

OSCYLOSKOPY (I NIE TYLKO) USB

www.ep.com.pl

ELEKTRONIKA PRAKTYCZNA + FTP

● Międzynarodowy magazyn elektroników konstruktorów ● Czerwiec ● 6/2021 ●

ELEKTRONIKA
PRAKTYCZNA

WYKRYWACZ MASZCZKI • EKRAN SMARTFONA JAKO INTERFEJS MOBILNY DO AMPLITUNERA, CZYLI APLIKACJA MOBILNA POPRZECZ UART W PRAKTYCE • SPECTRA – ANALIZATOR WIDMA SYGNAŁU AUDIO • MODUŁ DSP AUDIO DO RASPBERRY PI • EXPANDER 40 I/O DLA PI ZERO • MIKROKROKOWY STEROWNIK SILNIKA • MINIATUROWY WZMACNIACZ SŁUCHAWKOWY • STEROWNIK LED RGB Z UKŁADEM AL1783 STEROWANY PRZECZ I²C • WYŁĄCZNIK ZASILANIA Z OPÓŹNIENIEM • HYBRYDA BIS – WZMACNIACZ HYBRYDOWY Z NIEKONWENCJONALNYM ZASILANIEM

Projekty Podzespoły Technologia Opisy Karty Nowości Aparatura i narzędzia Rynek Automatyka Archaizm Karty AIT

EP W TOKU

NA EP.COM.PL

tylko Prenumeratorzy
mają dostęp do artykułów przed ich publikacją w EP

inspirujące, użyteczne projekty

Wykrywacz maseczki • Ekran smartfona jako interfejs mobilny do amplitunera, czyli aplikacja mobilna poprzez UART w praktyce • Spectra – analizator widma sygnału audio • Moduł DSP Audio do Raspberry Pi • Expander 40 I/O dla Pi Zero • Mikrokrokový sterownik silnika • Miniaturowy wzmacniacz słuchawkowy • Sterownik LED RGB z układem AL1783 sterowany przez I²C • Wyłącznik zasilania z opóźnieniem • Hybryda bis – wzmacniacz hybrydowy z niekonwencjonalnym zasilaniem

podzespoły, sprzęt, aplikacje

Sztuczna inteligencja w praktycznej elektronice • Praktyczne aplikacje scalonych układów AFE. Front-endy EKG do urządzeń wearable • Nie tylko oscyloskopy USB • Sprzętowy hackathon on-line • Analog Discovery Pro 3000 firmy Digilent. Nowy przenośny kombajn pomiarowy • ARM Keil – bezpieczeństwo w systemach IoT i embedded • Praca z płytką drukowaną – w co wyposażać stanowisko

tutoriale

Symulacja i pomiar, czyli LTspice i Analog Discovery 2 w rękach konstruktora. O czym nie mówiliśmy, a powinniśmy • Bez programowania Androida. Projekt z użyciem BLE i aplikacji mobilnej • Projektowanie interfejsów graficznych z użyciem TouchGFX • FPGA a open source. Prosty mikrokontroler DO6502 • Pomiar natężenia prądu w systemach wbudowanych. Praktyczny poradnik

kursy

Systemy dla Internetu Rzeczy. Pozyskiwanie energii słonecznej, śledzenie maksymalnego punktu mocy (MPPT) • Eksperymenty z FPGA. Coś dla graczy

AI I ML – ZASTOSOWANIA, KORZYŚCI I ZAGROŻENIA

TEMAT NUMERU

18,90 zł (w tym 8% VAT) • PRICE: 8 EUR



ISSN 1230-3526 Indeks 357677



9 771230 352214



**Zaprenumeruj
Elektronikę Praktyczną,
a zawsze dostaniesz
najnowszy numer wprost
do Twojej skrzynki!**

**na start
do 6* wydań gratis**

**po 5 latach
nieprzerwanej
prenumeraty
do 12* wydań gratis**

PREZENT

do każdej opłaconej prenumeraty

* Cena prenumeraty rocznej **na start** wynosi 207,90 zł. Przy zamówieniu prenumeraty dwuletniej za 340,20 zł oszczędność wynosi równowartość sześciu wydań Elektroniki Praktycznej.

Przedłużasz prenumeratę? Aby otrzymać zniżkę lojalnościową, przedłuż prenumeratę po zalogowaniu się do swojego panelu na www.avt.pl, gdzie znajdziesz atrakcyjną ofertę prenumeraty, która uwzględnia przysługujące Ci zniżki za lojalność. Po 5 latach nieprzerwanej prenumeraty otrzymasz **rabat 50%** na prenumeratę dwuletnią. Oferta dotyczy prenumeraty drukowanej.

Każdorazowo opłacenie prenumeraty jest premiowane prezentem. W tym numerze do wyboru mamy: koszulkę z logo Elektroniki Praktycznej (rozmiary L, XL) lub płytę muzyczną. Zgłoś się po prezent mailowo – prenumerata@avt.pl.

**Wszystkie opcje prenumeraty i e-prenumeraty znajdziesz na stronie
www.UlubionyKiosk.pl**

prenumerata@avt.pl

AVT-Korporacja sp. z o.o., ul. Leszczyńska 11, 03-197 Warszawa, konto 18 1050 1012 1000 0024 3173 1013

eprasa.pl 1d618dc1b2

Test na inteligencję

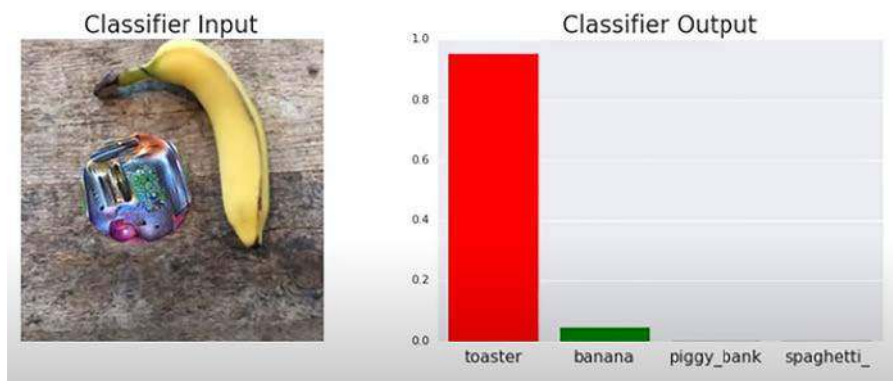
Jednym z przyjętych sposobów określenia zdolności maszyny do inteligentnego myślenia, podobnego do ludzkiego, jest test Turinga. Pomysłodawcą tego sposobu był Alan Mathison Turing, żyjący w latach 1912–1954 brytyjski matematyk i kryptolog, który mocno przyczynił się do rozwoju informatyki i sztucznej inteligencji w czasach kształtowania się tych technologii. Istota zaproponowanego przez niego eksperymentu polega na przeprowadzeniu konwersacji pisemnej. Osoba decydująca przez określony

czas prowadzi dialog, z użyciem klawiatury i monitora, z dwoma „rozmówcami” – człowiekiem oraz urządzeniem (programem działającym na komputerze). Po zakończeniu konwersacji osoba decydująca musi wskazać, która rozmowa odbyła się z udziałem człowieka, a w której uczestniczyło urządzenie. Brak możliwości jednoznacznego uzasadnionego zidentyfikowania rozmówców oznacza, że urządzenie przeszło test pozytywnie.

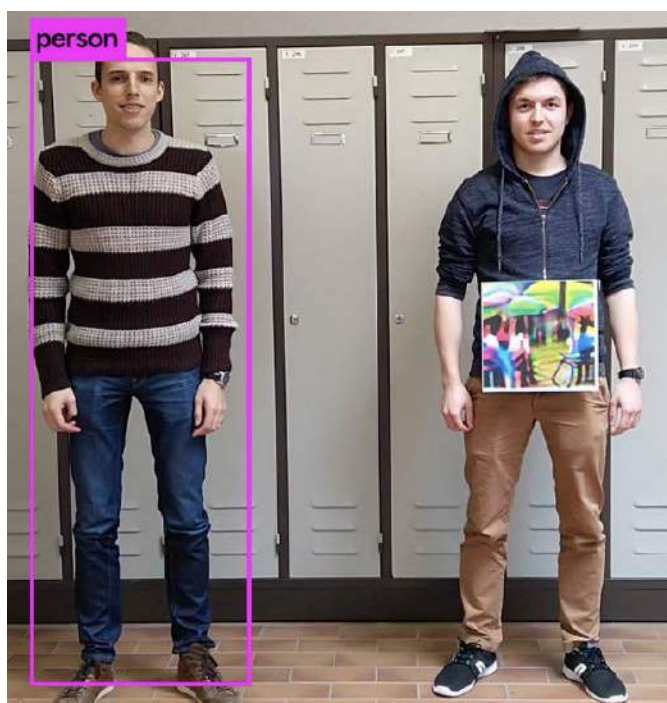
Metoda budzi wiele wątpliwości, ale ostatecznie trzeba zgodzić się z tym, że jeżeli urządzenie potrafiłoby „wmówić” komuś, że jest człowiekiem, to musiałoby być naprawdę inteligentne. Czy to trudne zadanie? Tak, nawet bardzo trudne. Od 1990 r. do 2019 r. organizowany był konkurs polegający na opracowaniu chatbota, który przeszedłby test Turinga. Nagrodą główną był 18-karatowy złoty medal oraz 100000 dolarów. W każdej edycji nie brakowało chętnych do podjęcia wyzwania, ale pomimo wieloletnich zmagania zwycięzcy nie wyłoniono. Natomiast sam konkurs zyskał niezbyt dobre opinie – ponieważ sędziowie wiedzieli, że mają do czynienia z programami, tak kierowali konwersacją, aby wymusić błędy charakterystyczne dla botów. Natomiast programiści uciekali się do różnych, niezbyt uczciwych zabiegów – np. chatboty miały udawać dzieci, nie do końca znające język i tamtejszą kulturę, czy osoby ze schizofrenią. Całe przedsięwzięcie zaczęło odbiegać od istoty sztucznej inteligencji. Pomimo to, startujące w konkursie boty, takie jak Mitsuku czy UberBot, uważa się za jedne z najlepszych na świecie. Co ciekawe – w 2019 r. do konkursu zgłoszono także polski projekt. Był to chatbot o nazwie Eva.

Czatboty to tylko mała sfera zastosowań AI. Pokładamy wielkie nadzieje w sztucznej inteligencji i znajdujemy dla niej kolejne zastosowania. Czy zdajemy sobie sprawę z tego, że daje się wykryć i oszukać? Ciekawy przypadek dotyczy amerykańskich uczniów, którzy zetknęli się z platformą edukacyjną Edgenuity. Zaczęli podejrzewać, że coś jest nie tak, ponieważ czasami otrzymywali oceny w kilka sekund po wysłaniu prac. Stało się oczywiste, że stoi za tym program komputerowy, który bazuje na jakimś algorytmie, sprawdzając ich prace. Szybko doszli do tego, że w odpowiedziach do pytań wystarczy napisać cokolwiek oraz listę słów kluczowych, aby uzyskać 100% poprawnych odpowiedzi.

Jeszcze bardziej interesujący jest kolejny przykład. Podczas konferencji naukowej Neural Information Processing Systems w Long Beach w USA zaprezentowano sposób na oszukanie inteligentnych systemów automatycznego rozpoznawania obiektów. Naukowcy zastosowali niewielkie wydrukowane obrazki, które umieścili obok banana. Efekt był zaskakujący – zamiast banana sztuczna inteligencja rozpoznała... toster (fotografia 1). Wcześniej owoc był rozpoznawany prawidłowo, ale gdy obok niego pojawił się specjalnie spreparowany, zniekształcony obraz toster, sztuczna inteligencja zgłupiała. Podobny efekt uzyskali naukowcy z Katolickiego Uniwersytetu Lowańskiego w Belgii. Dowiedli oni, że wystarczy zasłonić część sylwetki człowieka dużym zdjęciem ukazującym łatwo rozpoznawalne obiekty, aby stał się on niewidzialny dla systemów rozpoznawania ludzi z obrazów kamer czy zdjęć (fotografia 2). Co na ten temat powie Eva? Zapytajcie sami: <https://chatwitheva.com/>.

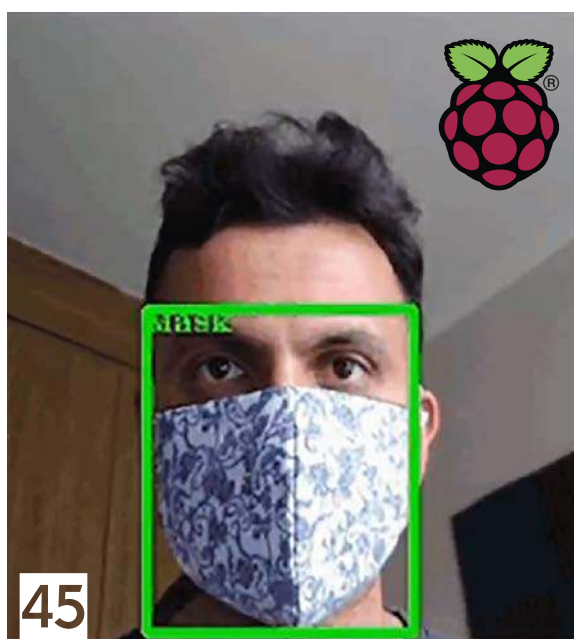


Fotografia 1. (źródło: <https://bit.ly/3fP5wna>)



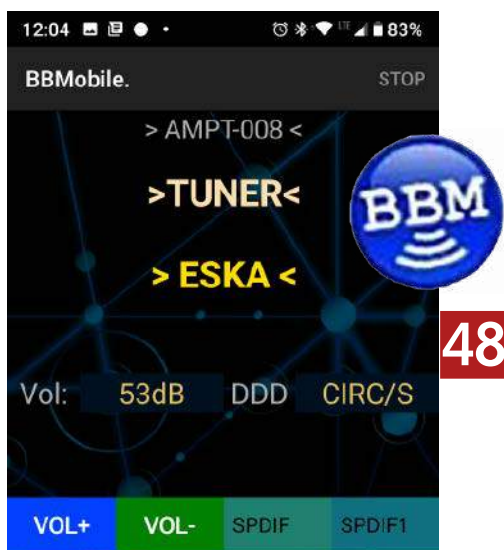
Fotografia 2. (źródło: <https://bit.ly/3fN0l5E>)

Damian Sosnowski



Wykrywacz maseczki

Jeszcze do niedawna maseczki były naszą codziennością i niektórych okolicznościach ciągle są wymagane (piszemy ten artykuł w czasie, gdy nakaz jeszcze obowiązuje) – w sklepach, centrach handlowych itd. były one wymagane. Upominanie nienoszących ich osób jest uporczywe, dlatego też dobrze jest oddać to zadanie w ręce maszyny.



Ekran smartfona jako interfejs mobilny do amplitunera, czyli aplikacja mobilna poprzez UART w praktyce

W zeszłym roku, w kilku wydaniach „Elektroniki Praktycznej” (EP4...7/20) ukazał się cykl artykułów na temat tworzenia aplikacji działającej na dowolnym smartfonie z użyciem języka JSON. Na smartfonie trzeba tylko zainstalować aplikację BBMobile, a cały design i funkcjonalność aplikacji realizowane są przez moduł sprzętowy.

Nie przeocz

Nowe podzespoły.....	5
Dodaj do obserwowanych	10
Koktajl newsów.....	106

Projekty soft

Wykrywacz maseczki	45
Ekran smartfona jako interfejs mobilny do amplitunera, czyli aplikacja mobilna poprzez UART w praktyce	48

Projekty

Spectra – analizator widma sygnału audio	17
Moduł DSP Audio do Raspberry Pi	23

Miniprojekty

Expander 40 I/O dla Pi Zero.....	30
Mikrokrokový sterownik silnika	32
Miniaturowy wzmacniacz słuchawkowy	35
Sterownik LED RGB z układem AL1783 sterowany przez I ² C.....	36
Wyłącznik zasilania z opóźnieniem.....	38

Temat numeru: AI i ML – zastosowania, korzyści i zagrożenia

Sztuczna inteligencja w praktycznej elektronice	40
---	----

Projekty czytelników

Hybryda bis – wzmacniacz hybrydowy z niekonwencjonalnym zasilaniem	52
--	----

Poradnik implementacji

Praktyczne aplikacje scalonych układów AFE (4). Front-endy EKG do urządzeń wearable	102
---	-----

Prezentacje

ARM Keil – bezpieczeństwo w systemach IoT i embedded	56
Praca z płytką drukowaną – w co wyposażać stanowisko.....	58
Sprzętowy hackathon on-line	62
Analog Discovery Pro 3000 firmy Digilent. Nowy przenośny kombajn pomiarowy.....	88

Wybór konstruktora:

Nie tylko oscyloskopy USB	80
---------------------------------	----

Notatnik konstruktora

Bez programowania Androida. Projekt z użyciem BLE i aplikacji mobilnej (1)	65
Projektowanie interfejsów graficznych z użyciem TouchGFX (4).....	70
FPGA a open source. Prosty mikrokontroler DO6502.....	75
Pomiary natężenia prądu w systemach wbudowanych (1). Praktyczny poradnik.....	91
Symulacja i pomiar, czyli LTspice i Analog Discovery 2 w rękach konstruktora (23). O czym nie mówiliśmy, a powinniśmy.....	96

Kursy

Systemy dla Internetu Rzeczy (49). Pozyskiwanie energii słonecznej, śledzenie maksymalnego punktu mocy (MPPT).....	109
Eksperymenty z FPGA (19). Coś dla graczy	116

Prenumerata	2
Od wydawcy.....	3
Kramik i rynek	122
Hity następnego numeru	123

nowe podzespóły

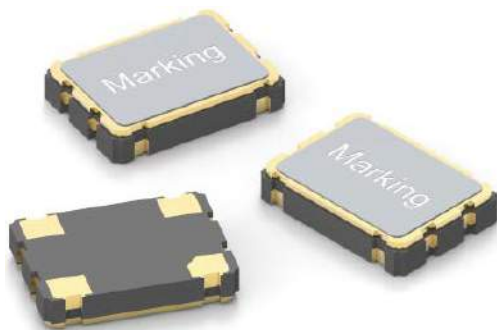
Z kilkuset nowości wybraliśmy te, których nie wolno przeoczyć. Bieżące nowości można śledzić na www.elektronikaB2B.pl

Specjalizowany mikrokontroler do odtwarzania komend głosowych

Do oferty firmy Rutronik wchodzi mikrokontroler S1C31D51 produkcji firmy Epson, zaprojektowany specjalnie do aplikacji komunikujących się z użytkownikiem poprzez reprodukcję dźwięków i komend głosowych za pomocą głośnika lub buzzera. Może on znaleźć zastosowanie w elektronice domowej, instalacjach przemysłowych, urządzeniach fitness oraz systemach alarmowych. Został oparty na rdzeniu ARM Cortex-M0+. Dodatkowo zawiera specjalną jednostkę sprzętową (HW Processor), odpowiedzialną za odtwarzanie mowy i podkładu dźwiękowego w dwóch niezależnych kanałach. Opracowany przez Epsona algorytm zapewnia optymalną jakość dźwięku.

HW Processor odciąża jednostkę CPU od zadań związanych z przetwarzaniem dźwięku, pozwalając jej realizować własne zadania związane ze sterowaniem. Dodatkowo wbudowane funkcje bezpieczeństwa, m.in. automatycznej kontroli integralności danych w pamięciach RAM i Flash oraz pamięci peryferyjnej QSPI Flash, pozwalają na realizowanie tych funkcji bez obciążania CPU. Opracowany przez firmę Epson koder-dekoder EOVS (*Epson Own Voice*) kompresuje sygnał audio (15635 kHz, 16 kbps), pozwalając znacznie zaoszczędzić pojemność pamięci. Oprogramowanie projektowe ESPER2 Voice Data Creation dla komputerów PC ułatwia generowanie komend głosowych w 12 językach, eliminując konieczność nagrania studyjnych. Może też importować gotowe nagrania w formacie WAV.

www.rutronik24.com



Oscylatory kwarcowe z wyjściami LVDS i LVPECL na zakres 100...156,25 MHz

Würth Elektronik powiększa ofertę oscylatorów kwarcowych WE-SPXO o nowe wersje z wyjściami LVDS i LVPECL oraz o nową serię oscylatorów 32,768 kHz. Są one zamykane w hermetycznych obudowach i charakteryzują się zwiększoną stabilnością w stosunku do wcześniejszych wersji. Nowe oscylatory z wyjściami LVDS

i LVPECL (IQXO-618-18, IQXO-618-33, IQXO-623 i IQXO-624) występują w wersjach o częstotliwości wyjściowej od 100 do 156,25 MHz. Są przeznaczone do zastosowań w aplikacjach o dużej przepustowości: Gigabit Ethernet, Fibre Channel, PCI Express i SONET/SDH. Zapewniają mniejszy błąd jitteru od wcześniejszych wersji i szerszy zakres dopuszczalnej temperatury pracy od -40 do +105°C.

Nowe oscylatory o częstotliwości wyjściowej 32,768 kHz (CFPS-102, CFPS-104, CFPS-107, CFPS-109, IQXO-402 i IQXO-404) mogą stanowić zamienniki tradycyjnych rezonatorów zegarkowych. Są produkowane w obudowach o trzech wymiarach: 3,2×2,5 mm, 2,5×2,0 mm i 2,0×1,6 mm. Zapewniają stabilność od ±25 ppm i mogą pracować w zakresie temperatury otoczenia od -40 do +125°C. Ich najważniejszą zaletą w stosunku do rezonatorów zegarkowych jest większa dokładność i stabilność częstotliwości w funkcji temperatury. Ponadto, w odróżnieniu od rezonatorów, nie wymagają dołączania komponentów zewnętrznych. Ich głównym zastosowaniem są mikrokontrolery z wbudowanym zegarem RTC.

www.we-online.com



First CK440 Clock Generator for Next-Gen Intel Server Platforms



Pierwszy generator sygnałów zegarowych zgodny ze specyfikacją CK440Q firmy Intel

Renesas Electronics wprowadza do sprzedaży generator sygnałów zegarowych 9SQ440 o małym poziomie jitteru, zaprojektowany do systemów komputerowych i centrów danych, opartych na platformie sprzętowej nowej generacji firmy Intel. Jest to pierwszy na rynku generator zgodny ze specyfikacją CK440Q. Został zaprojektowany specjalnie pod kątem przyszłych generacji mikroprocesorów Xeon i systemów opartych na szynie PCIe Gen5. Nadaje się do współpracy z oferowanymi przez Renesasa układami czasowymi do szyny PCIe, w tym z buforami zegarowymi PCIe Gen5.

Generator 9SQ440 pełni funkcję scentralizowanego generatora sygnałów zegarowych do taktowania mikroprocesorów serwerowych i układów współpracujących z szyną PCIe. Zawiera 20 wyjść różnicowych (7×100 MHz, 4×25 MHz + 9 wyjść multipleksowanych o częstotliwości od 25 do 100 MHz). Charakteryzuje się zawartością jitteru mniejszą niż 50 fs rms. Może pracować z modulacją *spread spectrum* (0%, 0,3% i 0,5%). Jest zamykany w 100-wyjściowej obudowie QFN o powierzchni 8×8 mm.

www.renesas.com

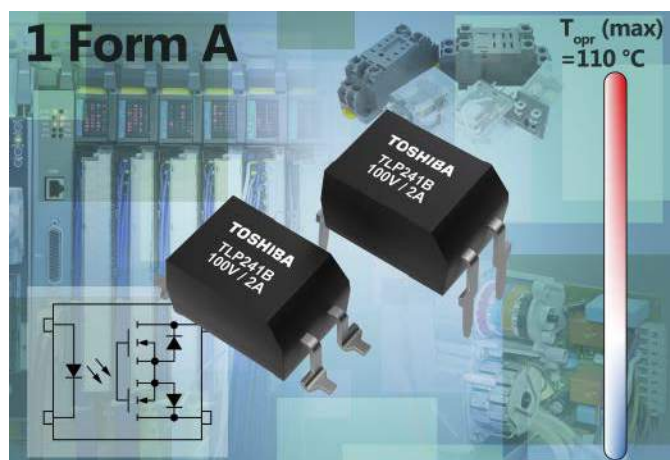


Miniaturowe, hermetyczne przekaźniki kontaktronowe o obciążalności 10 W/300 V

Firma Littelfuse powiększa swoje portfolio przekaźników kontaktronowych do testerów ATE, aparatury pomiarowej, telekomunikacji oraz instalacji alarmowych i przeciwpożarowych o nowe wersje produkowane w zminiaturyzowanych obudowach SIL i DIP. Przekaźniki te mogą przełączać napięcia do 300 VDC i zapewniają izolację między niskonapięciowym wejściem i zaciskami wyjściowymi do 2500 V rms. Są produkowane w czterech konfiguracjach: SPST-NO, SPST-NC, SPDT-CO i DPST-NO. Mogą pracować w przemysłowym zakresie temperatury od -40 do $+85^{\circ}\text{C}$. Są odporne na uderzenia mechaniczne do 50 g (11 ms) oraz na wibracje do 20 g (10...2000 Hz). Ich maksymalna częstotliwość przełączania wynosi 1 kHz, a niezawodność sięga miliardów cykli. Hermetyczna obudowa zapewnia odporność na narażenia środowiskowe. Opcjonalnym wyposażeniem jest ekran magnetyczny oraz dioda realizująca ochronę układu sterowania cewką.



www.littelfuse.com



2-amperowy fotoprzekaźnik przemysłowy w obudowie DIP4

TLP241B to kolejny wysokoprądowy fotoprzekaźnik z oferty firmy Toshiba, zoptymalizowany do zastosowań przemysłowych, np. w sterownikach PLC, instalacjach alarmowych, testerach ATE i systemach automatyki budynków. Zapewnia izolację wejście-wyjście do 5 kV rms i szeroki zakres dopuszczalnej temperatury pracy od -40 do $+110^{\circ}\text{C}$. Może przewodzić maksymalny prąd ciągły 2 A i impulsowy do 6 A. Dopuszczalne napięcie 100 V w stanie OFF między terminalami wyjściowymi pozwala na jego zastosowania w charakterze zamiennika większych, droższych i bardziej zawodnych przekaźników elektromechanicznych. Ze względu na brak elementów mechanicznych TLP241B charakteryzuje się krótkim czasem odpowiedzi i długim czasem bezawaryjnej pracy. Jest zamykany w obudowie DIP4 o wymiarach 6,4×4,6×3,7 mm z opcjonalnymi wyprowadzeniami typu Gull Wing.

www.toshiba.semicon-storage.com



Czujniki natężenia światła o wymiarach 2,0×1,0×0,5 mm

Sensory TSL2520 i TSL2521 to miniaturowe czujniki natężenia światła (ALS – ambient light sensor) zaprojektowane do najnowszych



smartfonów z wyświetlaczami *edge-to-edge*, pokrywającymi cały panel frontowy. Są obecnie najmniejszymi tego typu komponentami, zamykanymi w obudowach QFN o wymiarach zaledwie 2,0×1,0×0,5 mm z filtrem UV/IR. Dzięki dużej czułości optycznej i programowalnemu wzmocnieniu dają producentom dużą elastyczność w wyborze rodzaju szkła, pozwalając zapewnić bardzo dobre parametry rejestrowanych zdjęć i filmów w słabych warunkach oświetleniowych.



TSL2520 to czujnik ALS z czterema wewnętrznymi fotodiodami pracującymi w paśmie widzialnym i dwiema pracującymi w podczerwieni, zapewniający dużą dokładność pomiaru i eliminację interferencji IR. TSL2521 jest identyczny pod względem mechanicznym i funkcjonalnym, przy czym oferuje dodatkowo funkcję wykrywania migotania sztucznych źródeł światła (np. diod LED) w zakresie częstotliwości do 7 kHz. Pozwala to na realizację funkcji korekcji obrazu w smartfonie, eliminującej prążki i inne artefakty wywołane przez migotanie.

Oba czujniki umożliwiają pomiar natężenia światła w zakresie już od 1 mlx. Zapewniają szeroki zakres dynamiczny, równy 8192x. Pracują z napięciem zasilania od 1,8 V. Oferują konfigurowalne tryby oszczędnościowe, pozwalające obniżyć pobór prądu i wydłużyć czas pracy urządzeń bateryjnych.

www.ams.com

Termistor NTC do pracy w zakresie temperatury otoczenia do $+185^{\circ}\text{C}$

Firma Vishay zaprezentowała nowy wysokotemperaturowy termistor NTC z wyprowadzeniami ze stopu niklu i żelaza, przeznaczony do zastosowań w motoryzacji i innych aplikacjach wysokotemperaturowych. Jest to termistor z kwalifikacją AEC-Q200, mogący pracować w szerokim zakresie temperatury otoczenia od -55 do $+185^{\circ}\text{C}$.



Stop NiFe, zastosowany do produkcji wyprowadzeń termistora NTCLE350E4, wyróżnia się wyjątkowo małą rezystancją termiczną, pozwalającą uzyskać gradient termiczny $<0,01$ K/K i praktycznie eliminuje wydzielanie energii cieplnej do otoczenia, zapewniając dużą dokładność pomiaru. W przypadku innych przewodzących metali, np. miedzi, błąd pomiaru może sięgać kilku stopni. Aby zwiększyć niezawodność w środowisku o dużej wilgotności, termistor charakteryzuje się bardzo dobrą przyczepnością pomiędzy izolacją PEEK i lakierem epoksydowym. Minimalna średnica kulki pomiarowej (2,4 mm) zapewnia szybką odpowiedź, wynoszącą około 6 s w powietrzu.

Wyjątkowa dokładność termistora NTCLE350E4, krótki czas odpowiedzi oraz możliwość pracy w wysokiej temperaturze otoczenia



spawiają, że nadaje się on idealnie do pomiaru temperatury oleju i płynu chłodzącego silnik. Może być też stosowany w czujnikach MAP w kolektorze dolotowym i układach recyrkulacji spalin oraz w instalacjach HVAC.

Pozostałe parametry:

- rezystancja: 2,1...30 k Ω (tolerancja od 1%),
- beta (B25/85): od 3435 do 3984 K (tolerancja od $\pm 0,5\%$),
- maks. moc wydzielana: 100 mW.

www.vishay.com



Wzmacniacze pomiarowe do monitorowania natężenia prądu w środowiskach wysokotemperaturowych

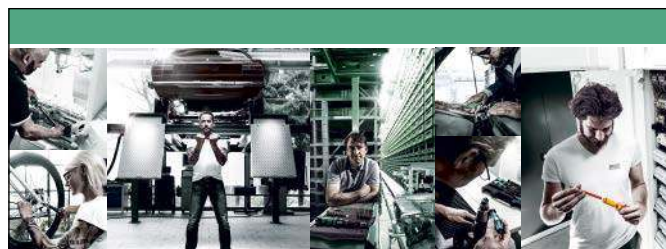
Wzmacniacze pomiarowe MCP6C02 i MCP6C04 firmy Microchip umożliwiają monitorowanie natężenia prądu w środowiskach wysokotemperaturowych. Są one identyczne pod względem funkcjonalnym, natomiast różnią się m.in. zakresem dopuszczalnej temperatury pracy oraz zakresem wejściowego napięcia sumacyjnego. MCP6C02 to wzmacniacz z kwalifikacją AEC-Q100 Grade 0, zaprojektowany do zastosowań w motoryzacji. Może pracować w zakresie temperatury otoczenia od -40 do $+150^{\circ}\text{C}$ i akceptuje napięcie wejściowe z zakresu od $+2,8$ do $+68$ V. MCP6C04 to wzmacniacz do aplikacji przemysłowych, pracujący w zakresie temperatury otoczenia od -40 do $+125^{\circ}\text{C}$ i akceptujący napięcia wejściowe od $+2,8$ do $+54$ V.

Układy MCP6C02 i MCP6C04 to wzmacniacze pracujące w konfiguracji high-side, wyróżniające się bardzo małym napięciem offsetu, bardzo małym błędem wzmocnienia oraz skutecznym tłumieniem tętnień zasilania i składowej sumacyjnej. Zapewniają precyzyjny pomiar małych sygnałów różnicowych na tle dużego napięcia sumacyjnego i w obecności silnych zaburzeń elektromagnetycznych. Umożliwiają programowanie wzmocnienia na jednej z trzech predefiniowanych wartości: 20, 50 lub 100 V/V. Architektura zero-drift pozwala na stosowanie wejściowych rezystorów pomiarowych o małej rezystancji i małej emisji ciepła.

Oba układy zawierają filtr EMI zapewniający ochronę przed interferencjami elektromagnetycznymi w.c.z. Mogą znaleźć zastosowanie w układach zarządzania zasilaniem i napędowych.

Wzmacniacz MCP6C02 jest zamykany w obudowach SOT-23-6 i VDFN-8 z wyprowadzeniami wettable flank, a jego ceny hurtowe wynoszą odpowiednio 1,22 USD i 1,41 USD przy zamówieniach 10 tys. sztuk. MCP6C04 jest zamykany w obudowie SOT-23-6, a jego cena hurtowa wynosi 0,98 USD.

www.microchip.com



Klucze i wkrętaki
dynamometryczne
dla profesjonalistów

Wera
BE A TOOL REBEL



T M E®

Electronic Components

TRANSFER MULTISORT ELEKTRONIK

USTRONNA 41, 93-350 ŁÓDŹ, POLSKA
TEL. 42 645 55 55, DSO@TME.EU

tme.eu

facebook.com/TME.eu
instagram.com/tme.eu
youtube.com/TMElectroniComponent

REKLAMA



Wielowarstwowe ferrytowe cewki indukcyjne SMD do zastosowań w filtrach obwodów NFC

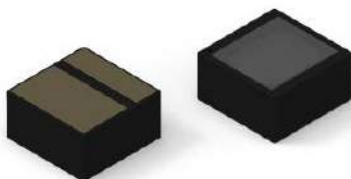
Firma TDK wprowadza na rynek serię wielowarstwowych ferrytowych cewek indukcyjnych SMD zaprojektowanych specjalnie do zastosowań w filtrach LC obwodów NFC. Cewki MLJ-H1005 dzięki wąskiemu przedziałowi tolerancji ($\pm 5\%$) pozwalają zminimalizować straty spowodowane niedopasowaniem impedancji anteny. Dodatkowo, dzięki zastosowaniu małostratnych materiałów konstrukcyjnych, wykazują one małe straty na częstotliwości 13,56 MHz w aplikacjach wysokoprądowych, ograniczając do minimum tłumienia mocy dostarczanej do anteny. Stanowią pod tym względem lepszą alternatywę dla cewek wcześniejszej serii MLJ-W. Ponadto do ich zalet należy niskoprofilowa konstrukcja i mały wpływ strumienia magnetycznego.



www.tdk.com

Laser IR o wymiarach do systemów przestrzennej identyfikacji obiektów

Würth Elektronik wprowadza do oferty laser IR do systemów pomiaru odległości i przestrzennego rozpoznawania obiektów, znajdujących zastosowanie m.in. w LiDAR-ach samochodowych i robotach autonomicznych. Model WL-VCSL (Vertical Cavity Surface Emitting Laser) pracuje na długości fali 940 nm i jest zamykany w miniaturowej obudowie SMD o wymiarach 3,5×3,5×1,8 mm. Zapewnia sprawność 35% i moc promieniowania sięgającą 2 W. Występuje w dwóch wersjach, różniących się polem emisji: 60°×45° i 110°×85°.



www.we-online.com



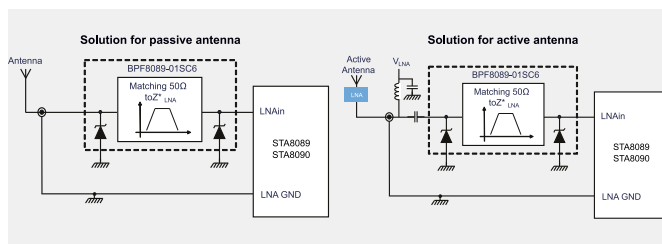
Przetworniki ultradźwiękowe na zakres częstotliwości 23...40 kHz

Oddział czujników firmy CUI powiększa ofertę przetworników ultradźwiękowych o nowe wersje nadajników, odbiorników i transceiverów z wyjściami analogowymi, oferujących zasięg transmisji od 0,2 do 18 m i kąt emisji wynoszący 75 lub 80 stopni. Są one zamykane w tanich, aluminiowych lub plastikowych obudowach do montażu przewlekane o średnicy od 9,8 mm i wysokości od 7 mm. Mogą pracować w zakresie dopuszczalnej temperatury otoczenia od -30 do +85°C.



Przetworniki serii CUSA i CUSP charakteryzują się zakresem częstotliwości znamionowych od 23 do 40 kHz i zakresem napięcia roboczego od 80 do 180 V. Ich poziom natężenia dźwięku (SPL) wynosi od 105 do 115 dB (30 cm), a zakres czułości od -75 do -68 dBV. Ceny hurtowe zaczynają się od 2,16 USD przy zamówieniach 100 sztuk.

www.cuidevices.com



Układ dopasowujący i zabezpieczający do odbiorników GNSS

STMicroelectronics wprowadza do oferty analogowy front-end o symbolu BPF8089-01SC6, przeznaczony do współpracy z odbiornikami GNSS. Pozwala on uprościć konstrukcję stopnia odbiorczego i zmniejszyć liczbę podzespołów dzięki wbudowanym funkcjom dopasowania anteny do impedancji 50 Ω oraz zabezpieczenia przed wyładowaniami ESD, realizowanymi zazwyczaj na bazie elementów dyskretnych. Zastępuje do 5 kondensatorów, rezystorów i cewek oraz dwa komponenty zabezpieczające, ograniczając wymaganą powierzchnię płytki drukowanej. Producent podaje na kartce katalogowej optymalny sposób prowadzenia ścieżek na płycie, zapewniający osiągnięcie optymalnych parametrów układu.

Układ BPF8089-01SC6 może współpracować bezpośrednio ze wzmacniaczami niskoszumowymi STA8089 i STA8090 firmy STMicroelectronics. Zapewnia zgodność ze standardem IEC 61000-4-2 (C=150 pF, R=330 Ω) w zakresie odporności na wyładowania ESD. Jest odporny na wyładowania do 4 kV przenoszone przez kontakt i do 15 kV przenoszone przez powietrze. Ponadto zapewnia odporność na impulsy napięciowe do 2 kV zgodnie z wymogami standardu militarnego MIL STD 883C (C=100 pF, R=1,5 k Ω).

Układ jest zamykany w obudowie SOT23-6L. Jego ceny hurtowe zaczynają się od 0,198 USD przy zamówieniach 1000 sztuk.

www.st.com



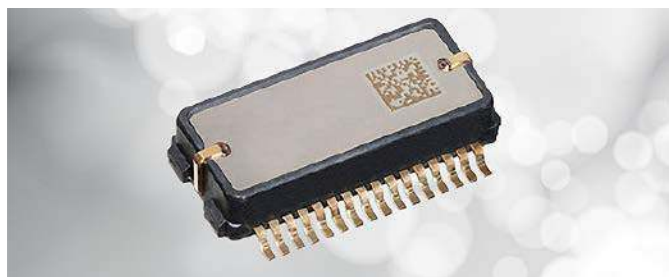
Czujniki różnicy ciśnień z obsługą do 7 zakresów pomiarowych konfigurowanych przez użytkownika

Superior Sensor Technology prezentuje pierwsze w branży czujniki różnicy ciśnień do zastosowań przemysłowych, obsługujące do 7 zakresów konfigurowanych przez użytkownika. Mogą one znaleźć wiele zastosowań, umożliwiając pomiar ciśnienia już od 62,5 Pa/0,25"

H₂O do 345 mb/5 psi przy dokładności 0,05% i całkowitym błędzie pomiaru (TEB) poniżej 0,15% FS. Są skompensowane termicznie w zakresie temperatury otoczenia od -20 do +85°C. W połączeniu z wbudowanym filtrem środkowozaporowym 50/60 Hz i filtrem dolnoprzepustowym (1...200 Hz) pozwalają uprościć projektowanie, skrócić czas wprowadzania produktów na rynek i zmniejszyć koszty inwentaryzacji. Nadają się idealnie do pomiaru ciśnienia suchego powietrza i nieagresywnych gazów, zapewniając bardzo dużą dokładność, 16-bitową rozdzielczość i stabilny punkt zerowy. Ich zakres zastosowań obejmuje aplikacje produkcyjne (zautomatyzowane systemy bezpieczeństwa, monitorowanie procesów chemicznych, testowanie wycieków), badawczo-rozwojowe (kalibracja urządzeń, akwizycja danych, zliczanie cząstek) i komercyjne (monitorowanie ciśnienia w kabinach, monitorowanie sal szpitalnych, drony).

Czujniki serii ND oraz płytki ewaluacyjne są dostępne w ilościach hurtowych m.in. w ofercie Digi-Key i Mouser Electronics.

www.superiorsensors.com



6-osiowy czujnik inercyjny MEMS do precyzyjnej kontroli maszyn przemysłowych

Nowy 6-osiowy czujnik inercyjny SCHA63T firmy Murata umożliwia precyzyjną kontrolę maszyn przemysłowych oraz stanowi wsparcie modułów pozycjonowania GNSS m.in. w robotach i pojazdach UAV. Jest czujnikiem MEMS wysokiej klasy, zapewniającym stabilność żyroskopu wynoszącą 1°/h, gęstość szumu 0,0015°/s/√Hz oraz błąd ortogonalności <0,14° po przeprowadzeniu kalibracji. Zawiera konfigurowalny filtr dolnoprzepustowy o częstotliwości odcięcia 13, 20, 46 lub 300 Hz oraz rozbudowany obwód auto-diagnostyki, pozwalający na zastosowania w aplikacjach krytycznych pod względem bezpieczeństwa. Komunikuje się z mikroprocesorem za pomocą interfejsu SPI.

Pozostałe cechy:

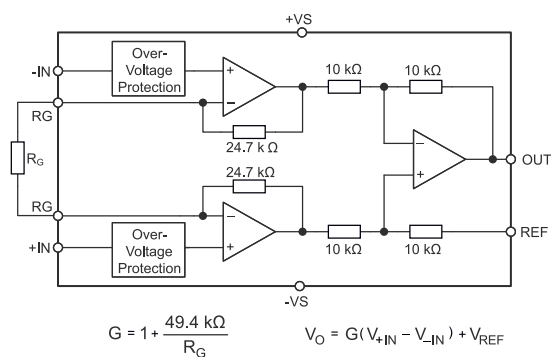
- zakres pomiaru szybkości kątowej: ±300°/s,
- zakres pomiaru przyspieszenia: ±6 g,
- zakres temperatury pracy: -40...+110°C,
- napięcie zasilania: 3,0...3,6 V,
- obudowa: SOIC-32 (19,7×12,2×4,6 mm),
- stabilny offset i czułość w całym zakresie dopuszczalnej temperatury pracy,
- bardzo dobra liniowość i duża odporność na wibracje.

www.murata.com

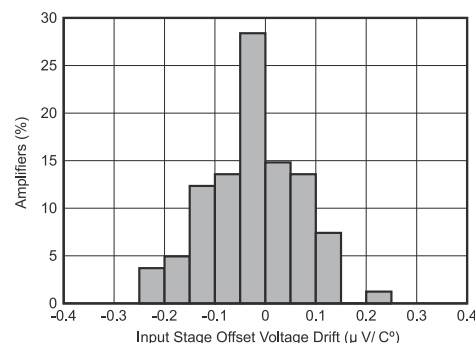
Precyzyjny wzmacniacz pomiarowy

Wzmacniacz pomiarowy INA821 firmy Texas Instruments charakteryzuje się małym poborem mocy i możliwością pracy w szerokim zakresie asymetrycznych i symetrycznych napięć zasilania 4,5...36 V i ±2,25...±18 V. Jego wzmacnienie może być programowane w szerokim zakresie od 1 do 10000 V/V za pomocą pojedynczego rezystora. Dzięki zastosowaniu w stopniu wejściowym tranzystorów super-beta układ charakteryzuje się małym wejściowym napięciem niezrównoważenia (typ. 10 μV, maks. 35 μV), małym dryftem temperaturowym i małym wejściowym prądem polaryzacji. Jest zabezpieczony przed przepięciami do ±40 V.

INA821 Simplified Internal Schematic



Typical Distribution of Input Stage Offset Voltage Drift



Układ INA821 został zoptymalizowany pod kątem zapewnienia skutecznego tłumienia składowej sumacyjnej. Przy wzmacnieniu równym 1 V/V współczynnik CMRR wynosi minimum 92 dB w całym zakresie dopuszczalnych napięć sumacyjnych. Układ jest produkowany w trzech typach obudów 8-wyprowadzeniowych: SOIC, VSSOP i WSON. Został przystosowany do pracy w szerokim zakresie temperatury otoczenia od -40 do +125°C.

Pozostałe parametry:

- dryft wzmacnienia: maks. 5 ppm/°C (G=1); 35 ppm/°C (G>1),
- gęstość wejściowego napięcia szumu: 7 nV/√Hz,
- pasmo: 4,7 MHz (G=1); 290 kHz (G=100),
- PSRR: min. 110 dB (G = 1),
- pobór prądu zasilania: maks. 650 μA,
- stabilność z obciążeniem pojemnościowym 1 nF.

www.ti.com

REKLAMA

ZAJRZYJ NA TE STRONY

 www.renex.com.pl	 www.ipctraining.pl	 www.reneline.pl
----------------------	------------------------	---------------------

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

www.gamma.pl
 PODZESPOŁY ELEKTRONICZNE

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.



Tracker GPS z LoRaWAN na platformie Dragino

W ramach opisywanego projektu zastosowano bramkę LoRaWAN firmy Dragino. Do bramki dołączone zostały różne moduły, które działają z serwerem TheThingsNetwork oraz dwa trackery GPS, które pozwalają na śledzenie pozycji poszczególnych sensorów. Najpierw do bramki podłączany jest węzeł GPS zbudowany z urzyciem Arduino, co pozwala na współdzielenie danych GPS. Następnie do modułu podłączany jest gotowy węzeł do śledzenia GPS LGT92 firmy Dragino i również z niego zbierane są dane dotyczące pozycji.

W systemie zastosowano nową bramkę Dragino LPS8. Jest to wewnętrzna bramka LoRaWAN, 8-kanalowa i z otwartymi źródłami (open source). W przeciwieństwie do bramki LG01-P można do niej podłączyć więcej węzłów i z łatwością obsłużyć stosunkowo większy ruch w sieci LoRa. Bramka LPS8 jest wyposażona w jeden koncentrator SX1308 LoRa i dwa transceivery SX1257 LoRa. Zawiera port USB i wejście zasilania USB typu C. Oprócz tego ma również port Ethernet, który można wykorzystać do połączenia z siecią LoRa. Można komunikować się z nim również poprzez Wi-Fi.

LPS8 używa technologii Semtech Packet i jest w pełni kompatybilna z protokołem LoRaWAN. Koncentrator LoRa zawarty w tej bramce zapewnia 10 programowalnych równoległych ścieżek demodulacji. Zawiera wstępnie skonfigurowane standardowe pasma częstotliwości LoRaWAN do użytku w różnych krajach. Niektóre z zalet bramki LPS8 LoRaWAN to:

- w pełni otwarta konstrukcja, oparta na systemie operacyjnym OpenWrt,
- emulacja do 49 demodulatorów LoRa,
- ma dziesięć równoległych, programowanych ścieżek demodulacji LoRa.

<https://bit.ly/2QLJ0BY>



Mobilna platforma Flightaware

„Chcę zbudować mobilny odbiornik PiAware ADS-B. Taki, który będzie działał, gdy poruszam się po różnych miejscach. Powinien mieć wbudowany GPS, dostarczający lokalizację do FA. Chcę również, aby ostrzegał mnie (z kursem, wysokością, prędkością, identyfikatorem), gdy statek powietrzny wkracza we wcześniej zdefiniowaną przestrzeń powietrzną (powiedzmy w promieniu 5 km od mojej aktualnej lokalizacji). Dane powinny być nadal przesyłane normalnie do aplikacji FlightAware”, pisze Kim Shapley, autor tego mobilnego stanowiska do obsługi FlightAware.

FlightAware to aplikacja przeznaczona do darmowego śledzenia pozycji samolotów na niebie. Pozwala ona na obserwacje prywatnych i komercyjnych lotów dzięki wykorzystaniu danych z odbiorników systemu ADS-B, oferowanych przez FlightAware w 197 krajach (firma posiada 30 tysięcy takich odbiorników na całym świecie). ADS-B to technologia, która pozwala samolotom nadawać swoją pozycję i inne dane z sensorów w czasie lotu w czasie zbliżonym do rzeczywistego.

Opisywany system korzysta z komercyjnie dostępnych części i telefonu komórkowego z systemem iOS. Pozwala na monitorowanie lotów,

jakie odbywają się dookoła nas i odbieranie komunikatów ADS-B poprzez zastosowany moduł USB FlightAware. System korzysta ze smartfona (iPhone 6S) oraz komputera jednopłytkowego (Raspberry Pi Zero W) i jest zasilany z powerbanku. Raspberry Pi uzupełnione jest modulem Pi Zero USB HAT, co pozwala na podłączenie do komputera modułu FlightAware Pro+ i sprzętowego modułu GPS, gdyż nie było możliwe wykorzystanie w aplikacji GPS-u wbudowanego w smartfon. Do modułu FlightAware dołączona jest pojedyncza antena 1090 MHz, która pozwala na odbiór transmisji ADS-B.

Obecnie system działa bardzo dobrze z Pi Zero z aplikacją zbudowaną na standardowej wersji PiAware v4.0 na karcie Micro SD o pojemności 32 GB. System obsługiwany jest przez przeglądarkę na telefonie – Raspberry Pi działa jako HotSpot, co pozwala na kontrolę aplikacji poprzez Hostname.local:8080. Oczywiście system poprawnie śledzi samoloty, ale na tym jeszcze nie koniec. Autor planuje uzupełnienie systemu o zewnętrzny moduł GPS, dodanie wyłącznika do Pi Zero, wraz ze skryptem zamykającym system, a także zaprojektowanie i druk w 3D odpowiedniej obudowy.

<https://bit.ly/2QOJuaF>

Emulator konsoli N64 na Raspberry Pi

Autor konstrukcji często umieszcza komputer jednopłytkowy w miejscach, które nie zostały zaprojektowane dla tego rodzaju maszyny. W swoim najnowszym projekcie postanowił umieścić Raspberry Pi w konsoli N64 i zbudować nowoczesny emulator gier retro.

Głównym celem projektu jest uzyskanie czegoś prawie nie do odróżnienia od oryginalnego, fabrycznego N64. W tym celu autor planuje:

- użyć wszystkich czterech oryginalnych portów sterujących i sprawić, aby wszystkie cztery kontrolery działały poprawnie,



- użyć oryginalnego wyłącznika zasilania z bezpiecznym obwodem wyłączającym Raspberry Pi,
- użyć oryginalnego przełącznika do resetowania systemu,
- użyć oryginalnego zasilacza lub przynajmniej sprawić, aby wyglądał, jakby był to oryginalny zasilacz do starej konsoli.

Jeśli chodzi o elementy wewnętrzne, sercem emulatora jest Raspberry Pi Compute Module o następującej specyfikacji:

- SoC Broadcom BCM2837B0 z 64-bitowym rdzeniem Cortex-A53 (ARMv8) taktowanym zegarem 1,2 GHz,
- 1 GB LPDDR2 pamięci SDRAM,
- pamięć eMMC Flash o pojemności 8 GB/16 GB/32 GB lub bez pamięci eMMC Flash w wariantcie Lite.

Parametry te, w realistycznym ujęciu, nie sprawią, że wszystkie gry na N64 będą działały idealnie, ale zdaniem autora zapewni to zadowalające efekty.

Pierwszym poważnym problemem było zastosowanie oryginalnych złączy dla kontrolerów. Są one bardzo niestandardowe, co było problemem w lutowaniu ich na PCB. Na szczęście udało się to zrobić, dzięki

REKLAMA

COMPUTER CONTROLS

Autoryzowany dystrybutor Altium w Polsce



ALTium
DESIGNER 21



ALTium 365

NIE CZEKAJ bądź efektywny od zaraz

Najlepsze oferty tylko do 30 czerwca 2021

- Wykorzystaj w swojej pracy ujednolicone środowisko Altium Designer i Altium 365
- Nie zwlekaj i zdobądź Altium Designer w preferencyjnych cenach

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

czemu autor może przystąpić do testów i uruchamiania całego systemu. Warto obserwować ten projekt i postępy autora w uruchamianiu emulatora N64 na Raspberry Pi w obudowie oryginalnej konsoli. <https://bit.ly/2RJ8rom>



Micro CRT Pi

Noah Backus postanowił zastosować stary, przenośny telewizor kineskopowy w zupełnie nowej roli jako wyświetlacz do komputera jednopłytkowego Raspberry Pi. Jego mikrotelewizor może odbierać sygnały AM oraz FM, a także transmisje VHF i UHF. Możliwe jest także wewnętrzne podanie sygnału wideo na jedną z nóżek układu AN5151N, który kontroluje telewizor. Jest to możliwe dzięki wyjściu wideo, w które wyposażone jest Raspberry Pi.

<https://bit.ly/2TbBSj7>



Wyświetlacz kursu bitcoin na Raspberry Pi 3

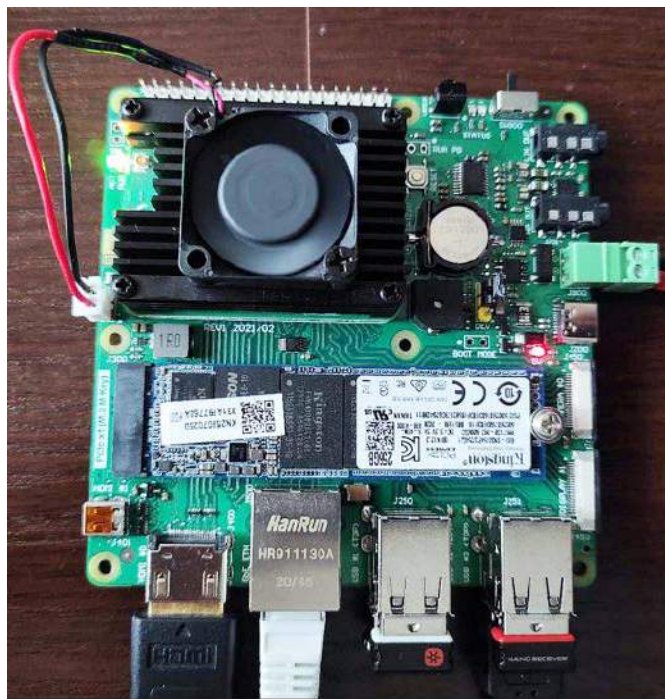
Ten projekt jest dokładnie tym, co opisuje jego nazwa. Ash Johnson, autor tej konstrukcji, postanowił zbudować prosty wyświetlacz kursu bitcoina. Za cały system odpowiedzialny jest komputer jednopłytkowy Raspberry Pi 3, który pobiera aktualny kurs bitcoina z sieci, korzystając z API *coindesk.com*. Do wyświetlania pobranych danych zastosowano ekran OLED ze sterownikiem SSD1305. Całość kontrolowana jest przez skrypt napisany w Pythonie, który uruchamiany jest na Raspberry Pi jako usługa pracująca w tle.

<https://bit.ly/3uh5JDS>

<https://bit.ly/2RNIxUC>

MirkoPC – płyta bazowa dla Raspberry Pi 4 Compute Module

MirkoPC to ciekawa płyta montażowa do modułu obliczeniowego Raspberry Pi 4. Przeznaczona jest do tworzenia zaawansowanych systemów wbudowanych dla różnych aplikacji.



Główne cechy platformy:

- pełna obsługa modułu Raspberry CM4,
- gniazdo M.2 2242/2280 M-Key (PCIe×1 Gen2 5 Gbps),
- Wi-Fi 2,4 GHz/5 GHz oraz Bluetooth 5.0 BLE (poprzez moduł CM4),
- slot na kartę microSD,
- dodatkowa pamięć eMMC 5.1 (standardowy układ FBGA-153),
- cztery porty USB 2.0 (działające w trybie hosta, złącza typu A),
- gigabitowy Ethernet na złączach RJ45 z obsługą IEEE1548,
- dwa porty HDMI 2.0 (4K przy 60 Hz),
- dwa interfejsy dla kamer (MIPI CSI),
- dwa interfejsy dla wyświetlacza (MIPI DSI),
- USB-C do zasilania (5 V DC z wydajnością prądową, co najmniej 3 A) i do uruchamiania – programowania firmware,
- złącze zasilania,
- wysokiej jakości przetwornik analogowo-cyfrowy (DAC) audio I²S (32-bit/384 kHz) ze wzmacniaczem słuchawkowym (DirectDrive, moc wyjściowa do 128 mW),
- zegar czasu rzeczywistego z podtrzymaniem baterijnym.



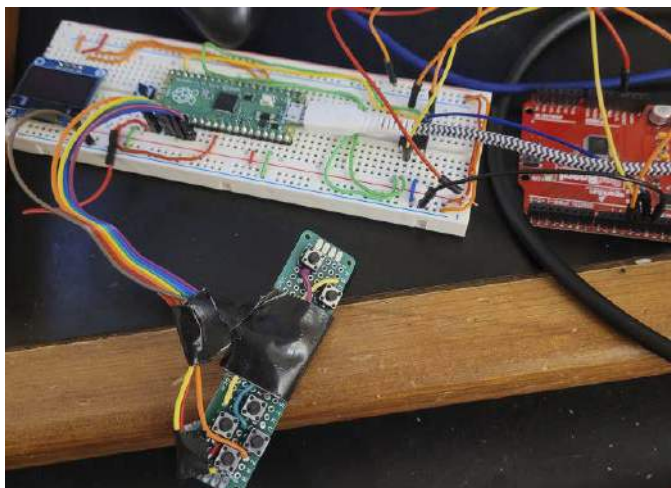
Konstrukcja oparta jest na czterowarstwowej płycie drukowanej o kontrolowanej impedancji. PCB ma wymiary 100×100 mm i pod koniec kwietnia zostało wypuszczone w wersji rev 1.1. Kompletna dokumentacja projektu dostępna jest w repozytorium autora na GitHubie. Informacje o postępach projektu znaleźć można z kolei na Twitterze, gdzie autor w miarę często stara się aktualizować status prac.

Autor systemu, Mirosław Folejewski, pisze w logu na portalu Hackaday.io, że najtrudniejszą rzeczą w tym projekcie była implementacja gniazda PCIe M.2. Na czas pisania tych słów nie było w pełni wiadomo, czy element ten będzie w 100% działał poprawnie, ale nadzieje są duże. Autor chciałby za pomocą interfejsu PCIe przetestować dołączanie do Raspberry Pi CM4 urządzeń takich jak dyski SSD NVMe czy moduł Coral Edge TPU.

Docelowe aplikacje systemu to, między innymi, platformy uczenia maszynowego, chmury obliczeniowe, serwery dyskowe (NAS), klient BOINC (sieci obliczeniowej), kontrolery automatyki domowej czy odtwarzacze multimedialne.

<https://bit.ly/3oV8JEU>

<https://bit.ly/2SkY36a>



Prosta konsola do gier wielkości karty kredytowej

Game-Card to konsola do gier o wymiarach typowej wizytówki. Składa się z wyświetlacza OLED, przycisków, Raspberry Pi Pico i baterii litowo-polimerowej o pojemności 2000 mAh. Gry są ładowane z zewnętrznej pamięci ROM i uruchamiane przez specjalny interpreter. Zarówno konsola, jak i układy ROM mają swoje płytki PCB, tak aby były możliwie kompaktowe.

Projekt ten ma dwa cele: system ma być mały i tani. Obecnie największym problemem wydaje się zasilanie. Akumulator i ładowarka kosztują więcej niż pozostałe komponenty razem wzięte (!), a także sprawiają, że system jest znacznie grubszy. Pierwszy prototyp bazował na ATmega328P (Arduino), ale od tego czasu projekt przeniósł się na znacznie mocniejszy Raspberry Pi Pico, ponieważ ten kosztuje tylko 2 dolary więcej, a oferuje znacznie większą wydajność obliczeniową.

Najwyższy poziom

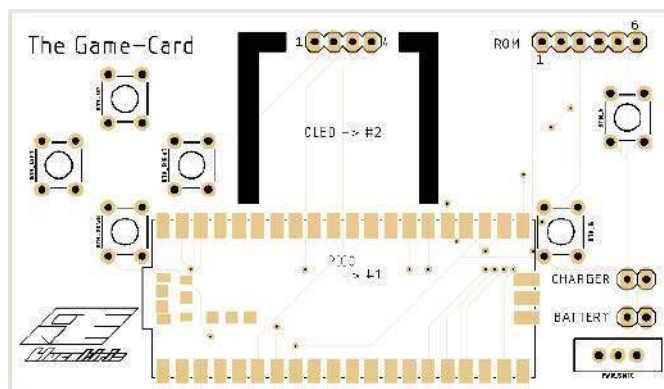
Na najwyższym poziomie abstrakcji, można napisać, że system pobiera dane wejściowe za pomocą przycisków, dostarcza dane wyjściowe do ekranu OLED, pobiera zasilanie przez złącze akumulatora i odbiera dane programu za pośrednictwem podłączonej pamięci ROM. Poniżej znaleźć można omówienie każdego z tych subsystemów bardziej szczegółowo.

System wejściowy

W oprogramowaniu jest klasa *Controller*, która jest interfejsem składającym się z zaledwie jednej funkcji *isPressed*, która przyjmuje wartość typu *ControllerInput*, która jest klasą wyliczeniową. To wyliczenie obecnie odpowiada liczbom całkowitym, które odnoszą się do pinów GPIO w Raspberry Pi Pico, chociaż nie trzeba tego robić w ten sposób. Istnieje implementacja interfejsu kontrolera o nazwie *ButtonController*, który jest oczywiście kontrolerem, który łączy się z fizycznymi przyciskami. Używają wewnętrznych oporników Pico, co pozwala uprościć PCB. Ta klasa wyodrębnia przyciski do postaci podającej, który przycisk jest wciśnięty, co jest przydatne w programowaniu.

Wyświetlacz

Podobnie jak klasa *Controller*, istnieje abstrakcyjny interfejs *Display*, który ma wiele funkcji. Układ sterownika wyświetlacza OLED, z jakiego korzysta system, to SSD1306. Istnieje klasa o takiej samej nazwie, która implementuje interfejs do warstwy abstrakcji dla tego wyświetlacza dołączonego do komputera poprzez I²C. Większość kodu specyficznego dla wyświetlacza OLED (zwłaszcza sekwencja startowa) została znaleziona i przekopiowana z różnych otwartych bibliotek Arduino. To, co można znaleźć w kodzie w repozytorium tego projektu, powstało jako integracja tych elementów.



System zasilania

Raspberry Pi Pico może być zasilane dowolnym napięciem od 3,3 V do 5 V, więc 3,7 V z ogniwa litowo-polimerowego doskonale nadaje się do bezpośredniego zasilania systemu. Dodatkowo, system zasilania konsoli ma przełącznik, który odłącza zasilanie od płytki i podłącza ją do ładowarki ogniwa Li-po.

Pamięć ROM

Obecny układ ROM to pamięć 25LC512, ale testowane były inne, podobne pamięci oraz pamięci SRAM, dlatego oprogramowanie do ich obsługi również jest zawarte w repozytorium. Interfejs dla pamięci ROM nazywa się RomChip. Ma on bardzo proste funkcje: odczyt, zapis i wyznaczenie rozmiaru. Obecny układ, 25LC512, to kontrolowana przez SPI pamięć EEPROM i właśnie tego używa Pico do komunikacji z nim.

Interpreter

Jest ich dużo, więc nie będą tutaj dokładnie opisane, ale jak zarzeka się autor, planuje on opublikować dokumentację ich wszystkich w repozytorium na GitHubie, więc będą one w pełni opisane.

Interpreter tworzy wątek wejścia/wyjścia przy użyciu funkcji wielordzeniowej Pico. Pobiera on odniesienia do interfejsów, dzięki czemu można podłączyć i grać w różne implementacje gier. Główny wątek odczytuje całą pamięć ROM (512 kb = 64 kB), ponieważ Pico ma 264 kB pamięci RAM. Następnie przechodzi przez każdą instrukcję, aż dotrze do końca programu. Następnie czyści ekran i kończy wykonywanie, chociaż zasadniczo większość gier tak naprawdę nie kończy swojego wykonywania tak szybko. W wątku aktualizacji, gdy otrzymana zostanie instrukcja z innego wątku, drugi rdzeń zaktualizuje rejestr z danymi wejściowymi z kontrolera, zaktualizuje ekran itd.

<https://bit.ly/3viVEHS>

REKLAMA

BORNICO Teraz większe MOŻLIWOŚCI

bornico.com.pl

- montaż kontraktowy elektroniki
- projektowanie urządzeń i systemów

Zakład Elektroniczny BORNICO

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





Ramka do obrazków z Discorda

Autor projektu (Frederik Hansen) założył serwer na Discordzie – aplikacji do wspólnych, grupowych czatów itp., który współdzieli ze znajomymi. Jako że jest to doskonała platforma do rozmów, wszyscy chętnie z niej korzystają. Pozwala również na przesyłanie obrazków, przez co stała się doskonałą platformą do wymiany memów.

Z tego też powodu – aby memy nie leżały po prostu w historii czasu, Hansen opracował cyfrową ramkę do zdjęć, która wyświetla obrazki przesyłane na czacie.

Jeśli chodzi o część sprzętową, to całość składa się z panelu LCD o rozdzielczości 720 p pozyskanego ze starego chromebooka i Raspberry Pi. Jako obudowa służy zwykła ramka do zdjęć (Ribba z IKEA). Dzięki grubości, wszystkie komponenty udało się zmieścić w jej wnętrzu.

Oprogramowanie na Raspberry Pi napisane jest w nodeJS. Komunikuje się ono z botem na kanale Discorda i wyświetla znajdujące się tam obrazki. Całość konfigurowana jest przez stronę web, na którą można wejść np. za pomocą smartfona. Kod programu można znaleźć na repozytorium autora na GitHubie. Autor udostępnia też pełny obraz karty SD z Raspbianem, skonfigurowanym do pracy w tym systemie.

<https://bit.ly/3oORmoU>

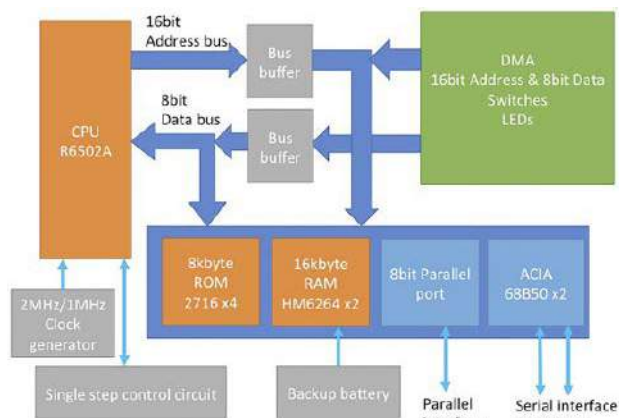
<https://bit.ly/3uIsKW9>



PERSEUS-8 – domowy komputer z układem 6502 DIY

Autorem tej przepięknej konstrukcji jest Mitsuru Yamada. W zeszłym roku zestawiał on samodzielnie PERSEUS-7 – komputer w minimalnej konfiguracji z procesorem 6502. Można było na nim uruchamiać oprogramowanie, jednakże było ono ograniczone do 1 kB, a autor potrzebował więcej, aby możliwe było uruchomienie bardziej złożonych programów. I tak narodził się PERSEUS-8. Główne cechy charakterystyczne nowej maszyny to:

- ustawienie adresu DMA o szerokości 16 bitów dla całej przestrzeni procesora 6502;



- zaimplementowanie 16 kB pamięci RAM z podtrzymaniem baterijnym, dzięki czemu oprogramowanie jest zawsze zapisywane, z tego 8 kB można przełączać za pomocą PROM;
- zawiera jednostopniowy obwód sterowania – ten sam sprawdzony obwód, co PERSEUS-7;
- obudowa komputera ma prawie taki sam wymiar jak starsza konstrukcja PERSEUS-3, ale obie strony płyty głównej są dostępne.

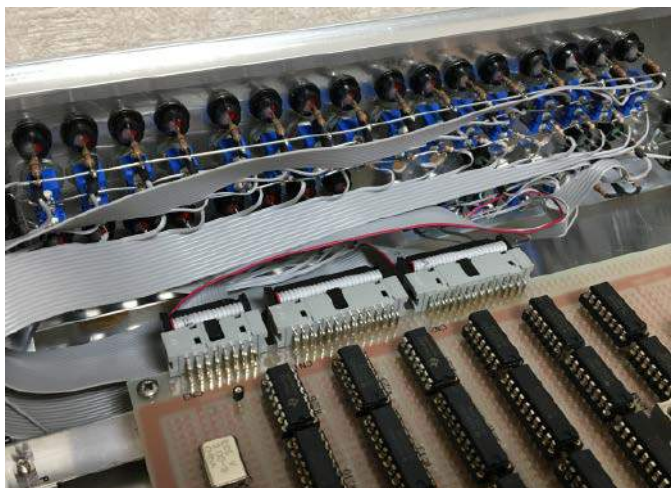
Ponieważ komputer ten korzysta z procesora, który ma około 45 lat, nie może w ogóle konkurować z dzisiejszymi konstrukcjami pod względem wydajności. Autor chciał jednak, aby PERSEUS-8 miał funkcje i wygląd, które przypominałyby koncepcje utracone przez dzisiejsze komputery, stąd też diody LED do wyświetlania wartości zmiennych i przełączniki do zadawania ich wartości i obsługi systemu.

Obudowa PERSEUS-8 jest zrobiona z aluminium. Do konstrukcji wykorzystano blachę o grubości 2 mm i kątowniki 15 mm i 10 mm. Jedynie dolna płyta jest wykonana z aluminium o grubości 1 mm. Do zamocowania każdej części użyto śrub i nakrętek M3. Ta metoda jest dość żmudna, ale pozwala na konstrukcję dowolnej obudowy. Wymiary zewnętrzne (bez uchwytów i gumowych nóżek) to 300×88×310 mm. Panel przedni został zmatowiony papierem ściernym, aby uzyskać matowe wykończenie, a panele boczne i górne pomalowano szarym lakierem w sprayu.

Rysunek powyżej przedstawia schemat blokowy sprzętu. Gdy procesor jest ustawiony w stan zatrzymania przełącznikiem na panelu przednim, adres pamięci i dane można ustawić za pomocą bezpośredniego dostępu do pamięci (DMA) przełącznikami bistabilnymi na panelu przednim, które służą do programowania.

Procesor to układ R6502A – taki sam jak w PERSEUS-7, ale taktowany z najwyższą możliwą częstotliwością taktowania, równą 2 MHz. Układ ma dwie 64 kb pamięci SRAM (HM6264BLSP-10L). Całkowita pojemność pamięci to 16 kB. Spośród nich 8 kB zostało zmapowanych na stronę zerowego obszaru strony i 8 kB na stronę obszaru wektora. Pamięci SRAM są zasilane baterią Ni-MH. Prąd zasilania tego układu w stanie podtrzymania wynosi tylko 0,56 µA. Jak twierdzi autor, pozwala to na podtrzymanie zapisu danych przez 10 lat. Przełącznik ochrony pamięci został tym razem zainstalowany na panelu przednim, ponieważ był trudny w użyciu na płycie głównej PERSEUS-3. W przypadku pamięci ROM można zamontować cztery 16 kb pamięci PROM typu 2716, które można zaprogramować za pomocą programatora PROM poza układem. ROM-y można wybierać między pamięcią RAM po stronie przestrzeni wektorowej za pomocą przełącznika na płycie głównej.

Szesnaście przełączników adresowych na panelu przednim jest podłączonych do magistrali adresowej przez bufony 3-stanowe (74HC244). Ośiem przełączników danych jest również podłączonych do magistrali danych za pośrednictwem buforów 3-stanowych (74HC244). Diody LED adresu wyświetlają adres wykonania CPU. W interfejsie szeregowym znajdują się diody LED sygnałów nadawczych i odbiorczych, dzięki czemu można bezpośrednio rozpoznać stan komunikacji.



Dokładny opis działania można znaleźć na stronie projektu na portalu Hackaday.io. Komputer ten może być wzbogacony o szereg urządzeń peryferyjnych. Ma dwukanałowy interfejs szeregowy RS-232C oraz 8-bitowy interfejs równoległy. Jako interfejs szeregowy używany jest *Asynchronous Communication Interface Adapter* (ACIA) zrealizowany na układzie MC68B50. Dla zegara szybkości transmisji użyto 76,92 kHz, co stanowi 1/13 zegara 1 MHz. 1/16 tej wartości jest przybliżeniem 4800 bps dla ACIA. Interfejs równoległy jest skonfigurowany za pomocą standardowej logiki 74LS374 i 74HC374.

Autor planuje niebawem opracować formalny program testowy i wgrać go do pamięci ROM jako oprogramowanie użytkowe. Za przykład daje np. test maszerujących wartości – jedną z najpopularniejszych metod testowania pamięci. Plany na przyszłość obejmują samodzielne stworzenie prostego interpretera z danymi zmiennoprzecinkowymi, jednakże jest to zadanie na co najmniej rok lub więcej

<https://bit.ly/3bR7Qbc>



The Gauntlet – klawiatura w postaci rękawiczki z przewodzącymi polami sterowana gestami

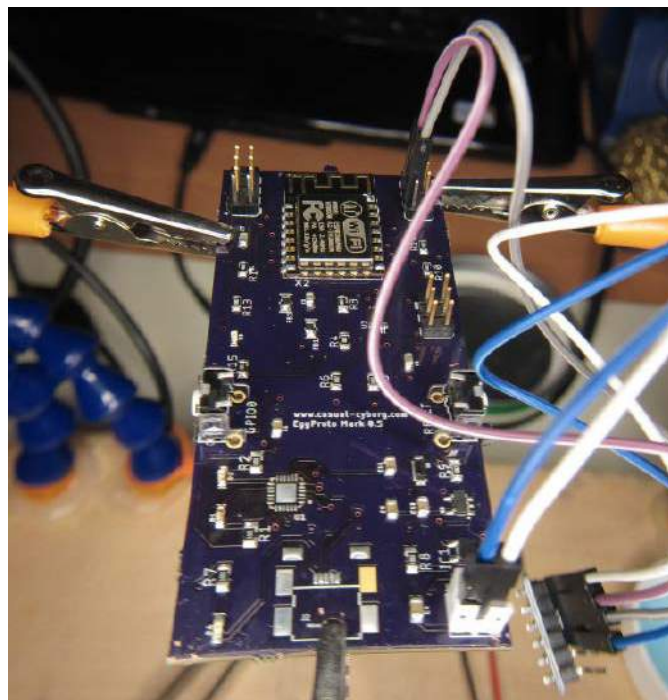
Zaprezentowany projekt to poręczna klawiatura, która używa palców jako klawiszy! Przewodzące punkty styku wzdłuż czterech palców pozwalają na układanie palców w różne kombinacje, które są przechwytywane przez mikrokontroler i przekształcane na wyjście sterujące klawiatury. Celem było zbudowanie kompletnego komputera do noszenia – aby wprowadzanie danych było znacznie łatwiejsze, konieczna była inna niż standardowa klawiatura. Tego rodzaju system

nadaje się do systemów rzeczywistości wirtualnej, rzeczywistości rozszerzonej, smartfonów, a także normalnego korzystania z komputera.

Arduino odczytuje siedem pinów i sprawdza, czy któryś z nich nie jest w stanie wysokim. Jeśli pin wchodzi w stan wysoki, zapisuje ten stan w tablicy boolowskiej i nie wyprowadza tablicy, dopóki wszystkie piny nie będą znowu w stanie niskim. Następnie mikrokontroler tłumaczy tablicę na wartość dziesiętną, która będzie indeksem kodu znaku, który wyprowadzi na urządzenie, do którego jest podłączona.

Siedem pinów odpowiada siedmiu przewodzącym polom na rękawicy, które można docisnąć do dwóch różnych padów (każdy pod napięciem 5 V), aby przyłożyć napięcie do poszczególnych pinów Arduino.

<https://bit.ly/34cRaXp>



BodiHub – płytka deweloperska dla systemów elektroniki noszonej

W przeciwieństwie do typowych systemów, urządzenia elektroniki noszonej stawiają wyższe wymagania. Człowiek dużo się rusza, obficie poci, skacze, kopie itp. Takie warunki powodują częste zrywanie połączeń lutowanych, udary elementów, pęknięcia płytek i obwodów... To prowadzi do zahamowania rozwoju produktu, ponieważ połowę czasu prototypowania spędzić można na próbach ustalenia, co się połuzowało, zepsuło lub wypadło.

Aby rozwiązać te problemy, uczestnik portalu Hackaday.io Casual Cyborg stworzył Bodihub: Inteligentną płytkę do rozwoju prototypów noszonych elementów Internetu Rzeczy, która pozwala na uczenie maszynowe. Łączy ona w sobie nadający się do noszenia moduł IoT, który ma wsparcie dla sieci neuronowej, z interfejsem chmurowym, który obsługuje back-endowe uczenie maszynowe. Ten projekt jest ciągłą ewolucją płytki deweloperskiej A Wearable IoT+ANN do pomiarów ciała.

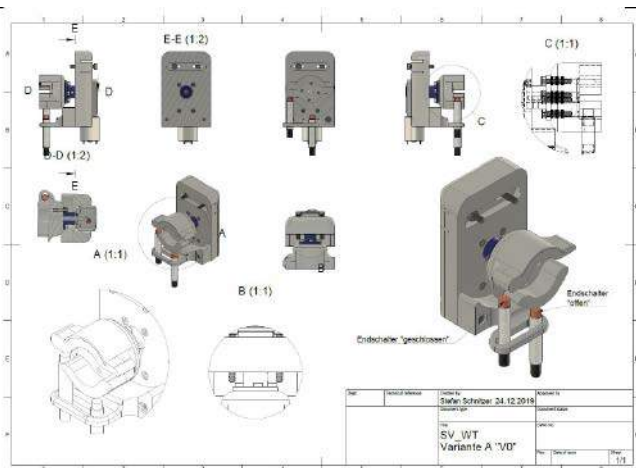
„Szukałem nadającej się do noszenia platformy programistycznej do systemów IoT do moich potrzeb i nie znalazłem tego, czego szukałem. Jasne, istnieje kilka modułów do noszenia. Niektóre były wykorzystane w jednym projekcie z żyroskopami i akcelerometrami do zegarków... ale bez portów analogowych. Niektóre były wyposażone w Bluetooth, ale nie obsługują Wi-Fi (...). Jeden jest przyszywany – nie chcę szyc i miałem dużo problemów z sygnałem w obwodach tego rodzaju. Żaden nie miał wszystkich funkcji wyprowadzonych. Dodatkowo większość z nich nie korzysta z ekosystemu Arduino,



który moim zdaniem jest najbardziej przyjazny dla użytkownika, wspierany przez społeczność i bogaty w sprzęt peryferyjny”, pisze autor. Dlatego też skonstruował własny moduł deweloperski. Oto lista funkcji, które zawiera pierwsza wersja jego płytki:

- Płytką oparta jest na chipie ESP8266 zgodnym ze standardem 802.11b/g/n. SoC ten wyrócił świat do góry nogami, kiedy pojawił się na rynku, ponieważ był to programowalny mikrokontroler z Wi-Fi, dostępny w niskiej cenie. Obecnie w układzie znajduje się ESP-12F, który ma 36 kB pamięci SRAM i 4 MB zewnętrznej pamięci Flash do przechowywania programów użytkownika. Jest to 32-bitowy mikrokontroler taktowany zegarem 80 MHz (max. do 160 MHz). Najważniejszą cechą układu jest kompatybilność z przyjaznym dla użytkownika ekosystemem Arduino. Daje to dostęp do rozległej społeczności, ogromnej biblioteki programistycznej i niezliczonych urządzeń peryferyjnych;
- Moduł zawiera złącza z pinami przeznaczonymi do I²C i SPI. Grupy złączy mają własne styki zasilania i masy, dzięki czemu można scentralizować system zasilania na płytce. Aby skorzystać z tej funkcji, wystarczy dołączyć standardową bibliotekę Arduino *Wire.h*. Piny te pasują do ośmiopinowych kabli taśmowych, które można łatwo nabyć na rynku;
- Na płytce znajduje się układ ADC, który zwiększa liczbę kanałów ADC do czterech i ma rozdzielczość aż 12 bitów! Piny wejścia analogowego zawierają własne styki zasilania. Daje to dostęp do szerokiej gamy czujników analogowych;
- Pokładowy system zasilania stabilizuje napięcie i ładuje baterię przez USB. Można użyć dowolnej baterii litowo-jonowej. Oszczędza to masę i rozmiar, jednocześnie zwiększając ilość energii dostępnej dla układu. Ogniwo podłączane jest przez 2-pinowe złącze JST;
- Moduł ma rozmiar zaledwie jednej trzeciej karty kredytowej. To sprawia, że jest mały, lekki i łatwy do przymocowania do ubrania, lub innego miejsca, które wymaga niewielkich rozmiarów;
- Układ ma standardowe złącze USB Micro B, służące do zasilania i komunikacji poprzez wbudowany kontroler USB do UART. Co można zrobić z tymi możliwościami? Oto mała lista projektów tworzonych za pomocą tej płyty deweloperskiej przez jej autora:
- Rękawice bokserskie, które rozmawiają ze sobą, gdy rozbijasz worek treningowy lub czyjąś twarz i informują, ile siły użyłeś;
- Zsynchronizowane narzędzia do synchronizacji rąk dla performerów, korzystających z tac ognia itp.;
- System śledzenia logistyki sieci kratowej, który wykorzystuje interfejsy rzeczywistości rozszerzonej;
- Nadający się do noszenia, syntetyczny „umysł kręgosłupa”, który analizuje ruchy ciała za pomocą pokładowej sieci neuronowej i przekazuje te dane do analizy do zaplecza uczenia maszynowego. Jest to znacznie lepszy interfejs komputera ludzkiego niż obecne systemy sprzężenia zwrotnego i sterowania.

<https://bit.ly/3oPvKco>



Zdalnie sterowany sterownik mechanicznego zaworu wody na Arduino

„Ze względów bezpieczeństwa chciałem mieć możliwość włączania i wyłączania wody w moim domu, ale mieszkam w wynajętym miejscu, w którym nie wolno modyfikować głównych rur i zaworów wodociągowych”, opisuje motywację do stworzenia układu Stefan Schnitzer. Dlatego też stworzył on ciekawy sterownik zaworu kulowego z kilkoma inteligentnymi funkcjami, który nie wymaga modyfikowania samego zaworu.

Pierwszym problemem była konstrukcja mechanicznej części projektu. Rozwiązaniem była drukowana w 3D złączka, która jest umieszczana bezpośrednio na normalnych uchwytach zaworów. Dwa zawory kulowe napędzane są tanimi serwosilnikami, a pozycje otwarcia i zamknięcia są monitorowane przez indukcyjne czujniki zbliżeniowe. Aby zamontować dwa (jeden na zimną i jeden na gorącą wodę) z tych urządzeń w systemie, użyto kilku więcej drukowanych w 3D części i trochę aluminium.

Ten prosty montaż pozwolił na zamontowanie elementów bez wiercenia otworów w skrzynce wodociągowej. Oprócz sterowników zaworów zamontowane są trzy czujniki temperatury DS18B20. Kontroler oparty jest na module Arduino, zamkniętym w wodoodpornej obudowie. Do samej płytki z mikrokontrolerem podłączono kontroler silnika w postaci mostka H, a w układzie znajdziemy także dwie przetwornice typu buck do ich zasilania.

Cała logika sterowania przekazana jest do głównego kontrolera inteligentnego domu. W ten sposób łatwo było zwizualizować stan zaworów i dodać kilka inteligentnych funkcji. Na przykład użyj czujników temperatury (różnica temperatur względem otoczenia), aby określić przepływ wody w rurach. W połączeniu z innymi funkcjami inteligentny dom wie, kiedy woda jest potrzebna, a kiedy nie. W przypadku wykrycia przepływu, gdy nie powinno być przepływu, zawory zamykają się.

<https://bit.ly/3hVZy5t>

Nikodem Czechowski, EP



Chcesz czytać nasze najnowsze artykuły jeszcze przed wydrukowaniem w EP? Zajrzyj na

www.ep.com.pl/EPwtoku





Spectra – analizator widma sygnału audio

Matematyka jest królową wszystkich nauk. Z tym cytatem spotkał się prawie każdy i nie sposób się z nim nie zgodzić, głównie wtedy, gdy zajmujemy się szeroko rozumianą elektroniką. Również i ja przekonałem się o tym wielokrotnie, zwłaszcza od czasu, gdy zacząłem zajmować się zagadnieniami z zakresu DSP, a zacząłem się nimi zajmować po natrafieniu na bardzo interesujący artykuł autorstwa Łukasza Podkalickiego (EP 12/2019) prezentujący arcyciekawe zagadnienia z zakresu DSP w odniesieniu do transformacji Fouriera w ujęciu realizacji na prostych, 8-bitowych mikrokontrolerach o ograniczonej mocy obliczeniowej i niewielkiej ilości pamięci RAM.

We wspomnianym artykule autor pokazuje, jak przy użyciu arytmetyki stałooprzecinkowej i współczynników wektora rotującego (tzw. *twiddle factors*) w prosty sposób jesteśmy w stanie wykonać dyskretną transformację Fouriera (DFT) sygnału audio, otrzymując widmo jego mocy. Jakby tego było mało, autor prezentuje również praktyczną realizację 3-punktowego „kolorofonu” bazującego na niewielkim mikrokontrolerze firmy Atmel typu ATtiny13 i programowej realizacji transformaty DFT.

Posiłkując się tym unikalnym materiałem, postanowiłem odświeżyć swój wcześniejszy projekt analizatora widma sygnału akustycznego pod nazwą „Spectrum” (EP 9/2019), który cieszył się sporym zainteresowaniem wśród Czytelników, lecz tym razem z zastosowaniem wspomnianej wcześniejszej techniki DSP. Warto przypomnieć, że w budowie urządzenia zastosowałem bardzo ciekawy układ scalonego MSGEQ7. Integruje on 7 filtrów pasmowo-przepustowych o częstotliwościach środkowych 63 Hz, 160 Hz, 400 Hz, 1 kHz, 2,5 kHz, 6,25 kHz oraz 16 kHz, 7 detektorów wartości

szczytowej i analogowy multiplekser wyjściowy. Okazało się jednak, że układ MSGEQ7 jest peryferium trudno dostępnym, a nawet jeśli uda się go pozyskać (głównie za sprawą azjatyckiego portalu aukcyjnego), to naabyt często okazuje się, że tak pozyskane peryferia nie są do końca sprawne lub, co gorsza, są to niedziałające podróbki.

Biorąc to wszystko pod uwagę, postanowiłem zbudować urządzenie o zbliżonej funkcjonalności, lecz pozbawione wcześniejszych rozwiązań sprzętowych, a realizujące całe skomplikowane zagadnienie filtracji sygnałów na drodze programowej. Co więcej, chciałem, by w odróżnieniu od pierwowzoru nowo projektowane urządzenie zbudowane było wyłącznie z popularnych i łatwo dostępnych elementów przeznaczonych do montażu przewlekane, co w zamyśle miało ułatwić montaż układu nawet przez mało doświadczonych elektroników. Jako pewne rozszerzenie funkcjonalności urządzenia źródłowego w niniejszym projekcie dodałem jeszcze jedną liniijkę LED, przez co zwiększyłem liczbę analizowanych pasm częstotliwościowych do ośmiu. Tak oto narodził się projekt Spectra.

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

W ofercie AVT* AVT5866

Podstawowe parametry:

- zasilanie: 5 VDC, ok. 170 mA,
- maksymalna amplituda analogowego sygnału wejściowego: 2,5 V (5 Vpp),
- analizowane pasma częstotliwościowe: 200 Hz, 400 Hz, 1 kHz, 2 kHz, 4 kHz, 6 kHz, 8 kHz, 12 kHz,
- liczba sposobów prezentacji widma sygnału: 5.

Projekty pokrewne na www.media.avt.pl:

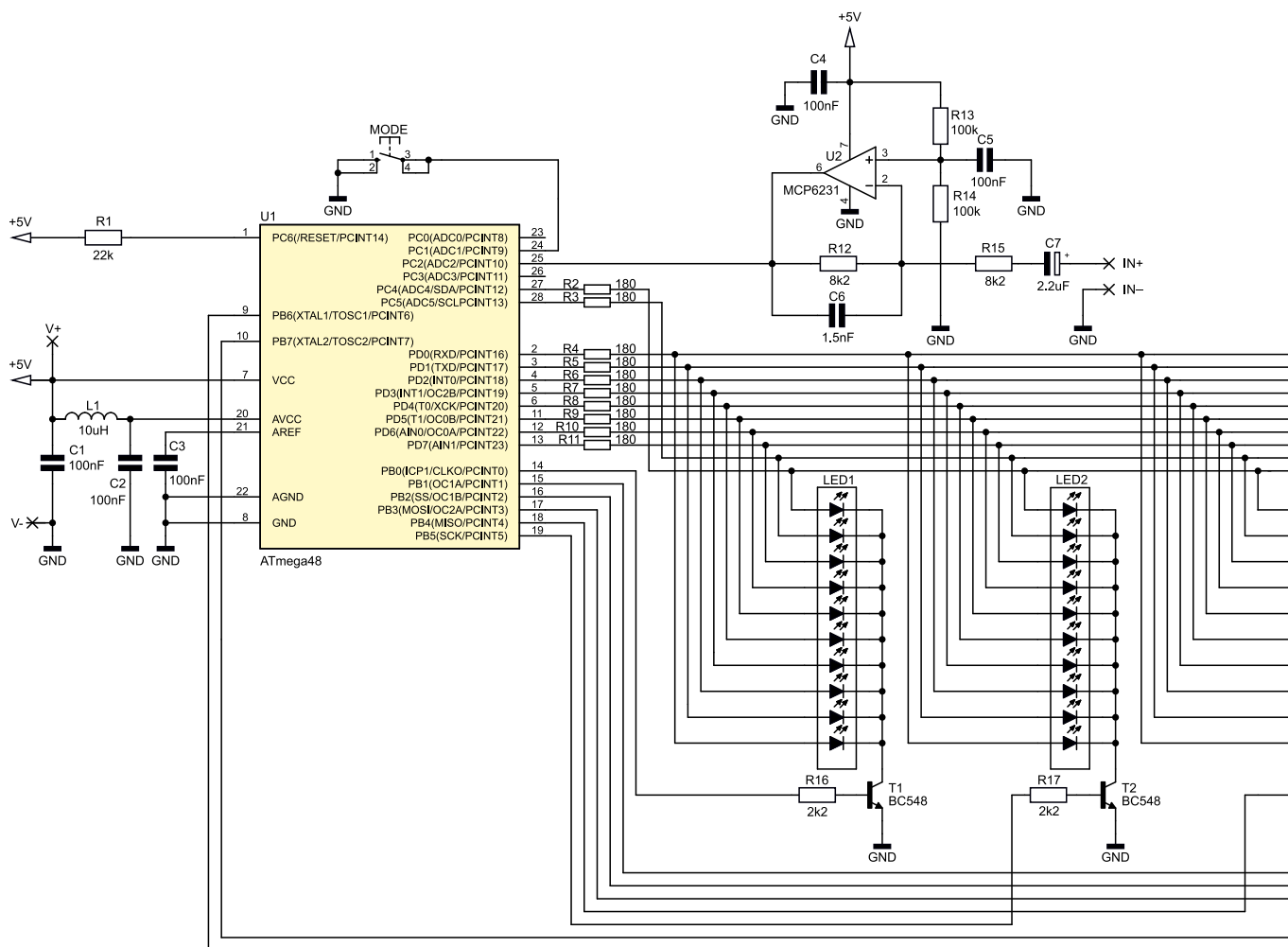
- AVT-5767 Stereofoniczny wskaźnik poziomuysterowania z funkcją Peak-Hold (EP 5/2020)
- AVT-5748 SpectrumDFT – analizator widma sygnału akustycznego (EP 3/2020)
- AVT-5712 Spectrum – prosty analizator widma sygnału akustycznego (EP 9/2019)
- AVT-5678 Stereofoniczny wskaźnikysterowania (EP 6/2019)
- AVT-5585 Sterownik wskaźnika wychyłowego do wzmacniacza (EP 1/2018)
- AVT-1716 Wskaźnikysterowania z pamięcią (EP 12/2012)
- AVT-1517 Wskaźnik nie tylkoysterowania (EP 9/2012)
- AVT-5219 Wizualizator do Winampa na USB (EP 1/2010)
- AVT-5210 Analizator widma sygnału audio (EP 11/2009)
- AVT-2864 Analogowo-cyfrowy analizator widma (Edw 5/2008)
- AVT-580 Procesor audio z equalizerem i analizatorem widma (EP 6-7/2004)
- AVT-2375 Wskaźnikysterowania 2x5 LED (Edw 9/1999)
- AVT-2353 Pseudoanalogowy VU-metr (Edw 4/1999)
- AVT-1190 Wskaźnikysterowania (EP 8/1998)

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 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 [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 [A+] – 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.



Rysunek 1. Schemat ideowy analizatora Spectra

Budowa i działanie

Schemat ideowy urządzenia został pokazany na **rysunku 1**. Jak widać, zaprojektowano bardzo prosty układ mikroprocesorowy, którego sercem jest niewielki mikrokontroler firmy Microchip (dawniej Atmel) o oznaczeniu ATmega48 taktowany wewnętrznym, wysokostabilnym sygnałem zegarowym o częstotliwości 8 MHz. Bardziej doświadczonych konstruktorów dziwić może fakt tak niskiej częstotliwości taktowania mikrokontrolera, biorąc pod uwagę rodzaj zadań, jakie będzie miał do wykonania (DSP). Uwierzyć mi na słowo, że jest to częstotliwość zupełnie wystarczająca. Przy okazji pozbywamy się dodatkowych elementów, które należałoby zastosować w celu zwiększenia częstotliwości taktowania poza wartości dostępne wewnątrz struktury naszego układu (mowa o zewnętrznym rezonatorze kwarcowym).

Głównym zadaniem naszego mikrokontrolera jest próbkowanie sygnału audio dostarczanego na wyprowadzenie ADC2 (PC2), co wykonywane jest przy użyciu wbudowanego przetwornika ADC pracującego w trybie 8-bitowym (tak naprawdę w trybie 10-bitowym, lecz z uwagi na wyrównanie wyniku konwersji do lewej łatwiej jest czytać 8-bitową wartość będącą wynikiem przetwarzania), zasilanego sygnałem zegarowym

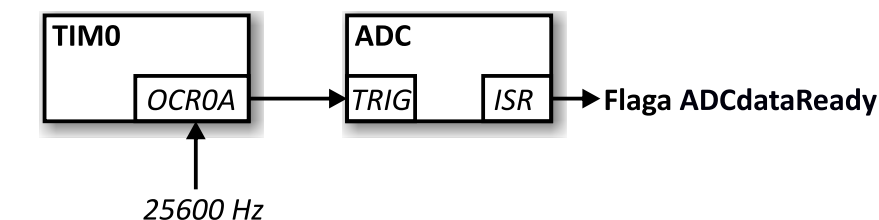
o częstotliwości 1 MHz, co zapewnia odpowiednio krótki czas przetwarzania sygnału (ok. 15,5 µs). Ten wymagany krótki czas konwersji wynika z dość dużej częstotliwości próbkowania wejściowego sygnału audio, a mianowicie z wartości 25600 próbek na sekundę. To z kolei wynika z pożądanego pasma przetwarzanego sygnału, które w tym przypadku, zgodnie z twierdzeniem Nyquista, wyniesie 12,8 kHz.

Zanim jednak sygnał audio trafi na wejście przetwornika ADC wbudowanego w strukturę mikrokontrolera, podlega filtracji w obwodzie prostego filtra dolnoprzestupowego pierwszego rzędu o częstotliwości granicznej około 12,8 kHz zbudowanego z użyciem wzmacniacza operacyjnego *rail-to-rail* pod postacią układu scalonego U2 (MCP6231) oraz kilku elementów biernych. Wzmocnienie w obwodzie filtra ustawiono na wartość 1, lecz można je skalować

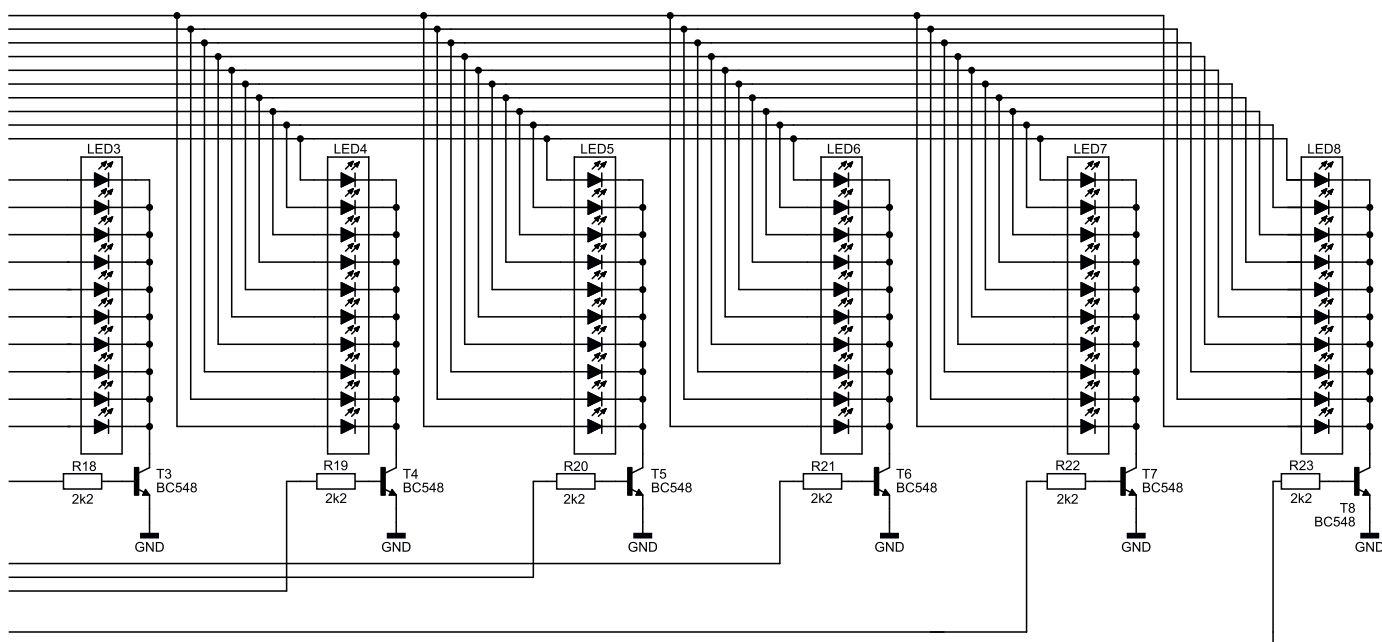
w górę, zmieniając stosunek rezystancji R12/R15, jednak cały czas pamiętając o częstotliwości granicznej wyznaczonej wzorem $f_c = 1 / (2 \cdot \pi \cdot R12 \cdot C6)$. Filtr w takiej konfiguracji odwraca fazę sygnału, lecz z uwagi na spolaryzowanie wejścia nieodwracającego wzmacniacza operacyjnego połowę napięcia zasilania oraz obecność kondensatora C7 to odwrócenie fazy nie ma znaczenia, jeśli chodzi o dalszą analizę DSP, gdyż sygnały przetwarzane przez przetwornik ADC znajdują się w zakresie napięć 0...5 V.

Dla porządku dodam, że wyzwalaczem konwersji wbudowanego przetwornika ADC jest układ czasowo-licznikowy Timer0 pracujący w trybie CTC, dla którego parametry dobrano w taki sposób, by porównanie zachodziło 25600 razy na sekundę.

Schemat blokowy systemu akwizycji danych urządzenia Spectra został pokazany na **rysunku 2**. W tym miejscu warto również



Rysunek 2. Schemat blokowy systemu akwizycji danych urządzenia Spectra



zaznaczyć, że producent mikrokontrolera, firma Microchip, podaje w stosownej dokumentacji dotyczącej przetwornika ADC dość niejednoznaczne, a przynajmniej niepełne informacje. Z jednej strony zaleca się, by częstotliwość taktowania przetwornika ADC nie była większa niż 200 kHz, zaś z drugiej podaje, że jeśli nie zależy nam na pełnej rozdzielczości peryferium (10 bitów), to częstotliwość tę możemy zwiększyć aż do 1 MHz (wartości, skądinąd wziętej z jeszcze innej części dokumentacji). Brak jednak jakichkolwiek wzorów czy informacji, w jaki sposób powiązać oczekiwaną rozdzielczość przetwornika z dopuszczalną częstotliwością taktowania tegoż peryferium. Na szczęście z pomocą przychodzi nam Internet i niezmiernie ciekawy, praktyczny artykuł umieszczony pod adresem <https://bit.ly/3o-WFoJZ>, gdzie empirycznie dowiedziono, jakie są limity tejże konstrukcji. Szczegółowo polecam lekturę tego artykułu, gdyż w sposób bardzo prosty tłumaczy zawile mechanizmy wpływające na proces akwizycji danych.

Wróćmy do schematu naszego urządzenia. Mikrokontroler realizuje (poza

procesem akwizycji danych i DSP) obsługę wyświetlacza LED zbudowanego z ośmiu 10-punktowych linijek LED połączonych w konfiguracji wspólnej katody oraz realizuje obsługę przycisku MODE przeznaczonego do zmiany sposobu prezentacji widma sygnału. Sterowanie pracą wyświetlacza LED jest wykonywane z użyciem mechanizmu multipleksowania, za który odpowiedzialny jest wbudowany w mikrokontroler układ czasowo-licznikowy Timer1 skonfigurowany w taki sposób, by generował przerwanie od porównania (tryb CTC układu) 480 razy na sekundę, czyli 60 razy na każdą z linijek LED. W przerwaniu takim w pierwszej kolejności wygaszane są wszystkie wspólne katody wyświetlacza LED (porty kolumn), następnie na porty PD0...PD7 oraz PC4 i PC5 (porty wierszy) wystawiana jest wartość odpowiadająca kolejnemu pasmu częstotliwościowemu. Następnie włączana jest kolejna wspólna katoda sterowana z portów PB0...PB7 mikrokontrolera (poprzez tranzystory sterujące T1...T8). Jest to typowe rozwiązanie mechanizmu multipleksowania stosowane w wielu systemach mikroprocesorowych, dzięki któremu możliwe jest sterowanie dużą liczbą diod LED przy ograniczonej liczbie portów mikrokontrolera. Dzięki temu ograniczamy przy okazji sumaryczny prąd pobierany

przez wyświetlacz LED, który w tym przypadku wynosi maksymalnie 10·16 mA. Tyle w kwestiach sprzętowych, w związku z czym przejdźmy do najważniejszych zagadnień programowych.

Program sterujący

Pora na odrobinę kodu związanego z systemem akwizycji danych. Kod odpowiedzialny za inicjalizację układu czasowo-licznikowego Timer0 będącego wyzwalaczem akwizycji oraz konfigurację przetwornika ADC pokazano na **listingu 1**. Dalej, na **listingu 2** pokazano kod odpowiedzialny za akwizycję danych przetwornika ADC. Jak widać, zebraniu kompletnej porcji danych towarzyszy zatrzymanie akwizycji danych (wyłączenie timera Timer0) oraz ustawienie flagi `ADCdataReady`, dzięki czemu możliwe jest przetworzenie danych przez program główny aplikacji.

Kilka słów uwagi wymaga zapis `ADCdata[Idx++] = ADCH - 128`. Odjęcie wartości 128 od wyniku przetwarzania przetwornika ADC (ADCH) wynika z faktu, że wejście ADC2 przetwornika spolaryzowano połową wartości napięcia zasilania, a więc jednocześnie napięcia referencyjnego po to, aby układ mógł przetwarzać rzeczywiste sygnały audio, które jak wiemy, przyjmują również wartości poniżej 0.

Ustawienia Fusebitów (ważniejszych):

```
CKSEL3..0: 0010
SUT1..0: 10
CKDIV8: 1
EEAVE: 1
```

Listing 1. Kod odpowiedzialny za inicjalizację akwizycji danych ADC

```
void initADC(void){
    //Konfiguracja Timera0 w trybie CTC, by zdarzenie porównania występowało 25600 razy na sekundę
    TCCR0A = (1<<WGM01); //Tryb CTC
    TCCR0B = (1<<CS01); //Preskaler = 8 (1MHz)
    OCR0A = 38; //Zdarzenie porównania 25600 razy na sekundę

    //Vref = AVcc (5V), wyrównanie wyniku do lewej (rozdzielczość 8-bitów), MUX = ADC2 (PC2)
    ADMUX = (1<<REFS0)|(1<<ADLAR)|(1<<MUX1);
    //Włączenie ADC, automatyczne wyzwalanie konwersji, zezwolenie na przerwanie ADC, Preskaler = 8 (1MHz)
    ADCSRA = (1<<ADEN)|(1<<ADSC)|(1<<ADIF)|(1<<ADPS1)|(1<<ADPS0);
    //Wybór źródła wyzwalania konwersji ADC: Timer/counter0 compare match A
    ADCSRB = (1<<ADTS1)|(1<<ADTS0);
    //Wyłączenie cyfrowych obwodów wejściowych na pinie ADC2 (PC2)
    DIDR0 = (1<<ADC2D);
}
```

Listing 2. Kod odpowiedzialny za akwizycję danych przetwornika ADC

```
ISR(ADC_vect){
    static uint8_t Idx;

    ADCdata[Idx++] = ADCH - 128;

    if(Idx == SAMPLES){
        Idx = 0;
        ADCdataReady = 1;
        STOP_ACQUISITION;
    }

    //Kasujemy flagę Timer0
    //Output Compare A Match
    //aby umożliwić wyzwalanie ADC
    // timerem
    TIFR0 = (1<<OCF0A);
}
```

Za przetworzenie zebranych danych, czyli obliczenie dyskretnej transformaty Fouriera z próbek sygnału zebranych w tablicy *ADCdata[]*, odpowiedzialna jest funkcja pokazana na **listingu 3**. Wprowadzono możliwość regulacji „czułości” funkcji DFT, a to z uwagi na przewagę tonów o niskich częstotliwościach w rzeczywistym sygnale audio oraz potrzebę zmniejszenia „czułości” funkcji w tym zakresie. Wynikiem działania funkcji *DFT()* jest obliczenie mocy poszczególnych prążków częstotliwości (specyfikowanych wartością argumentu *i*). I właśnie to zadanie stanowi główny problem obliczeniowy, o którym wspomniano na wstępie artykułu. Wynika to z liczby obliczeń stałoprzecinkowych wykonywanych w ramach pętli, z jakiej składa się wspomniana funkcja. Jak widać, liczba tych obliczeń zależy bezpośrednio od liczby punktów transformaty Fouriera (*SAMPLES*), która w naszym przypadku wynosi 128. Z kolei liczba punktów

Listing 3. Kod funkcji odpowiedzialnej za obliczenie dyskretnej transformaty Fouriera

```
#define MULF 64 //Multiplication factor, max 128

//Czas wykonania 1.45ms
uint16_t DFT(uint8_t i, uint8_t dftSensitivity){
    uint16_t a, b, Power;
    int32_t Re, Im;

    //Obliczamy moc sygnału dla szukanego
    //prążka częstotliwościowego i < (SAMPLES/2)+1

    Re = Im = a = 0;
    b = 3*SAMPLES/4;

    for (uint8_t j=0; j<SAMPLES; ++j){
        Re += (ADCdata[j] * Twiddle[a % SAMPLES])/MULF;
        Im -= (ADCdata[j] * Twiddle[b % SAMPLES])/MULF;
        a += i;
        b += i;
    }

    if(dftSensitivity == SENSIVITY_HI){
        Power = (Re*Re + Im*Im)/
            2048UL; else Power = (Re*Re + Im*Im)/
            16384UL;
    }

    return Power;
}
```

transformaty determinuje odległość kolejnych prążków mocy (tzw. *BIN*), a wynika z zależności: $BIN = \text{częstotliwość próbkowania} / \text{liczba punktów transformaty}$ (*SAMPLES*). Dla naszego przypadku $BIN = 200 \cdot (25,6 \text{ kHz} / 128)$, co oznacza, że kolejne wartości częstotliwości, dla których liczona jest moc sygnału, są wielokrotnością wartości 200 Hz. Wartość ta jest z kolei kompromisem pomiędzy rozdzielczością mocy (*BIN*) a czasem niezbędnym na wykonanie funkcji *DFT()* przy przyjętej liczbie punktów transformaty (*SAMPLES*). W naszym przypadku czas ten wynosi około 1,45 ms/8 (gdyż liczymy DFT wyłącznie dla wybranych 8 punktów). To determinuje częstotliwość odświeżania wykresu widma (tzw. *frame-rate*), która w tym przypadku wynosi około 80 Hz, a więc bardzo dużo, jak na nasze potrzeby. Dalsze zwiększanie liczby punktów transformaty (*SAMPLES*), choć pożądane, zmniejszyłoby częstotliwość odświeżania wykresu widma do wartości nieakceptowalnych i praktycznie nieużytecznych.

To jest główne ograniczenie software'owe naszej implementacji, o którym wspomniano wcześniej i wynika w głównej mierze z 8-bitowej architektury mikrokontrolera ATmega i maksymalnej, dostępnej częstotliwości taktowania przetwornika ADC (1 MHz). Zresztą nie bez powodu jako liczbę punktów transformaty wybrano wartość 128 będącą potęgą liczby 2. Wynikało to z faktu dzielenia, jakie wykonywane jest we wspomnianej wcześniej

pętli, a jak doskonale wiemy dzielenie przez liczbę będącą całkowitą potęgą liczby 2 jest o rząd wielkości szybsze, aniżeli dzielenie przez jakąkolwiek inną liczbę, gdyż tak naprawdę jest to zwykle przesuwanie wartości dzielonej w prawo.

Tyle w kwestii ograniczeń implementacji. Wróćmy zatem do naszej funkcji *DFT()*. Jak widać, i o czym powiedziano na wstępie artykułu, funkcja korzysta z tablicy *Twiddle[]* współczynników wektora rotującego, której wyznaczeniem zajmuje się funkcja *calculateTwiddleFactors()* pokazana na **listingu 4**. Oczywiście funkcja z listingu jest niejako nadmiarowa, gdyż współczynniki takie moglibyśmy wyznaczyć sobie w arkuszu kalkulacyjnym i zapisać na stałe w pamięci programu, co zmniejszyłoby zajętość pamięci RAM mikrokontrolera oraz kod obsługi aplikacji. Ja zdecydowałem się na wyznaczenie ich w trakcie działania programu, gdyż po pierwsze, jest to szybsze, zaś po drugie i nie mniej ważne, mikrokontroler nasz wyposażono w dużą ilość pamięci RAM, która jest nieużywana.

Warto podkreślić, że przed wykonaniem funkcji *DFT()* zebrana tablica danych wejściowych *ADCdata[]* poddawana jest okienkowaniu, które ma na celu ograniczenie tak zwanych wycieków widma sygnału, które samo w sobie wpływa niekorzystnie na wynikowe widmo sygnału. Zastosowana funkcja okna jest typu Hanna (Hanninga), zaś wyznaczeniem stosownych współczynników okna (*Window[]*) zajmuje się funkcja

Wykaz elementów:

Rezystory: (miniaturowe 1/4 W, raster 0,2")

R1: 22 kΩ

R2...R11: 180 Ω

R12, R15: 8,2 kΩ

R13, R14: 100 kΩ

R16...R23: 2,2 kΩ

Kondensatory:

C1...C5: 100 nF (raster 0,1")

C6: 1,5 nF (raster 0,1")

C7: 2,2 μF/10 V elektrolityczny (ø 4 mm, raster 2 mm)

Półprzewodniki:

LED1...LED8: linijka diodowa zielono-czerwona typu OSX10201-GGR1

T1...T8: BC548 (TO-92)

U1: ATmega48 (DIL-28)

U2: MCP6231 (DIL-08)

Pozostałe:

L1: dławik osiowy 10 μH typu DLA10-MN (miniaturowy, raster 0,2")

MODE: mikroswitch TACT 9 mm

Listing 4. Kod funkcji odpowiedzialnej za wyznaczenie współczynników wektora rotującego

```
#define PI2 6.2832 //2*Pi

void calculateTwiddleFactors(void){
    for(uint8_t i=0; i<SAMPLES; ++i)
        Twiddle[i] = (int8_t) (MULF*cos(i*PI2/SAMPLES));
}
```

Listing 5. Kod funkcji odpowiedzialnej za wyznaczenie współczynników okna Hanna (Hanninga)

```
void calculateWindowFactors(void){ //Okno Hanninga
    for(uint8_t i=0; i<SAMPLES; ++i)
        Window[i] = (int8_t) (MULF*(0.5-0.5*cos(i*PI2/(SAMPLES-1))));
}
```

Listing 7. Funkcja inicjalizacyjna mechanizmu multipleksowania

```
void initMultiplex(void){
    //Porty wierszy, jako porty wyjściowe ze stanem nieaktywnym "0"
    ROW_LSB_DDR = 0xFF;
    ROW_MSB_DDR |= (1<<ROW_MSB_1)|(1<<ROW_MSB_0);
    //Port kolumn, jako port wyjściowy ze stanem nieaktywnym "0"
    COLUMN_DDR = 0xFF;

    //Konfiguracja licznika Timer1 w celu generowania przerwania
    //do obsługi multipleksowania wyświetlacza LED (480 Hz)
    TCCR1B = (1<<WGM12)|(1<<CS12); //Tryb CTC, Preskaler = 256 @ 8MHz
    //480 Hz (przerwanie 480 razy na sekundę,
    //60 razy na sekundę dla każdego wyświetlacza LED)
    OCR1A = 64;
    //Uruchomienie przerwania Output Compare Match A (od porównania)
    TIMSK1 = (1<<OCIE1A);
}
```

calculateWindowFactors(), której ciało pokazano na **listingu 5**.

Spróbowany przebieg wejściowy zebrany w tablicy *ADCdata[]* jest przemnażany

przez funkcję okna *Window[]* przed wykonaniem transformaty Fouriera w ramach funkcji *DFT()*. Na sam koniec, po wykonaniu funkcji *DFT()*, a przed wyświetleniem

Listing 6. Plik nagłówkowy mechanizmu multipleksowania

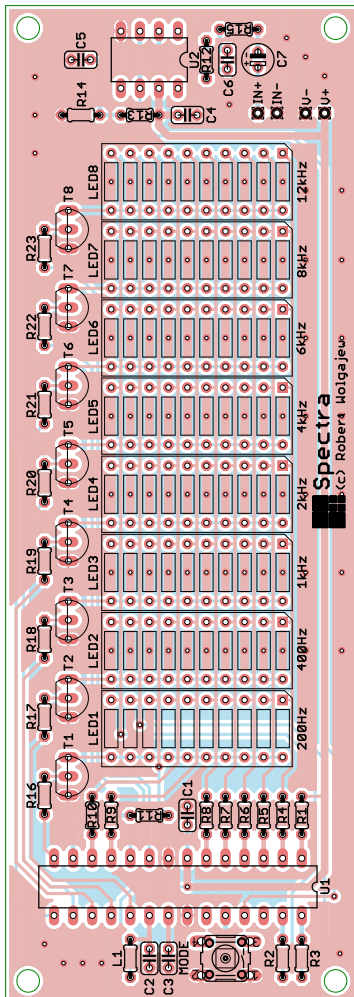
```
#define ROW_LSB_PORT PORTD
#define ROW_LSB_DDR DDRD
#define ROW_MSB_PORT PORTC
#define ROW_MSB_DDR DDRC
#define ROW_MSB_0 PC5
#define ROW_MSB_1 PC4

#define COLUMN_PORT PORTB
#define COLUMN_DDR DDRB
#define COLUMN_0 PB0
#define COLUMN_1 PB5
#define COLUMN_2 PB4
#define COLUMN_3 PB3
#define COLUMN_4 PB2
#define COLUMN_5 PB1
#define COLUMN_6 PB7
#define COLUMN_7 PB6
#define COLUMN_BLANK COLUMN_PORT = 0x00

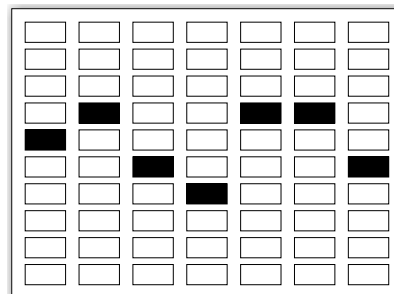
//Zmienna przechowująca wartość
//wyświetlaną na kolejnych słupkach
//wyświetlacza
extern uint16_t Led[8];
```

widma mocy sygnału, stosowne moce przeliczane są do skali logarytmicznej (dB), co wynika głównie z dużej dynamiki sygnału *Power[]* i konieczności pokazania go na ograniczonej rozdzielczości pionową wyświetlacza LED (10 diod) skali sygnału. Przeliczenie, o którym mowa, wykonywane jest według następującej zależności: $Power[i]=3,333\cdot\log_{10}(Power[i])$.

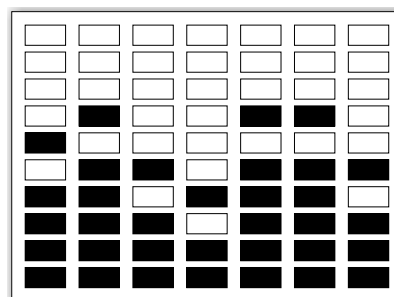
Tyle, jeśli chodzi o zagadnienia przetwarzania sygnałów DSP. Na koniec omówię w skrócie stosowne funkcje obsługi



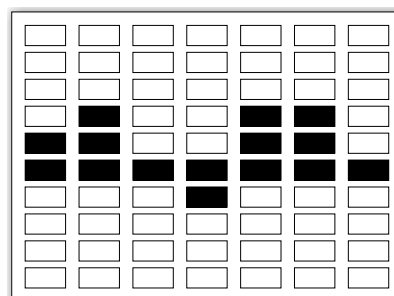
Rysunek 3. Schemat płytki PCB wraz z rozmieszczeniem elementów



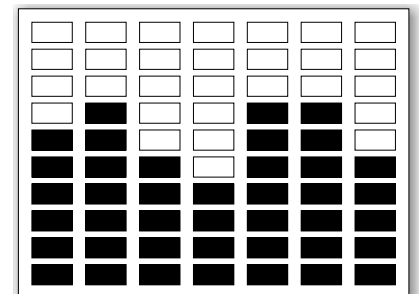
Point



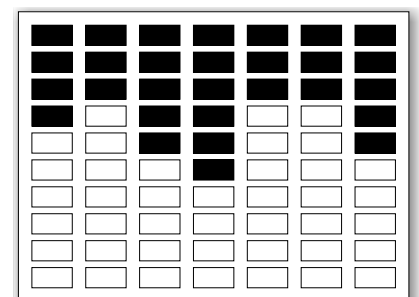
Bar maximum



Baseline



Bar



Bar negative

Rysunek 4. Wizualizacje przedstawiające 5 trybów wyświetlania informacji o widmie sygnału audio

mechanizmu multipleksowania odpowiada za wyświetlanie danych naszego ekranu LED. Zaczniemy od pliku nagłówkowego, którego zawartość pokazano na **listingu 6**. Dalej, na **listingu 7** pokazano funkcję inicjalizacyjną mechanizmu multipleksowania, która ustawia odpowiednie stany logiczne na portach wspólnych katod i anod wyświetlacza LED oraz inicjuje stosowny Timer sprzętowy.

Kolejną, prezentowaną funkcją jest funkcja obsługi przerwania realizująca mechanizm multipleksowania, której zawartość pokazano na **listingu 8**. Wspomniana funkcja korzysta z deklaracji zmiennej umieszczonej w pamięci Flash mikrokontrolera:

```
const uint8_t Columns[8] PROGMEM
= {
    (1<<COLUMN_0), (1<<COLUMN_1),
    (1<<COLUMN_2), (1<<COLUMN_3),
    (1<<COLUMN_4), (1<<COLUMN_5),
    (1<<COLUMN_6), (1<<COLUMN_7)};
```

Wprowadzenie tej zmiennej upraszcza dostęp do portów kolumn. Tyle w kwestiach programowych, przejdźmy zatem do zagadnień montażowych.

Montaż i uruchomienie

Schemat montażowy urządzenia Spectra został pokazany na **rysunku 3**. Zaprojektowano bardzo zwarty obwód drukowany z zastosowaniem wyłącznie elementów THT. Montaż urządzenia rozpoczynamy od przylutowania wszystkich rezystorów, następnie lutujemy diodki L1, kondensatory, potem układy

Listing 8. Funkcja obsługi przerwania realizująca mechanizm multipleksowania

```
//Zmienna przechowująca wartość
//wyświetlaną na kolejnych słupkach wyświetlacza
uint16_t Led[8];

ISR(TIMER1_COMPA_vect){
    //Numer kolejnego słupka do wyświetlenia
    static uint8_t Nr;

    //Wygaszenie wszystkich kolumn
    COLUMN_BLANK;

    //Wystawienie właściwych stanów na portach wierszy
    ROW_LSB_PORT = Led[Nr] & 0xFF;
    if(Led[Nr] & 0b100000000)
        ROW_MSB_PORT |= (1<<ROW_MSB_0);
    else
        ROW_MSB_PORT &= ~(1<<ROW_MSB_0);

    if(Led[Nr] & 0b1000000000)
        ROW_MSB_PORT |= (1<<ROW_MSB_1);
    else
        ROW_MSB_PORT &= ~(1<<ROW_MSB_1);

    //Załączenie odpowiedniej kolumny
    COLUMN_PORT = pgm_read_byte(&Columns[Nr]);
    //Wybranie kolejnej kolumny (słupka LED)
    Nr = (Nr+1) & 0x07;
}
```

scalone (które warto wyposażyć w stosowne podstawki) a na końcu wyświetlacze LED1...LED8 oraz wyłącznik MODE.

Poprawnie zmontowany układ nie wymaga żadnych regulacji i powinien działać tuż po włączeniu zasilania. Dla dociekliwych warto wspomnieć, iż program obsługi aplikacji urządzenia Spectrum przewiduje 5 trybów wyświetlania informacji o widmie sygnału audio, których przykładowe wizualizacje obrazuje **rysunek 4**. Co warto podkreślić, tryb trzeci (Bar maximum) integruje dodatkową funkcjonalność w postaci pokazywania wartości szczytowej w każdym z pasm częstotliwościowych.

Podsumowanie

Testy końcowe urządzenia z użyciem generatora arbitralnego wykazały bardzo dużą selektywność układu, jeśli chodzi o reakcję na wybrane częstotliwości analizowane przez nasze urządzenie oraz nieco mniejszą „odpowiedź” układu w przypadku „zwykłych” sygnałów audio (zwłaszcza tych o wysokich częstotliwościach), co skłoniło mnie do zwiększenia czułości funkcji liczącej dyskretną transformatę Fouriera DFT oraz do zmiany skali wynikowej tej funkcji na skalę logarymiczną, aby poprawnie odwzorować bardzo dużą dynamikę sygnału.

Robert Wołgajew, EP

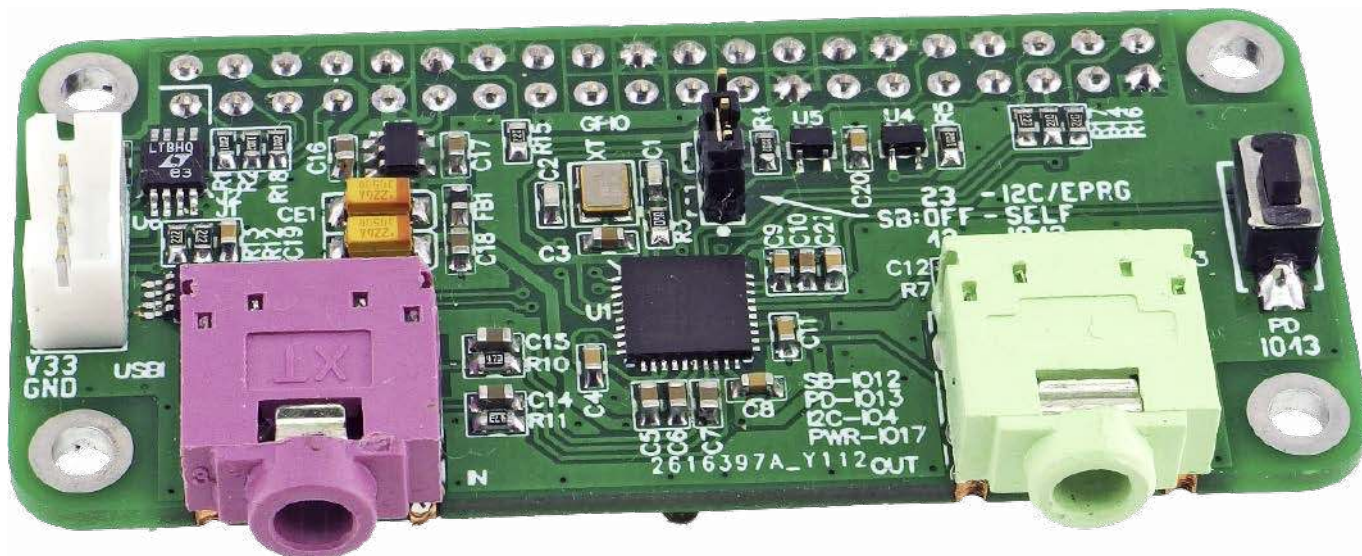
REKLAMA



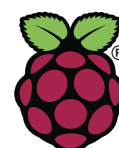
KIT AVT na wideo: [HTTP://BIT.LY/2SCLZTY](http://bit.ly/2SCLZTY)

O KIT-ach AVT przeczytasz również na Facebooku: [HTTP://BIT.LY/2BJVMN7](http://bit.ly/2BJVMN7)

 AVTEDU634 - MigoLEDki 0:25	 AVTEDU631 - Wskaźnik kierunku LED 0:38	 AVTEDU635 - Minipianino 0:36	 AVTEDU621 - Stroboskop policyjny LED 0:34	 AVTEDU630 - TermoEmotek 1:06	 AVT1996 - Bedlight - sterownik oświetlenia... 0:42
 AVT3144 - Klaskacz - przełącznik akustyczny 0:26	 AVT3250 - Bombka LED dla każdego - montaż 2:06	 AVT3165 - Odstraszacz kretów 0:28	 AVT5599 - Zdalnie sterowany włącznik 4-kanalowy 0:37	 AVT1484 - Wskaźnik temperatury silnika 0:26	 AVT5596 - Mieszacz kolorów RGB 0:40
 AVT1960 - Termometr z termoparą i alarmem 0:34	 AVT777 - Sterownik miniwiertarki modelarskiej 0:34	 AVT0001 - Uniwersalny regulator impulsowy 5A 0:42	 AVT5554 - Gra elektroniczna SNAKE 0:30	 AVT478 - Regulator obrotów wentylatorów 12V 0:30	 AVT720 - Błękitno-biały mrygacz 0:32
 AVT1853 - Iluminofonia LED RGB 1:28	 AVT2942 - Kogut dyskotekowy 1:06	 AVT3125 - Włącznik sterowany dowolnym pilotem 0:32	 AVT788 - Lampa LED reagująca na klasknięcia... 0:38	 AVT1900 - Animowany bałwanek LED 0:54	 AVT1651 - Gra - Kto pierwszy ten lepszy 0:34



Moduł DSP Audio do Raspberry Pi



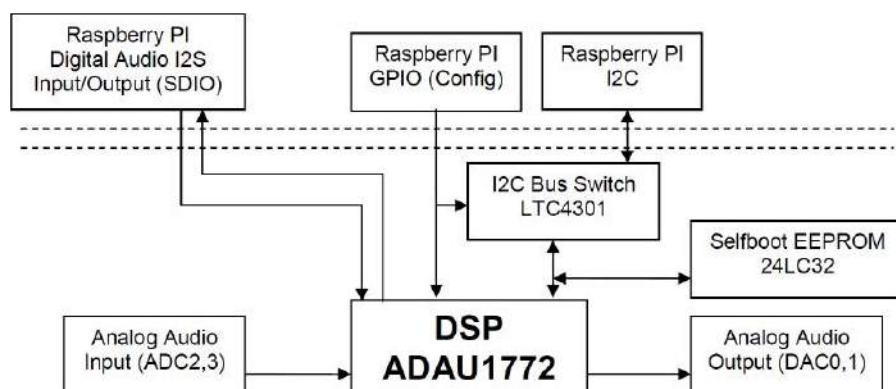
Połączenie Raspberry Pi i procesora DSP umożliwia tworzenie wbudowanych aplikacji multimedialnych o funkcjach nieosiągalnych przy zastosowaniu zwykłego DAC.

Moduł DSP zbudowany jest w oparciu na nowoczesnym układzie DSP firmy Analog Devices typu ADAU1772. Procesor jest konfigurowany i programowany za pomocą graficznego środowiska Sigma Studio, gdzie aplikacje tworzy się poprzez łączenie gotowych bloków funkcjonalnych. Docelowe parametry toru DSP mogą być zmieniane procesorem nadrzędnym, czyli bezpośrednio z Raspberry Pi.

Schemat blokowy modułu został pokazany na **rysunku 1**. Projekt zakładał osiągnięcie możliwie największej uniwersalności. Procesor DSP typu ADAU1772, którego budowę

wewnętrzną pokazuje **rysunek 2**, jest odpowiedzialny za obróbkę stereofonicznego sygnału audio w torze analogowym, z wykorzystaniem wbudowanych przetworników ADC i DAC audio. Jest połączony z cyfrowym interfejsem audio I²S płytki Raspberry Pi, co dodatkowo umożliwia obróbkę sygnału z pomocą komputerka lub pracę w roli przetwornika ADC/DAC audio z wbudowanym procesorem DSP.

Elastyczna konfiguracja modułu przewiduje pracę procesora ADAU1772 w trybie Selfboot, gdzie aplikacja i ustawienia pobierane są z pamięci EEPROM każdorazowo po włączeniu zasilania i może zostać modyfikowana na bieżąco poprzez magistralę I²C. Drugim trybem pracy jest ładowanie parametrów i przygotowanego algorytmu bezpośrednio do pamięci DSP z pominięciem EEPROM przez aplikację z Raspberry Pi.



Rysunek 1. Schemat blokowy modułu DSP

Budowa i działanie

Układ procesora U1 jest taktowany rezonatorem XT o częstotliwości 12,288 MHz zapewniając wzorcowy sygnał zegarowy zgodny z typowymi

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

W ofercie AVT* AVT----

Podstawowe parametry:

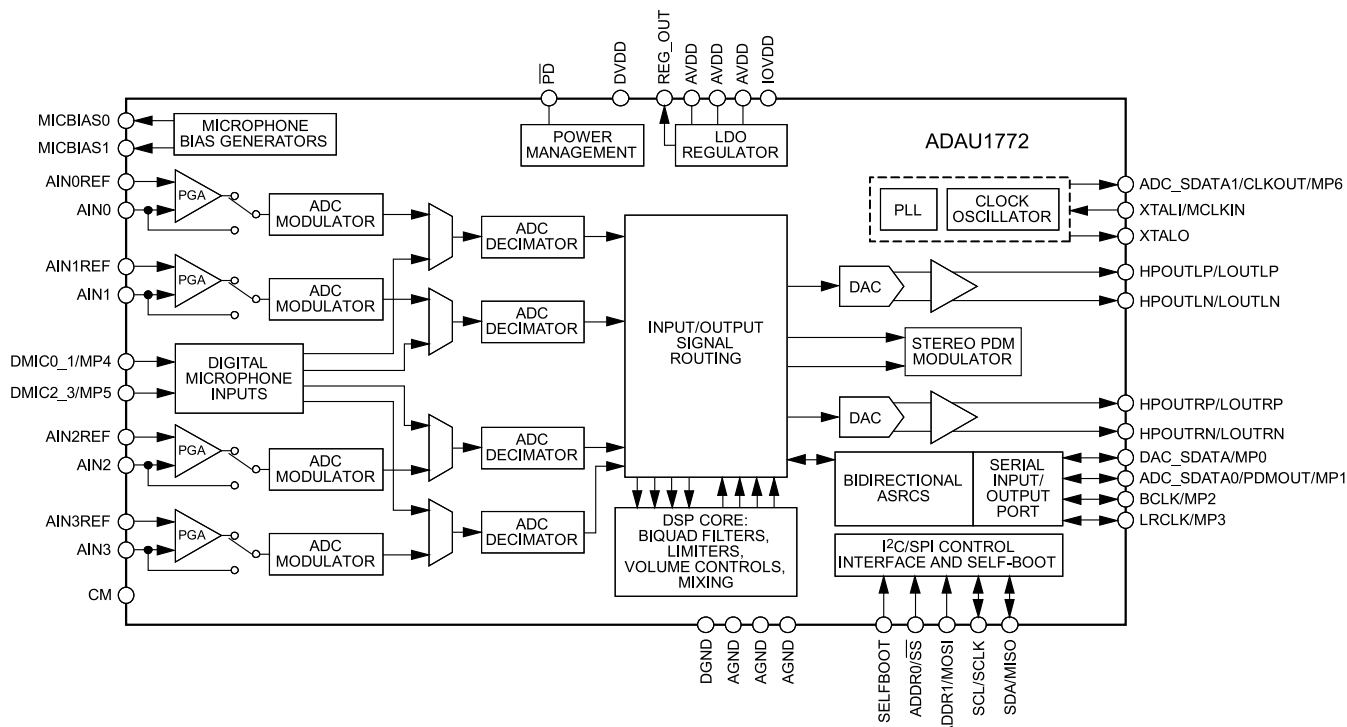
- zawiera procesor DSP typu ADAU1772 z wbudowanymi przetwornikami ADC i DAC audio,
- procesor pracuje w trybie Selfboot, czyli aplikacja i ustawienia pobierane są z pamięci EEPROM przy każdym uruchomieniu modułu,
- docelowe parametry toru DSP mogą być zmieniane procesorem nadrzędnym, czyli bezpośrednio z Raspberry Pi.

Projekty pokrewne na www.media.avt.pl:

- AVT-5858 Płytki bazowa dla Raspberry Pi Pico (EP 5/2021)
- AVT-5851 Dwukanałowy port szeregowy dla Raspberry (EP 3/2021)
- AVT-5847 Interfejs wyświetlacza TFT-RGB dla RPi Zero (EP 3/2021)
- Sterownik 18 LED dla Pi Zero (EP 2/2021)
- AVT-5837 Moduł do pomiaru napięcia i prądu z interfejsem I²C (EP 1/2021)
- AVT-5811 Odtwarzacz audio z Raspberry Pi (EP 10-12/2020)

*** 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 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 [B] (elementy wlutowane w płytkę PCB)
 - wersja [A] – płytki drukowane bez elementów i dokumentacji
 - wersja [A+] – płytki drukowane z elementami i dokumentacją
 - wersja [A+] – płytki drukowane [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.



Rysunek 2. Schemat blokowy układu ADAU1772

częstotliwościami próbkowania audio. Wejściowy analogowy sygnał audio z wejścia IN jest doprowadzony do przetworników ADC2 i ADC3 układu. Analogowy sygnał wyjściowy z przetworników DAC0 i DAC1 bloku DSP doprowadzony jest do wyjścia OUT. Oba gniazda są standardzie mini jack stereo 3,5 mm.

Sygnał cyfrowego audio I²S ze złącza GPIO Raspberry Pi jest doprowadzony do portów szeregowych SDI/SDO procesora ADAU1772. W przypadku, gdy częstotliwość próbkowania obrabianego sygnału cyfrowego nie jest zgodna z ustalonym próbkowaniem f_s w DSP, należy użyć bloków konwerterów częstotliwości próbkowania ASRC wbudowanych w ADAU1772.

Procesor jest zasilany napięciem V33 (3,3 V) uzyskiwanym z linii zasilającej V50 (+5 V) modułu Raspberry Pi oraz generowanym wewnątrz napięciem DVDD (1,1/1,2 V) zasilającym rdzeń DSP. Napięcie V33 i dodatkowo filtrowane zasilanie części analogowej DSP V33A dostarczane jest przez stabilizator LDO U3 typu MCP1812AT-033. Układ U3 wyposażony jest w funkcję rozładowania podłączonych pojemności wyjściowych, co eliminuje problemy z prawidłowym restartem U1. Napięcie wyjściowe stabilizatora kluczowane jest sygnałem PWR17 z linii GPIO17 Raspberry Pi. Podanie stanu wysokiego włączy stabilizator U3 i zasilanie DSP. Sygnał PDE13 z linii GPIO13 umożliwia sterowanie wyprowadzeniem !PD układu U1 odpowiedzialnym za wprowadzenie procesora w tryb obniżonego poboru mocy. Tryb PD może też zostać wyzwolony przyciskiem PD.

Inwerter U4 z otwartym kolektorem sumuje sygnały z GPIO13 i przycisku PB. Ustawienie linii GPIO13 w stan wysoki wprowadza U1 w tryb obniżonego poboru mocy niezależnie od stanu przycisku PD.

Tryb pracy DSP jest wybierany stanem wyprowadzenia 27 układu U1. Tryb Selfboot, w którym procesor pracuje samodzielnie, a aplikacja jest ładowana z pamięci EEPROM po włączeniu zasilania U2, aktywowany jest stanem wysokim sygnału SB. Stan niski podczas włączenia zasilania konfiguruje układ do pracy z ładowaniem aplikacji z I²C (lub z programatora USBi). Do wyboru trybu pracy służy zwora SB, która rozwarła aktywuje Selfboot, ze zwartymi wyprowadzeniami 2–3 tryb I²C, a ze zwartymi 1–2 przekazuje sterowanie trybem na wyprowadzenie GPIO12

(sygnał PGE12). Stan niski GPIO12 aktywuje tryb Selfboot. Zachowanie odpowiednich sekwencji sygnałów PWR17, PDE13, SBE12 umożliwia wprowadzenie ADAU1772 w wymagany tryb pracy.

Ostatnim, ale bardzo istotnym elementem jest układ U6 typu LTC4301CMS. Jest to sterowany bufor magistrali I²C separujący magistralę Raspberry Pi i ADAU1772 podczas ładowania programu z pamięci EEPROM. W przypadku bezpośredniego połączenia na magistrali dwóch układów master bez arbitrażu nastąpi zablokowanie magistrali. Wymuszony programowo niski stan sygnału I2C4 z GPIO4 aktywuje bufory i załączy magistralę I²C, gdy ADAU1772 zakończy cykl Selfboot. Połączenie realizowane jest dopiero po wykryciu sekwencji STOP lub IDLE magistrali, aby nie zakłócać jej działania. Rozłączenie magistral umożliwia także programowanie pamięci EEPROM lub konfigurację ADAU1772 poprzez złącze USBi zewnętrznym programatorem USBi.

Montaż i uruchomienie

Moduł jest zmontowany na niewielkiej dwustronnej płytce drukowanej zgodnej ze standardem Raspberry Pi Zero. Jej schemat wraz z rozmieszczeniem elementów został pokazany na **rysunku 4**. Sposób montażu jest klasyczny i nie wymaga dokładnego opisu, gotowa płytka wygląda tak jak na fotografii tytułowej.

Prawidłowo zmontowany moduł nie wymaga uruchamiania i po konfiguracji sprzętowo-programowej może być użyty w docelowej aplikacji. Przykładowym projektem postaram się ułatwić rozpoczęcie eksperymentów z DSP. Projekt będzie obsługiwał

Wykaz elementów:

Rezystory: (SMD0603, 1%)

R1, R2, R4, R5, R18: 10 kΩ
R3, R7, R9: 100 Ω
R6, R8, R10, R11: 47 kΩ
R12, R13: 2,2 kΩ
R14...R17: 22 Ω

Kondensatory:

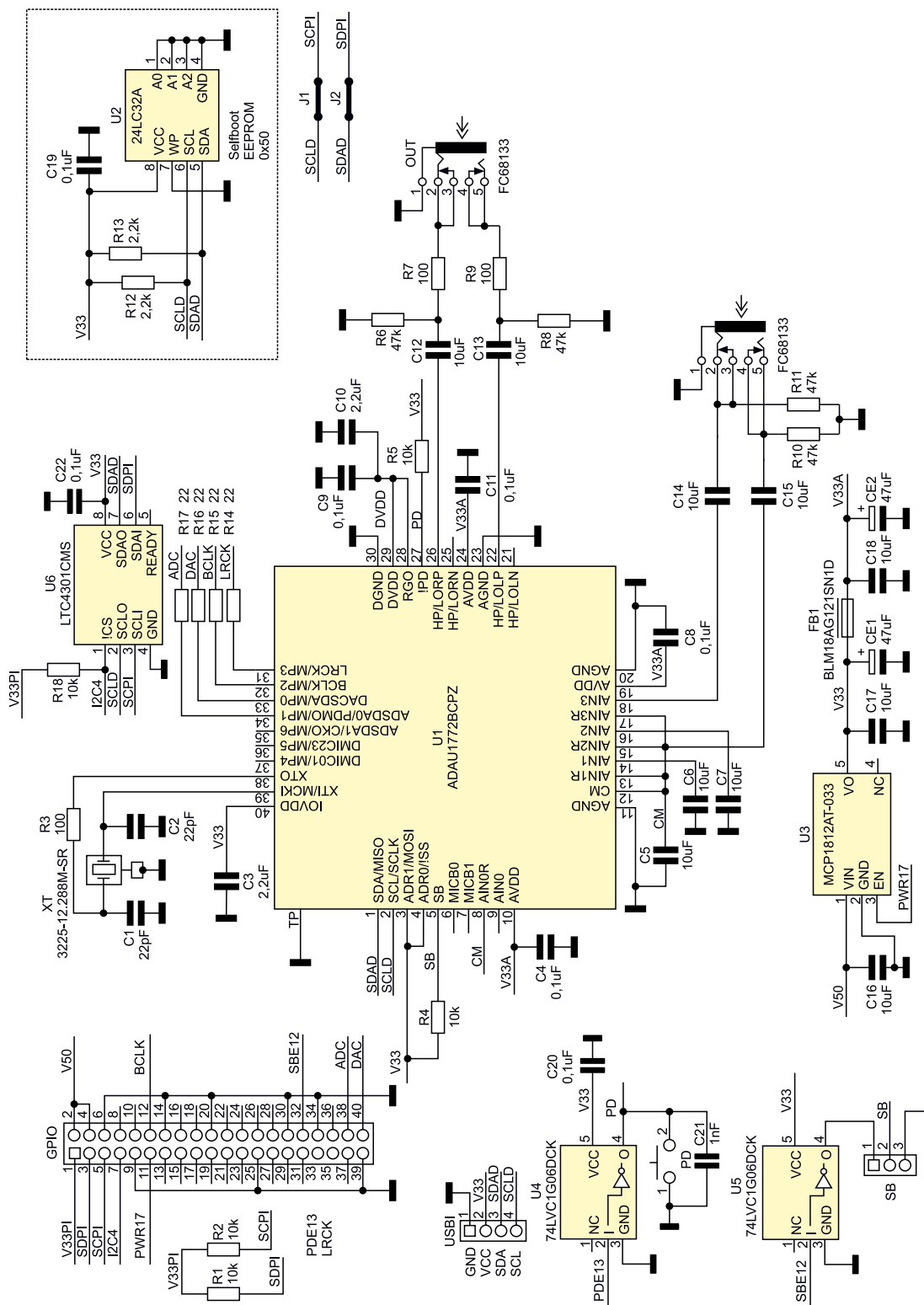
C1, C2: 22 pF SMD0603
C3, C10: 2,2 μF SMD0603
C4, C8, C9, C11, C19, C20, C22: 0,1 μF SMD0603
C5...C7, C12...C18: 10 μF SMD0603
C21: 1 nF SMD0603
CE1, CE2: 47 μF SMD3216 tantalowy 10 V

Półprzewodniki:

U1: ADAU1772BCPZ (LFCSP40)
U2: 24LC32A (MSOP8)
U3: MCP1812AT-033 (SOT-23-5)
U4, U5: 74LVC1606DCK (SC70-5)
U6: LTC4301CMS8 (MSOP8)

Pozostałe:

XT: rezonator kwarcowy 12,288M-SR CFPX-180 SMD
FB1: koralik ferrytowy BLM18AG121SN1D SMD603
GPIO: złącze żeńskie IDC40
USBi: złącze PH4 2 mm proste
IN, OUT: gniazdo mini jack stereo FC68133
SB: lista SIP3 2 mm + zwora
J1, J2: zwora PCB, opis w tekście
PD: mikroprzełącznik



Rysunek 3. Schemat ideowy modułu DSP

miksowanie sygnału ze źródła analogowego z wejścia IN z sygnałem cyfrowym I²S pochodzącym z Raspberry Pi.

Do oprogramowania procesora DSP potrzebne będzie środowisko Sigma Studio i programator USBi, a do konfiguracji komputer Raspberry Pi z aktywnym I²C i I²S DAC. W systemie Pi należy skonfigurować DAC, zgodny z PCM5102. Na wszelki wypadek należy zaktualizować system poleceniami:

```
sudo rpi-update
sync
```

sudo reboot

Następnie należy:

- usunąć z pliku `/etc/modprobe.d/raspi-blacklist.conf` linie:

```
blacklist i2c-bcm2708
blacklist snd-soc-pcm512x
blacklist snd-soc-wm8804
```
- usunąć driver w pliku `/etc/modules`:

```
snd_bcm2835
```
- skonfigurować DAC, zgodny z PCM5102, dodając w pliku `/boot/config.txt`

```
dtoverlay=hifiberry-dac
```

- skonfigurować ALSA, tworząc plik `/etc/asound.conf` z zawartością:

```
pcm.!default {
    type hw card 0
}
ctl.!default {
    type hw card 0
}
```

Następnie należy zrestartować PI i po uruchomieniu sprawdzić poprawność konfiguracji poleceniem:

```
sudo aplay -l
```

DAC powinien pojawić się na liście dostępnych urządzeń odtwarzających:

```
**** List of PLAYBACK Hardware
Devices ****
card 0: sndrpihifiberry [snd_
rpi_hifiberry_dac], device 0:
HiFiBerry DAC HiFi pcm5102a-
hifi-0 []
Subdevices: 1/1
Subdevice #0: subdevice #0
```

Do odtwarzania plików muzycznych, z racji uruchamiania w konsoli, używany będzie program aplay. Należy też przygotować kilka plików testowych audio 44,1 kHz w formacie *.wav, najlepiej z muzyką, której słuchanie nie sprawia nam przykrości. Potrzebne będzie też źródło sygnału analogowego np. wyjście audio z PC oraz wzmacniacz lub głośniki PC do odsłuchu efektów.

Przed przygotowaniem projektu w Sigma Studio, sprawdzimy poprawność sterowania DSP z poziomu Raspberry Pi:

- konfigurujemy sterowanie zasilaniem GPIO17 jako wyjście:
`echo 17 > /sys/class/gpio/export`
`echo out > /sys/class/gpio/gpio17/direction`
- sprawdzamy kluczowanie zasilania poleceniami:
`echo 1 > /sys/class/gpio/gpio17/value`
`echo 0 > /sys/class/gpio/gpio17/value`

Ustawienie stanu 1 załączy, a stanu 0 wyłączy +3,3 V dla DSP;

- następnie sprawdzamy sterowanie GPIO13 sygnałem PD, ustawiając go jako wyjście:
`echo 13 > /sys/class/gpio/export`
`echo out > /sys/class/gpio/gpio13/direction`
- sprawdzamy sterowanie PD poleceniami:
`echo 1 > /sys/class/gpio/gpio13/value`
`echo 0 > /sys/class/gpio/gpio13/value`

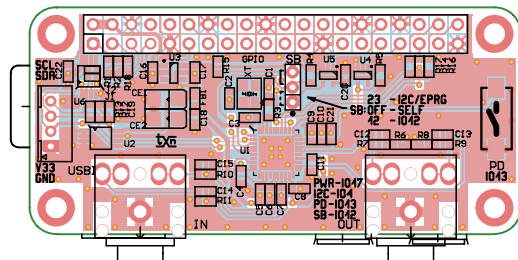
Ustawienie stanu 1 wprowadzi DSP w stan obniżonego poboru mocy, a stan 0 aktywuje DSP;

- następnie w położeniu zwory SB 1-2 sprawdzamy poprawność sterowania trybem pracy DSP, ustawiając GPIO12 jako wyjście:
`echo 12 > /sys/class/gpio/export`
`echo out > /sys/class/gpio/gpio12/direction`
- sprawdzamy sterowanie sygnałem SB poleceniami:
`echo 1 > /sys/class/gpio/gpio12/value`
`echo 0 > /sys/class/gpio/gpio12/value`
- ustawiamy stany;

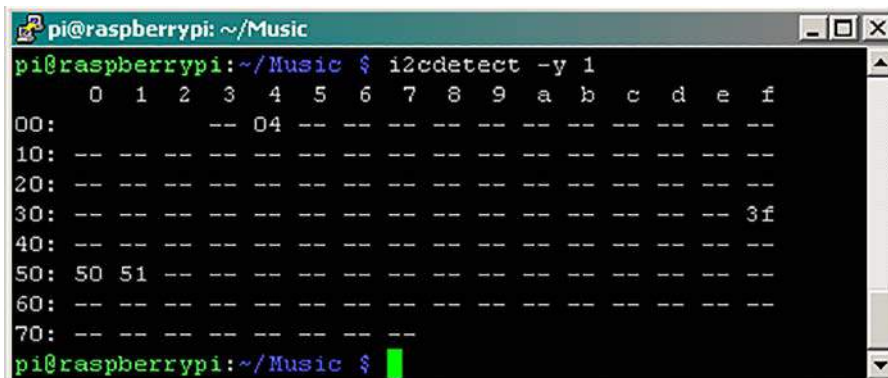
```
echo 1 > /sys/class/
gpio/gpio17/value
echo 0 > /sys/class/
gpio/gpio13/value
echo 0 > /sys/class/
gpio/gpio12/value
```

Teraz sprawdzamy działanie bufora LTC4301 poleceniem:
`i2cdetect -y 1`

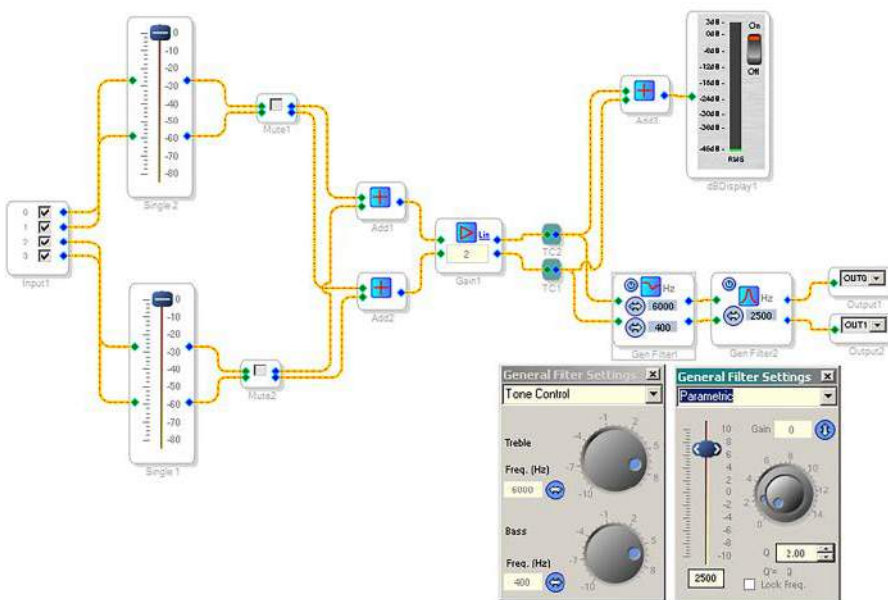
Na magistrali nie powinien pojawić się adres 0x3F DSP



Rysunek 4. Schemat płytki PCB z rozmieszczeniem elementów



Rysunek 5. Detekcja układów RaspbPI_Zero_DSP



Rysunek 6. Aplikacja testowa DSP

lub 0x50 pamięci EEPROM. Aktywując bufor poleceniami:

```
echo 4 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio4/
direction
echo 0 > /sys/class/gpio/gpio4/
value
```

i ponawiając polecenie:

```
i2cdetect -y 1
```

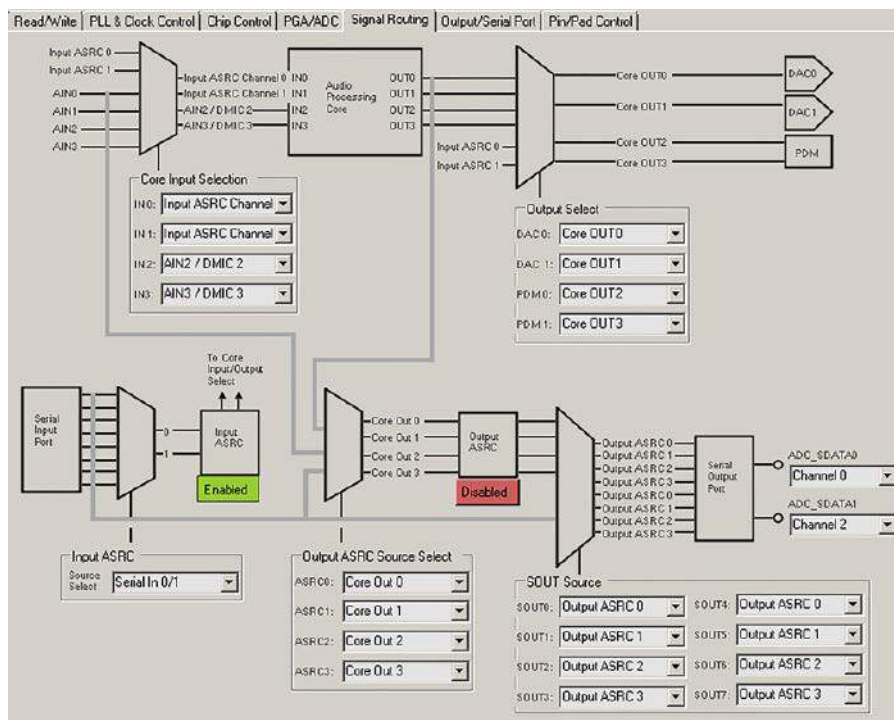
powinniśmy wykryć wszystkie układy modułu, co pokazano na **rysunku 5**. Pod adresem 0x3F widoczny jest ADAU1772, 0x50 pamięć EEPROM 24LC32, a 0x51 zajmuje programator USBi. UWAGA: pod adres 0x51 nie należy niczego zapisywać, bo uszkodzimy firmware programatora USBi.

W tym momencie można optymistycznie przyjąć, że sprzęt jest sprawny i należy zabrać się do oprogramowania, zakładając zworę SB w położenie 2-3 i odłączając I²C od procesora poleceniem:

```
echo 1 > /sys/class/gpio/gpio4/
value
```

Aplikacja testowa

Schemat aplikacji testowej ADAU1772_DSP-DAC.dspproj przygotowanej w Sigma Studio został pokazany na **rysunku 6**. Źródła tego projektu są dostępne w materiałach dodatkowych. Stereofoniczne sygnały z bloku wejść cyfrowych (wejście 0 i 1) i analogowych (wejście 2 i 3) z bloku Input1 doprowadzone



Rysunek 7. Konfiguracja sygnałów DSP

są do regulatorów poziomu Single1/2. Następnie poprzez bloki wyciszenia Mute1/2 są sumowane w blokach Add1/2 i wzmacnione blokiem Gain1, skąd doprowadzone są do bloku miernika poziomu dBDisplay1 i na blok parametryzowanych filtrów Gen Filter1/2, a następnie do wyjść analogowych OUT0/1. Poziom mierzony przez blok dbDisplay1 jest przechowywany w rejestrze DBREG0 i może zostać użyty w aplikacji sterującej np. do sprawdzenia obecności sygnału audio.

Pierwszy z filtrów realizuje funkcję regulatora barwy dźwięku, drugi regulatora parametrycznego. Schemat konfiguracji związanej z przepływem obrabianych sygnałów pokazano na **rysunku 7**. Sygnały z wejściowego interfejsu cyfrowego I²S (Serial In 0/1) poddane są konwersji częstotliwości próbkowania ASRC i razem z sygnałami z ADC2/3 doprowadzone są do rdzenia DSP pracującego z ustawioną częstotliwością próbkowania fs i po obróbce wyprowadzone są na przetworniki DAC0 i DAC1 z niezmiennym fs.

W pozostałych zakładkach skonfigurowane są bloki ADC/PGA/DAC oraz PLL i zasilanie układu. Aby wgrać aplikację do DSP, konieczne jest skonfigurowanie USBi i systemu zgodnie z **rysunkiem 8**. Z sygnałów programatora dostępnych na kablu USBi typu IDC10 łączymy z gniazdem USBi modułu tylko magistralę I²C: SDA, SCL oraz masę programatora, zachowując możliwie krótkie przewody.

Konfigurujemy system: procesor ADAU1772 z ustawionym adresem 0x7E (0x3F) i pamięć EEPROM z adresem 0xA0 (0x50). Należy też ustawić typ i parametry

pamięci zgodnie z **rysunkiem 9**, (prawy klawisz na ikonie IC2). W zakładce IC2 WinE2PromLoader należy pamięć wyczyścić (Clear E2Pro) i zaprogramować jej zawartość skompilowanym projektem, wracając do zakładki Config i wybierając opcję Write Latest Compilation to E2Prom (prawy klawisz na ikonie IC1).

Po założeniu zwory SB w położenie 2–3 (sterowanie z GPIO) restartujemy DSP sekwencją sygnałów PD/PWR/SB/PWR/PD poleceniami:

```
echo 1 > /sys/class/gpio/gpio13/value
echo 0 > /sys/class/gpio/gpio17/value
echo 0 > /sys/class/gpio/gpio12/value
echo 1 > /sys/class/gpio/gpio17/value
echo 0 > /sys/class/gpio/gpio13/value
echo 0 > /sys/class/gpio/gpio4/value
```

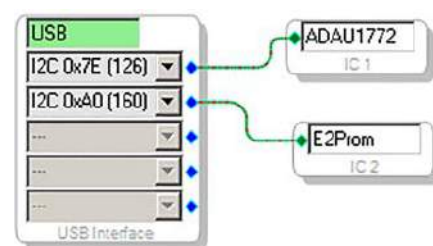
Jeżeli do wejścia analogowego mamy podłączony sygnał audio, do wyjścia głośnikowego powinien docierać słyszalny sygnał testowy. Teraz możemy sprawdzić miksowanie sygnału z I²S Raspberry Pi, odtwarzając przygotowany plik testowy 01.wav:

```
aplay 01.wav
```

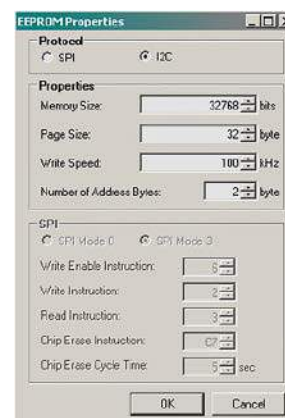
Zostanie on zmiksowany z sygnałem testowym z wejścia IN. Jeżeli wstępne testy przebiegły pomyślnie, procesor jest zaprogramowany i działający, pozostaje przetestowanie komunikacji i sterowania DSP przez I²C poleceniami:

```
echo 0 > /sys/class/gpio/gpio4/value
```

Załączmy bufor, dla pewności sprawdzamy obecność układów na magistrali:



Rysunek 8. Konfiguracja USBi



Rysunek 9. Konfiguracja EEPROM

i2cdetect-y 1

Wracając do Sigma Studio, po wykonaniu kompilacji i ładowania projektu, opcją Link Compile Download (klawisz F7) aktywowana jest ikona Export System Files, dzięki której możemy wyeksportować pliki konfiguracyjne niezbędne do stworzenia aplikacji sterującej DSP. Pliki zapisane są w formacie tekstowym i zawierają nagłówki zgodne z C/C++ określające parametry projektu i używane rejestry DSP. Warto zapoznać się z zawartością wygenerowanych plików, jest to niezbędne do utworzenia aplikacji sterującej DSP.

Za pomocą Sigma Studio możemy też wygenerować zawartość EEPROM, którą można wykorzystać do samodzielnego programowania, czy to skryptem czy zewnętrznym programatorem (Upload E2Prom To File w zakładce IC 2–WinE2PromLoader). Podczas eksportu zostanie wygenerowanych kilka typów plików:

- **.params* – zawiera wszystkie nazwy parametrów aplikacji DSP, adresy i wartości do zapisania w pamięci;
- **.hex* – zawiera wartości bajtów pamięci parametrów w formacie szesnastkowym, dane te są również odzwierciedlone w polu Parameter Data pliku **.params*;
- **.h* – które definiują nazwy, adresy i wartości wszystkich parametrów aplikacji DSP;
- **_REG.h* – plik nagłówkowy zawiera definicje wszystkich rejestrów sprzętowych wraz z wartościami początkowymi.

Oczywiście po każdej zmianie algorytmu konieczne jest ponowne generowanie kompletu plików konfiguracyjnych. Aby przyspieszyć tworzenie aplikacji, gdy konfiguracja DSP jest już określona i sprawdzona,

a zależy nam tylko na szybkim wyznaczeniu zmienianych parametrów bloków funkcjonalnych, należy użyć okienka Capture (**rysunek 10**). Każdorazowo po zmianie parametru Sigma Studio zapisuje rejestry DSP odpowiednio przeliczonymi wartościami. Dla przykładu, fragment parametrów zmiany poziomu tonów wysokich bloku Gen Filter1, która zapisana jest w pliku *.params, wygląda następująco:

```
Cell Name           = Gen Filter1
Parameter Name      =
FilterDouble10B1
Parameter Address   = 391
Parameter Value     =
-1.20945749431849
Parameter Data :
0xF6 , 0x53 , 0x07 , 0xF3 ,
```

Dla przyspieszenia pracy zmienione parametry mogą być skopiowane do schowka lub zapisane po kliknięciu prawym klawiszem w obszarze okienka Capture. Po przygotowaniu zestawu nastaw dla wszystkich bloków można przygotować oprogramowanie sterujące DSP.

Testowe sterowanie procesorem można przeprowadzić, korzystając z narzędzi i2c-tools. Adresy rejestrów sprzętowych i realizowane funkcje są ustalone i nie podlegają zmianie niezależnie od aplikacji DSP. Dla próby zmienimy poziom sygnału DAC, w tym celu w pliku ADAU1772_DSPDAC_IC_1_REG.h, w którym znajdują się definicje

Mode	Time	Cell Name	Parameter Name	Address	Value	Data	Bytes
Block Write	19:16:11 - 99ms	Gen Filter1	FilterDouble10B1	0x0187	1	0x08, 0x00, 0x00, 0x00	4
Block Write	19:16:11 - 99ms	Gen Filter1	FilterDouble11B1	0x01A7	0	0x00, 0x00, 0x00, 0x00	4
Block Write	19:16:11 - 99ms	Gen Filter1	FilterDouble12B1	0x01C7	0	0x00, 0x00, 0x00, 0x00	4
Block Write	19:16:11 - 99ms	Gen Filter1	FilterDouble13A1	0x01E7	0	0x00, 0x00, 0x00, 0x00	4
Block Write	19:16:11 - 99ms	Gen Filter1	FilterDouble14A1	0x0207	0	0x00, 0x00, 0x00, 0x00	4
Block Write	19:16:11 - 100ms	Gen Filter1	IC 1.CORE_CONTROL	0x0009		0x23	1
Block Write	19:16:11 - 100ms	Gen Filter1	FilterDouble10B1			0x08, 0x00, 0x00, 0x00	4
Block Write	19:16:11 - 100ms	Gen Filter1	FilterDouble11B1			0x00, 0x00, 0x00, 0x00	4
Block Write	19:16:11 - 100ms	Gen Filter1	FilterDouble12B1			0x00, 0x00, 0x00, 0x00	4
Block Write	19:16:11 - 100ms	Gen Filter1	FilterDouble13A1			0x00, 0x00, 0x00, 0x00	4
Block Write	19:16:11 - 100ms	Gen Filter1	FilterDouble14A1			0x00, 0x00, 0x00, 0x00	4
Block Write	19:16:11 - 100ms	Gen Filter1	FilterDouble20B1			0x08, 0x00, 0x00, 0x00	4
Block Write	19:16:11 - 100ms	Gen Filter1	FilterDouble21B1			0x00, 0x00, 0x00, 0x00	4
Block Write	19:16:11 - 100ms	Gen Filter1	FilterDouble22B1			0x00, 0x00, 0x00, 0x00	4
Block Write	19:16:11 - 100ms	Gen Filter1	FilterDouble21A1			0x00, 0x00, 0x00, 0x00	4
Block Write	19:16:11 - 101ms	Gen Filter1	FilterDouble22A1	0x016C	0	0x00, 0x00, 0x00, 0x00	4
Block Write	19:16:11 - 101ms	Gen Filter1	IC 1.CORE_CONTROL	0x0009		0x03	1
Block Write	19:16:11 - 101ms	Gen Filter1	FilterDouble20B1	0x018C	1	0x08, 0x00, 0x00, 0x00	4
Block Write	19:16:11 - 101ms	Gen Filter1	FilterDouble21B1	0x01AC	0	0x00, 0x00, 0x00, 0x00	4
Block Write	19:16:11 - 101ms	Gen Filter1	FilterDouble22B1	0x01CC	0	0x00, 0x00, 0x00, 0x00	4
Block Write	19:16:11 - 101ms	Gen Filter1	FilterDouble21A1	0x01EC	0	0x00, 0x00, 0x00, 0x00	4
Block Write	19:16:11 - 101ms	Gen Filter1	FilterDouble22A1	0x020C	0	0x00, 0x00, 0x00, 0x00	4

Rysunek 10. Zmiana parametrów

rejestrów DSP, szukamy sekcji DAC0_VOLUME, DAC1_VOLUME odpowiedzialnej za poziom sygnału wyjściowego DAC (pełną mapę ponad 70 rejestrów DSP możemy też oczywiście sprawdzić w dokumencie ADAU1772):

```
/* DAC0_VOLUME - Registers (IC
1) */
#define REG_DAC0_VOLUME_IC_1_
ADDR                0x30
#define REG_DAC0_VOLUME_IC_1_
BYTE                1
#define REG_DAC0_VOLUME_IC_1_
VALUE              0x2C
```

```
/* DAC1_VOLUME - Registers (IC
1) */
#define REG_DAC1_VOLUME_IC_1_
ADDR                0x30
#define REG_DAC1_VOLUME_IC_1_
BYTE                1
#define REG_DAC1_VOLUME_IC_1_
VALUE              0x2C
```

Jak widać, rejestry odpowiedzialne za poziom wyjściowy DAC znajdują się pod adresem 0x2F/0x30, a w rzeczywistości pod adresem 16-bitowym, 0x002F/0x0030, a poziom sygnału określony jest zapisanym

Listing 1. Zmiana poziomu sygnału dla bloku Single 1 na poziom 0 dB wymaga pokazanych sekwencji zapisów blokowych (skrypt vol0dB.sh)

```
#!/bin/bash
echo 'ADAU1772 vol 0dB'
i2cset -y 1 0x3F 0x01 0x84 0x08 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0xA4 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0xC4 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0xE4 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x02 0x04 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x00 0x09 0x23 1
i2cset -y 1 0x3F 0x00 0xE4 0x08 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0x04 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0x24 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0x44 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0x64 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x00 0xE9 0x08 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0x09 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0x29 0x00 0x00 0x00 0x00 0x00 1
```

Listing 2. Zmiana poziomu sygnału dla bloku Single 1 na poziom -20 dB wymaga pokazanych sekwencji zapisów blokowych (skrypt vol20dB.sh)

```
#!/bin/bash
echo 'ADAU1772 vol -20dB'
i2cset -y 1 0x3F 0x01 0x84 0x02 0x87 0xA2 0x6C 1
i2cset -y 1 0x3F 0x01 0xA4 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0xC4 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0xE4 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x02 0x04 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x00 0x09 0x23 1
i2cset -y 1 0x3F 0x00 0xE4 0x02 0x87 0xA2 0x6C 1
i2cset -y 1 0x3F 0x01 0x04 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0x24 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0x44 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0x64 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x00 0xE9 0x02 0x87 0xA2 0x6C 1
i2cset -y 1 0x3F 0x01 0x09 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0x29 0x00 0x00 0x00 0x00 0x00 1
```

Listing 3. Zmiana ustawień bloku Gen Filter 1 pomijająca jego działanie - bypass (skrypt toneoff.sh)

```
#!/bin/bash
echo 'ADAU1772 tone reg off'
i2cset -y 1 0x3F 0x01 0x87 0x08 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0xA7 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0xC7 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0xE7 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x02 0x07 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x00 0x09 0x23 1
i2cset -y 1 0x3F 0x00 0xE7 0x08 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0x07 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0x27 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0x47 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0x67 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x00 0xEC 0x08 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0x0C 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0x2C 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0x4C 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0x6C 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x00 0x09 0x03 1
i2cset -y 1 0x3F 0x01 0x8C 0x08 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0xAC 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0xCC 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x01 0xEC 0x00 0x00 0x00 0x00 0x00 1
i2cset -y 1 0x3F 0x02 0x0C 0x00 0x00 0x00 0x00 0x00 1
```

Listing 4. Przykładowe podbicie tonów niskich i wysokich (skrypt toneoff.sh)

```
#!/bin/bash
echo 'ADAU1772 tone reg set'
i2cset -y 1 0x3F 0x01 0x87 0xF1 0xC5 0x61 0xA8 1
i2cset -y 1 0x3F 0x01 0xA7 0x17 0x23 0x0C 0xC6 1
i2cset -y 1 0x3F 0x01 0xC7 0xF7 0x92 0x74 0xA1 1
i2cset -y 1 0x3F 0x01 0xE7 0x09 0x02 0x1A 0x0D 1
i2cset -y 1 0x3F 0x02 0x07 0xFE 0xC7 0x01 0xB5 1
i2cset -y 1 0x3F 0x00 0x09 0x23 1
i2cset -y 1 0x3F 0x00 0xE7 0xF1 0xC5 0x61 0xA8 1
i2cset -y 1 0x3F 0x01 0x07 0x17 0x23 0x0C 0xC6 1
i2cset -y 1 0x3F 0x01 0x27 0xF7 0x92 0x74 0xA1 1
i2cset -y 1 0x3F 0x01 0x47 0x09 0x02 0x1A 0x0D 1
i2cset -y 1 0x3F 0x01 0x67 0xFE 0xC7 0x01 0xB5 1
i2cset -y 1 0x3F 0x00 0xEC 0xF1 0xC5 0x61 0xA8 1
i2cset -y 1 0x3F 0x01 0x0C 0x17 0x23 0x0C 0xC6 1
i2cset -y 1 0x3F 0x01 0x2C 0xF7 0x92 0x74 0xA1 1
i2cset -y 1 0x3F 0x01 0x4C 0x09 0x02 0x1A 0x0D 1
i2cset -y 1 0x3F 0x01 0x6C 0xFE 0xC7 0x01 0xB5 1
i2cset -y 1 0x3F 0x00 0x09 0x03 1
i2cset -y 1 0x3F 0x01 0x8C 0xF1 0xC5 0x61 0xA8 1
i2cset -y 1 0x3F 0x01 0xAC 0x17 0x23 0x0C 0xC6 1
i2cset -y 1 0x3F 0x01 0xCC 0xF7 0x92 0x74 0xA1 1
i2cset -y 1 0x3F 0x01 0xEC 0x09 0x02 0x1A 0x0D 1
i2cset -y 1 0x3F 0x02 0x0C 0xFE 0xC7 0x01 0xB5 1
```

do nich bajtem. Wartość poziomu zmieniana jest z krokiem 0,125 dB. Wartości 0 dB odpowiada zapis 0x00, natomiast wartości -95,625 dB odpowiada zapis 0xFF. Za pomocą poleceń:

```
i2cset -y 1 0x3F 0x00 0x2F 0x10 i
i2cset -y 1 0x3F 0x00 0x30 0x10 i
```

możemy ustawić poziom sygnału DAC. W podobny sposób zmieniamy zawartość pozostałych rejestrów sprzętowych DSP, np. sprawdzamy poziom sygnału w rejestrze DBREG0:

```
i2cset -y 1 0x3F 0x00 0x0C
i2cget -y 1 0x3F
```

Odczytana wartość powinna zmieniać się w zależności od poziomu sygnału, 0 dB=0xFF. Ze względu na 16-bitowy adres konieczne jest rozbicie odczytu na dwa polecenia, pierwsze ustawia „wskaźnik adresu”, drugie odczytuje wartość rejestru.

Nieco bardziej skomplikowana jest sprawa parametryzacji realizowanych algorytmów. Tutaj też z pomocą przychodzi

funkcja Capture. Przed zmianą parametru najlepiej ustawić wartość domyślną, wyczyścić zawartość okienka Capture oraz zmienić pożądaną wartość, obserwując log aktywności komunikacji z DSP. Następnie, czy to poprzez schowek, czy zapis do pliku, należy wygenerować zestaw modyfikowanych rejestrów i ich zawartości oraz na ich podstawie przygotować oprogramowanie. Przykładowo, zmiana poziomu sygnału dla bloku Single 1 na poziom 0 dB wymaga sekwencji zapisów blokowych pokazanych na **listingu 1** (skrypt *vol0dB.sh*), natomiast zmiana na -20 dB wymaga sekwencji z **listingu 2** (skrypt *vol20dB.sh*). Zmiana ustawień bloku Gen Filter 1, pomijając jego działanie – bypass, jest możliwa po wysłaniu sekwencji z **listingu 3** (skrypt *toneoff.sh*), a przykładowe podbicie tonów niskich i wysokich pokazuje **listing 4** (skrypt *toneoff.sh*). Jeżeli skrypty działają, możemy uznać układ za przetestowany i przejść do właściwych aplikacji.

Podsumowanie

Na zakończenie warto wspomnieć o układzie kodeka ADAU1372, który jest uproszczoną wersją ADAU1772 pozbawioną rdzenia DSP, która jest w 100% zgodna sprzętowo i programowo. Liczba rejestrów konfiguracyjnych jest pomniejszona o rejestry rdzenia DSP, pozostałe zachowują adresacje i funkcje. W przypadku zastosowania ADAU1372 należy jedynie wyprowadzenie SB połączyć stale z masą układu. Zbędne są też układy U2 (24LC32), U5 (LVC06), U6 (LTC4301), elementy R12, R13, C19, a w miejsce U6 należy włutować zwory J1, J2 łączące magistralę I²C z Raspberry Pi.

Dla osób niemających styczności z Sigma DSP polecam opublikowany na łamach EP (EP 4/19...EP 11/19) kurs Audio DSP, gdzie znajdują się podstawowe informacje o środowisku Sigma Studio i procesorach Sigma DSP pominięte ze względu na ograniczoną objętość artykułu. Powodzenia w aplikacjach świata cyfrowego audio.

Adam Tatuś, EP

REKLAMA

ELPORTAL.pl

EdW możesz zamówić na

www.ulubionykiosk.pl

lub w Empikach i wszystkich większych kioskach z prasą.

Nie przegap czerwcowego wydania Elektroniki dla Wszystkich

W numerze między innymi:

Ploter – robot artysta

Co mogłoby się stać, gdyby dać komputerowi długopis do ręki? Możesz to sprawdzić! Wystarczy jedynie kilka silników i Arduino. A część mechaniczną można wykonać na zamówienie na drukarce 3D.

REWelacja, czyli precyzja komputerowej karty dźwiękowej

Opisywana niedawno przystawka pomiarowa oraz nowsze wersje programu REW pozwalają uwzględnić dodatkowe szkodliwe czynniki i mierzyć między innymi impedancję z zaskakującą dużą dokładnością

Droga do RRIO, czyli wzmacniacze operacyjne (nie tylko) dla początkujących

Zaczynamy omawiać ogromnie ważne kwestie niedoskonałości wejścia, wnętrza oraz wyjścia wzmacniaczy operacyjnych, czyli wiadomości niezbędne dla każdego prawdziwego elektronika.

Panorama audio

Technika audio gwałtownie się rozwija. Nie zawsze rozwój dotyczy polepszenia parametrów elektroakustycznych. Często polega tylko na zwiększeniu mobilności. Jak się nie zgubić w lawinie opisów i skrótów?

Sieci energetyczne dla elektroników

Sieci energetyczne prądu zmiennego 230 V 50 Hz wydają się beznadziejnie proste w porównaniu z układami elektronicznymi. Wniknięcie w szczegóły pokazuje, że wcale nie jest to takie oczywiste.

Ponadto w numerze:

- Generator losowych dźwięków
- Silniki prądu stałego
- Filozofia sieci. Protokół IP/ICMP
- Modułowe mierniki napięcia i prądu stałego
- Szkoła Konstruktorów:
 - Układ związany z awaryjnym zasilaniem zamrażarki, lodówki lub pompy obiegowej centralnego ogrzewania
 - Zaproponuj zastosowanie elektroniki do kontroli procesu kompostowania

Masz może 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

Expander 40 I/O dla Pi Zero

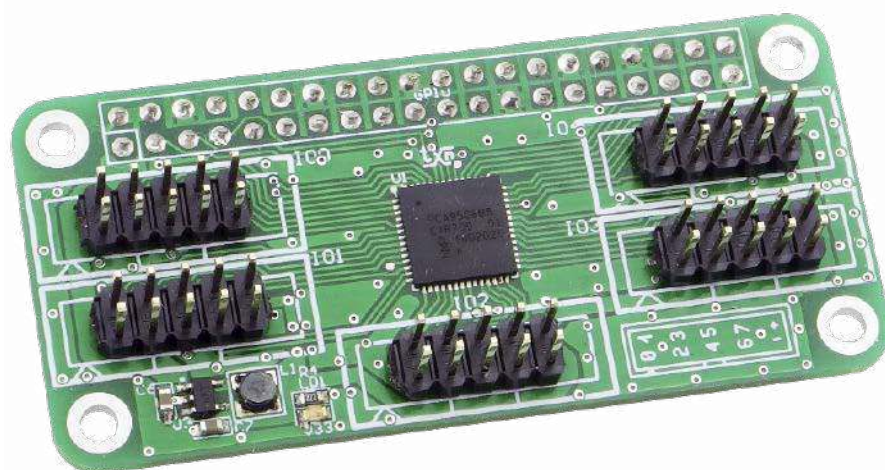


Zgodna mechanicznie z Raspberry Pi Zero nakładka z ekspanderem 40 cyfrowych portów GPIO będzie przydatna w rozbudowanych systemach monitorowania lub domowych systemach automatyki.

Nakładka umożliwia rozszerzenie liczby GPIO przy użyciu magistrali I²C. Pozwala na monitorowanie stanów wejść lub sterowanie do 40 wyjść cyfrowych o napięciu 3,3 V. Porty GPIO pogrupowane są w pięć ośmiobitowych banków 0...4, w których każde z wyprowadzeń może pełnić dowolną funkcję wejścia lub wyjścia, co zapewnia sporą elastyczność aplikacji. Dzięki prostej adresacji możliwa jest realizacja systemu do 320 punktów GPIO.

Budowa i działanie

W module zastosowano układ PCA9506BS firmy NXP, którego schemat wewnętrzny został pokazany na **rysunku 1**. Wyprowadzenia PCA9506BS w trybie wejścia przeznaczone są do współpracy z układami sterującymi typu push-pull. W przypadku monitorowania stanu wyjść typu „suchy” styk lub



otwarty kolektor/dren konieczne jest zastosowanie zewnętrznych rezystorów podciągających. Wejścia tolerują napięcia do 5 V.

Wyprowadzenia układu PCA9506BS w trybie wyjściowym przy zasilaniu 3,3 V mogą być źródłem prądu o wartości ok. 8 mA lub pochłaniać prąd ok. 12 mA, o ile nieprzekroczona jest moc strat i sumaryczny prąd wyprowadzeń zasilania (max. 500 mA). Pozwala to na bezpośrednie podłączenie i sterowanie diodami lub wyświetlaczami LED.

Schemat ideowy modułu został pokazany na **rysunku 2**. Został wyposażony w zwory adresowe dla PCA9506BS, umożliwiające ustawienie ośmiu adresów na magistrali I²C zgodnie z **tabelą 1**. Sygnały GPIO U1 wraz

z zasilaniem 3,3 V wyprowadzone są na złącza szpilkowe IDC10 oznaczone IO0...4. Ze względu na możliwy duży pobór prądu przy

Tabela 1. Adresacja układu PCA9506BS

A2	A1	A0	Adres
0	0	0	0x20 (0100 000X)
0	0	1	0x21 (0100 001X)
0	1	0	0x22 (0100 010X)
0	1	1	0x23 (0100 011X)
1	0	0	0x24 (0100 000X)
1	0	1	0x25 (0100 001X)
1	1	0	0x26 (0100 010X)
1	1	1	0x27 (0100 011X)

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

W ofercie AVT* AVT5868

Podstawowe parametry:

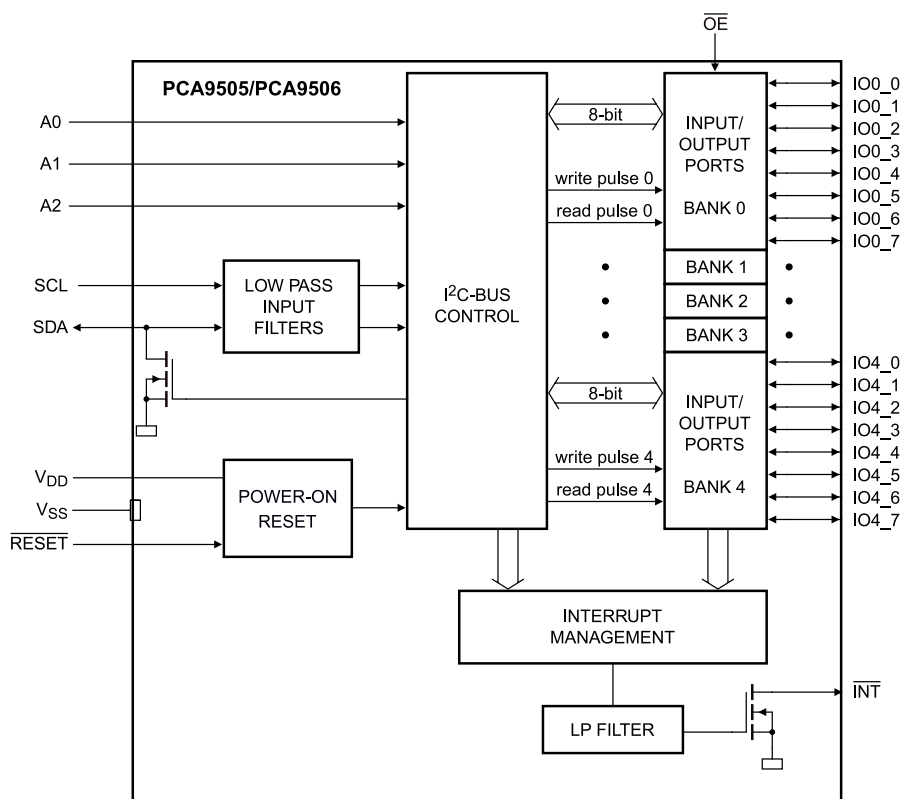
- umożliwia monitorowanie stanów wejść lub sterowanie do 40 wyjść cyfrowych o napięciu 3,3 V,
- każde z wyprowadzeń może niezależnie pełnić funkcję wejścia lub wyjścia,
- dzięki prostej adresacji możliwa jest realizacja systemu do 320 linii GPIO,
- zgodny mechanicznie z Raspberry Pi Zero.

Wykaz elementów:

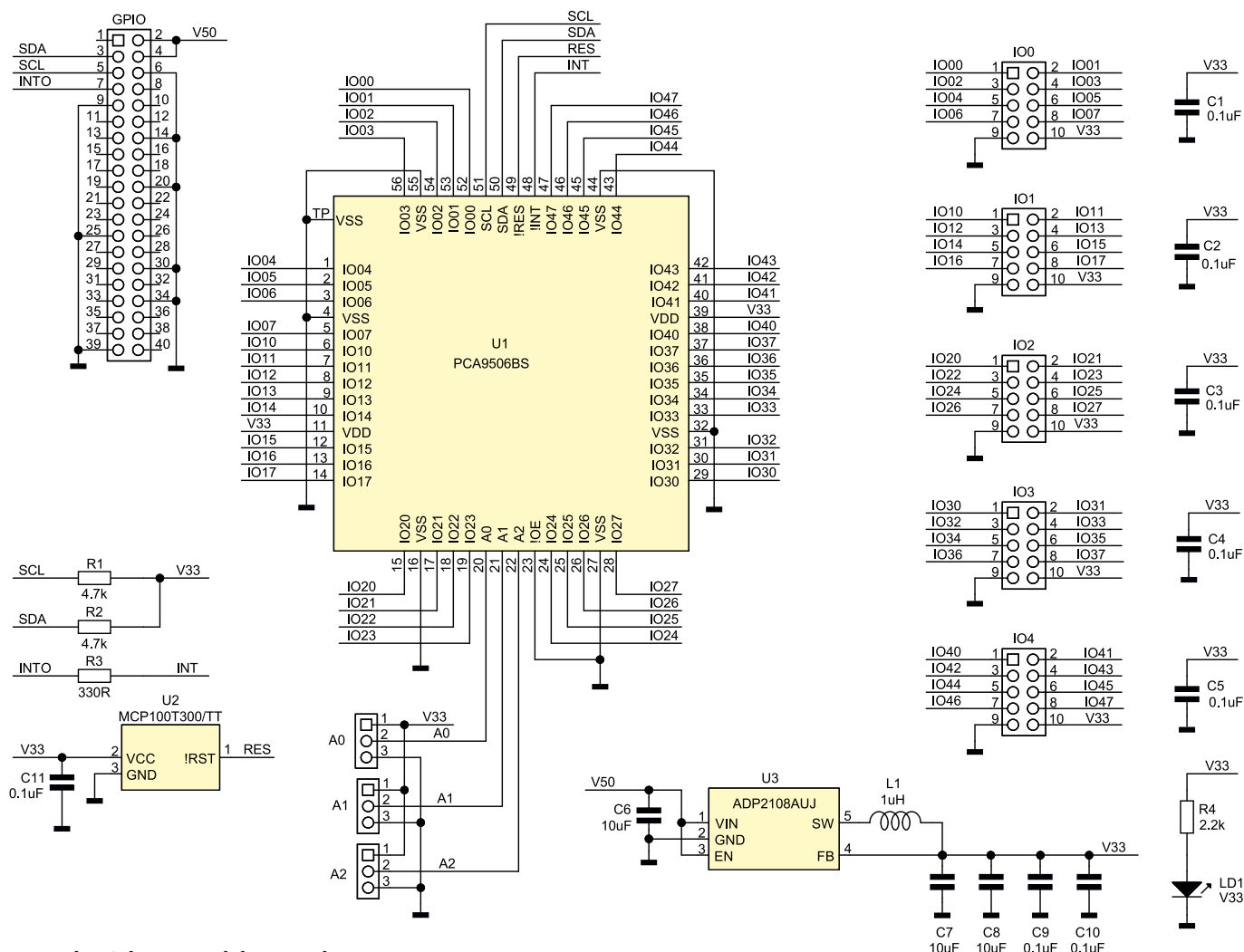
R1, R2: 4,7 kΩ (SMD0603, 1%)
 R3: 330 Ω (SMD0603, 1%)
 R4: 2,2 kΩ (SMD0603, 1%)
 C1...C5, C9, C10, C11: 0,1 μF (SMD0603)
 C6...C8: 10 μF (SMD0603)
 LD1: LED czerwony SMD0603
 U1: PCA9506BS (HVQFN56)
 U2: MCP100T300/TT (SOT-23)
 U3: ADP2108AUJZ SOT-23-5
 L1: 1 μH dławik WE-MAPI3020 (WE 74438336010)
 A0...A2: zwora PCB
 GPIO: złącze IDC 2x20 pin żeńskie
 IO0...IO4: listwa IDC 2x5 2,54 mm

*** 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 UKW) – jeśli występuje w projekcie, które należy samodzielnie wlutować w docelową 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:

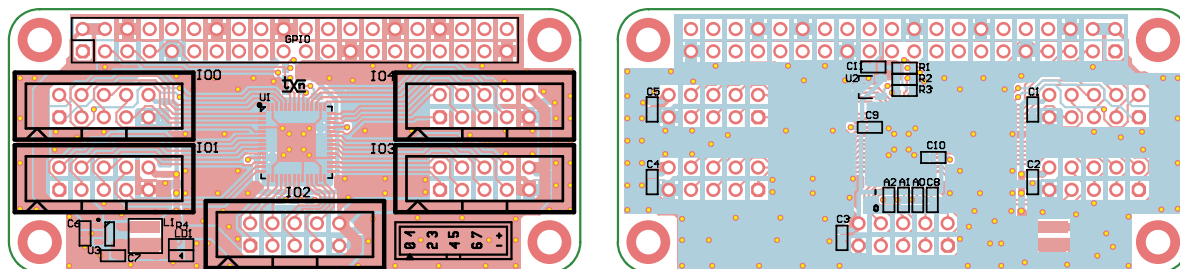
- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw
 - wersja [B] (elementy wlutowane w płytkę PCB)
 - wersja [A] – płytkę drukowaną bez elementów i dokumentacji
 - wersja [A+] – płytkę drukowaną z elementami i dokumentacją
 - wersja [UK] – zaprogramowany układ
 - wersja [UK+] – zaprogramowany układ z dokumentacją
- 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 1. Schemat wewnętrzny układu PCA9506BS



Rysunek 2. Schemat modułu expandera



Rysunek 3. Schemat płytki PCB z rozmieszczeniem elementów

Tabela 2. Rejestry konfiguracyjne układu PCA9506BS

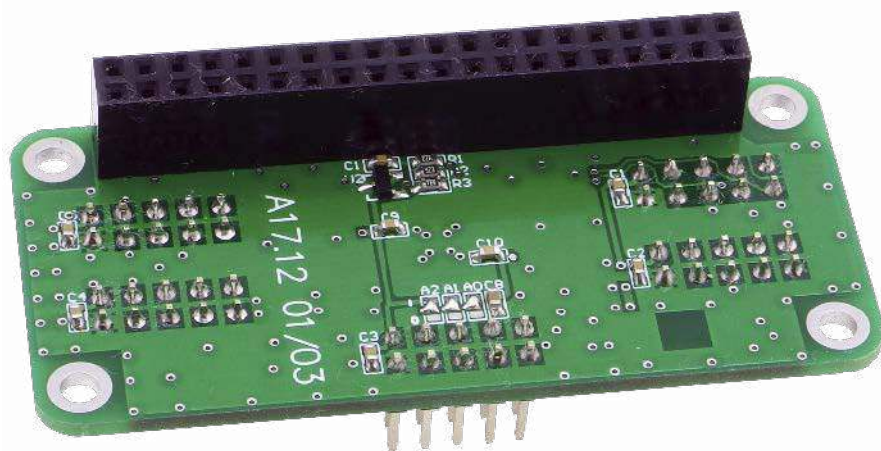
Subadres	Nazwa rejestru	Funkcja	Typ
0x00...0x04	IP0...4	Rejestr wejściowy, stan wejść GPIO0...4	odczyt
0x08...0x0C	OP0...4	Rejestr wyjściowy, stan wyjść GPIO0...4	odczyt/zapis
0x10...0x14	PI0...4	Rejestr inwersji GPIO0...4 0 – brak inwersji wejścia IPxx 1 – inwersja wejścia IPxx (domyślnie brak inwersji)	odczyt/zapis
0x18...0x1C	IOC0...4	Rejestr konfiguracji GPIO0...4 0 – ustawienie funkcji wyjścia Pxx 1 – ustawienie funkcji wejścia Pxx (domyślnie wszystkie Pxx ustawione są jako wejścia)	odczyt/zapis
0x20...0x24	MSK0...4	Rejestr maskowania przerwań GPIO0...4 0 – generowanie przerwania po zmianie stanu Pxx ustawionego jako wejście 1 – wyłączenie generowania przerwania (domyślnie przerwania wyłączone)	odczyt/zapis

sterowaniu wskaźnikami LED moduł wyposażony jest w niezależną od systemu zasilania Pi przetwornicę obniżającą U3 typu ADP2108 dostarczającą napięcie 3,3 V.

Diody LD1 sygnalizuje obecność zasilania. Układ U2 typu MCP100T-3.00 odpowiada za poprawny reset U1 po włączeniu zasilania. Rezystory R1 i R2 polaryzują magistralę I²C. Układ U1 ma możliwość opcjonalnego generowania sygnału przerwania, który poprzez rezystor R3 doprowadzony jest do GPIO4 Raspberry Pi. Jeżeli nie przewidujemy korzystania z przerwania, to nie montujemy na płycie rezystora R3.

Montaż i uruchomienie

Moduł jest zmontowany na dwustronnej płytce drukowanej, której schemat



Fotografia 1. Zmontowany moduł pokazany od strony bottom

i rozmieszczenie elementów zostało pokazane na **rysunku 3**. Sposób montażu jest klasyczny i nie wymaga opisu. Zmontowany moduł został pokazany na fotografii tytułowej oraz na **fotografii 1** od strony bottom.

Po montażu należy ustalić kropłą cyny adres modułu na zworach A2, A1, A0 zgodnie z tabelą 1. PCA9506BS ma pod adresami 0x00..0x27 pięć grup rejestrów IO oraz ustalających tryb pracy układu i konfiguracyjnych przerwan. Skrócony opis rejestrów pokazano w **tabeli 2**, pełny opis dostępny jest w karcie katalogowej.

Do szybkiego przetestowania działania modułu podłączonego do Pi użyjemy biblioteki `i2ctools`. Po ustawieniu adresu

bazowego zworami A2, A1 i A0, moduł powinien być widoczny po odczycie magistrali poleceniem:

```
i2cdetect -y 1
```

tak, jak pokazano na **rysunku 4** (ustawiony adres bazowy 0x27). Sterowanie portami i odczyt stanów może być wykonywany za pomocą prostych komend:

Stan wejść banków GPIO0...4 może być odczytany bez dodatkowej konfiguracji przy ustawieniach domyślnych PCA9506BS:

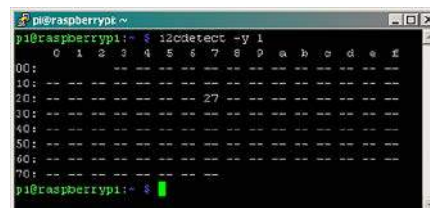
```
i2cget -y 1 0x27 0x00
```

```
..
```

```
i2cget -y 1 0x27 0x04
```

Stany wejść mogą zostać zanegowane:

```
i2cset -y 1 0x27 0x10 0xFF
```



Rysunek 4.

```
..
```

```
i2cset -y 1 0x27 0x14 0xFF
```

Ustawienie GPIO jako wyjście wykonujemy poleceniem:

```
i2cset -y 1 0x27 0x18 0x00
```

```
..
```

```
i2cset -y 1 0x27 0x1C 0x00
```

Stan wyjść zmieniamy poleceniem:

```
i2cset -y 1 0x27 0x08 0xFF
```

```
...
```

```
i2cset -y 1 0x27 0x1C 0xFF
```

Odblokowanie przerwań wykonujemy poleceniem:

```
i2cset -y 1 0x27 0x20 0x00
```

```
...
```

```
i2cset -y 1 0x27 0x24 0x00
```

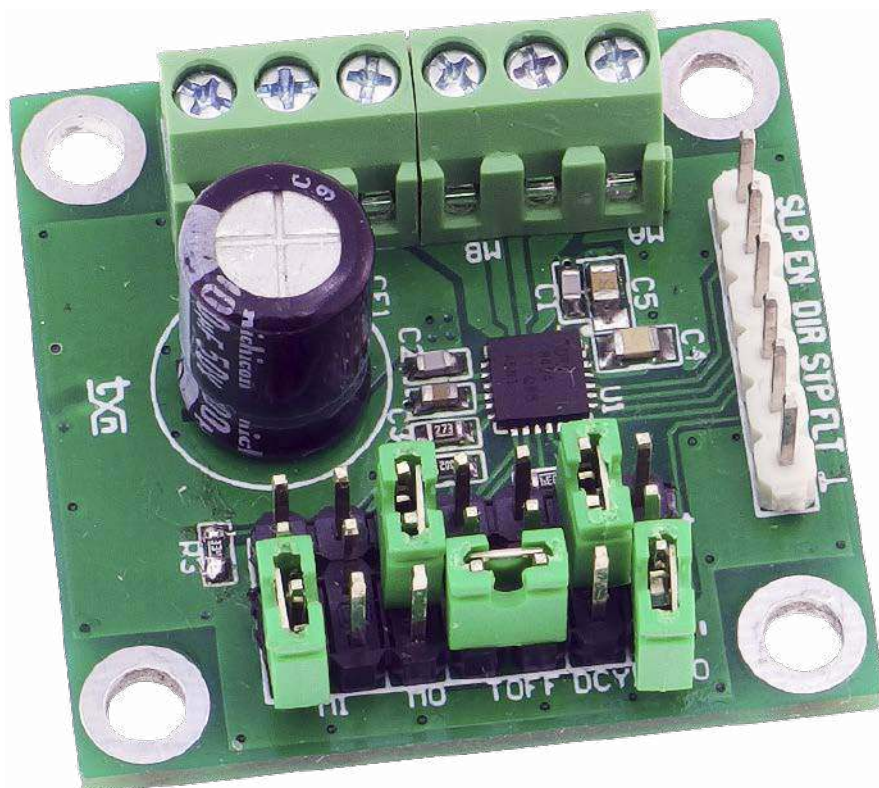
Ustawione przerwanie jest kasowane odczytem banku wejściowego 0x00..0x04 lub po przywróceniu stanu wyprowadzenia, które przerwanie wygenerowało. Jeżeli wszystko działa poprawnie, można moduł zastosować we własnej aplikacji.

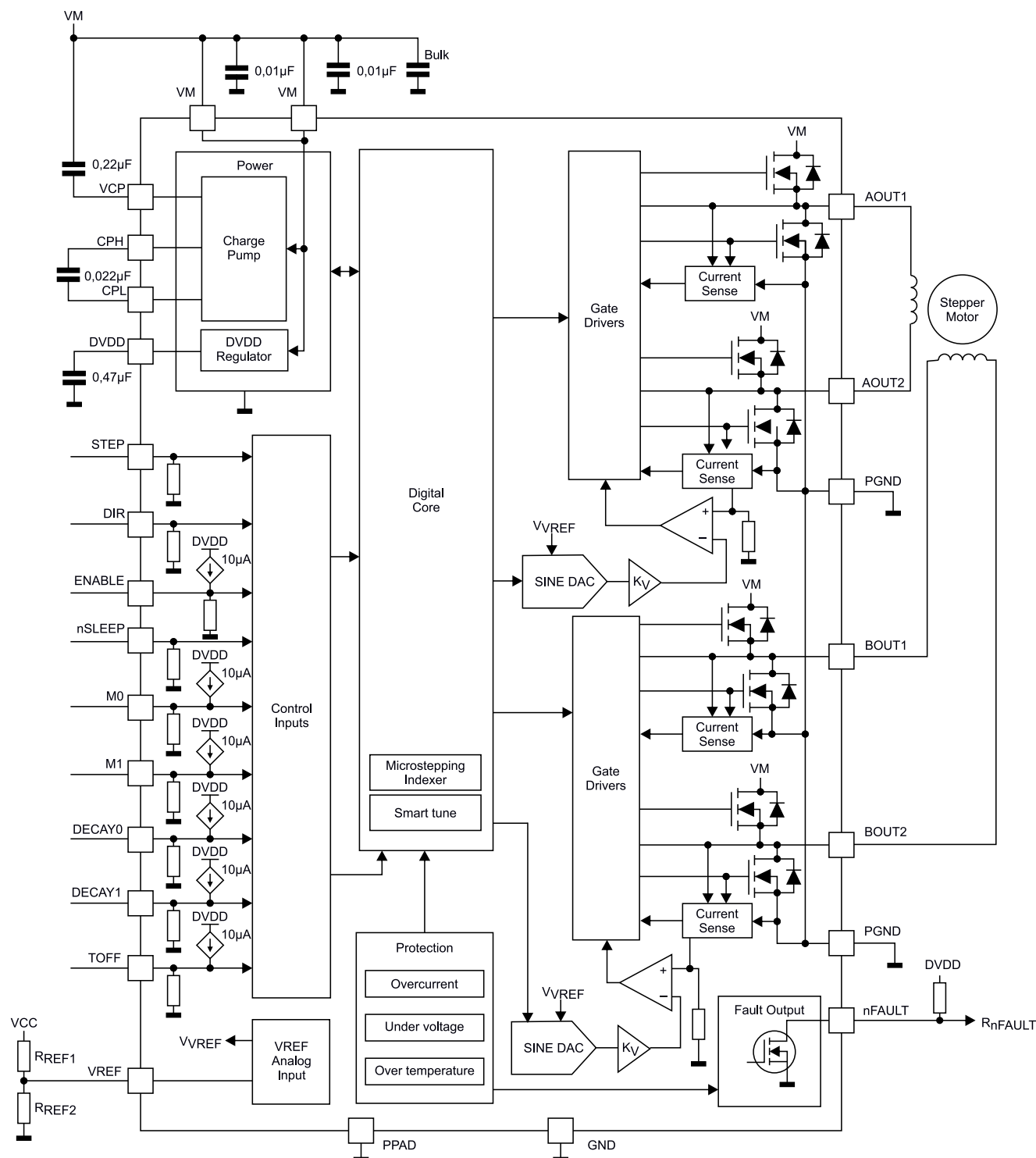
Adam Tatuś, EP

Mikrokrokový sterownik silnika

Silniki krokowe są jednymi z powszechniej stosowanych elementów elektromechanicznych. Ich prosta konstrukcja i osiągane parametry niestety okupione są dość skomplikowanym układem sterowania, szczególnie gdy wymagana jest praca z dokładnym pozycjonowaniem położenia wirnika tzw. mikro krokiem.

W ofertach producentów dostępnych jest wiele kompleksowych rozwiązań driverów silników krokowych. Jednym z najnowszych opracowań jest układ TI o oznaczeniu DRV8424, którego strukturę wewnętrzną pokazano na **rysunku 1**. Zawiera on kompletny sterownik silnika krokowego z obsługą mikrokroku, aż do 1/256 oraz stopień mocy z dwoma mostkami H na tranzystorach MOSFET. Charakteryzuje się niewielkim napięciem nasycenia i jest zdolny do dostarczenia ciągłego prądu 2,5 A na każdy mostek (w szczycie 4 A), przy napięciu zasilania w zakresie 4,5...33 V.





Rysunek 1. Schemat wewnętrzny DRV8424

Układ uzupełniono o sygnalizację stanów awaryjnych, takich jak przegrzanie, przeciążenie, zwarcie i blokadę podnapięciową, sygnalizowane na wyjściu nFAULT. Wbudowane układy pomiaru prądu uzwojeń eliminują konieczność stosowania niskoomowych rezystorów pomiarowych. Prosta i typowa aplikacja wykorzystująca sygnały STEP/DIR ułatwia zastosowanie modułu w układach elektromechanicznych, a w szczególności w robotyce amatorskiej, także podczas modernizacji, gdzie sterowanie mikro krokiem może podnieść dokładność pozycjonowania.

Budowa i działanie

Schemat modułu został pokazany na **rysunku 2**. Aplikacja układu DRV8424 jest bardzo prosta, moduł zasilany jest poprzez złącze PWR napięciem VM z zakresu 4,5...33 V, dobranym w zależności od zastosowanego silnika. Zasilanie odsprężane jest przez kondensatory CE1, C1 i C2. Należy pamiętać o odpowiedniej zewnętrznej pojemności filtrującej, zdolnej odebrać prądy hamowania z uzwojeń w docelowej aplikacji.

Z napięcia VM, poprzez wewnętrzną przetwornicę, generowane jest napięcie VCP potrzebne do zasilania drivera tranzystorów

MOSFET mostka H oraz napięcie pomocnicze DVDD. Kondensatory C4 i C5 są kondensatorami pompy ładunkowej VCP.

Uzwojenia silnika podłączone są do wyjść mostków U1 poprzez złącza MA i MB. Dzielnik R1, R2 określa napięcie odniesienia VREF dla układu regulacji prądu uzwojeń. Maksymalne dopuszczalne napięcie VREF wynosi 3,3 V. Prąd uzwojeń określony jest wzorem:

$$I_{fs} = V_{ref} / 1,32 \text{ [A]}$$

Podczas doboru dzielnika należy uwzględnić prąd wyprowadzenia Vref wynoszący ok. 8,25 µA. W modelu R1=27 kΩ, R2=10 kΩ, co ustala prąd uzwojeń na ok. 675 mA.

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

W ofercie AVT* AVT----

Podstawowe parametry:

- sterowanie standardowymi sygnałami STEP i DIR,
- obsługa mikrokoła aż do 1/256,
- dostarcza ciągłego prądu 2,5 A na każdą fazę (w szczycie 4 A),
- napięcie zasilania w zakresie 4,5...33 V.

Wykaz elementów:

C1, C2: 10 nF SMD0603
C3: 0,47 µF SMD0603
C4: 22 nF SMD0805
C5: 0,22 µF/50 V SMD0805 ceramiczny
CE1: 100 µF/50 V kondensator Low ESR 10 mm
U1: DRV8424RGER (VQFN24)
DCY0, DCY1, M0, M1, TOFF: złącze SIP3 2,54 mm + zwory
IO CONN: złącze SIP6 2,54 mm
MA MB, PWR CONN: złącze DG381-3.5-2

Uwagi! Elektroniczne zestawy do samodzielnego montażu.

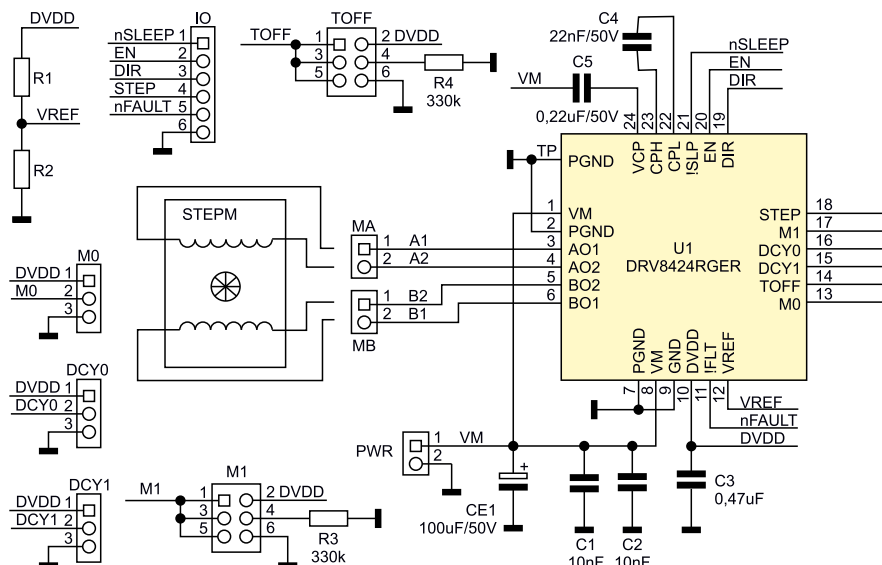
Wymagana umiejętność lutowania! Podstawową wersją zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] - jeśli występuje w projekcie), które należy samodzielnie wzlutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje 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: kit@avt.pl.



Rysunek 2. Schemat ideowy modułu DRV8824

Tabela 1. Konfiguracja trybu sterowania mostka

DCY0	DCY1	Zwora DCY0	Zwora DCY1	INCREASING STEPS	DECREASING STEPS
0	0	2-3	2-3	Smart tune Dynamic Decay	Smart tune Dynamic Decay
0	1	2-3	1-2	Smart tune Ripple Control	Smart tune Ripple Control
1	0	1-2	2-3	Mixed decay: 30% fast	Mixed decay: 30% fast
1	1	1-2	1-2	Slow decay	Mixed decay: 30% fast
Hi-Z	0	NC	2-3	Mixed decay: 60% fast	Mixed decay: 60% fast
Hi-Z	1	NC	1-2	Slow decay	Slow decay

Tabela 2. Konfiguracja indeksa

M0	M1	Zwora M0	Zwora M1	Tryb mikrokoła
0	0	2-3	5-6	1/1 100% I _{fs}
0	330 kΩ GND	2-3	3-4	1/1 71% I _{fs}
1	0	1-2	5-6	1/2 Non-circular (high torque at high RPM)
Hi-Z	0	NC	5-6	1/2
0	1	2-3	1-2	1/4
1	1	1-2	1-2	1/8
Hi-Z	1	NC	1-2	1/16
0	Hi-Z	2-3	NC	1/32
Hi-Z	330 kΩ GND	NC	3-4	1/64
Hi-Z	Hi-Z	NC	NC	1/128
1	Hi-Z	1-2	NC	1/256

Tabela 3. Konfiguracja układu kontroli tętnień

TOFF	Zwora TOFF	Current Ripple at a specific microstep
0	5-6	19 mA + 1% of ITRIP
1	1-2	19 mA + 2% of ITRIP
Hi-Z	NC	19 mA + 4% of ITRIP
330 kΩ GND	3-4	19 mA + 6% of ITRIP

Układ DRV8424 akceptuje zmiany nastaw indeksa i trybu pracy w locie, co ułatwia elastyczne sterowanie silnikiem.

Montaż i uruchomienie

Minimoduł zmontowany jest na dwustronnej płytce drukowanej, której schemat wraz

z rozmieszczeniem elementów został pokazany na **rysunku 3**. Sposób montażu jest klasyczny i nie wymaga opisu, należy tylko pamiętać, aby poprawnie przylutować pad termiczny.

Sprawdzenie układu po doborze dzielnika R1, R2 odpowiadającego

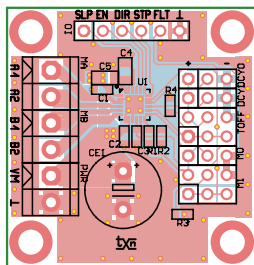
Podczas doboru prądu należy pamiętać, że maksymalny osiągnięty rzeczywisty prąd może zostać ograniczony dopuszczalną mocą strat i temperaturą struktury DRV8424.

Na złącze IO wyprowadzone są sygnały sterujące pracą układu U1:

- nSLEEP – stan niski wprowadza układ w tryb uśpienia i kasuje błędy;
- EN – stan wysoki aktywuje wyjścia układu;
- DIR – określa kierunek obrotów silnika;
- STEP – wejście impulsów kroków aktywne zboczem narastającym. Maksymalna częstotliwość taktowania to 500 kHz. Maksymalna wartość rzeczywista częstotliwości określona jest parametrami silnika i obciążenia;
- nFAULT – wyjście typu OD sygnalizujące błąd układu.

Układ sterowania układu DRV8424 konfigurowany jest zworami:

- DCY0/1 – określającymi tryb sterowania mostkiem H wpływającym na szybkość zaniku prądu uzwojeń w trybie Smart/Fast/Slow/Mixed lub w trybie kontroli tętnień, zgodnie z **tabelą 1**. Odpowiednie sterowanie tranzystorami mostka H umożliwia zanik prądu uzwojeń w trybie przeciwnapędu (Fast – reverse), recyrkulacji (Slow – brake) lub mieszanym (30/60% Fast);
- M0/M1 – określającymi tryb pracy indeksa, zgodnie z **tabelą 2**;
- TOFF – określającą tryb kontroli tętnień prądu uzwojeń, zgodnie z **tabelą 3**.



Rysunek 3. Schemat płytki PCB z rozmieszczeniem elementów

znamionowemu prądowi uzwojeń silnika i zapewnieniu odpowiednio wydajnego zasilania sprowadza się do ustawienia zwór DCY0/1, M0/M1, TOFF określających tryb pracy indeksera i układu sterowania mostkiem H. Po doprowadzeniu do złącza IO sygnału STEP z regulowanego generatora sygnału prostokątnego (3,3...5 V), określeniu kierunku obrotów DIR, aktywacji układu poprzez ustalenie stanu wysokiego EN i nSLEEP silnik powinien zacząć się

obracać, o ile częstotliwość sygnału STEP i ustawianie indeksa nie przekraczają maksymalnej prędkości obrotowej silnika.

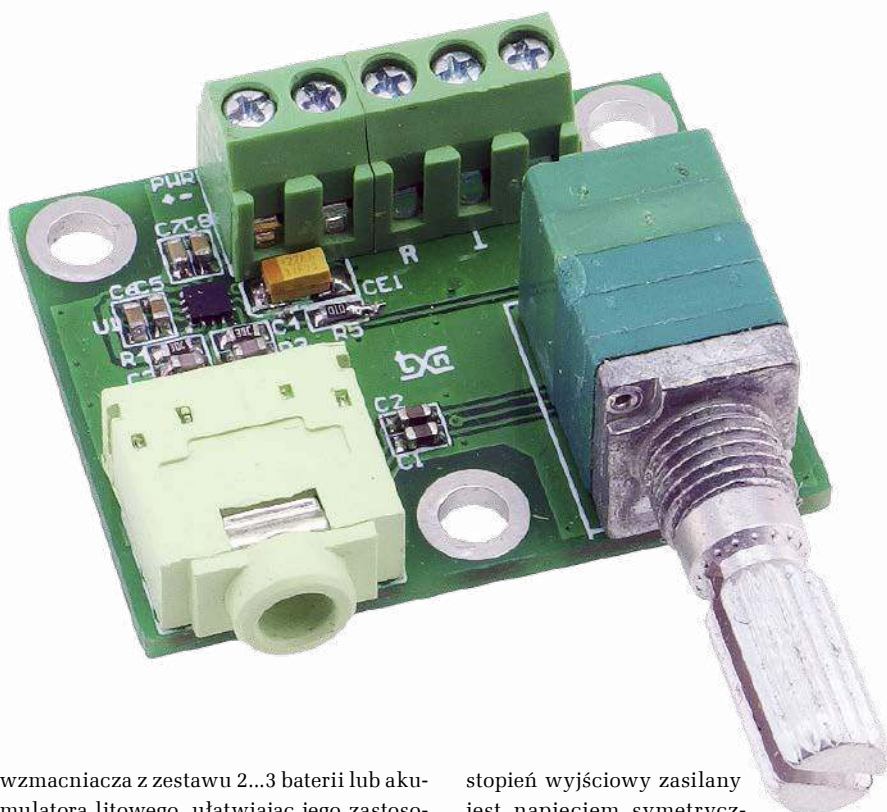
Pomimo prostego sterowania układem sterownika polecam zapoznanie się z kartą katalogową DRV8424 oraz notami aplikacyjnymi, gdzie w najdrobniejszych szczegółach opisane są tryby sterowania, które w artykule z oczywistych przyczyn opisane są w sposób skrócony.

Adam Tatuś, EP

Miniaturowy wzmacniacz słuchawkowy

Miniaturowy wzmacniacz słuchawkowy z zasilaniem bateryjnym lub akumulatorowym jest przydatny do podłączenia słuchawek niskomomowych do urządzeń wyposażonych tylko w wyjście liniowe.

Wzmacniacz bazuje na układzie IS31AP4912 firmy ISSI, którego schemat wewnętrzny został pokazany na rysunku 1. Najważniejszą cechą układu jest niskie i niesymetryczne napięcie zasilania mieszczące się w zakresie 2,7...5,5 V, co umożliwia zasilanie



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

W ofercie AVT* AVT----

Podstawowe parametry:

- moc wyjściowa dochodzi do 30 mW przy 32 Ω,
- wyłącznik sprzężony mechanicznie z potencjometrem RV1 automatycznie wprowadza układ w tryb obniżonego poboru mocy,
- napięcie zasilania mieści się w zakresie 2,7...5,5 V, co umożliwia zasilanie wzmacniacza z zestawu 2...3 baterii lub akumulatora litowego.

Wykaz elementów:

R1, R2, R3, R4: 20 kΩ SMD0603 1%
 R5: 100 kΩ SMD0603 1%
 RV1: potencjometr stereo z wyłącznikiem 20 kΩ log. PTD90SS P
 C1, C2: 1 μF SMD0603
 C3, C4: 10 pF SMD0603
 C5, C6: 2,2 μF SMD0603
 C7: 4,7 μF SMD0603
 C8: 0,1 μF SMD0603
 CE1: 22 μF/10 V kondensator tantalowy A SMD3216
 U1 IS31AP4912 (UTQFN-12)
 HP: gniazdo słuchawkowe mini jack 3,5 mm stereo (FC68133)
 IN: złącze 3 piny 3,5 mm DG381-3.5-3
 PWR: złącze 2 piny 3,5 mm DG381-3.5-2

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

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:

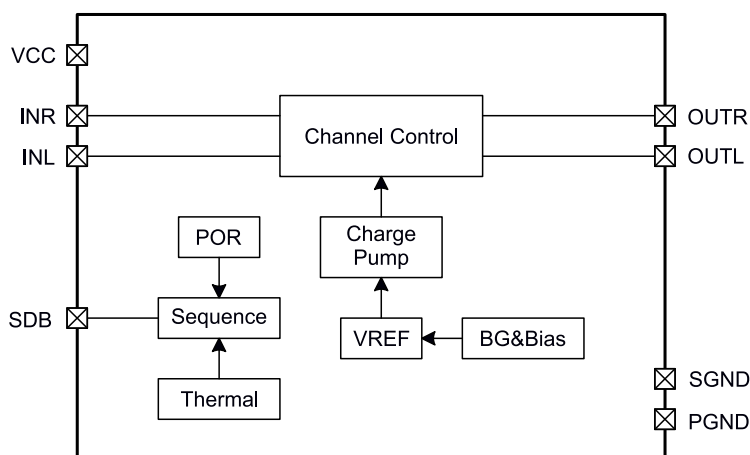
- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wzlutowane w płytkę PCB)
- wersja [A] – płytkę drukowaną bez elementów i dokumentacji Kity w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- wersja [Av] – płytkę drukowaną [A]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

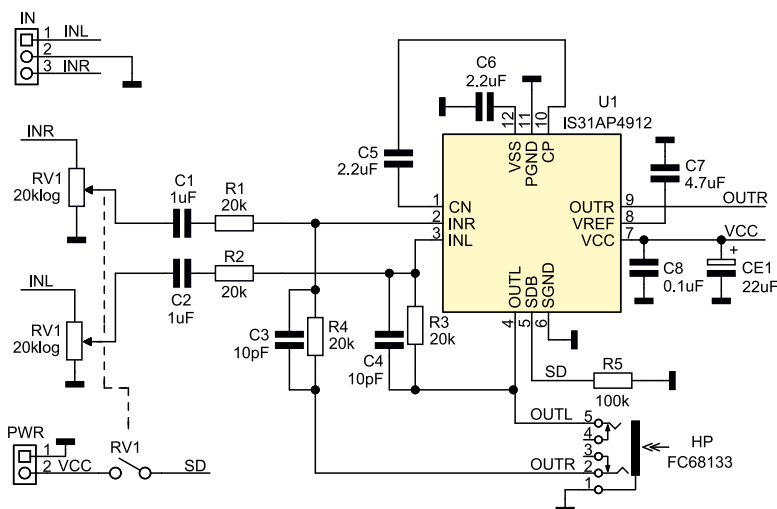
<http://sklep.avt.pl>. W przypadku braku dostępności na <http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.

wzmacniacza z zestawu 2...3 baterii lub akumulatora litowego, ułatwiając jego zastosowania przenośne. W aplikacji IS31AP4912 zbędne są kondensatory sprzęgające stopień wyjściowy ze słuchawkami, co pozwala zachować niewielki rozmiar aplikacji. Liniowy

stopień wyjściowy zasilany jest napięciem symetrycznym, uzyskiwanym z wbudowanej pompy ładunkowej. Wzmacniacz nie wymaga stosowania dławików lub cewek niezbędnych we wzmacniaczach klasy D.



Rysunek 1. Schemat wewnętrzny IS31AP4912 (za notą ISSI)



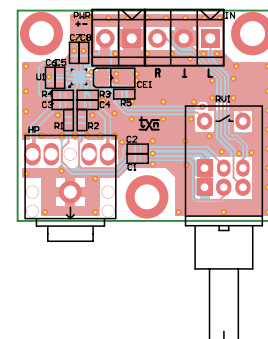
Rysunek 2. Schemat ideowy wzmacniacza

Budowa i działanie

Schemat modułu wzmacniacza słuchawkowego został pokazany na **rysunku 2**. Stereofoniczny sygnał wejściowy ze złącza IN doprowadzony jest poprzez potencjometr RV1, kondensatory sprzęgające C1, C2, określające dolną częstotliwość przenoszenia do układu U1. Rezystory R1, R3 określają wzmocnienie (R2, R4 dla kanału prawego). Kondensatory zapewniają C3, C4 stabilność wzmacniacza i ograniczają górną częstotliwość przenoszenia. Kondensatory C5, C6 są elementami pompy ładunkowej, pracującej

z częstotliwością ok. 250 kHz, pozostałe kondensatory odsprężają zasilanie.

Wzmocniony sygnał wyprowadzony jest na gniazdo HP typu mini jack 3,5 mm. Zasilanie 2,7...5,5 V z baterii lub akumulatora musi być doprowadzone do złącza PWR. Wyłącznik sprzężony mechanicznie z potencjometrem RV1 umożliwia wprowadzenie U1 w tryb obniżonego poboru mocy, gdy jest rozwarty i wejście SD=0 lub aktywuje układ, gdy jest zwarty, zmieniając stan wyprowadzenia SD na wysoki. W trybie obniżonego poboru mocy prąd pobierany ze źródła wynosi ok. 1 μ A.



Rysunek 3. Rozmieszczenie elementów płytki wzmacniacza

Wzmacniacz ma zabezpieczenie termiczne oraz eliminatory stanów nieustalonych. Moc wyjściowa może dochodzić do 30 mW przy 32 Ω w zależności od napięcia zasilania. Zniekształcenia wynoszą poniżej 1%, a odstęp od zakłóceń jest >90 dB, co jest wartością całkiem przyzwoitą.

Montaż i uruchomienie

Układ zmontowany jest na dwustronnej płytce drukowanej, montaż jest standardowy i nie wymaga opisu. Schemat płytki wraz z rozmieszczeniem elementów został pokazany na **rysunku 3**. Układ nie wymaga uruchamiania, a zmontowany ze sprawdzonych elementów działa po włączeniu zasilania.

Adam Tatuś

adam.tatus@ep.com.pl

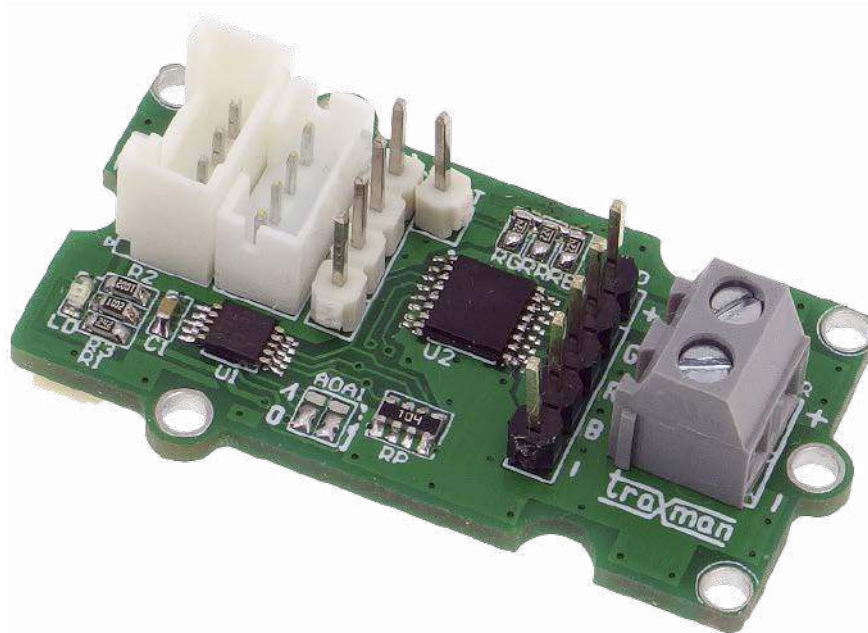
Sterownik LED RGB z układem AL1783 sterowany przez I²C

Prezentowany moduł rozszerza możliwości płytek uruchomieniowych o sterowanie diodami LED RGB poprzez magistralę I²C. Moduł pracuje w trybie prądowym i umożliwia indywidualną regulację każdego wyjścia (barwy), a dodatkowo pozwala na monitorowanie stanów awaryjnych.

Moduł wymaga zasilania dla dołączonych LED mieszczącego się w szerokim zakresie 6,5...60 V, podłączanego poprzez złącze PWR. To pozwala na sterowanie dosyć długimi łańcuchami szeregowymi (podłączonymi do potencjału VLED przez złącze LED), o ile nie zostanie przekroczona moc strat w układzie.

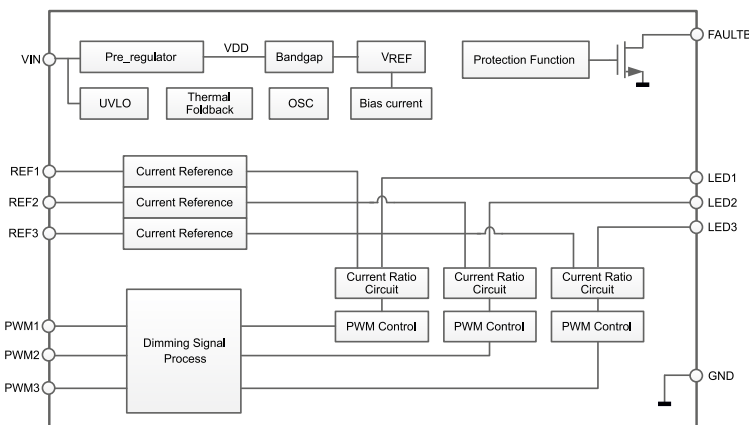
Budowa i działanie

Moduł bazuje na specjalizowanym, linio- wym, 3-kanałowym sterowniku LED typu AL1783 firmy Diodes, którego struktura wewnętrzna została pokazana na **rysunku 1**. Schemat modułu jest pokazany na **rysunku 2**.

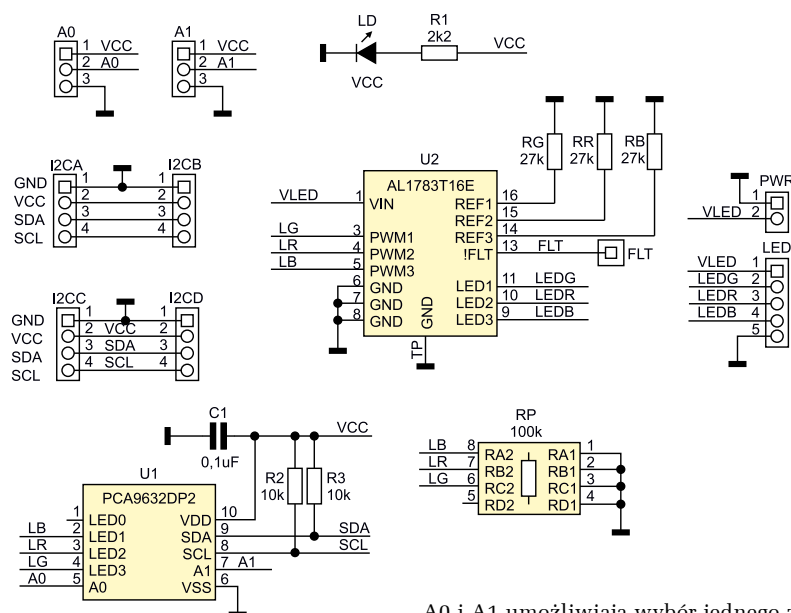


Układ AL1783 ustala niezależnie prąd diod w zakresie 20...250 mA, poprzez wartości dołączonych rezystorów RG, RR i RB, zgodnie

z charakterystyką z **rysunku 3**. Dla zastosowanych wartości 27 k Ω prąd ustawiony jest na 220 mA/kanał.



Rysunek 1. Struktura wewnętrzna układu AL1783



Rysunek 2. Schemat ideowy modułu

Sygnał FLT sygnalizuje stanem niskim przerwę lub zwarcie w aktywnym łańcuchu LED oraz przekroczenie temperatury układu lub zbyt niskie napięcie zasilania (<5,8 V). Wejścia PWM umożliwiają sterowanie układem U2 sygnałem ze zmiennym wypełnieniem regulującym jasność świecenia.

Jako generator PWM został zastosowany układ U1 typu PCA9632DP. Jest on konfigurowany poprzez magistralę I²C, zwory

A0 i A1 umożliwiają wybór jednego z czterech adresów 0x60...0x63. Magistrala doprowadzona jest do powielonych złączy I2Cx ułatwiających podłączenie do różnych standardów spotykanych na płytach uruchomieniowych (Grove, QWIIC, SIP 2,54 mm, PH 2 mm).

Montaż i uruchomienie

Montaż modułu przebiega w sposób klasyczny i nie wymaga dokładnego opisu. Schemat płytki PCB wraz z rozmieszczeniem elementów został pokazany na **rysunku 4**.

Listing 1. Przykładowy skrypt testowy

```
#!/bin/bash
echo "AL1783 RGB Driver ADR:0x60"

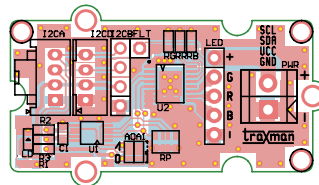
i2ctransfer -y 1 w13@0x60 0x80 0x80 0x14 0x00 0x00 0x00 0x00 0x80 0x40 0xAA 0xES

echo "Blue"
i2cset -y 1 0x60 0x03 0x55
sleep 1
i2cset -y 1 0x60 0x03 0x00
sleep 1

echo "Green"
i2cset -y 1 0x60 0x04 0x55
sleep 1
i2cset -y 1 0x60 0x04 0x00
sleep 1

echo "Red"
i2cset -y 1 0x60 0x05 0x55
sleep 1
i2cset -y 1 0x60 0x05 0x00
sleep 1

echo "All Off"
```



Rysunek 4. Schemat płytki PCB wraz z rozmieszczeniem elementów

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

W ofercie AVT* AVT----

Podstawowe parametry:

- umożliwia ustawienie prądu zasilania każdej struktury LED w zakresie 20...250 mA,
- każda barwa może być indywidualnie regulowana sygnałem PWM,
- sygnalizuje stany awaryjne,
- umożliwia dołączenie długich łańcuchów LED dzięki dopuszczalnemu napięciu maksymalnemu aż 60 V.

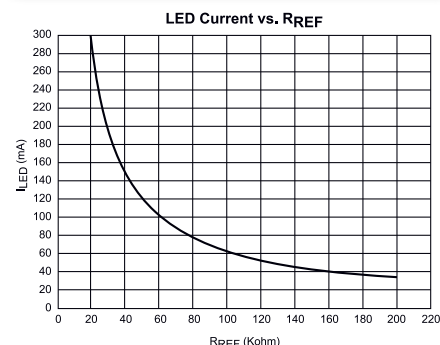
Wykaz elementów:

R1: 2,2 kΩ SMD0603 1%
 R2, R3: 10 kΩ SMD0603 1%
 RB, RG, RR: 27 kΩ SMD0603 1% (dobrać)
 RP: 100 kΩ drabinka rezystorowa CRA06S08
 C1: 0,1 μF SMD0603
 LD: dioda LED SMD0603
 U1: PCA9632DP2 (TSSOP10)
 U2: AL1783T16E (TSSOP16)
 FLT: złącze SIP1 2,54 mm
 I2CB: złącze SIP4 2,54 mm
 I2CC: złącze JST 1 mm
 LED: złącze SIP5 lub kielichowe
 PWR: złącze DG381-3.5-2

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] – płytka drukowana bez elementów i dokumentacji
- wersja [A+] – płytka drukowana z zestawem wymagającym 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.



Rysunek 3. Dobór prądu diody

W przypadku większych prądów warto układ U2 wyposażyć w niewielki radiator.

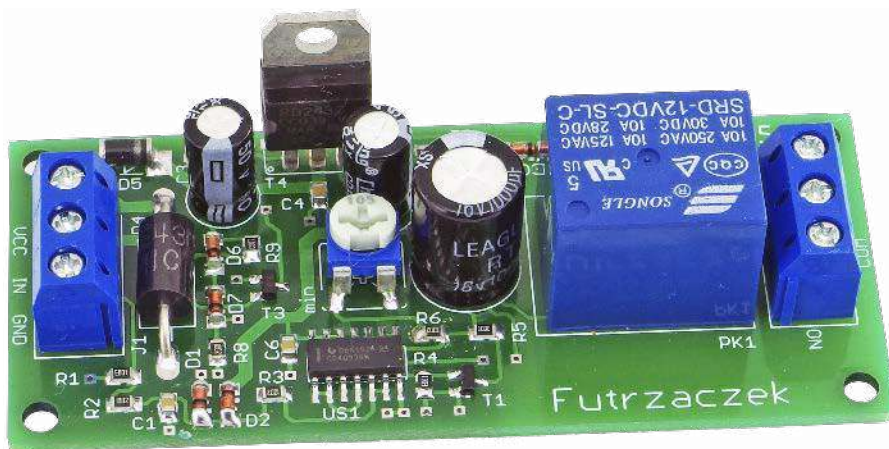
Poprawnie zmontowany moduł nie wymaga uruchamiania, należy dobrać tylko rezystory określające prąd LED. Dla sprawdzenia działania układu z Raspberry Pi przygotowano przykładowy skrypt testowy *AL1783.sh*, którego treść została pokazana na **listingu 1**. Dokładny opis rejestrów układu PCA9632 jest w nocie katalogowej.

Po podłączeniu modułu do Raspberry Pi i uruchomieniu skryptu zostaną zaświecone kolejno diody B, G, r. Jasność LED można ustawić, zmieniając wartość 0x55 na mniejszą lub większą niezależnie dla każdego koloru. Wartość 0x00 wygasza LED.

Adam Tatus
adam.tatus@ep.com.pl

Wyłącznik zasilania z opóźnieniem

Niektóre urządzenia wymagają wydłużenia czasu ich działania w porównaniu z resztą systemu. Może to być np. oświetlenie zewnętrzne, które działa jeszcze przez jakiś czas od wyłączenia stacyjki w pojeździe, czy też wentylator chłodzący wnętrze rozgrzanej maszyny jeszcze przez pewien czas od jej wyłączenia. Odcięcie dopływu prądu powinno nastąpić wtedy automatycznie.



Niektóre urządzenia muszą działać nieco dłużej niż pozostałe elementy systemu. Odcięcie dopływu prądu powinno nastąpić wtedy automatycznie, gdyż nikt nie zamierza wracać (ani tym bardziej czekać) do miejsca pracy tylko po to, aby przełączyć jeden przełącznik.

To jest zadanie, w którym sprawdzi się prezentowany układ – załączy obciążenie zaraz po podaniu sygnału napięciowego i wyłączy je po określonym czasie od jego zaniku.

Budowa

Schemat układu został pokazany na rysunku 1. Jedynymi elementami aktywnymi są trzy bramki NAND oraz dwa tranzystory, więc jego analiza nie będzie skomplikowana.

Wejście sygnału sterującego (linia TRIG) jest ściągane do masy rezystorem R1. Diody D1 i D2 ograniczają jego wartość do takiego poziomu, aby nie spowodować uszkodzenia następnych obwodów. Rezystor R2 ogranicza prąd tych diod, a w połączeniu z kondensatorem C2 stanowi prosty filtr dolnoprzepustowy, który eliminuje szybkozmienne zakłócenia, mogące przypadkowo wyzwolić układ.

Rezystor R3 ogranicza prąd diod zabezpieczających wbudowanych w strukturę bramki US1B. Jej wejścia zostały połączone ze sobą, więc zachowuje się jak odwracający przerzutnik Schmitta. Bramka US1A pełni funkcję zwykłego negatora, aby cały ten człon nie odwracał stanu logicznego.

Wystawienie stanu wysokiego przez wyjście bramki US1A powoduje załączenie tranzystora T1. Wówczas on, wymuszając przepływ prądu drenu, rozładowuje kondensator C2. Rezystor R5 ogranicza szybkość tego rozładowywania, aby przez T1 nie popłynął zbyt wysoki prąd chwilowy, powodujący jego uszkodzenie. Układ w tym stanie może trwać dowolnie długo. Napięcie na C2 będzie wtedy teoretycznie zerowe, a w praktyce nieco większe od zera, rzędu kilkunastu–kilkudziesięciu miliwoltów.

To wynika z dzielnika napięciowego, jaki tworzą połączone szeregowo R6 z P1 oraz R5 z R_{DSon} tranzystora T1.

Bramka US1C sprawdza napięcie na okładkach C2. Jeżeli jest ono dostatecznie niskie, interpretuje je jako stan niski i załącza tranzystor T2, który zasila cewkę przekaźnika. Zatem styki przekaźnika zostają przełączone niemal natychmiast po pojawieniu się sygnału sterującego. Po zaniku sygnału wyzwalającego potencjał bramki T1 spadnie do zera, czyli zostanie on zatkany. Rezystor R6 i potencjometr P1 będą powoli ładowały kondensator C2 – tym wolniej, im większa jest ich wypadkowa rezystancja. Ładowanie będzie trwało bardzo długo, ale najistotniejszy jest moment, w którym bramka US1C uzna potencjał jej wejść za logiczną „1”. Nastąpi wtedy wyłączenie cewki przekaźnika. C2 będzie się ładował dalej, aż do napięcia zasilającego tę część układu, ale nie ma to znaczenia. Przekaźnik pozostanie już wyłączony aż do następnego rozładowania C2.

Napięcie zasilania układu może mieścić się w szerokich granicach, a na dodatek może być ono silnie zakłócone. Dlatego stabilizator napięcia został zbudowany z kilku elementów dyskretnych, co ma zwiększać odporność na zakłócenia. Dodatkowo, na jego wejściu znalazł się dwukierunkowy transil D4, obcinający impulsy napięcia przekraczające wartość 43 V.

Idea działania tego stabilizatora jest bardzo prosta: dioda Zenera D8 jest spolaryzowana zaporowo, więc na jej zaciskach pojawia się napięcie około 13 V. Tranzystor T4 pełni funkcję wtórника napięciowego, zatem potencjał jego emitera wyniesie około 12,3 V. Pozostaje kwestia zapewnienia prawidłowej polaryzacji tej diody, co również nie jest trudne: źródło prądowe na tranzystorze T3 zapewnia jej prąd o natężeniu 6...7 mA. Diody D6 i D7 polaryzują jego bazę potencjałem około 1,4 V niższym od napięcia zasilającego, więc spadek napięcia na R9

jest zbliżony do spadku napięcia na pojedynczym, przewodzącym złączu p-n, czyli około 0,7 V. Rezystor R8 polaryzuje diody D6 i D7 w kierunku przewodzenia.

Montaż i uruchomienie

Układ został zmontowany na dwustronnej płytce drukowanej o wymiarach 80×35 mm. Jej wzór ścieżek oraz schemat montażowy

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

W ofercie AVT* AVT5867

Podstawowe parametry:

- podtrzymanie działania przekaźnika przez zadany czas od zaniku sygnału sterującego,
- płynna regulacja czasu podtrzymania od 11 s do 15 min,
- sygnał sterujący: dodatnie napięcie stałe o wartości 10 V...45 V,
- napięcie zasilania 18 V...43 V,
- przystosowany do zasilania z instalacji 24 V np. w pojazdach ciężarowych.

Wykaz elementów:

R1, R4, R7: 100 kΩ SMD0805
R2, R3, R6: 10 kΩ SMD0805
R5, R9: 100 Ω SMD0805
R8: 15 kΩ SMD0805
P1: 1 MΩ montażowy leżący
C1, C4, C6: 100 nF SMD0805
C1: 1000 µF/16 V raster 5 mm
C3, C5: 47 µF/50 V raster 2,54 mm
D1...D3, D6, D7: 1N4148 MiniMELF
D4: 1.5KE43CA (opis w tekście)
D5: SM4007 MELF lub podobna
D8: BZV55C13 MiniMELF
T1, T2: BSS123
T3: BC856 lub podobny
T4: BD234C lub podobny
US1: CD4093 S014
J1, J2: ARK3/500
PK1: JQC3FF 12 V
Radiator (opis w tekście)

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 projekcji), 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 [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 [A*] – płytkę drukowaną [A] + zaprogramowany układ [UK] i dokumentację
 - 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.

zostały pokazane na **rysunku 2**. W odległości 3 mm od krawędzi płytki znalazły się otwory montażowe o średnicy 3,2 mm.

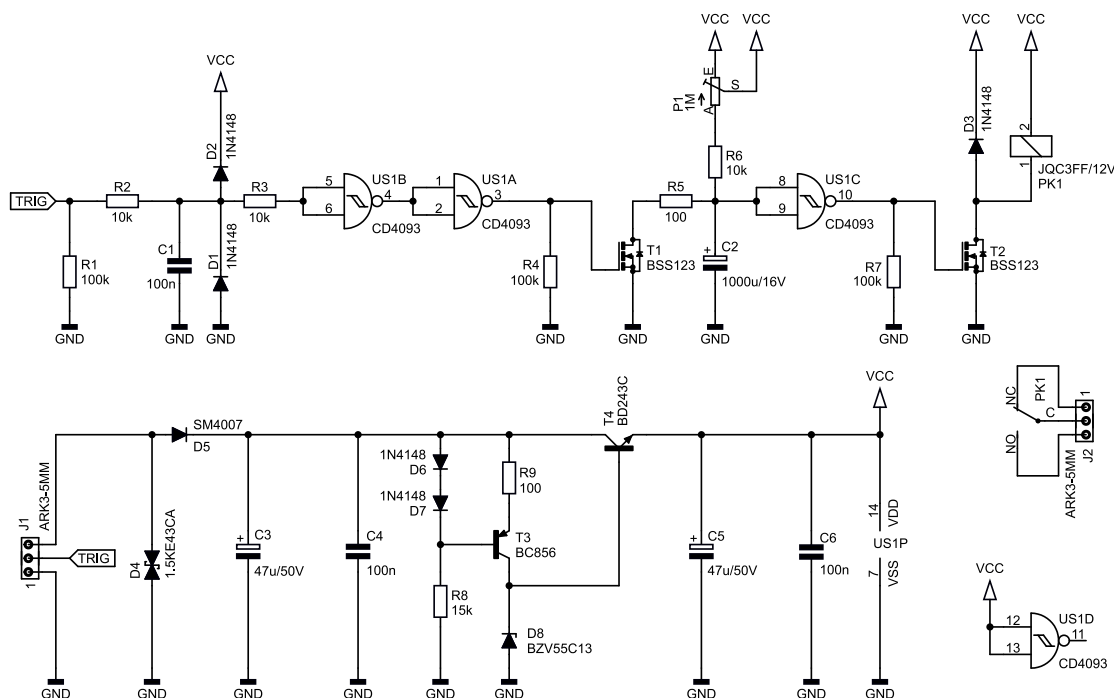
Montaż należy rozpocząć od przylutowania elementów w obudowach SMD. Znajdują się tylko na wierzchniej stronie płytki. Potem można przejść do wlutowania podzespołów w obudowach przewlekanych, według ich rosnącej wysokości.

Prawidłowo zmontowany układ jest gotowy do działania. Jediną czynnością uruchomieniową jest ustawienie potencjometru P1 w położeniu, które zapewni odpowiedni czas opóźnienia wyłączenia przełącznika PK1. W układzie prototypowym minimalna zmierzona wartość tego czasu to 11 s, a najdłuższa 15 min. Należy mieć na uwadze, że temperatura otoczenia będzie miała wpływ na ten parametr, ponieważ zmienia się wtedy pojemność elektrolitycznego kondensatora C2.

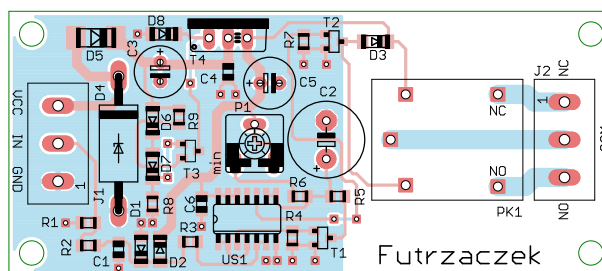
Podłączenie układu nie powinno nastręczać trudności: do złącza J1 należy doprowadzić zasilanie obecne cały czas (VCC) oraz masę (GND). Sygnał sterujący, czyli np. napięcie pochodzące ze stacyjki samochodowej, powinno być podłączone między zacisk IN a masę. Przełącznik zostanie zwarty (zwarzenie styków NO i COM) natychmiast po pojawieniu się sygnału sterującego, a rozwarzenie (zwarzenie styków NC i COM) nastąpi po upływie określonego czasu od jego zaniku.

Nie ma przeciwwskazań do tego, aby zmienić potencjometr P1 na inny, o mniejszej rezystancji, jeżeli wymagane jest opóźnienie rzędu pojedynczych sekund. W celu zapewnienia większej stabilności temperatury można pokusić się o zmianę kondensatora C2 na tantalowy, pamiętając o takim samym (lub wyższym) napięciu znamionowym, jak w przypadku oryginalnego, elektrolitycznego.

Pobór prądu ze źródła zasilania wynosi 9 mA przy wyłączonym przełączniku oraz 38 mA po jego załączeniu. Są to wartości zmierzone w układzie prototypowym przy napięciu 24 V. Napięcie zasilania może się



Rysunek 1. Schemat ideowy układu



Rysunek 2. Schemat montażowy i wzór ścieżek płytki PCB

wahać od 18 V do 43 V. Tak szeroki zakres czyni układ odpornym na zakłócenia, które mogą się przenosić po instalacji elektrycznej, na przykład w samochodzie ciężarowym. Przy napięciu stale przekraczającym 35 V do tranzystora T4 warto przykręcić niewielki radiator.

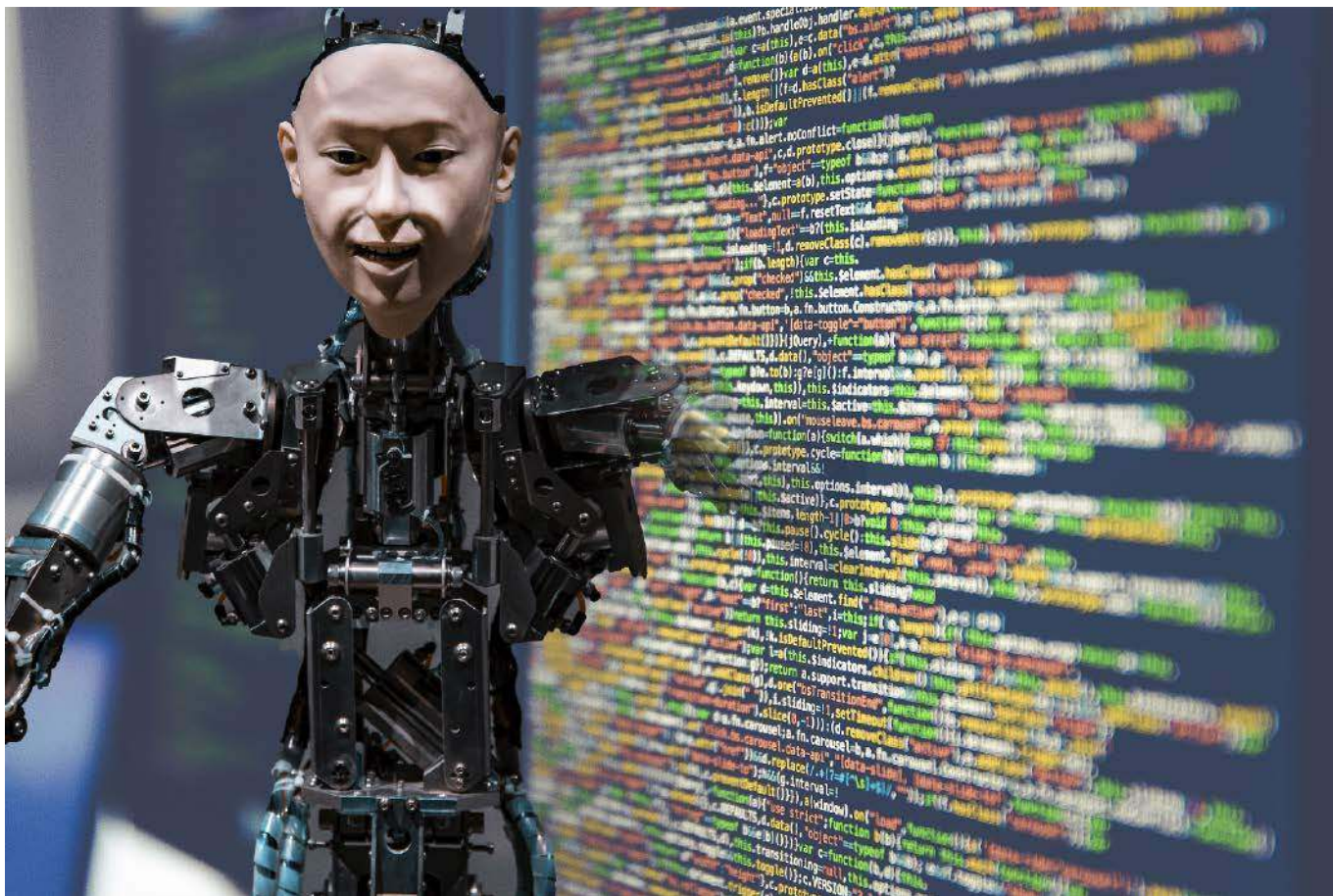
Napięcie nie powinno przekraczać 43 V, ponieważ jest to wartość, przy której otwiera się i zaczyna przewodzić tranzystor, znajdujący się na wejściu zasilania układu. Ma on za zadanie eliminować jedynie szybkozmiennne zakłócenia o charakterze szpilkowym, które mogłyby zniszczyć układ lub zaburzyć jego pracę.

Sygnał sterujący jest poddawany częściowej filtracji z zakłóceń, które mogłyby powodować samoczynne wyzwolenie przełącznika. Minimalny czas trwania impulsu sterującego można określić na około 1 s, zaś maksymalny czas jego trwania jest dowolnie długi. Jeżeli przed samoczynnym wyzwoleniem przełącznika nadejdzie następny

impuls (np. stacyjka została wyłączona na chwilę i znowu załączona), czas opóźnienia wyłączenia będzie liczony od nowa. Natężenie prądu pobieranego przez wejście sterujące nie jest wysokie: około 100 μ A przy 10 V, około 1,5 mA przy 24 V i niecałe 4 mA przy 43 V.

Układ można przystosować do zasilania napięciem o wartości 12 V, a dokładniej z zakresu 11 V...15 V, czyli np. z instalacji elektrycznej samochodu osobowego. Wystarczy wylutować tranzystor T4, a między pola lutownicze przeznaczone dla emitera i kolektora wstawić zworę z cienkiego drutu. Są to otwory: środkowy (kolektor) i prawy, znajdujący się bliżej rezystora R7 (emiter). Między tymi polami, na warstwie opisowej płytki, znajdują się trzy cienkie linie. Warto też wymienić tranzystor D4 na egzemplarz o niższym napięciu przewodzenia, np. 1.5KE18CA. Można wtedy nie montować elementów D6...D8, R8, R9 i T3.

Michał Kurzela, EP



Sztuczna inteligencja w praktycznej elektronice

Nie wszyscy sobie zdają sprawę z faktu, że sztuczna inteligencja otacza nas już na co dzień i właściwie każdego dnia z niej korzystamy. Mowa o wszelkiego rodzaju algorytmach, które pomagają nam w życiu lub służą do oceniania naszych preferencji, zachowań i klasyfikowania nas. I o ile jeszcze do niedawna AI (Artificial Intelligence) kojarzyło się i operowało głównie w chmurze, a więc gdzieś w wirtualnym świecie, obecnie algorytmy uczenia maszynowego implementuje się także w sprzęcie i w rozwiązaniach tworzonych przez elektroników. Jeśli wśród naszych czytelników są jeszcze osoby, które nie podjęły tego tematu, to nadszedł najwyższy czas, by się z nim zaznajomić, gdyż w przyszłości nie obejdzie się już bez sztucznej inteligencji.

Sztuczna inteligencja funkcjonuje w przestrzeni publicznej w sposób nieco nieokreślony. Przypomina to czasy sprzed 8 lat, gdy w „Elektronice Praktycznej” zaczynaliśmy pisać o Internecie Rzeczy i ciągle pojawiały się pytania o sens IoT, bo przecież komunikacja bezprzewodowa

między urządzeniami była możliwa i wcześniej. Dopiero po latach nadeszło powszechne zrozumienie IoT jako trendu, który opisuje pewne cechy niemal wszystkich nowoczesnych urządzeń elektronicznych. Idea Internetu Rzeczy spowodowała powstanie ogromnej liczby nowych rozwiązań, które wcześniej nie miały sensu bytu.

AI jako nowy trend

Ze sztuczną inteligencją jest podobna sytuacja, choć mamy tu zasadniczą różnicę. O ile w przypadku IoT nikt nie miał zbytnio wyobrażenia, czym taki Internet Rzeczy miałby być, to w odniesieniu do AI wizji mamy bardzo wiele, prezentowanych od lat w książkach, filmach i serialach science fiction. Rzecz w tym, że wizje te bardzo odbiegają od obecnych, praktycznych zastosowań sztucznej inteligencji. Wręcz można się często spotkać z opinią, że tworzone algorytmy i urządzenia wcale nie są inteligentne, bo nie mają cech ludzkich ani zdolności do rozumienia kwestii, do których nie zostały przygotowane. Rzeczywiście – wśród rozwiązań, które wykroczyły poza świat nauki, nie mamy jeszcze przykładów urządzeń, sprawiających wrażenie inteligentnych po ludzku.

Takiej ogólnej, sprawnej, sztucznej inteligencji jeszcze nie opracowano. Ale nie oznacza to, że praktyczne rozwiązania klasy AI nie

są użyteczne. Wręcz przeciwnie – są niezwykle pomocne w codziennym życiu i często nawet nie zdajemy sobie sprawy z faktu, że opierają swoje działanie na (specjalistycznej) sztucznej inteligencji. Bo to, czy dana aplikacja korzysta z AI, można dosyć łatwo określić, znając jej architekturę.

Od strony twórców rozwiązań zastosowanie AI polega na pozostawieniu pewnego, choćby małego fragmentu aplikacji algorytmowi, którego parametry nie są dobierane przez eksperta. Zamiast tego, stosując odpowiednie metody, można skorzystać z wybranego, względnie generycznego (choć odpowiednio dostosowanego) modelu, którego szczegóły działania dobierane są w procesie uczenia maszynowego. Zdejmuje to z inżyniera czy naukowca problem samodzielnego zrozumienia wszystkich zależności w dziedzinie, której dotyczy realizowany projekt, a które byłyby potrzebne do zaprogramowania klasycznego algorytmu działania systemu. Pracę tę wykonują odpowiednio wydajne komputery, które same poszukują powtarzających się wzorców i innych zależności, by następnie wywnioskować, w jaki sposób podobierać konkretne kryteria, na podstawie których będzie można uzyskiwać wyniki i decyzje, wartościowe z punktu widzenia użytkownika. I co ciekawe, sposobów na przygotowanie takiej aplikacji jest całkiem sporo.

Sztuczna inteligencja a uczenie maszynowe

Obok samej sztucznej inteligencji często pojawia się hasło uczenia maszynowego. Bywa, że oba te pojęcia są używane zamiennie, co jest w pewnym zakresie błędem, ale niezupełnie. Sztuczna inteligencja jest pojęciem nieco szerszym i w istotnym stopniu odnosi się nie tylko do praktyki inżynierskiej, ale i do wizji naukowej czy nawet społecznej. Sztuczną inteligencją można właściwie nazwać wszystko, co wydaje się w jakiś sposób inteligentne, a nie powstało w wyniku działań sił natury (biologii). Tymczasem do uczenia maszynowego zaliczamy zestaw całkiem konkretnych metod i algorytmów, które na różne sposoby pozwalają zbudować modele matematyczne, robiące wrażenie inteligentnych.

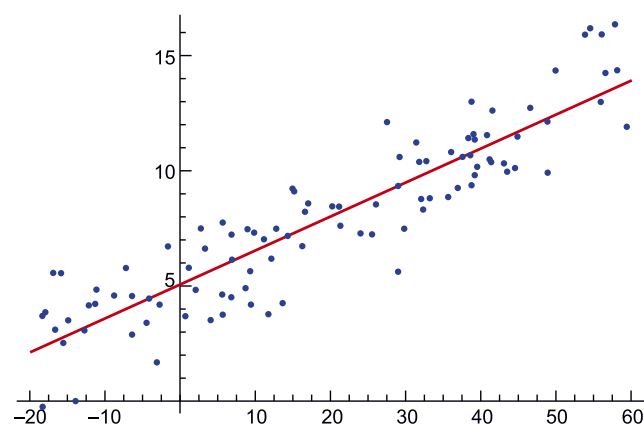
Można więc powiedzieć, że uczenie maszynowe to część całej, dużej dziedziny sztucznej inteligencji, odpowiadająca za budowanie modeli, ale taka klasyfikacja budzi kontrowersje. Problem w tym, że niektóre z metod uczenia maszynowego są na tyle nieskomplikowane, że ciężko je z czystym sercem nazwać sztuczną inteligencją. Dotyczy to przede wszystkim prostych statystyk, bardzo łatwych w implementacji, a nierzadko dających rewelacyjne rezultaty.

Na czym polega uczenie maszynowe?

Tak jak zostało to wspomniane wyżej, na uczenie maszynowe składa się szereg metod czy też algorytmów postępowania, które prowadzą do uzyskania rezultatu w postaci modelu realizującego określone funkcje. Co ważne, na początku takiego procesu mamy zbiory danych. Są one podstawą do przygotowania algorytmu, tak jak przeanalizowane wytyczne są podstawą do budowy klasycznych programów komputerowych czy obwodów elektronicznych. Można wręcz powiedzieć, że dane zastępują wytyczne i kreują ostateczny kształt modelu, przy mniejszej lub większej pomocy ze strony inżyniera budującego model.

Różne rodzaje uczenia maszynowego

Klasyczne uczenie maszynowe można podzielić na dwie kategorie: z nadzorem i bez nadzoru. Uczenie z nadzorem polega na wstępnym wskazaniu wytycznych, czyli np. celu klasyfikacji. Zresztą właśnie klasyfikacja może być świetnym przykładem zadania, jakie można powierzyć sztucznej inteligencji. Przykładowo, możemy zebrać szereg danych z czujników danego systemu i jednocześnie nazwać kilka stanów pracy tego systemu, które nas interesują. Podając zestawy danych wraz z przypisanymi im stanami pracy systemu i stosując któryś z algorytmów klasyfikacji (a więc klasycznego uczenia maszynowego nadzorowanego), jesteśmy w stanie uzyskać mechanizm,



Rysunek 1. Przykład prostej regresji liniowej

który pozwoli nam na całkiem poprawne klasyfikowanie kolejnych zestawów danych z czujników z takich systemów, tj. automatyczne przypisywanie im któryś z zdefiniowanych wcześniej etykiet.

Drugim rodzajem uczenia z nadzorem jest regresja, czyli dążenie do uzyskania modelu (najczęściej wielomianowego lub ew. liniowego), na podstawie którego będzie można, na bazie o jednych wartości wejściowych, z małym uchybem obliczać wartości wyjściowe (rysunek 1).

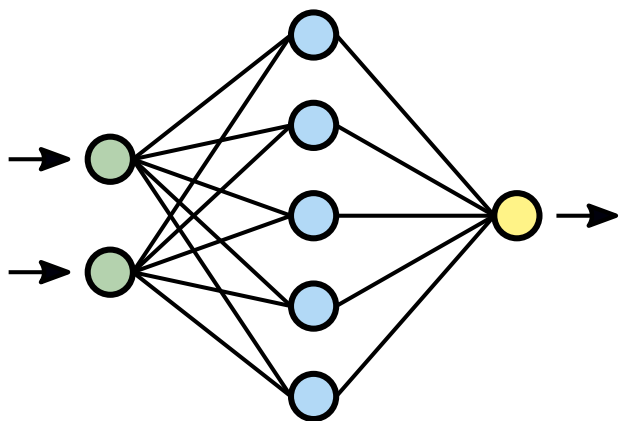
Zupełnie inaczej przebiega uczenie bez nadzoru. Przypomina ono pozostawienie dziecka w świecie bez nauczyciela, który mógłby mówić, jakich wyników oczekuje. Typowym przykładem jest grupowanie, które pozwala wykrywać nienazwane wcześniej stany. Można to sobie wyobrazić np. w kontekście działania pełnej czujników, złożonej maszyny przemysłowej. Jeśli nawet nie wiemy, co się może w niej popsuć, algorytm grupowania będzie mógł, w oparciu na podanych danych historycznych, wyznaczyć kilka rodzajów, silnie różniących się od siebie usterek, które później będziemy mogli jakoś po swojemu nazwać. Choć może nie jest to idealny przykład, tego typu algorytmy pozwalają łączyć w grupy podobne do siebie sytuacje, których podobieństwa wcale z góry nie widać. To użyteczny zabieg, gdyż pozwala odkryć pewne schematy i połączyć ze sobą różne zdarzenia, prowadząc do wytworzenia nowych kategorii sytuacji, a w konsekwencji np. do opracowania odpowiednich metod postępowania dla poszczególnych kategorii.

Podobnie jak grupowanie jest tak jakby nienadzorowaną wersją klasyfikacji, tak wyszukiwanie wzorców jest nienadzorowaną wersją regresji. Jeśli nie wiemy z góry, co od czego zależy w zbiorze danych, jaki posiadamy, to właśnie za pomocą wyszukiwania wzorców jesteśmy w stanie wykryć zależności, określić je zasadami i planować na ich podstawie dalsze działania.

Warto też wspomnieć o generalizacji, która również zaliczana jest do klasycznego uczenia maszynowego bez nadzoru. Służy do upraszczania zbiorów danych, tak by pozbyć się z nich wartości, które nie mają dużego znaczenia i które tylko zaciemniają obraz. Generalizacja nazywana jest również redukcją liczby wymiarów danych.

Uczenie ze wzmocnieniem

W powyżej opisanych metodach uczenie maszynowe wymaga posiadania wcześniej przygotowanych zbiorów danych, a często i odpowiedzi, jakie uznalibyśmy za poprawne dla konkretnych pakietów danych. Bywają sytuacje, gdy nie umiemy z góry podać dobrej odpowiedzi, ale za to potrafimy wskazać, czy jest ona lepsza, czy gorsza niż poprzednia. Korzystanie z tej wiedzy w trakcie procesu uczenia nazywamy uczeniem ze wzmocnieniem. W trakcie tego sposobu uczenia model próbuje samodzielnie uzyskać rezultaty, za które jest nagradzany lub karany i na bazie nagród dokonuje następnych zmian swoich parametrów. Tak jak i w przypadku innych metod uczenia maszynowego, nie mamy pewności, że dysponujemy wszystkimi informacjami, które potrzebne są do precyzyjnego opisanie modelu, ale



Rysunek 2. Uproszczony schemat jednokierunkowej sieci neuronowej, poszczególne kółka oznaczają sztuczne neurony

mamy nadzieję, że informacje te będą wystarczające by wytworzyć model względnie dobry. W uczeniu ze wzmocnieniem jest to bardzo typowa sytuacja i algorytm (a właściwie agent, bo tak się nazywa uruchomiony algorytm), stara się maksymalizować pewne kryterium (jakąś miarę), dostępną właśnie w postaci wzmocnień.

Sieci neuronowe

Kolejnym hasłem, jakie pojawia się w kontekście sztucznej inteligencji jest sieć neuronowa. Sieci neuronowe mogą być uczone na różne z powyższych sposobów, celem uzyskania modeli realizujących praktycznie dowolne zadania (rysunek 2). To, jak taka sieć jest zbudowana, tj. jakie neurony zastosowano, jakiego rodzaju połączenia pomiędzy neuronami i jakiego rodzaju dane i sposób trenowania są dopuszczalne, określa podtyp sieci. Wbrew względnie powszechnym przekonaniom, aby zrealizować algorytm sztucznej inteligencji wcale nie trzeba korzystać z sieci neuronowych. Wiele algorytmów opiera się o użycie zupełnie innych metod, często bazujących na statystyce.

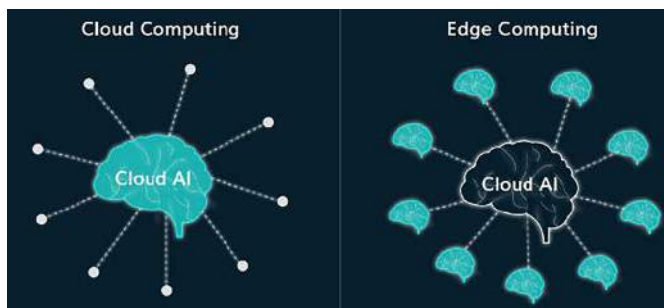
Nic poza kreatywnością i umiejętnościami twórcy nie stoi na przeszkodzie by próbować połączyć kilka z metod uczenia maszynowego. Jest to zresztą podejście, które nierzadko pozwala uzyskiwać najlepsze rezultaty. Przykładowo, sięgnięcie po którąś z metod klasyfikacji celem wstępnego pogrupowania danych lub zastosowanie metody generalizacji, by uprościć zbiory, może znacząco ułatwić dalsze trenowanie algorytmu, a nawet jego implementację w finalnym rozwiązaniu.

Sztuczna inteligencja w elektronice

Skoro wiemy już, na czym polega sztuczna inteligencja i co teoretycznie pozwala wykonywać, można się zastanowić, jak wykorzystać ją w praktyce i jakie zastosowania może znaleźć w życiu elektronika. Takich obszarów użycia jest bardzo wiele ze względu na ogromną uniwersalność idei sztucznej inteligencji. W końcu chodzi o zastąpienie pracą maszyny niektórych czynności, wykonywanych przez człowieka, które trudno opisać prostymi zasadami i algorytmami.

Na krańcu sieci

Sztuczną inteligencję spotkamy nie tylko w urządzeniach codziennego użytku, szczególnie tych skomputeryzowanych, pracujących z chmurą. Jest obecna również w niektórych narzędziach, stosowanych przez elektroników oraz może być implementowana w rozwiązaniach przez elektroników tworzonych. Szczególnie interesująca jest idea implementacji algorytmów AI w sprzęcie, tj. niekoniecznie poprzez sprzętowe wykonywanie algorytmów, bez udziału oprogramowania, ale przetwarzanie danych na tzw. krańcu sieci. Mowa o wszelkiego rodzaju urządzeniach i ich komponentach, które stały się na tyle wydajne, że są w stanie wykonywać zaprogramowane algorytmy sztucznej inteligencji bez potrzeby przesyłania danych do sieci i korzystania z mocy obliczeniowej dostępnej w chmurze



Rysunek 3. Idea implementacji algorytmów AI w sprzęcie, tj. przetwarzanie danych na tzw. krańcu sieci

(rysunek 3). Jeszcze kilka lat temu, taka architektura była nie do pomyslenia. Teraz staje się coraz bardziej popularna, szczególnie w dobie Internetu Rzeczy, który promuje rozwiązania o małym poborze mocy, w których komunikacja z siecią sprowadza się do przesyłania małych paczek danych.

Metoda postępowania

Co ciekawe, budowanie takich sieci nadal wymaga chmury lub całkiem potężnego komputera, który posłuży do przetworzenia danych i wytrenowania algorytmu. Istnieją dostawcy rozwiązań tego typu, którzy ułatwiają wdrażanie sztucznej inteligencji na krawędziach sieci. Zazwyczaj procedura przebiega następująco:

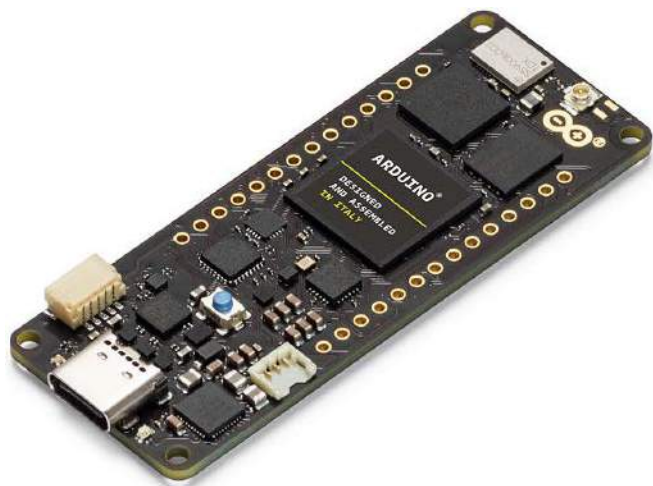
1. Użytkownik zbiera dane, które mają posłużyć szkoleniu sieci.
2. Dane dzielone są na odpowiednie zbiory trenujące i testowe oraz ewentualnie na zbiór danych do weryfikacji, czy algorytm się nie przetrenował w żaden sposób.
3. Dane wprowadzane są do systemu komputerowego, w którym następuje projektowanie – czy to sieci neuronowej, czy też jakiegoś innego algorytmu AI. Twórca (elektronik/specjalista od sztucznej inteligencji) kreuje całkowicie wirtualny model, aż do momentu uzyskania odpowiednio sprawnego rozwiązania.
4. Gotowy model załadowany jest do urządzenia sprzętowego w postaci skompilowanego kodu, realizującego potrzebne funkcje.

Powyższa procedura sprawia, że większość pracy, związanej z przygotowaniem algorytmu, odbywa się zupełnie w oderwaniu od sprzętu i docelowego mikrokontrolera czy też innego układu aplikacyjnego. Ważne jedynie, by docelowy układ był odpowiednio wydajny do realizacji ustalonego algorytmu. Obliczenia oparte o sieci neuronowe często wymagają niemałej mocy obliczeniowej, ale na szczęście na rynku jest już dosyć dużo firm, które dostarczają układy z obwodami wspomagającymi wykonywanie algorytmów sztucznej inteligencji.

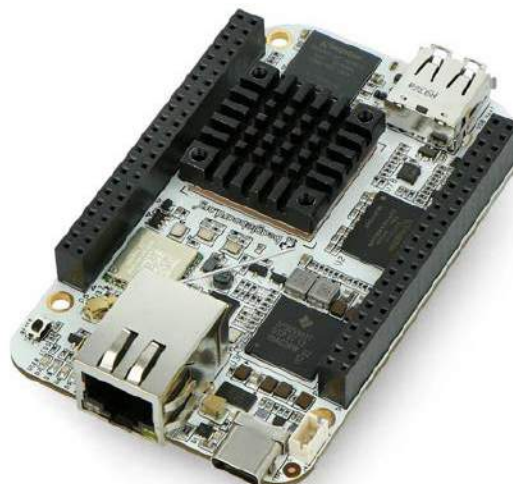
Przykłady zastosowań

Jednym z ciekawszych, z punktu widzenia elektroników, zastosowań AI jest wspomaganie projektowania układów scalonych. Typowy, nowoczesny układ scalony jest podzielony na wiele bloków, co oznacza, że ma konstrukcję modułową. Wśród modułów można wyróżnić np. bloki pamięci, jednostki obliczeniowe i układy interfejsowe. Projektant sam wybiera ile jakich bloków chciałby umieścić w tworzonej produkcie – wymaga to odpowiedniej wiedzy i zrozumienia tematu, ale jest zadaniem względnie prostym, często silnie zależnym bardziej od decyzji biznesowych i marketingowych niż inżynierskich.

Cała trudność pojawia się na etapie przekładania takiego schematu na tzw. layout, a więc rozplanowanie rozmieszczenia poszczególnych elementów. Od ich umieszczenia zależy wydajność i wiele innych parametrów powstającego podzespołu. W praktyce to właśnie od tego etapu zależy czy dany element scalony będzie konkurencyjny i sprawnie działający. Zadanie jest na tyle trudne, że pomimo dziesięcioleci badań, spełnienie wieloaspektowych kryteriów projektowych i technologicznych wymaga bardzo długiej pracy ze strony projektanta. Kluczowa jest nie tylko liczba modułów, ale ich rozmiary



Fotografia 1. Moduł Arduino Portenta H7 z dwurdzeniowym mikrokontrolerem STM32H747 który może realizować procesy utworzone za pomocą TensorFlow Lite



Fotografia 2. Minikomputer BeagleBone AI z procesorem Texas Instruments Sitara AM5729. Algorytmy sztucznej inteligencji mogą być implementowane dzięki bibliotece TIDL udostępnionej przez producenta

i parametry elektryczne, co przy milionach elementów do ułożenia staje się niezwykle skomplikowane.

To właśnie do takich zadań użyteczna okazuje się sztuczna inteligencja. Odpowiednie wyszkolenie algorytmów powinno pozwolić na przygotowanie mechanizmu samodzielnie układającego bloki i to z dużą sprawnością. Zresztą tego typu zadania już teraz powierza się AI, co ciekawe – np. w celu budowania układów przetwarzających modele sztucznej inteligencji (np. jednostki TPU – *Tensor Processing Unit*).

Kolejnym przykładem zastosowania sztucznej inteligencji są aplikacje przemysłowe, głównie związane z konserwacją predykcyjną. Dużym problemem w przemyśle jest trudność dokładnego przewidzenia momentu wystąpienia awarii instalacji. Ponieważ usterki tego typu powodują wstrzymanie całych linii produkcyjnych, co często skutkuje poważnymi stratami, prace serwisowe planuje się klasycznie – w oparciu na harmonogramach. Zaplanowany przestój jest dużo mniej kosztowny w realizacji niż niespodziewany – można go np. przeprowadzić w nocy lub w weekend, o ile zespół serwisowy jest skłonny pracować w takich godzinach. Rzecz w tym, że aby uniknąć nieprzewidzianych awarii, harmonogramy są planowane z zapasem, tj. tak, by czynności serwisowe nastąpiły na pewno, zanim maszyna ulegnie uszkodzeniu. W ten sposób naprawy prowadzone są nierzadko na maszynach całkiem sprawnych, tym bardziej jeśli realne obciążenie sprzętu nie było na tak wysokim poziomie, jak pierwotnie zakładano. Tego typu procedury są dalekie od optymalnych, ale trudno jest stworzyć sensowne algorytmy oparte na wiedzy eksperckiej, które z odpowiednim wyprzedzeniem pozwalałyby na wykrywanie anomalii i przewidywały popsucie się urządzenia.

To właśnie miejsce na zastosowanie sztucznej inteligencji. W czasach, gdy czujniki przemysłowe stały się niedrogie, dokładne i łatwe do połączenia w sieć, można zestawić całkiem zaawansowane układy pomiarowe, których wyniki są następnie analizowane przez złożone algorytmy. Ręczne opracowanie takich algorytmów jest niezwykle trudne i wymaga wiedzy eksperckiej, którą zdobywa się na przestrzeni lat doświadczenia. Co więcej, poszczególne maszyny i ich wersje różnią się od siebie na tyle, że czasem trudno przełożyć wiedzę zdobytą na używaniu jednego urządzenia na praktykę sprawdzającą się w innej sztuce, pracującej np. w innym zakładzie. Sztuczna inteligencja pozwala rozwiązać ten problem. Szybko można zaprojektować algorytm do monitorowania pracy maszyny, nie znając tak naprawdę szczegółów jej specyfiki. Dobór parametrów algorytmu następuje automatycznie. Efektem jest narzędzie, które przewiduje usterki (z odpowiednim marginesem błędu), na żywo oceniając stan maszyny.

Sztuczna inteligencja może być też użyta w procesie produkcji elektroniki. Istnieją firmy, które specjalizują się w monitorowaniu

produkcji i montażu płytek drukowanych, zbierając przez długi czas dane, wykorzystywane następnie celem analizy przestrzennej. Dane takie mogą opisywać procesy wytwarzania elektroniki i służyć do uczenia oraz optymalizacji algorytmów sztucznej inteligencji, by modernizować procesy produkcyjne. W nowoczesnych realizacjach tego typu stosuje się modelowanie trójwymiarowe, które daje lepszy wgląd w proces produkcyjny niż modelowanie dwuwymiarowe. Maszyny do inspekcji, obsługujące algorytmy AI, mierzą parametry charakterystyczne komponentów i połączeń lutowicznych w trzech wymiarach. Ważna jest odpowiednio wysoka wiarygodność danych, a więc zastosowanie dobrej jakości, sensownie ustawionych czujników. Za pomocą AI maszyny te potrafią wykrywać np. odkształcenia płytek czy też problemy wynikające ze zbyt bliskiego rozmieszczenia komponentów. Analizie poddawane są zarówno informacje z systemów optycznych, jak i elektrycznych testerów. Algorytmy pozwalają na kategoryzowanie komponentów według liczby wyprowadzeń, typu i innych cech. Efektem działania omawianych maszyn jest zwiększenie uzysku produkcyjnego i poprawienie jakości procesów.

Narzędzia do AI

Zabierając się do wdrożenia sztucznej inteligencji we własnych rozwiązaniach, warto wyposażać się w odpowiednie narzędzia, ułatwiające pracę. Wiele zależy od technologii, z jakiej chcemy skorzystać, ale znaczenie ma też oczywiście sam docelowy typ aplikacji. W ogólności



Fotografia 3. Płyta Ultra96-V2 wyposażona w układ Xilinx Zynq UltraScale+ oferuje inżynierom bardzo konkurencyjną cenowo platformę rozwojową do szybkiego prototypowania m.in. w zakresie sztucznej inteligencji



Fotografia 4. NVIDIA Jetson Nano to zestaw rozwojowy z układem graficznym Maxwell składającym się z 128 rdzeni wspierającym technologię CUDA. Przeznaczony jest zwłaszcza do przetwarzania bardzo dużych ilości danych lub obrazu i zapewnia odpowiednią wydajność do uruchamiania nowoczesnych rozwiązań AI w małym, energooszczędnym formacie

warto zainteresować się gotowymi zestawami elektronicznymi, przygotowanymi przez różnych producentów, które służą właśnie do rozpoczęcia prac i nauki (fotografie 1...4). Warto też podszkolić się z programowania, gdyż nierzadko będzie ono potrzebne, by sensownie wdrożyć nowe rozwiązanie z dziedziny sztucznej inteligencji.

Wielu producentów oferuje co prawda gotowe biblioteki, które wystarczy skonfigurować, ale dobrze, jeśli inżynier rozumie, co właściwie dzieje się w tle w jego rozwiązaniu, choć nie można wykluczyć, że w przyszłości wystarczające będzie jedynie graficzne układanie bloków i podawanie zbiorów danych do szkolenia algorytmów.

Zestawy do sztucznej inteligencji bardzo często zawierają komponenty do obrazowania – głównie kamery pracujące w świetle widzialnym, ale czasem też systemy działające w podczerwieni. To dlatego, że wiele widowiskowych rezultatów udaje się osiągnąć właśnie w ramach systemów wizyjnych. Bez AI trudno było definiować konkretne kryteria, pozwalające na ocenianie obrazu, a tym bardziej na rozpoznawanie szczegółów. Obecnie stało się to znacznie łatwiejsze, a ilości danych, generowane przez takie czujniki obrazów, przestały być już przytłaczające dla średniej wielkości mikrokontrolerów.



TensorFlow

Rysunek 4. TensorFlow to otwartoźródłowa biblioteka programistyczna stosowana w uczeniu maszynowym i głębokich sieciach neuronowych



Fotografia 5. Zintegrowany akcelerator sztucznej inteligencji Intel Neural Compute Stick 2

Do bardzo ciekawych rozwiązań należą też komponenty ze zintegrowanymi akceleratorami sztucznej inteligencji. Google opracowało kiedyś jednostki TPU (*Tensor Processing Unit*), a następnie jeszcze TPU Edge, przystosowane do pracy na krańcach sieci. Są one przygotowane przede wszystkim do pracy z oprogramowaniem TensorFlow – jednym z bardziej popularnych środowisk do programowania algorytmów uczenia maszynowego (rysunek 4).

Innym, ciekawym urządzeniem do AI jest Intel Neural Compute Stick, dostępny ostatnio w wersji 2 (fotografia 5). Bazuje na układzie Intel Movidius Myriad X Vision i dobrze współpracuje z takimi platformami jak Raspberry Pi. Pozwala na realizowanie głębokiego uczenia i pracę na krawędziach sieci. Producent dostarcza zestaw bibliotek funkcji i wstępnie optymalizowanych programów, a także bogate narzędzia deweloperskie.

Co na to rynek?

Opisane powyżej narzędzia do AI to jedynie niektóre z licznych przykładów, pokazujących znaczący wzrost zainteresowania sztuczną inteligencją na rynku elektroniki. Z wypowiedzi praktyków, jakie docierają do naszej redakcji oraz z rozmów z przedstawicielami firm z branży elektronicznej, jasno wynika, że sztuczna inteligencja staje się coraz powszechniej implementowanym rozwiązaniem. Nie oznacza to, że wszyscy przestawili się na tworzenie aplikacji sztucznej inteligencji i że inne projekty nie mają już racji bytu. Raczej chodzi o to, że w bardzo dużej części projektów, szczególnie tych bardziej złożonych i najbardziej obiecujących, uwzględnienie AI jest korzystne i opłacalne.

Jednocześnie wśród kierownictwa firm oraz samych inżynierów panuje obawa o brak kadr i odpowiednich umiejętności, by sztuczną inteligencję wdrażać. Pracownicy, zajmujący się projektowaniem elektroniki, nierzadko czują się niewystarczająco wspierani przez swoich pracodawców w zakresie zdobywania wiedzy o AI. Temat sztucznej inteligencji nie jest łatwy, ale po poświęceniu odpowiednio dużej ilości czasu wiele osób będzie w stanie się z nim zapoznać i zacząć implementować rozwiązania sztucznej inteligencji, czy to samodzielnie, czy w zespołach. Na pewno warto się nim zainteresować, gdyż będą to umiejętności niezwykle przydatne już w niedługiej przyszłości. Co więcej, zapoznanie się z algorytmami AI otwiera umysł na nowe pomysły, które mogą poprowadzić do stworzenia zupełnie nowej klasy rozwiązań i niezwykle innowacyjnych urządzeń. I o ile do stworzenia uniwersalnej sztucznej inteligencji jest nam jeszcze daleko, o tyle obecne możliwości budowania inteligencji specjalizowanej pozwalają z powodzeniem tworzyć ciekawe aplikacje oraz modernizować dotychczasowe konstrukcje, wprowadzając do nich wyjątkowo innowacyjne funkcje.

Marcin Karbowniczek, EP

Wykrywacz maseczki



Jeszcze do niedawna maseczki były naszą codziennością i w niektórych okolicznościach ciągle są wymagane (piszemy ten artykuł w czasie, gdy nakaz jeszcze obowiązuje) – w sklepach, centrach handlowych itd. Upominanie nienoszących ich osób jest uporczywe, dlatego też dobrze jest oddać to zadanie w ręce maszyny.

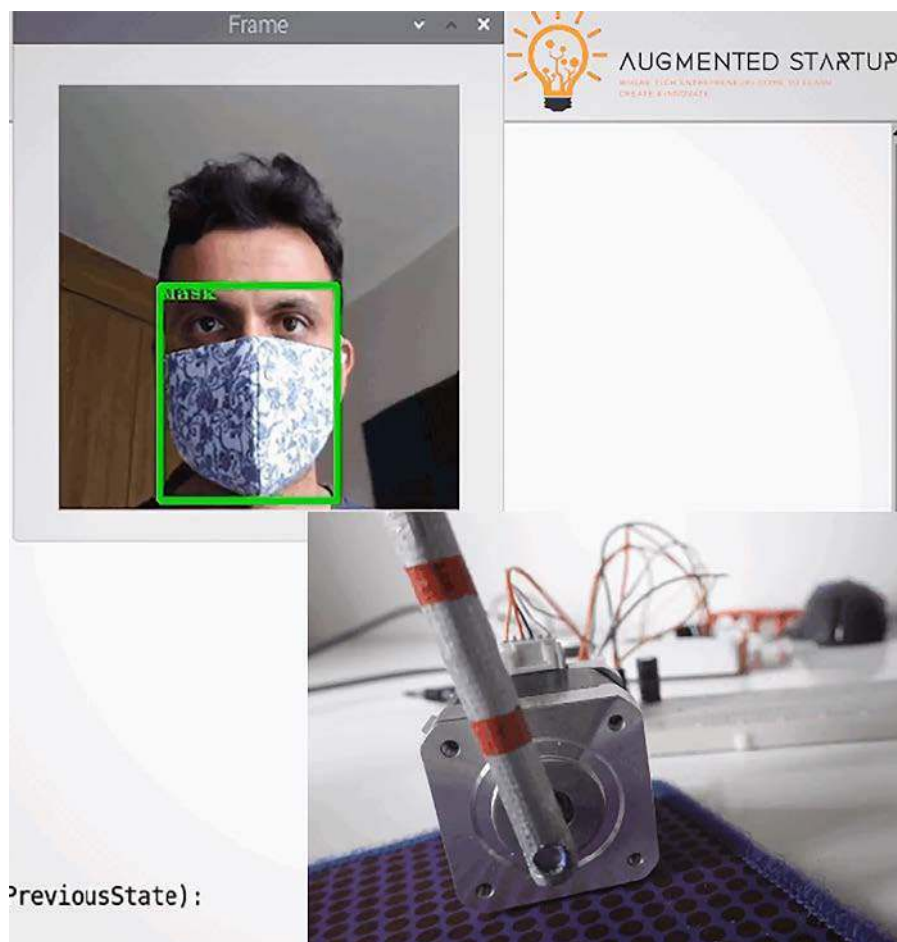
Opisany w artykule prosty system został wyposażony w kamerę, komputer jednopłytkowy Raspberry Pi oraz oprogramowanie oparte na OpenCV. Służy do wykrywania w czasie zbliżonym do rzeczywistego, czy osoba, na którą patrzy system, ma na twarzy założoną maseczkę. System kontroluje pracę prostego szlabanu zbudowanego na silniku krokowym. To tylko demonstracja działania, może on sterować dowolnym innym urządzeniem, np. wyświetlać informację o konieczności noszenia maseczki w danym miejscu.

Zaprezentowany algorytm wykorzystuje detektor obiektów COVID-19 mask/no-mask opracowany przez Google Colab. Jednak nic nie stoi na przeszkodzie, aby zaimplementować inny detektor obiektów lub przygotować własny. W dalszej części artykułu znajduje się skrócony opis trenowania użytej tutaj sieci neuronowej do zastosowań widzenia maszynowego, dzięki czemu możliwe jest samodzielne nauczanie systemu rozpoznawania innego rodzaju obiektów.

Zastosowane elementy

Do zestawienia omawianego urządzenia potrzebne będą następujące komponenty:

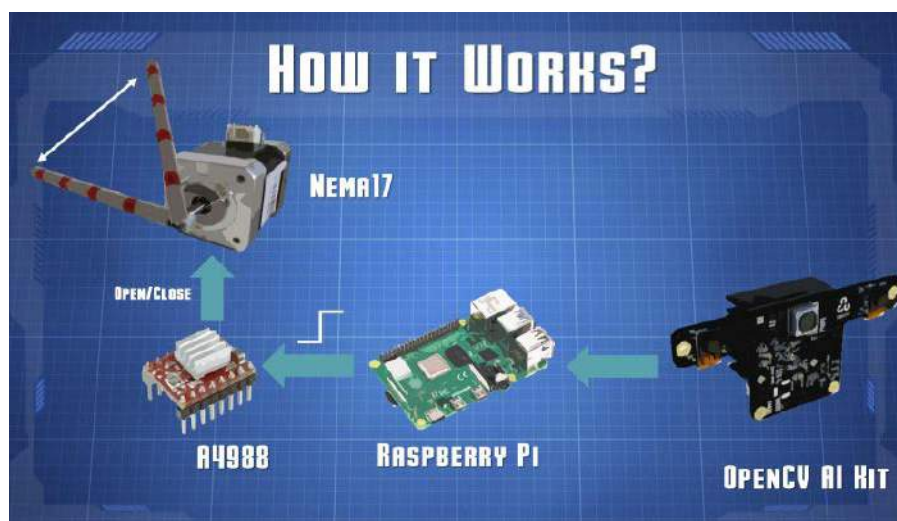
- Komputer jednopłytkowy Raspberry Pi z typowymi akcesoriami (zasilacz, karta pamięci itd.);
- Kamera OpenCV AI Kit. Autor zastosował model OAK D, ale można użyć również OAK-1;
- Silnik krokowy. Autor wybrał silnik w rozmiarze NEMA17, lecz można sterować dowolnym innym silnikiem, kompatybilnym z poniższym driverem lub zastosować własny driver;
- Sterownik silnika krokowego oparty na układzie A4988 lub inny, kompatybilny ze sterowaniem typu STEP/DIR;
- Zasilacz 12 V o dostatecznej wydajności prądowej do zasilania wybranego silnika i drivera;
- Płytkę uniwersalną lub płytkę stykową i drobne elementy, takie jak kondensatory i przewody do poprowadzenia zwolek.



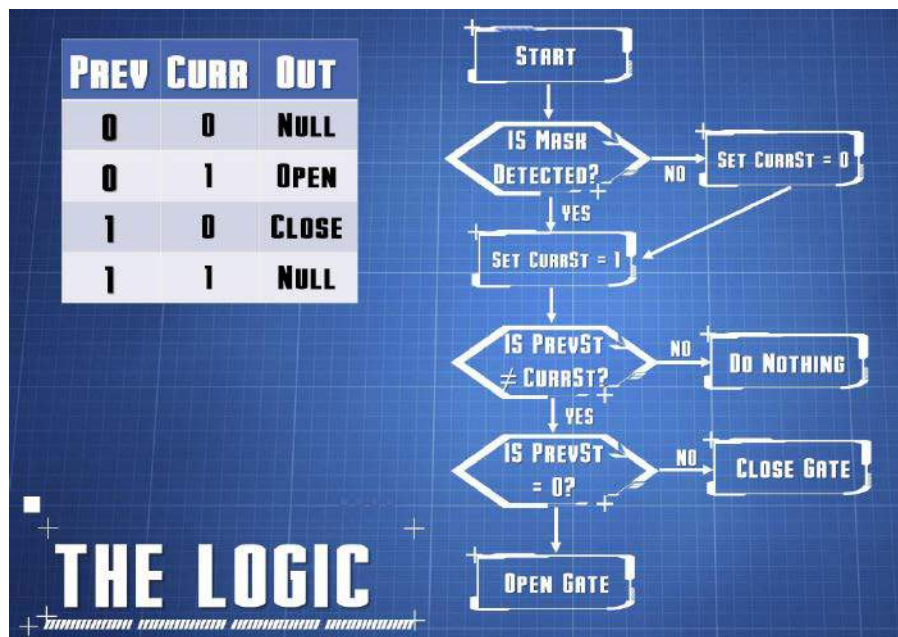
Opcjonalnie przydatna jest jeszcze drukarka 3D, która pozwala na wydrukowanie elementów mechanicznych, jakimi sterować ma silnik krokowy. W przypadku tego systemu jest to prosty szlaban, ale tylko wyobraźnia i moc silnika ogranicza to, co można do niego podłączyć w celu sygnalizacji braku maseczki.

Jak to działa

Obraz osoby z maseczką (lub bez) jest przechwytywany przez zestaw kamery AI OpenCV. Kamera zbiera 30 klatek na sekundę i z tą prędkością obrazy są przekazywane do Raspberry Pi poprzez USB. W samym Raspberry Pi pracuje oprogramowanie, oparte na platformie Roboflow, które analizuje zgromadzone obrazy.



Rysunek 1. Schemat blokowy urządzenia



Rysunek 2. Schemat blokowy ogólnego algorytmu działania urządzenia

System AI, zaimplementowany w Raspberry Pi, obsługuje pojedyncze wyjście cyfrowe, którego stan odpowiada obecności maseczki na osobie, na którą patrzy kamera. Pojedyncza linia GPIO steruje driverem silnika krokowego i otwiera oraz zamyka szlaban. Algorytm rozpoczyna działanie zaraz po uruchomieniu urządzenia. W przypadku wykrycia twarzy sprawdzane jest, czy dana osoba nosi maseczkę, czy nie. Jeśli nie, to stan linii wyjściowej ustawiany jest na 0, w innym przypadku, jeśli maseczka zostanie wykryta, stan wyjściowy ustawiany jest na 1. W następnym

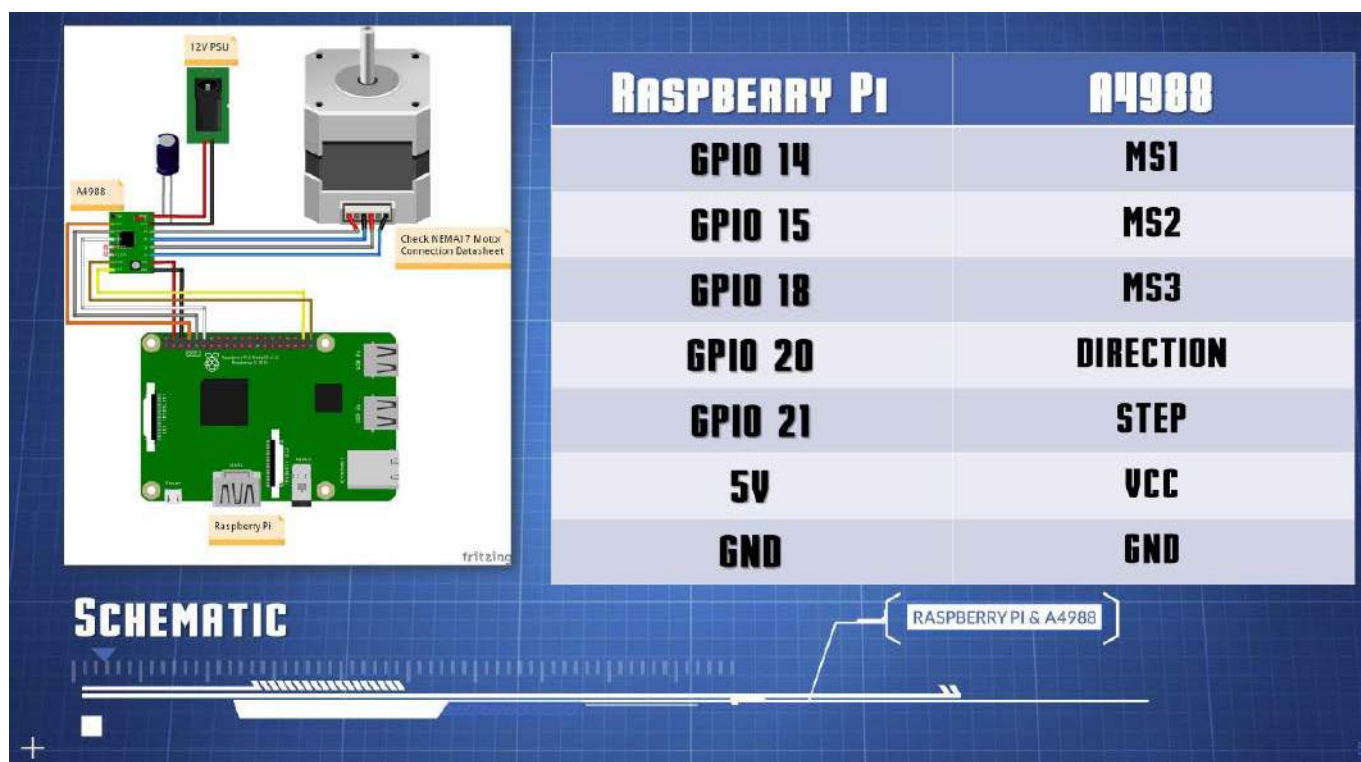
kroku algorytm porównuje, czy poprzedni stan jest taki sam jak aktualny. Zasadniczo poszukuje on przejścia lub zdarzenia wyzwalającego zbrocze. Jeśli nie ma takiego przejścia, nic nie robi, ale jeśli wykryte zostanie przejście, to sprawdzany jest poprzedni stan – jeśli był on zero, to układ otwiera bramkę, a w innym przypadku nic się nie dzieje. Jeśli jednak aktualny stan równy jest 1, co oznacza wykrycie maski, to szlaban jest otwierany. Jeśli poprzedni i obecny stan to jedynka, znaczy to, że użytkownik cały czas ma na sobie maseczkę i szlaban pozostaje cały czas otwarty.

Można by zapytać, dlaczego układ działa w tak skomplikowany sposób – monitoruje obecny i aktualny stan oraz porównuje je ze sobą? W tym systemie chcemy, aby brama otwierała się i zamykała. Jeśli nie będziemy kontrolować obecnego stanu, istnieje ryzyko, że system spróbuje otworzyć już raz otwartą bramkę, co będzie oznaczało przekroczenie szlabanu o kolejne 90 stopni i tak dalej.

Opracowany system zapewnia, że silnik nie przekręca nadmiernie szlabanu w żadnym kierunku. Łatwiejszą alternatywą jest to, że można otworzyć bramkę po wykryciu maski i zamknąć ją po określonej liczbie sekund, a następnie otworzyć ponownie po wykryciu maski.

Schemat

Schemat jest bardzo prosty, co pokazuje rysunek 3. Widać tam diagram połączeń komputera jednopłytkowego Raspberry Pi i sterownika silnika krokowego oraz sposób podłączenia silnika i zasilacza. Po prawej stronie rysunku umieszczono dodatkowo tabelkę, zawierającą listę połączeń poszczególnych linii GPIO Raspberry Pi. Po podłączeniu sterownika do systemu należy ustawić ograniczenie prądu układu A4988, aby było dopasowane do zastosowanego silnika. Za pomocą potencjometru regulowane jest napięcie odniesienia VREF w układzie. Ograniczenie prądowe jest w przybliżeniu równe dwukrotności napięcia VREF, co oznacza, że aby prąd uzwojenia silnika krokowego był równy 1 A, należy ustawić VREF=0,5 V. Warto pamiętać, że dla innych



Rysunek 3. Schemat połączeń Raspberry Pi, sterownika silnika krokowego i samego silnika (po lewej) oraz tabelka podsumowująca połączenia RPi i modułu z układem A4988 (po prawej)

sterowników silników krokowych sposób konfiguracji prądu maksymalnego może być inny – należy zastosować się do zaleceń producenta modułu.

Zestaw kamery AI OpenCV wystarczy podłączyć za pomocą dostarczonego kabla USB C. Aby sprawdzić, czy silnik krokowy jest prawidłowo podłączony (czy dobrze odnaleźliśmy pary uzwojeń), można podłączyć do nich diodę LED i zakręcić osią silnika. Jeśli dioda LED podłączona jest do jednego z uzwojeń, zaświeci się pod wpływem prądu zaindukowanego w uzwojeniu. To samo można powtórzyć z drugim uzwojeniem.

Oprogramowanie

Oprogramowanie do opisywanego modułu dostępne jest na repozytorium autora na GitHubie (<https://bit.ly/3yCkp3L>). Wystarczy sklonować zawartość wskazanego repozytorium w wygodnym dla nas miejscu. W pierwszej kolejności dobrze jest zrealizować samouczek, zawarty w folderze z aplikacją o numerze jeden. W artykule źródłowym znaleźć można m.in. film, który pokazuje, w jaki sposób należy zrealizować to zadanie.

Dalej należy skupić się na zawartości aplikacji 3. Znajdują się tam cztery pliki, o których należy pamiętać. Bazowe pliki aplikacji to `depthai_utils.py` i `main.py`.

Jeśli chcemy pominąć analizę oprogramowania, wystarczy uruchomić program na Raspberry Pi, wpisując w terminalu następujące polecenie, uruchamiające odpowiedni skrypt:

```
python3 main.py
```

Skrypt powinien się od razu uruchomić i zacząć kontrolować silnik krokowy szlabanu. Jeśli się nie uruchomi, przyczyną może być brak którejs z potrzebnych bibliotek. Aby je dodać, wystarczy wpisać polecenie:

```
python3 -m pip install -r requirements.txt
```

Wykorzystany model ML

Wykorzystana sieć neuronowa pozwala na zaawansowane wykrywanie obiektów. Sieć korzysta z frameworku Tensorflow, w ramach którego uruchamiana jest sieć SSD MobileNet V2. Sieć wytrenowana została w ramach Google Colab z użyciem 700 zdjęć osób w maskach i bez nich, na które naniesiono około 3600 oznaczeń.

Dokładny opis procesu uczenia sieci zawarto w repozytorium, które można znaleźć w linkach na końcu. Pokazany jest tam tzw. notatnik, który zawiera zapis krok po kroku procesu instalacji Tensorflow, MobileNet V2 i wymaganych przez nie bibliotek, a następnie uczenia sieci neuronowej rozpoznawania danych obiektów i generowania plików, wykorzystywanych przez algorytm.

Obrazy są złożone: różnią się znacznie skalą i kompozycją. Niemniej jednak detektor obiektów radzi sobie całkiem nieźle z tym stosunkowo niewielkim zestawem danych do takiego zadania. Szkolenie z zastosowaniem serwerów w chmurze, udostępnianych przez Google Colab, trwa około 2 godzin.

W zależności od tego, jaki GPU Colab przypisuje do instancji danego notatnika, szkolenie z użyciem 10 tysięcy kroków zajmuje od 1,5 do 2,5 godziny. To całkiem krótki czas, wymagany do wygenerowania gotowej do pracy sieci, zważywszy na jakość uzyskanego wyniku.

Następnie wykonywane są kolejne kroki, aby przekonwertować wynik nauki sieci na obiekt typu `blob` z wykorzystaniem OpenVINO jako etapu pośredniego. Pozwala to finalnie uruchomić model widzenia maszynowego na module DepthAI.

Nikodem Czechowski, EP

Źródła:

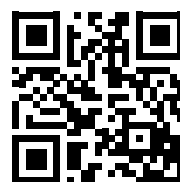
1. <https://bit.ly/2Sob6nw>
2. <https://bit.ly/2SoDpCj>
3. <https://bit.ly/3bS8WmJ>



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](http://bit.ly/2GADWTQ)



Ekran smartfona jako interfejs mobilny do amplitunera



czyli aplikacja mobilna poprzez UART w praktyce

W zeszłym roku, w kilku wydaniach „Elektroniki Praktycznej” (EP 4...7/20) ukazał się cykl artykułów na temat tworzenia aplikacji działającej na dowolnym smartfonie z użyciem języka JSON. Na smartfonie trzeba tylko zainstalować aplikację BBMobile, a cały design i funkcjonalność aplikacji realizowane są przez moduł sprzetowy.

Aplikacja BBmobile odbiera przez Bluetooth kod napisany w języku JSON i tworzy na ekranie smartfona interfejs użytkownika, a następnie cały czas odczytuje wszelkie polecenia z ekranu smartfona i przekazuje je zwrótnie do modułu BBMagic.

Moduł, w tym zastosowaniu potrzebny w wersji BBMobile, odbiera i nadaje dane poprzez łącze Bluetooth i przesyła je przez interfejs UART. Właściwy program trzeba napisać i zaimplementować w mikrokontrolerze (np. ARM) na własnej płytce drukowanej lub w dowolnym gotowym module (np. Arduino). Do komunikacji potrzebne są tylko dwa sygnały: TxD i RxD.

Komunikacja

Wyjście interfejsu UART naszego procesora oznaczone TxD należy podłączyć do wyprowadzenia nr 19 modułu, a wejście RxD

naszego procesora do wyprowadzenia nr 18 modułu. Na wyprowadzeniu 15 (oraz jego negacja na wyprowadzeniu 14) modułu BB-Magic pojawia się stan zera logicznego, gdy moduł jest sparowany ze smartfonem.

Napięcie zasilania z zakresu 1,8...3,3 V można podłączyć do wyprowadzeń 7 i/lub 21 modułu, a masę do wyprowadzeń 1 i 20 i dodatkowo (nie jest to konieczne) 22. Po włączeniu zasilania moduł ustawia prędkość transmisji na 9600 bodów bez sprawdzania parzystości. Zestaw poprawnych poleceń rozpoznawanych przez moduł zależy do tego, czy jest w stanie połączenia BLE (Bluetooth), czy nie. Przed połączeniem BLE obecność modułu można sprawdzić, wysyłając komendę `<hello` zakończoną parą znaków `\r\n`. Wszystkie komendy kończymy w ten sposób (hexadecymalnie 0x0D, 0x0A).

W odpowiedzi moduł powinien wysłać odpowiedź `>HI` zakończoną tak, jak komendy. Podstawowe informacje o tym module były podane, we wspomnianych na wstępie, wydaniach EP. Jednak moim zdaniem zabrakło opisu kilku ciekawych poleceń. Mam na myśli polecenia do tworzenia tak zwanych kontrolerek – EditText, Spinner oraz ProgBar.

Sprzet

Moduł BBMagic umieściłem w amplitunerze systemu 5.1 – AMPT-008 własnej konstrukcji. Sam amplituner został opisany w cyklu artykułów pt. „Amplituner kina domowego AMPL-008” w EP 5, EP 6 i EP 9 w 2013 roku. Amplituner ten jest wyposażony w procesor ARM typu LPC2387 oraz kodek typu WM8770 firmy Wolfson. Układ ten zawiera 1 przetwornik ADC STEREO (2 wejścia analogowe) oraz 4 przetworniki DAC STEREO (8 wyjść analogowych). Wejścia układu przełączane są za pomocą wbudowanego multiplexera analogowego – 8 wejść STEREO na 1 wyjście STEREO, które doprowadzone jest do układu PGA (regulacja wzmacnienia) i dalej na wejście przetwornika ADC.

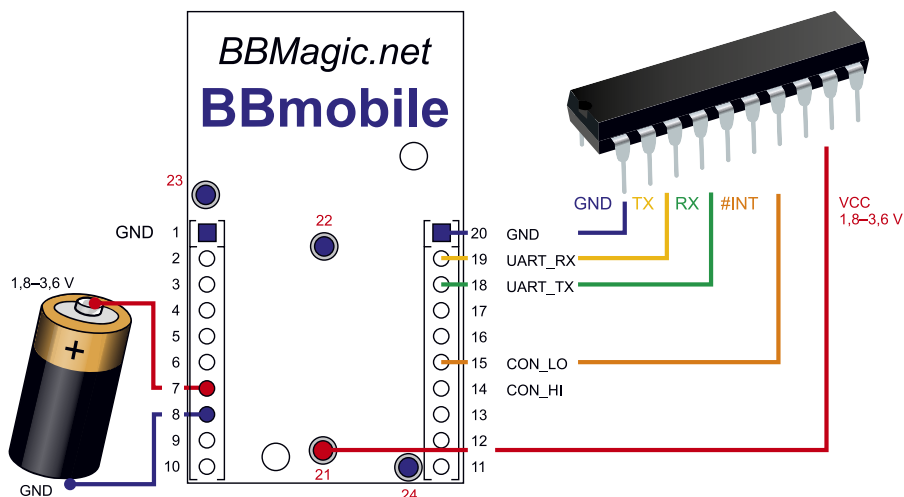
W moim amplitunerze umieściłem również cyfrowy dekodery Dolby Digital/DTS typu STA310 firmy STMicroelectronics. Układ jest wyposażony w wejście SPDIF i potrafi dekodować strumień danych w formacie PCM, Dolby Digital, DTS, MPEG i innych. Zawiera również procesor DSP, który może z 2 kanałów wejściowych (L, R) wygenerować 6 kanałów wyjściowych w formacie 5.1 (L, R, C, Sub, Ls, Rs). Procesor DSP może pracować w trybie Dolby Pro Logic lub Circle Surround.

Amplituner ma 5 wejść cyfrowych SPDIF (multiplexer cyfrowy doprowadza je na wejście układu STA310) i 4 wejścia analogowe STEREO (trzy z tyłu i jedno z przodu obudowy). Wyposażony jest w końcówki mocy typu D-Class. Dla kanałów głównych L i R zastosowałem układ typu TAS615, a dla kanałów C, Sub oraz Ls, Rs dwa układy typu TPA3123D2 produkcji firmy Texas Instruments. Jak widać z powyższego opisu, możliwości tej konstrukcji są dość szerokie. Amplituner wyposażyłem w wyświetlacz alfanumeryczny LCD 2x24 znaki. Jest to zdecydowanie małe „okno na świat” mojego amplitunera. Postanowiłem wykorzystać ekran smartfona jako alternatywny interfejs dla mojego amplitunera.

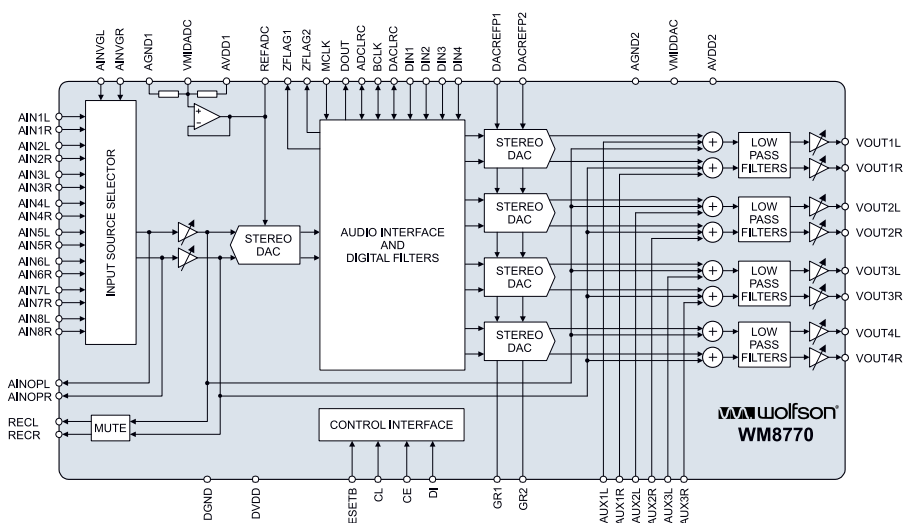
Aplikacja

Na wstępie chcę zwrócić uwagę na dwukierunkowość komunikacji pomiędzy smartfonem i amplitunerem. Moduł BBMagic podłączyłem do kanału UART2 procesora LPC2387. Sprawdzenie poprawności komunikacji można wykonać, wysyłając komendę <hello. Otrzymanie odpowiedzi >HI potwierdza prawidłowe połączenie i działanie modułu.

Na smartfonie (można do tego celu przeznaczyć nieużywany, starszy model) należy zainstalować aplikację BBMobile i włączyć Bluetooth. Po uruchomieniu aplikacji należy wybrać polecenie START, a następnie



Rysunek 1. Sposób podłączenia modułu BBMagic w wersji (z oprogramowaniem) BBMobile



Rysunek 2. Wewnętrzny schemat blokowy układu WM8770

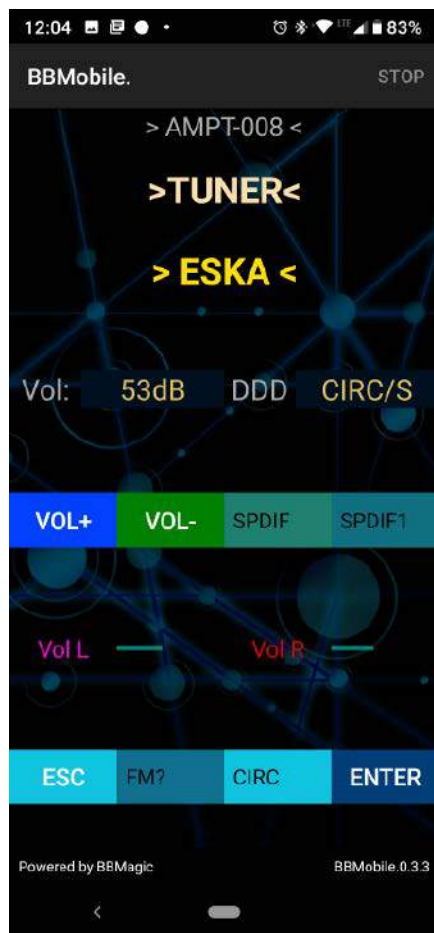
spośród znalezionych urządzeń (zwykle powinno być jedno) wybrać BBMobile. Od tego momentu aplikacja czeka na załadowanie poprzez Bluetooth tzw. Layoutu. W tym stanie w amplitunerze należy wywołać procedurę przesłania kodu Layout przez

interfejs UART2. W opisywanym zastosowaniu Layout jest definiowany kodem, który został pokazany na **listingu 1**.

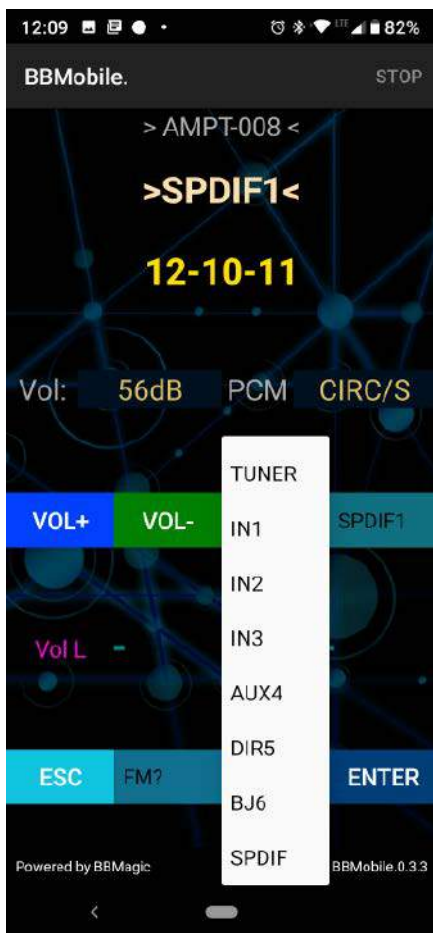
Główny blok Layout ma orientację pionową: "or":"v". Jako tło wybrałem image 6: "img":"6" (w programie zmieniam

Listing 1. Kod JSON definiujący Layout aplikacji mobilnej

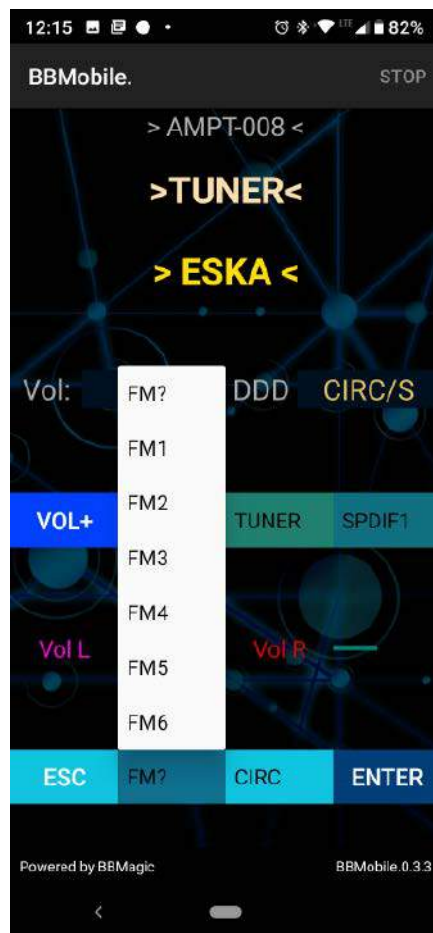
```
{
  "ty": "lout", "or": "v", "img": "6", "cs": [
    {
      "ty": "lout", "or": "v", "cs": [
        {
          "ty": "TextView", "n": "tv11", "te": "> AMPT-008 <", "tc": "174,174,174", "ts": "20", "w": "2",
          "ty": "TextView", "n": "tv12", "te": "INPUT", "tl": "bold", "tc": "255,224,174", "ts": "28", "w": "3",
          "ty": "TextView", "n": "tv13", "te": ">ABCDEFH<", "tl": "bold", "ts": "28", "tc": "255,224,0", "w": "3"
        }
      ]
    },
    {
      "ty": "lout", "or": "H", "cs": [
        {
          "ty": "TextView", "n": "tv21", "te": "Vol:", "tc": "174,174,174", "ts": "18", "w": "1",
          "ty": "TextView", "n": "tv22", "te": "99dB", "tc": "224,192,96", "bg": "0,16,32", "ts": "22", "w": "2",
          "ty": "TextView", "n": "tv23", "te": "DEC:", "tc": "174,174,174", "ts": "18", "w": "1",
          "ty": "TextView", "n": "tv24", "te": "PCM", "tc": "224,192,96", "bg": "0,16,32", "ts": "22", "w": "2"
        }
      ]
    },
    {
      "ty": "lout", "or": "H", "cs": [
        {
          "ty": "Button", "n": "b31", "te": "Vol+", "tc": "255,255,255", "bg": "0,64,255", "ts": "18",
          "ty": "Button", "n": "b32", "te": "Vol-", "tc": "255,255,255", "bg": "0,128,0", "ts": "18",
          "ty": "Spinner", "n": "sp33", "is": "IN?, TUNER, IN1, IN2, IN3, AUX4, DIR5, B36, SPDIF", "bg": "32,128,112", "ts": "22", "w": "1",
          "ty": "Spinner", "n": "sp34", "is": "SPDIF?, SPDIF1, SPDIF2, SPDIF3, SPDIF4, SPDIF5", "bg": "16,112,128", "ts": "22", "w": "1"
        }
      ]
    },
    {
      "ty": "lout", "or": "H", "cs": [
        {
          "ty": "TextView", "n": "tv41", "te": "Vol L", "tc": "174,255,174", "ts": "18", "w": "1",
          "ty": "ProgBar", "n": "pb1", "d": "1", "p": "25", "w": "1",
          "ty": "TextView", "n": "tv42", "te": "Vol R", "tc": "174,255,174", "ts": "18", "w": "1",
          "ty": "ProgBar", "n": "pb2", "d": "1", "p": "25", "w": "1"
        }
      ]
    },
    {
      "ty": "lout", "or": "H", "cs": [
        {
          "ty": "Button", "n": "b51", "te": "ESC", "tc": "255,255,255", "bg": "16,196,224", "ts": "18", "w": "1",
          "ty": "Spinner", "n": "sp52", "is": "FM?, FM1, FM2, FM3, FM4, FM5, FM6", "bg": "16,112,144", "ts": "22", "w": "1",
          "ty": "Spinner", "n": "sp53", "is": "DPL?, OFF, ProL, CIRC, DSTR0, DSLs, DSRs", "bg": "16,196,224", "ts": "22", "w": "1",
          "ty": "Button", "n": "b53", "te": "ENTER", "tc": "255,255,255", "bg": "0,62,120", "ts": "18", "w": "2"
        }
      ]
    }
  ]
}
```



Rysunek 3. Ekran początkowy interfejsu mobilnego dla amplituneru AMPT-008



Rysunek 4. Ekran interfejsu mobilnego, wybór wejścia AUDIO



Rysunek 5. Ekran interfejsu mobilnego, wybór zapamiętanej stacji radiowej

pseudo-losowo cyfrę 6 na wartość z zakresu 1...6, tło image 7 pomijam, ponieważ jest bardzo jasne). W głównym bloku Layout zaagnieżdżonych jest 5 kolejnych Layoutów.

Pierwszy ma orientację pionową "or": "v" i zawiera 3 kontrolki typu TextView: "tv11", "tv12", "tv13". W trakcie pracy amplitunera pole tv12 będzie zawierało nazwę aktualnego wejścia np. >TUNER<, >SPDIF1< itp. Aby zmienić nazwę wyświetlanego aktualnego wejścia np. na >SPDIF1<, należy wysłać do modułu następującą sekwencję znaków: `$set,tv12:te=">SPDIF1<"\r\n`

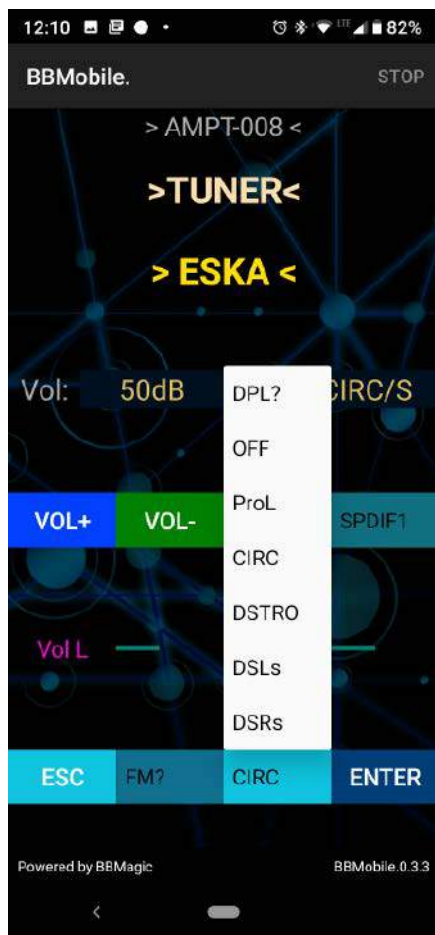
Pole tv13 będzie używane do wyświetlania informacji RDS lub aktualnego czasu.

Drugi blok Layout ma orientację poziomą: "or": "h" i zawiera 4 kontrolki typu TextView: tv21...24. Szerokość pól dla tych napisów jest różna ("w": "1" ... "w": "2" ...) i wynosi odpowiednio: 1/6, 2/6, 1/6 i 2/6. Powoduje to, że szerokość zajmowana przez napisy tv22 i tv24 jest dwa razy większa od szerokości napisów tv21 i tv23. Pola tv21 – Vol: oraz tv23 – DEC pozostaną niezmienione, ponieważ są to pola opisowe. Pola tv22 oraz tv24 będą zawierały aktualne wartości, odpowiednio: głośności i wybranego dekodera. Aby wyświetlić np. wartość głośności równą 70 dB, należy wysłać do modułu następującą sekwencję znaków: `$set,tv22:te="70dB"/r/n`

Trzeci blok Layout również ma orientację poziomą i zawiera dwie kontrolki typu Button. Są to przyciski Vol+ oraz Vol– odpowiedzialne za regulację głośności oraz dwie kontrolki typu Spinner, które zawierają rozwijaną listę wyboru. Kontrolka Spinner sp33 pozwala wybrać wejście AUDIO: TUNER, IN1...SPDIF. Kontrolka Spinner sp34 umożliwia wybór konkretnego (domyślnego) wejścia SPDIF: SPDIF1, SPDIF2...5. Wyboru odpowiedniej pozycji z rozwijanej listy dokonujemy na ekranie smartfona dotykowo. Po wybraniu którejś linii rozwijana lista się zwiija. O tym, że dokonaliśmy jakiegoś wyboru, nie jest automatycznie przesyłana żadna informacja. Aby zatwierdzić i przesłać wybór, naciskamy dowolny przycisk – kontrolkę Button. Specjalnie do tego celu umieściłem w ostatnim rzędzie przycisk ENTER. Po naciśnięciu tego przycisku moduł BBMagic prześle przez UART taką informację: `$by,b53"1",sp33"TUNER",sp34"SPDIF1",sp52"FM?",sp53"DPL?"` standardowo zakończoną znakami `\r\n`. Oczywiście informacje z kontrolki sp33, sp34, sp52 i sp53 są przesyłane po naciśnięciu dowolnego przycisku. Dla przycisku Vol+ przesłana informacja wygląda następująco: `$by,b31"1",sp33"TUNER",sp34"SPDIF1",sp52"FM?",sp53"DPL?"`

Tabela 1. Dokładny opis kontrolki typu Spinner – "ty": "Spinner"

Nazwa pola	Mnemonik	Opis
Type	ty	Określa typ kontrolki. W tym przypadku jest to Spinner – pole umożliwiające wybór jednego elementu z wielu.
Name	n	Umożliwia nadanie kontrolce unikalnej nazwy służącej do jej identyfikacji w interfejsie, np. "s1".
Mode	m	0 – kontrolka będzie miała postać okna dialogowego; 1 lub brak parametru – kontrolka będzie miała postać listy.
Items	is	Lista zawierająca elementy wyboru rozdzielone przecinkami, np. „jeden,dwa,trzy,cztery”.
Weight	w	Definiuje wielkość kontrolki jako ułamkową część całości widoku, gdzie licznik ułamka to wartość parametru 'w', a mianownik to suma wartości parametrów 'w' w wszystkich kontrolkach znajdujących się w Layoutcie.



Rysunek 6. Ekran interfejsu mobilnego, wybór trybu pracy procesora audio

Program obsługi zaimplementowany w amplitunerze dla przycisków Vol+ i Vol- nie analizuje danych z kontrolki typu Spinner, ponieważ ich działanie ograniczam do regulacji głośności. Te dane są analizowane tylko po naciśnięciu przycisku ENTER. Dokładny opis kontrolki Spinner jest zawarty w tabeli 1.

Czwarty blok Layout zawiera dwie kontrolki TextView z napisami Vol L i Vol R oraz dwa razy nowy typ kontrolki: ProgBar. Służą one do wyświetlania paska postępu, tzw. *progress bar*. Postanowiłem zastosować te kontrolki do przedstawienia w sposób graficzny poziomu sygnału analogowego w kanale lewym i prawym amplitunera. Długość rysowanego paska należy podać w procentach (0...100).

Tabela 2. Dokładny opis kontrolki typu ProgBar – "ty": "ProgBar"

Nazwa pola	Mnemonik	Opis
Type	ty	Określa typ kontrolki. W tym przypadku jest to ProgBar – pasek wizualizacji postępu.
Name	n	Umożliwia nadanie unikalnej nazwy kontrolce służącej do jej identyfikacji w interfejsie, np. "p1".
Determine	d	0 lub brak parametru – kontrolka sygnalizuje postęp bez określania stopnia wykonania; 1 – kontrolka pokazuje postęp określający stopień wykonania.
Progress	p	Aktualny postęp w procentach: przedział od 0 do 100.
Weight	w	Definiuje wielkość kontrolki jako ułamkową część całości widoku, gdzie licznik ułamka to wartość parametru 'w', a mianownik to suma wartości parametrów 'w' wszystkich kontrolki znajdujących się w Layoucie.

Dokładny opis kontrolki typu ProgBar jest zawarty w tabeli 2. Parametr *Determine* określa wygląd kontrolki ProgBar. Dla wartości 0 pasek postępu przyjmuje postać obracającego się wycinka okręgu i wygląda jak sygnał zajętości (zawieszenia się) aplikacji. Dla wartości 1 pasek postępu przyjmuje postać linii o długości proporcjonalnej do wartości parametru *Progress*. Nie mamy wpływu na kolor i grubość tej linii. Odświeżanie kilka razy na sekundę parametru *p* (*Progress*) pozwala wyświetlić namiastkę analogowego wskaźnika poziomu sygnałów audio w lewym i prawym kanale. Aby sterować wyświetlaniem, ustawiamy aktualną wartość parametru *progress* przez wysłanie do modułu (dla chwilowej wartości 75%) następującej sekwencji znaków: `$set,pb1:p="75"/r/n`

W ostatnim bloku Layout znalazły się dwa przyciski: ESC i ENTER oraz dwie rozwijane listy typu Spinner. Pierwsza sp52 pozwala określić numer zapamiętanej częstotliwości dla 6 stacji radiowych: FM1, FM2...FM6. Druga lista pozwala wybrać tryb pracy cyfrowego procesora audio (STA310): OFF, ProL (Dolby Pro Logic), CIRC (Circle surround).

Można zauważyć, że początek każdej listy zaczyna się od neutralnych pozycji "IN?", "SPDIF?", "FM?" oraz "DPL?". Są one potrzebne, ponieważ zawsze otrzymujemy informację o wszystkich polach Spinner, a nie zawsze chcemy zmieniać cztery parametry. Dlatego jeśli np. chcemy zmienić tylko numer wejścia analogowego na np. trzecie, to ustawiamy pierwsze pole wyboru na IN3

a pozostałe zostawiamy (lub ustawiamy) na neutralne. Czyli odpowiednio na: „SPDIF?”, „FM?” oraz „DPL?”. Po naciśnięciu przycisku ENTER wiemy, że chodzi tylko o ustawienie numeru wejścia analogowego na IN3, a numer domyślnego wejścia SPDIF pozostanie niezmieniony. Analogicznie pozostawimy niezmieniony domyślny numer częstotliwości radia FM oraz nie zmienimy trybu pracy procesora audio.

Podsumowanie

Należy pamiętać, że po odwróceniu smartfona o 90° nasz Layout zostanie przerysowywany do pozycji horyzontalnej. Należy przygotować naszą aplikację także do takiej pozycji smartfona, ponieważ po odwróceniu niektóre napisy mogą stać się nieczytelne.

Chcę jeszcze wspomnieć o kontrolce EditText (której ostatecznie nie użyłem). Pozwala ona wyświetlić dowolny tekst – podobnie jak kontrolka Spinner, ale tekst należy wprowadzić. Informacja o wprowadzonym tekście zostanie przesłana dopiero po naciśnięciu na dowolny przycisk.

Trzeba przyznać, że dość tanim kosztem (moduł BBMagic można kupić za około 55 złotych) można prawie dowolnie urządzić nie wyposażać w interfejs z ekranem dotykowym. Polecam użycie do tego celu „drugiego” smartfona. Zwykle każdy z nas zmienia co 2 lata swój smartfon na nowy. Warto stary, ale wciąż sprawny smartfon zastosować jako interfejs z aplikacją BBMobile.

Zbyszko Przybył
zbyszekjob@interia.pl

Chcesz czytać nasze najnowsze artykuły jeszcze przed wydrukowaniem w EP?

Zajrzyj na

www.ep.com.pl/EPwtoku

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. Przysyłanych tekstów nie zwracamy. Redakcja zastrzega sobie prawo do dokonywania skrótów.

Hybryda bis

– wzmacniacz hybrydowy z niekonwencjonalnym zasilaniem

W styczniowym wydaniu „Elektroniki Praktycznej” (EP 1/21) zaprezentowałem hybrydowy wzmacniacz lampowy. Tym razem przygotowałem całkiem odmienną wersję takiego rozwiązania. Różni się przede wszystkim tym, że nie jest zasilany transformatorem sieciowym, lecz przetwornicami impulsowymi o wysokiej sprawności. Dzięki temu wzmacniacz pracuje wydajniej, a grzeją się tylko lampy 6SN7.

Wnętrze wzmacniacza zostało pokazane na fotografii 1. Konstrukcja składa się z ośmiu modułów w postaci oddzielnych płytek PCB. Poszczególne moduły zostały oznaczone tak jak na rysunku 1. Na blok zasilania składają się:

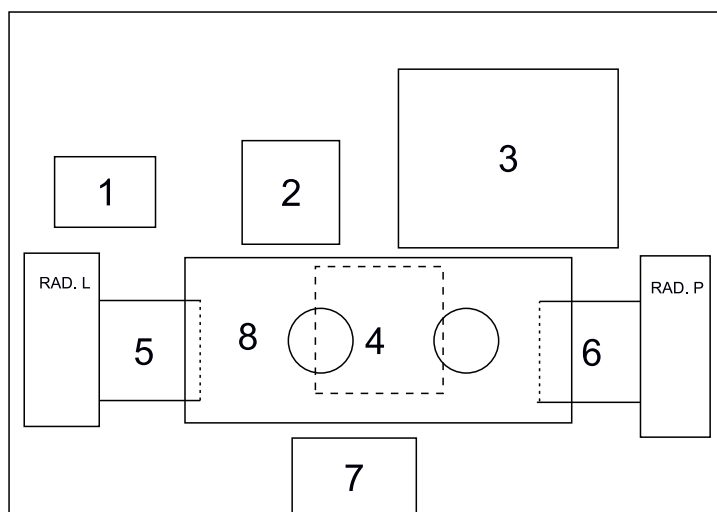
- **Przełącznik zasilania** (moduł nr 1), który załącza za pomocą wyłącznika głównego napięcie zasilania 12 V. Rozwiązanie to jest podyktowane tym, że układ przy pełnymysterowaniu pobiera prąd o wartości nawet 10 A;
- **Przetwornica podwyższająca** (moduł nr 2). Moduł o oznaczeniu STEP-UP DC 12 V AC 230 V w rzeczywistości okazał się urządzeniem, które daje na wyjściu napięcie stałe o wartości około 340 V. Dzięki temu znakomicie nadaje się do zasilania przedwzmacniacza lampowego;
- **Zasilacz napięcia symetrycznego** (moduł nr 3), który może dostarczyć napięć z zakresu $\pm 16... \pm 26$ V. Ze względu na zastosowane w dalszej części układu kondensatory elektrolityczne na napięcie 25 V, napięcie symetryczne zostało ustawione na wartość ± 24 V;
- **Kondensatory filtrujące** (moduł nr 4). Przetwornica napięcia symetrycznego ma na wyjściu zbyt małe pojemności, aby wzmacniacz mocy pracował wydajnie, więc była konieczność

dołączenia dodatkowych kondensatorów elektrolitycznych.

Tor audio to pozostałe moduły:

- **Wzmacniacze mocy** (moduły nr 5 i 6) zostały zbudowane z użyciem scalonych stopni typu LM3886. Mają one

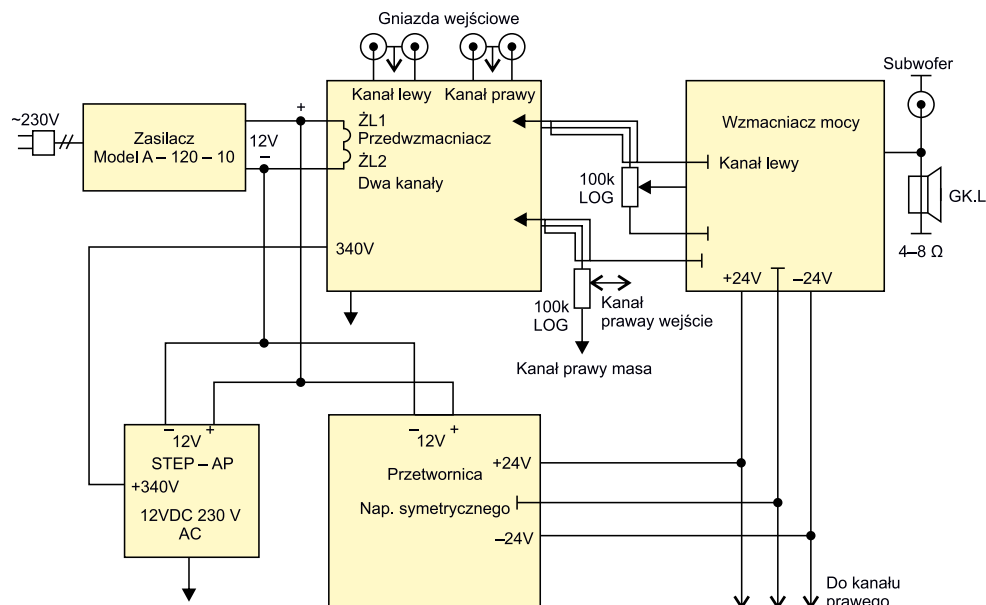
doskonałe parametry audio i mogą pracować w wielu aplikacjach m.in. dzięki szerokiemu wachlarzowi napięcia zasilania (12...40 V). Dodatkowo układ został wyposażony w szereg zabezpieczeń – jest odporny na zwarcie wyjścia



Rysunek 1. Oznaczenie i rozmieszczenie modułów składowych wzmacniacza

głośnikowego czy przegrzanie. Jest to bardzo istotne, ponieważ układ wymaga radiatora, który musi mieć kilkadziesiąt centymetrów kwadratowych. W prezentowanym wzmacniaczu zastosowano typowy radiator o długości 8 cm.

- **Potencjometr** (moduł nr 7). W układzie zastosowano potencjometr drabinkowy $2 \times 100 \text{ k}\Omega$. Zaletą tego elementu jest duża współbieżność (1%) w funkcji kąta obrotu, natomiast wadą jest duży skok głośności na ostatnim stopniu, ale nie jest to zbyt uciążliwością. Masy potencjometru poprowadzono osobno dla lewego i prawego kanału. Połączenia mas bloków wzmacniacza mocy i przedwzmacniacza zrealizowano za pomocą ekranów na przewodach połączeniowych. Zwiększa to znacznie tłumienie przesłuchu między kanałami, które wynosi około 40 dB.
- **Przedwzmacniacz lampowy** (moduł nr 8) został zrealizowany w układzie



Rysunek 2. Schemat blokowy wzmacniacza



Fotografia 1. Widok wewnętrznej konstrukcji wzmacniacza



Fotografia 2. Widok tylnej ścianki wzmacniacza

Wykaz elementów:

Rezystory: (metalizowane $0,25 \text{ W}$, chyba że wskazano inaczej)

R1, R3: 220Ω
 R1A: $1,5 \text{ k}\Omega$
 R2, R4: $1 \text{ M}\Omega$
 R5, R8: $68 \text{ k}\Omega$
 R6, R9: $1,2 \text{ k}\Omega$
 R7, R10: $4,7 \text{ k}\Omega$
 R11, R12: 470Ω
 R13, R14, R15, R17, R19, R20: $10 \text{ k}\Omega$
 R16, R18: 470Ω
 R21, R22: 10Ω , 1 W
 Pot: potencjometr $100 \text{ k}\Omega$ drabinka log.

Kondensatory:

C1, C2, C17, C19, C22, C23: $100 \text{ nF}/100 \text{ V}$ MKT
 C4, C6: $100 \text{ nF}/250 \text{ V}$ MKT
 C8, C9: $470 \text{ nF}/250 \text{ V}$ MKT
 C10, C11: $470 \text{ nF}/100 \text{ V}$ MKT
 C3: $10 \mu\text{F}/400 \text{ V}$ elektrolityczny
 C5, C7: $10 \mu\text{F}/250 \text{ V}$ elektrolityczny
 C12, C13: $470 \mu\text{F}/16 \text{ V}$ elektrolityczny
 C14: $10 \mu\text{F}/25 \text{ V}$ tantalowy
 C15, C16: $10000 \mu\text{F}/25 \text{ V}$ elektrolityczny
 C10A, C11A, C24, C26: $22 \mu\text{F}/25 \text{ V}$ elektrolityczny
 C18, C20, C21, C25: $470 \mu\text{F}/25 \text{ V}$ elektrolityczny

Półprzewodniki:

D1: dioda zielona fi 5 mm
 US1, US2: LM3886

Pozostałe:

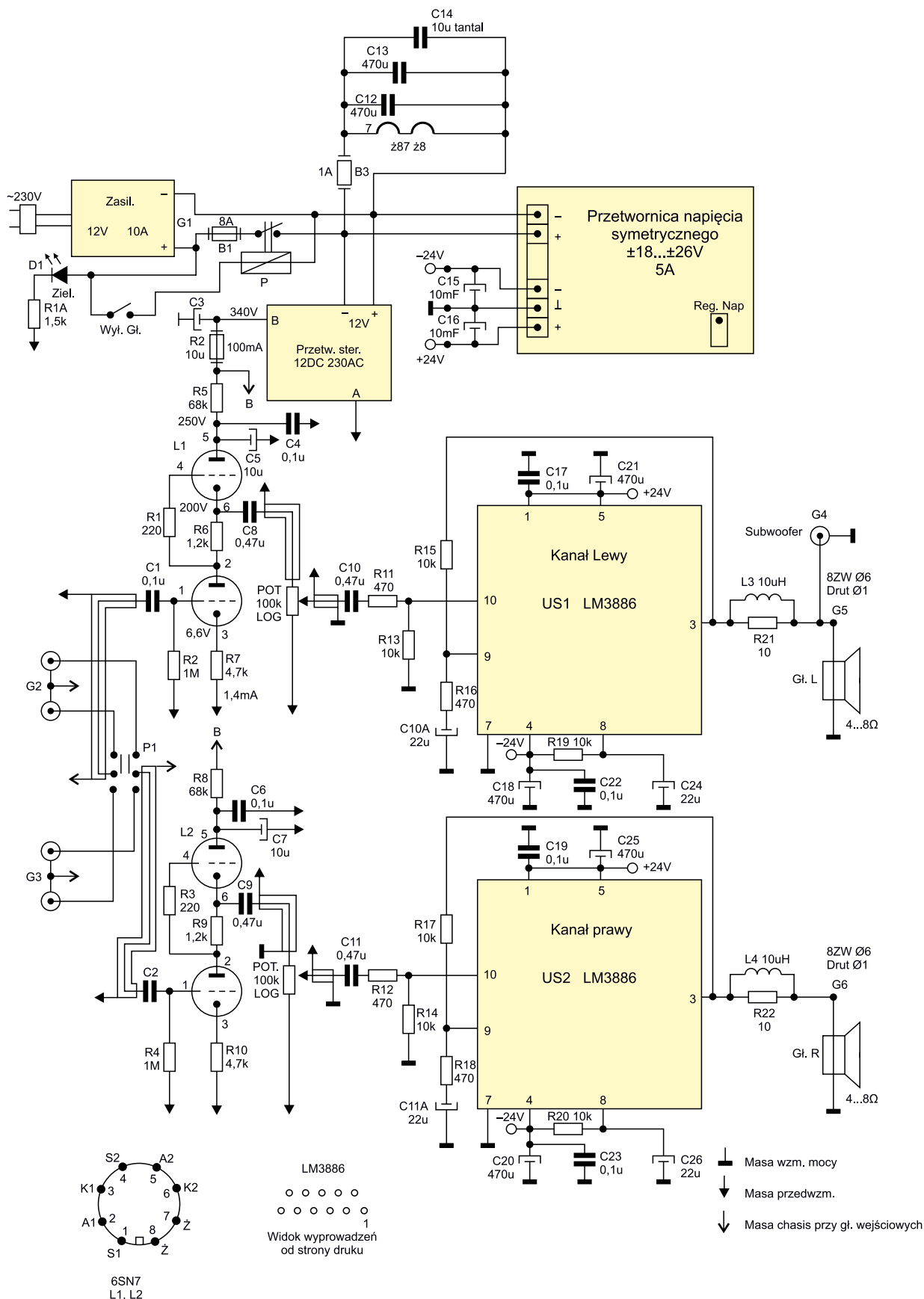
L1, L2: lampy typu 6SN7 TUNGSOŁ
 L3, L4: dławik powietrzny $10 \mu\text{H}$
 P: przekaźnik $12 \text{ V DC } 10 \text{ A}$
 P1: przełącznik przesuwany podwójny
 WYŁ. GŁ.: wyłącznik przechylny migowy $6 \text{ A } 250 \text{ V}$
 G1: gniazdo DC do obudowy $2,1 \times 5,5 \text{ mm}$
 G2, G3, G4: gniazdo cinch do obudowy
 G5, G6: gniazdo głośnikowe
 B1: gniazdo bezpiecznikowe w obudowie T261 + bezpiecznik 8 A szybki
 B2: gniazdo bezpiecznikowe do druku 5×20 + bezpiecznik 100 mA szybki
 B3: gniazdo bezpiecznikowe do druku 5×20 + bezpiecznik 1 A szybki
 zasilacz impulsowy model A-120-10 A
 przetwornica napięcia symetrycznego $\pm 16... \pm 26 \text{ V}$
 przetwornica step-up $12 \text{ V } 230 \text{ V AC}$

dwulampowym dla każdego kanału. Jest party na schemacie zaprezentowanym w wydaniu „Elektroniki Praktycznej” 1/2012 przez pana Jerzego Grnaderjana. Taka konfiguracja jest określana jako TTSA (Two Tubes Series Amplifier)

lub czasami SRPP (Shut Regulated Push Pull). Odsyłam do tego bardzo ciekawego artykułu, ponieważ są tam opisane różne warianty wórników katodowych nazywanych humorystycznie akrobatyką lampową (lampa na lampie).

Budowa i działanie

Schemat blokowy wzmacniacza został pokazany na **rysunku 2**. Dokładnie pokazuje on połączenia pomiędzy modułami. Natomiast schemat ideowy został pokazany na **rysunku 3**. Górna trioda jest dynamicznym



Rysunek 3. Schemat ideowy wzmacniacza

obciążeniem dolnej lampy, przez co układ ma dużą impedancję wejściową i małą impedancję wyjściową, co jest potrzebne do dobrego dopasowania impedancyjnego bloków wzmacniaczy.

Układ charakteryzuje się również tym, że przenosi symetrycznie sygnał. Po podaniu na wejście sygnału 1 kHz o wartości około 4,5 V RMS otrzymuje się nieznkształcony sygnał o wartości około 40 V RMS, co daje amplitudę Vpp ok. 114 V. Zwiększenie sygnału powoduje idealnie symetryczne obcinanie sinusoidy. Układ sprawdzono również pod względem pasma przenoszenia i wyniki wykazały pasmo 4 Hz...40 kHz przy spadku wzmocnienia 3 dB.

Jako stopnie mocy można zastosować gotowe wzmacniacze do zmontowania w postaci kitów AVT1922. Uprości to znacznie budowę urządzenia. Pomiar mocy wykazał, że na rezystorze 8 Ω odkłada się moc ok. 24 W RMS. Jest to wystarczająca wartość do nagłośnienia średniej wielkości pomieszczenia. Układ realizuje funkcję tłumienia stanów nieustalonych przy włączaniu zasilania. Odpowiadają za to elementy R19 i C24 w lewym kanale oraz R20 i C26 w prawym. Wzmocnienie układu ustalają elementy R15, R16 i C10A w lewym, a R17, R18 i C11A w prawym. Daje to czułość na poziomie 0,7 V.

Potencjometr włączony jest pomiędzy wzmacniaczem mocy a układem sterującym na lampach 6SN7. Skutkuje to minimalnymi zakłóceniami wnoszonymi przez przedwzmacniacz.

Cały wzmacniacz ma czułość 200 mV. Na wejście, poprzez przełącznik P1, można podać sygnał z odtwarzacza CD, gramofonu analogowego lub adaptera Bluetooth, którym można sterować laptopem lub smartfonem. Jednak należy pamiętać, że dopiero podłączenie dobrych kolumn głośnikowych zapewni właściwe odbieranie dźwięku.

Henryk Michałowski
michalowskihenio@gmail.com

REKLAMA

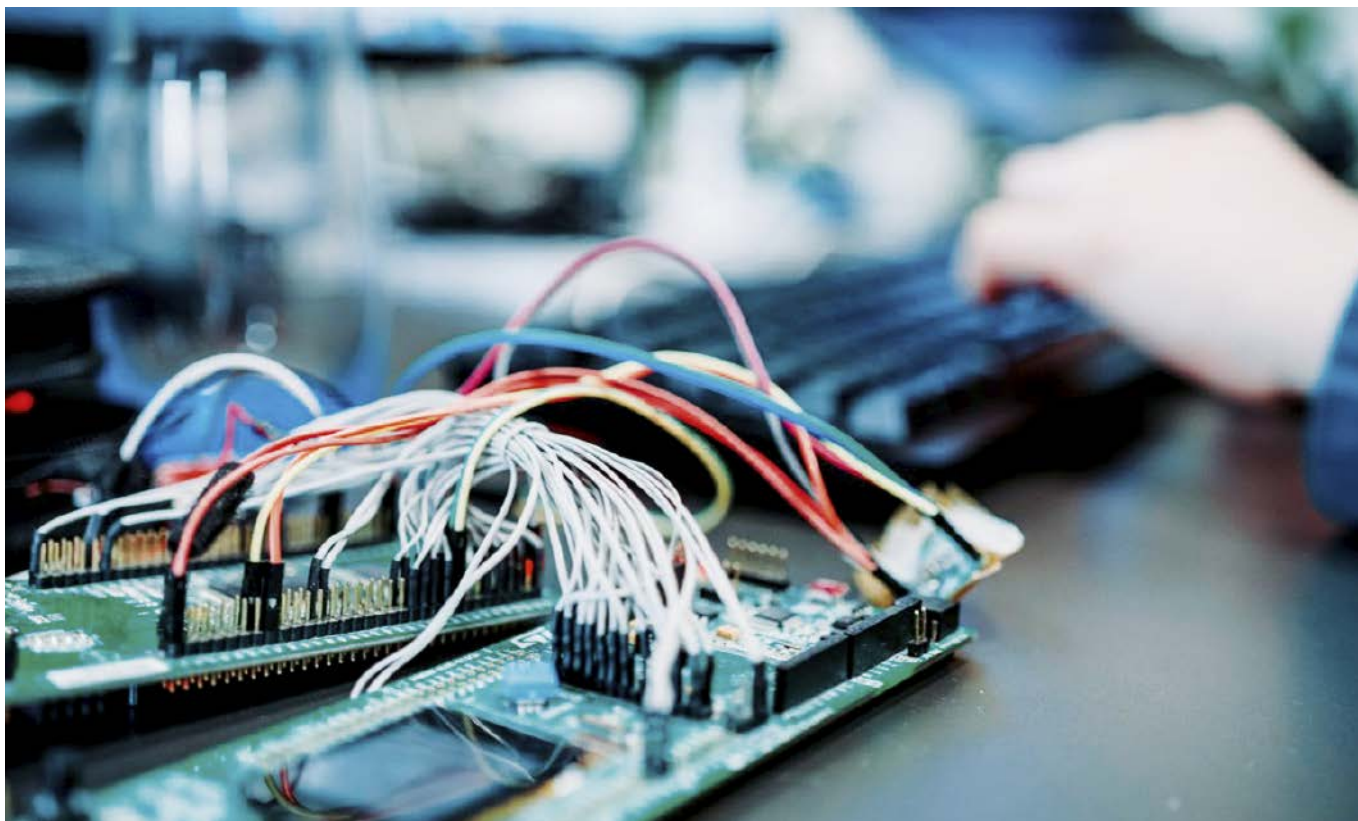
Ulubiony Kiosk zmienił się na lepsze!



Nowy, lepszy Ulubiony Kiosk stał się faktem! Nowy layout został zaprojektowany zgodnie z wiodącymi trendami, ale przede wszystkim w taki sposób, by jak najbardziej ułatwić przeglądanie i nawigowanie po serwisie.

Jednak nowy Ulubiony Kiosk to także zastosowaniu nowych technologii. Nasz serwis będzie teraz właściwie skalował się na ekranach o różnej rozdzielczości oraz lepiej wyświetlał na urządzeniach mobilnych.

Nie czekaj i wejdź na www.UlubionyKiosk.pl!



ARM Keil – bezpieczeństwo w systemach IoT i embedded

Warunkiem wprowadzenia na rynek wielu produktów, w branżach takich jak: AGD, motoryzacja, przemysł czy medycyna, jest spełnienie wymogów prawnych potwierdzających ich zgodność z normami bezpieczeństwa funkcjonalnego. Jednym z elementów procesu certyfikacji jest zabezpieczenie kodu urządzenia przed wykonaniem niewłaściwej operacji, jak również blokowanie dostępu przed nieautoryzowaną próbą przejęcia nad nim kontroli. Bezpieczeństwo w działaniu aplikacji wbudowanych jest więc kluczowym tematem podczas certyfikowania produktów.

W szczególności w ostatnim czasie, gdyż rosnąca liczba urządzeń IoT wprowadzanych na rynek wymusza działania również w tym zakresie. Rządy krajów zaczynają prace nad ustandaryzowaniem zabezpieczeń, a także nad określeniem wymogów, które będą spełniały aplikacje IoT. Informacje zaczynają napływać ze Stanów Zjednoczonych, o „rozsądnych zabezpieczeniach” i potrzebie ich stosowania w urządzeniach używanych przez instytucje rządowe. Jednak takie działania pozwala przypuszczać, że podobne rozwiązania zostaną wprowadzone także w innych krajach i będą dotyczyły zarówno przemysłu, jak rynku konsumenckiego.

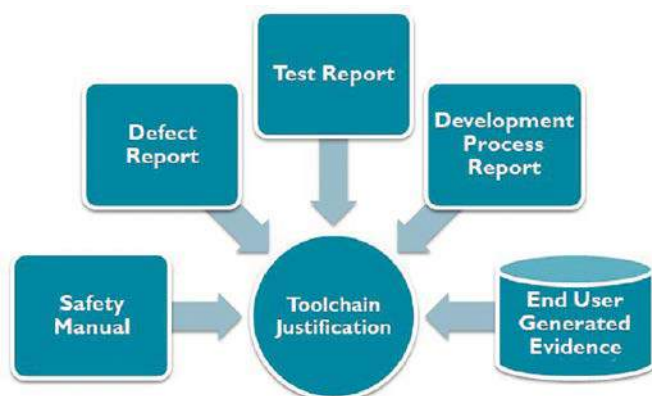
Bezpieczeństwo w aplikacjach IoT

IoT, czyli Internet Rzeczy, oznacza system oraz sieć połączonych urządzeń elektronicznych zdolnych do wymiany informacji pomiędzy

sobą w sposób automatyczny. Urządzenia IoT pozwalają na gromadzenie danych, udostępnianie ich użytkownikom oraz na komunikację przez Internet. Dlatego ważnym aspektem jest ochrona informacji i prawidłowa praca sprzętu podczas całego czasu użytkowania. W celu osiągnięcia tej funkcjonalności przydatna może być technologia TrustZone korzystająca z rozwiązania SoC (System on Chip) i podejście do bezpieczeństwa obejmujące cały system. Technologia TrustZone dla procesorów ARM Cortex-M zapewnia wysoki poziom ochrony. Zmniejsza także ryzyko ataku, izolując krytyczne oprogramowanie sprzętowe, zasoby i prywatne informacje od reszty aplikacji. Stanowi doskonały punkt wyjścia dla urządzenia w oparciu na wytycznych Platform Security Architecture (PSA). Do urządzeń IoT możemy zaliczyć między innymi inteligentne urządzenia typu smart home. Nietrudno zauważyć, że takie rozwiązania, choć bardzo przydatne człowiekowi, jak otwieranie bramy domu przy użyciu smartfonu, umożliwiają działanie osobom niepożądanym np. próby włamania się do obiektu. Jeszcze poważniej sprawa wygląda w przemyśle i medycynie.

Środowisko bezpiecznych aplikacji ARM Keil MDK Professional

Biorąc pod uwagę fakt, że bezpieczeństwo programu jest istotnym elementem podczas procesu certyfikacji Functional Safety, z pomocą w tym zakresie mogą przyjść bezpieczne rozwiązania firmy ARM. Należą do nich między innymi: certyfikowane kompilatory C, C++, zestaw kwalifikacyjny (Qualification Kit), modele symulacyjne czy optymalizacje systemu RTOS oraz technologia TrustZone.



Rysunek 1. Zestaw kwalifikacyjny to narzędzia pozwalające przejść do bezpiecznych rozwiązań

Warto dodać, że wymienione funkcjonalności może obsługiwać jedno wspólne środowisko, którym jest μ Vision MDK Professional.

ARM Keil rozwija narzędzia programistyczne dla układów opartych na architekturze ARM, dlatego zawsze zna najnowsze rozwiązania, udostępniając je w postaci paczek do ściągnięcia w swoim zintegrowanym środowisku. Narzędzia producenta używane są przez ogromną liczbę developerów, tworzących aplikacje dla produktów w lotnictwie, bankowości, motoryzacji, transporcie, produkcji, wojsku czy telekomunikacji.

Jak wykonać zmianę do kompilatorów bezpiecznych

Posiadacz aktualnej licencji środowiska μ Vision MKD Professional ma możliwość, bez dodatkowych kosztów, przejść do bezpiecznych rozwiązań. Może to wykonać, korzystając z zestawu kwalifikacyjnego, który zawiera: instrukcję bezpieczeństwa, raport procesu rozwoju, raporty z testu i raport usterek (rysunek 1). Są to narzędzia, które obejmują ponad 250 stron dokumentacji w formie wyjaśnień i potwierdzeń dotyczących zastosowań bezpiecznych kompilatorów. Należy pamiętać, że żaden kompilator nie pracuje w 100% bezbłędnie, jednak ARM dokłada wszelkich starań, aby tak było. Potwierdzeniem tych prac jest certyfikat wydany przez organizację TÜV.

Aby skorzystać z narzędzi zestawu kwalifikacyjnego, należy założyć profil na portalu developer.arm.com, a następnie ściągnąć narzędzia w najnowszej wersji dla Functional Safety, czyli v6.6.4. Na stronie znajdują się kolejno do pobrania: certyfikowany kompilator, zestaw kwalifikacyjny oraz certyfikat TÜV. Istnieje również możliwość pobrania poprzednich wersji narzędzi.

Po zgraniu paczek, w środowisku μ Vision należy wejść w menu Project-Manage-Project Items, gdzie dodajemy nowy kompilator do listy. Z kolei w Options for Target możemy wybrać kompilator, z którego chcemy korzystać w projekcie. Po jego dodaniu będzie to: v6.6.4 Long Term Maintenance (rysunek 2). Następnie należy skompilować projekt. Może wystąpić sytuacja, w której będzie potrzebne dostosowanie części kodu pod działanie nowego kompilatora, wtedy pomocna okaże się dokumentacja techniczna, dostępna w ściągniętej paczce.

Dodatkowe informacje

Więcej informacji znajduje się na kanale YouTube – „ARM Keil MDK #24 – zmiana rodzaju kompilatora”. W ofercie ARM występuje dodatkowo oprogramowanie FuSa RTS, które jest zestawem komponentów zakwalifikowanych do stosowania w najbardziej krytycznych dla bezpieczeństwa aplikacjach, w systemach motoryzacyjnych, medycznych i przemysłowych. Kolejną opcję zakupową stanowi ARM Development Studio, które zostało zaprojektowane specjalnie dla architektury ARM, zgodnie z zasadą „jeden zestaw narzędzi – dowolny projekt oparty na ARM”. Przyspiesza projektowanie, pomagając jednocześnie tworzyć solidne i wydajne produkty. Licencja kompilatora Functional Safety dostępna jest dla oprogramowania ARM Development Studio w wersji Gold lub Platinum.

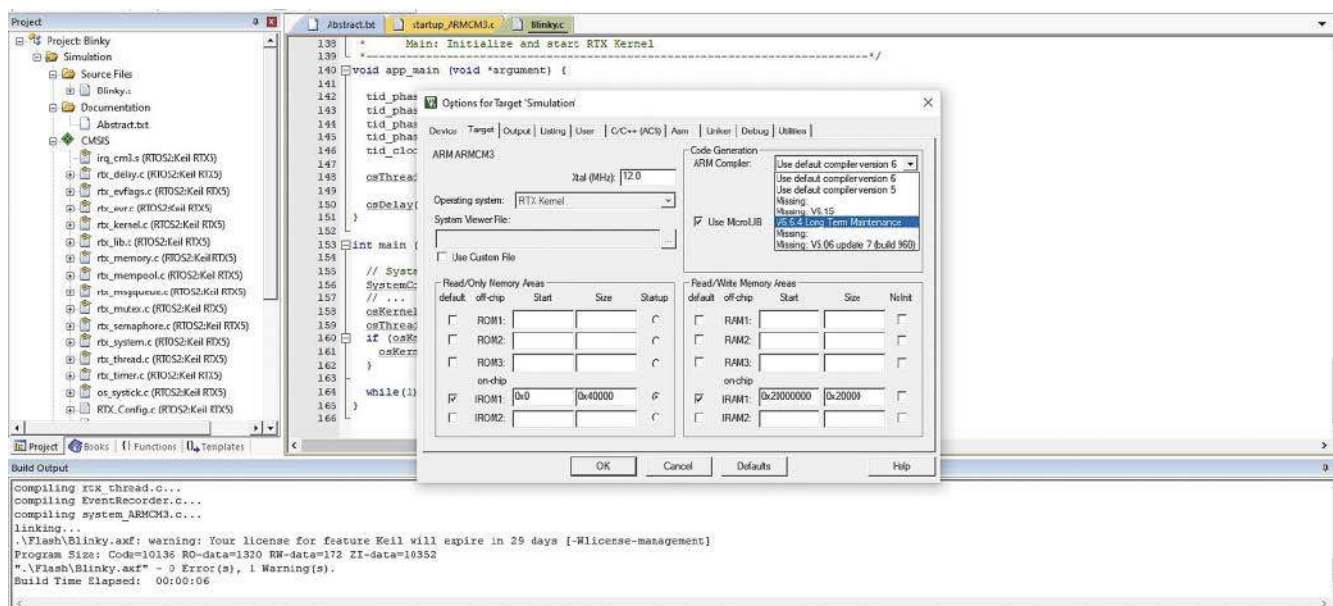
Wnioski

Podsumowując, rynek aplikacji IoT rośnie z roku na rok w bardzo szybkim tempie. Powstające urządzenia współpracują ze sobą, komunikując się i wymieniając dane. Takie działanie przynosi szereg korzyści użytkownikom, dostarczając sprawnie informacji czy też umożliwiając automatyzację pracy. Jednak wdrażanie aplikacji IoT stwarza także pewne zagrożenia. Dlatego tak ważne jest, aby zapewnić bezpieczeństwo na poziomie działania programu. Istnieje możliwość, aby usprawnić tworzenie bezpiecznego kodu przy użyciu certyfikowanych kompilatorów. Można również pracować z dostarczoną dokumentacją i korzystać z wyników testów czy zaleceń stworzonych dla zapewnienia bezpieczeństwa funkcjonalnego. Zastosowanie rozwiązań ARM dostarczy wielu korzyści, których efektem będzie opcja zakończenia projektu dla urządzenia w krótszym czasie. Dostępność i przedłużone wsparcie produktu pozwolą skorzystać z tych rozwiązań również w przyszłości, czyniąc pracę bardziej efektywną dla sprawdzonych rozwiązań.

Grzegorz Cuber

Computer Controls Sp. z o. o.

tel. 33 485 94 90, www.ccontrols.pl



Rysunek 2. Wybranie bezpiecznego kompilatora w środowisku μ Vision

Więcej informacji:

Transfer Multisort Elektronik Sp. z o.o.
93-350 Łódź, ul. Ustronna 41
tel. +48 42 645 55 55, dso@tme.pl, www.tme.eu



Praca z płytką drukowaną

– w co wyposażać stanowisko

Płytki drukowane PCB znajdziemy w każdym urządzeniu elektronicznym. Stanowią strukturę nośną dla elementów i zapewniają połączenia elektryczne między nimi. Połączenia są realizowane w postaci miedzianych ścieżek, do których są lutowane wyprowadzenia elementów. W zależności od rodzaju płytek, typu elementów i przeznaczenia, stosuje się różne narzędzia, preparaty i techniki montażu. W artykule wskażemy rozwiązania, które usprawnią pracę z płytkami PCB. Podpowiemy, jak przygotować niewielką pracownię na wszelkie czynności wykonywane przy różnych etapach produkcji, montażu i serwisu płytek drukowanych.

Płytki drukowane PCB mogą być wykonywane z różnych materiałów, choć najczęściej stosuje się w tym celu laminat z włókna szklanego, utwardzonego żywicą epoksydową. Ma on odpowiednie właściwości dielektryczne, ponadto jest wytrzymały, elastyczny, i może zostać przycięty do różnych kształtów i rozmiarów. Istnieją różne procesy wytwarzania płytek drukowanych na takim podłożu – od w pełni zautomatyzowanych linii produkcyjnych, z których dziennie zjeżdżają tysiące sztuk PCB, po małe pracownie, w których wytwarzane są pojedyncze egzemplarze. Obwody drukowane

często są projektowane i wykonywane także w warunkach domowych przez hobbystów i pasjonatów.

Nieodłącznym etapem przygotowania płytki jest jej montaż. Podobnie jak w przypadku wytwarzania PCB, tu także dostępne są różne procesy i technologie dostosowane do produkcji seryjnej, np. lutowanie rozpliwowe, oraz produkcji jednostkowej, jak lutowanie ręczne przy użyciu lutownicy kolbowej.

Zdarzają się również sytuacje, kiedy sprzęt elektroniczny ulega awarii. Wtedy na stanowisko warsztatowe trafia gotowa, zmontowana płytka, a czynności naprawcze zwykle polegają na wymontowaniu uszkodzonych elementów i zamontowaniu nowych sprawnych. Sprzęt i techniki naprawcze stosowane w tego rodzaju czynnościach mogą być zupełnie inne od tych stosowanych na etapie produkcji.

W tym artykule zajmiemy się bardzo obszernym tematem, jakim jest przygotowanie stanowiska i sprzętu oraz materiałów i preparatów niezbędnych do prac montażowo-naprawczych płytek PCB. Nie bierzemy pod uwagę profesjonalnych rozwiązań produkcyjnych, skupimy się na pracy z obwodami drukowanymi w niewielkim warsztacie i domowej pracowni.

Podstawy – odpowiednie miejsce

Przed rozpoczęciem jakiejkolwiek pracy należy wybrać odpowiednie miejsce. Najlepiej, jeśli będzie to duża przestrzeń, taka jak oddzielny pokój czy garaż lub piwnica. Jednak ostatnie miesiące pokazały,



Fotografia 1. Mata ESD

że często nie mamy zbyt wielkiego pola do manewru i takie pracownię chcemy organizować w domu.

Niezależnie od wybranego pomieszczenia warto jest zadbać o utrzymanie stałej temperatury. Jej zmiany mogą powodować powstawanie wilgoci, co bardzo niekorzystnie wpływa nie tylko na płytki PCB. Równie istotnym parametrem, bardzo często lekceważonym, jest wilgotność powietrza. Przy niskiej wilgotności bardzo łatwo powstają ładunki elektrostatyczne ESD, które przyczynią się do uszkodzania elementów półprzewodnikowych. W warunkach domowych należy też bezwzględnie zadbać o dużą cyrkulację powietrza w pomieszczeniu – pracowni. Najlepiej jeśli znajduje się tam okno z możliwością uchylecia i kratka wentylacyjna.

Zapobieganie gromadzeniu się ładunków elektrostatycznych wymaga przygotowania posadzki o właściwościach ESD. W domowych warunkach jest to trudne do realizacji. Dobrym zastępczym rozwiązaniem jest zastosowanie maty lub wykładziny ESD (**fotografia 1**), która dodatkowo zabezpiecza powierzchnię podłogową przed korozyjnym działaniem substancji chemicznych i pozwala na łatwe utrzymanie czystości.

Stanowisko pracy

Profesjonalny stół warsztatowy ma wiele zalet, jednak w domowych warunkach z powodzeniem sprawdzi się solidne biurko wyposażone w kilka niezbędnych elementów. Przede wszystkim należy zadbać o odpowiednie oświetlenie. Zalecanym rozwiązaniem jest zastosowanie co najmniej dwóch źródeł światła. Możemy użyć specjalnej lampy warsztatowej, która oświetla odpowiednio rozproszonym światłem, a dodatkowo, jeśli stanowisko ma półkę umieszczoną ponad obszarem roboczym, możemy zamontować pod nią listwę diod LED. Taka konfiguracja znacząco poprawi oświetlenie głównego blatu roboczego oraz obszaru dookoła niego.

Przy organizacji stanowiska pracy warto zadbać o przestrzeń do przechowywania narzędzi i materiałów. Dodatkowe szuflady, organizery na narzędzia i stojaki na płytki PCB (**fotografia 2**) usprawnią pracę i ułatwią utrzymanie porządku. Ze względów bezpieczeństwa wszystkie metalowe elementy takiego wyposażenia powinny zostać uziemione.

Praca z obwodami drukowanymi wymaga również odpowiedniego fotela. Na rynku dostępne są krzesła warsztatowe, których powierzchnie są odpowiednio wyprofilowane, dzięki czemu praca nie będzie dla nas



Fotografia 2. Stojak na płytki PCB



Fotografia 3. Stacja lutownicza typu WEL.WT1010HSET

aż tak męcząca i ułatwi utrzymanie właściwej pozycji. W ten sposób unikniemy w przyszłości problemów z kręgosłupem.

Co umieścić na stole warsztatowym

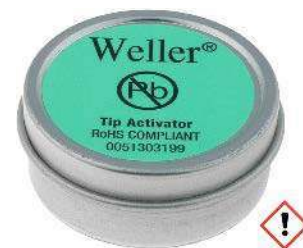
Podstawowym i najważniejszym wyposażeniem jest lutownica. Niezależnie od tego, czy nasza pracownia ma służyć działalności komercyjnej, czy tylko realizacji pasji, lutownica musi być solidnym, sprawdzonym i wielozadaniowym narzędziem (**fotografia 3**). Każda inwestycja w przyrząd tego typu zwraca się wielokrotnie, natomiast lutownica o niewłaściwych parametrach czy niedopracowanej konstrukcji tylko przysporzy kłopotów.

Dlatego wybieramy **stację lutowniczą** z regulacją i stabilizacją temperatury grota oraz z dużym zapasem mocy. Należy wiedzieć, że większa moc grzałki lutownicy nie oznacza tylko jej szybszego nagrzewania się. Dzięki temu możliwe będzie lutowanie elementów o dużych wyprowadzeniach czy połączonych z obszarami masy na płytce drukowanej, zamontowanych tak, by skutecznie rozpraszały ciepło.

Każda lutownica powinna być też wyposażona w **zestaw grotów** do różnych zastosowań. Do lutowania elementów SMD najbardziej przydatne będą groty precyzyjne – stożkowe lub ścięte jednostronnie. Niewielki rozmiar samej końcówki, pozwoli na precyzyjne rozgrzanie konkretnego padu lutowniczego. Przy pracy z układami scalonymi montowanymi powierzchniowo przydatny będzie również grot określany jako minifala. Pozwala on w łatwy sposób lutować gęsto rozmieszczone wyprowadzenia układów scalonych bez tworzenia zwarców. Oprócz tego warto wyposażać się w groty nieco większe, przeznaczone do lutowania i rozlutowywania elementów o znacznie większej powierzchni.

Często pomijaną kwestią jest konserwacja grotów. Poprzez stosowanie odpowiednich narzędzi i preparatów można znacznie przedłużyć żywotność tych elementów oraz usprawnić proces lutowania. Najbardziej popularnym sposobem konserwacji grotu jest używanie czyszcików wykonanych z metalowych wiórków lub specjalnych gąbek nasączonych wodą. Na stanowisku nie powinno zabraknąć także chemicznego preparatu do regeneracji grotów (**fotografia 4**). Dodatkowym godnym polecenia są maty do lutowania, które umieszczamy na blacie stołu warsztatowego. Zwykle są one wyposażone w niewielkie przegródki, znacznie ułatwiające utrzymanie porządku podczas lutowania i rozlutowywania obwodów, z którymi pracujemy.

Równie często jak do lutowania, lutownica będzie służyła do rozlutowywania i wylutowywania elementów. W tym procesie, dzięki któremu możemy usunąć dowolny komponent z płytki drukowanej, przydatny będzie odsysacz cyny (**fotografia 5**) oraz taśma rozlutowująca, popularnie nazywana



Fotografia 4. Regenerator grotów WEL.51303199

plecionką (fotografia 6). Dzięki nim będziemy mogli bez problemu usunąć nadmiar cyny z pola lutowniczego, co ułatwi demontaż elementu elektronicznego. Przy dużej liczbie takich zadań warto pomyśleć o stacji rozlutowującej (fotografia 7), która działa jak odsysacz z elektrycznie napędzanym podciśnieniem i grotem lutowniczym zintegrowanym w końcówce ssącej. Należy jednak pamiętać, że taka stacja wymaga regularnego czyszczenia.



Fotografia 5. Odsysacz lutowniczy typu ERS-VACX



Fotografia 6. Pleccionka miedziana do usuwania cyny

Obowiązkowym elementem każdego warsztatu elektronicznego jest również sprzęt kontrolno-pomiarowy. Multymetry, oscyloskopy czy analizatory logiczne pomogą zdiagnozować przyczynę niewłaściwego działania układu oraz umożliwiają zlokalizowanie usterki powstałej na płytce PCB. Warto wybrać solidne i sprawdzone modele (fotografia 8) z rozbudowanym systemem zabezpieczeń, aby służyły niezawodnie. Jednak nawet ogólne omówienie tej tematyki jest zagadnieniem na oddzielny artykuł.

Stacje lutownicze na gorące powietrze

W przypadku lutowania lub rozlutowywania niektórych elementów, zwłaszcza układów SMD z dużą liczbą wyprowadzeń, niezbędna jest stacja lutownicza na gorące powietrze, tzw. **hot air** (fotografia 9). Jednak korzystanie z lutownicy tego typu wymaga doświadczenia i warto najpierw poćwiczyć na płytkach przeznaczonych do utylizacji. Ponadto przydaje się do różnych innych czynności, takich jak obkurczanie rurek termokurczliwych czy suszenie płytek po umyciu.

Podczas używania stacji na gorące powietrze przydatna będzie taśma kaptonowa, która dzięki swojej odporności na wysokie temperatury pozwala uchronić niektóre elementy przed strumieniem gorącego powietrza. Tego rodzaju taśma jest szczególnie przydatna przy pracy z płytkami, na których znajdują się elementy wrażliwe

na wysoką temperaturę, np. złącza. Taśmą pokrywamy obszar wokół lutowanego elementu, dzięki czemu elementy umieszczone pod taśmą nie zostaną przypadkowo przesunięte lub odlutowane.

Jeśli mamy zamiar pracować z płytkami PCB, na których umieszczone są układy montowane powierzchniowo, możemy pomyśleć o zakupie profesjonalnej stacji do montażu BGA. Taka stacja, poza kierunkowym podgrzewaniem danego elementu od góry, podgrzewa również płytkę od spodu. Unikamy w ten sposób naprężeń powstałych w wyniku różnicy temperatur i eliminujemy efekt rozpraszania ciepła. Taką funkcję mogą pełnić także podgrzewacze, których używa się wraz ze stacją.

Przygotowanie własnej płytki

Jeśli nasz warsztat już został wyposażony w podstawowy sprzęt, możemy teraz zająć się samą płytką drukowaną. W zależności od tego, czy chcemy tylko polutować już gotową płytkę, czy też przygotować ją od zera, należy podjąć odpowiednie kroki.

W przypadku gdy samodzielnie wytwarzamy płytki, np. wstępne wersje prototypowe, potrzebujemy dodatkowego oprzyrządowania. Jeśli na płytce mają zostać umieszczone elementy przewlekane (montaż THT), musimy wykonać w niej odpowiednie otwory. Ich średnica jest najczęściej bardzo mała, dodatkowo są umieszczone dość gęsto, przez co potrzebne jest precyzyjne urządzenie umożliwiające wykonanie takich wierceń – jest nim miniwierarka (fotografia 10). Do pracy z miniwierarką niezbędny będzie również zestaw precyzyjnych wiertel.

Jeśli dysponujemy czystym laminatem i zamierzamy umieścić ścieżki i pola lutownicze na jego powierzchni, to musimy wykonać tzw. proces wytrawiania. Polega on na naniesieniu warstwy ochronnej na laminat zgodnie z mozaiką ścieżek i umieszczeniu takiej płytki w roztworze, który usunie nieosłoniętą miedź. Sposobów na naniesienie ścieżek jest wiele, od rysowania za pomocą specjalnego markera, przez drukowanie na papierze kredowym i przenoszenie wydruku metodą termotransferu, po metodę fotochemiczną, która nadaje się do wykonywania nawet skomplikowanych płytek.

Jeśli ścieżki są już naniesione, można przejść do samego wytrawiania. Płytkę umieszcza się w specjalnym roztworze B327 lub chlorku żelaza. Substancja ta spowoduje usunięcie miedzi w miejscach, gdzie nie ma warstwy osłaniającej. Gdy obwód jest już gotowy, należy go odpowiednio zabezpieczyć. W tym celu płytkę poddaje się procesowi cynowania, czyli naniesieniu warstwy cyny na warstwę miedzi. Można to zrobić ręcznie bądź skorzystać z gotowych preparatów chemicznych. Na koniec wykonaną płytkę warto pokryć tzw. soldermaseką. Jest to chemiczna ochrona, przypominająca lakier, elektrycznie obojętna, która poza ochroną obwodu przed utlenianiem zabezpieczy go przed przypadkowym zwarcie.

Lutowanie płytek PCB

Dla ułatwienia pracy płytkę PCB należy umieścić w specjalnym uchwycie (fotografia 11). Może on być wykonany w różnorodny sposób, czasami jest to zestaw tzw. trzeciej ręki wyposażony dodatkowo w lupe. Gdy obwód drukowany jest już stabilnie ustawiony, możemy rozpocząć proces lutowania. Do tego celu jest niezbędne spoiwo lutownicze. Na rynku dostępne są dwa rodzaje spoiwa – ołowiowe i bezołowiowe. Cyny ołowiowe, ze względu na zawartość szkodliwego pierwiastka, wykorzystywane są coraz rzadziej – głównie w zastosowaniach medycznych lub militarnych, gdzie istotna jest niezawodność. Spoiwem bezołowiowym lutuje się nieco trudniej, ale za to nie jest tak szkodliwe dla zdrowia. Do lutowania przydatne są również wszelkiego rodzaju topniki. Ich stosowanie znacznie ułatwia pracę, szczególnie w przypadku elementów SMD.

Podczas procesu lutowania, niezależnie od tego, jakiego spoiwa używamy, powstają szkodliwe dla naszego organizmu substancje lotne. Dlatego warto wyposażać się w tzw. odciąg oparów. Dzięki niemu nie trafią one do naszego układu oddechowego, ponieważ



Fotografia 7. Stacja rozlutowująca typu SP-1010DR



Fotografia 8. Multimetr stołowy typu AX-8450A

zostaną odesłane. Tego typu urządzenia dostępne są w trzech formach: ustawianych na biurku, montowanych na ruchomym ramieniu oraz jako skrzynki z wyprowadzoną rurą, którą możemy zamontować w dowolnym miejscu (fotografia 12).

Gdy płytkę PCB jest już zmontowana, najpewniej pokryta jest resztkami topnika czy innych osadów, których należy się pozbyć. Do tego celu możemy użyć najbardziej rozpowszechnionego alkoholu izopropylowego tzw. IPA lub środków czyszczących w postaci aerozolu. Często tego typu produkty mają dyszę zakończoną pędzelkiem ułatwiającym czyszczenie obwodów drukowanych. Płytki możemy również czyścić w myjce ultradźwiękowej, jednak do tego celu potrzebny jest odpowiedni środek czyszczący. Warto też zwrócić uwagę na ofertę topników typu no-clean, które dzięki specyficznemu składowi nie wymagają czyszczenia.

Płytkę PCB po wyczyszczeniu i wysuszeniu możemy pokryć warstwą specjalnego lakieru, który zabezpieczy obwód drukowany przed niepożądanym działaniem warunków zewnętrznych. Tak przygotowaną płytkę możemy uznać za skończoną i umieścić ją w obudowie urządzenia.

Dodatkowe narzędzia

Przy pracy z płytkami PCB z pewnością przydadzą nam się pęsety precyzyjne. Możemy ich użyć zarówno przy procesie lutowania, jak i rozlutowywania danego obwodu. Najbardziej powszechne są pęsety o ostrym zakończeniu, ale warto mieć na wyposażeniu także wersje z zagiętą oraz zakończoną płaskimi łapkami, za pomocą których będziemy mogli dopasować elementy SMD do pól lutowniczych.

Większość płytek PCB umieszczana jest w różnego rodzaju urządzeniach, które musimy otworzyć. Do tego celu przydatne będą narzędzia takie jak wkrętaki o różnych końcówkach, szczypce czy też obcinaczki. Takie narzędzia zazwyczaj są już na wyposażeniu warsztatu elektronicznego. Do pracy z PCB przydatne będą również różnego rodzaju środki chemiczne, wśród których wymienić można wspomniany już alkohol izopropylowy, topniki i substancje

do regeneracji grotów czy lakier PCB. Oprócz nich przydatna będzie również: kalafonia, która podobnie jak topnik ułatwia proces lutowania, spray/płyn typu „kontakt” służący do regeneracji np. potencjometrów czy kleje termoprzewodzące umożliwiające przytwierdzenie radiatora odprowadzającego ciepło.

Podsumowanie

Tematyka wyposażenia warsztatu oraz pracy z płytkami to bardzo szerokie zagadnienie. Dokonując wyboru sprzętu, narzędzi i materiałów, warto kierować tym, że im lepszy sprzęt wybierzemy, tym sprawniej wykonamy wszelkie czynności, bez niepotrzebnych komplikacji oraz w sposób bezpieczny dla nas samych.

TME
www.tme.eu



Fotografia 10. Wiertarka miniaturowa typu PG-142MD



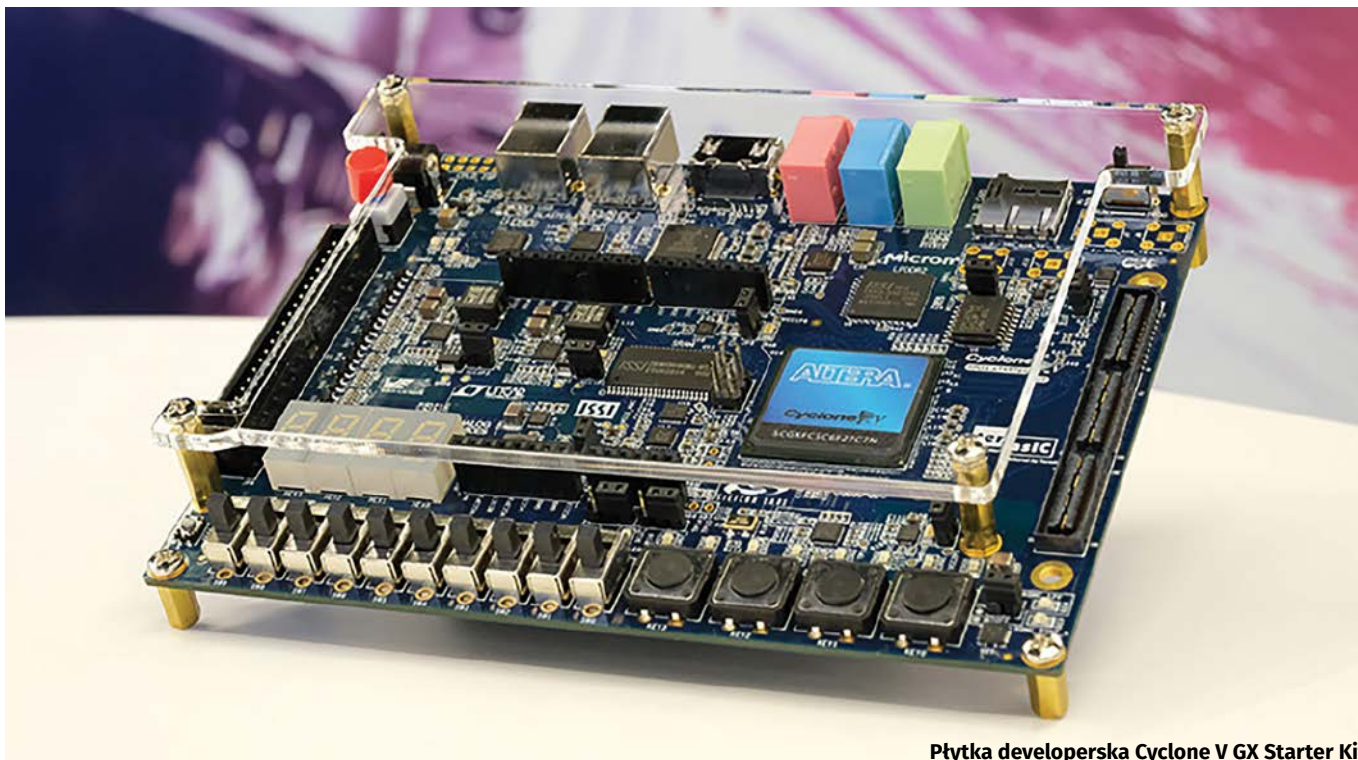
Fotografia 11. Uchwyt do płytek PCB typu WEL.ESF120ESD



Fotografia 9. Stacja na gorące powietrze typu ST-862D



Fotografia 12. Pochłaniacz oparów typu WEL.ZEROSMOGEL-KIT



Płytki developerska Cyclone V GX Starter Kit

Sprzętowy hackathon on-line

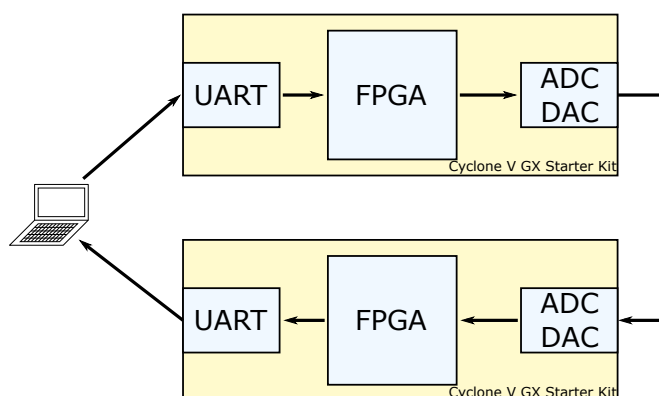


Nokia zajmuje się tworzeniem infrastruktury dla telefonii komórkowej. Nasi inżynierowie pracują nad tymi zagadnieniami już od ponad 25 lat, a w tym roku, w Centrum R&D Nokii w Krakowie będziemy świętować dziesięć lat rozwijania technologii LTE. Jednak nie spoczywamy na laurach. Obecnie intensywnie zajmujemy się technologią 5G, a już wkrótce także 6G. Szczególnie prężnie rozwijającym się działem jest programowanie układów FPGA. Są one wyjątkowo przydatne przy przetwarzaniu dużych ilości danych, z jakimi spotykamy się w nowoczesnej telekomunikacji.

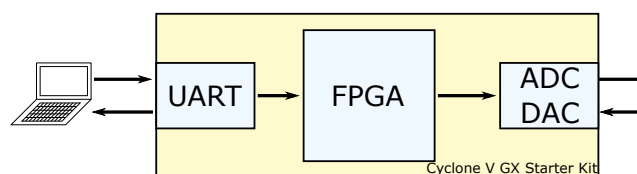
Aby dzielić się naszą wiedzą, w zeszłym roku zorganizowaliśmy konferencję oraz hackathon poświęcony tej tematyce. W artykule zaprezentuję kilka wspomnień tego wydarzenia. Jeżeli po lekturze, szanowny Czytelniku, zaczniesz żałować, że przegapiłeś to wydarzenie, to nic straconego. Już rozpoczęliśmy pracę nad drugą edycją, planowaną na listopad bieżącego roku. Wróćmy jednak do przeszłości...

Zadanie

Plan zrealizowania hackathonu FPGA pojawił się jeszcze w spokojnych czasach, gdy nikt nie słyszał o koronawirusie (czyli w połowie 2019 roku). Pierwotnie miał on odbyć się więc stacjonarnie, z początkiem kwietnia 2020 roku w Krakowie. Zadanie było planowane dla uczestników, którzy spotkają się na 24 godziny, a każda z drużyn miała otrzymać laptop z zainstalowanym oprogramowaniem oraz płytkę developerską Cyclone V GX Starter Kit[1], która została



Rysunek 1. Schemat zestawu do testowania rozwiązań



Rysunek 2. Stanowisko dla uczestnika

pokazana na fotografii tytułowej. Jest ona wyposażona w kodek audio SSM2603 oraz liniowe wejście i wyjście audio. Na potrzeby zadania zostały one połączone razem, co schematycznie pokazuje rysunek 1.

Celem było zaimplementowanie modemu wykorzystującego tak stworzone łącze do przesyłania danych z jak największą

przepustowością. Na obydwu płytkach znajdował się ten sam bitstream. Dane testowe są przesyłane z komputera poprzez port szeregowy. Układ FPGA musi zrealizować odpowiednią modulację i przekazać ją do kodeka audio. Następnie, na drugiej płytce, odebrane dane muszą zostać z powrotem odkodowane i przesłane do komputera. Na potrzeby testowania własnych pomysłów drużyny korzystały tylko z pojedynczej płytki ze zwartym wejściem i wyjściem (**rysunek 2**), natomiast zestaw z dwoma płytkami został użyty do oceniania rozwiązań.

Aby ułatwić rozpoczęcie pracy, uczestnicy otrzymali projekt startowy. Można go znaleźć w repozytorium [2]. Jak pokazano na **rysunku 3**, składa się on z trzech głównych części:

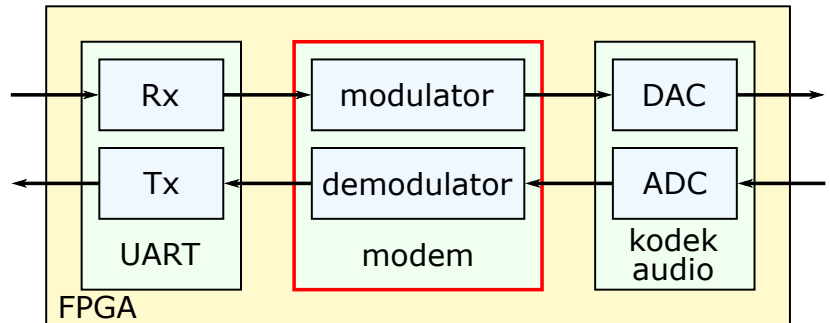
- obsługi portu szeregowego;
- modemu;
- interfejsu dla kodeka audio.

W dostarczonej wersji dostępna jest prosta implementacja modemu. Dzięki temu uczestnicy mogli zapoznać się z formatem danych wejściowych i wyjściowych. Mieli także dostępną bazę, na podstawie której mogli tworzyć własne rozwiązania.

Uczestnicy mieli do dyspozycji także dość spory zestaw narzędzi programistycznych. Głównym z nich było środowisko Quartus, umożliwiające generowanie wsadu oraz symulator Questa. Poza nimi można było też używać pakietu MATLAB&Simulink, Riviera-PRO oraz narzędzia do weryfikacji formalnej OneSpin.

Udostępnianie sprzętu

W połowie marca okazało się, że dwutygodniowy lock-down jednak zostanie z nami dużo dłużej. Kwietniowy termin był więc nie



Rysunek 3. Projekt startowy, uczestnicy modyfikują blok zaznaczony czerwonymi ramkami

do utrzymania. Z nadzieją, że sytuacja zdąży wrócić do normalności, nowy termin został ustalony na listopad. Szybko okazało się, że jeżeli chcemy, aby wydarzenie mogło się odbyć, to musimy przejść na formę on-line. Musieliśmy więc trochę zmodyfikować infrastrukturę. Zbudowane przez nas rozwiązanie zostało pokazane na **rysunku 4**. W naszym laboratorium w Krakowie uruchomiliśmy serwer, do którego podłączyliśmy poprzez USB programatory JTAG oraz porty szeregowo wszystkich dwudziestu płytek. Tak jak planowaliśmy wcześniej, zestawy dla drużyn miały podłączoną pętlę zwrotną. Natomiast wydzielone zestawy do oceniania składały się z dwóch połączonych razem płytek. Jedynym zadaniem serwera było zapewnienie dostępu do płytek za pomocą protokołu USBoverIP.

Dla każdej z drużyn przygotowaliśmy maszynę wirtualną z zainstalowanym oprogramowaniem. Dzięki udostępnieniu portów USB przez Internet maszyny wirtualne miały bezpośredni dostęp do sprzętu znajdującego się w laboratorium. Wszystkie narzędzia programistyczne łączyły się ze sprzętem w taki sposób, jakby był on fizycznie dołączony do gniazda USB komputera.

REKLAMA

ZAINTERESOWAŁ CIĘ POWYŻSZY ARTYKUŁ?

Dowiedz się więcej o aktualnych ofertach pracy w Nokii Kraków!

APLIKUJ



nokiakrakow.pl/oferty-pracy

CHCIAŁBYŚ POZNAĆ NAS LEPIEJ?

Dołącz do najbliższego Hackathonu i konferencji FPGA.

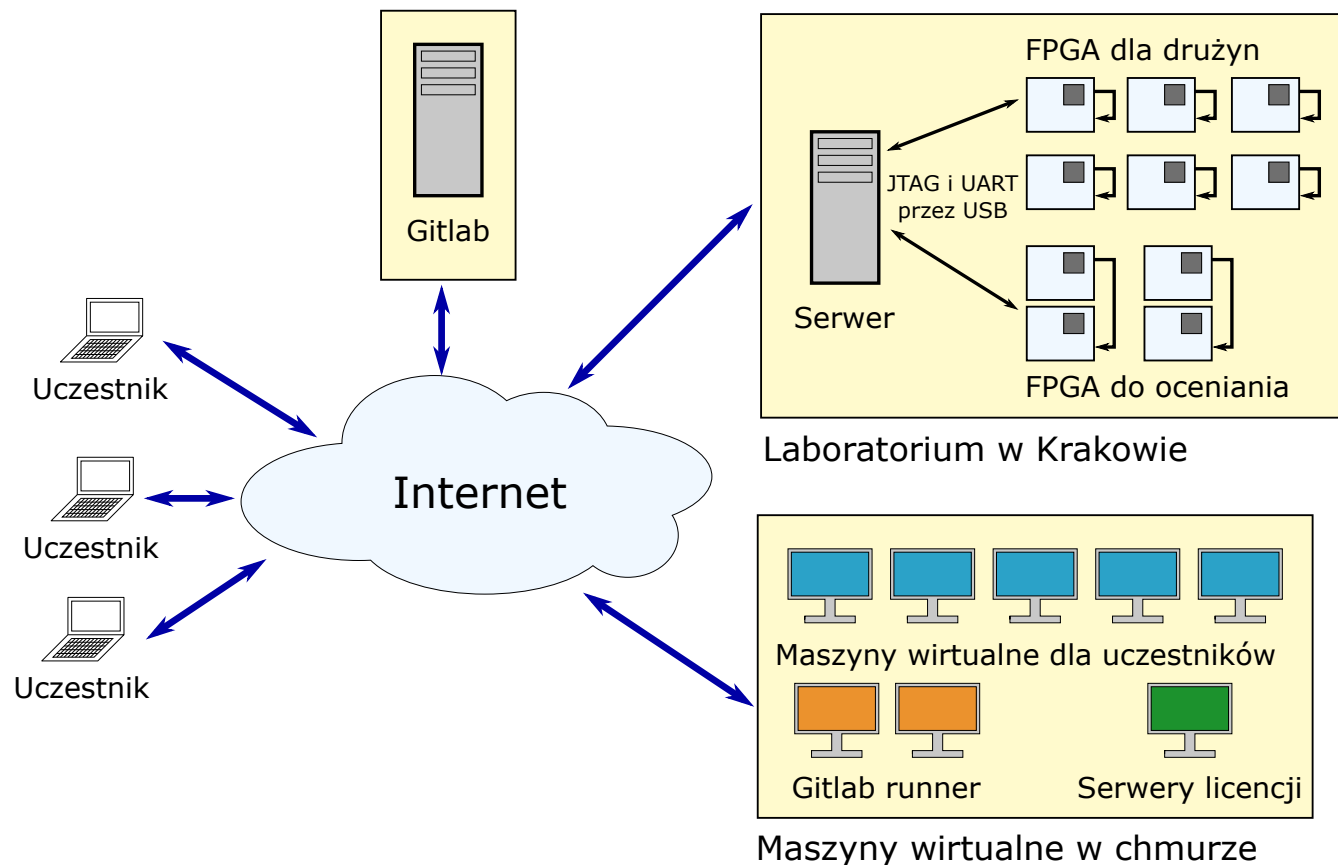
DOŁĄCZ



fpgahackathon.com

Nie tylko z zakresu FPGA!

NOKIA



Rysunek 4. Infrastruktura sieciowa hackathonu

Dysponując możliwością wirtualnego przepinania urządzeń pomiędzy maszynami, postanowiliśmy zwiększyć liczbę drużyn, które wzięły udział w zawodach, poprzez współdzielenie sprzętu: drużyny parzyste miały przydzielony sprzęt w godzinach parzystych, a nieparzyste w nieparzystych. Za przełączanie odpowiadał skrypt uruchomiony na serwerze.

Ocenianie rozwiązań

Do realizacji oceny rozwiązań użyliśmy narzędzi CI (*continuous integration* – ciągła integracja) udostępnianych przez platformę Gitlab. Każda z drużyn miała przypisane repozytorium, w którym umieszczała kolejne wersje swoich rozwiązań. Każdy commit powodował zakolejkowanie testu. Były one wykonywane po kolei, na jednym z dwóch zestawów. Obejmował on:

- skopiowanie rozwiązania na maszynę wirtualną,
- zaprogramowanie obydwóch układów FPGA tym samym projektem,
- uruchomienie skryptu mierzącego uzyskaną przepustowość,
- zwrócenie uzyskanego wyniku.

Wyniki były przekazywane do bazy danych i w czasie rzeczywistym prezentowane na stronie internetowej. Dzięki temu każda z drużyn mogła śledzić swoje postępy na tle pozostałych uczestników.

Przebieg konkursu

Zgodnie z początkowymi założeniami wydarzenie trwało 24 godziny. Publikacja zadania nastąpiła w sobotę o godzinie 12.00. W zawodach wzięło udział 28 drużyn. Dzięki zmianie formuły na wirtualną pojawił się zawodnicy z całego świata.

W czasie zawodów komunikacja odbywała się poprzez platformę Discord. Poza wspólnym kanałem każda z drużyn miała także swój własny pokój. W przypadku niejasności lub pytań technicznych zawodnicy mogli zaprosić do niego wolontariuszy z Nokii albo mentorów reprezentujących producentów zastosowanych narzędzi.

Po zaciętej rywalizacji, w niedzielę po południu, poznaliśmy wyniki. Zwyciężyła drużyna z Wojskowej Akademii Technicznej z Warszawy. Natomiast kolejne miejsca na podium zajęły ekipy z St. Petersburga oraz Kalifornii.

Co dalej?

Nasz krakowski zespół, bogatszy o zesłoroczne doświadczenia, rozpoczął już przygotowania drugiej edycji. Wstępnie możemy zdradzić, że jest ona planowana na listopad bieżącego roku i tak jak poprzednio, odbędzie się przez internet. Więcej informacji już wkrótce pojawi się na naszej stronie: www.fpgahackathon.com.

Zapraszamy wszystkich chętnych do sprawdzenia swoich sił w zawodach. Natomiast jeżeli chciałbyś już teraz dołączyć do naszego nokiowego zespołu, zachęcamy do sprawdzenia aktualnych ofert pracy: <https://nokiakrakow.pl/oferty-pracy/>.

NOKIA

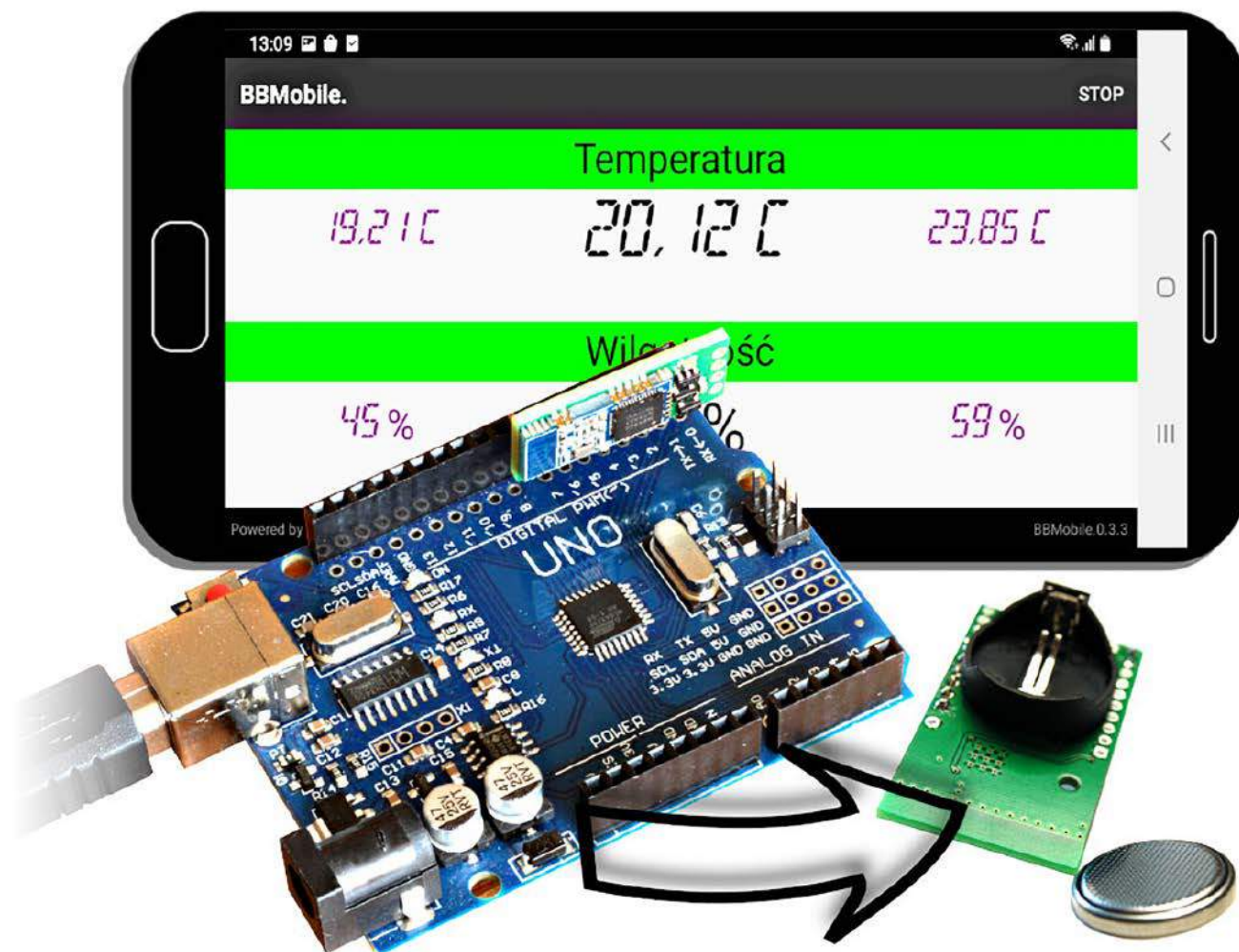
Rafał Kozik
FPGA design engineer
Nokia Kraków
rafkozik@gmail.com

Odnosiniki

- [1] <https://bit.ly/3yyBPY2>
[2] <https://bit.ly/3vgGKBS>

- [3] <https://bit.ly/2T5bXJW>
[4] <https://bit.ly/3ufJcr0>

<http://forum.ep.com.pl>



Bez programowania Androida

Projekt z użyciem BLE i aplikacji mobilnej (1)

W artykule zaprezentowano proces powstawania urządzenia elektronicznego, działającego z interfejsem mobilnym na ekranie smartfona. Co ciekawe, tworzenie projektu nie wymagało tworzenia aplikacji dla systemu Android – graficzny interfejs użytkownika jest budowany i obsługiwany przez samo urządzenie.

Dlaczego smartfon?

Rozwój techniki mikroprocesorowej, technologii radiowej i internetowej doprowadził nas do sytuacji, w której małe, trzymane w dłoni urządzenie z ekranem może stać się niemal całym światem. Widzimy to na ulicach i przystankach, w autobusach i tramwajach, parkach, placach zabaw, szkołach i biurach. W smartfonach połączonych w globalną sieć jest rozrywka, nauka, kultura, kontakty społeczne, wiedza, współzawodnictwo, biznes, spotkania towarzyskie, dyskusje polityczne itd. A bezkresne morze aplikacji dryfujących na maszynach

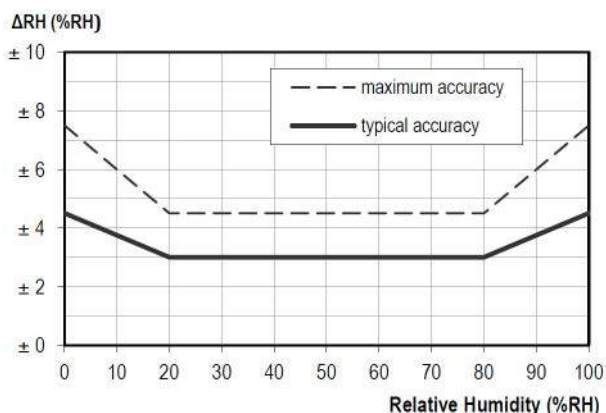
serwerowych tylko czeka na pobranie, zainstalowanie i wypróbowanie. W opisanym projekcie zastosujemy ten cudowny produkt ewolucji...

Pomysł i założenia

Zbudujemy rejestrator temperatury i wilgotności, ale nie taki, ot... zwyczajny. Nasze urządzenie będzie niezwykle, ponieważ wyposażymy je w wiele unikalnych cech zmiksowanych w jeden wielofunkcyjny produkt. Rejestrator będzie:

- bezprzewodowy, co oznacza, że może pracować w takim miejscu, które jest poza zasięgiem wzroku. Standardowe termometry i higrometry muszą być widoczne, w przeciwnym razie oczywiście nie odczytamy wskazania, czy to z wyświetlacza numerycznego, czy słupka cieczy w kapilarze;
- zawierał graficzny interfejs na ekranie smartfona, na którym czytelnie wyświetlane zostaną wszystkie informacje;
- potrzebował do pracy bardzo małej ilości energii, tak aby możliwe było zasilenie go nawet z baterii pastylkowej CR2032;

Parameter	Condition	Value	Units
Resolution ¹	12 bit	0.04	%RH
	8 bit	0.7	%RH
Accuracy tolerance ²	typ	±3.0	%RH
	max	see Figure 2	%RH
Repeatability		±0.1	%RH
Hysteresis		±1	%RH
Nonlinearity		<0.1	%RH
Response time ³	τ 63%	8	s
Operating Range	extended ⁴	0 to 100	%RH
Long Term Drift ⁵	Typ.	< 0.25	%RH/yr



Rysunek 1. Fragment dokumentacji układu SHT20 pokazujący jego najważniejsze parametry

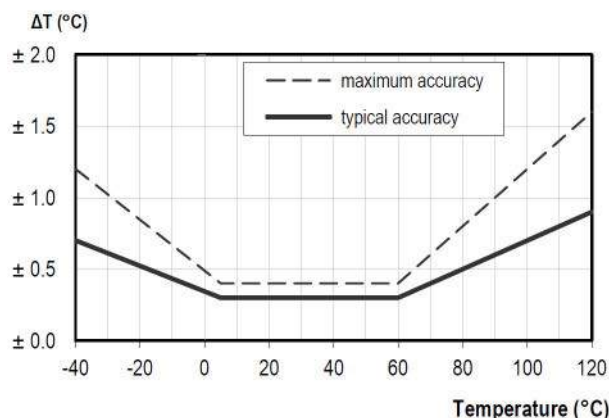
- wyposażony w interfejs Bluetooth Low Energy służący do zestawiania bezprzewodowego połączenia ze smartfonem lub tabletem;
- mierzył aktualną temperaturę i wilgotność z dużą dokładnością;
- wyświetlał dobową średnią, maksymalną oraz minimalną temperaturę i wilgotność;
- podawał średnią, maksymalną i minimalną temperaturę oraz wilgotność z ostatnich siedmiu dni;
- wyświetlał średnią, maksymalną oraz minimalną temperaturę i wilgotność zarejestrowaną od chwili włączenia zasilania;
- podawał liczbę dni, które upłynęły od startu procesu pomiarowego;
- mierzył i wyświetlał napięcie baterii/akumulatora, aby możliwe było określenie stanu źródła energii.

Najpierw narzędzia

Skoro mamy już pomysł i ogólne założenia, możemy przystąpić do określenia potrzebnych podzespołów i narzędzi. Ponieważ nasze urządzenie opiera się na energooszczędnej technologii **Bluetooth Low Energy (BLE)**, konieczne jest zastosowanie kompatybilnego chipu. Wybierzmy jeden z nordyckiej rodziny **NRF51**. Pomiar temperatury i wilgotności zrealizujemy sensorem **SHT20**. Będzie odpowiedni, ponieważ dysponuje interfejsem I²C oraz dużą dokładnością pomiarów przy zachowaniu reżimu energooszczędności. W trybie uśpienia pobiera typowo 0,15 μ A, a podczas pracy 300 μ A. Pomiar wilgotności względnej dostępny jest z typową dokładnością $\pm 3\%$ RH w szerokim zakresie temperatur od 20 do 80°C. W przedziale od 0 do 100°C błąd pomiaru nie przekracza 5% RH. Szczegóły zostały pokazane na **rysunku 1**. Warto wspomnieć, że sensor ten pracuje w pełnym zakresie wilgotności względnej, od 0% do 100% RH. Typowa dokładność pomiaru temperatury wynosi $\pm 0,3^\circ\text{C}$ w przedziale od 5 do 60°C. W pełnym przedziale od -40 do 120°C typowy błąd pomiaru jest nie większy niż $\pm 0,75^\circ\text{C}$. Dokładne dane zostały pokazane na **rysunku 2**.

Przystępując do realizacji, rozwijemy tajemnicę – jak to możliwe, że nie pisząc ani jednej linii kodu aplikacji mobilnej, wyświetlimy dane pomiarowe na ekranie smartfona lub tableta.

Parameter	Condition	Value	Units
Resolution ¹	14 bit	0.01	$^\circ\text{C}$
	12 bit	0.04	$^\circ\text{C}$
Accuracy tolerance ²	typ	±0.3	$^\circ\text{C}$
	max	see Figure 3	$^\circ\text{C}$
Repeatability		±0.1	$^\circ\text{C}$
Operating Range	extended ⁴	-40 to 125	$^\circ\text{C}$
Response Time ⁷	τ 63%	5 to 30	s
Long Term Drift ⁸	Typ.	< 0.02	$^\circ\text{C/yr}$



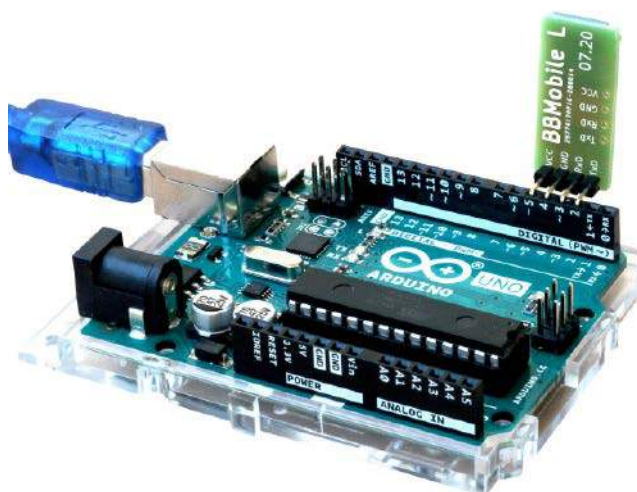
Rysunek 2. Fragment dokumentacji układu SHT20 pokazujący jego parametry w pełnym zakresie temperatury

Bez pisania aplikacji na Androida

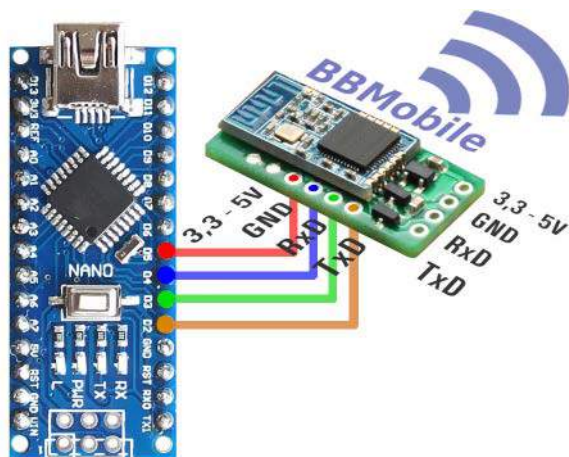
Opracowując elektroniczny produkt współpracujący bezprzewodowo z urządzeniem mobilnym (smartfon, tablet), standardowo mamy do wykonania dwa główne elementy:

1. samo urządzenie, najczęściej wyposażone w radiowy interfejs komunikacyjny: Bluetooth lub Wi-Fi;
2. aplikację instalowaną w systemie operacyjnym urządzenia mobilnego.

Takie sztapkowe podejście jest dość czasochłonne i generuje wysokie koszty realizacji projektu, ponieważ poza wykonaniem samego produktu, każdorazowo konieczne jest zaprojektowanie i zaimplementowanie odpowiedniej aplikacji mobilnej. Spróbujmy zoptymalizować ten proces, pomijając drugi jego etap. Zamiast projektować i tworzyć całą aplikację mobilną od zera, użyjemy uniwersalnej – BBMobile dla systemu Android, która pozwoli zbudować i obsłużyć graficzny interfejs użytkownika za pomocą tekstowych komend. Pozostałe do wykonania prace podzielimy na dwa etapy:



Rysunek 3. Sposób dołączenia modułu BBMobile do płytki Arduino UNO



Rysunek 4. Sposób dołączenia modułu BBMobile do płytki Arduino NANO

Parametry modułu BBMobile L

- ◆ Zasilanie: 3,3V do 5,0V,
- ◆ Średni pobór prądu: 1mA,
- ◆ Komunikacja przewodowa z mikrokontrolerem – port szeregowy – UART:
 - tolerancja 5V,
 - domyślnie 9600 bps, możliwe 4800 bps,
 - 8N1 – 8 bitów danych, bez bitu parzystości, 1 bit stopu,
- ◆ Komunikacja bezprzewodowa z urządzeniem mobilnym – BLE (Bluetooth Low Energy) – Bluetooth od wersji 4.0.
- ◆ Tworzenie aplikacji mobilnych poprzez przesłanie UARTem tekstowego kodu JSON,
- ◆ Komunikacja z aplikacją poprzez krótkie komunikaty tekstowe,
- ◆ Bezpłatna aplikacja BBMobile dla Androida od wersji 4.4.2
- ◆ Wymiary: 27 x 12 x 4 mm,

Rysunek 5. Fragment dokumentacji modułu BBMobile

- zaprojektowanie interfejsu graficznego oraz komunikacji z nim za pomocą płytki UNO (rysunek 3) bądź NANO (rysunek 4), kompatybilnej z Arduino IDE oraz modułu BBMobile (jego parametry zostały pokazane na rysunku 5);
- przeniesienie zaprojektowanych rozwiązań na ostateczną platformę hardware'ową.

Do projektowania interfejsu mobilnego wykonujemy następujące czynności:

1. Uruchamiamy Arduino IDE i otwieramy szkic *BBMobile_Ardu_Retransmitter.ino*. Szkic można pobrać np. z witryny www.bb-magic.net;
2. Programujemy płytkę UNO lub NANO. Uruchomiony program kopiuje dane pomiędzy dwoma portami szeregowymi. Wszystko,

Listing 1. Kod programu testowego *BBMobile_Ardu_Retransmitter.ino*

```
//-----
// MAIN CODE HERE, TO RUN REPEATEDLY
//-----
VOID LOOP()
{
  i++;

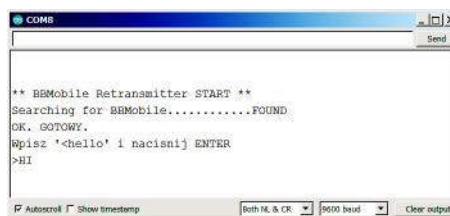
  // READ FROM MONITOR PORT AND SEND TO BBMOBILE PORT:
  IF (MON_SERIAL.AVAILABLE()) {
    INT INBYTE = MON_SERIAL.READ();
    BBM_SERIAL.WRITE(INBYTE);
  }

  // READ FROM BBMOBILE PORT AND SEND TO MONITOR PORT:
  IF (BBM_SERIAL.AVAILABLE()) {
    INT INBYTE = BBM_SERIAL.READ();
    MON_SERIAL.WRITE(INBYTE);
  }

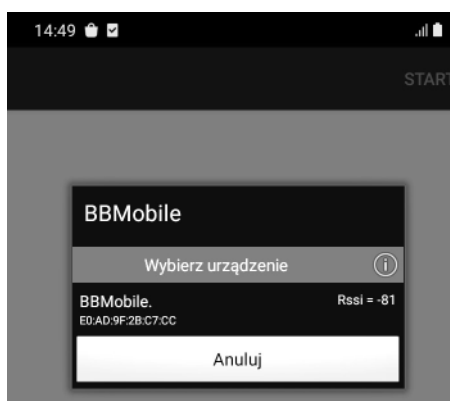
  // LED BLINKING
  IF (i == 4000) DIGITALWRITE(LED_BUILTIN, LOW); //-TURN THE LED OFF
  IF (i > 20000)
  {
    i = 0;
    DIGITALWRITE(LED_BUILTIN, HIGH); //-TURN THE LED ON
  }
}
```



Rysunek 6. Konfiguracja monitora portu szeregowego



Rysunek 7. Potwierdzenie komunikacji z modulem

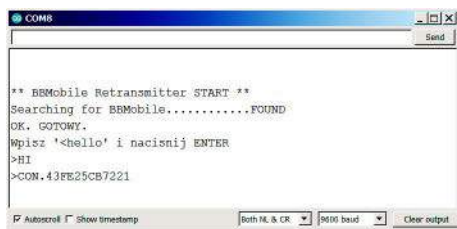


Rysunek 8. Uruchomienie aplikacji BBMobile

co pojawi się na sprzętowym porcie *BBM_SERIAL*, wysyłane jest na programowy port *MON_SERIAL* i odwrotnie. Główna część tego programu została zaprezentowana na [listingu 1](#);

3. W Arduino IDE uruchamiamy Serial Monitor (wybierając Tools → Serial Monitor lub naciskając Ctrl+Shift+M). Konieczne jest wybranie ustawień w prawym dolnym rogu: *Both NL&CR* oraz *9600 baud*, tak jak zostało to pokazane na [rysunku 6](#). W oknie zobaczymy powitalny komunikat oraz kolejno pojawiające się kropki informujące o oczekiwaniu na komunikację z modulem BBMobile;
4. Teraz podłączamy moduł tak, jak pokazano na rysunku 3 lub rysunku 4. Ponieważ BBMobile zasilany jest z pinów kontrolera (pin 4: GND, pin 5: VCC), bezpieczniej jest najpierw załadować program, a dopiero potem podłączyć moduł. Po jego wykryciu w oknie natychmiast pojawi się komunikat taki, jak na [rysunku 7](#);
5. Wpisanie do górnego okna Serial Monitora komendy *<hello* i naciśnięcie ENTER przetestuje komunikację z modulem. W odpowiedzi dostaniemy przyjazne *>HI* (rysunek 7);
6. Na smartfonie (z Bluetooth w wersji minimum 4.0) instalujemy i uruchamiamy aplikację BBMobile. System Android wymaga włączenia usługi lokalizacji (jeśli nie jest włączona), aby możliwe było skanowanie w poszukiwaniu urządzeń Bluetooth Low Energy;
7. Po uruchomieniu aplikacji dotykamy przycisku START w prawym górnym rogu ekranu i wybieramy z listy urządzenie o nazwie BBMobile, tak jak pokazuje to [rysunek 8](#);
8. Po chwili połączenie BLE zostanie zestawione, a w głównym oknie Serial Monitora pojawi się nowy komunikat, jak na [rysunku 9](#). Komunikat *>CON* informuje o nawiązaniu połączenia i po kropce zawiera sześciobajtowy adres urządzenia, z którym zestawiono komunikację.

Od tej chwili wszystkie komendy wpisane do górnego paska w oknie Serial Monitora zostaną przesłane wprost do aplikacji BBMobile, która będzie je wykonywała. W ten sposób możemy budować i kontrolować



Rysunek 9. Informacja o zestawieniu połączenia

interfejs na ekranie smartfona za pomocą prostych, tekstowych poleceń. Przystępujemy zatem do prac projektowych.

Pierwszy interfejs mobilny

Jeśli połączenie z aplikacją mamy zestawione, to główne okno Serial Monitora wygląda jak to z rysunku 9. Pora teraz na zbudowanie pierwszego interfejsu mobilnego za pomocą kilku prostych, tekstowych poleceń. Skopiujmy kod JSON (*JavaScript Object Notation*) widoczny na **listingu 2** i wklejmy go do górnego paska Serial Monitora, naciskając prawy przycisk myszy i wybierając Paste lub Wklej. Naciskamy ENTER i jeśli wklejony kod nie zawierał błędów, to w głównym oknie Serial Monitora pojawi się komunikat \$ok potwierdzający przyjęcie danych. Ekran smartfona powinien wyglądać jak ten z **rysunku 10**. Przesłany kod JSON jest definicją najprostszego chyba interfejsu zawierającego tylko dwa elementy. Są to nieedytowalne pola tekstowe TextView, które wyświetlają napisy. Nie będziemy tutaj dokładnie

analizować składni kodu JSON. Wiele informacji na ten temat można znaleźć pod adresem <http://bbmagic.net/co-to-jest-json-i-jak-go-uzyc/>.

Skoro wiemy już, że komunikacja z aplikacją BBMobile działa wysłanie, to możemy dodać do tego prostego interfejsu kolejne elementy. Nieco bardziej rozbudowany kod zawiera **listing 3**. Znajdują się tu już cztery kontrolki TextView ułożone wertykalnie (jedna pod drugą). Kontrolki zdefiniowane w liniach 3 i 5 (wyświetlające wartości temperatury i wilgotności) otrzymały swoje nazwy, odpowiednio: „t” i „h”. Nadanie kontrolkom unikalnych nazw jest konieczne, aby możliwe było późniejsze zmienianie wyświetlanej przez nie treści. Zwróćmy uwagę, że w czwartej linii kodu polskie znaki w słowie „Wilgotność” zostały zakodowane za pomocą podanych wprost wartości unicode. Takie rozwiązanie spowoduje poprawne ich wyświetlanie niezależnie od tego, jakiego edytora tekstu będziemy używać do przygotowania kodu JSON.

Wklejmy zatem gotowy kod do górnego paska Serial Monitora. Aplikacja mobilna wygląda teraz tak, jak na **rysunku 11**. Wpiszmy kolejne polecenie: `$set,t:te="19,8 C"`, zatwierdzając klawiszem ENTER i jeszcze kolejne `$set,h:te="48 %"` (i oczywiście ENTER na końcu). Łatwo zauważyć, że komendy te zmieniają wyświetlane wartości temperatury i wilgotności. Można je połączyć w jedno polecenie, które będzie wyglądało tak: `$set,t:te="19,8 C",h:te="48 %"`.

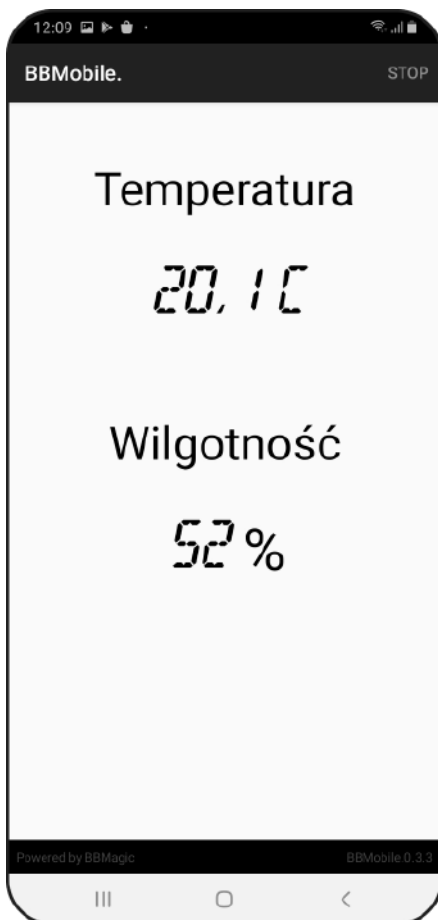
Ten interfejs mógłby już sprostac obsłudze bezprzewodowego termometru i higrometru. No, może wskazane byłoby jeszcze uatrakcyjnienie go jakąś grafiką. W linii pierwszej **listingu 4** polecenie `"img": "3"` wybiera grafikę o numerze porządkowym 3 jako obrazek

Listing 2. Kod JSON opisujący pierwszy interfejs

```
1 {"ty": "lout", "or": "V", "cs": [
2 {"ty": "TextView", "te": "\nRH T\nRejestrator", "tc": "255,0,255", "ts": "50", "tf": "03"},
3 {"ty": "TextView", "te": "\nTo jest projekt bez pisania aplikacji mobilnej", "tc": "80,80,80", "ts": "30"},
4 {"ty": "TextView", "w": "5"}]}
```



Rysunek 10. Wygląd pierwszej aplikacji mobilnej



Rysunek 11. Wygląd aplikacji mobilnej wyświetlającej parametry



Rysunek 12. Wygląd aplikacji mobilnej ze zmienionym tłem i kolorem czcionki

tła. Konieczna okazała się też zmiana koloru czcionki wyświetlanego tekstu, z poprzednio użytego czarnego (składowe R, G, B = 0, 0, 0) na nieco jaśniejszy (składowe R, G, B = 250, 250, 250). Efekt zmian został pokazany na **rysunku 12**.

Horyzontalne kontrolki dla wartości min. i max.

Zakładamy, że projektowane urządzenie będzie zapamiętywało wartości minimalne i maksymalne temperatury oraz wilgotności, zatem interfejs należy jeszcze nieco rozbudować. Sprawdźmy, czy dobrym pomysłem będzie użycie do tego celu horyzontalnego (jedna obok drugiej) ułożenia kontrolki. Oczywiście nie wszystkich, lecz tylko tych, które wskazują te same wielkości. Efekt wizualny tego pomysłu, tym razem zaprezentowany na obróconym o 90 stopni ekranie smartfona, pokazuje **rysunek 13**. Implementację tego rozwiązania pokazuje kod JSON z **listingu 5**. Opisuje on sześć głównych elementów interfejsu graficznego. Są to:

1. kontrolka wyświetlająca napis „Temperatura” (linia 2);
2. layout układający w swoim wnętrzu horyzontalnie (jedna obok drugiej) trzy kontrolki (linie od 3 do 6). Wyświetlają one wartości temperatury: minimalną, aktualną i maksymalną;
3. puste pole tekstowe podnoszące czytelność interfejsu (linia 7);
4. kontrolka wyświetlająca napis „Wilgotność” (linia 8);
5. drugi layout horyzontalny zawierający w swoim wnętrzu trzy kontrolki (linie od 9 do 12), tym razem wyświetlające wartości wilgotności: minimalną, aktualną i maksymalną;
6. puste pole tekstowe podnoszące czytelność interfejsu (linia 13).

Ta odsłona interfejsu jest już dość blisko celu, który określiliśmy na starcie. Wydaje się, że cyfry są zbyt duże, co uniemożliwi czytelne ich wyświetlenie w pionowej orientacji ekranu. Po prostu nie mieszczą się w jednym, poziomym wierszu i utworzą kolejną linię, co mocno obniży funkcjonalność interfejsu. Popracujemy nad tym w kolejnej części publikacji i zorganizujemy też sterowanie interfejsem.

Mariusz Żądło

Listing 3. Kod JSON rozbudowujący pierwszy interfejs

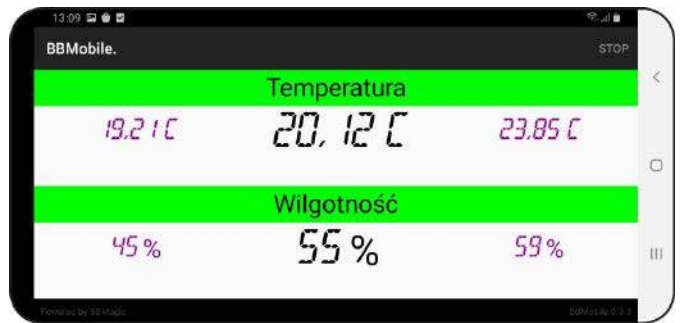
```
1 {"ty":"lout","or":"V","cs":[
2 {"ty":"TextView","te":"\nTemperatura","tc":"0,0,0","ts":"40"},
3 {"ty":"TextView","n":"t","te":"20,1 C","tc":"0,0,0","ts":"50","tf":"02"},
4 {"ty":"TextView","te":"\nwilgotno\u015B\u0107","tc":"0,0,0","ts":"40"},
5 {"ty":"TextView","n":"h","te":"52 %","tc":"0,0,0","ts":"50","tf":"02"},
6 {"ty":"TextView","w":"5"}]}
```

Listing 4. Kod JSON zmieniający wygląd interfejsu

```
1 {"ty":"lout","or":"V","img":"3","cs":[
2 {"ty":"TextView","te":"\nTemperatura","tc":"250,250,250","ts":"40"},
3 {"ty":"TextView","n":"t","te":"20,1 C","tc":"250,250,250","ts":"50","tf":"02"},
4 {"ty":"TextView","te":"\nwilgotno\u015B\u0107","tc":"250,250,250","ts":"40"},
5 {"ty":"TextView","n":"h","te":"52 %","tc":"250,250,250","ts":"50","tf":"02"},
6 {"ty":"TextView","w":"5"}]}
```

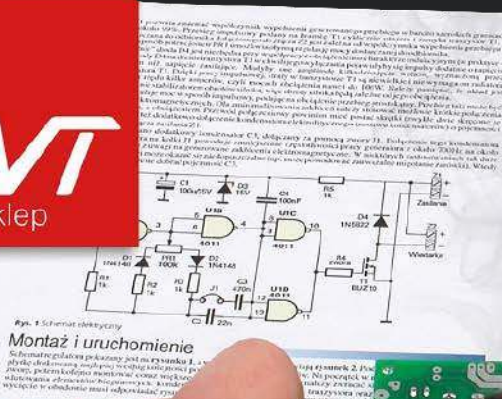
Listing 5. Kod JSON dodający do aplikacji wyświetlanie wartości maksymalnych i minimalnych

```
1 {"ty":"lout","or":"V","cs":[
2 {"ty":"TextView","te":"Temperatura","tc":"0,0,0","bg":"0,255,0","ts":"30"},
3 {"ty":"lout","or":"H","cs":[
4 {"ty":"TextView","n":"tm","te":"19,21 C","tc":"150,0,150","ts":"30","tf":"02"},
5 {"ty":"TextView","n":"t","te":"20,12 C","tc":"0,0,0","ts":"50","tf":"02"},
6 {"ty":"TextView","n":"tx","te":"23,85 C","tc":"150,0,150","ts":"30","tf":"02"}]},
7 {"ty":"TextView","te":"0,0,0","ts":"30"},
8 {"ty":"TextView","te":"Wilgotno\u015B\u0107","tc":"0,0,0","bg":"0,255,0","ts":"30"},
9 {"ty":"lout","or":"H","cs":[
10 {"ty":"TextView","n":"hm","te":"45 %","tc":"150,0,150","ts":"30","tf":"02"},
11 {"ty":"TextView","n":"h","te":"55 %","tc":"0,0,0","ts":"50","tf":"02"},
12 {"ty":"TextView","n":"hx","te":"59 %","tc":"150,0,150","ts":"30","tf":"02"}]},
13 {"ty":"TextView","w":"5"}]}
```



Rysunek 13. Wygląd aplikacji mobilnej rozbudowanej o wskazywanie wartości maksymalnych i minimalnych

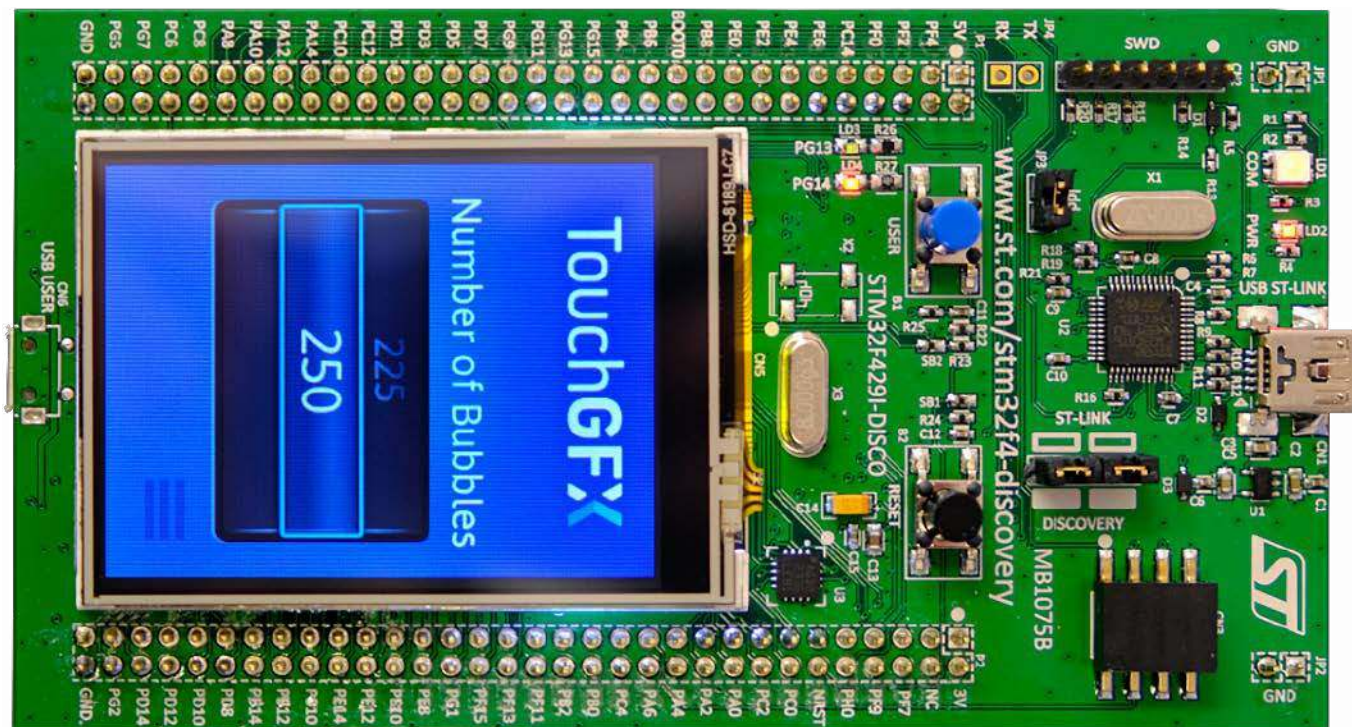
REKLAMA



**KITY
AVT**

KITY AVT
@KITYAVT • Elektronika

<http://bit.ly/2BjVMN7>



Projektowanie interfejsów graficznych z użyciem TouchGFX (4)



W poprzedniej części ukończyliśmy prosty interfejs użytkownika i rozpoczęliśmy analizę działania silnika grafiki TouchGFX. W ostatniej części tego cyklu dokończymy analizę, ponieważ pozwoli to zrozumieć, jak jest zorganizowany transfer danych do sterownika wyświetlacza i jak uniknąć wyświetlania zakłóceń na ekranie.

Silnik TouchGFX zobrazowaliśmy jako nieskończoną pętlę wykonującą cykle składające się z trzech podstawowych czynności:

- zbieranie zdarzeń – zbiera zdarzenia z ekranu dotykowego, naciśnięcia fizycznych przycisków, wiadomości/sygnały z systemów podrzędnych na przykład z interfejsów szeregowych,
- aktualizowanie modelu sceny – reaguje na zebrane zdarzenia, aktualizuje pozycje, animacje, kolory, obrazy,
- renderowanie modelu sceny – przerysowuje części modelu, który został zaktualizowany i wyświetla je na ekranie.

Zbieranie zdarzeń

W tej fazie silnik graficzny zbiera zdarzenia ze środowiska zewnętrznego. Te zdarzenia to zazwyczaj zdarzenia związane z ekranem dotykowym i przyciskaniem przycisków zewnętrznych. TouchGFX próbuje te elementy i przesyła wykryte zdarzenia do warstwy aplikacji. W przypadku ekranów dotykowych TouchGFX wykrywa następujące zdarzenia:

- kliknięcie – użytkownik nacisnął lub zdjął palec z wyświetlacza,
- przeciągnięcie – użytkownik przesunął palcem po wyświetlaczu (dotykając wyświetlacza),
- gesty – użytkownik szybko przesunął palec w wybranym kierunku, a następnie puścił. Nazywa się to machnięciem i jest rozpoznawane przez silnik graficzny.

Po wykryciu i zidentyfikowaniu są one przekazywane do aktualnie aktywnych elementów interfejsu użytkownika (np. widżetów).

Wyróżnionym zdarzeniem jest zdarzenie tick. Tick reprezentuje nową ramkę (lub krok w czasie) i jest zawsze wysyłane, także wtedy, gdy nie było innego zdarzenia. Jest ono używane przez aplikacje do uruchamiania animacji lub innych działań opartych na czasie, takich jak przejście do ekranu pauzy po upływie określonego czasu.

Aktualizowanie modelu

Jednym z zadań silnika graficznego jest współpraca z warstwą aplikacji mającą na celu aktualizację interfejsu użytkownika. Ta aktualizacja odzwierciedla zebrane zdarzenia. Silnik graficzny wie, który ekran (screen) jest aktualnie aktywny i przekazuje zdarzenia do obiektów tego ekranu.

Inaczej mówiąc, silnik informuje aplikację o zdarzeniach. W odpowiedzi aplikacja żąda przerysowania określonych elementów na wyświetlaczu. Aplikacja w modelu zachowanym (*retained*) nie rysuje bezpośrednio w odpowiedzi na zdarzenia, ale zmienia właściwości widżetów i żąda przerysowania przez warstwę Renderuj modelu silnika graficznego.

Jeśli na przykład wystąpi zdarzenie Click, silnik graficzny przeszukuje model sceny obiektu Screen, aby znaleźć widżet, który powinien odebrać zdarzenie. Niektóre widżety, takie jak Image i TextArea, nie obsługują zdarzeń Click i mają pustą procedurę obsługi zdarzeń, więc wtedy nic się nie dzieje. Inne widżety, takie jak przycisk, reagują na zdarzenie Click (naciśnięte lub zwolnione). Widżet Przycisk (*button*) zmienia swój stan, aby wyświetlić inny obraz po naciśnięciu i znów zmienia stan po ponownym zwolnieniu dotknięcia. Dlatego musi być ponownie „narysowany” w buforze ramki wyświetlacza. Co znamienne, silnik grafiki sam nie inicjuje tego przerysowania widżetów na podstawie zebranych zdarzeń. Widżety śledzą swój własny stan wewnętrzny i instruuja silnik, żeby przerysował część ekranu (na przykład prostokąt) obrazujący widżet.

Sama aplikacja może również reagować na zdarzenia na jeden z dwu sposobów:

- Przez konfigurację interakcji dla widżetu w TouchGFX Designer. Na przykład, możemy skonfigurować interakcję, aby kolejny widżet był widoczny po naciśnięciu przycisku. Ta interakcja jest wykonywana po zmianie stanu przycisku i zażądaniu ponownego rysowania od silnika graficznego. Jeśli używamy interakcji, aby wyświetlić inny (niewidoczny) widżet, aplikacja powinna również zażądać przerysowania z silnika graficznego;
- Reagowanie na zdarzenia na ekranie. Możliwe jest również reagowanie na zdarzenia na samym ekranie. Program obsługi zdarzeń to funkcje wirtualne w klasie Screen. Funkcje te można ponownie zaimplementować na ekranach w aplikacji. Można to na przykład wykorzystać do wykonania akcji za każdym razem, gdy użytkownik dotknie ekranu, bez względu na to, który widżet jest dotykany.

Aktualizacje czasowe

Procedura obsługi zdarzenia *handleTickEvent* jest wywoływana na początku każdej ramki. Dzięki temu aplikacja może przeprowadzać aktualizacje interfejsu użytkownika w oparciu na czasie. Przykładem może być zniknięcie widżetu po 10 sekundach. Zakładając, że mamy 60 klatek na sekundę, kod mógłby wyglądać tak, jak na **listingu 1**. Silnik graficzny wywołuje również procedurę obsługi zdarzeń w klasie *Model*. Ten program obsługi zdarzeń jest zwykle używany do wykonywania powtarzających się czynności, takich jak sprawdzanie kolejnych komunikatów lub próbkowanie GPIO (**listing 2**).

Jak omówiliśmy powyżej w przykładzie Button, widżety są odpowiedzialne za żądanie przerysowania, gdy zmienia się ich stan. Taki mechanizm nazywany jest unieważnionym obszarem (*invalidated area*). Kiedy przycisk zmienia stan (np. ze zwolnionego na wciśnięty) i wymaga przerysowania, obszar objęty widżetem przycisku staje się obszarem unieważnionym. Silnik graficzny przechowuje listę tych unieważnionych obszarów żądanych dla ramki. Wszystkie zebrane zdarzenia (dotknięcie, przycisk, tik) mogą skutkować jednym lub kilkoma unieważnionymi obszarami, więc w każdej klatce może być wiele unieważnionych obszarów.

Procedury obsługi zdarzeń w klasie Screen mogą również zażądać odświeżenia obszaru. Tutaj zmieniamy kolor widżetu Box, box1, w ramce 10 i żądamy przerysowania, wywołując metodę unieważnienia (*invalidate*) (**listing 3**). W pokazanym przykładzie silnik graficzny będzie wywoływał procedurę obsługi zdarzenia *handleTickEvent* w każdej ramce. W ramce 10 kod aplikacji żąda przerysowania obszaru objętego widżetem box1. W odpowiedzi na to silnik graficzny przerysuje ten obszar w buforze ramki, używając koloru zapisanego w argumentach *setColor*.

W kolejnym przykładzie interfejsu użytkownika pokazanym poniżej mamy widżety Button i Box. Jeśli wstawimy interakcję zmieniającą kolor ramki po kliknięciu przycisku, to otrzymamy dwa unieważnione obszary (zaznaczone na czerwono). Jeden dotyczy

Listing 1. Zależności czasowe

```
void handleTickEvent ( ) {
    tickCounter += 1;
    if ( tickCounter == 600 ) {
        // Przejście do 0 = niewidoczne w 20 //klatkach
        myWidget . startFadeAnimation ( 0 , 20 ) ;
    }
}
```

Listing 2. Odpytywanie GPIO w klasie Model

```
void Model :: tick ( ){
    // Przykładowe odpytywanie GPIO
    bool b = sampleGPIO_Input1 ( ) ;
    if ( b ){
        ...
    }
}
```

Listing 3. Przerysowanie widżetu box1

```
void handleTickEvent() {
    tickCounter += 1;
    if (tickCounter == 10) {
        // ustawienie koloru czerwonego
        box1.setColor(Color::getColorFrom24BitRGB(0xFF, 0x00, 0x00));
        // zezwolenie na przerysowanie
        box1.invalidate();
    }
}
```

przerysowania przycisku po jego dotknięciu, a drugi przerysowania koloru obszaru box (**rysunek 1**).

Renderowanie

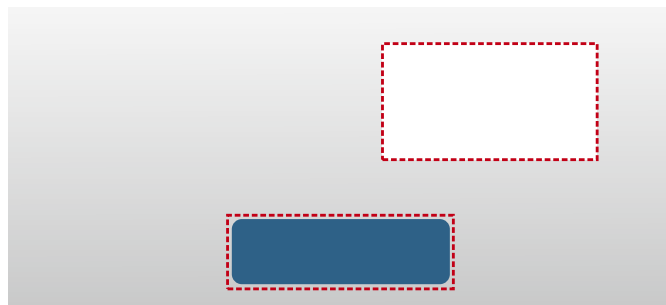
Tak jak to omówiliśmy, wynikiem fazy aktualizacji jest lista obszarów do przerysowania, nazwanymi obszarami unieważnionymi. Zadaniem fazy renderowania modelu jest przejrzanie tej listy i narysowanie widżetów pokrywających te obszary w buforze ramki. Faza renderowania modelu jest obsługiwana automatycznie przez silnik graficzny. Aplikacja ma za zadanie zdefiniowanie modelu sceny (widżety w interfejsie użytkownika) i unieważnienie określonych obszarów. Silnik graficzny obsługuje unieważnione obszary jeden po drugim. Dla każdego obszaru silnik skanuje model sceny i gromadzi listę widżetów objętych tym obszarem (częściowo lub w całości). Mając tę listę widżetów, silnik wywołuje metodę rysowania na widżetach, zaczynając od widżetu w tle, a kończąc na głównym widżecie.

Metody rysowania widżetu używają parametrów jego stanu np. koloru, podczas rysowania do bufora ramki. Wszelkie informacje potrzebne do narysowania widżetu muszą zostać zapisane w widżecie podczas fazy aktualizacji. W przeciwnym razie te informacje nie są dostępne w fazie renderowania.

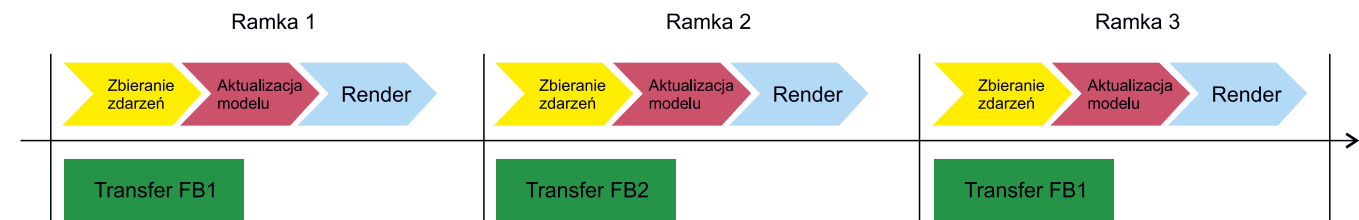
Stan wstrzymania Wait

Silnik graficzny TouchGFX czeka na sygnał gotowości przed aktualizacją i renderowaniem następnej klatki. Istnieją dwa powody, dla których należy czekać między klatkami zamiast ciągłego renderowania ramek tak szybko, jak to możliwe:

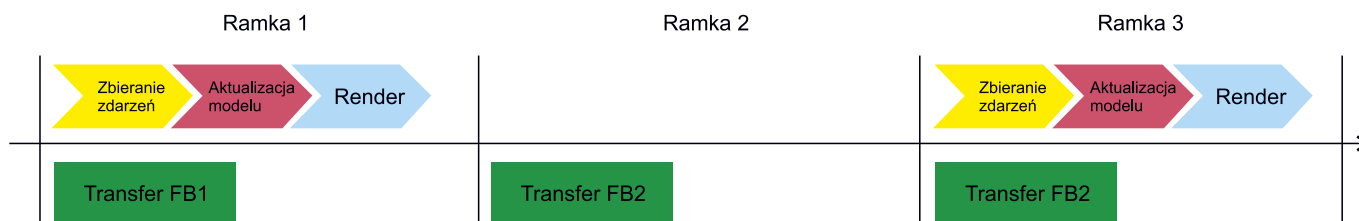
- Renderowanie jest zsynchronizowane z wyświetlaczem. Niektóre wyświetlacze wymagają wielokrotnego przesyłania bufora ramki. Podczas transmisji nie zaleca się arbitralnego renderowania



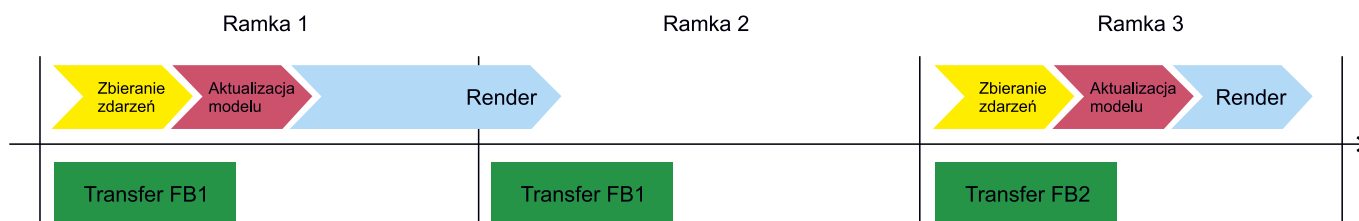
Rysunek 1. Przykład dwu obszarów unieważnionych interfejsu użytkownika



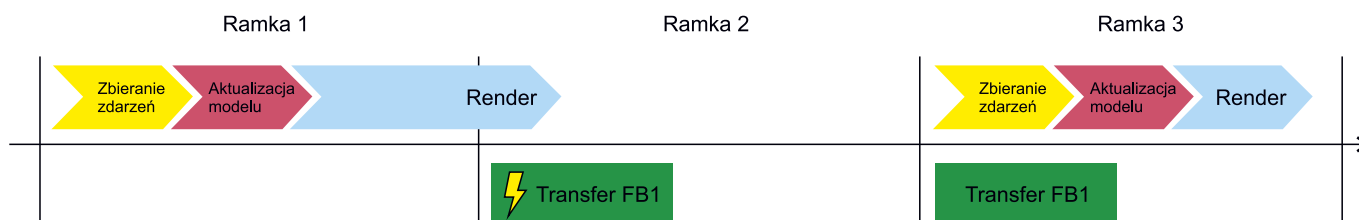
Rysunek 2. Podwójny bufor ramki



Rysunek 3. Ramka 2 nie jest aktualizowana



Rysunek 4. Dłuższe renderowanie ramki 1



Rysunek 5. Jeden bufor ramki i długie renderowanie

do bufora ramki. Dlatego silnik graficzny czeka przez krótki czas po uruchomieniu transmisji przed rozpoczęciem renderowania. Inne wyświetlacze wysyłają sygnał do mikrokontrolera, kiedy należy przesłać bufor ramki. Silnik graficzny czeka na ten sygnał;

- Ramki są renderowane ze stałą szybkością. Dla aplikacji często korzystne jest renderowanie klatek ze stałą szybkością, ponieważ ułatwia to tworzenie animacji trwających przez określony czas. Na przykład, jeśli masz wyświetlacz 60 Hz, dwusekundowa animacja powinna zostać zaprogramowana tak, aby kończyła się w 120 klatkach.

Czas oczekiwania silnika graficznego jest zwykle używany przez inne procesy o niższym priorytecie w aplikacji. W takich przypadkach czas nie jest marnowany, ponieważ procesy o niższym priorytecie i tak powinny być uruchomione w pewnym momencie.

Obsługa buforów ramki

Jak już wiemy, silnik graficzny synchronizuje się z wyświetlaczem przed aktualizacją bufora ramki. Po wyrenderowaniu do bufora ramki silnik musi również upewnić się, że wyświetlacz pokazuje zaktualizowany bufor ramki.

Dwa bufor ramki

W najprostszej konfiguracji dostępne są dwa bufor ramki. Silnik graficzny pracuje naprzemiennie między dwoma buforami ramki. Podczas rysowania ramki do bufora ramki drugi bufor ramki jest przesyłany i pokazywany na wyświetlaczu (rysunek 2). Zakładamy, że wyświetlacz ma równoległy interfejs RGB podłączony do kontrolera LTDC. Oznacza to, że bufor ramki musi być przesyłany do wyświetlacza w każdej ramce. Ponieważ mamy dwa bufor ramki, silnik

graficzny może rysować do jednego bufora ramki, podczas gdy drugi jest przesyłany do wyświetlacza. Ten schemat działa bardzo dobrze i jest preferowany, jeśli to możliwe.

Często zdarzają się ramki, w których aplikacja niczego nie aktualizuje. Oznacza to, że nic nie jest renderowane. Dlatego ten sam bufor ramki jest przesyłany ponownie w następnej ramce. Na rysunku 3 aplikacja nie rysuje niczego w ramce 2, więc silnik graficzny retransmituje bufor ramki 2 ponownie w ramce 3.

Typowy równoległy wyświetlacz RGB ma częstotliwość odświeżania około 60 Hz. Ta częstotliwość aktualizacji musi być utrzymywana przez mikrokontroler. Częstotliwość odświeżania 60 Hz oznacza, że mamy ok. 16 ms na renderowanie nowej ramki przed ponownym rozpoczęciem transmisji. W niektórych przypadkach ten czas jest dłuższy niż 16 ms, tak jak to pokazano na rysunku 4. W tym przypadku silnik graficzny po prostu ponownie przesyła tę samą ramkę. Renderowanie klatki 1 trwa dłużej niż 16 ms, więc ramka 0 wcześniej wyrenderowana do bufora ramki 1 jest ponownie przesyłana. Nowa ramka w buforze ramki 2 jest przesyłana w ramce 3. Gdy dostępne są dwa bufor ramki, czas renderowania może być bardzo długi. Poprzednia ramka jest retransmitowana do momentu udostępnienia nowej ramki.

Jeden bufor ramki

W niektórych systemach dostępna jest tylko pamięć dla jednego bufora ramki. Jeśli mamy równoległy wyświetlacz RGB, jesteśmy zmuszeni transmitować bufor ramki 1 w każdej ramce. Może to być problematyczne, ponieważ silnik graficzny jest zmuszony do rysowania do tego samego bufora ramki, który jednocześnie przesyłamy na wyświetlacz. Jeśli zostanie to zrobione bez zachowania pewnych

zasad, istnieje duże ryzyko, że wyświetlacz pokaże klatkę, która jest połączeniem poprzedniej i nowej klatki.

Jednym z rozwiązań jest wstrzymanie rysowania do zakończenia transferu i rysowanie tylko w przedziale czasowym przed ponownym rozpoczęciem transferu. Daje to mało czasu na narysowanie, ponieważ transfer zajmuje znaczną część całkowitego czasu ramki. Może się wtedy zdarzyć, że będą wysyłane niekompletne ramki, bo silnik nie zdążył z renderowaniem przed startem transferu.

Rozwiązaniem eliminującym potencjalnie tę wadę jest śledzenie, jaka część bufora ramki jest już wytransmitowana, a następnie ograniczenie renderowania do tej odpowiedniej części bufora ramki. W miarę postępu transferu coraz więcej bufora ramki jest dostępne dla algorytmów renderowania. Silnik graficzny zawiera algorytmy, które pomagają programiście upewnić się, że rysunek jest wykonywany poprawnie.

Sytuacja, w której czas renderowania jest dłuższy niż 16 ms, renderowanie nie zakończy się, gdy retransmisja rozpocznie się ponownie, jest pokazana na **rysunku 5**. Wtedy silnik graficzny musi upewnić się, że przesyłana część jest całkowicie renderowana. W przeciwnym razie wyświetlacz pokaże niedokończony bufor ramki.

Wydajność systemu graficznego

Przy omawianiu sytuacji pokazanej na rysunku 4 ustaliliśmy, że przy podwójnym buforze ramki czas renderowania może być dowolnie długi, bo do jego zakończenia jest ciągle wysyłana poprzednia ramka. To jest prawdziwe w przypadku, kiedy wyświetlane są obrazy statyczne. Sytuacja się zmienia, kiedy w interfejsie użytkownika chcemy zastosować animacje. Do płynnego wyświetlania animacji niezbędna jest możliwość wyświetlania określonej liczby ramek (klatek) na sekundę. Do tego ta liczba klatek nie powinna się zmieniać w czasie, bo powstaje wtedy wrażenie przyspieszenia lub opóźnienia animacji, co jest odbierane jako bardzo niekorzystne wrażenie wizualne. Żeby wyświetlać zaawansowane animacje, niezbędna jest określona wysoka wydajność układu interfejsu użytkownika. Wysoka wydajność jest tutaj definiowana jako uzyskiwanie dużej liczby klatek na sekundę przy jednoczesnym uzyskiwaniu pożądanego efektów graficznych i animacji.

Przypomnijmy sobie, jak główna pętla wpływa na liczbę klatek interfejsu użytkownika. Założmy ponownie, że do LTDC jest podłączony równoległy wyświetlacz RGB i dwa bufora ramki (rysunek 2). Zakładając, że wyświetlacz jest odświeżany 60 razy na sekundę, między każdym odświeżeniem jest około 16 ms: $1/60 = 0,01667$ s = 16,67 ms.

TouchGFX rozpoczyna rysowanie ramki do bufora ramki 2 w momencie rozpoczęcia przesyłania bufora ramki 1. Jeśli renderowanie ramki 1 zostanie zakończone przed rozpoczęciem następnego przesyłania, możemy przesłać bufor ramki 2. Jeśli nie zakończy się w ciągu 16,67 ms, bufor ramki 1 zostanie ponownie przesłany i wyświetlacz będzie wyglądał niezmieniony tak jak na rysunku 3. W takim przypadku utracimy jedną klatkę animacji. Czas dla faz zbierania zdarzeń i aktualizacji modelu jest zwykle niewielki, np. krótszy niż 1 ms. Dlatego w dalszej części rozważań dla uproszczenia przyjmujemy, że czas renderowania jako najdłuższy zawiera też fazy zbierania danych i aktualizacji modelu.

Jeśli czas renderowania w wielu klatkach przekroczy limit wynoszący 16,67 ms, liczba klatek na sekundę na wyświetlaczu realnie wyniesie 30 klatek na sekundę (fps). Jeśli renderowanie jest generalnie krótsze niż 16,67 ms, ale w niektórych klatkach dłuższych niż 16,67 ms, średnia liczba klatek na sekundę może być bliska 60 fps, ale animacja może nie wyglądać poprawnie. W zależności od aplikacji może wyglądać, że część animacji działa szybko (poprawnie), a część zwalnia. To nie jest pożądanym zjawiskiem. Jeżeli czas renderowania jest jeszcze dłuższy, na przykład nieco powyżej 33 ms, realna liczba klatek na sekundę spadnie do 20 fps, ponieważ mamy tylko nową ramkę gotową na co trzeci transfer. Zostało to pokazane na **rysunku 6**.

Zaawansowany użytkownik może mierzyć czas pomiędzy kolejnymi transferami. Silnik graficzny wywołuje funkcję w klasie GPIO, gdy rozpoczyna się faza zbierania zdarzeń i wykonuje kolejne wywołanie

po zakończeniu fazy renderowania. Aplikacja definiuje te funkcje. Mogą to być na przykład zmiany stanów linii GPIO. Potem za pomocą oscyloskopu można zmierzyć czas renderowania kolejnych ramek. Alternatywną metodą jest zliczanie utraconych klatek. Silnik graficzny zlicza liczbę transferów, które miały miejsce podczas ostatniej fazy zbierania-aktualizowania-renderowania. Aplikacja może łatwo sprawdzić tę wartość, aby zobaczyć, czy klatka została utracona, a tym samym zmniejszyła się liczba klatek na sekundę.

Kiedy czas renderowania jest za długi i klatki się gubią, a co za tym idzie, zmniejsza się liczba klatek na sekundę w jednej z naszych animacji, możemy to do pewnego stopnia skompensować poprzez jedno z działań:

- poczekaj – pozwól animacji trwać, co spowoduje wydłużenie czasu trwania animacji i prawdopodobnie płynną animację,
- pominię niektóre klatki – pomijając klatki, upewnij się, że ogólna animacja nie trwa dłużej, niż zamierzano.

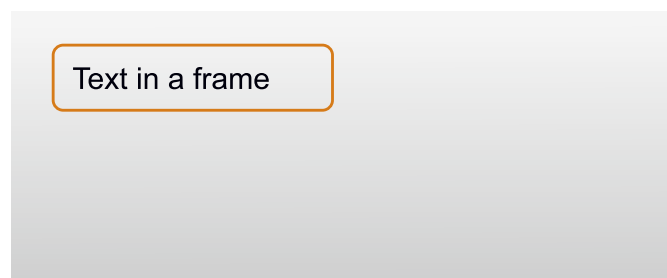
TouchGFX może zostać poinstruowany, aby automatycznie pomijał niektóre klatki, gdy zostaną utracone. Można to osiągnąć, zaznaczając animacje więcej niż raz na rzeczywistą klatkę. Może to pomóc w zwiększeniu płynności animacji, gdy czas renderowania jest nierówny.

Na czas renderowania wpływa wiele różnych czynników: rozmiar zaktualizowanych obszarów, użycie warstw, złożoność widżetów oraz dostępna sprzętowa obsługa przerysowania. Czas renderowania jest zwykle proporcjonalny do liczby pikseli, które należy zaktualizować. Jeśli więc przerysowanie animacji zajmuje zbyt dużo czasu, możliwym rozwiązaniem jest zmniejszenie obszaru animacji. Na przykład, jeśli masz obracający się obraz, a wydajność systemu nie jest wystarczająco dobra, można ją poprawić, zmniejszając rozmiar obrazu. Należy pamiętać, że silnik graficzny przerysowuje obszary, które aplikacja unieważniła. Dlatego ważne jest, aby unieważnić tylko te obszary, które faktycznie wymagają odświeżenia.

W typowej aplikacji grafika będzie składać się z różnych elementów, które są ułożone jeden na drugim w warstwach. Jeśli jeden z elementów jest aktualizowany, wszystkie elementy warstw muszą być zwykle przerysowane.

Typowy przykład został pokazany na **rysunku 7**. Ten interfejs użytkownika jest tworzony przez umieszczenie widżetu TextArea na widżecie Obraz wyświetlającym przezroczystą ramkę. To rozwiązanie jest bardzo często stosowane w aplikacji, bo jest bardzo proste i zapewnia dużą elastyczność. Można np. zmieniać ramkę w czasie wykonywania lub przesuwać ramkę i tekst w tle.

Jeśli tekst jest aktualizowany w czasie wykonywania i wymaga ponownego narysowania, silnik graficzny również musi przerysować tło i ramkę a następnie nowy tekst. Znacznie wydłuża to czas renderowania tekstu. Im więcej warstw do przerysowania, tym dłuższy czas jest potrzebny na przerysowanie.



Rysunek 7. Typowym przykładem jest obraz tła, ramka i tekst

FPS	Maksymalny czas renderowania
60	16,67 ms
30	33,34 ms
20	50,00 ms
15	66,67 ms

Rysunek 6. Maksymalny czas renderowania w połączeniu z liczbą klatek na sekundę (FPS)

We wszystkich typach renderowania silnik graficzny musi zapisać wynikowy piksel do bufora ramki. Okazuje się, że nakład mocy obliczeniowej na wyliczenie wartości reprezentującej piksel w buforze nie jest jednakowy w każdym przypadku. Na przykład w widzącej Box wszystkie piksele mają ten sam kolor i nie ma potrzeby wyliczania ich wartości za każdym razem, kiedy jest na przykład przesuwany czy obracany. W interfejsie zbudowanym z wielu takich elementów nie jest potrzebna duża moc obliczeniowa, ale z drugiej strony jakość i atrakcyjność takiego interfejsu jest problematyczna.

Podobnie jest w przypadku obrazków, ponieważ piksele są przechowywane w mapach bitowych w formacie gotowym do użycia. Obliczenie piksela zapisanego w buforze ramki polega na załadowaniu wartości koloru z odpowiedniego miejsca w mapie bitowej.

Wyświetlanie tekstu można traktować jak wyświetlanie małych bitmap. Jednak tu jest potrzebna trochę większa wydajność na przykład do obliczania pozycji kolejnych liter. Ładnie wyglądający tekst będzie wymagał wyświetlania z przezroczystością tak jak w przykładzie z rysunku 7. Przezroczystość zwiększa zapotrzebowanie na obliczenia niezbędne do narysowania elementu, ponieważ silnik graficzny musi najpierw narysować element za przezroczystym elementem (tak jak w przykładzie z rysunku 7). W kolejnym kroku silnik graficzny musi połączyć piksel tła z pikselem elementu przezroczystego i zapisać wynik do bufora ramki. Te obliczenia zajmują znacznie więcej czasu niż samo zapisanie obliczonego piksela.

Obracanie i operacje przeskalowania obrazków wymagają również dużej mocy obliczeniowej. Co prawda piksele są pobierane z mapy bitowej, ale do ich wyświetlania należy przeliczyć nową pozycję, uwzględniając rotację i/lub skalowanie. Rysowanie elementów geometrycznych jest jedną z najbardziej obciążających operacji. Nie możemy załadować koloru piksela z mapy bitowej, ale musimy obliczyć zarówno kształt figury (na przykład koła), jak i kolor poszczególnych pikseli.

Przy projektowaniu interfejsu trzeba pamiętać o kilku ważnych zasadach pozwalających na uniknięcie niepotrzebnego obciążania systemu (mikrokontrolera):

- Nie zlecaj silnikowi grafiki przerysowania niezmiennych elementów grafiki. Upewnij się, że przypadkowo nie unieważnisz (*invalidated area*) niepotrzebnych części wyświetlacza. Zwiększa to obciążenie systemu bez żadnych korzyści;
- Znajdź równowagę między jakością interfejsu a szybkością. Zmniejszenie złożoności elementów może poprawić szybkość i jakość działania. Często kluczem jest odpowiednia równowaga między tymi elementami;
- Wykorzystaj możliwości sprzętowe. Możliwości graficzne mikrokontrolera z akceleracją sprzętową (Chrom-ART) są często większe niż bez niej. Rozważ użycie mikrokontrolera z Chrom-ART;
- Zastąp konieczność obliczania elementów grafiki obrazami (bitmapami). Wyliczanie narysowania okręgu jest wolniejsze niż wyświetlenie obrazu koła. Obrazy mogą zastąpić wiele statycznych elementów;
- Dostosuj częstotliwość odświeżania ekranu. Jak to omówiliśmy, częstotliwość odświeżania jest sztywnym ograniczeniem czasu

renderowania. Jeśli czas renderowania przekracza częstotliwość odświeżania, liczba klatek spada. Jeśli czas renderowania jest tylko trochę dłuższy od częstotliwości odświeżania, możliwe jest obniżenie częstotliwości odświeżania wyświetlacza do np. 55 Hz (co odpowiada 18,2 ms) i utrzymanie wysokiej częstotliwości odświeżania.

Do tej pory skupiliśmy się na zapewnieniu interfejsowi użytkownika (HMI) poprawnego, płynnego działania. Jak wiemy, sprzęt potrzebny do tego celu: mikrokontroler, najlepiej z akceleratorem grafiki i pozostałe zasoby, powinien być dość wydajny i rozbudowany jak na systemy wbudowane (*embedded*). HMI w systemach wbudowanych to tylko (lub aż) interfejs zapewniający interakcję z właściwym systemem sterowania lub nadzoru. Należy sobie postawić pytanie, czy dość zaawansowany system obsługujący HMI można wykorzystać do tych innych zadań. Z poprzednich rozważań wiemy, że silnik grafiki TouchGFX może w pewnych przypadkach potrzebować zająć cały czas procesora. Przy skomplikowanych animacjach nawet ten czas może nie być wystarczający. Trudno sobie wyobrazić, żeby interfejs graficzny miał wyższe priorytety i zwalniał podstawowe procesy sterowania i nadzoru systemu wbudowanego.

W prostym urządzeniu z graficznym interfejsem użytkownika i tylko kilkoma prostymi zadaniami pomocniczymi, takimi jak prosty timer, pomiar temperatury czy funkcja zegara RTC, możliwe jest zorganizowanie całej aplikacji wokół kodu interfejsu użytkownika. Aplikacja robi bardzo niewiele poza regularnymi aktualizacjami interfejsu użytkownika, więc wykonywanie innych zadań może z dużym powodzeniem zostać osadzone w kodzie interfejsu użytkownika. Ale kiedy urządzenie zawiera bardziej zaawansowaną funkcjonalność, która „działa w tle” z oddzielnymi wymaganiami czasowymi, na przykład takimi jak sterowanie silnika BLDC, szybko staje się trudne zintegrowanie tych dwóch zadań w jedno przy jednoczesnym spełnieniu wymagań stawianym aplikacji.

Jak już wiemy, silnik graficzny musi rysować nowe ramki, aby obsługiwać płynny interfejs użytkownika. Jeśli wstrzymamy to rysowanie podczas wykonywania innych zadań, to liczba klatek na sekundę spadnie. Podobnie, jeśli inne zadania działają tylko między klatkami, w czasie bezczynności silnika grafiki, wtedy inne zadania będą wstrzymywane, gdy interfejs użytkownika renderuje złożone sceny. Te zależności utrudniają lub nawet uniemożliwiają przeplatanie zadania interfejsu użytkownika innymi złożonymi zadaniami sterowania systemem wbudowanym.

TouchGFX działa pod kontrolą systemu FreeRTOS. Funkcje systemowe pozwalają na uporządkowanie, kontrolę i współpracę pomiędzy uruchomionymi zadaniami. FreeRTOS zapewnia też standardowe mechanizmy komunikacji pomiędzy zadaniami, nadawanie im priorytetów i obsługę przerwań. Znacznie ułatwia to tworzenie złożonych aplikacji, ale w przypadku niedoborów wydajności nie likwiduje opisywanych wyżej problemów. Często, jeżeli chcemy mieć pewny i wydajny system sterowania i atrakcyjny interfejs HMI, trzeba obie te części podzielić na osobne systemy i zapewnić współpracę przez standaryzowany protokół, na przykład MODBUS.

Tomasz Jabłoński, EP

*O projektach, mini, soft i wielu
innych dyskutuj
na <https://forum.ep.com.pl>*

FPGA a open source

Prosty mikrokontroler DO6502



Materiały dodatkowe
<https://bit.ly/2SBK3VW>

W cyklu artykułów z EP 9/20 i EP 2/21 poznaliśmy otwarte narzędzia przeznaczone do syntezy FPGA oraz płytkę ewaluacyjną z układem ICE40 zrealizowaną jako projekt open hardware. Dysponując kompletem potrzebnych narzędzi pokażemy, w jaki sposób zbudować syntezyzowany mikrokontroler w oparciu na rdzeniu 65C02, znanym z pierwszych komputerów domowych lat 80. Prezentowany projekt ma charakter przede wszystkim edukacyjny. W praktycznych zastosowaniach lepszym rozwiązaniem będzie użycie rdzenia RISC-V. Jednak istotnym aspektem wyboru rdzenia jest jego prostota oraz sentyment do starych komputerów z lat 80. Projekt możemy uruchomić na płytce ewaluacyjnej „IceCore ICE40 HX” opisanej w poprzednim artykule.



Mikrokontroler typu DO6502 charakteryzuje się następującymi parametrami:

- rdzeń 65C02 taktowany zegarem 6,25 MHz,
- pamięć RAM 4 kB,
- pamięć ROM 8 kB,
- interfejs USART z 16-bajtowym buforem FIFO o prędkości transmisji 57600 bps,
- wyjściowy port GPIO sterujący 4 diodami LED.

Zestaw peryferiów prezentowanego mikrokontrolera jest skromny i ma na celu sprawdzenie, czy otwarte narzędzia do syntezy FPGA mogą być stosowane do projektowania rozbudowanych projektów, takich jak implementacja własnego mikrokontrolera. W miarę potrzeb projekt możemy rozbudować o dodatkowe układy peryferyjne, ponieważ obecnie zajmuje on jedynie 15% bloków układu ICE40, a zatem mamy wystarczająco dużo zasobów, aby znacząco rozbudować mikrokontroler.

Budowa mikrokontrolera

Schemat blokowy mikrokontrolera DO6502 został pokazany na **rysunku 1**. Sercem układu jest miękki rdzeń 65C02 pierwotnie opracowany przez Arleta Ottensa, a następnie zmodyfikowany przez Davida Banksa, i dostępny jest na platformie GitHub. Rdzeń jest w pełni zgodny z listą instrukcji mikroprocesora 65C02 i bez problemu przechodzi test Dorman 65C02 potwierdzający zgodność programową z listą instrukcji oryginalnego 65C02.

Interfejs zewnętrzny został dostosowany do specyfiki implementacji FPGA i ma dwie oddzielne magistrale: danych wejściowych (DI) oraz wyjściowych (DO). Mikroprocesor może współpracować jedynie z pamięciami synchronicznymi, gdzie dane na magistrali DI są oczekiwane w kolejnym cyklu po wystawieniu sygnału na magistrali adresowej ADDR. Pozostałe sygnały takie jak RDY, NMI, IRQ są zgodne z oryginalnym mikroprocesorem 65C02. Przykładowe sygnały na magistrali podczas wykonania programu z pamięci synchronicznej zostały pokazane na **rysunku 2**.

Tabela 1. Mapa pamięci układu

Adres	Rozmiar	Linia	Opis
0x0000	2048	RAMCS	Pamięć RAM
0x0400	256	PERIPHCS	Przestrzeń układów peryferyjnych
0xE000	4096	ROMCS	Wybór pamięci ROM

Jeżeli mikroprocesor chce odczytać dane z pamięci, linia WE ustawiana jest w stan 0, a następnie w kolejnym cyklu mikroprocesor odczytuje dane z magistrali DI. W przypadku gdy procesor chce zapisać dane, wystawia sygnał WE, jednocześnie wystawiając dane do zapisu na magistrali WE.

16-bitowa przestrzeń adresowa podzielona jest na 3 obszary za pomocą głównego dekodera adresowego oznaczonego na schemacie jako ADDR dekod. Dekoder na wyjściu ma sygnały wyboru pamięci RAMCS, sygnały wyboru pamięci ROMCS oraz sygnału wyboru przestrzeni układów peryferyjnych PERIPHCS. Mapa pamięci znajduje się w **tabeli 1**.

Listing 1. Dekoder adresowy zrealizowany jako standardowy układ kombinacyjny

```
module main_address_decoder(
    input [15:0] addr_bus,
    output rom_cs,
    output ram_cs,
    output periph_cs
);

function inbetween(input [15:0] low, input [15:0] value, input [15:0] high);
    inbetween = (value >= low && value <= high);
endfunction

function inany(input [15:0] value);
    inany = (^value==1'b0);
endfunction

reg [2:0] decoder_al;
always @(addr_bus) begin
    (* full_case, parallel_case *)
    case(1'b1)
        inany(addr_bus): decoder_al <= 3'b100;
        inbetween(16'h0000, addr_bus, 16'h0FFF): decoder_al <= 3'b001; //RAM
        inbetween(16'h0400, addr_bus, 16'h04FF): decoder_al <= 3'b010; //Periph
        inbetween(16'hE000, addr_bus, 16'hFFFF): decoder_al <= 3'b100; //BootROM
    endcase
end
assign ram_cs = decoder_al[0];
assign periph_cs = decoder_al[1];
assign rom_cs = decoder_al[2];
endmodule
```

Dekoder adresowy

Dekoder został zrealizowany jako standardowy układ kombinacyjny, którego wejście stanowią sygnały adresowe, natomiast wyjściami są linie sygnałowe wyboru pamięci oraz układów peryferyjnych (**listing 1**). Wejście dekodera stanowi 16-bitowa magistrala input, która jest porównywana z zadeklarowanymi przedziałami adresowymi z wykorzystaniem funkcji `in_between()`. W przypadku spełnienia przedziału dla danego zakresu na odpowiadających liniach wyboru peryferiów wystawiany jest stan wysoki. W aktualnej implementacji mikroprocesora w następnym cyklu zegarowym po zdjęciu sygnału RESET młodszy bajt magistrali adresowej znajduje się w stanie nieustalonym, zatem konieczne stało się dodanie do dekodera adresowego rozpoznawania tego stanu. W przypadku wykrycia stanu nieustalonego na dowolnej linii adresowej wybierana jest pamięć ROM, co umożliwia odczyt pierwszego rozkazu z pamięci oraz prawidłowy start mikroprocesora.

Pamięć RAM

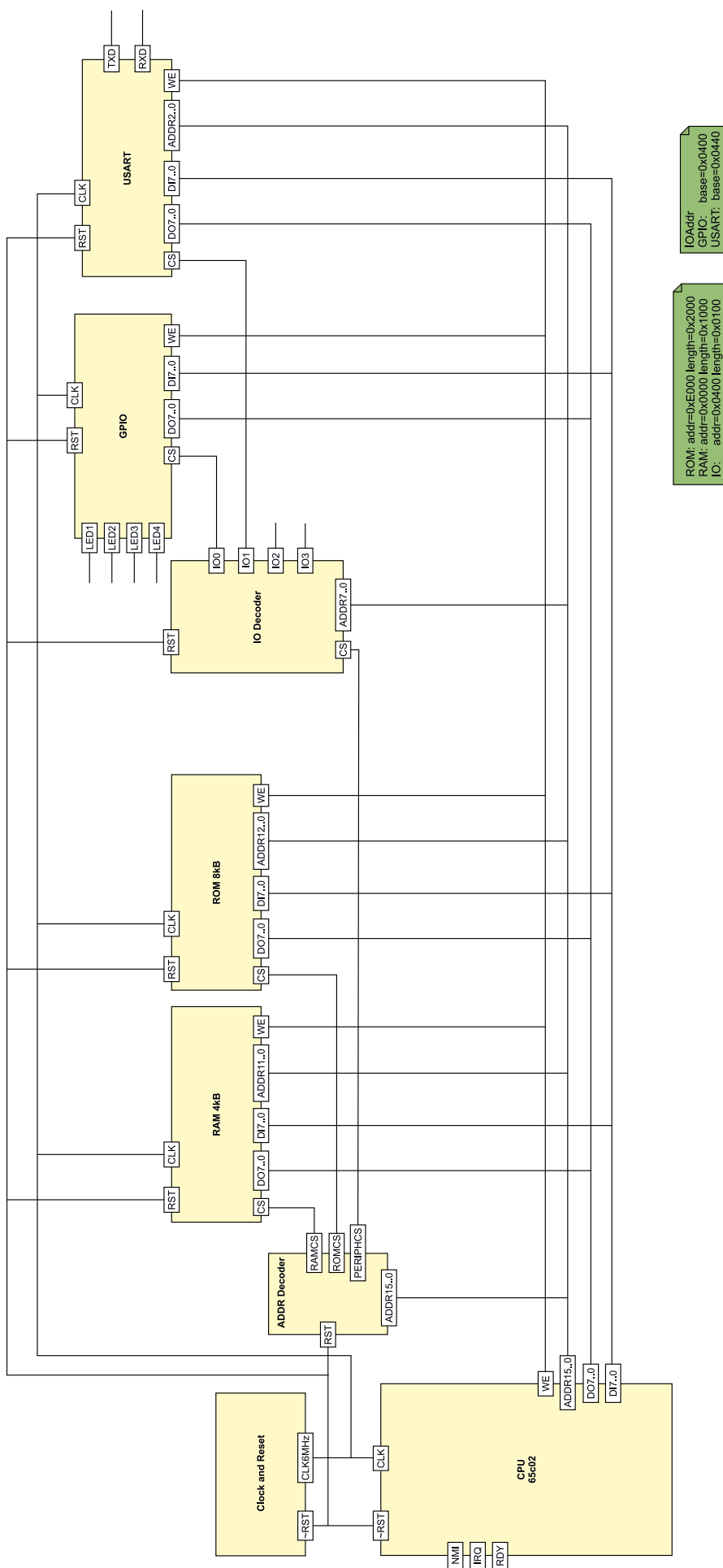
Blok pamięci RAM został zaimplementowany bezpośrednio w VERILOG-u bez użycia modułów specyficznych dla ICE40 celem sprawdzenia, jak otwarte narzędzie nextpnr poradzi sobie z rozpoznaniem bloków, które mogą być zaimplementowane w pamięci BlockRAM (**listing 2**).

Blok synchronicznej pamięci RAM (*internal_ram_notristate*) zawiera parametr konfiguracyjny BIT, który pozwala określić rozmiar pamięci poprzez liczbę bitów magistrali adresowej. Sygnał CLK jest sygnałem zegara pamięci synchronicznej, sygnał WE określa, czy dane będą odczytane (stan 0), czy zapisywane (stan 1). Linie danych pamięci RAM zostały podzielone na magistralę wejściową (DI) oraz magistralę wyjściową (DO), która dodatkowo wyposażona jest w bufor pozwalający na opóźnienie danych o jeden cykl zegarowy, zgodnie z wymaganiami stawianymi przez rdzeń. W bloku warunkowym `__ICARUS` znajduje się wstępne zerowanie pamięci RAM wymagane przez symulator ICARUS VERILOG i jest ono synteżowane tylko w przypadku symulacji.

Pamięć ROM

W analogiczny sposób zrealizowano blok pamięci ROM (**listing 3**), jedyną różnicą w stosunku do pamięci RAM jest brak linii WE. Zawartość pamięci ROM jest odczytywana na etapie syntezy za pomocą polecenia `readmemh` z pliku `capp.hex`. Plik ten generowany jest za pomocą narzędzia hexdump, które zamienia plik binarny wygenerowany przez kompilator na plik zawierający heksadecymalne wartości w postaci tekstu.

Moduły pamięci *internal_rom_no_tristate* i *internal_ram_no_tristate* nie mogą być bezpośrednio dołączone do procesora, ponieważ



Rysunek 1. Schemat blokowy mikrokontrolera DO6502



Rysunek 2. Przykładowe sygnały na magistrali podczas wykonania programu z pamięci synchronicznej

Listing 2. Kod opisujący blok pamięci RAM

```

/* RAM memory */
module internal_ram_no_tristate
#(parameter BITS=10)
(
    input clk,                // Clock input
    input we,                 // WR/RD signal
    input [BITS-1:0] addr,    // Address input
    input [7:0] din,          // Data input
    output reg [7:0] dout     //Data output
);
    reg [7:0] ram_memory[0:2**BITS-1];
    `ifdef ICARUS_
    integer i;
    initial begin
        for(i=0;i<(2**BITS-1);++i)
            ram_memory[i] = 0;
    end
    `endif
    always @(posedge clk) begin
        if(we) ram_memory[addr] <= din;
        else dout <= ram_memory[addr];
    end
endmodule

module internal_ram
#(parameter BITS=10)
(
    input clk,                // Clock input
    input rst,               // Reset signal
    input we,                 // WR/RD signal
    input oe,                 // Chip select signal
    input [BITS-1:0] addr,    // Address input
    input [7:0] din,          // Data input
    output [7:0] dout         //Data output
);
    wire [7:0] idata;
    internal_ram_no_tristate #(BITS(BITS)) ram(
        .clk(clk),
        .we(we&oe),
        .addr(addr),
        .din(din),
        .dout(idata)
    );

    reg oe_del;
    always @(posedge clk or negedge rst)
        if(~rst) oe_del <= 1'b0;
        else oe_del <= oe;

    tristate_buf tbuff(
        .oe(oe_del),
        .in(idata),
        .out(dout)
    );
endmodule

```

nie mają wyjścia trójszanowego aktywowanego sygnałem CS. Moduły *internal_ram* oraz *internal_rom* korzystają ze wspomnianych wcześniej modułów, wyposażając je w dodatkowy bufor trójszanowy aktywowany linią wyboru CS. Doświadczenia pokazały, że nextpnr bez większych problemów rozpoznaje i dokonuje syntezy pamięci ROM i RAM, umieszczając ją w BlockRAM bez potrzeby bezpośredniego odnoszenia się do komponentów specyficznych dla układu ICE40.

Dodatkowy dekodery

Główny dekodер adresowy ma tylko jedną linię CS, dla układów peryferyjnych konieczne jest użycie dodatkowego dekodera oznaczonego na schemacie jako IO decoder (**listing 4**) Jego zadaniem jest podział 256-bajtowej przestrzeni IO na 4 obszary po 64 bajty każdy. Blok zawiera 8-bitowy sygnał wejściowy, linię wejściową CS (pochodzącą z głównego dekodera) oraz 4 linie wyjściowe IO0...IO3 wyznaczające wybór układu w obszarze wejścia-wyjścia. Dekoder został zrealizowany w analogiczny sposób do głównego dekodera, z tym że nie ma tutaj konieczności rozpoznawania stanu nieustalonego.

Port GPIO

Do linii IO0 dekodera adresowego urządzeń dołączony został prosty wyjściowy Port GPIO mający 4 wyjścia (**listing 5**). Port ten został

Listing 3. Kod realizujący blok pamięci ROM

```

module internal_rom_no_tristate
#(parameter BITS=9)
(
    input clk,
    input [BITS-1:0] addr,
    output reg [7:0] data
);

localparam MEM_INIT_FILE="capp.hex";
reg [7:0] rom_memory[0:(2**BITS-1)];
initial begin
    if(MEM_INIT_FILE!="")
        $readmemh(MEM_INIT_FILE, rom_memory);
end
always @(posedge clk)
    data <= rom_memory[addr];
endmodule

/* ROM memory */
module internal_rom
#(parameter BITS=10)
(
    input clk,                // Clock input signal
    input rst,                // Enable signal
    input oe,                 // Address input
    input [BITS-1:0] addr,    // Data output
    output [7:0] data
);
wire [7:0] ntsdata;
internal_rom_no_tristate #(BITS(BITS)) rom(
    .clk(clk),
    .addr(addr),
    .data(ntsdata)
);
reg oe_del;
always @(posedge clk or negedge rst)
    if(~rst) oe_del <= 1'b0;
    else oe_del <= oe;

tristate_buf tbuf(
    .oe(oe_del),
    .in(ntsdata),
    .out(data)
);
endmodule

```

Listing 4. Dodatkowy dekodery adresowy IO decoder

```

module device_address_decoder(
    input [7:0] addr_in,
    input periph_cs,
    output periph1,
    output periph2,
    output periph3,
    output periph4
);
    reg [3:0] decoder;
    wire [2:0] addr = { periph_cs, addr_in[7:6] };
    always @ (addr) begin
        (* full_case,parallel_case *)
        casex(addr)
            3'b100: decoder = 4'b0001;
            3'b101: decoder = 4'b0010;
            3'b110: decoder = 4'b0100;
            3'b111: decoder = 4'b1000;
            default: decoder = 4'b0000;
        endcase
    end
    assign periph1 = decoder[0];
    assign periph2 = decoder[1];
    assign periph3 = decoder[2];
    assign periph4 = decoder[3];
endmodule

```

zrealizowany w najprostszy sposób i składa się z 4 przerzutników D z linią aktywacji układu CS. Linie D przerzutników dołączone są do 4 najmłodszych linii magistrali danych DI. Linie wyjściowe portu GPIO zmapowane są na 4 diody LED na płytce ewaluacyjnej ICE-CORE. Zatem zmieniając stan bitów w rejestrze GPIO, możemy sterować diodami LED.

Port szeregowy USART

Drugim urządzeniem naszego eksperymentalnego mikrokontrolera jest dwukierunkowy port szeregowy USART z 16-bajtowym buforem FIFO,

Listing 5. Kod prostego wyjściowego portu GPIO

```

module output_gpio(
    input clk,           // Clock
    input rst,           // Reset
    input [7:0] di,       // Data input
    input oe,            // Output enable
    input we,            // Write enable
    output reg [3:0] port //Output port
);

always @(posedge clk or negedge rst) begin
    if( ~rst ) begin
        port <= 4'd0;
    end
    else begin
        if( oe && we )
            port <= di[3:0];
        end
    end
end
endmodule

```

dostosowany do magistrali mikroprocesora 65C02 (**listing 6**). Interfejs modułu *tiny_usart()* jest zgodny z magistralą 65C02 oraz ma dodatkowo linię adresową wyboru układu CS, która jest dołączona do przestrzeni adresowej IO2 drugiego dekodera adresowego. Wyjście modułu stanowią dwie linie portu szeregowego TXD oraz RXD. Od strony programowej do dyspozycji mamy 3 rejestry. Rejestr danych DR, do którego należy wpisać daną, jaką chcemy wysłać lub odczytać daną odebraną. Rejestr statusu SR, który zawiera dwa bity: bit 0 (TXE), który mówi nam, że bufor nadawczy nadajnika jest pusty; bit 1 (RXNE) mówiący o tym, że w buforze odbiornika został odebrany znak i możemy go odczytać z rejestru danych. Rejestr kontrolny CR zawiera tylko jeden bit 0 (STX), ustawienie tego bitu powoduje rozpoczęcie transmisji.

Do realizacji zintegrowanego portu szeregowego zastosowano moduł *usart_tx*, który odpowiedzialny jest za realizację nadajnika oraz *usart_rx* zawierający implementację odbiornika. Moduł *usart_tx_fifo* zawiera magistralę danych oznaczoną jako data, oraz sygnał kontrolny valid, który należy ustawić, aby rozpocząć transmisję. W odpowiedzi na rozpoczęcie transmisji sygnał ready jest ustawiany po zakończeniu nadawania.

Analogiczny interfejs ma moduł odbiornika *usart_rx_fifo* oraz dodatkowo sygnał overflow oznaczający przepełnienie wewnętrznej kolejki FIFO. Sercem portu szeregowego jest maszyna stanów odpowiedzialna za generowanie sygnałów dla modułu nadajnika i odbiornika na podstawie wartości wpisywanych do rejestrów przez procesor 65C02.

Prędkość portu szeregowego została ustalona na stałą wartość 57600 bitów na sekundę i może być zmieniana na etapie syntezy poprzez zmianę współczynników dzielnika fraktalnego, które są parametryzowane za pomocą parametrów SPEED_KDIV oraz SPEED_NDIV. Port szeregowy, w razie potrzeby, może być łatwo rozbudowany o linię IRQ służącą do zgłaszania przerw, jednak z uwagi na eksperymentalny charakter projektu została ona pominięta.

Przykładowy program

Celem demonstracji możliwości mikrokontrolera przygotowano prosty program, którego zadaniem jest sprawdzenie poprawności działania rdzenia oraz jego skromnych peryferiów. Program został napisany w języku C i jest kompilowany za pomocą otwartoźródłowego kompilatora dla cc65 dla rodziny mikroprocesorów 6502.

Działanie programu polega na użyciu portu szeregowego jako konsoli szeregowej służącej do sterownia czterema diodami LED dołączonymi do portu GPIO układu. Pętla główna programu wygląda jak na **listingu 7**. Na początku program wysyła na konsolę szeregową komunikat zachęcający do wpisania na klawiaturze znaku z przedziału 1...4, a następnie oczekuje na odebranie pojedynczego znaku przez układ UART. Po odebraniu znaku sprawdzany jest dozwolony zakres znaków. Jeśli nie mieści się on w określonym przedziale, wówczas odsyłany jest komunikat błędny. W przypadku gdy znak mieści się w przedziale 1...4, stan odpowiadający mu diody LED zmieniający jest na przeciwny. Znak 1 będzie odpowiedzialny za zmianę stanu diody dołączonej do linii 49 i dalej analogicznie: linia 52, linia 55 i linia 56.

Funkcja *led_control()* jest odpowiedzialna za włączenie lub wyłączenie poszczególnych diod LED dołączonych do portu GPIO i sprowadza

Listing 6. Kod dwukierunkowego portu szeregowego USART z 16-bajtowym buforem FIFO

```

module tiny_usart(
    input clk,
    input rst,
    output tx_pin,
    input rx_pin,
    input [2:0] addr,
    input cs,
    input we,
    input [7:0] di,
    output [7:0] do
);
    reg utx_valid;
    wire utx_ready;
    reg [7:0] tx_dr;
    reg [7:0] rx_dr_rdy;
    reg [7:0] idata;
    wire [7:0] urx_bus;
    wire urx_valid;
    reg urx_ready;
    wire urx_overflow;

    localparam DR_ADDR = 3'd0;
    localparam SR_ADDR = 3'd1;
    localparam CR_ADDR = 3'd2;

    always @(posedge clk or negedge rst)
        if(~rst) begin
            tx_dr <= 8'd0;
            utx_valid <= 1'b0;
            urx_ready <= 1'b0;
            rx_dr_rdy <= 1'b0;
        end
        else begin
            if(cs && we) begin
                if(addr==DR_ADDR) begin
                    tx_dr <= di;
                end
                else if(addr==CR_ADDR) begin
                    utx_valid <= di[0];
                end
            end
            else if(cs && !we) begin
                if(addr==SR_ADDR) begin
                    idata <= { 5'd0, urx_overflow, rx_dr_rdy, utx_ready };
                end
                else if(addr==CR_ADDR) begin
                    idata <= { 7'd0, utx_valid };
                end
                else if(addr==DR_ADDR) begin
                    idata <= urx_bus;
                    rx_dr_rdy <= 1'b0;
                end
            end
            if(urx_valid) begin
                urx_ready <= 1'b1;
                rx_dr_rdy <= 1'b1;
            end
            if(utx_valid) utx_valid <= 1'b0;
            if(urx_ready) urx_ready <= 1'b0;
        end

    // Transmit component
    usart_tx_fifo #(.SPEED_NDIV(13)) utx(
        .clk(clk), .rst(rst),
        .valid(utx_valid), .ready(utx_ready),
        .data(tx_dr), .u_tx_pin(tx_pin)
    );

    // Receive component
    usart_rx_fifo #(.SPEED_NDIV(13)) urx( .clk(clk),
        .rst(rst),
        .valid(urx_valid), .ready(urx_ready),
        .overflow(urx_overflow), .data(urx_bus),
        .u_rx_pin(rx_pin)
    );

    reg cs_del;
    always @(posedge clk or negedge rst)
        if(~rst) cs_del <= 1'b0;
        else cs_del <= cs;

    tristate_buf tbuf(
        .oe(cs_del),
        .in(idata),
        .out(do)
    );
endmodule

```

się do wpisania stanu bitów do rejestru portu GPIO znajdującego się pod adresem bazowym 0x400 (**listing 8**).

Wysłanie łańcucha tekstowego na konsolę szeregową polega na cyklicznym wywołaniu funkcji *putchar()*, która przesyła kolejne znaki do bufora portu szeregowego (**listing 9**). Wysłanie pojedynczego znaku polega na sprawdzeniu w rejestrze statusu, czy bufor nadajnika jest wolny. W przypadku spełnienia tego warunku do rejestru DR wpisywany jest kod znaku, który chcemy przesłać, a w rejestrze kontrolnym CR ustawiany jest bit STX, co powoduje wpisanie kodu znaku do wewnętrznej kolejki FIFO portu szeregowego.

Za odbiór znaku z portu szeregowego odpowiada funkcja *getchar()* (**listing 10**). W aktywnej pętli sprawdza, czy bit RXNE w rejestrze SR

został ustawiony na 1, co oznacza, że w rejestrze DR znajduje się nowy znak do odczytania.

Aby program działał prawidłowo należy umieścić go w odpowiednim miejscu pamięci, do czego będziemy potrzebować skryptu linkera. Będziemy również potrzebować plik startowy *crt0.S*, który jest odpowiedzialny za ustawienie pamięci zgodnie z wymaganiami języka ANSI C. Mapę pamięci zawartą w pliku linkera pokazuje fragment skryptu:

```
MEMORY {
  ZP: start = $0, size = $100, type = rw, define =
  yes;
  RAM: start = $200, size = $0600, define = yes;
  ROM: start = $E000, size = $2000, file = %0;
}
```

Obszar **ZP** jest obszarem strony zerowej stanowiącym fragment pamięci RAM i rozpoczyna się od adresu 0. Dalszy obszar pamięci RAM kolejnych stron zawarty jest w sekcji RAM i rozpoczyna się od adresu 0x200. Obszar pamięci ROM rozciąga się od adresu 0xE000 aż do końca obszaru adresowego 0xFFFF, co oznacza, że wektory przerwań znajdować się będą bezpośrednio w pamięci ROM.

Struktura projektu oraz uruchomienie

Jedną z zalet otwartych narzędzi syntezy układów programowalnych jest możliwość wywołania ich z wiersza polecenia w dowolny sposób, bez konieczności używania zintegrowanych środowisk. Do zautomatyzowania procesu syntezy z pomocą przychodzą nam klasyczne narzędzia do automatyzacji budowania oprogramowania. W naszym projekcie użyto narzędzia *gnu make*, które przetwarza plik reguł *Makefile* i na tej podstawie stwierdza, które pliki źródłowe wymagają syntezy/kompilacji. Aby zbudować projekt, do dyspozycji mamy kilka głównych reguł zdefiniowanych w pliku *Makefile* realizujących następujące czynności:

- **make all** – kompletne zbudowanie projektu, które polega na kompilacji programu za pomocą kompilatora CC65 oraz syntezie projektu mikrokontrolera. Do syntezy wykorzystywane jest narzędzie

nextpnr-ice40, które na podstawie plików źródłowych Verilog i pliku *bin* wygenerowanego przez kompilator tworzy plik *retromcu.bin*, który stanowi plik wynikowy przeznaczony do konfiguracji układu ICE40;

- **make clean** – wyczyszczenie wszystkich plików pośrednich oraz wynikowych;
- **make sim** – dokonuje syntezy oraz symulacji projektu z wykorzystaniem pakietu *icarus-verilog*. W wyniku tego powstają pliki wynikowe LXT, które następnie możemy załadować do narzędzia służącego do wyświetlania wykresów GTK Wave;

Jeśli chodzi o strukturę projektu, to została ona podzielona na katalogi odpowiadające poszczególnym modułom projektu:

- *capp* – pliki źródłowe aplikacji w języku C i ASM, które następnie mogą być kompilowane za pomocą kompilatora CC65;
- *rtl* – pliki źródłowe Verilog projektu mikrokontrolera;
- *sim* – pliki testbench służące do weryfikacji poprawności działania plików źródłowych. Główne jednostki testowe powinny mieć nazwę kończącą się sufiksem *_top.v*. Dla wszystkich plików *top* generowane są pliki stanowiące wynik symulacji w postaci przebiegów czasowych dla programu GTK Wave.

Aby uruchomić projekt, należy dysponować skonfigurowanym środowiskiem, które zostało opisane w pierwszym odcinku cyklu, a ponadto należy doinstalować kompilator dla mikroprocesora cc65. W przypadku popularnych dystrybucji Linuksa można to zrobić za pomocą polecenia:

```
sudo apt install cc65
```

Jeśli dysponujemy systemem OS-X, możemy skorzystać z HomeBrew:

```
brew install cc65
```

Jeśli mamy gotowy zestaw narzędzi, w kolejnym kroku należy dokonać kompilacji oraz syntezy projektu, co możemy zrealizować za pomocą polecenia: **make**, czego rezultatem będzie powstanie pliku *retromcu.bin*.

Zanim uruchomimy mikrokontroler, do linii P32 stanowiącej wyjście TXD zestawu iCE-Core należy dołączyć wejście RXD przejściówki RS232 ↔ USB, natomiast do linii P33 (RXD) wyjście RXD przejściówki RS232 ↔ USB. Należy również uruchomić program terminalowy na wirtualnym porcie USB z następującymi parametrami transmisji: 57600,N,8,1. Konfiguracji zestawu dokonujemy poprzez przesłanie do wirtualnego portu szeregowego */dev/ttyACM0* pliku konfiguracyjnego za pomocą polecenia:

```
cat retromcu.bin > /dev/ttyACM0
```

Po przesłaniu pliku, w zestawie ICE-CORE powinna zaświecić się dioda CONF. W tym momencie przy wysyłaniu w programie terminalowym kodów klawiszy 1...4 powinny się naprzemiennie włączać diody LD1...LD4 przypisane poszczególnym klawiszom, a na konsolę szeregową będzie wysyłana informacja o włączeniu lub wyłączeniu wybranej diody.

Zakończenie

Zaprezentowany projekt zawiera jedynie podstawowe układy peryferyjne takie jak: GPIO, UART. W praktycznych zastosowaniach należałoby go wyposażać w dodatkowe peryferia np. układy czasowo-licznikowe oraz dodatkowy kontroler przerwań. Celem projektu nie było opracowanie praktycznego mikrokontrolera, a sprawdzenie, czy otwarte narzędzia programistyczne pozwalają na syntezę rozbudowanych projektów bazujących na syntezyzowanych mikroprocesorach. W praktycznych zastosowaniach lepszym rozwiązaniem będzie skorzystanie z bardziej wydajnego 32-bitowego rdzenia RISC-V zamiast z trzydziestoletniego CPU. Praktyczne doświadczenia pokazały, że *nextpnr-ice40* bez problemu poradził sobie z tego rodzaju zaawansowanym projektem i udało się uzyskać w pełni działający mikroprocesor z częstotliwością pracy 6,25 MHz.

Lucjan Bryndza, EP

lucjan.bryndza@ep.com.pl

Materiały dodatkowe:

- EP 2/21: FPGA a Open Source. Płytką prototypowa IceCore z układem Lattice ICE40, <https://bit.ly/3oP5oXM>
- EP 9/20: Przegląd otwartych narzędzi dla układów FPGA, <https://bit.ly/35XvZJT>

Listing 7. Pętla główna programu testowego

```
int main() {
  unsigned char ch, lds;
  puts("\r\nHello from 65c02 LED test\r\n");
  puts("Press 1-4 for led toggle\r\n");
  led_control(0xff);
  for(lds=0;;) {
    ch = getchar();
    if(ch < '1' || ch > '4') {
      puts("Error: Unknown key only 1-4 is allowed");
    } else {
      ch -= '0';
      lds ^= 1 << (ch-1);
      led_control(~lds);
      puts("Led: "); tohex(ch); puts(" toggled.\r\n");
    }
  }
  return 0;
}
```

Listing 8. Funkcja led_control()

```
#define LED_REG *(volatile unsigned char*)(0x0400)
static void led_control(unsigned char led) {
  LED_REG = led;
}
```

Listing 9. Funkcja putchar()

```
#define USART_DR_REG *(volatile unsigned char*)(0x0440)
#define USART_SR_REG *(volatile unsigned char*)(0x0441)
#define USART_CR_REG *(volatile unsigned char*)(0x0442)
#define USART_SR_TXE 0x01
#define USART_SR_RXNE 0x02
#define USART_CR_STX 0x01

static void putch(char ch) {
  while(!(USART_SR_REG & USART_SR_TXE) ) {}
  USART_DR_REG = ch;
  USART_CR_REG |= USART_CR_STX;
}
```

Listing 10. Funkcja getchar()

```
static unsigned char getchar(void) {
  while( !(USART_SR_REG & USART_SR_RXNE) ) {}
  return USART_DR_REG;
}
```



Nie tylko oscylloskopy USB

Aparatura pomiarowa stanowi absolutny fundament pracy biura projektowego, serwisu elektroniki czy laboratorium branżowego, a możliwość zaufania wynikom prowadzonych pomiarów jest kluczowa dla każdego, kto poważnie traktuje parametry tworzonych lub testowanych urządzeń. W ofertach producentów wyposażenia pomiarowego ze wszystkich półek cenowych, począwszy od niedrogich urządzeń dla amatorów, a skończywszy na wysokiej klasy aparaturze do zastosowań laboratoryjnych i naukowo-badawczych, można spotkać modele przeznaczone wyłącznie do współpracy z komputerem, nieposiadające klasycznego interfejsu użytkownika wbudowanego na panelu czołowym. Czy warto inwestować w tego typu aparaturę i o czym należy pamiętać, wybierając konkretny model urządzenia?

Zdążyliśmy już przyzwyczaić się do klasycznej budowy oscyloskopów cyfrowych, wyposażonych w wyświetlacz (początkowo niewielki, o proporcjach 4:3, ostatnio zaś coraz częściej duży i w dodatku panoramiczny) oraz zestaw typowych kontrolerów w postaci pokręteł czy przycisków fizycznych. Większość producentów nadal wprowadza do oferty także urządzenia w formie przystawek do komputerów PC i laptopów. Co więcej, choć wielu elektronikom oscylloskopy USB słusznie kojarzyły się głównie z tanimi modelami o próbkowaniu na poziomie kilku...kilkudziesięciu Msa/s, dziś w sprzedaży dostępne są modele mogące śmiało konkurować z topowymi oscyloskopami stacjonarnymi, zaś w ślad za nimi idą inne rodzaje urządzeń laboratoryjnych, także wykorzystujące oprogramowanie komputerowe jako interfejs użytkownika. Zanim przejdziemy do przedstawienia wybranych produktów z oferty czołowych przedstawicieli rynku aparatury pomiarowej, podsumujemy najważniejsze cechy urządzeń pomiarowych USB.

Zalety i wady przystawek oscyloskopowych

- (+) Kompaktowa forma i mobilność** – podstawową zaletą, jaką wymienia się w przypadku urządzeń pomiarowych w postaci przystawki USB, są niewielkie wymiary obudowy i znikoma

(w porównaniu z urządzeniami stacjonarnymi) masa. Dzięki temu aparatura USB doskonale nadaje się do zastosowań polowych, jako przyrządy przenośne współpracujące z laptopem. Kompaktowa forma tego typu sprzętów doskonale sprawdza się także w niewielkich pracowniach, w których postawienie kilku konwencjonalnych urządzeń odczuwalnie zmniejszyłoby ilość dostępnego miejsca;

- (+) Bezpośredni zrzut danych do PC** – kolejną ważną zaletą przystawek oscyloskopowych i innych tego typu urządzeń jest łatwość magazynowania wyników pomiarów (np. zrzutów ekranu i logów pomiarowych) bezpośrednio na dysku twardym komputera, co stanowi znaczne ułatwienie w porównaniu do żmudnego przetrzymywania danych z przyrządu za pośrednictwem pamięci typu pendrive. Choć zaleta ta może wydawać się mało istotna w codziennych pomiarach prowadzonych na żywo, to dla osób magazynujących wyniki pomiarów do celów archiwizacji bądź dalszej obróbki numerycznej będzie ona często priorytetem;
- (+) Atrakcyjna cena** – brak ekranu i kontrolerów interfejsu użytkownika znacząco redukuje koszty produkcji sprzętów, co w większości przypadków pozwala nabywcom zaoszczędzić spore środki finansowe – ma to znaczenie zwłaszcza w przypadku firm, które wyposażając pracownię pomiarową, muszą zainwestować w obszerniejszy zestaw aparatury. Choć ceny urządzeń pomiarowych na USB często nadal bywają wysokie, to zwykle zakup urządzenia stacjonarnego porównywalnego pod względem parametrów metrologicznych wiązałby się ze znacznie wyższym wydatkiem;
- (+) Możliwość galwanicznego odłączenia od sieci energetycznej** – niektóre modele oscyloskopów w formie przystawek komputerowych są zasilane z portu USB. Daje to dodatkową możliwość, o której użytkownicy tych urządzeń często zapominają – podłączenie oscyloskopu do laptopa, mającego własne, wewnętrzne źródło zasilania (akumulator) i niepołączonego z siecią energetyczną oraz zewnętrznymi urządzeniami (np. monitorami) pozwala na prowadzenie pomiarów w całkowitym odcięciu od potencjałów sieci. Taki „pływający” system pomiarowy może doskonale ułatwić pracę, dając pewną swobodę w podłączaniu masy oscyloskopu do nietypowych punktów badanego układu;
- (-) Wygoda obsługi** – chyba najistotniejszą wadą większości przyrządów obsługiwanych jedynie poprzez oprogramowanie

komputerowe jest pewna toporność, widoczna w działaniu wielu pakietów kontrolnych rozwijanych przez różnych producentów. Dla elektroników przyzwyczajonych do fizycznych pokręteł i przycisków (pozwalających na szybką oraz intuicyjną regulację podstawy czasu, wzmocnienia kanałów wejściowych itd.), przejście na kontrolki klikane na ekranie komputera stanowi zbyt dużą przeszkodę, znacznie utrudniającą sprawną obsługę takiego przyrządu. Co ciekawe, problem ten dotyczy w równym stopniu także stacjonarnej aparatury pomiarowej, sterowanej za pośrednictwem programu pracującego (zwykle) pod kontrolą systemu Windows bądź macOS. Zaskakujące jest to, że często nawet urządzenia wysokiej klasy zdradzają znaczne niedostatków pod względem użyteczności przeznaczonego dla nich oprogramowania;

- **(-) Ograniczona przepustowość** – konieczność przesyłania danych pomiarowych poprzez łącze USB zdecydowanie nie pomaga w uzyskaniu użytecznej prędkości odświeżania ekranu czy też w obsłudze dużych zestawów danych, płynących z wewnętrznej pamięci próbek (bufora) urządzenia. Dodatkowym ograniczeniem, praktycznie niezależnym od najlepszej nawet woli producentów aparatury, jest fakt, że większości systemów operacyjnych komputerów (np. Windows) daleko do miana systemów czasu rzeczywistego. Wielowątkowość i nieunikniony podział czasu procesora na obsługę procesów systemowych oraz wszelkich innych aplikacji uruchomionych w czasie działania aplikacji pomiarowej sprawiają, że od oprogramowania do obsługi urządzeń pomiarowych nie sposób wymagać pełnej responsywności, jakiej można spodziewać się po szybkich systemach wbudowanych stosowanych w urządzeniach mających własne, silnie zoptymalizowane HMI (human-machine interface). Z tego też względu producenci prześcigają się w implementacji customowych rozwiązań maksymalnie odciążających moc obliczeniową komputera i przepustowość łącza (USB 3.0 lub w tańszych modelach USB 2.0), co ma na celu optymalizację wydajności pomiarowej i przyspieszenie akwizycji dla bardziej wymagających aplikacji;
- **(-) Uzależnienie od dostępu do komputera** – kolejnym aspektem, który może przemawiać za odstąpieniem od zakupu przystawki oscyloskopowej na rzecz zintegrowanego skopometru lub niewielkiego oscyloskopu biurkowego, jest konieczność zapewnienia stałego dostępu do komputera, gdyż bez niego przyrząd sam w sobie nie dostarcza żadnych możliwości obsługi. Ma to znaczenie w przypadkach, gdy przyrząd ma służyć do prac wyjazdowych, a z jakichś względów konieczność przewożenia dodatkowego komputera przez serwisanta byłaby sporym utrudnieniem (np. pomiary w trudnych warunkach środowiska, które nie stanowią większego problemu dla niektórych modeli skopometrów, wykraczają zwykle poza zalecany zakres parametrów środowiska pracy większości przystawek USB);
- **(-) Ryzyko uszkodzenia komputera i łatwiejsze przenoszenie zakłóceń** – do wad urządzeń pomiarowych opartych na USB należy pewne ryzyko, które niesie podłączanie badanych układów (poprzez oscyloskop) do portu komputera. Najprostszym przykładem jest omyłkowe zwarcie masy sondy pomiarowej do napięcia zasilania. W pewnych przypadkach może to spowodować przepływ prądu wyrównawczego o dużej wartości, który stanowi zagrożenie dla obwodów komputera. Najlepszym rozwiązaniem jest zastosowanie komputera przenośnego odłączonego całkowicie od sieci energetycznej (jak opisaliśmy wcześniej) bądź zastosowanie oscyloskopu z wbudowanym izolatorem galwanicznym portu USB (lub systemem ochrony masy – więcej na ten temat w opisie produktów marki DreamSourceLab).



Fotografia 1. Oscyloskop z serii P924xA marki Keysight

Przegląd ofert producentów – oscyloskopy USB

W tej części artykułu przyjrzymy się propozycjom urządzeń z fabryki największych potentatów na rynku aparatury pomiarowej. Nie zabraknie także budżetowych oscyloskopów i innych urządzeń pomiarowych, przeznaczonych dla przedsiębiorstw i hobbystów o mniejszym zasobie gotówki.

Keysight

Firma Keysight, która przez wiele lat sprzedawała swoje produkty pod kultową marką Agilent, oferuje jedno z najszerzych portfolio w zakresie urządzeń opartych na szybkim interfejsie USB. Oscyloskopy z serii P924xA (fotografia 1) bazują na 1-gigahercowej technologii InfiniiVision, znanej z topowych modeli oscyloskopów stacjonarnych, które stanowią trzon ofertowy spadkobiercy pomiarowego działu Hewlett-Packard. Seria zawiera modele:

- P9241A – pasmo 200 MHz,
- P9242A – pasmo 500 MHz,
- P9243A – pasmo 1 GHz.

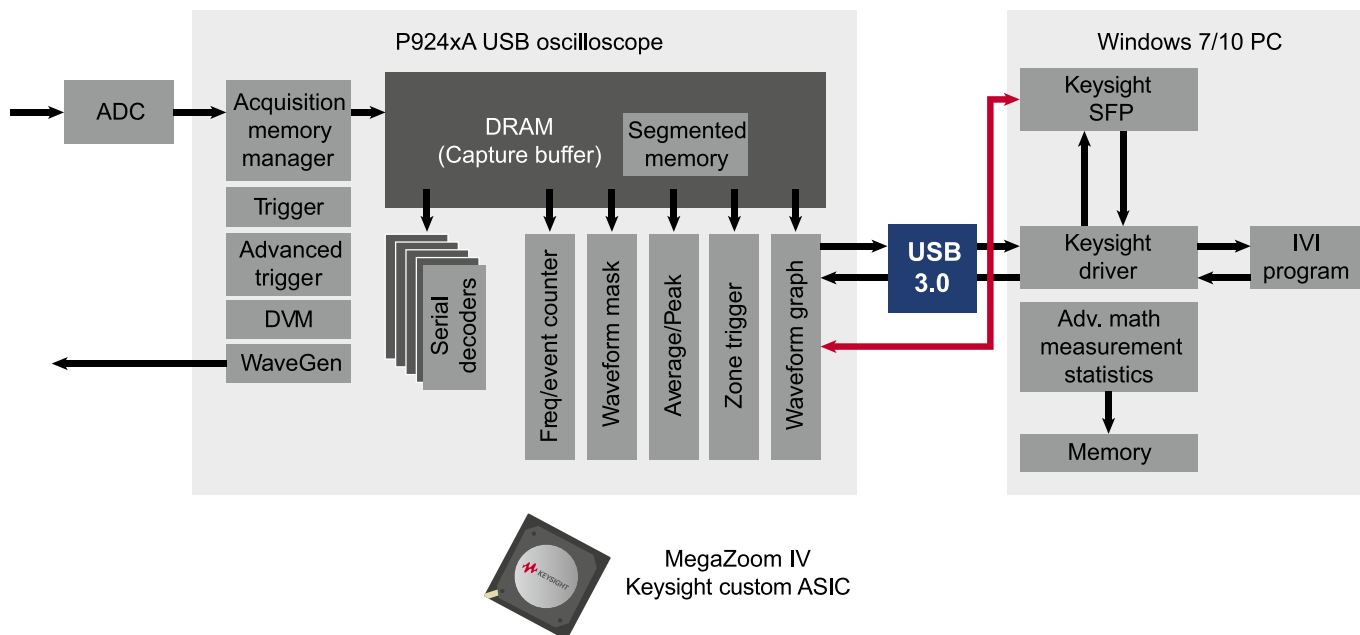
Urządzenia zostały skonstruowane w oparciu na opracowanym przez inżynierów firmy Keysight układzie ASIC typu MegaZoom IV, którego rdzeń zapewnia sprzętowe wsparcie dekodowania szeregowych interfejsów komunikacyjnych, obsługę masek sygnałowych, zaawansowanych trybów wyzwalania, czy wreszcie... wbudowanego generatora arbitralnego WaveGen o paśmie 20 MHz (rysunek 1). Producent nieprzypadkowo chwali się silną integracją funkcji pomiarowych na poziomie sprzętowym – taka architektura umożliwia bowiem efektywne odciążenie łącza USB 3.0 i procesora komputera nadrzędnego (jak również sterowników systemowych), dzięki czemu urządzenia z serii P924xA są w stanie obsługiwać częstotliwość akwizycji na poziomie aż miliona ramek na sekundę(!).

Opisywane oscyloskopy udostępniają prędkość próbkowania na poziomie 5 GSa/s (współdzieloną dla obydwu kanałów, zatem w trybie 2-kanałowym próbkowanie spada do 2,5 GSa/s, co ma zresztą miejsce w przeważającej większości klasycznych oscyloskopów cyfrowych). Wszystkie trzy modele z serii P924xA mają dwa kanały wejściowe o wybieranej impedancji (1 MΩ lub 50 Ω) i oferują pamięć próbek o sporej (choć wcale nie zawrotnej, jeśli spojrzeć na konkurencyjne oscyloskopy innych marek, np. Rigol) pojemności równej 4 Mpts.

Firma Keysight wprowadziła na rynek także modułowe oscyloskopy z serii U270xA: U2701A o paśmie 100 MHz i U2702A o paśmie 200 MHz (fotografia 2). Obydwa modele oferują dość standardową częstotliwość próbkowania (1 GSa/s, współdzielona pomiędzy obydwoma kanałami) i rozdzielczość przetworników ADC (8 bitów),



Fotografia 2. Oscyloskop modułowy z serii U270xA marki Keysight



Rysunek 1. Schemat funkcjonalny systemu pomiarowego opartego na oscyloskopie z serii P924xA

przewyższają za to wiele konkurencyjnych oscyloskopów sporą pamięcią próbek, która (dla pojedynczego kanału) wynosi 16 Mpts w trybie normalnym i 32 Mpts w trybie Single-shot. Urządzenia mogą pracować jako niezależne jednostki pomiarowe, zapewniają ponadto pełną kompatybilność z bazą U2781A, stanowiącą chassis systemu akwizycji danych i wspierającą podłączenie nawet do 6 modułów rozszerzeń (fotografia 3). Seria urządzeń kompatybilnych z chassis jest zresztą dużo obszerniejsza i obejmuje, oprócz samych oscyloskopów, także wielokanałowe moduły akwizycji danych, oznaczone numerami katalogowymi U2531A, U2541A oraz U2542A, wielofunkcyjne moduły DAQ U235xA czy też macierze szybkich przełączników sygnałowych U2751A.

Rigol

Najnowsza propozycja firmy Rigol – seria DS8000-R, obejmuje trzy modele wysokiej klasy oscyloskopów modułowych, które.... skutecznie wymykają się przedstawionemu wcześniej podziałowi na urządzenia pomiarowe ze zintegrowanym interfejsem HMI oraz urządzenia w formie przystawek. Rozwiązanie znanej chińskiej marki stoi dokładnie pośrodku, łącząc w sobie zalety obu tych typów przyrządów. Niskoprofilowa obudowa do montażu w szafach typu rack (fotografia 4) pozwala na zastosowanie w wielokanałowych systemach akwizycji danych (DAQ), zaś parametry sygnałowe zapewniają przepustowość nieustępującą dobrym oscyloskopom biurkowym.



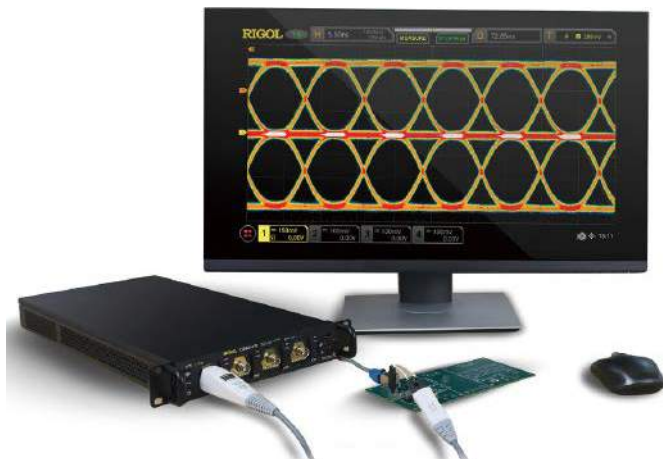
Fotografia 4. Oscyloskop modułowy z serii Rigol DS8000-R

Najniższy model DS8034-R oferuje prędkość próbkowania równą 5 GSa/s, zaś dwie pozostałe wersje aż 10 GSa/s. Maksymalna szybkość odświeżania to ponad 600 tysięcy ramek/sekundę, a pamięć próbek ma potężną pojemność aż 500 Mpts(!).

Oscyloskop nie ma żadnych zewnętrznych elementów obsługowych poza przyciskiem zasilania, klawiszami Force i Run/Stop. Co zatem stawia go w środku ww. podziału? W rzeczywistości urządzenia z serii DS8000-R mają wbudowany komputer, pracujący pod kontrolą systemu Linux i udostępniający pełną funkcjonalność, umożliwiającą pracę w roli samodzielnego urządzenia pomiarowego (tj. bez zastosowania jakichkolwiek chassis). Wystarczy podłączyć zewnętrzny monitor oraz.... mysz i klawiaturę (fotografia 5). Urządzenie może jednak z powodzeniem działać również jako przystawka do komputera PC,



Fotografia 3. Moduł chassis U2781A marki Keysight



Fotografia 5. Oscyloskop Rigol z serii DS8000-R jako samodzielne urządzenie pomiarowe (host)



Fotografia 6. Przykład zastosowania oscyloskopów Rigol z serii DS8000-R jako modułów wejściowych superszybkiego systemu akwizycji DAQ

choć w przypadku tak zaawansowanego urządzenia port USB device wydaje się zaledwie mało istotnym dodatkiem. Producent kusi najbardziej zaawansowanych odbiorców (w tym jednostki badawcze) możliwością rozbudowy systemu akwizycji danych do niewiarygodnej wprost liczby 512 kanałów (fotografia 6).

Pico Technology

Oscyloskopy z serii PicoScope można z powodzeniem traktować jako protoplastów rynku aparatury pomiarowej z portem USB. Dziś w portfolio marki znajdziemy aż sześć serii produktów, obejmujących zarówno podstawowe modele do ogólnych zastosowań, jak i specjalistyczne wersje o próbkowaniu w czasie ekwiwalentnym, którego częstotliwość jest określana już nie w giga-, lecz terahercach:

- **PicoScope 2000** – pasmo od 10 MHz do 100 MHz, 2 lub 4 kanały analogowe (opcjonalnie wersja MSO z 16 kanałami analizatora logicznego), próbkowanie do 1 GSa/s, pamięć wbudowana do 128 MSa (zależnie od modelu);
- **PicoScope 3000** – pasmo od 50 MHz do 200 MHz, 2 lub 4 kanały analogowe (opcjonalnie wersja MSO), próbkowanie 1 GSa/s, pamięć wbudowana do 512 MSa;
- **PicoScope 4000** – pasmo od 5 MHz do 20 MHz, 2, 4 lub 8 kanałów analogowych z rozdzielczością natywną 12 bitów, próbkowanie do 80 MSa/s, pamięć wbudowana do 256 MSa;
- **PicoScope 5000** – pasmo od 60 MHz do 200 MHz, 2 lub 4 kanały analogowe z rozdzielczością natywną od 8 do 16 bitów, opcja MSO z 16 kanałami cyfrowymi, próbkowanie do 1 GSa/s, pamięć wbudowana do 512 MSa;
- **PicoScope 6000** – pasmo od 300 MHz do 1 GHz, 4 lub 8 kanałów analogowych + 16 cyfrowych (opcja), próbkowanie 5 GSa/s, pamięć od 1 do 4 GSa, rozdzielczość ADC ustawiana na 8 lub 12 bitów (technologia FlexRes) (fotografia 7);



Fotografia 7. Oscyloskop PicoScope z serii 6000 z ośmioma kanałami analogowymi i opcją MSO



Fotografia 8. Oscyloskopy z serii PicoScope 9000

- **PicoScope 9000** – na tę serię składają się dwie podgrupy urządzeń, diametralnie różniące się parametrami i zakresem zastosowań:
 - Seria PicoScope 9300 zapewnia 16-bitową kwantyzację i próbkowanie w czasie rzeczywistym 1 MSa/s (ADC o rozdzielczości 16 bitów), ale za to tryb próbkowania sekwencyjnego dochodzi do częstotliwości efektywnej na poziomie nawet 15 TSa/s. Cała linia produktów obejmuje 9 modeli, z czego jeden (9321-20) ma wbudowany front-end fotoelektryczny, pozwalający na prowadzenie szybkich pomiarów bezpośrednio w dziedzinie optycznej (układ współpracuje z jedno- i wielomodowymi światłowodami, przesyłającymi sygnały w paśmie od 750 do 1650 nm, tj. we wszystkich trzech oknach transmisyjnych). Ponadto modele z serii 9311 (fotografia 8) mają wbudowane generatory impulsowe, służące do prowadzenia pomiarów reflektometrycznych (TDR/TDT);
 - Seria PicoScope 9400 zawiera cztery modele: 940x-05 – o paśmie 5 GHz (2 lub 4 kanały, próbkowanie w czasie ekwiwalentnym do 1 TSa/s) oraz 940x-16 (pasmo 16 GHz, 2 lub 4 kanały, próbkowanie w czasie ekwiwalentnym do 2,5 TSa/s). W urządzeniach zastosowano 12-bitowe przetworniki ADC o próbkowaniu 500 MSa/s, zaś do akwizycji sygnałów o szczególnie wysokiej częstotliwości (np. podczas pracy nad łączami Ethernet, HDMI czy PCI) mogą być wykorzystane specjalne sondy wysokoczęstotliwościowe (fotografia 9) z parzystymi pinami typu Pogo, umożliwiającymi bezstratne podłączenie masy sondy do masy badanego układu – rozwiązanie takie pozwala na zapewnienie integralności sygnałów niezbędnej



Fotografia 9. Sonda do badania sygnałów RF z serii PicoConnect 900



Fotografia 10. Oscyloskop cyfrowy USB marki DreamSourceLab

do badania urządzeń pracujących w paśmie rzędu 5 GHz, nieosiągalnej w przypadku klasycznych sond pasywnych z przewodem masowym.

DreamSourceLab

Aktualnie w ofercie DreamSourceLab znajdują się trzy modele oscyloskopów USB (fotografia 10) o paśmie od 50 do 100 MHz i próbkowaniu od 200 MSa/s do 1 GSa/s. Rozmiar pamięci próbek wynosi od 20 kSa do 2 MSa (w czasie rzeczywistym) oraz od 32 kSa do 256 MSa w trybie pojedynczej akwizycji. Firma wprowadziła ciekawe rozwiązanie sprzętowe, pozwalające na automatyczne rozłączenie mas oscyloskopu i komputera po przypadkowym zwarciu masy sondy do szyny zasilającej (rysunek 2).

Velleman

Belgijska firma Velleman, znana głównie z produkcji zestawów do samodzielnego montażu i drukarek 3D z serii Vertex, wprowadziła na rynek 5 modeli oscyloskopów przystawkowych:

- **PCSU200** – dwa kanały wejściowe do 12 MHz, pamięć próbek 4 kpts / kanał, dodatkowe wyjście generatora funkcyjnego do 1 MHz (sinus) lub 500 kHz (trójkąt/prostokąt);
- **PCSU1000** – dwa kanały analogowe 60 MHz + wejście wyzwalania zewnętrznego, próbkowanie do 50 MSa/s (lub sekwencyjne do 1 GSa/s), funkcja analizy widma do 25 MHz (FFT);



Fotografia 11. Oscyloskop USB PCSGU250 marki Velleman

- **PCSU02** – miniaturowa przystawka z wyprowadzonymi na stałe przewodami wejściowymi. Ma dwa kanały analogowe, pasmo do 500 kHz, próbkowanie do 2,5 MSa/s;
- **PCSU01** – model zbliżony do PCSU02, jeden kanał wejściowy o paśmie 200 kHz;
- **PCSGU250** – dwa kanały wejściowe do 12 MHz, próbkowanie do 25 MSa/s, wbudowany generator o próbkowaniu do 12,5 MHz (fotografia 11).

Multicomp

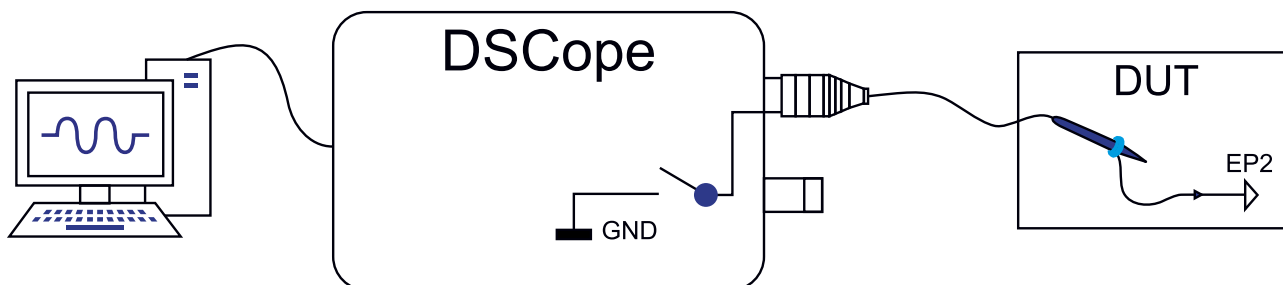
W ofercie firmy znajdują się cztery modele oscyloskopów USB (fotografia 12) o oznaczeniach:

- **MP720016 EU-UK** (2 kanały 25 MHz, do 40 Vpp, USB 2.0);
- **MP720017 EU-UK** (2 kanały 25 MHz, do 400 Vpp – izolowany interfejs USB 2.0);
- **MP720018 EU-UK** (2 kanały 60 MHz, do 40 Vpp, USB 2.0 + opcjonalny LAN);
- **MP720019 EU-UK** (2 kanały 100 MHz, do 40 Vpp), USB 2.0 + opcjonalny LAN).

Urządzenia oferują pamięć rekordu równą 5 kpts i próbkowanie 100 MSa/s (dwa najniższe modele) lub 10 Mpts i 1 GSa/s (kolejne wersje).

Owon

Marka Owon, także mająca swoją siedzibę w Chinach, zadebiutowała niezwykle tanimi oscyloskopami i innymi urządzeniami pomiarowymi, przeznaczonymi w rzeczywistości głównie do zastosowań amatorskich i dydaktycznych, a także podstawowych pomiarów



Rysunek 2. System ochrony masy, zastosowany w oscyloskopach DScope



Fotografia 12. Oscyloskop USB marki Multicomp (opis w tekście)

komercyjnych. Szybko jednak rozwinęła swoją ofertę o bardziej zaawansowane przyrządy, począwszy od oscyloskopów z próbkowaniem dochodzącym do 5 GSa/s, a skończywszy na analizatorach widma z pasmem pomiarowym do 7,5 GHz. W zakresie oscyloskopów wirtualnych (bo tak określa swoje przystawki USB firma Owon) producent oferuje trzy serie: VDS6000 (oscyloskopy 2-kanalowe, 1 GSa/s, pamięć 10 Mpts, pasmo 100 MHz oraz 4-kanalowe z pasmem 70 lub 100 MHz), a także VDS (modele 2- i 4-kanalowe o paśmie od 25 MHz do 100 MHz i próbkowaniu 100 MSa/s do 1 GSa/s). Model VDS1022I, choć najniższy pod względem metrologicznym (25 MHz, 100 MSa/s, 2 kanały) oferuje wbudowaną izolację portu USB, co daje możliwość prowadzenia pomiarów „pływających” (choć w materiałach producenta trudno jest znaleźć informację o parametrach bariery izolacyjnej, należy zatem traktować ją raczej jako sposób na przecięcie potencjalnych pętli masy niż jako środek ochrony elektrycznej...).

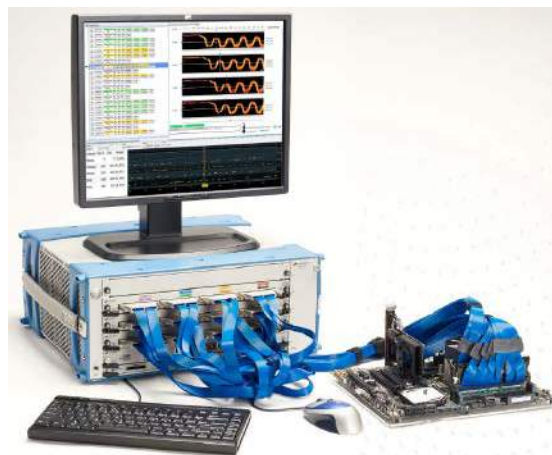
Hantek

Marka znana głównie z tanich oscyloskopów biurkowych i skopometrów, przeznaczonych głównie do zastosowań hobbystycznych, dydaktycznych i (zdaniem producenta) także do prowadzenia pomiarów w serwisach elektroniki samochodowej, wprowadziła na rynek cztery serie przystawek oscyloskopowych USB (pomijamy modele wycofane ze sprzedaży):

- **seria 1000** (aktualnie dostępny tylko jeden model 1008C) – 8 kanałów analogowych, ADC 12-bitowy, pamięć 4 kpts, próbkowanie 2,4 MSa/s;
- **seria 2000** – 2 kanały, pasmo od 40 do 100 MHz, próbkowanie 100...250 MSa/s;
- **seria 3000** – 2 lub 4 kanały, pasmo 60 MHz, próbkowanie 200 MSa/s, pamięć od 10 kpts do 16 Mpts;
- **seria 6000** – trzy modele 2-kanalowe o paśmie od 20 do 80 MHz, próbkowanie od 48 do 250 MSa/s.

Analizatory stanów logicznych

Przeglądając oferty producentów analizatorów stanów logicznych, można odnieść wrażenie, że dziś nikt nie korzysta już ze stacjonarnych urządzeń tego typu, których pierwsze modele pojawiły się na rynku aparatury pomiarowej kilkadziesiąt lat temu (niektórzy Czytelnicy być może pamiętają jeszcze „cacka” ówczesnej techniki, np. model Philips PM 3655 czy znacznie nowszy HP1670G). I choć nadal można spotkać w sprzedaży nowe analizatory wysokiej klasy, przeznaczone do najbardziej wymagających aplikacji (np. Keysight 16800), to zdecydowana większość rynku tych urządzeń podzielona została na dwie grupy: analizatory wbudowane w oscyloskopy MSO (stąd ich nazwa – mixed-signal oscilloscope) oraz wielokanałowe przystawki komputerowe. Do tej drugiej grupy należą zarówno proste, niedrogie urządzenia zapewniające możliwość badania systemów



Fotografia 13. Rozbudowany system analizy sygnałów logicznych, oparty na modułach Keysight U4164A

cyfrowych w liczbie do 8 kanałów jednocześnie, jak i znacznie bardziej rozbudowane modele o 16 i więcej kanałach cyfrowych, nie rzadko także wspieranych przez dodatkowe kanały analogowe. Takie rozwiązanie jest szczególnie cenne w badaniach urządzeń o sygnałach mieszanych, wyposażonych w przetworniki A/C i C/A, zaś obecność kanałów analogowych umożliwia traktowanie przystawek jako... miniatury oscyloskopów MSO (choć zwykle pasmo i częstotliwości próbkowania oraz zakres napięciowy kanałów wejściowych są silnie ograniczone względem parametrów oferowanych przez nawet najtańsze modele oscyloskopów biurkowych).

Wisienką na torcie będzie tu z pewnością system modułowy U4164A marki Keysight (fotografia 13), który może współpracować zarówno z zewnętrznym komputerem, jak i dedykowanym kontrolerem AXIe M9537A. Urządzenie umożliwia badanie sygnałów o przepływności 4 GB/s w 68 kanałach (tryb statyczny) oraz synchroniczną analizę danych (timing mode) do 10 GHz. System jest przeznaczony głównie do najbardziej wymagających aplikacji, np. bezpośredniego debugowania ruchu na szynach szybkich pamięci DDR4, DDR5, LPDDR4 i LPDDR5.

Większość obecnych na rynku analizatorów logicznych na USB ma jednak znacznie niższe parametry, a i tak stanowi nieocenioną pomoc dla projektantów oraz programistów systemów wbudowanych. Nawet najtańsze analizatory umożliwiają akwizycję i dekodowanie najpopularniejszych protokołów szeregowych, w tym rzecz jasna SPI, I²C, U(S) ART, a nawet 1-Wire, CAN, QSPI czy (w niektórych modelach nieco wyższej klasy) – nawet wolniejszych wersji interfejsu USB. Warto wymienić kilka najpopularniejszych modeli tego typu analizatorów.

Saleae

Marka obecna na rynku od wielu lat zasłynęła przede wszystkim dzięki doskonale dopracowanemu oprogramowaniu do analizy stanów logicznych i protokołów komunikacyjnych, co ważne – dostępnemu



Fotografia 14. Analizator stanów logicznych Saleae Logic 8 Pro



Fotografia 15. Analizator stanów logicznych i protokołów Zeroplus LAP-C Pro(32256M)

nie tylko na systemy Windows, ale także Mac OSX i Linux. Firma oferuje trzy modele:

- **Logic 8** – 8 kanałów wejściowych (cyfrowych lub analogowych), próbkowanie cyfrowe do 100 MSa/s i analogowe do 10 MSa/s; interfejs USB 2.0) (fotografia 14);
- **Logic Pro 8** – 8 kanałów wejściowych, próbkowanie cyfrowe do 500 MSa/s i analogowe do 50 MSa/s; interfejs USB 3.0;
- **Logic Pro 16** – 16 kanałów wejściowych, próbkowanie cyfrowe do 500 MSa/s i analogowe do 50 MSa/s; interfejs USB 3.0.

Zeroplus

Omawiając analizatory stanów logicznych, nie sposób pominąć także obszernej oferty firmy Zeroplus. Marka opracowała aż trzy linie produktowe, zapewniające zróżnicowane parametry pomiarowe i bogaty przekrój funkcji dekodowania protokołów komunikacyjnych:

- **Seria LAP-F1** – zawiera dwa modele z końcówką -40 oraz -64, różniące się liczbą kanałów (odpowiadającą oznaczeniu modelu). Obydwa produkty oferują próbkowanie 200 MSa/s (tryb State) lub 1 GSa/s (tryb Timing) i pamięć próbek równą 4, 64, 256 lub 512 Mpts;
- **Serial LAP-C Pro** – zawiera cztery modele o próbkowaniu 250 MHz lub 1...2 GHz (zależnie od trybu pracy) i pamięci na kanał równej 64, 128 lub nawet 256 Mpts (model LAP-C Pro 32256M) (fotografia 15);
- **Seria LAP-C** – obejmuje modele o próbkowaniu do 200 MSa/s i liczbie kanałów od 16 do 32. Rozmiar pamięci dostępnej dla pojedynczego kanału waha się od 32 kpts do nawet 2 Mpts.
- Dodatkowo firma Zeroplus wprowadziła 8-kanałowy model **LAP-Educator** o próbkowaniu do 100 MHz i pamięci na kanał równej 32 kpts.

DreamSourceLab

Firma wprowadziła na rynek trzy modele analizatorów stanów logicznych (fotografia 16):

- **DSLogic Plus** – 16 kanałów, próbkowanie 400 MSa/S, pamięć 16 GSa (w trybie strumieniowym) lub 256 Msa (w trybie buforowania);
- **DSLogic U3Pro16** – 16 kanałów, próbkowanie 1 GSa/S, pamięć 16 GSa (w trybie strumieniowym) lub 2 GSa (w trybie buforowania);
- **DSLogic U3Pro32** – 32 kanały, próbkowanie 1 GSa/S, pamięć 16 GSa (w trybie strumieniowym) lub 2 GSa (w trybie buforowania).



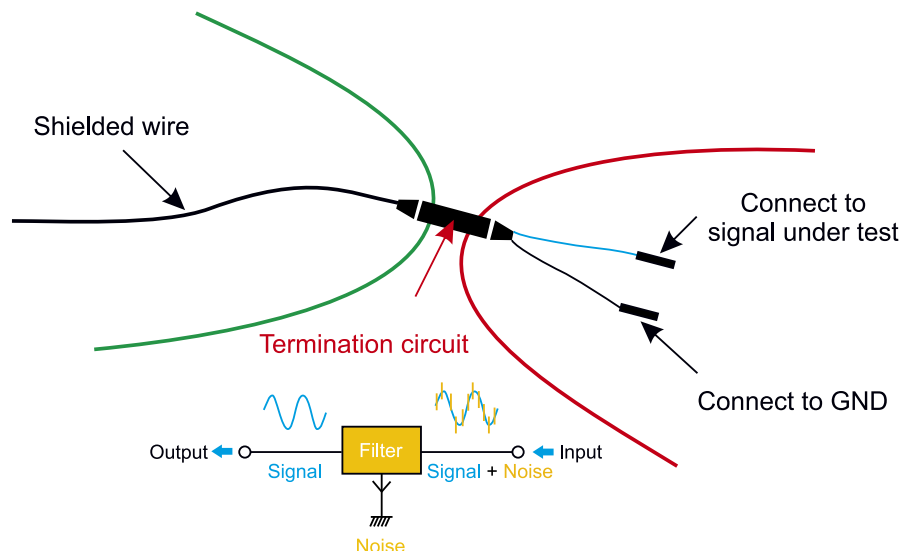
Fotografia 16. Analizator logiczny DSLogic

Interesującym rozwiązaniem jest zastosowanie przewodów koncentrycznych z filtrami przelotowymi (rysunek 3) – dzięki temu już na najwcześniejszym etapie akwizycji urządzenie dba o integralność sygnałową, minimalizując niepożądane odbicia i redukując szumy wysokoczęstotliwościowe.

Digilent

Marka Digilent w kooperacji z firmami Analog Devices i Xilinx wprowadziła na rynek interesujące rozwiązanie, przeznaczone przede wszystkim (choć nie tylko) do celów akademickich. Analog Discovery 2 (fotografia 17) to niewielki kombajn pomiarowy, zapewniający funkcjonalność:

- 2-kanałowego oscyloskopu 14-bitowego z wejściami różnicowymi, oferującego próbkowanie 100 MSa/s i pasmo do 30 MHz,
- dwukanałowego generatora arbitralnego (± 5 V, 14 bitów, 100 MSa/s, 12 MHz),
- 16-kanałowego analizatora logicznego (1,8/3,3/5 V, 100 MSa/s),
- 16-kanałowego generatora sekwencji cyfrowych (3,3 V, 100 MSa/s),
- 16-kanałowego kontrolera GPIO,
- dwukanałowego woltomierza,
- analizatora sieci z funkcjami wyznaczania charakterystyk Bodego, Nyquista, Nicholasa w paśmie od 1 Hz do 10 MHz,
- analizatora widma,
- urządzenia do akwizycji danych (DAQ),
- analizatora impedancji elementów LC,
- analizatora protokołów SPI, I²C, CAN, UART, AVR,



Rysunek 3. Konstrukcja przewodów wejściowych analizatora DSLogic



Fotografia 17. Moduł Analog Discovery 2

- dwukanałowego zasilacza regulowanego o napięciach wyjściowych ± 5 V,
- wzmacniacza audio.

Moduł wspiera matlabowy pakiet Data Acquisition Toolbox, środowisko LabView oraz SDK dla języków Python i C++, co daje

wprost nieograniczone możliwości eksperymentowania w pracowniach studenckich, niewielkich prototypowniach czy też młodych firmach (startupach), opracowujących projekty z zakresu systemów wbudowanych.

Podsumowanie

Przedstawione w artykule reprezentatywne przykłady przystawek oscyloskopowych i analizatorów stanów logicznych doskonale obrazują, jak bogaty jest obecnie rynek urządzeń pomiarowych niewyposażonych w klasyczny interfejs użytkownika. Okazuje się, że aparatura bazująca wyłącznie na oprogramowaniu komputerowym może – pod pewnymi warunkami – skutecznie zastąpić lub przynajmniej uzupełnić konwencjonalną bazę sprzętową pracowni pomiarowej. Należy jednak pamiętać o istotnych ograniczeniach i zasadach pracy, pozwalających na bezpieczne użytkowanie, bez stwarzania ryzyka porażenia elektrycznego bądź uszkodzenia komputera i/lub samego urządzenia pomiarowego.

Celowo pominęliśmy równie obszerne segmenty rynku obejmujące analizatory widma, analizatory sieci (VNA), a także inne rodzaje przyrządów pomiarowych na USB (generatory arbitralne, mostki RLC, analizatory impedancji itp.) – bowiem nawet skrótowy opis najważniejszych urządzeń tego typu daleko wykracza poza ramy pojedynczego artykułu.

inż. Przemysław Musz
przemyslaw.musz@ep.com.pl

REKLAMA

m.technik

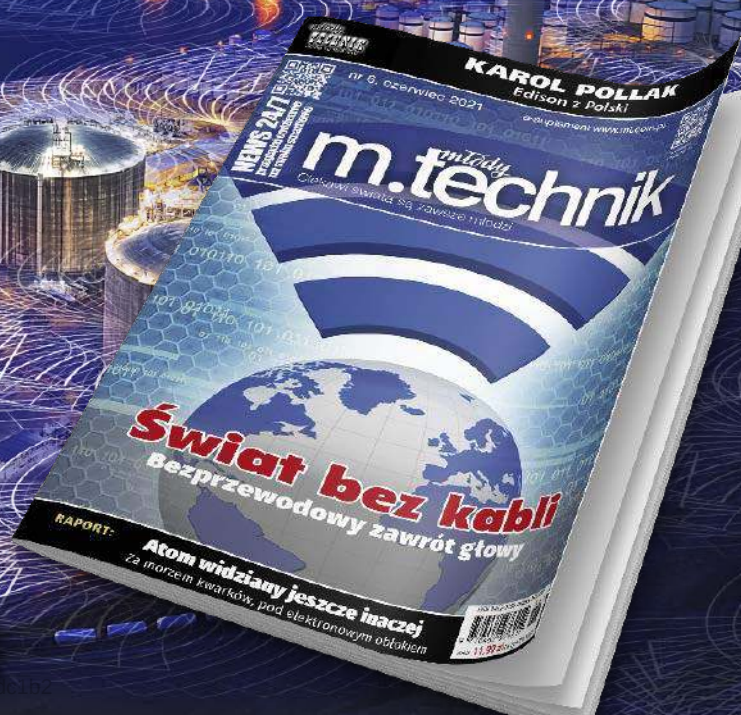
Ciekawi świata są zawsze młodzi

w prezencie na każdą okazję



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

przejrysz i kupisz na
www.ulubionykiosk.pl





Analog Discovery Pro 3000 firmy Digilent

Nowy przenośny kombajn pomiarowy

Firma Digilent poszerzyła rodzinę oferowanych przyrządów pomiarowych o wielofunkcyjne urządzenia Analog Discovery Pro 3000. Podobnie do bardzo popularnych w naszym kraju Analog Discovery 2 nowe przyrządy oferują wiele funkcji przydatnych (nie tylko!) w laboratorium elektroniki, zapewniając doskonałe parametry pomiarowe, wygodę użycia i obsługę za pomocą dojrzałego, wygodnego oprogramowania.

Oferta przyrządów pomiarowych firmy Digilent jest dość skromna, składała się bowiem do niedawna z jednego urządzenia: zintegrowanego oscyloskopu-generatora-zasilacza-analizatora Analog Discovery 2. W niewielkiej obudowie producent zintegrował wszystkie najważniejsze funkcjonalności, dzięki czemu jeden przyrząd był w stanie zaspokoić wszystkie potrzeby pomiarowe typowego laboratorium elektronicznego.

Dojrzały sprzęt

Duża popularność Analog Discovery 2 i rosnące wymagania klientów zachęciły firmę Digilent do opracowania i wprowadzenia do produkcji zaawansowanej wersji zintegrowanego przyrządu pomiarowego, łączącego w sobie:

- 2- lub 4-kanalowy oscyloskop (w zależności od modelu),
- analizator widma, analizator sieci, analizator impedancji,
- 16-kanalowy rejestrator-analizator stanów logicznych,
- 16-kanalowy generator przebiegów cyfrowych,
- 2-kanalowy generator arbitralny, za pomocą którego można generować dowolne przebiegi analogowe i cyfrowe,
- analogowy i cyfrowy data logger,
- statyczne wyjścia cyfrowe, sterowane z poziomu aplikacji sterującej lub za pomocą dedykowanego oprogramowania bazującego na SDK WaveForm od firmy Digilent,

Więcej informacji:

Wyłącznym dystrybutorem edukacyjnym firmy Digilent w Polsce jest KAMAMI.pl.



- wielokanałowy woltomierz (o liczbie kanałów takiej samej jak liczba wejść oscyloskopowych),
- zasilacz o programowanym napięciu wyjściowym.

Urządzenie oznaczone jako ADP3450 jest przyrządem z 4-kanalowym oscyloskopem, a ADP3250 jest wyposażony w oscyloskop 2-kanalowy. W obydwu przypadkach rozdzielczość w osi Y wynosi 14 bitów, a maksymalna częstotliwość próbkowania to 125 MHz. W przypadku akwizycji sygnałów okresowych można wirtualnie zwiększyć częstotliwość próbkowania do 500 MHz w każdym kanale, w czym pomaga mechanizm oversamplingu. Deklarowane przez producenta pasmo analogowe kanałów oscyloskopowych wynosi 55 MHz, a bezpieczny dla obwodów wejściowych zakres napięć mieści się w przedziale ± 25 V w stosunku do masy przyrządu.

Także w generatorze arbitralnym zastosowano przetwornik o rozdzielczości 14 bitów. Maksymalna częstotliwość próbkowania sygnałów wyjściowych wynosi 125 MHz, przy której możliwe jest uzyskanie pasma analogowego 15 MHz. Zakres napięć wyjściowych generowanych przebiegów wynosi $\pm 3,5$ V dla maksymalnego dopuszczalnego obciążenia.

Linie cyfrowe są konfigurowalne: można zmieniać ich kierunkowość i referencyjne napięcie zasilania w zakresie od 1,2 do 3,3 V. W trybie rejestratora-analizatora urządzenie może automatycznie dekodować protokoły SPI, I²C, UART, CAN, I²S, 1-Wire, HDMI CEC oraz kody Manchester, a także protokoły definiowane przez użytkownika. W trybie generatora sekwencji cyfrowych użytkownik może wykorzystać m.in. funkcje: clock, pulse, random, number, binary counter, Gray counter, Johnson counter, decimal counter, walking 0/1, może także definiować własne wzorce.

Analizatory analogowe są dołączone do kanałów wejściowych zintegrowanego oscyloskopu, za ich pomocą można m.in. wykonać analizę FFT podglądanego przebiegu (z oknami Rectangular, Triangular, Hamming, Hann, Cosine, Blackman-Harris, Flat Top, Kaiser) w paśmie do 50 MHz. Z kolei analizator sieci pozwala używać sygnału

referencyjnego o częstotliwości do 10 MHz, a uzyskiwane wyniki mogą być prezentowane w postaci wykresów Nicholasa, Nyquista lub jako przebiegi odnoszące się do osi czasu.

Podsumowanie tego krótkiego zestawienia wybranych cech i parametrów wyraźnie wskazuje na zamiary twórców przyrządu: maksymalizacja funkcjonalności przy zapewnieniu zakresów parametrów najczęściej potrzebnych w typowych laboratoriach elektronicznych. Bez epatowania wielkimi liczbami, które jak pokazuje praktyka, w wielu przypadkach nie mają pokrycia w realnych parametrach przyrządów.

Wygodne i stabilne oprogramowanie

Wszystkie przyrządy z rodziny Analog Discovery są obsługiwane przez bezpłatne oprogramowanie o nazwie WaveForms. Jest to takie samo oprogramowanie, jakie znają użytkownicy Analog Discovery 2, przy czym dostępne w nim nastawy są automatycznie dostosowywane do aktualnie wybranego przyrządu.

Z poziomu WaveForms są realizowane nastawy wszystkich przyrządów zintegrowanych w urządzeniach, odpowiada ono także za wyświetlanie przebiegów analizowanych i mierzonych sygnałów. Przykładowe widoki okna programu obsługującego Analog Discovery Pro pokazano na rysunkach 1...3. Warto zauważyć, że dzięki wbudowanemu w oprogramowanie edytorowi skryptów użytkownicy mogą pisać własne procedury testowe w języku JavaScript, dzięki czemu możliwa jest wygodna automatyzacja pomiarów lub analiza ich wyników. Przykład wyglądu okna edytora skryptu pokazano na rysunku 4.

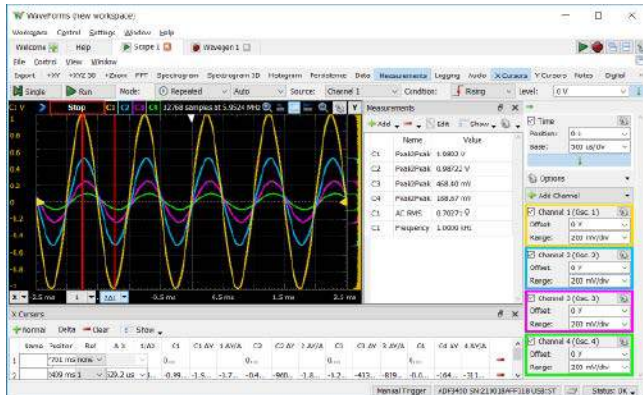
Oprogramowanie WaveForms jest dostępne w wersjach na systemy operacyjne iOS, Windows i Linux, dostępne jest także kompletny, dobrze udokumentowany SDK, który umożliwia przygotowanie własnych programów używających przyrządów Analog Discovery Pro. Sposób przygotowania SDK pozwala na używanie bibliotek z poziomu kompilatorów lub interpreterów języków C, C++, C#, Visual Basic oraz Python. Dostępne są także toolkity pozwalające używać Analog Discovery Pro w środowiskach LabVIEW oraz MATLAB.

Linux w aplikacjach pomiarowych

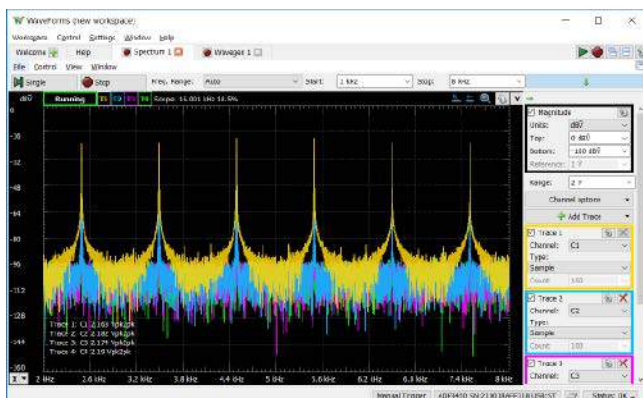
Sercem przyrządu Analog Discovery Pro jest układ z rodziny ZYNQ firmy Xilinx. Integruje on w swojej strukturze konfigurowalną część FPGA oraz 2-rdzeniowy mikroprocesor Cortex-A9, który działa pod opieką systemu operacyjnego Linux. Producent przyrządu umożliwił opcjonalne użycie interfejsu konsoli do komunikacji oprogramowania systemowego z użytkownikiem, co wymaga uruchomienia przyrządu w trybie Linux Mode Boot (rysunek 5). Funkcję interfejsu użytkownika spełnia w tym trybie program terminalowy (rysunek 6).

Zaawansowane tryby wyzwalania

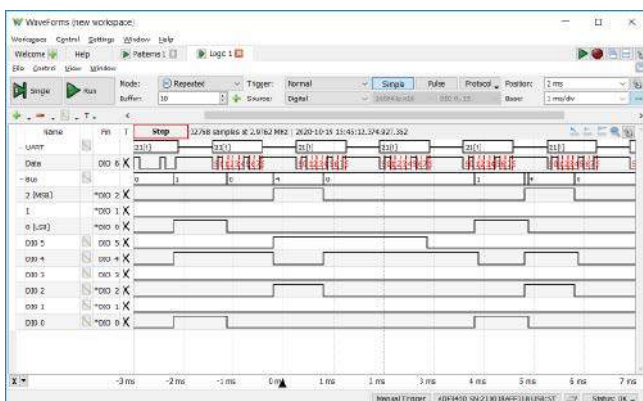
Dzięki integracji wielu przyrządów w jednym urządzeniu, Analog Discovery Pro oferuje użytkownikom wiele możliwości zaawansowanego wyzwalania. Przykładowo możliwe jest wyzwolenie podstawy czasu w oscyloskopie po odebraniu i zdekodowaniu przez



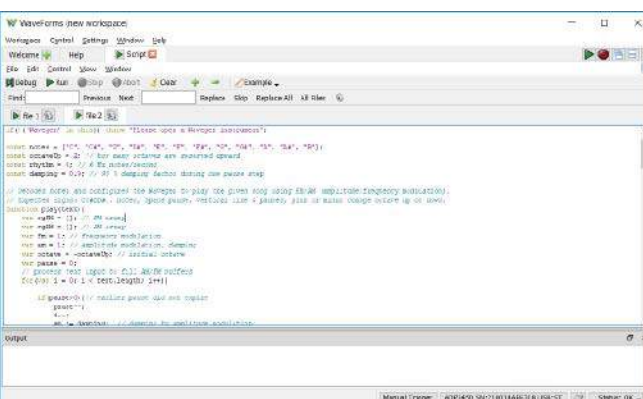
Rysunek 1. Okno WaveForms w trybie oscyloskopu



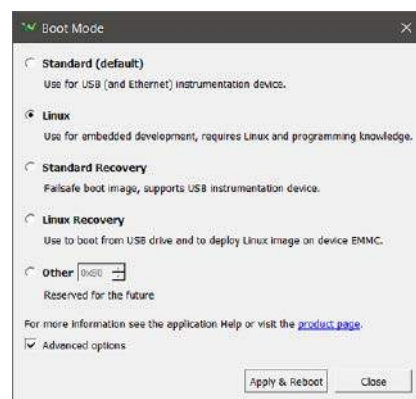
Rysunek 2. Okno WaveForms w trybie analizatora widma



Rysunek 3. Okno WaveForms w trybie analizatora stanów logicznych



Rysunek 4. Okno WaveForms w trybie edytora JavaScript



Rysunek 5. Możliwe tryby bootowania Analog Discovery Pro

PREZENTACJE

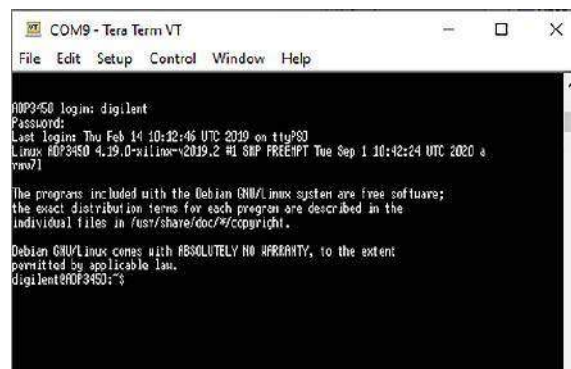
analizator stanów logicznych ramki danych o określonej zawartości. Innym przykładem jest możliwość wygenerowania sekwencji tworzącej przebieg analogowy na wyjściu generatora arbitralnego po przekroczeniu zadanej wartości napięcia zmierzzonego w wybranym kanale analogowym. Analog Discovery Pro jest wyposażony także w dwa standardowe wejścia wyzwalające, których funkcjonalności można konfigurować z poziomu aplikacji WaveForms.

Wszelchstronna komunikacja

Aplikacja WaveForms komunikuje się z Analog Discovery Pro za pomocą interfejsu USB lub Ethernet. Urządzenie wyposażono w 4 dodatkowe złącza USB-A, które są dostępne z poziomu systemu Linux. Można je wykorzystać do dołączenia pamięci Flash USB lub dongle'a Wi-Fi.

Zaawansowany support użytkowników

Z myślą o użytkownikach Analog Discovery Pro firma Digilent przygotowała bardzo rozbudowaną dokumentację oraz przykłady użycia, które są – tradycyjnie – dostępne bezpłatnie. I tak, oprócz Starter Guide'a, który prowadzi użytkownika za rękę po podstawowych możliwościach przyrządu, producent przygotował także przykłady



Rysunek 6. Okno terminalu konsoli Linux dołączonego do Analog Discovery Pro

ilustrujące autonomiczne wykonywanie pomiarów. Wśród nich znajduje się także „sztuczka” pozwalająca na wykorzystanie maksymalnej częstotliwości próbkowania w torze analogowym do 500 MHz, która to wartość nie jest dostępna w domyślnym trybie pracy przyrządu. Szczegóły na stronie producenta.

KAMAMI.pl

REKLAMA



KOMPUTERY RASPBERRY PI I MODUŁY ARDUINO



AVT SPV Sp. z o.o.
03-197 Warszawa, ul. Leszczynowa 11
Sklep stacjonarny czynny jest od poniedziałku do piątku w godzinach: 8:00–16:00, tel. 22 257 84 66
e-mail: handlowy@avt.pl, tel. 22 257 84 51 (w godzinach: 9:00–15:00)

<http://sklep.avt.pl>

Pomiary natężenia prądu w systemach wbudowanych (1)

Praktyczny poradnik

Pomiar parametrów analogowych, czy to za pomocą przetworników ADC obecnych w strukturach mikrokontrolerów, czy też z użyciem specjalizowanych przetworników lub front-endów scalonych, należy do podstawowych funkcjonalności przeważającej części systemów wbudowanych. O ile jednak napięcia mogą być w wielu przypadkach dostarczane bezpośrednio do wejść przetworników, o tyle w przypadku prądu zawsze konieczne jest zastosowanie mniej lub bardziej złożonych metod konwersji I/U.

Zakres aplikacji układów do pomiaru natężenia prądu jest bardzo szeroki, a dobór konkretnej metody zależy zarówno od pożądanego zakresu oraz dokładności pomiarowej, jak i szeregu innych czynników. Niebagatelną rolę odgrywa tutaj m.in. ewentualna separacja galwaniczna układu pomiarowego od obwodu wykonawczego, rodzaj uzyskiwanej informacji (binarna lub ilościowa), kształt przebiegów prądowych czy też minimalne wymagane pasmo częstotliwości. Oto najczęściej spotykane zastosowania pomiarów amperometrycznych – z pominięciem rzecz jasna samych amperomierzy, watomierzy i innych przyrządów pomiarowych jako takich:

- monitorowanie poboru prądu na szynach zasilania elementów wykonawczych, modułów rozszerzeń czy też wszelkich podzespołów większego systemu, realizowane w celu detekcji potencjalnych awarii;
- sterowanie silników – w celu dostarczenia informacji zwrotnej dla algorytmu sterującego (w bardziej zaawansowanych napędach) oraz detekcji przeciążenia i/lub braku połączenia z silnikiem (otwarcie obwodu);
- odczyt danych z pętli prądowej 4...20 mA (lub rzadziej 0...20 mA),
- określanie stanu naładowania akumulatora i (pośrednio) stopnia jego zużycia (tzw. liczniki kulombów);
- pomiar rezystancji (współpraca z czujnikami RTD, sensorami wilgotności względnej i stężenia gazów, czujnikami nacisku, itd.) [1].

Podstawy fizyczne

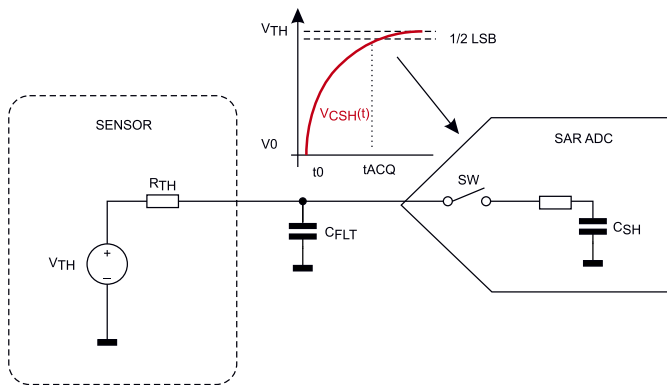
Stosowane powszechnie metody pomiaru prądu można podzielić na trzy podstawowe grupy:

- **Pomiar spadku napięcia na boczniku** – w tym przypadku stosuje się bezpośrednio prawo Ohma w postaci $I=U/R$, a w roli rezystora bocznikowego mogą być stosowane różnego rodzaju elementy – począwszy od standardowych rezystorów THT lub SMD, poprzez specjalnie do tego celu zaprojektowane rezystory dużej mocy, aż po boczniki instalacyjne, montowane poza płytką drukowaną i umożliwiające podłączenie poprzez zaciski śrubowe;



- **Pomiar z wykorzystaniem zjawiska Halla** – do pomiaru zastosowany jest miniaturowy hallotron, sprzężony magnetycznie z przewodnikiem, przez który przepływa mierzony prąd. Pole magnetyczne, o natężeniu wprost proporcjonalnym do wartości natężenia prądu, powoduje wytworzenie napięcia różnicowego na zaciskach pomiarowych płytki hallotyonu, zasilanej za pomocą precyzyjnego źródła prądowego;
- **Pomiar za pomocą przekładnika prądowego** – metoda także opiera się na sprzężeniu magnetycznym, jednak w tym przypadku, w odróżnieniu od klasycznego transformatora, w roli uzwojenia pierwotnego występuje przewód, przewleczony przez otwór magnetowodu toroidalnego, na którym nawinięte jest uzwojenie wtórne. Zatem zachowana pozostaje liniowość (w znacznym zakresie prądów wejściowych), a dobór odpowiedniego stopnia przekładni takiego transformatora umożliwia uzyskanie czułości pomiarowej odpowiedniej do danej aplikacji.

Niejako osobną kategorię stanowi **pomiar za pomocą wzmacniacza transimpedancyjnego (TIA)**. Ta metoda wymaga najczęściej niewielkiego układu ze wzmacniaczem operacyjnym w konfiguracji odwracającej. Najprostsza wersja wzmacniacza TIA zawiera tylko jeden rezystor (R_f) w pętli sprzężenia zwrotnego, na którym występuje spadek napięcia zależny od wartości prądu wejściowego, wprowadzonego przez źródło (np. fotodiodę) podłączone do wejścia odwracającego wzmacniacza. Wejście nieodwracające jest wykorzystywane do ustalenia offsetu napięciowego (napięcia odniesienia,



Rysunek 1. Wewnętrzna pojemność przetwornika SAR (CSH) ogranicza impedancję wejściową konwertera tym bardziej, im wyższa jest ustawiona częstotliwość próbkowania

V_{ref}), np. przez podłączenie sztucznej masy. W celu poprawy stabilności i dodatkowego ograniczenia poziomu szumów, do rezystora R_i zwykle dołącza się równoległy kondensator C_i o niewielkiej wartości. Od strony fizycznej opisywany sposób pomiaru jest jednak wyłącznie modyfikacją metody bocznikowej.

Pomiar napięcia na boczniku

Podstawową zaletą metody bocznikowej okazuje się napięciowy charakter sygnału wyjściowego oraz możliwość pracy zarówno z prądem stałym, jak i zmiennym, w dość szerokim zakresie częstotliwości (ograniczonym efektami pasożytniczymi bocznika, w tym przede wszystkim jego indukcyjnością). Możliwość uzyskania relatywnie wysokiej czułości, dzięki doborowi odpowiedniej rezystancji bocznika, jest jednak okupiona dużymi stratami mocy na rezystorze, co nie tylko zmniejsza ilość energii dostarczanej do obciążenia, ale co gorsza, znacząco obniża dokładność metody. Efekt samopodgrzewania bocznika powoduje bowiem zmianę jego rezystancji, a to bezpośrednio pociąga za sobą wprowadzenie dodatkowego błędu pomiaru, współistniejącego obok niedokładności układu pomiarowego (np. nieliniowości przetwornika ADC i tolerancji oraz temperaturowego dryfu źródła napięcia odniesienia), a także tolerancji rezystancyjnej samego bocznika.

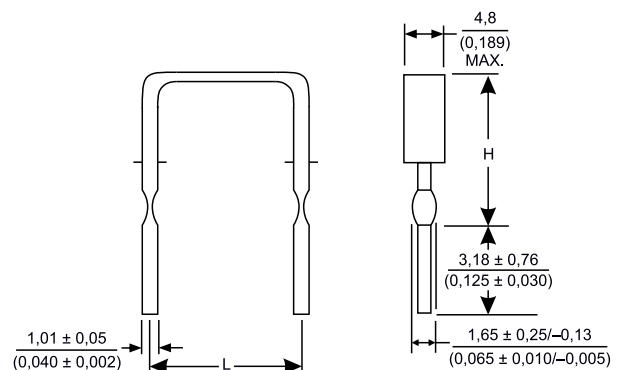
Problem efektu termicznego w boczniku można zredukować następującymi metodami:

- **Zastosowanie akwizycji bezpośredniej i przetwornika ADC o wysokiej rozdzielczości** – nowoczesne przetworniki i uniwersalne front-endy scalone mają konwertery ADC o rozdzielczości rzędu 20...24 bitów, którą można dodatkowo zwiększyć poprzez nadpróbkowanie (*oversampling*) z cyfrowym uśrednieniem wielu wyników pomiaru (co jest rzecz jasna okupione bardzo istotnym spadkiem efektywnej częstotliwości odczytów). Przetworniki o rozdzielczości 24 bitów i więcej są rekomendowane do aplikacji z bezpośrednim przetwarzaniem sygnałów wejściowych z czujników (lub boczników prądowych), tj. wykorzystujących akwizycję napięć z pominięciem wzmacniacza wejściowego. Bezpośredni pomiar napięcia na boczniku jest prawie zawsze możliwy dzięki znaczącej dysproporcji pomiędzy rezystancją bocznika a impedancją wejściową przetworników. Bowiem w większości przypadków rezystor bocznikowy ma wartość przynajmniej kilkaset razy niższą niż rezystancja zastępcza wejścia przetwornika, nawet w przypadku najbardziej niekorzystnej konfiguracji. Taka konfiguracja występuje w przetwornikach typu SAR przy dużej częstotliwości próbkowania – obniżenie impedancji wejściowej następuje na skutek impulsowego poboru prądu, spowodowanego częstym ładowaniem wewnętrznej pojemności (**rysunek 1**). Dzięki temu dokładność pomiaru bezpośredniego nie ucierpi w znaczącym stopniu, w myśl „złotej zasady metrologii” wprowadzającej wymóg zapewnienia wielokrotnie wyższej rezystancji wejściowej woltomierza niż rezystancja zastępcza badanego obwodu;

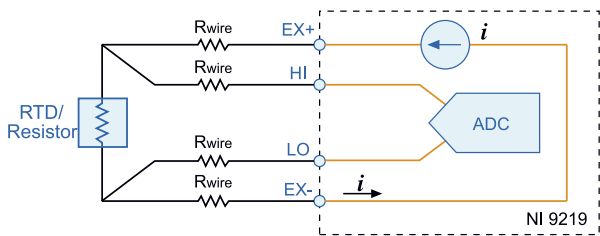
- **Zastosowanie wzmacniacza wejściowego** – włączenie dodatkowego wzmacniacza pomiędzy bocznik a wejście przetwornika pozwala zmniejszyć wartość rezystancji bocznika proporcjonalnie do współczynnika wzmocnienia. To zaś bezpośrednio przekłada się na mniejsze straty mocy, a co za tym idzie, także na osłabienie efektu samopodgrzewania. Możliwe jest także zmniejszenie wymiarów bocznika, jeżeli ma to znaczenie dla mechanicznych aspektów danej aplikacji (ilość miejsca na PCB, dostępna przestrzeń wewnątrz obudowy, itd.);
- **Zastosowanie bocznika o dużych wymiarach** (dużej dopuszczalnej mocy strat P_{tot}) – rozwiązanie możliwe do wdrożenia zarówno poprzez zwiększenie obudowy elementu (np. zastosowanie rezystora SMD2512 zamiast SMD1206), jak i przejście na bocznik „z prawdziwego zdarzenia” (np. przewlekany rezystor dużej mocy o możliwie małej tolerancji);
- **Dobór bocznika o niewielkim współczynniku TCR** – w większości przypadków zastosowanie zwykłego rezystora (nawet o większej wartości P_{tot}) będzie ograniczało dokładność pomiaru z uwagi na relatywnie wysoką wartość temperaturowego współczynnika rezystancji. Warto zatem rozważyć zastosowanie specjalnego rezystora bocznikowego. W handlu dostępne są zarówno wersje przewlekane (np. w formie precyzyjnie dobranej blaszki metalowej o dużej stabilności rezystancji – **rysunek 2**), jak i SMT (zwykle w rozmiarach SMD1206 lub większych).

W praktyce okazuje się, że nawet po zastosowaniu dobrego przetwornika z precyzyjnym źródłem napięcia odniesienia oraz wysokiej klasy bocznika dokładność pomiaru prądu dla większych natężeń może pozostać niezadowalająca. W takich przypadkach znaczenie zyskuje bowiem szereg aspektów powiązanych bezpośrednio z konstrukcją PCB – sposób doprowadzenia prądu (z mierzonego obwodu do bocznika) oraz napięcia wyjściowego (z bocznika do wejścia układu pomiarowego). Duże wartości prądu powodują odkładanie (trudnych do oszacowania) spadków napięcia na rezystancji ścieżek, wyprowadzeń elementu pomiarowego oraz (nade wszystko) na rezystancji połączeń lutowanych pomiędzy bocznikiem a płytką (lub przewodami a bocznikiem, w przypadku rezystorów umieszczonych poza PCB).

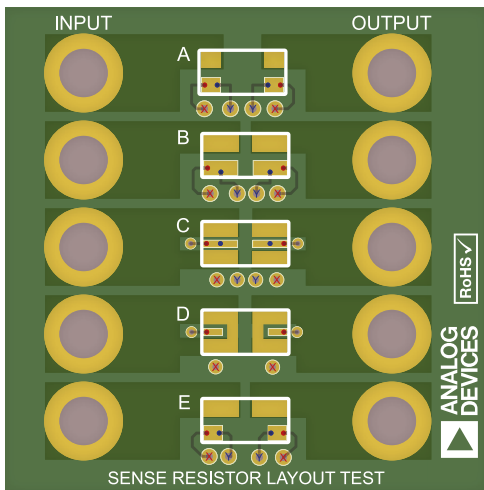
Identyczny problem, choć w zupełnie innej skali liczbowej, występuje przy próbach pomiaru rezystancji czujnika (np. termorezystora PT100 lub PT1000), podłączonego do wejścia urządzenia pomiarowego stosunkowo długim (i/lub cienkim) przewodem dwużyłowym. Rezystancja samego przewodu „dokłada się” bowiem do rezystancji czujnika, wprowadzając dodatkowy błąd pomiarowy o składowej stałej (zależnej od stosunku rezystancji mierzonej i sumarycznej rezystancji przewodów) oraz zmiennej (zależnej od warunków otoczenia, przede wszystkim z powodu niezerowego współczynnika termicznego miedzi, efektów utleniania, itd.). We wszystkich tego typu przypadkach z pomocą przychodzi czteroprzewodowa metoda Kelvina (**rysunek 3**), wykorzystująca parę przewodów zasilających („prądowych”) oraz pomiarowych („napięciowych”). Metoda Kelvina jest fundamentalnym rozwiązaniem problemu rezystancji przewodów



Rysunek 2. Precyzyjny rezystor bocznikowy w wersji THT, typ PWR-4412-2A marki Bourns



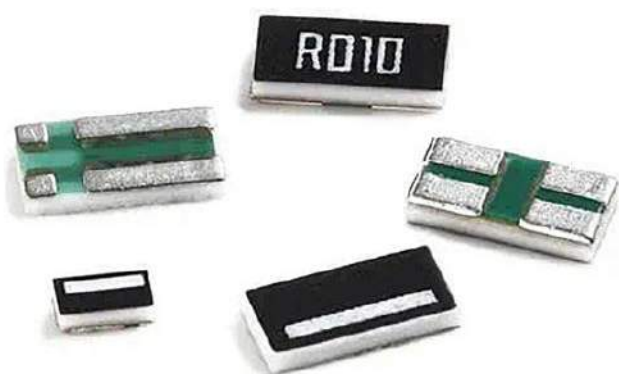
Rysunek 3. Metoda 4-przewodowa Kelvina; na pomiar spadku napięcia na R_x nie wpływają wartości rezystancji przewodów prądowych R_{s1} i R_{s4} (ze względu pomiar różnicowy) oraz przewodów napięciowych R_{s2} i R_{s3} (z uwagi na wysoką impedancję wejściową przetwornika ADC)



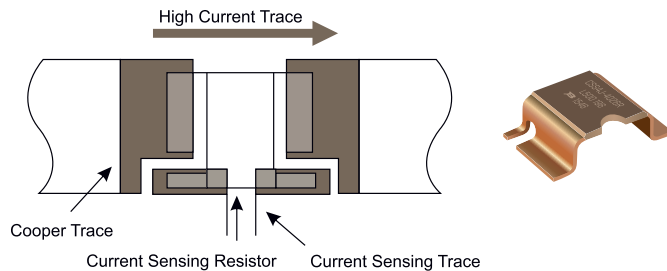
Rysunek 4. Różne warianty padów lutowniczych dla 4-przewodowego połączenia bocznika w postaci typowego rezystora SMD dużej mocy

pomiarowych, stosowanym niemal we wszystkich gałęziach metrologii elektronicznej, a nawet medycznej, gdzie oddaje nieocenione usługi w pomiarach bioimpedancyjnych (w których udział impedancji elektroda-skóra przyjmuje wartości na tyle wysokie, że w żadnym przypadku nie mogą one zostać pominięte w metodzie pomiarowej).

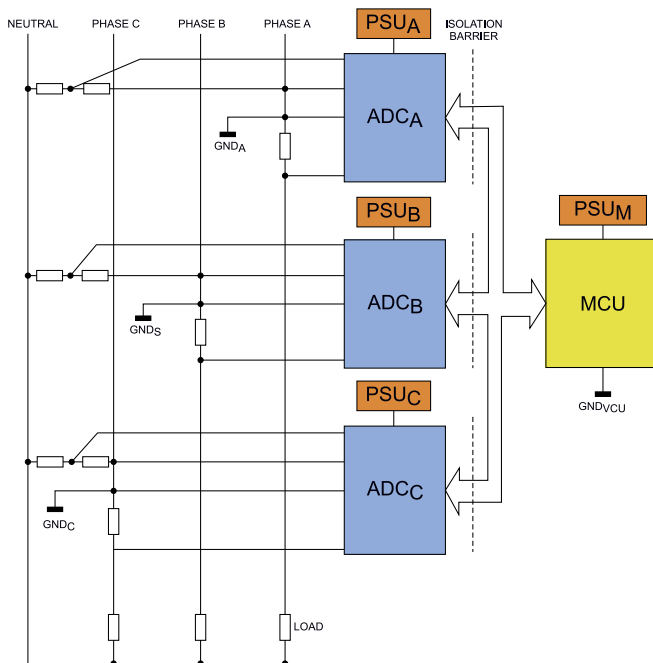
Co ciekawe i ważne: praktyka pokazuje, że w przypadku pomiarów większych wartości prądu znaczenie zyskuje nawet... kształt padów lutowniczych bocznika (dotyczy to elementów SMD) oraz sposób wprowadzenia do nich zakończeń ścieżek PCB. Tematyka ta została dogłębnie opisana w artykule *Optimize High-Current Sensing Accuracy by Improving Pad Layout of Low-Value Shunt Resistors*, wydanym w periodyku „Analog Dialogue” [2]. Uzyskane wyniki mogą przyprawić o zawrót głowy – błąd pomiaru napięcia na boczniku dla różnych footprintów (**rysunek 4**) rezystora SMD2512 wynosił od 0,6% do aż 5%, zaś w przypadku pomiaru na padach parzystych



Fotografia 1. Różne rodzaje rezystorów bocznikowych SMD z poczwórnymi padami lutowniczymi marki Yageo



Rysunek 5. Rezystor bocznikowy SMD w postaci wyprofilowanej metalowej blaszki z padami do podłączenia 4-przewodowego oraz rekomendowany footprint

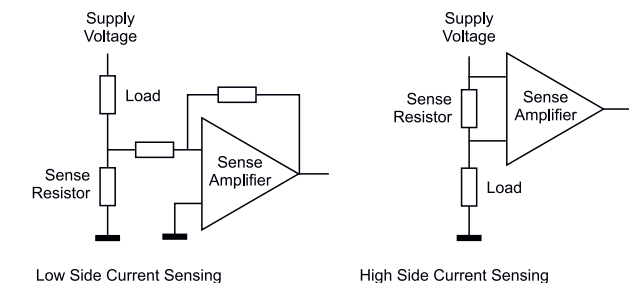


Rysunek 6. Schematyczne zobrazowanie topologii układu wejściowego 3-fazowego licznika energii elektrycznej; osobne przetworniki ADC₁...ADC₃ wymagają rozdzielonych galwanicznie zasilaczy (PSU₁...PSU₃) oraz izolowanej szyny komunikacyjnej, gdyż mamy analogowe są współpólne z poszczególnymi przewodami fazowymi

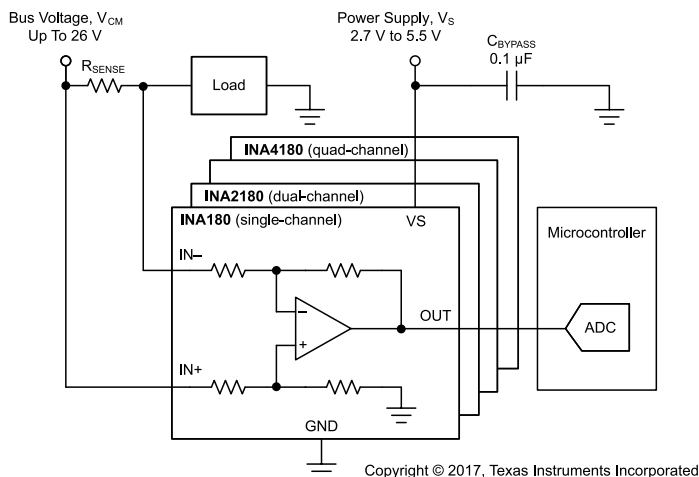
(tj. z zastosowaniem połączenia 2-przewodowego) błąd wyniósł aż 22,8 %! [3]. W aplikacjach wymagających wysokiej precyzji pomiaru warto rozważyć zastosowanie specjalnego bocznika SMD, wyposażonego w podwójne (dzielone) pady lutownicze i mającego postać zbliżoną do klasycznego rezystora (**fotografia 1**) bądź specjalnie wyprofilowanego rezystora metalowego (**rysunek 5**).

Do podstawowych wad pomiarów bocznikowych, oprócz wspomnianych strat mocy i efektów z tym związanych, należy konieczność bezpośredniego połączenia wejścia układu pomiarowego z bocznikiem. Ma to szczególne znaczenie w obwodach sieciowych oraz wszelkich innych, w których obwód prądu mierzonego powinien (np. ze względów bezpieczeństwa) pracować za barierą galwaniczną. W przypadku urządzeń wymagających jednoczesnego pomiaru napięcia i prądu w tym samym obwodzie, często stosowaną konfiguracją jest pomiar w układzie „pływającym”, tj. z masą układu pomiarowego u wspólnioną z przewodem fazowym (przykład układu wejściowego trójfazowego licznika energii z masami analogowymi połączonymi z odpowiednimi przewodami fazowymi zaprezentowano na **rysunku 6**).

W ogólnym ujęciu pomiar prądu za pomocą bocznika może odbywać się w konfiguracji *high-side* bądź *low-side*, zależnie od tego, w którym miejscu obwodu względem masy układu jest ustawiony bocznik (**rysunek 7**). Choć do obsługi boczniców w wielu praktycznych układach mogą być z powodzeniem użyte wzmacniacze pomiarowe bądź

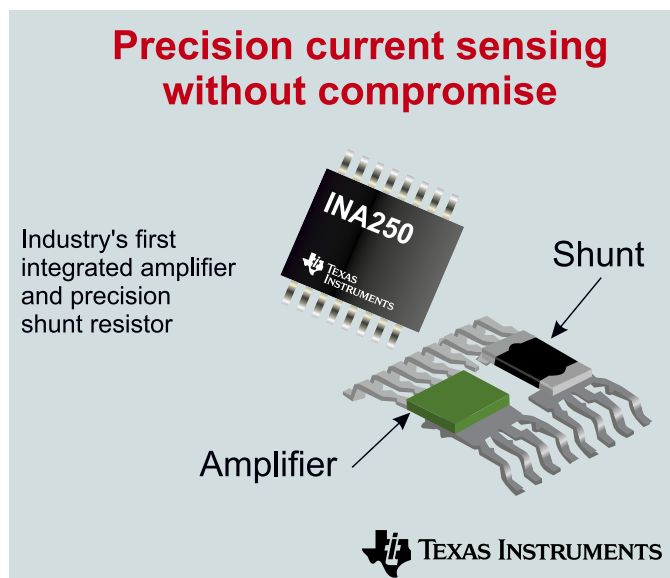


Rysunek 7. Schematyczne przedstawienie topologii pomiaru prądu typu high-side oraz low-side

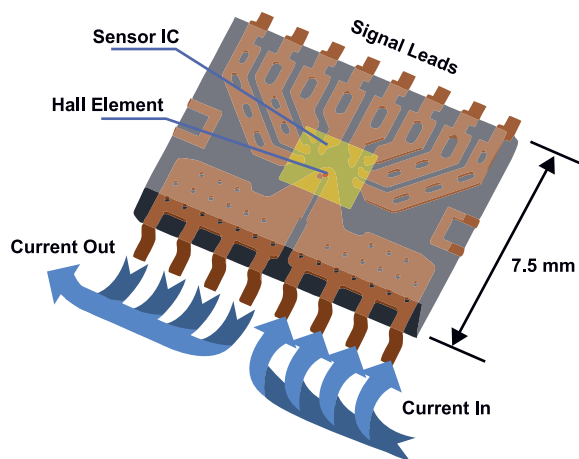


Rysunek 8. Schemat aplikacyjny wzmacniacza z rodziny INAx180

różnicowe, to doskonałą alternatywę dla nich stanowią specjalizowane, scalone wzmacniacze określane handlowo jako tzw. *current sense amplifiers*. Doskonały przykład to rodzina układów INAx180 marki Texas Instruments, zawierająca modele 1-, 2- oraz 4-kanalowe (odpowiednio: INA180, INA2180 oraz INA4180) o fabrycznie ustalonych wzmocnieniach równych 20 V/V, 50 V/V, 100 V/V lub 200 V/V (**rysunek 8**). Układy mogą pracować w każdej z przedstawionych konfiguracji i są odporne na napięcia wspólne rzędu od -0,2 V aż do 26 V, przy zasilaniu „strony wyjściowej” z przedziału od 2,7 V do 5,5 V. Z tego też względu doskonale nadają się one do aplikacji w typowych, niskonapięciowych systemach wbudowanych, współpracujących na wspólnej masie z obciążeniami zasilanymi napięciem do 24 VDC. Ciekawym rozwiązaniem jest pierwszy na rynku wzmacniacz



Rysunek 9. Wzmacniacz pomiarowy ze zintegrowanym boczniakiem typu INA250 marki TI



Rysunek 10. Budowa hallotronowego czujnika prądu

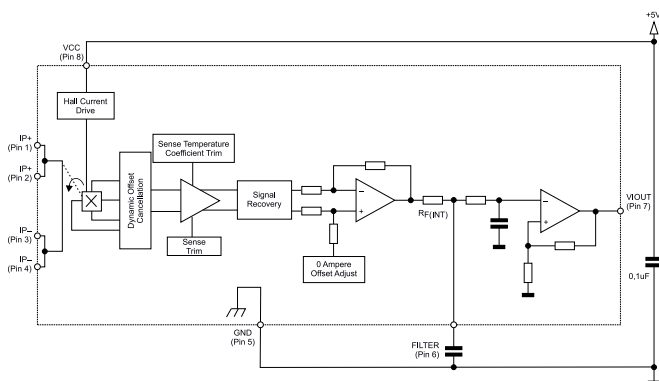
zintegrowany w jednej obudowie z boczniakiem – INA250, także produkcji TI (**rysunek 9**).

Pomiar za pomocą hallotronu

Niekwestionowaną zaletą pomiarów prądu za pomocą hallotronu jest całkowita izolacja galwaniczna obwodu wejściowego od przewodnika, przez który płynie mierzony prąd, dzięki czemu można uniknąć łączenia masy części niskonapięciowej z przewodem fazowym (lub w ogólności – obwodem mierzonym). Do wad hallotronów należy natomiast bardzo mała czułość, ograniczona przez samą zasadę działania tej metody (efekt Halla) – stąd konieczne jest stosowanie wzmacniaczy o dość dużym wzmocnieniu, a metoda praktycznie nie nadaje się do pomiaru niewielkich wartości prądu. „Dyskretnie” aplikacje czujników Halla w pomiarach amperometrycznych spotyka się w codziennej praktyce stosunkowo rzadko. Zamiast nich furorę zrobiły hybrydowe układy scalone, zawierające w jednej obudowie miniaturowy przewodnik o bardzo niskiej rezystancji, a także strukturę półprzewodnikową z płytką hallotronu oraz wzmacniaczem i układami kondycjonowania sygnału. Wbudowana elektronika ogranicza od góry pasmo przepustowe scalonych sensorów prądu, ale kluczowe plusy – prostota implementacji i stosunkowo niski koszt – w większości przypadków przeważają szalę decyzji projektowej na korzyść takiego właśnie rozwiązania. Schemat budowy scalonego czujnika prądu pokazano na **rysunku 10**, zaś na **rysunku 11** zaprezentowano schemat funkcjonalny scalonego czujnika prądu ACS712 firmy Allegro Microsystems. Układ zapewnia izolację o wytrzymałości 2,1 kV, pasmo przepustowe aż 80 kHz i zakres pomiarowy od ± 5 A do ± 30 A.

Pomiar za pomocą przekładnika

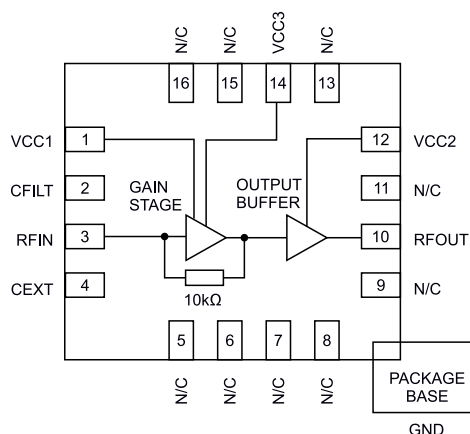
Całkowita izolacja galwaniczna pomiędzy obwodem mierzonym i układem pomiarowym, a także możliwość pracy z bardzo wysokimi natężeniami prądu przemiennego to najważniejsze zalety przekładników prądowych. Bez większego problemu można kupić zarówno



Rysunek 11. Schemat funkcjonalny czujników prądu z rodziny ACS712



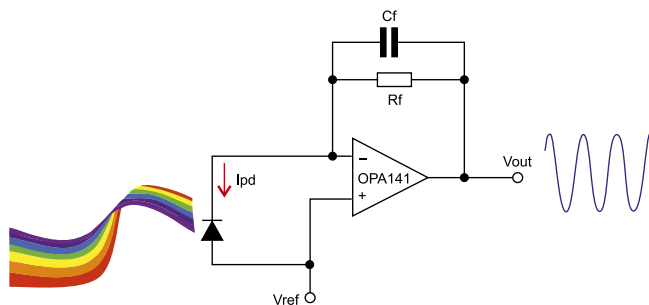
Fotografia 2. Przekładniki do montażu przewlekane marki Talema



Rysunek 12. Budowa scalonego wzmacniacza transimpedancyjnego 10 kΩ typu HMC799LP3E

niewielkie transformatory prądowe montowane na PCB (**fotografia 2**) i dostosowane do pomiaru prądów na poziomie kilku amperów, jak i duże przekładniki wysokoprądowe, obsługujące linie zasilania o wydajności nawet kilku kA. Dodatkowo, niektóre modele są wyposażone w dwuczęściowy magnetowód (split-core), który umożliwia założenie przekładnika na przewód zasilania w całkowicie nieinwazyjny sposób, tzn. bez konieczności przecinania lub rozłączania kabla podawanego pomiarowi.

Pewnym problemem z punktu widzenia projektanta systemu wbudowanego wykorzystującego przekładnik jest... prądowy charakter sygnału wyjściowego, który wymusza zastosowanie układu konwertera I/U. Należy przy tym pamiętać, że uzwojenie wtórne transformatora prądowego jest samo w sobie źródłem prądu, dzięki czemu konwersja sygnału do postaci napięciowej może być z powodzeniem wykonana za pomocą... precyzyjnego rezystora. Warto jednak przemyśleć dodatkowe ograniczenie napięcia, które zabezpieczy obwód wejściowy przetwornika (bądź przedwzmacniacza) przed uszkodzeniem, spowodowanym przez skok napięcia, wywołany przez potencjalne przeciążenie obwodu pierwotnego.



Rysunek 13. Podstawowa konfiguracja wzmacniacza transimpedancyjnego na bazie wzmacniacza operacyjnego

Pomiar z użyciem wzmacniacza TIA

Wzmacniacze transimpedancyjne są stosowane do pomiaru niewielkich prądów stałych lub zmiennych, w tym również krótkich, szybkozmiennych sygnałów (np. impulsowych). Z tego też względu odpowiednio zaprojektowanych, szerokopasmowych wzmacniaczy o dużej stabilności używa się do realizacji układów komunikacyjnych z wykorzystaniem światłowodów. Choć na rynku dostępne są gotowe, scalone wzmacniacze transimpedancyjne (np. HMC799LP3E o paśmie przepustowym 700 MHz i poziomie szumu wejściowego zaledwie 150 nA), to w wielu urządzeniach preferowane są klasyczne wzmacniacze operacyjne, czy to w swojej podstawowej konfiguracji, takiej jak na **rysunku 13**, czy też w bardziej złożonej topologii układowej, stosowanej w niektórych typach urządzeń pomiarowych np. w celu zapewnienia lepszej skuteczności tłumienia zakłóceń.

Podsumowanie

Tematyka pomiaru natężenia prądu ma fundamentalne znaczenie dla działania wielu urządzeń, w tym także bazujących na systemach wbudowanych. Dobór odpowiedniego rozwiązania jest jak zawsze podyktowany dążeniem do znalezienia złotego środka pomiędzy parametrami metrologicznymi, kosztami produkcji, rozmiarami, czy też dodatkowymi parametrami, związanymi m.in. z bezpieczeństwem elektrycznym (izolacja galwaniczna) oraz rodzajem współpracującego układu. Zaprezentowane w artykule informacje nie wyczerpują zagadnienia pomiarów amperometrycznych, ale nakreślają obraz obecnego stanu technologii i dają wskazówki do dalszych poszukiwań rozwiązania optymalnego do zastosowania w ramach konkretnej aplikacji.

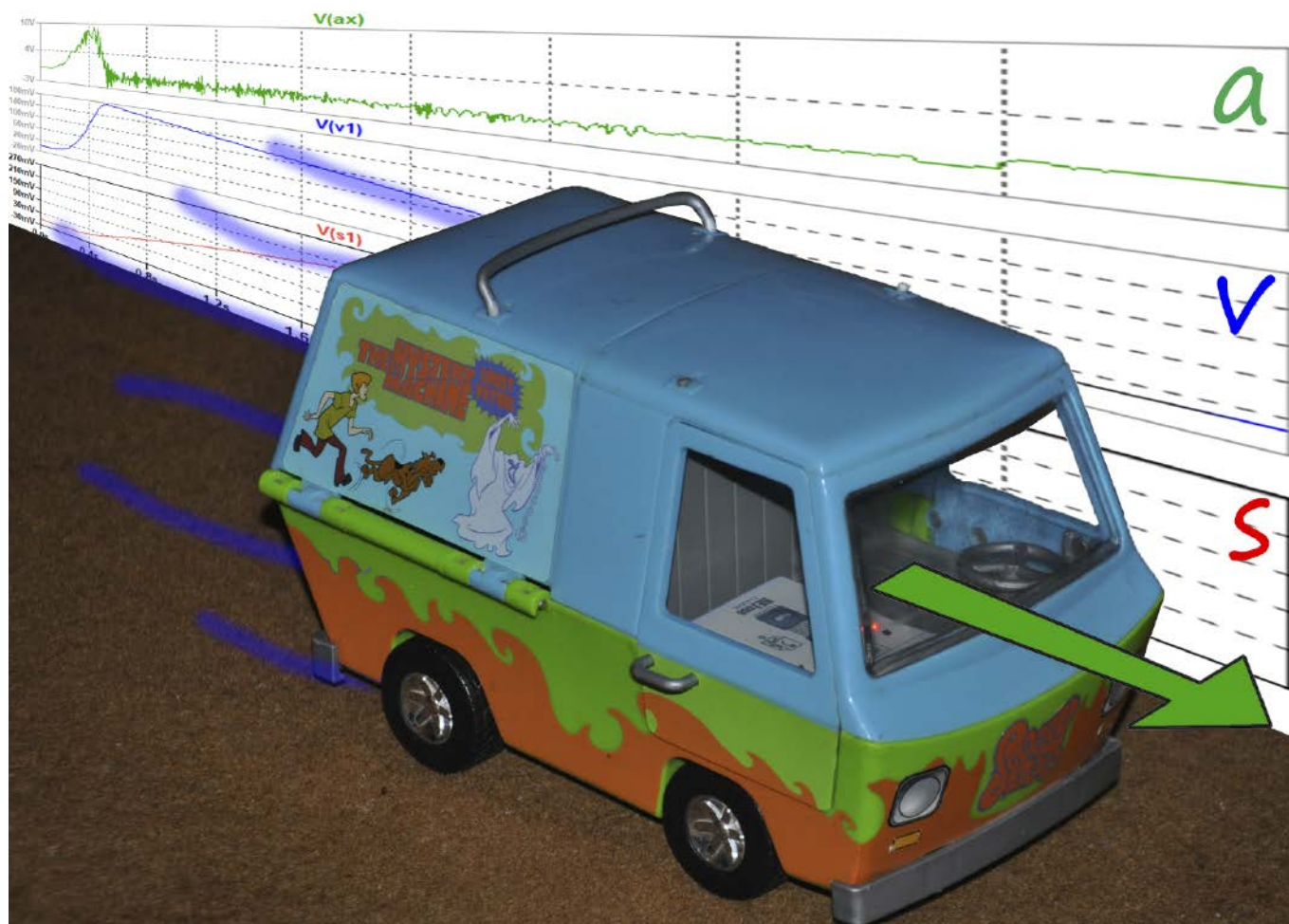
inż. Przemysław Musz, EP
przemyslaw.musz@ep.com.pl

- [1] Dla ścisłości należy dodać, że w tego typu zastosowaniach często wykorzystuje się pomiar pośredni rezystancji poprzez określenie spadku napięcia na mierzonym oporze, wywołanego przez precyzyjne źródło prądowe. Budowa źródeł prądowych często opiera się jednak na sprzężeniu zwrotnym od sygnału z bocznika, czyli... wracamy znów do (pośredniego) pomiaru prądu.
- [2] „Analog Dialogue” 46-06 Back Burner, June (2012).
- [3] Rekomendowany przez autora artykułu footprint to wariant C na rysunku 5.

Chcesz czytać nasze najnowsze artykuły jeszcze przed wydrukowaniem w EP?

Zajrzyj na

www.ep.com.pl/EPwtoku



Symulacja i pomiar, czyli LTspice i Analog Discovery 2 w rękach konstruktora (23)

O czym nie mówiliśmy, a powinniśmy

Niniejszy cykl trwa już bardzo długo. To wystarczający powód do tego, by przeanalizować poruszane zagadnienia. Okazało się, że jest kilka tematów, o których nie było mowy, a z racji tego, że należałoby je zaliczyć do podstawowych, powinny być choćby wspomniane. Nadrabiamy zaległości.

Rozpoczynamy od ważnego zagadnienia, które nie było omawiane wprost, a jest jedną z ważniejszych cech symulatora LTspice.

Hierarchiczna struktura schematów

Hierarchiczna struktura schematów symulatora LTspice jest jedną z najważniejszych cech tego programu. Zwiększa ich czytelność, szczególnie wtedy, gdy są znacznie rozbudowane, pozwala też

definiować własne elementy. Zalety schematów hierarchicznych pokazano na przykładzie. Naszym zadaniem jest symulacja układu elektronicznego selektywnie filtrującego kilka częstotliwości. Użytkownik musi mieć oczywiście możliwość łatwego strojenia filtrów. Problem rozwiązaliśmy, stosując mostki „podwójne T”.

Pierwszym, naturalnym podejściem jest narysowanie kilku takich mostków na jednym schemacie. Ponieważ mamy zapewnić łatwą możliwość strojenia, warto zapewnić parametryczne zadawanie wartości każdego elementu. Jeden mostek zawiera 6 elementów. Gdybyśmy chcieli zastosować np. 4 takie mostki, przestrojenie ich wszystkich wymagałoby zmiany wartości aż 24 elementów. Parametryzacja zredukuję tę liczbę do 8. Przykład kompletnego schematu pokazano na **rysunku 1a**, a wykresy częstotliwościowe na **rysunku 1b**. Parametry elementów każdego filtru umieszczono tak, aby było wyraźnie widoczne, do którego z nich się odnoszą.

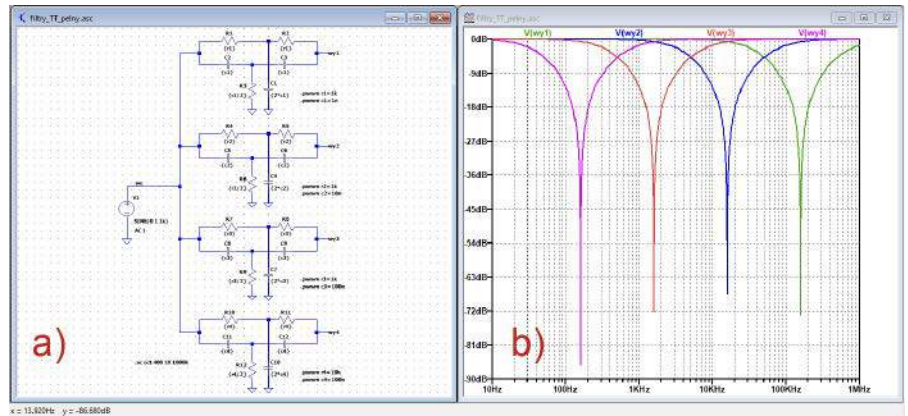
Teraz tę samą symulację przeprowadzimy z zastosowaniem schematu hierarchicznego. Najpierw należy zbudować podstawową strukturę filtra. Rysujemy więc schemat tak, jak to robimy w przypadku każdego symulowanego układu. Nie wprowadzamy jednak konkretnych wartości elementów, a definiujemy je parametrycznie (**rysunek 2**). Ważne jest nadanie etykiet wszystkim węzłom, do których będzie konieczny dostęp ze schematu o wyższym poziomie hierarchicznym. W tym przypadku jest to węzeł wejściowy („we”) i wyjściowy („wy”). Schemat zapisujemy w formacie ASC, a więc takim, jak każdy symulowany układ. Warto ponadto nadać nazwę jednoznacznie kojarzącą się z funkcją układu. Niech to będzie *filtrTT.asc*.

Aby korzystać z tak zaprojektowanego filtra, konieczne jest zdefiniowanie jego oznaczenia schematowego. Otwieramy więc odpowiednie okno edytora: File → New Symbol (**rysunek 3**). Na ekranie ukazuje się pusta plansza z symbolem kotwicy tworzonego elementu. Rysujemy wokół niej prostokąt (Draw → Rect lub przycisk R na klawiaturze) o rozmiarach 4×2 oczka siatki. Kolejną czynnością jest naniesienie wyprowadzeń filtra (Edit → Add Pin Port lub przycisk P), do których doprowadzamy połączenia z prostokątem symbolizującym filtr. Każdy taki element musi mieć swoje oznaczenie (etykieta) i numer porządkowy, które wprowadzamy bezpośrednio po wybraniu powyższej komendy. Dodatkowo możemy decydować o tym, czy etykieta ma być wyświetlona na schemacie, czy nie. Jeśli decydujemy się na wyświetlanie, możemy dodatkowo określić jej położenie względem pinu – opcje TOP, LEFT, RIGHT, DOWN, Vertical Text i Offset (**rysunek 4**). Aby uczynić symbol filtra bardziej czytelnym, w obrębie prostokąta umieściliśmy ponadto grafikę przypominającą charakterystykę filtra TT. Jest ona wykonana przy użyciu polecenia Draw-Line.

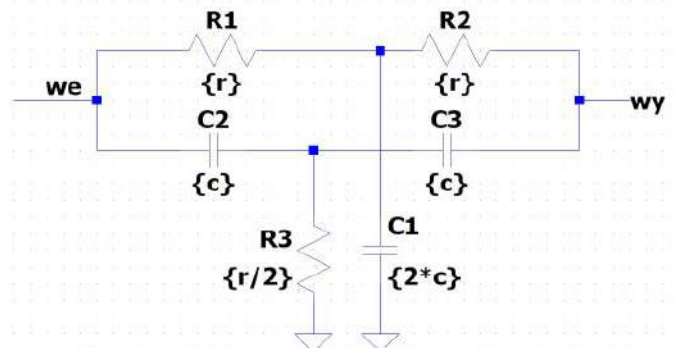
Umieszczanie etykiet z jednej strony ułatwia później, w trakcie tworzenia schematu symulowanego układu, rozpoznawanie funkcji poszczególnych wyprowadzeń, jednak nie mamy możliwości ustawiania wielkości liter i dla niewielkich elementów będą one zaciemniały schemat. W takich przypadkach korzystniejsze jest wyłączenie wyświetlania etykiet – opcja NONE (Not Visible) i zamiast nich wstawienie tekstu komentarza (Draw → Text). Teraz już można wybrać wielkość liter, np.: Font size=0,625. Tak zdefiniowany symbol będzie bardziej czytelny na schemacie.

Pozostaje jeszcze uzyskanie dostępu do parametrów filtra, które powinny być jakoś przekazywane do najwyższego poziomu hierarchii przez symbol schematowy. W tym celu definiujemy domyślne parametry R i C. Przyjmujemy np., że R będzie równe 1 kΩ, a C 10 nF. Przy każdym wprowadzeniu filtra na schemat jego parametry będą określone tak przyjętymi wartościami. Możemy je oczywiście zmienić już na schemacie układu symulowanego w dowolnym momencie. Będzie to potrzebne do strojenia filtra w trakcie symulacji. Parametry nanosimy poprzez atrybut SpaceLine. Wykonujemy polecenie Edit-Attributes, wybieramy opcję Edit Attributes i wypełniamy odpowiednie pole (**rysunek 5**). Warto ponadto uczynić ten atrybut widocznym na schemacie, co następuje po wskazaniu go w oknie atrybutów (Ctrl+W) i umieszczeniu w stosownym miejscu na schemacie. Wynik powyższych prac zapisujemy w pliku o takiej samej nazwie jak schemat, tylko z automatycznie zmienionym rozszerzeniem na ASY (*filtrTT.asy*). Element warto zapisać w domyślnym folderze zawierającym biblioteki elementów. Będzie wówczas zawsze widoczny w programie. Jeśli o to nie zadbamy, może być później problem z dostępem do niego. Aby stał się widoczny, tworzony schemat układu symulowanego powinien być zapisany w folderze, w którym znajduje się plik *filtrTT.asy* lub w którymś z nadrzędnych.

Na tym kończą się prace nad definiowaniem własnego elementu należącego do najniższego poziomu struktury schematu. Możemy przystąpić do symulacji z jego zastosowaniem. Teraz nie musimy rysować każdego filtra, korzystamy z przygotowanego elementu schematowego.



Rysunek 1. Symulacja układu z czterema filtrami selektywnymi: a) schemat, b) charakterystyki częstotliwościowe

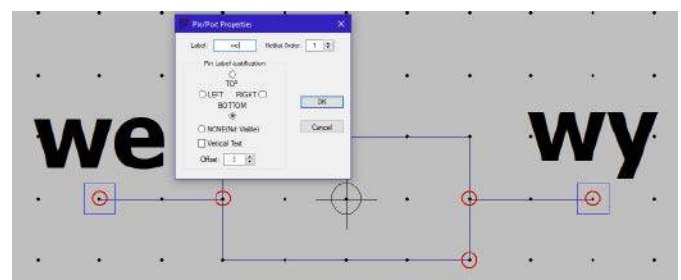


Rysunek 2. Schemat filtra „TT” z parametrycznie określonymi wartościami elementów

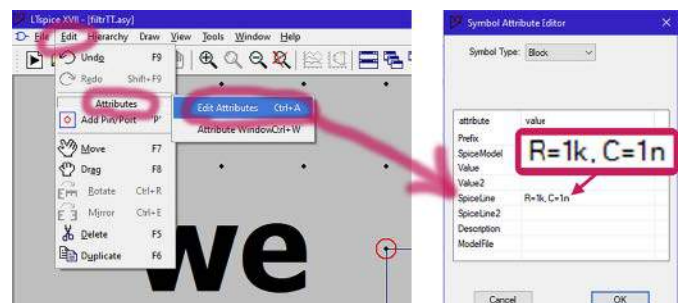
Otwieramy czysty arkusz i umieszczamy na nim symbol filtra (Edit → Component lub przycisk F2). Jeśli elementu nie ma w zestawieniu, wybieramy na widocznym u góry okna liście Top Directory katalog, w którym został zapisany nasz filtr. Powinien się po tym pokazać element filtrTT (**rysunek 6**).



Rysunek 3. Menu z poleceniami otwierającymi arkusz edycji nowego symbolu

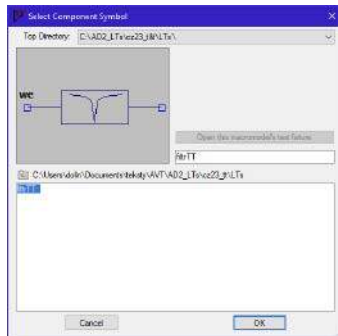


Rysunek 4. Okno właściwości wprowadzającego pinu



Rysunek 5. Okno wprowadzania domyślnych wartości elementów filtra

Wybieramy go i od razu powielamy w czterech egzemplarzach. Następnie stroimy filtry. Można to wykonać na kilka sposobów. Po pierwsze, klikamy na wartości elementów wyświetlane pod prostokątem symbolizującym filtr (**rysunek 7**). Otwiera się po tym okno edycji parametrów, w którym wpisujemy żądane wartości. Druga metoda polega na kliknięciu prawym przyciskiem myszki w obrębie elementu symbolizującego filtr.



Rysunek 6. Okno wyboru elementu bibliotecznego użytkownika

Oprócz możliwości zmiany parametrów filtru można również przeprowadzić edycję jego symbolu schematowego (przycisk ekranowy Open Symbol) oraz schematu filtru (Open Schematic) – **rysunek 8**.

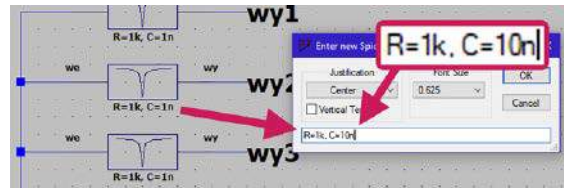
Filtry stroimy dokładnie tak samo jak w pierwszej symulacji, tzn.: $R1=1\text{ k}\Omega$, $C1=1\text{ nF}$ (1. filtr), $R2=1\text{ k}\Omega$, $C2=10\text{ nF}$ (2. filtr), $R3=1\text{ k}\Omega$, $C2=100\text{ nF}$ (3. filtr) i $R4=10\text{ k}\Omega$, $C2=100\text{ nF}$ (4. filtr). Warto zauważyć, że nadal pozostaje możliwość parametrycznego określania wartości elementów tak, jak to zaprezentowano w ostatnim 5. filtrze. Pełny schemat układu filtrującego i wyniki symulacji zostały pokazane na **rysunku 9**.

Funkcje

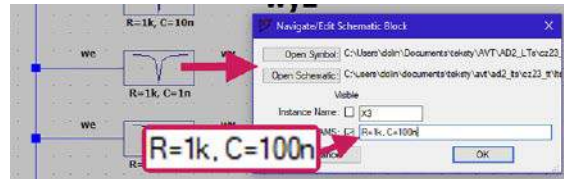
Funkcje były już wprawdzie opisywane w 13. odcinku, w którym omawialiśmy symulację metodą Monte Carlo. Wracamy do nich, ale w nieco innym kontekście. Tym razem, oprócz typowej funkcji definiowanej przez użytkownika, czyli przez nas, zastosujemy jedną z wielu funkcji bibliecznych symulatora LTSpice. Ale po kolei...

Ostatnio zadano mi pytanie: „No dobrze, ale właściwie po co nam symulator?” Poniższy przykład może być odpowiedzią. Wyobraźmy sobie, że budujemy system nawigacji bezwładnościowej oparty na akcelerometrze. Nie jesteśmy pewni ani rozwiązań układowych, ani parametrów technicznych, które należy przyjąć do realizacji projektu. Zamiast budować w ciemno urządzenie i pracować na jego fizycznym modelu, co oczywiście wiąże się z kosztami, może lepiej sprawdzać jego działanie na modelu wirtualnym. Zastosujemy to podejście do projektowanej nawigacji. Nie będziemy tu budować kompletnego urządzenia, sprawdzimy jedynie koncepcję na uproszczonym jednowymiarowym rozwiązaniu. Mamy zatem jednoosiowy akcelerometr ustawiony w kierunku ruchu. Jeżeli nie dysponujemy żadnym fizycznym akcelerometrem, możemy wygenerować jakieś teoretyczne dane wejściowe. Ja akurat miałem takie urządzenie, więc użyłem go do zebrania rzeczywistych danych wejściowych. Problem wózka rozwiązałem, korzystając z samochodzika zabawki. Akcelerometr zainstalowałem w jego kabinie. Wózek rozpędzamy i puszczaemy, pozwalając na wyhamowanie na skutek sił tarcia. Przyspieszenie jest w trakcie eksperymentu rejestrowane, a następnie zapisywane w formacie tekstowym w postaci par liczb: (czas, przyspieszenie). Dane takie można bez trudu zaimportować w symulatorze do źródła napięciowego z funkcją PWL. Zapewniamy w ten sposób połączenie świata rzeczywistego z wirtualnym.

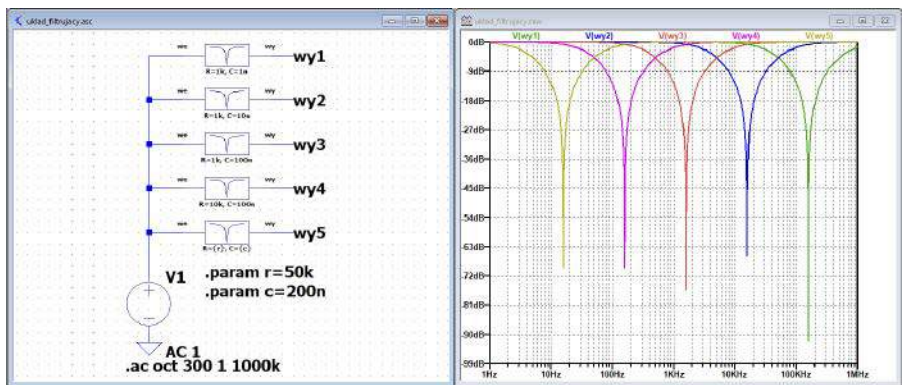
Na schemacie z **rysunku 10** jest to źródło V1, a pochodzący z tego wirtualnego akcelerometru sygnał jest dostępny w węźle opisanym etykietą AX. Po wykonaniu wstępnej symulacji czasowej okazuje się, że sygnał ten jest dość mocno zaszumiony (**rysunek 11**). Widać w nim każdą nierówność podłoża, po którym porusza się wózek. Istnieje więc podejrzenie, że konieczna będzie jakaś filtracja tego sygnału. Ale jak dobrać rząd filtru i jego częstotliwość graniczną? Czy w ogóle ten zabieg faktycznie będzie potrzebny?



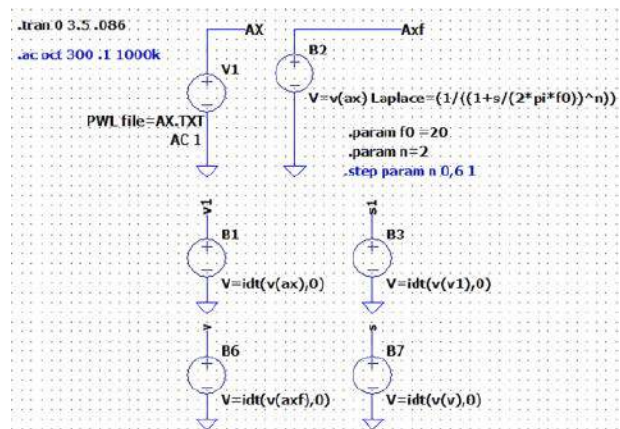
Rysunek 7. Okno wprowadzania parametrów elementu z niższego poziomu hierarchicznego – metoda 1



Rysunek 8. Okno wprowadzania parametrów elementu z niższego poziomu hierarchicznego – metoda 2

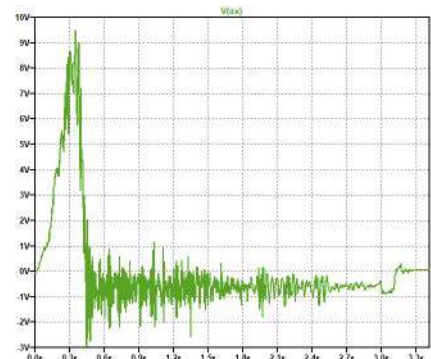


Rysunek 9. Wyniki symulacji układu filtrującego z zastosowaniem schematu hierarchicznego



Rysunek 10. Schemat do sprawdzenia koncepcji układu nawigacji bezwładnościowej

Do rozstrzygnięcia tej wątpliwości zastosujemy model filtru dolnoprzepustowego. Nie będziemy go jednak budować na piechotę z elementów elektronicznych – wzmacniaczy operacyjnych, kondensatorów, rezystorów itp. Skorzystamy z cyfrowego modelu opisanego tylko transmitancją. W symulacji posłużymy



Rysunek 11. Wykres rzeczywistego przyspieszenia zarejestrowanego podczas poruszania się wózka

się behawioralnym źródłem napięciowym B2, którego napięcie jest opisane przekształceniem Laplace'a: $V=v(ax)$
 $Laplace=(1/((1+s/(2\cdot\pi\cdot f_0))^n))$. Zbudowaliśmy w ten sposób filtr dolno-przepustowy filtrujący sygnał z węzła AX. Ma on parametrycznie zadawaną częstotliwość graniczną f_0 , regulowany jest też rząd filtru n .

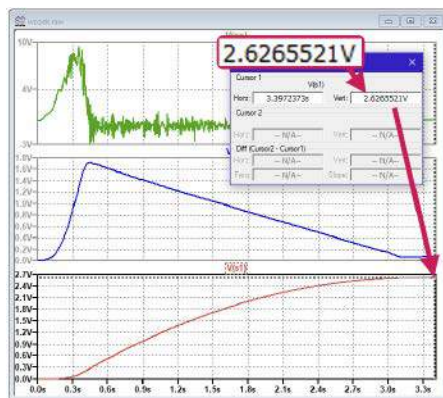
Przed nami najważniejsze zadanie: musimy obliczyć trajektorię wózka. Z fizyki wiemy, że całka z przyspieszenia to prędkość, a całka z prędkości to droga. Powinniśmy więc dwukrotnie scałkować sygnał z akcelerometru. Znowu korzystamy ze źródła behawioralnego. Tym razem napięcie źródła jest obliczane za pomocą funkcji bibliotecznej `idt` liczącej całkę. Ma ona co najmniej jeden parametr, którym jest funkcja podcałkowa. W naszym przypadku do obliczenia prędkości poprzez całkowanie przyspieszenia AX użyjemy źródła B1, a do obliczenia drogi skorzystamy z analogicznego źródła B3 całkującego tym razem prędkość (węzeł „v1”). Sygnał ze źródła B3 opisany etykietą „s1” jest odległością, którą pokonał wózek. Wynik symulacji został pokazany na **rysunku 12**.

Dystans, jaki pokonał wózek, można wyznaczyć, mierząc końcową wartość na wykresie $v(s1)$. Ustawiamy więc kursor na końcu tego wykresu i odczytujemy wynik: 2,63. Jednostka „wolt” wynika oczywiście z zastosowania w symulacji źródła napięciowego. Wyniki z akcelerometru są podawane w m/s^2 , mamy więc proste przełożenie metrów na wolt. Ostatecznie możemy więc stwierdzić, że wózek przejechał 2,63 metra. Pozostaje jeszcze weryfikacja wyniku z rzeczywistym pomiarem odległości i... okazuje się, że zmierzony taśmą dystans to 2,65 m. To wręcz niewiarygodna zgodność... Ciesząc się z takiego wyniku, nie mogę jednak odnieść wrażenia, że to chyba jednak zbieg jakichś wyjątkowo korzystnych okoliczności.

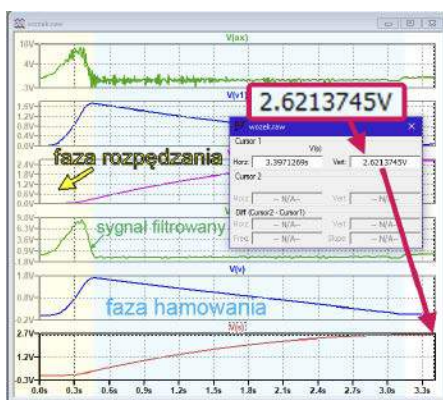
W tej sytuacji wracamy do pytania zadanego na wstępie: czy konieczna jest filtracja sygnału akcelerometru? Trudno przypuszczać, abyśmy uzyskali tą metodą lepszy wynik. Dla formalności jednak zastosujemy filtr. Ustalamy częstotliwość graniczną 20 Hz i rząd filtru równy 2. Wyniki przedstawiono na **rysunku 13**. Wniosek jest dość zaskakujący. Okazuje się, że można zrezygnować z filtrowania sygnału.

Wykresy

Obsługę wykresów powinniśmy już w zasadzie dobrze znać. Jest jednak kilka zagadnień, które warto zebrać w jednym miejscu. Wiemy już, że w jednym oknie wykresu można umieszczać kilka paneli, a każdy z nich może zawierać pogrupowane tematycznie wykresy. Dopuszczalne



Rysunek 12. Wyniki symulacji koncepcyjnego układu nawigacji bezwładnościowej bez filtrowania sygnału



Rysunek 13. Wyniki symulacji koncepcyjnego układu nawigacji bezwładnościowej z zastosowaniem filtra 2. rzędu o częstotliwości granicznej 20 Hz

jest kopiowanie wykresów z poszczególnych węzłów obwodu do kilku, a nawet wszystkich paneli. Panel dodaje się, np. klikając prawym przyciskiem myszki w oknie wykresu, a następnie wybierając polecenie Add Plot Pane. Autorzy programu przyjęli, że nowy panel pojawia się u góry okna. Mnie osobiście bardzo to przeszkadza. Zwykle operacja ta jest wykonywana po to, by wprowadzić jakiś dodatkowy wykres, który nie jest najważniejszy w całej kolekcji, a jako ten mniej ważny powinien znaleźć się – moim zdaniem – na dalszej pozycji. To oczywiście bardzo subiektywna ocena. Szkoda, że nie ma możliwości ustalania jakiejś opcji w parametrach symulatora, która dawałaby pewien manewr w tym zakresie. Tymczasem chyba nie ma możliwości zmiany kolejności paneli, pozostaje więc ręczne przerzucanie wykresów. Czynność jest prosta, ale na skutek różnych reakcji symulatora na naciskanie poszczególnych przycisków, czy to klawiatury komputera lub myszki, należy zachować odpowiednią kolejność ich naciskania.

Aby przenieść dany wykres z jednego panelu do innego, należy najechać kursorem myszki na nazwę wykresu widoczną nad nim, nacisnąć lewy przycisk myszki i trzymając go, przenieść kursor dożądanego panelu. W chwili zwolnienia przycisku wykres zostanie przeniesiony. Czasami chcemy pozostawić oryginał w pierwotnym panelu, a dany wykres tylko skopiować do innego. W tym przypadku, po naciśnięciu lewego przycisku myszki, dodatkowo naciskamy przycisk `Ctrl` na klawiaturze. Powoduje to dodanie znaku plusa kursorze myszki, co wyraźnie sygnalizuje zamiar wykonania kopii wykresu.

Co zrobić, jeśli chcemy usunąć jakiś wykres, a nawet cały panel? Do kasowania wszelkich elementów przewidziano użycie przycisku `F5`. Jego naciśnięcie powoduje zmianę wyglądu kursora myszki, który przybiera teraz postać nożyczek. Naprowadzenie kursora na nazwę wykresu (w górnej części panelu) i kliknięcie lewym przyciskiem myszki powoduje usunięcie wykresu. Cały panel wraz z jego zawartością usuwamy, klikając prawym przyciskiem myszki w obszarze wykresu, a następnie wybierając komendę Delete this Pane. Na szczęście operacja jest odwracalna dzięki komendzie Undo (ikonka zawiniętej strzałki w lewo na pasku narzędziowym).

Czasami konieczne jest powiększenie jakiegoś fragmentu wykresu. Intuicyjnie korzystamy z ikonki lupy ze znakiem `+`. Możemy wówczas otoczyć interesujący nas fragment wykresu, korzystając z lewego przycisku myszki. Bardzo często, szczególnie wtedy, gdy wycinamy jakiś wąski przedział czasowy, oś pionowa ulega przeskalowaniu. W rezultacie wykres wygląda tak, jakby uległ spłaszczeniu, a więc skala pionowa nie jest wykorzystywana w pełni. Do optymalnego rozciągnięcia wykresu w pionie służy ikonka z wizerunkiem wykresu z dwiema strzałkami pionowymi (**rysunek 14**). Przydatną, chociaż nieco dziwnie działającą czynnością jest przesuwanie wykresu w dowolnym kierunku wewnątrz panelu. W tym celu należy naprowadzić kursor myszki na wykres, nacisnąć przycisk `Ctrl` i poruszając myszką, ustawić wykres w żądanym położeniu. Po zwolnieniu przycisku wykres jest jednak odświeżany z minimalną zmianą położenia. Jest to efekt niepożądany i nieco irytujący.

Symulując układy z dużą liczbą elementów i połączeń, często chcemy sprawdzić wzajemne zależności między poszczególnymi sygnałami. Tworzymy więc wykresy z kilkoma panelami, a w każdym umieszczamy po kilka przebiegów. Coraz trudniej przychodzi nam orientowanie się, który wykres należy do którego węzła obwodu. Można sobie radzić, nadając etykiety poszczególnym sygnałom. Bywa, że schemat jest tak gęsty, że nie ma już miejsca na etykietę, albo traci on na czytelności z powodu nadmiaru symboli. W takich przypadkach dobrym trikiem jest naprowadzenie kursora myszy na nazwę interesującego nas wykresu (widoczną nad panelem), naciśnięcie lewego `Alt` i kliknięcie lewym przyciskiem myszki. Wskazany węzeł lub element zostaje podświetlony (standardowo na żółto) – **rysunek 15**.

I jeszcze jeden trik związany z wykresami i opcjonalnymi pomiarami, które można za ich pomocą wykonać. I tym razem korzystamy z lewego przycisku `Alt`, po naciśnięciu którego wskazujemy jakiś element na schemacie i klikamy lewym przyciskiem myszki. Pojawia się wówczas dodatkowy wykres mocy wydzielanej na tym elemencie. Jest to wykres

czasowy, a więc widzimy wartości chwilowe tej mocy. Po naciśnięciu *Ctrl* i kliknięciu na nazwę wykresu pojawia się okno, w którym jest wyświetlana moc średnia oraz energia wydzielona w elemencie w przedziale czasowym obejmowanym przez wykres.

Na zakończenie rozważań na temat wykresów warto wspomnieć o możliwości niezależnego skalowania osi czasu w poszczególnych panelach. Zwykle chcemy, aby wszystkie wykresy były wyświetlane współbieżnie. Pozwala to obserwować wzajemne powiązania między poszczególnymi sygnałami. Są jednak wyjątkowe sytuacje, których niezależne skalowanie może być przydatne. Przykładem niech będzie multiwibrator przedstawiony na schemacie z **rysunku 16**. Generuje on przebieg prostokątny dostępny na kolektorach obu tranzystorów. Można go obejrzeć w dłuższym czasie, co pozwoli np. przeprowadzić analizę widmową. Jednak w układzie można dostrzec stan nieustalony w chwili jego włączania, któremu warto przyjrzeć się z bliska, nie tracąc przy tym wglądu w całość. Zna-

ne już nam zoomowanie wykresu za pomocą lupy obejmuje wszystkie wykresy, we wszystkich panelach. Korzystając z tego narzędzia, utracilibyśmy zatem możliwość podglądu całego przebiegu. Aby na jednym wykresie wyświetlić wybrany przebieg w powiększeniu, musimy kliknąć prawym przyciskiem myszki na wykresie i wyłączyć opcję *Sync.Horiz.Axes*. Następnie otwieramy nowy panel i umieszczamy w nim kopię przebiegu wybranego do powiększenia. Teraz klikając na nim prawym przyciskiem myszki, wybieramy komendę: *View → Manual Limits*, i w polach *Horizontal Axis* wprowadzamy odpowiednie parametry czasowe, które zdefiniują nam okno powiększenia. Niestety, od tego momentu nie jest możliwe stosowanie narzędzia „lupa”.

Nieudokumentowane polecenia LTspice

Takie nieudokumentowane polecenia to prawdziwy kłosek dla użytkowników. Znając je i umiejętnie wykorzystując, stawiamy się – to nic, że we własnym mniemaniu – na pozycji wybitnych fachowców od tematu. Potrafimy przecież robić to, o czym inni użytkownicy nie mają pojęcia.

Kilka takich sztuczek było już użytych w niniejszym cyklu, choć nie było to specjalnie nagłaśniane. Okazuje się, że takich nieudokumentowanych poleceń jest dość dużo. Ciekawe, dlaczego autorzy programu nie zawarli ich w opisie programu? Może nie dają gwarancji na poprawne ich działanie? Kto wie, może pod jakąś kombinacją klawiszy ukryty jest jakiś symulator lotu, jak to było w jednej z pierwszych wersji Excela. Takie polecenia są mniej lub bardziej potrzebne, ale warto je znać. Wymienimy więc tylko niektóre.

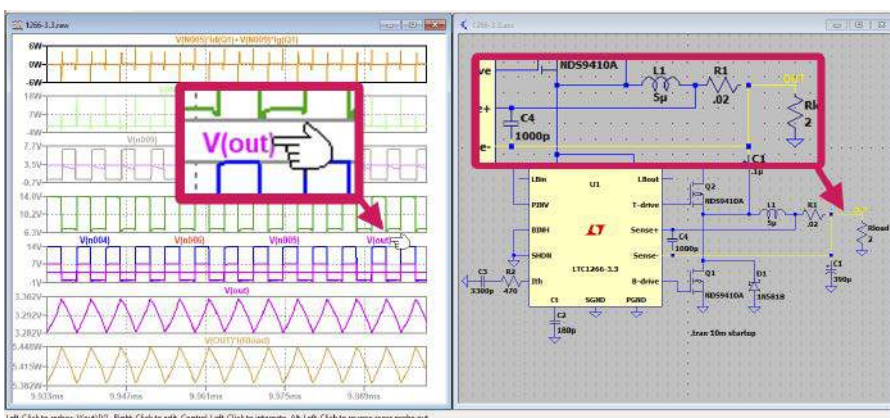
1. Etykieta z negacją. To polecenie jest szczególnie przydatne podczas symulacji układów cyfrowych. Bardzo często występują w nich sygnały proste i zanegowane. Sygnały zanegowane są oznaczane zwykle nadkreślonym napisem tak, jak to pokazano na **rysunku 17**. Efekt taki jest uzyskiwany przez dodanie znaku podkreślenia przed tekstem etykiety.
2. Wyświetlanie ukrytych parametrów elementów. Elementy takie, jak kondensatory czy cewki mają szereg dodatkowych parametrów, które nie są standardowo wyświetlane na schemacie, aby nie zmniejszać jego czytelności. Na przykład dla kondensatora są to: pojemność (to oczywiste), napięcie znamionowe, prąd znamionowy, ekwiwalentna rezystancja szeregową (ESR), ekwiwalentna indukcyjność szeregową, ekwiwalentna rezystancja równoległa i ekwiwalentna

pojemność równoległa. Parametry te można sprawdzić, wyświetlając właściwości elementu (kliknięcie prawym przyciskiem myszki na elemencie). Wyświetlana jest wówczas tabelka z parametrami. Podobne informacje można uzyskać, naprowadzając kursor myszki na dany kondensator i naciskając jednocześnie przyciski *Shift-Ctrl-Alt-H*. Wszystkie zdefiniowane dla tego elementu ukryte parametry są wyświetlone na schemacie standardowo żółtą czcionką w pobliżu oznaczenia schematowego (**rysunek 18**). Domyślny, żółty kolor nie jest najlepszym rozwiązaniem, tym bardziej że zastosowano najmniejszą czcionkę.

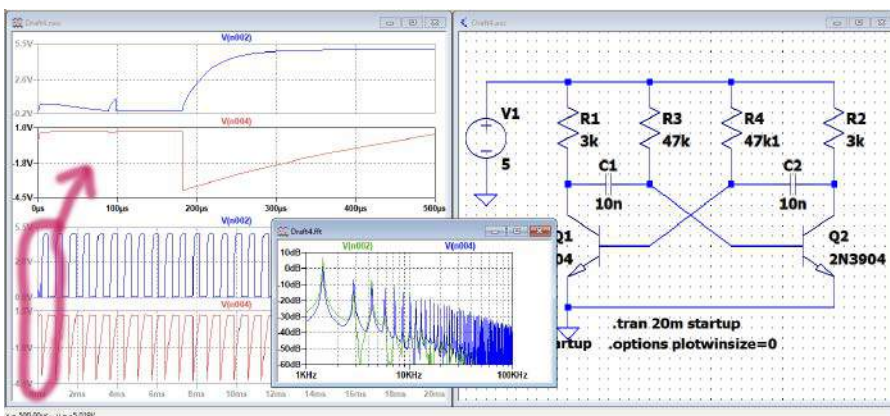
3. Generowanie przebiegów definiowanych numerycznie w pętli skończonej lub nieskończonej. Z tej cechy już korzystaliśmy, ale warto



Rysunek 14. Ikona rozciągania wykresu w pionie

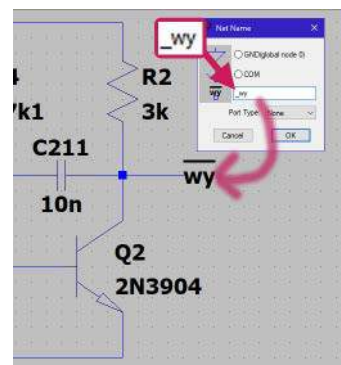


Rysunek 15. Ilustracja metody podświetlania węzła na schemacie, do którego odnosi się wskazany przebieg na wykresach



Rysunek 16. Przykład stosowania niezależnego skalowania osi czasu w różnych panelach na wykresie

ją przytoczyć jeszcze raz tutaj. Jak wiemy, użytkownik może wygenerować niemal dowolny własny przebieg, korzystając np. ze źródła napięciowego z opcją *PWL*. Dane numeryczne przebiegu są zapisywane w wierszach zawierających parametry **czas**, **napięcie**. Dane te, jeśli jest ich niewiele i można je wprowadzić ręcznie, najwygodniej jest zapisać w wewnętrznej tablicy symulatora. Gdy robi



Rysunek 17. Ilustracja metody wprowadzania etykiet ze znakiem negacji

się ich zbyt dużo, można je zachowywać w plikach dyskowych w formacie tekstowym.

W poniższym przykładzie zapisano w ten sposób krótkie sygnały sinusoidalne, które wykorzystamy do generowania tekstu zapisanego alfabetem Morse'a. Kropka trwa 150 ms, kreska 450 ms, odstęp między elementami znaku trwa 150 ms, a odstęp między znakami trwa 450 ms. Do generowania telegramu zastosujemy źródło napięciowe V1 z opcją PWL. Argumentami będą pliki z zapisanymi kropkami i kreskami połączone za pomocą nieudokumentowanej opcji **REPEAT ENDREPEAT**. Naszym zadaniem jest nadanie komunikatu „AVT” za pomocą dźwiękowych znaków Morse'a. Dla sprawdzenia efektu końcowego zastosujemy komendę `.wave avt.wav 16 44.1k v(wy)` zapisującą efekt pracy naszego automatycznego klucza telegraficznego w pliku dźwiękowym *avt.wav*. Definicja sygnału generowanego przez źródło V1 jest następująca:

```
PWL repeat for 1(file=kropka.txt)endrepeat repeat for 1(file=kreska.txt)endrepeat repeat for 1(file=kropka.txt)endrepeat +450m 0 repeat for 3(file=kropka.txt)endrepeat repeat for 1(file=kreska.txt)endrepeat +450m 0 repeat for 1(file=kreska.txt)endrepeat
```

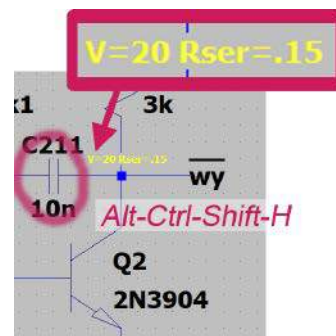
Jak widać, w powyższym zapisie łączono są różne techniki określania parametrów. Opcje **REPEAT** mogą być ponadto zagnieżdżane. Na **rysunku 19** pokazano wygenerowany przebieg w całości i jeden powiększony jego fragment. Komunikat można odsłuchać z pliku *avt.wav*.

4. Renumeracja elementów. Niezwykle przydatna funkcja dla użytkowników lubiących porządek na schemacie. W trakcie projektowania często usuwamy ze schematu różne elementy, a zamiast nich wstawiamy inne. LTspice automatycznie numeruje każdy wstawiany do schematu element, nadając mu kolejny numer. Przy wielokrotnie powtarzanej operacji zamiany elementów tworzą się więc dziury w numeracji. Naciskając kombinację przycisków **Shift-Ctrl-Alt-R**, wymuszamy renumerację elementów od zera. Prosty zabieg, a jakże przydatny.
5. Symulacja wymiany typów elementów. Założmy, że projektujemy wzmacniacz tranzystorowy. Punkty pracy wyznaczone, napięcia się zgadzają, ale parametry częstotliwościowe zwykle nie są łatwe do obliczenia. Wymagają wielu obliczeń, a i tak trudno uwzględnić wszystkie elementy, które mogą wpływać na charakterystykę częstotliwościową. Jedna mała pojemność złącza kolektor-baza zastosowanego tranzystora może spowodować, że cały projekt nadaje się do kosza. Ratunkiem jest symulacja metodą prób i błędów. Po prostu bierzemy kolejne typy tranzystorów i patrzymy, który z nich nadaje się do projektu, a który nie. Wspaniale byłoby móc porównywać np. charakterystyki układu z poszczególnymi typami tranzystorów, ale trzeba by tworzyć kilka niemal identycznych schematów różniących się jedynie typami tranzystorów. Jest prostsza metoda polegająca na zastosowaniu polecenia `.step param`, którego argumentem byłby typ tranzystora. Problem jednak polega na tym, że polecenie `param` nie dopuszcza użycia nienumerycznych argumentów. Na szczęście w symulacjach dozwolone jest używanie aliasów. Będziemy więc w nich ukrywać alfanumeryczne oznaczenia elementów. Alias poprzedzamy tekstem „ako:”, po którym następuje biblioteczny typ tranzystora. W przykładzie użyjemy tranzystorów:

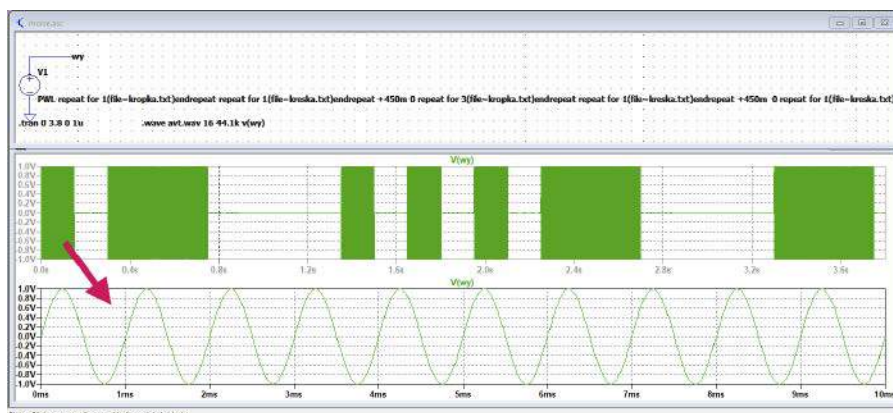
2N3904, MAT02 (oznaczenie w bibliotece MAT-2 – jest to para tranzystorów w jednej obudowie) oraz tranzystorów BC547B i BC547C. Zaczynamy od przygotowania aliasów:

```
.model 3904 ako:2N3904
.model 2 ako:MAT-02
.model 5472 ako:BC547B
.model 5473 ako:BC547C
```

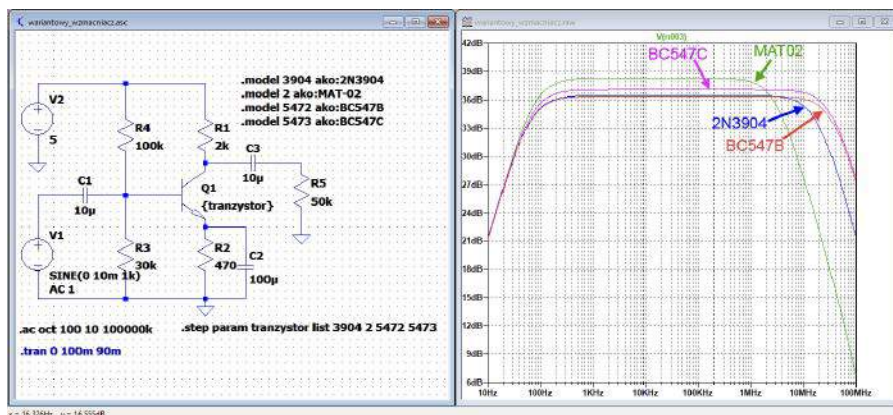
Nadajemy parametryczny typ tranzystora na schemacie, np.:



Rysunek 18. Wyświetlanie ukrytych parametrów elementów



Rysunek 19. Przebieg generowany przez automatyczny klucz telegrafu Morse'a



Rysunek 20. Wariantowe charakterystyki wzmacniacza zależne od zastosowanych typów tranzystorów

{tranzystor}. Musimy jeszcze wstawić odpowiednio zredagowane polecenie wymiany tranzystorów:

```
.step param tranzystor list 3904 2 5472 5473
```

Wszystko jest gotowe, rozpoczynamy symulację. Zbadamy charakterystykę częstotliwościową naszego wariantowego wzmacniacza. Jest ona pokazana wraz ze schematem na **rysunku 20**. Wszystko stało się jasne. Musimy tylko zdecydować, czy bardziej nam zależy na wzmocnieniu, czy na paśmie.

Czas leci...

Ten odcinek ukazuje się dokładnie po dwóch latach od zapowiedzi cyklu. To czas, po którym może już występować „zmęczenie materiału”. Na temat symulatora LTspice można by napisać jeszcze wiele, ale podjąłem decyzję o zakończeniu kursu. Ostatni odcinek ukazuje się za miesiąc. Nie wykluczam powrotu do pewnych tematów, jeśli uznaję, że są warte opublikowania. Będą to jednak już tylko pojedyncze artykuły.

Jarostaw Doliński, EP



Praktyczne aplikacje scalonych układów AFE (4)

Front-endy EKG do urządzeń wearable

W poprzedniej części Poradnika Implementacji szczegółowo omówiliśmy układy AFE przeznaczone do implementacji w torach wejściowych klinicznych urządzeń elektrokardiograficznych i elektroencefalograficznych. Tym razem zajmiemy się przykładami miniaturowych układów, których konstrukcja i parametry umożliwiają zastosowanie w nieco mniej wymagających (pod względem funkcjonalnym) urządzeniach typu wearable.

Zaprezentowane do tej pory front-endy są dedykowane przede wszystkim do poważnych aplikacji medycznych, choć najprostsze, 16-bitowe wersje AFE mogą być także z powodzeniem stosowane w mniej wymagających aplikacjach monitorów sportowych (jak już wspomnieliśmy przy okazji opisu rodziny ADS119x). Nie sposób nie wspomnieć jednak o jeszcze prostszych układach wejściowych, których konstrukcja jest dostosowana niemal wyłącznie do zastosowań typu fitness, takich jak pasy piersiowe i inne urządzenia wearable. Konstrukcja tych układów jest bowiem silnie zoptymalizowana odnośnie do poboru mocy – o ile ich parametry pomiarowe przeważnie (choć nie zawsze!) znacznie ustępują większym front-endom biosygnalowym, o tyle pod względem energooszczędności (a także wymiarów obudowy) zazwyczaj to właśnie prostsze układy przodują w projektach małych urządzeń mobilnych.

AD8232 i AD8233

Układy AD8232 i AD8233 wyraźnie odróżniają się od swoich „większych kolegów” nie tylko rozmiarami, ale przede wszystkim

konstrukcją i stopniem złożoności. Największą różnicą, która rzuca się w oczy po pierwszym spojrzeniu na schematy blokowe tych układów (rysunki 1 i 2), jest brak... przetwornika ADC. Oba produkty są bowiem w istocie specjalizowanymi wzmacniaczami pomiarowymi, przestrajanymi za pomocą zaledwie kilku wejść cyfrowych i obudowanymi kilkoma dodatkowymi blokami: RLD, układem detekcji impulsów rozrusznika oraz... uniwersalnym wzmacniaczem operacyjnym do dowolnego wykorzystania przez użytkownika. Pomędzy wersjami AD8232 i AD8233 istnieje jednak kilka zasadniczych różnic, zestawionych w tabeli 1.

Obydwa układy są wyposażone w sterowany za pomocą osobnej linii wejściowej (FR) układ szybkiego powrotu (fast restore), zmieniający charakterystykę czasową filtra górnoprzepustowego. Układ ten jest konieczny, aby urządzenie bazujące na tak prostym AFE mogło poradzić sobie z gwałtownymi odchyleniami izoliny, powstającymi m.in. w wyniku artefaktów ruchowych. Problem ten nie jest krytyczny w układach wyposażonych w przetworniki analogowo-cyfrowe o wysokiej rozdzielczości, pozwalającej na cyfrowe usunięcie dryfu izoliny bez szkody dla sygnału różnicowego (użytecznego). Konstruktorzy układów AD8232/33 przewidzieli jednak, że będą one implementowane w urządzeniach wyposażonych w przetworniki ADC o stosunkowo małej liczbie bitów (co jest typowe w znakomitej większości współczesnych mikrokontrolerów) – stąd taka, a nie inna metoda wirtualnego zwiększenia zakresu dynamiki. Układ fast restore opiera swoje działanie na komparatorze okienkowym, który po przekroczeniu progów przełączania (po wykroczeniu sygnału wyjściowego wzmacniacza pomiarowego IA poza użyteczny zakres

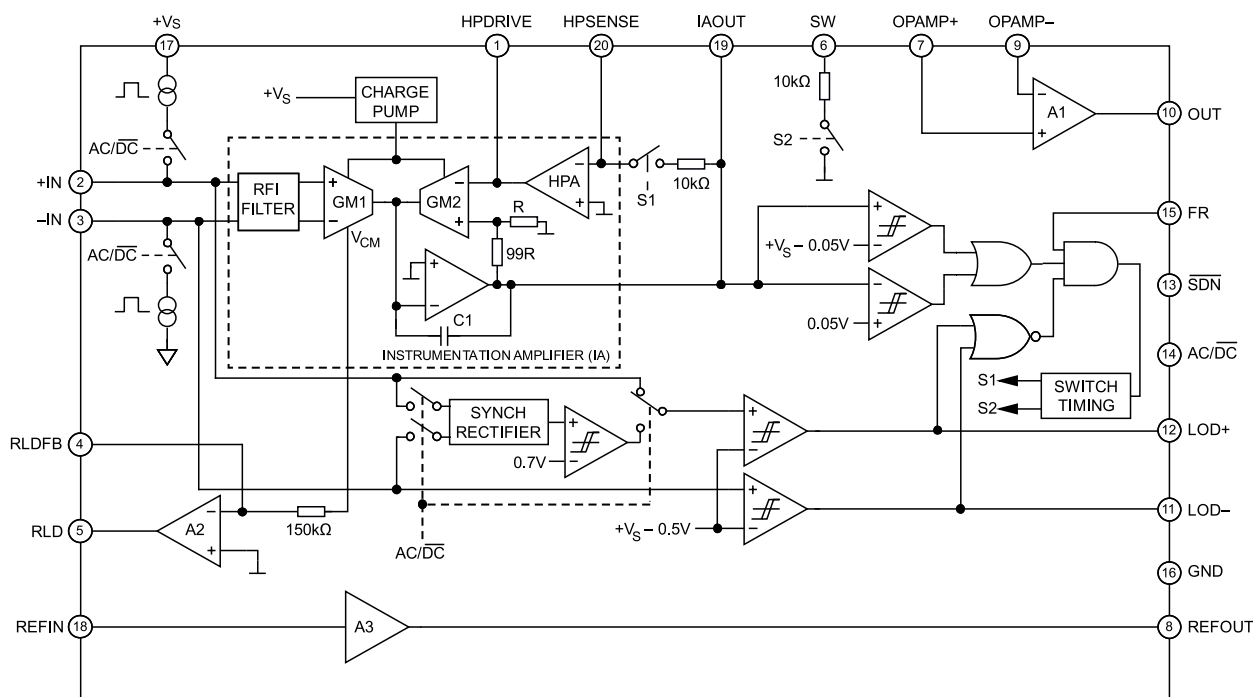
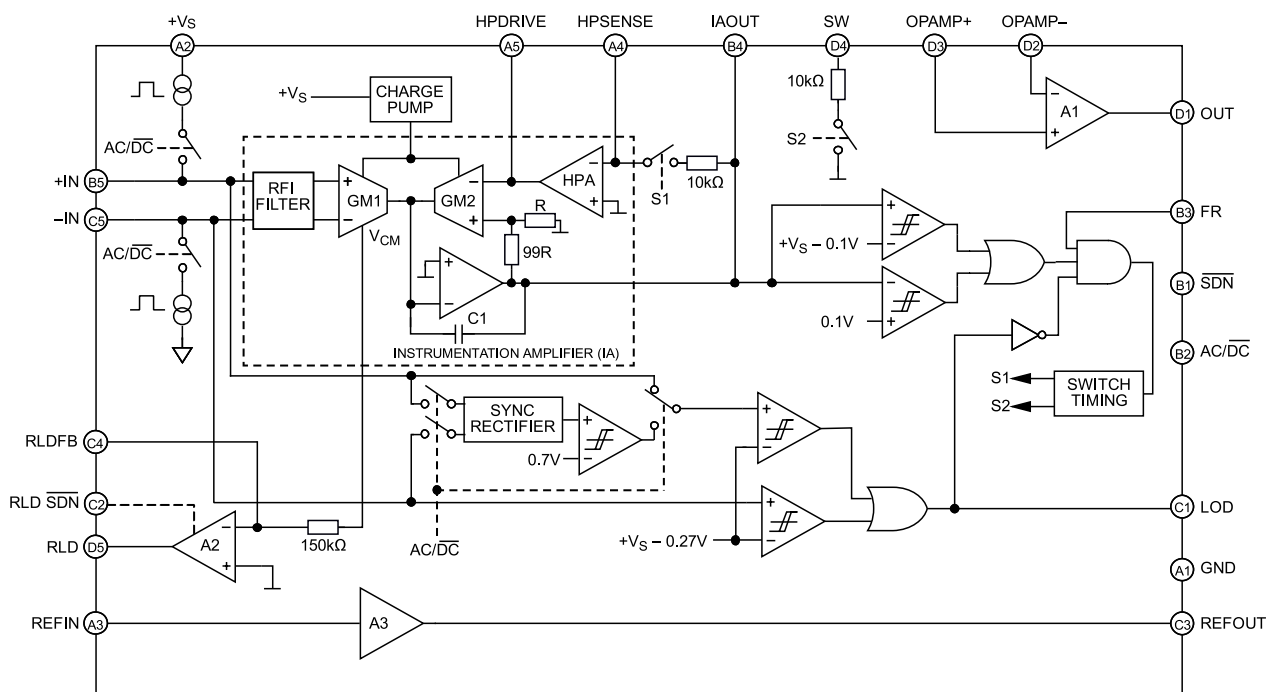
Tabela 1. Porównanie parametrów układów AD8232 i AD8233

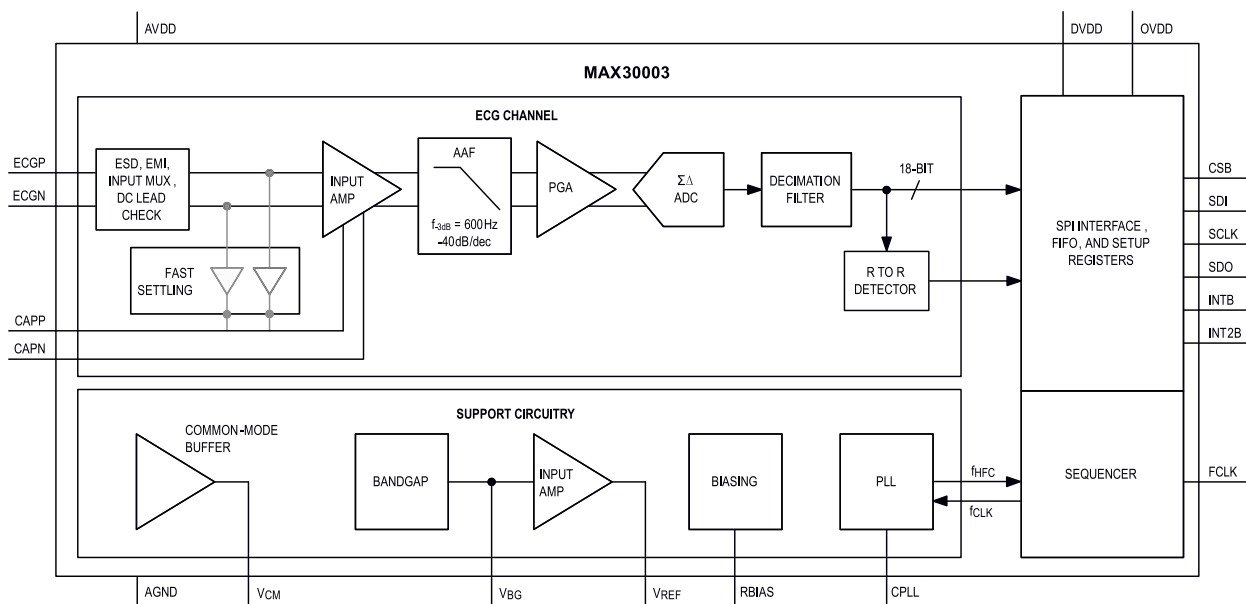
Parametr	AD8232	AD8233
Pobór prądu [μA]	170	50
Szum wejściowy @ BW=0,5..40 Hz [μVpp]	14	8,5
Obsługa detekcji odłączenia elektrod w trybie shutdown	-	x
Osobne wejście shutdown dla wzmacniacza RLD	-	x
Rozdzielone monitorowanie odłączenia elektrod +IN i -IN	x	-
Obudowa	LFCSP20 (4×4 mm)	WLCSP20 (2×1,7 mm)

dynamiki) łączy dwa klucze analogowe S1 i S2, skracające czas odpowiedzi filtra górnoprzepustowego (HPA).

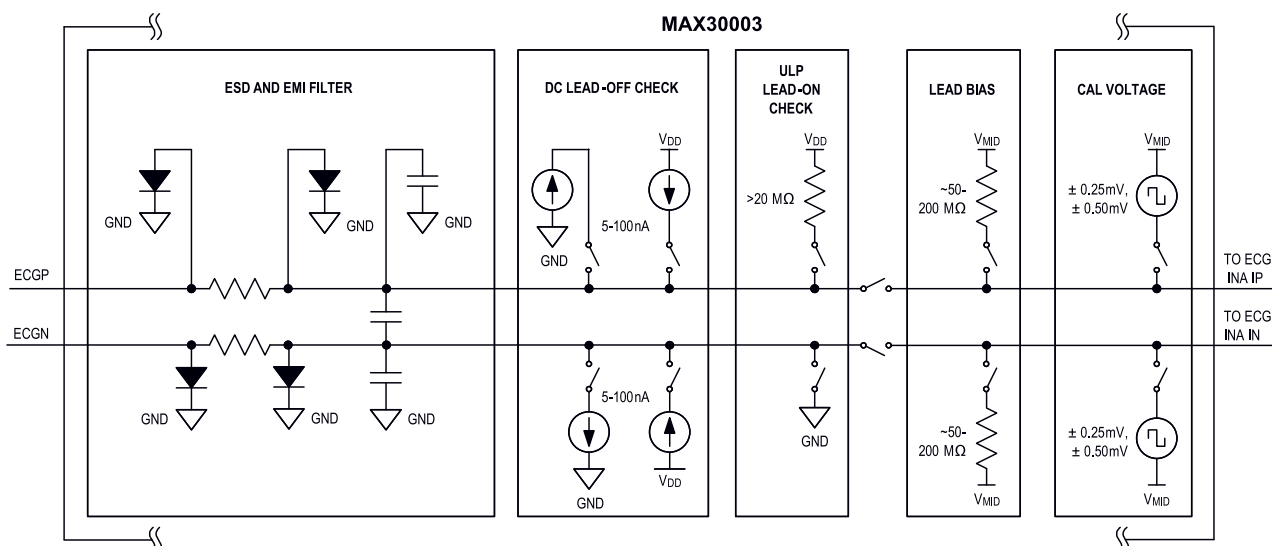
Interesujące rozwiązanie, o którym warto wspomnieć, zastosował producent także w przypadku samego wzmacniacza instrumentalnego. W przeciwieństwie do typowo „napięciowych” topologii,

opisanych pokrótce w pierwszej części cyklu (EP 3/21), w układach AD8232/33 zastosowano... dwa wzmacniacze transkonduktancyjne, których wyjścia podłączone są do integratora odwracającego z kondensatorem C1. Wynikowe wzmocnienie składowej zmiennej sygnału wejściowego wynosi 100 V/V, zaś kompletna wolnozmienna


Rysunek 1. Schemat funkcjonalny układu AD8232

Rysunek 2. Schemat funkcjonalny układu AD8233



Rysunek 3. Schemat funkcjonalny układu MAX30003



Rysunek 4. Schemat układu przełączników kanału wejściowego MAX30003

jest sukcesywnie kompensowana przez układ blokowania składowej stałej.

MAX30003

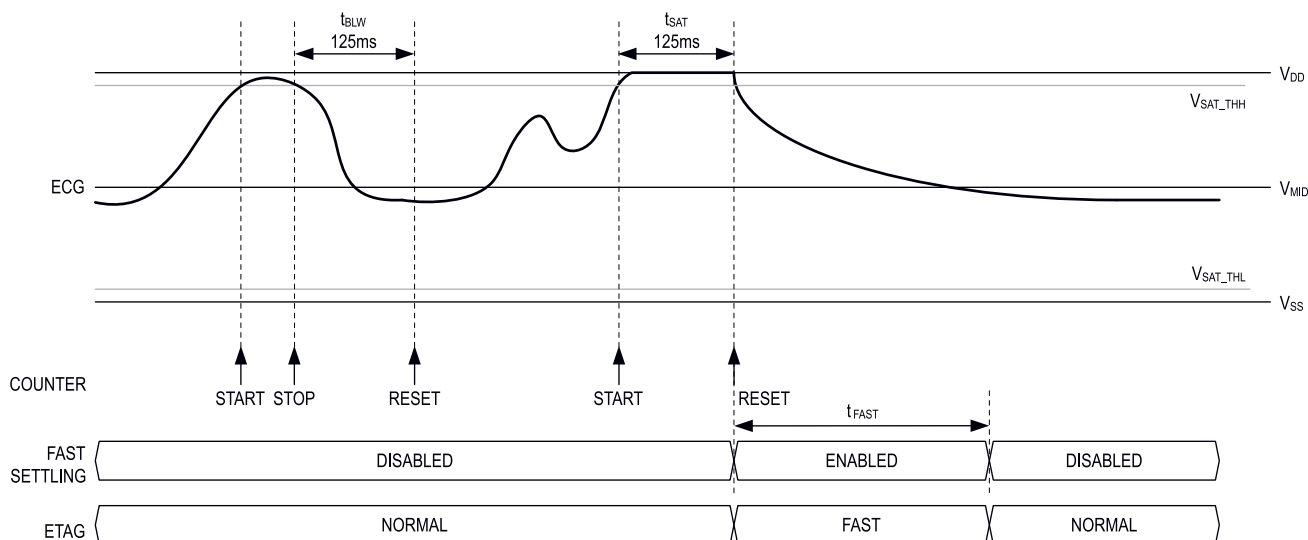
Firma Maxim, świadoma potencjału, jaki niosą ze sobą nowoczesne aplikacje medycznej elektroniki wearable, opracowała własną odpowiedź na AFE konkurentów. Układ MAX30003 (**rysunek 3**) należy do coraz liczniejszej rodziny front-endów medycznych, która zasłynęła m.in. dzięki zintegrowanym, hybrydowym czujnikom pulsoksymetrycznym z wbudowaną optyką (omówimy je szerzej w kolejnym odcinku niniejszego cyklu). MAX30003 jest pozycjonowany jako produkt klasy klinicznej (clinical grade), co oznacza, że firma Maxim dołożyła starań, by produkt mógł spełniać nietatwe wymagania aparatury medycznej. Najważniejsze parametry układu to:

- liczba kanałów wejściowych (odprowadzeń): 1,
- CMRR >100 dB,
- rozdzielczość natywna/efektywna: 18 bitów/15,5 bitu,
- poziom szumów wejściowych: 5,4 μVpp (0,82 μVrms) @BW=150 Hz,
- impedancja wejściowa: >500 M Ω ,
- dopuszczalny offset DC: ± 650 mV,
- zakres dynamiki (AC): 65 mVpp,
- zasilanie 85 μW @ $V_{cc}=1,1$ V (!),

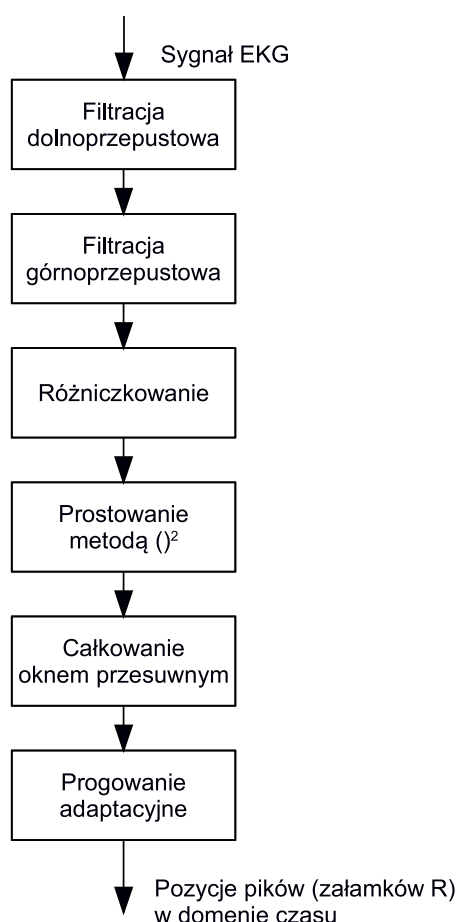
- wbudowana pamięć wyników (FIFO, 32 słowa),
- obudowa: TQFN28/WLP30.

Układ ma parę wejść CAPP/CAPN, służących do podłączenia zewnętrznego kondensatora dla filtru górnoprzepustowego, a także dwa wyjścia cyfrowe (INTB, INT2B) udostępniające przerwania dla nadrzędnego mikrokontrolera. Układ wejściowy (**rysunek 4**) zawiera podstawowe filtry EMI w topologii różnicowo/wspólnej (więcej na ten temat w poprzedniej części cyklu) i podwójne zabezpieczenia ESD. Co ciekawe, producent zdecydował się na zastosowanie podwójnego układu detekcji odłączenia/dołączenia elektrod, z którego jeden pracuje w trybie normalnym, zaś drugi w trybie obniżonego poboru mocy (ULP). Cyfrowo przełączane rezystory (lead bias) pomagają ustalić właściwy punkt pracy (napięcie wspólne) kanału wejściowego – użytkownik może wybrać elementy o wartości 50, 100 lub 200 M Ω .

Podobnie jak w przypadku omówionych wcześniej układów AD8232 i AD8233, także front-end MAX30003 zawiera sprzętowe wsparcie dla przyspieszania powrotu izoliny po wystąpieniu silnych zakłóceń (artefaktów ruchowych, impulsów z urządzeń elektrochirurgicznych bądź defibrylatora). Zasadę działania tej funkcji przedstawiono na **rysunku 5** – wejście wzmacniacza PGA w stan saturacji uruchamia timer sprzętowy o okresie przepełnienia równym 125 ms. Jeżeli sygnał nie wróci przez ten czas do poprawnego zakresu, częstotliwość progowa filtra górnoprzepustowego jest tymczasowo podwyższana,



Rysunek 5. Zasada działania funkcji fast recovery układu MAX30003



Rysunek 6. Schemat blokowy algorytmu Pan-Tompkins

dzięki czemu napięcie szybciej powraca do właściwego zakresu. Użytkownik może także programowo wymusić zadziałanie funkcji fast recovery w razie potrzeby.

Prawdopodobnie najciekawszą funkcją układu MAX30003, której próżno szukać w innych front-endach elektrokardiograficznych spośród opisanych wcześniej produktów, jest blok sprzętowego wsparcia obliczeń odstępu R-R. Funkcja bazuje na znanym w świecie informatyki medycznej algorytmie Pan-Tompkins (rysunek 6), a jej implementacja w strukturze AFE znakomicie ułatwia tworzenie aplikacji wskazujących częstość rytmu serca (np. pulsometry sportowe) czy też zmienność tegoż rytmu (tzw. HRV – heart rate variability).

W tym miejscu warto jeszcze wspomnieć o ciekawym zestawie ewaluacyjnym, przygotowanym przez inżynierów firmy Maxim na bazie


Fotografia 1. Wygląd noszonego monitora EKG MAX-ECG-MONITOR (źródło: <https://bit.ly/2QHhZP>)

front-endu MAX30003. MAX-ECG-MONITOR to niewielkie urządzenie w postaci samoprzylepnej łatki (patch) o średnicy 35,6 mm i grubości 8 mm, zasilane baterią CR2025 i komunikujące się z dedykowaną aplikacją mobilną za pośrednictwem łącza Bluetooth Smart. Dodatkowo gadżet ma także wbudowany czujnik inercyjny MEMS i termometr. Wygląd urządzenia został pokazany na fotografii 1.

Podsumowanie

Układy AFE przeznaczone do celów akwizycji biosygnali na dobre wrosły już w rynek elektroniki medycznej. W tym artykule staraliśmy się możliwie szczegółowo przedstawić niuanse implementacyjne oraz podobieństwa i różnice pomiędzy różnymi modelami układów, dostępnych obecnie na rynku. Jak zawsze, za wyborem konkretnego front-endu do określonej aplikacji (czy to urządzenia klasy klinicznej, czy też wearable pulsometru dla sportowców amatorów) musi stać dogłębna analiza założeń projektowych, wymaganych parametrów (rozdzielczości, poziomu szumów, rozmiarów układu, liczby wymaganych do pracy komponentów zewnętrznych, poboru mocy czy też obecności wbudowanej pamięci FIFO) oraz... ceny układu. Warto bowiem pamiętać, że ceny poszczególnych układów różnią się i to znacząco, nawet w obrębie tej samej rodziny produktów, zaś selekcja komponentów zawsze wiąże się z nieuchronnym kompromisem czy też – jak mówi się w inżynierskim gronie – problemem zbyt krótkiej koldry.

W następnej części Poradnika Implementacji przyjrzymy się kolejnej grupie front-endów medycznych. Tym razem na warsztat weźmiemy układy przeznaczone do akwizycji sygnałów fotopletyzmoграфических i aplikacji pulsoksymetrycznych, tak powszechnych w dobie pandemii covid-19. Opis konkretnych propozycji z oferty największych producentów rynku półprzewodników poprzedzimy wstępem teoretycznym, który przystępnie nakreśli podstawy prowadzenia nieinwazyjnych biopomiarów optycznych.

inż. Przemysław Musz
przemyslaw.musz@ep.com.pl

koktajl niusów

Czy lodówka to rzeczywiście lodówka?

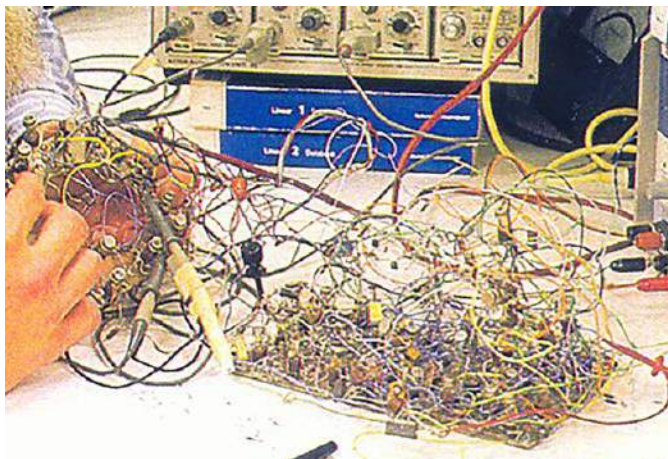
Nie chodzi o filozoficzne rozważania, ale o poprawność tytułowego pojęcia w znaczeniu współczesnych urządzeń chłodzących (takich, jak na fotografii). W rzeczywistości jest to chłodziarko-zamrażarka, którą producenci eksponują pod krótką, wygodną nazwą lodówka. Natomiast właściwe znaczenie tego słowa odnosi się do szafki/pojemnika do przechowywania lodu wodnego (prawdziwego lub sztucznego) schładzającego obecną w niej/nim żywność i napoje, by przedłużyć ich trwałość.



Pierwsze egzemplarze zaczęto budować jeszcze w latach trzydziestych XIX wieku w Stanach Zjednoczonych, do których lód był zwożony tonami na statkach z Kanady i północnych wybrzeży Nowej Anglii. Zatem to stary wynalazek, w dodatku znany jedynie na Zachodzie. Na ziemiach polskich raczej się o nim tylko słyszało. Jeśli się pojawiał, to bardzo rzadko, ponieważ był trudno osiągalny za czasów komunizmu i kosztowny.

Wspomniana wcześniej szafka ewoluowała około 160 lat i stała się czymś zupełnie innym niż pierwowzór. Co ma w tej sytuacji zrobić człowiek, który nigdy w życiu nie widział lodówki? Nie można się dziwić, że nazywa tak nowoczesne urządzenie. Jak sądzicie, czy jest szansa na to, że właściwe określenie będzie powszechnie stosowane?

źródło fotografii: <https://bit.ly/3oMqExn>



Gdyby nie było układów scalonych

Amerykański inżynier elektroniki Robert Allen Pease, bardziej znany jako Bob Pease (<https://bit.ly/34jUbow>), słynął z „pokręconych” obwodów elektrycznych (zwłaszcza analogowych) opartych na płytkach prototypowych z mnóstwem elementów dyskretnych i kabelków połączeniowych. Jeden z nich został pokazany na fotografii i jest to przetwornik napięciowo-częstotliwościowy (V-F) odpowiadający układowi scalonemu LM131 firmy National Semiconductor.

Taka konstrukcja wymaga wiele czasu i pracy. Tylko prawdziwy pasjonat elektroniki jest w stanie coś takiego zbudować, a Pease takim, właśnie był. Pozostałe osiągnięcia inżyniera można znaleźć w Internecie. Celowo nie podajemy adresów stron, by nie psuć Wam zabawy z samodzielnych poszukiwań. Polecamy także książkę jego autorstwa

„Projektowanie układów analogowych”. Zapoznanie się z jego pracami będzie najlepszym wyrazem uznania dla nieżyjącego już inżyniera.

źródło fotografii: <https://bit.ly/3oLgInV>



Arduino nie zawsze oryginalne

Na fotografii widać popularny zestaw Uno R3 w dwóch wersjach. Wyraźne logo wyróżnia oryginalną platformę. W podróbkach jest ono nieobecne bądź jest nieudolnie skopiowane. Różnice wykazuje też stylizowany napis modelu płytki oraz czcionka opisów wyprowadzeń i komponentów, której krój jest niebanalny. Charakterystyczny jest także kolor płytki, który trudno uzyskać, nie znając jego proporcji (a nawet jeżeli, to może się okazać, że jest on zastrzeżony). Przy dokładnej obserwacji widać różnicę w podzespołach – w ich barwach i rozstawieniu, a nawet w ścieżkach, które są uwypuklone jak numery na karcie płatniczej. Choć cena oryginału jest dużo wyższa niż podróbki, to jakość wykonania ją rekompensuje.

Podrabiane komponenty elektroniczne to poważny problem rozprzestrzeniony na ogromną skalę. Pamiętajcie o tym i przy zamawianiu elementów kierujcie się hasłem: „Arduino fajne, o ile oryginalne”.

źródło fotografii: <https://bit.ly/3yECmOY>

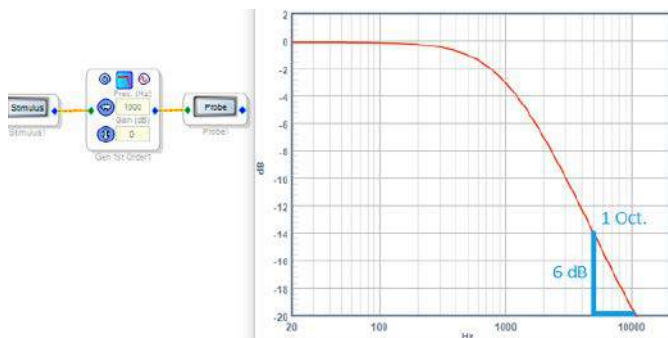
Promieniowanie niejonizujące – czy jest się czego obawiać?

Zgodnie z wyjaśnieniem Światowej Organizacji Zdrowia (WHO): „jest to termin oznaczający promieniowanie elektromagnetyczne o częstotliwościach, przy których energia fotonów jest zbyt mała, by wywołać proces jonizacji cząstek” (*non-ionizing radiation is the term given to radiation in the part of the electromagnetic spectrum where there is insufficient energy to cause ionization*).



Proces jonizacji to proces powstawania jonu, czyli kationu lub anionu – elektrycznie nieobojętnego atomu. Wspomniane częstotliwości obejmują zakres od 0 Hz do 3 PHz (równoważnie jest to długość fali poniżej 100 nm). Wspomnianym promieniowaniem są m.in.: fale radiowe i światło widzialne, czyli coś, co na co dzień nas otacza. Na rysunku został pokazany symbol ostrzegający przed nimi. Jest on przydatny w przypadku wysokich poziomów tego promieniowania mogących wywrzeć istotny wpływ na organizm człowieka. Tylko takie jest zagrożenie, żadne inne.

źródło rysunku: <https://bit.ly/2QUmUNW>

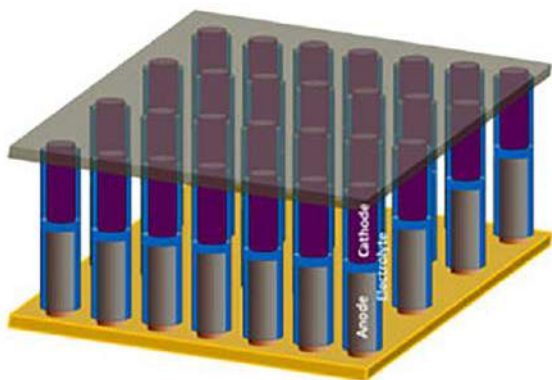


Oktawa, wbrew nazwie, nie równa się osiem!

W odniesieniu do elektroniki wynosi tylko dwa. Dlaczego? Ponieważ chodzi o pewną właściwość tego pojęcia, a dokładniej mowa o stosunku (ilorazie) dwóch częstotliwości kolejnych dźwięków w muzyce, wynoszącym dwa. To, że jest zawartych między nimi osiem stopni skali muzycznej, lub równoważnie dwanaście półtonów, jest nieważne. Tym niemniej są to dwa niewykluczające się rozumienia tego samego pojęcia, z czego każde jest stosowane w różnej dziedzinie. Dobrym rozwiązaniem byłoby rozgraniczenie tych pojęć i nadanie im odrębnych określeń. Jednak w ten sposób stracilibyśmy związek „ze źródłem”, z którego pochodzi oktawa.

Elektronicy powinni wiedzieć, że w charakterystykach transmitancji filtrów analogowych, w których stromość zbocza wynosi 6 dB na oktawę, odpowiada to 20 dB na dekadę (co pokazuje rysunek). W zasadzie to przez te filtry mówi się o tytułowym pojęciu w elektronice.

źródło rysunku: <https://bit.ly/2RHMGom>



Przepis na małą baterię

Schemat konstrukcji pokazuje rysunek. Potrzebne jest podłoże miedziane z wgłębieniami, które będzie jedną ze stron baterii. Następnie nikiel lub cyna do formowania anod baterii w kształcie walców pasujących do wgłębien. W dalszej kolejności tlenek polietyleny w postaci żelu do utworzenia cienkiej warstwy separującej anody (kolor niebieski) – będzie to elektrolit baterii. Wreszcie polianiliny do ukształtowania katod w formie walca (kolor fioletowy) ponad wspomnianym tlenkiem. Jeszcze tylko folia aluminiowa łącząca katody i pełniąca funkcję drugiej strony baterii. Gotowe!

Wszystko wydaje się proste, dopóki nie zdamy sobie sprawy ze skali, w której trzeba wszystko wykonać. Potrzebujemy maksymalnie setek nanometrów, więc metody domowe odpadają. W grę wchodzi tylko porządne laboratorium naukowe z dobrej klasy mikroskopem elektronowym, idealnym do tego rodzaju zadań.

Liczba cykli ładowania i rozładowania takiego ogniwa jest mała – wynosi tylko 20. Zatem miniaturyzacja została dokonana kosztem sprawności. Na szczęście ostatnie można zwiększyć, eksperymentując z materiałami i ich grubościami. Jest to cel przyświecający obecnie naukowcom na całym świecie. Całkiem przypadkiem odkryli ponad dwukrotne wydłużanie się anod z dwutlenku cyny podczas ładowania. Zaobserwowali to jeszcze w 2010 roku w amerykańskim

instytucie badawczym Sandia National Laboratories, który udostępnił zdjęcie ilustrujące wspomniane zjawisko pod adresem: <https://bit.ly/3hS8RDI>.

źródło rysunku: <https://bit.ly/3wwzlyp>

O początkach telefonii w Polsce

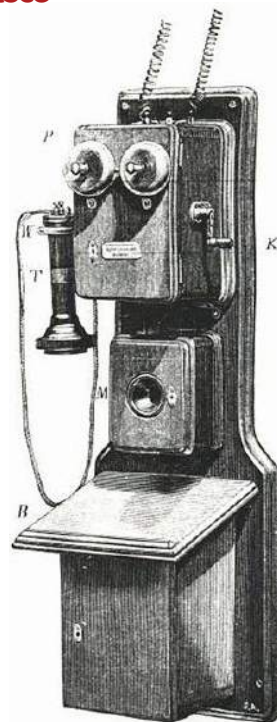
W czasach, gdy naszego kraju nie było na mapach i zabory trwały w najlepsze, Aleksander Graham Bell uzyskał patent na telefon. Jego wynalazek pojawił się w Warszawie – stolicy Królestwa Polskiego (Imperium Rosyjskie) i pierwszy raz na ziemiach polskich 7 i 9 grudnia 1877 roku przeprowadzono pomyślne próby łączności z jego zastosowaniem. Była to linia telegraficznej Kolei Warszawsko-Wiedeńskiej (obecnie linia kolejowa nr 1) między Warszawą a Skierniewicami. Ówczesna prasa doniosła, że „(...) głos jest wyraźny tylko przy zupełnej ciszy, jaka panowała w czasie prób czynionych o godzinie drugiej w nocy. W innym czasie dźwięki cichsze przez telefon przesyłane giną zupełnie, a dźwięki mocniejsze zmać są innymi odgłosami. Ponieważ dla oszczędności nie zakładano dla telefonu oddzielnych drutów, ale korzystano z telegraficznych, więc podczas prób w dzień robionych telefon ustawicznie brzmiał odgłosem pukania telegrafistów na pobliskich stacjach”. Opisane wyniki potwierdziły użyteczność rozwiązania. Niespełna miesiąc później odbyły się jego publiczne pokazy w Warszawie, a także powstała pierwsza linia telefoniczna łącząca cukiernię Semadeniego w Ogrodzie Saskim z zakładem optyka Jakuba Pika przy ul. Niecałej 7. Ponadto odbyła się próba transmisji koncertu muzycznego z Ogrodu Saskiego do gmachu Zarządu Wodociągów przy ul. Dobrej na Powiślu (już wtedy puszczano muzykę przez telefon!).

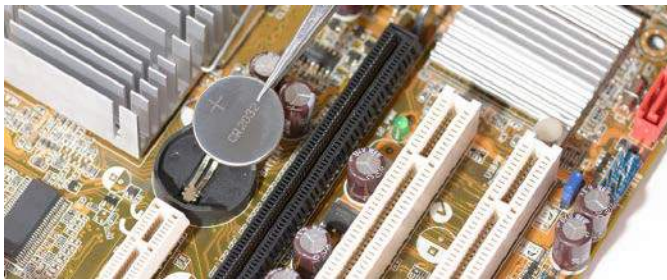
Dalsza historia to sieć telefoniczna projektu amerykańskiego towarzystwa The International Bell Telephone Company, którą uruchomiono 13 lipca 1882 roku – zachęcano do niej w następujący sposób: „Towarzystwo telefonowe umie robić reklamę. Pragnie ono i szerszą publiczność obznajomić szczegółowo z użytkowaniem telefonu. W tym celu, jak nas zapewniamy, połączy ono dwie najbardziej uczęszczane przez publiczność cukiernie Loursa w Saskim Ogrodzie i w Hotelu Europejskim telefonowymi drutami. Doświadczenie z tą linią i objaśnienie każdemu, kto zechce poznać bliżej ich konstrukcję, rozpoczyna się z dniem 1 lipca br.”. Na fotografii został pokazany aparat dołączony do tej sieci. Przed jej uruchomieniem wspomniano jeszcze, że, „W dniu jutrzejszym otwarta będzie czynność telefonów warszawskich. Sto połączeń zastrzeżonych przez ministerstwo w koncesji jest już dokonanych (...)”. W dniu uruchomienia przyłączono 106 abonentów, po czym dalej było „(...) abonentów 163 [na koniec roku – W.Ł.], w roku 1883 liczba ta podniosła się do 302, w roku 1884 do 416, a na koniec w ubiegłym roku [1885 – W.Ł.] liczono 520 abonentów. Ogółem rozmów prowadzono w r. 651 497, najwięcej z nich było w czerwcu, a najmniej w marcu...”.

źródło fotografii: <https://bit.ly/3oNcID8>

Bateria CMOS – skąd jej nazwa?

Chodzi przede wszystkim o baterię CR2032, którą należy włożyć w gniazdo zamontowane na płycie głównej komputera. Jest ona potrzebna, ponieważ podtrzymuje zasilanie pamięci RAM, wykonanej w technologii CMOS, zawierającej konfigurację systemu BIOS

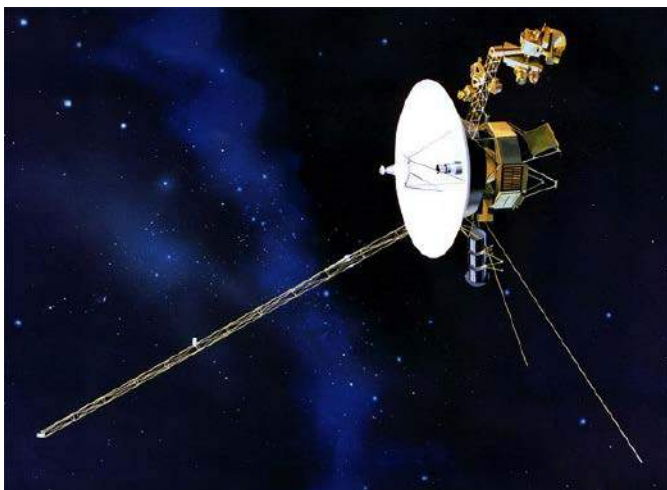




po wyłączeniu komputera. Nazwa wynikała z tego, że technologia CMOS nie była masowo stosowana w układach scalonych jak dzisiaj. Mimo upływu czasu pozostała, choć jest już tylko potoczem, o czym należy pamiętać.

Ponadto wskazana pamięć, wbrew temu, co się mówi, nie jest nieulotna (*non-volatile*), tzn. nie jest pamięcią NVRAM. Najlepiej świadczy o tym wyjęcie omawianej baterii sprawiające, że system BIOS trzeba na nowo skonfigurować.

źródło fotografii: <https://bit.ly/3ukZr6b>



Która transmisja radiowa odznacza się największym opóźnieniem?

Odpowiedź wydaje się oczywista – transmisja z najbardziej oddalonym od Ziemi obiektem zbudowanym przez człowieka, tzn. z sondą Voyager 1, która dotychczas (początek maja 2021 roku) pokonała dystans w kosmosie co najmniej 152 jednostek astronomicznych (AU – *Astronomical Unit*). Jedną taką jednostką, będącą średnią odległością Ziemi od Słońca, to 149 597 870 700 metrów, w zaokrągleniu równa 150 milionów kilometrów. Obliczając, otrzymujemy ponad 22 miliardy 800 milionów kilometrów przemierzone przez sondę.

Światło pokonuje ten dystans w czasie ponad 21 godzin, podobnie jak fala radiowa. Podany czas określa, ile trzeba czekać, nim komenda (polecenie) wysłana do Voyagera 1 zostanie wykonana. Nie wolno zapominać, że jest to jednostronne opóźnienie w łączności i przy założeniu, że nie ma tłumienia po drodze (w przypadku przestrzeni kosmicznej jest to zawsze spełnione), w dodatku stale wzrasta. Sonda ciągle przemieszcza się z prędkością powyżej 38 tysięcy kilometrów na godzinę. Jej aktualny status można sprawdzić pod adresem: <https://go.nasa.gov/2TjdyvV>. Polecamy się z nim zapoznać.

źródło fotografii: <https://bit.ly/3vqeo8s>

Zmierzch telefonii trzeciej generacji w Polsce

Okazuje się, że w ciągu najbliższych kilku lat operatorzy komórkowi planują stopniowe wyłączanie nadajników 3G, by zastąpić je stacjami Single RAN obsługującymi jednocześnie sieci drugiej oraz czwartej i piątej generacji. W szczególności chce to poczynić firma T-Mobile – w ciągu dwóch najbliższych lat. Jak mówi jej prezes Andreas Maierhofer „Przygotowujemy T-Mobile do wyłączenia już przestarzałej

technologii, jaką jest sieć 3G. Już teraz mniej niż 5% przesyłanych danych jest wysyłanych w sieci 3G i udział ten stale spada. Wyłączenie 3G nastąpi do 2023 r. Podobną sytuację obserwujemy w rozmowach głosowych. Prowadzonych jest coraz więcej połączeń głosowych w technologii VoLTE. Widać to w statystykach, gdzie 30% wszystkich połączeń głosowych odbywa się w VoLTE”. Ostatnie ma przy tym wzrosnąć dzięki wspomnianej usłudze w ofertach na kartę i MIX. Ponadto zgodnie ze słowami tej samej osoby: „Również w pierwszej połowie tego roku będziemy oferować technologię 5G klientom prepaid i uruchomimy kilka programów przeznaczonych dla klientów, których telefony i routery nie mają dostępu do technologii LTE, aby zapewnić im znacznie lepsze wrażenia z usług sieciowych T-Mobile”.

Obecnie jest to jedyna taka deklaracja – pozostali operatorzy jeszcze się nie określili w tej sprawie. Tym niemniej to kwestia czasu, kiedy i oni to poczynią. Czy jest się czego obawiać? Nie – sprzęt mamy coraz nowszy, a i tak pozostanie jeszcze sieć 2G w awaryjnych sytuacjach. Tak naprawdę dopiero jej wyłączenie będzie powodem do grozy, lecz nie uprzedzamy faktów. Na razie cieszymy się tym, co jest i tym, co nadchodzi, czyli siecią 5G. I znów oddajmy głos prezesowi firmy T-Mobile, który mówi, że: „Pracujemy intensywnie nad rolloutem sieci 5G. Obecnie gotowych na nowy standard jest ponad 1600 wież, co przekłada się na zasięg 5G w T-Mobile dla populacji rzędu 6,3 mln. Kontynuujemy inwestycje w tym zakresie, ale coraz istotniejsza staje się kwestia rozdysponowania częstotliwości. Liczymy, że aukcja 5G zostanie uruchomiona i przeprowadzona bardzo szybko, ponieważ to zdecydowanie przyspieszy potencjał rolloutu 5G. Jesteśmy przygotowani na pozyskanie częstotliwości dla 5G”. Oby to się udało, czego serdecznie życzymy.

źródło rysunku: <https://bit.ly/3bPSJOU>;

<https://bit.ly/3oOhkZU>; <https://bit.ly/3yDzYIf>

Ostatni VHS nie taki stary

Jest nim wydany 14 marca 2006 roku, na nośniku z fotografii, film „A History of Violence” (Historia przemocy) w reżyserii Davida Cronenberga, oparty na powieści pod tym samym tytułem autorstwa Johna Wagnera i Vinca’a Locke’a. Rzecz



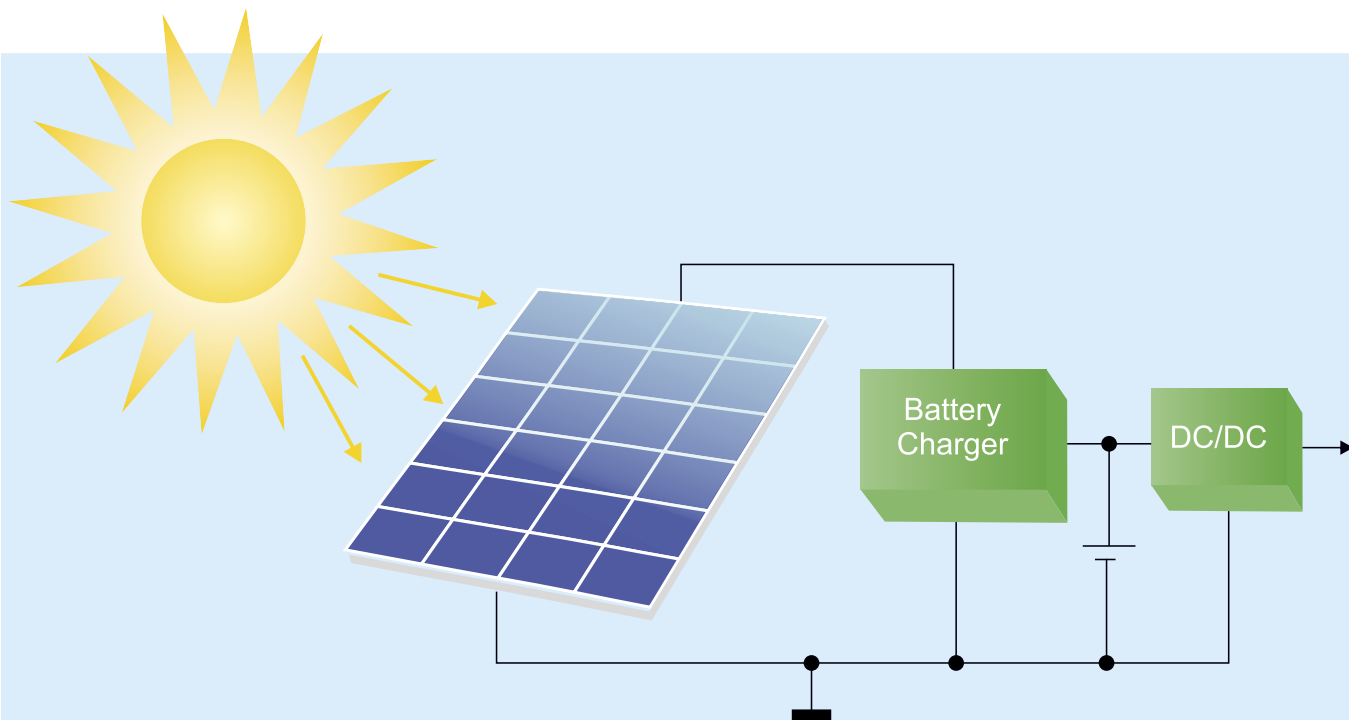
nie do zdobycia – dostępne są tylko urywki. Jeden z nich to pierwsze osiem minut „taśmy” do obejrzenia na portalu YouTube pod adresem: <https://bit.ly/3yMMPZ0>. To niewiele, ale dobrze widać tę charakterystyczną jakość obrazu, dużo gorszą od tej oferowanej współcześnie. Chciałoby się o tym przekonać, ale czy jest sposób, by to zrobić, skoro magnetowidy nie są produkowane od 2008?

Jakkolwiek by patrzeć, VHS to już przeżytek o znaczeniu tylko kolekcjonerskim. Choć może będzie przeżywał renesans jak obecnie kasety magnetofonowe i płyty winylowe. Tymczasem zachęcamy do odkurzenia nagrań sprzed lat i archiwizowania ich w formacie komputerowym. VHS to nośnik, który ulega degradacji i szkoda by dawne wspomnienia bezpowrotnie zniknęły.

źródło fotografii: <https://bit.ly/3hQuaoY>

Jakub Tyburski

jakub.tyburski@wat.edu.pl



Systemy dla Internetu Rzeczy (49)

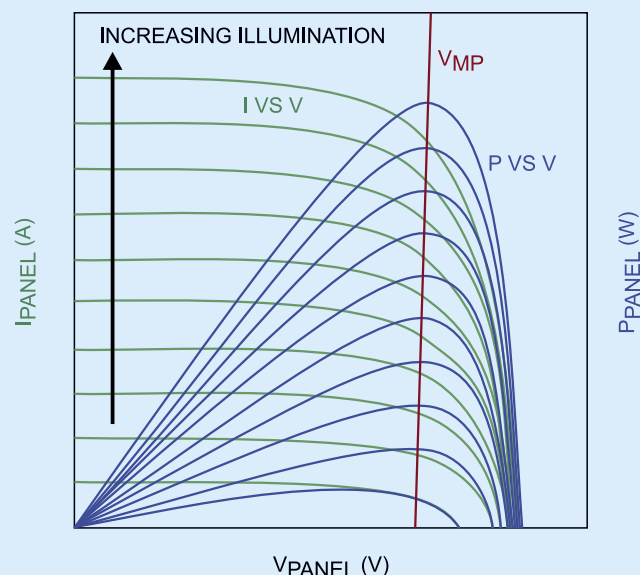


Pozyskiwanie energii słonecznej, śledzenie maksymalnego punktu mocy (MPPT)

Do produkcji energii elektrycznej nadają się różne źródła energii z otoczenia, w tym drgania mechaniczne, różnica temperatur i padające światło. Jednym z wyzwań stojących przed aplikacjami wykorzystującymi źródło fotowoltaiczne jest brak mocy wejściowej w ciemności i przy słabym oświetleniu. W przypadku większości zastosowań wymaga to użycia elementów magazynujących energię, takich jak superkondensatory lub akumulatory, który są wystarczająco duże, aby zapewnić podtrzymanie zasilania przez najdłuższy okres oczekiwania w ciemności.

Źródła fotowoltaiczne mogą być modelowane elektrycznie przez źródło prądu połączone równolegle z diodą. Bardziej złożone modele wykazują efekty wtórne, ale do naszych celów ten model jest wystarczający [7]. Dwa typowe parametry charakteryzujące ogniwo fotowoltaiczne to napięcie w układzie otwartym i prąd zwarcia. Typowe krzywe prądu i napięcia ogniwa fotowoltaicznego zostały pokazane na **rysunku 1**. Należy zauważyć, że prąd zwarcia jest wyjściem źródła prądowego modelu, podczas gdy napięcie w obwodzie otwartym jest napięciem przewodzenia diody modelu. Wraz ze wzrostem poziomu oświetlenia wzrasta prąd z generatora, a krzywe wykresu I/V (zielony) przesuwają się w górę.

Aby uzyskać maksymalną moc z ogniwa fotowoltaicznego, rezystancja wejściowa przetwornika mocy musi być dopasowana



Rysunek 1. Krzywa charakterystyki I/V typowego panelu słonecznego [2]

do rezystancji wyjściowej ogniwa, co skutkuje pracą w punkcie maksymalnej mocy. Aby zapewnić maksymalny pobór mocy, napięcie wyjściowe ogniwa powinno pracować na szczycie krzywej mocy.

Moc, jaką można wygenerować za pomocą ogniwa fotowoltaicznego, zależy od wielu czynników. Moc wyjściowa ogniwa jest

proporcjonalna do poziomu światła padającego na ogniwo, całkowitej powierzchni ogniwa i wydajności ogniwa. Większość ogniw fotowoltaicznych jest przystosowana do pracy w pełnym, bezpośrednim świetle słonecznym (1000 W/m^2), ale takie idealne warunki raczej nie wystąpią w większości zastosowań. W przypadku urządzeń zasilanych światłem słonecznym szczytowa moc dostępna z ogniwa może łatwo zmieniać się wielokrotnie z dnia na dzień z powodu pogody, pory roku, zamglenia, kurzu i zmian kąta padania światła słonecznego. Ogniwo mocno nagrzane w pełnym słońcu zmniejsza wydajność o 15...30%. Typowa moc wyjściowa ogniwa krystalicznego w pełnym słońcu wynosi około 40 mW na cm^2 kwadratowy, w zależności od charakterystyki ogniwa. Ogniwo fotowoltaiczne o powierzchni kilku cm^2 kwadratowych wystarcza do uruchomienia wielu zdalnych czujników i do podładowania baterii [7].

Śledzenie maksymalnego punktu mocy (MPPT)

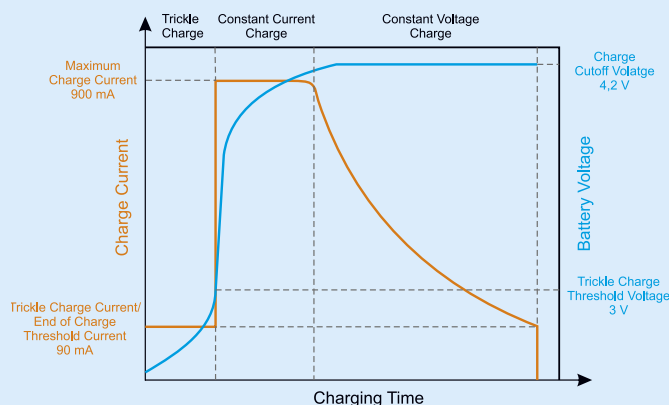
Algorytm śledzenia maksymalnego punktu mocy MPPT (*Maximum Power Point Tracking*) może zapewnić dostarczanie przez panel słoneczny maksymalnej mocy wyjściowej przy różnych obciążeniach i przy różnych poziomach światła słonecznego. Możemy określić prąd zwarcia I_{SC} oraz napięcia otwartego układu V_{OC} z przecięcia punktów krzywej wykresu I/V (zielony). Wartości I_{SC} i V_{OC} rosną wraz ze wzrostem poziomu oświetlenia. Wraz ze wzrostem napięcia wyjściowego V_{PANEL} prąd wyjściowy I_{PANEL} stopniowo maleje, a następnie spada w dół w sposób określony przez specjalne punkty V_{MP} maksymalnej mocy. Mnożąc napięcie i prąd, co daje moc, i przyjmując napięcie wyjściowe jako oś x, otrzymujemy krzywą P/V (kolor niebieski). Moc wyjściowa P_{PANEL} osiąga szczyt, gdy napięcie wyjściowe wynosi V_{MP} .

Chociaż maksymalna moc wzrasta wraz z oświetleniem, V_{MP} zmienia się nieznacznie. W rezultacie można wybrać pojedyncze napięcie śledzące V_{MP} , które zapewnia pracę w pobliżu maksymalnego punktu mocy dla szerokiego zakresu poziomów oświetlenia. Nawet jeśli punkt pracy nie będzie dokładnie w punkcie maksymalnej mocy przy ekstremalnych poziomach oświetlenia, redukcja mocy wyjściowej w stosunku do ideału wynosi zwykle tylko 5...10%. Z reguły maksymalne napięcie sterujące punktu mocy powinno wynosić około 75...80% napięcia układu otwartego ogniwa. Śledzenie ogniwa przy tym napięciu skutkuje prądem wyjściowym ogniwa na poziomie 75...80% prądu zwarcia [7].

Układ scalony zarządzania energią słoneczną CN3065

Układ scalony CN3065 firmy Consonance Electronic [6] to ładowarka akumulatorów przeznaczona głównie do ładowania akumulatorów litowo-jonowych lub litowo-polimerowych z pojedynczym ogniwem [7]. Układ jest wyposażony w wewnętrzny tranzystor MOSFET z kanałem P. Ładowanie akumulatorów odbywa się w trybach stałego prądu i stałego napięcia. Ciągły prąd ładowania można zaprogramować do 1000 mA za pomocą zewnętrznego rezystora. Nie jest wymagana żadna dioda blokująca ani rezystor pomiarowy. Wbudowany 8-bitowy przetwornik ADC umożliwia automatyczne regulowanie prądu ładowania w oparciu na możliwościach dołączonego zasilacza.

Układ CN3065 dobrze nadaje się do systemów zasilanych energią słoneczną lub aplikacji, które wymagają ładowania akumulatora litowo-jonowego lub akumulatora litowo-polimerowego z zasilaczem wejściowym, którego możliwości prądowe są ograniczone. Wyjścia $nDONE$ i $nCHRG$, typu otwarty dren, wskazują stan ładowarki. Wewnętrzny obwód regulacji termicznej zmniejsza zaprogramowany prąd ładowania, jeśli temperatura struktury scalonej układu wzrasta do około 115°C . Ta funkcja chroni układ CN3065 przed nadmiernym przegrzaniem i pozwala użytkownikowi zwiększać możliwości obsługi zasilania bez ryzyka uszkodzenia układu lub elementów zewnętrznych. Inną korzyścią płynącą z zastosowania



Rysunek 2. Fazy ładowania akumulatora litowego [2]

regulacji termicznej jest to, że prąd ładowania można ustawić zgodnie z typowymi temperaturami otoczenia dla danego zastosowania, z gwarancją, że ładowarka automatycznie zmniejszy prąd w najgorszych warunkach.

Układ CN3065 ma wejście TEMP dołączane do termistora, służące do pomiaru temperatury akumulatora. Jeśli temperatura akumulatora jest zbyt wysoka lub zbyt niska i cykl ładowania jest zawieszany, funkcję wykrywania temperatury akumulatora można wyłączyć, podłączając wejście TEMP do GND.

Wewnętrzny obwód blokady podnapięciowej (UVLO) monitoruje napięcie wejściowe i utrzymuje ładowarkę w trybie wyłączenia, aż VIN wzrośnie powyżej napięcia blokady podnapięciowej (3,0 V typ.). Obwód UVLO ma wbudowaną histerezę 0,1 V.

Układ ma komparator dla stanu uśpienia, który utrzymuje ładowarkę w tym stanie, jeśli VIN spadnie poniżej progu trybu uśpienia – $V_{BAT} + 40 \text{ mV}$. Z trybu uśpienia ładowarka wyjdzie, gdy VIN wzrośnie o 90 mV powyżej napięcia akumulatora.

Układ pracuje z napięciem wejściowym do 6 V i umożliwia ładowanie prądem stałym do 1000 mA (typowo 900 mA). Pobór prądu przez sam układ wynosi typowo 650 μA w trakcie ładowania akumulatora, 3,5 μA w stanie czuwania ($V_{BAT}=V_{REG}$) oraz 3 μA (max.) w stanie uśpienia ($V_{IN}=0$).

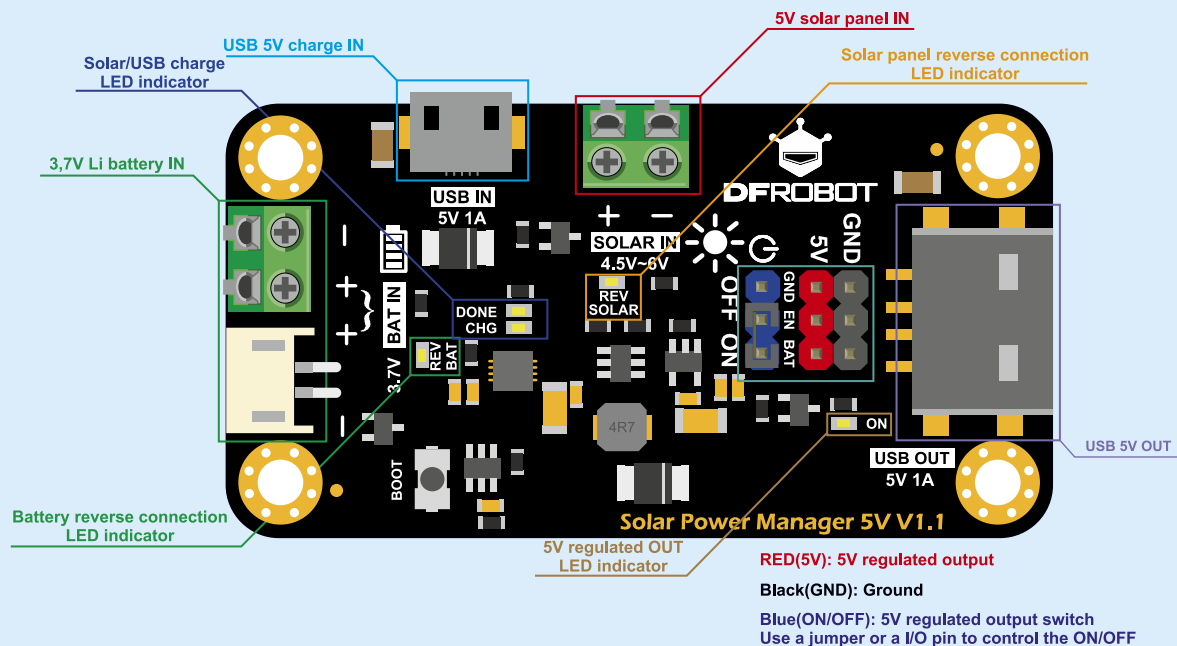
Cykl ładowania akumulatora

Napięcie odniesienia, wzmacniacz błędów i dzielnik rezystora zapewniają regulację napięcia z dokładnością do 1%, która może spełniać wymagania akumulatorów litowo-jonowych i litowo-polimerowych. Gdy napięcie wejściowe nie jest obecne, ładowarka przechodzi w tryb uśpienia, obniżając prąd rozładowania akumulatora do mniej niż 3 μA . To znacznie zmniejsza pobór prądu z akumulatora i wydłuża czas czuwania.

Cykl ładowania rozpoczyna się, gdy napięcie na wejściu VIN wzrośnie powyżej poziomu UVLO oraz rezystor nastawy prądu RISET jest podłączony pomiędzy wejście ISET oraz masę. Wtedy wyjście $nCHRG$ jest ustawiane w niski stan logiczny, aby wskazać, że cykl ładowania jest w toku.

Układ scalony zarządzania energią słoneczną CN3065 bezpiecznie i szybko ładuje baterię litową przez trzy fazy: ładowanie podtrzymujące, ładowanie stałym prądem, ładowanie stałym napięciem (rysunek 2).

Ładowanie podtrzymujące. Ponieważ rezystancja wewnętrzna akumulatora litowego staje się wysoka, gdy napięcie jest niskie, ładowanie dużym prądem na początku nie jest dobrym sposobem. W takich warunkach temperatura akumulatora wzrasta, a żywotność baterii spada. Gdy napięcie akumulatora (doprowadzone do wejścia FB) jest niższe niż napięcie progowe ładowania podtrzymującego wynoszące 3 V, moduł przechodzi w fazę ładowania podtrzymującego, ładując akumulator prądem wynoszącym 10% maksymalnego



Rysunek 3. Moduł zarządzania energią słoneczną DFR0559 [2]

prądu ładowania (do 90 mA), aż do momentu, gdy napięcie akumulatora będzie wyższe niż 3 V.

Ładowanie prądem stałym. Gdy napięcie akumulatora jest wyższe niż 3 V, moduł wchodzi w fazę ładowania akumulatora prądem stałym do 900 mA. W trybie prądu stałego prąd ładowania jest ustalany przez RISET. Wyjście nCHRG jest nadal ustawione na niski stan logiczny.

Ładowanie stałonapięciowe. Gdy napięcie akumulatora zbliża się do napięcia odcięcia ładowania 4,2 V, moduł wchodzi w fazę ładowania stałonapięciowego. Moduł kontynuuje ładowanie akumulatora stałym napięciem, a prąd ładowania stopniowo maleje. Gdy prąd ładowania spada do 10% maksymalnego prądu ładowania (90 mA), ładowanie kończy się. Wyjście nCHRG jest ustawiane w wysoki stan logiczny a wyjście nDONE w stan niski. Aby wznowić cykl ładowania, wystarczy usunąć napięcie wejściowe i ponownie je przyłożyć.

Ładowanie automatyczne. Jeśli akumulator jest w pełni naładowany (dioda DONE świeci się), źródło wejściowe modułu (SOLAR IN lub USB IN) jest włączone, a napięcie akumulatora spada z powodu samorozładowania lub obciążenia poniżej 4,05 V, to nowy cykl ładowania rozpocznie się automatycznie.

Układ scalony zarządzania energią słoneczną CN3065 maksymalizuje moc wyjściową, utrzymując napięcie wyjściowe w pobliżu VMP, w sposób zwany algorytmem stałonapięciowym MPPT. Dzięki tej funkcji moduł ma lepszą wydajność konwersji w porównaniu do zwykłej liniowej ładowarki akumulatorów.

Moduł zarządzania energią słoneczną

Moduł Solar DFR0559 Power Manager firmy DFRobot jest modulem zarządzania energią słoneczną o małej mocy i wysokiej wydajności, przeznaczony do panelu słonecznego 5 V o mocy do 10 W (rysunek 3). W module został zastosowany układ scalony zarządzania energią słoneczną CN3065 [2]. Moduł nadaje się do projektów IoT, związanych z energią odnawialną. Jego podstawowe parametry to:

- napięcie wejściowe z panelu solarne (SOLAR IN): 4,5...6 V,
- napięcie wyjściowe 5 V, 1 A (1,5 A max.),
- wydajność (akumulator 3,7 V 900 mA): 5 V/USB OUT: 89% przy 10% obciążenia, 86% przy 50% obciążenia, 83% przy 90% obciążenia,
- wydajność ładowania USB/słonecznego: 73%,
- maksymalny prąd spoczynkowy: <1 mA,
- temperatura pracy: -40...+85°C,

- wymiar: 33×63 mm/1,30×2,48 cala,
- waga: 12 g.

Na dolnej stronie płytki drukowanej jest wykonane złożone pole druku z opisem „Cooling Fin”. Jest ono przeznaczone do przyklejenia (poprzez samoprzylepną podkładkę termoprzewodzącą) radiatora (dostarczany w zestawie). Poprawia on parametry termiczne pracy układu scalonego CN3065, zmontowanego po przeciwnej stronie płytki drukowanej. Jest to konieczne przy stosowaniu panelu słonecznego o mocy od 5 W (10 W) przy maksymalnym prądzie ładowania akumulatora (900 mA). Układ scalony ma funkcję ochrony przed przegrzaniem, która automatycznie ograniczy prąd ładowania.

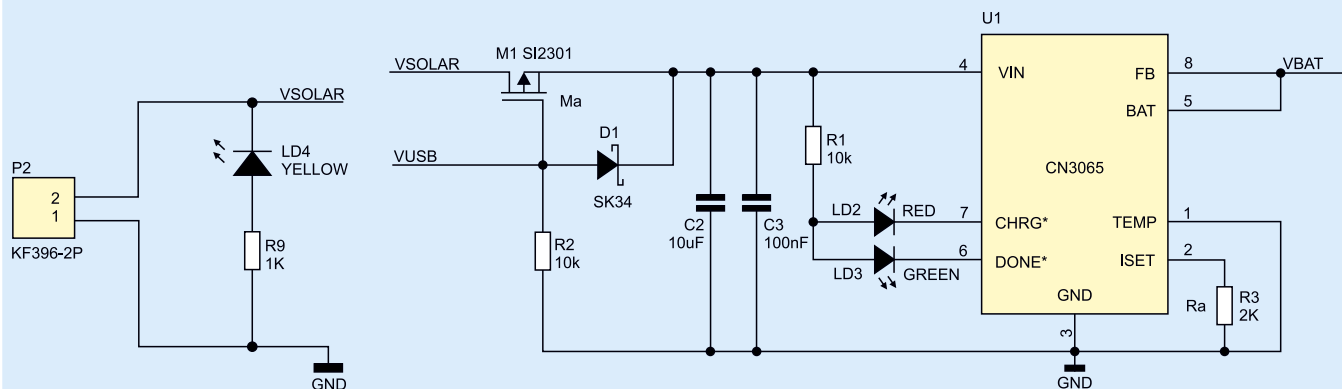
Dołączanie panelu słonecznego. Panel słoneczny o napięciu 4,4...6 V (typowo 5 V) jest dołączany do złącza „SOLAR IN” (P2). Do zacisków wejściowych dołączona jest dioda LED „REV SOLAR” (LD4, żółta/pomarańczowa), która świeci, gdy zostanie podłączone napięcie wejściowe o odwrotnej polaryzacji (rysunek 4). Dalej jest zastosowany tranzystor M1, który zapobiega uszkodzeniu układu scalonego CN3065 przy odwrotnym podłączeniu napięcia wejściowego. Można zastosować dowolny rodzaj paneli, o ile napięcie w obwodzie otwartym jest mniejsze niż 6,5 V, w przeciwnym razie moduł może zostać trwale uszkodzony.

Ładowanie akumulatora ze złącza USB. Na płytce drukowanej modułu jest zamontowane gniazdo „USB IN 5V 1A” typu microUSB (U3) dołączone do linii VUSB (rysunek 3). Gniazdo służy tylko do ładowania akumulatora. Zaleca się używanie zasilacza sieciowego o napięciu 5 V i prądzie 1 A.

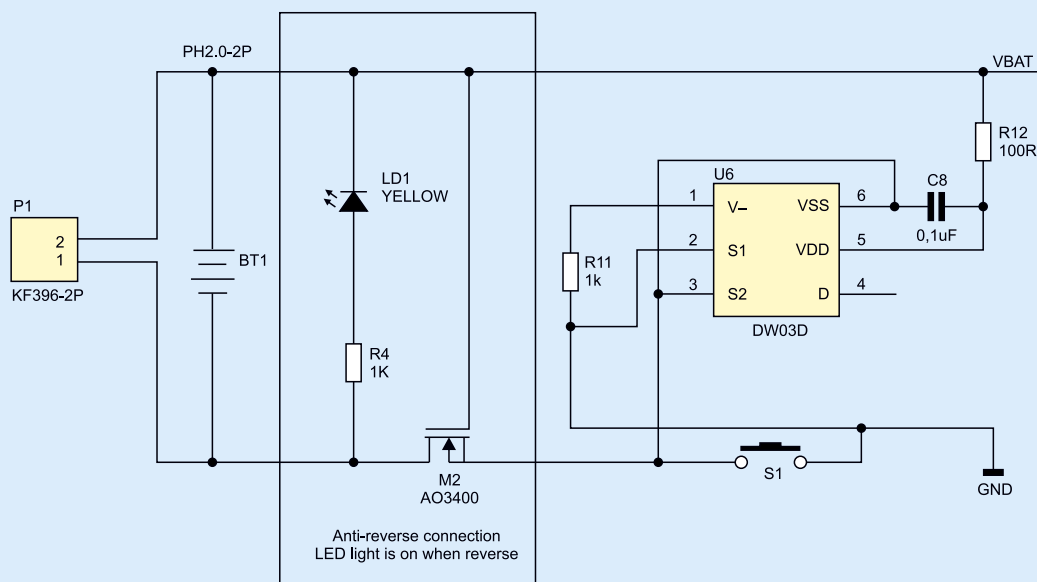
Automatyczne przełączanie USB IN/SOLAR IN. Zarówno wejście „USB IN”, jak i „SOLAR IN” są w stanie dostarczyć maksymalny prąd ładowania 900 mA. Aby zapobiec konfliktowi ładowania spowodowanemu przez użycie obu wejść, ładowanie USB ma wyższy priorytet. Gdy gniazdo „USB IN” jest zasilane przez zasilacz sieciowy, to prąd płynie przez diodę D1, a wejście „SOLAR IN” jest odłączone przez tranzystor M1 (rysunek 4). Po wyłączeniu zasilania „USB IN” wejście „SOLAR IN” wznowia ładowanie.

Prąd ładowania akumulatora jest ustalany wartością rezystora Ra (rysunek 4) według wzoru $I_{CHG} = 1800/Ra$. Dla $Ra = 2 \text{ k}\Omega$ daje to 900 mA.

Dołączanie akumulatora. Akumulator jest dołączany do wejścia „BAT IN” poprzez dwa złącza (rysunek 3) połączone równolegle (P1) w standardzie JST PH2.0 oraz KF396 3,96 mm (rysunek 5). Zwykle podłączany jest jeden akumulator litowo-polimerowy/litowo-jonowy



Rysunek 4. Obwód dołączenia panelu słonecznego oraz układ scalony zarządzania energią słoneczną CN3065 [1]



Rysunek 5. Układ dołączenia akumulatora [1]

3,7 V (4,2 V po całkowitym naładowaniu) do jednego ze złączy. Maksymalny prąd ładowania wynosi 900 mA zarówno dla zasilania z wejścia „USB IN”, jak i „SOLAR IN”.

Do zacisków wejściowych dołączona jest dioda LED „REV BAT” (LD1, żółta/pomarańczowa), która świeci, gdy zostanie podłączone napięcie wejściowe odwrotnej polaryzacji. Dalej jest zastosowany tranzystor M2, który zapobiega uszkodzeniu układu scalonego CN3065 przy odwrotnym podłączeniu napięcia wejściowego.

Do zabezpieczenia akumulatora jest zastosowany układ zabezpieczający DW06D (U6). Jest on dołączony do linii VBAT pomiędzy zaciskiem dodatnim akumulatora i wyprowadzeniem BAT układu scalonego zarządzania energią słoneczną CN3065. Wyprowadzenie S2 układu DW06D jest dołączone do zacisku ujemnego akumulatora, a wyprowadzenie S1 do masy modułu. W ten sposób układ kontroluje przepływ prądu i umożliwia odłączenie akumulatora od modułu. Układ DW06D zabezpiecza akumulator litowy przed przeładowaniem (4,3 V, próg 4,1 V), nadmiernym rozładowaniem (2,5 V, próg 2,9 V), odwrotnym podłączeniem oraz nadmiernym prądem (3 A).

Po każdym dołączeniu akumulatora do gniazdka BAT IN modułu należy przycisnąć przycisk „BOOT” (S1) w celu aktywowania układu zabezpieczenia i wyprowadzenia napięcia wyjściowego na gniazdko USB OUT oraz listwę 5 V (czerwona).

Wskaźniki LED określają status systemu [2]:

- jeśli podłączone jest źródło zasilania modułu (USB IN lub SOLAR IN), to wykonywane jest ładowanie akumulatora i dioda CHG (czerwona) się świeci;

- jeśli akumulator jest w pełni naładowany, to dioda CHG gaśnie, a dioda DONE (zielona) zaczyna świecić;
- jeśli akumulator nie jest podłączony, świecą się jednocześnie diody CHG i DONE.

Regulatory napięcia wyjściowego

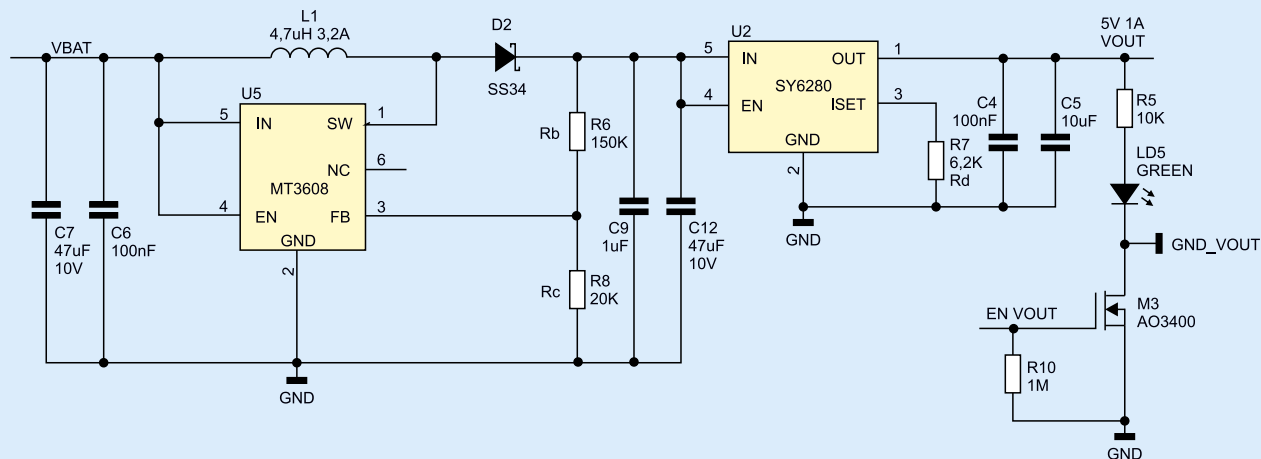
Moc dostarczana na wyjście modułu jest pobierana bezpośrednio z akumulatora, z linii VBAT (rysunek 6).

Najpierw został zastosowany impulsowy regulator DC-DC podwyższający MT3608 (U5) o niskiej mocy spoczynkowej. Napięcie wyjściowe jest ustalane przez dzielnik rezystancyjny według wzoru $V_{out1} = 0,6 \cdot (1 + R_b/R_c)$. Dla zastosowanych wartości daje to 5,1 V (do 2 A). Układ pracuje z częstotliwością 1,2 MHz ze sprawnością ok. 93%.

Napięcie to jest podawane na układ przełącznika zasilania SY6280 (U2) o niskim spadku napięcia ($R_{on} = 80 \text{ m}\Omega$). Układ dostarcza napięcie wyjściowe VOUT 5 V 1 A dołączone do gniazdka „USB OUT” typu USB-A. Układ SY6280 ma funkcję ograniczania prądu wyjściowego. Poziom ograniczenia jest ustawiany zewnętrznym rezystorem R7 według wzoru $I_{lim} = 6800/R7$. Dla zamontowanego na module rezystora oznacza to ograniczenie do ok. 1 A prądu.

Układy konwertera DC-DC MT3608 oraz klucza zasilania SY6280 są zawsze włączone. Napięcie wyjściowe VOUT 5 V jest zawsze generowane i doprowadzane do gniazdka USB OUT, gdy jest aktywne jedno z wejść USB IN, BAT IN lub SOLAR IN.

Jednak kluczowane jest dołączenie masy gniazdka USB OUT (USB-A). Zastosowany jest do tego tranzystor M3 sterowany z wejścia

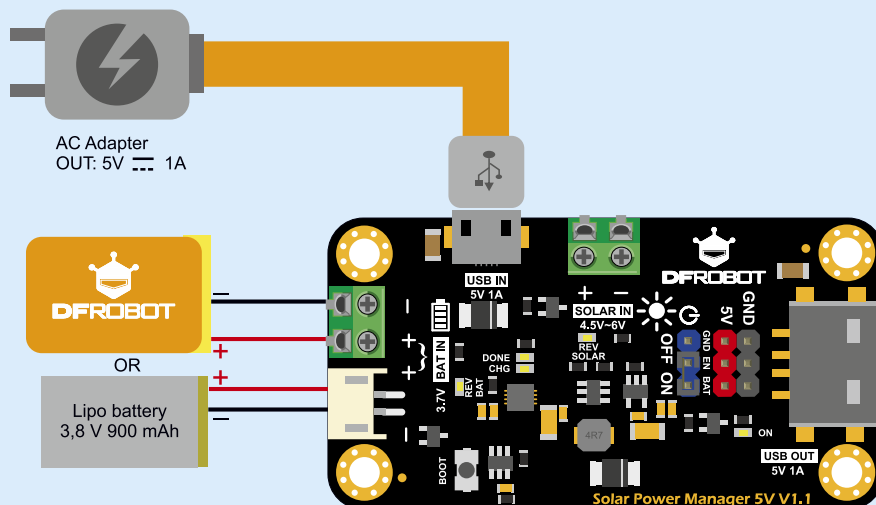


Rysunek 6. Układy regulatorów napięcia wyjściowego [1]

EN_VOUT (aktywne poziomem wysokim 3,3 V lub 5 V). Stan włączenia zasilania gniazdka USB OUT jest sygnalizowany świeceniem zielonej diody LED (LD5).

Linia EN_VOUT jest dołączona do środkowego wyprowadzenia listwy P5 (niebieskie, oznaczenie Q). Pozostałe wyprowadzenia złącza są dołączone do linii VBAT (oznaczenie „ON”) oraz masy modułu (oznaczenie „OFF”). Zwora założona na pozycję ON włącza zasilanie dla gniazdka USB OUT. Masa „GND VOUT” gniazdka VOUT (USB) jest odizolowana od masy modułu „GND”. Przy braku połączenia pomiędzy masami gniazdka USB OUT nie ma dostarczanego zasilania. Włączony tranzystor M3 daje połączenie pomiędzy tymi masami i włącza zasilanie gniazdka wyjściowego.

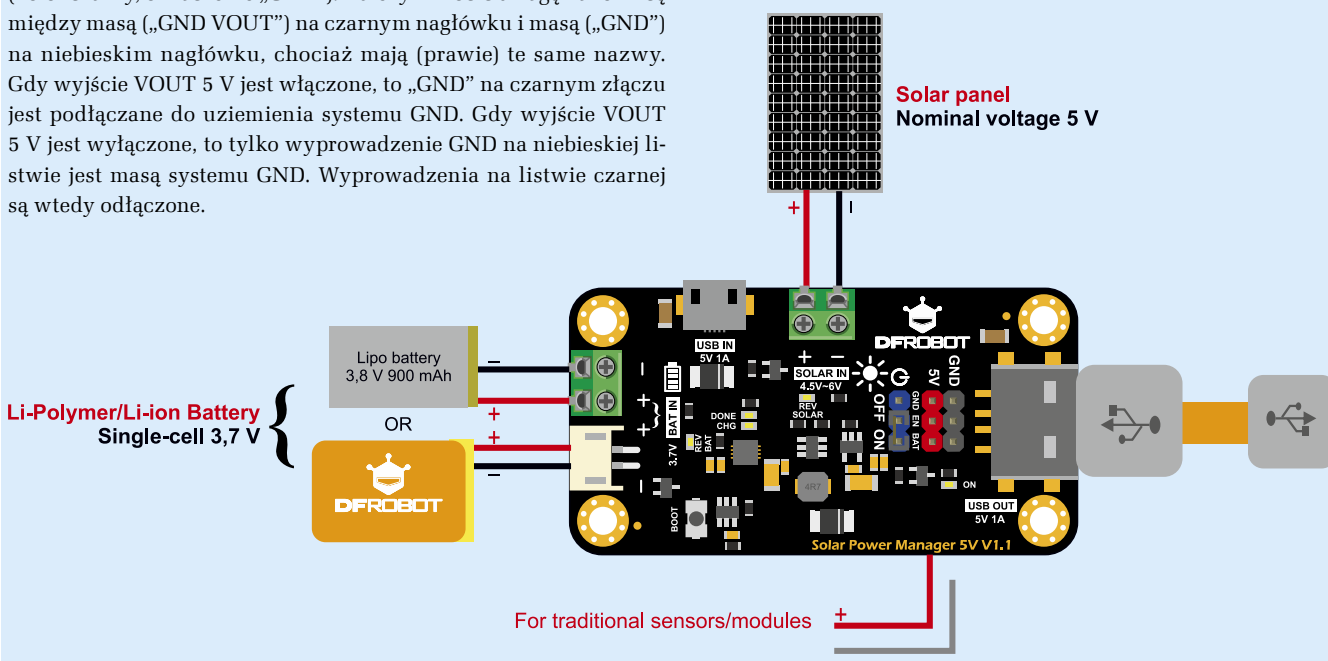
Napięcie wyjściowe VOUT (5 V, 1 A) jest także dołączone do trzech wyprowadzeń listwy P4 (kolor czerwony, oznaczenie „5V”). Całkowita moc wyjściowa gniazdka USB OUT i złącza 5 V nie powinna przekraczać $5 \text{ V} \cdot 1,5 \text{ A} = 7,5 \text{ W}$. Masa wyjściowa „GND VOUT” jest dołączona do wszystkich wyprowadzeń listwy P3 (kolor czarny, oznaczenie „GND”). Należy zwrócić uwagę na różnicę między masą („GND VOUT”) na czarnym nagłówku i masą („GND”) na niebieskim nagłówku, chociaż mają (prawie) te same nazwy. Gdy wyjście VOUT 5 V jest włączone, to „GND” na czarnym złączu jest podłączane do uziemienia systemu GND. Gdy wyjście VOUT 5 V jest wyłączone, to tylko wyprowadzenie GND na niebieskiej listwie jest masą systemu GND. Wyprowadzenia na listwie czarnej są wtedy odłączone.



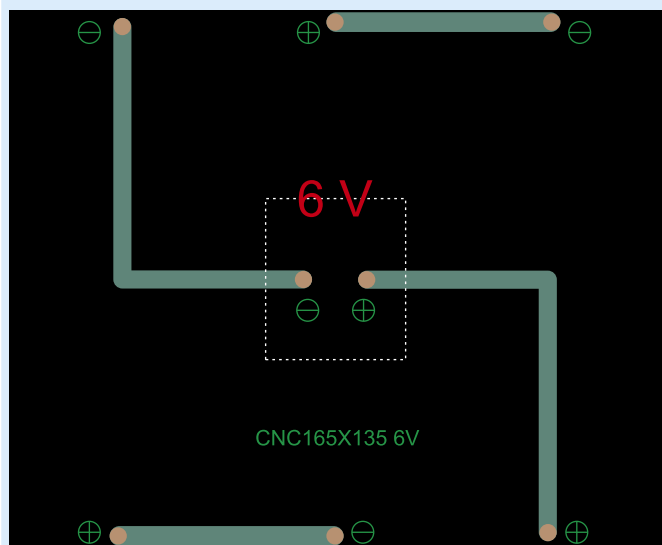
Rysunek 7. Konfiguracja ładowania akumulatora przez USB [1]

Używanie wejścia USB do ładowania akumulatora

Ładowanie akumulatora za pomocą panelu słonecznego typowo jest powolne i zależne od natężenia oświetlenia. Szczególnie trwa



Rysunek 8. Podstawowa konfiguracja pracy modułu DFR0559 [1]



Rysunek 9. Panel słoneczny CNC165X135 3,5 W, 6 V [3]

to długo, gdy akumulator jest bardzo rozładowany. Ładowanie akumulatora przez USB umożliwia szybkie jego naładowanie (na przykład w nocy). Wtedy ładowanie za pomocą panelu słonecznego umożliwia w ciągu dnia uzupełnienie na bieżąco zużytej energii.

Konfiguracja połączeń podczas ładowania akumulatora przez USB jest pokazana na **rysunku 7**. Do modułu jest podłączony akumulator (wejście BAT IN) oraz ładowarka z wtykiem microUSB (wejście USB IN). Może być także dołączony panel słoneczny do wejścia SOLAR IN. Źródło ładowania zostanie automatycznie przełączone na napięcie z wejścia USB IN. Podłączenie napięcia do wejścia USB IN powoduje od razu wystawienie napięcia wyjściowego 5 V, bez konieczności przyciskania przycisku BOOT, a nawet bez konieczności podłączania akumulatora do wejścia BAT IN. Maksymalny prąd ładowania akumulatora wynosi 900 mA, bez względu na moc dołączonej ładowarki.

Podstawowa konfiguracja pracy modułu DFR0559

Podstawowa konfiguracja pracy modułu została pokazana na **rysunku 8**. Napięcie wyjściowe 5 V można pobierać jednocześnie z gniazdka USB OUT oraz z listew 5 V – GND (czerwona – czarna). Według dokumentacji można pobierać maksymalnie 1,5 A prądu.

Panel słoneczny

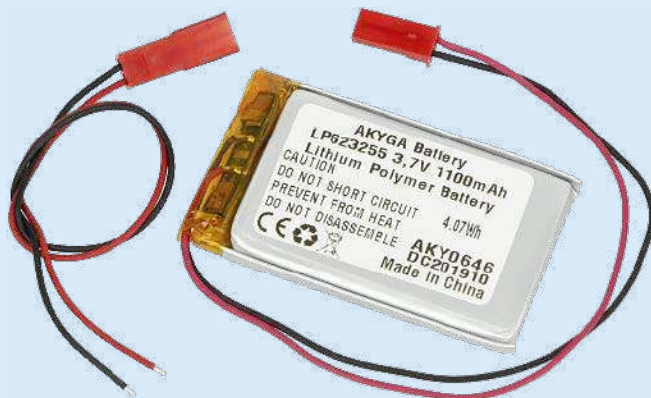
Do prób został zastosowany panel słoneczny CNC165X135 (AKU-06988, Botland) o mocy 3,5 W i napięciu wyjściowym 6 V (580 mA) [3]. Wymiary płyty to: 165×135×3 mm. Wyprowadzeniami panelu są pola lutownicze (pady) do przylutowania przewodów (**rysunek 9**). Panel zawiera monokrystaliczne ogniwa o wysokiej wydajności z wodoodporną powłoką poliuretanową wzmocnioną płytą pilśniową. Dzięki solidnemu uszczelnieniu panel słoneczny jest niezwykle trwały i niezawodny.

Akumulator

Akumulator marki Akyga został wykonany z ogniwo Li-Pol (lito-polimerowych) o napięciu znamionowym 3,7 V i pojemności 1100 mAh [4]. Cechuje go duża gęstość energii, niewielkie wymiary, mała masa i powolne samorozładowywanie. Akumulator jest wyposażony w przewód o długości 150 mm zakończony 2-pinowym złączem JST typu żeńskiego (raster 2,54 mm). W zestawie jest konektor do złącza JST (**fotografia 1**).

Specyfikacja akumulatora Li-Pol jest następująca:

- zakres temperatury pracy: -20...+60°C;
- nominalny prąd ładowania/rozładowania: 0,2 C (0,22 A)/0,2 C;



Fotografia 1. Akumulator Li-Pol 1100 mA, 3,7 V [4]

- maksymalny prąd ładowania/rozładowania: 1 C (1,1 A)/1 C;
- zabezpieczenie: PCM;
- wymiary: 55×32×6,2 mm.

Próby praktyczne

Praktyczna praca z pozyskiwaniem energii słonecznej została przeprowadzona z użyciem następujących elementów:

- moduł zarządzania energią słoneczną DFR0559 Power Manager firmy DFRobot [2];
- ogniwo słoneczne CNC165X135 3,5 W/6 V 165×135×3 mm (Botland) [3];
- akumulator Li-Pol Akyga 1100 mAh 1S 3,7 V (Botland) [4].

Pomiar napięcia na zaciskach panelu słonecznego woltomierzem jest prosty. Dla panelu wystawionego bezpośrednio na światło słoneczne bez obciążenia pomiar daje poziom maksymalnego napięcia panelu (układ otwarty). Podobnie prosty jest pomiar prądu zwarciovego dostarczanego przez panel słoneczny realizowany bezpośrednio przez amperomierz. Trzeba wtedy pamiętać o wewnętrznej rezystancji amperomierza włączonej szeregowo w obwód. Jednak typowo jest ona nieduża i znacząco nie zmienia działania układu. Tak samo, poprzez szeregowo włączenie amperomierza, można zmierzyć prąd pobierany z panelu słonecznego podczas pracy z modulem.

Pomiar napięcia i prądu akumulatora wygląda podobnie, jednak należy go wykonywać bezpośrednio na wyprowadzeniach akumulatora. Zacisk ujemny akumulatora nie jest połączony bezpośrednio z masą modułu. Napięcie zacisku dodatniego względem masy modułu dosyć mocno się zmienia. Pomiar napięcia wyjściowego 5 V ponownie należy wykonywać w stosunku do masy wyjściowej „GND VOUT”, która jest dołączona do wszystkich wyprowadzeń listwy P3 (kolor czarny, oznaczenie „GND”). Należy pamiętać, aby po podłączeniu akumulatora do modułu przycisnąć przycisk „BOOT”. Dopiero wtedy zostanie aktywowana praca modułu.

Próby wykazały, że układ CN3065 stara się utrzymać napięcie pracy panelu słonecznego VMP na poziomie ok. 4,6 V, co stanowi ok. 75...80% napięcia nieobciążanego panelu ok. 5,6...6,45 V, przy prądzie ładowania ok. 85...92% prądu zwarciovego. Pomiar nie są łatwe ze względu na zmiany warunków oświetlenia, które mocno rzutują na wyniki.

Została przeprowadzona próba długotrwałego obciążenia modułu prądem 24 mA (rezystor 200 Ω, 0,1%). Początkowo akumulator był całkowicie naładowany VBAT=4,13 V (świeci DONE). Szybko moduł przeszedł w tryb ładowania (świeci CHG) z dosyć zmiennymi warunkami słonecznymi (prąd ładowania malejący od ok. 11 mA do zera po 4 godzinach, VBAT=4,07 V). Dalej były warunki nocne z brakiem doładowania. Po kolejnych 4 godz. było VBAT=3,88 V a po następnych 9 godzinach 3,64 V. Pomimo słonecznego następnego dnia i ciągłego ładowania (w szczycie 110 mA) napięcie na koniec wynosiło 3,591 V, czyli całkowita pozyskana energia był mniejsza niż pobór stałym prądem 24 mA.

Podłączenie do wejścia SOLAR IN zasilacza regulowanego pokazuje, że dioda CHG zapala się przy wzroście napięcia wejściowego do ok. 3,6 V. Jednak dla prądu obciążenia 24 mA, 5 V oznacza to kontynuowanie rozładowania akumulatora. Dopiero od napięcia ok. 4,36 V rozpoczyna się ładowanie akumulatora. Prąd ładowania wzrasta przy wzroście napięcia wejściowego. Ale potem maleje i dla napięcia na akumulatorze ok. 4,08 V prąd spada do zera. Dioda CHG świeci, ale akumulator nie zostaje naładowany pomimo napięcia wejściowego nawet ok. 6 V.

Podsumowanie

W warunkach dużego poziomu oświetlenia stosowanie układu pozyskania energii ze światła sprawdza się dobrze. Ustawione na stałe napięcie wyjściowe (5 V) układu CN3065 sprawia kłopot przy próbie bezpośredniego zasilania typowych układów komunikacji bezprzewodowej – z napięciem około 3,3 V.

Badania praktyczne użycia modułu DFR0559 pokazały mankament polegający na braku wskazywania poziomu naładowania i rozładowania akumulatora. Pokazywane są tylko dwa stany akumulatora: naładowany (świeci LED DONE) oraz ładowany (świeci LED CHG). W większości zakresu pracy akumulatora pomiar napięcia na nim nie pokazuje stopnia rozładowania. Dla systemów

IoT wymagających niezawodnej pracy długookresowej może to być bardzo duży problem.

Henryk A. Kowalski

Instytut Informatyki, Politechnika Warszawska

Literatura:

- [1] Solar Power Manager 5 V SKU DFR0559, DFRobot, <https://bit.ly/3yCZBZS>
- [2] DFRobot Solar Power Manager – Moduł zarządzania energią słoneczną – 5 V, Botland, <https://bit.ly/3ulygrT>
- [3] Ogniwo słoneczne 3,5 W/6 V 165×135×3 mm, AKU-06988, Botland, <https://bit.ly/2SuvK5r>
- [4] Akumulator Li-Pol Akyga 1100 mAh 1S 3,7 V, Botland, <https://bit.ly/3vhV9O9>
- [5] Lithium Ion Battery Charger for Solar-Powered Systems CN3065, Consonance Electronic, <https://bit.ly/3vmOAd3>
- [6] CN3065 Układ zarządzania ładowaniem baterii litowej, który może być zasilany przez panel słoneczny, Consonance Electronic, (język chiński), <https://bit.ly/2RDLzqr>
- [7] Harvest Energy from a Single Photovoltaic Cell, Nathan Bourgoine, Journa; of Ananlog Innovation, April 2011, Analog Devices (Linear Technology), <https://bit.ly/3yzuuP3>

REKLAMA

ELEKTRONIKA PRAKTYCZNA

MATERIAŁY DODATKOWE

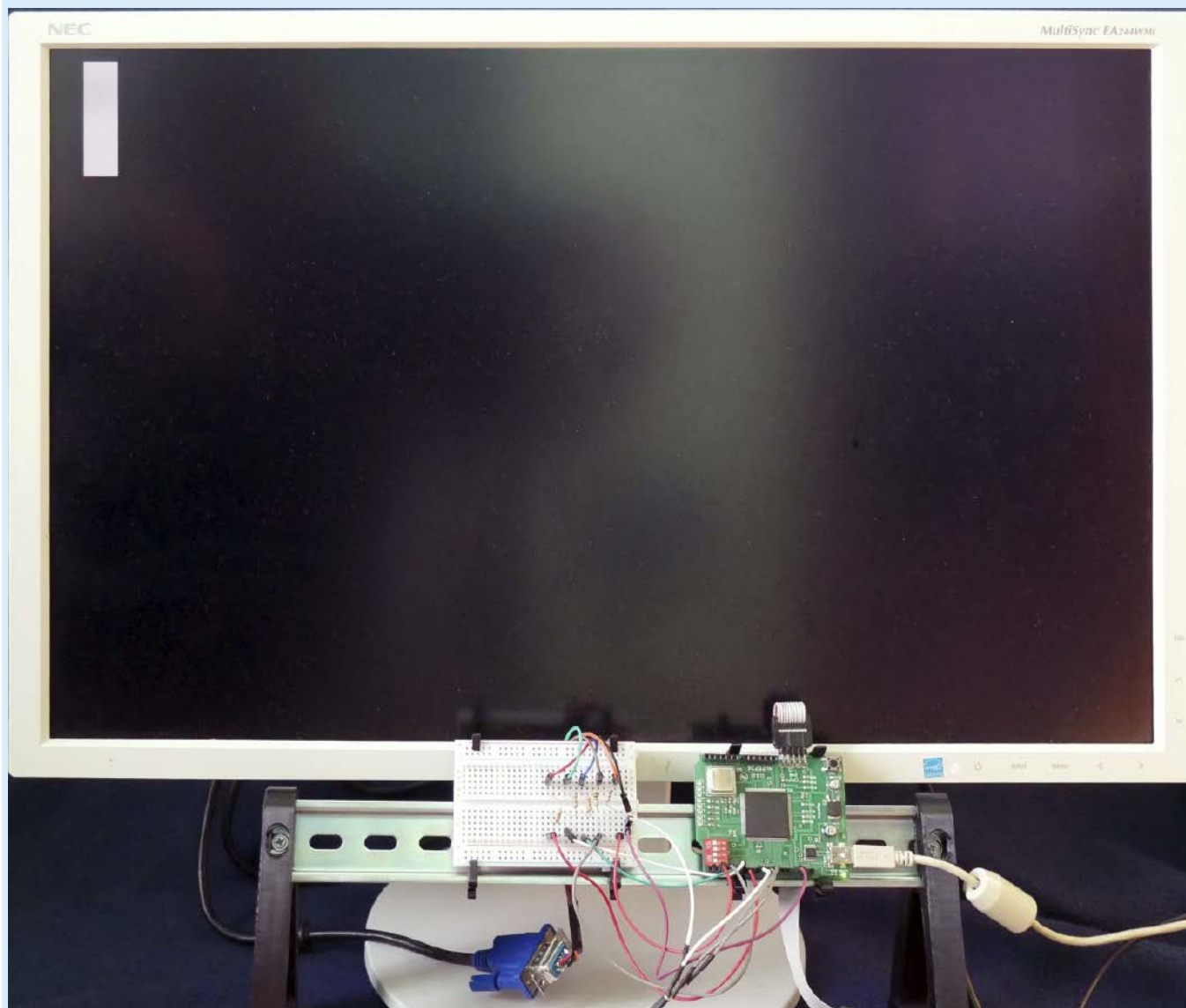


MEDIA

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 do biblioteki osobistej,
4. Odpowiedzieć na proste pytanie dotyczące bieżącego numeru,
5. Pobrać pliki.





Eksperymenty z FPGA (19)

Coś dla graczy

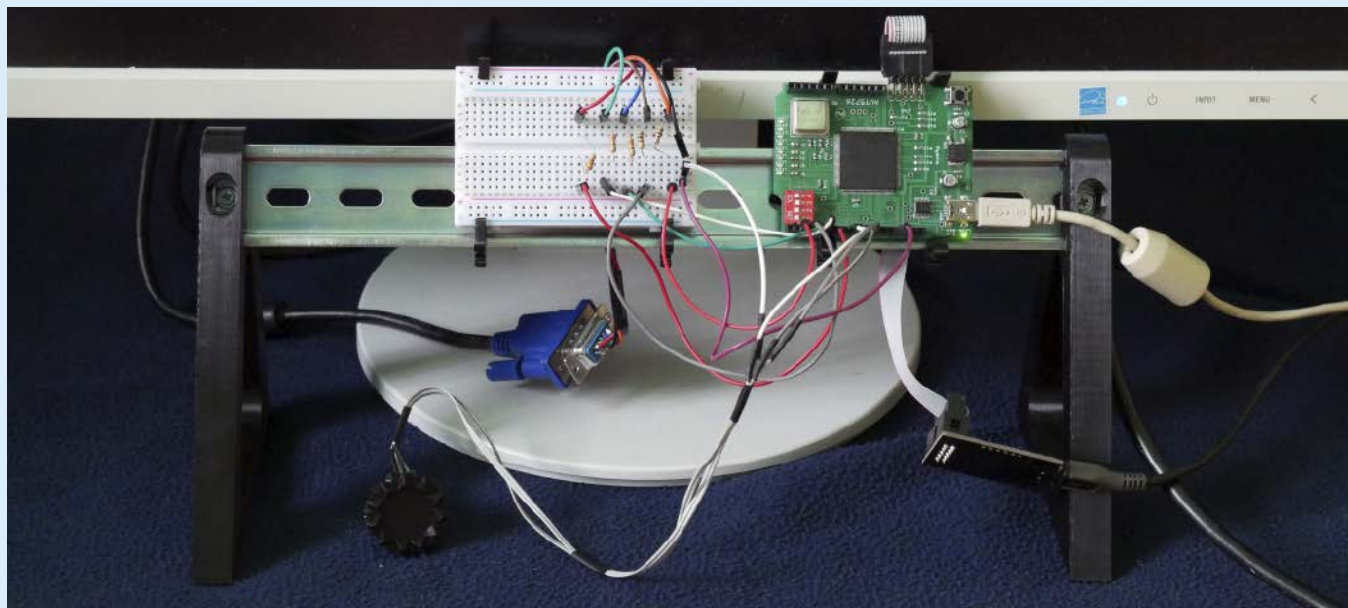
Umiemy już obsługiwać interfejs VGA. Teraz spróbujemy użyć go do wyświetlenia gry PONG. Tak jak poprzednio przed przystąpieniem do wykonywania eksperymentów zachęcam do aktualizacji repozytorium z przykładami [1]. Można to szybko wykonać poprzez wywołanie polecenia **git pull**.

Niemal każdy spotkał się z grą Pong. Istnieje ona od niemal 50 lat. Jest to prosta symulacja tenisa stołowego. Poruszając paletką, staramy się odbić piłkę z nadzieją, że przeciwnikowi nie uda się do niej dotrzeć na czas.

Aby umożliwić sterowanie, potrzebujemy kontrolera dla gracza. Użyjemy w tym celu enkodera inkrementalnego z przyciskiem. Sposób jego podłączenia do płytki Rysino został pokazany na **rysunku 1**. Poza tym użyjemy także portu VGA, ale tę część już znamy z poprzednich odcinków.

Połączone wszystkie elementy naszej „konsoli do gier” prezentuje **fotografia 1**. Enkodera nie dało się pewnie zamocować w płytce stykowej, więc dla podniesienia komfortu gry warto dołutować do niego przewody. Płytki zostały zamocowane na szynie DIN (TH35) za pomocą wydrukowanych zaczepów. Dzięki temu całość jest bardziej stabilna i można ją łatwiej przenosić z miejsca na miejsce. Mamy gotowy sprzęt, więc musimy przygotować oprogramowanie.

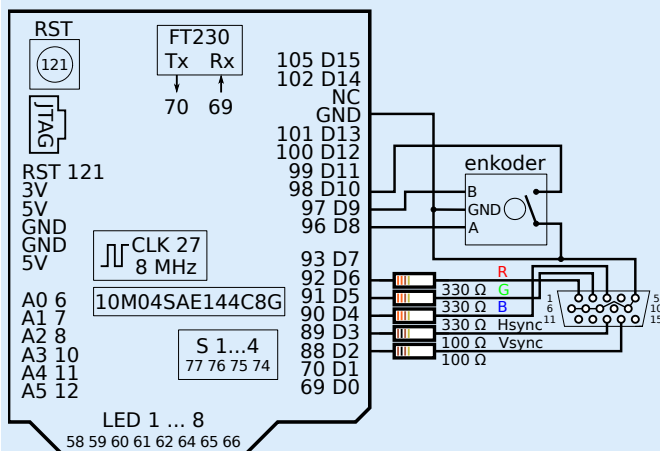




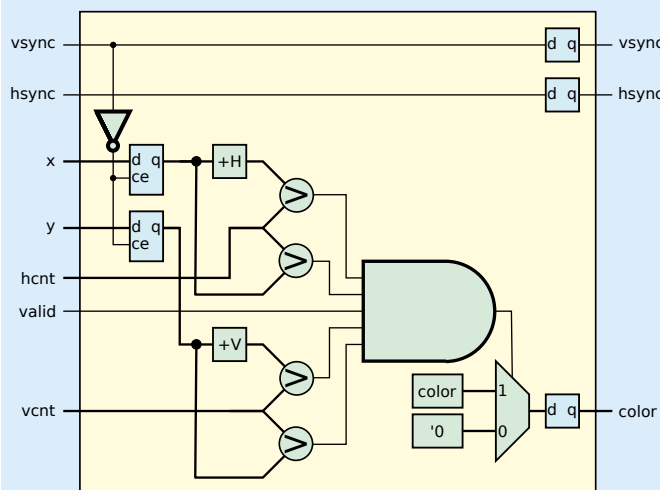
Fotografia 1. Połączone elementy zestawu testowego

Rysujemy prostokąt

Grafika w naszej grze będzie bardzo prosta. Potrzebna nam będzie tylko jedna figura geometryczna: prostokąt. Jak się okaże, jego narysowanie jest całkiem proste. Zaprojektujemy moduł, który będzie przyjmował współrzędne lewego górnego rogu figury i używał do zaświecenia odpowiednich pikseli. Schemat blokowy projektu prezentuje rysunek 2.



Rysunek 1. Schemat połączenia peryferiów do Rysino



Rysunek 2. Moduł wyświetlający prostokąt

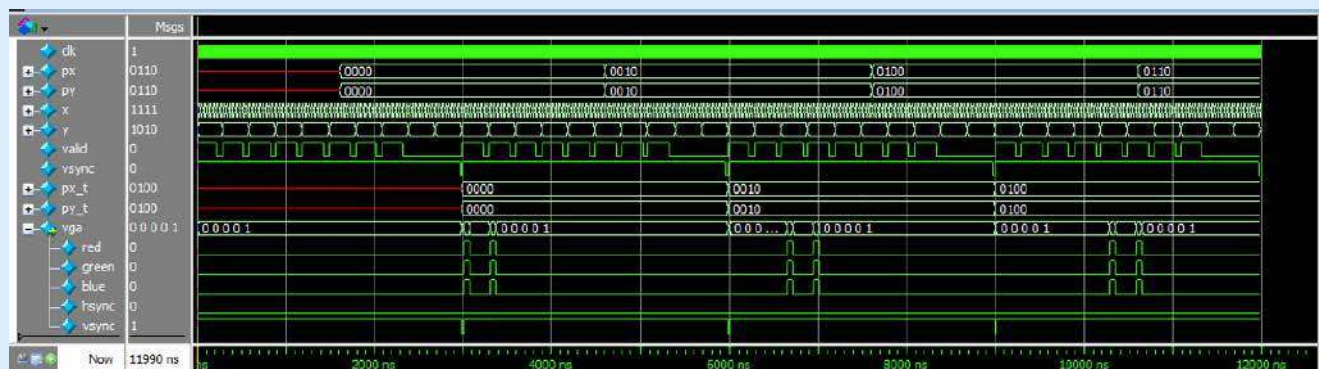
Współrzędne określające położenie są zapisywane w rejestrach, gdy aktywny jest sygnał synchronizacji pionowej (czyli w stanie niskim). Dzięki temu położenie figury nie zmienia się w środku

Listing 1. Struktura vga_t (15_VGA/vga_pkg.sv)

```
10 package vgaPkg;
11
12 typedef struct packed {
13     logic red;
14     logic green;
15     logic blue;
16     logic hsync;
17     logic vsync;
18 } vga_t;
19
20 endpackage : vgaPkg
```

Listing 2. Implementacja modułu rectangle (16_PONG/rectangle.sv)

```
10 module rectangle #(
11     parameter V = 20,
12     parameter H = 80,
13     parameter color = 3'b111,
14     parameter logX = 9,
15     parameter logY = 8
16 ) (
17     input wire clk,
18     input wire [logX-1:0]px,
19     input wire [logY-1:0]py,
20     input wire [logX-1:0]x,
21     input wire [logY-1:0]y,
22     input wire valid,
23     input wire vsync,
24     input wire hsync,
25     output vgaPkg::vga_t vga
26 );
27
28 logic x_in, y_in;
29 logic [logX-1:0]px_t;
30 logic [logY-1:0]py_t;
31
32 always_ff @(posedge clk)
33     if (vsync == 1'b0)
34         {px_t, py_t} <= {px, py};
35
36 assign x_in = x >= px_t && x < (px_t+H);
37 assign y_in = y >= py_t && y < (py_t+V);
38
39 always_ff @(posedge clk)
40     if (x_in && y_in && valid)
41         {vga.red, vga.green, vga.blue} <= color;
42     else
43         {vga.red, vga.green, vga.blue} <= '0;
44
45 always_ff @(posedge clk) begin
46     vga.hsync <= hsync;
47     vga.vsync <= vsync;
48 end
49 endmodule
```



Rysunek 3. Symulacja wyświetlania prostokąta

Listing 3. Generowanie wymuszenia dla symulacji (16_PONG/rectangle_tb.sv)

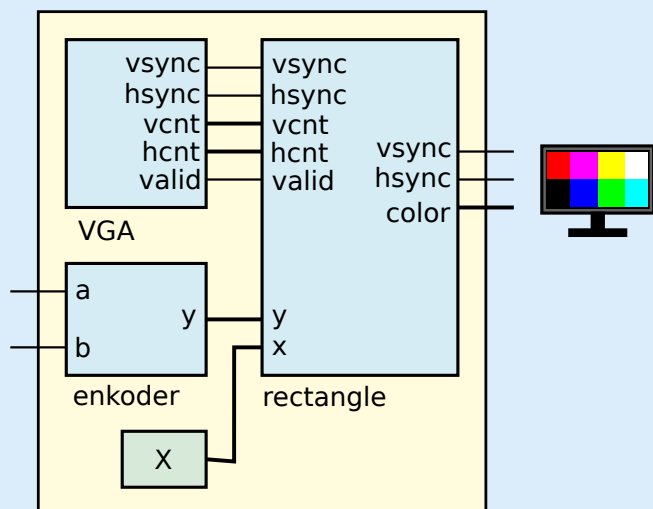
```

26  initial begin
27      for (frame = 0; frame < 4; frame++)
28          for (y = 0; y < 10; y++)
29              for (x = 0; x < 15; x++)
30                  @(posedge clk);
31  $stop;
32  end
33
34  assign vsync = !(y == 9 && x == 14);
35  assign valid = (y < 8 && x < 12);
36
37  always_ff @(posedge clk)
38      if (x == 5 && y == 5) begin
39          px = 2*frame;
40          py = 2*frame;
41      end

```

ramki. Wymiary prostokąta są określone przez dwie stałe: **H** (pionowa) i **V** (pionowa). Cztery komparatory sprawdzają, czy obecnie przetwarzany piksel o współrzędnych **hcnt** i **vcnt** należy do prostokąta. Na tej podstawie multiplexer wybiera, czy na wyjście zostanie podany kolor czarny (wektor zerowy), czy zdefiniowany przez stałą **color**. Wyjście modułu zatrzasknięte jest w przerzutnikach. Sygnały synchronizacyjne po prostu „przepływają” przez moduł. Dzięki temu mają takie samo opóźnienie, jak sygnał mówiący o kolorze.

Aby zmniejszyć liczbę sygnałów, które musimy podłączać, stworzymy strukturę **vga_t**, której kod został pokazany na **listingu 1**. Mamy w niej wszystkie pięć sygnałów złącza VGA. Możemy jej użyć nie tylko wewnątrz modułów, ale także jako wejścia i wyjścia. Jej użycie zostało pokazane m.in. na **listingu 2**, który prezentuje moduł **rectangle**. Stanowi w nim jedyne wyjście. Ale zacznijmy od początku. Definiujemy tu pięć parametrów. Pierwsze – **V i H**, to wymiary, a **color** to kolor prostokąta. Ostatnie dwa (**logX** i **logY**) to liczba bitów potrzebnych do reprezentacji współrzędnej piksela. Użyjemy



Rysunek 4. Połączenie generowania prostokąta z blokiem VGA

ich między innymi do zdefiniowania rozmiarów wejść: położenia figury oraz współrzędnych piksela. Poza tym dostajemy także sygnał **valid**, synchronizację oraz oczywiście zegar. Nie mamy tu resetu – nie jest on potrzebny, ponieważ poprawny stan ustali się najpóźniej po pierwszej ramce obrazu.

W liniach 31...34 widzimy zatrzasknięcie położenia prostokąta wywołane przez sygnał synchronizacji pionowej (czyli raz na ramkę). Samo sprawdzenie, czy obecny piksel należy do prostokąta, zaczyna się w liniach 35...36. Testujemy osobno oś **x** i **y**, a wynik zapisujemy w tymczasowych zmiennych. Używamy słowa kluczowego **assign**, więc tworzymy tu logikę kombinacyjną. Korzystamy z nich w wierszu 39, gdzie utworzony jest multiplexer wybierający stan obecnego piksela. Na końcu umieszczamy jeszcze rejestry dla sygnałów synchronizacyjnych (44...47).

Aby sprawdzić nasz nowy moduł w symulacji, musimy przygotować testbench. Jego fragment odpowiedzialny za generowanie wymuszeń widzimy na **listingu 3**. Najpierw, w wierszach 26...32, ustalamy czas trwania symulacji na cztery ramki. Aby uprościć symulację ustalamy rozmiar ekranu na 15×10 pikseli. Nie jest to „prawdziwy” format, obsługiwany przez wyświetlacz. Za to jest na tyle mały, że wyniki będą łatwe do interpretacji. Sygnały synchronizacyjne powstają w liniach 34 i 35. Pozostaje nam jeszcze ustalenie położenia prostokąta, które będzie zmieniało się po każdej ramce (37...41).

Uruchomiamy symulację – włączamy program ModelSim, przechodzimy do folderu 16_PONG i wykonujemy skrypt: **do rectangle.do**

Uzyskane wyniki prezentuje **rysunek 3**. W pierwszej linii widzimy sygnał zegarowy. Dalej znajdują się pozostałe wejścia. Na samym końcu widzimy wyjście. W jednej linii jest wypisana cała struktura (opisana jako **vga**). Możemy ją rozwinąć i zobaczyć każdy z sygnałów składowych osobno. Zachęcam Czytelnika do przeanalizowania wygenerowanych przebiegów i rozstrzygnięcia, czy moduł zachowuje się zgodnie z oczekiwaniami.

Spróbujmy teraz połączyć nasz nowy moduł z generatorem sygnału VGA. Na początek naszym celem będzie jedynie wyświetlenie prostokąta symbolizującego paletkę gracza oraz zmiana jego pozycji za pomocą enkodera. Schemat modułu został pokazany na **rysunku 4**. Sama integracja modułu VGA z modulem **rectangle** jest prosta: łączymy wyjścia z wejściami. Natomiast sygnały wygenerowane przez blok **rectangle** trafią bezpośrednio na wyjścia układu FPGA. Pozostaje nam jeszcze obsługa enkodera. Nie będziemy się jednak w nią zagłębiać, ponieważ jest prawie identyczna jak ta, którą przygotowaliśmy w trzecim odcinku naszego cyklu. Jednak tym razem do licznika dodane zostało nasycenie. W grze spowoduje to, że dolna i górna krawędź ekranu nie są połączone.

Implementacja tego projektu została pokazana na **listingu 4**. Główną zmianą w stosunku do poprzednich eksperymentów z VGA jest użycie struktury jako wyjścia w module topowym. Jednak Quartus radzi sobie z tym bez żadnego problemu. Możemy przypisać każdy element struktury do poszczególnych wyjść układu FPGA. Sama



Rysunek 5. Symulacja wyświetlania paletki

zawartość modułu jest bardzo prosta: znajdziemy tu połączone razem instance znanych nam już modułów. Aby zwiększyć prędkość przewijania, wartość uzyskana z enkodera została pomnożona razy 8. Zostało to zrealizowane w linii 57 za pomocą dołączenia trzech zerowych bitów na początku wektora.

Do przetestowania modułu `rectangle_top` użyjemy monitora portu VGA przygotowanego w poprzednim odcinku. Sposób jego połączenia prezentuje **listing 5**. Poza nim testbench zawiera także generowanie sygnałów imitujących ruch enkodera. Symulację uruchamiamy poleceniem:

`do rectangle_top.do`

Jej wykonanie zajmie dość dużo czasu (do kilkudziesięciu minut). Zawiera ona ponad osiem pełnych ramek. W sprężeniu będzie to trwało nieco ponad jedną dziesiątą sekundy. Wygenerowane przebiegi prezentuje **rysunek 5**, a uzyskane obrazy **rysunek 6**. Analizując je, należy wziąć pod uwagę to, że pierwsza zapisana ramka nie

jest poprawnie zsynchronizowana. Dopiero kolejne odzwierciedlają efekt uzyskany na monitorze.

Wróćmy do przebiegów czasowych. Pierwsze dwa wiersze to zegar generowany przez rezonator kwarcowy oraz reset. Projekt jest taktowany zegarem `clk_vga` uzyskanym z pętli PLL. Sygnały `ea` i `eb` to wymuszenia imitujące działanie enkodera. Aktualną wartość zadaną przez enkoder widzimy w wierszu `py`. Natomiast `py_t` jest to położenie zatraskiwane co ramkę w module `rectangle`. Na końcu znajdziemy wyjściowy sygnał `vga`.

Testy w sprężeniu

Dla przetestowania efektów naszej pracy w sprężeniu uruchamiamy środowisko Quartus i ładujemy projekt `16_PONG/rectangle.qpf`. Posłuży nam on także do kolejnych testów, dlatego przed włączeniem budowy musimy ustawić moduł `rectangle_top` jako nadrzędną instancję projektu (*top level entity*). Teraz możemy rozpocząć budowę projektu. Gdy się zakończy, pozostanie nam już tylko zaprogramowanie płytki. Efekt został pokazany na fotografii otwierającej artykuł. Gdy poruszymy gałką enkodera, prostokąt powinien zmienić swoje położenie. Finalny efekt możemy także zobaczyć na filmie [2].

Pitka

Piłka będzie reprezentowana jako kwadrat. Jednym z celów postawionych przed naszym projektem jest rozdzielenie logiki gry od generowania obrazu. Tę drugą funkcję będzie pełnił moduł `rectangle`. Zastanowimy się teraz, w jaki sposób możemy generować ruch piłki tak, aby mógł on zająć się jej wyświetleniem.

Implementacja jest pokazana na **listingu 6**. Poza standardowym zegarem i resetem mamy tu jeszcze dwa wejścia. Pierwsze z nich, `game` mówi nam, czy obecnie trwa rozgrywka, a drugie `ce`, czy należy obliczyć kolejną klatkę animacji. Mamy także trzy wyjścia: pozycja piłki w dwóch osiach oraz `reflection`. Przyjmuje ono stan wysoki, gdy nastąpiło odbicie piłki od pionowych krawędzi ekranu. Sygnał ten przyda nam się przy rozstrzygnięciu, czy gra powinna się zakończyć.

Listing 4. Generowanie wymuszenia dla symulacji (16_PONG/rectangle_top.sv)

```
10 module rectangle_top #(
11     parameter H = 800,
12     parameter V = 525,
13     parameter H_BIT = $clog2(H),
14     parameter V_BIT = $clog2(V)
15 ) (
16     input wire clk,
17     input wire rst,
18     input wire a,
19     input wire b,
20     output vga_pkg::vga_t vga
21 );

33 encoder #(
34     .MAX((480-RECTANGLE_V)/8+1)
35 ) enc (
36     .clk(clk_vga),
37     .rst(rst),
38     .a(a),
39     .b(b),
40     .x(py));

42 vga #(H(H), .V(V)) vga_inst (
43     .clk(clk_vga),
44     .rst(rst),
45     .hsync(hsync_vga),
46     .vsync(vsync_vga),
47     .valid(valid_vga),
48     .hcnt(hcnt_vga),
49     .vcnt(vcnt_vga));

51 rectangle #(
52     .V(RECTANGLE_V), .H(RECTANGLE_H),
53     .logX(H_BIT), .logY(V_BIT)
54 ) rect (
55     .clk(clk_vga),
56     .px(10),
57     .py({py, 3'd0}),
58     .x(hcnt_vga),
59     .y(vcnt_vga),
60     .valid(valid_vga),
61     .vsync(vsync_vga),
62     .hsync(hsync_vga),
63     .vga(vga));
```

Listing 5. Generowanie wymuszenia dla symulacji (16_PONG/rectangle_top.sv)

```
55 rectangle_top dut (
56     .clk(clk),
57     .rst(rst),
58     .a(ea),
59     .b(eb),
60     .vga(vga));

62 vga_monitor #(
63     .NAME("rectangle")
64 ) monitor (
65     .clk(dut.clk_vga),
66     .rst(rst),
67     .c({vga.red, vga.green, vga.blue}),
68     .hsync(vga.hsync),
69     .vsync(vga.vsync));
```

Listing 6. Generowanie ruchu piłki (16_PONG/ball.sv)

```

10 module ball #(
23 ) (
24   input wire clk,
25   input wire rst,
26   input wire game,
27   input wire ce,
28   output logic reflection,
29   output logic [LOG_X-1:0] ball_x,
30   output logic [LOG_Y-1:0] ball_y
31 );

48   always_ff @(posedge clk)
49     if (!rst)
50       by_init <= BALL_Y_MIN;
51     else begin
52       if (by_init < BALL_Y_MAX-BALL_V)
53         by_init <= by_init + 1'd1;
54       else
55         by_init <= BALL_Y_MIN;
56     end

69   end else if (!game) begin

73     by_tmp <= by_init;

79   end else begin
80     if (ce) begin
81       bx_tmp <= bx + ball_vx;
82       by_tmp <= by + ball_vy;
83     end
84     if (ce1) begin
85       if (bx_tmp < BALL_X_MIN) begin
86         bx <= BALL_X_MIN;
87         ball_vx <= -ball_vx;
88         reflection_tmp <= '1;
89       end else if (bx_tmp > BALL_X_
MAX - BALL_H) begin
90         bx <= BALL_X_MAX - BALL_H;
91         ball_vx <= -ball_vx;
92         reflection_tmp <= '1;
93       end else begin
94         bx <= bx_tmp;
95         reflection_tmp <= '0;
96     end

```

W wierszach 48...56 widzimy licznik, który cały czas jest inkrementowany z częstotliwością głównego zegara. Jest on używany do losowania początkowej pozycji piłki. Jak widzimy w linii 79, gdy nie jest prowadzona gra, jego zawartość jest kopiowania do aktualnego położenia piłki w pionie.

W czasie rozgrywki pozycja piłki w obu osiach jest inkrementowana o aktualną prędkość (80...82). W następnym takcie zegara następuje sprawdzenie, czy położenie mieści się w zadanym zakresie. Jeżeli nie następuje odbicie, czyli zmiana znaku prędkości dla danej współrzędnej. Jego przebieg dla osi poziomej widzimy w wierszach 85...96. Wersja dla osi pionowej jest analogiczna.

Generowanie wymuszeń na potrzeby testów pokazuje listing 7. W wierszach 26...32 generowany jest sygnał **ce**. Docelowo będzie

Listing 7. Testy dla modułu ball (16_PONG/ball_tb.sv)

```

26   initial begin
27     ce <= 1'b0;
28     @(posedge clk);
29     forever begin
30       ce <= !ce;
31       @(posedge clk);
32     end
33   end
34
35   initial begin
36     game <= 1'b0;
37     repeat(10) @(posedge clk);
38     game <= 1'b1;
39     repeat(400) @(posedge clk);
40     game <= 1'b0;
41     repeat(40) @(posedge clk);
42     game <= 1'b1;
43     repeat(800) @(posedge clk);
44     $stop;
45   end

```

on w stanie wysokim przez jeden cykl zegara na każdą ramkę obrazu. Na potrzeby symulacji zmienia się jednak w każdym takcie zegara. Drugą częścią jest wytworzenie sygnału **game**, które znajdziemy w wierszach 35...45. W różnych odstępach czasu następuje jego zmiana, co pozwoli nam zaobserwować, w jaki sposób moduł będzie na niego reagował.

Symulację uruchamiamy rozkazem:

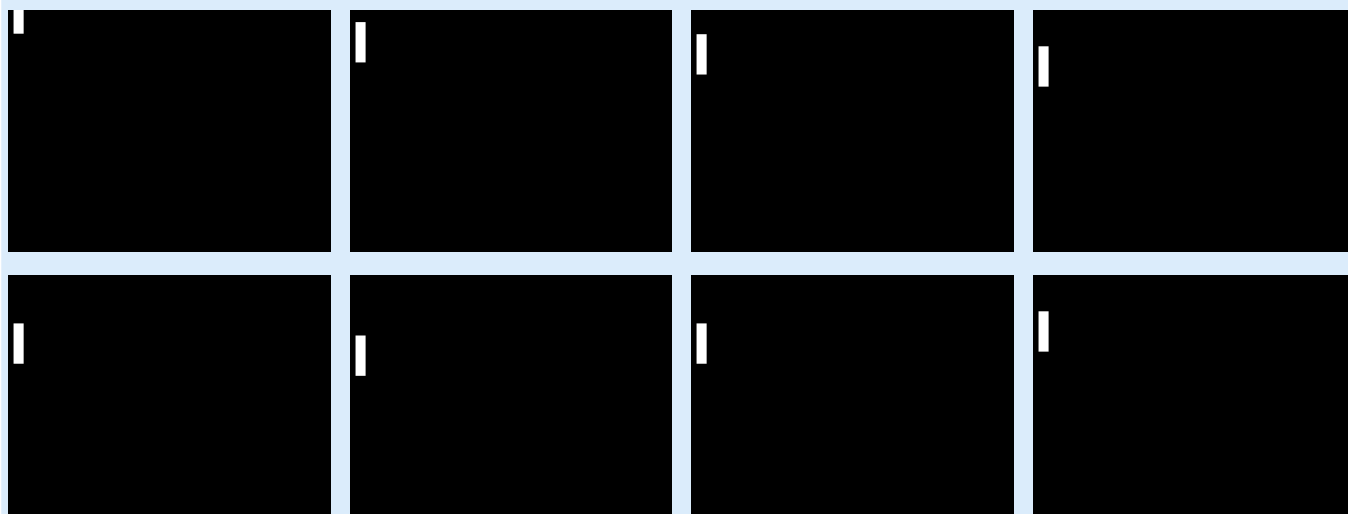
do ball.do

Wyniki prezentuje rysunek 7. Pierwsze dwa wiersze to zegar oraz reset. Dalej znajdziemy sygnał **game** oraz **ce**. Z wyjątkiem najbardziej interesujących nas **ball_x** oraz **ball_y**. Jest to aktualne położenie piłki. Wykresy pokazują je zarówno w formie tekstowej, jak i graficznej. Gdy gra jest wstrzymana (**game == 0**), oś pozioma jest ustawiana na środek ekranu, a na pionowej następuje „losowanie” pozycji. Widzimy także, że każde odbicie od pionowej krawędzi ekranu generuje stan wysoki na wyjściu **reflection**.

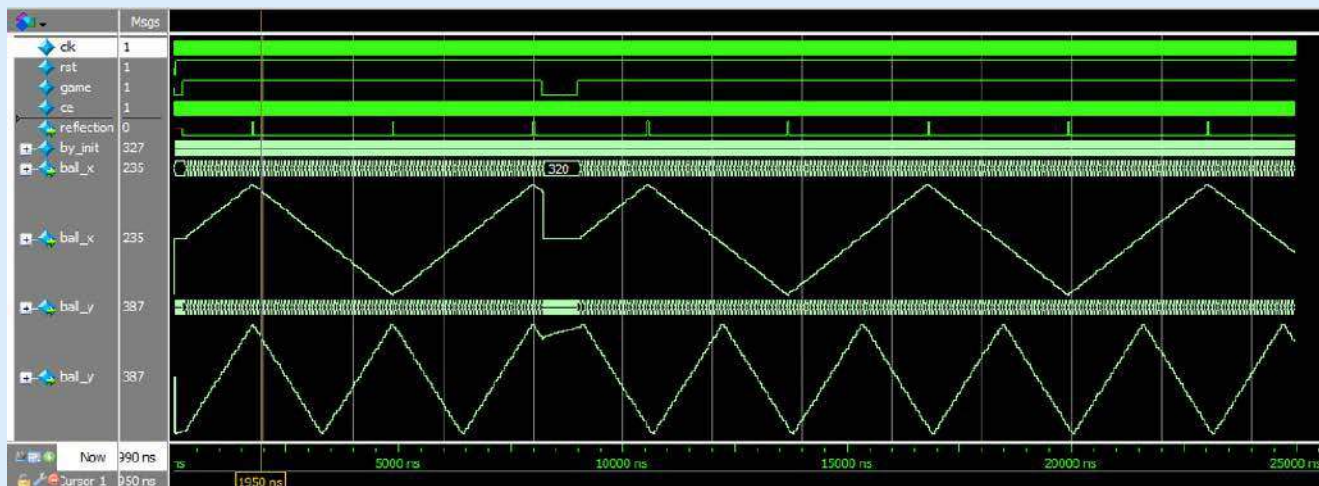
Spróbujmy teraz uruchomić samą piłkę w sprzęcie. W tym celu możemy wziąć poprzedni projekt i zastąpić piłką moduł enkodera. Uzyskany wynik pokazuje rysunek 8. Sygnał **ce** jest generowany na narastającym zboczu synchronizacji pionowej. Nowością jest sygnał **button** sterowany przez przycisk enkodera. Steruje on wejściem **game**.

Fragment implementacji tego rozwiązania widoczny jest na listingu 8. Sposób połączenia modułów jest klasyczny. Piłka jest kwadratem o boku długości 20 pikseli. Wymiary te są potrzebne zarówno przy obliczaniu, symulacji jak i wyświetlaniu. Sygnał **ce** przyjmuje stan wysoki na zboczu narastającym synchronizacji pionowej (wiersz 51).

Ponieważ symulowanie topowych modułów zajmuje sporo czasu, zrezygnowałem z niego w tym przypadku. Poprzestałem jedynie



Rysunek 6. Wyniki z monitora portu VGA



Rysunek 7. Symulacja ruchu piłki

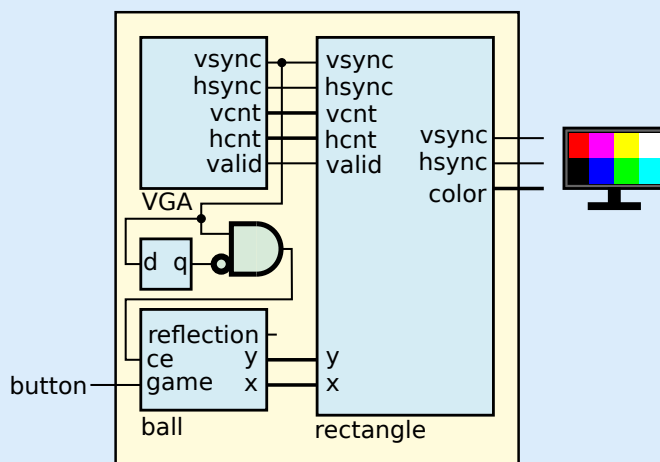
Listing 8. Połączenie modułu piłki (16_PONG/ball_top.sv)

```

35 vga #(.H(H), .V(V)) vga_inst (
36   .clk(clk_vga),
37   .rst(rst),
38   .hsync(hsync_vga),
39   .vsync(vsync_vga),
40   .valid(valid_vga),
41   .hcnt(hcnt_vga),
42   .vcnt(vcnt_vga));
43
44 always_ff @(posedge clk_vga)
45   vsync_vga_r <= vsync_vga;
46
47 ball #(.BALL_H(20), .BALL_V(20)) ball_inst (
48   .clk(clk_vga),
49   .rst(rst),
50   .game(button),
51   .ce(vsync_vga & !vsync_vga_r),
52   .ball_x(ball_x),
53   .ball_y(ball_y));
54
55 rectangle #(
56   .V(20), .H(20),
57   .logX(H_BIT), .logY(V_BIT)
58 ) rect (
59   .clk(clk_vga),
60   .px(ball_x),
61   .py(ball_y),
62   .x(hcnt_vga),
63   .y(vcnt_vga),
64   .valid(valid_vga),
65   .vsync(vsync_vga),
66   .hsync(hsync_vga),
67   .vga(vga));

```

na teście pojedynczych modułów. Generowanie obrazu piłki sprawdzimy za pomocą projektu *rectangle.qpf*. Przed rozpoczęciem budowy musimy jednak zmienić topowy moduł na *ball_top.sv*. Po uruchomieniu po ekranie będzie sunąć piłka odbijająca się od jego brzegów. Natomiast gdy naciśniemy przycisk, zobaczymy proces losowania pozycji startowej. Piłka zostanie przeniesiona na środek ekranu i z dużą prędkością będzie poruszać się po linii pionowej. Gdy zwolnimy



Rysunek 8. Połączenie piłki do modułu vga

przyciski, rozpocznie swój normalny ruch od ostatniego położenia. Wynik możemy zobaczyć także na filmie [2].

Podsumowanie

W tym odcinku wyświetliliśmy na ekranie prostokąt. W pierwszej wersji jego położenie było zadawane za pomocą enkodera, a w drugim widzieliśmy na bieżąco generowaną animację ruchu piłki. W następnym odcinku przygotujemy algorytm sterujący położeniem przeciwnika oraz uruchomimy całą grę PONG.

Rafał Kozik
rafkozik@gmail.com

Materiały dodatkowe:

[1] Repozytorium <http://bit.ly/33uYPxs>

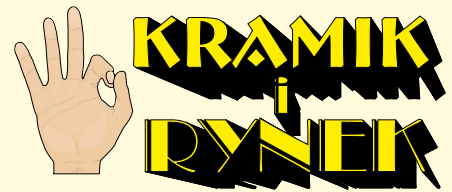
[2] Film demonstrujący działanie projektu <https://bit.ly/3fZkHsP>

Chcesz czytać nasze najnowsze artykuły jeszcze przed wydrukowaniem w EP? Zajrzyj na

www.ep.com.pl/EPwtoku



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 Krzykowski, tel. 22 257 84 60, e-mail: grzegorz.krzykowski@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 2021 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

RACK i Eurocarta 19" Wyposażenie szaf 19"
www.obudowa.pl
 Producent obudów dla elektroniki tel. 032-230-2301

www.ep.com.pl

Hurtownia elementów elektronicznych "AKSOTRONIK" zaprasza do swojego sklepu internetowego
 Zaloguj się i kupuj ON-LINE na naszej stronie:
WWW.AKSOTRONIK.COM.PL

Aksotronik
 ELEMENTY ELEKTRONICZNE

Magnesy neodymowe oraz ferrytowe
 Ceny od 0.10zł

Kostki elektryczne zaciskowe
 Ceny od 0.22zł

Szczotki węglowe do elektronarzędzi
 Ceny od 2.60zł/kpl

Pudełka/organizery
 Ceny od 0.95zł

Przełączniki klawiszowe wodoszczelne-pyłoszczelne
 Ceny od 2.40zł

Druty oporowe od 0.16 do 0.31mm
 Ceny od 5.70zł

Przewodniki do przewodów
 Ceny od 11.00zł

Złącza hermetyczne Superseal
 Ceny od 1.10zł /kpl

Przełączniki do elektronarzędzi zwykłe i elektromagnetyczne
 Ceny od 7.00zł

Zestawy śrubek M2, M3 z nakrętkami i podkładkami
 Ceny od 2.50zł

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 optoelektryczne), elementy dystansowe, złącza, przełączniki, elementy akustyczne, rezystory, kondensatory, kwarce, podstawki, moduły Arduino
 Zapraszamy do kontaktu: **INFO@aksotronik.com.pl, tel: (22) 783-20-51**

ELMAX 1988
OBWODY DRUKOWANE
 Produkcja, Projektowanie, Montaż

Certyfikat Underwriters Laboratories
 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
 Płytki czołowe 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

NINIEJSZE OGŁOSZENIA SĄ INFORMACJĄ HANDLOWĄ I NIE STANOWIĄ OFERTY W MYŚL ART. 66, § 1 KODEKSU CYWILNEGO. EWENTUALNIE PODANE CENY MOGĄ ULEC ZMIANIE.



Miesięcznik „Elektronika Praktyczna” (12 numerów w roku) jest wydawany przez AVT-Korporacja Sp. z o.o. we współpracy z wieloma redakcjami zagranicznymi.

Wydawnictwo:

AVT-Korporacja Sp. z o.o.
 03-197 Warszawa, ul. Leszczyńska 11
 tel. 22 257 84 99, faks 22 257 84 00

Wydawca:

Wiesław Marciniak

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:

Damian Sosnowski

Redaktor Programowy,

Przewodniczący Rady Programowej:

Piotr Zbysiński

Menedżer Magazynu:

Katarzyna Gugala

Szef Pracowni Konstrukcyjnej:

Grzegorz Becker

Redakcja strony internetowej **www.ep.com.pl**

MAD Sp. z o.o.

Zespół marketingu i reklamy:

Katarzyna Gugala, tel. 22 257 84 64
 Bożena Krzykawska, tel. 22 257 84 42
 Grzegorz Krzykowski, tel. 22 257 84 60

Sekretarz Redakcji:

Grzegorz Krzykowski, tel. 22 257 84 60

DTP i okładka:

MAD Sp. z o.o.

Stali Współpracownicy:

Nikodem Czechowski, Jakub Tyburski, 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



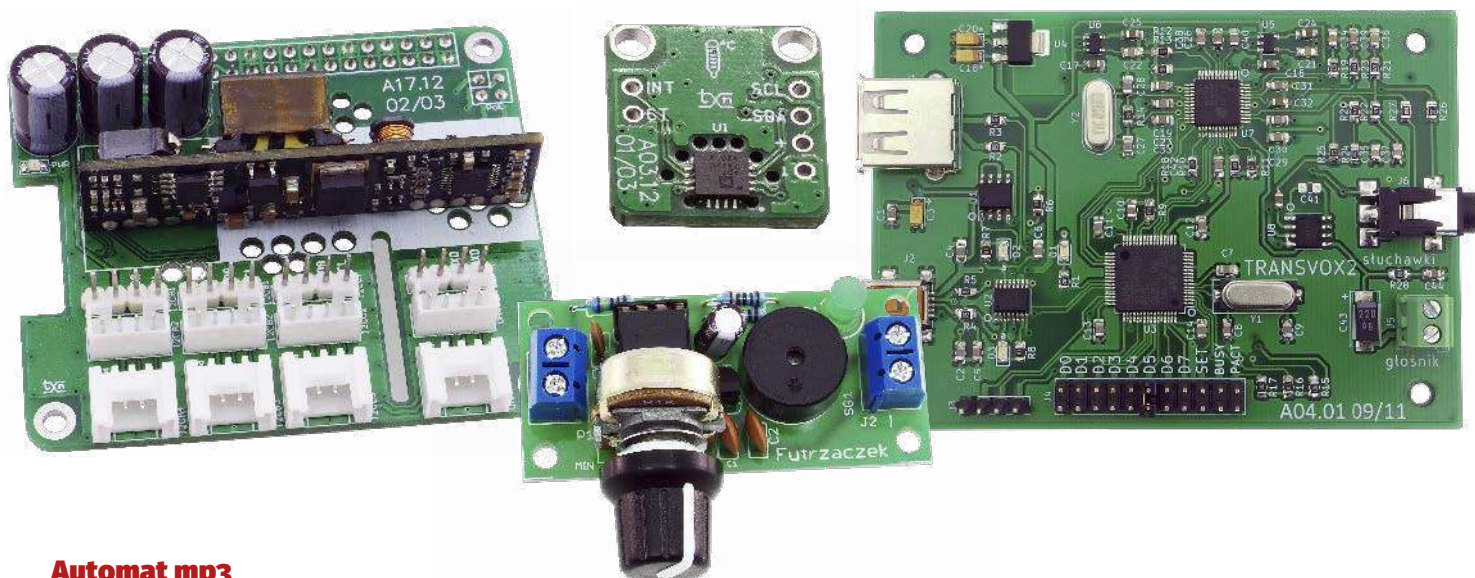
Wydawnictwo
 AVT-Korporacja Sp. z o.o.
 należy do **Izby Wydawców Prasy**

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 umieszczanie na stronach internetowych całości lub fragmentów publikacji zamieszczanych w „Elektronice Praktycznej” jest dozwolone wyłącznie po uzyskaniu zgody redakcji. Redakcja nie odpowiada za treść reklam i ogłoszeń zamieszczanych w „Elektronice Praktycznej”.





Automat mp3

Format plików mp3 jest powszechnie używany do zapisu muzyki i dźwięków. Projekt nazwany Transvox2, pomyślany został jako sterowany automat do szybkiego, selektywnego odtwarzania plików zgromadzonych na zewnętrznym nośniku, mogących być muzyką lub informacjami, np. reklamowymi. Do komunikacji z urządzeniem korzystamy z portu USB lub prostego interfejsu równoległego.

Zasilacz PoE dla Raspberry Pi

Zasada działania PoE (Power over Ethernet) jest prosta – oprócz transmisji danych, jednym standardowym przewodem przesyłane jest także zasilanie. W wersji podstawowej wykorzystane są wolne pary interfejsu 10 M/100 M (żyły 4, 5+7, 8), co pozwala na zastosowanie typowych gniazd i transformatorów sprzęgających. W wersji rozszerzonej do przesyłu zasilania wykorzystane są pary transmisyjne (żyły 1, 3+3, 6), co wymusza zastosowanie specjalizowanych gniazd zintegrowanych lub transformatorów sprzęgających zdolnych do pracy z większymi prądami uzwojeń. Zaprezentowana nakładka ma wbudowany zasilacz PoE zgodny z IEEE802.3af i może dostarczyć 5 V przy obciążeniu do 2,5 A (max. 13 W).

Wykrywacz zadanej rezystancji

Szukanie zwarc w układach i systemach elektronicznych to niełatwe zadanie. Można się o tym przekonać zwłaszcza wtedy, kiedy szukany upływ ma rezystancję kilkudziesięciu lub kilkuset omów, na co nie reagują typowe sygnalizatory akustyczne w multimetrach. Próg detekcji w prezentowanym układzie jest płynnie regulowany, przez co można nim szybko „przedzwonić” podejrzane obwody.

Dokładny przetwornik temperatury z układem ADT7422

Rozwój systemów monitorujących parametry życiowe człowieka, no i oczywiście istniejąca pandemia wymaga jako podstawowego parametru świadczącego o kondycji organizmu dokładnego pomiaru temperatury ciała. Coraz więcej producentów oferuje scalone przetworniki temperatury o zwiększonej dokładności. Jednym z nich jest zastosowany w minimodule układ ADT7422 opracowany przez Analog Devices. W swojej strukturze integruje sensor temperatury, układ przetwornika ADC, komparatory progowe oraz układ komunikacji I²C. Dokładność przetwarzania pozwala spełnić normy termometru do pomiaru temperatury ciała pacjenta (ASTM E1112), zachowując dokładność $\pm 0,1^\circ\text{C}$ w zakresie $25...50^\circ\text{C}$ i $\pm 0,25^\circ\text{C}$ w zakresie $-20...105^\circ\text{C}$, bez potrzeby kalibracji przetwornika.

a ponadto tematy wiodące EP 7/2021:

- BLUETOOTH 5 WE WŁASNYM PROJEKCIE
- INTERFEJSY BEZDOTYKOWE
I STEROWANIE GESTAMI

Wykaz firm ogłaszających się w tym numerze „Elektroniki Praktycznej”

AKSOTRONIK.....	122
ARMEL	122
BORNICO.....	13
COMPUTER CONTROLS	
.....	11, 56
ELMAX.....	122
GAMMA	9
KAMAMI	88
MICROCHIP.....	124
NOKIA	62, 63
PIEKARZ	9
RENEX	9
TME	7, 58



Przebij się przez zakłócenia

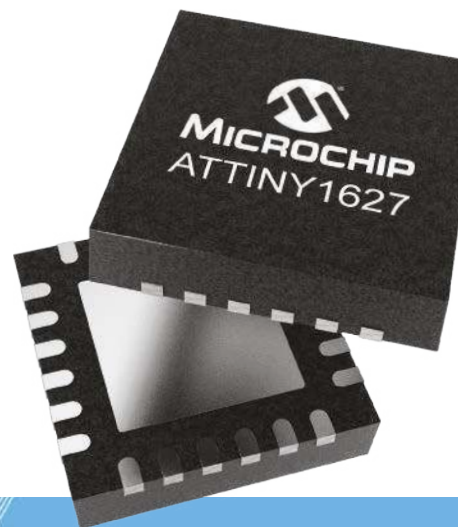
Szybkie i niezawodne interfejsy czujników do trudnych warunków pracy

Rodzina mikrokontrolerów ATtiny1627 zawiera 12-bitowy różnicowy przetwornik ADC ze wzmacniaczem o programowalnym wzmocnieniu (PGA), co umożliwia pomiar sygnałów o małej amplitudzie, wydobywanie sygnału użytecznego z zaszumionego tła i szybkie przetwarzanie informacji w aplikacjach działających w czasie rzeczywistym. Seria ATtiny1627 jest kompatybilna z rodzinami tinyAVR® 1 i 0, a migracja między nimi jest bardzo prosta.

Układy ATtiny1627 doskonale pasują do węzłów czujników, a także do małych i wydajnych aplikacji sterujących. Dzięki maksymalnie dwóm USART-om można łatwo skonfigurować komunikację z różnymi interfejsami. Zastosowania w węzłach czujnikowych mogą obejmować detektory ruchu z sensorami pasywnej podczerwieni (PIR), pomiary temperatury za pomocą termopar, małych rezystancji i prądów w tym boczników i enkoderów magnetycznych. Drugi USART należący do rodziny ATtiny1627 umożliwia komunikację z kilkoma interfejsami w aplikacji.

Kluczowe właściwości

- Szybka i dokładna akwizycja sygnału za pomocą 12-bitowego różnicowego przetwornika analogowo-cyfrowego (ADC)
- Pomiar sygnałów o małej amplitudzie za pomocą wzmacniacza PGA
- Lepsze tłumienie zakłóceń dzięki sprzętowemu nadpróbkowaniu do 1024 próbek z akumulacją wyniku



microchip.com/attiny1627



eprasa.pl 1d618dc1b2

Nazwa i logo Microchip, logo Microchip i tinyAVR są zastrzeżonymi znakami towarowymi firmy Microchip Technology Incorporated w USA i innych krajach. Wszystkie inne znaki towarowe są własnością ich zarejestrowanych właścicieli. © 2021 Microchip Technology Inc. Wszelkie prawa zastrzeżone. D530010230B. MEC2371-POL-05-21