równej lub większej, od energii przerwy energetycznej w danym materiale. Emisja z kolei przypada dokładnie w energii przerwy energetycznej. Oznacza to, że dana dioda LED reaguje na światło o długości fali nie dłuższej niż emitowana. Czerwone i podczerwone diody LED reagować będą na cały zakres światła białego, diody zielone na kolor niebieski i zielony, ale już nie żółty i czerwony etc. Dodatkowo, pod uwagę wziąć trzeba kolor i sposób wykonania obudowy diody LED. Diody białe często mają powierzchnię pokrytą tzw. fosforem, który konwertuje światło (najczęściej ultrafioletowe) na światło białe. Takie diody są kiepskimi fotodetektorami. Podobnie jest w przypadku kolorowych diod z obudowami z kolorowego plastiku, ograniczają one pasmo światła docierającego do struktury i mogą redukować czułość elementu. Dobierając LED jako fotodetektor w naszym urządzeniu, trzeba mieć to na uwadze.

Diody LED nie są przez producenta charakteryzowane do pracy w tym trybie. Oznacza to, że jeżeli chcemy zastosować je w naszym projekcie, trzeba samodzielnie zmierzyć ich podstawowe parametry jako fotodetektora i zastosować je w projekcie z dość dużym zapasem. Niewielkie zmiany w procesie produkcji, o których dostawca elementów nawet nie musi informować w karcie katalogowej, jako że nie mają przełożenia na parametry LED mogą mieć istotny wpływ na pracę tego elementu jako fotodiody. Z uwagi na to stosowanie LED-ów jako elementów światłoczułych w urządzeniach produkowanych masowo może być problematyczne, ale doskonale sprawdzi się w krótkich, jednorazowych seriach czy pojedynczych urządzeniach.

Eleganckie aplikacje diod LED jako fotodetektorów

Diodę półprzewodnikową można podłączyć jako detektor światła do układu na dwa sposoby. Możliwe jest podłączenie w tzw. trybie fotowoltaicznym i w trybie fotoprzewodzącym. Panele słoneczne



Rysunek 1. Schemat ideowy układu z diodą LED podłączoną do układu jako sygnalizator i jako sensor z użyciem portu analogowego mikrokontrolera

działają w trybie fotowoltaicznym; światło na nie świeci, anoda staje się bardziej dodatnia niż katoda, a prąd (proporcjonalny do natężenia padającego światła) płynie w dowolnym obwodzie połączonym między anodą i katodą. Dioda jest spolaryzowana w takim układzie w kierunku przewodnictwa, a jej pojemność jest kilkukrotnie wyższa niż w diodzie spolaryzowanej w kierunku zaporowym.

Z kolei w trybie fotoprzewodzącym światło pada na fotodiodę spolaryzowaną zaporowo. Powoduje to również przepływ prądu proporcjonalnego do natężenia padającego światła, zmianę koncentracji nośników mniejszościowych i w konsekwencji prądu upływu diody. Trybu fotoprzewodzącego najlepiej jest używać dla sygnałów zmiennych, ponieważ odpowiedź częstotliwościowa jest w tym trybie lepsza, kosztem tego, że pomiar światła w tym trybie jest trudniejszy niż w trybie fotowoltaicznym.

Bardzo ciekawym zastosowaniem diody LED w urządzeniu jestysterowanie jej za pomocą analogowego pinu mikrokontrolera. Dzięki temu dioda LED może pracować w dwóch trybach – jako zwykły wskaźnik – kontrolka oraz jako fotodetektor. Wystarczy zastosować mikrokontroler wyposażony w wielofunkcyjny pin, który może pracować jako wyjście cyfrowe oraz jako wejście analogowe. W tego rodzaju piny wyposażone są układy takie jak mikrokontrolery ADUC7023, wiele układów z rodziny STM32 czy też stosowane w modułach Arduino układy ATmega firmy Microchip. Możliwe jest wtedy używanie diody zarówno jako sygnalizatora – wystarczy spolaryzować ją w kierunku przewodzenia, jak i pomiar jej fotoprądu w trybie fotowoltaicznym. Schemat takiego układu pokazano na **rysunku 1**. Gdy pin mikrokontrolera skonfigurowany jest do pracy w trybie cyfrowym, tak aby dioda LED1 funkcjonowała jako sygnalizator, opornik R1 ustala prąd płynący przez diodę w momencie, gdy pin mikrokontrolera przyjmuje wartość zerową (wyjście znajduje się wtedy na potencjale masy). Prąd płynący przez opornik R2 jest pomijalny, gdyż ten opór powinien być znacznie większy niż R1. W momencie, gdy pin mikrokontrolera przyjmie stan wysoki, potencjał VCC na jego wyjściu sprawi, że różnica potencjałów na LED1 będzie zerowa, a co za tym idzie, prąd przez diodę nie popłynie. W momencie, gdy pin mikrokontrolera zostanie skonfigurowany jako wejście analogowe (do wbudowanego ADC), będzie miał wysoką impedancję wejściową. Teraz, gdy na diodę LED1 padnie światło, wygeneruje ono pewien fotoprąd, który następnie popłynie przez oporniki R1 i R2. Zakładając, że impedancja wejściowa mikrokontrolera jest dużo większa od R2 (dla układów z rodziny Microchip ATmega impedancja wejściowa pinów analogowych wynosi około 100 M Ω), napięcie mierzone przez ten pin wynosić będzie:

$$I_{\text{foto}} \times R2$$

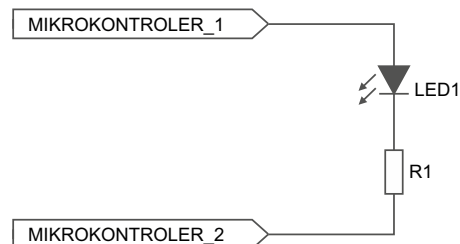
gdzie: I_{foto} to wygenerowany na LED1 fotoprąd, który dla typowej diody LED wynosić będzie kilkanaście...kilkadziesiąt mikroamperów.

Jeśli zatem jako R2 zastosujemy opornik o rezystancji kilkuset kiloomów, to na wejściu ADC zobaczymy kilka woltów (na przykład dla wspomnianego wyżej prądu 20 μ A i opornika 220 k Ω na wejściu ADC uzyskamy 4,4 wolta, czyli niemalże maksymalne napięcie wejściowe dla zasilanych 5 V układów AVR). Oczywiście realnie napięcie jest ograniczone spadkiem napięcia na diodzie LED1 i na oporniku R1. O ile spadek napięcia na oporniku R2 będzie bardzo mały i można go pominąć, o tyle spadek napięcia na typowej czerwonej diodzie LED wynosi typowo 2,2 V, a np. na diodzie niebieskiej aż 3,2 V. Pozostawia to jednak dostatecznie dużo napięcia do 5 V (czy nawet 3,3 V w przypadku diody czerwonej), aby wykorzystać LED jako prosty sensor natężenia światła.

Dodatkowo, jeśli nie chcemy lub nie możemy wykorzystać pinu analogowego w naszym mikrokontrolerze, to za pomocą dwóch cyfrowych pinów możemy zestawić układ o podobnym działaniu co powyżej, opierający się jednak na zupełnie innej zasadzie działania [3]. Na **rysunku 2** zaprezentowano schemat ideowy podłączenia diody LED w takim układzie. Aby zapalić diodę LED1, wystarczy pin 1 (mikrokontroler_1) ustawić w stanie wysokim, a pin 2 (mikrokontroler_2) w stanie niskim. Spowoduje to przepływ prądu przez diodę. Wartość prądu będzie ograniczona poprzez opornik R1, należy go dobrać do wymaganego prądu naszej diody. Jednocześnie spowoduje to rozładowanie się wewnętrznej pojemności diody (w kierunku zaporowym). Jeśli teraz chcemy wykorzystać diodę LED1 z rysunku 2 jako sensor światła, należy zamienić wartości na pinach 1 i 2 – na pinie 1 ustawiamy wartość niską, a na pinie 2 wysoką. Spowoduje to naładowanie wewnętrznej pojemności diody. Zajmuje to nie więcej niż kilka milisekund. Następnie pin 2 przełączany jest z wyjścia na wejście. Wejście mikrokontrolera ma wtedy wysoką, ale skończoną impedancję i powoli rozładowuje naładowaną diodę. Czas rozładowania diody jest proporcjonalny do natężenia oświetlenia, padającego na strukturę diody i wynosi, w zależności od natężenia oświetlenia, od poniżej kilkuset mikrosekund do kilku milisekund. Możliwe jest wykorzystanie tego mechanizmu nie tylko do statycznego mierzenia natężenia oświetlenia, ale również jako sposób komunikacji pomiędzy dwoma urządzeniami, znajdującymi się blisko siebie i świecącymi nawzajem w swoje diody LED [3]. Tego rodzaju rozwiązanie jest niezwykle praktyczne i tanie – funkcja komunikacji z wykorzystaniem diod LED dodawana jest tylko i wyłącznie w oprogramowaniu i nie wymaga wzbogacenia układu o interfejsy bezprzewodowe – optyczne (np. IrDA lub RC-5) lub radiowe (np. 433 MHz czy Wi-Fi), które zwiększają koszt systemu.

Podsumowanie

Niepozorna dioda LED może być również zadziwiająco czułą fotodiodą, która umożliwia nie tylko pomiar natężenia oświetlenia



Rysunek 2. Schemat ideowy układu z diodą LED podłączoną do układu jako sygnalizator i jako sensor z użyciem dwóch linii cyfrowych mikrokontrolera

otoczenia (np. projektowane przez nas urządzenie może wykrywać, czy jest dzień, czy noc lub czy włączone jest w pomieszczeniu światło), ale także zestawienie prostego systemu komunikacji half-duplex bez zwiększania kosztów projektowanego systemu.

Jednak trzeba pamiętać, że nie tylko dioda LED może być fotodiodą – tak zachować się może także każda inna dioda, w tym diody zabezpieczające wejścia czy diody prostownicze, a to oznacza, że jeżeli wykorzystujemy elementy w szklanych lub innych przezroczystych obudowach, światło padające na nie jest bez znaczenia i może wprowadzać do naszego układu zakłócenia, chociażby takie, jak przydźwięk sieciowy, pochodzący z modulowanego 100 Hz światła żarówek czy świetlówek.

Warto może na koniec zauważyć, że o ile dioda LED może działać jako fotodioda, o tyle nie ma efektu odwrotnego, tj. fotodioda (zwykła krzemowa) nie może emitować światła. Wynika to z właściwości struktury energetycznej krzemu, a dokładniej z faktu, że ekstrema wykresów energia-pęd $W(k)$ dla pasma przewodnictwa i walencyjnego są przesunięte, a więc rekombinacja skośna nośników zamienia energię elektronu w kwant ciepła (fonon), a nie w kwant światła (foton). W tych materiałach energia tracona przez elektron przy przejściu z pasma przewodnictwa do pasma walencyjnego zamienia się w drgania sieci krystalicznej – ciepło, a nie w emitowane światło.

Nikodem Czechowski, EP

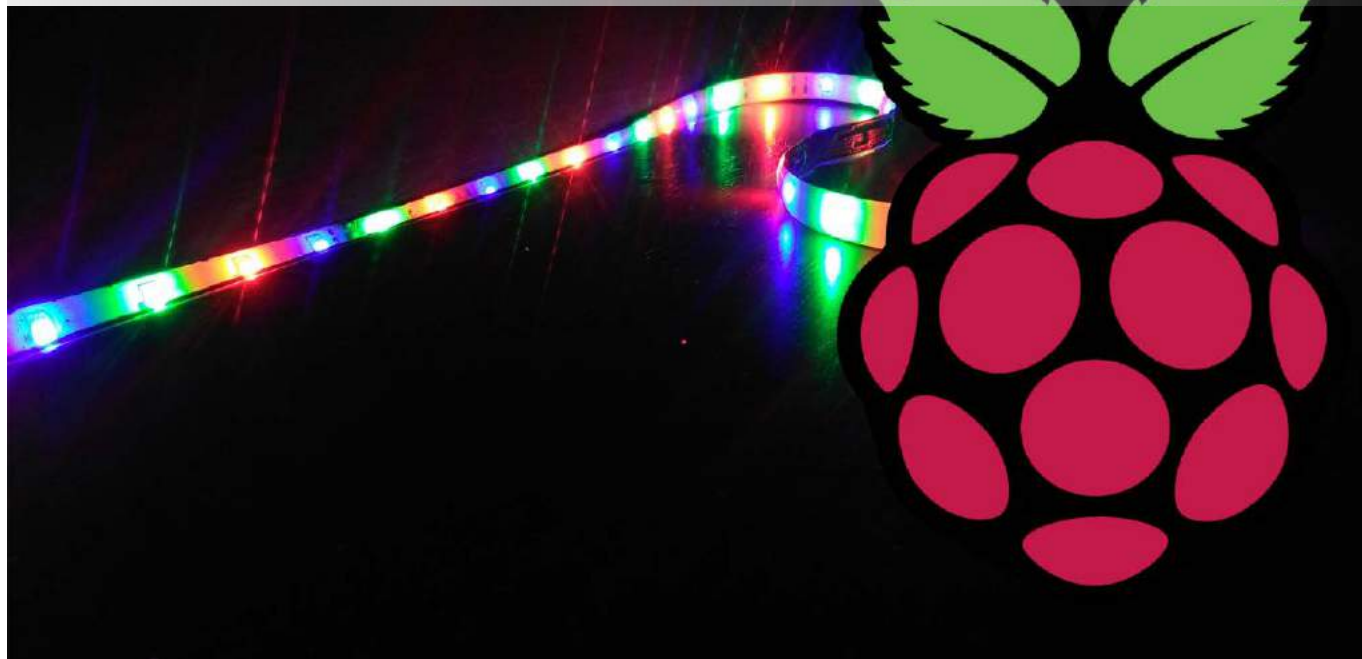
Źródła:

- [1] J. Bryant, „Glass Diodes May See the Light – and Hum”, Rarely Asked Questions 45, Design News, maj 2009, <http://bit.ly/2MmkpPp>
- [2] J. Bryant, „LEDs are Photodiodes Too”, Rarely Asked Questions 108, Analog Dialogue, sierpień 2014, <http://bit.ly/2ITg3Nm>
- [3] P. H. Dietz, W.S. Yerazunis, D.L. Leigh, „Very Low-Cost Sensing and Communication Using Bidirectional LEDs”, ACM International Conference on Ubiquitous Computing (UbiComp), październik 2003.

REKLAMA

KITy AVT na wideo <http://bit.ly/2ScLZTy>
O KIT-ach AVT przeczytasz również na Facebooku
<http://bit.ly/2BjVMN7>





Sterowanie diodami RGB z Raspberry Pi

Dioda RGB pozwala na wyświetlenie dowolnego koloru. Może funkcjonować jako kontrolka, a w postaci modułów i taśm z wieloma diodami służy zwykle jako źródło oświetlenia. W artykule opiszemy, jak sterować diodami RGB z komputera Raspberry Pi.

Dioda RGB składa się z trzech diod elektroluminescencyjnych o różnych barwach, umieszczonych w jednej obudowie. Uzyskany wypadkowy kolor światła wynika z niezależnego sterowania jasnością każdej ze składowych barw. A ponieważ barwy zlewają się w jednym obszarze, mamy wrażenie płynnej zmiany kolorów w niemalże całym zakresie widzenia ludzkiego oka. Istnieje wiele sposobów na uzyskanie takiego sterowania z użyciem minikomputera Raspberry Pi (lub jednego z podobnych modułów, pracujących pod kontrolą systemu operacyjnego Linux). Opiszemy dwa z nich, które sterują diodą za pomocą trzech przebiegów PWM.

Mieszanie barw w RGB

Model kolorów RGB jest addytywnym modelem opisu tworzenia kolorów, w którym światło czerwone, zielone i niebieskie jest dodawane do siebie na różne sposoby w celu odtworzenia szerokiego spektrum kolorów. Addytywny model kolorów oznacza, że wynikowe widmo koloru jest sumą widm poszczególnych jego składników. Nazwa modelu pochodzi od skrótów angielskich nazw trzech podstawowych kolorów, używanych w tym modelu – Red (czerwony), Green (zielony) i Blue (niebieski). W uproszczeniu model ten zobrazowany jest na **rysunku 1**.

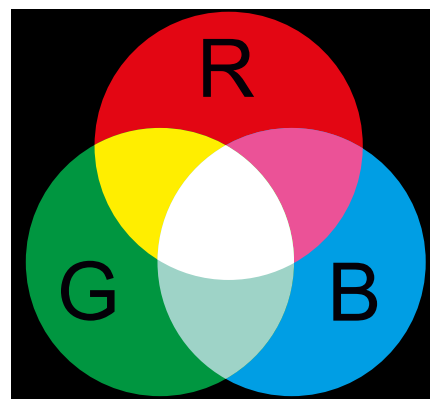
Głównym celem stosowania modelu kolorów RGB jest opis, reprezentacja i wyświetlanie kolorów w systemach elektronicznych, zwłaszcza takich jak telewizory i komputery, aczkolwiek jego historia sięga początków fotografii analogowej. Model RGB ma za sobą solidną teorię w zakresie postrzegania kolorów przez ludzi.

Aby utworzyć dowolny kolor z barw składowych, należy nałożyć na siebie trzy wiązki światła. Każda z nich może mieć dowolną

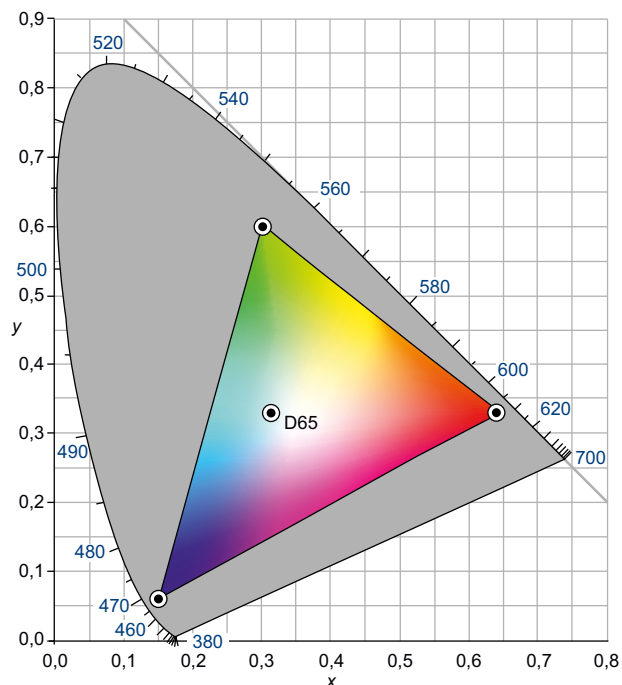
intensywność, od całkowitego wyłączenia (0) do pełnego włączenia (100%). Zerowa intensywność dla każdego komponentu daje najciemniejszy kolor – czarny (bez światła), a pełna intensywność wszystkich barw daje w sumie kolor biały. Jakość tej bieli jest zależna od precyzyjnego doboru składowych źródeł światła i ich widm. Gdy intensywność wszystkich składowych barw jest taka sama, wynikiem jest kolor szary o jasności zależnej od intensywności barw. Gdy intensywności są różne, rezultatem jest zabarwiony odcień, mniej lub bardziej nasycony w zależności od różnicy najsilniejszej i najsłabszej intensywności zastosowanych kolorów podstawowych.

Gdy jeden ze składników ma najwyższą intensywność, wynikowy kolor jest odcieniem zbliżonym do tego podstawowego koloru, a gdy dwa składniki mają tę samą najsilniejszą intensywność, wówczas kolor jest odcieniem jednego z kolorów uzupełniających (cyjan, magenta lub żółty). Widać to dokładnie na wykresie przedstawionym na **rysunku 2**. Jest to tzw. gamut, czyli przestrzeń barwna – w tym przypadku sRGB. Na wykresie zaznaczone są cztery punkty – trzy to barwy podstawowe, a czwartym jest tzw. D65, czyli umowny kolor biały; odpowiada on kolorowi emisji ciała doskonale czarnego o temperaturze 6500 K.

Na wykresie przedstawiono gamut sRGB z zaznaczonymi trzema punktami – składowymi kolorami oraz kolorem białym o temperaturze barwowej 6500 K (D65.)



Rysunek 1. Uprozczone przedstawienie addytywnego mieszania barw w modelu RGB



Rysunek 2. Wykres przestrzeni barwnej CIE_{xy}1931 do przedstawiania gamutu (przestrzeni barwowej) w modelu RGB

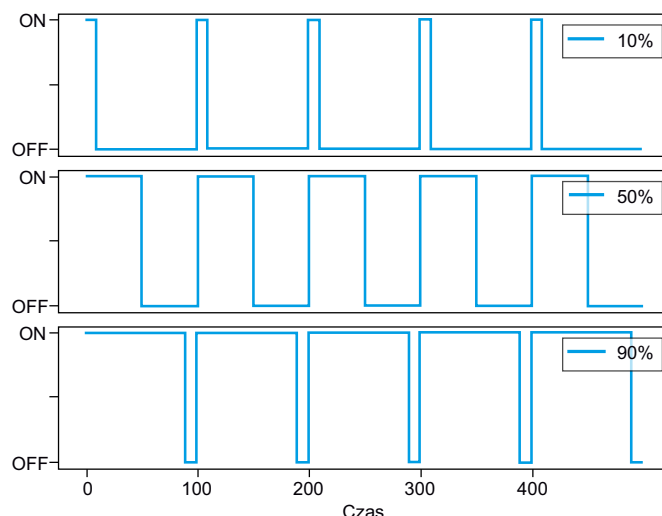
Sam model kolorów RGB nie definiuje, co oznacza dokładnie kolor czerwony, zielony czy niebieski. Z uwagi na to wyniki ich mieszania nie są bezwzględnie określone, ale zależą od kolorów podstawowych. Dopiero po zdefiniowaniu dokładnych kolorów podstawowych ogólny model kolorów staje się konkretną absolutną przestrzenią kolorów (gamutem), taką jak wspomniane sRGB czy Adobe RGB. Zdefiniowanie ścisłej przestrzeni barwowej jest szczególnie istotne w systemach telewizyjnych, oprogramowaniu graficznym etc. W przypadku kontroli prostej diody RGB nie jest to krytyczne, o ile nasza aplikacja nie musi ściśle odwzorowywać pewnych kolorów.

Sterowanie PWM

Sterowanie diod elektroluminescencyjnych za pomocą impulsów o zmiennym wypełnieniu jest bardzo praktycznym rozwiązaniem. Do sterowania diodą LED w ten sposób wystarczy pojedyncze wyjście cyfrowe, które może generować przebieg prostokątny. Modulacja PWM polega na zmianie wypełnienia przebiegu prostokątnego, który podawany jest na wyjście cyfrowe układu. W dużym uproszczeniu tego rodzaju sterowanie polega na naprzemiennym włączaniu i wyłączaniu zasilania diody. Stosunek czasu, gdy zasilanie jest załączone i jest podawane na wyjściu, do ogólnego czasu trwania jednego okresu nazywa się właśnie wypełnieniem sygnału. Wartość ta podawana jest w procentach. Dla wypełnienia 50% zasilanie będzie podawane na diodę LED przez połowę czasu trwania okresu. Na **rysunku 3** zaprezentowano trzy przebiegi tego rodzaju, różniące się wypełnieniem (pokazano przebiegi o wypełnieniu (od góry): 10%, 50% oraz 90%).

Jeśli zasilanie jest przełączane na diodzie dostatecznie często, to mruganie diod nie będzie w ogóle widoczne dla człowieka – wystarczy około kilkuset herców, aby mruganie było niezauważalne, nawet jeżeli dioda lub obserwator są w ruchu. Rekomenduje się używanie częstotliwości co najmniej 1 kHz do kluczowania sygnału PWM dla diod LED.

W systemach cyfrowych sygnał PWM generuje się najczęściej, wykorzystując przebieg prostokątny o znacznie krótszym okresie (wyższej częstotliwości) niż wyjściowy przebieg PWM. Układ odpowiedzialny za generowanie tego sygnału zlicza okresy tego zegara – przez określoną liczbę cykli sygnał na wyjściu jest załączony, potem wyłącza się, po to tylko, żeby po określonej liczbie cykli zresetować się, załączyć zasilanie i ponownie rozpocząć zliczanie cykli. To, ile maksymalnie



Rysunek 3. Przykładowe przebiegi PWM o wypełnieniu 10% (na górze), 50% (pośrodku) i 90% (na dole)

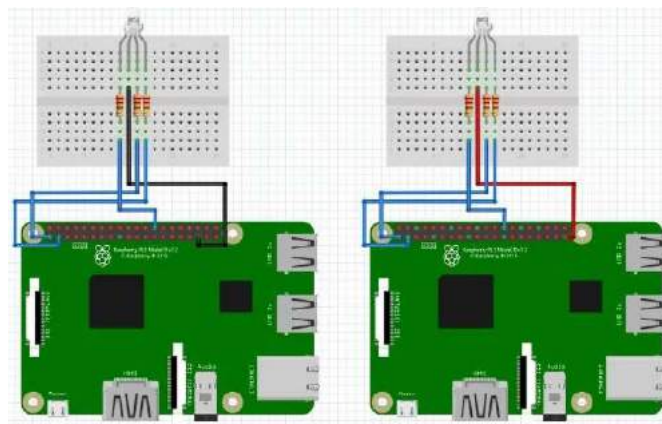
cykli zliczać może dany system, mówi nam o rozdzielczości modułu PWM – na przykład 10-bitowy licznik jest w stanie zliczać do 1024, co oznacza, że cały cykl sygnału podzielony może być na 1024 fragmentów, co daje nam rozdzielczość PWM równą ok. 0,1%.

Generowanie PWM na Raspberry Pi

Aby wygenerować przebieg PWM, skorzystamy z pythonowej biblioteki GPIO, która pozwala na obsługę wyjść cyfrowych RPi. W bibliotece znajdziemy moduł GPIO.PWM, który pozwala na generowanie przebiegu o zmiennej modulacji. Przebieg ten generowany jest programowo – tworzony jest osobny wątek dla każdego wyjścia PWM w systemie, co zapewnia dostateczną precyzję generowania przebiegu PWM, wymaganą do zadań takich jak sterowanie diodami LED itp.

Na **rysunku 4** zaprezentowano dwa schematy podłączenia diody RGB do modułu Raspberry Pi, aby sterować nią poprzez sygnały PWM. W zależności od tego, czy zastosowana przez nas dioda ma wspólną katodę (schemat po lewej, katoda do masy – czarny przewód), czy wspólną anodę (schemat po prawej, anoda do zasilania 3,3 V – czerwony przewód), bazujemy na odpowiednim diagramie z **rysunku 4**. Jeśli chcemy diody zasilac z napięcia 5 V, konieczne trzeba dodać rezystory na wszystkich kanałach RGB. Oporniki stosowane są do ograniczenia prądu płynącego przez LED-y. Dla typowych elementów wartość ta powinna wynosić 150 Ω dla kanału czerwonego i po 100 Ω dla kanałów zielonego i niebieskiego.

Na **listingu 1** zaprezentowano kod programu do generowania przebiegów PWM na pinach, do których podłączone są poszczególne diody



Rysunek 4. Dwa schematy podłączenia diody RGB do modułu Raspberry Pi, dla modelu ze wspólną katodą (po lewej) oraz ze wspólną anodą (po prawej)

```

Listing 1. Kod programu do sterowania dyskretną diodą RGB
z wykorzystaniem przebiegów PWM na komputerze jednopłytkowym
Raspberry Pi
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM) # konfiguracja konwencji oznaczenia pinów GPIO
GPIO.setwarnings(False) # wyłączenie ostrzeżeń podczas pracy kodu
RUNNING = True

# definicja pinów wyjściowych dla poszczególnych kanałów RGB
green = 20
red = 21
blue = 22

# definiujemy wszystkie kanały jako wyjścia
GPIO.setup(red, GPIO.OUT)
GPIO.setup(green, GPIO.OUT)
GPIO.setup(blue, GPIO.OUT)

# wybór częstotliwości przebiegu PWM
Freq = 100

# definicja kanałów PWM na poszczególnych pinach
RED = GPIO.PWM(red, Freq)
GREEN = GPIO.PWM(green, Freq)
BLUE = GPIO.PWM(blue, Freq)

try:
    # główna pętla programu
    while RUNNING:
        # uruchomienie PWM, 100 oznacza 100% wypełnienia
        RED.start(100)
        GREEN.start(1)
        BLUE.start(1)

        for x in range(1,101):
            # zmiana wypełnienia sygnału PWM
            GREEN.ChangeDutyCycle(x)
            # opóźnienie pomiędzy każdą zmianą
            # (czas w sekundach)
            time.sleep(0.05)

        for x in range(1,101):
            RED.ChangeDutyCycle(101-x)
            time.sleep(0.025)

        for x in range(1,101):
            GREEN.ChangeDutyCycle(101-x)
            BLUE.ChangeDutyCycle(x)
            time.sleep(0.025)

        for x in range(1,101):
            RED.ChangeDutyCycle(x)
            time.sleep(0.025)

        # aby przerwać działanie programu, wystarczy
        # nacisnąć dowolny klawisz na klawiaturze
        except KeyboardInterrupt:

            RUNNING = False
            GPIO.cleanup() # wyłączenie wszystkich kanałów PWM

```

w diodzie RGB, według schematu z rysunku 4. Aby zademonstrować działanie, wystarczy uruchomić program.

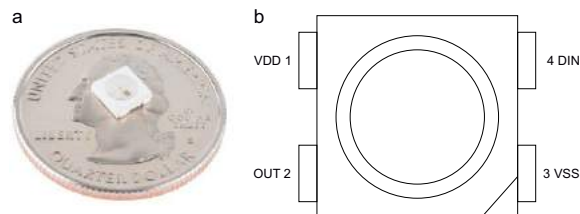
Scalone diody RGB ze zintegrowanym sterownikiem

Inne podejście do sterowania diodą RGB oferują scalone moduły, które integrują w jednej obudowie trzy diody – czerwoną, zieloną i niebieską – oraz kontroler, który generuje przebiegi PWM dla każdego z tych kanałów. Do komunikacji pomiędzy takim modulem a sterownikiem w postaci minikomputera Raspberry Pi służy interfejs cyfrowy.

Jednymi z najpopularniejszych diod RGB ze zintegrowanym sterownikiem są układy WS28xx. Przykładowa dioda tego rodzaju pokazana jest na **rysunku 5a**. Są one dostępne w szerokiej gamie elementów, różniących się liczbą diod w strukturze – niektóre z nich zawierają trzy diody – czerwoną, zieloną i niebieską, a inne dodatkowo uzupełnione są diodą białą, tworząc tzw. diodę RGBW. Symbol elektryczny takiej diody pokazano na **rysunku 5b**. Ma ona cztery piny, dwa z nich (VCC i GND) służą do zasilania, a dwa pozostałe (DIN oraz DOUT) do komunikacji.

Protokół komunikacji z diodami RGB WS2812B

Protokół komunikacji z modułami tego rodzaju bazuje na kodowaniu NRZ, czyli bez powrotu do zera. Jest to rodzaj kodowania, który zera i jedynki pozwala zapisać jako przejścia pomiędzy dwoma stanami (napięciami), a to, czy dane zboczy jest zerem, czy jedynką (czy resetem), uwarunkowane jest zależnościami czasowymi. Na **rysunku 6** zaprezentowano takie przejścia dla układu WS2812B.



Rysunek 5. Dioda RGB ze zintegrowanym sterownikiem, typu WS28xx

Moduł WS2812B ma wewnętrzny, 24-bitowy zatrask, który zatraskuje dane wejściowe, w momencie gdy na wejście DIN podany zostanie komplet bitów po resecie. Kolejne bity podawane na wejście podawane są do układu, który wzmacnia je, regeneruje zboczy etc. i podaje na wyjście – DOUT. Dzięki temu można łączyć kaskadowo ze sobą diody RGB tego rodzaju w szereg, podłączając wyjście DOUT jednej diody do wejścia DIN kolejnej diody. Na **rysunku 7** zaprezentowano schemat kaskadowego układu tego typu.

Dane przesyłane do układu są uporządkowane w formacie GRB – zielony/czerwony/niebieski, gdzie pierwszy w ramce jest najstarszy bit. Każdej składowej odpowiada osiem bitów, co pozwala uzyskać 16777216 kolorów (2^{24}). Aby sterować diodą WS2812B z minikomputera Raspberry Pi, podłączamy jej wejście danych do jednego pinu danych. Jeśli chcemy podłączyć więcej modułów RGB, łączymy je kaskadowo, jak pokazano na **rysunku 7**. Do sterowania takim modulem można użyć biblioteki **neopixel** (którą trzeba przed uruchomieniem programu pobrać i zainstalować za pomocą narzędzia **pip**:

```

sudo pip3 install rpi_ws281x
adafruit-circuitpython-neopixel

```

Kod programu do sterowania łańcuchem diod LED RGB WS2812 pokazano na **listingu 2**. Program to proste demo, które prezentuje możliwości kontrolowania kolorów.

Podsumowanie

Diody RGB dają duże możliwości – mogą świecić na dowolny kolor, dzięki czemu np. po zastosowaniu ich jako kontrolki w naszym urządzeniu jedną diodą można przekazać użytkownikowi wiele informacji. Z użyciem komputera jednopłytkowego Raspberry Pi sterowanie taką diodą (czy samym elementem dyskretnym, czy modulem z wbudowanym sterownikiem) jest bardzo proste i nie wymaga dużych nakładów pracy.

Nikodem Czechowski, EP

Źródła:

1. <http://bit.ly/2p31ro7>, 2. <http://bit.ly/2MJlvVu>

Listing 2. Kod programu do sterowania łańcuchem diod LED RGB WS28

```

import time
from neopixel import *
import argparse

# Konfiguracja listwy LED
LED_COUNT      = 16      # Liczba kaskadowo połączonych diod
LED_PIN        = 18      # Numer pinu GPIO podłączonego do diody
LED_FREQ_HZ    = 800000  # Częstotliwość komunikacji z LED-ami w hercach (typowo 800 kHz)
LED_DMA        = 10      # Kanał DMA wykorzystywany do generowania sygnału (typowo 10)
LED_BRIGHTNESS = 255     # Ustawienie jasności - 0 to najciemniej a 255 najjaśniej
LED_INVERT     = False    # Odwrócenie sygnałów, ustawmy na True, jeżeli np. odwracamy sygnał na translatorze poziomów
LED_CHANNEL    = 0        # set to '1' for GPIOs 13, 19, 41, 45 or 53

# Definicja funkcji, które w różny sposób animować będą diody

def colorWipe(strip, color, wait_ms=50):
    """Przejście koloru przez diody, jedna po drugiej."""
    for i in range(strip.numPixels()):
        strip.setPixelColor(i, color)
        strip.show()
        time.sleep(wait_ms/1000.0)

def theaterChase(strip, color, wait_ms=50, iterations=10):
    """Animacja z przechodzącymi i goniącymi się kolorami."""
    for j in range(iterations):
        for q in range(3):
            for i in range(0, strip.numPixels(), 3):
                strip.setPixelColor(i+q, color)
            strip.show()
            time.sleep(wait_ms/1000.0)
        for i in range(0, strip.numPixels(), 3):
            strip.setPixelColor(i+q, 0)

def wheel(pos):
    """Generacja tęczowego wzoru przez wszystkie pozycje."""
    if pos < 85:
        return Color(pos * 3, 255 - pos * 3, 0)
    elif pos < 170:
        pos -= 85
        return Color(255 - pos * 3, 0, pos * 3)
    else:
        pos -= 170
        return Color(0, pos * 3, 255 - pos * 3)

def rainbow(strip, wait_ms=20, iterations=1):
    """Wyswietlanie wszystkich kolorów tęczy na wszystkich pikselach naraz."""
    for j in range(256*iterations):
        for i in range(strip.numPixels()):
            strip.setPixelColor(i, wheel((i+j) & 255))
        strip.show()
        time.sleep(wait_ms/1000.0)

def rainbowCycle(strip, wait_ms=20, iterations=5):
    """Wyswietlanie wszystkich kolorów tęczy w postaci jednorodnego rozkładu na wszystkich diodach."""
    for j in range(256*iterations):
        for i in range(strip.numPixels()):
            strip.setPixelColor(i, wheel((int(i * 256 / strip.numPixels()) + j) & 255))
        strip.show()
        time.sleep(wait_ms/1000.0)

def theaterChaseRainbow(strip, wait_ms=50):
    """Przechodzące kolory tęczy przez diody."""
    for j in range(256):
        for q in range(3):
            for i in range(0, strip.numPixels(), 3):
                strip.setPixelColor(i+q, wheel((i+j) % 255))
            strip.show()
            time.sleep(wait_ms/1000.0)
        for i in range(0, strip.numPixels(), 3):
            strip.setPixelColor(i+q, 0)

# Program główny
if __name__ == '__main__':
    # Przetwarzanie argumentów, z jakimi uruchomiono skrypt
    parser = argparse.ArgumentParser()
    parser.add_argument('-c', '--clear', action='store_true', help='wyczyść ekran wychodząc z programu')
    args = parser.parse_args()

    # Utworzenie obiektu NeoPixel z odpowiednią konfiguracją
    strip = Adafruit_NeoPixel(LED_COUNT, LED_PIN, LED_FREQ_HZ, LED_DMA, LED_INVERT, LED_BRIGHTNESS, LED_CHANNEL)
    # Inicjalizacja biblioteki - musi być ona uruchomiona, zanim zaczniemy korzystać z jej funkcji
    strip.begin()

    print('Naciśnij Ctrl-C aby wyjść.')
    if not args.clear:
        print('Wykorzystaj argument "-c" aby wyczyścić LEDy wychodząc')

    try:
        while True:
            print('Animacje z przechodzeniem koloru.')
            colorWipe(strip, Color(255, 0, 0)) # Przejście czerwonego
            colorWipe(strip, Color(0, 255, 0)) # Przejście niebieskiego
            colorWipe(strip, Color(0, 0, 255)) # Przejście zielonego
            print('Animacja z goniącymi się kolorami.')
            theaterChase(strip, Color(127, 127, 127)) # Animacja z kolorem białym
            theaterChase(strip, Color(127, 0, 0)) # Animacja z kolorem czerwonym
            theaterChase(strip, Color(0, 0, 127)) # Animacja z kolorem niebieskim
            print('Rainbow animations.')
            rainbow(strip)
            rainbowCycle(strip)
            theaterChaseRainbow(strip)

    except KeyboardInterrupt:
        if args.clear:
            colorWipe(strip, Color(0,0,0), 10)

```

AudioDSP zestaw z procesorem Sigma DSP ADAU1701

Na łamach EP opublikowany został cykl artykułów w formie kursu o procesorach Sigma DSP od Analog Devices. W ofercie sklepu Kamami dostępny jest zestaw uruchomieniowy AudioDSP dla procesora tej rodziny typu ADAU1701A. Jest to doskonała platforma uruchomieniowa. Pozwoli poznać możliwości sprzętowe, opanować zasady konfiguracji procesora DSP oraz praktycznie korzystać ze środowiska Sigma Studio.

AudioDSP jest przystępną cenowo platformą sprzętową dla procesora sygnałowego Analog Devices ADAU1701A z rodziny Sigma DSP. Zestaw wraz z programatorem USBi i darmowym środowiskiem Sigma Studio umożliwia obróbkę sygnału audio zarówno ze źródeł analogowych jak i cyfrowych (SPDIF/I²S). Modułowa budowa zestawu AudioDSP umożliwia elastyczne dopasowanie zasobów sprzętowych do wymogów aplikacji, ułatwiając szybką rozbudowę funkcjonalną przy zastosowaniu modułów własnego pomysłu lub gotowych z oferty AVT, Kamami czy Digilent.

Wygląd zestawu prezentuje **fotografia 1**. Pełny zestaw składa się z czterech modułów:

- płytki bazowej z procesorem DSP ADAU1701A_DSP,
- płytki interfejsu użytkownika ADAU1701A_GPIO,
- płytki interfejsu cyfrowego audio ADAU1701A_SPDIF (2 sztuki).

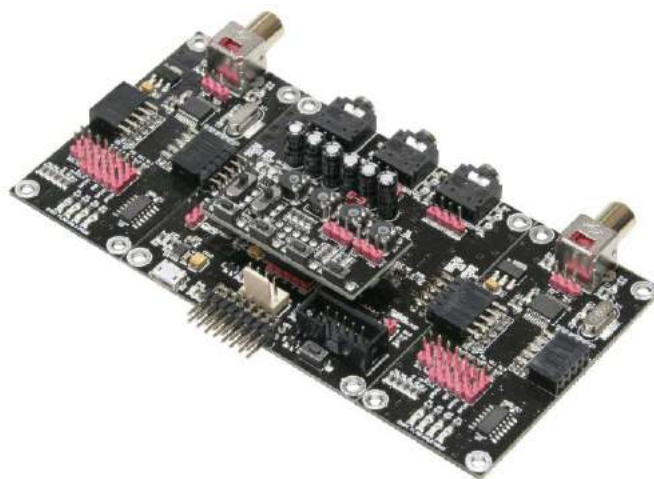
Dla rozpoczęcia pracy z AudioDSP konieczny jest jeszcze programator USBi (**fotografia 2**). Umożliwia on pełną konfigurację sprzętową ADAU1701A, symulację w czasie rzeczywistym oraz programowanie pamięci programu (EEPROM) zestawu oraz konfigurację poprzez magistrale I²C układów audio produkowanych przez Analog Devices.

Procesor

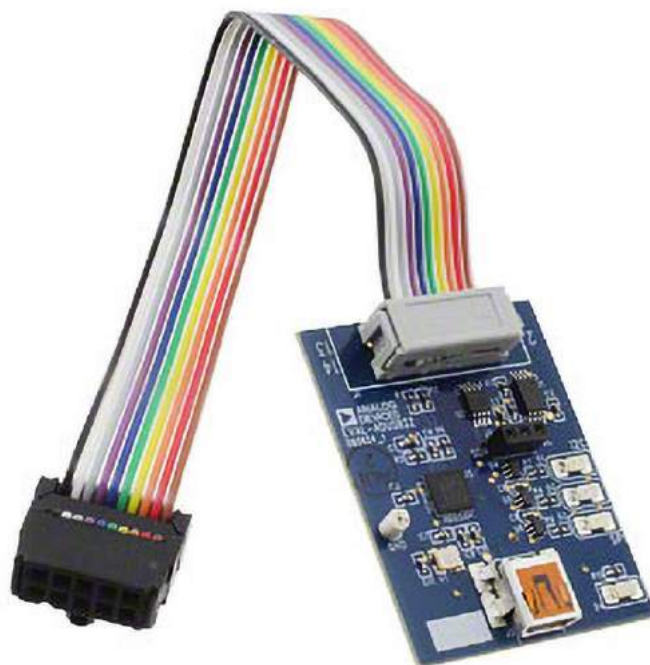
Aby w pełni wykorzystać zestaw AudioDSP konieczne jest zapoznanie się z budową blokową jego serca, czyli procesora DSP. Układ ADAU1701A jest nie tylko procesorem DSP, ale całym zintegrowanym systemem cyfrowej obróbki sygnału audio, wyposażony w przetworniki AD/DA, uniwersalne porty GPIO oraz cyfrowe interfejsy audio. Współpracuje z zewnętrzną pamięcią programu EEPROM I²C, dzięki czemu możliwa jest budowa systemu obróbki sygnału audio, bez zewnętrznego procesora sterującego i przy zastosowaniu niewielkiej liczby elementów.

Schemat wewnętrzny układu ADAU1701 pokazano na **rysunku 3**. Jego najistotniejsze cechy to:

- wydajność na poziomie 50 MIPS, 28/56 bitowy rdzeń DSP,
- wbudowany oscylator i konfigurowalny blok PLL,
- obsługa częstotliwości próbkowania do 192 kHz,
- wbudowane dwa przetworniki ADC 24 bitowe (SNR 100 dB),
- wbudowane cztery przetworniki DAC 24 bitowe (SNR 104 dB),
- dwukierunkowy interfejs cyfrowy (8 kanałów), z możliwością elastycznej konfiguracji dwukanałowego cyfrowego interfejsu szeregowego (I²S, LJ, RJ) oraz wielokanałowego trybu multipleksowanego czasowo (TDM),
- programowalne GPIO, w tym cztery wejścia analogowe, dla realizacji interfejsów użytkownika, bez konieczności zastosowania procesora zewnętrznego,

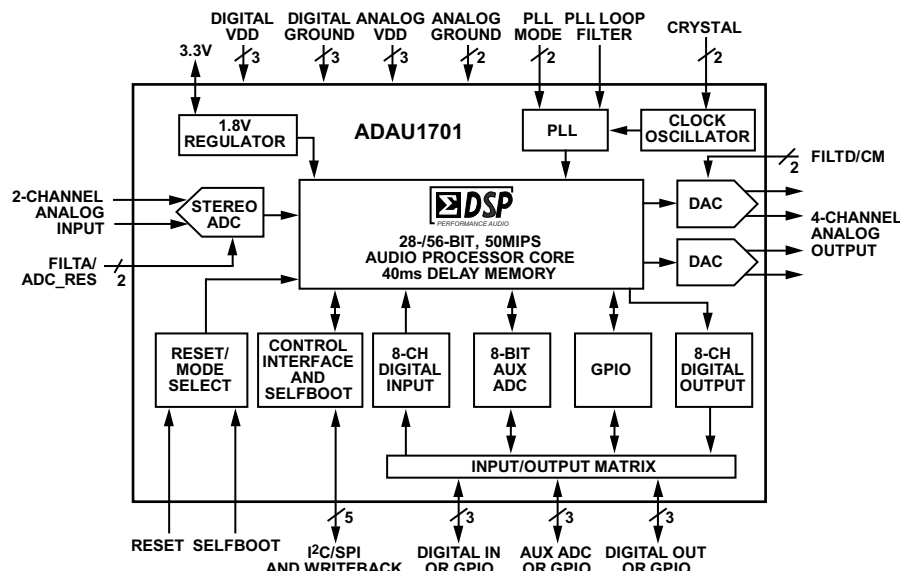


Fotografia 1. Wygląd zestawu AudioDSP

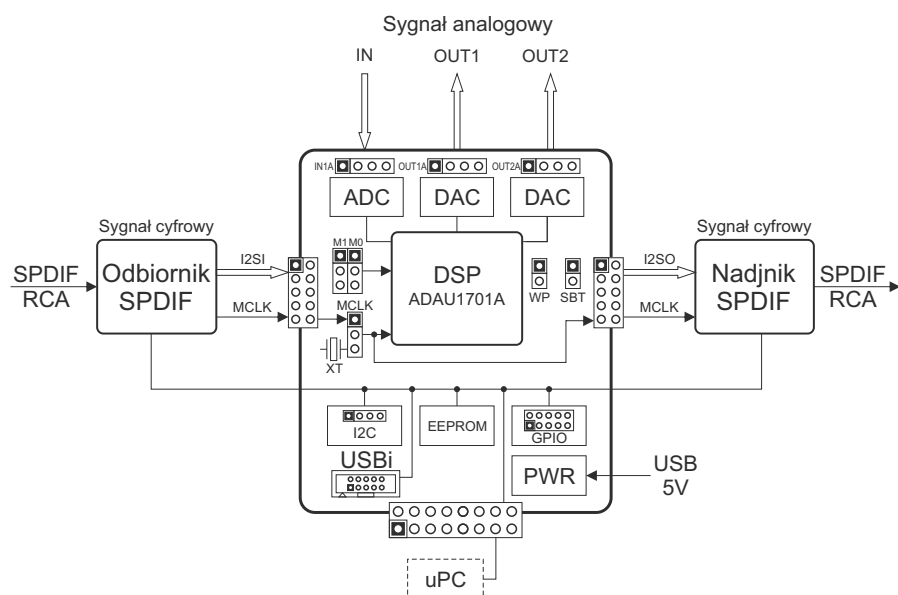


Fotografia 2. Programator USBi

- obsługa zewnętrznej pamięci programu i konfiguracji I²C EEPROM,
- możliwa samodzielna praca DSP (tryb *SELFBOOT*), bez zewnętrznego procesora sterującego,
- tryb *WriteBack*, umożliwiający zachowanie parametrów aplikacji DSP w pamięci EEPROM przy pracy samodzielnej, podczas zaniku zasilania,
- zasilanie 3,3 V, wbudowany stabilizator dla zasilania rdzenia DSP 1,8 V,
- konfiguracja i programowanie za pomocą intuicyjnego graficznego środowiska Sigma Studio.



Rysunek 3. Schemat blokowy ADAU1701A



Rysunek 4. Schemat blokowy zestawu AudioDSP

Opis elementów zestawu

Schemat blokowy zestawu, ilustrujący przepływ sygnałów audio, sterujących oraz połączenia pomiędzy poszczególnymi modułami przedstawia **rysunek 4**. Podstawą zestawu jest płyta bazowa ADAU1701A_DSP, której wygląd pokazano na **fotografii 5** a schemat na **rysunku 6**. Umożliwia bezpośrednią obróbkę analogowego sygnału audio z próbkowaniem 44,1 kHz dla wlutowanego rezonatora 11,2896 MHz lub 48 kHz dla rezonatora 12,288 MHz. Zwora MCLK umożliwia wybór źródła taktowania DSP – z wbudowanego rezonatora lub zewnętrznym sygnałem MCLK np. z modułu interfejsu cyfrowego (złącze I2SI). Buforowany sygnał zegarowy MCLK doprowadzony jest na złącze I2SO i można wykorzystać go do taktowania zewnętrznych układów DAC, np. cyfrowych wzmacniaczy mocy, nie posiadających własnego oscylatora. Dwie zwory M0 i M1 ustawiają mnożnik pętli PLL rdzenia DSP.

Płyta AudioDSP wyposażona jest w dwa wejścia analogowe (stereo) i cztery wyjścia analogowe (dwa kanały stereo) o poziomie liniowym. Sygnały audio doprowadzone są do DSP poprzez gniazda mini Jack 3,5 mm stereo oraz dodatkowo wraz z zasilaniem 5 VDC wyprowadzone na złącza szpilkowe IN1A, OUT1A, OUT2A. Ułatwia to podłączenie zgodnych modułów rozszerzeń m.in. dla przedwzmacniaczy, mikrofonów MEMS, wzmacniaczy mocy SSM itp.



Rysunek 5. Płyta bazowa AudioDSP

Wszystkie porty ADAU1701A wyprowadzono na złącza szpilkowe (R=2,54 mm) i pogrupowano funkcjonalnie na port sterujący GPIO, wejście I2SI oraz wyjście I2SO sygnału cyfrowego interfejsu audio.

Płyta wyposażona jest w złącze magistrali I²C w standardzie 3,3 V, zgodne m.in. z modułem ADUCino360, oraz port UPC udostępniający sygnały niezbędne do sterowania DSP także poprzez zewnętrzny mikrokontroler.

Wbudowana pamięć EEPROM typu 24LC256 przechowuje program i dane aplikacji w trybie samodzielnym (*SELFBOOT*) zwalniając z konieczności sterowania DSP zewnętrznym mikrokontrolerem.

Wszystkie złącza rozszerzeń posiadają wspólną magistralę I²C ułatwiającą konfigurowanie DSP oraz współpracujących układów zewnętrznych. Płyta posiada wbudowany niskoszumny stabilizator 3,3 V typu ADP3338 zasilający procesor ADAU1701A. Napięcie 1,8 V zasilające rdzeń DSP stabilizowane jest w obwodzie tranzystora Q1. Podczas programowania zestaw zasilany jest z programatora USBi. W przypadku pracy samodzielnej w docelowej aplikacji przewidziano dodatkowe

gniazdo zasilające mikro USB (5 V/500 mA lub większej dobranej w zależności od poboru mocy modułów rozszerzeń). Led PWR sygnalizuje obecność zasilania 5 V.

Na płycie znajdują się także dwa obwody pomocnicze. Pierwszy ADM811 generuje sygnał RESET (RES), z możliwością zdalnego wyzwolenia poprzez programator USBi oraz manualnie z przycisku RES. Drugi obwód oparty o ADM812, odpowiada za automatyczne generowanie sygnału *WriteBack* (WB) informującego DSP o zaniku zasilania i uruchamiającego procedurę zapisu parametrów i konfiguracji aplikacji do pamięci nieulotnej EEPROM. Zwory SBT (*Selfboot*) oraz WP (*Write Protect*) konfiguruje tryb pracy DSP.

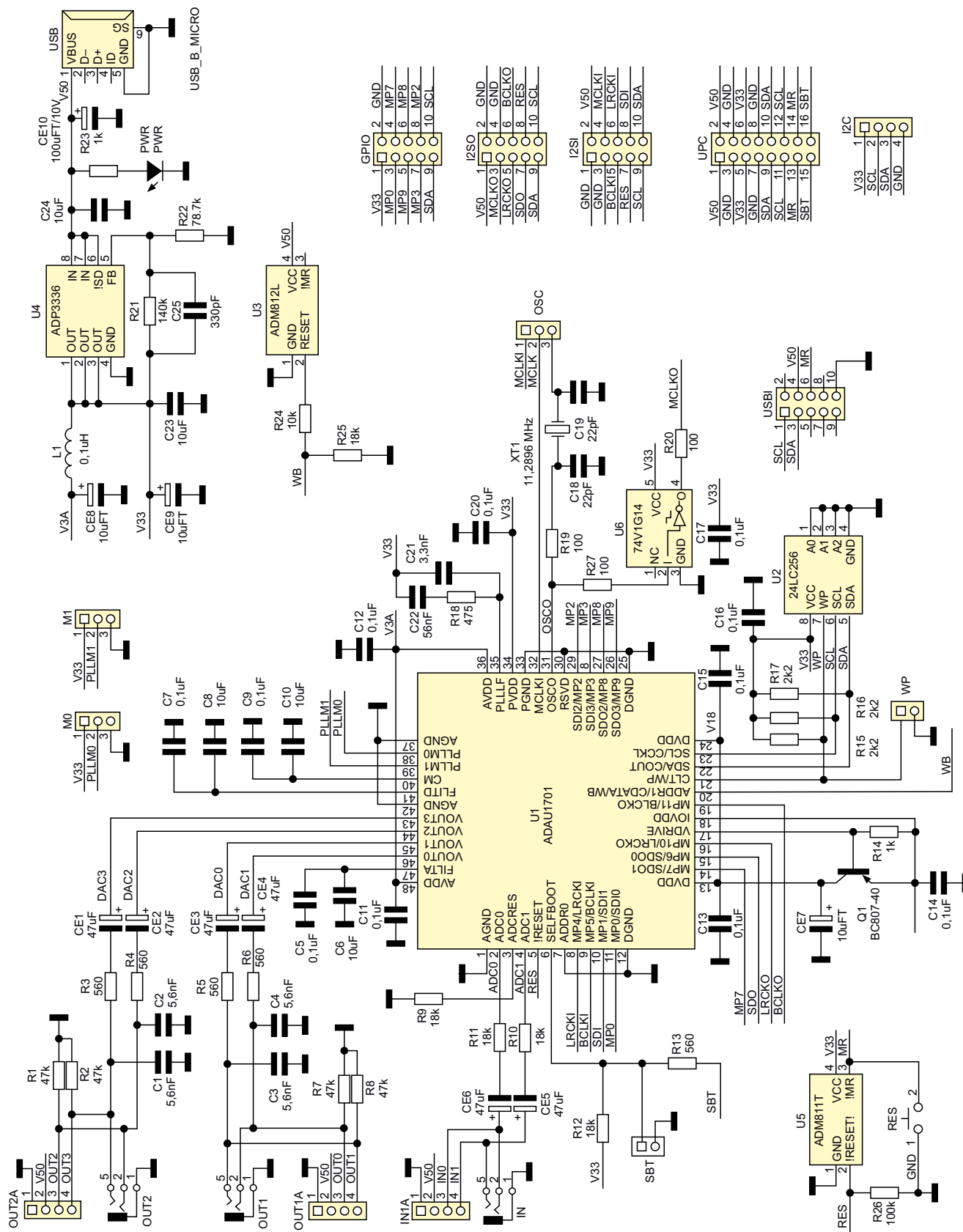
Drugim elementem zestawu jest płyta interfejsu użytkownika – DSP ADAU1701A_GPIO, której wygląd prezentuje **fotografia 7**. Procesor ADAU1701, posiada programowalne, wielofunkcyjne wyprowadzenia we/wy GPIO umożliwiające sterowanie pracą programu za pomocą zewnętrznych sygnałów analogowych i cyfrowych oraz podstawowe dwustanowe sterowanie układów współpracujących. Pozwala to na rezygnację z zastosowania zewnętrznego procesora sterującego nawet w złożonych aplikacjach.

Płyta DSP ADAU1701A_GPIO posiada wbudowane typowe manipulatory GPIO, takie jak cztery przyciski, dwa przełączniki suwakowe, cztery potencjometry oraz dwie buforowane diody LED. Połączenie płytek DSP i GPIO nie wymaga dodatkowych przewodów, płytkę

rozszerzeń GPIO montuje się korzystając z gniazda GPIO, bezpośrednio na płytce bazowej DSP (fotografia 1).

Schemat płytki pokazano na **rysunku 8**. Potencjometry RV1, RV2 podłączone są na stałe do wyprowadzeń GPIO2 i 3. Potencjometry RV3, RV4 współdzielą porty MP8 i 9 z przyciskami chwilowymi PB3 i 4, wybór aktywnej funkcji możliwy jest zworami MP8 i 9. Porty MP0 i 1 połączone są z przyciskami chwilowymi PB1

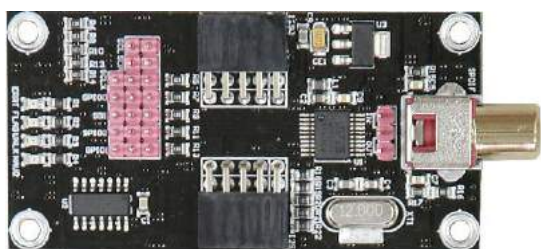
i PB2 z równolegle połączonymi przełącznikami suwakowymi SW1 i SW2. Porty MP0 i 1 wyposażone są także w diody LED sygnalizujące ich stan, gdy pracują jako wyjścia. Bufory są niezbędne, gdyż wydajność prądowa ADAU1701 jest niewielka, konieczne trzeba o tym pamiętać w przypadku używania wyjść GPIO we własnej aplikacji. Szeregowe rezystory 470 Ω chronią przed skutkami błędnej konfiguracji. Wyprowadzenia GPIO pracują w standardzie



Rysunek 6. Schemat płytki bazowej AudioDSP



Fotografia 7. Płytką interfejsu użytkownika

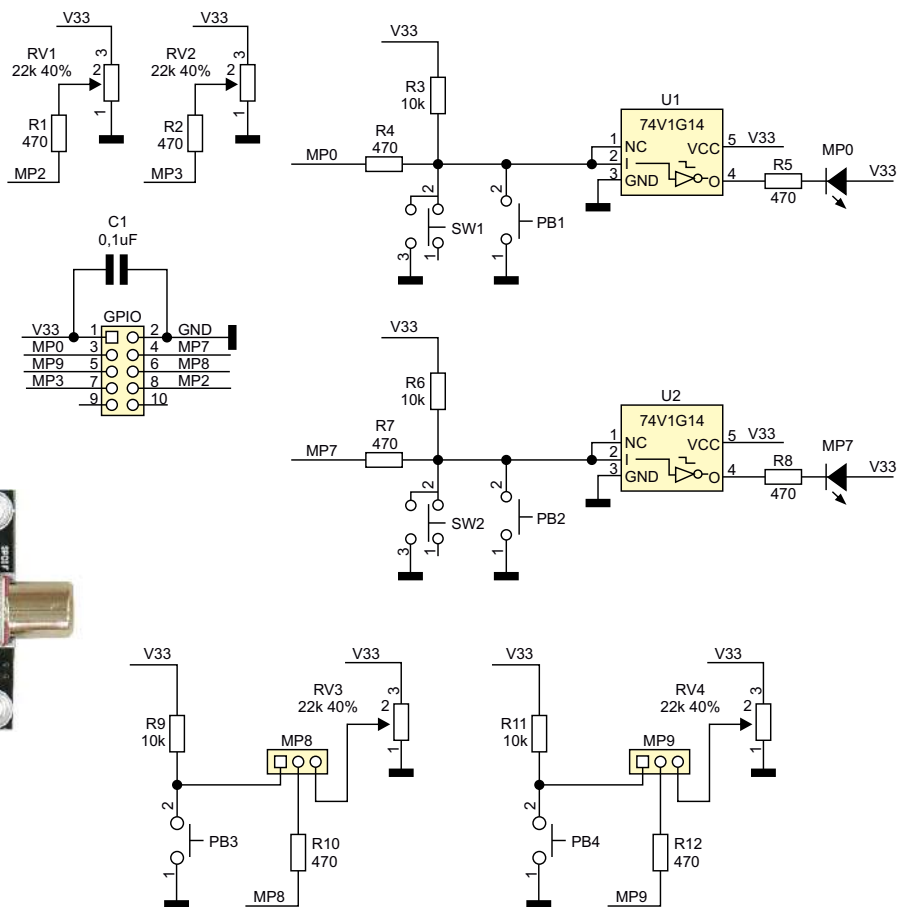


Fotografia 9. Płytką interfejsu cyfrowego

napięciowym 3,3 V, przekroczenie tego napięcia trwale uszkodzi procesor.

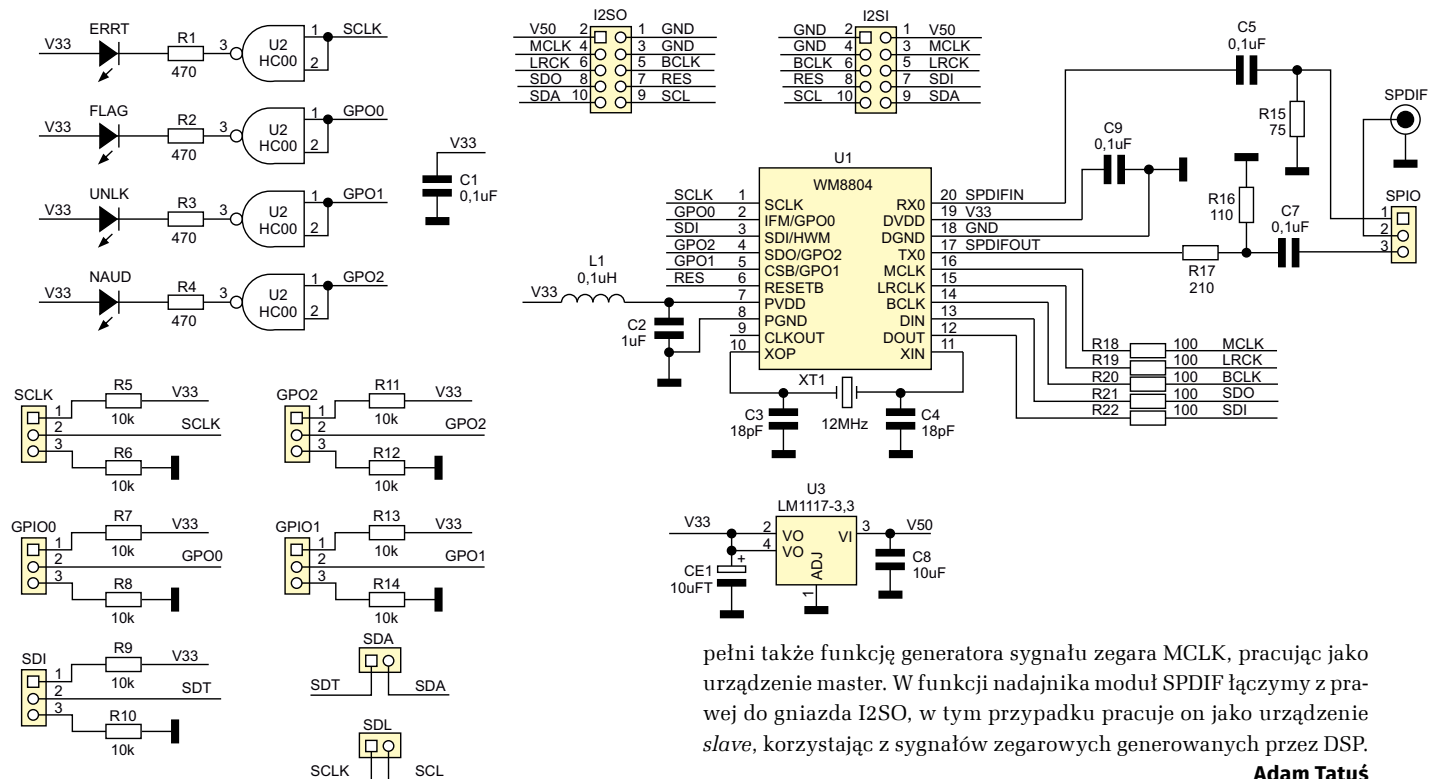
Trzecią i ostatnią płytką zestawu jest płytka interfejsu ADAU1701_SPDIF (2 sztuki), której wygląd prezentuje **fotografia 9**. Płytkę powstała, aby umożliwić bezpośrednią obróbkę sygnału cyfrowego w standardzie SPDIF (ADAU1701A nie posiada wbudowanego interfejsu SPDIF). Schemat modułu pokazano na **rysunku 10**.

Moduł jest konfigurowalny i realizuje funkcje odbiornika lub nadajnika. Bazuje na popularnym układzie WM8804. Zwory konfiguracyjne umożliwiają wprowadzenie WM8804 w tryb pracy samodzielnej



Rysunek 8. Schemat płytki interfejsu użytkownika

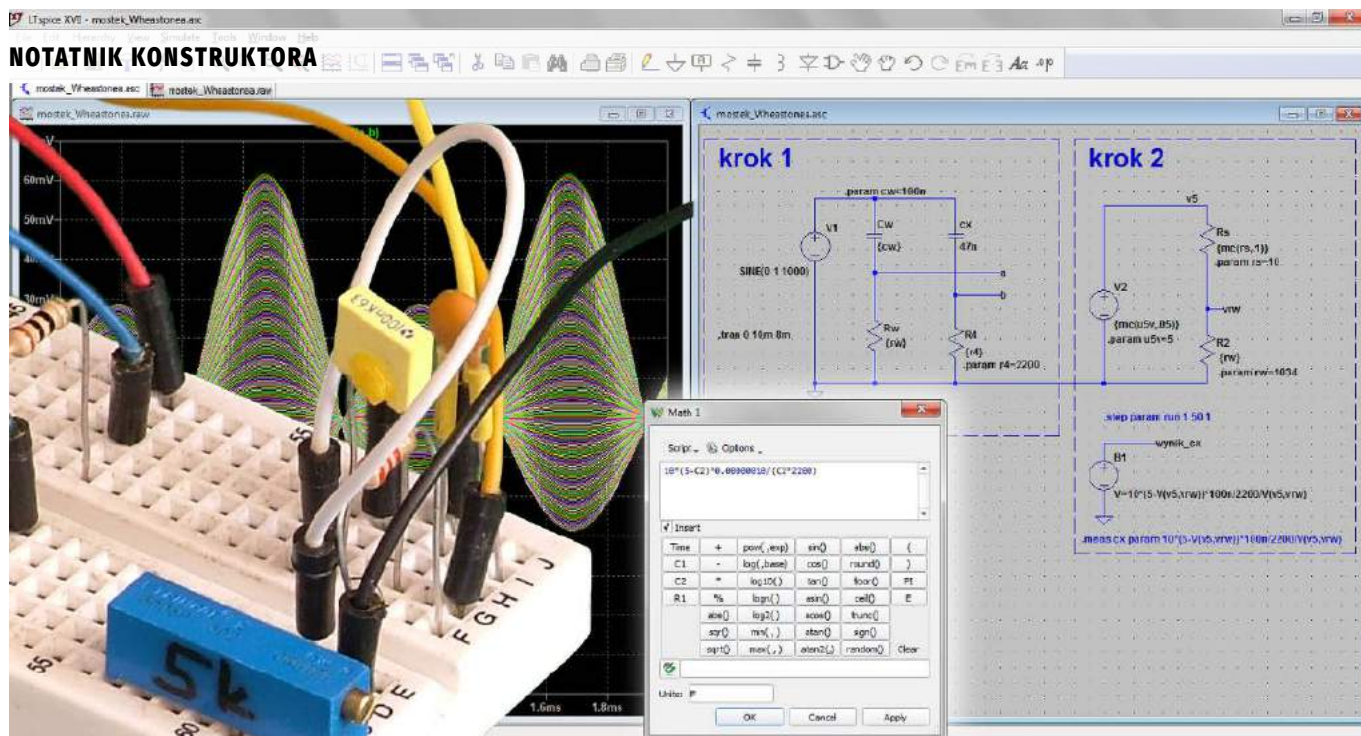
lub konfigurowanej przez I²C, określają standard transmisji szerego-
wej oraz konfiguruje tryb odbiornik-nadajnik SPDIF. Diody LED sy-
gnalizują aktualny status WM8804. Po konfiguracji modułu SPDIF
w funkcji odbiornika podłączamy go z lewej strony DSP (gniazda we/
wy na górnej krawędzi) do złącza I2SI płytki DSP. W tym przypadku



Rysunek 10. Schemat płytki interfejsu cyfrowego

pełni także funkcję generatora sygnału zegara MCLK, pracując jako urządzenie master. W funkcji nadajnika moduł SPDIF łączymy z prawej do gniazda I2SO, w tym przypadku pracuje on jako urządzenie *slave*, korzystając z sygnałów zegarowych generowanych przez DSP.

Adam Tatus
adam.tatus@ep.com.pl



Symulacja i pomiar czyli LTspice i Analog Discovery 2 w rękach konstruktora (4)



Pomiar pojemności mostkiem Wheatstone'a

O tym, że na dokładność pomiaru bardzo często decydujący wpływ ma zastosowana metoda wie nawet początkujący elektronik. Czasami mamy pomysł jak można by mierzyć określony parametr, ale nie jesteśmy pewni czy sprawdzi się on w praktyce. Nawet jeśli z technicznego punktu widzenia przyjęta metoda umożliwiałaby wykonanie takiego pomiaru, zawsze warto, a nawet należy rozważyć jaką niepewnością czy błędem będzie ona obciążona. Rozważmy dość nietypowe zastosowanie mostka Wheatstone'a do... pomiaru pojemności.

Charles Wheatstone (**fotografia 1**) miał szczęście urodzić się w roku 1802, co dawało mu szansę znalezienia się w zaszczytnym gronie wielu dziewiętnastowiecznych odkrywców. Początkowo jednak nic takiego biegu wydarzeń nie wróżyło, gdyż ojciec małego Charlesa, jako zawodowy muzyk i producent instrumentów muzycznych pragnął podobnie wykształcić syna. Dbał zresztą o to należycie posyłając go do dobrych szkół. Plany ojca wkrótce się spełniły. Charles zainteresował się muzyką, terminował nawet w fabryce instrumentów.

Nabył w niej praktycznych umiejętności i doświadczenia w zakresie budowy przyrządów precyzyjnych, co zaowocowało skonstruowaniem własnej harmonii. Szybko jednak okazało się, że muzyka nie była jedyną pasją Charlesa Wheatstone'a. Interesował się też m.in. optyką i elektrycznością. W 1837 roku opracował pierwszy telegraf elektryczny, ale jego najbardziej znanym wynalazkiem jest mostek prądu stałego pozwalający na bardzo precyzyjny pomiar rezystancji. My spróbujemy użyć tego urządzenia do pomiaru pojemności, ale przeciwieństwiemu prądu stałego raczej nie da się mierzyć pojemności. Pierwszą zatem modyfikacją będzie zasilenie mostka napięciem zmiennym i przyjęcie, że wszystkie elementy mostka są impedancjami, nie rezystancjami. Podobny tok rozumowania przeprowadził prawdopodobnie inny wynalazca Max Karl Werner Wien (1866-1938) opracowując swoją wersję mostka, ale czy jego modyfikacje były



Fotografia 1. Charles Wheatstone
(źródło <http://bit.ly/2MYibFV>)

prowadzone w celu uzyskania narzędzia do pomiaru pojemności?

Przygotowanie do pomiarów – obliczenia

Wróćmy zatem do mostka Wheatstone'a. W trzech jego gałęziach były umieszczone rezystory o znanych wartościach, dodatkowo jeden z nich był regulowany. Badaną rezystancję umieszczano w czwartej gałęzi. Zasada pomiaru polegała na zrównoważeniu mostka za pomocą regulowanej oporności, a następnie obliczeniu rezystancji mierzonej na podstawie zależności wyznaczającej warunek równowagi mostka. W naszej konstrukcji rezystancje zastąpimy impedancjami (**rysunek 1**). Musimy wyznaczyć warunek równowagi mostka. Zakładamy, że obciążenie mostka jest pomijalne, czyli traktujemy go tak, jakby nie był obciążony w ogóle. Wynika z tego, że prąd I_L płynie zarówno przez impedancję Z_1 , jak i przez Z_3 . Podobnie prąd I_p płynie przez impedancje Z_2 i Z_4 . Prądy te są równe:

$$I_L = \frac{U_Z}{Z_1 + Z_3} \quad I_p = \frac{U_Z}{Z_2 + Z_4}$$

$$U_A = I_L Z_3 \quad U_B = I_p Z_4$$

Napięcie wyjściowe między węzłami A i B jest równe:

$$U_{AB} = U_A - U_B$$

W stanie równowagi jest ono równe 0, wynika z tego zależność:

$$\frac{U_Z Z_3}{Z_1 + Z_3} = \frac{U_Z Z_4}{Z_2 + Z_4}$$

stąd:

$$Z_3 (Z_2 + Z_4) = Z_4 (Z_1 + Z_3)$$

i ostatecznie:

$$Z_2 Z_3 = Z_1 Z_4 \quad (1)$$

Gdybyśmy zasilili mostek napięciem stałym a impedancje zamienili rezystancjami otrzymalibyśmy wariant Charlesa Wheatstone'a, jednak mostek ten chcemy przystosować do pomiaru pojemności. Dokonujemy więc modyfikacji polegającej na zastąpieniu rezystorów górnych gałęzi pojemnościami (**rysunek 2**), przy czym jedna z nich – C_X będzie pojemnością mierzoną, natomiast C_W będzie dokładnie znaną pojemnością wzorcową. Zmodyfikujemy teraz warunek równowagi mostka. Ogólne impedancje zastępujemy przyjętymi elementami, a więc:

$$Z_1 = \frac{1}{j\omega C_W}$$

$$Z_2 = \frac{1}{j\omega C_X}$$

$$Z_3 = R_W$$

$$Z_4 = R_4$$

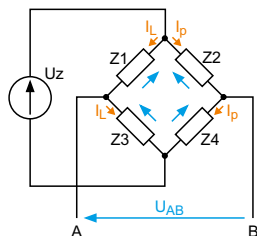
Warunek równowagi ma teraz postać:

$$\frac{R_W}{\omega C_X} = \frac{R_4}{\omega C_W}$$

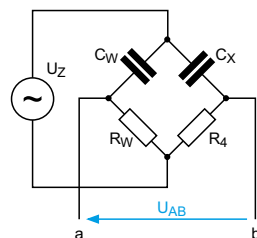
z czego wynika, że mierzona pojemność jest równa:

$$C_X = \frac{R_W C_W}{R_4} \quad (2)$$

Mamy wszystko, co jest teoretycznie potrzebne do pomiaru pojemności C_X . Spróbujmy jeszcze nieco usprawnić nasz układ pomiarowy tak, aby wszystkie czynności mogły być wykonane wyłącznie za pomocą Analog Discovery 2. Jak wynika z zależności (2) pojemność C_X jest zależna od elementów stałych o znanych wartościach tj. C_W i R_4 , a także od regulowanej rezystancji wzorcowej R_W . Rezystorem



Rysunek 1. Schemat mostka Wheatstone'a z impedancjami zamiast rezystancji



Rysunek 2. Schemat mostka, który zastosowano w pomiarach

R_W będzie w naszym układzie precyzyjny potencjometr wieloobrotowy. Zmodyfikujemy układ tak, aby możliwe było mierzenie jego rezystancji. Trudność polega na tym, że przyrząd Analog Discovery 2 nie ma wbudowanego omomierza. Pomiar będzie więc 2-etapowy. W pierwszym kroku zrównoważymy mostek kręcąc suwakiem potencjometru R_W i obserwując jednocześnie na oscyloskopie napięcie wyjściowe. W drugim kroku rezystor R_W zostanie odłączony od mostka i dołączony do źródła napięcia stałego 5 V przez rezystor szeregowy R_S . Umożliwi on zmierzenie prądu płynącego w 2 kroku przez rezystor R_W a znając prąd i napięcie będzie już można obliczyć rezystancję R_W z prawa Ohma. Schemat pomiarowy całego stanowiska pokazano na **rysunku 3**. Źródło dodatniego napięcia stałego V_+ jest dostępne w Analog Discovery 2, W1 to 1 kanał generatora arbitralnego wytwarzający sygnał sinusoidalny o częstotliwości 1000 Hz i amplitudzie 3 V. Pierwszy kanał oscyloskopu dołączamy bezpośrednio do punktów „a” i „b” korzystając z możliwości różnicowego trybu pracy wejścia. Drugi kanał będzie mierzył różnicowe napięcie na rezystorze szeregowym R_S o znanej oporności 10 Ω . Analog Discovery 2 ma możliwość definiowania własnych wyrażeń matematycznych służących do obliczania wartości mierzonego parametru. Skorzystamy z tej cechy tak, aby w drugim kroku przyrząd podawał obliczony wynik uwalniając nas od obliczeń. W tym celu oporność R_W we wzorze (2) należy przedstawić tak, aby odpowiadała wielkościom występującym lub mierzonym w rzeczywistości. Są to: $V_+ = 5$ V, $R_S = 10$ Ω , U_{CH2} – napięcie zmierzone 2 kanałem oscyloskopu zależne od ustawionej rezystancji potencjometru R_W . Zakładamy ponadto, że rezystor R_4 będzie miał oporność 2200 Ω , a pojemność kondensatora wzorcowego C_W będzie równa 100 nF.

Rozpatrując obwód zastosowany w drugim kroku pomiaru zawierający źródło napięcia V_+ , rezystor R_S , przełącznik ustawiony w pozycji 2 i rezystor R_W można dla niego zapisać równanie:

$$V_+ - I_2 R_S - I_2 R_W = 0$$

przy czym

$$I_2 = \frac{U_{CH2}}{R_S}$$

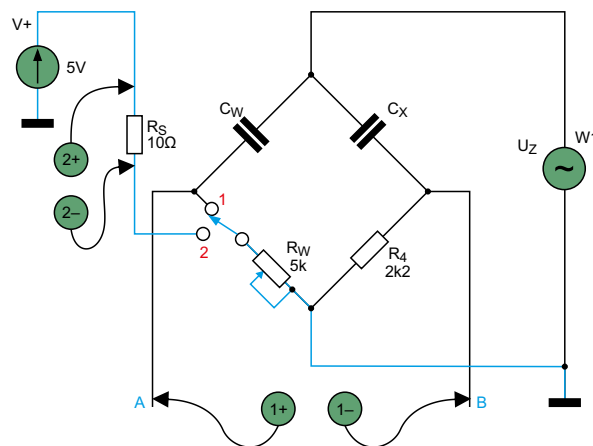
gdzie U_{CH2} – napięcie zmierzone 2 kanałem oscyloskopu.

Z zależności tej wynika, że rezystancja potencjometru R_W jest równa:

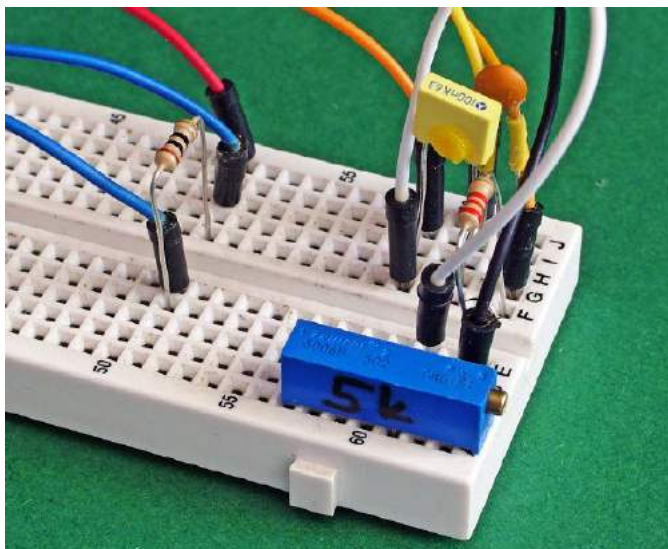
$$R_W = \frac{V_+ - I_2 R_S}{I_2} = \frac{V_+ - \frac{U_{CH2}}{R_S} R_S}{\frac{U_{CH2}}{R_S}} = \frac{V_+ R_S - U_{CH2} R_S}{U_{CH2}}$$

Powyższą zależność można już podstawić do wyrażenia (2), co pozwoli bezpośrednio obliczyć pojemność mierzoną na podstawie danych uzyskanych z narzędzi pomiarowych Analog Discovery 2.

$$C_X = \frac{R_W C_W}{R_4} = \frac{R_S (V_+ - U_{CH2}) C_W}{U_{CH2} R_4} \quad (3)$$



Rysunek 3. Schemat pomiarowy z użyciem zestawu Analog Discovery 2



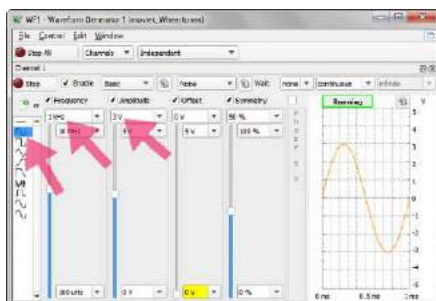
Fotografia 2. Układ pomiarowy zmontowany na płytce stykowej

W wyrażeniu zastosowanym w Analog Discovery 2 wpisujemy konkretne wartości liczbowe, które przyjęliśmy wcześniej. Mamy więc ostatecznie:

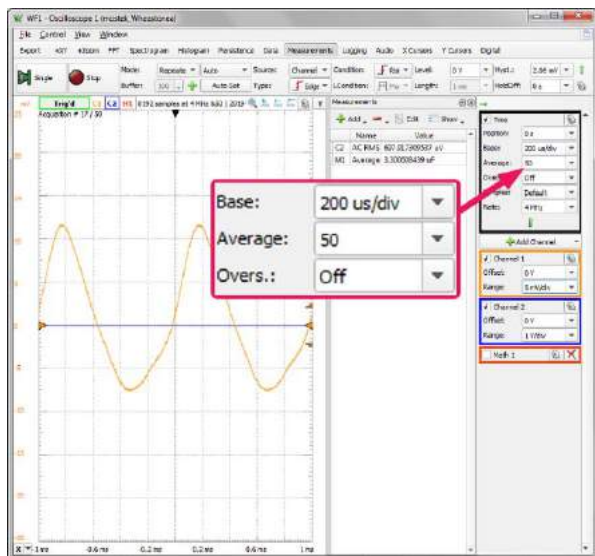
$$C_X = \frac{10 \cdot (5 - U_{CH2}) \cdot 0,0000001}{U_{CH2} \cdot 2200} \quad (4)$$

Taką formułę trzeba będzie zapisać w odpowiednim polu programu Waveforms. Liczba 10 widniejąca w liczniku reprezentuje rezystancję rezystora szeregowego, natomiast liczba 0,0000001 to pojemność wzorcowa. Celowo nie wymnożono ich ze sobą, aby można było później ewentualnie wpisać w formułę Analog Discovery 2 dokładne wartości tych elementów zmierzone wcześniej precyzyjnym miernikiem.

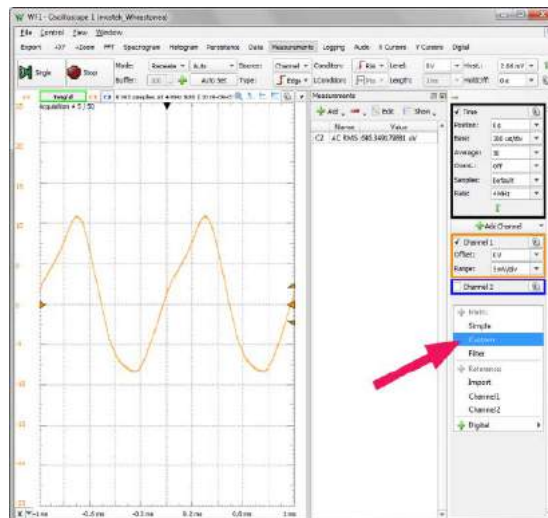
Wykonamy pomiary dwóch kondensatorów. Pierwszy z nich będzie miał pojemność 100 nF, drugi 47 nF. W obu przypadkach zastosujemy ten sam kondensator wzorcowy C_w o pojemności 100 nF. Teoretycznie, dla zapewnienia symetrii



Rysunek 4. Parametry generatora ustawione w Analog Discovery 2



Rysunek 5. Przebieg występujący w stanie zrównoważenia mostka

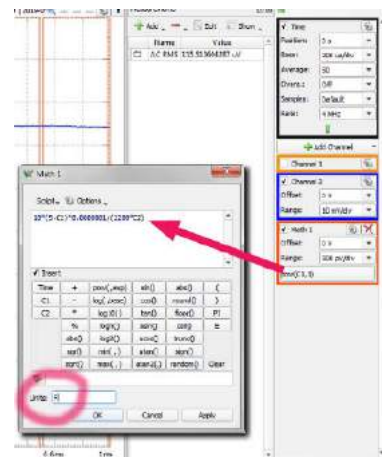


Rysunek 6. Uruchomienie okna, w którym wprowadzana jest formuła obliczeniowa

układu pojemność ta powinna być jak najbliższa pojemności mierzonej, ale nie jest to warunek, który musi być bezwzględnie spełniony. Nasuwa się też pytanie: czy na wynik pomiaru ma również wpływ częstotliwość zasilania mostka? Jak widać, we wzorze (2) częstotliwość nie występuje, więc teoretycznie nie. Trzeba jednak pamiętać, że reaktancje obu kondensatorów są zależne od częstotliwości. W zależności od pojemności kondensatorów mostka i częstotliwości napięcia zasilającego będzie się zmieniał stosunek podziału napięć między kondensatorami i rezystorami. Największą precyzję regulacji uzyskamy, gdy napięcie zasilające będzie dzielone mniej więcej w połowie między kondensatorem a rezystorem jednej gałęzi mostka. Można więc rozpatrzyć wprowadzenie dodatkowego kroku pomiarowego, w którym po wstępnym zrównoważeniu mostka zostanie dobrana częstotliwość napięcia zasilającego tak, aby w napięciu punktach „a” i „b” było równe mniej więcej połowie napięcia zasilającego.

Pomiar, krok 1 – równoważenie mostka

Mamy już wszystkie dane, przystępujemy więc do pomiarów. Układ tradycyjnie montujemy na płytce stykowej (fotografia 2). Nie stosujemy mechanicznego przełącznika widocznego na schemacie. Zamiast niego po prostu przepniemy kabelek bezpośrednio na płytce do odpowiedniego styku. W pierwszym kroku potrzebny będzie oscyloskop i generator arbitralny. Narzędzia te uruchamiamy z zestawu Analog Discovery 2. W generatorze wybieramy przebieg sinusoidalny, ustawiamy częstotliwość 1 kHz i amplitudę 3 V (rysunek 4). W oscyloskopie natomiast wybieramy podstawę czasu 200 μs/dz, czułość kanału 1. 500 mV/dz, wyzwalanie z kanału 1. Uruchamiamy oba przyrządy i kręcąc potencjometrem R_w obserwujemy przebieg z kanału 1. Obracając suwak z jednego skrajnego położenia w drugie zaobserwujemy wyraźne minimum mierzonego napięcia. W miarę zbliżania się do tego minimum można stopniowo zwiększać czułość kanału 1. Niestety, całkowite zero jest trudne do uzyskania. W moim układzie uznałem, że mostek



Rysunek 7. Okno wprowadzania formuły obliczeniowej

jest zrównoważony, gdy napięcie wyglądało tak, jak na **rysunku 5**. W celu uzyskania większej precyzji podczas równoważenia mostka można włączyć uśrednianie. Ja zastosowałem uśrednianie 50 akwizycji, a jako zero przyjąłem minimum napięcia AC RMS mierzonego w kanale 1 oscyloskopu.

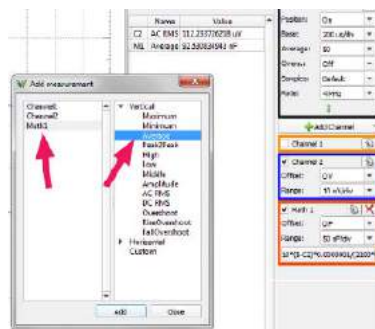
Można, a nawet należy zadać sobie pytanie, dlaczego nie jest możliwe uzyskanie idealnego zera? Prawdopodobnie jest to spowodowane różnicowym charakterem pomiaru. Najdrobniejsze przesunięcia fazowe wynikające choćby z pojemności układowych, mogą być przyczyną powstawania przebiegów o niezerowej amplitudzie. Nie przejmowałbym się jednak problemami tego typu, gdyż i tak nie spodziewam się dużej dokładności tego pomiaru.

Z zerowaniem mostka związany jest jeszcze zakres regulacji. Należało wybrać kompromis pomiędzy dokładnością i zakresem zerowania. Jak wynika z formuły (2) kondensator mierzony może być co najwyżej R_w/R_4 razy większy niż kondensator wzorcowy. W naszym przypadku jest to 2,27. Nie jest to dużo. Od dołu natomiast jest ograniczony możliwością wyznaczania małych rezystancji, co fizycznie oznacza pomiar małych prądów. Tu już zaczynają być istotne zagadnienia takie jak: rozdzielczość przetwornika A/C, szumy itp.

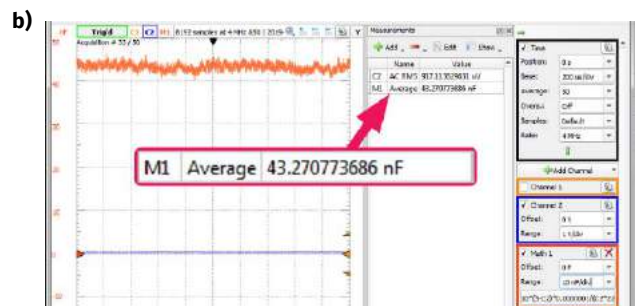
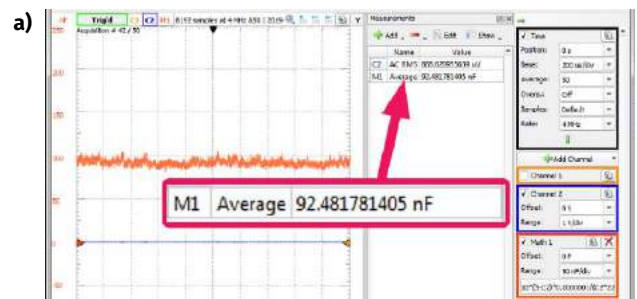
Pomiar, krok 2 – pomiar rezystancji R_w i obliczenie mierzonej pojemności

Teraz, gdy mostek jest już wyzerowany możemy przełączyć kabelek w pozycję „2”, co pozwoli zmierzyć prąd płynący przez potencjometr R_w , a następnie obliczyć jego rezystancję i na tej podstawie wyznaczyć pojemność badanego kondensatora. Musimy pamiętać o włączeniu źródła napięcia stałego ustawionego na 5 V. Zgodnie z obietnicą wprowadzamy też formułę, która pozwoli wyświetlić wynik bezpośrednio w jednostkach pojemności (w faradach). W tym celu klikamy prawym przyciskiem myszki w wolnym polu pod parametrami kanałów pomiarowych (**rysunek 6**). W wyświetlonym menu wybieramy opcję „Custom”. Pojawia się wówczas ramka „Math 1”, w której po kliknięciu na dolne pole wpisujemy wyznaczoną wcześniej formułę (4). Dla porządku ustalamy również jednostkę „F” (farad) w polu „Units” (**rysunek 7**). Pozostaje jeszcze zdefiniowanie pomiaru automatycznego, którym jest wartość średnia z kanału „Math 1”. W tak utworzonym polu będzie wyświetlana wartość mierzonej pojemności (**rysunek 8**). Na **rysunku 9a** pokazano rezultat pomiaru pierwszego kondensatora – ok. 92 nF, przy znamionowej pojemności 100 nF. Wynik całkiem poprawny, tym bardziej, że do obliczeń przyjęto znamionową pojemność kondensatora wzorcowego, a więc 100 nF, a nie jego rzeczywistą.

Teraz nie zmieniając kondensatora wzorcowego zmierzmy kondensator o znamionowej pojemności 47 nF. Powtarzamy dokładnie te same czynności. Okazuje się, że drugi kondensator ma pojemność ok. 43 nF (**rysunek 9b**). Wydaje się, że jest to też



Rysunek 8. Definiowanie pomiaru automatycznego podającego pojemność mierzonego kondensatora

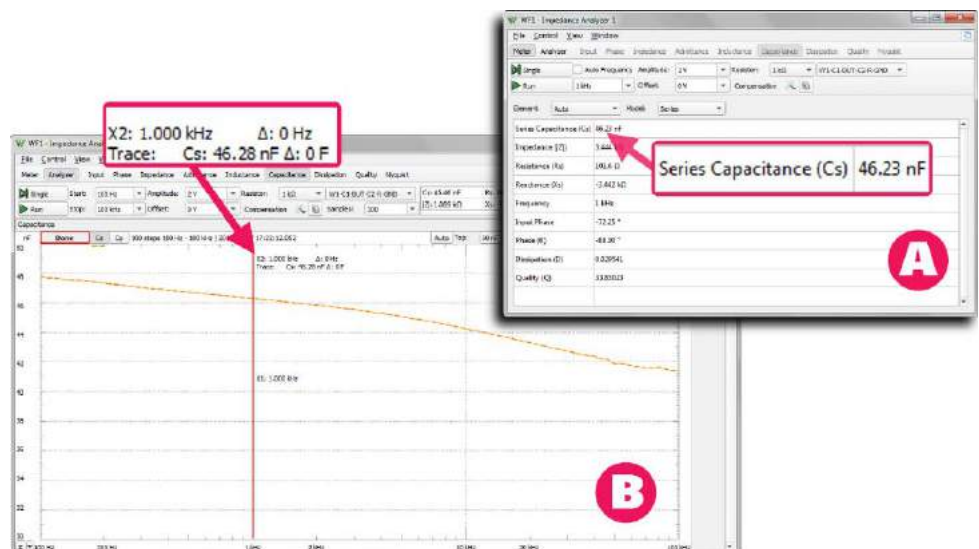


Rysunek 9. Tabela wyników pomiarów automatycznych zawierająca m.in. pojemność mierzonego kondensatora a) 100 nF, b) 47 nF

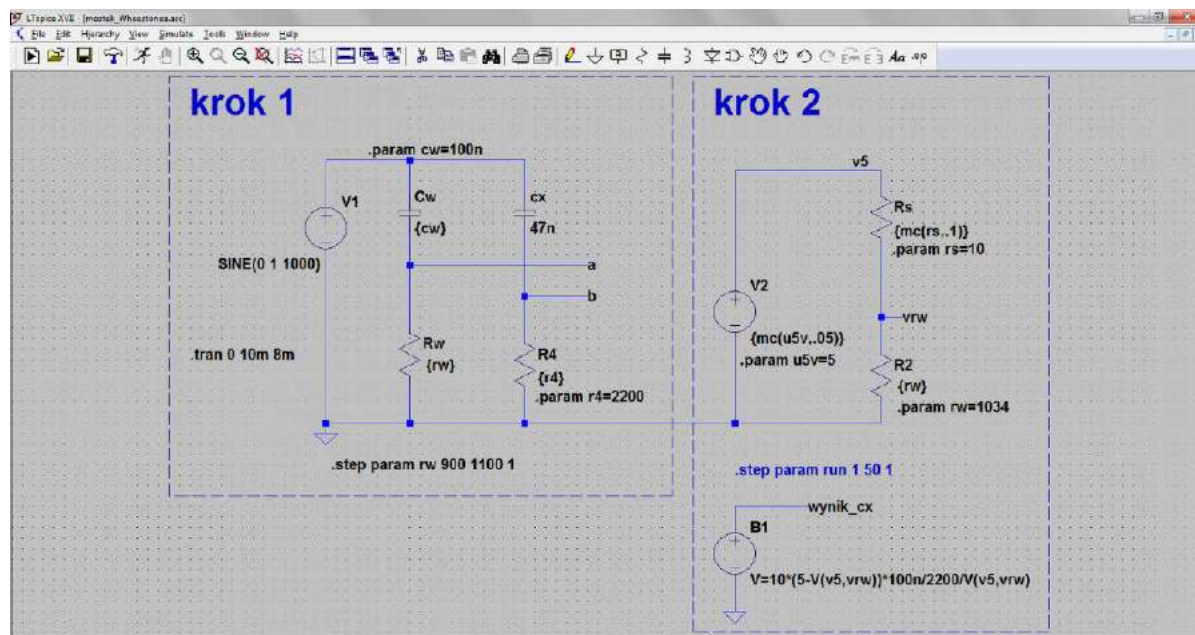
dość sensowna wartość, ale mimo to nie popadajmy w nadmierny optymizm. Zweryfikujmy teraz powyższe wyniki mierząc te same pojemności narzędziem *Impedance* dostępnym w Analog Discovery 2.

Pomiary pojemności narzędziem *Impedance* Analog Discovery 2

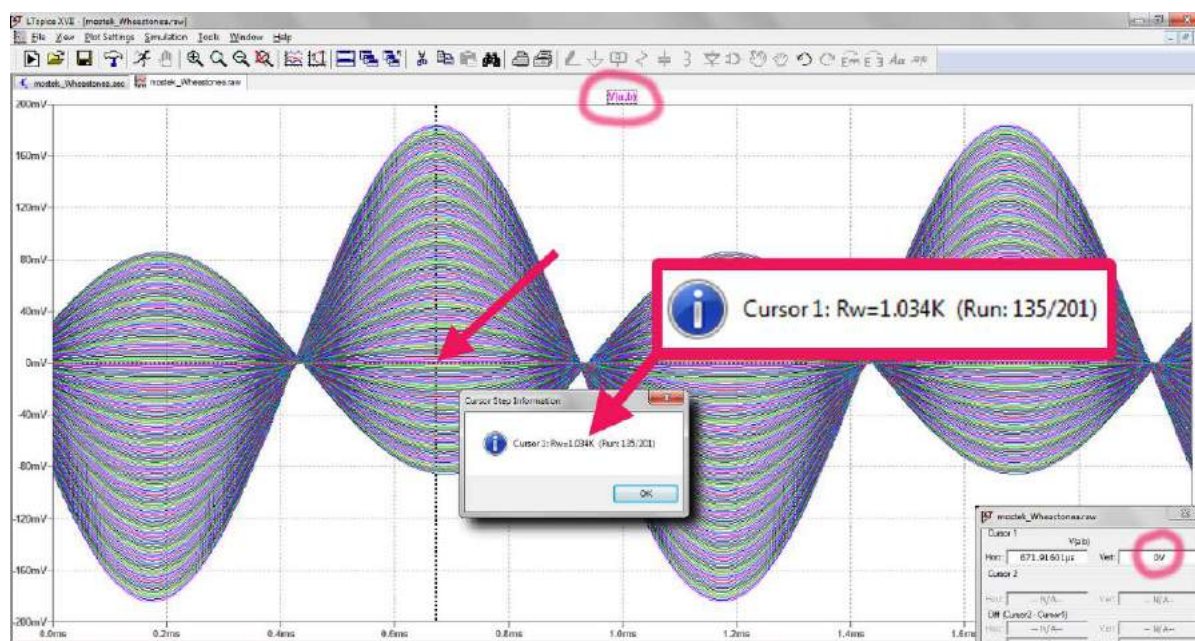
Impedance to narzędzie dodane do Waveforms 2015 jako jedno z ostatnich długo po premierze urządzenia, ale szybko okazało się, że jest ono bardzo przydatne w warsztacie elektronika. Jest przeznaczone do pomiaru niemal wszystkich parametrów impedancyjnych dwójników. Nie będziemy go teraz omawiać dokładnie, być może będzie jeszcze okazja do tego. *Impedance* użyjemy do szybkiego pomiaru pojemności kondensatorów, które wcześniej mierzyliśmy mostkiem. Do dyspozycji mamy dwa tryby pracy. W pierwszym z nich – „Meter” urządzenie mierzy parametry impedancyjne przy ustalonej częstotliwości. Pojemność kondensatora C_x (znamionowo 47 nF) zmierzona przy częstotliwości 1 kHz, a więc takiej samej jak w pomiarach mostkowych, była równa 46,23 nF (43,27 nF pomiar mostkiem) – **rysunek 10a**. Na tej podstawie można by przypuszczać, że pomiar mostkiem był jednak mniej dokładny, ale nie musi to być



Rysunek 10. Pomiar pojemności narzędziem *Impedance* Analog Discovery 2, a) w trybie „Meter”, b) w trybie „Analyzer”



Rysunek 11. Schemat układu pomiarowego w symulatorze LTspice



Rysunek 12. Przebiegi wyświetlane w pierwszym kroku symulacji w LTspice

prawda, gdyż w pomiarach mostkowych przyjęliśmy pojemność znamionową kondensatora wzorcowego, a nie rzeczywistą.

Na zakończenie pomiarów skorzystamy jeszcze z ciekawej cechy narzędzia *Impedance*, jaką jest pomiar parametrów elementów biernych w funkcji częstotliwości. Czasami wyniki mogą być bardzo zaskakujące. Szczególnie wtedy, gdy okaże się, że na przykład pojemność danego kondensatora dość drastycznie zmienia się w całym zakresie częstotliwości występujących w urządzeniu, w którym kondensator ma być zastosowany. Popatrzmy jak to jest w przypadku naszego kondensatora C_X o znamionowej pojemności 47 nF. Już wiemy, że przy 1 kHz jest równa 46,23 nF (pomiar AD2). Wynik takiego pomiaru pokazano na **rysunku 10b**. Jak widać kondensator 47 nF ma rzeczywistą pojemność 47,7 nF przy 100 Hz i 40,7 nF przy 100 kHz.

Przydatność metody

Już na wstępie artykułu wyraziłem wątpliwość odnoszącą się do praktycznej przydatności przedstawionej metody pomiarowej. Ponieważ nie było przeciwwskazań teoretycznych postanowiłem ją sprawdzić i zaprezentować rezultaty. Uzyskane wyniki były dość

sensowne, co mogłoby dawać nadzieję na praktyczną przydatność tej metody. Należałoby jednak przymknąć oko na dość złożone operacje manualne niezbędne do wykonania pomiaru.

Natomiast spory niepokój budzi złożona postać wyrażenia matematycznego, z którego wyznaczana jest pojemność mierzzonego kondensatora. Sprawdźmy, jak bardzo uzasadnione są te obawy. Poniższy fragment artykułu zapewne zniechęci część Czytelników do dalszej lektury, tych wytrwałych zachęcam jednak do uważnego przeczytania ze zrozumieniem. W wyrażeniu, z którego jest wyznaczana pojemność kondensatora występuje aż 5 zmiennych mających wpływ na wynik końcowy. Oszacujmy więc, z jakim błędem powinniśmy się liczyć wykonując przedstawiony pomiar. Przyjmujemy więc, że formuła obliczeniowa jest funkcją 5 zmiennych (3), dla każdej z nich należy więc obliczyć pochodną cząstkową. Mamy zatem:

$$\frac{\partial C_X}{\partial R_S} = \frac{(V_+ - U_{CH2})C_W}{U_{CH2}R_4}$$

$$\frac{\partial C_X}{\partial V_+} = \frac{\partial}{\partial V_+} \left(\frac{R_S \cdot V_+ \cdot C_W}{U_{CH2}R_4} - \frac{R_S \cdot C_W}{R_4} \right) = \frac{R_S \cdot C_W}{U_{CH2}R_4}$$

$$\frac{\partial C_X}{\partial U_{CH2}} = -\frac{R_S \cdot V_+ \cdot C_W}{R_4 \cdot U_{CH2}^2}$$

$$\frac{\partial C_X}{\partial C_W} = \frac{R_S (V_+ - U_{CH2})}{R_4 \cdot U_{CH2}}$$

$$\frac{\partial C_X}{\partial R_4} = -\frac{R_S (V_+ - U_{CH2}) \cdot C_W}{R_4^2 \cdot U_{CH2}}$$

Przyjmujemy teraz, że poszczególne parametry wraz z tolerancjami są równe:

$$R_W = 10 \Omega \pm 10\% \quad U_{CH2} = 49,5 \text{ mV} \pm 5\% \quad R_4 = 2200 \Omega \pm 10\%$$

$$V_+ = 5 \text{ V} \pm 5\% \quad C_W = 100 \text{ nF} \pm 20\%$$

Całkowity błąd bezwzględny pomiaru pojemności kondensatora C_X jest równy różniczkę zupełnej:

$$\Delta C_X = \left| \frac{\partial C_X}{\partial R_S} \right| R_S \cdot \delta R_S + \left| \frac{\partial C_X}{\partial V_+} \right| V_+ \cdot \delta V_+ + \left| \frac{\partial C_X}{\partial U_{CH2}} \right| U_{CH2} \cdot \delta U_{CH2} +$$

$$\left| \frac{\partial C_X}{\partial C_W} \right| C_W \cdot \delta C_W + \left| \frac{\partial C_X}{\partial R_4} \right| R_4 \cdot \delta R_4$$

Podstawiając przyjęte parametry do powyższego wyrażenia okazuje się, że błąd bezwzględny pomiaru C_X jest równy... $\pm 22,8 \text{ nF}$ (sic!), co w przypadku kondensatora 47 nF oznacza błąd względny 48%.

Weryfikacja metody pomiarowej w symulatorze LTspice

Właściwie powinniśmy taką symulację przeprowadzić na początku, aby sprawdzić czy zaproponowana metoda pomiarowa ma w ogóle jakiś sens. Pomiar na szczęście nie był bardzo skomplikowany, więc za dużo czasu nie straciliśmy.

Schemat wirtualnego pomiaru pokazano na **rysunku 11**. Symulacja, tak jak pomiar rzeczywisty przebiega w dwóch etapach. W pierwszym kroku bazujemy na lewej części schematu opisanej „krok 1”. Wartości elementów są podane parametrycznie, wyjątkiem jest kondensator mierzony C_X o pojemności 47 nF . Kondensator wzorcowy C_W ma pojemność 100 nF , a mostek jest zasilany napięciem sinusoidalnym o amplitudzie 3 V i częstotliwości 1000 Hz . Rezystor R_W odpowiadający na tym schemacie potencjometrowi z pomiaru rzeczywistego ma parametrycznie zadaną rezystancję r_w . Będzie ona krokowo zmieniana w trakcie symulacji komendą:

.step param rw 900 1100 1

Oznacza to, że mostek będziemy równoważyć w zakresie zmian R_W od 900 do 1100 omów z krokiem 1Ω . Aby wyeliminować ewentualne stany nieustalone po wirtualnym uruchomieniu układu podajemy wyniki ostatnich 2 milisekund analizy spośród 10 , którymi objęta jest cała symulacja. Zapewnia to komenda:

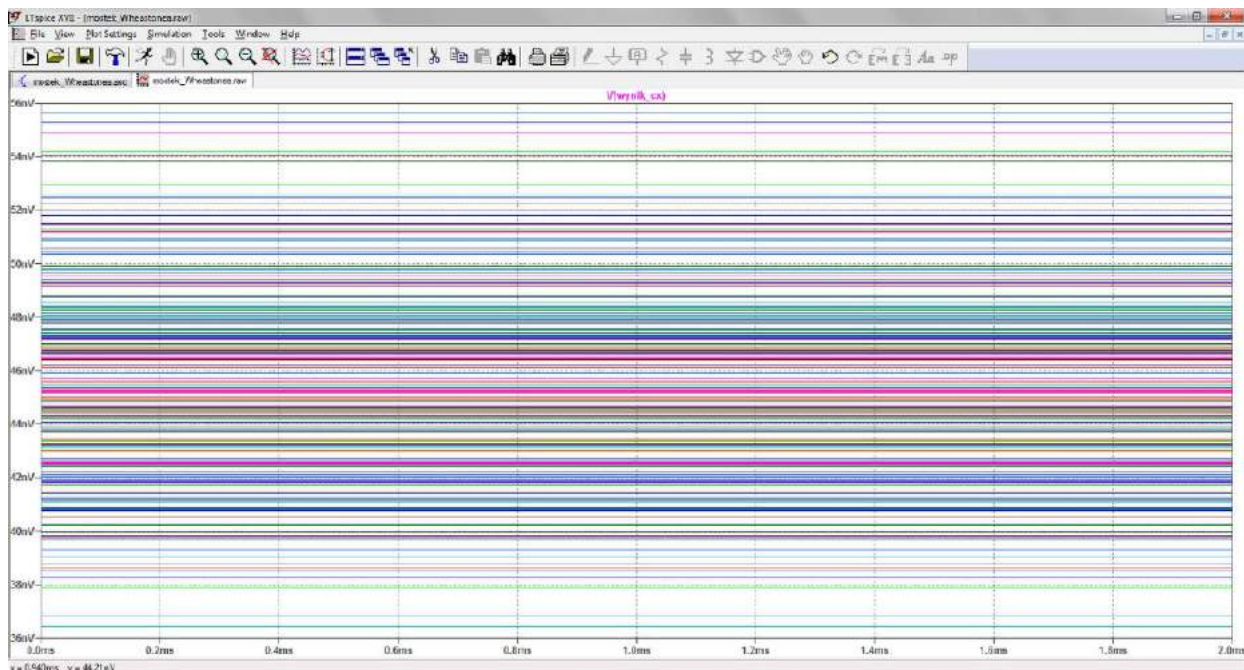
.tran 0 10m 8m

Symulacja rozpoczyna się po wydaniu polecenia *Simulate* → *Run* (ikona z grafiką biegacza). Zostaje wyświetlone puste okno, w którym będzie umieszczony wykres wskazanego przebiegu. Interesuje nas wyjście mostka, czyli punkty „a” i „b”. Naprowadzamy więc kursor na linię „a”, naciskamy prawy przycisk myszki i przytrzymując go przesuwamy kursor na linię „b”, po czym zwalniamy przycisk. W oknie wykresu pojawia się rodzina przebiegów (**rysunek 12**). Klikamy na tytuł wykresu („V(a,b)” nad wykresem), w wyniku czego pojawia się kursor pomiarowy i tabela opisująca jego parametry. Teraz przesuwamy ten kursor tak, aby znalazł się nad jednym z maksimum wykresu i za pomocą przycisków „strzałka w górę”, „strzałka w dół” ustawiamy go do takiej pozycji, w której napięcie jest równe zero lub osiąga wartość jak najbliższą zero. Po kliknięciu prawym przyciskiem na pionową kreskę kursora pomiarowego zostaje wyświetlone okno „Cursor Step Information”, z którego odczytujemy, że mostek został wyzerowany dla $R_W = 1034 \Omega$ (rysunek 12). Poszło gładko, ale w ogólnym przypadku może się zdarzyć, że mostka nie uda się wyzerować. Należy wówczas zwiększyć zakres zmian R_W kosztem wydłużenia czasu symulacji i zagęszczenia wykresu. Pamiętajmy ponadto, że dla wartości elementów przyjętych w układzie, tzn. pojemności oporności potencjometru $R_W = 5 \text{ k}\Omega$ maksymalna pojemność mierzona, dla której uda się wyzerować mostek jest równa ok. 227 nF . Można to łatwo sprawdzić w symulacji lub obliczyć, do czego zachęcam już samodzielnie Czytelników.

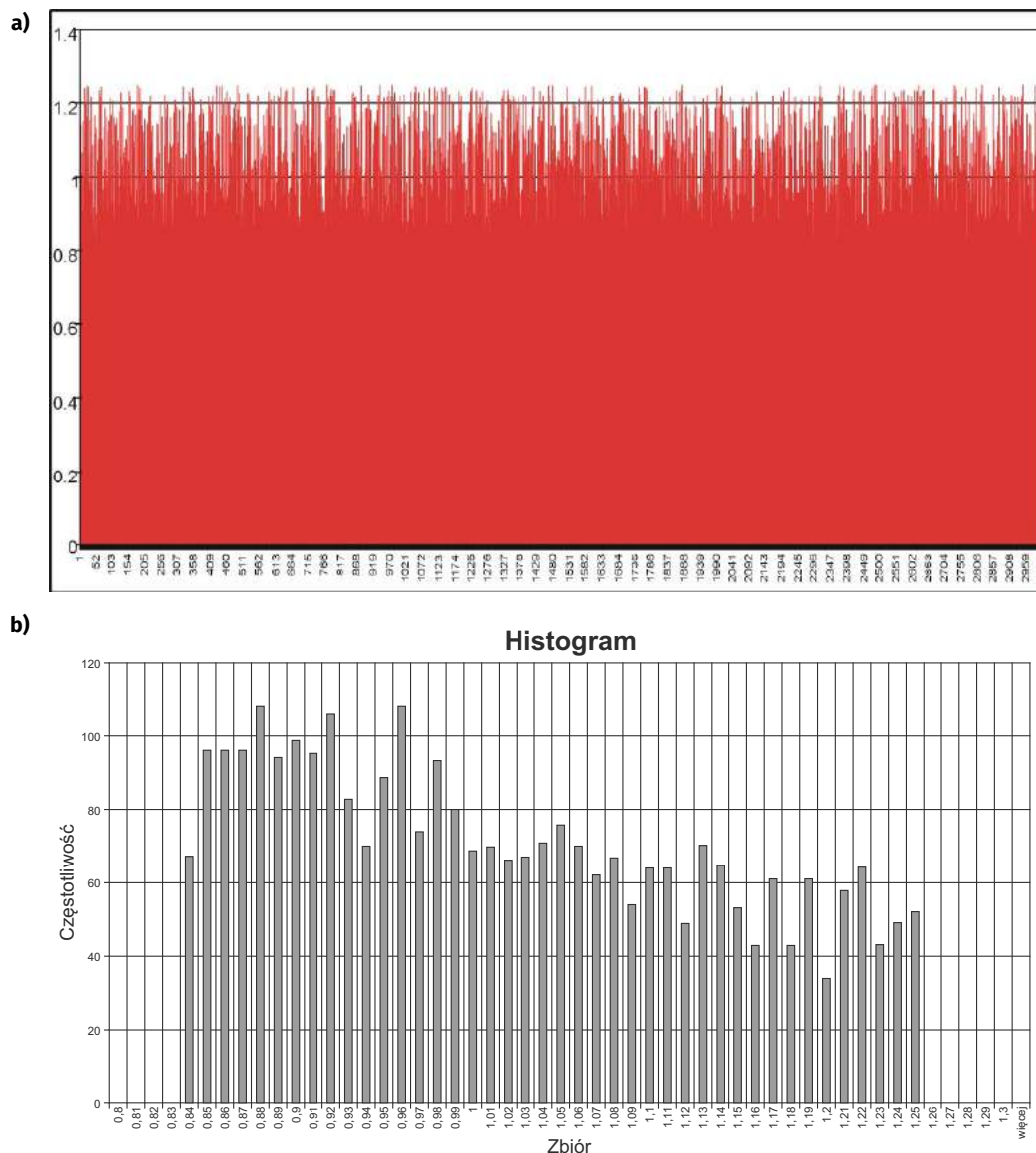
Wracamy do naszych pomiarów. Przechodzimy do kroku 2. Na wstępie należy zakomentować polecenie krokowej zmiany R_W z 1. etapu i odkomentować polecenie:

.step param run 1 50 1

Oznacza ono uruchomienie kolejno następujących po sobie 50 symulacji, co będzie potrzebne do zgromadzenia danych dla pomiarów wykonywanych z zastosowaniem metody *Monte Carlo*. Polega ona na losowej zmianie wartości wskazanych parametrów w każdej kolejnej symulacji, a zatem zbadanie rozrzutu wyników, jakich można będzie spodziewać się w układzie fizycznym. Rezystor R_W



Rysunek 13. Wyniki symulacji metodą Monte Carlo



Rysunek 14. Badanie rozkładu liczb losowych generowanych komendą mc, a) wartości kolejnych próbek, b) rozkład generowanych liczb

powinien dokładnie odpowiadać rezystorowi R_w z kroku 1, więc musimy pamiętać o wprowadzeniu jego rezystancji 1034 Ω (param $rw=1034$). Rozrzut wyników jest spowodowany rozrzutem takich parametrów, jak: rezystancja rezystora R_s i napięcie zasilające V_2 . Dlatego parametry te zapisujemy w postaci:

```
{mc(rs,.1)}
```

```
i
```

```
{mc(u5v,.05)}
```

oraz definiujemy rs i $u5v$:

```
.param rs=10
```

```
.param u5v=5
```

Powyższy zapis oznacza, że rezystor R_s będzie miał rezystancję 10 Ω z tolerancją $\pm 10\%$ (zapis .1). A napięcie zasilające będzie miało 5 V $\pm 5\%$.

Dla zobrazowania wyników w postaci graficznej dość nietypowo użyjemy programowanego źródła napięciowego (*Arbitrary behavioral voltage source*), które w standardowej bibliotece elementów LTspice'a ma symbol „bv”. Ponieważ jest ono definiowane należy w polu „Value” w jego właściwościach wpisać formułę, którą stosowaliśmy w rzeczywistym układzie do obliczania pojemności mierzonej, a więc:

$$V=10 \cdot (5 - V(v5, vrw)) \cdot 100n / 2200 / V(v5, vrw)$$

Zwróćmy uwagę na to, że w wyrażeniu tym są zapisane znamionowe wartości elementów R_s i V_2 , a nie rzeczywiste. Zakres zmian zmierzonej pojemności C_x jest widoczny w oknie wykresu

po kliknięciu na linię opisaną etykietą „wynik_cx”. Do zobrazowania użyto źródła napięciowego, więc oś pionowa jest opisana w voltach nie w faradach, na co przysmykamy oko (**rysunek 13**).

Problem z Monte Carlo

Metoda *Monte Carlo* to bardzo ciekawe narzędzie symulatora LTspice przewidziane do symulacji rozrzutu wybranych parametrów badanego układu wynikających np. z tolerancji. Wszystko było by dobrze, gdyby nie jedna ewidentna niedoróbka symulatora. Problem polega na tym, że program generuje wartości losowe z rozkładem... zbliżonym do równomiernego, ale też dalekim od ideału. Pozwoliłem sobie wykonać mały test. Wykonałem symulację z elementem o nominalnej wartości 1 i tolerancji 20%. Dla tego testu nie jest istotna fizyczna interpretacja elementu. Symulator wygenerował 3000 próbek. Proces ten pokazano na **rysunku 14a**. Widoczne są na nim wartości kolejno losowanych próbek. Następnie korzystając z możliwości Excela zbadałem rozkład wylosowanych wartości (**rysunku 14b**). Hmm... trudno nazwać go równomiernym, z całą pewnością nie jest to też rozkład normalny, któremu podlega rozrzut parametrów elementów fizycznych.

Jest to temat, któremu warto się jeszcze przyjrzeć, co prawdopodobnie nastąpi w którymś z kolejnych odcinków. A w następnym temat bardziej „lajtowy”, po tym nieco przydługim. Zajmiemy się dopasowaniem mocy.

Jarostaw Doliński, EP

niezbędnika elektronika

1. Intel Quartus Prime 19.3

Rewolucyjne oprogramowanie Intel Quartus Prime Design zawiera wszystko, czego potrzebujesz do projektowania układów Intel FPGA, SoC i złożonych programowalnych urządzeń logicznych (CPLD), od wprowadzania projektu i syntezy do optymalizacji, weryfikacji i symulacji. Znacząco zwiększone możliwości urządzeń z wielomilionowymi elementami logicznymi zapewniają projektantom idealną platformę do spełnienia możliwości projektowych nowej generacji.

2. NXP i.MX Pins Tool

Narzędzie Pins dla procesorów i.MX jest następcą oprogramowania Processor Expert. Nowe narzędzie Pins sprawia, że konfiguracja pinów jest łatwiejsza i szybsza dzięki intuicyjnemu i łatwemu in. erfejsowi użytkownika, który następnie generuje normalny kod C, który można następnie wykorzystać w dowolnej aplikacji C i C++. Narzędzie Pins konfiguruje sygnały pinów multiplexowanych, a także tworzy pliki wycinków drzewa urządzeń (.dtsi) i raporty w formacie CSV.

3. ST-STM32CubeIDE

Zaawansowana platforma programistyczna C/C++ z konfiguracją urządzeń peryferyjnych, generowaniem kodu, kompilacją kodu i funkcjami debugowania dla mikrokontrolerów i mikroprocesorów STM32. Opiera się na frameworku Eclipse/CDT i łańcuchu narzędzi GCC do programowania oraz GDB do debugowania. Umożliwia integrację setek istniejących wtyczek, które uzupełniają funkcje Eclipse IDE. STM32CubeIDE integruje wszystkie funkcjonalności STM32CubeMX, oferując kompleksowe działanie narzędzi oraz oszczędzając czas instalacji i programowania

4. TI MSP430 tools

Pakiet zawiera kompletne narzędzia do debugowania i kompilatora C/C++ typu open source do aplikacji opartych na mikrokontrolerach MSP430. Te bezpłatne kompilatory GCC obsługują wszystkie układy MSP430 i MSP432 bez ograniczeń rozmiaru kodu. Mogą być używane samodzielnie z wiersza poleceń lub poprzez oprogramowanie Code Composer Studio (również jest zawarty w pakiecie). Uzupełnieniem narzędzi są pliki przykładowych programów oraz manuały i tutoriały.

EP 10/2019

1. Lattice Radiant Software 1.1
Oferuje wszystkie najlepsze w swojej klasie narzędzia i funkcje, które pomagają użytkownikom efektywnie i wydajnie tworzyć aplikacje oparte na układach FPGA firmy Lattice

2. CodeWarrior Development Studio for Microcontrollers v11.1
Zintegrowane środowisko programistyczne (IDE) od NXP Semiconductors

3. LPCpresso v8.2.2
Zintegrowane środowisko programistyczne (IDE) od NXP bazujące na Eclipse

4. S32 Design Studio for Power Architecture v2.1
Bezpłatne zintegrowane środowisko programistyczne (IDE) dla motoryzacyjnych i bardzo niezawodnych mikrokontrolerów opartych na architekturze Power Architecture (rdzeń e200) i ARM

5. ST-TrueStudio-9.3.0
Elastyczne i rozszerzalne środowisko, z wbudowanymi funkcjami edycji, kompilacji i debugowania kodu dla programistów mikrokontrolerów STM32

EP 9/2019
1. ADI-LTSpice-XVII-9-2019
Program do projektowania i testowania analogowych i mieszanych układów elektronicznych.

2. AutotraxDEX-2019-8.09
Pakiet narzędzi do projektowania płytek drukowanych.

3. Intel Quartus-Prime-19.2
Bezpłatny pakiet oprogramowania przeznaczony do projektowania FPGA firmy Intel.

4. STM32CubeMX-5.3.0
Narzędzie ułatwiające konfigurację mikrokontrolerów STM32 za pomocą interfejsu graficznego.

EP 8/2019

1. Infineon-DAVE-4.4.2
Pakiet programistyczny DAVE firmy Infineon, w wersjach dla Win64 i Win32. Generuje kod dla najnowszych mikrokontrolerów XMCCxxx i XMCCxxx.

2. Lattice Semiconductor-Diamond-FPGA
Kompletne środowisko projektowania oparte na GUI, dla układów FPGA firmy Lattice.

3. Microchip MPLAB Harmony
Najnowsza wersja środowiska programistycznego firmy Microchip wspierająca programowanie mikrokontrolerów z rdzeniem PIC32 tej firmy

4. NXP-i.MX-Pin-Tool
Konfigurator graficzny przeznaczony dla procesorów i.MX.

EP 7/2019

1. ADI-LTSPICE
LTSpice to program do projektowania i testowania analogowych i mieszanych układów elektronicznych.

2. ADI-SigmaStudio
Graficzne narzędzie programistyczne dla procesorów SigmaDSP. Pozwala inżynierom bez dotąd zdefiniowania w pisaniu kodu DSP na łatwiejsze wdrożenie DSP do swoich projektów.

3. CodeWarrior for QorIQ LS Series ARMv8 ISA
Środowisko programistyczne (IDE) opracowane przez Freescale dla procesorów serii QorIQ LS. Układy te oferują wyjątkową wydajność obliczeniową i energetyczną, dedykowane są do aplikacji sieciowych.

4. Infineon-ModusToolbox
Nowy pakiet narzędzi programistycznych opracowanych przez Cypress, który ma usprawnić projektowanie produktów dla Internetu Rzeczy.

5. Infineon-PSOC-Software
Pakiet narzędzi programistycznych opracowanych przez Cypress, który ma pomóc w pełnym wykorzystaniu możliwości układów PSOC. Zawiera PSOC Creator, PSOC Designer i PSOC Programmer.

6. SiliconLabs-SimplicityStudio4
Simplicity Studio to zintegrowane środowisko programistyczne (IDE) oparte na Eclipse.

Zawiera zestaw narzędzi do profilowania energii, konfiguracji i analizy sieci bezprzewodowej.

7. TI-Code Composer Studio Version 9
Code Composer Studio to zintegrowane środowisko programistyczne (IDE), które obsługuje portfolio mikrokontrolerów TI i procesorów wbudowanych.

EP 6/2019
1. Labcenter-Proteus
Pakiet oprogramowania EDA, zawiera aplikacje wspomagające tworzenie schematów, projektowanie płytek i narzędzia do symulacji.

2. NXP-MCUpresso-Config-Tools

Oprogramowanie służące do konfiguracji mikrokontrolerów zamontowanych na płytkach MCUpresso.

3. NXP-MCUpresso-IDE
Środowisko IDE MCUpresso oparte na Eclipse dla mikrokontrolerów NXP w tym mikrokontrolerów LPC.

4. ST-STM32CubeIDE
Konfigurator graficzny wspomagający programowanie mikrokontrolerów STM32.

5. ST-TouchGFXDesigner
Oprogramowanie ułatwiające tworzenie interfejsu graficznego w systemach z STM32.

6. ST-TrueStudio-9.2.0
Elastyczne i rozszerzalne środowisko, z funkcjami edycji, kompilacji i debugowania dla mikrokontrolerów STM32.

EP 05/2019
1. FlexGUI Mixed Signal Audio Software
Oprogramowanie firmy Cirrus Logic przeznaczone do obsługi płytek ewaluacyjnych do budowania zaawansowanych torów obróbki sygnałów audio.

2. WISCE Development Tools Suite
Narzędzie programowe firmy Cirrus Logic służące do wspierania procesów konstrukcji oraz produkcji urządzeń.

3. Intel Quartus Prime Pro Edition 19.1.0.240
Bezpłatny pakiet oprogramowania, środowisko projektowe dla układów FPGA.

4. EasyEDA version 3.0.11
Program do projektowania i symulowania obwodów oraz projektowania płytek drukowanych.

EP 4/2019

1. Atollic TrueSTUDIO for STM32
Znany pakiet IDE do tworzenia programów dla mikrokontrolerów STM32. Zawiera kompilator, debugger, linker. Nowa wersja mimo sporych zmian w organizacji i przeznaczeniu samego oprogramowania dodaje też istotną nowość: jak się dostać do poziomu menu budowania do opcji dezasmblacji pliku oraz listowania symboli w plikach binarnych i obiektowych.

2. Siemens LOGO! Web Editor
Najnowsza wersja edytora stron internetowych dla sterownika LOGO! v8

3. Analog Devices LTspice XVII
Najnowsza wersja popularnego symulatora układów elektronicznych, teraz rozwijana przez firmę Analog Devices.

4. Siemens LOGO!Soft Comfort v8.1
Najnowsza wersja środowiska projektowego przeznaczającego dla sterowników LOGO! firmy Siemens.

5. WaveDrom Editor v1.6.3
WaveDrom tworzy wykres czasowy lub przebieg na podstawie opisu tekstowego. Środowisko zawiera język opisu, silnik renderujący i edytor. Działa w przeglądarce lub może być zainstalowany lokalnie.

6. Tina 9 TI
Symulator układów elektronicznych wspierany i rozwijany przez firmę Texas Instruments.

7. TI Code Composer Studio 8.3 for Windows
Pakietu programów przeznaczonych do tworzenia aplikacji dla procesorów firmy Texas Instruments.

8. NXP Pins Tool for i.MX Processors
Konfigurator graficzny przeznaczony dla procesorów i.MX.

9. NXP MCUpresso IDE
Środowisko IDE MCUpresso oparte na Eclipse z wbudowanymi funkcjami edycji, kompilacji i debugowania kodu.

EP 03/2019
1. Atollic TrueSTUDIO for STM32
Pakiet IDE do tworzenia programów dla mikrokontrolerów STM32.

2. Siemens LOGO! Web Editor
Edytor stron internetowych dla sterownika LOGO! v8

3. Analog Devices LTspice XVII
Popularny symulator układów elektronicznych, teraz rozwijany przez firmę Analog Devices.

4. Siemens LOGO!Soft Comfort v8.1
Środowisko projektowe przeznaczone dla sterowników LOGO! firmy Siemens.

5. WaveDrom Editor v1.6.3
WaveDrom tworzy wykres czasowy lub przebieg na podstawie opisu tekstowego.

6. Tina 9 TI
Symulator układów elektronicznych wspierany i rozwijany przez firmę Texas Instruments.

7. TI Code Composer Studio 8.3 for Windows
Pakietu programów przeznaczonych do tworzenia aplikacji dla procesorów

firmy Texas Instruments.

8. NXP Pins Tool for i.MX Processors
Konfigurator graficzny przeznaczony dla procesorów i.MX.

9. NXP MCUpresso IDE
Środowisko IDE MCUpresso oparte na Eclipse z wbudowanymi funkcjami edycji, kompilacji i debugowania kodu.

EP 02/2019

1. SiFive Freedom Studio
Zintegrowane środowisko programistyczne przeznaczone do tworzenia programów dla mikrokomputerów SiFive RISC-V.

2. SiFive RISC V OpenOCD GCC
Wtyczka dla popularnego edytora Eclipse umożliwiająca tworzenie programów dla mikrokomputerów SiFive RISC-V.

3. SiFive Segger RISC V Embedded Studio
Środowisko programistyczne firmy Segger pozwalające na tworzenie i uruchamianie programów dla mikrokomputerów SiFive RISC-V.

4. NXP Pins Tool for i.MX Application Processors
Konfigurator graficzny przeznaczony dla procesorów i.MX.

5. ST TouchGFX Designer 4.10
Oprogramowanie ułatwiające tworzenie interfejsu graficznego w systemach embedded wykorzystujących mikrokontrolerów STM32.

EP 1/2019

1. Analog Devices - ADALM2000
Oprogramowanie typu „przenośne laboratorium” dla modułu ADALM2000 firmy Analog Devices.

2. ST STM32CubeMX release 5.0.0
Konfigurator graficzny mikrokontrolerów STM32.

3. ST TouchGFX
Zaawansowane rozwiązanie graficznego interfejsu użytkownika dla mikrokontrolerów. Uruchamiany na 32-bitowych mikrokontrolerach, TouchGFX zapewnia wysokiej klasy grafikę.

4. SEGGER Embedded Studio for ARM Version 4.12
IDE przeznaczona do tworzenia programów dla mikrokontrolerów z rdzeniem ARM zawierająca nowy linker i bibliotekę LTO.

5. ST Atollic TrueSTUDIO for STM32 9.2
Znany pakiet IDE do tworzenia programów dla mikrokontrolerów STM32.

6. NXP MCUpresso IDE
Środowisko IDE MCUpresso oparte na Eclipse z wbudowanymi funkcjami edycji, kompilacji i debugowania kodu.

7. NXP MCUpresso Config Tools
Pakietu programów służących do konfiguracji mikrokontrolerów zamontowanych na płytkach MCUpresso. Generuje kod w języku C, który może być wykorzystany w tworzonej aplikacji.

8. Espressif Systems ESP32 IDF
Najnowsza wersja środowiska programistycznego dla układów ESP32 firmy Espressif.

9. FTDI EVE Tools
Pakiet programów narzędziowych przeznaczonych dla układu graficznego FT800.

10. FTDI FT9XX Tools
Pakiet programów narzędziowych przeznaczonych do tworzenia aplikacji dla układów z rodziny FT9XX, serii mikrokontrolerów z 32-bitowym rdzeniem RISC FT32 oraz układów Vinculum.

EP 12/2018

1. Analog Devices SigmaStudio 4.2
Program firmy Analog Devices przeznaczony do graficznego tworzenia aplikacji dla procesorów DSP tego producenta.

2. Silicon Labs Simplicity Studio v4
Pakietu programów wspomagających tworzenie aplikacji dla mikrokontrolerów firmy Silicon Labs.

3. Cypress PSoC Creator 4.2
Środowisko programistyczne firmy Cypress przeznaczone do tworzenia aplikacji dla mikrokontrolerów PSoC.

4. ST Atollic TrueSTUDIO 9.1.0
Zintegrowane środowisko programistyczne dla mikrokontrolerów z rdzeniem ARM.

5. Cypress WICED Studio 6.2.1
Platforma do tworzenia oprogramowania dla urządzeń IoT.

6. TI TINA 9
Symulator układów elektronicznych wspierany i rozwijany przez firmę Texas Instruments.

7. TI Code Composer Studio 8
Pakietu programów przeznaczonych do tworzenia aplikacji dla procesorów firmy Texas Instruments.

EP 11/2018

1. Microchip MPLAB X IDE v5.05
Środowisko programistyczne firmy Microchip wspierające programowanie mikrokontrolerów z rdzeniami dsPIC, PIC i AVR.

2. Analog Devices LTspice XVII
Popularny symulator układów elektronicznych rozwijany przez firmę Analog Devices.

3. ST AlgoBuilder
Aplikacja pozwalająca na graficzne projektowanie rozwiązań w oparciu o różne algorytmy sterowania i kontroli.

4. Intel Quartus Prime Pro Edition 18.1.0.222
Bezpłatny pakiet oprogramowania przeznaczony do projektowania FPGA.

5. Infineon DAVE v4.4.2
Pakiet programistyczny DAVE firmy Infineon, w wersjach dla Win32 i Win64, z bibliotekami Peripheral Library.

6. ST X-CUBE-MEMS-XT1
Biblioteka przeznaczona dla mikrokontrolerów STM32. Umożliwia współpracę z układami: A3G4250D, AIS328BD, AIS3624DQ, H3LIS331DL, HTS221, IIS-2DLP, IIS2MDC, ISM303DAC, ISM330DL, LIS2DH12, LIS2DH12, LIS2MDL, LIS3MDL, LIS2ZH, LPS22HB, LPS25HB, LPS33HW, LSM303AGR, LSM6DSO, LSM6DSX, LSM6DSL, LSM6DSO, LSM6DSR oraz LSM6DSL.

7. ST UNICLO-GUI
Interfejs graficzny dla rozszerzeń programowych X-CUBE-MEMS1 i X-CUBE-MEMS-XT1 oraz płytki STM32 Nucleo (X-NUCLEO-IKS01A1 i X-NUCLEO-IKS01A2).

EP 10/2018

1. NXP Pins Tool for i.MX Processors
Intuicyjny interfejs graficzny przyspieszający i ułatwiający konfigurowanie peryferiów oraz wyprowadzenie procesorów z rodziny i.MX, generujący kod źródłowy z konfiguracją dla kompilatorów C/C++.

2. Microchip MPLAB Harmony Integrator Software Framework v2.06
Najnowsza wersja środowiska programistycznego firmy Microchip wspierająca programowanie mikrokontrolerów z rdzeniem PIC32 tej firmy.

3. Microchip MPLAB X IDE v5.05
Najnowsza wersja środowiska programistycznego firmy Microchip wspierająca programowanie mikrokontrolerów z rdzeniami dsPIC, PIC i AVR.

1. Lattice Semiconductor Diamond-3.10
IDE dla układów FPGA firmy Lattice.

2. Lattice Semiconductor LMS for Diamond 3.10
Środowisko programistyczne, zawierające dwa narzędzia. Jedno generuje plik HDL z opisem sprzętu, a drugie umożliwia tworzenie uruchamianego na nim oprogramowania.

3. Lattice Radiant Software 1.0
IDE wspierające tworzenie oprogramowania dla układów FPGA iC4E40 UltraPlus.

4. ST Atollic TrueSTUDIO for STM32 9.0.1
Bezpłatne, zintegrowane środowisko programistyczne dla mikrokontrolerów STM32.

5. SEGGER Embedded Studio for ARM 3.4.0
IDE firmy Segger służącego do tworzenia oprogramowania dla mikrokontrolerów z rdzeniem ARM.

EP 7/2018

1. AutotraxDEX-2018.619
Pakiet do projektowania płytek drukowanych. Program w wersji demonstracyjnej, która pomimo pełnej funkcjonalności nie tworzy plików produkcyjnych w formacie Gerber.

2. CADSTAR-Express-2018
Pakiet programów firmy Zuken do projektowania płytek drukowanych. Wersja demonstracyjna, ograniczona do 300 pinów i 50 komponentów.

3. ST-MCU-FINDER-PC
Wyszukiwarka mikroprocesorów z rodziny STM8 i STM32 oraz przeznaczonych do nich narzędzi deweloperskich.

4. ST-STM32Cube-4.26.0
Konfigurator graficzny wspomagający programowanie urządzeń zbudowanych z użyciem mikrokontrolerów STM32.



Zasilacze LED do zadań specjalnych

Większość z nas, czytając artykuł zatytułowany „zasilacz LED”, ma na myśli powszechnie stosowane modele w metalowych/plastikowych obudowach umieszczane bezpośrednio w oprawie lub też zabudowane ścianką kartonowo-gipsową. Tymczasem wraz z rozwojem technologii poszerza się obszar zastosowań diod LED. Stosowane są nie tylko w celach oświetleniowych czy dekoracyjnych, ale również w aplikacjach przemysłowych np. w laserach dużej mocy, naświetlaczach UV, itd. Powoduje to również konieczność ciągłego rozwoju urządzeń zasilających. W poniższym artykule krótko opisano kilka niestandardowych przykładów zasilaczy LED.

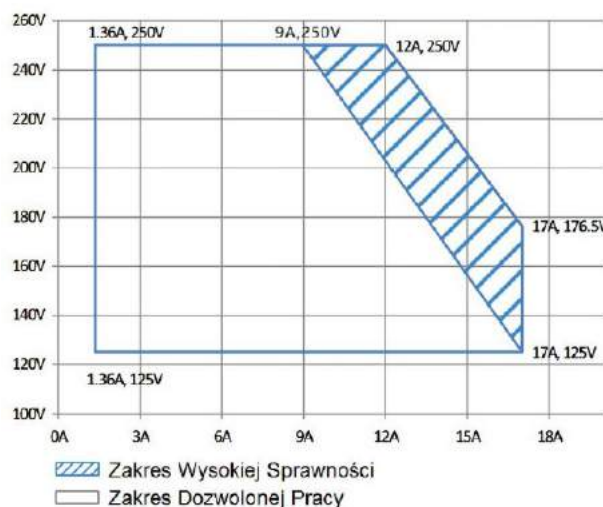
Więcej informacji:

ELMARK Automatyka S.A.
05-075 Warszawa, ul. Niemcewicza 76
tel. 22 541 84 60 wew. 1
tel. 22 773 79 37, 22 778 99 25
zasilacze@elmark.com.pl, www.elmark.com.pl



CSP-3000

Pierwszy niestandardowy zasilacz, o którym warto wspomnieć, jest przeznaczony właśnie do aplikacji przemysłowych, takich jak wspomniany wcześniej laser dużej mocy czy naświetlacz UV. Model CSP-3000, to aż 3 kW mocy przy dostępnych wariantach napięciowych 120/250/400 VDC. Każdy z modeli pozwala na regulację napięcia wyjściowego, dzięki czemu sumaryczny zakres napięciowy 3 modeli



Rysunek 1. Charakterystyka prądowo-napięciowa modelu CSP-3000-250

zawiera się w przedziale od 90 do 400 VDC. Dodatkowo producent przewidział funkcję regulacji prądu wyjściowego oraz możliwość łączenia równoległego do 3 jednostek – co zwiększa parametry całego systemu. Istnieje również możliwość zabezpieczenia PCB dodatkową powłoką ochronną (wersja opcjonalna).

Przykładową charakterystykę modelu CSP-3000-250 pokazano na **rysunku 1**. Regulacja napięcia jest możliwa w zakresie 125...250 VDC, a prądu w zakresie od 1,36 A do 17 A. Właśnie dzięki tak szerokiej charakterystyce zasilacz może pracować z wieloma typami diod mocy. Urządzenie zostało wyposażone w szereg zabezpieczeń takich jak: zwarcie, przeciążeniowe, nadnapięciowe, temperaturowe, czy też informujące o awarii wentylatora. Producent przewidział 5 lat gwarancji.



Seria UHP

W branży zasilania diod LED istotną rolę odrywają również zasilacze bezwentylatorowe. Coraz częściej możemy spotkać aplikacje, gdzie zasilanych jest kilkadziesiąt lub nawet kilkaset metrów taśm LED i jednym z wymogów jest brak wentylatora. Do tej pory



jedynym w takiej sytuacji było zastosowanie kilku zasilaczy połączonych równolegle, zamontowanych na szynie. Na szczęście od niedawna mamy serię UHP, która została poszerzona o modele 1000/1500/2500 W. Producent przewidział kilka wersji o napięciach wyjściowych 12/24/36/48 VDC (w zależności od mocy wyjściowej). Dodatkowo warto wspomnieć o niskim profilu tych urządzeń (41...60 mm), dzięki czemu mogą być montowane w miejscach o ograniczonej przestrzeni. Niewątpliwymi atutami są również dodatkowe funkcje, takie jak możliwość zdalnego włączania/wyłączania, wyjście DC OK czy opcjonalna wersja z komunikacją PMBus/CANBus. Wszystkie modele pracują w szerokim zakresie temperatur (od -30 do +70°C) oraz mają 5-letnią gwarancję.

Podsumowanie

W miarę rozwoju nowoczesnych technologii LED rośnie zapotrzebowanie na coraz nowsze konstrukcje zasilaczy. Praca bezwentylatorowa, wysokie napięcie wyjściowe przy jednoczesnej dużej mocy wyjściowej czy dodatkowa magistrala komunikacyjna – są to jedynie niektóre z coraz liczniejszych wymagań stawianych zasilaczom tego typu. Można się spodziewać, że w przyszłości zasilacze staną się urządzeniami jeszcze bardziej złożonymi pod względem technologicznym.

REKLAMA



Seria XLG

25W / 50W / 75W / 100W / 150W / 200W / 240W

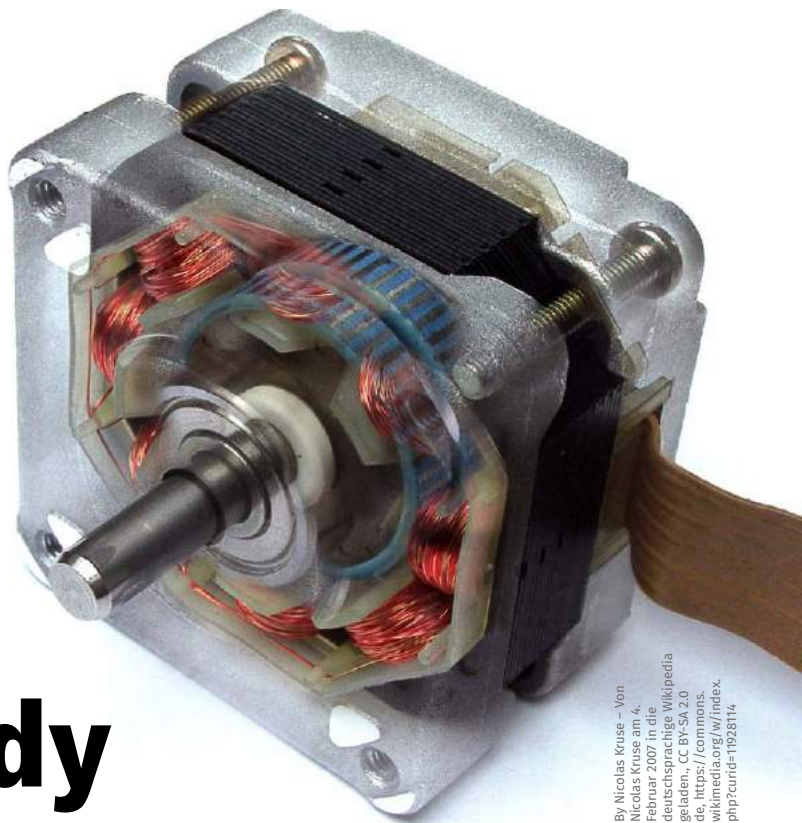
Wysoka niezawodność w niskiej cenie

- Zasilacze o stałej mocy wyjściowej
- Szczelna obudowa IP67
- Odporność na przepięcia 6kV/4kV (opcja 10kV)
- Zgodność z normami IEC 61347 / UL8750
- Wersja „AB” z opcją ściemniania



Silnik elektryczny jest znany od ponad 180 lat. To bardzo długa historia. Tym bardziej jeśli rozpatrzmy ją w kontekście szybkości rozwoju całej techniki. Ale też silnik przeszedł niewyobrażalną metamorfozę. Zmiany objęły nie tylko konstrukcję i wykorzystanie nowych materiałów, ale zmieniły się też zastosowania. Powstała niezliczona wręcz liczba typów i odmian silnika elektrycznego. Wśród nich jest silnik krokowy.

Silniki krokowe i serwonapędy

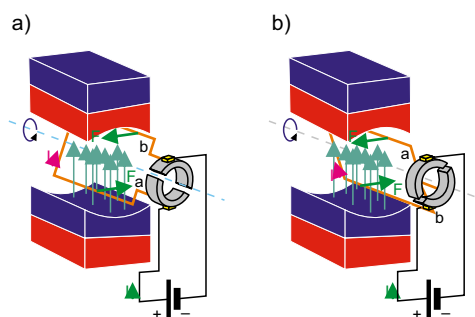


By Nicolas Kruse - Von Nicolas Kruse am 4. Februar 2007 in die deutschsprachige Wikipedia geladen., CC BY-SA 2.0 de, <https://commons.wikimedia.org/w/index.php?curid=11928114>

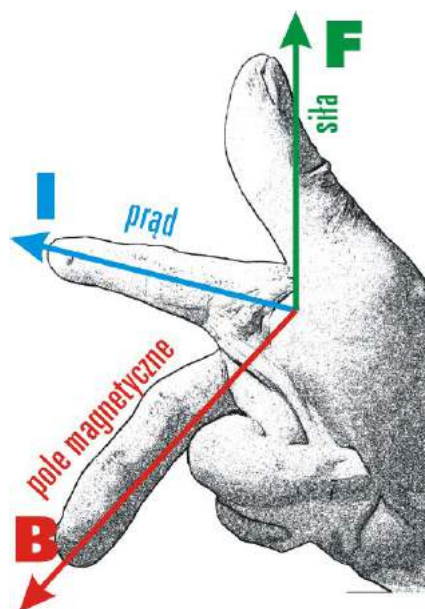
Michaela Faradaya (1791–1867), fizyka i chemika, kojarzymy przede wszystkim jako konstruktora komory, której wnętrzu jest izolowane od zewnętrznego pola elektrycznego, zwanej od nazwiska autora klatką Faradaya. Powstała w wyniku szeregu eksperymentów z elektrycznością, ale wypływające z nich wnioski przyczyniły się również do opracowania innego epokowego wynalazku, jakim był silnik elektryczny. Samego Faradaya nie można jednak bezpośrednio łączyć z tym urządzeniem. Konstruktor pierwszego silnika był inny fizyk – Peter Barlow. Swój wynalazek przedstawił w roku 1822. Nie była to jednak konstrukcja nadająca się do poważniejszych zastosowań. Za taką należy uznać silnik opracowany przez Thomasa Davenporta. Urządzenie było na tyle rozwinięte, że w roku 1837 zostało opatentowane w USA. Wkrótce powstało wiele nowych wersji silników, z którymi związani są tacy wynalazcy jak: Moritz Jacobi czy niezawodny Thomas Alva Edison. Skonstruowany w roku 1880 silnik Edisona miał wymiary 2,5×4 cm i osiągał ok. 4000 obrotów na minutę. Nie można również nie wspomnieć o zaczynającym swoją karierę w laboratorium Edisona Nikoli Tesli. Jego pomysły szybko wzbudziły zazdrość pracodawcy, który w pewnym okresie traktował go jak swojego największego wroga.

Wszystko zaczęło się od silnika prądu stałego

Zasada działania silnika elektrycznego prądu stałego jest dla nas dzisiaj wręcz banalnie prosta. Ale to tak, jak z pytaniami, na które zna się odpowiedź – wydają się łatwe. Spójrzmy zatem, o co w ogóle w silnikach chodzi? Na **rysunku 1** przedstawiono budowę typowego silnika prądu stałego. Mamy dwa magnesy stałe (czy też dwa bieguny jednego magnesu) wytwarzające jednorodne pole magnetyczne oraz ramkę przewodzącą prąd, która obraca się pod wpływem oddziaływania pola magnetycznego na przewodnik z prądem. Ramka jest zasilana ze źródła prądu stałego. Pojawia się jednak dość istotny problem – jak dołączyć źródło zasilające ramkę, aby nie poskręcały się przewody. Najprostszym nasuwającym się rozwiązaniem jest zastosowanie odpowiedniego złącza składającego się z dwóch blaszek wykonanych jako półpięścienie i dwóch stykających się z nimi tzw.



Rysunek 1. Budowa typowego silnika prądu stałego



Rysunek 2. Reguła 3 palców prawej ręki pozwalająca określić zwrot siły elektrodynamicznej

szczotek. Kierunek obrotu ramki wynika ze zwrotu siły elektrodynamicznej wytwarzanej na skutek oddziaływania pola magnetycznego na przewodnik z prądem (**rysunek 2**). No dobrze, ale przełączenie blaszek komutatora po wykonaniu półobrotu spowoduje zmianę kierunku przepływu prądu w ramce. I całe szczęście, że tak jest. Gdyby nie nastąpiło takie przełączenie, silnik zablokowałby się w pozycji, w której płaszczyzna ramki znalazłaby się w położeniu prostopadłym do linii sił pola magnetycznego i dalsze obroty nie byłyby możliwe. Zastosowanie komutatora pozwoliło więc połączyć dwa warunki niezbędne do pracy silnika.

Konstrukcja silnika prądu stałego była cały czas rozwijana. Komutator był niezbędny do jego pracy, ale jednocześnie stanowił najsłabszy element silnika. W uzwojeniach silników dużej mocy musiał płynąć prąd o dużym natężeniu, a to powodowało iskrzenie styków w chwili przełączania. Szczotki zużywały się więc systematycznie w trakcie eksploatacji i konieczna była ich okresowa wymiana. Naturalne było zatem dążenie do takiej zmiany konstrukcji silnika, aby wyeliminować z niego komutator.

Silniki bezszczotkowe (BLDC – BrushLess Direct-Current)

Podobno nie ma rzeczy niemożliwych. Niezbędny, wydawać by się mogło, element silnika elektrycznego został jednak wyeliminowany. Powstała nowa klasa silników – silniki bezszczotkowe. Ogólna zasada działania została jednak zachowana. I w tym przypadku należało w jakiś sposób wytworzyć moment napędowy wprowadzający wirnik w ruch obrotowy. W silniku BLDC magnesy stałe umieszczono w obrotującym się wirniku, a uzwojenia cewek przełożono do stojana. Jeśli w silniku klasycznym możliwe było choćby teoretyczne stosowanie tylko jednej cewki, to w silniku bezszczotkowym tak już się nie da. Wynika to z konieczności przełączania pola magnetycznego statora, za którym podążają magnesy wirnika. No właśnie – przełączania. Kolejna różnica między silnikiem klasycznym a BLDC polega na konieczności odpowiedniego sterowania uzwojeniami silnika. W tym przypadku nie wystarczy dołączenie jakiegoś źródła napięcia do cewek, konieczne jest zastosowanie odpowiedniego sterownika, który będzie przełączał napięcie do kolejnych cewek według ustalonego porządku. Można wnioskować, że zmieniając częstotliwość przełączania napięcia, można regulować obroty wirnika. W klasycznym silniku regulacja obrotów mogła być prowadzona przez zmianę napięcia zasilającego cewki. Powodowało to zmianę natężenia płynącego prądu, a więc i siłę elektrodynamiczną. W efekcie skutkowało zmianą prędkości obrotowej, ale też niestety zmianą momentu obrotowego. W silniku BLDC moment obrotowy jest niemal stały, niezależnie od prędkości obrotowej.

Podstawowym problemem sterowania silnikami BLDC jest konieczność synchronizacji przełączania uzwojeń z położeniem wirnika. Sterownik musi więc w jakiś sposób tę informację pozyskiwać, co w praktyce jest realizowane kilkoma metodami. Jedną z nich polega na zastosowaniu halotronów umieszczonych w stojanie. Czujniki te wytwarzają impulsy napięciowe w chwili przechodzenia nad nimi magnesów wirnika. Impulsy są następnie używane do sterowania uzwojeniami. Nie wymaga to skomplikowanego układu, wystarczy względnie prosta logika. Oczywiście niezbędny jest też odpowiednio zaprojektowany stopień mocy.

Inna metoda detekcji położenia wirnika jest oparta na znanym zjawisku indukowania siły elektromotorycznej w przewodniku poruszającym się w polu magnetycznym. W tym przypadku mówimy o tzw. sile przeciw-elektromotorycznej. Analizując ten parametr, można ustalić, w jakim

położeniu w danej chwili znajduje się wirnik i na tej podstawie dokonywać przełączeń zasilania cewek. Wadą metody jest brak siły elektromotorycznej w chwili, gdy wirnik się nie obraca. Jednak nie będziemy się teraz zajmować tym problemem.

Silniki BLDC mają kilka istotnych zalet w stosunku do silników klasycznych. Przede wszystkim wyeliminowano z nich zużywające się elementy: szczotki i komutator, co przyczyniło się także do znacznej redukcji hałasu wytwarzanego podczas pracy. Uzyskano ponadto dużą sprawność silnika i możliwość regulacji prędkości w dużym zakresie. Niestety, silniki bezszczotkowe są znacznie droższe od klasycznych.

Niedaleka droga od silnika bezszczotkowego do krokowego

Silnik krokowy do złudzenia przypomina swoją budową silnik bezszczotkowy. O ile jednak silnik bezszczotkowy obraca się ruchem ciągłym, o tyle silnik krokowy może w sposób kontrolowany wykonywać obrót o określony kąt – tzw. krok lub jego wielokrotność. I w tym przypadku niezbędny jest specjalny sterownik podający impulsy zasilające do uzwojeń silnika. Położenie wirnika jest uzależnione bezpośrednio od liczby podanych impulsów, co teoretycznie eliminowałoby konieczność stosowania czujników Halla, które jak wiemy, były niezbędne w silnikach BLDC.

Silniki krokowe, podobnie jak inne silniki, ewoluowały przez lata, co zaowocowało powstaniem wielu ich odmian. Być może zaskakująca będzie informacja o tym, że pierwsze silniki krokowe były stosowane w systemach kontroli położenia i zdalnego sterowania już w roku 1920 przez brytyjską marynarkę wojenną. Konstrukcja była podobna do dzisiejszych rozwiązań, różniły się znacznie budową i wielkością. Należy pamiętać, że silników krokowych nie można rozpatrywać w oderwaniu od sterowników. Bez nich nie mogą działać. Można zatem powiedzieć, że rozwój silników krokowych wynika pośrednio z postępu w zakresie produkcji elementów półprzewodnikowych.

Podział silników krokowych ze względu na budowę

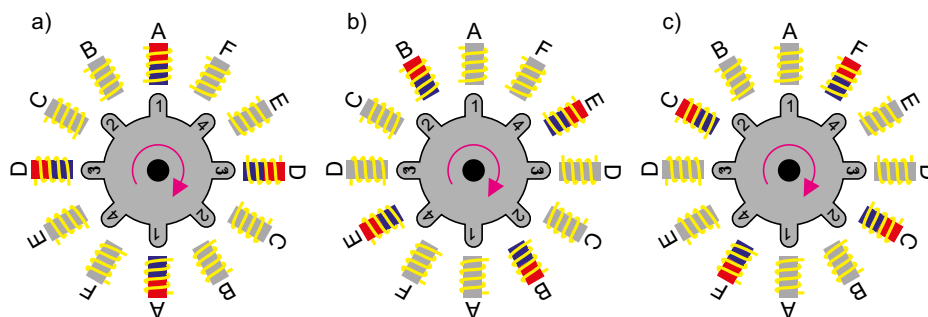
Obecnie stosowanych jest kilka odmian silników krokowych. Oprócz budowy wewnętrznej różnią się one także sposobem sterowania. Biorąc pod uwagę jeszcze inne względy techniczne, mamy sporą liczbę opcji.

Ze względu na budowę wyróżniamy trzy podstawowe typy silników:

1. silnik o zmiennej reluktancji (VR – *Variable Reluctance*)
2. silnik z magnesem stałym (PM – *Permanent Magnet*)
3. silnik hybrydowy (HB – *HyBrid*)

Silnik VR (o zmiennej reluktancji)

Reluktancja to opór magnetyczny definiowany jako stosunek siły magnetomotorycznej do całkowitego strumienia magnetycznego w pasywnym obwodzie magnetycznym przy założeniu, że strumień magnetyczny nie jest rozpraszany poza rdzeń. Reluktancja jest więc odpowiednikiem oporności w obwodzie elektrycznym. Strumień magnetyczny kieruje się więc przede wszystkim drogami o najmniejszej



Rysunek 3. Ilustracja zasady działania silnika VR (ze zmienną reluktancją)

reluktancji, podobnie jak prąd „wybiera” obwody o najmniej-
szej rezystancji.

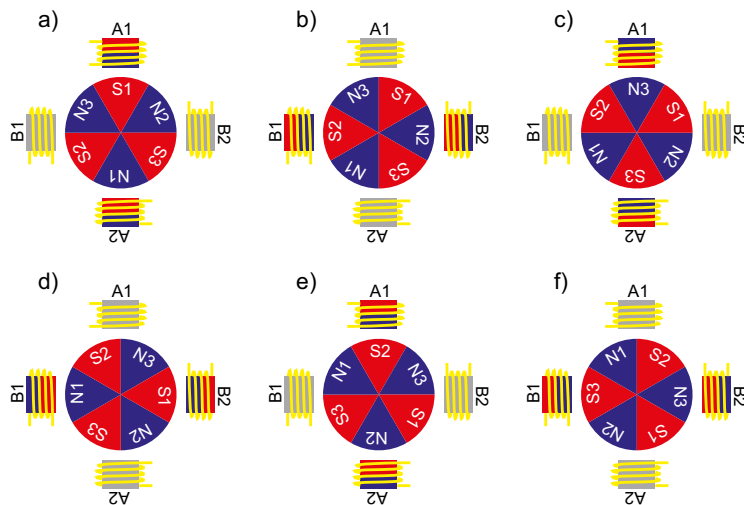
Wirnik silnika VR nie zawiera magnesu stałego (trwałego). Jest zbudowany z tarczy przypominającej koło zębate, która jest wykonana z miękkiego materiału ferromagnetycznego (**rysunek 3**). Kształt wirnika jest tak dobrany, że reluktancja obwodu magnetycznego zależy od jego położenia. Stojan natomiast zawiera zespół uzwojeń pogrupowanych w przeciwległe pary. Nietrudno zauważyć, że liczba cewek nie jest równa liczbie zębów wirnika. Jest to konieczne do wytworzenia momentu obrotowego powstającego w wyniku dążenia układu do osiągnięcia minimalnej reluktancji. Gdyby nie było tej asymetrii, silnik zatrzymałby się w neutralnym położeniu i jego praca nie byłaby możliwa. Uzwojenia są kolejno przełączane za pośrednictwem odpowiednio skonstruowanego sterownika. Na przykład w silniku przedstawionym na rysunku 3 w pierwszej fazie są włączone cewki A i D. Powoduje to przyciągnięcie do nich zębów odpowiednio 1 i 3. W kolejnym kroku zostaną włączone uzwojenia B i E, a ponieważ najbliższe im w poprzednim ustawieniu znajdowały się zęby odpowiednio 2 i 4, to one zostaną przyciągnięte do tych uzwojeń. Podobnie wyglądają kolejne fazy. Jak widać, rotor tego silnika będzie się obracał w prawo, w każdym kroku o 15° . Silniki ze zmienną reluktancją są obecnie wypierane przez silniki innych typów.

Silnik PM (z magnesem stałym)

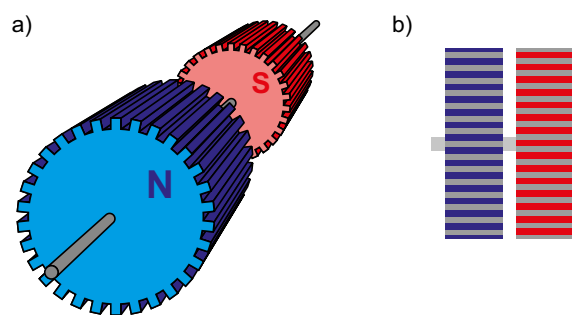
Jest powszechnie stosowany od ok. 1960 roku i stał się podstawowym silnikiem krokowym używanym w aplikacjach automatyki przemysłowej. Silnik z magnesem stałym charakteryzuje się lepszymi parametrami w porównaniu z silnikami VR. Wirnik jest w nim wykonany w postaci magnesu o przekroju kołowym z promieniowo rozmieszczonymi wieloma parami biegunów. Na **rysunku 4** pokazano przykład silnika z trzema parami biegunów N-S. Sterowanie odbywa się w tym przypadku za pośrednictwem dwóch par uzwojeń A1–A2 i B1–B2. Pełny cykl pracy składa się z kilku taktów. W pierwszym (**rysunek 4a**) prąd płynący przez uzwojenie A1 wytwarza biegun N przyciągający biegun S1 rotora. Jednocześnie cewka A2 wytwarza biegun S i przyciąga biegun N1. W drugim takcie następuje przełączenie sterowania uzwojeniami. Teraz biegun N od strony rotora wytwarza uzwojenie B1 (**rysunek 4b**). Z ustawienia rotora z poprzedniego cyklu wynika, że najbliższym uzwojeniu B1 biegunem S jest S2. Zostanie on przyciągnięty do uzwojenia B1 w drugim cyklu. Domyślamy się, że w trzeciej fazie znowu trzeba przełączyć sterowanie uzwojeniami. Gdybyśmy jednak zrobili tak samo, jak w pierwszej fazie, rotor obróciłby się w przeciwną stronę, gdyż najbliższym biegunem S dla uzwojenia A1 jest w drugiej fazie biegun S1. Aby zachować kierunek obrotu wirnika, konieczna jest zmiana polaryzacji uzwojenia A w trzeciej fazie. Teraz uzwojenie A1 wytwarza biegun S od strony rotora, co powoduje przyciągnięcie do niego bieguna N3 rotora (**rysunek 4c**). W podobny sposób odbywa się sterowanie silnikiem w kolejnych fazach. Zauważmy, że w każdym kroku naszego silnika rotor wykonuje obrót o 30° . Aby zmniejszyć ten kąt, w silnikach występuje zwykle więcej uzwojeń i par biegunów rotora. W praktyce uzyskuje się kąty obrotu wirnika równe $7,5^\circ$... 15° . Odpowiada to 48...24 skokom wirnika na jeden obrót.

Silnik HB (hybrydowy)

Jest połączeniem idei silników PM i VR. O ile zasada działania poprzednich silników była stosunkowo prosta, to znacznie się komplikuje dla silnika HB. Jego rotor jest zbudowany z dwóch namagnesowanych pierścieni o zębatej powierzchni, osadzonych na osi (**rysunek 5**). W rzeczywistości powinniśmy mówić raczej o wyżłobieniach czy szczelinach niż o zębach, ale będziemy je nazywać zębami. Bieguny uzwojeń stojana mają również wyżłobienia dopasowane do ząbków rotora (**fotografia 6a**). Ząbki te wraz z ząbkami rotora



Rysunek 4. Ilustracja zasady działania silnika PM (z magnesem stałym)

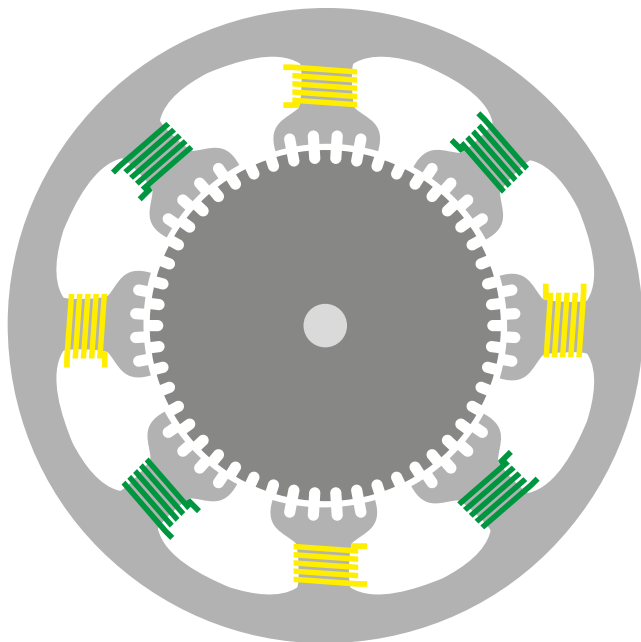


Rysunek 5. Budowa rotora silnika HB (hybrydowego): a) rzut ukośny, b) rzut prostopadły



Fotografia 6. Silnik hybrydowy: a) stator, b) rotor

tworzą obwód magnetyczny. Należy zwrócić uwagę na to, że zęby w magnesach rotora są przesunięte względem siebie w ten sposób, że ząbek magnesu N trafia w szczelinę magnesu S (**rysunek 5b**). Przedstawiony układ jest więc niesymetryczny. Na pewnym odcinku ząbki rotora będą się pokrywały z ząbkami bieguna N magnesu rotora, ale dla bieguna S już tak nie będzie wskutek wspomnianego już przesunięcia ząbków w tych elementach rotora. W pracującym silniku, gdy uzwojenia stojana są odpowiednio zasilane, rotor będzie przyjmował taką pozycję, aby dla danego wysterowania uzwojeń zapewnić minimalną reluktancję obwodu magnetycznego. Przy ustalonym wysterowaniu pozycja taka będzie stabilna. W silniku hybrydowym jedno uzwojenie ma kilka biegunów. Najczęściej są stosowane dwa odpowiednio połączone uzwojenia, każde mające po cztery bieguny (**rysunek 7**). Silniki hybrydowe charakteryzują się dużą rozdzielczością. Typowe kąty pojedynczych kroków wynoszą $0,9^\circ$... $3,6^\circ$ co odpowiada od 100 do 400 kroków na obrót.



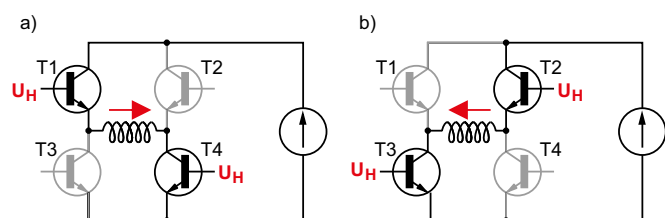
Rysunek 7. Przekrój silnika hybrydowego

Ogólnie o sterowaniu silnikami krokowymi

Budowa wewnętrzna silników krokowych jest, jak mogliśmy się przekonać, dość zróżnicowana. Jest to jednak dopiero początek dylematów konstruktora, który musi dokonać wyboru odpowiedniego typu silnika do swojej aplikacji. Właściwie od strony praktycznej jest to zagadnienie drugoplanowe, o wyborze silnika będzie raczej decydował sposób sterowania niż jego budowa wewnętrzna. Rozpatrzmy więc kolejny podział silników uwzględniający typy uzwojeń i konfiguracje połączeń. Będzie od tego zależała m.in. liczba wyprowadzeń, a więc i stopień skomplikowania sterownika.

Z wcześniejszego opisu wynika, że do sterowania silnikami krokowymi na pewno konieczne jest sekwencyjne przełączanie uzwojeń. Rozpatrywany był też przypadek wymagający zmiany kierunku przepływu prądu w uzwojeniach. Można wyobrazić sobie jeszcze jedną konfigurację, w której uzwojenia są przełączane nie tylko sekwencyjnie, ale też następuje zmiana polaryzacji napięcia zasilającego cewki. Do takich wymagań potrzebne są odpowiednio skonstruowane sterowniki. Najprostszym rozwiązaniem układowym jest dobrze wszystkim znany prosty klucz tranzystorowy. Umożliwia on tylko włączanie lub wyłączanie cewki. Zdecydowanie większe możliwości ma tzw. mostek H pozwalający zmieniać kierunek przepływu prądu przez cewkę. Zasadę działania takiego mostka przedstawiono na **rysunku 8**. Podanie napięcia sterującego (U_H) na bazy tranzystorów T1 i T4 spowoduje ich włączenie, tranzystory T2 i T3 będą wyłączone. Prąd popłynie przez uzwojenie w kierunku od T1 do T4 (rysunek 8a). Przełączenie sterowania na tranzystory T2 i T3 spowoduje zmianę kierunku przepływu prądu (rysunek 8b). W spotykanych mostkach wykonywanych często jako układy scalone stosowane są nie tylko tranzystory NPN czy PNP, ale też MOSFET z kanałem typu N lub kanałem typu P.

W silnikach krokowych często występują dwa uzwojenia. Można tak powiedzieć, jeśli będą one rozpatrywane w znaczeniu elektrycznym.



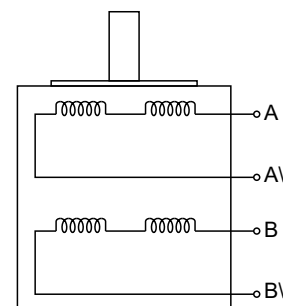
Rysunek 8. Ilustracja zasady działania mostka H: a) włączona sekcja T1, T4, b) włączona sekcja T2, T3

W rzeczywistości każde uzwojenie składa się najczęściej z kilku odpowiednio połączonych cewek przesuniętych względem siebie o pewien kąt, tak jak pokazano np. na rysunku 7 (jedno uzwojenie zaznaczone kolorem żółtym, drugie zielonym). Końcówki uzwojeń są oznaczane symbolami A, A\ i B, B\ informującymi o tym, która z nich jest początkiem, a która końcem uzwojenia. Kierunek połączenia uzwojeń ze sterownikiem jest istotny, zamiana wyprowadzeń może spowodować, że silnik nie będzie działał. Spotykane są silniki, w których uzwojenia mają odczep w środku pozwalający, wbrew pozorom, zmniejszyć liczbę połączeń ze sterownikiem, a zatem i jego konstrukcję, ponieważ dwie podzielone cewki są traktowane jak jedna. Możliwe konfiguracje pokazano na **rysunkach 9...12**.

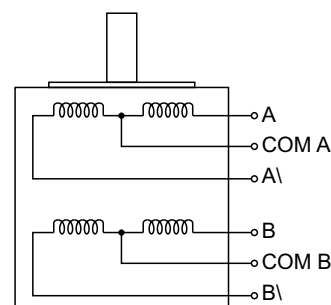
Dość liczne kombinacje uzwojeń mają ścisły związek z kolejną klasyfikacją silników krokowych. Klasyfikacji tej nie można jednak rozpatrywać w oderwaniu od innych metod sterowania. Dalszy opis będzie więc uwzględniał powyższe zagadnienia.

Sterowanie unipolarne

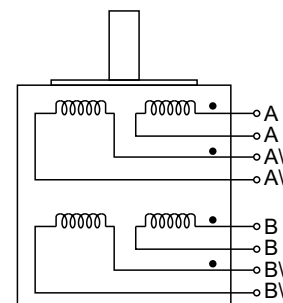
Charakteryzuje się tym, że w uzwojeniach silnika prąd płynie zawsze w tym samym kierunku. Rozpatrzmy silnik z często spotykanymi sześcioma wyprowadzeniami. Uzwojenia mają więc odczep w środku dzielący każde z nich na dwie połowy. Ważne jest, że prąd płynie zawsze w kierunku od odczepu (COM) do końcówek A lub A\ (w zależności od występowania). Dzięki temu uzyskuje się jednak zmianę kierunku strumienia magnetycznego w rdzeniu statora, a zatem i biegunowość cewek. Pełny cykl pracy składa się z 4 taktów, przedstawionych na **rysunku 13**. Jak widać, w każdej chwili włączona jest tylko jedna cewka (połowa uzwojenia). Sekwencja sterowania np.: A, B, A\, B\, powoduje w tym przykładowym silniku obrót rotora o 90° w lewo w każdym kroku. Niestety, ponieważ zawsze aktywna jest tylko połowa każdego uzwojenia, w każdym takcie jest wykorzystywana tylko połowa możliwości silnika, co oczywiście jest jego wadą. Z uwagi na sekwencyjne przełączanie uzwojeń



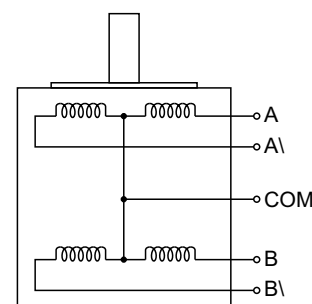
Rysunek 9. Silnik z dwoma uzwojeniami 4-przewodowy



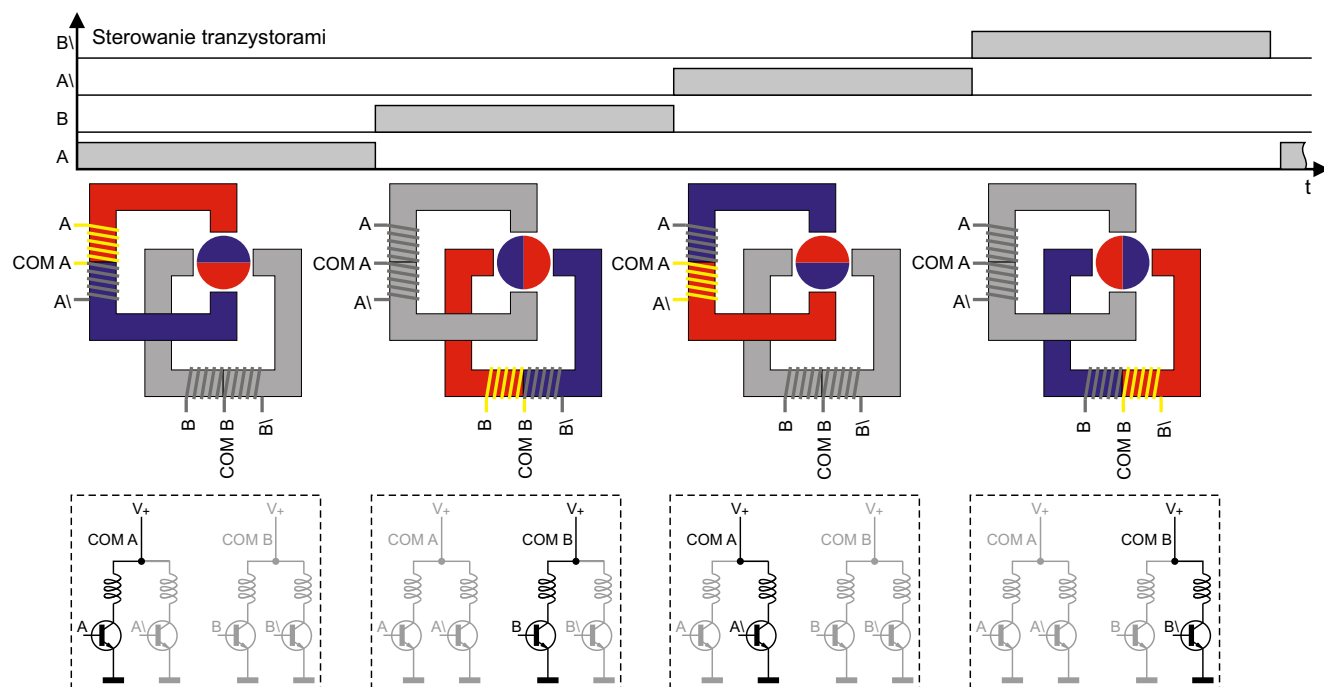
Rysunek 10. Silnik z dwoma dzielonymi uzwojeniami 6-przewodowy



Rysunek 11. Silnik z dwoma podwójnymi uzwojeniami 8-przewodowy



Rysunek 12. Silnik z dwoma dzielonymi uzwojeniami ze wspólnym odczepem 5-przewodowy



Rysunek 13. Sterowanie 1-fazowe (falowe) silnika unipolarnego

i aktywność tylko jednego uzwojenia w każdym takcie, sterowanie takie określane jest jako falowe, jednofazowe. Ponadto można zauważyć, że w każdym takcie rotor wykonuje maksymalny możliwy obrót. Mówimy więc o sterowaniu pełnokrokowym.

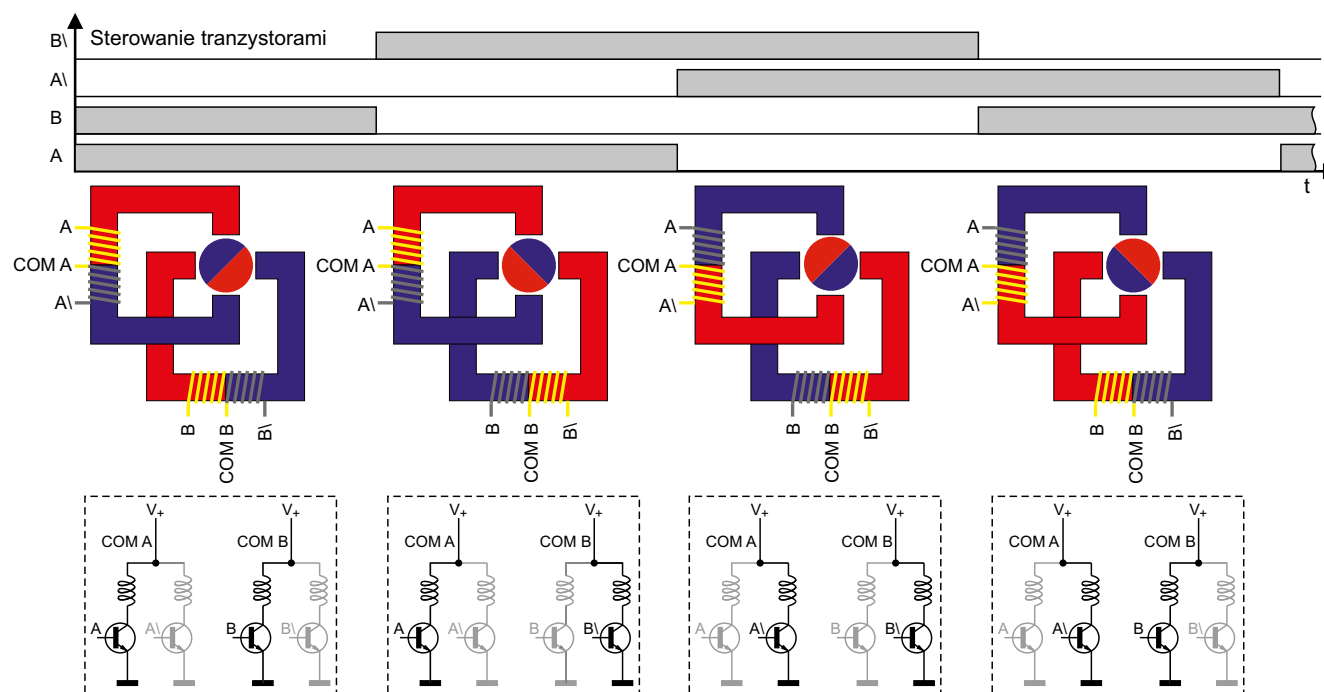
Rozważmy inną metodę sterowania tym samym silnikiem (**rysunek 14**). W tym przypadku zawsze aktywne będą jednak dwie cewki. Mówimy zatem o sterowaniu dwufazowym. Nietrudno dostrzec, że w każdym takcie rotor wykonuje również obrót o maksymalny możliwy kąt (90°). Mamy więc do czynienia ze sterowaniem pełnokrokowym.

W przedstawionych przykładach omówiono sterowanie silnikami zawierającymi dwa uzwojenia. Jeśli do jakiejś aplikacji zostanie zastosowany silnik o większej liczbie uzwojeń, ogólna zasada nie zmieni się. Bardziej złożony będzie oczywiście sam sterownik, gdyż musi on obsługiwać większą liczbę wyprowadzeń.

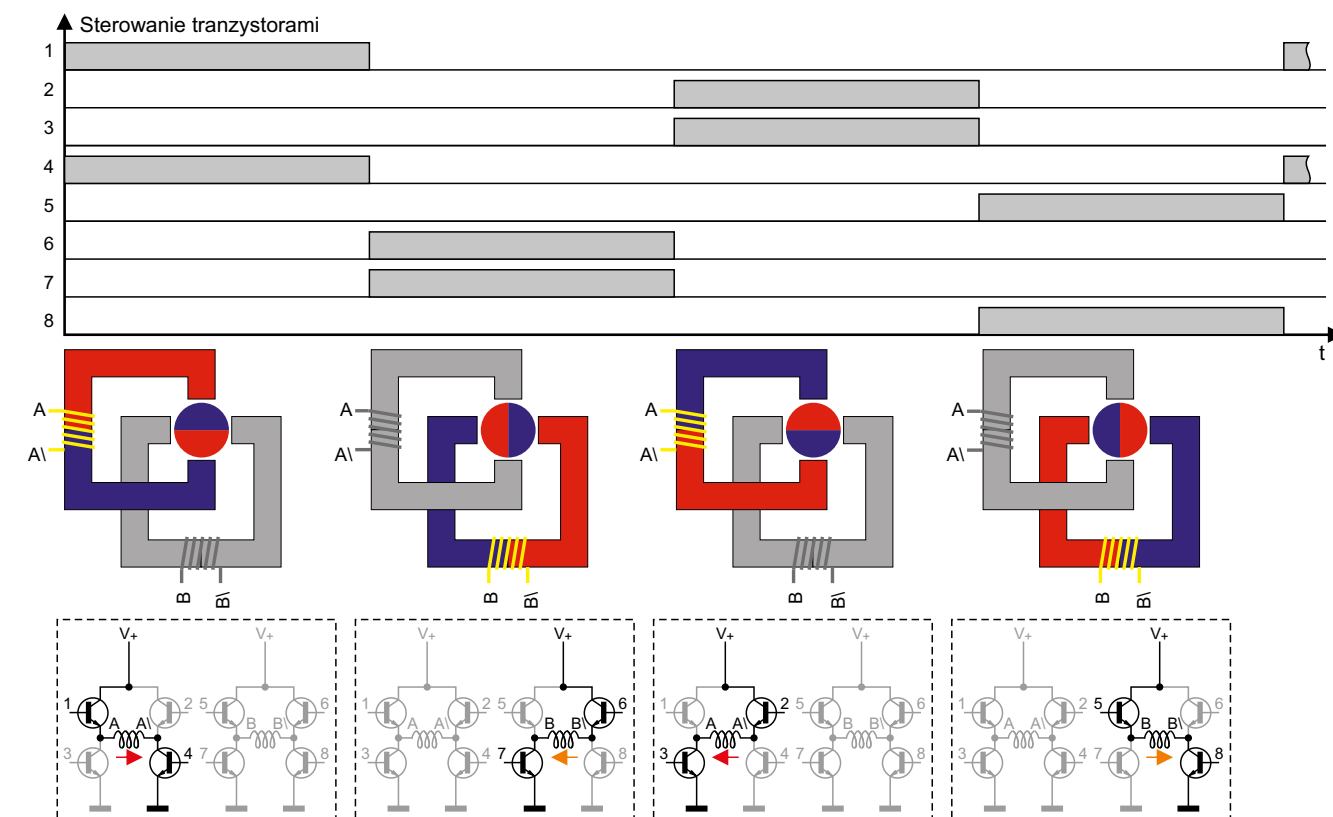
Sterowanie bipolarne

Wymaga takiej konstrukcji sterownika, która zapewni zmianę kierunku przepływu prądu w uzwojeniach silnika. Można ją zrealizować w oparciu o mostek H. Do przełączenia kierunku prądu nie jest wymagana zmiana polaryzacji napięcia zasilającego. Mostek spełnia swoje zadanie (przełączanie kierunku prądu w cewce) nawet wtedy, gdy jest zasilany jednym napięciem.

Sterowanie bipolarne silnikami krokowymi jest bardzo podobne do unipolarnego. Przykładowy pełnokrokowy cykl pracy przedstawiono na **rysunku 15**. W tym przypadku cewka nie jest dzielona, lepsze jest zatem wykorzystanie silnika. Pełny cykl składa się z 4 taktów: A, B, A', B. Sekwencja sygnałów sterujących mostkiem jest widoczna na wykresie. Wynika z niej, że w pierwszym takcie włączane są tranzystory 1 i 4, w drugim takcie 6 i 7, w trzecim takcie 2 i 3 i w czwartym takcie 5 i 8. Taką sekwencję musi zapewnić sterownik silnika.



Rysunek 14. 2-fazowe sterowanie pełnokrokowe silnika unipolarnego



Rysunek 15. Falowe sterowanie pełnokrokowe silnikiem bipolarnym

Dla formalności należy wspomnieć o pełnokrokowym bipolarnym sterowaniu dwufazowym (rysunek 16). W jednym takcie używane są dwa uzwojenia rotora, a w poszczególnych taktach następuje zmiana kierunku płynącego w nich prądu. Sterownik powinien zapewnić sekwencję pokazaną na wykresie czasowym z rysunku 16.

Sterowanie półkrokowe

W przedstawionych teoretycznych przykładach kąt obrotu rotora dla pełnego kroku był równy 90° . Wiemy, że można go zmniejszyć, zwiększając liczbę uzwojeń silnika, ale są oczywiście fizyczne ograniczenia, których nie da się pokonać. Z punktu widzenia użytkowników im mniejszy jest kąt pojedynczego kroku, tym większą można uzyskać precyzję sterowania. Nasuwający się pomysł na rozwiązanie problemu to zastosowanie odpowiednio do potrzeb zaprojektowanej przekładni mechanicznej (np. zębatej lub ślimakowej). Pomysł nie jest zły. Można go nawet spotykać w wielu aplikacjach, ale nie eliminuje występujących przy dużych krokach drgań i rezonansów mechanicznych. Szukano więc innych rozwiązań, a jednym z nich jest sterowanie półkrokowe.

Należy zadać sobie pytanie, czy dla silnika o danej budowie wewnętrznej możliwe jest takie sterowanie uzwojeniami, żeby rotor mógł obracać się o kąt mniejszy niż dla pełnego kroku. Przyjrzyjmy się więc sekwencji sterującej silnikiem z dwoma uzwojeniami z odcięciem, pokazanej na rysunku 17. Jest to ten sam silnik, którego inny sposób sterowania był omawiany w oparciu o rysunek 13. Tym razem w jednym takcie może być aktywne jedno lub dwa uzwojenia, a pełny cykl zawiera 8 taktów. Nietrudno zauważyć, że dla sekwencji sterowania $AB\backslash, A, AB, B, A\backslash B, A\backslash, A\backslash B, B\backslash$ kąt obrotu rotora jest równy 45° , a więc stanowi połowę kąta obrotu dla pełnego kroku. Z tego względu przedstawioną metodę sterowania nazwano półkrokową.

Sterowanie mikrokrokowe

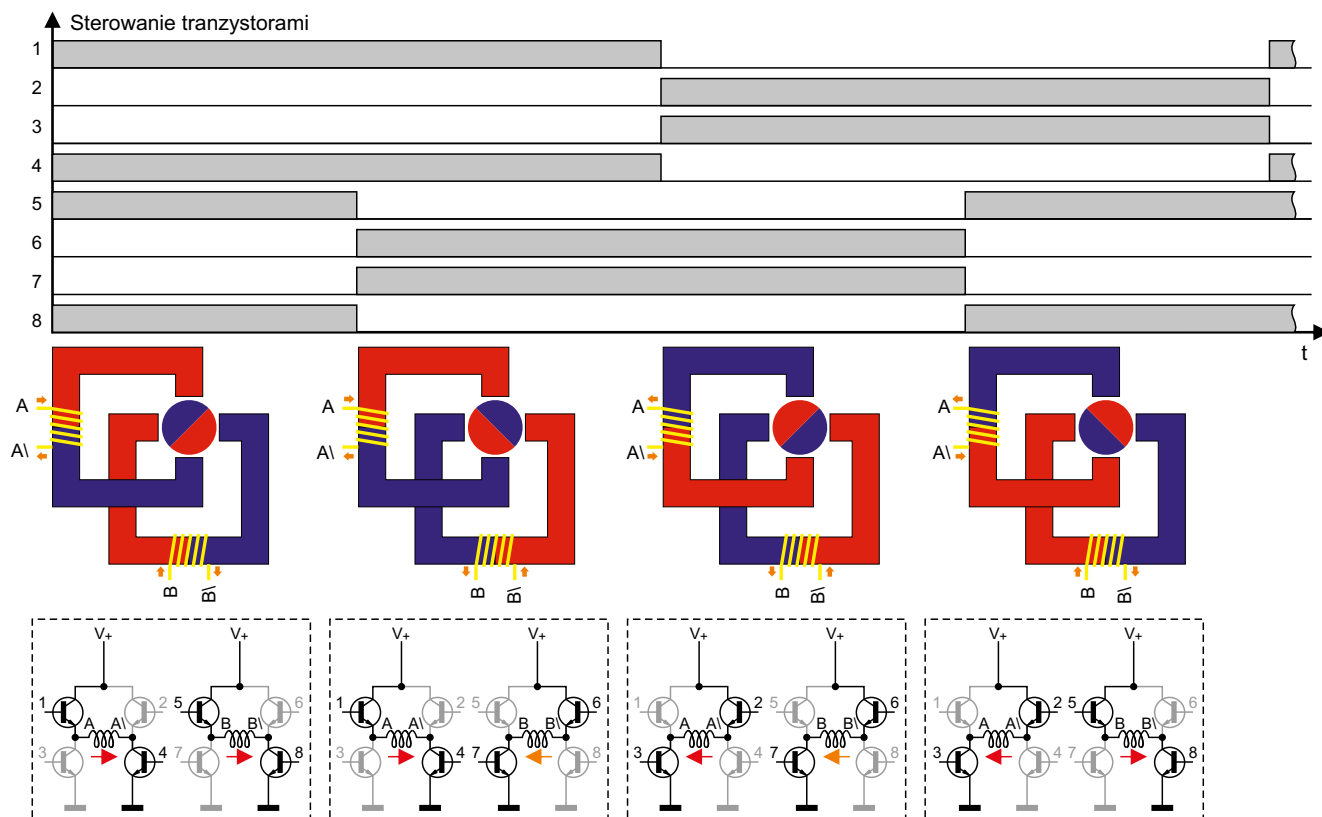
Sterowanie półkrokowe rozwiązało problem rozdzielczości i drgań silnika krokowego, ale nadal efekt nie był idealny. Sama nazwa – silnik krokowy, może sugerować, że kąty obrotu rotora są zawarte tylko w pewnym ograniczonym zbiorze wartości wynikających z budowy wewnętrznej

silnika i metodzie cyfrowego poniekąd sterowania. Mówiąc o cyfrowym sterowaniu, mamy na myśli fakt, że dane uzwojenie przyjmuje w trakcie pracy tylko dwa stany: może być włączone albo wyłączone. W tej sytuacji jedyną metodą poprawiania rozdzielczości jest zwiększanie liczby uzwojeń, co z czysto mechanicznych względów jest mocno ograniczone. No dobrze, ale strumień magnetyczny wytwarzany w rdzeniu cewki jest proporcjonalny do natężenia prądu płynącego w uzwojeniach. Odpowiednio sterując tym strumieniem, można więc uzyskać pośrednie położenia rotora. To jest właśnie idea sterowania mikrokrokowego. Nie zapomnijmy jednak o tym, że mamy do czynienia z silnikiem krokowym, a więc sterownik powinien mieć w każdej chwili pełną kontrolę nad nim. Musi mieć możliwość zatrzymania silnika w określonym położeniu. Jeśli rotor ma być ustawiany w pozycjach pośrednich między pełnymi krokami, konieczne będzie generowanie odpowiednio dobranych dla tego ustawienia napięć zasilających uzwojenia silnika. Nawet bez głębszej analizy można przypuszczać, że równomierny obrót rotora uzyska się dla przesuniętych o 90° przebiegów sinusoidalnych. Sterownik silnika krokowego generuje jednak z definicji napięcia o wartościach dyskretnych. Zatem powinien jak najdokładniej odzwierciedlać przebiegi analogowe. Oprócz wspomnianych już przebiegów sinusoidalnych spotykane są np. przebiegi prostokątne.

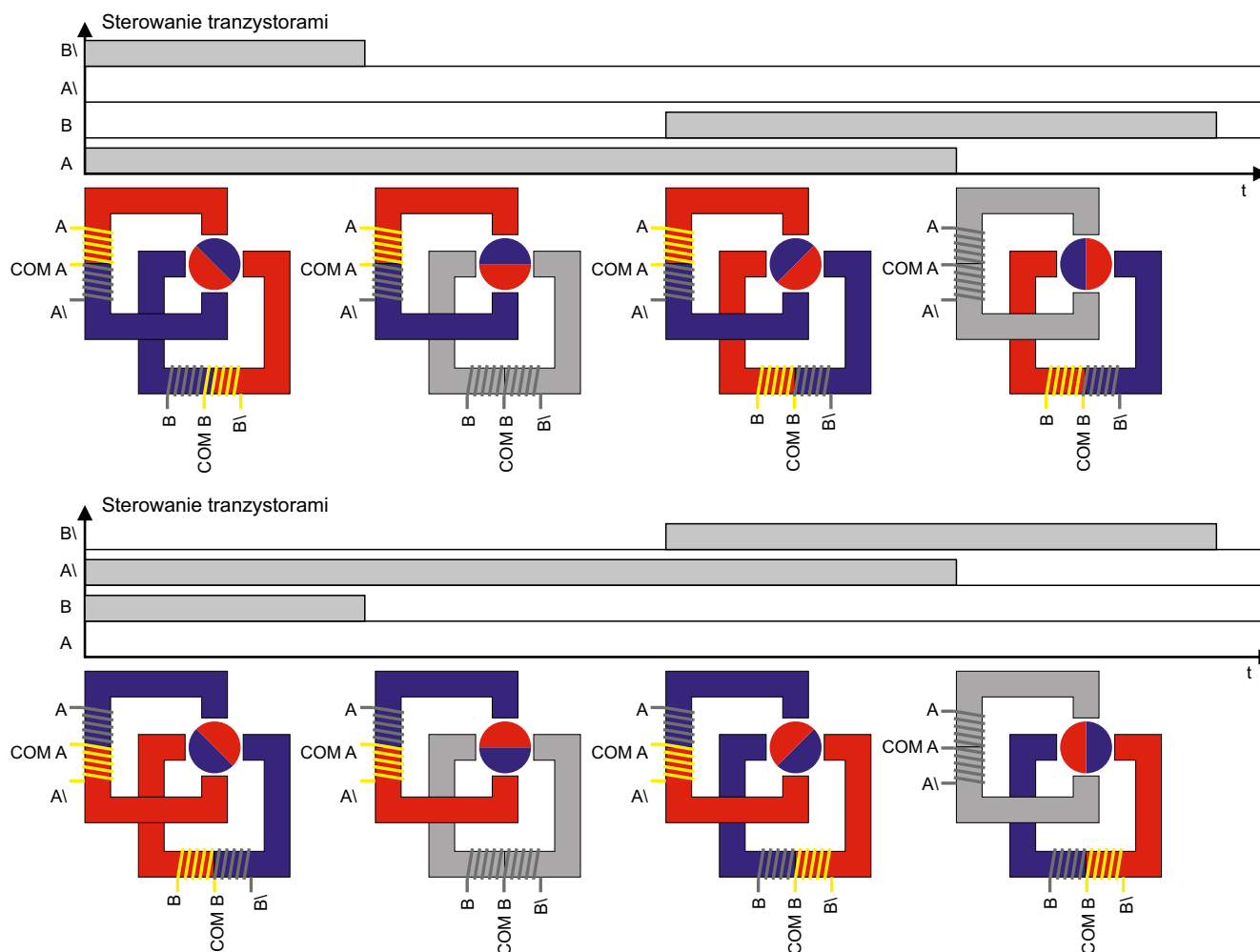
Sterowniki zapewniające pracę mikrokrokową są dostępne w postaci układów scalonych oferowanych przez wielu producentów. Przykład sterowania mikrokrokowego pokazano na rysunku 18. Zaznaczono natężenia prądów w uzwojeniach A i B uzyskiwane za pośrednictwem odpowiednio zaimplementowanego algorytmu. Już przy powierzchownym spojrzeniu na wykres nasuwa się wniosek, że przy ustalonej częstotliwości taktowania silnik będzie poruszał się znacznie wolniej niż przy sterowaniu pełno- czy nawet półkrokowym. Jest to cena za zwiększenie rozdzielczości i zmniejszenie drgań i rezonansów mechanicznych.

Niekorzystne zjawiska mechaniczne i elektryczne występujące w silniku krokowym

Silnik krokowy, nawet mimo małej wielkości, jest dość złożonym urządzeniem mechanicznym. Jego elementy mają swoje wymiary i masę, a co za tym idzie charakteryzują się również określoną



Rysunek 16. 2-fazowe sterowanie pełnokrokowe silnikiem bipolarnym



Rysunek 17. Sterowanie półkrokowe silnikiem unipolarnym

bezwładnością. W chwili wykonywania kroku rotor musi obrócić się o charakterystyczny dla danego typu silnika kąt. Nawet przy impulsowym wymuszeniu przepływu prądu w uzwojeniach, rotor, jak każdy obiekt fizyczny, najpierw musi być rozpędzony, a następnie wyhamowany w pozycji końcowej wykonywanego kroku. Jest to ruch bardzo gwałtowny, szczególnie przy sterowaniu pełnokrokowym. Rotor nigdy nie zatrzymuje się nagle (w zerowym czasie) w pozycji końcowej. Zawsze podlega on pewnym drganiom o gasnącym charakterze. Dopiero gdy one wygasną, można mówić o wykonaniu kroku. Czas wygasania drgań własnych zależy od własności mechanicznych samego silnika, ale też np. od obciążenia wału. Zagadnienie to staje się więc szczególnie istotne, gdy obciążenie może ulegać zmianom. Czas wygaszania drgań własnych ogranicza częstotliwość sterowania silnikiem. Impulsy sterujące nie mogą pojawiać się przed wygaśnięciem drgań, gdyż spowoduje to zablokowanie silnika. Jest to główny czynnik ograniczający maksymalną prędkość obrotową silnika. Efekty te są minimalizowane przez stosowanie sterowania półkrokowego, a przede wszystkim mikrokrokowego.

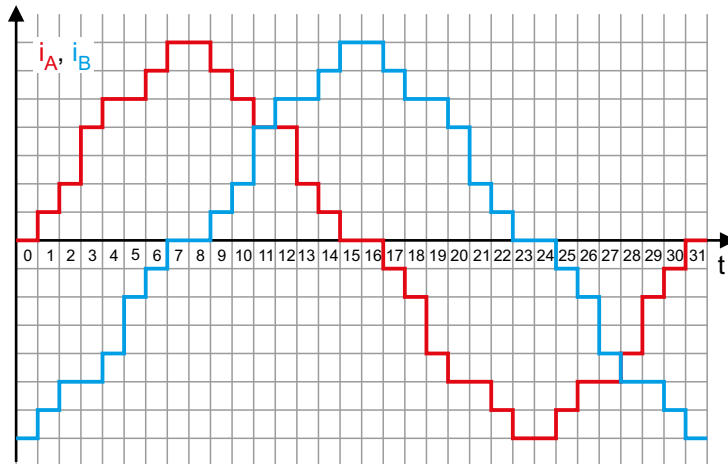
Bezwładność może być nie tylko mechaniczna. Silnik zawiera uzwojenia, które są elementami indukcyjnymi. Jeśli mówimy o zasilaniu impulsowym układów z indukcyjnościami, to wcześniej czy później musimy się liczyć z problemami. Pojawiają się one wraz ze wzrostem częstotliwości sterowania. Cewki o określonej indukcyjności i rezystancji tworzą układ RL o stałej czasowej L/R . Parametr ten, jak wiadomo, decyduje o szybkości narastania prądu w takim układzie pobudzonym sygnałem impulsowym. Przy zbyt dużej częstotliwości sterowania prąd nie zdąży narosnąć do swojej maksymalnej wartości, gdyż wcześniej nastąpi przełączenie. Nie wytworzy więc pełnego strumienia magnetycznego w rdzeniu, a to będzie przyczyną zmniejszenia momentu obrotowego silnika, a w przypadku krytycznym jego zatrzymania. Jest to ważny wniosek. Okazuje się, że moment obrotowy silnika krokowego zmniejsza się wraz ze wzrostem prędkości obrotowej (szybkości przełączania). W stałej czasowej rezystancja występuje w mianowniku, można więc sądzić, że jej zwiększenie skróci czas narastania prądu w uzwojeniu. Jednak dodatkowa rezystancja ograniczy maksymalną wartość prądu płynącego w uzwojeniu, a dodatkowo spowoduje wzrost strat.

Opisane zjawisko stanowi dość poważny problem dla konstruktorów i jest rozwiązywany innymi, dużo bardziej zaawansowanymi metodami. Dodatkowa rezystancja jest np. zastępowana źródłem prądowym, zwiększane jest napięcie zasilające, a nawet wprowadza się dodatkowe źródło zasilające, które jest na krótko uruchamiane w pierwszej fazie taktu. Spotykane są też sterowniki pracujące techniką choperową. Wytwarzają one serię impulsów o dużo większej częstotliwości od częstotliwości przełączania, ale zapewniają odpowiednio wysoki prąd średni. Wszystkie te metody są realizowane przez mocno specjalizowane sterowniki silników krokowych.

Warto jeszcze zwrócić uwagę na to, że praca silnika krokowego nie zawsze polega na ciągłym obracaniu wirnika, jak w zwykłym silniku. W ogólnym przypadku może być wymagane obrócenie rotora do określonej pozycji i zatrzymanie go. Aby jednak było to możliwe dla obciążonego silnika, w ustalonej pozycji przez uzwojenia musi przepływać prąd nawet rzędu kilku amperów. Silniki nagrzewają się więc często do wysokich temperatur. Czynnik ten powinien być brany pod uwagę przy projektowaniu aplikacji.

Gubienie kroków

Urządzenia z silnikami krokowymi w wielu przypadkach mogą pracować w otwartej pętli. Oznacza to, że mikrokontroler sterujący nie otrzymuje żadnej informacji zwrotnej o bieżącym położeniu wirnika. Nie stanowi to żadnego problemu w prawidłowo działającym urządzeniu, gdyż procesor wysyła takie sygnały sterujące, które powinny skutkować obrotem wirnika do założonej pozycji. Niejako więc



Rysunek 18. Przebiegi prądowe w uzwojeniach silnika krokowego przy sterowaniu mikrokrokowym

procesor zna pozycję rotora. W praktyce mogą wystąpić jednak sytuacje, które zakłócą pracę takiego algorytmu. Wiemy już, że moment obrotowy silnika spada wraz z prędkością. Może się zdarzyć, że obciążenie silnika okaże się zbyt duże, aby rotor zajął założoną pozycję. Mówimy w takim przypadku o gubieniu kroków. Rozwiązaniem tego problemu jest dobranie silnika o odpowiedniej mocy (zapewniającej wystarczający moment obrotowy dla największego dopuszczalnego obciążenia) albo zastosowanie jakiegoś rozwiązania ze sprzężeniem zwrotnym, np. z użyciem enkodera.

Scalone sterowniki silników krokowych

Wiemy już, że do sterowania silnika krokowego potrzebny jest układ generujący wymaganą sekwencję sterującą oraz stopień końcowy o wymaganej wydajności prądowej. W prostych aplikacjach do generowania przebiegów w zupełności może wystarczyć mikrokontroler, jednak do realizacji bardziej wymagających algorytmów, np. do obsługi mikrokrokowej, pracy z dużymi prędkościami obrotowymi itp., zaangażowanie procesora może być zbyt duże. Zwykle wykonuje on jeszcze szereg innych zadań. Konieczne jest wówczas sięganie po specjalizowane sterowniki dostępne w postaci układów scalonych. Podobnie jest z końcówkami mocy. Można je realizować z dyskretnych tranzystorów bipolarnych lub MOSFET, lecz nie ma to większego uzasadnienia, gdyż bez problemu można je zastąpić scalonymi mostkami H. Znacznie ułatwiają projektowanie sterowników nawet dwufazowych silników bipolarnych a ich ceny nie są wysokie. Przykładowe układy tej grupy to: popularny driver ULN2803A, a także L293D, L297, L298, PBD3517/1, TLE5205.

Do bardziej wymagających aplikacji warto natomiast dobrze zapoznać się z ofertą scalonych kompletnych sterowników przystosowanych do obsługi różnych typów silników, umożliwiających pracę mikrokrokową z optymalizacją prądu uzwojeń (np. choperowych) itp. Przykładem niech będą sterowniki firmy Trinamic. Są to układy scalone, jak również gotowe moduły przeznaczone do aplikacji sterowania ruchem. Komunikują się z urządzeniem nadrzędnym (mikrokontrolerem) za pośrednictwem specjalnie opracowanego protokołu TMCL (Trinamic Motion Control Language), który może być bardzo łatwo implementowany we własnych systemach użytkownika. Termin „sterowanie ruchem” oznacza, że układy te mają zaszyte algorytmy optymalizujące przyspieszenia i opóźnienia przemieszczającego się za pośrednictwem silników krokowych obiektu. Są w nich wykorzystywane m.in. charakterystyki typu: 3-punktowa, 6-punktowa i s-shape. Obszerny opis zamieściliśmy w EP 10/2018.

Serwonapędy

Urządzenia wykonawcze służące do kontroli ruchu i precyzyjnego pozycjonowania nazywane są serwonapędami. To podstawowe elementy

współczesnych systemów automatyki przemysłowej. Pracują w zamkniętej pętli sprzężenia, w której źródłem informacji zwrotnej jest enkoder, czujnik Halla lub rezolwer (czujnik kąta). Serwonapęd zapewnia precyzyjne sterowanie z zachowaniem dużej dynamiki ruchu. Podstawowym elementem serwonapędu jest serwosilnik. Najczęściej jest to bezszczotkowy silnik synchroniczny prądu przemiennego lub prądu stałego. W razie konieczności jest on uzupełniany przekładnikami planetarnymi i specjalnymi hamulcami (np. elektromagnetycznymi). Spotykane są również rozwiązania z silnikami krokowymi. Asortyment serwosilników i serwosterowników jest bardzo bogaty. Dostępne są zestawy przystosowane do pracy z prądami dochodzącymi do kilkudziesięciu amperów. Ze względu na możliwość generowania silnych zakłóceń elektromagnetycznych często są wyposażane w filtry EMC. Z kolei praca w ciężkich warunkach przemysłowych narzuca wymóg zapewnienia wysokiego stopnia ochrony dla obudów (IP54, IP55).

Servonapęd, a właściwie jego sterownik, reaguje w czasie rzeczywistym na ewentualne zmiany obciążenia wału serwosilnika, zmiany napięcia zasilającego lub inne zakłócenia, odpowiednio je kompensując i zawsze ustawiając położenie wału w zadanym położeniu. Moce serwonapędów używanych w automatyce przemysłowej osiągają wartości nawet kilku kilowatów, a typowe prędkości obrotowe są rzędu 3000 obr./min dla pracy ciągłej i 5000 obr./min dla pracy chwilowej. Rozróżniane są również wartości chwilowe i ciągle uzyskiwanych momentów siły. Przykładowe nominalne wartości momentów siły dla serwomechanizmów o mocach rzędu kilkuset watów zawierają się w przedziale od ok. 0,5 Nm do ok. 3 Nm, ale np. serwomechanizm SMH180D-0440 produkowany przez firmę Kinco o mocy 4,4 kW charakteryzuje się nominalnym momentem siły wynoszącym 28 Nm.

Servonapędy są instalowane w systemach automatyki zawierających wiele różnych urządzeń, nie tylko serwonapędów. Poszczególne elementy muszą się komunikować ze sobą, jeśli nie bezpośrednio, to przez jakiś element nadzorujący. Do komunikacji są stosowane interfejsy sieciowe, takie jak: Profibus, Profinet, CAN, Modbus, EtherCAT czy podobne. Niektórzy producenci opracowują też własne protokoły.

Świetlana przyszłość automatyki

Opisane w artykule silniki krokowe i serwonapędy to nierozłączne elementy automatyki nie tylko przemysłowej. Już dziś w każdym biurze, a nawet domu można znaleźć kilka urządzeń, w których istotną funkcję pełnią silniki krokowe (czytniki płyt CD, drukarki). Często użytkownicy nawet o tym nie wiedzą, choć mają do tego prawo, gdyż prawdopodobnie większość osób określanych jako nietechniczne nawet nie wie, co to są silniki krokowe.

O rozwój silników krokowych i serwonapędów w sprzęcie profesjonalnym raczej nie musimy się martwić – wymuszają go po prostu potrzeby rynku i stale rosnące wymagania użytkowników. Od pewnego czasu można obserwować rosnące zainteresowanie silnikami krokowymi wśród hobbystów. Powstają amatorskie konstrukcje obrabiarek CNC, ploterów czy nawet drukarek 3D. Większość obsługiwana przez tanie sterowniki dostępne w wielu sklepach internetowych i oczywiście Arduino lub Raspberry Pi. Taką działającą aplikację można wykonać w jeden wieczór, a eksperyment może być doskonałą zachętą do zainteresowania się automatyką. Ta profesjonalna jest jednak o wiele bardziej skomplikowana i wymagająca bardzo obszernej i niełatwej do opanowania wiedzy.

Jarosław Doliński, EP

REKLAMA

m.technik
Ciekawi świata są zawsze młodzi
w prezencie na każdą okazję

przejrysz i kupisz na
www.ulubionykiosk.pl

UWAGA SZPIEG!
łatwiej powiedzieć, kto cię nie śledzi

KAPOTA: Jak handlować w XXI wieku
Zakupy online i offline – różnica zanika

ELI WHITNEY
bawelniany ołciec rewolucji

nr 12, grudzień 2019

m.technik
Ciekawi świata są zawsze młodzi

<http://bit.ly/2DKgsBJ>

Dylemat konstruktora

Specjalizacją naszej firmy są maszyny do automatycznego pakowania. Są to konstrukcje niestandardowe i większość podzespołów i komponentów produkujemy sami. Opracowaliśmy wiele autorskich rozwiązań, które stale unowocześniamy. Interesuje nas współpraca z dostawcą dysponującym szeroką ofertą serwonapędów oraz niezbędnego osprzętu.

Polecane rozwiązanie

Serwonapędy to układy wykonawcze stosowane w automatyce przemysłowej, maszynach i na liniach produkcyjnych ze względu na możliwość precyzyjnego pozycjonowania, sterowania prędkością oraz dużą dynamikę ruchu. Serwonapęd pracuje w pętli sprzężenia zwrotnego i składa się z trzech podstawowych elementów: silnika, sterowania oraz enkodera. Wysokowydajne serwonapędy oferowane przez WObit charakteryzują się dobrymi właściwościami dynamicznymi i regulacyjnymi, a także płynną, precyzyjną pracą. Kilka serii silników o znamionowym momencie obrotowym od 0,32 Nm aż do 48 Nm wraz ze sterownikami serwo o szerokich możliwościach gwarantuje wybór odpowiedniego napędu. Nowością jest seria silników wyposażona w enkoder absolutny jednoobrotowy lub wieloobrotowy. Enkoder absolutny pamięta

pozycję w przypadku wyłączenia zasilania, zatem serwonapęd może kontynuować pracę bez konieczności bazowania, co sprawdza się we wszelkich układach zawierających więcej niż jedną oś.



... Więcej informacji

Zespół firmy WObit tworzą specjaliści posiadający duże doświadczenie i wiedzę inżynierską w dziedzinach takich jak pomiary, napędy i sterowanie, robotyka oraz mechanika. Niezwykle bogata oferta produktów zawiera całą gamę czujników, od niezbędnych w automatyce czujników pomiaru kąta i przemieszczenia liniowego, przez czujniki siły i ciśnienia, skanery laserowe, po czujniki zbliżeniowe, momentu czy czujniki koloru. W zakresie napędów firma proponuje silniki prądu stałego szczotkowe o mocach od 1 W do 300 W,

silniki bezszczotkowe do 1100 W, a także silniki DC z przekładniami i silniki krokowe 2-, 3- i 5-fazowe, oraz serwonapędy. Ofertę uzupełniają przydatne konstruktorom i elektronikom elementy z tworzywa sztucznego, panele HMI i sterowniki PLC oraz niezbędne w układach pozycjonowania prowadnice liniowe i śruby kulowe. WObit to firma o dużym doświadczeniu, gotowa dostarczyć pełne spektrum najnowocześniejszych produktów w dziedzinie automatyki.

Publikacje:

www.automatykab2b.pl:

- Realizacja przemieszczeń liniowych – moduły liniowe firmy WObit
- Precyzyjna kontrola ruchu w oparciu o silniki krokowe firmy WObit z zadajnikami trajektorii ruchu
- Jak zbudować robota? Komponenty robotyki przemysłowej w ofercie WObit

Kontakt:

P.P.H. WObit E.K.J. Ober s.c.
62-045 Pniewy, Dęborzyce 16
tel. 61 22 27 410
faks 61 22 27 439
wobit@wobit.com.pl
www.wobit.com.pl, www.silniki.pl

Dylemat konstruktora

Nasza firma produkuje urządzenia do precyzyjnego dozowania substancji takich jak pasty lutownicze, kleje, smary, masy zalewowe i uszczelniające. Substancje muszą być dozowane w dokładnie ustalonej dawce, bez rozlewania i ściekania, dlatego napęd mechanizmu tłoczącego musi zapewniać wysoką dynamikę oraz dawać możliwość precyzyjnej kontroli. Nawiązemy współpracę z dostawcą silników BLDC oraz niezbędnego osprzętu.

Polecane rozwiązanie

OEM Automatic posiada w swojej ofercie silniki produkcji firmy Crouzet. Są to wysokowydajne silniki bezszczotkowe nowej generacji o mocy wyjściowej od 34 do 192 W. Dostępne są modele z wbudowanym sterownikiem prędkości TNi21 oraz regulatorem pozycji SMi21. Łącząc w sobie doskonałą gęstość mocy, zintegrowaną elektronikę oraz liczne funkcje sterowania

prędkością, momentem obrotowym i pozycją, silniki BLDC z serii DCmind idealnie nadają się do zastosowań w różnego rodzaju aplikacjach przemysłowych. Charakteryzują się wyjątkowo długą żywotnością (ponad 20000 godzin pracy), bardzo niskim poziomem hałasu <50 dBA, wysoką sprawnością (zoptymalizowane sterowanie silnikiem) oraz doskonałą dynamiką pracy.

... Więcej informacji

OEM Automatic jest częścią szwedzkiego koncernu OEM International – firmy powstałej w 1974 roku, która posiada obecnie 34 oddziały na całym świecie. Jest wiodącym dostawcą komponentów i systemów dla największych odbiorców automatyki przemysłowej i elektroniki w Europie.

Oferta skierowana jest do producentów oraz użytkowników maszyn, szaf sterowniczych i rozdzielnic, integratorów systemów automatyki oraz odbiorców komponentów elektronicznych. Jej wyjątkowość wynika z połączenia wysokiej jakości produktów, profesjonalnej obsługi i specjalistycznej wiedzy zatrudnianych specjalistów produktowych i inżynierów sprzedaży. Największymi atutami firmy są: doświadczenie i znajomość lokalnego rynku, know-how oraz wysoki poziom wiedzy o produktach i aplikacjach,

efektywna logistyka, aktywne wsparcie systemowe na poziomie projektowania i wdrażania, a także wysokiej jakości materiały marketingowe i informacyjne.



Publikacje:

www.automatykab2b.pl:

- OEM Automatic przeprowadził się do nowej siedziby
 - Silniki BLDC firmy Crouzet serii DCmind ze zintegrowaną elektroniką sterującą oraz interfejsem komunikacyjnym CANopen
 - Miniaturowe siłowniki elektryczne firmy Actuonix
- IRA 2020, str. 232

Kontakt:

OEM Automatic Sp. z o.o.
02-234 Warszawa, ul. Działkowa 121 A
tel. 22 863 27 22, faks 22 863 27 24
info@pl.oem.se, www.oemautomatic.pl

Dylemat konstruktora

Nasza firma jest pionierem w produkcji systemów automatyki laboratoryjnej. Korzyści z automatyzacji w tym obszarze są oczywiste: pozwalają osiągnąć szybsze wyniki, wyższą wydajność, mniej błędów i niższe koszty personelu. Naszym produktom stawiane są wygórowane wymagania funkcjonalne a jednocześnie muszą one spełniać najwyższe standardy bezpieczeństwa i jakości. Nawiązemy współpracę z dostawcą precyzyjnych i sprawdzonych systemów napędowych.

Polecane rozwiązanie

W napędach Ezi-SERVOII-PLuSE zastosowano silniki krokowe, które dzięki zabudowanemu enkoderowi pracują w zamkniętej pętli sprzężenia zwrotnego. Takie rozwiązanie eliminuje możliwość zgubienia kroku nawet przy nagłych zmianach obciążenia. Złożone algorytmy filtracji realizowane przez procesor sygnałowy i ciągła kontrola wektora prądu pozwalają na uzyskiwanie płynnego ruchu obrotowego wału silnika od 0,2 obrotu/minutę. Dzięki regulacji prądu silnika, w zależności od zmieniającego się obciążenia, udało się ograniczyć ciepło wytwarzane w silniku.

Sterowniki mają funkcję tabeli pozycji. Lista złożona maksymalnie z 256 wierszy danych zawiera informacje o ruchu i zapisywana jest w pamięci flash-rom sterownika. W pojedynczym wierszu oprócz pozycji docelowej i prędkości obrotowej znajdują się m.in. takie informacje, jak czas przyspieszania i zwalniania, numer wiersza, który ma być wykonany w następnej kolejności, czy licznik powtórzeń umożliwiający realizację pętli. Zapisaną tabelę pozycji możemy uruchomić za pomocą portu Ethernet lub wejścia cyfrowego. Uzyskany w ten sposób pozycjoner może pracować bez potrzeby stosowania sterownika nadrzędnego.

... Więcej informacji

Firma Eldar istnieje na rynku od końca lat dziewięćdziesiątych. Oferuje szeroki asortyment produktów z branży automatyki przemysłowej, za cel stawia sobie dostarczanie kontrahentom tylko sprawdzonych produktów. W ofercie znajdziemy m.in. serwonapędy, napędy krokowe i serwokrokowe, a także zasilacze, czujniki, przekładnie i sterowniki. Oferta skierowana jest głównie do producentów maszyn, biur konstruktorskich oraz firm, których działalność związana jest z serwisem i modernizacjami maszyn.

Staranny dobór dostawców skutkuje ciągłym rozwojem firmy. Do współpracy zostali zaproszeni dostawcy z Włoch, Korei, Izraela, Słowenii oraz Tajwanu, dzięki czemu oferta firmy Eldar obejmuje produkty takich marek jak Lika, Fastech, LiteON, Servotronic czy Riko.



Publikacje:

www.automatykab2b.pl:

- Zintegrowane z napędami silniki krokowe z interfejsem CANopen
- Sterowanie napędami Ezi-Servo Plus-R za pomocą interfejsu szeregowego RS485
- Ezi-SERVO – więcej niż napęd

Kontakt:

Eldar
45-531 Opole, ul. Morcinka 51
tel. 77 442 04 04, tel./faks 77 453 22 59
eldar@eldar.biz, www.eldar.biz
www.fastech.pl

Oferta prenumeraty Elektroniki Praktycznej

**Prenumerata
wersji
drukowanej**

roczna
176,00 zł
(rabat 8%)

dwuletnia
256,00 zł
(rabat 33%)

**DO 50%
ZNIŻKI NA
PRENUMERATĘ!**

Prenumerata ze zniżką lojalnościową
za nieprzerwany staż prenumeraty
(opcja dostępna po zalogowaniu na www.avt.pl):
roczna (staż 1 rok) – 160,00 zł (rabat 16%)
roczna (staż 2 lata) – 144,00 zł (rabat 25%)
roczna (staż 3 lata) – 128,00 zł (rabat 33%)
dwuletnia (staż 3 lata) – 224,00 zł (rabat 41%)
dwuletnia (staż 5 lat) – 192,00 zł (rabat 50%)

Prenumerata e-wydań

- roczna – 91,80 zł (rabat 15%)
- dwuletnia – 151,20 zł (rabat 30%)

Prenumeratory wersji drukowanej za równoległe
e-wydania płać 20% ceny: opłata za e-prenumeratę
równoległą wynosi 21,60 zł/rok i 43,20 zł/2 lata.

Każdorazowo opłacenie prenumeraty jest premiowane prezentem.

W tym numerze są to do wyboru:

- koszulka z logo EP (roz. L, XL)
- płyta Krzysztofa Napiórkowskiego „Semi Electric”

Zamów swój prezent mailowo – prenumerata@avt.pl.



Przywileje Prenumeratorów

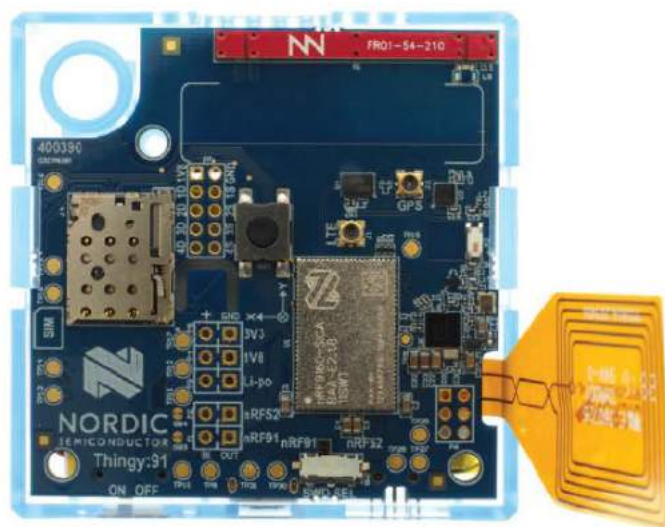
- masz prawo do rabatu od 30 do 50% na www.UlubionyKiosk.pl
(wystarczy podczas zamówienia podać numer swojej prenumeraty)
- masz prawo do rabatu na www.sklep.avt.pl
- z każdym numerem swojej prenumeraty otrzymasz DVD Niezbędnik Elektronika

Zamów prenumeratę EP w dogodny dla siebie sposób:

- mailowo – prenumerata@avt.pl
- przelewem na konto: AVT-Korporacja sp. z o.o.,
ul. Leszczyńska 11, 03-197 Warszawa, ING Bank Śląski 18 1050 1012 1000 0024 3173 1013
- a najlepiej na **WWW.AVT.PL** – jeśli zamawiasz po raz pierwszy na www.avt.pl lub przedłużysz
ją po zalogowaniu do swojego Panelu Prenumeratora, otrzymasz dodatkowo kody rabatowe
na bezpłatne pobranie dowolnych czasopism w wersji elektronicznej ze strony www.UlubionyKiosk.pl

Szanowny Kliencie, od 25 maja 2018 roku w krajach Unii Europejskiej obowiązuje Ogólne rozporządzenie o ochronie danych osobowych (RODO). Zachęcamy do zapoznania się z poniższą klauzulą informacyjną.

Administratorem Twoich danych jest AVT-Korporacja sp. z o.o. z siedzibą ul. Leszczyńska 11, 03-197 Warszawa, e-mail: prenumerata@avt.pl. Chodzi o dane osobowe, które zbieramy, aby móc wysłać Ci nasze czasopisma w formie drukowanej lub elektronicznej oraz inne towary (np. prezenty), a także w innych prawnie usprawiedliwionych celach, w tym marketingu bezpośredniego naszych produktów i usług (tzw. uzasadniony interes administratora). Podanie danych jest dobrowolne, ale niezbędne do zrealizowania zamówienia na prenumeratę. Twoje dane osobowe mogą być przekazane Poczcie Polskiej, która będzie dostarczać do Ciebie przesyłki. Bez Twojej zgody nie prześlemy i nie będziemy dokonywać obrotu (nie użyczymy, nie sprzedamy) Twoich danych osobowych innym osobom lub instytucjom. Twoje dane osobowe możemy przekazać jedynie podmiotom uprawnionym do ich uzyskania na podstawie obowiązującego prawa (np. sądy lub organy ścigania) – ale tylko na ich żądanie w oparciu o stosowną podstawę prawną. Będziemy przetwarzać Twoje dane osobowe przez 5 lat od zakończenia roku obrotowego, w którym wystąpiła ostatnia płatność. Dane osobowe do celów marketingowych będziemy przetwarzać do czasu wycofania przez Ciebie zgody na przetwarzanie lub do czasu usunięcia danych. Informujemy, że masz prawo do żądania od administratora dostępu do Twoich danych, ich sprostowania, usunięcia, ograniczenia ich przetwarzania, wniesienia sprzeciwu wobec przetwarzania Twoich danych lub ich przenoszenia. W każdej chwili możesz odwołać zgodę na przetwarzanie Twoich danych osobowych oraz możesz zażądać, by Twoje wszystkie dane zostały przez nas usunięte.



Podziękowania dla pana Macieja Michna z Centrum badań i rozwoju Nordic Semiconductor w Krakowie za udostępnienie modułów sprzętowych Nordic Thingy:91.

Systemy dla Internetu Rzeczy (31)

Nordic Thingy:91 – platforma do prototypowania dla mobilnego IoT

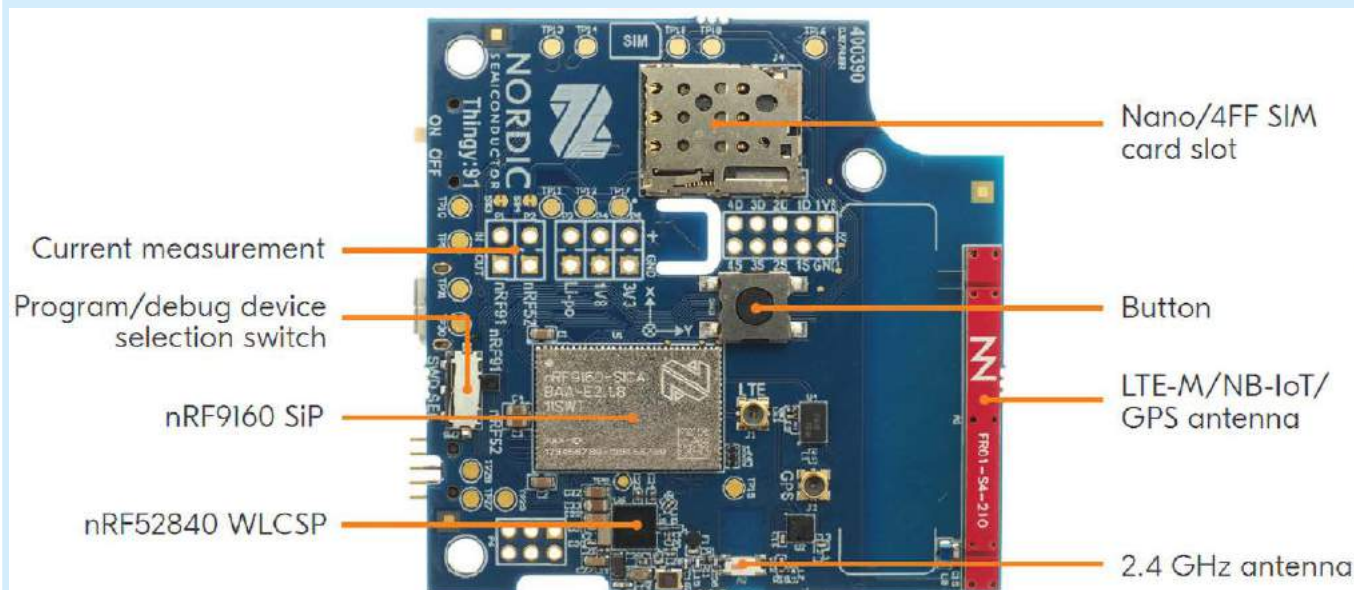


Rok 2019 jest rokiem komunikacji komórkowej LTE 5G. Jednak technologia LTE jest złożona, co może zniechęcać do sprawdzania jej możliwości. Właśnie pojawiła się nowa platforma szybkiego prototypowania dla mobilnej komunikacji IoT. Zestaw Nordic Thingy:91 firmy Nordic Semiconductor integruje dwie dziedziny komunikacji. Zastosowany w zestawie moduł nRF9160 SiP (*System in Package*) obsługuje komunikację dużego zasięgu i niskiej mocy standardu LTE-M, NB-IoT oraz dodatkowo GPS (*Global Positioning System*). Układ scalony nRF52840 SoC (*System on Chip*) obsługuje komunikację standardu Bluetooth Low Energy, Thread oraz NFC (*Near Field Communication*). Unikalną cechą zestawu jest osiągnięcie wysokich własności komunikacji radiowej przy zastosowaniu zintegrowanych anten wewnętrznych. Zestaw Thingy:91 zawiera szereg czujników: temperatury, wilgotności, ciśnienia, jakości powietrza, koloru i poziomu oświetlenia, dwa trzyosiowe czujniki ruchu oraz wiele elementów dodatkowych: przyciski, trzy diody RGB LED, głośniczek i antenę NFC. Zestaw Thingy:91 posiada globalną certyfikację do pracy w sieciach komórkowych. Jest on idealną platformą do prototypowania systemów komunikacji mobilnego IoT z zasilaniem bateryjnym i weryfikowania koncepcji. Zainstalowane fabrycznie oprogramowanie zapewnia śledzenie zasobów w czasie rzeczywistym.

Poprzedni zestaw Nordic Thingy:52 IoT Sensor Kit (PCA2020) spotkał się z bardzo dużym zainteresowaniem [S21]. Został w nim zastosowany układ scalony nRF5232 z obsługą protokołu Bluetooth Low Energy 5, NFC oraz ANT. Moduł Thingy:52 zawiera szereg czujników: temperatury, wilgotności, ciśnienia, jakości powietrza (CO₂ oraz TVOC), koloru

i poziomu oświetlenia, czujnik ruchu niskiej mocy i 9-osiowy czujnik ruchu. Postęp w dziedzinie układów MEMS jest tak duży, że obecnie wiele zastosowanych scalonych czujników zostało już wycofanych z produkcji.

Nowy zestaw Thingy:91 (PCA20035) firmy Nordic Semiconductor ma tą samą obudowę co poprzedni (6×6 cm) i obsługuje również te same



Rysunek 1. Płytką drukowaną zestawu Thingy:91, widok od góry

możliwości komunikacyjne oraz pomiarowe. Dodatkowo jest bardziej rozbudowany i posiada certyfikacje CE, FCC oraz globalną certyfikację do pracy w sieciach komórkowych [1]. Zawiera trzy osobne anteny: dla komunikacji NFC, dla komunikacji BLE oraz trzecią dla komunikacji GPS, LTE-M i NB-IoT. Producent udostępnia za darmo liczne aplikacje dla zestawu, kody źródłowe oprogramowania firmowego oraz schematy i projekty sprzętowe [4]. Standardy komunikacji mobilnej LTE-M oraz NB-IoT były omówione w poprzednim artykule serii [S22].

Moduł nRF9160 SiP

Moduł nRF9160 SiP integruje procesor ARM Cortex-M33 (64 MHz) z dużą pamięcią Flash do 1 MB oraz RAM do 256 KB [4]. Moduł zawiera modem LTE, układ GPS, układy peryferyjne i zarządzanie zasilaniem w obudowie o rozmiarach 10×16×1,2 mm. Oferuje kompletne rozwiązanie do pracy z komunikacją mobilną IoT. Wymaga tylko zewnętrznej baterii, karty SIM oraz anteny (pojedyncze wyprowadzenie 50 Ω).

- Modem obsługuje komunikację w standardach:
 - 3GPP LTE release 13: Cat-M1 (eMTC) oraz Cat-NB1 (NB-IoT),
 - 3GPP LTE release 14: Cat-NB1 (NB-IoT) oraz Cat-NB2 (NB-IoT),
 - Obsługuje tryby pracy z obniżonym poborem mocy: DRX, eDRX, PSM
 - Zakresy pracy LTE: 700-960 MHz, 1710-2200 MHz
- Układ obsługuje interfejs UICC (*Universal Integrated Circuit Card*). Oznacza to obsługę kart standardowych oraz eUICC (*Embedded UICC*).
- Układ wymaga napięcia zasilania w granicach 3,0...5,5 V (typ. 3,8 V).
- Posiada certyfikat GFC do pracy na całym globie
- Bezpieczeństwo: ARM TrustZone, ARM CryptoCell 310
- Wiele układów peryferyjnych: 4×SPI/UART/TWI, PDM, I²S, PWM, ADC
- Programowana moc wyjściowa do 23 dBm

Budowa wewnętrzna modułu była omawiana w poprzednich artykułach serii [S22, S27].

Układ scalony nRF52840

Układ scalony nRF52840 firmy Nordic Semiconductor bazuje na rdzeniu ARM Cortex-M4F (64 MHz) z dużą pamięcią Flash do 1 MB oraz RAM do 256 kB [5].

- Układ umożliwia w paśmie 2,4 GHz transmisję w sieciach z protokołem Bluetooth 5 (2 Mbps, 1 Mbps, Long Range), Bluetooth mesh, Thread, Zigbee, IEEE 802.15.4 oraz ANT.

- Możliwa jest jednoczesna obsługa transmisji z obsługą stosu BLE 5 oraz stosu OpenThread. Zapewnia to mechanizm Dynamic Multiprotocol z obsługą stosów S140 v5 SoftDevice oraz stosu OpenThread RF.
- Programowana moc wyjściowa w zakresie od -20 dBm do +8 dBm.
- Wiele układów peryferyjnych: UART, SPI, TWI, PDM, I²S, QSPI, PWM, 12-bit ADC, NFC-A, USB 2.0
- Bezpieczeństwo: 128-bit AES CCM, ARM CryptoCell
- Układ zapewnia bezpieczne bootowanie oraz aktualizację oprogramowania poprzez radio.
- Układ może pracować z różnymi systemami RTOS włącznie z systemem FreeRTOS.
- Układ pracuje z napięciem zasilania 1,7 V do 5,5 V.
- Układ ma certyfikację konsorcjum Thread Group.
- Cechy unikalne: Układ nRF52840 posiada dedykowany moduł kryptograficzny ARM TrustZone CryptoCell 310.

Budowa układu scalonego a szczególnie zagadnienia zasilania niskiej mocy były omawiane w poprzednim artykule serii [S29].

Środowisko programowe

Dla modułu nRF9160 są dostępne narzędzia programowe: nRF Connect for Cloud, LTE Link Monitor, nRF Connect for Desktop, Segger Embedded Studio oraz nRF Connect Software Development Kit (SDK) z przykładowymi aplikacjami. Pakiet programowy nRF Connect SDK zawiera wszystko co jest potrzebne do rozpoczęcia pracy. Aplikacja nRF Link Monitor pozwala na testowanie połączeń komunikacyjnych i uzyskiwanie informacji o pracy sieci.

Dokumentacja

Dokumentacja dotycząca zestawu Thingy:91 jest dostępna poprzez portal firmy Nordic Semiconductor na stronie produktu „*Nordic Thingy:91, Cellular IoT Prototyping platform*” [1]. Znajduje się tam odnośnik do opisu zamieszczonego w pliku pdf: „*Nordic Thingy:91 Product Brief v1.2*”. Po kliknięciu na przycisk *Get Documentation* otwierana jest strona *Documentation library* z dostępem do kilku opisów w formacie plików pdf oraz stron html. Dokładny opis modułu Thingy:91 jest zawarty w dokumencie „*Nordic Thingy:91 User Guide v1.0*” [2].

Dolna część strony Thingy:91 jest zorganizowana w postaci trzech zakładek:

- Overview – Zwiera opisy i odwołania do modułu nRF9160 DK z układem nRF9160 serii nRF91 a także do procesorów nRF9160

SiP oraz nRF52840. Jest też odwołanie do artykułów informacyjnych, w tym bardzo ciekawy tekst „*Simplified development of cellular IoT prototypes is now possible in just a few days with launch of Nordic Thingy:91*” [3] oraz „*Simplify Cellular IoT Prototyping With Nordic Thingy:91*” [14].

- Download – Zawiera odnośniki do źródłowych plików aplikacyjnych. Schematy płytki i anteny NFC oraz pliki konstrukcyjne znajdują się w pliku zip „*Thingy91 – Hardware files 1_0_0*”. Zawiera on plik schematu zestawu „*PCA20035_Schematic_And_PCB, rev.1.0.0*”. Po kliknięciu na element na schemacie zestawu pokazywane jest okno ze szczegółowym.informacjami, włącznie z odnośnikiem do pliku opisu elementu.
- Get started – W polu *Connect and test* jest opisane postępowanie od wyjęcia modułu z pudełka do połączenia z aplikacją mobilną. Poniżej znajduje się kilka prezentacji wideo.

Opis modułu nRF9160 SiP znajduje się w pliku „*nRF9160 Product Specification v1.0*” dostępnym poprzez stronę produktu [4]. Opis układu scalonego SOC znajduje się w pliku „*nRF52840 Product Specification v1.1*” dostępnym poprzez stronę produktu [5].

Przygotowanie modułu do pracy

Moduł jest umieszczony w elastycznym pudełku w kolorze pomarańczowym. Zaslepka na bocznej ścianie pudełka umożliwia dostęp do gniazdka Micro-USB. Otwarta duża boczna ścianka zapewnia dostęp do przezroczystej plastikowej obudowy. W niej umieszczona jest bateria litowa dołączona do płytki dwoma przewodami. Na przeciwnej ścianie bocznej pudełka są otwory na gniazdko Micro-USB, wyłącznik zasilania oraz złącze debugowania. Płytkę drukowaną modułu jest mocowana w pudełku z użyciem trzech zatrzasków.

W narożniku plastikowego pudełka jest wykonany spory otwór. Służy on jako światłowód wyprowadzający światło dwóch diod LED RGB. Plastikowe pudełko posiada również światłowód doprowadzający światło z powierzchni pudełka do scalonego czujnika koloru umieszczonego na płytce drukowanej. Oraz światłowód wyprowadzający światło diody LED RGB na powierzchnię pudełka jako podświetlenie obiektu do badania koloru odbitego.

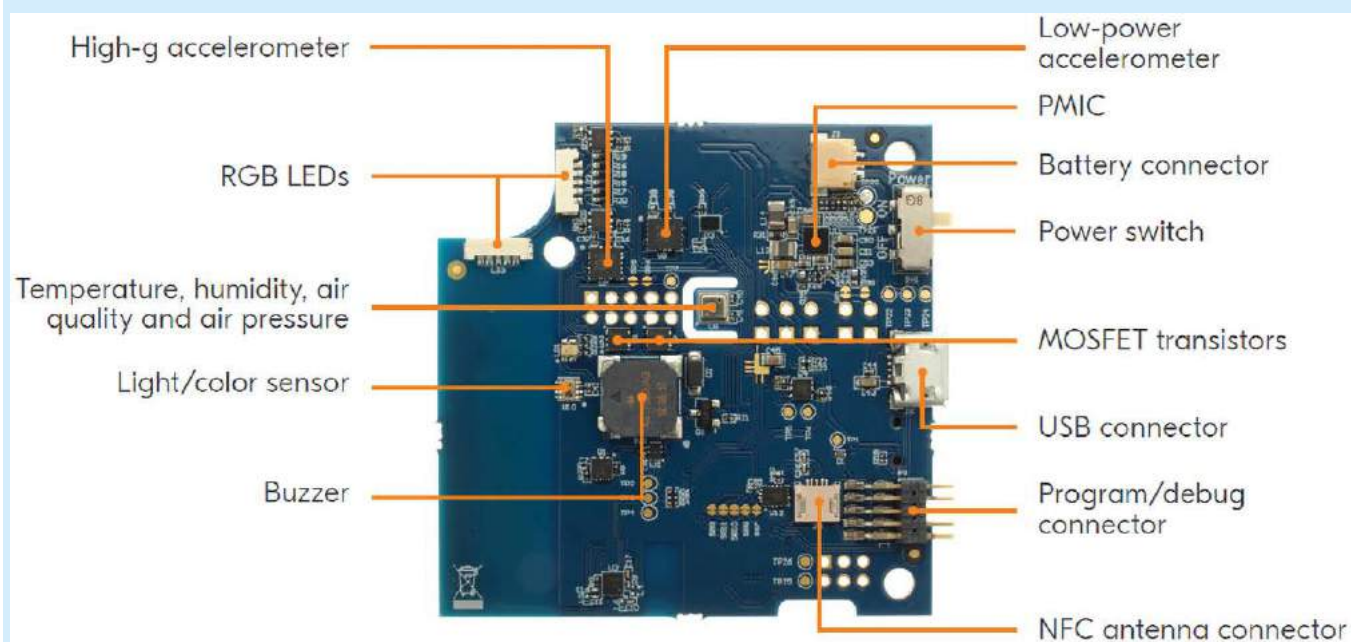
Na fotografii tytułowej jest widoczna pomarańczowa elastyczna obudowa oraz dolna strona płytki drukowanej zestawu Thingy:91 umieszczonej w obudowie plastikowej [2]. Widoczna jest też elastyczna antena NFC. Pierwszym krokiem do rozpoczęcia pracy z zestawem Thingy:91 jest zdjęcie pomarańczowego elastycznego

pudełka. Dopiero wtedy jest dostęp do wyłącznika zasilania Power. Do ładowania baterii jest przeznaczone gniazdko Micro-USB umieszczone na bocznej ścianie plastikowego pudełka. Gniazdko można dołączyć do gniazdka USB komputera lub do ładowarki USB. Bateria jest ładowana tylko gdy przełącznik POWER jest w pozycji ON. Po włączeniu zasilania modułu można założyć pomarańczowe elastyczne pudełko. Dostęp do gniazdka Micro-USB jest możliwy pod zasłepką na bocznej ścianie pudełka.

Przy zaglądaniu do „środka” należy zachować ostrożność. Po wyjęciu przezroczystej plastikowej obudowy z pomarańczowego elastycznego pudełka należy ostrożnie podważyć płytkę drukowaną na dwóch bocznych zatrzaskach. Podnoszenie płytki należy wykonać po przeciwnej stronie niż gniazdko Micro-USB.

Budowa modułu

Zestaw Thingy:91 zawiera szereg czujników: temperatury, wilgotności, ciśnienia, jakości powietrza, koloru i poziomu oświetlenia, dwa czujniki ruchu oraz wiele elementów dodatkowych: dwa przyciski, trzy diody RGB LED, głośniczek i antenę NFC. Zastosowane w module czujniki nie zostały dodatkowo skalibrowane (oprócz kalibracji producentów czujników). Góra płytki drukowanej modułu Thingy:91 jest pokazana na **rysunku 1**. Jest to strona przykryta pomarańczową elastyczną obudową. W środku płytki jest zamontowany przycisk SW3 umieszczony bezpośrednio pod ikoną graficzną firmy na powierzchni pomarańczowej obudowy. Przeciśnięcie w tym miejscu obudowy powoduje zadziałanie przycisku. Po tej samej stronie jest też umieszczony moduł nRF9160 SiP (U1), układ scalony nRF52840 (U6, w obudowie WLCSP), gniazdko karty SIM (J4, nano-SIM) oraz przełącznik SWD SELECT (SW2) wyboru układu do debugowania. Mała antena pasma 2,4 GHz typu 2450AT18D0100 (A2) jest umieszczona na dole płytki drukowanej. Na prawej krawędzi płytki umieszczona jest długa antena pasm LTE i GPS typu FR01-S4-210 (A1). Na prawo od modułu nRF9160 SiP (U1) umieszczone są dwa gniazdko typu SWF z wyłącznikiem do dołączania zewnętrznych anten w miejsce obu anten wbudowanych. Obok nich jest umieszczony mały przycisk SW4 służący do obsługi bootowania układu scalony nRF52840 (U6). Obok anteny A1 (puste miejsce na rysunku 1) jest na płytce przyklejona etykieta podająca kod modułu (PCA20035), wersję (1.0.0) oraz datę produkcji (2019.33). Poniżej jest podany numer IMEI, PIN oraz numery identyfikacji FCC.



Rysunek 2. Płytkę drukowaną zestawu Thingy:91, widok od dołu

Widok dolnej strony płytki drukowanej zestawu Thingy:91 jest pokazany na **rysunku 2**. W środku, w specjalnym wycięciu, jest umieszczony czujnik temperatury, wilgotności, jakości powietrza i temperatury BME680 (U9). Na lewo od niego jest umieszczony 3 osiowy czujnik przyspieszeń dużej wartości ADXL372 (U7). Na prawo jest umieszczony 3 osiowy czujnik przyspieszeń niskiej wartości ADXL362 (U8). U góry zostało umieszczone gniazdko dołączania baterii J3 oraz wyłącznik zasilania POWER (SW1). Obok jest zamontowany układ scalony zarządzania zasilaniem ADP536x (U11) typu PMIC. Poniżej jest zamontowane gniazdko USB-micro (J6). Jeszcze niżej jest zamontowane złącze do podłączania anteny NFC (J5) i złącze debugowania P8 (na krawędzi). W lewym górnym rogu, obok wycięcia w płytce, są umieszczone dwie diody LED RGB typu MSL0104RGBU1 (LD2, LD3). Poniżej znajduje się scalony czujnik koloru BH1749NUC (U10) oraz dioda LED RGB (LED1). Prawie w środku płytki znajduje się brzęczyk AST7525MATRQ (Bz1) sterowany poprzez tranzystor. Na obu stronach płytki drukowanej rozmieszczone są zworki, punkty pomiarowe oraz pola lutownicze dla dodatkowych złącz.

Schemat blokowy zestawu Thingy:91 został pokazany na **rysunku 3**. Zasilanie zestawu Thingy:91 jest pobierane z baterii typu Li-ion Polymer LP793444, 1440 mAh 3,7 V firmy Wamtechnik. Używamy nazwy „bateria” ale jest to oczywiście akumulator umożliwiający wielokrotne ładowanie. Złącze Micro-USB (5 V) dostarcza napięcia VBUS do układu scalonego ADP536x firmy Analog Devices. Służy on do ładowania akumulatora i dostarczania napięć zasilania zestawu. Napięcie VLi-Ion z baterii jest podawane na układ ADP536x, który dostarcza główne napięcie zasilania VSYS. Z niego jest zasilany bezpośrednio tylko moduł nRF9160 SiP. Układ ADP536x dostarcza też napięcia 1,8 V z którego jest zasilany układ scalony nRF52840 oraz czujniki przyspieszeń (ADXL362 i ADXL372) i zespolony czujnik BME680. Reszta układów zestawu Thingy:91 jest zasilana z napięcia 3,3 V również dostarczanego poprzez układ scalony ADP536x. To napięcie może być wyłączone po przejściu zestawu Thingy:91 w stan oszczędzania energii (*sleep mode*).

Pomiaru prądu zasilania modułu jest możliwy po rozłączeniu zwór SB1...SB4. Na płytce modułu Thingy:91 znajdują się pola lutownicze dla zamontowania dodatkowych złączy P3...P7. Moduł Thingy:91 zawiera też dodatkowo cztery tranzystory N-MOS (Q6, Q7) dołączone bezpośrednio do wyprowadzeń I/O układu nRF9160 oraz do złącza P7. Przełącznik SWD SELECT (SW2) wybiera czy standardowe 10-cio nóżkowe złącze debugowe P8 jest dołączone do modułu nRF9160 SiP czy do układu scalonego nRF52840. Pozwala to na ich programowanie i debugowanie.

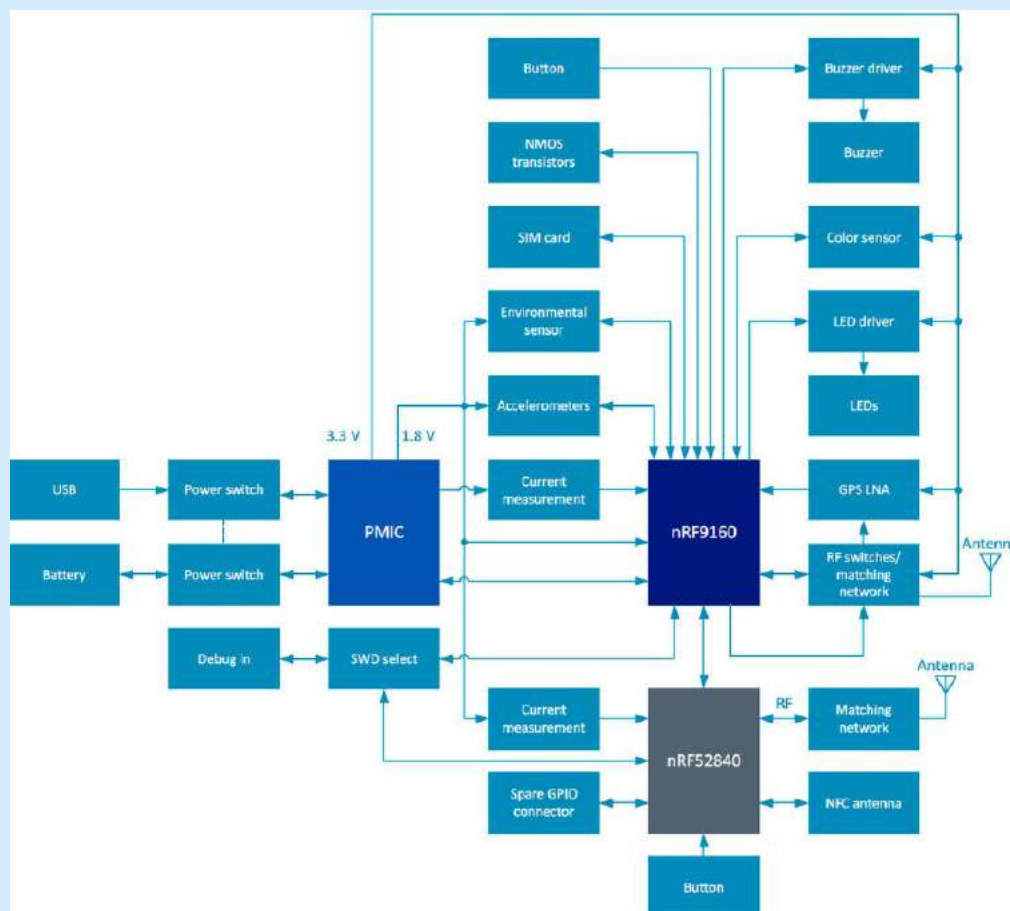
Moduł nRF9160 SiP pracuje w standardowej konfiguracji. Wyprowadzenie mocy radiowej ANT modułu jest doprowadzone do gniazdka typu SWF z wyłącznikiem. Umożliwia to dołączenie zewnętrznego

kabla koncentrycznego do wykonania pomiarów modułu, przy odłączonych układach płytki. Sygnał jest następnie dołączony do dynamicznego układu dopasowującego. Dwa multipleksery radiowe typu QM12038 (U2, U3) umożliwiają modułowi nRF9160 SiP wybór optymalnego układu dopasowania w zależności od pasma pracy [4]. Jedną z konfiguracji pracy układu dopasowania pozwala na pracę układu GPS modułu nRF9160 SiP. Dobre dopasowanie systemu antenowego jest bardzo istotne w systemach radiowych małej mocy. Tym bardziej, że stosowane w zestawach anteny mają typowo bardzo małe rozmiary. Ma to szczególne znaczenie dla pasm LTE poniżej 1 GHz. Obecnie dostępne aplikacje umożliwiają pracę zestawu Thingy:91 w następujących pasmach LTE: 2, 3, 4, 8, 12, 13, 20 oraz 28 [13].

Sygnał radiowy GPS, ze wspólnej z sygnałem LTE anteny A1, jest podawany do gniazdka typu SWF z wyłącznikiem. Umożliwia to dołączenie bardziej wydajnej anteny zewnętrznej. Dalej sygnał jest podawany na niskoszumny wzmacniacz U4 i wejście GPS modułu nRF9160 SiP. Wszystkie układy czujników są dołączone do modułu nRF9160 SiP.

Układ scalony nRF52840 pracuje w standardowej konfiguracji. Wyprowadzenie mocy radiowej ANT modułu jest doprowadzone do scalonego filtra SAFFB2G45MA0F0A (F1) typu SAW skonfigurowanego na pasmo ISM 2,4 GHz. Dalej sygnał jest dołączony do osobnej anteny A2.

Zestaw Thingy:91 jest wyposażony w antenę NFC dołączoną do gniazdka J5 i dalej bezpośrednio do dwóch dedykowanych wyprowadzeń układu scalonego nRF52840. Wyprowadzenia zasilania gniazdka USB-micro (J6) zestawu Thingy:91 są dołączone do układu zasilania zestawu (zasilanie całości i ładowanie baterii). Dostarcza ono także zasilanie dla modułu peryferyjnego USB układu scalonego nRF52840 (U6). Sygnały danych są bezpośrednio dołączone do wyprowadzeń D- oraz D+ układu scalonego nRF52840.



Rysunek 3. Schemat blokowy zestawu

Wyprowadzenia P0.18 do P0.25 modułu nRF9160 SiP są połączone z odpowiednimi wyprowadzeniami P0.11/15/20/2,1 P1.00, P0.25/19/22 nRF9160 SiP układu scalonego nRF52840. Zapewnia to bezpośredni interfejs komunikacyjny pomiędzy dwoma układami.

Diody RGB

Dwie diody RGB (LD2, LD3) są umieszczone w lewym górnym rogu dolnej powierzchni płytki drukowanej, obok wycięcia w płytce (rysunek 2). Pracują one równolegle, sterowane trzema sygnałami RGB z tranzystorem MOS jako kluczem w każdym kanale. Trzecia dioda RGB (LD1) jest umieszczona obok scalonego czujnika koloru BH1749NUC (U10). Wysyła ona oświetlenie pozwalające na pomiar koloru światła odbitego od przedmiotu umieszczonego koło obudowy.

Akcelerometri

Trzyosiowy akcelerometr ADXL362 firmy Analog Devices jest układem wykonanym w technologii MEMS [7]. Pobiera on nadzwyczaj mało prądu 1,5 μ A dla pomiarów 100 Hz oraz tylko 270 nA w trybie uśpienia (pomiar ok 6 Hz) z wykrywaniem zdarzeń. Posiada programowalny generator sygnału przerywania aktywowanego po wykryciu ruchu w trzech osiach lub swobodnego upadku. Układ umożliwia skalowanie zakresu pomiarowego od ± 2 g do ± 8 g oraz częstotliwości pomiarów od 12,5 Hz do 400 Hz. Obsługuje łącze komunikacyjne standardu SPI z osobnym napięciem do 1,8 V. Każde poruszenie modułu przez użytkownika może być wykryte i wybudza moduł nRF9160 SiP.

Drugi czujnik ruchu zestawu, trzyosiowy akcelerometr ADXL372 firmy Analog Devices też jest układem wykonanym w technologii MEMS [8]. Układ umożliwia pomiary do ± 200 g oraz częstotliwości pomiarów od 200 Hz do 3200 Hz.

Czujnik otoczenia

Do pomiaru jakości powietrza, temperatury, wilgotności oraz ciśnienia został zastosowany zintegrowany układ scalony BM680 firmy Bosch typu MEMS [6]. Jest on najbardziej skomplikowanym czujnikiem zamontowanym na płytce Thingy:91. Układ scalony ma bardzo małe wymiary 3×3×0,93 mm. Pobiera on tylko 0,15 μ A (uśpienie) do 12 mA (nagrzewanie) prądu (w zależności od trybu pracy) przy zasilaniu 1,70 V do 3,6 V. Układ obsługuje łącze komunikacyjne standardu I²C oraz SPI.

Wewnętrzny czujnik gazu tego układu scalonego zawiera płytkę podgrzewaną do odpowiedniej temperatury (typowo pomiędzy 200 oraz 400°C). Pozwala to na uzyskanie bardzo dobrej jakości pomiaru, bardzo krótkiego czasu opóźnienia rozpoczęcia pomiaru (poniżej 1 sek.) i bardzo znaczącej redukcji pobieranej mocy. Czujnik do pomiaru jakości powietrza wykrywa wiele różnych substancji lotnych. Gazy, które mogą być wykrywane przez BME680, obejmują lotne związki organiczne (VOC) – także wydychane przez człowieka, z farb (takich jak formaldehyd), lakiery, produkty do usuwania farby, środki czyszczące meble, sprzęt biurowy, kleje i alkohol [6].

Pomiary są podawane w jednostkach indeksu jakości powietrza IAQ (*Index for Air Quality*). Przyjmuje on wartości od zera do 500 z rozdzielczością 1. Wartości IAQ ok. 25 oznaczają dobrą typową jakość powietrza, wartości ok 250 oznaczają zanieczyszczone powietrze – mogące wywoływać bóle głowy.

Pomiar temperatury jest wykonywany w zakresie -40...85 °C z dokładnością ± 1 °C. Pomiar wilgotności jest wykonywany w zakresie 0...100 % z dokładnością ± 3 %. Pomiar ciśnienia atmosferycznego jest wykonywany w zakresie 300-1100 hPa z dokładnością $\pm 0,6$ hPa.

Czujnik koloru

Czujnik koloru BH1749NUC firmy ROHM Semiconductor posiada cztery osobne kanały pomiarowe: czerwony, zielony, niebieski oraz dodatkowo podczerwień [9]. Przetworniki ADC w każdym kanale

dają 16-bitowe wyniki pomiarów. Układ scalony pracuje z zasilaniem w zakresie 2,3 V do 3,6 V przy typowym poborze prądu 190 μ A. Układ umożliwia komunikację po szynie I²C (400 kHz) dla napięcia do 1,8 V. Układ cechuje się bardzo dużą rozdzielczością pomiaru, do 0,0125 Lx oraz bardzo dużym zakresem pomiarowym od 0,0375 Lx do 80000 Lx. Jest zamknięty w małej obudowie 2,10×2,00×0,6mm. Na płytce drukowanej obok czujnika koloru jest umieszczona dioda LED RGB, dzięki temu możliwy pomiar koloru światła odbitego od przedmiotu umieszczonego koło obudowy.

Rozpoczęcie pracy z zestawem Thingy:91

Na początek należy przestawić wyłącznik zasilania POWER (SW1) z położenia OFF do położenia ON (I). Jeśli bateria jest naładowana to zaczną błyskać diody LED na niebiesko.

Pierwsze dołączenie zestawu kablem USB do komputera skończyło się porażką – nie zostały znalezione sterowniki zastawu dla systemu Windows 7. Na systemach Windows 10 i Linux nie było problemów. Na portalu technicznym DevZone firmy Nordic Semiconductor jest (świeży) opis rozwiązania problemu [15]. Trzeba zainstalować pakiet programowy nRF5 SDK. Wtedy w folderze `\\nRF5_SDK_xx\\examples\\usb_drivers` znajdują się przydatne drajwery „*nordic_cdc_acm*”.

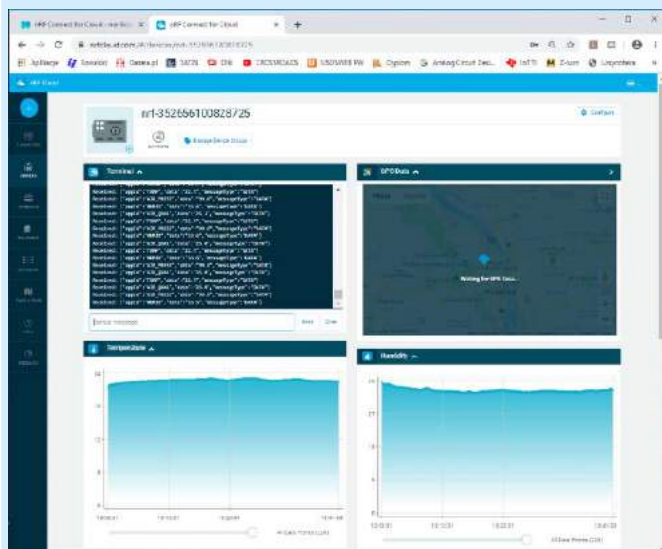
W oknie Menadżera urządzeń należy kliknąć na linię „Inne urządzenia → Thingy:91 UART” (pierwszą z dwóch) i wybrać „Aktualizuj Oprogramowanie Sterownika”. Następnie wybierz „Przeglądaj mój komputer ...” i wskaż folder ze sterownikami. Po zakończeniu instalacji pojawia się w Menadżerze urządzeń linia „nRF USB CDC Demo (COMxx)”. To jest właśnie port przeznaczony do komunikacji z zestawem Thingy:91.

Serwis nRF Connect for Cloud

W komplecie fabrycznym zestawu Thingy:91 dołączona jest karta nano/4FF SIM firmy iBasis powiązana z darmowym transferem 10 MB. W manualu do zestawu Thingy:91 [2] jest dokładnie opisany sposób aktywowania tej karty SIM. Służy do tego specjalna strona serwisu „nRF Connect for Cloud” [10]. Serwis umożliwia pracę z odbiornikiem GPS, aktualizowanie oprogramowania firmowego modułu nRF9160 a także układu nRF52840, aktualizację oprogramowania aplikacyjnego oraz daje wiele innych możliwości. W zasadzie jedyną, krótką informację o serwisie można znaleźć tylko na stronie produktu [16].

Postępowanie przy uruchamianiu pracy z serwisem:

1. Otwórz stronę „nRF Connect for Cloud” [10].
2. Zarestruj się.
3. Z menu, po lewej stronie okna, wybierz *Devices*.
4. Kliknij w menu na niebieski przycisk „+”.
5. W oknie *Add New* wybierz pozycję *LTE Device*.
6. W nowym oknie *Verify SIM Card* wpisz *SIM ICCID* oraz *PUK*. Numer ICCID jest wydrukowany bezpośrednio na karcie SIM. Numer PUK oraz PIN jest nadrukowany na plastikowej płytce dostarczanej z kartą SIM.
7. Przy wyłączonym zasilaniu zestawu Thingy:91 włóż załączoną w zestawie kartę SIM do gniazdka (J4, rysunek 1).
8. Włącz zasilanie wyłącznikiem *Power* (rysunek 2).
9. Zaznacz prostokąt potwierdzenia warunków obsługi.
10. Kliknij przycisk *Verify SIM*. Jeśli nie ma dostępu do obsługi mobilnego IoT to nic więcej nie da się zrobić. Do wykonania rejestracji zestawu potrzebne jest połączenie serwisu *nRF Connect for Cloud* z zestawem w celu wymiany kluczy.
11. W nowym oknie *Add LTE Device* wpisz IMEI oraz HWID. Oba numery są wydrukowane na naklejce umieszczonej na górnej powierzchni płytki drukowanej zestawu (rysunek 1).
12. Kliknij przycisk *Add Device*. Jeśli zestaw Thingy:91 jest w zasięgu aktywnej stacji bazowej LTE to zestaw i karta SIM zostaną dodane do serwisu. Na zakładce *SIM CARDS* są pokazywane informacje o zarejestrowanych kartach.



Rysunek 4. Serwis nRF Connect for Cloud podczas transmisji danych on-line z zestawu Thingy:91

13. Z menu, po lewej stronie okna, wybierz **DEVICES**. Wyświetlany jest zarejestrowany zestaw Thingy:91 z podaniem jego numeru IMEI.
14. Kliknij na ikonkę **Lte** po prawej stronie. Zostaną otworzone okna (rysunek 4):
 - Terminal – z informacjami o komunikacji,
 - GPS Data – z informacjami o lokalizacji geograficznej,
 - Asset Orientation – z informacjami o zmianie położenia zestawu oraz cztery okna z aktywnym on-line wykresem odczytów.

Są to okna:

- Temperature – odczyt temperatury,
- Humidity – odczyt wilgotności,
- Air Pressure – odczyt ciśnienia atmosferycznego oraz
- Reference Signal Received Power – poziom odbieranego sygnału radiowego (dBm). Ze względu na słaby sygnał zbierany z anteny moduł GPS może nie dostarczać danych.

Trzeba pamiętać, że pobieranie danych z zestawu Thingy:91 do chmury serwisu powoduje zużywanie predefiniowanej puli

darmowego transferu danych, która bardzo szybko się zużywa. Całość działa w sposób automatyczny, jednak jeśli chcemy zobaczyć szczegóły dotyczące sposobu pracy połączenia LTE to trzeba użyć innego programu.

nRF Connect for Desktop

Program *nRF Connect for Desktop* (wersja 3.2.0 lub wyższa) jest platformą do uruchamiania aplikacji (app) przeznaczonych do obsługi specyficznych zadań dla obsługi mobilnego IoT [11]. Po zainstalowaniu programu, przy pierwszym uruchomieniu, może pojawić się informacja o konieczności aktualizacji programu. Nawet jeśli została pobrana najnowsza wersja ze strony produktu. Aktualizację należy koniecznie wykonać. Następnie należy kliknąć na przycisk *Install* na prawo od napisu *LTE Link Monitor*.

LTE Link Monitor

Aplikacja *LTE Link Monitor v1.1.1* jest wtyczką (app) do programu *nRF Connect for Desktop* [11]. Komunikuje się z zestawem Thingy:91 poprzez port szeregowy realizowany z użyciem połączenia USB. Aplikacja jest przeznaczona do komunikacji z modemem LTE z użyciem poleceń AT. Pokazuje informacje o pracy dołączonego modemu jak: siła sygnału, informacje sieciowe i wiele innych. Jest bardzo dobrze opisana w manualu „*nRF Connect LTE Link Monitor v1.1.1 User Guide*” [12].

Po uruchomieniu aplikacji otwierana jest zakładka „*Terminal*” (rysunek 5).

Najpierw należy w prawym dolnym rogu okna aplikacji odznaczyć opcję filtru „*Auto device/port*”.

Następnie należy z menu „*Select device*” wybrać port COM (nRF USB CDC Demo) zestawu Thingy:91 (PCA20035). Pokazywane są dwa porty COM, ale tylko jeden jest aktywny. Aplikacja automatycznie łączy się z zestawem. Przebieg wymiany poleceń komunikacyjnych AT jest pokazany w dużym czarnym polu. W celu sprawdzenia połączenia należy w linii poleceń wpisać „*AT*” i kliknąć na przycisk „*Send*”. Poprawna odpowiedź zestawu to „*OK*”. Polecenia AT są opisane w manualu „*nRF91 AT Commands Reference Guide*”.

Jeśli nie ma dostępu do obsługi mobilnego IoT to w prawym górnym rogu okna aplikacji pulsuje kolorami pole LTE, co wskazuje na poszukiwanie sieci LTE. I nic więcej nie da się zrobić. Na dole

Wybrane pozostałe artykuły kursu Systemy dla Internetu Rzeczy

[S21] Zestaw Nordic Thingy:52 IoT Sensor Kit, EP 12/2018

[S22] Układy scalone z obsługą mobilnej komunikacji IoT, EP 1/2019

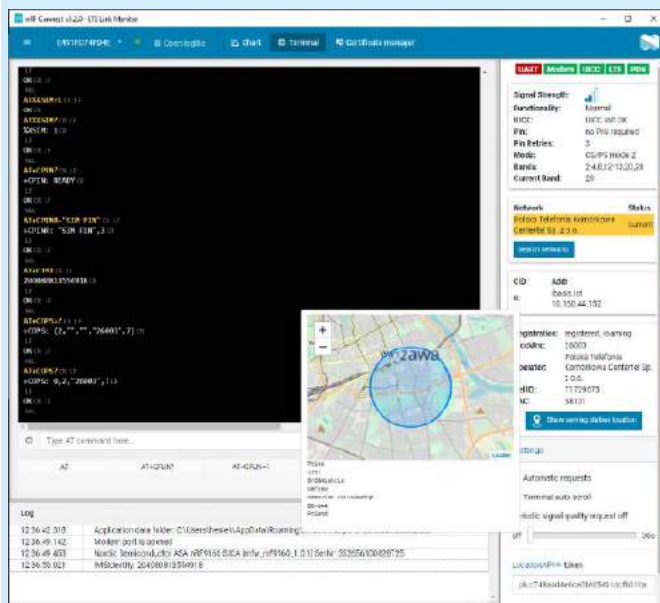
[S27] Wgląd w układy SoC (System on Chip), moduły SiP (System in Package) oraz elastyczne układy SoP (Silicon-on-Polymer), EP 6/2019

[S29] Zestaw uruchomieniowy nRF52840 DK, EP 9/2019

Literatura

- [1] Nordic Thingy:91, Cellular IoT Prototyping platform, Product Page, Nordic Semiconductor <http://bit.ly/2MVC14V>
- [2] Nordic Thingy:91 User Guide v1.0, 2019-08-22, Nordic Semiconductor <http://bit.ly/2WrgoMY>
- [3] Simplified development of cellular IoT prototypes is now possible in just a few days with launch of Nordic Thingy:91, 22 Aug 2019, Nordic Semiconductor <http://bit.ly/2JtTteF>
- [4] nRF9160 System-in-Package, Low power SiP with integrated LTE-M/NB-IoT modem and GPS, Nordic Semiconductor <http://bit.ly/2WD2N4o>
- [5] nRF52840 System on Chip, Advanced Bluetooth 5, Thread and Zigbee multiprotocol SoC, Nordic Semiconductor <http://bit.ly/2Y5kCc8>
- [6] BME680 Low power gas, pressure, temperature & humidity sensor, Bosch Sensortec <http://bit.ly/2jTKhH>
- [7] ADXL362 Micropower, 3-Axis, ± 2 g/ ± 4 g/ ± 8 g Digital Output MEMS Accelerometer, Analog Devices <http://bit.ly/34ajuT8>
- [8] ADXL372 Micropower, 3-Axis, ± 200 g Digital Output, MEMS, Analog Devices <http://bit.ly/3478oTj>

- [9] BH1749NUC Digital 16bit Serial Output Type Color Sensor IC, ROHM <http://bit.ly/2Pp8ty5>
- [10] nRF Connect for Cloud, Strona serwisu, Nordic Semiconductor www.nrfcloud.com
- [11] nRF Connect for Desktop, Product Page, Nordic Semiconductor <http://bit.ly/2Cugbz5>
- [12] nRF Connect LTE Link Monitor v1.1.1 User Guide v1.0, 2019-09-13 <http://bit.ly/2PqIV4W>
- [13] nRF Connect SDK, v1.0.99, Oct 03, 2019, Nordic Semiconductor <http://bit.ly/2MUBLYk>
- [14] Simplify Cellular IoT Prototyping With Nordic Thingy:91, Bjørn Kvaale, September 4, 2019, Nordic Semiconductor <http://bit.ly/2JHln5Z>
- [15] Thingy91 USB drivers, DevZone, 10.09.2019, Nordic Semiconductor <http://bit.ly/32Ss6CL>
- [16] nRF Connect for Cloud, Strona produktu, Nordic Semiconductor <http://bit.ly/2oniCAe>



Rysunek 5. Aplikacja LTE Link Monitor v1.1.1 dołączona do zestawu Thingy:91

okna aplikacji jest okno *Log* w którym pokazywane są informacje oraz komunikaty o błędach. W prawym panelu okna pokazywane są informacje dotyczące modemu oraz sieci komórkowej. Pierwsza linia pokazuje kolorami stan połączenia. Najechanie myszką na pole powoduje wyświetlenie listy dostępnych kolorów oraz ich znaczenie. Poniżej są pokazywane szczegółowe informacje, zaczynając od siły sygnału radiowego, używanego kanału LTE (20, 791...849 MHz), oraz danych właściciela sieci LTE (Orange) i stanu/typu połączenia komórkowego (roaming). Kliknięcie na przycisk *Show serving station location* powoduje wyświetlenie okna z informacją o lokalizacji używanej stacji bazowej wraz z mapką zasięgu (rysunek 5). Na zakładce *Chart* pokazywany jest wykres on-line pokazujący zdarzenia oraz siłę sygnału RSRP (dBm).

Jakiej informacji zabrakło? O używanym standardzie komunikacji, w tym przypadku był to standard LTE-M (Cat-M1).

Inne przydatne aplikacje

Aplikacja *Getting Started Assistant v1.0.2* służy do instalowania i konfigurowania środowiska programowego potrzebnego do pracy z projektami programowymi dla zestawu Thingy:91. Instalowany jest pakiet programowy „nRF Connect SDK” zawierający biblioteki i przykładowe projekty. Instalowane jest środowisko GNU ARM Embedded Toolchain oraz SEGGER Embedded Studio, a także oprogramowanie

dla systemu operacyjnego czasu rzeczywistego Zephyr. Instalacja trwa długo i jest bardzo skomplikowana.

Aplikacja *Programmer v1.2.3* służy do programowania modułów nRF9160 SIP z zastosowaniem zestawu nRF9160 DK jak również debugera J-Link. Jest jeszcze aplikacja *Trace Collector v1.0.0*, ale jest ona dopiero w trakcie udostępniania dla użytkowników i nie udało się uzyskać o niej informacji.

Podsumowanie

Większość projektów Internetu Rzeczy jest duża i złożona. Obejmują cały łańcuch zagadnień: czujników, urządzeń wbudowanych, łączności, chmury, analiz i informacji biznesowych. Wszystkie te obszary są złożone same w sobie i tylko największe firmy dysponują specjalistyczną wiedzą we wszystkich tych obszarach. Wiele młodych start-upów ma ograniczone zasoby i po prostu nie może z tym konkurować.

Problem polega na tym, że bez zbudowania działającego prototypu w celu wykazania rentowności, trudno będzie uzyskać finansowanie. Podstawową ideą Nordic Thingy: 91 jest zastąpienie jednego lub więcej z tych elementów w łańcuchu zagadnień, aby przetestować i udowodnić swój pomysł. Systemy wbudowane i komunikacyjne stają się coraz bardziej złożone, co jest szczególnie prawdziwe w przypadku sieci komórkowej. Thingy: 91 usuwa tę złożoność, a co najważniejsze, firma deklaruje, że całość działa od momentu otwarcia pudełka [13]. Jednak wydaje się, że deklaracja producenta o prototypowaniu w kilka dni, zamieszczona w artykule [3], wydaje się zbliżona do rzeczywistości. Przeprowadzone pierwsze próby z zestawem Thingy:91 wyglądały jakby „praca szła” zaraz po rozpakowaniu zestawu. Nie udało się jednak ustalić jak skorzystać z komunikacji BLE oraz NFC.

Zestaw Thingy:91 ma bardzo „elegancką” konstrukcję elektroniczną. Umożliwia komunikację na wiele sposobów: LTE-M/NB-IoT, GPS, BLE/Bluetooth mesh/Thread/Zigbee/IEEE 802.15.4/ ANT, NFC oraz USB/UART. Udostępnia bardzo nowoczesne scalone układy czujnikowe, szczególnie układ pomiaru jakości powietrza. Układ scalony zarządzania zasilaniem (ADP536x firmy Analog Devices) jest tak nowy, że nie jest jeszcze dostępny jego opis. Jednak to są dopiero pierwsze tygodnie po wprowadzeniu zestawu Thingy:91 na rynek. I podobnie jak w przypadku poprzedniego zestawu IoT tej firmy wsparcie użytkowe i programowe ułatwi życie programistom.

Dopiero teraz startuje w Polsce udostępnianie transmisji standardu LTE-M/NB-IoT. Powyższy opis jest tylko pierwszym spojrzeniem na zestaw Thingy:91. Kolejnym krokiem będzie rozpoczęcie programowania modułu SIP. Ale to już w przyszłości.

Henryk A. Kowalski
Instytut Informatyki
Politechnika Warszawska

REKLAMA

O projektach, mini, soft i wielu
innych diskutuj
na <https://forum.ep.com.pl>

ISIXRTOS v3 mini system operacyjny dla mikrokontrolerów rodziny M0/M3/M4/M7 (8)



Panel dotykowy

W poprzednim odcinku uruchomiliśmy wyświetlacz graficzny oraz przykładowe demo pokazujące możliwości graficzne biblioteki **libgfx**, jednak jak łatwo zauważyć brakuje nam jeszcze warstwy wejściowej, umożliwiającej wprowadzenie danych.

Wyświetlacz zestawu ewaluacyjnego STM32F469I-DISCO posiada pojemnościowy panel dotykowy, który możemy użyć jako źródło wprowadzania danych. Sercem panelu zintegrowanego z zestawem jest kontroler FT6x06 zapewniający następującą funkcjonalność:

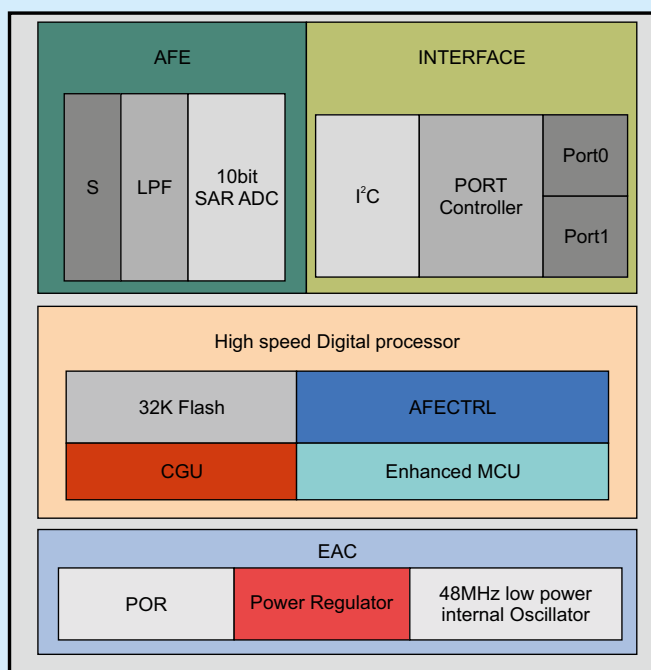
- obsługa bezwzględnych współrzędnych x, y,
- obsługę gestów oraz wielodotyku,
- wsparcie dla auto kalibracji, której zadaniem jest dostosowanie panelu dotykowego do warunków środowiskowych.

Od strony elektrycznej dostęp do rejestrów kontrolera dotykowego możliwy jest za pośrednictwem magistrali I²C oraz dodatkowej linii przerwań zewnętrznych, umożliwiającej zgłaszanie zdarzeń bez konieczności ciągłego sprawdzania zawartości rejestrów układu. Budowę wewnętrzną układu przedstawiono na **rysunku 1**. Blok AFE stanowi część sprzętową i odpowiedzialny jest za interakcję z panelem dotykowym. Zawiera on blok generatora sygnałów szybkochybiennych oraz analogową część odbiorczą, umożliwiającą detekcję dotyku opartą o zmiany pojemności. Mikrokontroler z wbudowaną pamięcią ROM zawiera program zajmujący się obsługą panelu oraz odciażający procesor główny od monotonnej czynności ciągłego skanowania panelu. Komunikacja z hostem zewnętrznym odbywa się za pomocą magistrali I²C. Sposób podłączenia kontrolera do mikrokontrolera w zestawie ewaluacyjnym przedstawiono na **rysunku 2**.

Linie SDA i SCL panelu dotykowego zostały dołączone do portów PB8, PB9, które w trybie alternatywnym stanowią interfejs magistrali I²C1 mikrokontrolera. Linia przerwania informująca o wystąpieniu zdarzenia została podłączona do portu PJ5, który może być używany przez kontroler przerwań zewnętrznych EXTI. Sposób dołączenia panelu dotykowego do kontrolera odbywa się w sposób klasyczny, tak jak w przypadku każdego urządzenia z magistralą I²C.

Oprogramowanie

Za obsługę i generowanie zdarzeń od urządzeń peryferyjnych w bibliotece **libgfx** odpowiedzialny jest podsystem **input**. Hierarchię klas tego podsystemu przedstawiono na **rysunku 3**. Klasę bazową dla wszystkich klas pochodnych stanowi klasa **input_dev** umożliwiająca wstrzykiwanie zdarzeń do podsystemu graficznego. Z klasy bazowej dla urządzeń wejściowych dziedziczą poszczególne kategorie urządzeń takie jak: panel dotykowy, klawiatura oraz myszka. Tworząc

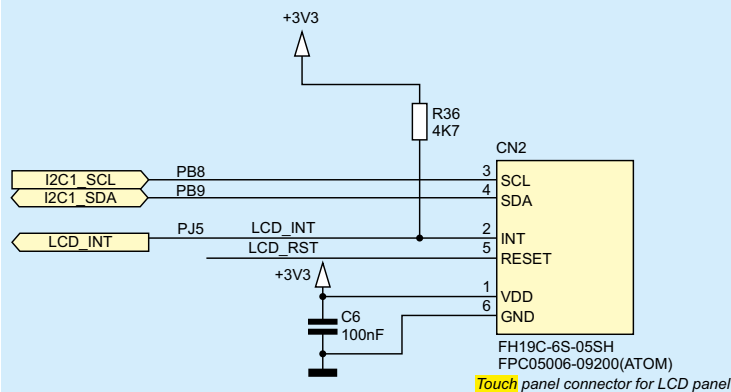


Rysunek 1. Budowa kontrolera dotykowego FT6x06

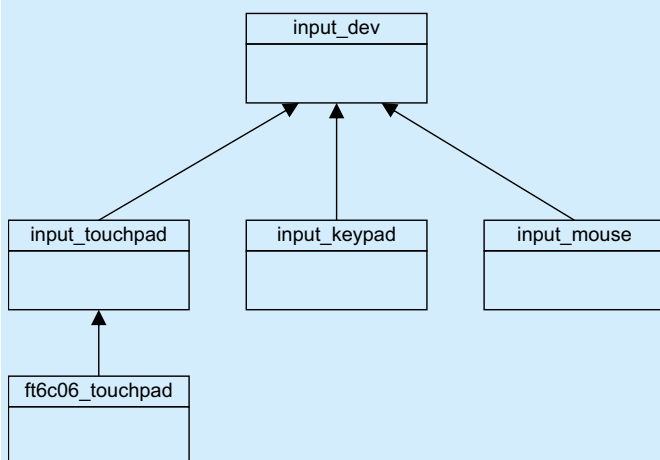
sterownik urządzenia wejściowego dla biblioteki graficznej należy dziedziczyć z wybranej kategorii urządzenia wejściowego. W przypadku panelu dotykowego klasa **ft6x06_touchpad** odpowiedzialna za obsługę panelu dotykowego zestawu a jej jedynym publicznym interfejsem jest jej konstruktor oraz metoda **start()** umożliwiająca uruchomienie wątku odpowiedzialnego za obsługę kontrolera (**listing 1**).

Listing 1. Konstruktor oraz metoda umożliwiająca uruchomienie wątku odpowiedzialnego za obsługę kontrolera

```
public:
    //Constructor
    ft6x06_touch(const char disp_name[],
                periph::drivers::i2c_master& i2c,
                gui::frame& frame);
    // Start the main thread
    void start() noexcept;
```



Rysunek 2. Podłączenie panelu dotykowego do mikrokontrolera w zestawie STM32F469I-DISCO



Rysunek 3. Hierarchia klas podsystem.in.ut w bibliotece libgfx

Konstruktor przyjmuje szereg argumentów potrzebnych do działania sterownika i są to odpowiednio: nazwa panelu konfiguracyjnego, która jest później wykorzystywana przez podsystem DTS do odczytu parametrów konfiguracyjnych, referencja do obiektu kontrolera I²C oraz referencja do klasy frame, która odpowiedzialna

```
Listing 2. Metoda thread() odpowiedzialna za obsługę zdarzeń
oraz odczyt danych z kontrolera
// Main thread for read input events
void ft6x06_touch::thread() {
    int ret;
    //Initialize touchpad hardware into the poll mode
    if((ret=initialize()) {
        dbg_err("Unable to configure touchpad errno: %i",ret);
        return;
    } else {
        dbg_info("Touchpad initialized");
    }
    for(touch_stat ts={};ts={}) {
        isix::wait_ms(50);
        ret = get_state(ts);
        if(ret>0) { //! If number of touches is greater than one
            //Print touch info
            print_touch(ts);
            //Report touch to the gui library
            report_touch(std::move(ts));
        } else if(ret<0) {
            dbg_err("Touch screen failed");
        }
    }
}
```

```
Listing 3. Strukturę touch_tag, która opisuje zdarzenie dotykowe
/** Touch screen tag */
struct touch_tag {
    static constexpr auto max_touch = 2;
    unsigned char num_touches; //! Number of touches
    unsigned short x[max_touch]; //! Position x
    unsigned short y[max_touch]; //! Position y
    unsigned char weight[max_touch]; //! Touch weight
    touchevents::touchevt eventid[max_touch]; //! Touch event id
    unsigned char area[max_touch]; //! Touch area
    touchgestures::touchgests gestureid; //! Gesture id
};
```

jest za rysowanie elementów na ekranie. Za obsługę zdarzeń oraz odczyt danych z kontrolera odpowiedzialna jest metoda `thread()` (listing 2).

Metoda ta rozpoczyna pracę od wywołania metody `intialize()` odpowiedzialnej za konfigurację układu ft6x06. W przypadku gdy inicjalizacja przebiegła pomyślnie wykonywana jest pętla nieskończona, która w cyklu 50 ms odczytuje z rejestrów konfiguracyjnych status kontrolera a jako wynik zwraca ilość zdarzeń jaka została wykryta. W przypadku wykrycia jakiegokolwiek zdarzenia wywoływana jest metoda `print_touch()` wypisująca na ekranie aktualnie wykryte zdarzeniem oraz metoda `report_touch()`. Zadaniem tej metody jest przekazanie zdarzeń dotykowych do klasy frame zarządzającej oknami na wyświetlaczu. Wspomniane wyżej funkcje jako jedyny argument przyjmują strukturę `touch_tag`, która opisuje zdarzenie dotykowe i pozwala później poszczególnym widżetom podjąć reakcję na poszczególne zdarzenia (listing 3). Do najważniejszych elementów tej struktury należą: pozycja bezwzględna wskazującą miejsce, gdzie ekran został dotknięty, szerokość obszaru dotknięcia oraz identyfikator gestu realizowanego za pomocą wielodotyku np. podniesienie okna, rozciągnięcie okna, zbliżenie oddalenie itp.

Wróćmy teraz na chwilę do funkcji inicjalizującej sterownik panelu dotykowego (listing 4). W metodzie tej w pierwszej kolejności

```
Listing 4. Funkcji inicjalizująca sterownik panelu dotykowego
//Initialize touchpad
int ft6x06_touch::initialize() noexcept
{
    int ret {};
    do {
        ret = detect_device();
        if(ret) break;
        const periph::dt::device_conf_base* base {};
        ret = periph::dt::get_periph_devconf(m_name,base);
        if(ret) break;
        const auto entry = reinterpret_
        cast<const periph::display::fb_info*>(base);
        if(entry->n_layers<1) {
            ret = periph::error::inval;
            break;
        }
        ret = touch_enable(entry->layers[0].width, entry-
        >layers[0].height);
        if(ret) break;
    } while(0);
    return ret;
}
```

```
Listing 5. Metodzie touch_enable(), której zadaniem jest
odpowiednie skonfigurowanie układu
//Touch screen enable
int ft6x06_touch::touch_enable(
    unsigned short sizex,
    unsigned short sizey) noexcept
{
    if(sizex<sizey) {
        m_orientation = TS_SWAP_NONE;
    } else {
        m_orientation = TS_SWAP_XY | TS_SWAP_Y;
    }
    //Start the device
    int ret {};
    do {
        ret = calibrate();
        if(ret) break;
        ret = disable_it();
        if(ret) break;
    } while(0);
    return ret;
}
```

wywoływana jest metoda `detect_device()`, której zadaniem jest stwierdzenie obecności dołączenia kontrolera do magistrali I²C. Działanie tej metody sprowadza się do odczytania zawartości rejestru identyfikacyjnego i porównaniu go z danym wzorcem. Jeśli wartość odczytana będzie różna od wartości wzorcowej lub nie uda się odczytać zawartości rejestru, zgłoszony będzie błąd, w wyniku czego wątek sterownika zakończy działanie. W przypadku, gdy

stwierdzimy iż kontroler został prawidłowo podłączony do magistrali, z bazy DTS odczytywana będzie aktualna rozdzielczość ekranu dotykowego. Rozdzielczość ta potrzebna jest metodzie `touch_enable()`, której zadaniem jest odpowiednie skonfigurowanie układu (listing 5).

W pierwszej kolejności sprawdzamy w jakiej orientacji pracuje wyświetlacz i w przypadku, gdy stwierdzimy, że znajduje się on w orientacji poziomej, ustawiane są dodatkowe bity konfiguracyjne zamieniające oś X z osią Y, które w późniejszym etapie będą przesłane do odpowiedniego rejestru. W kolejnym kroku wywoływana jest metoda `calibrate()`, której zadaniem jest kalibracja panelu dotykowego, tak aby dostosować go aktualnych warunków środowiskowych, a następnie wywoływana jest metoda `disable_int()`, której zadaniem jest wyłączenie zgłaszania przerw na linii INT kontrolera, ponieważ w aktualnej implementacji sterownik pracuje w trybie odpytowania bez systemu przerw.

Najważniejszą metodą wywoływaną w pętli głównej w cyklu 50 ms jest metoda `get_state()`, której zadaniem jest sprawdzenie czy wystąpiło zdarzenie oraz odczytanie informacji o typach zdarzeń dotykowych z rejestrów układu a następnie odpowiednie wypełnienie struktury przechowującej informację o zdarzeniu (listing 6). W pierwszej kolejności wywoływana jest metoda `get_touch()`, która z rejestrów kontrolera odczytuje ilość zdarzeń jaka została wykryta. W przypadku gdy liczba zdarzeń jest większa od 0, wówczas w pętli odczytywana jest pozycja (x,y) miejsca dotknięcia, jednak z uwagi iż wyświetlacz może być ustawiony w pozycji pionowej lub pionowej, odczytane z rejestrów układu współrzędne należy przeliczyć tak, aby odpowiadały one rzeczywistym współrzędnym ekranowym. Po przeliczeniu współrzędnych wywoływana jest metoda `get_info()`, której zadaniem jest odczytanie dodatkowych informacji o zdarzeniu takich jak m.in. typ gestu czy obszar w jakim nastąpiło zdarzenie. W kolejnym kroku w pętli switch..case następuje przeliczenie wartości rejestru zawierającego kod zdarzenia na identyfikatory zdarzeń występujące w bibliotece graficznej `libgfx`.

Mamy już gotowy sterownik umożliwiający zgłaszanie zdarzeń. Pozostało nam jeszcze utworzenie instancji klasy jako składowej prywatnej w klasie `tft_livedemo` przedstawionej w poprzednim odcinku, co zrealizowano bezpośrednio pliku `tft_livedemo.hpp`

```
gfx::drv::ft6x06_touch m_touch { „display”, m_i2c, frame };
```

Jako pierwszy argument przekazujemy do konstruktora nazwę stanowiącą identyfikator urządzenia wyświetlacza, potrzebny do odczytania danych z konfiguracji DTS, referencję do sterownika magistrali I²C oraz referencję do klasy `frame` biblioteki graficznej, która odpowiedzialna jest za zarządzanie oknem głównym biblioteki graficznej.

Aby uruchomić powyższy przykład należy podłączyć do komputera zestaw stm32f469i-disco z wykorzystaniem kabla mini USB oraz pobrać kod źródłowy z repozytorium wpisując polecenie: `git clone -b pub/ep1119 --recursive https://www.boff.pl/cgit/public/isixsamples`

W kolejnym kroku należy skompilować program oraz zaprogramować płytkę ewaluacyjną tak powstałym kodem maszynowym, co możemy zrealizować wydając następujące polecenia:

```
waf configure --crystal-hz=8000000 --debug waf
waf
waf program
```

Listing 6. Metoda `get_state()`, której zadaniem jest sprawdzenie czy wystąpiło zdarzenie oraz odczytanie informacji o typach zdarzeniach

```
// Get gesture state
int ft6x06_touch::get_state(touch_stat& stat) noexcept {
    using namespace detail::regs::ft6x06;
    using namespace gfx::input;
    int ts_status;
    do {
        /* Check and update the number of touches active detected */
        ts_status = detect_touch();
        if (ts_status<0) break;
        stat.num_touches = ts_status;
        ts_status = 0;
        if (stat.num_touches>0) {
            uint16_t Raw_x[c_max_nb_touch];
            uint16_t Raw_y[c_max_nb_touch];
            for (int index = 0; index < stat.num_touches; index++){
                /* Get each touch coordinates */
                ts_status = get_xy( (Raw_x[index]), (Raw_y[index]));
                if(ts_status) break;
                if (m_orientation & TS_SWAP_XY) {
                    auto tmp = Raw_x[index];
                    Raw_x[index] = Raw_y[index];
                    Raw_y[index] = tmp;
                }

                if (m_orientation & TS_SWAP_X) {
                    Raw_x[index] = FT_6206_MAX_WIDTH - 1 - Raw_x[index];
                }

                if (m_orientation & TS_SWAP_Y) {
                    Raw_y[index] = FT_6206_MAX_HEIGHT - 1 - Raw_y[index];
                }

                auto xDiff = Raw_x[index] > m_x[index] ?
                    (Raw_x[index] - m_x[index]) : (m_x[index] - Raw_x[index]);

                auto yDiff = Raw_y[index] > m_y[index] ?
                    (Raw_y[index] - m_y[index]) : (m_y[index] - Raw_y[index]);

                if ((xDiff + yDiff) > 5) {
                    m_x[index] = Raw_x[index];
                    m_y[index] = Raw_y[index];
                }

                stat.x[index] = m_x[index];
                stat.y[index] = m_y[index];
                uint32_t weight {};
                uint32_t area {};
                uint32_t event {};
                ts_status = get_info(index, weight, area, event);
                if(ts_status) break;
                stat.weight[index] = weight;
                stat.area[index] = area;

                /* Remap touch event */
                switch (event) {
                    case FT6206_TOUCH_EVT_FLAG_PRESS_DOWN:
                        stat.eventid[index] = touchevents::press_down;
                        break;
                    case FT6206_TOUCH_EVT_FLAG_LIFT_UP:
                        stat.eventid[index] = touchevents::lift_up;
                        break;
                    case FT6206_TOUCH_EVT_FLAG_CONTACT:
                        stat.eventid[index] = touchevents::contact;
                        break;
                    case FT6206_TOUCH_EVT_FLAG_NO_EVENT:
                        stat.eventid[index] = touchevents::undefined;
                        break;
                    default:
                        ts_status = error::error_not_supported;
                        break;
                }
            }
        }

        /* Get gesture Id */
        ts_status = get_gesture_code();
        if(ts_status<0) break;
    } while (0);
    return ts_status?ts_status:stat.num_touches;
}
```

Równocześnie do wirtualnego portu szeregowego utworzonego przez programator należy podłączyć dowolny program terminalowy oraz ustawiając następujące parametry transmisji: 115200,n,8,1.

Po wykonaniu powyższych czynności na wyświetlaczu powinno pojawić się okno z demonstracyjnymi widżetami biblioteki graficznej. Dotykając teraz poszczególnych elementów na ekranie na konsoli szeregowej będziemy mogli zaobserwować informację o pozycji dotknięcia, obszarze dotknięcia oraz ewentualnych gestach. Jednak nadal interfejs dotykowy w żaden sposób nie będzie reagował na zdarzenia. O tym jak zrobić, aby wspomniane elementy reagowały na dotyk, dowiemy się z kolejnego odcinka, gdy zostanie przedstawione więcej szczegółów na temat biblioteki `libgfx`.

Lucjan Bryndza, EP
lucjan.bryndza@boff.pl

RISC-V – budujemy własny mikrokontroler (3)

Otwarte oprogramowanie jest spotykane w wielu dziedzinach od kilkudziesięciu lat. W wielu przypadkach wypierało swoich płatnych konkurentów. Zadaniem, jakie przed sobą stawiają członkowie projektu RISC-V, jest przeprowadzenie podobnej rewolucji w dziedzinie procesorów.

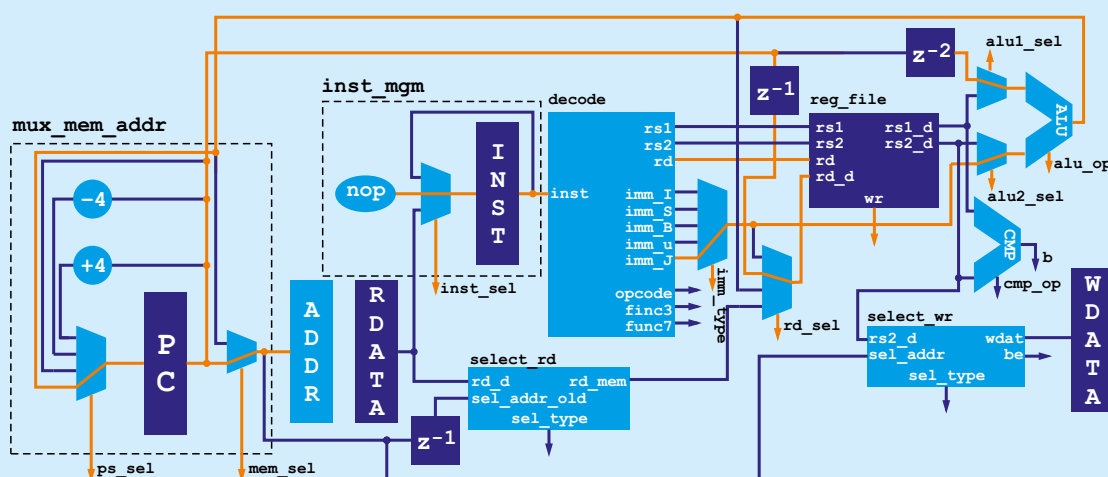


Skoki bezwarunkowe

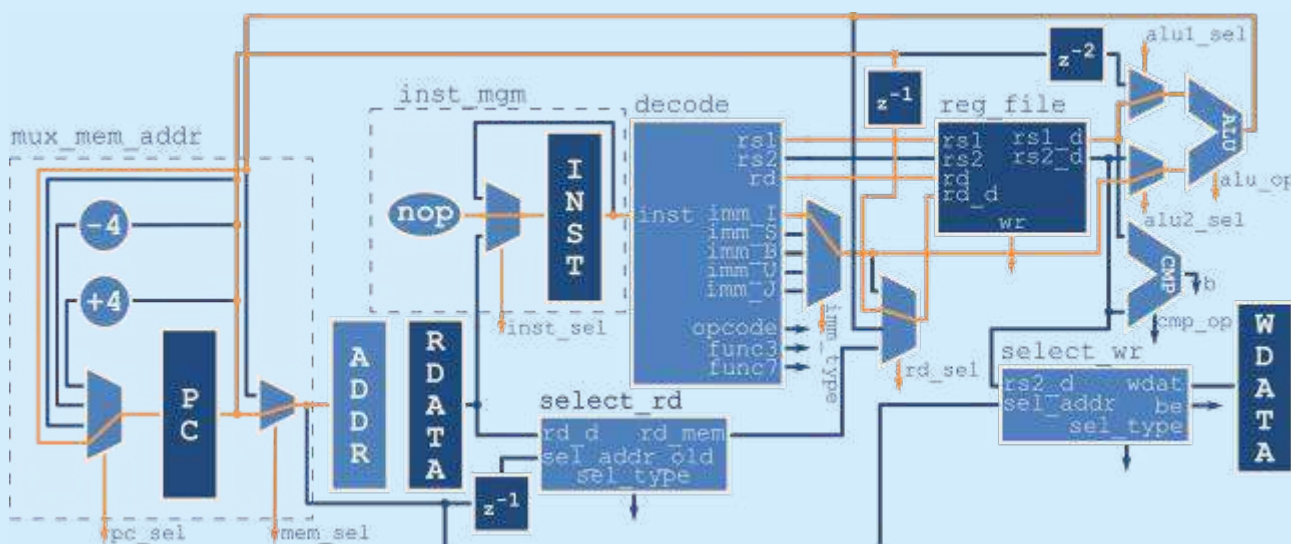
Przyjrzyjmy się teraz trochę bardziej skomplikowanym poleceniom: skokom bezwarunkowym. Pierwszy z nich **jal** jest pokazany na **rysunku 29**. Jak już wiemy pozwala on na dodanie do rejestru PC stałej przy równoczesnym zapisaniu adresu następnej instrukcji do rejestru. Multiplexer *imm_type* wybiera stałą typu J, a *alu2_sel* wpuszcza ją do ALU. Pierwszym operandem wybranym przez *alu1_sel* jest „płynąca wraz z rozkazem” wartość rejestru PC. Ścieżka *alu_op* wybiera dodawanie. *pc_sel* podaje wynik do rejestru PC. Ponieważ

rozkaz z pod nowego adresu wejdzie do fazy wykonania dopiero za dwa cykle część kontrolna będzie musiała zastąpić dwa kolejne rozkazy instrukcjami *nop*, co obrazuje konfiguracja rejestru *inst_sel*. Natomiast do rejestru *rd* zostanie wpisany adres następnego rozkazu, który byłby wykonany gdyby skok nie nastąpił. Jest to wartość PC opóźniona o 1 takt zegara.

Drugim rodzajem skoku bezwarunkowego jest instrukcja **jalr**. Jak widzimy na **rysunku 30** jest ona bardzo podobna. Jednak tym razem kolejny rozkaz jest pobrany z adresu będącego sumą wartości



Rysunek 29. Przepływ danych dla instrukcji *jal*



Rysunek 30. Przepływ danych dla instrukcji *jalr*

z rejestru i (krótszej) stałej. Ponieważ najważniejsze elementy obsługi obu rozkazów są analogiczne przyjrzymy się dokładnie tylko pierwszej z nich: *jal*. Jako przykład posłuży nam króciutki kod z **listingu 2**. Dla przypomnienia został on przedstawiony w **tabeli 5**, ale tym razem, także w postaci binarnej i z podaniem adresów w pamięci. Pierwsza instrukcja zeruje rejestr x5, a następnie w nieskończonej pętli następuje jego inkrementacja.

Potok instrukcji dla początku i dwóch pierwszych obiegów pętli pokazuje **rysunek 31**. Jak już wiemy zaraz po resetie przez dwa takty zegara część sterująca wymusza dwie instrukcje *nop*. W trzecim cyklu instrukcja spod adresu 0x00 jest wykonywana, czyli następuje wyzerowanie rejestru x5. W kolejnym takcie wykonana jest pierwsza instrukcja z ciała pętli. Widzimy jednak, że w tym kroku ustawiony już jest adres 0x0c. Kolor czerwony symbolizuje, że ten fragment pamięci nie zawiera żadnych sensownych danych. Na tym etapie mikrokontroler jeszcze nie wie, że za chwilę nastąpi skok, a ładowany adres jest błędny. Dopiero w następnym cyklu jest wykonana instrukcja skoku. W tym momencie w potoku znajdują się już dwie nieprawidłowe instrukcje. Tu wchodzi część kontrolna, która dokładnie tak samo jak w przypadku resetu wymusza zastąpienie dwóch kolejnych rozkazów instrukcją *nop*. Widzimy, że przez ten czas rdzeń wyczyści potok z nieprawidłowych rozkazów, a docelowa instrukcja spod adresu 0x04 dojdzie do fazy wykonaj. Widzimy więc, że w przeciwieństwie do operacji na stałych i rejestrach skok zajmuje aż trzy takty zegara. Jest to ceną jaką płacimy za potokowe przetwarzanie instrukcji. Jak łatwo możemy policzyć jeden obieg pętli zajmuje cztery cykle zegara.

Zachęcam do samodzielnego uruchomienia symulacji rdzenia z przygotowanym programem. Tak jak wcześniej uruchamiamy ModelSima, przechodzimy do głównego folderu repozytorium i uruchamiamy:

do scripts/simulate_jal.do

Powinniśmy zobaczyć przebiegi podobne do tych z **rysunku**

Tabela 5. Program demonstrujący skok bezwarunkowy (code/jal.S)

Adres	Binarny	Po assemblerze	Kod
0x00	0x00000293	<code>addi x5, x0, 0</code>	<code>addi x5, x0, 0</code>
			<code>loop:</code>
0x04	0x00128293	<code>addi x5, x5, 1</code>	<code>addi x5, x5, 1</code>
0x08	0xffdff0ef	<code>jal x1, -4</code>	<code>jal x1, loop</code>

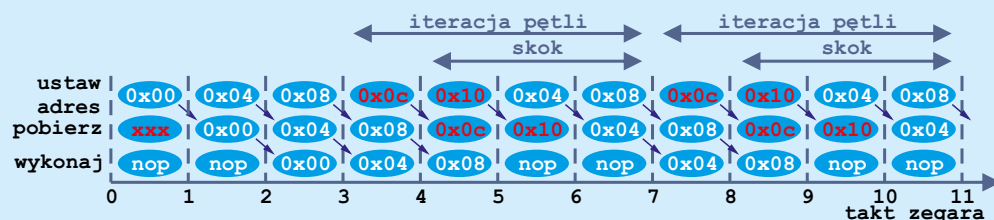
32. Jeżeli interesują nas przebiegi innych sygnałów możemy je przeciągnąć z panelu *Objects* (ang. obiekty) i umieścić w panelu *Wave* (ang. fale). Następnie musimy wrócić na początek symulacji wpisując polecenie:

`restart -f`

A następnie uruchomić symulację (w tym przypadku na 13 μ s):

`run 13us`

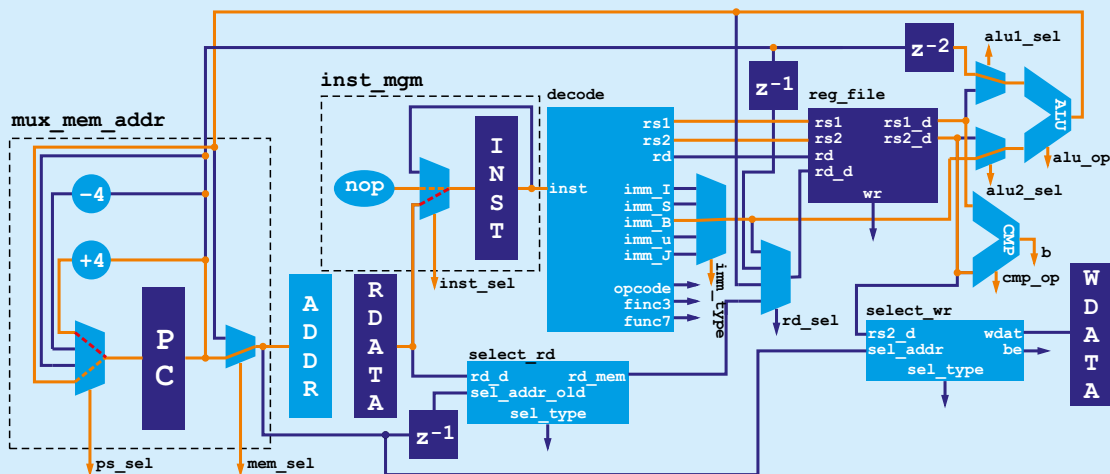
Podobnie jak poprzednio w pierwszych dwóch wierszach mamy sygnał zegarowy i reset. Gdy reset opadnie rozpoczyna się praca rdzenia. Aż do momentu, gdy zostanie wykonana pierwsza instrukcja wartość rejestru x5 (ostatni wiersz) jest nieustalona. Symbolizuje to czerwona linia. Dopiero pierwszy rozkaz powoduje jego wyzerowanie. W kolejnym takcie następuje pierwsza inkrementacja x5. W kolejnym cyklu rozpoczyna się wykonanie skoku. Widzimy, że do rejestru x1 wpisany został adres powrotu, czyli 12 (szesnastkowo 0x0c). Warto też zwrócić uwagę, że odczyt niezainicjalizowanej pamięci powoduje pojawienie się w symulacji czerwonej kreski,



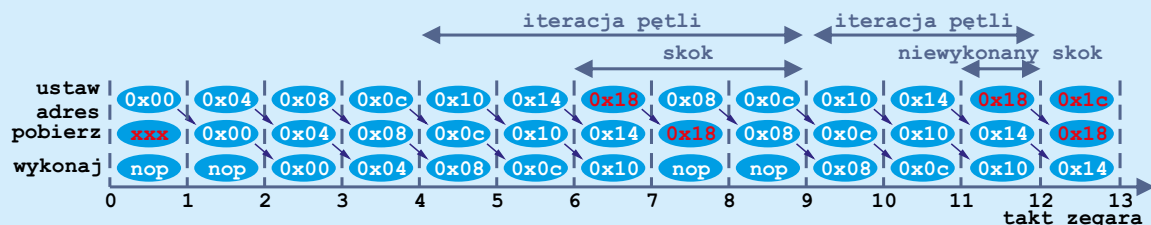
Rysunek 31. Potok instrukcji przy wykonywaniu programu z tabeli 5. Numery określają spod jakiego adresu pochodzi rozkaz



Rysunek 32. Symulacja wykonania kodu z tabeli 5



Rysunek 33. Przebieg danych dla instrukcji warunkowych. Ścieżki zależne od warunku są zaznaczone liniami przerywanymi: pomarańczowe dla wykonanego, a czerwone dla niewykonanego skoku.



Rysunek 34. Potok instrukcji przy wykonywaniu programu z tabeli 6. Numery określają spod jakiego adresu pochodzi rozkaz

czyli nieznanego stanu. Nie zostanie on jednak wykonany, ponieważ przez kolejne dwa cykle będzie ona nadpisana instrukcją o kodzie 0x00000013, czyli *nop*. Licząc liczbę cykli przypadających na kolejne wartości w rejestrze x5 widzimy, że zgodnie z oczekiwaniem obrót pętli trwa cztery cykle zegara.

Skoki warunkowe

Skoki bezwarunkowe pozwalają na wywoływanie i powroty z podprogramów. Jeżeli jednak chcemy, aby nasz program podejmował decyzję potrzebujemy skoku warunkowego, czyli takiego, które zachodzi jedynie, gdy spełniony jest określony warunek. Przepływ danych przedstawia **rysunek 33**. Z rozkazu odczytywane są numery rejestrów, których wartości są przekazywane do CMP. Za pomocą ścieżki *cmp_op* wybierany jest rodzaj warunku do sprawdzenia. Jeżeli został on spełniony ścieżka b przybiera stan wysoki. Równocześnie ALU sumuje wartość rejestru PC (opóźnioną o dwa cykle) ze stałą bezpośrednią. Na podstawie stanu ścieżki b część kontrolna wybiera, czy kolejna wartość PC zostanie załadowana z ALU (skok), czy przyjmie wartość o 4 większą od poprzedniej (brak skoku). Jeżeli skok nastąpi, to analogicznie jak dla instrukcji *jal* konieczne będzie wyczyszczenie potoku poprzez wymuszenie instrukcji *nop* przez dwa cykle zegara. W **tabeli 6** znajduje się znany nam już kod z **listingu 4**, ale tym razem z podaną lokalizacją poszczególnych rozkazów w pamięci. Na **rysunku 34** pokazany jest potok instrukcji wykonywanych podczas obu iteracji wewnętrznej pętli.

Tak jak we wcześniejszych przykładach efekt pracy pierwszego rozkazu zostanie zapisany na 3 zboczu zegara. Pierwszy skok warunkowy ma miejsce podczas taktów od siódmego do dziewiątego. Ponieważ został on wykonany, konieczne jest wyczyszczenie potoku. Następnie w taktie 12 następuje drugie wywołanie instrukcji. Tym razem warunek nie jest spełniony. Ponieważ zwykły potok nie został zaburzony, wykonanie rozkazu zajęło tylko jeden takt zegara. W następnym kroku następuje przejście do rozkazu z adresu 0x14, czyli znanego już nam skoku bezwarunkowego do początku programu.

Tabela 6. Program demonstrujący skok bezwarunkowy (code/branch.S)

Adres	Binarny	Po assemblerze	Kod
			start:
0x00	0x00200093	<code>addi x1, x0, 2</code>	<code>addi x1, x0, 2</code>
0x04	0x00000113	<code>addi x2, x0, 0</code>	<code>addi x2, x0, 0</code>
			loop:
0x08	0x00110133	<code>add x2, x2, x1</code>	<code>add x2, x2, x1</code>
0x0c	0xffff08093	<code>addi x1, x1, -1</code>	<code>addi x1, x1, -1</code>
0x10	0xfe101ce3	<code>bne x0, x1, -8</code>	<code>bne x0, x1, loop</code>
0x14	0xfedff06f	<code>jal x0, -20</code>	<code>j start</code>

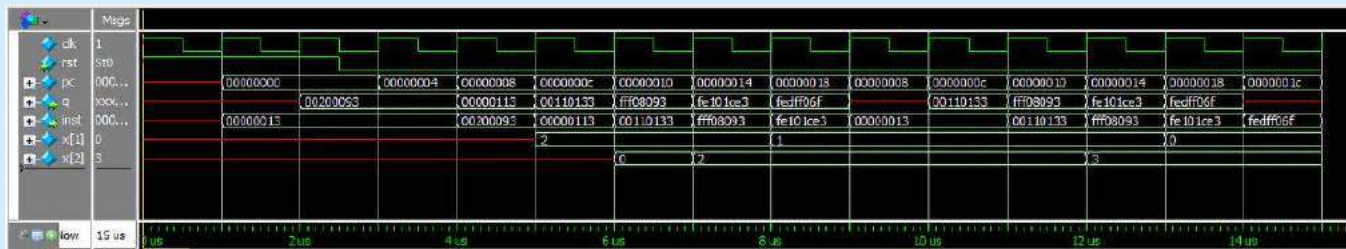
Jego działanie poznaliśmy już w poprzednim kroku. Warto zwrócić uwagę, że ostatni obieg pętli, w którym skok nie zostaje wykonany trwa o dwa takty zegara krócej niż wcześniejsze. Jeżeli chcielibyśmy użyć pętli do wygenerowania opóźnienia powinniśmy uwzględnić ten fakt. Tak jak poprzednio zachęcam do zasymulowania działania tego programu w ModelSimie:

do scripts/simulate_b.do

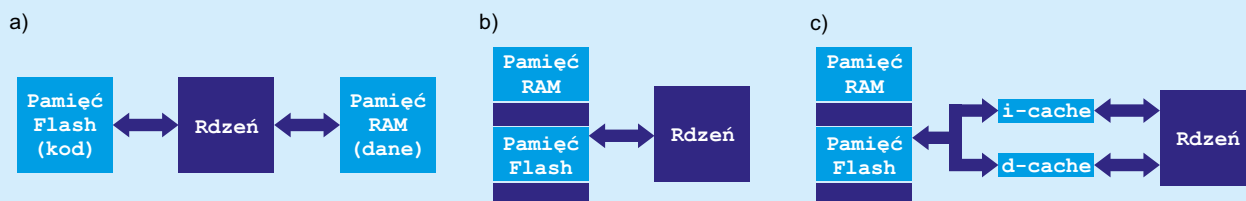
Fragment uzyskanych przebiegów pokazuje **rysunek 35**. Warto go porównać z rysunkiem 34. Ponieważ rdzeń rozpoczyna pracę po wyzerowaniu sygnału reset zerowy cykl zegara ma miejsce w 2 µs symulacji.

Zapis do pamięci

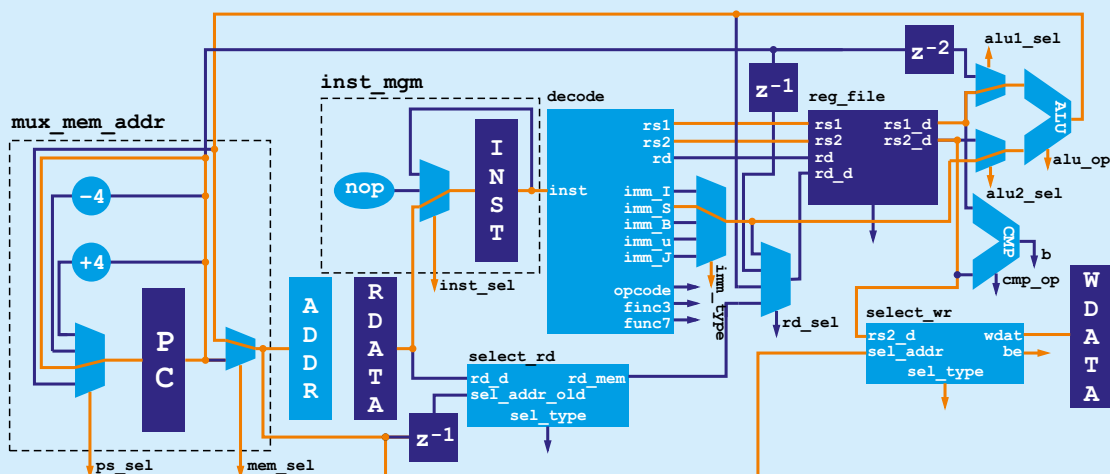
Żeby zaprojektować obsługę zapisu i odczytu z pamięci musimy wrócić do architektury naszego rdzenia. Dla przypomnienia różne jej rodzaje są pokazane na **rysunku 36**. Część a przedstawia architekturę Harwardzką występującą na przykład w popularnych mikrokontrolerach AVR. Pamięci danych jest w niej niezależna od pamięci programu. Dzięki temu odczyt i zapis do RAM nie wpływają na wczytywanie kolejnych instrukcji. Ma ona też swoje wady. Odczytywanie



Rysunek 35. Symulacja wykonania kodu z tabeli 6



Rysunek 36. a) Architektura Harwardzka z rozdzieloną przestrzenią adresową programu i danych (AVR). b) Architektura von Neumana (ARM Cortex-M0). c) Architektura Harwardzka z cachem (ARM Cortex-M7)



Rysunek 37. Przepływ danych podczas zapisu do pamięci

stałych z pamięci Flash jest dużo trudniejsze, a uruchomienie programu z pamięci RAM nie jest możliwe.

Na rysunku 36b pokazana jest architektura von Neumana. Tutaj zarówno program jak i dane znajdują się we wspólnej przestrzeni adresowej, pomimo że mogą być one wykonane w różnej technologii. Dostęp do nich odbywa się przez tę samą szynę. Przykładem mikrokontrolera z taką organizacją jest ARM Cortex-M0.

Dla szybszych rdzeni stosuje się jeszcze inne podejście. Program i dane nadal znajdują się we wspólnej przestrzeni adresowej, ale dostęp do nich odbywa się poprzez *cache* (pamięć notatnikowa). Tę sytuację pokazuje rysunek 36c. Jest to stosunkowo mała, ale szybka pamięć, która pośredniczy w dostępie do przestrzeni adresowej. Rozwiązuje konflikt pomiędzy wykorzystaniem magistrali dla dostępu do danych i do pobierania programu. Przykładem mikrokontrolera, w którym zastosowano taką architekturę jest ARM Cortex-M7. Występuje także w większości procesorów (Intel, ARM Cortex-A). Cache same w sobie są ciekawym (i dość złożonym) zagadnieniem. Zainteresowanym czytelnikom polecam wykłady [5], [14] oraz książkę [12].

W budowanym przez nas rdzeniu zastosowana jest architektura von Neumana (rysunek 36b).

Rysunek 37 pokazuje przepływ danych przy zapisie do pamięci. Adres pamięci jest ustawiony na wartość z wyjścia ALU, czyli sumę stałej bezpośredniej oraz wartości z rejestru. Wartość rejestru PC nie ulega zmianie, ponieważ na następnym zlocu zegara nie nastąpi odczyt kolejnej instrukcji.

Do pamięci zostanie zapisana wartość z drugiego odczytanego rejestru. Przed zapisem musi jednak zostać odpowiednio przygotowana, za co odpowiada blok *select_wr*. Ścieżka *sel_type* konfiguruje, czy zapis ma dotyczyć pojedynczego bajtu, połowy, czy całego słowa. Tu pojawia się problem. Pamięć obsługuje jedynie dostęp do pełnych słów, spod adresów wyrównanych do czterech bajtów. Na szczęście udostępnia jednak dodatkowy czterobitowy sygnał *be* (*byte enable* – bajt włączony). Zapisane zostaną tylko te bajty słowa, dla których odpowiedni bit jest ustawiony.

Pierwszym zadaniem bloku *select_wr* jest odpowiednie przesunięcie danych do zapisania, aby odpowiadały docelowej pozycji w pamięci. Odpowiedzialny za to fragment kodu pokazuje **listing 12**. Przy zapisie pojedynczego bajtu (SB) jak łatwo się domyślić dostępne są cztery możliwości. Przy zapisie połowy słowa (SH) sprawa jest trudniejsza. Teoretycznie także mamy cztery możliwości, ale tylko dwie z nich w całości mieszczą się w pojedynczym słowie w pamięci. Dzieje się tak gdy adres jest podzielny przez liczbę bajtów. O takim adresie mówimy, że jest wyrównany (*aligned*). Zapis (a także odczyt) pod niewyrównany adres wymaga rozbicia go na dwie operacje w pamięci. RISC-V ISA zostawia twórcą poszczególnych implementacji

Listing 12. Fragment modułu *select_wr* (*core/select_wr.sv*) przesuwający dane, aby pasowały do odpowiedniego słowa pamięci

```
20 always_comb
21   case (sel_type)
22     selectPkg::SB:
23       case (sel_addr)
24         2'b00: wdata = {24'b0, rs2_d[7:0]};
25         2'b01: wdata = {16'b0, rs2_d[7:0], 8'b0};
26         2'b10: wdata = {8'b0, rs2_d[7:0], 16'b0};
27         2'b11: wdata = {rs2_d[7:0], 24'b0};
28         default: wdata = {24'b0, rs2_d[7:0]};
29       endcase
30     selectPkg::SH:
31       case (sel_addr)
32         2'b00: wdata = {16'b0, rs2_d[15:0]};
33         2'b10: wdata = {rs2_d[15:0], 16'b0};
34         default: wdata = {16'b0, rs2_d[15:0]};
35       endcase
36     selectPkg::SW: wdata = rs2_d;
37     default: wdata = rs2_d;
38   endcase
```

Listing 13. Fragment modułu *select_wr* (*core/select_wr.sv*) przygotowujący sygnał *be*

```
40 always_comb
41   case (sel_type)
42     selectPkg::SB: be = 4'b0001 << sel_addr;
43     selectPkg::SH: be = 4'b0011 << sel_addr;
44     selectPkg::SW: be = 4'b1111;
45     default: be = 4'b1111;
46   endcase
47 endmodule
```

wyбір, czy taki dostęp do pamięci będzie obsługiwany sprzętowo, programowo, bądź spowoduje zgłoszenie wyjątku. Ponieważ w naszej minimalistycznej implementacji nie są one wywoływane przyjmujemy, że wyniki niewyrównanych operacji na pamięci jest niezdefiniowany. Ostatni przypadek to zapis całego słowa (SW). W tym przypadku wyrównany adresu musi być podzielny przez cztery, więc mamy tylko jedną możliwość: po prostu przesyłamy niezmienną wartość z rejestru do pamięci.

Drugim zadaniem *select_wr* jest przygotowanie sygnału *be*. Odpowiadającą za to logikę znajdziemy na **listingu 13**. Dla pojedynczego bajtu ustawiona jest jedna 1 na polu odpowiadającym dwóm najmłodszym bitom adresu. Dla połowy słowa ustawione są dwa młodsze albo dwa starsze bity. Przy zapisie całego słowa ustawione są wszystkie cztery bity sygnału *be*.

Tabela 7. Program demonstrujący zapis danych do pamięci (*code/store_sim.S*)

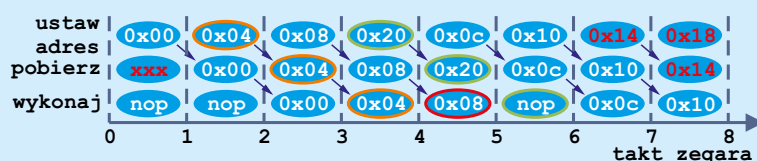
Adres	Binarny	Po assemblerze	Kod
0x00	0x00a00093	<i>addi x1 x0 10</i>	<i>li x1, 10</i>
0x04	0x02102023	<i>sw x1 32(x0)</i>	<i>sw x1, 0x20, x0</i>
0x08	0x00100093	<i>addi x1 x0 1</i>	<i>li x1, 1</i>
0x0c	0x00200093	<i>addi x1 x0 2</i>	<i>li x1, 2</i>
0x10	0x00300093	<i>addi x1 x0 3</i>	<i>li x1, 3</i>

Wiemy już w jaki sposób przygotowane są dane do zapisu. Musimy obsłużyć jeszcze jeden problem. Ponieważ pamięć danych i programu mają wspólną magistralę, wartość spod zapisywanego adresu zostanie wciągnięta do potoku instrukcji. Musimy więc wykryć w którym miejscu się pojawi i zastąpić ją bezpiecznym rozkazem *nop*. Aby poczuć problem przeanalizujemy wykonanie kodu z tabeli 7. Pierwsza instrukcja ładuje do rejestru liczbę dziesięć, która zostanie załadowana pod adres 0x20. Pozostałe trzy wpisują do rejestru x1 kolejno liczby 1, 2 i 3. Są one dodane po to, aby łatwiej było zaobserwować, w który momencie w potoku znajduje się „nieproszona” wartość. Na rysunku 38 pokazano przepływ instrukcji. Kolorem pomarańczowym zaznaczona jest instrukcja *sw*. Gdy wchodzi do fazy wykonania adres jest ustawiony na 0x20. Przepływ tej instrukcji przez potok zaznaczono kolorem zielonym. Warto zwrócić uwagę na instrukcję 0x08. W momencie gdy dokonywany jest zapis znajduje się już ona w fazie pobierz. Dzięki temu może być ona wykonana od razu po zapisie, co zaznaczono kolorem czerwonym. Jest tu jednak mała pułapka. Gdyby zapis odbywał się pod adres 0x08 wykonana byłaby instrukcja znajdująca się pod tym adresem przed dokonaniem zapisu, mimo że w kodzie znajduje się ona za nim! W ten sposób zyskujemy skrócenie trwania zapisu z 3 do 2 cykli zegara kosztem dodania bardzo nieoczywistego zachowania w pewnym szczególnym przypadku. Ponieważ podobne sztuczki są także implementowane w większości procesorów dodano specjalną rodzinę instrukcji zwanych barierami, które gwarantują kolejności zapisu i odczytu danych. Dla naszego prostego rdzenia taką barierą będzie po prostu dodanie instrukcji *nop* po zapisie, który mógłby sprawić problem.

Abu uruchomić zaprezentowany program w symulatorze Model-Sim wywołujemy polecenie:

```
do scripts/simulate_store.do
```

Po jego wykonaniu zobaczymy efekt podobny do tego z rysunku 39. W ostatnim wierszu wyświetlona jest zawartość ósmego słowa



Rysunek 38. Przepływ instrukcji podczas wykonywania programu z tabeli 7

Tabela 8. Program demonstrujący odczyt danych z pamięci (code/load_sim.S)

Adres	Binarny	Po assemblerze	Kod
0x00	0x00a00093	<i>addi x1 x0 10</i>	<i>li x1, 10</i>
0x04	0x02102023	<i>lw x1 20(x0)</i>	<i>lw x1, 0x14, x0</i>
0x08	0x00100093	<i>addi x1 x0 1</i>	<i>li x1, 1</i>
0x0c	0x00200093	<i>addi x1 x0 2</i>	<i>li x1, 2</i>
0x10	0x00300093	<i>addi x1 x0 3</i>	<i>li x1, 3</i>
0x14	0x000000ff		

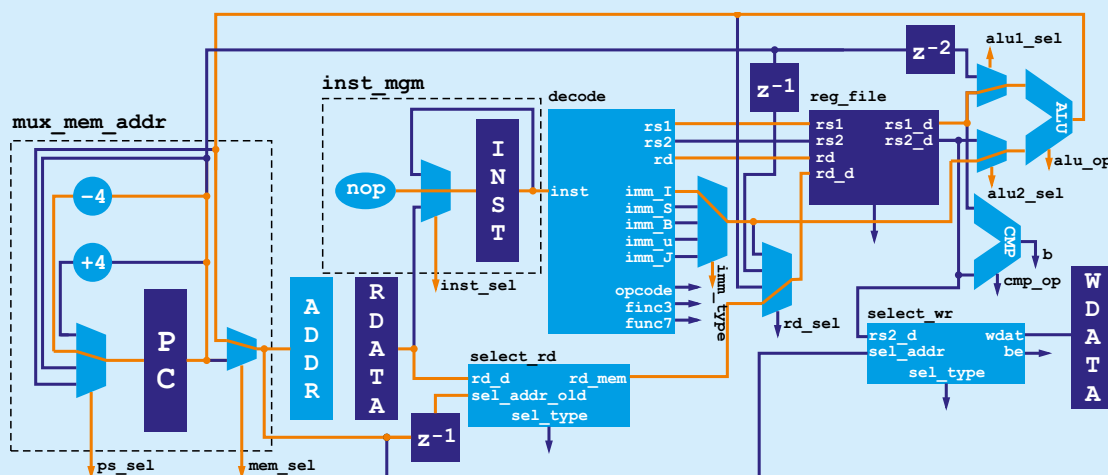
pamięci RAM, które zawiera bajty od 32 do 36. Widzimy że w 6 μ s następuje zapis do niego zawartości rejestru x1. Następnie w kolejnym takcie do x1 wpisywana jest jedynka. Na zboczach zegara w 8 μ s wykonana jest wstrzyknięta instrukcja *nop*, która zastępuje dane wczytane w czasie zapisu. Ponieważ wcześniej to słowo pamięci nie było zapisane oznaczone jest czerwoną kreską, czyli jako niezdefiniowane. Wpisanie do x1 liczby 2 ma miejsce dopiero podczas kolejnego taktu (9 μ s).

Odczyt z pamięci

Odczyt danych z pamięci jest operacją tylko trochę bardziej skomplikowaną niż zapis. Jak widzimy na rysunku 40 wybór adresu przebiega analogicznie. Tym razem jednak z samym zapisem do rejestru musimy poczekać, aż dane pojawią się w rejestrze RDATA. Ma to miejsce dopiero na kolejnym zboczach zegara po tym jak instrukcja



Rysunek 39. Symulacja wykonania kodu z tabeli 7



Rysunek 40. Przepływ danych podczas odczytu z pamięci

znajdzie się w fazie wykonaj. Wtedy dopiero zostanie ustawiona ścieżka *wr*, a sam zapis do rejestru nastąpi dopiero na trzecim taktie zegara. Blok *select_rd* pełni odwrotną funkcję do *select_wr*. Na podstawie dwóch najmłodszych bitów adresu oraz rozmiaru wczytywanych danych wycina z odczytanego słowa pamięci potrzebne dane. Następnie przesuwane są one na początek rejestru. Aby adres „płynął” razem z danymi przed podaniem na blok *select_rd* jest on opóźniony o jeden takt zegara.

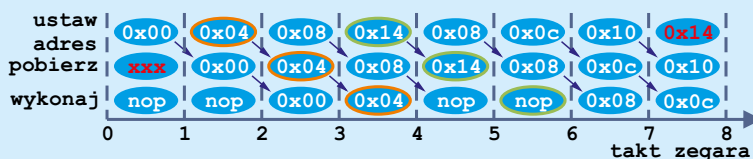
W przypadku odczytu całego słowa jest ono po prostu wpisywane do rejestru. Natomiast dla połowy słowa oraz bajtu dochodzi dodatkowa możliwość. Rozkazy *sh* i *sb* traktują pobieraną wartość jako liczbę ze znakiem, więc na niewykorzystane bity rejestru dopełniają najstarszym bitem zmiennej. Jednak jeżeli chcemy reprezentować tylko liczby dodatnie możemy skorzystać z instrukcji *shu* i *sbu*, które zawsze dopełnią je zerami. Różnicę

Listing 14. Różnica pomiędzy ładowaniem liczb z i bez znaku (*core/select_rd.sv*).

```

29 selectPkg::SH:
30   case (sel_addr_old)
31     2'b00: rd_mem = {rdata[15] ? 16'hffff : 16'h0, rdata[15:0]};
32     2'b10: rd_mem = {rdata[31] ? 16'hffff : 16'h0, rdata[31:16]};
33     default: rd_mem = 'b0;
34   endcase
44 selectPkg::SHU:
45   case (sel_addr_old)
46     2'b00: rd_mem = {16'h0, rdata[15:0]};
47     2'b10: rd_mem = {16'h0, rdata[31:16]};
48     default: rd_mem = 'b0;
49   endcase

```



Rysunek 41. Przepływ instrukcji podczas wykonywania programu z tabeli 8



Rysunek 42. Symulacja wykonania kodu z tabeli 8

między implementacją obu tych przypadków obrazuje fragment kodu z **listingu 14**.

Program demonstracyjny z **tabeli 8** jest bardzo podobny do poprzedniego. Różni się zamianą instrukcji *sw* na *lw* oraz wpisaniem liczby 255 do słowa pamięci pod adresem 0x14. **Rysunek 41** pokazuje potok instrukcji podczas wykonywania przygotowanego programu. Na pomarańczowo oznaczono instrukcję *lw*. W momencie jej wykonania ustawiany jest adres danych do wczytania, co zostało oznaczone kolorem zielonym. Aby kolejne instrukcje były wykonywane dopiero po wczytaniu danych do rejestru wartość PC jest przywracana do poprzedniej pozycji poprzez odjęcie 4. Taka realizacja odczytu z pamięci jest prostsza ale zajmuje 3 cykle. Alternatywnym rozwiązaniem mógłby być zapis instrukcji spod adresu 0x08 w dodatkowym rejestrze. Dzięki temu mogłaby być ona zładowana w miejsce drugiej instrukcji *nop* (oznaczonej kolorem zielonym). Spowodowałoby to skrócenie czasu wykonania o jeden takt zegara. Zachęcam do zabawy z własnymi modyfikacjami. Na **rysunku 42** pokazano wynik symulacji, którą można uruchomić poleceniem:

do scripts/simulate_load.do

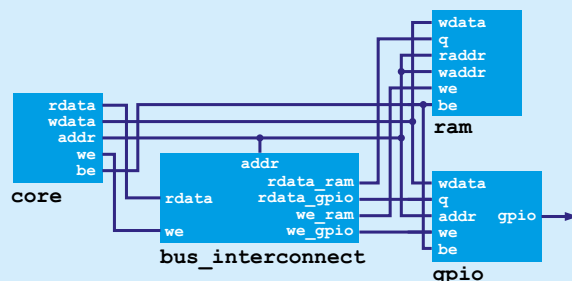
Za wygenerowanie odpowiednich sygnałów sterujących odpowiada moduł *ctrl* zaimplementowany w pliku *core/ctrl.sv*.

Peryferia

Mamy już gotowy rdzeń. Żeby mógł pracować potrzebujemy jeszcze peryferiów. Najprostszym sposobem na ich dodanie są tak zwane MMIO (*Memory-Mapped Input/Output* – wejścia/wyjścia zmapowane do pamięci). Technika ta polega na przypisaniu rejestrów poszczególnych podzespołów adresów. Dzięki czemu dostęp do nich nie wymaga żadnych dodatkowych instrukcji. Po prostu korzystamy z rozkazów z rodzin *store* i *load*.

Zajmiemy się tylko najprostszym możliwym peryferium, czyli wyjściem cyfrowym. Docelowo posłuży ono do sterowania diodami świecącymi.

Rysunek 43 pokazuje schemat naszego miniaturowego układu mikroprocesorowego. Implementacja jego poszczególnych elementów znajduje się w folderze *peripheral*. Kod bloku pamięci RAM



Rysunek 43. Schemat układu mikroprocesorowego

pochodzi z szablonu dostarczonego przez firmę Intel, który gwarantuje że w czasie syntezy nie zostanie on złożony z podstawowych elementów logicznych. Zamiast tego wyinferowana zostanie blokowa pamięć RAM. W projekcie przyjęto, że ma ona rozmiar 1 kB. W przestrzeni adresowej zajmuje pozycję od 0 do 0x400. Moduł wyjść cyfrowych składa się z jednego rejestru. Jego 4 najmłodsze bity są wyprowadzona na ścieżkę *gpio*. Zostaną one podłączone do czterech diod świecących znajdujących się na płytce prototypowej. Ponieważ do wyjść układu FPGA podłączone są ich katody, moduł *gpio* wystawia zanegowane wartości. Dzięki temu logiczne 1 odpowiada zaświeconej, a 0 zgaszonej diodzie. W przestrzeni adresowej odpowiada im adres 0x2000_0000. Moduł *bus_interconnect* pełni rolę multiplexera. Na podstawie najstarszych bitów adresu ustala czy aktywny ma być RAM, czy *gpio*. Odpowiada za to, aby z jednej strony rdzeń otrzymał odpowiednie dane wejściowe, a z drugiej aby sygnał *we* trafił do odpowiedniego bloku.

Listing 15. Program sterujący diodami LED (*code/blink_slow.S*).

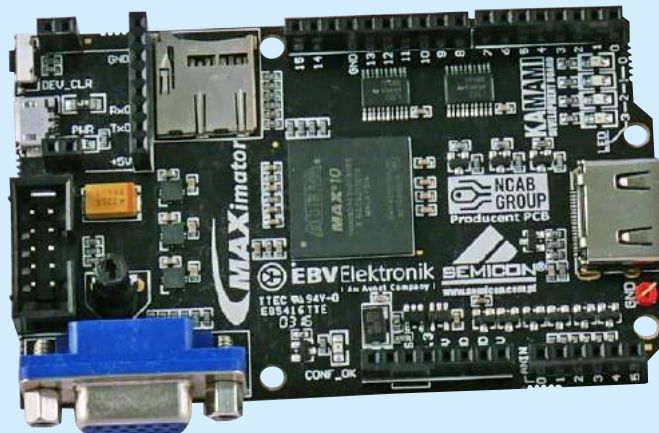
```

01 li    x1, 0x20000000
02 li    x2, 0
03 start:
04 li    x3, 1000000
05 loop:
06 addi  x3, x3, -1
07 bnez  x3, loop
08 addi  x2, x2, 1
09 sw    x2, 0, x1
10 j     start

```



Rysunek 44. Symulacja działania programu migającego diodą



Fotografia 45. Zestaw startowy maXimator

Na **listingu 15** pokazany jest program, który zaświeca diody LED w sekwencji odpowiadającej kolejnym liczbom binarnym. W pierwszej linii do rejestru x1 wpisywany jest wybrany wcześniej adres portów wyjściowych. Rejestr x2 służy do przygotowania kolejnych stanów wyjść. Główna pętla programu zaczyna się od etykiety *start*. W liniach od 4 do 7 znajduje się pętla opóźniająca. Najpierw rejestr x3 jest inicjalizowany, a następnie jest on dekrementowany dopóki nie zostanie wyzerowany. Za pętlą inkrementowany jest rejestr x2. Wpisanie jego zawartości na wyjście cyfrowe następuje w linii 9. Ostatnią instrukcją jest skok do początku pętli głównej. Aby miganie było widoczne dla człowieka wartość wpisywana do x3 musi być stosunkowo duża. Na potrzeby symulacji przygotowano program *code/blink.S*. Jediną zmianą jest zmniejszenie liczby iteracji do 5. Można ją uruchomić poleceniem:

do scripts/simulate_blink.do

Uzyskany wynik prezentuje **rysunek 44**. Tym razem nie są na nim pokazane kolejne instrukcje, ponieważ przy tym powiększeniu byłby one nieczytelne. Poza zegarem i resetem pokazany jest stan trzech wykorzystanych rejestrów. Gpio_state obrazuje wartość wpisaną

do rejestru wyjść, a gpio przedstawia w systemie binarnym stan podawany na wyjścia układu FPGA.

Podsumowanie

Mam nadzieję, że udało mi się zainteresować czytelników tematem tworzenia własnych mikrokontrolerów. Ostatnim etapem kursu będzie uruchomienie rdzenia „w krzemie”. Naszą docelową platformą będzie płyta MAXimator (**fotografia 45**) wyposażona w układ MAX10 firmy Intel (dawniej Altera) [13].

Rafał Kozik
rafkozik@gmail.com

Bibliografia:

- [1] <http://bit.ly/2MFFapx>, dostęp na dzień 29.04.2019
- [2] <http://bit.ly/2YPOEGY>, dostęp na dzień 29.04.2019
- [3] Zeloof S., First IC, <http://bit.ly/33dwwEx>, dostęp na dzień 29.04.2019
- [4] <http://bit.ly/2MHjQQL>
- [5] Computer Structures, MIT 6.004, Spring 2019, <http://bit.ly/33dx4dL>
- [6] MIT 6.004 RISC-V ISA Reference Card, <http://bit.ly/2TcOS43>
- [7] The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Document Version 2.2, Editors Andrew Waerman and Krste Asanovic, RISC-V Foundation, May 2017
- [8] <http://bit.ly/31jY2OV>
- [9] <http://bit.ly/2yDgVjP>
- [10] Quartus Prime Lite Edition, <https://intel.ly/2MC6TYp>
- [11] Ataki na procesory – PortSmash, TLBleed, Foreshadow, <http://bit.ly/2M307eL>
- [12] Patterson D., Hennessy J., Computer Organization and Design: The Hardware Software Interface (RISC-V Edition), Morgan Kaufmann, 1st ed, 2017
- [13] MAXimator Altera MAX10 FPGA Evaluation Board, <http://bit.ly/2YjneNp>
- [14] Great Ideas in Computer Architecture (Machine Structures), CS61C Berkeley with Nicholas Weaver, Spring 2019
- [15] Moduł dla Visual Studio Code <http://bit.ly/2YPOjug>

Miesięcznik „Elektronika Praktyczna” (12 numerów w roku) jest wydawany przez AVT-Korporacja Sp. z o.o. we współpracy z wieloma redakcjami zagranicznymi.

Wydawca:

AVT-Korporacja Sp. z o.o.
03-197 Warszawa, ul. Leszczyńska 11
tel.: 22 257 84 99, faks: 22 257 84 00

Adres redakcji:

03-197 Warszawa, ul. Leszczyńska 11
tel.: 22 257 84 60
faks: 22 257 84 00
e-mail: redakcja@ep.com.pl
www.ep.com.pl

Redaktor Naczelny:

Wiesław Marciniak

**Redaktor Programowy,
Przewodniczący Rady Programowej:**
Piotr Zbysiński

**Zastępca Redaktora Naczelnego,
Redaktor Prowadzący:**
Damian Sosnowski

**Zastępca Redaktora Naczelnego,
Menedżer Magazynu**
Marcin Karbowiczek

Szef Pracowni Konstrukcyjnej:
Grzegorz Becker, tel.: 22 257 84 58

Redaktor strony internetowej www.ep.com.pl

Dariusz Welik

Zespół marketingu i reklamy:

Katarzyna Gugala, tel.: 22 257 84 64
Adam Kęska, tel.: 22 257 84 63
Bożena Krzykawska, tel.: 22 257 84 42
Grzegorz Krzykowski, tel.: 22 257 84 60
Maksymilian Hoser, tel.: 22 257 84 65

Sekretarz Redakcji:

Grzegorz Krzykowski, tel.: 22 257 84 60

DTP i okładka:

MAD Sp. z o.o.

Stali Współpracownicy:

Jacek Bogusz, Lucjan Bryndza, Jarosław Doliński,
Andrzej Gawryluk, Krzysztof Górski, Tomasz Jabłoński,
Michał Kurzela, Szymon Panecki, Sławomir Skrzyński,
Ryszard Szymański, Adam Tatuś, Robert Wołgajew

Uwaga!

Kontakt z wymienionymi osobami jest możliwy via e-mail, według schematu: imię.nazwisko@ep.com.pl

Prenumerata w Wydawnictwie AVT

www.avt.pl/prenumerata
lub tel.: 22 257 84 22
e-mail: prenumerata@avt.pl
www.sklep.avt.pl, tel.: 22 257 84 66



Prenumerata w RUCH S.A.

www.prenumerata.ruch.com.pl
lub tel.: 801 800 803, 22 717 59 59
e-mail: prenumerata@ruch.com.pl



Wydawnictwo
AVT-Korporacja Sp. z o.o.
należy do Izby Wydawców Prasy

**Copyright AVT-Korporacja Sp. z o.o.
03-197 Warszawa, ul. Leszczyńska 11**

Projekty publikowane w „Elektronice Praktycznej” mogą być wykorzystywane wyłącznie do własnych potrzeb. Korzystanie z tych projektów do innych celów, zwłaszcza do działalności zarobkowej, wymaga zgody redakcji „Elektroniki Praktycznej”. Przedruk oraz umieszczenie na stronach internetowych całości lub fragmentów publikacji zamieszczonych w „Elektronice Praktycznej” jest dozwolone wyłącznie po uzyskaniu zgody redakcji. Redakcja nie odpowiada za treść reklam i ogłoszeń zamieszczonych w „Elektronice Praktycznej”.





Wykaz firm ogłaszających się w tym numerze „Elektroniki Praktycznej”

Reklamy stron internetowych na str. 12

AKSOTRONIK.....	139
ARMEL	139
BORNICO.....	8
COMPUTER CONTROLS.....	11
CONTRANS TI	87
DELTA-OPTI	139
ELDAR	119
ELMARK AUTOMATYKA	106, 107
ELMAX.....	139
FARNELL ELEMENT14.....	140
FERYSER	12
GAMMA	12
HAMMOND	9
MICROCHIP	7
OEM AUTOMATIC.....	118
PIEKARZ	12
REL POL	84, 85
RENEX	12
ST MICROELECTRONICS.....	
.....	5, 39, 41, 45, 51
WOBIT.....	117

uniTerm – termostat z wyświetlaczem OLED

Niewielkie urządzenie, którego głównym zadaniem jest utrzymanie zadanej temperatury w trybie schładzania lub ogrzewania. Na graficznym wyświetlaczu OLED prezentowane są parametry pracy, a dla lepszego efektu wzbogacone prostą animacją. Spójny i czytelny interfejs użytkownika sprawia, że obsługa urządzenia jest łatwa i intuicyjna.

Radiowy sterownik serwomechanizmów

Serwomechanizmy mogą ustawić oś w ściśle określonym położeniu. Zaprezentujemy układ, który umożliwia wybranie jednego z trzech wcześniej ustalonych wychyleń za pomocą przełącznika. Jednak informacja o ustawieniu będzie przesyłana drogą radiową, od nadajnika z przełącznikiem do odbiornika połączonych z serwowym. Urządzenie można zastosować np. do zamykania, uchylania i pełnego otwierania wywietrznika.

Sztuczne obciążenie zmiennoprądowe w.cz. o impedancji 50 ohm

W drugiej części publikacji zaprezentujemy projekt sztucznego obciążenia w. cz. o dopuszczalnej ciągłej mocy strat rzędu 10...20 W przeznaczonego do pracy w pasmie KF. Urządzenie pozwala na komfortowe testowanie urządzeń nadawczych, ponieważ oprócz precyzyjnego wyjścia pomiarowego zostało wyposażone także w 12 detektorów progowych ze wskaźnikami na diodach LED. Dzięki temu można szybko i łatwo ocenić oddawaną moc w zakresie nawet do 40 W.

Hybrydowy wzmacniacz słuchawkowy na lampie NuTube

Wzmacniacz lampowy łączy się z urządzeniem o sporym gabarycie, znacznym ciężarze, dużym poborze mocy i ilością generowanego ciepła. W czasach lampowej świetności istniały lampy miniatury przystosowane do urządzeń przenośnych, lecz opracowana przez firmę Korg lampa Nutube 6P1 mocno podnosi poprzeczkę.

Śnieżynka LED

W oczekiwaniu na dzień przyozdabiania świątecznego drzewka proponujemy czytelnikom EP wykonanie Śnieżynki LED. Tego niewielkiego gadżetu można użyć w charakterze dekoracji np. na biurko. Będzie urozmaicheniem każdego wnętrza i wprowadzi świąteczny klimat. A wcześniej, montaż układu może być doskonałym pretekstem, aby wspólnie z rodziną zrobić coś niecodziennego.

a ponadto tematy wiodące EP12:

CZUJNIKI DO POMIARU PARAMETRÓW ŚRODOWISKOWYCH

oraz

KOMPUTERY I MODUŁY JEDNOPŁYTKOWE

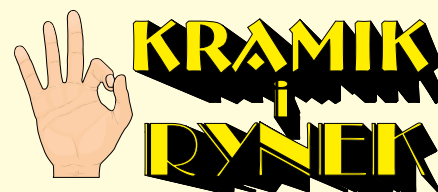
Ogłoszenia i reklamy hurtowni, sklepów, importerów, producentów, dealerów, itp. są płatne. Cena podstawowego modułu (35x20 mm) wynosi 66 zł + VAT. Koszt minimalnej ramki dla ogłoszeń o wielkości 3 modułów wynosi 198 zł + VAT. Moduły można łączyć zarówno w pionie jak i w poziomie. Maksymalna szerokość ogłoszenia to 5 modułów, wysokość 12 modułów. Rabaty stosujemy wyłącznie dla reklam powyżej 8 modułów: 4-6 emisji 10%, 7-11 emisji 15% i od 12 emisji 25%.

Oferta specjalna:

- publikacja fragmentów cennika w ramce o wielkości: 8 modułów w pionie cena 264 zł + VAT, 9 modułów w poziomie 305 zł + VAT
- rabat specjalny dla firm poszukujących pracowników wynosi 25% (wyłącznie dla dużych reklam).

Wszelkich informacji udziela **Grzegorz Krzykawski, tel. 22 257 84 60**, e-mail: grzegorz.krzykawski@ep.com.pl.

Reklamy do tej rubryki mogą być przygotowane przez Zamawiającego w CMYK, w formacie CDR (wersja do 2018), AI, EPS (tekst zmieniony na krzywe), PSD, PDF (programy Adobe w wersji do CC 2019 włącznie PC/Mac) z próbnym wydrukiem. Małe reklamy mogą być przygotowane w redakcji (gratis) na podstawie odręcznego szkicu. Opracowania te nie będą jednak wówczas uzgadniane z Zamawiającym przed oddaniem do druku.



Redakcja nie odpowiada za treść reklam i ogłoszeń zamieszczonych w Elektronice Praktycznej

Hurtownia elementów elektronicznych "AKSOTRONIK" zaprasza do swojego sklepu internetowego. Zaloguj się i kupuj ON-LINE na naszej stronie:

WWW.AKSOTRONIK.COM.PL

Przetwornica Step-Down LM2596 3,40 PLN

Moduł WIFI do Arduino 16,60 PLN

Elektromagnes 6V oraz 12V Cena od 15,90 PLN

Woltomierze LED AC oraz DC Cena od 7,50 PLN

Kable do płytek stykowych Cena od 1,30 PLN

Aksotronik ELEMENTY ELEKTRONICZNE WWW.AKSOTRONIK.COM.PL

Czujniki indukcyjne Cena od 7 PLN

Anteny WIFI oraz GSM Cena od 2,80 PLN

Silniki krokowe 200 oraz 400 kroków Cena od 30 PLN

Moduł bluetooth HC05 do Arduino Cena 20 PLN

Moduły z przekładnikiem Cena od 3,60 PLN

Zarówki LED do samochodu Cena od 1,60 PLN

Uwaga!!! Powyższe ceny dotyczą zakupów minimalnych ilości hurtowych, poprzez nasz sklep internetowy. W swojej ofercie posiadamy m.in.: półprzewodniki (diody, układy scalone, tranzystory, triaki, elementy optoelektroniczne), elementy dystansowe, złączki, przełączniki, elementy akustyczne, rezystory, kondensatory, kwarce, podstawki, moduły Arduino.

Zapraszamy do kontaktu: INFO@aksotronik.com.pl, tel: (22) 783-20-51

RACK i Eurocarta 19" Wyposażenie szaf 19"

www.obudowa.pl

Producent obudów dla elektroniki tel. 032-230-2301

KITY AVT

<http://bit.ly/2FNvrIN>

www.ep.com.pl

www.delta.poznan.pl

Wielkość matrycy: 2,1 Mpx (Sony IMX290)
Obiektów: 2,8 - 12 mm (kąt widzenia 85° - 23°)
Rozdzielczość: 1920 x 1080 (1080p)
Zasięg oświetlacza IR: 60 m
Protokoły sieciowe: TCP/IP, NTP, HTTP, ONVIF 2.4
PoE 802.3af, Klasa szczelności IP66
3D-DNR, WDR, detekcja ruchu, H.265 / H265+
Aplikacje dla: Android, iOS
Bardzo dobra jakość w jeszcze lepszej cenie

▶ **APTI-250C61-2812WP**

Wielkość matrycy: 2,1 Mpx (przetwornik 1/2.9")
Obiektów: 3,6 mm (kąt widzenia 69°)
Rozdzielczość: 1920 x 1080 (1080p)
Przepływność (bitrate): 64 ... 12000 kbit/s
Protokoły sieciowe: TCP/IP, NTP, HTTP, ONVIF 2.4
PoE 802.3af, Klasa szczelności IP66
3D-DNR, WDR, detekcja ruchu, H.265 / H265+
Aplikacje dla: Android, iOS
Bardzo dobry stosunek jakości do ceny

▶ **APTI-201D2-36WP**

Wielkość matrycy: 4,0 Mpx (przetwornik 1/2.5")
Obiektów: 2,8 - 12 mm (kąt widzenia 81° - 23°)
Rozdzielczość: 2560 x 1440
Obsługa audio - zewnętrzny mikrofon
Protokoły sieciowe: TCP/IP, NTP, HTTP, ONVIF 2.4
PoE 802.3af, Klasa szczelności IP66, H.265
3D-DNR, F-DNR (Defog), WDR, detekcja ruchu, Aplikacje dla: Android, iOS
Wysoka rozdzielczość dostępna w atrakcyjnej cenie

▶ **APTI-41VA3-2812WP**

Największy wybór w Polsce

ELMAX 1988

OBWODY DRUKOWANE
Produkcja, Projektowanie, Montaż

Certyfikat Underwriters Laboratories
UL 94V-0 E480148 TYPE 1

Zakład produkcyjny:
05-260 Marki
ul. Duża 1
tel. 22 781 63 95
22 761 95 80
fax. 22 781 63 95 w 23
www.elmax.waw.pl
elmax@elmax.waw.pl

Płytki jednostronne
Płytki dwustronne
Płytki na podłożu aluminium
Płyty czolowe FR4

Serie dowolne
Prototypy
Maksymalny wymiar płytek 1w 630 mm

Dokumentacja technologiczna
Dokumentacja konstrukcyjna
Trawione szablony SMD

Montaż elektroniki
Krótkie terminy
Wykonania super expresowe

Aktywny kalkulator prototypów na stronie internetowej

Pokrycie Sn lub SnPb inne na życzenie
Maski, opisy montażowe w różnych kolorach

Sprawdź nasz największy asortyment od wiodących dostawców

- Produkty w magazynie, gotowe do wysyłki
- Oferta produktowa od etapu projektowania, po utrzymanie ruchu
- Doskonała, lokalna obsługa klienta
- Kwotacje na produkty spoza katalogu i niestandardowe produkty dostosowane do indywidualnych potrzeb klienta

KLEIN
TOOLS



Honeywell

OMRON

NXP

ON Semiconductor ON

maxim
integrated.

Panasonic

KEYSIGHT
TECHNOLOGIES

multicomp PRO

METCAL

Life Is On | Schneider
Electric

ST life.augmented

VISHAY

RENESAS

MICROCHIP

ROHDE & SCHWARZ

FLUKE

Tektronix

TE

molex

Weller

pl.farnell.com

 **Farnell**
AN AVNET COMPANY