Wydanie elektroniczne dostępne na www.ulubionykiosk.pl

15 stron Crash Course STM32C0

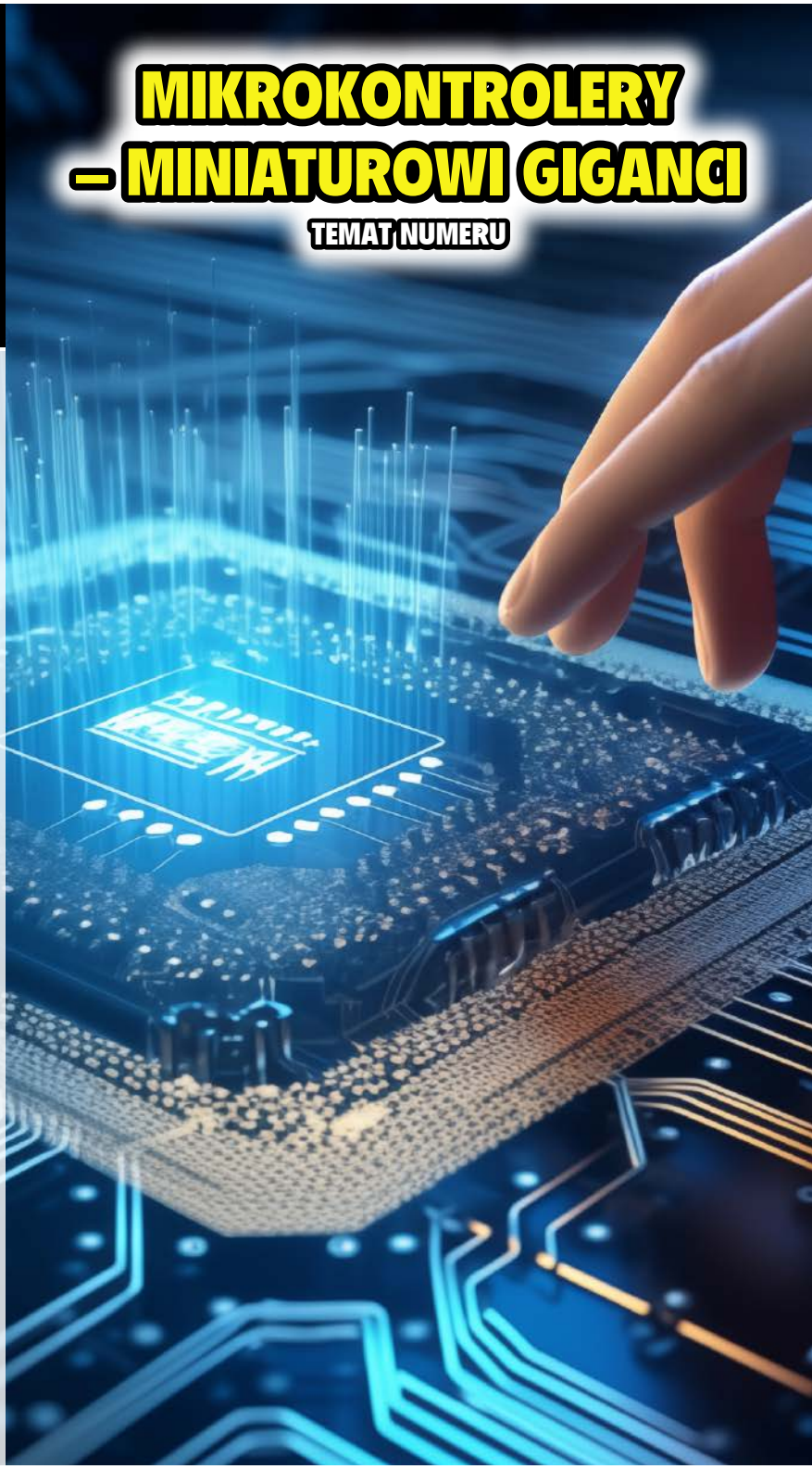
ELEKTRONIKA PRAKTYCZNA

EP.com.pl

● Międzynarodowy magazyn elektroników konstruktorów ● marzec ● 3/2024 ●

Tylko Prenumeratorzy

- mają dostęp do artykułów przed ich publikacją w EP na www.ep.com.pl – **EP W TOKU**
- mają dostęp do materiałów dodatkowych, takich jak pliki źródłowe projektów na naszym serwerze **FTP** www.ulubionykiosk.pl/media



inspirujące, użyteczne projekty
• powerTracker. Rejestrator parametrów elektrycznych • Sterownik silników do AVTDuino UNO R4 • Zasilacz PoE 5/12 V 30 W • Dwuportowy mikrohub USB-C do systemów wbudowanych • Prosty higrometr RGB

podzespoły, sprzęt, aplikacje
• Multimetr stołowy Owon XDM3051. 5½ cyfry w przystępnej cenie – da się? • Narzędzia sprzętowe dla programistów systemów wbudowanych • Przekrojowa oferta mikrokontrolerów w portfolio firmy Micros • Mikrokontrolery z rodziny AVR DB oraz AVR DD – zaprojektowane do pracy w obwodach o zróżnicowanym poziomie napięć • Zestawy ewaluacyjne i inteligentne wyświetlacze w ofercie ELTY – wsparcie prototypowania oraz produkcji elektroniki • Programowanie pamięci mikrokontrolerów • Ewolucja peryferiów analogowych w mikrokontrolerach 8-bitowych • Moduły SoM od Laird Connectivity – gdy mikrokontroler to (zdecydowanie) za mało • Nowy enkoder przyrostowy FAULHABER IEP3 • Nowoczesne materiały termoprzewodzące • Chłodzenie w urządzeniach elektronicznych • Przegląd rynku mikrokontrolerów 2024

tutoriale
• Internet Rzeczy w pomiarach środowiskowych. Moduł czujników środowiska SEN5x firmy Sensirion • Chłodzenie w urządzeniach elektronicznych – przewodnik projektanta • Zasilanie energooszczędnych mikrokontrolerów

kursy
• Crash Course STM32C0 – programowanie mikrokontrolerów ARM w rejestrach • Kurs FPGA Lattice. Odtwarzacz melodii

ISSN 1230-3526 Indeks 357677
9 771230 352245 03
18,90 zł (w tym 8% VAT) • PRICE: 8 EUR

CHŁODZENIE W URZĄDZENIACH ELEKTRONICZNYCH

eprasa.pl 65e3eb482b

wiosenne

–50%

na wybrane roczne prenumeraty drukowane

101,40 zł
12 wydań w roku

113,40 zł
12 wydań w roku

89,40 zł
12 wydań w roku

Zaprenumeruj wybrane czasopismo z rabatem aż 50%!

Promocja wiosenna dotyczy rocznych prenumerat drukowanych czasopism:
• Elektronika dla Wszystkich (12 wydań w roku) • Elektronika Praktyczna (12 wydań w roku) • Młody Technik (12 wydań w roku)

Zamów prenumeratę na www.UlubionyKiosk.pl/prenumerata lub poprzez dokonanie przelewu na konto AVT-Korporacja sp. z o.o., ul. Leszczynowa 11, 03-197 Warszawa, ING BANK ŚLĄSKI 18 1050 1012 1000 0024 3173 1013 (w tytule wpłaty podaj nazwę czasopisma).

Masz opłaconą bieżącą prenumeratę? Już teraz przedłuż ją z rabatem 50%.

Promocja trwa do 31.05.2024 i nie łączy się z innymi promocjami Wydawnictwa AVT.
Koszt wysyłki (list standardowy nierejestrowany) pod wskazany adres w Polsce ponosi wydawnictwo.

eprasa.pl 65e3eb482b

Bitwa o peryferia – oraz inne trendy na rynku mikrokontrolerów

Szanowni Czytelnicy,

przyszło nam żyć w ciekawych czasach! Na szczęście oryginalne, negatywne znaczenie tego starego chińskiego przysłowia można w niektórych przypadkach zmienić i dostrzec w nim wiele aspektów jak najbardziej pozytywnych. W świecie mikrokontrolerów istotnie zachodzi sporo interesujących zmian, a – jak zwykle bywa w takich sytuacjach – rosnąca konkurencja oraz wynikająca z niej walka o klienta prowadzą do istotnej dywersyfikacji rynku dostępnych komponentów. Odbywa się to z niewątpliwą korzyścią dla nas, konstruktorów – wszak im większy wybór, tym łatwiej możemy dobrać procesor optymalny do projektowanego urządzenia.

Co ciekawe, wiele zapowiadanych przed laty zmian w globalnym obrazie rynku MCU albo w ogóle nie wprowadzono w życie, albo poszły one – niczym przysłowiowy kot – własnymi ścieżkami. Gdy kilkanaście lat temu mikrokontrolery ARM szturmem wdzierały się na rynek, wielu wieszczyło nieuchronną śmierć poczciwych 8-bitowców. Tymczasem popularne rodziny AVR czy PIC nie tylko nie wymarły, ale zyskały silne wsparcie w postaci unowocześnionych układów, które zaskakują właśnie na polu tytułowych peryferiów. Okazuje się, że nowe serie procesorów 8-bitowych stopniowo przejmują niektóre zalety znacznie potężniejszych układów. „Małe" mikrokontrolery otrzymują m.in. obwody nadzorujące poprawność pracy oscylatora, taktowanego zewnętrznym rezonatorem kwarcowym – rozwiązania tego typu są doskonale znane chociażby z układów STM32, podobnie jak systemy synchronizacji i przesyłania sygnałów sterujących bądź wyzwalających pomiędzy dwoma blokami sprzętowymi, bez udziału rdzenia. Do 8-bitowców trafiają także wzmacniacze operacyjne, przetworniki DAC czy też unowocześnione, 12-bitowe konwertery ADC. Z drugiej strony, niektóre rozwiązania zastosowane w najnowocześniejszych mikrokontrolerach 8-bitowych – takie jak wbudowane translatory poziomów logicznych czy bloki „gumowej logiki", dające namiastkę prościutkiej macierzy FPGA (pracującej niezależnie od rdzenia i wspierającej go w wykonywaniu niskopoziomowej obróbki sygnałów logicznych) – przewyższają wielu 32-bitowych konkurentów.

W diametralnie innej sytuacji znaleźli się producenci, którzy niejako przyzwyczaili się już do wdrażania kolejnych rodzin procesorów 32-bitowych na bazie architektury ARM. Bogactwo wbudowanych bloków komunikacyjnych, obliczeniowych czy też sterujących potrafi przyprawić o zawrót głowy – dokumentacja najnowszych procesorów ma nierzadko ponad trzy i pół tysiąca stron (a mówimy tutaj tylko o pojedynczym dokumencie, określanym jako *reference manual* – do tego dochodzą jeszcze noty katalogowe, dziesiątki not aplikacyjnych i dokumentów wspomagających pracę z tymi „kombajnami" współczesnej techniki cyfrowej)…

Co zatem mogą zrobić półprzewodnikowi potentaci, by nie oddać 8-bitowego kawałka rynkowego tortu swoim konkurentom? Bitwa na peryferia w rozumieniu opisanego powyżej dodawania kolejnych funkcjonalności nie miałaby w ich przypadku większego sensu, powodowałaby bowiem jeszcze silniejsze oddalanie się od celu. Istotnie, nie tędy droga. Firma ST Microelectronics podeszła do problemu od zupełnie innej strony, niejako pod prąd. Wprowadzona przez ST rodzina procesorów STM32C0 bazuje wprawdzie na 32-bitowym rdzeniu ARM Cortex-M0+ (podobnie jak tysiące innych modeli MCU dostępnych na rynku), ale jej zadaniem jest… zmiażdżenie konkurencji pod względem finansowym. Producent chwali się, że ceny nowych mikrokontrolerów startują już od 24 centów(!), a w zestawie z tymi elementami użytkownicy otrzymują nie tylko sprawdzoną, wydajną i całkiem energooszczędną architekturę, ale w dodatku mogą łatwo przesiąść się na nie z dużo wydajniejszych modeli. STM32C0 mają bowiem pełne wsparcie dla programistów w postaci znanego i lubianego (choć nie przez wszystkich!) ekosystemu STM32Cube.

A co z programistami, którzy z jakichś względów nie chcieli lub nie potrzebowali dotąd przesiadać się na 32-bitowce? Zdajemy sobie sprawę z tego, że metody programowania procesorów w rejestrach oraz z użyciem wysokopoziomowych bibliotek HAL są od siebie biegunowo odległe, stąd zmiana sposobu pracy wymaga całkowitego przestawienia się na inny model tworzenia kodu.

Dlatego właśnie w tym numerze „Elektroniki Praktycznej" przygotowaliśmy zapowiadany w poprzednim wydaniu materiał specjalny – Crash Course STM32C0, czyli… jednoodcinkowy, ekspresowy kurs niskopoziomowego programowania, nie tylko bez użycia HAL, ale nawet z pominięciem bibliotek LL (Low Level). Wielu programistom embedded pisanie kodu na procesory ARM z użyciem samych tylko gołych rejestrów wydaje się niezwykle trudne, a niektórym wręcz niemożliwe. Dlatego podjęliśmy wyzwanie – w ramach zaledwie 15 stron artykułu omówiliśmy zarówno proces konfiguracji odpowiedniego środowiska programistycznego, jak i podstawy tworzenia kodu na niemal najniższym (zaraz po asemblerze) poziomie abstrakcji. Mało tego – wspólnie uruchomimy nawet bardziej rozbudowane bloki peryferyjne, nawiązując komunikację z zewnętrznym czujnikiem (i to na dwa sposoby).

W numerze marcowym przygotowaliśmy także szereg innych atrakcji. W dziale „Elektronika w Praktyce" znalazł się poradnik dotyczący metod, komponentów i materiałów umożliwiających efektywne chłodzenie elektroniki. Publikujemy ponadto projekty: powerTracker (niewielki rejestrator napięcia, prądu, mocy i ładunku) i sterownik silników w formacie Arduino Shield. Tradycyjnie nie zabrakło również kilku ciekawych i użytecznych miniprojektów czy też kolejnych odcinków kursu programowania FPGA oraz cyklu poświęconego czujnikom środowiskowym w aplikacjach IoT.

Zapraszam do lektury!

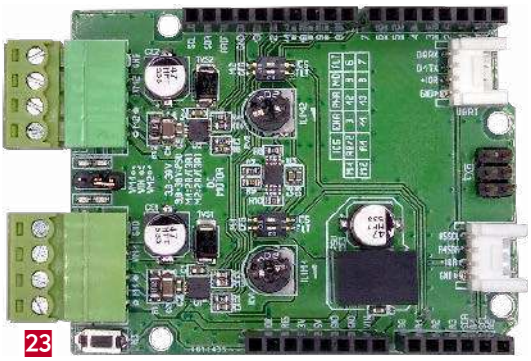
Przemysław Musz

eprasa.pl 65e3eb482b

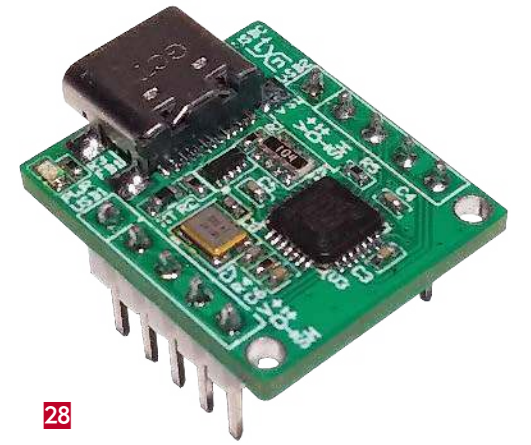
ELEKTRONIKA PRAKTYCZNA

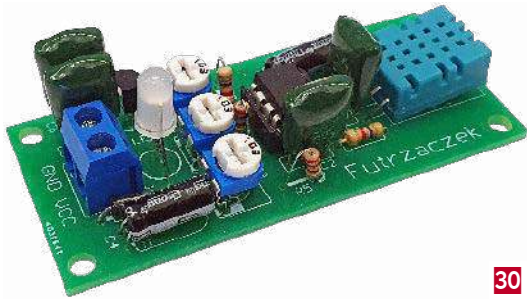
Nr 3 (374) • Marzec 2024

14


23

26


28


30

Nie przeocz
Nowe podzespoły ... 6
Konkurs ... 11
Dodaj do obserwowanych ... 12
Koktajl niusów ... 112

Projekty
powerTracker. Rejestrator parametrów elektrycznych ... 14
Sterownik silników do AVTDuino UNO R4 ... 23

Miniprojekty
Zasilacz PoE 5/12 V 30 W ... 26
Dwuportowy mikrohub USB-C do systemów wbudowanych ... 28
Prosty higrometr RGB ... 30

Prezentacje
Przekrojowa oferta mikrokontrolerów w portfolio firmy Micros ... 33
Mikrokontrolery z rodziny AVR DB oraz AVR DD
– zaprojektowane do pracy w obwodach o zróżnicowanym poziomie napięć ... 36
Zestawy ewaluacyjne i inteligentne wyświetlacze w ofercie ELTY
– wsparcie prototypowania oraz produkcji elektroniki ... 48
Programowanie pamięci mikrokontrolerów ... 50
Ewolucja peryferiów analogowych w mikrokontrolerach 8-bitowych ... 52
Moduły SoM od Laird Connectivity
– gdy mikrokontroler to (zdecydowanie) za mało ... 54
Nowy enkoder przyrostowy FAULHABER IEP3 ... 63
Nowoczesne materiały termoprzewodzące ... 64
Chłodzenie w urządzeniach elektronicznych ... 80

Temat numeru: Mikrokontrolery – miniaturowi giganci
Narzędzia sprzętowe dla programistów systemów wbudowanych ... 39
Przegląd rynku mikrokontrolerów 2024 ... 42

Sprzęt
Multimetr stołowy Owon XDM3051. 5½ cyfry w przystępnej cenie – da się? ... 56

Notatnik konstruktora
Zasilanie energooszczędnych mikrokontrolerów ... 60

Elektronika w praktyce
Chłodzenie w urządzeniach elektronicznych – przewodnik projektanta ... 67

Moduły w aplikacjach
Internet Rzeczy w pomiarach środowiskowych (3).
Moduł czujników środowiska SEN5x firmy Sensirion ... 82

Kursy
Crash Course STM32C0
– programowanie mikrokontrolerów ARM w rejestrach ... 89
Kurs FPGA Lattice (17). Odtwarzacz melodii ... 104

Prenumerata ... 2
Od wydawcy ... 5
Hity następnego numeru ... 115

Dodatkowe materiały do artykułów znajdziesz na https://ulubionykiosk.pl/media
Szczegóły na stronie 32

eprasa.pl 65e3eb482b

MIKROKONTROLERY
HITY SPRZEDAŻY

Micros sp.j. W.Kędra i J.Lic
ul. E.Godlewskiego 38, 30-198 Kraków
tel.: +48 12 636 95 66, e-mail: bok@micros.com.pl
www.micros.com.pl

Megawin Technology oferuje wiele mikrokontrolerów 8-bitowych z szybkim rdzeniem 8051 i bogatym zestawem układów peryferyjnych oraz 32-bitowe układy z rdzeniem Cortex-M0 i Cortex-M3 kompatybilne z rozwiązaniami innych producentów. W przypadku mikrokontrolerów opartych na rdzeniu Cortex M0 producent oferuje trzy wersje tych urządzeń: Base Line - wersja podstawowa układów z tej rodziny, USB Line - wzbogacona o interfejs USB 2.0 działający z szybkością 12Mb oraz Value Line - układy dostępne w małych obudowach TSSOP20/QFN32. Wśród oferty Megawin dostępne są układy kompatybilne pod względem obudowy i schematu wyprowadzeń (pin to pin) z popularnymi układami z serii STM32F. Producent w swoim asortymencie posiada również narzędzia deweloperskie takie jak płytki ewaluacyjne czy debuggery, oraz zapewnia wsparcie programowe. **W sklepie Micros czeka na Ciebie bogaty wybór mikrokontrolerów Megawin - 30 opcji do wyboru.**

Symbol	MG82F6B001AS8	MG32F02A064AD64	MG32F103CBT6	MG32F103RBT6	MG32F104RCT6
Wbudowana pamięć	8kB-FLASH, 1kB-RAM, 512B-EEPROM	64kB-FLASH, 8kB-SRAM,	128kB-FLASH, 28kB-SRAM,	128kB-FLASH, 28kB-SRAM,	256kB-FLASH, 36kB-SRAM,
Częstotliwość	16MHz	48MHz	72MHz	72MHz	96MHz
I/O	6	59	37	51	51
Interfejsy	UART, SPI, TWI	UART, SPI, TWI,	UART, SPI, TWI, USB	UART, SPI, TWI, USB	UART, SPI, TWI, USB
Obudowa	SOP08	LQFP64	LQFP48	LQFP64	LQFP64

ST

Rodzina mikrokontrolerów STM32, produkowanych przez firmę STMicroelectronics obejmuje szeroki zakres kontrolerów, które różnią się między sobą pod względem wydajności, funkcji, pamięci, a także innych parametrów. Jednakże, wszystkie one korzystają z zaawansowanej architektury rdzeni ARM Cortex-M, co zapewnia wysoką wydajność, efektywność energetyczną i szerokie wsparcie dla narzędzi programistycznych. Dzięki temu są one popularnym wyborem wśród programistów i inżynierów, którzy projektują systemy wbudowane dla różnych zastosowań, od prostych urządzeń peryferyjnych po zaawansowane systemy kontrolne. W ofercie Microsa znajdziemy produkty z serii: STM32F0, STM32F1, STM32F2, STM32F3, STM32F4, STM32F7, STM32L1, STM32L2, STM32L4(+). Dostępne są też inne rodziny układów, takie jak: ST6, ST7, ST10 i STM8. **W sklepie Micros czeka na Ciebie bogata oferta mikrokontrolerów ST - aż 303 pozycje.**

Symbol	STM32F030F4P6	STM32F103C8T6	STM32F103CBT6	STM32F407VGT6	STM32F103RET6
Wbudowana pamięć	16kB-FLASH, 4kB RAM	64kB-FLASH, 20kB-SRAM	128kB-FLASH, 20kB-SRAM	1MB-FLASH, 192kB-SRAM	512kB-FLASH, 64kB RAM
Częstotliwość	48MHz	72MHz	72MHz	168MHz	72MHz
I/O	15	37	37	82	51
Interfejsy	UART, SPI, TWI	USART, SPI, TWI, CAN, USB	USART, SPI, TWI, CAN, USB	UART/USART, SPI, TWI, CAN, ETHERNET, USB	UART/USART, SPI, TWI, CAN, USB
Obudowa	TSSOP20	LQFP48	LQFP48	LQFP100	LQFP64

ATMEL

ATMEL (aktualnie MICROCHIP) produkuje trzy główne grupy mikrokontrolerów: MCS-51, AVR i ARM. Mikrokontrolery z rodziny AVR to nowoczesne procesory z optymalnie zaprojektowaną strukturą wejść/wyjść, architekturą RISC, wykonujące wszystkie instrukcje w jednym cyklu zegara. Przemyślana konstrukcja pozwala na istotne ograniczenie elementów zewnętrznych. Wbudowane funkcje wewnętrzne (w zależności od modelu): zegar, timery, UART, SPI, TWI, CAN, rezystory pull-up, PWM, przetworniki A/D, watchdog i wiele innych. **W sklepie Micros znajdziesz szeroki wybór mikrokontrolerów Atmel - aż 489 różne opcje.**

Symbol	ATMEGA128A-AU	ATMEGA2561-16AU	ATMEGA8A-PU	ATMEGA16A-AU	ATMEGA168PA-AU
Wbudowana pamięć	128kB-FLASH, 4kB RAM, 4kB-EEPROM	256kB-FLASH, 8kB RAM, 4kB-EEPROM	8kB-FLASH, 1kB RAM, 512B-EEPROM	16kB-FLASH, 1kB RAM, 512B-EEPROM	16kB-FLASH, 1kB RAM, 512B-EEPROM
Częstotliwość	16MHz	16MHz	16MHz	16MHz	20MHz
I/O	53	54	23	32	23
Interfejsy	USART, SPI, TWI	USART, SPI, TWI	USART, SPI, TWI	USART, SPI, TWI	USART, SPI, TWI
Obudowa	TQFP64	TQFP64	PDIP28W	TQFP44	TQFP32

MICROCHIP

Firma MICROCHIP znana jest z bardzo interesujących mikrokontrolerów PIC, których dużą zaletą jest wzajemna kompatybilność oraz długotrwała dostępność. Pośród dostępnych modeli znaleźć można zarówno wersje 8-bitowe, 16-bitowe, jak i 32-bitowe. W ofercie Microsa znaleźć można produkty z rodzin: PIC10F, PIC12F, PIC16F, PIC18F, PIC24F, PIC32M, dsPIC30F i dsPIC33F, z czego te dwie ostatnie obejmują układy wyspecjalizowane w cyfrowym przetwarzaniu sygnałów. **W sklepie Micros znajdziesz obfitą gamę mikrokontrolerów Microchip, obejmującą aż 404 różne pozycje do wyboru.**

Symbol	PIC18F67K22-I/PT	PIC16F628A-I/P	PIC12F675-I/SN	PIC12F1822-I/SN	PIC16F1823-I/SL
Wbudowana pamięć	128kB-FLASH, 4kB RAM, 1kB-EEPROM	3,5kB-FLASH, 128B-EEPROM, 224B-SRAM	1,75kB-FLASH, 64B RAM, 128B-EEPROM	3,5kB-FLASH, 256B-EEPROM, 128B-SRAM	3,5kB-FLASH, 128B RAM, 256B-EEPROM
Częstotliwość	64MHz	20MHz	20MHz	32MHz	32MHz
I/O	53	16	6	5	11
Interfejsy	USART, SPI, TWI	USART	-	UART, SPI, TWI	USART, SPI, TWI
Obudowa	TQFP64	PDIP18	SOP08	SOP08	SOP14

NXP

Dużą popularnością wśród rodzin ARM-ów, cieszy się seria LPC firmy NXP. W ramach tych produktów oferta firmy Micros koncentruje się na podgrupach: LPC1100, LPC1200, LPC1300, LPC1500, LPC1800, LPC2000, LPC3000, LPC4000, LPC4300. **W sklepie Micros znajdziesz obszerny wybór mikrokontrolerów NXP - aż 61 różnych modeli.**

Symbol	LPC2132FBD64/01	LPC2148FBD64	LPC1765FBD100	LPC1115FBD48/303	LPC1768FBD100
Wbudowana pamięć	64kB-FLASH, 16kB RAM	512kB-FLASH, 40kB RAM	256kB-FLASH, 64kB RAM	64kB-FLASH, 8kB RAM	512kB-FLASH, 64kB RAM
Częstotliwość	60MHz	60MHz	100MHz	50MHz	100MHz
I/O	47	45	70	42	70
Interfejsy	UART, SPI, TWI	UART, SPI, TWI, USB	UART, SPI, TWI, CAN, USB	UART, SPI, TWI	UART, SPI, TWI, CAN, USB, ETHERNET
Obudowa	LQFP64	LQFP64	LQFP100	LQFP48	LQFP100

GEEHY

Geehy Semiconductor produkuje alternatywne wersje procesorów STM32F, bazując na własnym „know-how" i procesie technologicznym. Ich układy APM to odpowiedniki na zasadzie FFF (fit-form-function), czyli alternatywny projekt o takiej samej funkcjonalności i rozkładzie pinów w obudowach. Geehy jest oficjalnym partnerem (licencjonobiorcą) firmy ARM, co oznacza, że zgodność binarna wytwarzanych układów jest duża i nie ma z nimi problemów kompatybilności oprogramowania. **W sklepie Micros znajdziesz szeroki wybór mikrokontrolerów Geehy - 36 różnych opcji.**

Symbol	APM32F103RET6	APM32E103VET6	APM32F003F6P6	APM32F407VGT6	APM32F103RBT6
Wbudowana pamięć	512kB-FLASH, 128kB RAM	512kB-FLASH, 128kB RAM	32kB-FLASH, 4kB RAM	1MB-FLASH, 192kB RAM	128kB-FLASH, 20kB-RAM
Częstotliwość	96MHz	120MHz	48MHz	168MHz	128MHz
I/O	51	80	16	82	51
Interfejsy	UART/USART, SPI, TWI, CAN, USB	UART/USART, SPI, TWI, CAN, USB	USART, SPI, TWI	UART/USART, SPI, TWI, CAN, USB, ETHERNET	USART, SPI, TWI, CAN, USB
Obudowa	LQFP64	LQFP100	TSSOP20	LQFP100	LQFP64

eprasa.pl 65e3eb482b

nowe podzespoły

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

Mikrokontrolery o mocy obliczeniowej >3000 CoreMark z rdzeniem Arm Cortex-M85

Renesas Electronics prezentuje pierwszą serię mikrokontrolerów – o bardzo małych opóźnieniach i mocy obliczeniowej przekraczającej 3000 punktów w teście CoreMark – przeznaczonych do zastosowań w deterministycznych systemach czasu rzeczywistego. Mikrokontrolery RA8, wyposażone w jednostkę obliczeniową ARM Cortex-M85, osiągającą wydajność 6,39 CoreMark/MHz w testach EEMBC CoreMark, mogą być stosowane w aplikacjach, które wcześniej wymagały tradycyjnych mikroprocesorów. Umożliwiają łatwe przenoszenie projektów z aplikacji zbudowanych w oparciu o starsze mikrokontrolery z rodziny RA. Ponadto korzystają z nowej technologii ARM Helium z rozszerzeniem MVE (M-Profile Vector Extension), zapewniającej 4-krotny wzrost szybkości cyfrowego przetwarzania sygnałów i algorytmów uczenia maszynowego, w stosunku do mikrokontrolerów opartych na jednostce ARM Cortex-M7. W niektórych zastosowaniach, na przykład związanych z konserwacją predykcyjną, pozwala to na wyeliminowanie dodatkowego procesora DSP.

Poza dużą mocą obliczeniową, mikrokontrolery serii RA8 oferują zaawansowane mechanizmy zabezpieczeń, w tym technologię ARM TrustZone, obsługę protokołu Renesas Security IP (RSIP-E51A), Root-of-Trust, interfejs Octal SPI z deszyfrowaniem w locie (DOTF), bezpieczne debugowanie z uwierzytelnieniem i funkcje ochrony przed manipulacją. Architektura Armv8.1-M wprowadza rozszerzenie zabezpieczeń Pointer Authentication and Branch Target Identification (PACBTI), zapobiegające atakom na oprogramowanie, których celem byłoby naruszenie bezpieczeństwa pamięci. Otwiera to drogę do zastosowań w systemach z certyfikatami PSA Certified Level 2 + Secure Element (SE), NIST CAVP i FIPS 140-3.

www.renesas.com

Moduł komunikacyjny z obsługą Bluetooth LE i UWB

Laird Connectivity wprowadza do oferty moduł komunikacyjny Sera NX040 z obsługą standardów Bluetooth LE i UWB (ultra-wideband), oparty na transceiverach Trimension SR040 produkcji NXP i nRF52833 produkcji Nordic Semiconductor. Przy jego opracowywaniu

skupiono się na zapewnieniu jak największego stopnia integracji, który pozwoli zminimalizować liczbę komponentów współpracujących, takich jak układy zegarowe czy filtry. Moduł jest polecany do zastosowań przede wszystkim w systemach lokalizacji i pozycjonowania. Występuje w wersjach z anteną wewnętrzną i złączem do anteny zewnętrznej. Pracuje pod kontrolą systemu operacyjnego Zephyr. Producent dostarcza do niego zestaw przykładowych aplikacji w języku Python. Użytkownicy mają cztery możliwości programowania modułu: z wykorzystaniem języka skryptowego, zestawu komend AT, platformy Nordic nRFConnect SDK lub z poziomu aplikacji mobilnej.

www.lairdconnect.com

Kondensatory elektrolityczne o podwyższonej niezawodności i małej rezystancji ESR

Firma TTi dodała do oferty dwie serie kondensatorów elektrolitycznych produkcji Kemet, charakteryzujących się podwyższoną niezawodnością i bardzo małą rezystancją ESR, wynoszącą od 5 mΩ. Są to kondensatory z elektrolitem stałym i katodą z przewodzącego polimeru, zachowujące pojemność znamionową również w zakresie dużych częstotliwości. Wykazują małą rezystancję ESR – typową dla kondensatorów ceramicznych MLCC, dużą pojemność (porównywalną z kondensatorami elektrolitycznymi) oraz duży współczynnik CV – charakterystyczny dla kondensatorów tantalowych. Mogą być stosowane w filtrach i układach polaryzacji wzmacniaczy w.cz. oraz jako elementy podtrzymujące zasilanie.

W ramach serii HRA T540/T541 produkowane są kondensatory o pojemności od 4,7 do 1500 μF (120 Hz, 25°C) i napięciu znamionowym od 2,5 do 63 VDC. Charakteryzują się one prądem upływu ≤0,1 CV (SμA), tangensem kąta strat ≤10% i zakresem dopuszczalnej temperatury pracy od –55 do +125°C. Występują w trzech klasach niezawodnościowych o współczynniku awaryjności B (0,1% @ 1000 h), C (0,01% @ 1000 h) i D (0,001% @ 1000 h).

www.ttieurope.com

eprasa.pl 65e3eb482b

Zintegrowane 600-woltowe stopnie mocy do układów sterowania silnikami BLDC

Toshiba Electronics Europe poszerza ofertę zintegrowanych stopni mocy (IPD) do układów sterowania silnikami bezszczotkowymi prądu stałego. Do jej oferty wchodzą dwa warianty 600-woltowe, TPD4163K i TPD4164K, zamykane w obudowach HDIP30 o wymiarach 32,8×13,5×3,5 mm. Zajmują one mniejszą o 20% powierzchnię montażową od wcześniejszych odpowiedników TPD4123x, TPD4144x i TPD4135x, dostępnych w obudowach DIP26. Są polecane do zastosowań w pompach, wentylatorach, klimatyzatorach itp. Podobnie jak wcześniejsze wersje, zawierają wbudowane sterowniki bramek.

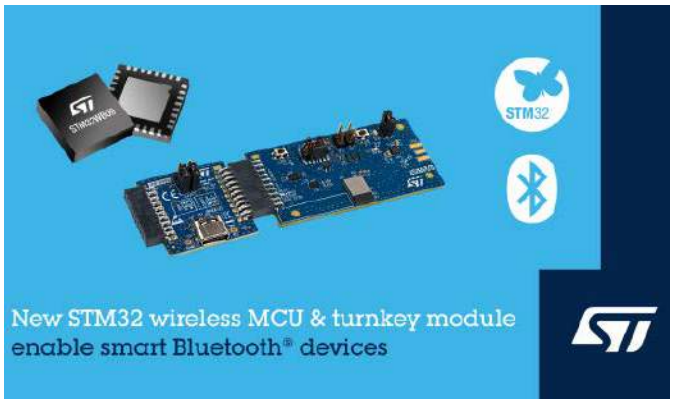
TPD4163K i TPD4164K różnią się maksymalnym prądem wyjściowym, wynoszącym 1 A i 2 A, napięciem nasycenia wewnętrznych tranzystorów IGBT (odpowiednio 2,6 V i 3,0 V) oraz napięciem przewodzenia wewnętrznych diod (odpowiednio 2,6 V i 3,0 V). Maksymalne napięcie zasilania sekcji wysokonapięciowej (VBB) obu układów wynosi 600 V, co zwiększa marginesy projektowe i może być szczególnie ważne w aplikacjach o niestabilnym napięciu sieciowym.

www.toshiba.semicon-storage.com

Kable USB-C o stopniu ochrony IP67 do montażu panelowego

Do oferty firmy Samtec wkracza seria wodoszczelnych kabli USB-C do montażu panelowego, produkowanych w izolacji z czarnego tworzywa termoplastycznego z silikonową uszczelką. Są one przystosowane do mocowania za pomocą sześciokątnej nakrętki, w wycięciach o średnicy 16 mm. W ramach serii BPCU dostępne są warianty o długości 0,25 m, 0,5 m, 0,75 m i 1,0 m, wyposażone opcjonalnie w pozłacane styki. Zapewniają one stopień ochrony IP67 po zainstalowaniu w gnieździe. Spełniają wymogi standardu USB 3.2 Gen 2x1 (10 Gbps) oraz Power Delivery w zakresie mocy do 100 W (5 A @ 20 V).

www.eltronika.pl



Nowe kontrolery „bezprzewodowe" Bluetooth Low Energy 5.3

STMicroelectronics wprowadza na rynek dwa nowe układy, umożliwiające realizację nowej generacji inteligentnych urządzeń do łączności bezprzewodowej krótkiego zasięgu. Najnowsza specyfikacja Bluetooth daje projektantom duże możliwości w zakresie tworzenia takich aplikacji, jak bezprzewodowe sygnalizatory i urządzenia zdolne do obliczania swojej lokalizacji w pomieszczeniach – z dokładnością do centymetra. Nie brakuje oczywiście wszystkich funkcjonalności standardu Bluetooth znanych z wcześniejszych jego edycji.

Nowy mikrokontroler „bezprzewodowy" STM32WB09 łączy w jednym chipie dużą moc obliczeniową z obwodami radiowymi Bluetooth. Zawiera najnowsze oprogramowanie Bluetooth 5.3, a jednocześnie pozwala projektantom korzystać z obszernego środowiska projektowego do mikrokontrolerów STM32, obejmującego m.in. zestaw narzędzi deweloperskich i przykładowe kody, przyspieszające tworzenie aplikacji. STM32WB09 jest dostępny m.in. w miniaturowej obudowie chip-scale o powierzchni 2,99×2,83 mm. Jego ceny hurtowe zaczynają się od 1,36 USD (wersja w obudowie QFN32).

Mniej doświadczonym użytkownikom, niemającym dostępu do rozbudowanych zasobów inżynieryjnych, STMicroelectronics oferuje moduł bezprzewodowy STM32WB1MMC z certyfikowanym mikrokontrolerem, wstępnie zintegrowany z zewnętrznymi komponentami do realizacji systemu radiowego, a także oprogramowaniem Bluetooth. Pozwala on ominąć wiele wyzwań technologicznych, napotykanych podczas projektowania z użyciem układów RF. Moduł pomaga zmniejszyć ryzyko projektowe podczas tworzenia zaawansowanych urządzeń bezprzewodowych, wymagając od użytkownika

REKLAMA

HAMMOND®

1556 - Obudowy IP54 FR ABS

Dowiedz się więcej:
hammondmfg.com/1556

eusales@hammondmfg.com • + 44 1256 812812

eprasa.pl 65e3eb482b

jedynie podstawowych umiejętności z zakresu techniki w.cz. Dzięki temu sam konstruktor może skupić się na opracowywaniu własnego oprogramowania sprzętowego. Ceny hurtowe STM32WB1MMC zaczynają się od 5,32 USD przy zamówieniach 10 tys. sztuk.

Aby jeszcze bardziej ułatwić korzystanie z opisywanego modułu Bluetooth, firma STMicroelectronics przygotowała płytkę ewaluacyjną B-WB1M-WPAN1, upraszczającą rozpoczęcie prac projektowych. Na płytce znajdują się m.in. czujniki ruchu, temperatury i ciśnienia barometrycznego oraz złącze do zewnętrznej anteny. Cena B-WB1M-WPAN1 wynosi 30 USD.

Ważniejsze cechy STM32WB09:

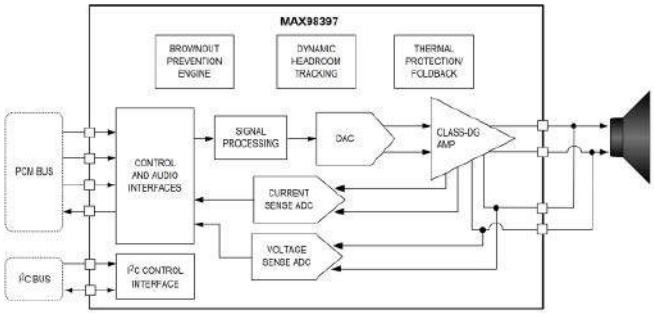
- jednostka obliczeniowa: ARM Cortex-M0+,
- pamięć: 64 kB RAM + 512 kB Flash,
- radio: BLE 5.3 (long range, advertising ext., 2 Mbps, AoA/AoD, izochroniczne kanały),
- czułość odbiornika: –104 dBm,
- moc nadajnika: +8 dBm,
- bezpieczeństwo: AES, TRNG, RSA, secure boot,
- interfejsy: USART, LPUART, I²S, SPI, I²C,
- funkcje analogowe: 12-bitowy przetwornik A/C (1 Mbps),
- 3 timery 16-bitowe, DMA, RTC, IWDG, SysTick, interfejs IR,
- zakres temperatury pracy: od –40 do +105°C.

www.st.com

Najmniejszy resetowalny bezpiecznik termiczny SMD o wymiarach 4,7×2,8×0,94 mm

Bourns wprowadza na rynek najmniejszy z dotychczasowej oferty, resetowalny bezpiecznik termiczny, zamykany w obudowie SMD o wymiarach 4,7×2,8×0,94 mm. Model SW charakteryzuje się temperaturą progową +70°C ±5°C i temperaturą resetu +49°C ±9°C. Może znaleźć zastosowanie w ochronie nadprądowej m.in. systemów oświetleniowych LED, silników czy też cewek w systemach ładowania bezprzewodowego. Ponadto może pełnić funkcję elementu zabezpieczającego przed przegrzaniem. Jego maksymalna rezystancja wewnętrzna wynosi 5 mΩ, prąd znamionowy 3 A @ +60°C, a dopuszczalne napięcie szczytowe: 100 V. Bezpiecznik SW wykazuje dużą odporność na wilgoć i korozję.

www.bourns.com



Energooszczędny wzmacniacz audio klasy DG z funkcją IVSENSE i detektorem napięcia zasilania

MAX98397 to energooszczędny, monofoniczny wzmacniacz audio klasy DG z funkcją IVSENSE i detektorem napięcia zasilającego. Korzysta on z dwóch napięć zasilania: V_{BAT} (3...5,5 V) i P_{VDD} (3...28 V), przełączanych w zależności od poziomu sygnału wejściowego. Dzięki dużej częstotliwości próbkowania (192 kHz) i paśmie o szerokości >50 kHz może znaleźć zastosowanie również w aplikacjach ultradźwiękowych.

MAX98397 zawiera czujniki prądu i napięcia wyjściowego (ISENSE/VSENSE), umożliwiające zaimplementowanie algorytmów ochrony głośnika w układzie host. Modulacja spread-spectrum oraz kontrola szybkości narastania zboczy pozwalają zminimalizować poziom generowanych zaburzeń EMI, a także wyeliminować filtr wyjściowy, niezbędny w przypadku wzmacniaczy klasy D.

W przypadku urządzeń zasilanych bateryjnie wewnętrzny detektor napięcia zasilającego zapobiega możliwości wystąpienia awarii poprzez zmniejszenie udziału układu w całkowitym poborze mocy systemu. Realizuje to poprzez ograniczanie mocy wyjściowej wzmacniacza, gdy napięcie zasilania zmniejsza się poniżej zaprogramowanej wartości progowej.

Układ zawiera interfejs PCM do transmisji danych audio oraz I²C do transmisji danych sterujących. Interfejs PCM obsługuje formaty I²S, left-justified i TDM. Unikalna struktura taktowania eliminuje potrzebę stosowania zewnętrznego zegara systemowego do komunikacji PCM, a to zmniejsza liczbę wyprowadzeń i upraszcza projekt płytki drukowanej. Funkcja zabezpieczenia termicznego automatycznie zmniejsza moc wyjściową, gdy temperatura układu przekroczy próg określony przez użytkownika – co pozwala na ciągłe odtwarzanie muzyki, również w podwyższonych temperaturach pracy.

MAX98397 jest zamykany w obudowie WLP-35 o rozstawie wyprowadzeń 0,4 mm. Może pracować w temperaturze otoczenia od –40 do +85°C.

Pozostałe parametry:

- moc wyjściowa: 51 W (8 Ω, 28 V, THD+N=10%),
- sprawność: do 93% (8 Ω, 24 V),
- THD+N: –85 dB (6 W, 4 Ω, 1 kHz),
- pobór mocy w trybie spoczynkowym: do 25,3 mW,
- wyjściowe napięcie szumu: 21,5 μV rms,
- zakres dynamiczny: 117,8 dB.

www.analog.com

Diody OSLON LED 3. generacji do reflektorów samochodowych

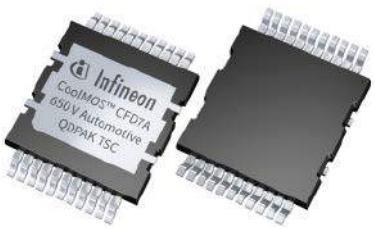
Firma ams OSRAM wprowadza na rynek 3. generację diod LED OSLON Submount PL do reflektorów samochodowych. Dzięki możliwości bezpośredniego montażu na radiatorach pozwalają one obniżyć ogólne koszty systemów oświetleniowych, eliminując potrzebę projektowania płytek drukowanych.

Seria OSLON Submount PL obejmuje diody z kwalifikacją AEC-Q102 o dwóch wartościach powierzchni emisyjnej (2,1 mm² dla wersji 2-chipowej KW2 C2LNL3.TK oraz 3,3 mm² dla wersji 3-chipowej KW3 C3LNL3.TK). Są one produkowane w wariantach o strumieniu wynoszącym odpowiednio: 790...1060 lm i 1185...1525 lm dla I_F=1000 mA. Różnią się też napięciem przewodzenia, wynoszącym odpowiednio: 6,5 V i 9,47 V.

Oba modele przystosowane zostały do pracy w temperaturze otoczenia od –40 do +130°C, z maksymalnym prądem przewodzenia 1500 mA. Zapewniają odporność na wyładowania ESD do 8 kV. Ich jasność jest średnio o 9% wyższa od diod wcześniejszej generacji.

www.ams-osram.com

Superzłączowe tranzystory MOSFET serii CoolMOS CFD7A w nowych obudowach QDPAK



Superzłączowe tranzystory MOSFET serii CoolMOS CFD7A są obecnie produkowane w nowych obudowach QDPAK z chłodzeniem dolnym (BSC) lub górnym (TSC). Zapewniają one parametry termiczne porównywalne z wcześniejszymi wersjami w obudowach TO247, natomiast wykazują lepsze parametry elektryczne, co pozwala

eprasa.pl 65e3eb482b

zwiększyć sprawność energetyczną układów ładowania i przetwornic DC-DC w sektorze motoryzacyjnym.

Tranzystory CFD7A z nowej oferty charakteryzują się maksymalnym napięciem V_{DS} równym 650 V, rezystancją $R_{DS(on)}$ od 17 mΩ i ładunkiem bramki od 32 nC. Pozwalają uzyskać większą gęstość mocy i optymalnie wykorzystać dostępną przestrzeń na płytce drukowanej. Dzięki małej indukcyjności pasożytniczej źródła minimalizują poziom zaburzeń EMI, natomiast podwójne wyprowadzenie źródła zwiększa dokładność pomiaru prądu. Szybka dioda regeneracyjna wykazuje mniejszy nawet o 30% ładunek Qrr – w porównaniu z tranzystorami wcześniejszej serii CoolMOS CFDA.

www.infineon.com

Kontroler NFC z wbudowanym elementem SE i obsługą usług cyfrowych STPay-Mobile

STMicroelectronics prezentuje kontroler NFC nowej generacji z wbudowanym elementem SE (secure element), zapewniający urządzeniom mobilnym dostęp do bezpiecznych, bezdotykowych usług sprzedaży biletów i płatności za pośrednictwem platformy STPay-Mobile.

ST54L zapewnia obsługę wielu usług, w tym płatności mobilnych NFC, biletów komunikacji miejskiej, aplikacji kontroli dostępu (np. cyfrowy kluczyk samochodowy) oraz innych, wymagających karty eSIM. Pracuje pod kontrolą bezpiecznego systemu operacyjnego Thales do urządzeń mobilnych, działającego na wbudowanym elemencie SE, co jest sprawdzoną technologią, opracowaną wspólnie przez STMicroelectronics i Thales.

W porównaniu z odpowiednikami poprzedniej generacji, ST54L zawiera większą o 50% pamięć nieulotną o pojemności 3,3 MB. Pozwala to na załadowanie do urządzenia dodatkowych usług, zapewniając jednocześnie użytkownikom dużą ilość miejsca na pobranie dodatkowych profili eSIM.

ST54L został już wprowadzony do masowej produkcji i stanowi element najnowszych smartfonów Google Pixel 8. Uzyskał ponadto akceptację kilku głównych producentów platform mobilnych, co gwarantuje jego bezproblemową integrację przez wytwórców OEM.

www.st.com

Seria cewek powietrznych o dużej dobroci na zakres indukcyjności od 22 do 120 nH

Firma Bourns prezentuje nową serię cewek powietrznych AC4842R o dużej dobroci (>100), produkowanych w wersjach na zakres indukcyjności od 22 do 120 nH. Są to komponenty do montażu SMT, pokryte warstwą epoksydową, kompatybilne z automatami pick and place. Charakteryzują się wąskim przedziałem tolerancji (±10%), rezystancją uzwojenia od 4,2 do 17,3 mΩ i zakresem prądów znamionowych od 1,5 do 3,5 A. Mogą być stosowane w obwodach strojenia w.cz. w odbiornikach TV i wszelkiego typu urządzeniach radiowych. Są przystosowane do pracy w temperaturze otoczenia od –25 do +125°C.

www.bourns.com

Miniaturowy spektrometr UV na zakres długości fali 190...440 nm – do badania jakości wody

Firma Hamamatsu Photonics opracowała miniaturowy spektrometr, wrażliwy na promieniowanie UV, zamykany w obudowie o wymiarach 20,1×12,5×10,1 mm. Pracuje on w zakresie długości fali od 190 do 440 nm i zapewnia rozdzielczość widmową (FWHM) wynoszącą typowo 5,5 nm. Poza małymi wymiarami, do jego zalet należy też duża czułość i mały pobór mocy.

C16767MA został zrealizowany w technologii MEMS. Rozdziela promieniowanie UV na wiele podzakresów i mierzy intensywność promieniowania w każdym z nich. Jest polecany do zastosowań m.in. przy badaniach jakości wody, pozwalając określić zawartość różnych rodzajów zanieczyszczeń, np. azotynów, kwasu azotowego i zanieczyszczeń organicznych (BOD) z wykorzystaniem spektrofotometrii absorpcyjnej. Ułatwia projektowanie i produkcję tanich analizatorów jakości wody, które można powszechnie instalować w różnych ośrodkach badawczych, poprawiając dokładność oceny jakości wody na dużych obszarach.

www.hamamatsu.com

Seria mikrokontrolerów PIC18-Q24 z nowymi funkcjami zabezpieczania kodu i danych

Coraz więcej urządzeń elektronicznych współpracuje z usługami w chmurze, dlatego też bardzo ważne staje się stosowanie – już na poziomie chipa – zaawansowanych środków bezpieczeństwa, zapewniających

REKLAMA

eprasa.pl 65e3eb482b

ochronę danych i oprogramowania firmware. Do tego celu firma Microchip zaprojektowała rodzinę mikrokontrolerów PIC18-Q24, wyposażonych w nowe mechanizmy zabezpieczeń.

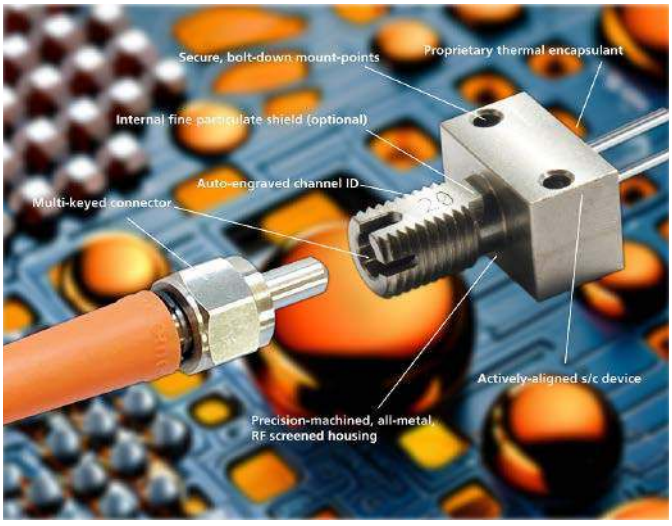
Układy te oferują funkcję PDID (Programming and Debugging Interface Disable), pozwalającą zablokować dostęp do interfejsu programowania/debugowania ICSP oraz blokującą nieautoryzowane próby odczytu, modyfikacji i usunięcia oprogramowania firmware.

Ponieważ wiele bezpiecznych systemów często komunikuje się z zewnętrznymi czujnikami, pamięciami i mikroprocesorami, w mikrokontrolerach PIC18-Q24 wprowadzono wielonapięciowe linie MVIO (Multi-Voltage I/O). Eliminują one potrzebę stosowania zewnętrznych przesuwników poziomu i umożliwiają bezpośrednią współpracę z wejściami/wyjściami cyfrowymi o różnych napięciach roboczych. Oprócz zmniejszenia złożoności płytki PCB i zredukowania kosztów podzespołów, linie MVIO sprawiają, że mikrokontrolery PIC18-Q24 doskonale nadają się do zarządzania systemem, realizując zadania związane z monitorowaniem oraz telemetrią na potrzeby większych mikroprocesorów. Te pozornie rutynowe zadania są zazwyczaj najbardziej podatne na ataki hakerów, próbujących uzyskać dostęp do systemów embedded.

Kolejną nowością jest opcja niezmiennego programu ładującego (immutable bootloader) do aplikacji, które wymagają bezpiecznych metod aktualizacji oprogramowania firmware.

Mikrokontrolery PIC18-Q24 zawierają 10-bitowy przetwornik A/C z funkcją obliczeniową (ADCC) o szybkości próbkowania do 300 kSps, 2 bloki komunikacyjne SPI/I²C, 4 kontrolery DMA, 6 wyjść PWM o 16-bitowej rozdzielczości, 2 timery uniwersalne i 2 moduły wykrywania przejścia przez zero (ZCD). Cechy te sprawiają, że nowe układy nadają się doskonale do zastosowań w automatyce przemysłowej, motoryzacji, elektronice użytkowej i systemach zarządzania energią.

www.microchipdirect.com



Kodowane gniazda SMA do systemów optycznych eliminują błędy połączeń

Firma OMC, specjalizująca się w projektowaniu i produkcji komponentów z sektora optoelektroniki, wdrożyła nową rodzinę kodowanych złączy światłowodowych SMA z wieloma rowkami wpustowymi. Nowe komponenty pozwalają uzyskać do 50 różnych konfiguracji mechanicznych wtyk-gniazdo. Zaprojektowane początkowo do instalacji 4-kanałowych, sprzyjają eliminowaniu ryzyka błędów instalacyjnych.

W wielokanałowych systemach optycznych typowym sposobem podłączania kabli światłowodowych do odpowiedniego nadajnika lub odbiornika na płytce drukowanej jest oznaczenie każdego kabla tulejką identyfikacyjną i dopasowanie go do gniazda na podstawie schematu instalacji. Jednak nie jest to metoda odporna na błędy, które przy dużej liczbie połączeń dość łatwo popełnić i które okazują się czasochłonne w późniejszej identyfikacji.

Aby wyeliminować ryzyko błędów instalacyjnych po stronie użytkownika, firma OMG zaprojektowała nową rodzinę kodowanych gniazd i wtyków, oferujących wszystkie zalety złączy SMA: solidną konstrukcję, łatwy montaż i nakrętkę zabezpieczającą. Dodatkowo zawierają one dwa rowki wpustowe – jeden pionowy, ustalający orientację wtyku i drugi kodujący, możliwy do ustawienia w jednym z wielu położeń kątowych, a ustalający numer kanału. Dzięki temu systemowi niewykonalne staje się podłączenie niewłaściwego kabla optycznego do gniazda w urządzeniu nadawczo-odbiorczym. W przypadku większej liczby kanałów wtyki i gniazda urządzeń z niestandardowymi położeniami rowków kodujących mogą być również wykonywane na indywidualne zamówienia.

Nowy system kodowania jest dostępny dla całej rodziny gniazd i wtyków SMA firmy OMC, w tym H22, H1 i Vertical SMA (VSMA), zajmującego powierzchnię jedynie 8×7 mm. Dodatkową zaletą jest fakt, że system wpustów eliminuje ruchy obrotowe, co dodatkowo zwiększa niezawodność złącza.

Do produkcji złączy wielokluczowych stosowane są tokarki CNC z przesuwną głowicą, wykonujące rowki pod odpowiednim kątem oraz grawerujące na złączu unikalny numer kanału. Gniazdo i wtyk będą pasowały do siebie tylko wtedy, gdy będą miały wygrawerowane to samo oznaczenie. Tokarka może automatycznie wytwarzać ponad 50 unikalnych par gniazd i wtyków.

www.omc-uk.com

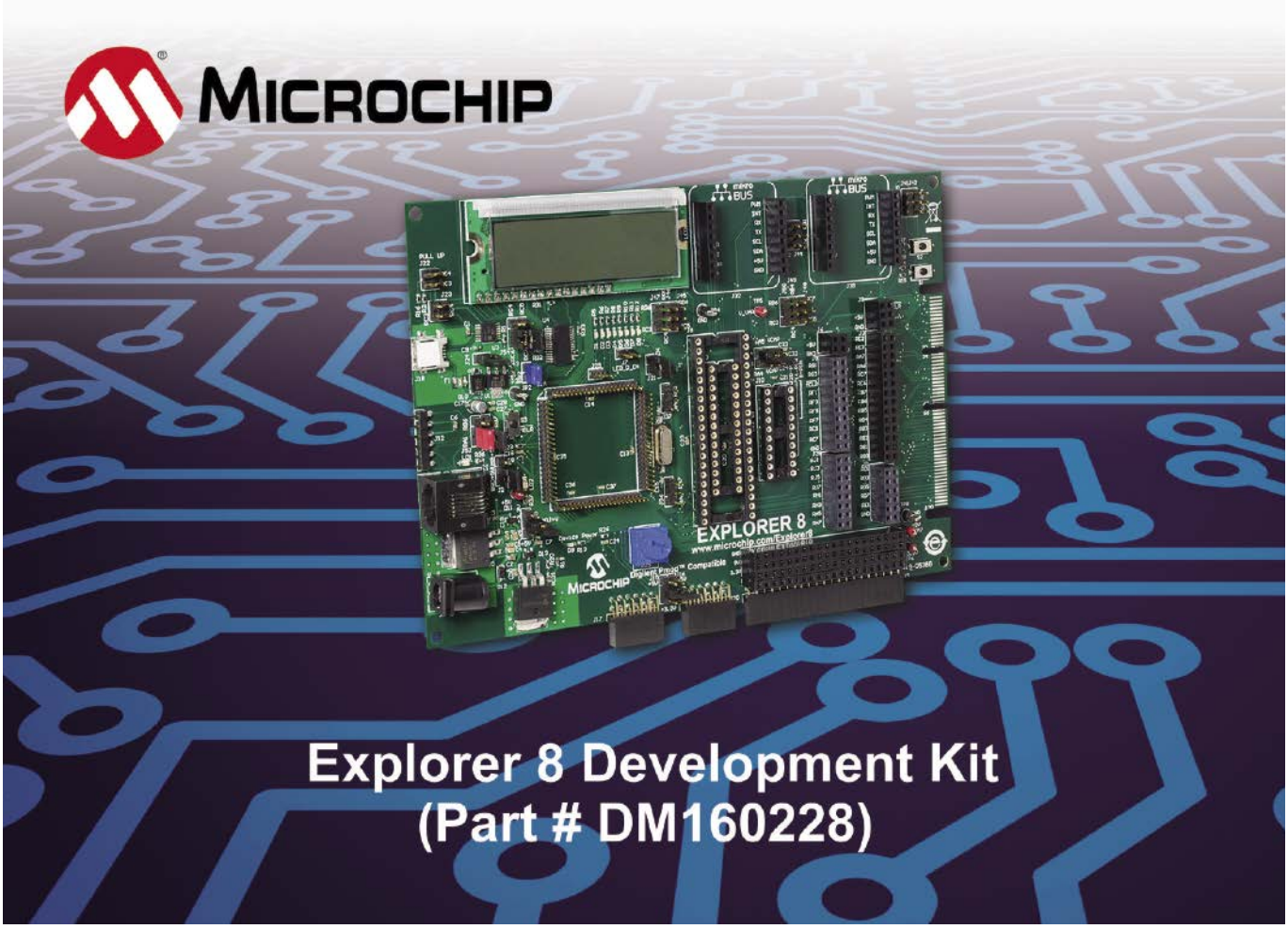
Microchip rozszerza ofertę rozwiązań mSiC o sterownik bramek mSiC 3,3 kV XIFM typu plug-and-play

Postępująca elektryfikacja środków transportu zwiększa zakres zastosowań technologii półprzewodnikowych opartych na węgliku krzemu (SiC). Aby ułatwić konstruktorom wdrażanie rozwiązań SiC w systemach średniego i wysokiego napięcia, m.in. sieciach elektroenergetycznych czy też pojazdach ciężkich, firma Microchip opracowała gotowy do użytku sterownik bramki mSiC™ 3,3 kV, bazujący na opatentowanej technologii Augmented Switching.

Kompaktowy moduł jest wyposażony w cyfrowy układ sterowania, zintegrowany zasilacz i niezawodny interfejs światłowodowy, znacznie podnoszący odporność na zakłócenia. Sterownik bramki ma wstępnie skonfigurowane profile sterowania bramki „on/off", pozwalające zoptymalizować wydajność systemu.

Solidna bariera galwaniczna zapewnia izolację wzmocnioną (na poziomie 10,2 kV) pomiędzy obwodem pierwotnym a obwodem wtórnym. Moduł oferuje funkcję monitorowania temperatury, ochronę przed nadmiernym spadkiem (UVLO) lub wzrostem napięcia (OVLO) czy też zintegrowane zabezpieczenia przeciwzwarciowe i przeciwprzeciążeniowe (DESAT). Driver spełnia wymogi normy EN 50155 obowiązującej w zakresie aplikacji kolejowych. Producent deklaruje, że zastosowanie nowego sterownika bramki SiC może skrócić czas cyklu projektowania docelowego systemu nawet o 50%, w porównaniu z tradycyjnymi rozwiązaniami analogowymi.

eprasa.pl 65e3eb482b



Wygraj zestaw ewaluacyjny Microchip Explorer 8

W kolejnym konkursie organizowanym przez firmę Microchip pod patronatem „Elektroniki Praktycznej" możesz wygrać rozbudowany zestaw deweloperski Microchip Explorer 8 (DM160228). Ale nawet jeżeli tu akurat szczęście Ci nie dopisze, i tak otrzymasz kupon z 15% rabatem oraz bezpłatną wysyłkę jednej z tych płytek.

Zestaw Microchip Explorer 8 to w pełni funkcjonalna platforma rozwojowa do pracy z 8-bitowymi mikrokontrolerami PIC. Narzędzie stanowi wszechstronną bazę do eksperymentów i prototypowania, obejmującą kilka opcji dołączanych czujników, komunikację z układami zewnętrznymi oraz podstawowy interfejs użytkownika. Dodatkowo płytka oferuje sporo miejsca na dalszą rozbudowę, co czyni ją doskonałym rozwiązaniem dla programistów i inżynierów poszukujących narzędzia z największą liczbą obsługiwanych 8-bitowych mikrokontrolerów PIC.

Explorer 8 jest najnowszą propozycją należącą do serii narzędzi klasy korporacyjnej przeznaczonych do 8-bitowych procesorów PIC. Płytka stanowi efekt ewolucji popularnego zestawu Explorer PIC18 i została zaktualizowana, tak aby wykorzystać najważniejsze cechy omawianych mikrokontrolerów, w tym: niezależne bloki peryferyjne, kompatybilność ze wszystkimi MCU z rodziny PIC, czy też uniwersalne wsparcie układów w obudowach 8-, 14-, 20-, 28-, 40-, 44-, 64- i 80-pinowych.

Dzięki zestawowi deweloperskiemu Explorer 8 możesz mieć pod ręką pojedynczą platformę programistyczną, która umożliwia łatwą aktualizację starszego projektu, przy jednoczesnej obsłudze naszych najnowszych modeli MCU. Wbudowane komponenty pozwalają na prostsze tworzenie interfejsu użytkownika, konwersję mocy, ładowanie akumulatorów, czy też budowę rozwiązań z zakresu IoT przy użyciu wydajnego, 8-bitowego mikrokontrolera.

Explorer 8 oferuje również spore pole do popisu w zakresie dalszej rozbudowy. Dwa interfejsy Diligent Pmod, dwa gniazda płytek MikroElectronika Click i dwa niestandardowe złącza rozszerzeń umożliwiają szybką adaptację – w miarę zmieniających się potrzeb projektowych. Zestaw obsługuje wiele narzędzi programistycznych, w tym emulator obwodów PICkit 5, ICD 5 i MPLAB REAL ICE.

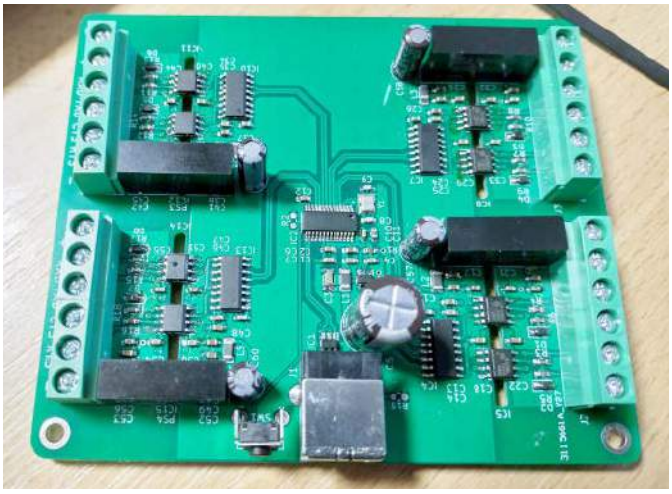
Aby zyskać szansę na wygranie zestawu rozwojowego Microchip Explorer 8 lub otrzymanie kuponu z 15% zniżką i bezpłatną wysyłką, odwiedź stronę: https://page.microchip.com/E-Prak-Exp8.html i wprowadź swoje dane w krótkim formularzu zgłoszeniowym online.

Powodzenia!

eprasa.pl 65e3eb482b

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.



Poczwórny konwerter USB-Serial

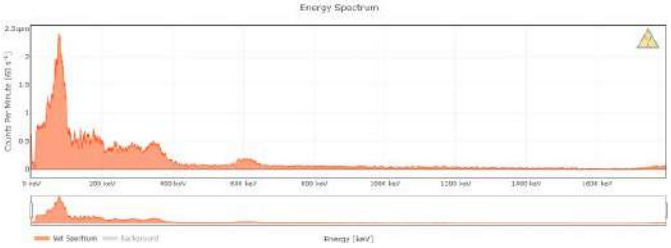
Ten prosty konwerter zbudowany został na bazie niedrogich podzespołów produkcji azjatyckiej i pozwala na współpracę maksymalnie czterech urządzeń z interfejsem szeregowym UART 3,3 V. Połączenie z komputerem PC realizowane jest poprzez gniazdo USB-B, zaś sam port został wyposażony w zabezpieczenie ESD w postaci układu USBLC6-2SC6. Zasilacz USB 5 V jest chroniony przed przepięciami lub zwarciami za pomocą ogranicznika prądu MT9700 (lub BL2554) z limitem ustawionym na 1,5 A.

Do czterokanałowego koncentratora USB CH334U autor projektu podłączył cztery konwertery USB – UART typu CH340C, które – po wpięciu całości do portu USB komputera PC – są widziane przez system operacyjny jako wirtualne porty COM. Sygnały RX, TX, CTS i RTS zostały doprowadzone na złącza wyjściowe za pomocą izolatorów pojemnościowych π122M31. Do zasilania strony zewnętrznej każdej z barier galwanicznych (tj. od strony urządzeń UART) autor zastosował z kolei izolowane przetwornice modułowe DC-DC typu IVB0105S3V3.

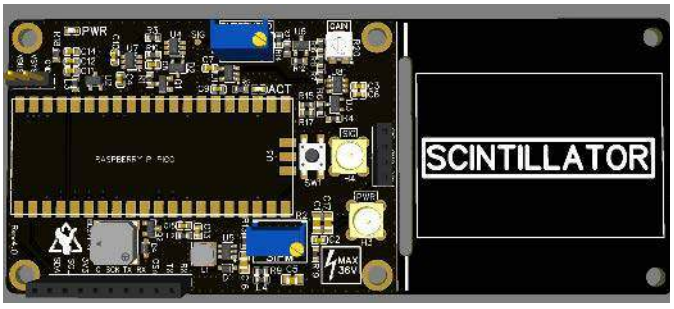
https://tiny.pl/dh8hk

Zintegrowany, kieszonkowy spektrometr promieniowania gamma

Projekt bazuje na module Raspberry Pi Pico i jest wyposażony w scyntylator (np. typu NaI(Tl)) oraz krzemowy fotopowielacz (SiPM). Cały proces przetwarzania sygnałów realizuje mikrokontroler RP2040. Detektor używa krzemowego fotodetektora (SiPM) zamiast klasycznego fotopowielacza, dzięki czemu cała konstrukcja jest znacznie mniejsza i bezpieczniejsza w porównaniu do urządzeń starszej generacji – w tym przypadku nie są już bowiem wymagane wysokie napięcia do zasilania poszczególnych dynod. Widma promieniowania gamma, zmierzone za pomocą urządzenia, mogą być zapisywane do pamięci Flash współpracującej z mikrokontrolerem bądź przesyłane do komputera za pomocą wirtualnego portu szeregowego USB.



Całkowity rozmiar urządzenia wynosi 120 mm × 50 mm, z czego około 70 mm × 50 mm zajmuje elektronika, zaś pozostałe 50 mm × 50 mm przewidziane jest na montaż scyntylatora. Widok płytki drukowanej można zobaczyć na poniższym renderze.



Zastosowany układ pomiarowy współpracuje z fotopowielaczami krzemowymi przeznaczonymi do działania w zakresie napięcia zasilania od 27,5 V do 33,8 V i oferuje rozdzielczość spektrum na poziomie 4096 punktów. Układ wykazuje około 20 µs czasu martwego w trybie spektrometru oraz około 5 µs w trybie licznika Geigera (tj. bez różnicowania amplitudy impulsów).

Całość pobiera około 20 mA prądu przy zasilaniu napięciem 5 V. Moduł posiada wbudowany buzzer do sygnalizacji wykrytych impulsów. Dodatkowo na płytce znalazły się wyprowadzone piny zasilające i złącza I²C, SPI oraz UART, co pozwala na dołączenie dodatkowych modułów (np. wyświetlacza czy karty SD). W standardowej wersji oprogramowania zaimplementowane jest już wsparcie obsługi ekranów OLED ze sterownikami SSD1306 lub SH110x.

Układ elektroniczny został zmontowany na 4-warstwowej płytce drukowanej, przy czym wszystkie komponenty trafiły na górną stronę PCB. Autor wybrał głównie podzespoły w obudowach 0603.

eprasa.pl 65e3eb482b

Połączenie SiPM z płytką detektora można zrealizować za pomocą krótkich kabli przylutowanych bezpośrednio do listwy goldpin (widocznej po prawej stronie płytki) lub zastosować złącza MCX. Następnie całą konstrukcję SiPM/scyntylatora warto położyć na PCB i zabezpieczyć za pomocą opaski zaciskowej. Autor przewidział także odcięcie części PCB, przeznaczonej na zamocowanie zespołu detektora, wzdłuż białych linii, co daje pewną elastyczność podczas montażu w obudowie.

W zależności od rozmiaru scyntylatora (który wpływa na ilość emitowanych przezeń fotonów), a także pożądanego zakresu widma oraz rozdzielczości, należy odpowiednio dostosować napięcie progowe dyskryminatora. Na stronie projektu zaprezentowane zostały wyniki wszystkich pomiarów wykonanych przy użyciu scyntylatora o objętości 30,5 cm^3.

Aby umożliwić analizowanie zarejestrowanych widm i na żywo prezentować dane z portu szeregowego, autor projektu zaprogramował specjalną aplikację internetową, dostępną pod adresem: https://spectrum.nuclearphoenix.xyz/.

https://tiny.pl/dh8q1
https://tiny.pl/dh8qj

Boja monitoringu akustycznego

Celem projektu było stworzenie systemu do podwodnego monitoringu akustycznego przy użyciu 4 podwodnych elementów i boi powierzchniowej. Łączność systemu zrealizowana została za pomocą jednoparowej sieci Ethernet.

Jednostka dolna (podwodna) składa się z 4 hydrofonów, czujnika temperatury, sensora głębokości, systemu akwizycji danych akustycznych (PAMELA DAQ produkcji Applied Ocean Sciences) oraz jednoparowego nadajnika Ethernet. Wiadomości z urządzenia są przesyłane w górę, tj. do boi powierzchniowej, skąd moduł radiowy może przekazywać je do odbiorcy znajdującego się na brzegu.

Moduł akwizycji danych Applied Ocean Science – PAMELA DAQ – współpracuje z niezależnem rejestratorem, zawierającym kodek SGTL5000 i wzmacniacz. Do modułu akwizycji danych podłączone są m.in. hydrofony naukowe Aquarian AS-1, czujnik ciśnienia Blue Robotics do pomiaru głębokości (MS5837-30BA) i czujnik temperatury Blue Robotics (TSYS01). Dodatkowo moduł uzupełnia pakiet LiPo (3s5p), który zasila cały system.

Przetwornik Ethernetu jednoparowego Sparkfun pozwala na komunikację na odległości niemal kilometra ze stacją bazową utrzymującą się na powierzchni wody. Stacja bazowa wyposażona jest w moduł radiowy i system do śledzenia pozycji.

Oprócz integracji wszystkich komponentów, głównymi wyzwaniami przy konstrukcji systemu okazały się problemy związane z wykonaniem szczelnych połączeń do pracy na dużej głębokości. W tym celu autor opracował specjalne przepusty, które pozwalają na przewlekanie przewodów przez obudowę boi, a jednocześnie utrzymują oczekiwaną szczelność konstrukcji.

Finalnie wnętrze boi ma zostać zalane żywicą. Jak wskazuje autor, większość profesjonalistów używa do tego celu Loctite DP420, ale on postanowił sięgnąć po odpowiednik – Infinity Bond EP420, który jest znacznie tańszy (16 $ za 50 ml) i ma podobne właściwości. Zastosowanie żywicy powinno zapewnić konstrukcji wodoodporność. Sam autor wątpi wprawdzie, czy rozwiązanie takie sprawdzi się na naprawdę dużych głębokościach, jednak ma nadzieję, że boja będzie działała na głębokości sięgającej ok. 30 metrów pod wodą.

https://tiny.pl/dh8qk

REKLAMA

eprasa.pl 65e3eb482b

AVT sklep

W ofercie AVT*
AVT6027

Podstawowe parametry:
- zakres pomiarowy mierzonego napięcia: 0...34 V,
- rozdzielczość pomiaru napięcia: 10 mV,
- zakres pomiarowy mierzonego prądu: 0...10 A,
- rozdzielczość pomiaru prądu: 1 mA,
- zakres pomiarowy mierzonej mocy: 0...340 W,
- rozdzielczość pomiaru mocy: 100 mW,
- zakres pomiarowy dostarczonego ładunku: 0...100 Ah,
- rozdzielczość pomiaru ładunku: 10 mAh,
- częstotliwość rejestracji danych na karcie SD: 5 razy na sekundę,
- maksymalny pobór prądu ze źródła napięcia zasilającego: 40 mA.

Dodatkowe materiały do pobrania ze strony www.ulubionykiosk.pl/media

AVT5969	powerMonitor – miniaturowy miernik parametrów elektrycznych (EP 2/2023)
AVT5954	Warsztatowy Multi Tester (EP 10/2022)
AVT5766	Ampera – miniaturowy miernik prądu stałego (EP 5/2020)
AVT5672	USBtester – monitor parametrów zasilania USB (EP 5/2019)
Projekt 232	Miliamperomierz TRMS (EP 9/2017)
AVT5507	Miernik UIPTR (EP 7/2015)
AVT5488	Kontroler obciążenia portu USB (EP 2/2015)
AVT1823	Monitor prądu USB (EP 8/2014)
AVT5399	Dwukanałowy multimetr panelowy (EP 6/2013)
AVT5386	Podwójny woltomierz i amperomierz (EP 3/2013)

* **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]** – płytka drukowana bez elementów i dokumentacji.

Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- **wersja [A+]** – płytka drukowana **[A]** + zaprogramowany układ **[UK]** i dokumentacja,
- **wersja [UK]** – zaprogramowany układ.

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik PDF! Podczas składania zamówienia upewnij się, którą wersję zamawiasz!
http://sklep.avt.pl

W przypadku braku dostępności na stronie sklepu osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: **kity@avt.pl**.

powerTracker

Rejestrator parametrów elektrycznych

Pomiar parametrów zasilania urządzenia elektronicznego wymaga zastosowania dwóch osobnych mierników (woltomierza i amperomierza) lub laboratoryjnego miernika mocy. Pierwsze rozwiązanie wiąże się z koniecznością żmudnego podłączania przewodów w odpowiedniej konfiguracji, drugie jest przeważnie zarezerwowane tylko dla profesjonalnych laboratoriów pomiarowych. Opisane urządzenie umożliwia rejestrację napięcia i prądu stałego, mocy oraz ładunku dostarczonego do testowanego odbiornika – doskonale sprawdzi się zatem w rozmaitych aplikacjach warsztatowych oraz serwisowych.

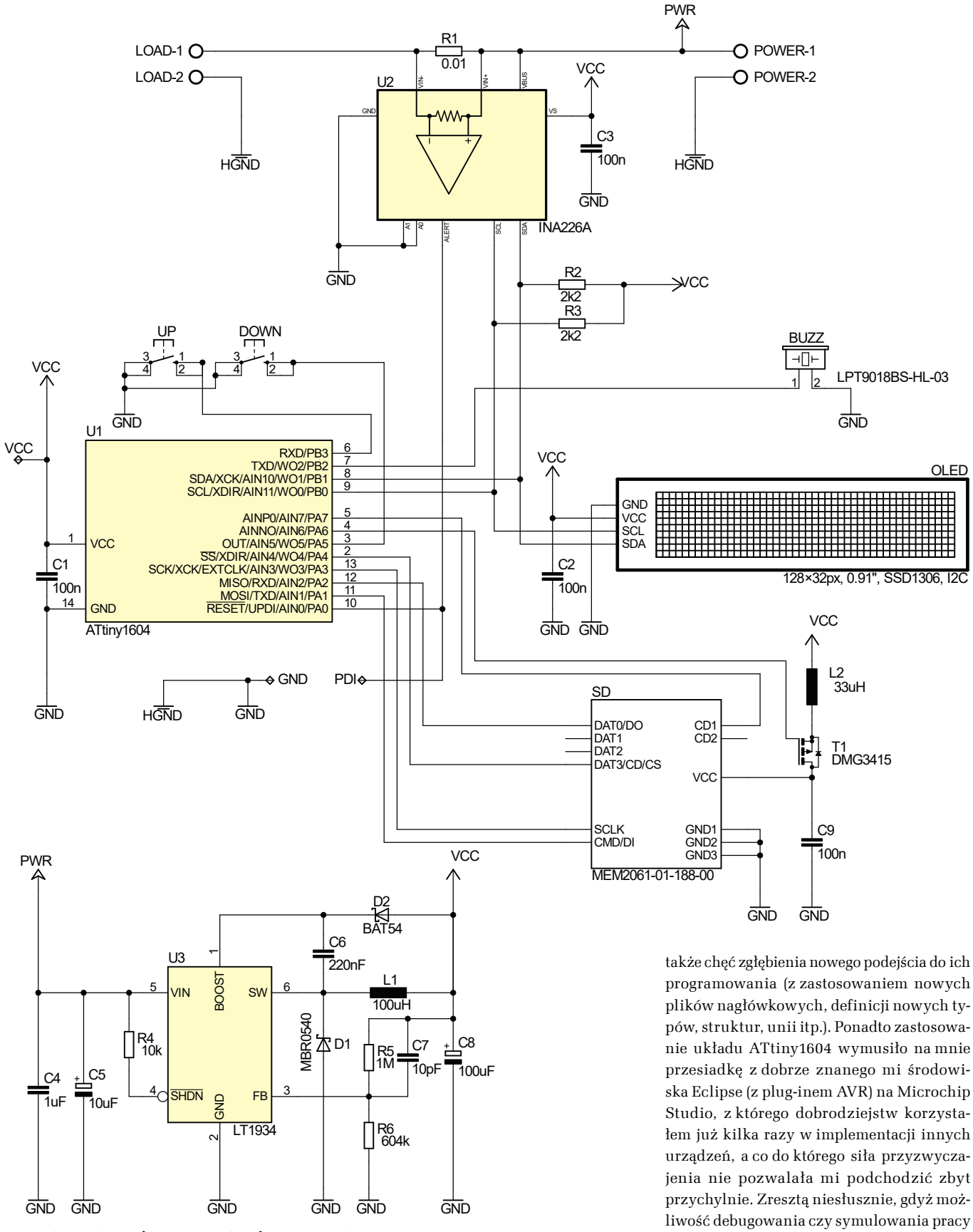
Pomysł na ten projekt podsunął mi jeden z naszych Czytelników (przy okazji: pozdrowienia dla Pana Janusza), który w mailu skierowanym do mnie wyrażał uznanie dla mojej inwencji twórczej – jednym z jej przejawów miał być projekt opublikowany w numerze 1/2023 „Elektroniki Praktycznej", urządzenie o nazwie powerMonitor. Zadaniem tej konstrukcji było monitorowanie parametrów elektrycznych odbiornika energii prądu stałego. Niestety, a może i na szczęście, do zachwytów dodana została przysłowiowa łyżka dziegciu. Wspomniany Czytelnik zauważył, iż tego typu monitor powinien być wyposażony w opcję rejestrowania mierzonych wartości elektrycznych, by można było je później poddać analizie – gdyż ich prezentacja na tak małym wyświetlaczu jest dyskusyjna. Cóż, nie sposób nie zgodzić się z tego rodzaju tezą, w związku z czym zabrałem się ochoczo za prace projektowe. Postanowiłem zbudować urządzenie o zbliżonej funkcjonalności, lecz wyposażone w możliwość rejestracji danych na karcie micro SD. W ten prozaiczny sposób narodził się projekt o nazwie powerTracker, który jest tematem niniejszego artykułu. Przystępując do prac konstrukcyjnych, poczyniłem wstępne założenia co do funkcjonalności urządzenia, które prezentują się następująco:

- możliwość pomiaru napięcia na zaciskach odbiornika,
- możliwość pomiaru prądu i mocy pobieranej przez odbiornik,
- możliwość pomiaru ładunku przekazanego do odbiornika,
- możliwość generowania alarmów po przekroczeniu konfigurowalnych parametrów – niezależnie dla każdej z wielkości elektrycznych (alarm powyżej lub poniżej ustawionej wartości),
- możliwość rejestracji parametrów elektrycznych na karcie micro SD sformatowanej zgodnie z systemem plików FAT16/FAT32.

Jako że, konstruując poprzednie urządzenie, zdobyłem niezbędne doświadczenie w kwestii wykorzystywanych peryferiów, postanowiłem użyć ich i tym razem. Jedynym dodatkowym problemem, z jakim musiałem się zmierzyć, była implementacja obsługi karty standardu SD oraz systemu plików, co z góry wykluczało zastosowanie prostego mikrokontrolera ATtiny84 z poprzedniego projektu. Tym razem musiałem sięgnąć po bardziej nowoczesny procesor, którego zasoby sprzętowe sprostałyby stawianemu przed nim zadaniu. Tak oto powstał system mikroprocesorowy, który pokazano na **rysunku 1**.

Jak widać, mamy tu bardzo prosty układ, którego sercem jest niewielki, acz nowoczesny mikrokontroler firmy Microchip (dawniej Atmel) typu ATtiny1604 (należący do nowej rodziny 0-series AVR), taktowany wewnętrznym generatorem RC o częstotliwości 3,333 MHz. Odpowiada on za sprzętową implementację interfejsu I²C (nazywanego przez firmę Microchip mianem TWI), przy użyciu którego mikrokontroler realizuje obsługę układu INA226 – będącego specjalizowanym, bardzo dokładnym, 16-bitowym, różnicowym przetwornikiem ADC – oraz niewielkiego, ale bardzo efektownego wyświetlacza OLED o rozdzielczości 128×32 px,

eprasa.pl 65e3eb482b



Rysunek 1. Schemat ideowy urządzenia powerTracker

stanowiącego element graficznego interfejsu użytkownika. Wybór tego konkretnego mikrokontrolera z szerokiej rodziny układów firmy Microchip podyktowany był chęcią utrzymania małego footprintu obudowy (podobnie jak ATtiny84) oraz wymogiem dostępności większej ilości pamięci Flash i RAM. Inną, trochę mniej oczywistą przyczyną wyboru tego układu była chęć poznania nowej rodziny mikrokontrolerów AVR oraz sposobu obsługi zaktualizowanych peryferiów, w jakie je wyposażono – ale także chęć zgłębienia nowego podejścia do ich programowania (z zastosowaniem nowych plików nagłówkowych, definicji nowych typów, struktur, unii itp.). Ponadto zastosowanie układu ATtiny1604 wymusiło na mnie przesiadkę z dobrze znanego mi środowiska Eclipse (z plug-inem AVR) na Microchip Studio, z którego dobrodziejstw korzystałem już kilka razy w implementacji innych urządzeń, a co do którego siła przyzwyczajenia nie pozwalała mi podchodzić zbyt przychylnie. Zresztą niesłusznie, gdyż możliwość debugowania czy symulowania pracy mikrokontrolera jest nie do przecenienia.

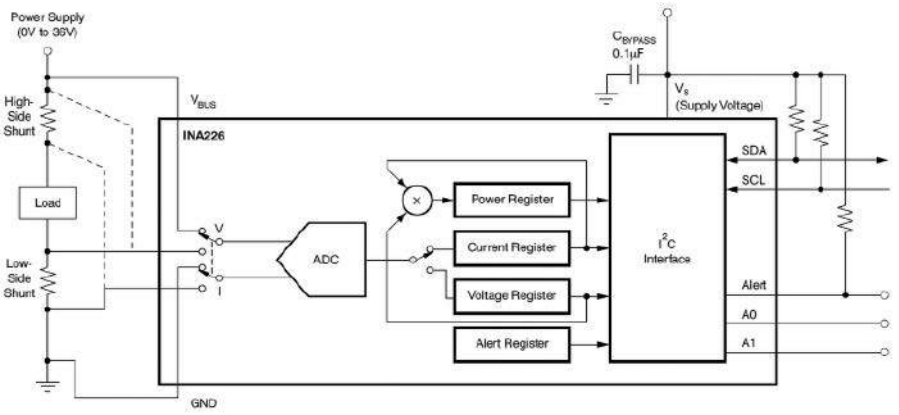
Mikrokontroler w naszym systemie odpowiedzialny jest dodatkowo za obsługę 2 microswitchy UP i DOWN. W celu eliminacji drgań styków używa on wbudowanego 16-bitowego układu czasowo-licznikowego TCB0, pracującego w trybie Periodic Interrupt. Obsługuje również buzzer piezoelektryczny

eprasa.pl 65e3eb482b

(bez wewnętrznego generatora), co realizowane jest z kolei z użyciem 16-bitowego układu czasowo-licznikowego TCA0, pracującego w trybie Frequency Generation (i generującego przebieg prostokątny o częstotliwości 4 kHz na wyprowadzeniu W02 (PB2), zasilający wspomniany buzzer). Wspomniany wcześniej, specjalizowany przetwornik ADC mierzy spadek napięcia na rezystorze szeregowym R1 (10 mΩ), dzięki czemu możliwe jest wyznaczenie prądu pobieranego przez badane urządzenie. Nie jest to jednak zwykły, zewnętrzny przetwornik ADC, jakich wiele na rynku, a specjalizowany układ przeznaczony do pomiaru prądu, napięcia i mocy urządzeń zasilanych napięciem stałym. Jako że jest to element dość wyjątkowy, warto choćby skrótowo zaznajomić się z jego specyfikacją. Tak jak napisano wcześniej, układ INA226 produkcji firmy Texas Instruments jest bardzo dokładnym, 16-bitowym, różnicowym przetwornikiem pomiarowym ADC, przeznaczonym do współpracy z bocznikiem rezystancyjnym. Układ ten wyróżnia się następującymi cechami użytkowymi:

- szeroki zakres napięć zasilania 2,7...5,5 V,
- bardzo duża dokładność pomiaru rzędu 0,1%,
- możliwość pracy w systemach o szerokim zakresie napięć szyny zasilającej 0…36 V,
- możliwość pracy w konfiguracjach Low-side i High-side,
- bezpośredni pomiar napięcia, prądu i mocy,
- konfigurowalny czas przetwarzania wbudowanego przetwornika ADC,
- konfigurowalna funkcja uśredniania pomiarów,
- dwa tryby pracy wbudowanego przetwornika ADC: ciągły i wyzwalany na żądanie,
- możliwość alarmowania po przekroczeniu zadanego poziomu prądu, napięcia szyny zasilającej odbiornik bądź mocy pobieranej przez odbiornik.

Jak widać, układ INA226 idealnie wpisuje się w wymagania naszej aplikacji, oferując niespotykaną dotąd funkcjonalność i dokładność pomiarów. Schemat blokowy tego peryferium pokazano na **rysunku 2**.



Rysunek 2. Schemat blokowy układu INA226

Opisywany układ dokonuje ciągłego (lub wyzwalanego manualnie przez aplikację użytkownika) pomiaru dwóch wartości napięć: napięcia szyny zasilającej odbiornik (***VBUS***) oraz napięcia na zaciskach bocznika rezystancyjnego (***VSHUNT***), włączonego w szereg z odbiornikiem. Na podstawie tych dwóch wielkości oraz zawartości rejestru konfiguracyjnego CALIBRATION (którego zawartość zależy od wymaganej rozdzielczości pomiaru i parametrów zastosowanego bocznika rezystancyjnego), układ oblicza prąd oraz moc pobieraną przez odbiornik, po czym udostępnia je aplikacji użytkownika, ładując wyniki obliczeń do stosownych rejestrów konfiguracyjnych, jak również ustawiając dedykowane flagi zakończenia konwersji. Ponadto, dzięki wyposażeniu w grupę specjalnych rejestrów konfiguracyjnych (odpowiedzialnych za porównywanie zmierzonych i obliczonych wartości z wartościami progowymi) oraz dedykowane wyprowadzenie oznaczone jako **ALERT**, umożliwia generowanie alarmów po przekroczeniu zdefiniowanych przez użytkownika progów dotyczących: napięcia szyny zasilającej, napięcia na boczniku pomiarowym i mocy pobieranej przez odbiornik. Co więcej: producent tego peryferium wyposażył je w możliwość niezależnej konfiguracji czasu przetwarzania przetwornika ADC, oddzielnie dla napięcia szyny zasilającej i napięcia bocznika rezystancyjnego, a także możliwość uśredniania tychże wielkości spośród wielu, kolejno następujących pomiarów. Dzięki takiemu podejściu zwiększono wydatnie funkcjonalność układu i możliwość dostosowania trybu jego pracy do wymagań konkretnej aplikacji. Pamiętać należy jedynie, że wydłużenie czasu przetwarzania wbudowanego przetwornika ADC wydatnie podwyższa uzyskaną dokładność pomiaru, zaś uśrednianie większej liczby próbek zdecydowanie zwiększa odstęp sygnału od szumu, w związku z czym w rzeczywistych aplikacjach należy dobierać maksymalne i możliwe do zaakceptowania wartości tychże parametrów, kierując się dla przykładu szybkością zmian badanych przebiegów jako kryterium wyjściowym. Co oczywiste, by poznać wszystkie możliwości drzemiące w układzie INA226, należałoby sięgnąć do jego noty aplikacyjnej lub do… mojego artykułu wspomnianego wcześniej. Szczegółowo omawiam w nim wszelkie zagadnienia implementacyjne, dlatego nie będę ich powielał w niniejszym opracowaniu, tym bardziej że jest to dość obszerna lektura.

Uważny Czytelnik zauważy z pewnością dość rozbudowany blok zasilający, zbudowany z użyciem nowoczesnej, scalonej przetwornicy step-down typu LT1934 firmy Analog Devices, która w zastosowanej wersji może dostarczać prąd o wartości 300 mA. Dlaczego zastosowałem przetwornicę i w dodatku tak nietypową? To proste. Chciałem, by nasze urządzenie było zasilane z tego samego źródła, co badany odbiornik, czyli de facto było włączane w szereg pomiędzy źródłem zasilania a badanym odbiornikiem. Jako że założyłem sobie dość szeroki zakres napięć dla źródeł zasilania, niemożliwe stało się zastosowanie zwykłego stabilizatora liniowego (nawet LDO) w celu zasilenia systemu mikroprocesorowego z uwagi na zbyt dużą moc traconą i problem z odprowadzeniem ciepła (nie mówiąc już nawet o sprawności takiego układu). Okazało się dość szybko, iż wśród dostępnych na rynku półprzewodników trudno jest znaleźć taki, który spełniałby

Wykaz elementów:

Rezystory:
R1: pomiarowy 10 mΩ 1 W 1% (SMD 1206)
R2, R3: 2,2 kΩ (SMD 0805)
R4: 10 kΩ (SMD 0805)
R5: 1 MΩ 1% (SMD 0805)
R6: 604 kΩ 1% (SMD 0805)

Kondensatory:
C1...C3, C9: 100 nF (SMD 0805, X7R)
C4: 1 µF (SMD 0805, X7R)
C5: 10 µF/50 V typu 50SVF10M PANASONIC (polimerowy, LOW ESR, SANYO SMD C6)
C6: 220 nF (SMD 0805, X7R)
C7: 10 pF (SMD 0805, X7R)
C8: 100 µF/10 V (tantalowy, LOW ESR, C/6032-28R)

Półprzewodniki:
U1: ATtiny1604 (SOIC14)
U2: INA226A (MSOP-10)
U3: LT1934 (SOT23)
T1: DMG3415 (SOT23)
D1: MBR0540 (SOD123)
D2: BAT54 (SOT23)
OLED: wyświetlacz OLED 128x32 px, 0,91", sterownik SSD1306, magistrala I²C, wymiary 38 x 12 mm

Pozostałe:
L1: 100 µH typu DLG-0504-101
L2: 33 µH (SMD 0805)
UP, DOWN: microswitch SMD typu DTSM-65N-V-B lub podobny
BUZZ: przetwornik piezoelektryczny SMD typu LPT9018BS-HL-03-4.0-12-R
LOAD, POWER: złącze śrubowe AK500/2
SD: gniazdo karty micro SD typu MEM2061-01-188-00-A (GLOBAL CONNECTOR TECHNOLOGY)

eprasa.pl 65e3eb482b

oczekiwane założenia (napięcie wejściowe rzędu 34 V i wyjściowe 3,3 V), a do tego występował w obudowie, która będzie łatwa do montażu przez amatora. Prawdę mówiąc, była to jedyna scalona przetwornica, która spełniała wszystkie wstępne założenia projektowe. Nie będę w tym miejscu wchodził w szczegóły implementacyjne, gdyż stosowne informacje znajdziemy w drobiazgowej dokumentacji producenta układu – zwłaszcza że zastosowane rozwiązanie jest typową implementacją zaproponowaną w nocie aplikacyjnej, zapewniającą szeroki zakres napięć zasilających (3,3...34 V) oraz maksymalny, dopuszczalny prąd obciążenia rzędu 300 mA, co z zapasem zaspokaja zapotrzebowanie energetyczne urządzenia (zresztą największy prąd pobiera wyłącznie karta SD w trybie zapisu). Niemniej jednak warto wspomnieć o pewnym ograniczeniu tejże przetwornicy. Otóż układ LT1934 może pracować poprawnie już przy napięciu wejściowym rzędu 3,3 V, ale do startu samej przetwornicy niezbędne jest nieco większe napięcie – mianowicie 4,5 V. Gdy przetwornica uruchomi się, napięcie zasilające można zmniejszyć do wspomnianego poziomu, jednak to ograniczenie należy mieć zawsze na uwadze. Co prawda można mu zaradzić, modyfikując minimalnie zaproponowaną implementację poprzez podłączenie anody diody D2 do napięcia wejściowego (VIN) zamiast wyjściowego (VCC). W takim przypadku ograniczymy jednak zakres napięć zasilających: tym razem od góry do maksymalnej wartości 20 V, co wynika z dopuszczalnego napięcia na wyprowadzeniu BOOST przetwornicy, które w takim układzie wynosi w przybliżeniu $2\times V_{IN}$ i nie może przekraczać wartości 40 V. Innym rozwiązaniem jest pozostanie przy schemacie pierwotnym i wstępne obciążenie wyjścia przetwornicy prądem minimalnym ok. 12 mA, co zapewni poprawny start układu nawet przy napięciu wejściowym 3,3 V, nie ograniczając tym samym maksymalnego poziomu napięcia wejściowego, lecz znacznie pogarszając sprawność układu (połowa prądu zostanie w tym przypadku „zmarnowana" na podgrzewanie rezystora obciążającego). Ostateczne rozwiązanie pozostawiam Czytelnikom, gdyż każdy musi określić, na jakie ograniczenia jest skłonny sobie pozwolić. Ja zdecydowałem się na wersję z implementacją producenta układu. Z myślą o dociekliwych Czytelnikach dodam, iż wartość napięcia wyjściowego ustalono na 3,3 V. Co ważne, już w tym miejscu chciałbym zaznaczyć, że w celu zapewnienia optymalnej pracy przetwornicy (w tym maksymalizacji jej sprawności) niezbędne jest zastosowanie odpowiedniej jakości kondensatorów (zwłaszcza elektrolitycznych o niskim ESR), co znajduje odzwierciedlenie w opisie listy elementów. Nie warto bagatelizować tego zagadnienia.

Przejdźmy zatem do kwestii implementacyjnych, które w tym przypadku ograniczę do opisu problemów związanych z obsługą karty SD oraz systemu plików FAT16/FAT32, gdyż pozostałe zagadnienia programistyczne zostały już poruszone w artykule dotyczącym poprzedniego mojego urządzenia. Jak zapewne się domyślacie, obsługa kart SD (w naszym przypadku micro SD) nie jest zagadnieniem łatwym, a jeśli dodać do tego jeszcze sam system plików FAT16/FAT32, zaczyna robić się nieciekawie. Na szczęście ktoś zrobił to wcześniej za nas i to na tyle dobrze (a przy okazji uniwersalnie), że nawet firma Microchip w swoich przykładach podaje wykorzystanie tego właśnie opracowania (nota aplikacyjna AN2799). Mowa o dobrze znanym module Petit FatFs autorstwa elm-chan, który jest uproszczoną wersją modułu FatFS tego samego autora, przeznaczoną na mikroprocesory o niewielkich zasobach pamięci Flash i RAM. Wersja ta ma co prawda pewne ograniczenia, ale za to można ją bez problemu uruchomić na tak skromnych mikrokontrolerach, jak rodzina Tiny AVR, gdyż wymaga jedynie 44 bajtów dostępnej pamięci RAM! Niesamowite, nieprawdaż? Ograniczenia, o których mowa, to:

- brak możliwości tworzenia nowych plików (obsługiwane są wyłącznie pliki istniejące),
- brak możliwości zmiany rozmiaru plików,
- data/godzina utworzenia/modyfikacji pliku nie ulega zmianie nawet po operacji zapisu,
- operacja zapisu musi zaczynać się i kończyć w ramach tego samego sektora,
- atrybut „tylko do odczytu" pliku nie blokuje operacji zapisu.

Jak widać, są to ograniczenia o niezbyt wielkim ciężarze gatunkowym, lecz jeśli w Waszym rozwiązaniu wydają się istotne, zawsze możecie sięgnąć po moduł FatFS (tego samego autora), który – co oczywiste – wymaga już znacznie większych zasobów sprzętowych. Dla naszego projektu ograniczenia takie, jak wyżej opisane, nie mają żadnego praktycznego znaczenia, a skutkiem ich istnienia jest konieczność przygotowania na karcie SD „pustego" pliku o predefiniowanej nazwie i odpowiedniej wielkości (by można było w nim zapisywać dane – wszak moduł Petit FatFs nie może zmienić jego rozmiaru). Co więcej, firma Microchip przygotowała przykład wykorzystania tego modułu – możemy go wygenerować ze środowiska Microchip Studio (przykład AVR42776), a następnie dołączyć do nowego projektu. Co prawda wspomniane oprogramowanie przygotowano dla mikrokontrolera ATtiny817, lecz bez większego trudu powinniśmy dostosować je do używanego typu układu. Tak naprawdę modyfikacji wymaga głównie plik diskio_avr.c (i związany z nim diskio.h), zawierający niskopoziomowe funkcje dostępu do medium komunikacyjnego (SPI) oraz plik pffconf.h, w którym włączyć musimy kompilację warunkową niezbędnych funkcji bibliotecznych oraz obsługę używanych systemów plików (za pomocą stosownych definicji), co ma oczywisty wpływ na rozmiar kompilowanego projektu. Jak widzicie, zagadnienie nie wydaje się szczególnie skomplikowane, a mimo to na wielu forach dyskusyjnych znaleźć można setki wątków, w ramach których użytkownicy opisują swoje nieudane próby uruchomienia wspomnianego oprogramowania. Z tego właśnie powodu zilustruję niezbędne modyfikacje przykładu firmy Microchip, dostosowane do naszego projektu (i mikrokontrolera), rozszerzające jego funkcjonalność o możliwość detekcji obecności nośnika w gnieździe karty SD oraz sterowania zasilaniem

```
//Definicje portów interfejsu SPI karty SD -> DO, DI, SCK, CS
#define SD_PORT_NAME PORTA
#define SD_SCK_MASK PIN3_bm                                  //PA3 -> SCK karty SD
#define SD_DO_MASK PIN2_bm                                   //PA2 -> MISO -> DO karty SD
#define SD_DI_MASK PIN1_bm                                   //PA1 -> MISO -> DI karty SD
#define SD_CS_MASK PIN4_bm                                   //PA4 -> CS karty SD
#define SD_DO_PINCTRL_REG PIN2CTRL

//Definicje pozostałych portów karty SD -> DETECT, POWER
#define SD_POWER_MASK PIN6_bm                                //PA6 -> POWER karty SD (0 -> zasilanie włączone, 1 -> zasilanie wyłączone)
#define SD_DETECT_MASK PIN7_bm                               //PA7 -> CARD DETECT karty SD (Karta obecna w gnieździe -> 1)
#define SD_DETECT_PINCTRL_REG PIN7CTRL
#define SD_POWER_ON SD_PORT_NAME.OUTCLR = SD_POWER_MASK
#define SD_POWER_OFF SD_PORT_NAME.OUTSET = SD_POWER_MASK
#define SD_IS_PRESENT (SD_PORT_NAME.IN & SD_DETECT_MASK)

//Definicje dla portu CS
#define SELECT() SD_PORT_NAME.OUTCLR = SD_CS_MASK            //CS = 0
#define DESELECT() SD_PORT_NAME.OUTSET = SD_CS_MASK          //CS = 1
#define SELECTING ((SD_PORT_NAME.DIR & SD_CS_MASK) && !(SD_PORT_NAME.OUT & SD_CS_MASK))
```

Listing 1. Zmieniony względem oryginału fragment zawartości pliku nagłówkowego diskio.h modułu obsługującego karty SD (w ramach modułu Petit FatFs)

eprasa.pl 65e3eb482b

```
static void init_spi(void)
{
    //Porty MOSI, CS, SCK i POWER jako wyjściowe, zaś MISO i DETECT, jako wejściowe podciągnięte do VCC
    SD_PORT_NAME.DIRSET = SD_DI_MASK|SD_SCK_MASK|SD_CS_MASK|SD_POWER_MASK;
    SD_PORT_NAME.SD_DO_PINCTRL_REG = PORT_PULLUPEN_bm;
    SD_PORT_NAME.SD_DETECT_PINCTRL_REG = PORT_PULLUPEN_bm;

    SPI0.CTRLA = SPI_MASTER_bm|SPI_PRESC_DIV16_gc;          //SPI Master, fclk = 208 kHz dla F_CPU = 3.333 MHz
    SPI0.CTRLB = SPI_SSD_bm;                                //Pin SS kontrolowany przez oprogramowanie
    SPI0.CTRLA |= SPI_ENABLE_bm;                            //Włączenie modułu SPI

    //Włączamy zasilanie karty SD i odczekujemy chwilę na start sterownika karty
    SD_POWER_ON;
    _delay_ms(100);
}
```

Listing 2. Zmieniony względem oryginału fragment zawartości pliku diskio_avr.c, pokazujący zredefiniowane ciało funkcji inicjalizującej interfejs SPI (w ramach modułu Petit FatFs)

wspomnianego medium. Sterowanie zasilaniem karty znajduje zastosowanie w przypadku problemów z poprawną jej inicjalizacją, co zdarzyć się może z różnych powodów. W takim wypadku możemy wyłączyć, a później włączyć jej zasilanie, następnie odczekać czas niezbędny na inicjalizację jej wewnętrznego sterownika, po czym ponownie spróbować zainicjować go zgodnie z naszymi oczekiwaniami. Na **listingu 1** pokazano zmieniony względem oryginału fragment zawartości pliku nagłówkowego diskio.h, w ramach którego zebrano definicje wyprowadzeń karty SD oraz nazwy stosownych rejestrów sterujących.

Dalej, na **listingu 2**, zaprezentowano zmieniony względem oryginału fragment zawartości pliku diskio_avr.c, w ramach którego zredefiniowano ciało funkcji inicjalizującej interfejs SPI.

I na koniec, w pliku pffconf.h, włączamy niezbędne opcje kompilacji, co pokazano na **listingu 3.**

Prawda, że proste? Na koniec, już w ramach aplikacji urządzenia powerTracker, zaimplementowano funkcję inicjalizującą kartę SD wraz z niezbędnym „montażem" woluminu oraz obsługą systemu plików, której ciało pokazano na **listingu 4.**

To wszystko, jeśli chodzi o niezbędne modyfikacje modułu Petit FatFs. Na koniec zagadnień implementacyjnych opiszę jeszcze prosty moduł do obsługi interfejsu I²C, wykorzystujący sprzęg TWI znajdujący się na pokładzie mikrokontrolera ATtiny1604. Przyznam, że nie jest to szczególnie nowatorskie rozwiązanie, jednak interfejs TWI w nowych mikrokontrolerach AVR z serii 0 różni się znacząco (pod względem sposobu obsługi i rozmieszczenia oraz znaczenia rejestrów konfiguracyjnych) od starszych wersji tychże procesorów, przez co uważam, że opracowanie, takie jak powyższe, jest niezbędne. Warto podkreślić, iż nie będzie to jednak rozwiązanie na wskroś uniwersalne, gdyż zależało mi na prostocie rozwiązań programistycznych – lecz w zastosowaniach tak łatwych, jak opisane tutaj, moim zdaniem okazuje się wystarczająco dobre. Podobnie jak poprzednio, stosowny driver (w tym używający przerwań TWI) możemy wygenerować automatycznie z poziomu środowiska Microchip Studio. Jak możecie jednak sami się

```
#define _USE_READ 0   /* Enable pf_read() function */
#define _USE_DIR 0    /* Enable pf_opendir() and pf_readdir() function */
#define _USE_LSEEK 1  /* Enable pf_lseek() function */
#define _USE_WRITE 1  /* Enable pf_write() function */

#define _FS_FAT12 0   /* Enable FAT12 */
#define _FS_FAT16 1   /* Enable FAT16 */
#define _FS_FAT32 1   /* Enable FAT32 */
```

Listing 3. Fragment zawartości pliku pffconf.h – kod pozwalający na włączenie stosownych opcji kompilacji (w ramach modułu Petit FatFs)

```
FRESULT initSdCard(void)
{
    DSTATUS Status;
    FRESULT Result;
    uint8_t Attempts = 1;


    //Najpierw sprawdzamy, czy karta SD jest w ogóle włożona do gniazda
    if(SD_IS_PRESENT)
    {
        //Inicjujemy napęd fizyczny. Jeśli inicjalizacja się nie powiodła, to wyłączamy
        //i włączamy zasilanie karty, przy czym proces ponawiamy max. trzykrotnie
        do
        {
            Status = disk_initialize();     //Status powinien być RES_OK

            if(Status != RES_OK)
            {
                SD_POWER_OFF;
                _delay_ms(10);
                SD_POWER_ON;
                _delay_ms(100);
            }
            else
            {
                //Poprawnie zainicjowano kartę SD, więc zwiększamy prędkośc magistrali SPI
                //-> fclk = 1.66 MHz dla F_CPU = 3.333 MHz
                SPI0.CTRLA = SPI_MASTER_bm|SPI_CLK2X_bm|SPI_PRESC_DIV4_gc|SPI_ENABLE_bm;
            }
        }
        while((Status != RES_OK) && (Attempts++ < 4));

        if(Status != RES_OK) return FR_DISK_ERR;

        //Montujemy wolumin
        Result = pf_mount(&fileSystem);     //Result powinien być FR_OK, jeśli nie jest
                                            //taki to zwracamy numer błędu
        if(Result != FR_OK) return Result;

        //Otwieramy plik
        Result = pf_open("Tracker.txt");    //Result powinien być FR_OK, jeśli nie jest
                                            //taki to zwracamy numer błędu
    if(Result != FR_OK) return Result; else pf_lseek(fileOffset);

        return FR_OK;
    }
    else return FR_NOT_READY;               //Karta nie jest włożona do gniazda
}
```

Listing 4. Funkcja inicjalizująca kartę SD wraz z niezbędnym „montażem" woluminu i obsługą systemu plików

przekonać, rozwiązanie proponowane przez producenta jest bardzo skomplikowane – jeśli bowiem ma pozostać w pełni uniwersalne, musi operować na dość wysokim poziomie abstrakcji. Zacznijmy od pliku nagłówkowego, którego ciało pokazano na **listingu 5.**

Dalej, na **listingu 6** zaprezentowano ciało funkcji inicjalizującej sprzęg TWI mikrokontrolera AVR 0-series.

Na **listingu 7** pokazano z kolei ciało funkcji odpowiedzialnej za wygenerowanie sygnału START interfejsu I²C i przesłanie adresu klienta (wraz z bitem kierunku R/W), zaś na **listingu 8** – ciało funkcji odpowiedzialnej za wygenerowanie sygnału STOP interfejsu I²C.

I na koniec dwie kluczowe funkcje, pozwalające na zapis i odczyt bajtu poprzez interfejs I²C, których ciała pokazano odpowiednio na **listingach 9** i **10.**

Tyle – w telegraficznym skrócie – w kwestiach implementacyjnych. Przejdźmy zatem do schematu montażowego naszego urządzenia, zaprezentowanego na **rysunku 3**.

Jak widać, zaprojektowano bardzo zgrabną, dwustronną, niewielką płytkę drukowaną o nietypowym wyglądzie. Swoim kształtem dopasowana jest ona do obudowy opisanej

eprasa.pl 65e3eb482b

```
//Definicje portów magistrali I²C
#define I2C_PORT_NAME PORTB
#define I2C_SDA_PINCTRL_REG PIN1CTRL     //PB1
#define I2C_SCL_PINCTRL_REG PIN0CTRL     //PB0

//Definicje częstotliwości magistrali oraz czasu narastania zboczy
#define I2C_FREQUENCY 400000UL           //Częstotliwość magistrali [Hz]
#define I2C_RISE_TIME 100UL              //Czas narastania zboczy zależny od impedancji magistrali [ns]
#define I2C_BAUD (uint8_t)((((((float) F_CPU/(float) I2C_FREQUENCY)) - 10 - ((float) F_CPU * I2C_RISE_TIME/1000000)))/2)

//Definicje bitu potwierdzenia (ACK)
#define NACK 0
#define ACK 1

//Definicje statusów wykonania funkcji
#define OK 0
#define FAILED 255
```

Listing 5. Plik nagłówkowy modułu obsługi interfejsu I²C mikrokontrolerów AVR 0-series

w dalszej części artykułu. Warto również podkreślić, że dla zminimalizowania rozmiaru obwodu drukowanego, przewidziano tutaj montaż elementów po obu stronach laminatu. Lutowanie urządzenia rozpoczynamy od warstwy TOP, na której w pierwszej kolejności umieszczamy wszystkie półprzewodniki. Proces ten najłatwiej wykonać przy użyciu stacji lutowniczej na gorące powietrze i odpowiednich stopów lutowniczych. Jeśli jednak nie dysponujemy tego rodzaju sprzętem, można również zastosować inną metodę. Najprostszym sposobem montażu elementów o tak dużym zagęszczeniu wyprowadzeń, niewymagającym jednocześnie posiadania specjalistycznego sprzętu, jest użycie zwykłej stacji lutowniczej, dobrej jakości cyny z odpowiednią ilością topnika oraz dość cienkiej plecionki lutowniczej, która umożliwi usunięcie nadmiaru cyny spomiędzy wyprowadzeń układów. Należy przy tym uważać, by nie uszkodzić termicznie tego rodzaju elementów. Następnie lutujemy elementy bierne, gniazdo karty micro

```
void i2cInit(void)
{
    //Podciągnięcie portów SDA i SCL pod VCC
    I2C_PORT_NAME.I2C_SDA_PINCTRL_REG = PORT_PULLUPEN_bm;
    I2C_PORT_NAME.I2C_SCL_PINCTRL_REG = PORT_PULLUPEN_bm;

    TWI0.MBAUD = I2C_BAUD;                  //Ustawienie częstotliwości magistrali
    TWI0.MCTRLA = TWI_ENABLE_bm;            //Włączenie modułu I²C -> Tryb Master bez
                                            //przerwań (tzw. pooling)
    TWI0.MSTATUS = TWI_BUSSTATE_IDLE_gc;    //Domyślny tryb magistrali (Idle)
}
```

Listing 6. Ciało funkcji inicjalizującej sprzęg TWI mikrokontrolera AVR 0-series

```
uint8_t i2cStart(uint8_t Address)
{
    //Wysyłamy sygnał START z adresem klienta. Adres zawiera w sobie bit R/W (bit 0) decydujący o kierunku transmisji
    TWI0.MADDR = Address;

    while (!(TWI0.MSTATUS & (TWI_WIF_bm | TWI_RIF_bm)));   //Czekamy na zakończenie operacji

    //Jeśli doszło do błędu lub arbitrażu na magistrali i dostęp do niej został utracony, zwracamy FAILED i czekamy na zwolnienie magistrali
    if(TWI0.MSTATUS & TWI_ARBLOST_bm)
    {
        while(!(TWI0.MSTATUS & TWI_BUSSTATE_IDLE_gc));      //Czekamy na zwolnienie magistrali
        return FAILED;
    }
    //Jeśli klient nie potwierdził swojego adresu, wysyłamy sygnał STOP i czekamy na zwolnienie magistrali
    else if(TWI0.MSTATUS & TWI_RXACK_bm)
    {
        TWI0.MCTRLB |= TWI_MCMD_STOP_gc;                    //Wysyłamy sygnał STOP
    while(!(TWI0.MSTATUS & TWI_BUSSTATE_IDLE_gc));          //Czekamy na zwolnienie magistrali
        return FAILED;
    }

    return OK;
}
```

Listing 7. Ciało funkcji odpowiedzialnej za wygenerowanie sygnału START interfejsu I²C i przesłanie adresu klienta

```
void i2cStop(void)
{
    TWI0.MCTRLB |= TWI_MCMD_STOP_gc;                    //Wysyłamy sygnał STOP
    while(!(TWI0.MSTATUS & TWI_BUSSTATE_IDLE_gc));      //Czekamy na zwolnienie magistrali
}
```

Listing 8. Ciało funkcji odpowiedzialnej za wygenerowanie sygnału STOP interfejsu I²C

```
uint8_t i2cWriteByte(uint8_t Byte)
{
    TWI0.MDATA = Byte;
    TWI0.MCTRLB = TWI_MCMD_RECVTRANS_gc;        //Wysyłamy bajt danych
    while(!(TWI0.MSTATUS & TWI_WIF_bm))         //Czekamy na zakończenie transmisji

    //Jeśli doszło do błędu lub arbitrażu na magistrali i dostęp do niej został utracony
    //zwracamy FAILED
    if(TWI0.MSTATUS & (TWI_ARBLOST_bm | TWI_BUSERR_bm)) return FAILED;

    return!(TWI0.MSTATUS & TWI_RXACK_bm);       //Zwracamy bit ACK
}
```

Listing 9. Ciało funkcji odpowiedzialnej za zapis bajtu poprzez interfejs I²C

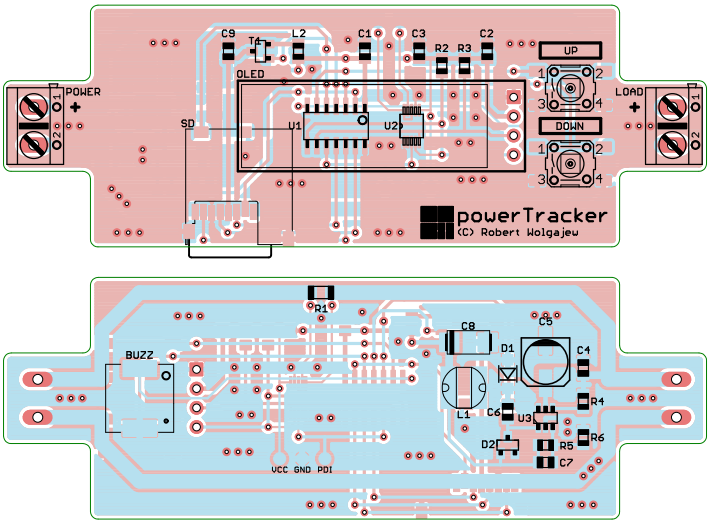
```
uint8_t i2cReadByte(uint8_t Ack)
{
    uint8_t Byte;

    while(!(TWI0.MSTATUS & TWI_RIF_bm));            //Czekamy na zakończenie odczytu
    Byte = TWI0.MDATA;
    //Wysyłamy sygnał ACK, gdy oczekujemy więcej danych (i wznawiamy transmisję) lub sygnał NACK
    if(Ack) TWI0.MCTRLB = TWI_MCMD_RECVTRANS_gc|TWI_ACKACT_ACK_gc; else TWI0.MCTRLB = TWI_ACKACT_NACK_gc;

    return Byte;
}
```

Listing 10. Ciało funkcji odpowiedzialnej za odczyt bajtu poprzez interfejs I²C

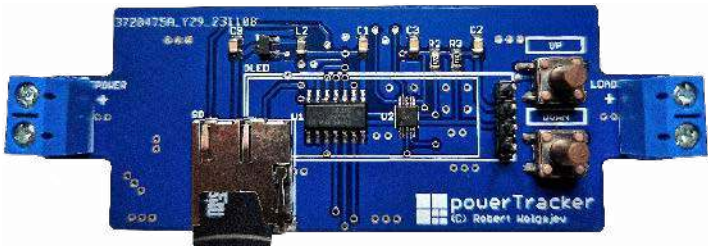
eprasa.pl 65e3eb482b

SD, przyciski UP i DOWN oraz złącza ARK o oznaczeniach POWER i LOAD, które docelowo będą wystawać poza obrys obudowy. Na samym końcu, do tak przygotowanej płytki, instalujemy wyświetlacz OLED, zwyczajnie wlutowując wyprowadzenia jego złącza typu goldpin w przeznaczone do tego celu pola lutownicze (należy sprawdzić polaryzację zasilania), gdyż połączenia elektryczne zapewniają mu jednocześnie wystarczająco stabilny montaż mechaniczny. Wyświetlacz ten mocujemy w taki sposób, by górna płaszczyzna szkła wyświetlacza była umieszczona w odległości 5 mm od górnej powierzchni obwodu drukowanego (musimy przy tym zadbać, aby końcówki złącza goldpin nie wystawały ponad płaszczyznę szkła). Najłatwiej dokonać takiego montażu, przycinając wyprowadzenia złącza goldpin na wysokość 5 mm od płaszczyzny obwodu drukowanego. W tym momencie przechodzimy na warstwę BOTTOM, gdzie – jak poprzednio – w pierwszej kolejności przylutowujemy półprzewodniki, następnie elementy bierne, a na końcu buzzer piezoelektryczny SMD. Poprawnie zmontowany układ powinien działać tuż po włączeniu zasilania. Na **fotografii 1** pokazano zmontowane urządzenie powerTracker od strony warstwy TOP, tuż przed przylutowaniem wyświetlacza OLED, zaś na **fotografii 2** – to samo urządzenie od strony warstwy BOTTOM.

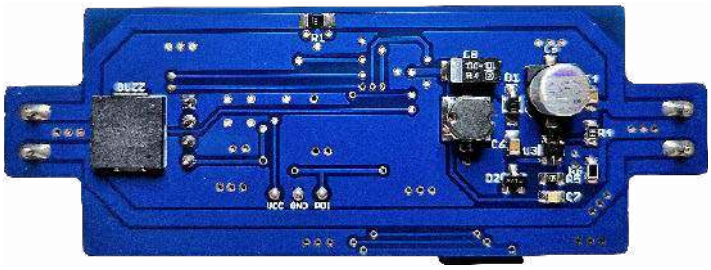
Warto podkreślić, że ścieżki przewodzące duże prądy (pomiędzy gniazdami POWER i LOAD) zaprojektowano w taki sposób, by możliwy był przepływ przez nie prądu stałego o wartości do 10 A. Co również istotne, zastosowany rodzaj bocznika pozwala na rozproszenie mocy rzędu 1 W, co spełnia założenia aplikacji. Jeżeli bocznik nagrzewałby się do nieakceptowalnej temperatury, należy zastosować element o większej dopuszczalnej mocy strat – bądź dwa elementy (o odpowiedniej rezystancji) połączone szeregowo lub równolegle. Przejdźmy teraz do zagadnień związanych z obsługą urządzenia.



Rysunek 3. Schemat montażowy urządzenia powerTracker



Fotografia 1. Zmontowane urządzenie powerTracker, widok od strony warstwy TOP tuż przed przylutowaniem wyświetlacza OLED



Fotografia 2. Zmontowane urządzenie powerTracker, widok od strony warstwy BOTTOM

Obsługa

Projektując interfejs użytkownika modułu powerTracker, kierowałem się zarówno zasadą maksymalnego uproszczenia sposobu obsługi układu, jak i chęcią wyposażenia go w odpowiednią paletę możliwości. W realizacji tego celu posłużyłem się niewielkim, lecz bardzo atrakcyjnym wyświetlaczem OLED o rozdzielczości 128x32 px oraz dwoma mikroprzełącznikami oznaczonymi umownie jako UP i DOWN. Program obsługi aplikacji rozróżnia krótkie i długie naciśnięcie każdego z przycisków. Zgodnie z tymi założeniami powstał bardzo czytelny, graficzny interfejs użytkownika, w ramach którego wyświetlane są wyłącznie 2 ekrany systemu menu o następujących funkcjonalnościach:

- ekran główny wskazujący wartość napięcia na zaciskach odbiornika, prądu pobieranego przez odbiornik, mocy i ładunku dostarczanego do odbiornika oraz ikonę prezentującą aktywność funkcji rejestracji pomiarów na karcie SD (migający symbol RECORD),
- ekran ustawień, w ramach którego dokonujemy ustawień funkcji alarmowania (w tym typu alarmu oraz wartości progowych).

Warto podkreślić, że w przypadku aktywnej funkcji rejestracji pomiarów na karcie SD próbki rejestrowane są 5 razy na sekundę – a po przekroczeniu wielkości pliku następuje wyzerowanie wskaźnika zapisu, przez co nowe pomiary nadpisują stare wartości. W związku z powyższym warto przygotować odpowiednio duży plik danych wypełniony, dla przykładu, samymi spacjami. Zapisywane wartości zgrupowane są w wiersze (zakończone znakami CR i LF) oraz rozdzielone przecinkami tak, jak ma to miejsce w plikach typu CSV. Wspomniany plik, który należy przygotować wcześniej, musi znajdować się w głównym katalogu karty i musi nosić nazwę „Tracker.txt". Wygląd wszystkich ekranów systemu menu pokazano na **rysunku 4**.

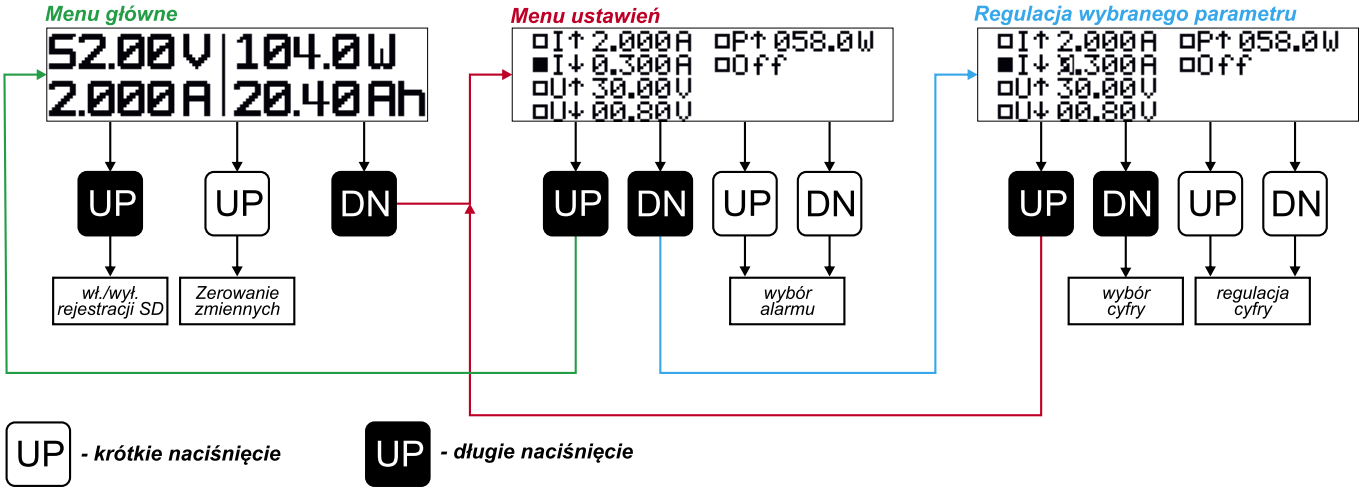
Krótkie wciśnięcie przycisku UP powoduje wyzerowanie wskazania ładunku dostarczonego do odbiornika, zaś długie wciśnięcie tego przycisku – włączenie/wyłączenie procesu rejestracji danych na karcie SD, co sygnalizowane jest miganiem symbolu RECORD na ekranie głównym urządzenia. Włączeniu

Rysunek 4. Wygląd wszystkich ekranów systemu menu urządzenia Tracker

eprasa.pl 65e3eb482b

Tabela 1. Lista wszystkich możliwych błędów zgłaszanych przez urządzenie powerTracker wraz z opisem ich znaczenia dla systemu mikroprocesorowego

Tekst błędu	Opis	Sugerowane działanie
SD disk error	Błąd sprzętowy obsługi karty SD lub nieprawidłowa struktura systemu plików	Sprawdź połączenia gniazda karty SD z systemem mikroprocesorowym. Sprawdź (w komputerze PC) strukturę systemu plików
SD not ready	Brak karty SD w gnieździe karty lub karta SD nie odpowiada na wysyłane komendy	Włóż kartę SD do gniazda lub zmień egzemplarz karty SD
No file	Brak pliku Tracker.txt w katalogu głównym karty SD	Utwórz plik Tracker.txt w katalogu głównym karty SD. Plik musi mieć odpowiednią do oczekiwanej pojemności wielkość pliku
No file system	Brak lub nieprawidłowy system plików na karcie SD	Sformatuj kartę SD zgodnie z systemem plików FAT16/FAT32



Rysunek 5. Diagram prezentujący sposób obsługi urządzenia powerTracker

procesu rejestracji danych na karcie SD towarzyszy inicjalizacja tejże karty i montaż systemu plików – ten ostatni może z różnych względów zakończyć się niepowodzeniem (np. jeśli karta micro SD nie znajduje się w złączu karty lub plik danych „Tracker.txt" nie istnieje). W takim wypadku zostanie wyświetlony stosowny komunikat błędu, zaś obsługa karty nie będzie możliwa. Listę wszystkich możliwych błędów zgłaszanych przez urządzenie powerTracker, wraz z opisem ich znaczenia dla systemu mikroprocesorowego, pokazano w **tabeli 1**.

Kontynuując zagadnienie obsługi systemu powerTracker, warto dodać, że długie przyciśnięcie przycisku DOWN na ekranie głównym powoduje z kolei przejście urządzenia do ekranu ustawień alarmowania. Powyższe zasady to wyłącznie drobny wycinek funkcjonalności przycisków sterujących, gdyż ich funkcja w procesie obsługi urządzenia (wliczając w to rozróżnianie krótkiego bądź długiego przyciśnięcia) determinowana jest miejscem w systemie menu, w jakim znajduje się aktualnie urządzenie. Diagram prezentujący kompletny sposób obsługi urządzenia powerTracker pokazano na **rysunku 5**.

Na koniec warto podkreślić, że aktywna funkcja alarmu powoduje cykliczne sygnalizowanie (dźwiękiem wbudowanego buzzera) faktu przekroczenia ustawionego progu alarmowania (poniżej lub powyżej ustawionej wartości) do czasu ustąpienia warunków alarmu.

Obudowa

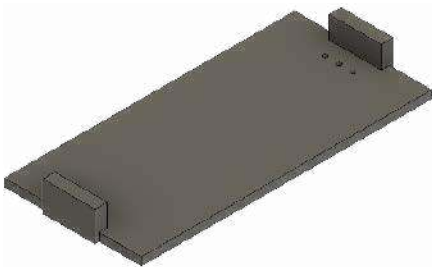
I na sam koniec obiecany „ekstras" – mianowicie gotowy panel obudowy, opracowany w aplikacji do projektowania 3D przez mojego niezawodnego kolegę, **Bartłomieja Wawrzyszko**, zajmującego się hobbystycznie projektami tego rodzaju. Tak oto powstał model obudowy, którego widok 3D pokazano na **rysunku 6**.

Wspomniana obudowa tak naprawdę składa się z dwóch elementów: części górnej (pokazanej na rysunku 6), w której umieszczono otwory na elementy interfejsu użytkownika (okienko wyświetlacza OLED i przyciski funkcyjne) oraz otwór (w bocznej ściance)

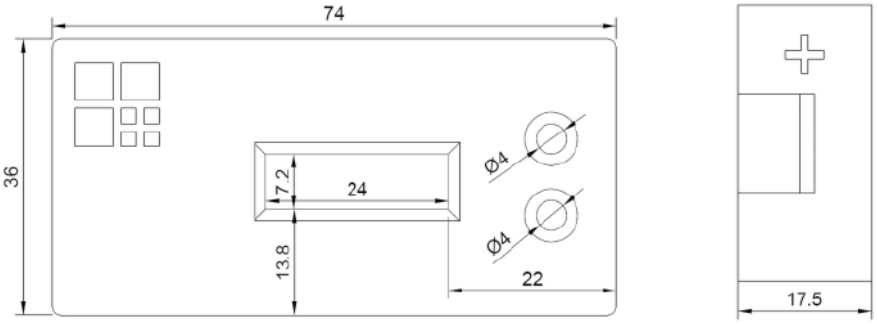
Rysunek 6. Widok 3D obudowy urządzenia powerTracker

REKLAMA

eprasa.pl 65e3eb482b



Rysunek 7. Widok 3D klapki stanowiącej drugi element obudowy urządzenia powerTracker



Rysunek 8. Widok 2D obudowy urządzenia powerTracker z zaznaczeniem wymiarów kluczowych krawędzi

na gniazdo karty SD – a także części dolnej pełniącej funkcję klapki, za pomocą której zamykamy obudowę od dołu. Wygląd 3D (od wewnętrznej strony obudowy) klapki, stanowiącej drugi element obudowy, pokazano na **rysunku 7**.

Z kolei **rysunku 8** pokazano widok 2D wspomnianej obudowy z zaznaczeniem wymiarów kluczowych krawędzi.

Stosowne pliki projektu obudowy (do wydrukowania na drukarce 3D) znajdują się w powiązanych z artykułem materiałach dodatkowych, dostępnych na stronie ep.com.pl. Dodatkowo podpowiem, że – jeśli nie dysponujecie odpowiednim urządzeniem umożliwiającym wydrukowanie obudowy według załączonych plików – z powodzeniem możecie zlecić takie zadanie firmie świadczącej usługi druku 3D. Wybierając tego rodzaju rozwiązanie, decydujcie się na wydruk w technologii MJF (*Multi Jet Fusion*) lub STS (*Selective Laser Sintering*). Pierwsza z nich polega na druku 3D ze sproszkowanych tworzyw sztucznych (poliamidów), poprzez selektywne natryskiwanie na nie lepiszcza (które skleja ze sobą poszczególne warstwy modelu), a następnie zgrzewanie w wysokiej temperaturze, co powoduje ich trwałe zespojenie się. Atutem wydruków 3D tego rodzaju jest wysoka wytrzymałość mechaniczna produkowanych części. Uzyskuje się ją dzięki jednolitej strukturze, która ma 100-procentowe wypełnienie. Druga z polecanych przeze mnie metod, SLS, to przyrostowa technika druku przestrzennego, polegająca na warstwowym spiekaniu laserem proszku polimerowego, który stopniowo tworzy trójwymiarowy wydruk. Detale realizowane w tej technologii charakteryzują się bardzo dobrym przyleganiem warstw i świetnymi właściwościami mechanicznymi, szczególnie odpornością na zgniatanie czy ścieranie. Ponadto cechują się one ogromną dokładnością, dlatego można wykorzystywać je nawet w bardzo precyzyjnych zastosowaniach. Szczerze polecam tego rodzaju rozwiązanie, dodając, że zamieszczona na fotografii tytułowej obudowa została wydrukowana właśnie w tej technologii (STS), a koszt całej usługi (wydruku 2 elementów obudowy) wyniósł około 5 $. Prawda, że tanio? I zupełnie na sam koniec słowo komentarza na temat programowania mikrokontrolera – należy wykonać je przy użyciu nowoczesnego interfejsu programowania UPDI, w który wyposażono nasz układ. Istnieją co najmniej 2 sposoby realizacji tego zadania: albo przy użyciu programatora firmy Microchip o nazwie MPLAB SNAP – albo programatora UPDI mojego autorstwa o nazwie sUPDI, który był tematem artykułu z numeru 4/2021 naszego miesięcznika. Co ciekawe, jest to możliwe bez żadnych specjalnych przygotowań mikrokontrolera, gdyż wyprowadzenie UPDI/RESET układu standardowo ustawione jest za pomocą bitów konfiguracyjnych (fuse-bitów) właśnie jako linia interfejsu UPDI. Niemniej jednak uważny Czytelnik zauważy zapewne, iż w aplikacji urządzenia powerTracker do tego wyprowadzenia podłączone jest jednocześnie wyjście ALERT układu INA226, służące do alarmowania po przekroczeniu wartości progowych, będących przedmiotem pomiaru. O ile takiej funkcjonalności nie wyklucza standardowe ustawienie fuse-bitów mikrokontrolera, gdyż wyprowadzenie w trybie UPDI może być używane (z pewnymi ograniczeniami) jako zwykłe wejście (z wewnętrznym podciąganiem), o tyle sama obecność podłączenia tego pinu z wyjściem ALERT przetwornika INA226 może uniemożliwić programowanie mikrokontrolera (zwłaszcza gdy jest w stanie alarmowania – stanie niskim). Oznacza to, że w przypadku tego rodzaju problemów, wyprowadzenie ALERT układu INA226 należy odłączyć od pinu UPDI podczas programowania mikrokontrolera. Z drugiej strony: mogłoby się również wydawać, że podczas normalnej pracy układu stan wyprowadzenia ALERT wspomnianego przetwornika mógłby doprowadzić do wprowadzenia mikrokontrolera w tryb programowania. Tak, sytuacja taka jest teoretycznie możliwa, lecz nie może mieć miejsca w przypadku naszego urządzenia. Aby wejść w tryb programowania, należy, po spełnieniu pewnych timingów wejściowych, przesłać bajt synchronizacji o wartości 0x55, czego układ INA226 z pewnością wykonać nie może. Osoby szczególnie wrażliwe na tego typu łączenia funkcjonalności mogą, po zaprogramowaniu mikrokontrolera, przestawić (za pomocą zaprogramowania fuse-bitu FUSE.SYSCFG0) funkcję wyprowadzenia UPDI/RESET jako normalnego portu I/O, jednak po wykonaniu takiej operacji wejście w tryb programowania wymagać będzie programatora HV (wysokonapięciowego).

Robert Wołgajew, EP

REKLAMA

Świat projektantów i programistów dla elektroniki w nowej odsłonie. Odwiedź wiecznie młody

ELPORTAL.pl

Znajdziesz nas również na Facebooku: facebook.com/ElportalPL

eprasa.pl 65e3eb482b

AVT sklep

W ofercie AVT*
AVT6028

Podstawowe parametry:
- dwa niezależnie sterowane kanały,
- prąd szczytowy uzwojeń: 3 A/kanał,
- napięcie zasilania silników: 3...30 V (niezależne w obydwu kanałach),
- możliwość sterowania kierunkiem i prędkością obrotów,
- możliwość monitorowania prądu silników,
- konstrukcja na bazie scalonych sterowników pełnomostkowych MAX22202,
- wbudowana przetwornica DC/DC typu buck z opcją wyboru źródła zasilania logiki (napięcie zasilania kanału 1 lub 2).

Dodatkowe materiały do pobrania ze strony www.ulubionykiosk.pl/media

AVT6023	Nakładka Ethernet PoE do AVTDuino (EP 2/2024)
AVT5850	Płytka bazowa dla Arduino Nano Every (EP 3/2021)
AVT5819	Płytka bazowa dla Arduino MKR (EP 11/2020)
AVT5777	Moduł interfejsu ethernet dla Arduino MKR Zero (EP 6/2020)
AVT5738	Wyświetlacz 7-segmentowy dla Arduino MKR Zero (EP 1/2020)
–––––	Sterownik silników prądu stałego dla Arduino MKR (EP 3/2019)
AVT5636	Płytka edukacyjna dla Arduino (EP 5/2018)
AVT1795	AVTduino Battery Shield (EP 3/2014)
AVT1722	AVTduino miniLCD – miniaturowy panel operatora dla Arduino (EP 1/2013)
AVT1686	AVTRelDuino Shield. Moduł wykonawczy dla Arduino (EP 8/2012)

* **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]** – płytka drukowana bez elementów i dokumentacji.

Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- **wersja [A+]** – płytka drukowana **[A]** + zaprogramowany układ **[UK]** i dokumentacja,
- **wersja [UK]** – zaprogramowany układ.

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik PDF! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! **http://sklep.avt.pl**

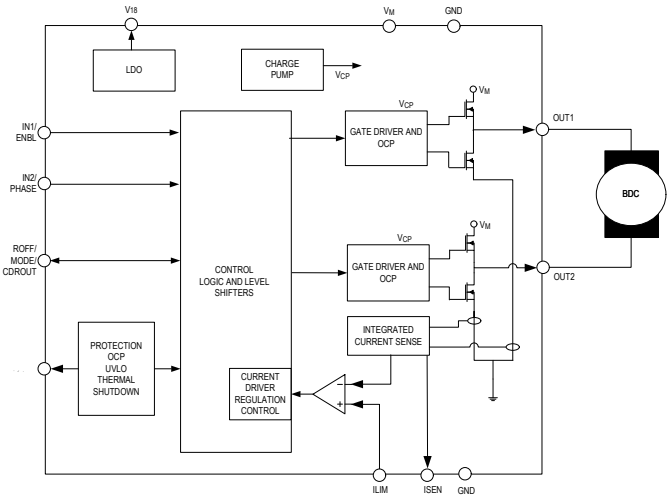
W przypadku braku dostępności na stronie sklepu osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: **kity@avt.pl**.

Sterownik silników do AVTDuino UNO R4

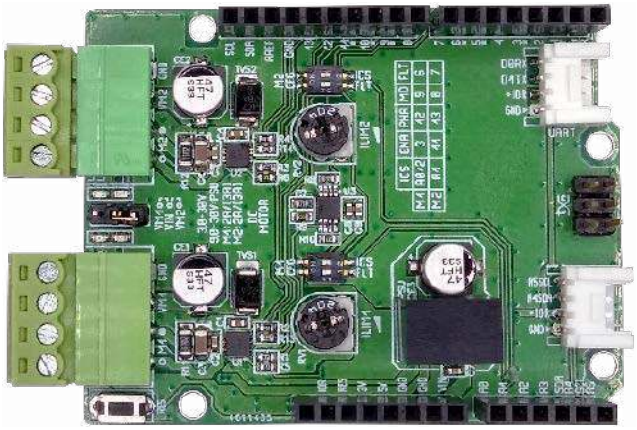
Przedstawiona konstrukcja kontynuuje cykl odświeżający nakładki Arduino, pozwalający na korzystanie z całego potencjału wersji R4 popularnej płytki bazowej. Opisany moduł zastępuje leciwe sterowniki silników oparte na układach L293 oraz L298 nowoczesnymi sterownikami MAX22202, zachowując możliwie wysoki stopień kompatybilności wstecznej.

Nakładka umożliwia dwukierunkowe sterowanie sygnałem PWM dwóch silników prądu stałego z niezależną regulacją prędkości. Prąd szczytowy uzwojeń może sięgnąć maksymalnie 3 A przy zasilaniu każdego silników z osobnego źródła napięcia 3...30 VDC. Wbudowana opcjonalna przetwornica obniżająca ułatwia zasilanie płytki bazowej, gdy napięcie jednego z silników mieści się w zakresie 9...30 V. Zastosowane układy sterowników zapewniają płynną regulację prądu uzwojeń, z wartością ustalaną potencjometrem montażowym; mają ponadto funkcję monitorowania prądów uzwojeń oraz sygnalizację stanów awaryjnych.

Nakładka wyposażona jest w dwa nowoczesne, pełnomostkowe układy sterownika silnika prądu stałego typu MAX22202 (Analog Devices), którego budowę przedstawiono na **rysunku 1**.



Rysunek 1. Budowa wewnętrzna MAX22202 (za notą Analog Devices)



Układ MAX22202 zawiera wszystkie bloki niezbędne do sterowania silnikiem prądu stałego, a jego aplikacja ogranicza się do zaledwie kilku elementów zewnętrznych. Wbudowany mostek MOSFET charakteryzuje się niewielką rezystancją w stanie przewodzenia $R_{ds(on)}$~0,3 Ω, co zmniejsza straty i umożliwia pracę przy niskich napięciach zasilania, już od 3 V, bez zmniejszenia wydajności prądowej. MAX22202 ma wbudowany układ pomiaru prądu silnika, niewymagający zewnętrznego rezystora pomiarowego – dzięki temu jego aplikacja jest uproszczona, a straty mocy zminimalizowane. Układ pracuje w dwóch trybach: pierwszy to włączona regulacja prądu silnika, przy którym maksymalny prąd uzwojeń ustalany jest rezystancją podłączoną do wprowadzenia ILIM, drugi to praca z wyłączoną regulacją, gdy do wyprowadzenia ILIM podłączony jest rezystor 14 kΩ, a aktywne pozostaje tylko ograniczenie prądu

REKLAMA

LASEROWE
SZABLONY
DO MONTAŻU SMT

Materiał: stal nierdzewna CrNi
Zakres grubości blach: 0,020–1,000 mm
Wycinamy również detale
o dowolnych kształtach

LASTENIC
LASER STENCILS

LASTENIC LASER & ELECTRONICS sp. z o.o.
58-100 Świdnica, ul. Husarska 5
tel. 74 851 48 77, 697 977 732
www.lastenic.com info@lastenic.com

eprasa.pl 65e3eb482b

maksymalnego 3,5 A. Regulacja prądu silnika ułatwia np. rozruch napędu bez przeciążeń prądowych i zbędnych obciążeń mechanicznych. W każdym trybie możliwa jest regulacja PWM. Wartość rezystora podłączonego do wyprowadzenia ILIM ustalającego prąd silnika I_{trip} wyznacza się ze wzoru:

$$I_{trip}\ [A] = K_{ilim}/R_{lim}\ [k\Omega]$$

gdzie K_{ilim}=~50 KV (R_{lim}: 18...250 kΩ)

Jeżeli w danej aplikacji przewidywane jest sterowanie silnika w zamkniętej pętli sprzężenia zwrotnego lub konieczna jest informacja o prądzie uzwojeń, to wyprowadzenie ISEN udostępnia potrzebny sygnał monitorujący. Podłączony do niego rezystor obciąża wbudowane źródło prądowe, a uzyskiwane napięcie wyjściowe opisane jest wzorem:

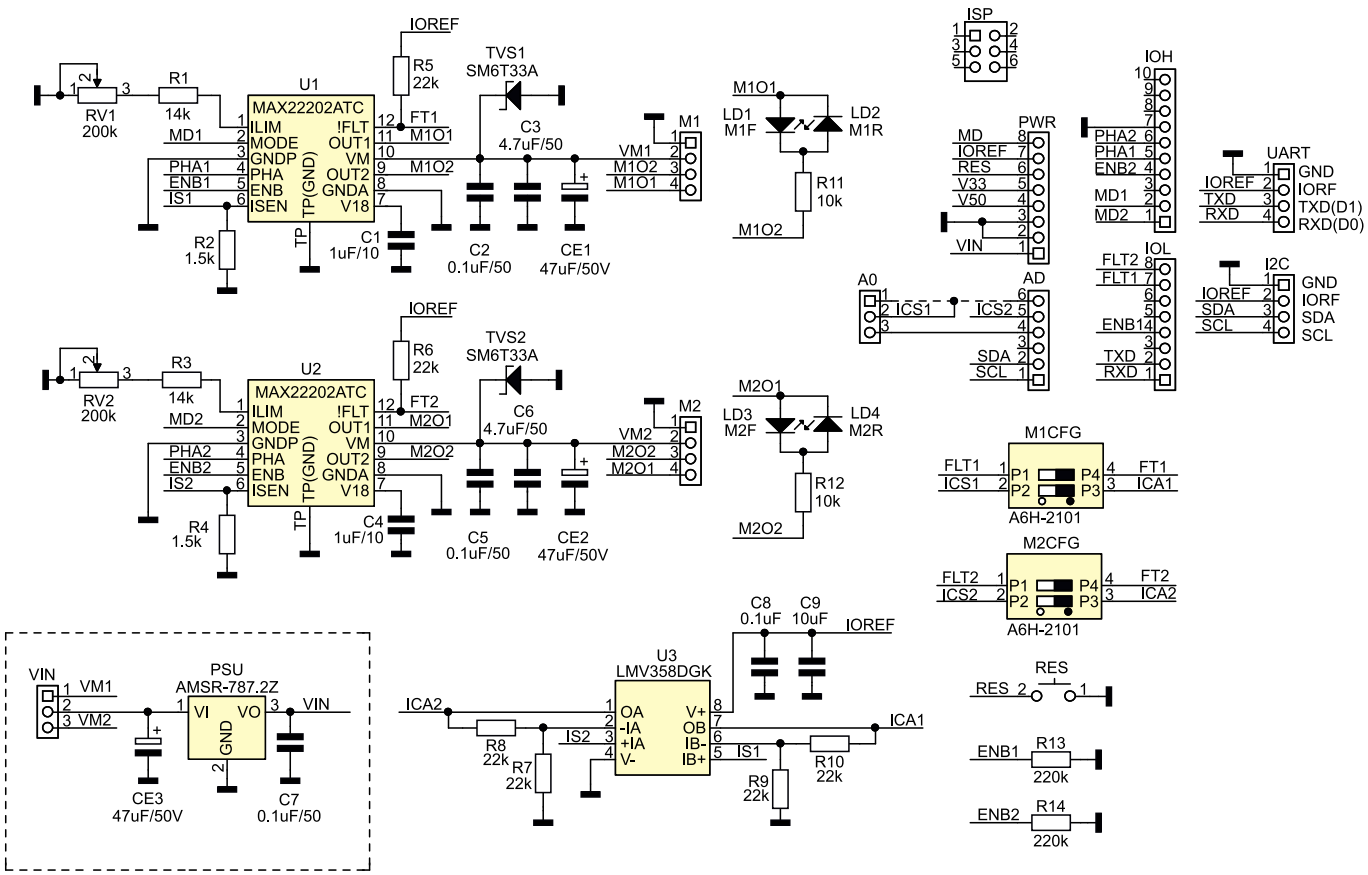
$$R_{isen}\ [\Omega] = K_{isen} \cdot V_{fs}\ [V]/I_{max}\ [A]$$

gdzie K_{isen}=~5 kA/A

Stany awaryjne – takie jak przeciążenie, przegrzanie lub zbyt niskie napięcie zasilania – sygnalizowane są stanem niskim na wyprowadzeniu FAULT.

Za sterowanie silnikiem odpowiadają sygnały ENABLE, PHASE(DIR) oraz MODE, zgodnie z **tabelą 1**. Jeżeli stan niski na wyprowadzeniu MODE trwa dłużej niż (1,5...3,5 ms), układ przechodzi w stan obniżonego poboru mocy. W przypadku włączonej regulacji prądu silnika i przekroczenia jej nastawy, sterowanie mostkiem przełączane jest na wewnętrzne, ograniczające prąd uzwojenia niezależnie od stanu wyprowadzeń.

Schemat nakładki przedstawiono na **rysunku 2**. Układy U1, U2 (MAX22202) pracują w typowej aplikacji. Każdy z silników podłączony jest wraz z napięciem zasilania do odpowiadającego mu złącza M1, M2. Odsprzęganie zasilania zapewniają kondensatory C2, C3, CE1 i C5, C6, CE2, zaś transile TVS1, TVS2 zabezpieczają drivery przed skutkami przepięć. Pojemności filtrujące mają minimalne wartości, niezbędne do zapewnienia poprawnej pracy U1, U2 (każdorazowo należy zapewnić odpowiednią filtrację i przekroje przewodów zasilających dobranych pod kątem wymagań współpracujących silników). Regulacja prądu silnika odbywa się za pomocą potencjometrów RV1, RV2. Ustawienie ślizgaczy w maksymalnym położeniu wyłącza regulację, ustalając rezystancję widzianą przez wyprowadzenie ILIM na R1, R3=14 kΩ. Sygnał monitorujący prąd silnika, dostępny na wyprowadzeniu ILIM, przetwarzany jest na napięcie i dodatkowo buforowany przez wzmacniacz U3 o wzmocnieniu ustalonym na 2 V/V. Maksymalny mierzony prąd to 3,5 A, co odpowiada napięciu 2 V na wyprowadzeniach A0 lub A2 i A1. Zakres prądu i napięcia może zostać zoptymalizowany poprzez dobór rezystorów R2, R10 i R4, R8 odpowiednio do każdego z silników – należy pamiętać tylko, by nie przekraczać zakresu napięcia wejściowego ADC (zależnego od użytej płytki bazowej). Bezpiecznie jest pracować w zakresie 0...3 V, ponieważ gwarantuje



Rysunek 2. Schemat nakładki

Wykaz elementów:

Rezystory: (1%)
R1, R3: 14 kΩ (SMD 0603)
R2, R4: 1,5 kΩ (SMD 0603)
R5...R10: 22 kΩ (SMD 0603)
R11, R12: 10 kΩ (SMD 1206)
R13, R14: 220 kΩ (SMD 0603)

Kondensatory:
C1, C4: 1 μF/10 V (SMD 0603, X7R)
C2, C5, C7: 100 nF/50 V (SMD 0603, X7R)
C3, C6: 4,7 μF/50 V (SMD 1206, X7R)
C8: 100 nF/50 V (SMD 0603, X7R)
C9: 10 μF/10 V (SMD 0603, X7R)
CE1...CE3: 47 μF/50 V elektrolityczny, typ EEEFT1H470AP

Półprzewodniki:
LD1, LD3: dioda LED czerwona (SMD 0603)
LD2, LD4: dioda LED zielona (SMD 0603)
TVS1, TVS2: transil SM6T33A (SMB)
U1, U2: MAX22202ATC (TDFN12)
U3: LMV358DGK (VSSOP8_065)

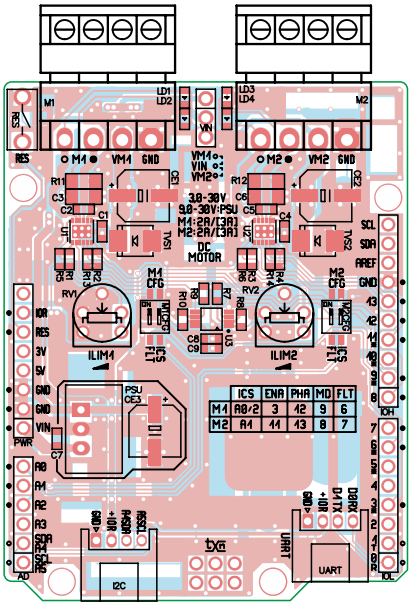
Pozostałe:
AD: złącze szpilkowe 6 pin 2,54 mm (13...15 mm)
I²C, UART: złącze Grove kątowe
IOH: złącze szpilkowe 10 pin, 2,54 mm (13...15 mm)
IOL, PWR: złącze szpilkowe 8 pin, 2,54 mm (13...15 mm)
ISP: złącze szpilkowe 2×3 pin, 2,54 mm (13...15 mm) (opcja)
M1, M2: złącze śrubowe kompletne 4 pin 3,81 mm kątowe
M1CFG, M2CFG: przełącznik DIP switch, 2 sekcje, typ A6H-2101
PSU: przetwornica AMSR-787.2Z (opcja)
VIN: złącze SIP3 + zwora (opcja)
RES: przycisk TACT-switch 6×3 mm
RV1, RV2: potencjometr montażowy 200 kΩ

eprasa.pl 65e3eb482b

Tabela 1. Sterowanie MAX22202 (za notą Analog Devices)

PHASE	ENBL	MODE	OUT1	OUT2	Description
1	1	X	H	L	Forward (current from OUT1 to OUT2)
0	1	X	L	H	Reverse (current from OUT2 to OUT1)
X	0	1	L	L	Brake (slow decay)
1	0	0	L	H	Fast-decay synchronous rectification (*) Sleep mode if following a longer than tSLEEP brake status
0	0	0	H	L	Fast-decay synchronous rectification (*). Sleep mode if following a longer than tSLEEP brake status

zgodność z płytkami zasilanymi napięciami zarówno 3,3 V, jak i 5 V. Zwora A0 (zalutowana domyślnie w położeniu A0) umożliwia zmianę wyprowadzenia monitorującego prąd silnika M1 z A0 na A2, co jest istotne w aplikacjach, w których chcemy wykorzystać wyprowadzony na A0 przetwornik D/A w UNO R4. Sygnał monitorujący prąd silnika nie podlega w układach U1 i U2 żadnej obróbce oprócz skalowania zakresu – o stosowne przeliczenie należy zatem zadbać we własnym zakresie, zwłaszcza jeśli sterujemy silnikiem sygnałem PWM. Kiedy sygnał monitorujący prąd silników nie jest niezbędny w aplikacji, może zostać całkowicie odłączony od wyprowadzeń A0(2), A1 przy użyciu przełączników DIP M1CFG, M2CFG, a linie te zostaną uwolnione do użycia w innych celach. Stan awaryjny driverów: FLT1, FLT2, dostępny jest na wyprowadzeniach D6, D7 i także może zostać od nich odłączony przełącznikami DIP M1CFG, M2CFG. Sterowanie silnikami sprowadza się do trzech sygnałów: ENB1/2 (załączającego silnik z możliwą regulacją PWM), PHA1/2 (odpowiadającego za zmianę kierunku) oraz MD1/2 (odpowiadającego za tryb pracy). Przypisanie sygnałów jest zgodne z Motor Shield. Diody LD1...LD4 – podłączone bezpośrednio na wyjściach OUT układów U1 i U2 – sygnalizują wysterowanie silnika, dzięki czemu możliwa jest też obserwacja napięcia indukowanego w silniku przy braku wysterowania uzwojenia (przejście do pracy prądnicowej) i kontrolowanie rzeczywistego stanu wyjść mostka.



Rysunek 3. Rozmieszczenie elementów na płytce drukowanej

W celu ułatwienia aplikacji nakładki, na złącza Grove wyprowadzono magistralę I²C (wyprowadzenia A4, A5) oraz UART (D0, D1), umożliwiającą podłączenie czujników zasilanych napięciem IOREF. Opcjonalnie układ uzupełnia przetwornica obniżająca PSU typu AMSR-787.2Z, pozwalająca na zasilanie płytki bazowej z jednego z napięć zasilających silniki, doprowadzonych do złącza VIN – położenie współpracującej z nim zwory ustala wybór napięcia VM1 lub VM2. Przetwornica obniża wybrane napięcie do 7,2 V (obciążalność maksymalna 500 mA). Do poprawnej pracy konwertera DC/DC wymagane jest napięcie wejściowe 9...30 V, co nieznacznie ogranicza możliwości driverów, ale upraszcza aplikację w przypadku, gdy podłączamy silniki do typowego napięcia z przedziału 9...24 V, np. z pakietu akumulatorów. Gdy wymagane jest sterowanie silnikami niskonapięciowymi 3...9 V, napięcie do płytki bazowej musi być doprowadzone niezależnie. W przypadku AVTDuino UNO R4 Plus, aby wykorzystać zasilanie z napięcia silników, należy założyć na płytkę bazową zworę VIN; w przypadku innych płytek Arduino trzeba koniecznie sprawdzić dopuszczalny zakres zasilania VIN.

Układ zmontowany jest na dwustronnej płytce drukowanej zgodnej z Arduino Shield Rev3. Rozmieszczenie elementów przedstawiono na **rysunku 3.** Sposób montażu nie wymaga opisu, montaż zwory VIN i przetwornicy PSU jest opcjonalny (zgodnie z uwagami w tekście). W zależności od przewidywanego zastosowania, przedłużane złącza szpilkowe PWR, AD, IOL, IOH, ISP (wysokość 13...15 mm) można zastąpić „stackowalnymi" złączami Arduino, umożliwiającymi montowanie modułów w kanapki.

Prosty szkic z **listingu 1** ułatwia przetestowanie sterowania silnikami przy użyciu PWM i zmiany kierunku. W przypadku zastosowania szkiców z Arduino Motor Shield należy zmodyfikować funkcję obsługującą wyprowadzenie BRAKE, która w AVTDuino UNO_R4_DCMotor Shield pełni funkcję MODE i konieczna jest jej negacja.

```
// Test MAX22202 UNO_R4_DCMotor
int dirPin1 = 12;
int pwmPin1 = 3;
int modePin1 = 9;
int pwm1 = 0;

int dirPin2 = 13;
int pwmPin2 = 11;
int modePin2 = 8;
int pwm2 = 0;

int step = 1;

void setup() {
  //define pins
  pinMode(dirPin1, OUTPUT);
  pinMode(pwmPin1, OUTPUT);
  pinMode(modePin1, OUTPUT);
  pinMode(dirPin2, OUTPUT);
  pinMode(pwmPin2, OUTPUT);
  pinMode(modePin2, OUTPUT);
  digitalWrite(modePin1, LOW);
  digitalWrite(modePin2, LOW);
  digitalWrite(dirPin1, LOW);
  digitalWrite(dirPin2, LOW);
  analogWrite(pwmPin1, 0);
  analogWrite(pwmPin2, 0);
}

void loop() {
  // M1 speed up/down
  digitalWrite(modePin1, HIGH);
  delay(5);
  // M1 >>
  digitalWrite(dirPin1, LOW);
  for (pwm1 = 0; pwm1 <= 255; pwm1++) {
    analogWrite(pwmPin1, pwm1);
    delay(10);
  }
  delay(3000);
  for (pwm1 = 0; pwm1 <= 255; pwm1++) {
    analogWrite(pwmPin1, 255 - pwm1);
    delay(10);
  }
  delay(1000);

  // M1 <<
  digitalWrite(dirPin1, HIGH);
  for (pwm1 = 0; pwm1 <= 255; pwm1++) {
    analogWrite(pwmPin1, pwm1);
    delay(10);
  }
  delay(3000);
  for (pwm1 = 0; pwm1 <= 255; pwm1++) {
    analogWrite(pwmPin1, 255 - pwm1);
    delay(10);
  }
  delay(1000);
  // M1 fast decay
  digitalWrite(modePin1, LOW);
  analogWrite(pwmPin1, 0);

  // M2 speed up/down
  digitalWrite(modePin2, HIGH);
  delay(5);

  // M2 >>
  digitalWrite(dirPin2, LOW);
  for (pwm2 = 0; pwm2 <= 255; pwm2++) {
    analogWrite(pwmPin2, pwm2);
    delay(10);
  }
  delay(3000);
  for (pwm2 = 0; pwm2 <= 255; pwm2++) {
    analogWrite(pwmPin2, 255 - pwm2);
    delay(10);
  }
  delay(1000);

  // M2 <<
  digitalWrite(dirPin2, HIGH);
  for (pwm2 = 0; pwm2 <= 255; pwm2++) {
    analogWrite(pwmPin2, pwm2);
    delay(10);
  }
  delay(3000);
  for (pwm2 = 0; pwm2 <= 255; pwm2++) {
    analogWrite(pwmPin2, 255 - pwm2);
    delay(10);
  }
  delay(1000);
   // M2 fast decay
   digitalWrite(modePin2, LOW);
   analogWrite(pwmPin2, 0);
}
```

Listing 1. Przykładowy szkic testowy

Jeżeli wszystko działa poprawnie, moduł może zostać użyty we własnej aplikacji.

Adam Tatuś, EP

eprasa.pl 65e3eb482b

AVT sklep

W ofercie AVT*

AVT6029

Podstawowe parametry:
- Dwa wyjścia stałonapięciowe: 12 V/2 A oraz 5 V/2,5 A
- Łączna moc wyjściowa: 24 W (ciągła)/30 W (szczytowa)
- Sprawność głównej przetwornicy 12 V: 89%
- Możliwa regulacja napięcia pierwszego stopnia układu w zakresie 10,8 V...12,8 V
- Konstrukcja oparta na module Ag5712
- Zdublowane gniazda RJ45, zapewniające łatwą rozbudowę sieci przy jednoczesnym korzystaniu z funkcjonalności PoE i transmisji LAN
- Wbudowane wskaźniki obecności napięcia na obu wyjściach (2×LED SMD)

Dodatkowe materiały do pobrania ze strony www.ulubionykiosk.pl/media

–––––	Zasilacz do płytek stykowych ze złączem USB-C (EP 10/2023)
AVT5977	Stabilizator napięcia symetrycznego z regulacją współbieżną (EP 3/2023)
AVT5963	Zasilacz warsztatowy, część 1 i 2 (EP 12/2022, 01/2023)
–––––	Modułowy zasilacz warsztatowy (EP 5/2022)
–––––	Regulowany zasilacz warsztatowy – RPS-02 (EP 4/2022)
AVT5915	Zasilacz 5 V/1 A z szerokim zakresem napięć wejściowych (EP 1/2022)
AVT5908	Beztransformatorowy impulsowy zasilacz sieciowy (EP 12/2021)
AVT5872	Regulowany zamiennik stabilizatora 78xx (EP 7/2021)
AVT1990	Regulowany zasilacz do płytek stykowych (EP 8/2018)
–––––	Precyzyjny regulowany zasilacz stabilizowany (EP 2/2018)

* **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]** – płytka drukowana bez elementów i dokumentacji.

Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- **wersja [A+]** – płytka drukowana **[A]** + zaprogramowany układ **[UK]** i dokumentacja,
- **wersja [UK]** – zaprogramowany układ.

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik PDF! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! **http://sklep.avt.pl**

W przypadku braku dostępności na stronie sklepu osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: **kity@avt.pl**.

Zasilacz PoE 5/12 V 30 W

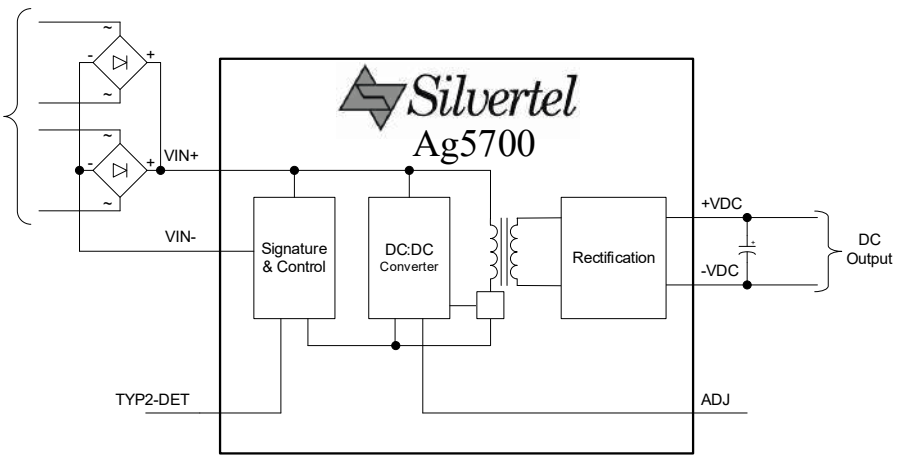
Zasilacz umożliwia zastosowanie standardu 802.3at (PoE+) w celu uzupełnienia lub rozbudowy urządzeń korzystających z interfejsu Ethernet, a niewyposażonych w funkcję zasilania. Dostarcza napięcia 5 V/2,5 A i 12 V/2 A o sumarycznej mocy 24 W (szczytowo 30 W), co poszerza zakres zastosowań i ułatwia dostarczanie energii do urządzeń zawierających logikę 5-woltową oraz obwody sterowane, np. jak przekaźniki zasilane z 12 V lub dyski HDD. Zasilacz wyposażony został w gniazda obejściowe do transmisji Ethernet100M, co rozwiązuje jednocześnie dwa problemy: z zasilaniem i i transmisją danych do urządzenia.

Układ bazuje na module Ag5712 Silvertel, którego budowę wewnętrzną przedstawiono na **rysunku 1**.

Ag5712 pozwala spełnić wymogi IEEE802.3 PoE, przy utrzymaniu pełnej kompatybilności z IEEE802.3bt, IEEE802.3at i IEEE802.3af. Ag5712 jest wstępnie skonfigurowany jako urządzenie typu 2 klasy 4, umożliwia pobieranie do 30 W mocy z PSE, przy nominalnym napięciu wyjściowym 12 V i sprawności do 89%. Wbudowana przetwornica zabezpieczona została przed przeciążeniem oraz zwarciem wyjścia, a także przegrzaniem.

Schemat zasilacza przedstawiono na **rysunku 2**. Zasilanie z routera, switcha lub inżektora PoE+ wraz z sygnałami transmisji doprowadzone jest typowym okablowaniem Cat5 do gniazda ETHPoE. Same przebiegi cyfrowe nie podlegają żadnym modyfikacjom i są bezpośrednio wyprowadzone na gniazdo ETH, które należy połączyć z interfejsem sieciowym urządzenia

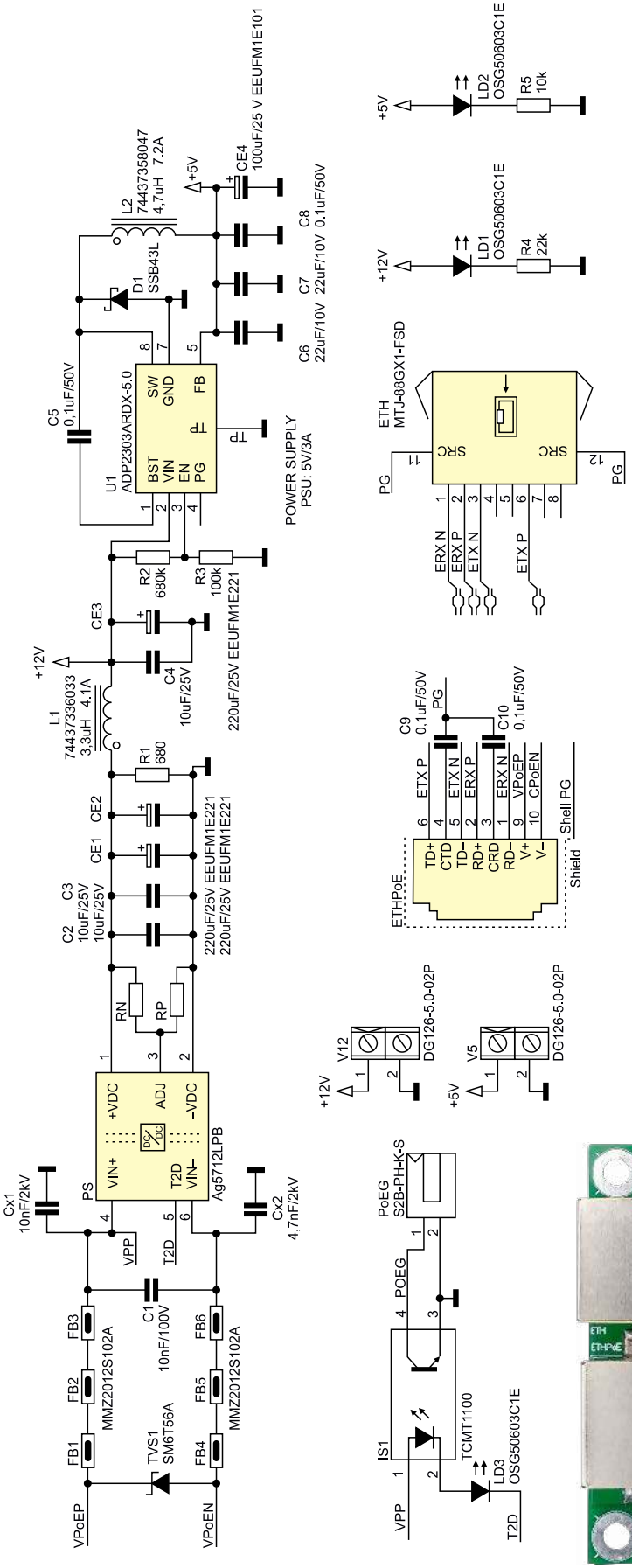
UWAGA: w module przetwornicy występują napięcia do 60 V, należy zachować szczególną ostrożność podczas uruchamiania układu. W celu zwiększenia bezpieczeństwa polecamy dodatkowo zabezpieczyć moduł przekładką lub taśmą izolacyjną.



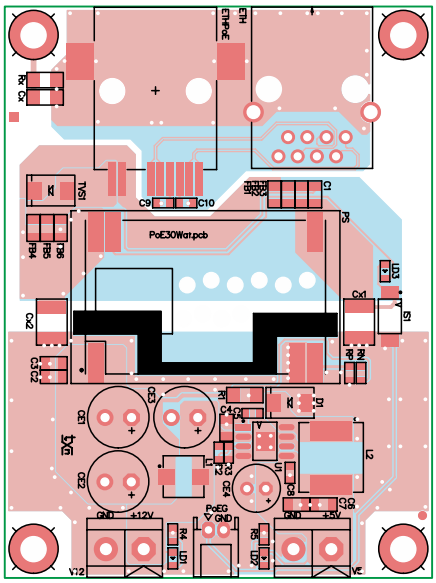
Rysunek 1. Schemat blokowy Ag5712 (za notą Silvertel)

eprasa.pl 65e3eb482b

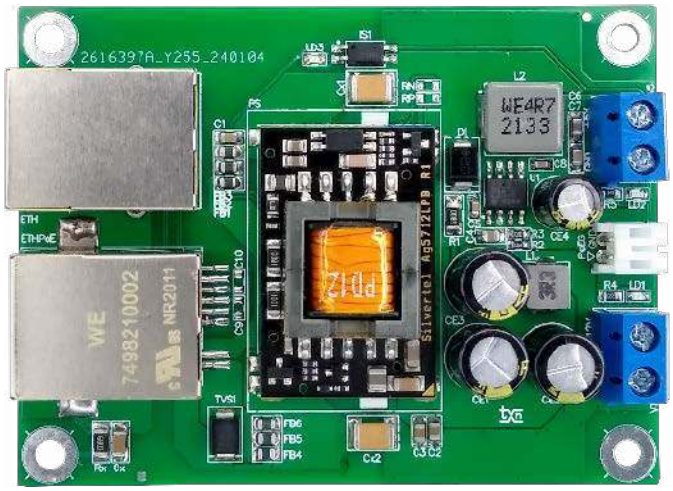
(lub pozostawić niepodłączone, jeżeli używamy tylko trybu zasilania). Modułowe gniazdo ETHPoE typu WE 7498210002 wyposażone jest w transformatory separacyjne dla kanałów transmisji RX/TX oraz dwa mostki prostownicze umożliwiające zastosowanie obu schematów transmisji zasilania, zarówno przez pary sygnałowe, jak i przez pary niewykorzystane w interfejsie 100M. Zasilanie z PoE+ – po zabezpieczeniu przez transil TVS1 i dodatkowej filtracji zakłóceń poprzez FB1...FB6 – doprowadzone zostaje do modułu PS. Wyjście T2D służy do sygnalizacji typu współpracującego PSE. Jeżeli moduł podłączony został do PSE zgodnego z PoE+, T2D ustawiane jest w stan niski (co sygnalizuje tym samym dostępność pełnej mocy modułu). Transoptor IS1 separuje sygnał od potencjału pierwotnej strony przetwornicy, zaś dioda LD3 sygnalizuje zgodność z PoE+. Jeżeli PSE jest niezgodne z PoE+, wówczas wyjście T2D ustawiane zostaje w stan wysoki, jednocześnie informując współpracujący układ o ograniczeniu dostępnej mocy do 12 W. Sygnał POEG dostępny jest na złączu PoEG. Rezystory RN, RP służą do korekcji napięcia wyjściowego przetwornicy, zgodnie z notą aplikacyjną. Wlutowując rezystor RN=68 kΩ, obniżamy napięcie przetwornicy do 10,8 V, natomiast wlutowując zworę w miejsce RP – podnosimy napięcie wyjściowe do 12,8 V.



Rysunek 2. Schemat ideowy zasilacza



Rysunek 3. Rozmieszczenie elementów modułu



Fotografia 1. Zmontowany moduł

eprasa.pl 65e3eb482b

Wykaz elementów:

Rezystory: (1%)
R1, Rx: 680 Ω (SMD 1206)
R2: 680 kΩ (SMD 0603)
R3: 100 kΩ (SMD 0603)
R4: 22 kΩ (SMD 0603)
R5: 10 kΩ (SMD 0603)
RN, RP: opis w tekście (SMD 0603)

Kondensatory:
C1: 10 nF/100 V (SMD 0805, X7R)
C2...C4: 10 µF/25 V (SMD 0805, X7R)
C5, C8...C10: 100 nF/50 V (SMD 0603, X7R)
C6, C7: 22 µF/10 V (SMD 0805, X7R)
CE1...CE3: 220 µF/25 V elektrolityczny (typ EEUFM1E221, low ESR, D=8 mm, P= 3,5 mm)
CE4: 100 µF/25 V elektrolityczny (typ EEUFM1E101, low ESR D=6,3 mm, P= 2,5 mm)
Cx: 2,2 nF/1 kV (SMD 1206, X7R)
Cx1: 10 nF/2 kV (SMD 1812, X7R)
Cx2: 4,7 nF/2 kV (SMD 1812, X7R)

Półprzewodniki:
D1: dioda Schottky'ego 40 V/3 A SSB43L (SMB)
LD1...LD3: dioda LED OSG50603C1E (SMD 0603)
TVS1: transil 600 W SM6T56A (SMB)
IS1: transoptor TCMT1100 (SSOP4)
U1: ADP2303ARDZ-5.0 (SO-8)

Pozostałe:
FB1...FB6: dławik ferrytowy 1000 Ω @ 100 MHz 0,3 Ω 0,5 A, typ MMZ2012S102A (SMD 0805)
L1: dławik ferrytowy 3,3 µH/32 mΩ 4,1 A, typ 74437336033
L2: dławik ferrytowy 4,7 µH/19,7 mΩ 7,2 A, typ 74437358047
ETH: złącze RJ45 8P8C, typ MTJ-88GX1-FSD
ETHPoE: złącze RJ45 PoE + WE-RJ45 LAN PoE + transformator, typ 7498210002
PoEG: złącze JST THT PH 2,00 mm 2-pin, typ S2B-PH-K-S
PS: przetwornica PoE+ Silvertel 12 V/24 W(30 W), typ Ag5712LPB
V5, V12: złącze 5,00 mm 2-pin, typ Degson DG126-5.0-02P

Napięcie wyjściowe 12 V – po dodatkowej filtracji za pomocą elementów CE1...3, L1, C4 – doprowadzone jest do złącza V12, a dioda LD1 sygnalizuje jego obecność na zaciskach wspomnianego gniazda śrubowego. W celu uzyskania napięcia 5 V zastosowana została przetwornica obniżająca U1 typu ADP2303-5, której napięcie wyjściowe dostępne jest na złączu V5, a jego obecność sygnalizuje dioda LD2.

Zasilacz zmontowano na dwustronnej płytce drukowanej, a rozmieszczenie elementów przedstawiono na **rysunku 3**. Montaż nie wymaga opisu, kondensatory C9, C10 nie są lutowane.

Zmontowany moduł prezentuje **fotografia 1**.

Moduł nie wymaga uruchamiania – po podłączeniu do PSE PoE+ należy jedynie sprawdzić obecność napięć 5 V i 12 V sygnalizowanych przez LD1/2, podczas gdy LD3 sygnalizuje zasilanie z PoE+. Należy pamiętać o sumarycznej obciążalności maksymalnej zasilacza wynoszącej 24 W (szczytowo 30 W). W każdym przypadku trzeba zapewnić odpowiednie chłodzenie modułu oraz – przy pracy z obciążeniem zbliżonym do granicznego – zapewnić cyrkulację powietrza. Każdorazowo zalecane jest wykorzystanie sygnału PoEG do ograniczania poboru mocy urządzeniu docelowym, aby zapewnić niezawodną współpracę także z PSE o mniejszej mocy (standard PoE af).

Jeżeli wszystko pracuje poprawnie, moduł można wykorzystać we własnej aplikacji.

Adam Tatuś, EP

AVT sklep

W ofercie AVT*
AVT6030

Podstawowe parametry:
- port wejściowy: gniazdo USB-C,
- dwa porty USB wyprowadzone na złącza szpilkowe 5-pin,
- wbudowany stabilizator 3,3 V i zabezpieczenia TVS,
- zintegrowane klucze na liniach zasilania,
- ograniczenie prądowe 800 mA.

Dodatkowe materiały do pobrania ze strony www.ulubionykiosk.pl/media
----- Dwukanałowy port szeregowy z pełnym interfejsem UART (EP 1/2024)
----- Moduł czterech wyjść HighSide dla RPi Pico (EP 11/2023)
----- Filtr zasilania dla Raspberry Pi (EP 9/2023)
----- Ekspander GPIO RPi z taśmą FPC (EP 8/2023)
----- Sterownik unipolarnego mikrosilnika krokowego dla RPi Pico (EP 7/2023)
----- Sterownik dwóch silników krokowych do Raspberry Pi (EP 6/2023)

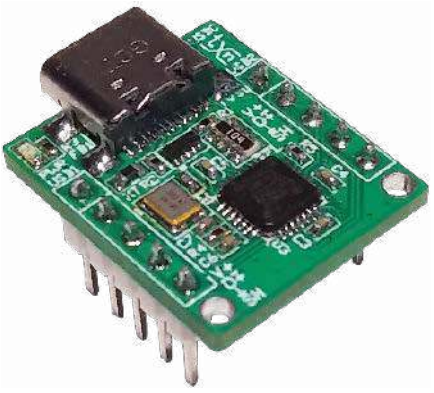
* **Uwaga!** Elektroniczne zestawy do samodzielnego montażu. Wymagana umiejętność lutowania! Podstawową wersją zestawu jest wersja **[B]** nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji **[B]** zawiera elementy elektroniczne (w tym **[UK]** – jeśli występuje w projekcie), które należy samodzielnie 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]** – płytka drukowana bez elementów i dokumentacji.

Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- **wersja [A+]** – płytka drukowana **[A]** + zaprogramowany układ **[UK]** i dokumentacja,
- **wersja [UK]** – zaprogramowany układ.

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik PDF! Podczas składania zamówienia upewnij się, którą wersję zamawiasz!
http://sklep.avt.pl

W przypadku braku dostępności na stronie sklepu osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.

Dwuportowy mikrohub USB-C do systemów wbudowanych



Hub USB-C o niewielkim rozmiarze przyda się przy rozbudowie systemów SBC, w których klasyczny hub i jego okablowanie zajmowałyby zbyt dużo miejsca.

Gdy nasz SBC ma tylko jeden port USB (np. Raspberry Pi Zero), a potrzebujemy ich więcej, oczywistym rozwiązaniem okazuje się zastosowanie huba USB. Ale gdy mamy niewielką ilość przestrzeni, współpracujące urządzenia USB są umieszczone w tej samej obudowie (np. ekran dotykowy, pendrive, karta muzyczna audio lub konwertery interfejsów szeregowych, które dodatkowo sumarycznie nie pobierają zbyt dużego prądu z USB), okazuje się, że ich typowe okablowanie zajmuje więcej miejsca niż cała aplikacja. W tym przypadku proponowane rozwiązanie może wprowadzić nieco porządku. Zmiana gniazd USB na złącza szpilkowe (o typowym dla PC wyprowadzeniu sygnałów)

Wykaz elementów:

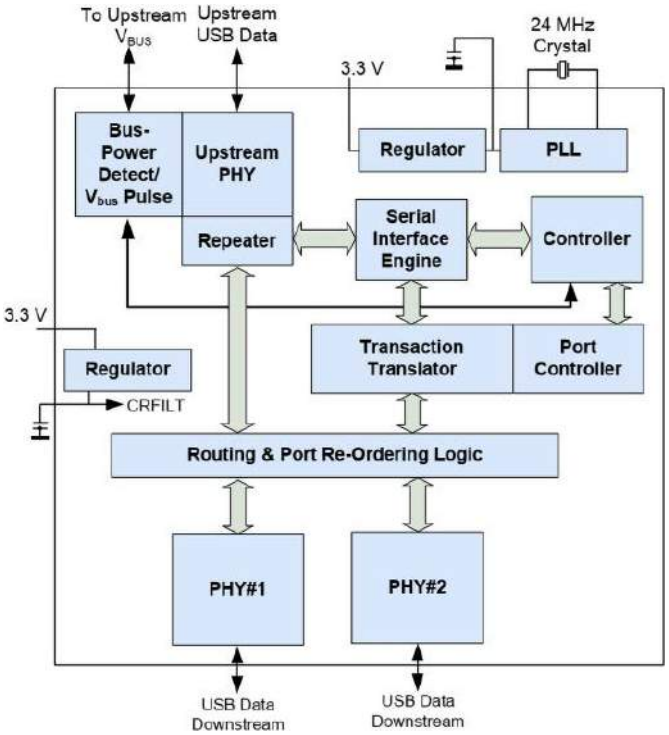
Rezystory:
R1, R2: 5,1 kΩ (SMD 0402, 1%)
R3: 2,2 kΩ (SMD 0402, 1%)
R4: 12 kΩ (SMD 0402, 1%)
R5: 47 kΩ (SMD 0402, 1%)
RP1: drabinka 4x100 kΩ (CRA06S08)

Kondensatory:
C1, C2, C9...C12: 10 µF (SMD 0402, X7R)
C3...C6, C13: 100 nF (SMD 0402, X7R)
C7, C8: 22 pF (SMD 0402, C0G)

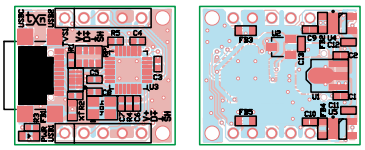
Półprzewodniki:
U1: MCP1700T-3302MB (SOT-89)
U2: MCP100T300/TT (SOT-23)
U3: USB2412 (QFN28_050)
U4, U5: AP2151AW-7 (SOT-25)
TVS1: transil ESD204 (USON10)
PWR: LED SMD (0603)

Pozostałe:
FB1...FB5: dławik ferrytowy BLM18EG101 (SMD 0603)
USB1, USB2: złącze szpilkowe 2,54 mm (SIP5)
USBC: gniazdo USB C, typ USB4110
XT: rezonator kwarcowy 24 MHz (SMD 3,2×2,5 mm)

eprasa.pl 65e3eb482b



Rysunek 1. Schemat blokowy USB2412 (za notą SMSC/Microchip)



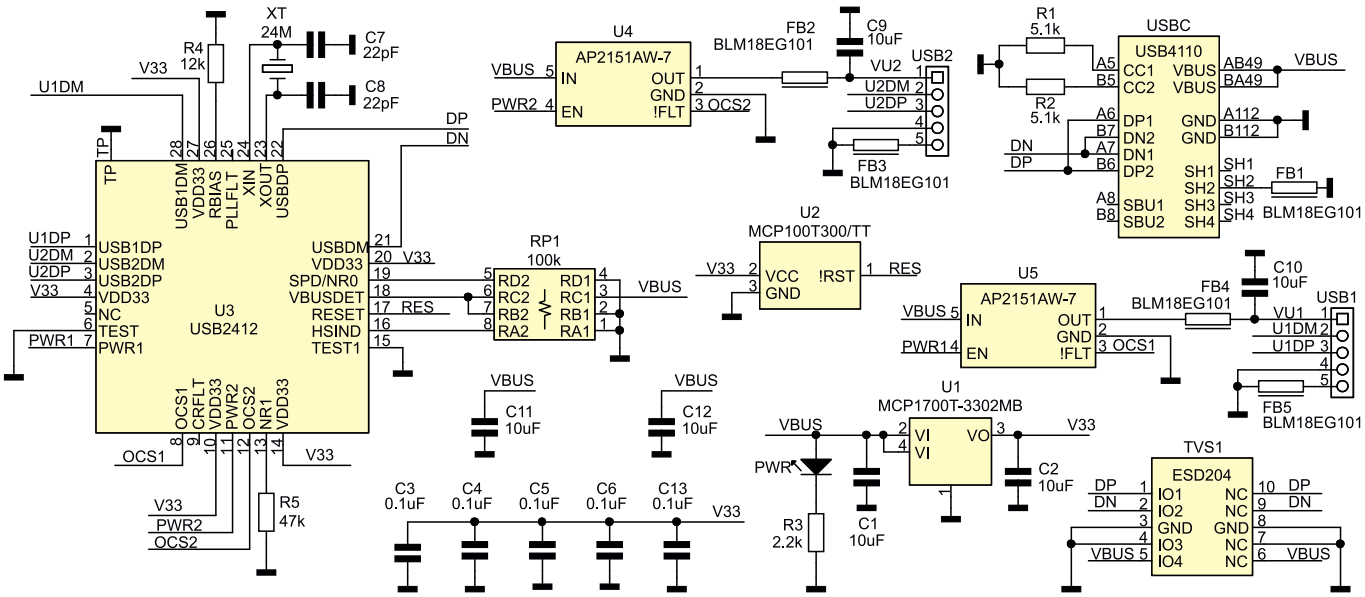
Rysunek 3. Rozmieszczenie elementów

w układ zarządzania zasilaniem odpowiedzialny za kontrolę i zabezpieczenie urządzenia USB. Sygnały PWR1/2 sterują kluczami włączonymi na liniach zasilających, a linie OCS1/2 powiadamiają U3 o przeciążeniu (awarii), spowodowanym przez podłączone urządzenia. Próg sygnalizacji przeciążenia układu AP2151 ustalono na ok. 800 mA. Złącza szpilkowe USB1/2 zapewniają dostęp do portów downstream USB2412, pozostałe elementy układu odsprzęgają i filtrują napięcie zasilające.

Układ zmontowany jest na niewielkiej, dwustronnej płytce drukowanej. Rozmieszczenie elementów zaprezentowano na **rysunku 3**, zaś zmontowany hub pokazano na **fotografii otwierającej**.

Układ – po poprawnym montażu – nie wymaga uruchamiania i podłączony do PC, powinien automatycznie zgłosić się jako *Generic USB Hub*. Stałe połączenia wewnętrzne z urządzeniami USB należy wykonać przewodem przeznaczonym do okablowania USB bądź wykonanym samodzielnie z odcinków przewodu TSK1160/TSK1161 i odpowiednich wtyków/gniazd lub z przewodów prefabrykowanych (np. AK674, USBAJ-2 BQ CABLE), stosowanych w komputerach PC.

Adam Tatuś, EP



Rysunek 2. Schemat ideowy układu

zwiększa elastyczność zastosowań huba, pozwalając zaoszczędzić trochę miejsca, a standardowy rozstaw złączy ułatwia zastosowanie także podczas prototypowania układów z użyciem płytek stykowych.

Hub oparty jest o popularny kontroler typu USB2412, którego strukturę wewnętrzną zaprezentowano na **rysunku 1**.

Dzięki wysokiej skali integracji układ zawiera w sobie praktycznie wszystkie elementy niezbędne do realizacji huba USB, ograniczając implementację do zaledwie kilku zewnętrznych komponentów biernych. Schemat ideowy układu pokazano na **rysunku 2**. Hub zasilany jest z portu SBC poprzez złącze USB-C, a ochronnik TVS1 zabezpiecza układ przed skutkami przepięć na liniach danych USB i zasilaniu. Stabilizator U1 dostarcza napięcia 3,3 V zasilającego kontroler USB2412. U3 taktowany jest kwarcem XT 24 MHz, natomiast poprawny restart po włączeniu zasilania zapewnia U2 typu MCP100T. Rezystory podłączone pomiędzy wyprowadzenia NR0/1 a masę informują U3, że urządzenia portów downstream USB1/2 mogą być odłączane/podłączane podczas pracy (nie są wpięte do huba na stałe). Każdy z portów USB1/2 wyposażony został

REKLAMA

eprasa.pl 65e3eb482b

AVT sklep

W ofercie AVT*

AVT6031

Podstawowe parametry:
- pomiar wilgotności względnej powietrza w pomieszczeniach zamkniętych,
- prezentacja wyniku za pomocą trójkolorowej diody LED (RGB),
- szybki odczyt – jedna z trzech barw świecenia informuje o stanie wilgotności powietrza,
- scalony, fabrycznie skalibrowany czujnik wilgotności,
- możliwość indywidualnego dostosowania jasności każdej z trzech barw,
- małe wymiary, prosta budowa,
- praca w zakresie temperatur 0...50°C,
- zasilanie napięciem stałym w przedziale 9...35 V,
- pobór prądu: około 10...20 mA.

Dodatkowe materiały do pobrania ze strony www.ulubionykiosk.pl/media

–––––	Czujnik punktu rosy (EP 8/2023)
–––––	Wysokościomierz barometryczny (EP 12/2022)
AVT5955	Termo-higrometr LED (EP 10/2022)
–––––	Stacja pogodowa WS-01 (EP 3/2022)
AVT5722	microStation – mała stacja pogodowa (EP 10/2019)
AVT5654	Amatorska stacja pogodowa (EP 12/2018)
AVT5639	Bezprzewodowy czujnik warunków atmosferycznych (EP 10/2018)
AVT5605	wiStation – domowa stacja pogodowa z prognozą pogody (EP 9/2017)
AVT5566	THPStation – rozbudowany termometr z Wi-Fi (EP 1/2017)
AVT5328	Miernik wybranych parametrów środowiskowych (EP 2/2012)
AVT961	Domowa stacja pogodowa (EP 12/2006)
AVT957	Moduł pomiaru temperatury (EP 11/2006)

* **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]** – płytka drukowana bez elementów i dokumentacji.

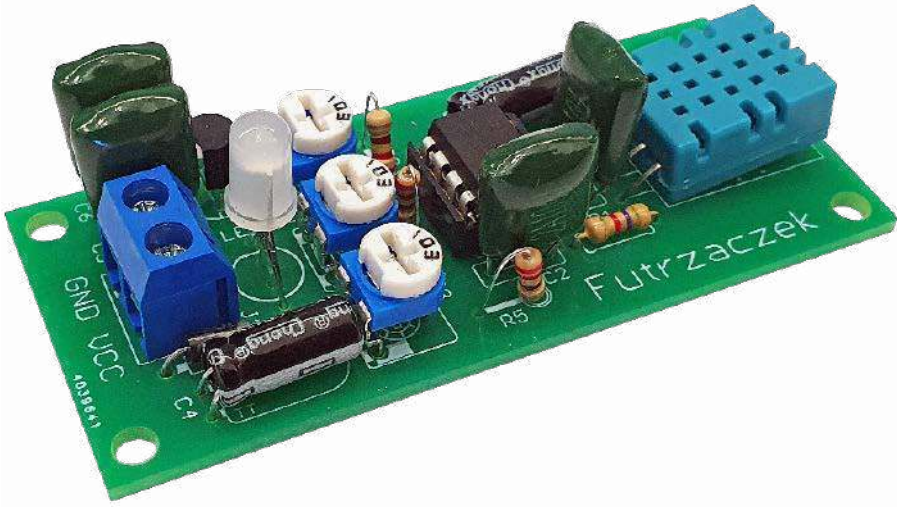
Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- **wersja [A+]** – płytka drukowana **[A]** + zaprogramowany układ **[UK]** i dokumentacja,
- **wersja [UK]** – zaprogramowany układ.

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik PDF! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! **http://sklep.avt.pl**

W przypadku braku dostępności na stronie sklepu osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: **kity@avt.pl**.

Prosty higrometr RGB

Względna wilgotność powietrza w pomieszczeniach to czynnik istotny dla naszego stanu zdrowia oraz samopoczucia. Choć rozmaite układy potrafią ją mierzyć, to znajomość dokładnej wartości tego parametru okazuje się często niepotrzebna. W wielu wypadkach szybka, prosta do zrozumienia informacja będzie cenniejsza, ponieważ nie wymaga odczytywania i porównywania zmieniających się wskazań liczbowych. Omawiany w artykule higrometr służy do zgrubnej oceny poziomu wilgotności powietrza.



Wilgotność powietrza na poziomie 69%: to dużo czy mało? A 33%? Owszem, osoby szczególnie wrażliwe – na przykład cierpiące na przewlekłe choroby górnych dróg oddechowych – są zmuszone do szczególnie wnikliwego analizowania parametrów powietrza w swoich domach, w tym jego wilgotności. Jednak w większości przypadków tak szczegółowe informacje nie są potrzebne, wręcz utrudniają szybką ocenę sytuacji: za sucho czy zbyt wilgotno? Właśnie po to powstał niniejszy układ.

Jeżeli wilgotność względna zawiera się w przedziale 40...60%, dioda LED wbudowana w układ świeci na zielono – jest to powszechnie przyjęty przedział, w którym wartość tego parametru można uznać za prawidłowy. Poniżej 40% zobaczymy – zamiast barwy zielonej – czerwoną (zbyt sucho). Natomiast dla wartości większych niż 60% układ emituje światło niebieskie (zbyt wilgotno). To proste komunikaty, szybkie w odczycie i bardzo czytelne.

Budowa

Schemat ideowy omawianego układu znajduje się na **rysunku 1**. Głównym podzespołem zawiadującym jego pracą jest mikrokontroler US2 typu ATtiny13PA-PU z 8-bitowym rdzeniem AVR. Ma wystarczającą liczbę konfigurowalnych wyprowadzeń, więc nie zachodzi potrzeba stosowania dodatkowych układów pośredniczących. Ten mały, prosty i tani mikrokontroler ma tylko 1 kB pamięci programu, lecz jest to pojemność w zupełności wystarczająca do tego zastosowania.

W roli sensora wilgotności pracuje gotowy moduł DHT11. Jest fabrycznie skalibrowany, dzięki czemu użytkownik nie musi przeprowadzać tego procesu samodzielnie. Wprawdzie wykonywany przez czujnik pomiar nie jest zbyt dokładny, ponieważ dopuszczalny błąd wynosi ±4%, lecz w tym zastosowaniu nie ma to większego znaczenia. Do komunikacji z mikrokontrolerem służy jednoprzewodowa magistrala cyfrowa, którą do potencjału +5 V podciąga zewnętrzny rezystor R2 – zgodnie z wymaganiami noty katalogowej układu.

Przedział wartości, w którym znajduje się aktualnie zmierzona wilgotność powietrza, wskazywany jest poprzez załączenie jednej z trzech struktur wchodzących w skład trójkolorowej diody LED1 o wspólnej katodzie.

Wykaz elementów:

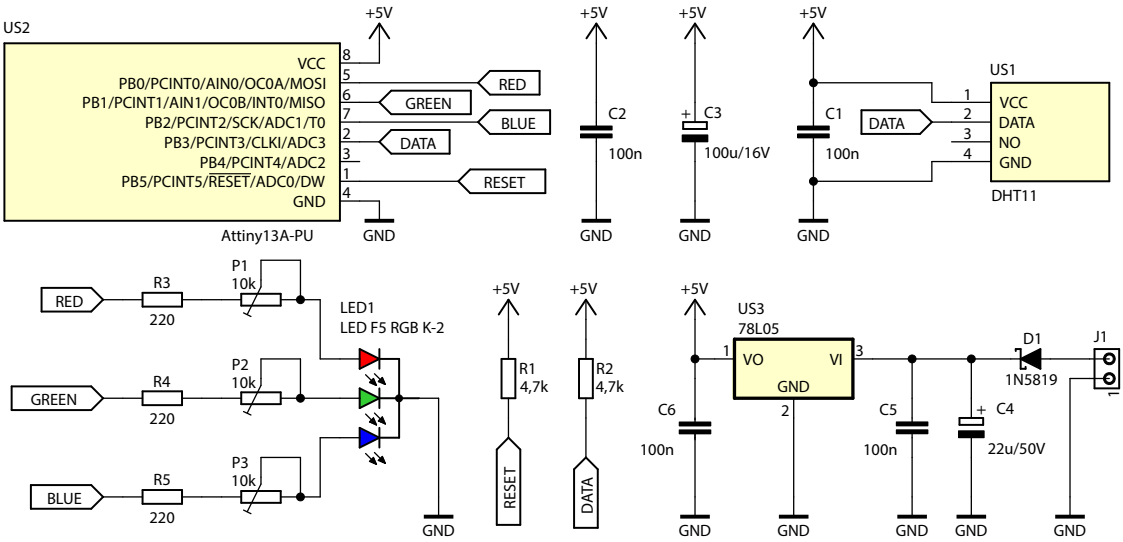
Rezystory: (THT o mocy 0,25 W)
R1, R2: 4,7 kΩ
R3...R5: 220 Ω
P1...P3: potencjometry montażowe leżące 10 kΩ

Kondensatory:
C1, C2, C5, C6: 100 nF (raster 5 mm, MKT)
C3: 100 µF 16 V (raster 2,5 mm)
C4: 22 µF 50 V (raster 2,5 mm)

Półprzewodniki:
D1: 1N5819
LED1: LED F5 RGB K-2 (opis w tekście)
US1: DHT11
US2: ATtiny13A-PU (DIP8)
US3: 78L05 (TO92)

Inne:
J1: złącze ARK2/500

eprasa.pl 65e3eb482b



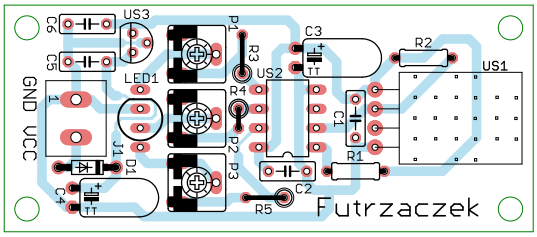
Rysunek 1. Schemat ideowy prostego higrometru RGB

Prąd każdej ze struktur jest ograniczony przez szeregowe połączenie rezystora 220 Ω i potencjometru, którego rezystancję można regulować w zakresie 0…10 kΩ. Wybrano takie rozwiązanie, ponieważ umożliwia proste i wygodne ustalenie pożądanej jasności każdej z barw, które są przez nasz wzrok interpretowane z różną intensywnością.

Do poprawnej pracy mikrokontrolera oraz czujnika temperatury wymagana jest stabilna wartość napięcia zasilającego, na przykład 5 V. Zapewnia je niewielki stabilizator liniowy US3 typu 78L05. Pobór prądu omawianego urządzenia jest na tyle niski, że z chłodzeniem stabilizatora nie będzie problemów nawet podczas pracy w pełnym zakresie dopuszczalnego dla układu 78L05 napięcia wejściowego. Dioda D1 chroni układ przed uszkodzeniem w razie omyłkowej zamiany biegunowości zasilania, które należy podłączyć do zacisków złącza J1.

Montaż i uruchomienie

Układ został zmontowany na jednostronnej płytce drukowanej o wymiarach 30 mm × 70 mm. Jej wzór ścieżek oraz schemat montażowy przedstawia **rysunek 2**. W odległości 3 mm od krawędzi płytki znalazły się cztery otwory montażowe, każdy o średnicy 3,2 mm.



Rysunek 2. Schemat montażowy i wzór ścieżek płytki

Montaż proponuję rozpocząć od elementów o najmniejszej wysokości obudowy, czyli rezystorów i diody D1. Układ US1 może leżeć poziomo na powierzchni laminatu. Pod mikrokontroler proponuję zastosować podstawkę, aby ułatwić jego programowanie oraz wymianę w razie uszkodzenia. Zmontowany układ można zobaczyć na **fotografii 1**.

Na etapie uruchamiania urządzenia konieczne jest zaprogramowanie pamięci Flash mikrokontrolera dostarczonym wsadem oraz zmiana ustawień bitów zabezpieczających. Oto ich nowe wartości:

Low Fuse = **0x7A**

High Fuse = **0xF9**

Szczegóły są widoczne na **rysunku 3**, który zawiera widok okna konfiguracji tychże bitów z programu BitBurner. W ten sposób zostanie wyłączony preskaler sygnału zegarowego oraz włączy się Brown-Out Detector, który wprowadzi mikrokontroler

Rysunek 3. Szczegóły ustawienia bitów zabezpieczających

w stan zerowania, jeżeli jego napięcie zasilające spadnie poniżej 4,3 V. Taka procedura znacznie zmniejsza ryzyko zawieszenia się układu podczas uruchamiania.

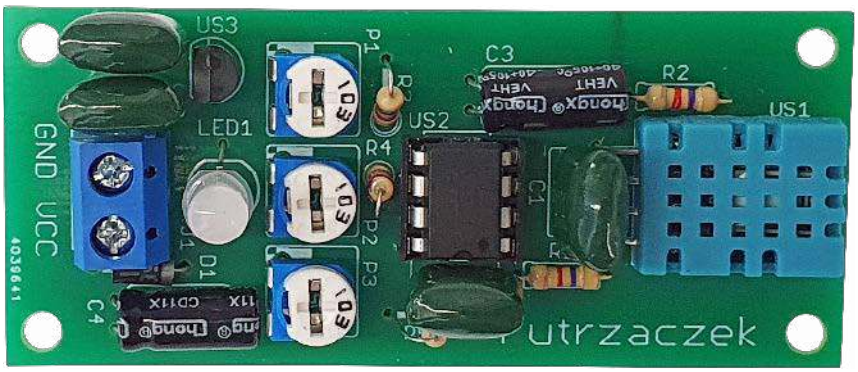
Poprawnie zaprogramowany układ jest gotowy do działania po podłączeniu zasilania do zacisków złącza J1. Napięcie zasilające powinno mieścić się w granicach 9…35 V. Obie te granice wynikają z warunków wskazanych przez producenta stabilizatora 78L05 jako niezbędne do zapewnienia prawidłowej pracy komponentu. Pobór prądu przez ten układ wynosi 10…20 mA, zależnie od natężenia prądu płynącego przez aktualnie załączoną strukturę diody LED1. W prototypie użyto diody o kodzie handlowym LED F5 RGB K-2, lecz może to być dowolna inna dioda RGB małej mocy ze wspólną katodą i identycznym układem wyprowadzeń.

Po włączeniu zasilania układ przez kilka sekund próbuje nawiązać komunikację z czujnikiem, po czym odświeża wynik co dwie sekundy. Barwa świecenia diody LED1 informuje o wilgotności względnej powietrza:

- zielona: prawidłowa (40…60%),
- czerwona: zbyt sucho (0…40%),
- niebieska: zbyt wilgotno (60…100%).

Jeżeli podczas komunikacji z czujnikiem DHT11 wystąpiłby błąd, dioda LED1 będzie co kilka sekund emitować krótkie, czerwone błyski. Oznacza to uszkodzenie połączeń lub któregoś z elementów na płytce – układ zgłasza konieczność dokonania serwisu.

Michał Kurzela, EP



Fotografia 1. Szczegółowy widok zmontowanego układu

eprasa.pl 65e3eb482b

Ulubiony Kiosk

Pobierz bezpłatnie multimedialne dodatki do tego wydania Elektroniki Praktycznej

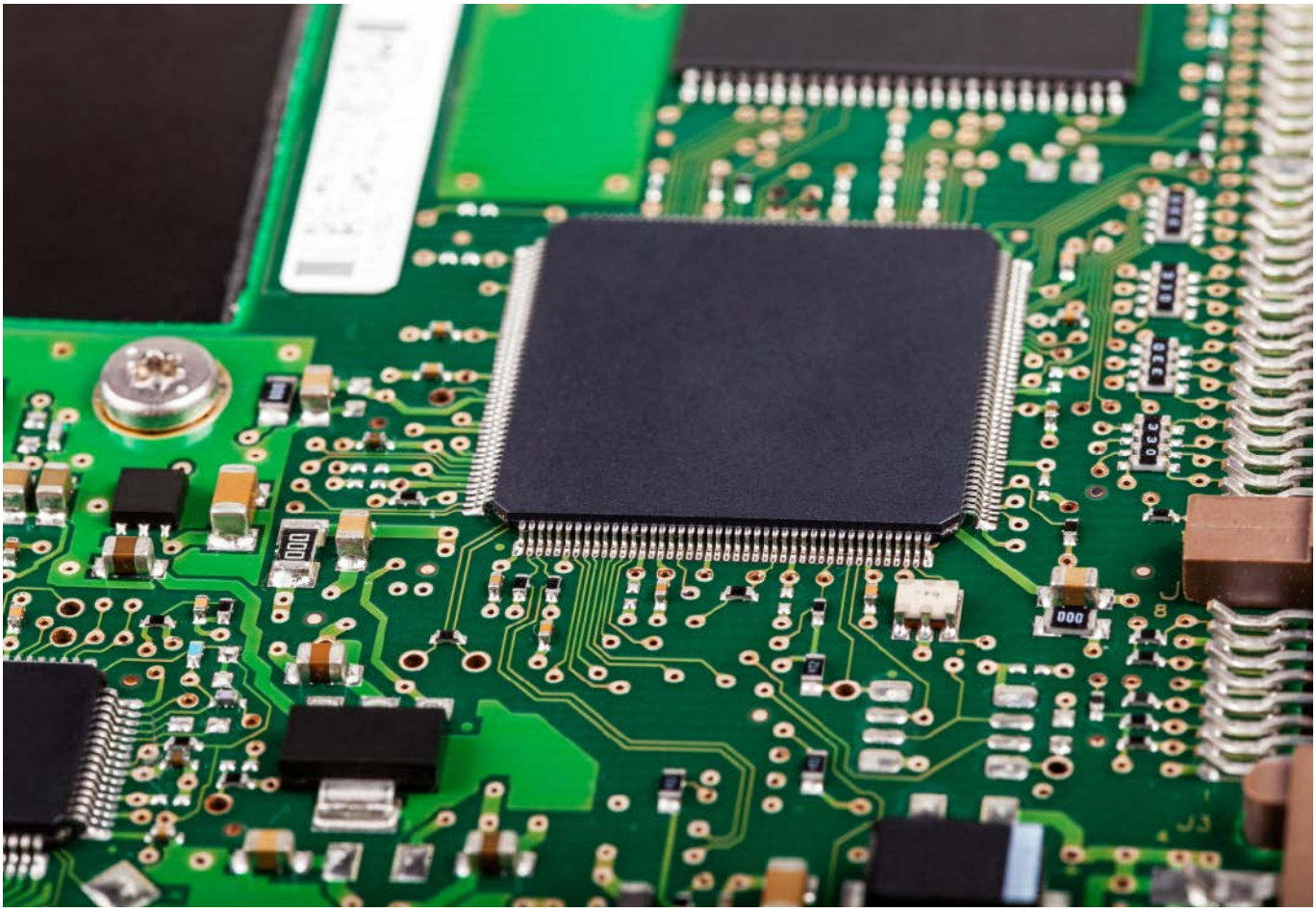
Projekty, miniprojekty, materiały do artykułów i kursów oraz wiele innych!

* **Kupiłeś magazyn** w Ulubionym Kiosku lub masz prenumeratę? Multimedialne dodatki będą odblokowane automatycznie!

* **Zakupiłeś czasopismo** u zewnętrznego dystrybutora? Odblokuj bibliotekę multimediów samodzielnie.

Szczegóły na **UlubionyKiosk.pl/media**

eprasa.pl 65e3ed482b



Przekrojowa oferta mikrokontrolerów w portfolio firmy Micros

Trudno nie zgodzić się ze stwierdzeniem, że mikrokontrolery napędzają współczesną elektronikę i to we wszystkich jej obszarach aplikacyjnych. Dlatego w ofercie firmy Micros – polskiego dystrybutora komponentów elektronicznych, systemów oświetlenia, minikomputerów, narzędzi oraz wielu innych akcesoriów dla elektroników – znalazła się obszerna kategoria poświęcona w całości mikrokontrolerom i obejmująca tysiące zróżnicowanych produktów dostarczanych przez najważniejszych graczy na rynku półprzewodników.

Więcej informacji:
Więcej informacji nt. oferty Micros można znaleźć na stronie: www.micros.com.pl
Micros sp.j. W. Kędra i J. Lic
30-198 Kraków, ul. E. Godlewskiego 38
tel. +48 12 636 95 66, e-mail: bok@micros.com.pl

Dlaczego nie istnieje jeden mikrokontroler idealny do wszystkich aplikacji?

Współczesna oferta mikrokontrolerów obejmuje niezwykle szeroki wachlarz układów – od prostych, 8-bitowych procesorów w obudowach o zaledwie kilku wyprowadzeniach, aż po potężne, wielordzeniowe procesory aplikacyjne do zastosowań związanych z obsługą multimediów, obróbką obrazu czy też uruchamianiem rozbudowanych algorytmów sztucznej inteligencji (AI). I właśnie zakres docelowych zastosowań wymusza użycie różnorodnych mikrokontrolerów w określonych typach projektów. Niewielkie urządzenia przenośne, gadżety mobilne, sprzęt medyczny czy też zdalne czujniki IoT o zasilaniu bateryjnym wymagają procesorów typu ultra-low power, czyli zoptymalizowanych pod względem oszczędzania energii i to na wszystkie możliwe sposoby. W aplikacjach związanych ze sterowaniem napędami (w tym przede wszystkim silnikami BLDC) konieczny okazuje się bogaty zestaw bloków peryferyjnych – zaawansowanych timerów z funkcjami zabezpieczającymi, przetworników ADC czy też zintegrowanych obwodów do analogowego kondycjonowania sygnałów.

eprasa.pl 65e3eb482b

A to wciąż tylko dwa spośród tysięcy możliwych zastosowań, w których wybór mikrokontrolera ma kluczowe znaczenie dla funkcjonalności i niezawodności docelowego urządzenia.

Microchip (dawniej Atmel)

Firma Microchip – jeszcze przed przejęciem Atmela – słynęła z doskonałych mikrokontrolerów PIC oraz procesorów sygnałowych dsPIC. Fuzja z producentem kultowych już układów AVR oraz SAM sprawiła, że Microchip stał się jednym z kilku najważniejszych graczy na rynku półprzewodników. Dziś oferta układów AVR, PIC16, PIC18, PIC24 czy PIC32 obejmuje szeroką gamę mikrokontrolerów, które znaleźć można w niezliczonych obszarach aplikacyjnych.

ST Microelectronics

ST Microelectronics to jeden z potentatów na rynku nowoczesnych mikrokontrolerów z rdzeniem ARM. Wszystkie oferowane przez producenta układy można podzielić na cztery główne grupy (**rysunek 1**): procesory „radiowe" (z wbudowanym transceiverem), ultraenergooszczędne, a także układy standardowe (do typowych aplikacji) oraz mikrokontrolery o wysokiej wydajności, przeznaczone do zastosowań związanych m.in. z obsługą multimediów, systemów operacyjnych wysokiego poziomu itp.

Rysunek 1. Portfolio mikrokontrolerów marki ST Microelectronics

Megawin

Oferta firmy **Megawin** obejmuje zróżnicowane modele mikrokontrolerów o architekturze 8051 oraz ARM Cortex-M3 i ARM Cortex-M0. Układy wyróżniają się doskonałym wyposażeniem peryferyjnym oraz wyjątkowo przystępnym stosunkiem jakości do ceny, co przekłada się na spore oszczędności w przypadku produkcji wielkoseryjnej. Warto dodać, że kilka spośród modeli procesorów Megawin zapewnia kompatybilność z układami STM32F oraz STM8S.

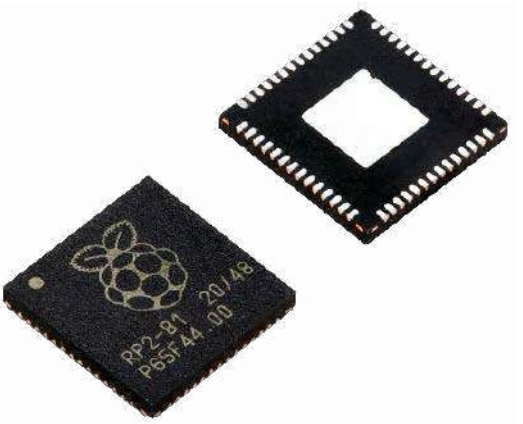
Geehy Semiconductor

Marka Geehy Semiconductor wdrożyła ponad 20 serii mikrokontrolerów, przy czym prawie połowa z nich jest przeznaczona do aplikacji z sektora motoryzacyjnego. Dostępne są układy o architekturze ARM Cortex w wersjach M0+, M3 oraz M4. Większość układów zapewnia kompatybilność z procesorami serii STM32, dzięki czemu producenci OEM mogą w prosty i szybki sposób przejść na alternatywne rozwiązanie o doskonałym stosunku jakości do ceny.

NXP

Mikroprocesorowe portfolio firmy NXP obejmuje zarówno ultranowoczesne procesy aplikacyjne i wydajne mikrokontrolery

Fotografia 1. Przykładowe mikrokontrolery z rodziny LPC11xx marki NXP



Fotografia 2. Mikrokontrolery RP2040 marki Raspberry Pi

eprasa.pl 65e3eb482b

Rysunek 2. Zarys portfolio mikrokontrolerów firmy Megawin

na bazie architektury ARM, jak i szeroką gamę mniejszych mikrokontrolerów do różnych aplikacji. NXP dzierży także spuściznę takich marek, jak Motorola oraz Freescale, do dorobku której należą m.in. rodziny procesorów HC08S, HCS12 czy też Kinetis.

Inni producenci

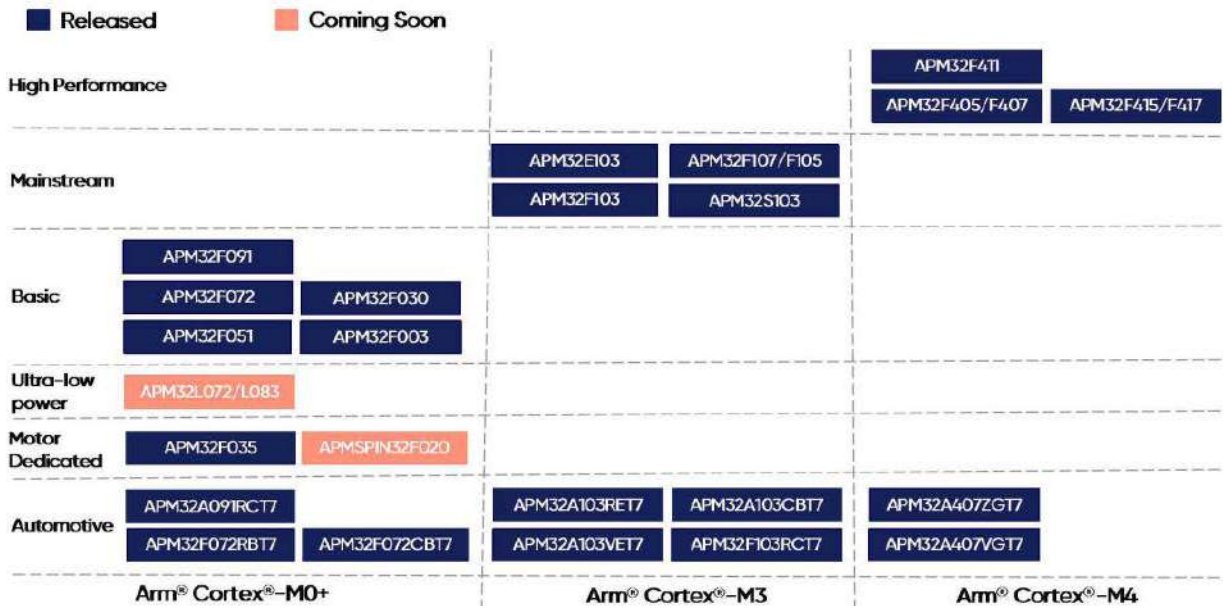
W ofercie firmy Micros znajdują się także mikrokontrolery pochodzące zarówno z pracowni innych wiodących marek – w tym największych półprzewodnikowych koncernów (**Renesas**, **Silicon Labs**, **Cypress**, **Texas Instruments**, **Analog Devices**), jak i producentów mniej znanych na rynku polskim – **Gigadevice**, **Mindmotion**, **Levetop** czy też **Holychip**. Dość niedawno – bo zaledwie trzy lata temu – na rynek wkroczył wydajny, dwurdzeniowy procesor **RP2040** marki **Raspberry Pi**, który także trafił do asortymentu polskiego dystrybutora wraz z kilkoma modelami bazujących na tym układzie płytek deweloperskich.

Warto wiedzieć, że oprócz wymienionych powyżej firm istnieje także wielu innych producentów mikrokontrolerów. Wśród dostawców firmy Micros znajdują się m.in. marki **Allwinner**, **AMD**, **Bridgetek**, **CKS**, **Infineon**, **Maxim Integrated**, a nawet **Toshiba** czy **Zilog**. Szeroką ofertę układów uzupełniają wydajne procesory sygnałowe, przeznaczone do aplikacji wymagających uruchamiania algorytmów cyfrowego przetwarzania sygnałów (DSP) – w tym zakresie przodują marki takie, jak Analog Devices, Texas Instruments oraz NXP.

Nie tylko mikrokontrolery

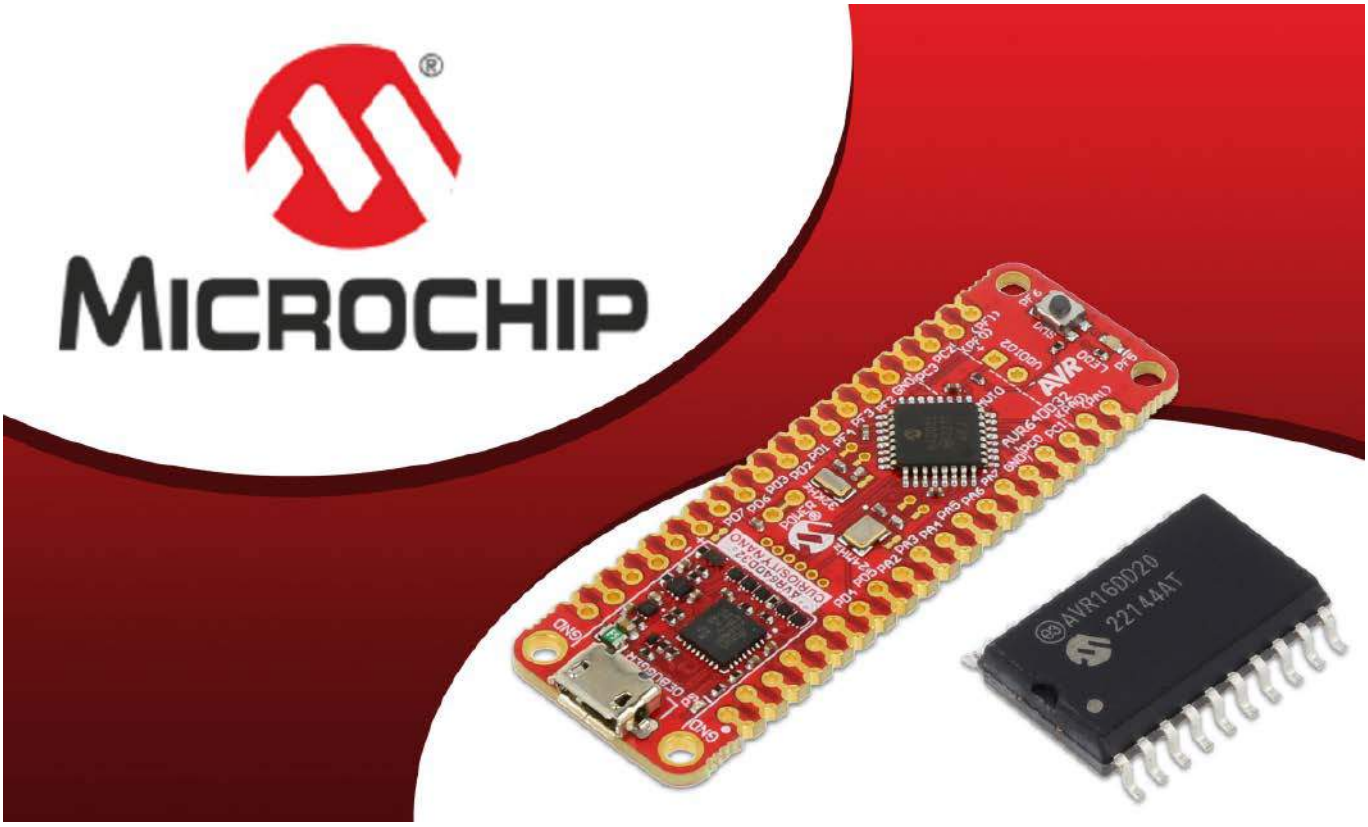
W nowoczesnych urządzeniach IoT klasyczne mikrokontrolery okazują się zwykle niewystarczające – znacznie efektywniejszym rozwiązaniem jest zastosowanie specjalizowanych procesorów „radiowych" (zintegrowanych z transceiverem pracującym w odpowiednim paśmie transmisyjnym) bądź – co jeszcze bardziej ułatwia konstrukcję docelowego urządzenia – gotowych modułów RF. W tym zakresie Micros oferuje szeroką gamę modułów marki **EBYTE**, inteligentne moduły Bluetooth oraz subminiaturowe moduły typu SiP marki **Silicon Laboratories**, transceivery produkcji **Nordic Semiconductor**, **Laird Connectivity**, **Rayson**, **Onsemi**, **Panasonic, Radiocontrolli**, **Microchip**, **Renesas** czy **ST Microelectronics**. Nie mogło oczywiście zabraknąć kultowych już modułów Wi-Fi/Bluetooth marki **Espressif**, szybkich transceiverów **AI-Thinker** czy też propozycji takich producentów, jak **Murata**, **Olimex**, **Texas Instruments**, **WIZnet**, **SparkLAN** oraz **Wireless-Tag**.

Warto wspomnieć, że mikroprocesorową część bogatej oferty firmy Micros uzupełniają uniwersalne układy pamięci typu Flash, EPROM, EEPROM, DRAM, FRAM, MRAM, SDRAM oraz SRAM.



Rysunek 3. Rodziny mikrokontrolerów ARM w ofercie marki Geehy Semiconductor

eprasa.pl 65e3eb482b



Mikrokontrolery z rodziny AVR DB oraz AVR DD

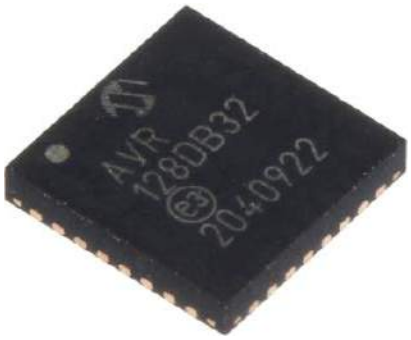
– zaprojektowane do pracy w obwodach o zróżnicowanym poziomie napięć

Jeszcze do niedawna bardzo typowym problemem, z którym borykali się projektanci obwodów elektronicznych, były różne standardy napięć zasilających mikrokontroler (np. 3,3 V) oraz układy peryferyjne (np. 5 V). Jeśli miały ze sobą współpracować, konieczne było stosowanie dodatkowych układów translacji napięć, co komplikowało urządzenie i wymuszało podniesienie ceny. Ale to już przeszłość.

Firma Microchip, znana z szerokiego portfolio 8-bitowych mikrokontrolerów z rodzin PIC i AVR, wprowadziła do sprzedaży nowo opracowane układy, które łączą w sobie najlepsze cechy obydwu wymienionych grup produktowych, a jednocześnie oferują projektantom urządzeń elektronicznych szerokie możliwości i ułatwiają budowę jeszcze bardziej zintegrowanych obwodów. Tak właśnie jest w przypadku energooszczędnych, 8-bitowych mikrokontrolerów z rodzin AVR DD/DB. Dostarczane są wraz z wyjątkowym zestawem peryferiów pracujących niezależnie od rdzenia (CIP, *Core Independent Peripherals*), służących przede wszystkim sprawnemu przetwarzaniu sygnałów analogowych. Zapewniają też bezpieczne funkcjonowanie aplikacji zgodnie z normami IEC 61508 oraz ISO 26262. Dzięki nim mikrokontrolery AVR DD/DB idealnie odpowiadają potrzebom mniej skomplikowanych aplikacji. Mogą też służyć jako układy towarzyszące (wspierające) w bardziej złożonych projektach.

Ogólne informacje o AVR DB i AVR DD

Prezentowane mikrokontrolery opracowano, by przyspieszyć i ułatwić etap projektowania, a tym samym szybciej wdrożyć produkcję docelowego urządzenia. Microchip zrealizował ten cel przez zintegrowanie w jednej strukturze rdzenia AVR z szeregiem peryferiów, pozwalających na wykonywanie wielu zadań, które dawniej musiały być obsługiwane przez układy zewnętrzne. Co więcej, układy



Fotografia 1. Mikrokontroler AVR128DB32-I-RXB z serii AVR DB

eprasa.pl 65e3eb482b

z serii DB i DD dostępne są w wielu zróżnicowanych obudowach, od klasycznych obudów DIP, aż po VQFN o polu podstawy jedynie 5×5 mm (**fotografia 1**). Odpowiada to potrzebom nowoczesnych obwodów elektronicznych, takich jak zdalne urządzenia typu IoT albo programowalne sterowniki, przekaźniki i czujniki przemysłowe [1].

Należy zaznaczyć, że prezentowane układy mają wszystkie typowe cechy i funkcje znane z produktów Microchipa. Są taktowane sygnałem zegarowym o częstotliwości 24 MHz, mają do 128 kB pamięci Flash (programowej), 16 kB SRAM (operacyjnej) i 512 B nieulotnej pamięci EEPROM tradycyjnie stosowanej do przechowywania danych o nastawach, numerach seryjnych itp. Obudowy mikrokontrolerów wyposażone są w maksymalnie 64 piny o zróżnicowanej funkcjonalności. Oprócz tradycyjnej roli portów wejścia/wyjścia (czyli GPIO), zostały połączone z przetwornikami analogowo-cyfrowymi oraz szeregowymi interfejsami SPI, I²C, UART. W mikrokontrolerach zintegrowano również zegar czasu rzeczywistego (RTC), liczniki/timery (służące m.in. do generowania sygnałów PWM), komparator analogowy, źródła napięcia odniesienia (o kilku wartościach). Co więcej, w układzie znalazł się nawet przetwornik cyfrowo-analogowy (DAC) oraz zaawansowany watchdog. Inne rozbudowane podzespoły, działające niezależnie od rdzenia (CIP), omówiono bardziej szczegółowo w dalszej części tekstu [2].

Funkcjonalność MVIO

MVIO (Multi-Voltage Input/Output) to konfigurowalny, zintegrowany translator napięcia poziomów logicznych (level shifter). Dzięki niemu część wyprowadzeń mikrokontrolera może być zasilana ze źródła o innej wartości napięcia niż to, z którego zasilany jest rdzeń jednostki. Takie rozwiązanie ogromnie usprawnia i poszerza możliwości mikrokontrolera – pozwala mu na bezpieczną komunikację z układami pracującymi na innych poziomach logicznych (np. 5 V przy zasilaniu 3,3 V). Oznacza to, że projektanci mogą swobodnie dobierać docelowe elementy obwodu bez obaw, że różnice w ich charakterystyce elektrycznej wywołają potrzebę umiejscowienia na PCB dodatkowych układów pośredniczących.

System MVIO został zaprojektowany do pracy z sygnałami logicznymi (cyfrowymi). Oczywiście, jego funkcjonalność obejmuje zarówno sygnały binarne, jak i sprzętowe magistrale komunikacyjne (np. SPI, I²C) czy generatory PWM wbudowane w układ. Bezpośrednie sterowanie elementami zewnętrznymi ma jeszcze tę zaletę, że jest w pełni nadzorowane przez mikrokontroler. Jeżeli wartość napięcia zasilającego MVIO (VDDIO2) spadnie poniżej zadanej wartości, możliwe jest wygenerowanie przerwania, co pozwala na szczegółowe monitorowanie stanu urządzenia. MVIO akceptuje napięcia z zakresu od 1,6 V do 5,5 V DC. W przypadku całkowitego braku zasilania na linii VDDIO2, piny MVIO pracują jako wejścia/wyjścia trójstanowe.

Peryferium MVIO można łatwo skonfigurować w środowisku programistycznym MPLAB X oraz ATMEL STUDIO 7 [3].

Zestawy uruchomieniowe

Producent ułatwił zapoznanie się z funkcjonalnością układów AVR DB i AVR DD (**fotografia 2**) poprzez przygotowanie

Fotografia 2. Mikrokontroler AVR32DD20-ISO z serii AVR DD

REKLAMA

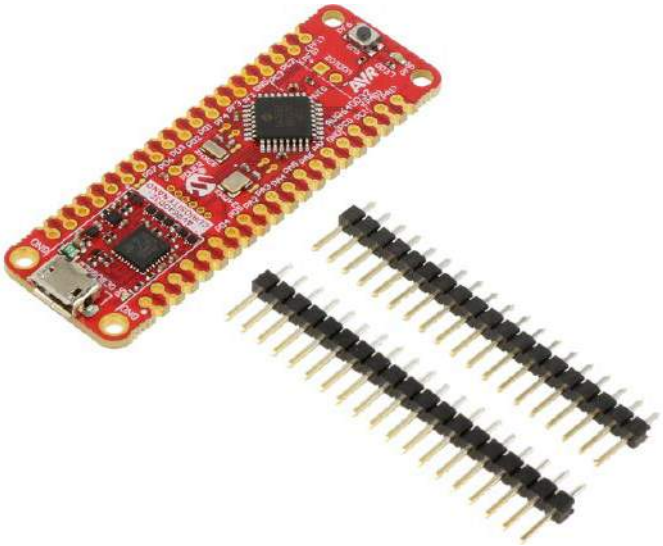
TME Electronic Components
TRANSFER MULTISORT ELEKTRONIK

SEGGER
It simply works!

PROGRAMATORY I DEBUGGERY SEGGER
JUŻ W OFERCIE TME

Transfer Multisort Elektronik Sp. z o.o.
Łódź, Polska, dso@tme.pl
Dołącz do nas:
tme.eu
tme.com

eprasa.pl 65e3eb482b



Fotografia 3. Zestaw ewaluacyjny EV72Y42A – płytka uruchomieniowa z mikrokontrolerem z serii AVR DD oraz kompletem złączy szpilkowych

zestawów uruchomieniowych. Zostały one wykonane w formacie niewielkich płytek PCB z wyprowadzeniami do montażu złączy kołkowych lub instalacji w technologii powierzchniowej (SMT). Z katalogu TME można nabyć dwie płytki: EV72Y42 z mikrokontrolerem AVR64DD32 (**fotografia 3**) oraz EV35L43A z mikrokontrolerem AVR128DB48.

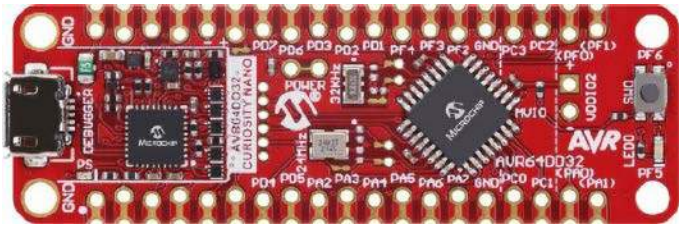
Obydwie należą do rozwijanej przez Microchipa rodziny Curiosity Nano (**fotografia 4**) – platformy mającej na celu sprawne, stopniowe wdrażanie elektroników w zagadnienia związane z programowaniem oraz aplikacją mikrokontrolerów AVR w produktach i indywidualnych projektach. Pod względem zawartości zestawu obydwa modele mają zbliżoną konstrukcję. Do komunikacji z komputerem PC oraz jako główne źródło zasilania służą w nich gniazda Micro USB. Zawierają sygnalizacyjne diody LED (wskaźniki obecności napięcia zasilającego oraz diody obsługiwane programowo), rezonatory kwarcowe taktujące rdzeń oraz (wbudowany w strukturę mikrokontrolera) zegar czasu rzeczywistego (RTC), a także przełącznik typu Micro Switch (podłączony do pinu I/O). Co najważniejsze, w skład obwodu wchodzi zintegrowany debugger, pozwalający na szczegółową analizę pracy mikrokontrolera. W związku z funkcjonalnością MVIO na płytkach znalazły się też osobne wyprowadzenia, umożliwiające doprowadzenie alternatywnego napięcia zasilania.

Inne możliwości układów DB i DD

Oprócz systemu MVIO, w prezentowanych mikrokontrolerach zastosowano szereg innych rozwiązań. Są to m.in. peryferia typu CIP, czyli pracujące niezależnie od rdzenia. Poniżej przedstawiono najważniejsze z nich – chociaż należy podkreślić, że pełna funkcjonalność serii AVR DB/DD jest bardzo szerokim tematem, a poszczególne układy mogą się między sobą różnić pod względem konstrukcyjnym.

CFD (Clock Failure Detect)

CFD (Clock Failure Detect) – niezmiernie ważny ze względu na bezpieczeństwo funkcjonowania aplikacji. Jest to obwód nadzorujący obecność sygnału taktującego z zewnętrznego oscylatora (generatora kwarcowego). Gdy sygnał zaniknie, mikrokontroler automatycznie przełączy się na użycie wewnętrznie generowanego sygnału zegarowego. Jednocześnie zostanie wywołane przerwanie, z którego projektant może skorzystać w aplikacji, np. w celu zatrzymania programu lub przejścia do trybu awaryjnego, zapewniającego bezpieczne, nieprzerwane działanie obwodów.



Fotografia 4. Niewielka płytka PCB stanowi pełny zestaw uruchomieniowy do mikrokontrolerów z serii AVR DD

EVSYS (Event System)

EVSYS (Event System) to system pozwalający zintegrowanym urządzeniom peryferyjnym na bezpośrednie przesyłanie sygnałów do siebie nawzajem, niezależnie od rdzenia. Poprzez EVSYS wyjście z jednego podzespołu może zostać przekazane do wielu innych podzespołów peryferyjnych. Poprawia to czas reakcji i zmniejsza zużycie energii, umożliwiając jednocześnie realizację bardziej złożonych zadań, gdyż rdzeń CPU nie jest obciążany nieustanną obsługą prostych, cyklicznych operacji (np. warunkowych, dokonywanych w celu monitorowania portów I/O).

CCL (Custom Logic)

CCL (Custom Logic) występuje jedynie w rodzinie AVR DD. Jest to zestaw wbudowanych, programowalnych bramek logicznych i przerzutników. Dzięki możliwości konfiguracji połączeń, ich wejścia i wyjścia mogą być dołączone do fizycznych wyprowadzeń (pinów) mikrokontrolera. Alternatywnie konstrukcja pozwala na bezpośrednie połączenie CCL ze zintegrowanymi w układzie peryferiami, jak również odczytywanie przez program mikrokontrolera stanów wyjściowych CCL (chociaż samo działanie pozostaje niezależne od rdzenia). W wielu przypadkach pozwoli to wyeliminować zewnętrzne układy logiczne, a co za tym idzie, zminimalizować koszt opracowania urządzenia i powierzchnię PCB [4].

Zintegrowany wzmacniacz operacyjny

Zintegrowany wzmacniacz operacyjny (3 szt.) umieszczono w układach z grupy AVR DB. Obecność tych peryferiów mieszanych (mixed-signal) pozawala mikrokontrolerowi pełnić funkcje analizatora i przetwornika sygnałów analogowych, redukując liczbę wymaganych do tego celu komponentów zewnętrznych. Te ostatnie można ograniczyć do multiplekserów lub drabinek rezystorowych (jeżeli zachodzi taka potrzeba).

Wbudowany przetwornik analogowo-cyfrowy (ADC)

Standardowo układy z obydwu prezentowanych rodzin mają zintegrowany przetwornik analogowo-cyfrowy (ADC) o rozdzielczości 12 bitów, odznaczający się maksymalną częstotliwością próbkowania na poziomie 130 kSps (tysięcy próbek na sekundę). Do jego obsługi można zastosować umieszczone w mikrokontrolerze źródło napięcia odniesienia o minimalnym dryfcie termicznym (niskim współczynniku temperaturowym), dostarczające stabilne napięcie referencyjne o wartości 1,024 V, 2,048 V, 2,500 V oraz 4,096 V. W połączeniu z wymienionymi wyżej peryferiami, takimi jak EVSYS czy wewnętrzny komparator, przetwornik ADC pozwala na błyskawiczną, efektywną i precyzyjną obróbkę sygnałów analogowych.

Transfer Multisort Elektronik Sp. z o.o.

Materiały dodatkowe:

[1] https://www.youtube.com/watch?v=M-myqg-2c5s
[2] https://www.youtube.com/watch?v=EH1FozhBKHc
[3] https://www.youtube.com/watch?v=RSpu1iM-hT8
[4] https://www.youtube.com/watch?v=beZXfAUR-PE

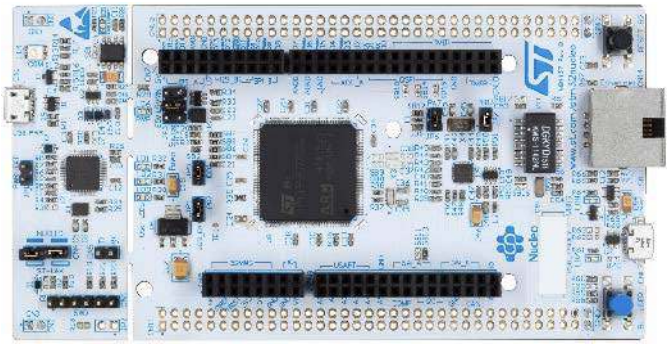
eprasa.pl 65e3eb482b

Narzędzia sprzętowe dla programistów systemów wbudowanych

Narzędzia przeznaczone dla programistów embedded coraz częściej integrują w sobie szereg dodatkowych funkcji, które znacznie ułatwiają prace związane z debugowaniem i uruchamianiem kodu. W artykule zaprezentujemy subiektywny wybór kilku najciekawszych propozycji z oferty firm najprężniej działających na rynku programatorów, debuggerów oraz zestawów ewaluacyjnych i rozbudowanych platform deweloperskich.

Programatory/debuggery

Jeszcze do niedawna zadania programatorów/debuggerów sprzętowych ograniczały się zwykle do zapisywania i odczytywania pamięci Flash mikrokontrolera oraz (w przypadku bardziej zaawansowanych rodzin procesorów, w tym ARM) do monitorowania oraz modyfikacji zawartości rejestrów i zmiennych. Dziś producenci narzędzi dla programistów coraz częściej integrują w nich szereg funkcji dodatkowych, znakomicie zwiększających użyteczność. Jako przykład warto tutaj wymienić programatory ST-Link zintegrowane z płytkami ewaluacyjnymi z serii STM32 Nucleo. Przykładowo – płytka NUCLEO-F439ZI zawiera (w „górnej" części, połączonej z głównym modułem za pomocą trzech wyłamywalnych łączników – **fotografia 1**) nowoczesny interfejs programistyczny typu STLINK/V2-1, STLINK-V3E lub STLINK-V3EC. Oprócz standardowego programowania i debugowania, umożliwia on także komunikację w trybie wirtualnego portu szeregowego – jeden z bloków USART głównego procesora STM32F439ZI jest sprzężony z mikrokontrolerem zawiadującym pracą interfejsu, co po odpowiedniej konfiguracji umożliwia przesył danych do/z procesora, bez konieczności stosowania zewnętrznego konwertera USB-UART.



Fotografia 1. Zestaw ewaluacyjny NUCLEO-F439ZI z mikrokontrolerem STM32F439ZI i wbudowanym programatorem/debuggerem ST-Link (http://t.ly/W9NrH)

Jeszcze ciekawsze rozwiązania konstrukcyjne zastosowała firma ST Microelectronics w opracowanym przez siebie urządzeniu STLINK-V3PWR (**fotografia 2**). Narzędzie, oprócz pełnienia funkcji programatora/debuggera, może także odgrywać rolę źródła mierzącego (SMU – *source measurement unit*), przeznaczonego do badania i profilowania poboru mocy układów energooszczędnych, np. czujników IoT czy urządzeń noszonych. Pomimo kompaktowych rozmiarów, narzędzie pozwala na zasilanie docelowego urządzenia napięciem w zakresie od 1,6 V do 3,6 V, przy poborze prądu nieprzekraczającym 500 mA (zabezpieczenie OCP jest ustawione na 550 mA). Co ważne, programowalna częstotliwość próbkowania może wynosić od 1 Sps do 100 kSps, przy czym zakres dynamiki pomiaru prądu jest niebywale szeroki – mierzalne są już natężenia na poziomie kilku nanoamperów, a górna granica zakresu jest równa 500 mA. Przyrząd oferuje interfejsy JTAG oraz SWD (Serial Wire Debug), wspiera ponadto komunikację przez port wirtualny (VCP). Mało tego – producent udostępnił nawet… wbudowany mostek USB – SPI/I²C/CAN/GPIO, dzięki czemu użytkownicy mogą jeszcze łatwiej debuggować swoje projekty poprzez bezpośredni dostęp do warstwy fizycznej lokalnych interfejsów szeregowych. Warto dodać, że – jak zawsze w przypadku narzędzi deweloperskich ze stajni ST – także i w tym przypadku sprzęt ma pełne wsparcie ze strony IDE. Jednostka SMU sprawnie współpracuje bowiem z oprogramowaniem STM32CubeMonitor-Power (**rysunek 1**), pozwalając na wygodne prowadzenie akwizycji oraz zapisywanie danych pomiarowych, rejestrowanych przy określonych przez użytkownika parametrach zasilania i wyzwalania.

Inne ciekawe rozwiązanie opracowała – doskonale znana na rynku narzędzi dla programistów embedded – firma MikroElektronika. Niewielkie urządzenie o nazwie UNI CODEGRIP (**fotografia 3**) to w pełni bezprzewodowy, uniwersalny programator/debugger przeznaczony do użycia z mikrokontrolerami STM32,

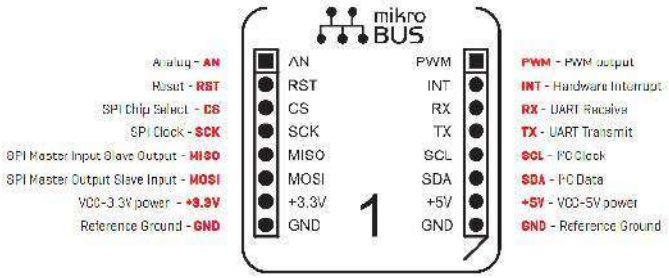
Fotografia 2. Programator/debugger z wbudowaną jednostką SMU – STLINK-V3PWR marki ST Microelectronics (http://t.ly/-Iu07)

Rysunek 1. Okno programu STM32CubeMonitor – Power (http://t.ly/cLilL)

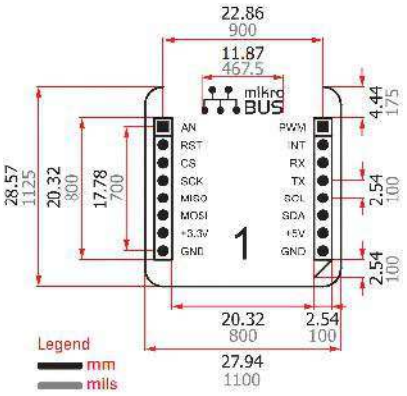
eprasa.pl 65e3eb482b

Kinetis, TIVA, CEC, MSP, PIC, dsPIC, PIC32, AVR oraz GD32, co daje łącznie ponad 2700 modeli obsługiwanych układów(!). Komunikację z komputerem zapewnia wbudowany moduł Wi-Fi, choć nic nie stoi na przeszkodzie, by skorzystać z konwencjonalnego, przewodowego połączenia przez USB-C. Producent oferuje także szereg kompatybilnych z urządzeniem adapterów, pozwalających na podłączenie programatora/debuggera do różnych urządzeń docelowych.

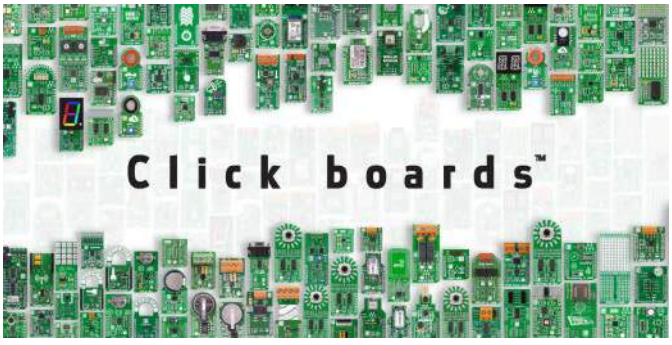
Fotografia 3. Uniwersalny, bezprzewodowy programator/debugger UNI CODEGRIP marki MikroElektronika (http://t.ly/7R7Y-)



Rysunek 2. Opis wyprowadzeń złączy modułów rozszerzeń w standardzie mikroBUS (http://t.ly/Pk8cc)



Rysunek 3. Najważniejsze wymiary modułu/gniazda w standardzie mikroBUS (http://t.ly/Pk8cc)



Fotografia 4. Portfolio marki MikroElektronika obejmuje obecnie ponad 1500 modułów Click board, kompatybilnych ze standardem mikroBUS stosowanym w uniwersalnych zestawach ewaluacyjnych 7. i 8. generacji tego samego producenta (http://t.ly/5QiFB)

Platformy do prototypowania modułowego

Sporym zainteresowaniem inżynierów cieszą się rozmaite platformy sprzętowe, przeznaczone do budowy prototypów systemów embedded przy pomocy łatwych w użyciu modułów czujnikowych, komunikacyjnych i wykonawczych, łączonych ze wspólną płytą bazową. Choć opis ten wielu osobom skojarzy się zapewne w pierwszej

Fotografia 5. Płyta bazowa 7. generacji – EasyMx PRO v7 for STM32 marki MikroElektronika (http://t.ly/9ospP)

Fotografia 6. Płyta bazowa 8. generacji – Fusion marki MikroElektronika, przeznaczona do pracy z mikrokontrolerami PIC, dsPIC, PIC24 oraz PIC32 (http://t.ly/P7JLj)

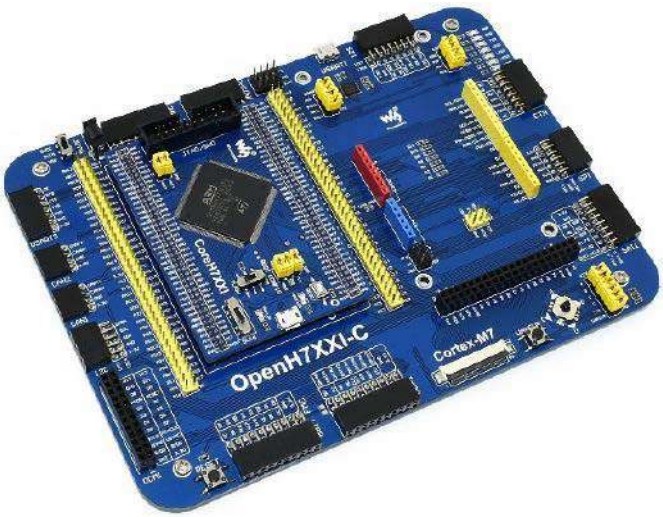
Fotografia 7. Moduły programatora/debuggera oraz zasilacza na płycie Fusion marki MikroElektronika (http://t.ly/P7JLj)

eprasa.pl 65e3eb482b

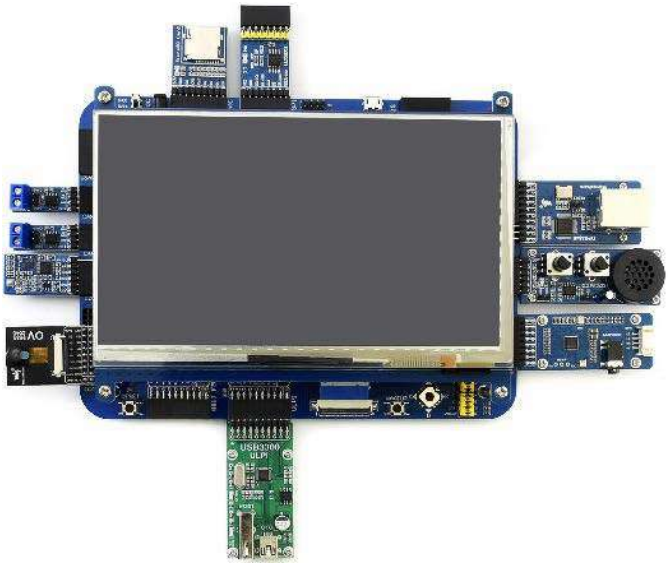
Fotografia 8. Zintegrowane podpórki przymocowane do płyty z serii Fusion marki MikroElektronika (http://t.ly/P7JLj)

chwili z popularnymi płytkami Arduino oraz nakładkami Arduino Shield, to w zastosowaniach komercyjnych na prowadzenie wysuwa się znów firma MikroElektronika. Opracowany przez nią otwarty standard mikroBUS (**rysunki 2** i **3**) stanowi fundament potężnego ekosystemu deweloperskiego, zawierającego obecnie przeszło 1500 modułów Click board (**fotografia 4**) oraz szeroki wachlarz płyt bazowych. Do tych ostatnich zaliczyć można zarówno rozbudowane płyty nieco starszej, 7. generacji (**fotografia 5**), jak i ultra-nowoczesne, w pełni profesjonalne zestawy ewaluacyjne 8. generacji (**fotografia 6**). Te ostatnie można już określić mianem „klasy luksusowej" – doskonałe wyposażenie w postaci bogatego zestawu bloków peryferyjnych, przycisków, złączy komunikacyjnych oraz slotów pod moduły rozszerzeń, zostało bowiem uzupełnione przez szereg konstrukcyjnych „smaczków". Zintegrowany programator/debugger Wi-Fi, a także moduł pełniący funkcję zasilacza UPS, są ukryte pod specjalnymi osłonami (**fotografia 7**), zaś wygodną pracę z płytą umożliwia system zintegrowanych podpórek (**fotografia 8**). Elastyczność aplikacyjną zapewnia system wymiennych kart z różnymi typami mikrokontrolerów, łączonych z płytą bazową przy pomocy złączy typu Mezzanine.

Konstruktorzy poszukujący prostszego, budżetowego narzędzia, także zapewniającego wygodny dostęp do linii GPIO oraz możliwość rozszerzania systemu o dodatkowe moduły, mogą skorzystać z oferty marki Waveshare. Opracowany przez nią ekosystem STM32 Open składa się z płyt bazowych z wymiennymi modułami „procesorowymi" (**fotografia 9**) oraz całkiem obszernej serii modułów dodatkowych, wpinanych w odpowiednie gniazda na brzegach płytki (**fotografia 10**). Firma oferuje nie tylko płyty bazowe, moduły



Fotografia 9. Płyta bazowa z serii STM32 Open marki Waveshare z zainstalowanym modułem zawierającym mikrokontroler (http://t.ly/91LdE)



Fotografia 10. Płyta STM32 Open z zamontowanym wyświetlaczem graficznym i szeregiem modułów dodatkowych (http://t.ly/91LdE)

z mikrokontrolerami oraz płytki rozszerzeń, ale także rozmaite modele wysokiej jakości wyświetlaczy graficznych, zaś całość uzupełniają repozytoria typu wiki, w których znajdują się liczne programy przykładowe, biblioteki oraz tutoriale ułatwiające rozpoczęcie pracy z ekosystemem Waveshare.

Podsumowanie

W artykule zaprezentowaliśmy kilka wybranych narzędzi przydatnych programistom i konstruktorom systemów embedded. Własne konstrukcje zestawów ewaluacyjnych, programatorów/debuggerów oraz rozmaitych narzędzi pomocniczych ma w swojej ofercie każdy producent mikrokontrolerów, zaś całości dopełniają uniwersalne platformy prototypowe, oferowane przez partnerów półprzewodnikowych potentatów – oraz rozmaite firmy niezależne. Warto zatem śledzić ten dynamiczny segment rynku, gdyż odpowiedni dobór narzędzi sprzętowych może niebywale poprawić komfort pracy, skrócić czas budowy prototypu, a w efekcie – przyspieszyć wprowadzenie produktu na rynek.

inż. Przemysław Musz, EP

REKLAMA

eprasa.pl 65e3eb482b

Przegląd rynku mikrokontrolerów 2024

W dzisiejszym świecie mikrokontrolery stanowią fundament wielu urządzeń, od prostych sprzętów codziennego użytku, po zaawansowane systemy automatyzacji i robotyki. Branża ta cały czas się rozwija i na rynku pojawiają się nowe układy. Niektóre z nich przechodzą inkrementalny rozwój, inne są propozycjami wręcz przełomowymi, a jeszcze inne dostarczają znanych od lat, niezmiennych możliwości, ale przy niższym zużyciu energii czy niższej cenie. Przyjrzyjmy się, jakiego rodzaju układy wprowadzono w ostatnim czasie na rynek.

Mikrokontrolery to niewielkie, specjalizowane układy scalone, które odgrywają kluczową rolę w sterowaniu różnymi funkcjami elektronicznymi. Ich wszechstronne zastosowanie obejmuje obsługę czujników, wykonywanie obliczeń, kontrolę włączania i wyłączania elementów wykonawczych oraz komunikację z innymi urządzeniami. Mikrokontroler łączy w sobie procesor, pamięć programu, pamięć danych, różne interfejsy wejścia/wyjścia oraz inne funkcjonalności, takie jak przetworniki analogowo-cyfrowe (ADC) i cyfrowo-analogowe (DAC).

Omawiany rodzaj układów scalonych zaprojektowany został do wykonywania pewnych z góry określonych zadań, przy ograniczonych zasobach – np. w układach o niewielkich rozmiarach i relatywnie małym zużyciu energii. Taka uniwersalność po części wynika z elastyczności, jaką stwarza programowalność, a po części z ogromnej liczby różnorodnych mikrokontrolerów dostępnych na rynku. Niektóre z nich są przeznaczone do ogólnych zastosowań, ale istnieje wiele układów, które dedykowane są do realizacji zadań w bardzo wąskiej niszy. Mikrokontrolery wykorzystywane są zatem do sterowania szeroką gamą konstrukcji, od systemów automatyki domowej, przez aplikacje medyczne, samochodowe systemy sterowania, po elektronikę przenośną, AGD czy urządzenia audio/wideo. Można je znaleźć w niemalże każdym urządzeniu elektronicznym, jakie mamy w domu.

W tym artykule skupimy się na najnowszych trendach w sektorze mikrokontrolerów oraz możliwościom, jakie oferują nowe układy. Przyjrzyjmy się, jakie rodziny mikrokontrolerów zadebiutowały w ostatnim czasie, a także: jakiego rodzaju mikrokontrolery zyskują, a jakie tracą na popularności, ze względu na kierunek rozwoju branży.

8-bitowce wiecznie żywe

Układy AVR to rodzina ośmiobitowych mikrokontrolerów, stworzona przez firmę Atmel. Obecnie produkowane są przez Microchipa, który w 2016 roku kupił Atmela. AVRy obecne są na rynku od 1997 roku i zdobyły od tego czasu ogromną popularność, głównie dzięki swojej prostej budowie i łatwemu programowaniu. Do dalszego ich upowszechniania przyczyniło się wykorzystanie układów z tej rodziny w szalenie popularnych wśród hobbystów modułach Arduino.

Rysunek 1. Flagowy przedstawiciel rodziny AVR DD – układ AVR32DD14

AVRy nie ograniczają jednak się do zastosowań hobbystycznych. Układy te często stosowane w urządzeniach konsumenckich czy przemysłowych, tym bardziej, że w ofercie firmy dostępnych jest wiele linii AVR, wyposażonych w funkcje dostosowane do bardzo wyspecjalizowanych wymagań.

Podobnie jest z nową rodziną układów AVR DD, opracowanych do zastosowań ogólnych i przemysłowych (oferują one bowiem wsparcie norm bezpieczeństwa funkcjonalnego).

Rodzina mikrokontrolerów AVR DD bazuje na znanym rdzeniu AVR ze sprzętowym mnożnikiem działającym w dwóch cyklach zegara, pracujący z prędkością taktowania do 24 MHz. Rdzeń ten pozwala na dostęp do wyjść i wejść w jednym cyklu zegara, ma wbudowany dwupoziomowy kontroler przerwań. Mikrokontrolery AVR DD wyposażane są w pamięć Flash o pojemności 16, 32 lub 64 kB, SRAM o pojemności 2, 4 lub 8 kB oraz EEPROM o pojemności 256 bajtów, w zależności od układu – i są dostępne w obudowach 14-, 20-, 28- oraz 32-pinowych (VQFN, TQFP, SSOP, SOIC, jak i SPDIP). Zasilane są napięciem w zakresie od 1,8 V do 5,5 V.

Producent określa wytrzymałość pamięci na zapis i wymazywanie: dla Flash jest to 10 000 cykli, a dla pamięci EEPROM: 100 000 cykli. Retencja danych wynosić ma 40 lat w temperaturze 55°C. Podawanie tych informacji na pierwszej stronie karty katalogowej nie jest typowe, ale powinno być zrozumiałe, jeśli zwrócimy uwagę na informacje o wsparciu dla norm IEC 61508 oraz ISO 26262. Układ wspiera dodatkowo obliczenia CRC dla pamięci Flash oraz szereg typowych zabezpieczeń, takich jak watchdog, układ Power-on Reset (POR), czy detektor zapadu napięcia (BOD).

Nowe układy programowane są przy pomocy jednoliniowego, zintegrowanego interfejsu do programowania i debugowania (UPDI), podobnie jak wszystkie inne nowoczesne mikrokontrolery AVR.

Opisywaną rodzinę układów mikrokontrolerów wyposażono w zintegrowane bloki CIP. Elementy te redukują obciążenie procesora dzięki wykorzystaniu rozbudowanego systemu obsługi zdarzeń i konfigurowalnych peryferiów niestandardowych (CCL). MVIO rodziny AVR DD umożliwia dwukierunkową komunikację z układami pracującymi w innej w domenie zasilania, co eliminuje konieczność stosowania zewnętrznych konwerterów poziomów logicznych. Elastyczność nowych mikrokontrolerów sprawia, że są doskonałe do zastosowań wymagających sterowania w czasie rzeczywistym w systemach kontroli procesów przemysłowych, urządzeniach gospodarstwa domowego, instalacjach samochodowych czy urządzeniach Internetu Rzeczy (IoT). Mikrokontrolery AVR DB są dobrze radzą sobie też z przetwarzaniem sygnałów – wyposażone są w trzy wbudowane wzmacniacze operacyjne, 12-bitowy przetwornik ADC, 10-bitowy przetwornik DAC, a także układ detekcji przejścia przez zero i komparator analogowy. Peryferia te ułatwiają m.in. budowę interfejsów użytkownika opartych na pojemnościowej technologii dotykowej.

Wszystkie mikrokontrolery AVR są w pełni obsługiwane przez kompleksowy ekosystem rozwojowy, obejmujący środowisko MPLAB X i Studio – bezpłatne IDE Microchipa z wbudowanym kompilatorem GCC oraz potężne narzędzia konfiguracji kodu MPLAB Code

eprasa.pl 65e3eb482b

Rysunek 2. AVR4DD32 Curiosity Nano Evaluation kit firmy Microchip

Configurator (MCC) i START, generujące fabrycznie zwalidowany kod w C, który pomoże w poprawnym rozpoczęciu każdego projektu. Szybkie prototypowanie urządzeń z mikrokontrolerami AVR DD ułatwia zestaw ewaluacyjny AVR4DD32 Curiosity Nano Evaluation (patrz **rysunek 2**).

Microchip oferuje również licencję na MPLAB® XC8 Functional Safety Compiler, która jest certyfikowanym przez TÜV SÜD pakietem kompilatora, obsługującym mikrokontrolery 8-bitowe PIC i AVR.

Popularne STM32 w nowej odsłonie

W 2023 roku firma STMicroelectronics wprowadziła do swojej oferty dwie nowe linie układów z rdzeniem ARM: STM32U5 oraz STM32C0. Różnią one się diametralnie zarówno parametrami, jak i docelowym zakresem aplikacji.

Seria STM32U5 to pierwsze mikrokontrolery ogólnego przeznaczenia, które otrzymały certyfikat NIST dla ich wbudowanego źródła entropii – generatora prawdziwie losowych liczb, przeznaczonego do wsparcia algorytmów szyfrowania. Certyfikat ten upraszcza i przyspiesza wdrożenie aplikacji, które potrzebują końcowej certyfikacji SP800-90B.

Nowe mikrokontrolery rozszerzają pojemność pamięci Flash do 4 MB i SRAM do 3 MB. Dzięki bogatemu wyposażeniu

Rysunek 3. Zestaw deweloperski Discovery z układem STM32U-5A9NJH6Q do testowania możliwości graficznych układu

peryferyjnemu (obejmującemu nawet akcelerator graficzny 2,5D) nowe mikrokontrolery idealnie nadają się do aplikacji, takich jak sensory środowiskowe, napędy przemysłowe, automatyzacja budynków, inteligentne urządzenia AGD, urządzenia przenośne, sterowanie e-mobilnością i inne rozwiązania, zwłaszcza te stosowane w odległych, trudno dostępnych miejscach. Ponieważ miliardy takich urządzeń są wdrażane na całym świecie w miarę upowszechniania inteligentnych rozwiązań do pracy i codziennego użytku, te właśnie

REKLAMA

mikrokontrolery mają szansę zrewolucjonizować wiele aspektów naszego życia, poprawiając sprawność energetyczną urządzeń i wzmacniając bezpieczeństwo cybernetyczne. Seria STM32U5 wykorzystuje najnowszą ich generację rdzenia ARM – Cortex-M33 – która wprowadziła ulepszenia zwiększające wydajność, sprawność oraz odporność na ataki hackerskie.

Nowe funkcje oszczędzania energii, obecne w układach z serii STM32U5, obejmują autonomiczne peryferia i tryb autonomiczny o niskim poborze mocy (LPBAM). LPBAM pozwala aplikacji zachować krytyczną funkcjonalność, podczas gdy rdzeń i inne nieużywane bloki przechodzą w dowolny z elastycznych trybów oszczędzania energii. Z tego stanu mikrokontroler może szybko wybudzić rdzeń, aby efektywnie przetworzyć partię danych, a następnie przejść z powrotem w tryb energooszczędny.

Seria STM32U5 wykracza poza typowe ograniczenia wydajności graficznej, które zwykle dotyczą oszczędnych energetycznie mikrokontrolerów. Układy z zaawansowaną jednostką przetwarzania grafiki (GPU) NeoChrom mogą obsługiwać rozbudowany interfejs graficzny, co wcześniej było możliwe tylko w drogich systemach opartych na dużych mikroprocesorach. Teraz mały mikrokontroler może zapewnić doświadczenia użytkownika podobne do tych znanych ze smartfonów, a w ramach rozwoju interfejsu graficznego można wykorzystać framework TouchGFX, który obecnie obsługuje format SVG i bogate zasoby graficzne.

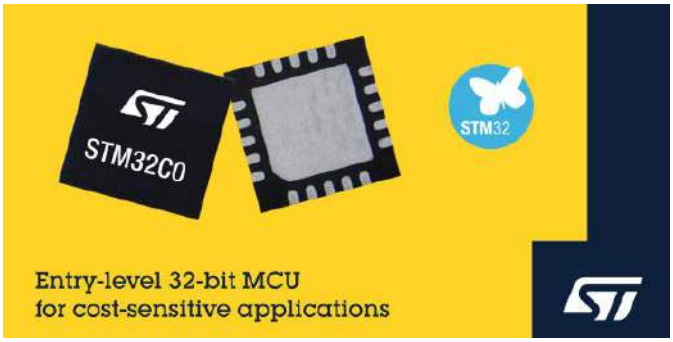
Mikrokontrolery STM32U5 są dostępne w kompaktowej obudowie LQFP100, co pozwala na uproszczenie konstrukcji PCB przy minimalnej liczbie warstw. Konstruktorzy mogą szybko dostarczać nowe projekty, korzystając z zasobów, takich jak pakiety oprogramowania STM32CubeU5, nowe płytki deweloperskie NUCLEO-U545RE i NUCLEO-U5A5ZJ oraz zestaw demonstracyjny STM32U5A9J-DK.

Seria STM32U5 wzmacnia bezpieczeństwo cybernetyczne dzięki rdzeniowi Cortex-M33 z jednostką ochrony pamięci i architekturze TrustZone firmy ARM. Mikrokontrolery te wyposażone są również w akceleratory kryptograficzne do obsługi zaawansowanych algorytmów AES i wsparcie architektury klucza publicznego (PKA), zapewniają też odporność na ataki fizyczne. Obsługa kodów korekcji błędów (ECC) w pamięciach Flash i SRAM zapobiegają jednocześnie uszkodzeniom danych.

STMicroelectronics poszerzył także swoją linię ekonomicznych mikrokontrolerów, wprowadzając przystępną cenowo serię STM32C0. Firma kieruje ją do producentów m.in. urządzeń AGD, pomp przemysłowych, wentylatorów, czujników dymu itp., w których zazwyczaj wykorzystywane są prostsze mikrokontrolery 8-bitowe i 16-bitowe. Nowoczesny, 32-bitowy design STM32C0 pozwala na wprowadzenie ulepszeń przy zachowaniu zbliżonego kosztu oraz zużycia energii. Jednocześnie prostota omawianych komponentów i bezpłatny, rozbudowany ekosystem deweloperski (obejmujący narzędzia sprzętowe oraz pakiety oprogramowania) ułatwiają nowym użytkownikom wejście do świata 32-bitowych mikrokontrolerów. Seria STM32C0 jest objęta 10-letnim programem długowieczności ST dla produktów przemysłowych, aktualizowanym w cyklu rocznym.

ST zaprojektowało nową serię STM32C0 tak, aby projektanci mogli zdziałać więcej przy mniejszym wysiłku: mikrokontrolery umieszczone są w małych i ekonomicznych obudowach z tylko jedną parą pinów zasilania, co pozwala na wykorzystanie większej liczby linii I/O na potrzeby aplikacji, przy jednoczesnym obniżeniu kosztów materiałowych systemu. Ponadto na pokładzie znajduje się dokładny (1%) oscylator, co eliminuje potrzebę stosowania zewnętrznych układów taktowania. Ze względu na potrzebę zachowania „ciągłości produktowej" STM32, układy STM32C0 umożliwiają łatwą migrację do serii STM32G0 przy zachowaniu spójnego układu wyprowadzeń.

Seria STM32C0 integruje różne peryferia, w tym ADC z szybkością próbkowania do 1,7 MSps i sprzętową obsługą nadpróbkowania 16-bitowego, a także wiele timerów, w tym zaawansowany timer kontrolny do np. sterowania silnikiem. Dzięki rdzeniowi ARM Cortex-M0+, taktowanemu zegarem do 48 MHz, układy te osiągają do 44 DMIPS i uzyskały CoreMark o wartości 114, pobierając prąd zasilania na poziomie 80 μA/MHz.



Rysunek 4. Zapowiedź rodziny mikrokontrolerów STM32C0

Seria STM32C0 ułatwia projektantom wejście w świat cenionego ekosystemu rozwojowym STM32. Dostępna jest tania płytka deweloperska STM32C0 Nucleo (NUCLEO-C031C6) oraz dwa zestawy demonstracyjne STM32C0 Discovery (STM32C0116-DK i STM32C0316-DK). Zasoby oprogramowania obejmują narzędzie konfiguracyjne STM32CubeMX, wsparcie dla Microsoft Azure RTOS oraz pakiet STM32CubeC0 z warstwą abstrakcji sprzętowej (HAL) i bibliotekami niskopoziomowymi.

Nowi członkowie rodziny PIC

Mikrokontrolery PIC firmy Microchip są bodajże najstarszą z prezentowanych tutaj rodzin mikrokontrolerów. Pierwszymi układami z tej rodziny były PIC1650 i PIC1655. Przez wiele lat układy PIC ewoluowały, by stać się finalnie jedną z najpopularniejszych i wszechstronnych rodzin mikrokontrolerów na świecie. Większość procesorów PIC to układy 8-bitowe, choć w ofercie Microchip znajdują się też układy 16-bitowe (PIC24 oraz dsPIC30) oraz 32-bitowe (PIC32MX). W ostatnim czasie rodzina PIC powiększyła się o cztery układy 8-bitowe: PIC16F13145, PIC16F171, PIC16F181 oraz PIC18Q71.

PIC16F131 to kolejny krok w wykorzystaniu programowalnej logiki w mikrokontrolerach. Omawiana rodzina 8-bitowców ma wbudowane nowe moduły logiki konfigurowalnej (CLB), oferujące rozwiązania sprzętowe na miarę potrzeb i eliminujące konieczność stosowania zewnętrznych komponentów logicznych. Mikrokontrolery te wyposażono również w niezależne moduły peryferiów (CIP), co w połączeniu z CLB umożliwia w pełni sprzętową realizację rozmaitych funkcji na bazie logiki kombinacyjnej, bez potrzeby angażowania samego procesora do tego celu. Proces opracowywania oprogramowania jest dodatkowo uproszczony dzięki narzędziu graficznemu, które pomaga syntetyzować niestandardowe projekty logiczne do implementacji w CLB. Rodzina PIC16F131 została zaprojektowana do aplikacji wykorzystujących niestandardowe protokoły czy też sterujących procesami czasie rzeczywistym, np. w przemyśle czy w motoryzacji.

Dzięki CLB mikrokontrolery wkraczają w teren zdominowany dotychczas przez układy programowalne (FPGA), umożliwiając użycie tych mikrokontrolerów w nowych obszarach rynku. Ponieważ działanie CLB nie zależy od prędkości zegara, prezentowane rozwiązanie wyróżnia się niskim zużyciem energii. CLB może być używany do podejmowania prostych decyzji podczas uśpienia procesora, co dodatkowo zwiększa energooszczędność. Mikrokontrolery z tej rodziny wyposażone zostały ponadto w szybki, 10-bitowy przetwornik ADC, 8-bitowy DAC, komparatory, timery oraz moduły komunikacji szeregowej (I^2C i SPI), dzięki czemu mogą wykonywać wiele zadań na poziomie systemowym, bez użycia CPU. Rodzina ma być dostępna w różnych obudowach – od 8 do 20 pinów. Układy te wyposażono w maksymalnie 24-kilobajtową pamięć. Maksymalna częstotliwość taktowania to 32 MHz.

PIC16F171 to z kolei nowa rodzina mikrokontrolerów wyposażonych w rozbudowany zestaw analogowych peryferiów, umożliwiających zastosowania układu np. w aplikacjach czujnikowych. Układy dostępne są w obudowach z wyprowadzeniami w liczbie od 8 do 44,

eprasa.pl 65e3eb482b

w wersjach z pamięcią Flash 7 kB do 28 kB i taktowaniem do 32 MHz. Mikrokontrolery te zawierają wbudowany niskoszumny wzmacniacz operacyjny, 12-bitowy ADC z możliwością pracy w trybie różnicowym, dwa 8-bitowe przetworniki DAC, do czterech 16-bitowych peryferiów dedykowanych do generacji sygnałów PWM i wiele innych peryferiów przeznaczonych do sterowania i komunikacji. Podobny charakter ma rodzina mikrokontrolerów PIC16F18146 – zawierająca dodatkowo wydajny zestaw niezależnych od rdzenia peryferiów (CIP), predysponujący ją do projektów systemów o ograniczonych rozmiarach, które pracują m.in. z wieloma czujnikami. Dzięki 12-bitowemu ADC z wbudowaną opcją wykonywania prostych obliczeń, dwóm 8-bitowym DAC i kilku interfejsom komunikacyjnym, omawiana rodzina pozwala na łączenie różnych rodzajów czujników w jednym projekcie. Mikrokontrolery PIC16F171 mogą działać jako prcesory towarzyszące w bardziej złożonych systemach, takich jak automatyka domowa, sterowanie procesami przemysłowymi, motoryzacja czy Internet rzeczy (IoT).

Rodzina układów PIC18-Q71 łączy w sobie potężne CIP z wysokim poziomem integracji bloków analogowych, co ma w zamyśle uprościć prowadzenie pomiarów, a jednocześnie optymalizować wydajność systemu oraz obniżać koszty urządzenia. Mikrokontrolery te zawierają 12-bitowy, różnicowy ADC, konfigurowalne wzmacniacze operacyjne, szybkie komparatory analogowe oraz 8-bitowy port do routingu sygnałów z peryferiów cyfrowych. Dodatkowo na pokładzie znalazły się dwa zintegrowane, 16-bitowe modulatory PWM, 20-bitowy oscylator programowalny oraz 32-bitowy licznik uniwersalny. Bogaty zestaw bloków CIP – w połączeniu z menedżerem peryferiów analogowych (APM) – umożliwia prostą konfigurację bloków analogowych oraz szybkie reagowanie na zdarzenia systemowe. Nowa rodzina produktów dostępna jest w różnych opcjach obudów i wersjach pamięci do wielu zastosowań, w tym oświetlenia LED, systemów konserwacji predykcyjnej, urządzeń medycznych, automatyki domowej, sterowania procesami przemysłowymi, motoryzacji i Internetu Rzeczy (IoT).

Wszystkie nowe mikrokontrolery PIC programowane mogą być przy użyciu zintegrowanego środowiska MPLAB X. Ten darmowy pakiet IDE to oprogramowanie o dużej elastyczności, zawierające potężne narzędzia wspierające konfigurację, rozwijanie i debugowanie mikrokontrolerów Microchipa. Oprócz samego IDE, firma oferuje wiele darmowych i płatnych akcesoriów w ramach ekosystemu MPLAB. Dostępne są również płytki rozwojowe do omawianych mikrokontrolerów, które pozwalają na szybką ocenę przydatności tych układów w naszych zastosowaniach.

Mikrokontrolery ze wsparciem dla uczenia maszynowego

Firma Infineon wprowadziła w zeszłym roku nową rodzinę mikrokontrolerów do portfolio znanych układów PSoC (Programmable System on a Chip), opartych na architekturze ARM Cortex. PSoC Edge został zaprojektowany z myślą o nowej generacji aplikacji związanych ze sterowaniem i przetwarzaniem danych, dlatego oferuje sprzętowe wsparcie uczenia maszynowego (ML). Omawiana rodzina układów zapewnia nowy poziom doświadczeń użytkowników, ponieważ umożliwia tworzenie bardziej inteligentnych produktów o wyższej użyteczności, a ponadto obniża barierę interakcji między ludźmi a maszynami i dodaje świadomość kontekstu do aplikacji końcowych. Czyni to przy zachowaniu wysokiego poziomu ochrony prywatności i bezpieczeństwa, dzięki zastosowaniu wbudowanych technologii Infineon Edge Protect.

Układy PSoC Edge są oparte na wydajnym rdzeniu ARM Cortex-M55 ze wsparciem dla DSP Helium w połączeniu z ARM Ethos-U55 – lub na rdzeniu ARM Cortex-M33 w połączeniu z modułem NNLite od Infineon o niskim zużyciu energii. NNLite to własny akcelerator sprzętowy Infineona, przeznaczony do przyspieszania obliczeń sieciach neuronowych, wykorzystywanych w uczeniu maszynowym i aplikacjach AI. Wsparcie włączonych przez cały czas czujników i gwarancja szybkiej odpowiedzi sprawiają, że układy te są idealne do zaawansowanych segmentów IoT oraz przemysłowych, takich jak inteligentny dom, systemy bezpieczeństwa, urządzenia noszone, robotyka i wiele innych. Rodzina procesorów PSoC Edge charakteryzuje się dużą ilością pamięci wbudowanej, w tym nieulotną pamięcią RRAM oraz obsługą szybkiej, bezpiecznej pamięci zewnętrznej. Układy oferują skalowalność w coraz bardziej zaawansowanych, inteligentnych systemach nowej generacji, a jednocześnie zapewniają przenośność oprogramowania.

Możliwości obliczeniowe rodziny PSoC Edge rozszerzają wsparcie zaawansowanych aplikacji graficznych, głosowych, dźwiękowych i opartych na widzeniu maszynowym na istniejące portfolio PSoC. Nowe układy sprzyjają łatwej migracji aplikacji w obrębie rodziny oraz ścieżkę aktualizacji z istniejących konstrukcji (korzystających ze starszych generacji PSoC).

Wspierani przez bogaty zestaw narzędzi, programiści mogą wykorzystać zalety PSoC Edge do swoich obecnych i przyszłych potrzeb projektowych. Dzięki kompleksowej dokumentacji i uznanemu w branży oprogramowaniu ModusToolbox (z integracją z systemem AI Imagimob Studio), PSoC Edge skraca czas wprowadzenia nowych produktów na rynek.

Platforma oprogramowania ModusToolbox firmy Infineon zapewnia zbiór narzędzi deweloperskich, bibliotek i zasobów uruchomieniowych do systemów wbudowanych, umożliwiających

REKLAMA

IoT w Twoim domu!

Wireless Gecko Series 2
System-on-Chip

Najważniejsze protokoły radiowe IoT – pre-certyfikowane moduły i skalowalna seria układów System-on-Chip (SoC)

- Wieloprotokołowy SoC z ZigBee, Thread, Bluetooth Mesh i Matter
- SoC dla Bluetooth Low-Energy i Bluetooth Mesh

Myślisz o projekcie urządzenia końcowego, bramie, węźle low-power czy urządzeniu wielo-protokołowym? Zacznij od GLYN-SUPPORT:

www.glyn.com | biuro@glyn.pl

FIRST CLASS PREMIUM SUPPORT
GLYN
High-Tech Distribution

eprasa.pl 65e3eb482b

elastyczne i wszechstronne działania deweloperskie. ModusToolbox obsługuje szeroki zakres aplikacji, w tym systemy konsumenckie, IoT, aplikacje przemysłowe, systemy inteligentnego domu, urządzenia noszone i inne. Imagimob Studio to platforma do rozwoju Edge AI, zintegrowana z ModusToolbox, która umożliwia kompleksowy rozwój ML, od akwizycji danych do wdrożenia kompletnego modelu ML. Gotowe projekty przykładowe i modele Imagimob sprawiają, że rozpoczęcie pracy z uczeniem maszynowym jest bardzo proste. Wspólnie z PSoC Edge, Imagimob umożliwia wdrożenie zaawansowanych modeli uczenia maszynowego w zastosowaniach brzegowych, z wykorzystaniem wsparcia akceleratorów, oferowanych w tej rodzinie mikrokontrolerów.

ARM to nie wszystko

W sektorze mikrokontrolerów zauważyć można pewną dominację układów z rdzeniem ARM, z wyjątkiem opisanych wcześniej procesorów 8-bitowych. Nie oznacza to jednak, że na rynku nie ma obecnie innych rozwiązań. Są one cały czas rozwijane, czego przykładem mogą być nowe produkty firmy Renesas.

Wprowadzona do sprzedaży we wrześniu zeszłego roku nowa rodzina mikrokontrolerów RL78/G24 firmy Renesas to układy, które charakteryzują się najwyższą wydajnością spośród wszystkich mikrokontrolerów w rodzinie RL78. Ta popularna linia produktowa obejmuje MCU 8- i 16-bitowe, przeznaczone do aplikacji energooszczędnych. Dzięki dedykowanemu akceleratorowi FAA oraz szybkiemu procesorowi, zdolnemu do osiągnięcia częstotliwości pracy do 48 MHz, układ ten może zaoferować bardzo wiele, nawet przy swoim niezwykle małym poborze prądu.

Nowy mikrokontroler wzbogacony został o poszerzony zestaw peryferiów, w tym nowe funkcje analogowe i zegarowe, dzięki czemu idealnie sprawdza się w aplikacjach sterowania silnikami, zasilaniem czy oświetleniem. Dzięki FAA układ może ponadto sprawnie i niezależnie od procesora dystrybuować zadania do peryferiów (sterowanie falownikiem, szyfrowanie, pomiary czy proste operacje arytmetyczne), co znacząco zwiększa osiągi procesora.

Mikrokontrolery z omawianej serii wyposażono w 12-bitowe ADC z trzema kanałami o jednoczesnym próbkowaniu. Przetwornik oferuje wbudowane funkcje przetwarzania danych z szybkością do 1 μs na próbkę. Na pokładzie znalazł się też szybki komparator (do 50 ns opóźnienia). Mikrokontrolery z omawianej rodziny wyposażono w DALI-2 – standardowy interfejs do komunikacji z systemami oświetlenia. Układy te dostępne są w opcji z 64 lub 128 kB pamięci programu, 12 kB RAM, od 16 do 58 GPIO w obudowach QFN, QFP, SSOP i LGA. Najmniejszy z dostępnych układów w rodzinie mierzy zaledwie 3 mm × 3 mm.

Renesas dostarcza również od pewnego czasu moduł RX23W – z pełnym wsparciem dla Bluetooth 5.0 Low Energy, przeznaczony do sterowania i komunikacji bezprzewodowej w urządzeniach końcowych IoT. Wyposażony jest on w 32-bitowy rdzeń taktowany częstotliwością do 54 MHz, zintegrowane peryferia obsługujące komunikację Bluetooth Low Energy, antenę, oscylator i inne kluczowe

Rysunek 5. Mikrokontrolery z rodziny RL78/G24

Rysunek 6. Mikrokontrolery z rodziny RX32W

komponenty toru radiowego. Moduł jest certyfikowany zgodnie z wymaganiami Bluetooth SIG. Bardzo mała obudowa LGA (6,1 mm × 9,5 mm) o 83 pinach umożliwia projektowanie kompaktowych urządzeń, z mniejszą liczbą zewnętrznych komponentów, co zmniejsza koszty materiałowe systemu. Układ oferuje również funkcję bezpieczeństwa – Trusted Secure IP (TSIP) – która zapewnia ochronę przed zagrożeniami urządzeń IoT, takimi jak podsłuchiwanie, manipulacja czy wirusy. Ponadto mikrokontroler RX23W wyposażony jest w bogaty zestaw interfejsów, niezbędnych w wielu aplikacjach IoT – w tym wsparcie klawiszy dotykowych, USB i CAN.

Do programowania nowego modułu RX23W przeznaczone jest to samo środowisko deweloperskie, co w przypadku innych mikrokontrolerów Renesas (RX, RL78 itd.), czyli Renesas CS+. Wbudowany Smart Configurator umożliwia automatyczne generowanie kodu dla peryferiów i Bluetooth za pomocą GUI, a także konfigurowanie ustawień pinów. Środowisko wspiera niestandardowe profile Bluetooth, z kolei Bluetooth Test Tool Suite dostarcza graficznego narzędzia do wstępnej oceny systemów bezprzewodowych i weryfikacji funkcjonalności BLE.

NXP dla przemysłu i systemów Internetu Rzeczy

Firma NXP wprowadziła na rynek na początku tego roku nową serię mikrokontrolerów MCX – rodziny A14x oraz A15x. Są to uniwersalne mikrokontrolery, będące częścią większej rodziny MCX, która współdzieli ten sam rdzeń ARM Cortex-M33. Rodzina MCX powstała z myślą o połączeniu najlepszych elementów popularnych układów NXP z nowymi, innowacyjnymi funkcjami.

Seria MCX A to mikrokontrolery wielozadaniowe zaprojektowane do obsługi szerokiego spektrum zastosowań, oferujące skalowalność, niskie zużycie energii oraz inteligentne peryferia. Innowacyjna architektura zasilania ma na celu wsparcie efektywnego wykorzystania wejść/wyjść oraz wysokiej sprawności energetycznej przy prostym obwodzie zasilania, możliwym do zrealizowania w mniejszym niż dotychczas formacie. Przeznaczona do obsługi większej liczby pinów GPIO, seria MCX A pozwala projektantom korzystać z mniejszej obudowy, prostszego projektu płytki i niższych kosztów materiałów, a dodatkowo jest wspierana przez środowisko deweloperskie MCUXpresso.

Skalowalność pozostaje ważnym filarem serii MCX. Rodzina układów MCX A odgrywa w tym portfolio znaczącą rolę, jako podstawowa seria układów wielozadaniowych. Jest ona przeznaczona do szerokiego spektrum zastosowań obejmujących wiele rynków, takich jak m.in. komunikacja przemysłowa, inteligentne pomiary, automatyzacja i sterowanie, czujniki oraz urządzenia o niskim poborze mocy i/lub zasilanie bateryjnie.

MCX A14x taktowany jest zegarem o częstotliwości 48 MHz, a MCX A15x – do 96 MHz. Układy występują w obudowach QFN32, QFN48 i LQFP64, a zgodność rozkładu wyprowadzeń z innymi produktami NXP upraszcza migrację oraz aktualizacje systemów. Peryferia są również wspólne, dzięki czemu tworzenie oprogramowania także wymaga mniej wysiłku. Maksymalnie 8 pinów zdolnych jest do obsługi sporego obciążenia (do 20 mA), a dodatkowo są one

eprasa.pl 65e3eb482b

odporne na na napięcia do 5 V, co w wielu sytuacjach niweluje potrzebę stosowania konwerterów poziomów logicznych.

MCX A używa podsystemu zasilania z LDO niewymagającym kondensatora na wyjściu i zdolnym do pracy w zakresie od 1,7 V do 3,6 V. Kluczową zaletą architektury zasilania MCX A jest jego wyjątkowo wysoka sprawność – 59 μA/MHz w trybie aktywnym (przy zasilaniu 3 V, w temp otoczenia równej 25°C), podczas uruchomienia z wewnętrznej pamięci Flash – lub 6,5 μA w trybie głębokiego uśpienia, z pełną retencją danych w pamięci SRAM i wybudzaniem w czasie 10 μs. Dodatkowo dostępny jest też prawdziwy tryb głębokiego uśpienia z wyłączoną pamięcią, który pozwala osiągnąć pobór prądu na poziomie poniżej 400 nA, niestety kosztem czasu wybudzania (który rośnie do 2,78 ms).

Mikrokontrolery MCX A integrują w sobie wszystkie najważniejsze peryferia, w tym UART, SPI, I^2C oraz kilka liczników, choć w strukturze układów znalazło się również kilka godnych uwagi dodatków, które wyróżniają go spośród konkurentów.

MIPI I^3C – następca I^2C pojawił się w tym układzie nieprzypadkowo. NXP aktywnie uczestniczy w wdrażaniu nowej magistrali MIPI I^3C w różnych rodzinach mikrokontrolerów. Interfejs ten został opracowany przez MIPI alliance, aby odświeżyć I^2C, zapewniając wyższą prędkość transmisji danych interfejsami dwuprzewodowymi. I^3C przedstawiany jest czasami jako alternatywa dla SPI. Wsparcie NXP dla MIPI I^3C otwiera nowe możliwości aplikacji, takich jak mostkowanie protokołu I^3C oraz inteligentne peryferia z szybkim interfejsem dwuprzewodowym.

Układy MCX A mają również kontroler USB Full Speed ze zintegrowanym PHY – do działania w roli USB Device. Istotną cechą podsystemu USB w MCX A jest programowanie w systemie (ISP) za pomocą boot ROM. Oznacza to, że produkty z tymi mikrokontrolerami mogą być aktualizowane za pomocą interfejsu USB.

Funkcje ISP są wbudowane w boot ROM MCX A. Błędne aktualizacje oprogramowania nie stanowią już problemu, ponieważ dzięki funkcji boot ROM istnieje opcja wgrania programowania zapasowego, gwarantująca działanie niezależnie od zawartości pamięci FLASH. Warto zauważyć, że funkcje ISP są dostępne również poprzez interfejsy UART, I^2C i SPI. Ułatwieniem w korzystaniu z funkcji boot ROM jest możliwość wykorzystania narzędzia MCUXpresso SEC, które pozwala na opracowanie własnych procesów programowania produkcyjnego i aktualizacji za pomocą intuicyjnego interfejsu graficznego, z możliwością wyjścia operacji wiersza poleceń dla skryptowania. SDK układów dostępne jest jako open-source, a specjalne SDK do Zabezpieczania Danych (SPSDK) umożliwia zaawansowanym użytkownikom dalsze dostosowanie operacji z wiersza poleceń, jeśli zachodzi taka potrzeba.

W architekturze pamięci MCX A można wyróżnić jeszcze dwie kolejne ważne cechy. Pierwsza to kontroler pamięci cache o niskim zużyciu energii (LPCAC), wspierający obsługę pamięci podręcznej o wielkości 4 kB i dołączony do magistrali ARM Cortex-M33. Dzięki LPCAC dane i instrukcje mogą być dostępne z zachowaniem niskiej latencji. Dodatkowo rośnie też dostępność magistrali dla innych peryferiów, takich jak DMA, ponieważ wydajność procesora może zostać rozprzęgnięta z wydajnością pamięci systemowej. Jeśli mechanizm pamięci podręcznej nie jest wymagany, pamięć LPCAC 4 kB może zostać przekształcona w zwykłą pamięć instrukcji. Krytyczny kod programu i wektory przerwań mogą być umieszczone w tej części pamięci dla zapewnienia przewidywalnego timingu w aplikacjach sterowania w czasie rzeczywistym.

Podsekcja RAM w MCX A ma możliwość skorzystania z ECC, która może być używana w połączeniu z modułem raportowania błędów (ERM) w przypadku aplikacji wymagających większej niezawodności.

MCX A zawiera ADC o prędkości 4 MSPS, a także szybkie komparatory ze zintegrowanym 8-bitowym DAC generującym napięcia odniesienia. Podsystem analogowy może być połączony z podsystemem kontroli silnika, który obejmuje z kolei peryferia FlexPWM.

Połączenie możliwości komunikacji z zaawansowanymi peryferiami analogowymi i podsystemem kontroli silnika pokazuje, że układy MCX A są idealne do zastosowań takich, jak sterowniki napędów BLDC (także z implementacją algorytmów FOC), sterowania precyzyjnych serwomechanizmów w aplikacjach pozycjonowania, ale także w układach rozproszonego zarządzania akumulatorami (BMS).

Dzięki środowisku deweloperskiemu MCUXpresso, wejście w ten ekosystem jest relatywnie nietrudne. NXP oferuje programistom wolność wyboru sposobu tworzenia oprogramowania. Pakiet MCUXpresso oferuje podstawowe zestawy narzędzi deweloperskich (SDK), zintegrowane środowisko programistyczne (IDE) i narzędzia do konfiguracji mikrokontrolera. SDK dla MCX A obejmuje również niskopoziomowe sterowniki peryferyjne, konfiguratory i oprogramowanie Middleware, takie jak np. stos dla urządzenia USB. Jest ono dostarczane na licencji open-source. SDK może być używane również z innymi IDE, np. Visual Studio Code, środowiskami opartymi na Eclipse IDE, IAR Embedded Workbench i ARM Keil MDK.

SDK MCX A zawiera przykłady wszystkich powszechnych przypadków implementacji USB. Warto wspomnieć, że MCUXpresso IDE zawiera zaawansowane narzędzie konfiguracyjne do generowania niestandardowego kodu do urządzeń USB poprzez zarządzanie kodem inicjalizacji i deskryptorami urządzenia.

Ciekawym aspektem dostarczanego pakietu narzędzi i SDK jest wsparcie dla nowoczesnych przepływów pracy CI/CD. Coraz więcej projektów systemów wbudowanych rozwijanych jest z zastosowaniem zwinnych metodologii i paradygmatów ciągłej integracji i ciągłego wdrażania. Projekty oparte na samym IDE są powszechne w przypadku rozwoju oprogramowania dla mikrokontrolerów, ale narzędzia dla MCX A obsługują również kompilacje z wiersza poleceń. Wewnątrz SDK znajdują się skrypty CMake, które można użyć w połączeniu z systemami Make lub Ninja.

Wraz z udostępnieniem serii MCX A, NXP ponownie wprowadza na rynek swoją sprawdzoną platformę rozwojową FRDM, tym razem rozbudowaną o nowe funkcje. Płytka rozwojowa FRDM-MCXA153 zapewnia wszystkie narzędzia do szybkiego prototypowania aplikacji opartych na tym nowym mikrokontrolerze.

Podsumowanie

Rynek mikrokontrolerów cały czas się zmienia. Mimo że obserwujemy na nim głównie znane i szeroko stosowane rodziny, to z każdym rokiem są one wzbogacane o kolejne układy z nowymi możliwościami. Opisane w powyższym artykule nowinki na rynku mikrokontrolerów pokazują, jakie są trendy całego rynku elektronicznego. Z jednej strony miniaturyzacja i redukcja poboru mocy przez elementy elektroniczne, z drugiej zaś – rozwój nowych możliwości, głównie w zakresie obsługi coraz większej liczby sensorów, integracji z systemami IoT, czy też – coraz częściej – wsparcia algorytmów AI.

Nikodem Czechowski, EP

Źródła

1. Microchip AVR® DD Product Brief
2. IEC 61508 Functional Safety Packages for AVR® DD Family, Microchip Release Note DS40002215A
3. https://tiny.pl/dmhzn
4. https://tiny.pl/dmh3w
5. https://tiny.pl/dmhz2
6. https://tiny.pl/dmhz8
7. https://tiny.pl/dmhzv
8. https://tiny.pl/dmh3f
9. https://tiny.pl/dmhzz
10. https://tiny.pl/dmhz3
11. https://tiny.pl/dmh3q
12. https://tiny.pl/dmh3x
13. https://tiny.pl/dmh3t
14. https://tiny.pl/dmh37

eprasa.pl 65e3eb482b

Fotografia 1. Panoramiczny wyświetlacz LCD 11,9" marki Waveshare z panelem ze szkła hartowanego, oferujący unikalną rozdzielczość 320×1480 px

Zestawy ewaluacyjne i inteligentne wyświetlacze w ofercie ELTY

– wsparcie prototypowania oraz produkcji elektroniki

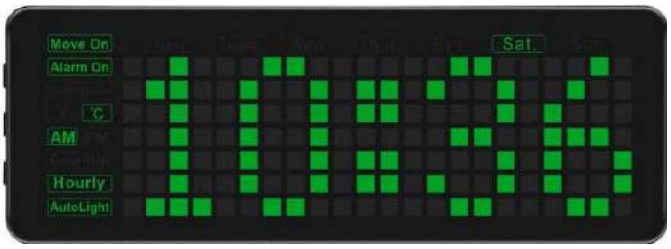
Praca nad urządzeniami mikroprocesorowymi nierozerwalnie wiąże się z koniecznością szybkiego i sprawnego tworzenia prototypów, na których uruchomione zostaną wszystkie najważniejsze bloki funkcjonalne docelowego produktu. Z drugiej strony, spore znaczenie dla redukcji czasu potrzebnego na wdrożenie danej aplikacji ma także zastosowanie gotowych rozwiązań, znacznie przyspieszających proces pisania oprogramowania wbudowanego. W artykule zaprezentujemy trzech producentów, którzy we wspomnianych obszarach rynku – prototypowaniu oraz usprawnieniu tworzenia rozbudowanych interfejsów HMI – mają szczególnie dużo do zaoferowania.

Waveshare – czołowy producent narzędzi deweloperskich

Firma Waveshare już od wielu lat wyznacza nowe standardy na rynku narzędzi dla elektroników i programistów systemów wbudowanych. W ofercie producenta znajdują się setki zestawów ewaluacyjnych, modułów rozszerzeń, wyświetlaczy, a także minikomputerów jednopłytkowych (SBC) oraz rozmaite akcesoria – w tym zasilacze, adaptery, obudowy, przewody FPC, a także programatory/debuggery sprzętowe, kamery płytkowe i wiele, wiele innych. Cechą szczególną Waveshare, która wyróżnia firmę na tle konkurentów, jest doskonałe wsparcie użytkowników – w obszernym repozytorium opartym na systemie Wiki dostępne są szczegółowe dokumentacje, przewodniki, biblioteki programistyczne oraz przykładowe kody źródłowe, które zdecydowanie skracają czas potrzebny na zapoznanie się z daną platformą lub modułem oraz uruchomienie i zaprzęgnięcie go do pracy w opracowywanym projekcie.

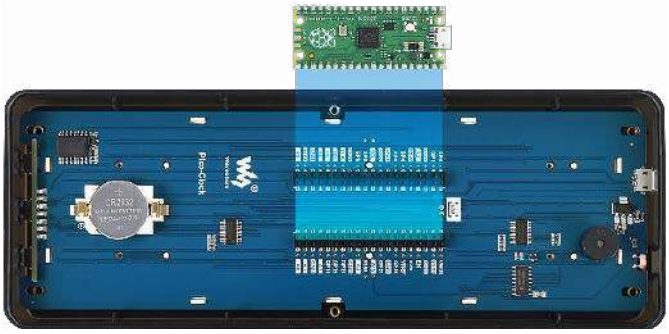
Więcej informacji:
Dystrybutorem marek Waveshare, Nextion oraz DWIN w Polsce jest firma **ELTY Sp. z o.o.**
43-516 Zabrzeg, ul. Miliardowicka 17 A
tel. +48 603 871 075
e-mail: elty@elty.pl, www.elty.pl

Spora część produktów z oferty Waveshare jest przeznaczona do pracy z minikomputerami Raspberry Pi oraz płytkami uruchomieniowymi Raspberry Pi Pico. Doskonała jakość narzędzi deweloperskich Waveshare sprawia, że znajdują one zastosowanie nie tylko podczas prac prototypowych, ale często implementowane są również w docelowych rozwiązaniach – dotyczy to zwłaszcza wysokiej klasy wyświetlaczy o szerokich kątach widzenia i wiernym odwzorowaniu barw, w części przypadków wyposażonych nawet w solidny panel ze szkła hartowanego. Uwagę zwracają przede wszystkim rozwiązania niestandardowe – interesujący przykład można zobaczyć na **fotografii 1**. Niektóre z zestawów ewaluacyjnych Waveshare w ogóle nie



Fotografia 2. Zestaw ewaluacyjny Waveshare do budowy zegara na bazie Raspberry Pi Pico (widok z przodu)

eprasa.pl 65e3eb482b



Fotografia 3. Wnętrze obudowy zegara zaprezentowanego na fotografii 2

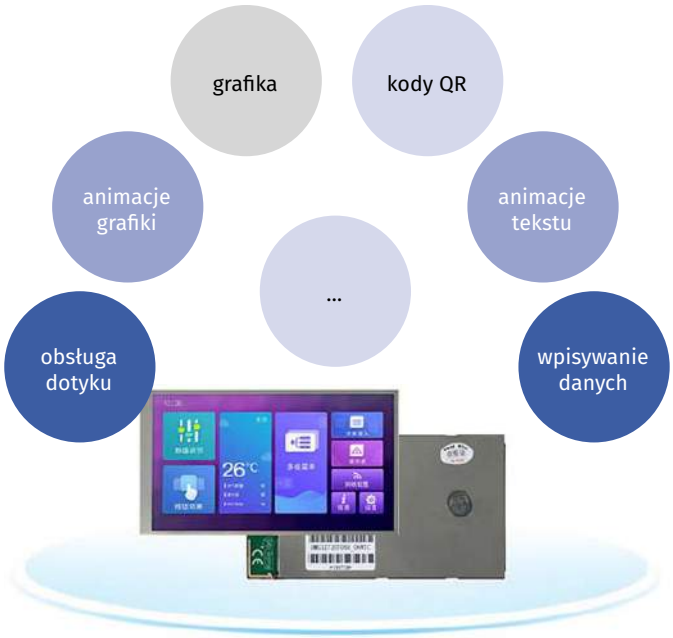
przypominają typowych rozwiązań, spotykanych na biurkach elektroników i programistów – za przykład niech posłuży estetyczny zegar LED, będący w istocie płytą bazową do wpięcia płytki Raspberry Pi Pico (**fotografie 2** i **3**).

Inteligentne wyświetlacze graficzne

W wielu współczesnych projektach urządzeń konieczna jest implementacja wysokiej jakości interfejsów HMI, bazujących na ekranach dotykowych. Tworzenie całego interfejsu „od zera" okazuje się często nie lada wyzwaniem, zwłaszcza gdy procesor zastosowany w urządzeniu ma mocno ograniczone zasoby pamięci danych oraz RAM, a także stosunkowo niewielką moc obliczeniową. W takich sytuacjach z pomocą przychodzą wyświetlacze inteligentne, które – oprócz samego ekranu graficznego oraz panelu dotykowego – zawierają również własny procesor, odpowiedzialny za generowanie i odświeżanie zawartości GUI. Komplet plików graficznych, wraz z niezbędnym programem sterującym, jest przechowywany na karcie microSD lub we wbudowanej pamięci Flash wyświetlacza, dzięki czemu główny procesor urządzenia może ograniczyć swoją interakcję z ekranem do wysyłania prostych komend (np. do zmiany tekstu czy przełączenia na inny widok) oraz odbierania powiadomień o działaniach użytkownika (kliknięcie przycisku, przesunięcie suwaka, etc.).

W ofercie naszej firmy znalazły się inteligentne wyświetlacze z serii Nextion (**fotografia 4**) oraz panele HMI marki DWIN (**fotografia 5**). Wyświetlacze Nextion mają wsparcie opracowanego przez producenta środowiska programistycznego Nextion Editor, zapewniającego intuicyjne narzędzia do projektowania interfejsów HMI w trybie WYSIWYG, zaś do komunikacji z procesorem nadrzędnym służy prosty zestaw komend UART. Oferta marki DWIN obejmuje natomiast

Fotografia 4. Przykładowy wyświetlacz inteligentny z serii Nextion marki IteadStudio



Fotografia 5. Przykładowy wyświetlacz inteligentny marki DWIN

szeroki wachlarz różnych rozwiązań – prostsze z nich działają na zasadzie zbliżonej do opisanych powyżej wyświetlaczy Nextion, choć oczywiście w tym przypadku mamy do dyspozycji inne oprogramowanie projektowe (DGUS TOOL). Bardziej rozbudowane modele bazują na systemie operacyjnym Android lub Linux i są przeznaczone do zaawansowanych urządzeń, w których wymaga się bezkompromisowej płynności animacji, wielopoziomowego menu z obsługą okien i aplikacji oraz dużych możliwości w zakresie obsługi multimediów.

REKLAMA

Moduły rozszerzeń do mikrokontrolerów, w tym do Raspberry Pi Pico

Wyświetlacze HMI firm Nextion, DWIN

ELTY.PL elektronika i ty

Elty sp. z o.o.
43-518 Zabrzeg, ul. Miliardowicka 17 A
elty@elty.pl, https://elty.pl/

eprasa.pl 65e3eb482b

Programowanie pamięci mikrokontrolerów

Programowanie układów pamięci jest operacją nieodłącznie związaną z techniką mikroprocesorową, z mikrokontrolerami. Już na etapie uruchamiania prototypu zbyt długi czas programowania pamięci może irytować konstruktora, o ile przy każdorazowej drobnej korekcie projektu istnieje konieczność przeprogramowania pamięci dużej pojemności. Wielominutowe programowanie oznacza bierne czekanie – a więc przerwę w pracach. Problem ulega zwielokrotnieniu na etapie produkcji seryjnej – gdy mamy do zaprogramowania tysiące układów każdego dnia. Czas temu poświęcony jest de facto czasem traconym, spowalniającym linię produkcyjną.

Pamiętam lata, kiedy programowaliśmy pamięci EPROM o pojemności 1 kB. Nikt się wtedy nie zastanawiał, jak optymalizować zapis pamięci nieulotnych. Aby zrozumieć, dlaczego dziś stanowi to istotny problem, rzućmy okiem na…

Wielkość współczesnych układów programowalnych

Z historycznych układów programowalnych, tzn. pamięci nieulotnych, mikrokontrolerów z wbudowaną pamięcią programu i programowalnych układów logicznych, swoje wpływy na organizacje procesu programowania układów w firmie utraciły te ostatnie. Strukturę nawet największych współczesnych układów CPLD/FPGA definiuje się zazwyczaj w układzie docelowym, a czas programowania struktury jest relatywnie krótki.

Krytyczne dla procesu produkcji stały się natomiast pamięci wielkiej pojemności typu NAND czy eMMC. Dzisiejsze aplikacje często wymagają olbrzymiej ilości danych, a i procesory – o mocy obliczeniowej nieporównywalnie większej niż we wspomnianych latach – wymagają większych kodów programów. Do tego nadchodzi jeszcze sztuczna inteligencja.

Jeżeli weźmiemy pod uwagę czas cyklu Erase/Program/Verify przykładowych dwóch układów współczesnych pamięci (jak w **tabeli 1**), to – przy produkcji seryjnej rzędu 100 tys. sztuk – na samo zaprogramowanie musimy poświęcić kilkadziesiąt lub kilkaset ośmiogodzinnych dniówek.

Te wielkości pokazują skalę problemu. Pierwszym krokiem ku jego rozwiązaniu jest zwielokrotnienie ilości jednocześnie programowanych układów poprzez zastosowanie off-liniowych programatorów wielogniazdowych (typowo 4-, 8-podstawkowych) lub – w przypadku ISP – wielokanałowych (typowo 4- , 8-, 16- kanałowych). Takie liczby układów lub zmontowanych płytek PCB, które trzeba kolejno umieszczać w podstawkach lub igłowych łożach testowych, kwalifikują proces do automatyzacji. I tu rysują się dwa kierunki działania:

- w przypadku programowania ISP – integracja programowania w ramach ręcznych i automatycznych stacji ICT testowania strukturalnego lub funkcjonalnego,
- w przypadku tradycyjnego programowania w programatorach off-line'owych – wyrzucenie programowania poza linię

Tabela 1. Czasy programowania współczesnych pamięci

Model pamięci	Typ (pojemność)	Czas EPV (s)	Wielkość produkcji	Potrzebna liczba dniówek (8 h)
Micron MT29F1G08ABAEAWP	NAND Flash (128 MB)	21,3	100 000	74,02
Sandisk SDIN8DE2-8G	eMMC (8 GB)	274	100 000	951,39

eprasa.pl 65e3eb482b

technologiczną i przygotowanie preprodukcyjne zaprogramowanych układów na automatach programujących.

Kryterium wielkości produkcji i czasu potrzebnego na programowanie jest kluczowe dla prawidłowego zaplanowania procesu programowania układów w firmie. Jeśliby robić to tradycyjnie w programatorach, nawet wielogniazdowych, to często tylko do tego zadania trzeba delegować pracowników. Jeśliby z kolei realizować to w układzie docelowym w technologii ISP (In-System-Programming), to czas (zazwyczaj szeregowego) zapisu danych do pamięci spowolni linię produkcyjną. Pewnym rozwiązaniem problemu jest jednoczesne programowanie kilku, kilkunastu układów – płyt PCB – jednocześnie, z użyciem wielokanałowych programatorów ISP. Ale i to stanowi barierę nie do przejścia przy masowej produkcji. W tym ostatnim przypadku pozostają nam praktycznie tylko automatyczne off-line'owe stacje programujące.

Ciągle nowe układy, ciągle nowe algorytmy

Inną cechą współczesnego rynku układów programowalnych jest olbrzymia liczba producentów, różnorodność typów układów, dynamika przejawiająca się w licznych premierach nowych podzespołów, a tym samym różnorodność i ogrom algorytmów, które – jak oczekujemy – nasz programator powinien obsługiwać. Niestety, te czasy minęły. Dziś firmy i osoby projektujące nowatorskie rozwiązania, oparte o najnowsze układy programowalne, muszą liczyć się z tym, że odpowiadające im algorytmy nie zostały jeszcze zaimplementowane. Powszechną praktykę stanowi dodawanie algorytmu za dodatkową opłatą. Jeśli – ograniczając koszty – zdecydujemy się na jakiś niszowy programator niszowej firmy, musimy liczyć się z tym, że producent nie będzie w stanie dodać nowego algorytmu, bo nie będzie miał dostępu do jego specyfikacji technicznych. Minęły bowiem czasy, gdy w kartach katalogowych publikowano parametry techniczne algorytmów. Dziś producenci układów programowalnych wspierają w tym zakresie jedynie wyselekcjonowanych wytwórców programatorów. I nie stanowi to ich złej woli lub zabiegu ograniczającego konkurencyjność. Stopień komplikacji algorytmów oraz liczba problemów związanych z ich implementacją wymusiłyby u producentów podzespołów konieczność rozbudowy służb obsługujących i autoryzujących wszystkich twórców programatorów. Stąd rada – jeśli chcesz utrzymać się w awangardzie producentów elektroniki, kupując programatory, obstawiaj renomowane firmy. Na przykład jakie?

ELNEC – jakość za rozsądną cenę

Światowa firma ze Słowacji, jeden z największych producentów programatorów off-line'owych. Flagowe programatory BeeProg obsługują wyjątkowo dużą liczbę układów, a cechą wyróżniającą jest to, że update'y oprogramowania z zaimplementowanymi kolejnymi nowymi układami są dostępne gratis w całym okresie ich życia. Koszt ich implementacji jest częściowo wliczony w cenę adapterów, z odpowiednimi dla danej obudowy podstawkami, dopasowujących mechanicznie i elektrycznie układy do gniazd programatora, ale mimo to strategia pozostaje unikatowa. Programatory są też przystosowane do programowania ISP.

BPM Microsystems – najszybsze automaty na rynku

„Mercedesy" ze Stanów w świecie programatorów off-line'owych. Zawsze nowatorskie, najszybsze, z bogatym wyposażeniem. „Game makerami" są automaty programujące 10. generacji (**fotografia otwierająca artykuł**), które mogą pobierać samoczynnie układy z tub, z tacek (pojedynczych lub ze stosu) bądź taśm. Programują równolegle do 48 układów w kilku site'ach, dokonują inspekcji optycznej wyprowadzeń, oznaczają laserowo układy i pakują je do taśm lub tacek (pojedynczych lub na stos).

SMH Technologies/Algocraft – od najprostszych do wysublimowanych rozwiązań ISP

Pełna oferta z Italii, i to nie turystyczna, dla integratorów automatycznych systemów testowania chcących włączyć w nie programowanie ISP pamięci oraz mikrokontrolerów. Technologie bazowe FlashRunner i WriteNow są odpowiednie do wszystkich zastosowań, ale producent oferuje też wysublimowane rozwiązania dla krytycznie szybkich układów. Możliwość programowania w sieci.

JTAG Technologies – programowanie w ramach testowania

Z Niderlandów, ze źródeł „boundary-scan" mamy rozwiązania pozwalające programować układy via interfejs JTAG. W tym przypadku od testowania metodą ścieżki krawędziowej dochodzimy do programowania – poniekąd przy okazji. Możemy upiec dwie pieczenie przy jednym ogniu.

AMTEC.PRO – automaty dedykowane

Wracamy do śródziemnomorskich klimatów, tym razem do Słowenii. Przepakowując w automatycznych kabinach FlexCell układy z taśm, tacek i tub do taśm tłoczonych, możemy przy okazji je zaprogramować w programatorach firmy ELNEC. W tym wariancie otrzymujemy automatyczne rozwiązania budżetowe dedykowane pod klienta.

WG Electronics – ekspert w programowaniu

Ponieważ temat programowania układów jest w spektrum zainteresowań autora jeszcze od czasów z przed założenia firmy, uzurpujemy sobie prawo do bycia znawcą tematu. Tak wiec, jako WG Electronics wspieramy naszych klientów w analizie ich potrzeb i w doborze odpowiednich rozwiązań. A tu mamy pełne portfolio programatorów zarówno off-line'owych, jak i ISP – od relatywnie tanich, jednokrotnych, dla hobbystów czy konstruktorów, do wielokrotnych programatorów produkcyjnych ręcznych i automatycznych. Współpracujemy z renomowanymi producentami światowymi, którzy zapewniają dostęp do algorytmów programowania najnowszych układów programowalnych wszystkich światowych producentów, a tym samym zapewniają rozwój i update'y swoich urządzeń teraz i w przewidywalnej przyszłości.

Wybierając dziś właściwe rozwiązanie, zapewnicie sobie jego skalowalność i rozwój w dalszej perspektywie. Może więc warto wydać obecnie trochę więcej (nie mamy bowiem w ofercie najtańszych programatorów), aby nie wydawać w przyszłości kilka razy na podobne urządzenia. Zapraszamy do współpracy.

Tadeusz Górnicki
WG Electronics Sp. z o.o.
www.wg.com.pl

eprasa.pl 65e3eb482b

Fotografia 1. Nowoczesne pojemnościowe systemy dotykowe wymagają szybkiej adaptacji do zmian temperatury i poziomu wilgoci

Ewolucja peryferiów analogowych w mikrokontrolerach 8-bitowych

Mikrokontrolery odgrywają znaczącą rolę w naszej codzienności, znajdując miejsce w niemal wszystkich otaczających nas urządzeniach elektronicznych – od motoryzacji i lotnictwa, poprzez elektronikę użytkową, aż po automatykę przemysłową czy aparaturę medyczną.

Choć 8-bitowe mikrokontrolery są obecne na rynku już od prawie pięćdziesięciu lat, to nowe aplikacje – takie jak samochody i rowery elektryczne, automatyka domowa czy też urządzenia Internetu Rzeczy (IoT) – w wykładniczym tempie napędzają popyt na małe, niedrogie procesory. W rezultacie mikrokontrolery 8-bitowe znacznie wyewoluowały, aby dotrzymać kroku funkcjonalnościom wymaganym w nowoczesnych aplikacjach. Przeanalizujemy trzy typowe zastosowania, w których nowa klasa 8-bitowych MCU z zaawansowanymi funkcjami analogowymi wspiera budowę nowoczesnych systemów.

Zarządzanie baterią, monitorowanie i optymalizacja

W przypadku wielu systemów wbudowanych używanych w odległych lokalizacjach (np. w sensorach IoT) głównym źródłem zasilania jest bateria lub akumulator. Monitorowanie żywotności i stanu naładowania źródła energii jest niezwykle ważną funkcją, zapewniającą bezpieczną i niezawodną pracę urządzenia przez wiele tygodni, miesięcy, a nawet lat.

W klasycznych aplikacjach problem monitorowania stanu baterii lub akumulatora jest rozwiązywany na drodze pomiaru podstawowych parametrów źródła energii (napięcia i prądu) oraz obliczania na ich podstawie pozostałego czasu pracy urządzenia. Zadania te realizuje najczęściej sam mikrokontroler, a ich celem jest zarówno informowanie użytkownika (lub nadrzędnego urządzenia) o stanie naładowania, jak i optymalizacja wydajności zasilania. Mikrokontroler ze zintegrowanym przetwornikiem analogowo-cyfrowym (ADC) próbkuje zatem wartości prądu i napięcia akumulatora oraz konwertuje je na dane cyfrowe, które odpowiedni program może wykorzystać do oszacowania stanu baterii. Jeśli potrzebne jest ponadto równoczesne monitorowanie temperatury akumulatora, zintegrowany w strukturze mikrokontrolera wzmacniacz operacyjny może zostać użyty jako bufor napięcia polaryzującego czujnik temperatury. Wbudowane wzmacniacze operacyjne są dostępne w szerokiej gamie nowoczesnych 8-bitowych mikrokontrolerów i mogą nie tylko obniżyć koszty implementacji systemu, ale także zredukować powierzchnię płytki drukowanej w aplikacjach wymagających dodatkowego stopnia wzmocnienia przed wejściem przetwornika ADC.

Aby zoptymalizować wydajność źródła zasilania i wydłużyć czas pracy systemu, mikrokontroler powinien oferować różne tryby zarządzania zużyciem energii – co pozwoli zrównoważyć zapotrzebowanie na wydajność, przy jednoczesnej redukcji poboru mocy. Kluczem do długiej żywotności baterii w systemie wbudowanym jest możliwość ograniczenia aktywności systemu, gdy nie jest on aktywny. Elastyczne poziomy konfiguracji pozwalają systemowi zużywać minimalną ilość energii do realizacji bieżących zadań, często bez nadzoru ze strony rdzenia. Funkcje takie, jak tryby IDLE, DOZE lub SLEEP, zapewniają daleko idące oszczędności energetyczne. Dodatkowo w najnowszych mikrokontrolerach Microchip (PIC i AVR) przetworniki ADC, wzmacniacze operacyjne oraz przetworniki DAC można również włączać i wyłączać programowo lub ustawić tak, aby uruchamiały rdzeń MCU i cyfrowe bloki peryferyjne po osiągnięciu określonych progów, co zapewnia dodatkową elastyczność i oszczędność energii w aplikacjach bateryjnych. Dzięki funkcjom niskiego poboru mocy, dostępnym w nowoczesnych mikrokontrolerach, żywotność baterii jest maksymalizowana – jednocześnie zmniejsza się zużycie prądu, rozpraszanie mocy, a więc i... koszty.

Wbudowane funkcje zarządzania systemem

Konstrukcje aplikacji wbudowanych z biegiem czasu ulegają coraz większemu skomplikowaniu. Konieczne okazuje się zatem rozdzielenie zadań przetwarzania danych w sposób zapewniający błyskawiczną reakcję systemu, aby zmaksymalizować wygodę użytkownika czy też przestrzegać rygorystycznych norm bezpieczeństwa. Aby to osiągnąć, złożone systemy często korzystają z nowoczesnych 8-bitowych mikrokontrolerów ze względu na zintegrowane w ich strukturach peryferia analogowe, wykonujące zadania „porządkowe", w tym:

eprasa.pl 65e3eb482b

Rysunek 1. Mikrokontrolery ze zintegrowanymi peryferiami analogowymi pomagają projektantom obniżyć koszty i skrócić czas wprowadzenia produktu na rynek, jednocześnie poprawiając responsywność systemu

sterowanie szynami zasilającymi, monitorowanie jakości środowiska, wykrywanie błędów i awarii czy też arbitraż komunikacji między kilkoma chipami na płycie głównej. Funkcje te mają kluczowe znaczenie w wielu zastosowaniach, m.in. w infrastrukturze centrów danych, systemach zarządzania budynkami, punktach końcowych inteligentnych sieci i zastosowaniach związanych z zapewnieniem bezpieczeństwa, np. w niektórych sprzętach AGD.

Jedną z takich aplikacji może być serwer znajdujący się w centrum danych. Chociaż na każdej płycie głównej umieszczony został wydajny procesor CPU i szeroka gama procesorów aplikacyjnych do obsługi różnych zadań, wiele współczesnych serwerów korzysta także z nowoczesnych mikrokontrolerów 8-bitowych jako układów do zarządzania rozmaitymi funkcjami systemowymi. „Małe" mikrokontrolery stanowią zazwyczaj interfejs czujników środowiskowych (temperatury, wilgotności, napięcia na szynach zasilania, etc.) i są zaprogramowane do raportowania stanu na magistrali systemowej podczas sekwencjonowania zasilania innych urządzeń na płycie głównej. W takich przypadkach 8-bitowe mikrokontrolery doskonale nadają się do tych prostszych zadań. Niektóre układy, takie jak PIC i AVR firmy Microchip, oferują niezależne od rdzenia bloki peryferyjne (CIP), które współpracują z blokami analogowymi procesora. Wzmacniacze operacyjne lub przetworniki ADC zapewniają możliwość wstępnego kondycjonowania sygnałów wejściowych (wzmocnienie, filtracja analogowa, etc.). Natomiast bloki CIP zaprojektowano tak, by automatyzowały zadania systemowe bez konieczności angażowania rdzenia ani nawet nadzoru z jego strony, co umożliwia zredukowanie ilości kodu, który trzeba napisać, zdebugować i przetestować. Poszczególne peryferia CIP komunikują się ze sobą, co dodatkowo pomaga zwiększyć wydajność i responsywność systemu, umożliwiając jednoczesną obsługę wielu zadań.

Mikrokontrolery PIC i AVR idą o krok dalej i używają wyrafinowanych peryferiów analogowych, aby umożliwić zaawansowane obliczenia niezależne od rdzenia, takie jak uśrednianie, nadpróbkowanie czy filtracja dolnoprzepustowa. Funkcje te pomagają przyspieszyć reakcję systemu i zapewniają efektywną eliminację zakłóceń podczas pracy w najbardziej wymagających środowiskach. Mało tego – peryferia analogowe można łączyć z wieloma innymi sygnałami wewnątrz mikrokontrolera (w tym z timerami, źródłami sygnału zegarowego, cyfrowymi sygnałami peryferyjnymi itp.) w celu automatycznego wyzwalania określonych procesów lub generowania przerwań, powiadamiających procesor o konieczności podjęcia określonych zadań.

Rysunek 2. Procesory z rodziny PIC18-Q71 firmy Microchip zawierają szereg inteligentnych, analogowych bloków peryferyjnych, w tym różnicowy przetwornik analogowo-cyfrowy (ADC)

Interfejs użytkownika

Technologia wykrywania dotyku jest stosowana w szerokiej gamie produktów elektronicznych, od smartfonów, poprzez urządzenia RTV i AGD, aż po wyposażenie samochodów. W projektowaniu kierownic i desek rozdzielczych stopniowo odchodzi się od klawiszy mechanicznych na rzecz eleganckich, elastycznych interfejsów użytkownika. Przyciski dotykowe muszą natychmiast reagować na interakcję użytkownika, gwarantować odporność na fałszywe wyzwalacze i dostosowywać się do wielu różnych warunków środowiskowych, w tym szybkich wahań temperatury czy nadmiernej wilgotności, a dodatkowo powinny prawidłowo reagować nawet na dotyk dłoni w rękawiczkach.

We współczesnych interfejsach pojemnościowych sama tylko moc obliczeniowa 32-bitowych procesorów nie jest w stanie zastąpić specjalnie zaprojektowanych interfejsów analogowych, które można znaleźć w nowoczesnych mikrokontrolerach 8-bitowych. Nowe układy z oferty firmy Microchip, w tym rodziny produktów PIC18-Q71 i AVR EA, są wyposażone w różnicowe przetworniki ADC z zaawansowanymi możliwościami filtrowania, pozwalającymi na znaczną redukcję ilości wymaganych interwencji rdzenia, potrzebnych do wdrożenia niezawodnej detekcji dotyku. Te wysoko wyspecjalizowane przetworniki ADC odznaczają się doskonałą charakterystyką tłumienia zewnętrznych zaburzeń i oferują wbudowane funkcje automatycznego dostrajania oraz kalibracji w celu zwiększenia odporności na zakłócenia, a także tolerancji na wilgoć i wodę. W połączeniu z łatwymi w użyciu narzędziami programistycznymi firmy Microchip mikrokontrolery te zapewniają kompletne rozwiązanie do tworzenia wymagających interfejsów dotykowych, narażonych na trudne warunki otoczenia.

Podsumowanie

W miarę wzrostu wymagań, dotyczących mikrokontrolerów na przestrzeni ostatnich 50 lat, w systemach wbudowanych zacierają się granice pomiędzy tym, co musi być „analogowe", a tym, co może pozostać w domenie cyfrowej. Wraz z ewolucją projektów systemów wbudowanych, 8-bitowy mikrokontroler ewoluował od prostych układów obliczeniowych, do pełnoprawnych rozwiązań typu System-on-Chip (SoC), zbudowanych z myślą o obsłudze większości współczesnych zadań, jakie stawia się przed systemami embedded. Począwszy od prostego monitorowania napięć zasilania, a skończywszy na obowiązkach głównego sterownika w złożonych projektach, mikrokontrolery ze zintegrowanymi peryferiami analogowymi umożliwiają projektantom przeniesienie niektórych funkcji, wykonywanych zwykle poza chipem, na główny procesor. Takie rozwiązanie znakomicie poprawia responsywność systemu i zmniejsza koszty BOM. Inteligentne analogowe bloki peryferyjne, dostępne w 8-bitowych mikrokontrolerach PIC i AVR, płynnie integrują się z peryferiami cyfrowymi, zapewniając jeszcze większą funkcjonalność i elastyczność dla konstruktorów wyrafinowanych systemów wbudowanych.

za: **Stephanie Pinteric, Wayne Freeman,**
Ewolucja funkcjonalności analogowej w 8-bitowym mikrokontrolerze i jej nowa, innowacyjna rola w elektronice

Więcej informacji o peryferiach analogowych najnowszych mikrokontrolerów 8-bitowych Microchipa możesz znaleźć na stronie: microchip.com/8bit.

eprasa.pl 65e3eb482b

Moduły SoM od Laird Connectivity

– gdy mikrokontroler to (zdecydowanie) za mało

Co roku najnowsze serie mikrokontrolerów przekraczają kolejne granice mocy obliczeniowej, rozmiaru pamięci programu oraz pamięci RAM czy też możliwości oferowanych przez peryferia sprzętowe. Gdy w docelowej aplikacji konieczna okazuje się obsługa wysokiej jakości obrazów, dźwięku lub wideo, wysokopoziomowych systemów operacyjnych czy algorytmów sztucznej inteligencji (AI), naturalnym wyborem konstruktorów jest skręcanie w kierunku procesorów aplikacyjnych. Praktyczna implementacja tych układów jest jednak dość trudna, czaso- i pracochłonna, a zarazem… kosztowna. We wszystkich tych kłopotach z pomocą inżynierom przychodzą gotowe moduły minikomputerów SoM.

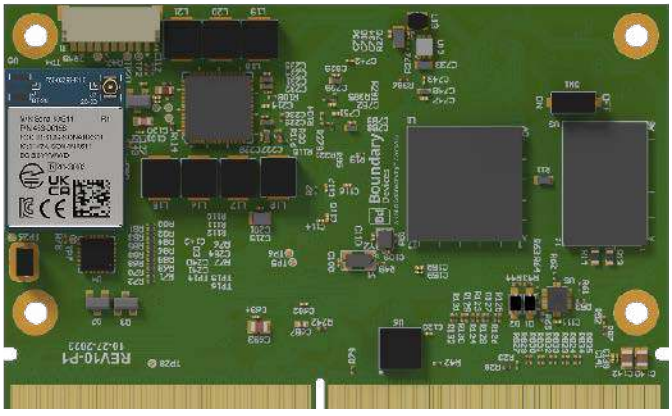
Rynkowa oferta minikomputerów jednopłytkowych w formie modułów SoM stale się poszerza, ale tylko niektóre firmy mogą poszczycić się modelami o wydajności obliczeniowej umożliwiającej uruchamianie najnowszych systemów operacyjnych czy algorytmów uczenia maszynowego, nierzadko pracujących na dodatek w reżimie czasu rzeczywistego. Firma Laird Connectivity specjalizuje się nie tylko w nowoczesnych modułach komunikacyjnych dla aplikacji IoT, z którymi zapewne kojarzy ją wiele osób – w ofercie amerykańskiego przedsiębiorstwa o 38-letniej tradycji znalazło się także pokaźne portfolio minikomputerów SoM oraz SBC, opartych w większości przypadków na potężnych procesorach aplikacyjnych marki NXP z serii i.MX 6, i.MX 8 oraz i.MX 9, a także MediaTek Genio. W artykule zaprezentujemy trzy wybrane modele, należące do serii Nitrogen8M oraz Tungsten.

Nitrogen8M Mini SMARC

Linia produktów Nitrogen8M obejmuje kilka modeli minikomputerów, przy czym wszystkie oparte są na wydajnych procesorach aplikacyjnych z serii i.MX 8. Moduł Nitrogen8M Mini SMARC (**fotografia 1**) bazuje na układzie i.MX 8M Mini, wyposażonym w czterordzeniową jednostkę ARM Cortex-A53 o taktowaniu do 1,8 GHz i dodatkowy koprocesor ARM Cortex-M4 (400 MHz), wspierający główną jednostkę w realizacji rozmaitych zadań niskopoziomowych. Układ ma wbudowane GPU 2D/3D, szeroką gamę interfejsów komunikacyjnych (w tym PCIe, USB2.0, Gigabit Ethernet, MIPI-DSI, MIPI-CSI, I²S, SPIDF oraz UART, I²C i SPI), a także obszerny zestaw peryferiów wspierających realizację algorytmów szyfrujących oraz zapewniających ochronę krytycznego kodu i danych aplikacji. Na pokład minikomputera trafiły także: rozbudowany układ zarządzania zasilaniem (PF8121 marki NXP), ultranowoczesny moduł radiowy Sona Wi-Fi 6/6E + Bluetooth 5.3/5.4 oraz pamięci (RAM LPDDR4 o pojemności 2 GB, 4 GB lub 16 GB i eMMC o pojemności 16 GB). Całość mieści się na płytce drukowanej o wymiarach zaledwie 82×50 mm, wyposażonej w standardowe złącze krawędziowe. Uruchomienie tego oraz innych modułów kompatybilnych ze standardem SMARC ułatwi płyta bazowa Universal SMARC Carrier (**fotografia 2**), która rozprowadza sygnały z głównego złącza systemowego minikomputera na poszczególne gniazda interfejsowe, w tym HDMI, USB, Ethernet (RJ-45), DSI, CSI – a także szereg złączy szpilkowych i rozmaitych dodatkowych złączy rastrowych.

Minikomputer Nitrogen8M Mini SMARC jest przeznaczony do szerokiego zakresu aplikacji docelowych, w tym kontrolerów inteligentnego budynku, przemysłowych systemów IoT, automatów vendingowych, czy nowoczesnych urządzeń medycznych. Co ważne – zwłaszcza w przypadku tych ostatnich – producent zapewnia przynajmniej 10-letnią dostępność modułów w sprzedaży, co niewątpliwie stanowi ukłon w stronę rynków wymagających stabilnych, przewidywalnych łańcuchów dostaw przez cały okres życia produktu. Inżynierowie oprogramowania wbudowanego mogą wybrać jeden

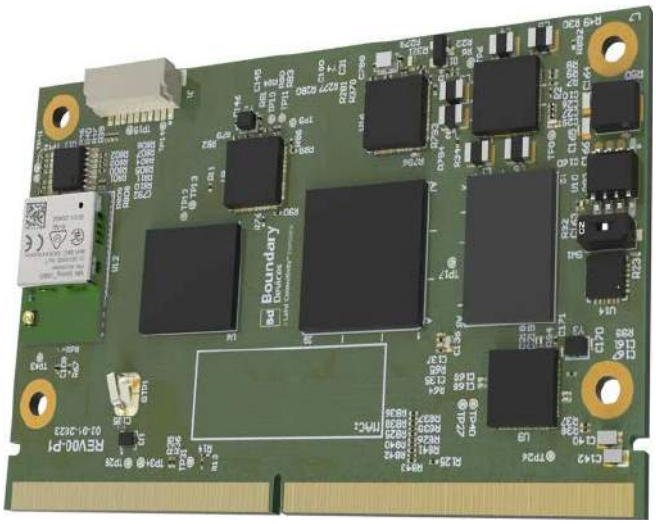
Więcej informacji:
Oficjalnym dystrybutorem modułów marki Laird Connectivity w Polsce jest firma **GLYN Poland**.
50-075 Wrocław, ul. Krupnicza 13, tel. 71 7828-758
e-mail: sales@glyn.pl, www.glyn.pl



Fotografia 1. Widok modułu Nitrogen8M Mini SMARC

Fotografia 2. Płyta bazowa do szybkiego prototypowania z użyciem minikomputerów z serii SMARC – Universal SMARC Carrier

eprasa.pl 65e3eb482b



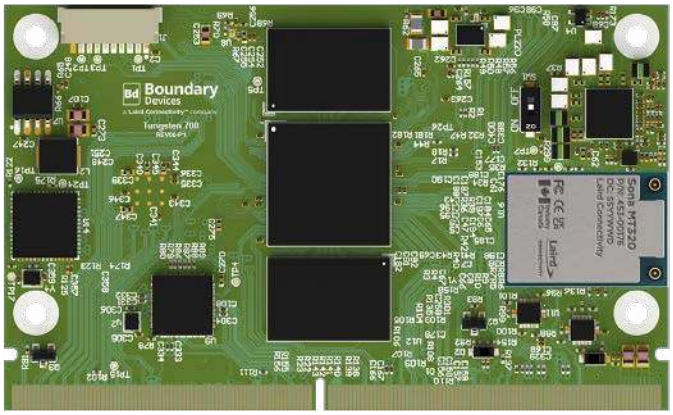
Fotografia 3. Widok modułu Nitrogen8M Plus SMARC

spośród wielu systemów operacyjnych – płytka wspiera bowiem systemy Android, Ubuntu, Zephyr RTOS, FreeRTOS, a także customowe dystrybucje linuksowe budowane w oparciu o Yocto oraz Buildroot.

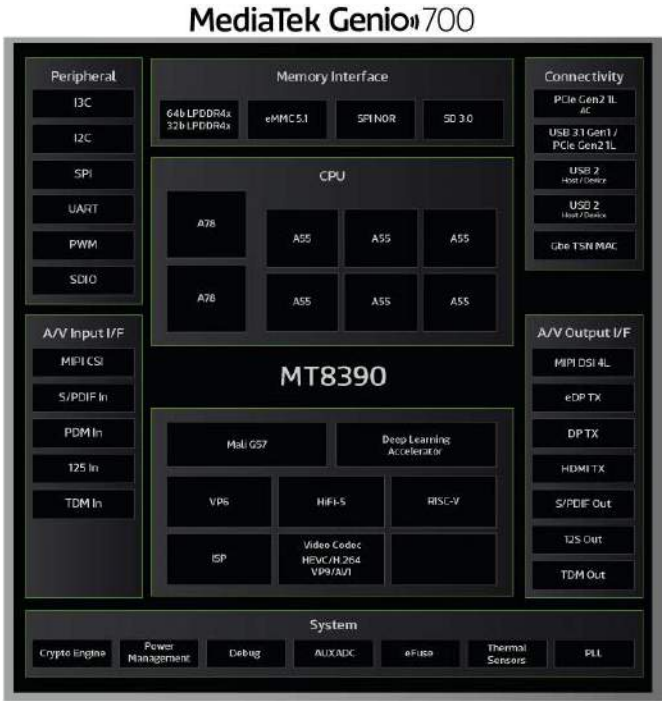
Nitrogen8M Plus SMARC

Płytka Nitrogen8M Plus SMARC (**fotografia 3**) daje już konstruktorom oraz integratorom systemów znacznie większe możliwości i to zarówno w zakresie sterowania niskopoziomowego oraz komunikacji z otoczeniem sprzętowym, jak też wysokopoziomowego przetwarzania danych. Choć cały moduł ma identyczne wymiary jak opisana poprzednio wersja Mini, to zastosowany tu układ NXP i.MX 8M Plus oferuje aż 800-megahercowy koprocesor ARM Cortex-M7. W oczywisty sposób podnosi to walory modułu w aplikacjach związanych m.in. z komunikacją za pośrednictwem szyny CAN bądź gigabitowego Ethernetu, pracującego w ramach sieci TSN (Time Sensitive Networking), od których wymaga się wysokiej dostępności oraz minimalnej latencji w transmisji danych. Mało tego – oprócz koprocesora Cortex-M7, w układzie znalazł się także rdzeń Tensilica DSP HiFi 4, czyli 32-bitowa jednostka zmiennoprzecinkowa, przeznaczona do realizacji wymagających zadań z zakresu zaawansowanych aplikacji audio – w tym przetwarzania sygnałów zbieranych przez macierze mikrofonów, np. w celu rozpoznawania komend głosowych z użyciem algorytmów AI. W procesorze i.MX 8M Plus znajdziemy także akcelerator uczenia maszynowego o wydajności 2,25 TOPS, koprocesory do obróbki grafiki 2D i 3D, a nawet dwa procesory ISP (Image Signal Processor), które w połączeniu z podwójnym interfejsem kamer MIPI-CSI mogą realizować rozbudowane zadania z zakresu przetwarzania obrazu stereowizyjnego w czasie rzeczywistym.

Za komunikację radiową w przypadku Nitrogen8M Plus SMARC odpowiada moduł Sterling LWB5+ Wi-Fi 5/Bluetooth 5.2, oparty na dwuzakresowym układzie CYW4373E marki Cypress. Dostarczaniem



Fotografia 4. Widok modułu Tungsten700 SMARC



Rysunek 1. Uproszczony schemat blokowy procesora MediaTek Genio 700 (t.ly/WO8Cm)

stabilnego zasilania do wszystkich podsystemów minikomputera zajmuje się natomiast układ PCA9450, integrujący w sobie aż sześć wydajnych przetwornic typu buck, pięć stabilizatorów LDO, 400-miliamperowy przełącznik zasilania, 2-kanałowy translator poziomów logicznych oraz układ taktowania 32,768 kHz do obsługi zegara RTC. Wszystkie minikomputery z omawianej serii są wyposażone w 16 GB pamięci eMMC oraz 2, 4 lub 8 GB pamięci RAM typu LPDDR4.

Tungsten700 SMARC

Na koniec naszej prezentacji pozostawiliśmy moduł Tungsten700 SMARC (**fotografia 4**), będący praktyczną implementacją potężnego procesora aplikacyjnego MediaTek Genio 700 (**rysunek 1**), zbudowanego w oparciu o dwurdzeniową jednostkę główną (w tej roli ARM Cortex-A78 o taktowaniu do 2,2 GHz) oraz…. sześciordzeniowy procesor wspomagający ARM Cortex-A55 o taktowaniu maksymalnym równym 2,0 GHz. Całość współpracuje z koprocesorem graficznym ARM Mali-G57 MC3 GPU, pozwalającym na płynną obsługę dwóch ekranów w jakości 4K30/4K60. Za przetwarzanie dźwięku odpowiada rdzeń Tensilica HiFi 5 Audio DSP, zaś ogromny koprocesor AI wspiera aplikacje sztucznej inteligencji, dostarczając zawrotną wydajność obliczeniową na poziomie aż 3,7 TOPS.

Komunikacja radiowa w przypadku minikomputera Tungsten700 SMARC jest możliwa dzięki transceiverowi Sona MT320 Wi-Fi 6/Bluetooth 5.3. W chwili pisania niniejszego artykułu moduł SoM występuje w dwóch wersjach, różniących się wielkością wbudowanej pamięci RAM (4 GB lub 8 GB), przy czym każda z nich ma wbudowane 16 GB pamięci eMMC.

Podsumowanie

W artykule zaprezentowaliśmy zaledwie trzy spośród kilkunastu modeli minikomputerów SoM marki Laird Connectivity, jednak już tak niewielka próbka możliwości producenta pokazuje, że mamy do czynienia z wytwórcą świadomym potencjału drzemiącego w najnowszych procesorach aplikacyjnych. Nic więc dziwnego, że firma zyskała status „złotego partnera" (Gold Partnership) firmy NXP. Warto też dodać, że oprócz modułów sprzętowych firma Laird Connectivity oferuje także doskonale przygotowaną, szczegółową bazę dokumentacji, modele 3D minikomputerów, tutoriale, a także kompletne repozytoria dla programistów Linuksa.

eprasa.pl 65e3eb482b

Multimetr stołowy Owon XDM3051

5½ cyfry w przystępnej cenie – da się?

Gdy zaczynałem przygodę z elektroniką, cyfrowe multimetry 3½ cyfry były standardem zarówno wśród amatorów, jak i profesjonalistów. Multimetr stołowy 4½ cyfry miał wtedy cenę poza zasięgiem wszystkich amatorów (i wielu zawodowców), a instrument wyższej klasy kosztował tyle co samochód. Ale w ostatniej dekadzie zdarzyła się cicha rewolucja. Pierwszym jej efektem były tanie oscyloskopy cyfrowe, oferujące klasę dokładności i funkcjonalność instrumentów kilka razy droższych. Drugim, przez wielu niezauważonym, był wysyp multimetrów lepszej klasy w niższych cenach. Jeden z tych instrumentów przedstawiony zostanie w niniejszym artykule.

Najważniejsze cechy:
- 5½ cyfry, 200 000 zliczeń,
- Pomiar AC TrueRMS do 100 kHz,
- Ekran TFT 4” 480×320 px,
- Akwizycja do 150 odczytów na sekundę,
- Interfejsy USB, RS232, LAN, zgodne z SCPI.

Owon to chiński producent aparatury pomiarowej znany głównie z produkcji oscyloskopów. W swojej ofercie ma też generatory funkcyjne, analizatory widma, zasilacze oraz multimetry. Jeszcze kilka lat temu firma nie cieszyła się dobrą opinią, głównie ze względu na przeciętną jakość wykonania, oprogramowanie pełne błędów i „zamiłowanie do plastiku”. W ostatnim czasie jednak Owon próbuje poprawić swoją opinię, stąd nowe, bardziej profesjonalne serie instrumentów, w tym multimetry stołowe serii XDM. Urządzenia te mają dość szeroką rozpiętość cenową i różne klasy dokładności. Najwyższy model w serii to bohater tej recenzji, XDM3051. Nie jest to najtańszy multimetr 5½ cyfry na naszym rynku – tańszym modelem jest na przykład Uni-T UT805A. Ba, w praktyce elektronika, nawet profesjonalnego, nieczęsto zdarza się potrzeba aż tak dużej rozdzielczości i dokładności pomiaru – w wielu przypadkach rozdzielczość 4½ cyfry w zupełności wystarczy, a ceny instrumentów o takich parametrach zaczynają się poniżej 200 złotych (popularny w ostatnich latach Aneng AN870). Nawet multimetry stołowe 4½ cyfry nie kosztują więcej niż tysiąc złotych (modele od Uni-T, Aneng czy właśnie Owon). Dlaczego zatem na własne potrzeby wybrałem najdroższy model Owona? Odpowiedź jest prosta: dla jednego zakresu pomiarowego, którego tańsze modele nie mają albo nie oferują adekwatnej rozdzielczości i dokładności.

Zakup, zawartość pudełka, pierwsze wrażenie

Multimetr zakupiłem u największego w Polsce dystrybutora części elektronicznych. Za ten zakup zapłaciłem 2333,14 PLN. Owon zapakował swój produkt w solidny karton z dodatkową rączką do przenoszenia. Jedyną informacją o zawartości jest nalepka w rogu pudełka z napisem „Digital Multimeter”. W środku

eprasa.pl 65e3eb482b

kartonu znajduje się samo urządzenie, zabezpieczone piankowymi nakładkami i zawinięte w piankową torbę – dziwi brak typowej dla urządzeń elektronicznych torebki z pochłaniaczem wilgoci. Poza samym instrumentem w kartonie znajdują się też: kabel zasilający, para sond pomiarowych i dodatkowych krokodylków, kabel USB, zapasowe bezpieczniki, instrukcja obsługi oraz płyta CD z oprogramowaniem i instrukcją w pliku PDF. Pewną ciekawostką jest naklejka umieszczona na gnieździe zasilającym – ostrzegająca, by najpierw upewnić się co do ustawienia napięcia zasilania (XDM3051 pracuje z napięciami 115 V lub 230 V), co stanowi przejaw przywiązania do detali. Z tyłu urządzenia, poza gniazdem zasilania z bezpiecznikiem oraz przełącznikiem napięć, znajdziemy zaskakujące jak na tę cenę bogactwo gniazd: dwa gniazda BNC – jedno to wyjście pomocnicze (sygnalizujące zakończenie pomiaru lub przekroczenie zdefiniowanych wartości), a drugie to wejście zewnętrznego wyzwalania pomiaru. Poniżej znajdują się gniazda portu RS232, USB typu B i LAN. Miernik wspiera protokół SCPI i pozwala na zdalne sterowanie instrumentem. Dodatkowo konstruktorzy uwzględnili gniazdo Kensington dla linki zabezpieczającej oraz zacisk uziemienia. Cała obudowa jest metalowa, z tworzywowymi ramkami oraz panelem frontowym. Urządzenie ma wymiary 23,5×11×29,5 cm i waży 3,06 kg.

Lewą część frontu zajmuje zaskakująco czytelny ekran TFT – Owon bardzo dobrze dobrał czcionki oraz kolory, wszystko jest czytelne nawet dla osób z problemami ze wzrokiem. Lewy dolny róg to także lokalizacja sprzętowego włącznika zasilania – wyłączony miernik nie pobiera prądu. Pod ekranem znajdują się przyciski wyboru typu pomiaru: napięcie stałe/zmienne, prąd stały/zmienny, rezystancja/test ciągłości/test diody, pojemność, częstotliwość i temperatura. Po prawej stronie ekranu znajdują się przyciski funkcyjne, których przeznaczenie zmienia się zależnie od wybranej opcji. Na prawo od nich – przyciski dodatkowych funkcji i menu, klawisze nawigacyjne pozwalające też na zmianę parametrów i ręczny wybór zakresu pomiarowego, a poniżej nich – przyciski trybu wyświetlania wyników oraz gniazdo USB do ich zapisywania w zewnętrznej pamięci. Po prawej stronie znajduje się grupa pięciu gniazd pomiarowych i gniazda bezpiecznika dla zakresu prądowego. Dwa gniazda z lewej strony służą do pomiaru rezystancji metodą czteroprzewodową (Kelvina). Potrzebne do tego okablowanie trzeba dokupić oddzielnie. Pierwsze gniazdo po prawej (licząc od góry) służy do wszystkich pomiarów (z wyjątkiem natężenia prądu), następne jest gniazdo wspólne, a pod nim – gniazdo do pomiaru prądu. Multimetr potrafi naprzemiennie mierzyć napięcie i prąd oraz wyświetlać oba wyniki naraz. Instrukcja, choć udziela informacji na temat każdego pomiaru, nie pokazuje, jak jednocześnie mierzyć napięcie i prąd – moim zdaniem to drobny brak, zwłaszcza jeśli taki multimetr zakupi hobbysta.

O ile nie można narzekać na wygląd, estetykę i obsługę samego multimetru, to jednak pojawiają się w nim pewne wady techniczne i konstrukcyjne. Multimetr ma rozkładaną podpórkę i pokryte gumą narożniki, ale to rozwiązanie nie sprawdza się wcale – nawet przy złożonej podpórce XDM3051 nie leży płasko na blacie, gumy z przodu nie dotykają blatu, a te z tyłu dotykają go tylko krawędzią. Rozłożenie podpórki nie poprawia sytuacji – cała wykonana została z tworzywa i na zwykłym, fornirowanym blacie instrument łatwo się ślizga. O ile wciskanie przycisków funkcyjnych nie przemieszcza miernika, o tyle próba wciśnięcia włącznika zasilania wymaga już przytrzymania przyrządu, podobnie zresztą podłączanie sond. Jest to poważna wada świadcząca o tym, że ktoś w firmie Owon nie przemyślał projektu do końca. Wspomniane gumki można zdjąć, ale wymaga to rozkręcenia obudowy. Co ciekawe, ustawienie miernika na boku lub do góry nogami, by gumowe elementy dobrze przylegały do podłoża, drastycznie poprawia przyczepność. Drugi, mniejszy problem widzimy przy zmianie ustawień, a konkretnie jasności ekranu – po każdym wyłączeniu instrument wraca do domyślnych 50% i nie zapisuje preferencji użytkownika. Jedynym tego wyjaśnieniem jest niedopatrzenie ze strony programisty. Przy okazji, multimetr ma opcję aktualizacji oprogramowania, ale na stronach Owon brak jakiejkolwiek wersji do pobrania dla tego modelu.

Urządzenie włącza się w czasie około ośmiu sekund, wewnątrz obudowy znajduje się dość intensywnie pracujący wentylator. Program miernika startuje w trybie pomiaru napięcia stałego z automatycznym zakresem ustawionym na 200 mV.

Podsumowując pierwsze wrażenie, trzeba zaznaczyć, że drastycznie popsuły je wspomniane dwa niedopatrzenia ze strony inżynierów Owon. Łatwość, z jaką miernik „wędruje" po blacie, jest powodem do zmartwień, bo nie sądzę, by gwarancja uwzględniała uszkodzenia w wyniku upadku na podłogę. Szkoda, zwłaszcza jeśli spojrzymy na specyfikację.

Specyfikacja

XDM3051 jest w tej chwili najdokładniejszym multimetrem w ofercie firmy Owon. Wielu zaskoczyć może fakt, że porównywalne cenowo instrumenty konkurencyjnych marek wypadają nieznacznie gorzej. Najtańszym multimetrem o porównywalnych parametrach u tego samego dystrybutora jest droższy o ponad 280 złotych PeakTech P4096, co nie powinno dziwić, gdyż jest to faktycznie Owon XDM3051 sprzedawany pod inną marką. Oba urządzenia pod względem wyglądu, możliwości, parametrów, a nawet instrukcji obsługi okazują się identyczne.

Jak dokładny jest ten multimetr? Dla wszystkich zakresów napięcia stałego specyfikacja określa dokładność na ±0,015% ±0,004%. W testach porównawczych, wykonanych przez innych posiadaczy omawianego multimetru, błąd zakresu 200 mV przy pomiarze np. 100 μV wynosi jedynie ±2 μV. Dla napięć zmiennych w najbardziej użytecznym zakresie częstotliwości 45 Hz...20 kHz błąd pomiaru wynosi ±0,2% ±0,05% – w sam raz do regulacji poziomów sygnałów w układach audio, gdyż błąd jest 5...10 razy mniejszy niż w przypadku oscyloskopów cyfrowych. Poniżej 20 Hz i powyżej 20 kHz błąd już rośnie do 1%, a w zakresie 50...100 kHz – aż do 3%.

Sytuacja staje się ciekawa po spojrzeniu na zakresy pomiaru prądu. W tej kategorii cenowej tylko Owon oferuje bowiem zakres mikroamperowy, a dokładniej 200 μA z rozdzielczością 1 nA. To właśnie ten zakres pomiarowy przesądził o moim wyborze akurat modelu XDM3051. Błąd określony we wspomnianym zakresie wynosi ±0,055% ±0,005%.

REKLAMA

eprasa.pl 65e3eb482b

Przy pomiarze prądu zmiennego TrueRMS tracimy zakresy 200 µA i 2 mA. W paśmie 45 Hz...20 kHz błąd pomiaru wynosi ±0,5% ±0,1%. Maksymalny zakres pomiaru prądu stałego i zmiennego wynosi 10 A, przy prądzie powyżej 10,5 A multimetr ostrzega o przekroczeniu zakresu.

Dla zakresów rezystancji Owon podaje błąd pomiaru czteroprzewodowego lub dwuprzewodowego po wstępnym wyzerowaniu instrumentu funkcją pomiaru względnego. Podane są też wartości prądu używane do pomiarów. Dla najczęściej stosowanych zakresów 2 kΩ, 20 kΩ i 200 kΩ błąd wynosi ±0,02% ±0,003%. Zakres 200 Ω ma błąd ±0,03% ±0,005%, a 2 MΩ ±0,04% ±0,004%. Dwa ostatnie zakresy, 10 MΩ i 100 MΩ to odpowiednio: ±0,25% ±0,003% i ±1,75% ±0,004%. Specyfikację instrumentu ujęto w **tabeli 1**.

Ciekawie prezentuje się kwestia testu diody – maksymalne napięcie wynosi 2 V, co oznacza, że w przypadku większości diod LED nie poznamy wartości napięcia przewodzenia. Mimo to sprawne diody LED będą świecić. Oddzielna tabela podaje specyfikację pomiaru temperatury. Multimetr obsługuje osiem typów termopar z (niezbyt dokładną) kompensacją zimnego końca, oraz dwa popularne rodzaje platynowych czujników rezystancyjnych: PT100 i PT385. Specyfikację prezentuje **tabela 2**.

Warto jeszcze poruszyć kwestię bezpieczeństwa. Multimetr ten ma klasyfikację CAT I 1000V i CAT II 600V. Maksymalne napięcie na gnieździe Lo względem uziemienia może wynieść 500 V_{p-p}, co jest wartością wystarczającą z punktu widzenia wielu pomiarów w pracujących układach. W razie pomyłkowego podłączenia przewodu pomiarowego do gniazda pomiaru rezystancji metodą czteroprzewodową wejście to zniesie napięcie do 200 V. Maksymalne napięcie stałe, jakie multimetr może zmierzyć, wynosi 1000 V, zmienne zaś 750 V.

Tabela 1. Parametry multimetru XDM3051

Funkcja pomiaru	Zakres pomiaru	Częstotliwość lub prąd testowy	Dokładność (% pomiaru + % zakresu)
Napięcie stałe	200 mV...1000 V	...	±0,015±0,004
Napięcie zmienne	200 mV...750 V	20 Hz...45 Hz	±1,5±0,01
		45 Hz...20 kHz	±0,2±0,05
		20 kHz...50 kHz	±1±0,05
		50 kHz...100 kHz	±3±0,05
Prąd stały	200 µA	...	±0,055±0,005
	2 mA		±0,055±0,005
	20 mA		±0,095±0,02
	200 mA		±0,07±0,008
	2 A		±0,17±0,02
	10 A		±0,250±0,01
Prąd zmienny	20 mA...10A	20 Hz...45 Hz	±1,5±0,01
		45 Hz...2 kHz	±0,5±0,01
		2 kHz...10 kHz	±2,5±0,2
Rezystancja	200 Ω	1 mA	±0,03±0,005
	2 kΩ	1 mA	±0,02±0,003
	20k Ω	100 µA	±0,02±0,003
	200 kΩ	10 µA	±0,02±0,003
	2 MΩ	1 µA	±0,04±0,004
	10 MΩ	200 nA	±0,25±0,003
	100 MΩ	200 nA || 10MΩ	±1,75±0,004
Test diody	2 V	1 mA	±0,05±0,01
Test ciągłości	2000 Ω	1 mA	±0,05±0,01
Częstotliwość/okres	200 mV...750 V	20Hz...2 kHz	±0,01±0,003
		2 kHz...20 kHz	±0,01±0,003
		20 kHz...200 kHz	±0,01±0,003
		200 kHz...1 MHz	±0,01±0,006
	20 mA...10 A	20 Hz...2 kHz	±0,01±0,003
		2 kHz...10 kHz	±0,01±0,003
Pojemność	2 nF	200 nA	±3±1
	20 nF	200 nA	±1±0,5
	200 nA	1 µA	±1±0,5
	2 µF	10 µA	±1±0,5
	200 µF	100 µA	±1±0,5
	2000 µF	1 mA	±2±0,5

Praca z multimetrem i funkcje dodatkowe

Owon stworzył przejrzysty i dopracowany interfejs w tej serii multimetrów. Rozpoczęcie pracy nie wymagało zagłębiania się w instrukcję, gdyż niemal wszystko zostało przejrzyście zaprezentowane. Wybór jednej z funkcji pomiarowych wyświetla dostępne dla niej opcje konfiguracyjne, jak na przykład częstotliwość pomiarów, filtracja czy wybór zakresu (automatycznego lub ręcznego). Ze względu na tę ostatnią opcję na panelu umieszczony został dedykowany przycisk. Przycisk Dual pozwala włączyć pomiar drugiej wartości poniżej wskazania wyniku pomiaru głównego. I tak na przykład dla pomiaru napięcia zmiennego można włączyć dodatkowy pomiar częstotliwości lub okresu. Można też wyświetlać pomiary napięć stałych i zmiennych razem, pomiary prądów stałych i zmiennych razem czy dowolną kombinację pomiaru napięcia i prądu. Tu ciekawostka: instrukcja pokazuje, jak połączyć przewody pomiarowe dla każdego typu pomiaru, ale nie pokazuje kombinacji przewodów dla jednoczesnego pomiaru napięcia i prądu – ot, niedopatrzenie.

Pod przyciskiem Graph kryją się tryby wyświetlania wyniku: standardowy, skala paskowa, wykres trendu i histogram. Do każdego trybu – oprócz głównego – mamy dostępne opcje konfiguracyjne, jak na przykład zakres pokazywanych wartości. Warto dodać, że zmiana trybu wyświetlania nie wymazuje zebranych już w danym trybie próbek. Tryb skali jest użyteczny, zwłaszcza gdy zdefiniujemy własny jej zakres, przy kalibrowaniu podłączonego do instrumentu układu. Z kolei tryb wyświetlania trendu w formie wykresu przydać się może przy długoterminowym badaniu układu. W tym miejscu warto wspomnieć o menu kryjącym się pod przyciskiem Record, w którym można ustawić automatyczny lub ręczny zapis wartości do pamięci wewnętrznej lub zewnętrznej. Wartości zapisane ręcznie, tj. za pomocą przycisku Save, można wyeksportować z wewnętrznej pamięci urządzenia na nośnik USB. W przypadku pomiarów automatycznych należy najpierw podłączyć pamięć i wybrać zapis do niej. Wyniki rejestrowane są w formacie csv. Szkoda jednak, że nie można zapisać też zrzutu ekranu. Przycisk Run/Stop włącza i wyłącza akwizycję danych, co nie wiąże się z ich zapisem na nośniku. Kolejna, użyteczna funkcjonalność kryje się pod przyciskiem Math, dzięki któremu można włączyć wyświetlanie

Tabela 2. Specyfikacja pomiaru temperatury. Układ kompensacji zimnego końca dla termopar ma dokładność ±2°C

Rodzaj sondy	Typ sondy	Zakres temperatur	Dokładność
RTD	385 ppm/K	−200°C...660°C	±0,16°C
Termopara	B	20°C...1820°C	±0,76°C
	E	−270°C...1000°C	±0,5°C
	J	−210°C...1200°C	±0,5°C
	K	−270°C...1370°C	±0,5°C
	N	−270°C...1300°C	±0,5°C
	R	−70°C...1760°C	±0,5°C
	S	−50°C...1760°C	±0,6°C
	T	−270°C...400°C	±0,5°C

eprasa.pl 65e3eb482b

statystyk: wartości średniej, minimalnej, maksymalnej, zakresu wartości od minimalnej do maksymalnej, odchylenia standardowego oraz liczby zebranych próbek. Można też ustalić limity, przy których przekroczeniu multimetr zmieni kolor tła na czerwony i – jeśli te opcje są włączone – zmieni stan wyjścia Aux oraz włączy alarm akustyczny. Ostatnią funkcją dostępną w omawianym menu jest wyświetlanie wyników w dB i dBm. Dla pomiaru mocy w dBm można wybrać jedną z szesnastu wartości rezystancji odniesienia, od 50 Ω do 8 kΩ. Domyślną wartością jest 600 Ω. Dla dB trzeba również wskazać wartość odniesienia w dBm. W tym menu zdublowana została też funkcja pomiaru relatywnego, normalnie dostępna przy wyborze typu pomiaru.

W menu Utility można wybrać język, zmienić jasność wyświetlacza, ustawić datę i godzinę, zresetować multimetr, przetestować ekran i klawiaturę oraz zaktualizować firmware, choć – jak już wspomniałem wcześniej – na stronie firmy Owon nie znalazłem żadnej wersji firmware. Ostatnie menu, Port, pozwala ustawić parametry związane z portami znajdującymi się z tyłu urządzenia: funkcje gniazd BNC, ustawienia sieci LAN oraz portu szeregowego. W przypadku tego ostatniego można wybrać prędkość od 1200 baud do 115200 baud, liczbę bitów danych, bit parzystości czy bity stopu. Domyślnie ustawiono 9600, 8 bitów, brak bitu parzystości i 1 bit stopu.

Praca z multimetrem jest przyjemna – instrument okazuje się responsywny, nie zacina się ani nie zawiesza. Jak wspomniałem, ekran wyróżnia się czytelnością i nie stwarza problemu z odczytem wartości, nawet z nieco większej odległości. Jedynie statystyki wyświetlane są czcionką, która jest mniejsza nawet od czcionki w menu po prawej stronie. Z mojego punktu widzenia jest ona za mała. Ten problem może dotyczyć też starszych użytkowników. Instrukcja zaleca, by pozwolić instrumentowi pracować przez pół godziny, nim zacznie się go używać, ze względu konieczność stabilizacji temperatury; i faktycznie na początku wartości mocno „skaczą", nawet gdy sondy nie są do instrumentu podłączone. Potrzeba przynajmniej kilkunastu minut, by temperatura się ustabilizowała. Czas oczekiwania będzie wypełniony wątpliwym „umilaczem" w formie szumu wentylatora. Nie jest on najcichszy, a do tego przymocowany został do metalowej ramy – stykająca się z nią obudowa staje się dodatkowym „głośnikiem". Producent powinien był przemyśleć budowę tego elementu i dodać choćby gumowe podkładki tłumiące.

Podsumowanie, czyli: czy warto zainwestować w XDM3051?

Projektując serię XDM, zwłaszcza modele XDM3041 i XDM3051, firma Owon chciała pokazać, że jest w stanie opracować multimetr profesjonalny, który nie wygląda tanio i nie ma problemów oraz bolączek budżetowych produktów z Chin. Nie do końca to się udało, ale – pomimo drobnych niedoróbek – Owonowi udało się stworzyć pełnoprawny, profesjonalny instrument, który sprawdzi się w wielu różnych sytuacjach. Jednocześnie jest to jeden z najtańszych multimetrów 5½ cyfry, przy czym nie sprawia wrażenia taniości. Solidny, funkcjonalny interfejs to także dowód, Owon chce przekonać do siebie profesjonalistów – i w tym przypadku myślę, że mu się to udało. Jeśli kryterium wyboru nowego multimetru stołowego są parametry, możliwości i funkcjonalność, a nie logo na obudowie, to Owon XDM3051 okaże się solidnym wyborem dla zarówno profesjonalistów, jak i doświadczonych hobbystów, którzy w pełni wykorzystają jego potencjał, szczególnie przy projektowaniu i testowaniu układów o bardzo niskim poborze prądu.

Paweł Kowalczyk

REKLAMA

UWAGA! Tylko prenumeratorzy czasopism „Elektronika dla Wszystkich", „Elektronika Praktyczna", „Świat Radio" oraz „Elektronik" mogą korzystać z atrakcyjnych rabatów w Sklepie AVT:

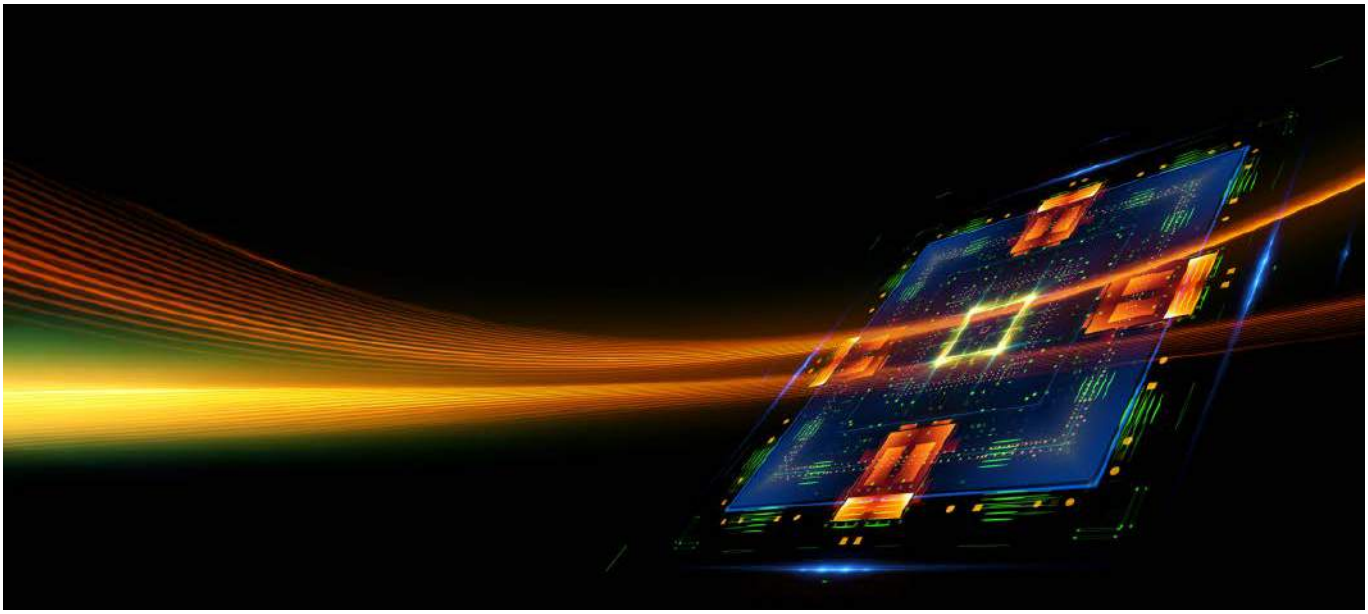
- ✓ do 50% na wydania specjalne czasopism Wydawnictwa AVT
- ✓ 20% na kity w wersji A (płytki drukowane do projektów AVT)
- ✓ 10% na pozostałe wersje kitów: (A+, B, C, D)
- ✓ 10% na książki
- ✓ 5% na pozostałe produkty z oferty sklepu

Ponadto każdy prenumerator ww. czasopism korzysta z rabatów od 30% do 50% na zakup czasopism z oferty www.UlubionyKiosk.pl

KLUB AVT ELEKTRONIKA

Jak uzyskać rabat? Podczas zamówienia powołaj się na swój numer prenumeraty – otrzymasz go mailowo po zakupie prenumeraty wraz z kartą członkowską Klubu AVT-Elektronika.

Regulamin Klubu AVT-Elektronika znajdziesz na stronie https://sklep.avt.pl/klub-avt-elektronika



Zasilanie energooszczędnych mikrokontrolerów

W miarę zwiększania zakresu zastosowań mikrokontrolerów we wszystkich rodzajach aplikacji, czynnikiem coraz istotniejszym przy ich wyborze stało się niskie zużycie energii. Wymagania dotyczące ekologii oraz lepszego wykorzystania zasobów, wydłużenie czasu pracy na baterii, etc. przekładają się na wzrost zainteresowania układami o niskim poborze mocy podczas pracy oraz trybie gotowości czy uśpienia. Z drugiej strony rosną również oczekiwania dotyczące funkcjonalności i mocy obliczeniowej, co zazwyczaj nie idzie w parze z obniżeniem zużycia energii. Przyjrzyjmy się, w jaki sposób należy wybierać i zasilać współczesne mikrokontrolery, aby zaspokoić oba te wymagania w nowoczesnych projektach cyfrowych, a także jak wymaganie to wpływa na firmware.

Współczesne urządzenia korzystające z mikrokontrolerów często zasilane są za pomocą wewnętrznego źródła energii (baterii lub akumulatorów) bądź za pośrednictwem przewodów sygnałowych. Wspólne dla tych wszystkich aplikacji jest wymaganie bardzo niskiego zużycia energii, ale jednocześnie urządzenia te często muszą się cechować wystarczającą mocą obliczeniową do spełnienia innych wymagań produktu. Niniejszy artykuł ma za zadanie zwrócić uwagę na niektóre z wyzwań projektowania współczesnych systemów mikroprocesorowych i sposób, w jaki są one zasilane.

Szacowanie zużycia energii

Aby zmaksymalizować żywotność układu przy zasilaniu bateryjnym, projektanci systemów muszą minimalizować zużycie energii przez cały okres życia urządzenia. Całkowite zużycie mocy *P* (1) i pobór energii *E* (2) są określane jako:

$$P = I \cdot U \quad (1)$$

gdzie *I* to pobierany przez układ prąd, a *U* to jego napięcie zasilania oraz

$$E = P \cdot t \quad (2)$$

gdzie *t* to czas pracy układu.

W celu oszacowania zużycia mocy lub energii przez mikrokontroler wystarczy tylko przejrzeć kartę katalogową układu, aby określić prąd pobierany przy wymaganej przez projektanta częstotliwości taktowania CPU. Należy pomnożyć ten prąd przez napięcie zasilania, a następnie przez czas pracy, aby uzyskać wartość zużytej w tym okresie energii. Na pierwszy rzut oka zadanie wydaje się zatem dosyć proste... W rzeczywistości przy takiej analizie należy wziąć pod uwagę jeszcze kilka innych elementów:

- Czy system przechodzi w tryb gotowości, gdy mikrokontroler nie jest uruchomiony?
- Czy system musi automatycznie wybudzać się np. w określonych odstępach czasu?
- Czy system dokonuje pomiarów sygnałów analogowych?
- Czy system musi rejestrować jakiekolwiek dane do analizy lub transmisji w późniejszym czasie?

W większości systemów bazujących na mikrokontrolerach odpowiedź przynajmniej na jedno z powyższych pytań brzmi „tak". Z uwagi na to prosta metoda – polegająca na obliczaniu energii z prądu i napięcia mikrokontrolera – nie zapewnia dokładnej reprezentacji zużycia przez niego energii w całym cyklu pracy. Aby uzyskać kompletny obraz tego, jak nasz układ pracować będzie przy zasilaniu bateryjnym, należy uwzględnić cztery typy zużycia energii w systemie:

Zasilanie w trybie gotowości – w typowych aplikacjach mikrokontrolery spędzają większość swojego życia w trybie gotowości. Znaczna część układu pozostaje uśpiona, a system oczekuje na sygnał do wybudzenia układu, co spowoduje uruchomienie procesora i wymaganych w danej aplikacji peryferiów. W wielu systemach zasilanych bateryjnie ten pobór energii jest największym w całym cyklu życia urządzenia.

Zasilanie peryferiów mikrokontrolera – układy te integrują w sobie wiele peryferiów umożliwiających m.in. komunikację z otoczeniem czy też pomiary zewnętrznych sygnałów. W systemach, w których

eprasa.pl 65e3eb482b

mikrokontroler odpowiedzialny jest za np. akwizycję danych z sensorów, konieczność zasilania wbudowanego w układ przetwornika analogowo-cyfrowego (ADC) może znacząco wpłynąć na całkowity pobór energii systemu.

Zasilanie podczas rejestrowania danych – w większości aplikacji mikrokontroler rejestruje dane do analizy lub w celu późniejszego przesłania ich dalej. Dane te są zapisywane w pamięci nieulotnej wbudowanej w sam układ lub dołączonej zewnętrznie. W zależności od tego, ile danych i jak często jest zapisywanych, pobór energii będzie ulegał zmianie.

Zasilanie aktywne – zużycie energii podczas aktywnego przetwarzania danych przez mikrokontroler. To wartość energii, którą układ wykorzystuje podczas normalnej pracy, tj. realizacji kodu, zapisanego w firmware.

Informacje, jakie musimy zebrać przed rozpoczęciem projektowania systemu, to – oprócz całkowitego zużycia energii, stanowiącego odzwierciedlenie średniego poboru prądu przez mikrokontroler – także chwilowe zużycie w różnych momentach pracy układu. Wykres przykładowej charakterystyki poboru prądu przez mikrokontroler w czasie pracy pokazano na **rysunku 1**.

Standardowo mikrokontrolery w układach niskomocowych przez większość czasu są w jakimś stanie uśpienia bądź standby, z wyłączonymi peryferiami i wieloma elementami wewnętrznymi, a nawet uśpionym rdzeniem, zależnie od konkretnego mikrokontrolera i jego ustawień. W tym trybie procesor pobiera z zasilacza pojedyncze mikroampery. Układ jest aktywowany tylko przez krótką chwilę i zużywa wtedy relatywnie duży prąd – typowo na poziomie kilkunastu bądź kilkudziesięciu miliamperów. Tak zróżnicowany poziom poboru energii ma spore implikacje, jeśli chodzi o sekcję zasilania, która musi wykazywać wysoką sprawność w szerokim zakresie dynamiki.

Projekt zasilacza

Z uwagi na specyficzne wymagania opisane powyżej konieczne jest dobranie odpowiedniego zasilacza do mikrokontrolera. Musi on zapewniać wymagane napięcie (lub napięcia) systemowi i maksymalizować sprawność podczas pracy w typowym dla danego systemu zakresie prądów.

W pierwszej kolejności należy wybrać napięcie zasilania. Wybór ten może wydawać się prosty: „im niższe napięcie, tym niższa moc". W pewnym sensie zdanie to jest prawdą, ale niekoniecznie zawsze znajdzie odzwierciedlenie w rzeczywistej pracy urządzeń. Wiele procesorów ma wewnętrzny stabilizator napięcia, obniżający napięcie i oferujący stałe zużycie energii w przeważającej części zakresu wejściowych napięć zasilania. Dzięki temu zasilanie układu napięciem 3 V lub 5 V (a w niektórych przypadkach nawet 1,8 V) nie wpływa na wynikowe zużycie energii przez mikrokontroler. Oczywiście zasilanie całego systemu 3 V może dobrze wpłynąć na całkowite zużycie energii, ale jest to wybór zależny od obwodów znajdujących się poza samym mikrokontrolerem.

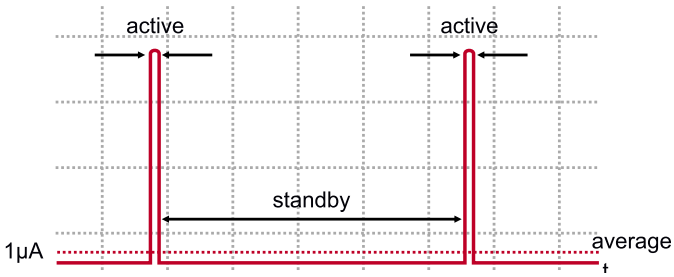
Obecnie wiele układów pracować może przy bardzo niskich napięciach – na przykład procesor RL78/L12 firmy Renesas jest w stanie działać przy zasilaniu od 1,6 V. Dzięki tak szerokiemu zakresowi napięć realne okazuje się wydłużenie czasu pracy na baterii lub akumulatorze bez konieczności częstej wymiany lub ładowania źródła energii. Warto zauważyć, że maksymalna częstotliwość taktowania układu jest często zależna od napięcia i podczas pracy z bardzo niskim napięciem zasilania niemożliwe będzie osiągnięcie maksymalnej mocy obliczeniowej układu. Z tego względu należy zachować ostrożność przy konfiguracji zegara przy zasilaniu procesora napięciami od 1,8 V do 3,3 V i upewnić się, że częstotliwość taktowania jest odpowiednia dla danego napięcia. Maksymalna częstotliwość wewnętrznego oscylatora w przypadku wspomnianego RL78/L12 przy 1,6 V może wynosić od 1 MHz do 4 MHz (dla porównania – w zakresie od 2,7 V do 5,5 V układ może już pracować z taktowaniem od 1 MHz do 24 MHz).

Dodatkowo należy pamiętać, że możliwe jest okresowe obniżenie napięcia zasilania układu podczas jego bezczynności – niższe napięcie pozwala podtrzymać zawartość rejestrów i pamięci RAM, jednakże rozwiązanie takie komplikuje projekt systemu zasilania i wymaga zachowania ostrożności przy tworzeniu oprogramowania (m.in. sekwencjonowania napięć podczas wyłączania i włączania systemu z zachowaniem odpowiednich opóźnień, aby nie wpłynąć na układy zewnętrzne podłączone do mikrokontrolera). Opisane podejście przekłada się także na wydłużenie czasu uruchamiania i wybudzania układu ze stanu uśpienia. Ostateczna decyzja co do zastosowania koncepcji sekwencjonowania napięć wynika z prostej kalkulacji opłacalności – czy warto tak komplikować układ w celu np. wydłużenia czasu pracy na jednym naładowaniu akumulatora?

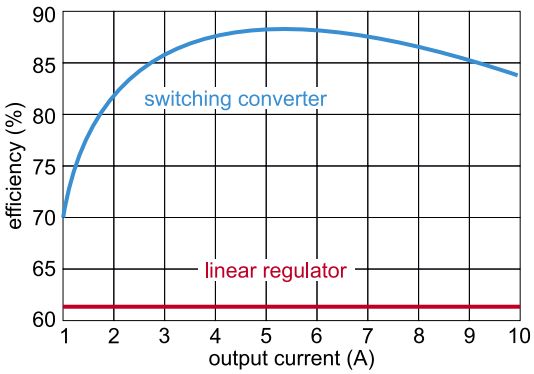
Drugim aspektem, który należy wziąć pod uwagę podczas projektowania sekcji zasilania, jest szeroki zakres prądu wyjściowego – od pojedynczych mikroamperów do nawet kilkudziesięciu miliamperów (lub więcej, jeśli weźmiemy pod uwagę również zasilanie pozostałych układów w systemie). Nie jest to problemem dla stabilizatorów liniowych, jednakże ich zastosowanie wiąże się z relatywnie niską sprawnością systemu, szczególnie jeśli napięcie zasilania mikrokontrolera jest dużo niższe od napięcia wejściowego. Znacznie wyższą sprawność (nawet ponad 95%) zaoferować mogą przetwornice impulsowe, jednakże charakteryzują się one zazwyczaj kiepską sprawnością przy niewielkich obciążeniach, jak pokazano na **rysunku 2**.

Jednym ze sposobów rozwiązania tego problemu okazuje się równoległe dołączenie do stabilizatora impulsowego układu LDO, którego sprawność wprawdzie jest niższa niż typowa sprawność stabilizatora impulsowego, ale w przypadku niewielkich prądów zależność ta odwraca się. Wystarczy znaleźć punkt przecięcia obu charakterystyk, jak pokazano na **rysunku 3** i przełączać pomiędzy dwoma stabilizatorami, np. wykorzystując ich wejście zezwalające.

Rozwiązanie z użyciem LDO oraz stabilizatora impulsowego nie należy oczywiście do najprostszych i należy pamiętać, że konieczne jest przełączanie stabilizatorów przez zewnętrzne układy (np. sam mikrokontroler). Można przy okazji takiego rozwiązania ustawić również dwa różne napięcia na wyjściach obu układów.

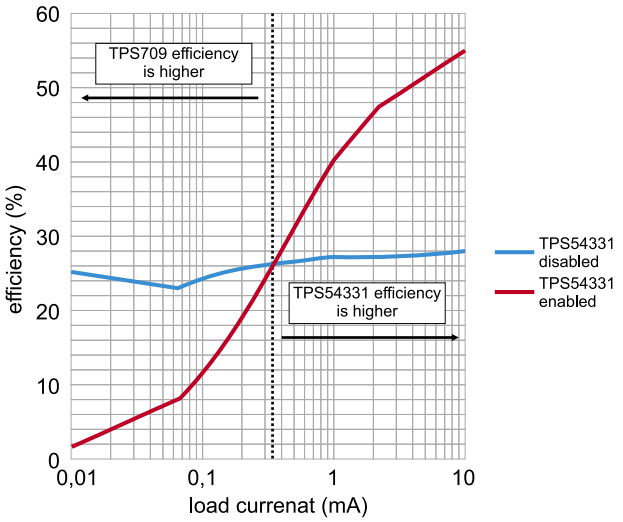


Rysunek 1. Wykres poboru prądu typowego mikrokontrolera w funkcji czasu



Rysunek 2. Przykładowe krzywe sprawności stabilizatora liniowego (linia czerwona) oraz impulsowego (linia niebieska) w funkcji pobieranego prądu

eprasa.pl 65e3eb482b



Rysunek 3. Sprawność stabilizatora LDO TPS709 (linia niebieska) i impulsowego TPS54331 (linia czerwona) w funkcji obciążenia. Widać, że powyżej około 0,3 mA stabilizator impulsowy ma wyższą sprawność niż LDO

Oczywiście można również wybrać stabilizator impulsowy zaprojektowany pod kątem wysokiej sprawności przy niewielkich obciążeniach. Uwagę należy tutaj zwrócić głównie na obecność funkcji tzw. pulse skipping, czyli pomijania impulsów. Modulacja ta jest jedną z technik zmniejszania strat przełączania przy zachowaniu dobrej stabilizacji napięcia wyjściowego. Tryb pomijania impulsów polega na utrzymywaniu tranzystorów MOSFET w stanie wyłączenia przez kilka cykli przełączania, przez co straty związane z kluczowaniem w układzie sterowania tranzystorem MOSFET ulegają zredukowaniu, a ogólna sprawność wzrasta. Przetwornice wyposażone w taką opcję idealnie nadają się do aplikacji w systemach bateryjnych.

Architektura oprogramowania

Aby zoptymalizować pobór energii w systemie, programiści powinni rozważyć użycie szeregu funkcji i możliwości w oprogramowaniu mikrokontrolera. Zaprezentowana poniżej lista nie jest wyczerpująca, ale zawiera szereg kluczowych koncepcji, które warto rozważyć poza wspomnianymi wcześniej trybami uśpienia itp.

Wykonywanie oprogramowania z pamięci nieulotnej lub RAM

Tworząc system, należy dokładnie przeanalizować, czy oprogramowanie mikrokontrolera ma być wykonywane z pamięci nieulotnej, czy operacyjnej układu – ogromnie wpływa to na pobór prądu. Uruchamianie kodu z pamięci RAM oferuje zwykle niższe zużycie energii, jednak praktyczne aplikacje okazują się często zbyt obszerne, by móc wykonywać je tylko z pamięci RAM.

Włączone lub wyłączone zegary magistrali

Funkcjonowanie mikrokontrolera wymaga dostępu do pamięci i bloków peryferyjnych, co z kolei generuje potrzebę włączenia licznych zegarów dla ich magistrali i musi zostać uwzględnione przy szacowaniu aktywnego prądu. Z tego też powodu istotne okazuje się wyłączanie zegarów peryferiów nieużywanych w danym momencie pracy MCU.

DMA i CIP

Wiele mikrokontrolerów wyposażonych jest w kontroler bezpośredniego dostępu do pamięci (DMA), który pozwala peryferiom na wykonywanie pewnych działań niezależnie od aktywności rdzenia. Rozwiązanie takie umożliwia znaczne obniżenie prądu na poziomie systemu. Ważne, aby rozważyć, czy jest to możliwe w przypadku konkretnego peryferium lub zadania, ponieważ można w ten sposób zaoszczędzić znaczną ilość prądu.

Pewne rozwinięcie opisanego podejścia stanowią tzw. Core Independent Peripherals (CIP), obecne w 8-bitowych układach PIC i AVR firmy Microchip. Są one peryferiami mikrokontrolera, oferującymi dodatkowe funkcjonalności do obsługi różnorodnych zadań, bez konieczności interwencji CPU. Działanie modułów CIP procesora bazuje na osobnych instrukcjach wprowadzonych do jego pamięci. Dzięki takiej automatyzacji moduły te nie obciążają CPU i pozwalają mu na realizację innych zadań lub przejście w stan bezczynności.

Akceleracja

Mikrokontrolery przeznaczone do aplikacji o niskim zużyciu energii często wyposażone są w specjalne akceleratory, które obniżają liczbę cykli oraz energii wymaganej do realizacji określonych zadań. Na przykład: szyfrowanie AES256 może wymagać nawet 7 000 cykli klasycznego procesora, a akcelerator sprzętowy realizuje ten sam proces w zaledwie ok. 500 cykli. W tym przypadku akceleracja sprzętowa może zredukować czas aktywnego przetwarzania ponad 10-krotnie, a dodatkowo wiele akceleratorów może działać asynchronicznie, dzięki czemu – w czasie gdy realizują one swoje zadanie – procesor może wykonywać inne instrukcje, jeszcze wyraźniej skracając czas aktywności.

Używanie zoptymalizowanego kodu

Kod mikrokontrolera powinien zostać zoptymalizowany pod kątem najkrótszego czasu wykonania, za pomocą odpowiedniej struktury kodu i optymalizacji kompilatora. Jakkolwiek zalecenie to może brzmieć trywialnie i być oczywiste, to – jak pokazuje praktyka – często w pośpiechu i nawale prac projektowych optymalizacja kodu jest traktowana bardzo powierzchownie. Redukcja czasu realizacji zadań w trybie aktywnym procesora pozwala na istotne zmniejszenie zużycia energii przez układ.

Podsumowanie

Zasilanie mikrokontrolerów o niskim poborze mocy nie jest proste, podobnie jak tworzenie całych systemów, które mają zużywać mało prądu. W obecnych czasach jednak coraz więcej aplikacji stawia takie wymagania. Systemy zasilane bateryjnie, rozproszone sensory, urządzenia IoT etc. – wszędzie tam znajdziemy systemy mikroprocesorowe zoptymalizowane do zużywania jak najmniejszej ilości energii.

Powyższy artykuł jest jedynie zarysowaniem pewnego szerszego tematu, ale mam nadzieję, że pozwoli na wprowadzenie w zagadnienie optymalizacji zużycia energii w systemach wykorzystujących mikrokontrolery. Oprócz opisanych powyżej podstaw, warto zapoznać się z notami aplikacyjnymi mikrokontrolerów różnych producentów, które dotyczą stosowania ich układów w aplikacjach o niskim bądź ultraniskim poborze prądu. Istnieje też wiele rodzin mikrokontrolerów specjalnie zaprojektowanych w tym celu, takich jak STM32L0/4/5 oraz – niedawno dodana – seria STM32U5, PIC32CM Lx00, AVR z serii DD czy wreszcie układy takie, jak MAX32670 lub C8051F98x. Procesory te często wyposażone są w dosyć unikatowe rozwiązania, pozwalające zredukować pobór prądu w systemie.

Nikodem Czechowski, EP

Źródła:
1. Jacob Borgeson, Stefan Schauer, Horst Diewald „Benchmarking MCU power consumption for ultra-low-power applications", Texas Instruments White Paper SLAY023, listopad 2012
2. Michael Hartshorne „How to Improve Buck Converter Light Load Efficiency with an LDO" Powerhouse, Texas Instruments, 2014
3. Sitthipong Angkititrakul, Haitao Hu, „Design and analysis of buck converter with pulse-skipping modulation", 2008 IEEE Power Electronics Specialists Conference

eprasa.pl 65e3eb482b

Nowy enkoder przyrostowy FAULHABER IEP3

Enkodery inkrementalne odgrywają kluczową rolę w nowoczesnych systemach sterowania prędkościowego i pozycyjnego. Najnowszy enkoder IEP3 marki FAULHABER umożliwia niezwykle precyzyjną kontrolę miniaturowych napędów komutatorowych oraz krokowych, stosowanych w najbardziej wymagających aplikacjach.

Imponująca dokładność dzięki najnowszym zdobyczom technologii półprzewodnikowej

Schönaich – FAULHABER – poszerza swoją ofertę produktów o enkoder przyrostowy IEP3, który dzięki zastosowaniu najnowocześniejszych układów scalonych, gwarantuje doskonałą rozdzielczość i dokładność. Niezwykle lekki i kompaktowy enkoder IEP3, o średnicy zewnętrznej korpusu równej zaledwie 8 mm, cechuje się niebywale wysoką rozdzielczością sięgającą 10 000 linii na obrót, zapewnianą przez ultranowoczesne chipy o wysokim stopniu interpolacji. W wersji standardowej rozdzielczość można zaprogramować w zakresie od 1 do 4096 linii na obrót. Co więcej zastosowana technologia półprzewodnikowa gwarantuje niezwykłą precyzję pozycjonowania z dokładnością do 0,3°m, jak również wysoką powtarzalność rzędu 0,05°m, uzyskaną dzięki wbudowanej funkcji kompensacji dokładności.

Uniwersalność w każdym calu

Enkoder IEP3 może być zasilany zarówno napięciem 5 V, jak i 3,3 V, dzięki czemu doskonale nadaje się do użycia w urządzeniach akumulatorowych, a także wszelkich innych systemach wbudowanych o standardowym napięciu zasilania równym 3,3 V. Szeroki zakres temperatur pracy (od –40 do +125°C) dodatkowo poszerza zakres docelowych zastosowań. Enkoder IEP3 może być wykorzystywany w wielu zróżnicowanych aplikacjach, przede wszystkim w branżach wymagających jednocześnie wysokiej dokładności i wyjątkowo kompaktowej konstrukcji. Należą do nich m.in. protetyka (nowoczesne protezy medyczne), optyka (układy przestrajania i pozycjonowania w teleskopach, mikroskopach, aparaturze laserowej czy kamerach), przemysł półprzewodników i robotyka.

Więcej informacji:
FAULHABER Polska sp. z o.o.
60-204 Poznań, ul. Górki 7
tel. +48 61 278 72 53, info@faulhaber.pl
https://www.faulhaber.com

Montowany na osi enkoder z 2-biegunowym magnesem ma prostą, trwałą i niezawodną konstrukcję. Dzięki modułowej budowie, charakterystycznej dla enkoderów FAULHABER należących do serii IE3, sensory IEP3 stanowią platformę bez problemu współpracującą z wieloma modelami silników. Omawiany model zapewnia pełną kompatybilność z napędami komutatorowymi (DC) z serii 0816SR, 1016SR i 1024SR oraz z silnikami krokowymi, należącymi do linii produktowych AM0820, AM1020 i AM1524.

W ofercie marki Faulhaber znajduje się ponadto wiele akcesoriów do wykonywania podłączeń elektrycznych, w tym złącza oraz gotowe kable PVC lub FEP o różnych długościach, które znacznie skracają czas integracji docelowych systemów i redukują koszty, związane z samodzielną lub zleconą produkcją wiązek przewodowych.

Podsumowanie najważniejszych zalet enkoderów IEP3:

- wysoka rozdzielczość do 10 000 linii na obrót,
- najnowocześniejsza technologia półprzewodnikowa z funkcją kompensacji dokładności, zapewniająca wysoką precyzję pozycjonowania i powtarzalność,
- niska masa własna i kompaktowe wymiary,
- kompatybilność z napędami z serii 0816SR, 1016SR, 1024SR AM0820, AM1020 i AM1524 marki FAULHABER.

eprasa.pl 65e3eb482b

Nowoczesne materiały termoprzewodzące

Problemy związane z chłodzeniem urządzeń stają się tym istotniejsze, im intensywniej postępuje miniaturyzacja w elektronice. Efektywniejsza kontrola termiczna często pozwala bowiem na uzyskanie większej niezawodności, a w efekcie – przedłużenie żywotności sprzętu. Właśnie z tego względu dobór właściwych materiałów termoprzewodzących ma niezwykle istotne znaczenie w procesie projektowania systemu chłodzenia.

Jeżeli komponent wytwarzający ciepło jest zaizolowany, jego temperatura wzrasta aż do osiągnięcia stanu równowagi, czyli wyrównania się ilości ciepła wytwarzanego przez ten element z ciepłem oddawanym do otoczenia. Jeśli temperatura równowagi termicznej będzie zbyt wysoka, czas życia elementu ulegnie drastycznemu skróceniu, a w skrajnym przypadku dojdzie nawet do awarii urządzenia. Większość elementów elektronicznych podczas swojej pracy generuje przeważnie niewielką ilość ciepła. Jednakże niektóre komponenty – jak tranzystory i diody mocy czy też procesory komputerowe – wytracają znaczną ilość energii i wymagają podjęcia pewnych środków w celu przedłużenia ich żywotności oraz zwiększenia niezawodności.

Szybkość odprowadzania ciepła jest większa przy przemieszczaniu się mas powietrza niż podczas pracy w warunkach statycznego otoczenia. Jedną z możliwości kontroli temperatury będzie zatem zainstalowanie wentylatorów, umożliwiających wzmocnienie efektu konwekcji (taka technika chłodzenia nosi nazwę konwekcji wymuszonej). Zresztą nawet zainstalowanie ogólnej wentylacji skutkuje obniżeniem temperatury pracy urządzeń – w porównaniu do ilości ciepła wydzielanego przez taką samą instalację w pomieszczeniu bez otworów wentylacyjnych.

Fotografia 1. HTC – bezsilikonowa pasta termoprzewodząca (0,9 W/m·K) w opakowaniu o masie netto równej 1 kg

Więcej informacji:
Dystrybutorem produktów opisanych w artykule jest firma **Semicon Sp. z o.o.**
04-761 Warszawa, ul. Zwoleńska 43/43A
tel. +48 22 615 83 40 e-mail: info@semicon.com.pl
www.semicon.com.pl

Ciepło z wnętrza komponentu jest oddawane do otoczenia przez powierzchnię jego obudowy. Szybkość strat ciepła zwiększa się wraz z polem tej powierzchni – małe urządzenie o mocy 10 W osiągnie wyższą temperaturę niż podobne, ale w większej obudowie. Jednym ze sposobów na chłodzenie podzespołów będzie więc zwiększenie ich efektywnej powierzchni, co można uzyskać przez dołączenie metalowych radiatorów. Radiatory mogą być wytwarzane metodą tłoczenia lub wylewania, zazwyczaj z miedzi, aluminium lub ich stopów (z uwagi na doskonałe przewodzenie ciepła). Elementy te często mają odpowiednie użebrowanie, ponieważ dzięki temu można uzyskać większą powierzchnię rozpraszającą ciepło do otoczenia.

Wytworzenie idealnie dopasowanych powierzchni komponentu i radiatora jest w praktyce niemożliwe, więc zestawienie mikroskopijnych nierówności obydwu elementów spowoduje powstanie licznych szczelin powietrznych, obejmujących sumarycznie sporą część całkowitego pola styku. Powietrze – jako mierny przewodnik ciepła – wytwarza termiczną barierę ograniczającą efektywność chłodzenia. Aby przezwyciężyć ten problem, stosuje się materiały termoprzewodzące, przeznaczone do wypełniania prześwitu pomiędzy chłodzonym komponentem a radiatorem, co redukuje opór termiczny, prowadząc do szybszej utraty ciepła i obniżenia temperatury pracy urządzenia.

Na rynku istnieje wiele rodzajów materiałów termoprzewodzących. Jednym z czołowych producentów past termoprzewodzących jest firma Electrolube, wytwarzająca materiały na bazie wypełniaczy mineralnych o wysokiej przewodności cieplnej, zawieszonych w płynnym nośniku w postaci oleju silikonowego lub mineralnego. Pasty na bazie silikonowej – takie jak HTS czy HTSP – mogą pracować w wyższych temperaturach niż ich odpowiedniki bezsilikonowe (HTC, HTCP).

Należy pamiętać, że silikony mają tendencję do migracji, dlatego mogą powodować problemy w niektórych aplikacjach, zanieczyszczając np. styki przekaźnikowe.

Modyfikacja przewodności termicznej past jest możliwa poprzez zmianę ilości wypełniaczy lub ich składu. Wersje „P" wyżej wymienionych past mają wyższą zawartość wypełniaczy oraz odpowiednią ich kompozycję, a rozwiązanie takie służy zwiększeniu efektywności transferu ciepła. Zazwyczaj wyższa zawartość wypełniaczy wiąże się też z większą lepkością past, co może prowadzić do trudności przy ich dozowaniu.

Ulepszoną wersją HTC jest pasta HTCX, która charakteryzuje się podwyższoną przewodnością

Fotografia 2. Pasta HTCP w kartuszu o pojemności 20 ml

eprasa.pl 65e3eb482b

Fotografia 3. Silikonowa pasta termoprzewodząca typu HTS

cieplną, niższym wypływem oleju i mniejszymi stratami masy na skutek odparowania.

Najwyższą przewodność termiczną (do 3,40 W/mK), przy jednoczesnym utrzymaniu zalet oleju bezsilikonowego, zapewnia pasta HTCPX, która swoje wyjątkowe własności zawdzięcza użyciu sproszkowanych tlenków różnych metali.

Warto dodać, że zdolność pasty do utrzymania swojej początkowej konsystencji w czasie długotrwałej eksploatacji znacząco ułatwia demontaż komponentów w przypadku naprawy lub modyfikacji urządzenia.

W niektórych okolicznościach wymagane jest jednak pełne utwardzenie materiału termoprzewodzącego. Produkty Electrolube pod nazwą TCER i TCOR są materiałami silikonowymi utwardzanymi w temperaturze pokojowej i wypełnionymi mieszanką mineralną – wprowadzone pomiędzy radiator a obudowę elementu, żelują do konsystencji gumy. W przypadku TCER proces ten zachodzi pod wpływem wilgoci otoczenia i powoduje wydzielanie etanolu. Pasta łączy właściwości podkładek kauczukowych oraz tradycyjnych past termoprzewodzących. Dzięki niskiej lepkości materiał jest łatwy w aplikacji również przy użyciu pistoletu. Podobne cechy ma TCOR, który – żelując pod wpływem wilgoci – wydziela obojętny oksym. Produkt ten, oprócz typowych zastosowań do rozpraszania ciepła, może służyć jako klej o niedużej sile adhezji, izolator lub uszczelniacz.

Skład kleju TBS tego samego producenta stanowi dwuskładnikowa żywica epoksydowa, która utwardza się do stanu stałego i efektywnie przykleja radiator do komponentu. W niektórych urządzeniach taki sposób mocowania jest korzystny, ale wiąże się z problemami przy demontażu.

Przy wszystkich materiałach termoprzewodzących bardzo istotne jest zapewnienie dokładnego wypełnienia na styku radiatora z urządzeniem i wyparcie całego zgromadzonego w szczelinie powietrza. Efekt taki uzyskuje się zwykle przez centralne nałożenie pasty na powierzchnię styku radiatora lub urządzenia i równoległe złączenie obu powierzchni. Ilość nakładanego materiału musi być kontrolowana, tak aby wystarczyło go do pełnego usunięcia niepożądanego gazu, ale bez niepotrzebnego nadmiaru pasty wyciskanego przy krawędziach. Osiąga się to, stosując automatyczne dozowanie i urządzenia montażowe.

Im wyższa jest przewodność termiczna pasty czy żywicy, tym mniejszy opór cieplny, a tym samym niższa temperatura pracy komponentu. Materiały termoprzewodzące mają niższą przewodność cieplną niż radiator, a więc wskazane jest utrzymanie minimalnej możliwej grubości warstwy. Oczywiście nie można dopuścić do istnienia w niej szczelin powietrznych – kontrolowanie grubości warstwy jest możliwe poprzez wmieszanie do pasty albo żywicy małych kuleczek szklanych, które ustalają odległość łączonych powierzchni. Kontakt elementu z radiatorem można ponadto poprawić, stosując śruby lub zaciski.

Fotografia 4. Silikonowa pasta termoprzewodząca typu HTSP

Znając przewodność cieplną materiału termoprzewodzącego, grubość jego warstwy i powierzchnię styku z radiatorem, można obliczyć opór cieplny na granicy stykających się powierzchni, a tym samym – temperaturę równowagi cieplnej, zaś znajomość wartości wydzielanej mocy pozwala przewidzieć temperaturę osiąganą przez radiator. Kontrolę termiczną w obwodach

REKLAMA

Materiały termoprzewodzące

SEMICON®

WACKER

ELECTROLUBE
THE SOLUTIONS PEOPLE

– Kleje i pasty termoprzewodzące
– Doskonała wytrzymałość na rozciąganie
– Pasty termoprzewodzące z przewodnością do 5,5 W/mK
– Szeroki zakres temperaturowy przy małych stratach na odparowanie
– Nie utwardza się, umożliwia naprawę elementów

Innowacyjne produkty
Innowacyjne technologie

Semicon Sp. z o.o.
ul. Zwoleńska 43/43a, 04–761 Warszawa | 22 615–73–71
www.semicon.com.pl | zamowienia@semicon.com.pl

dużej mocy można uzyskać także przez umieszczenie elektroniki w puszce metalowej z użebrowaniem, a następnie zalanie jej materiałem termoprzewodzącym. Firma Electrolube produkuje wiele takich materiałów, z których najbardziej popularna jest dwuskładnikowa żywica ER 2183. Również w tym przypadku istotne jest, aby w trakcie zalewania nie zostało wmieszane powietrze, które ograniczałoby transfer ciepła do obudowy metalowej. Jeżeli nie jest wymagana szczególnie wysoka przewodność zalewy, można użyć żywicy ogólnego przeznaczenia, np. ER 2188.

Najwyższą ochronę w wielu aplikacjach – łącznie z błyskawicznie rozwijającym się przemysłem oświetlenia LED – zapewnia żywica ER 2220. Jej przewodność termiczna wynosi 1,54 W/m·K, a temperatura pracy mieści się w zakresie –400°C...+1300°C. Jest to żywica samogasnąca, spełniająca wymogi normy UL 94V-0.

Oprócz żywic epoksydowych, coraz większe znaczenie zyskują także materiały silikonowe, których maksymalna temperatura pracy dochodzi do 2000°C. Najlepszym przykładem jest tu dwuskładnikowa, samogasnąca żywica SC2003.

Żywice poliuretanowe wykazują wprawdzie mniejszą odporność temperaturową, ale utrzymują swoje doskonałe parametry elektryczne i przewodność termiczną w trudnych warunkach, np. morskich czy tropikalnych. Typowym przykładem tutaj jest żywica UR 5633.

Wypełniacze mineralne mają wyższe przewodnictwo termiczne niż baza żywiczna, co oznacza, że materiały z domieszką wypełniaczy są lepszymi przewodnikami ciepła niż żywice pozbawione wypełniaczy. Co więcej – im wyższa jest ich zawartość, tym lepsza przewodność termiczna. Niestety, wraz z ilością wypełniaczy rośnie lepkość, a to wiąże się z kolei z większym ryzykiem wmieszania powietrza w zalewę.

Obecnie na rynku znajduje się również wiele materiałów termoprzewodzących innych producentów. Należy wspomnieć tu o takich firmach, jak np. Momentive, Polytec-PT czy Cotronics. W ofercie marki Momentive dostępne są wysokolepkie kleje TSE 3281 i TSE3882, wydzielające przy żelowaniu produkty oksymowe i mające termoprzewodność na poziomie 2 W/m·K. Żaden z tych produktów nie powoduje korozji powierzchni metalowych.

Firma Polytec-PT produkuje kleje termoprzewodzące, w tym TC 420 i TC 430. Pierwszy z nich jest dwuskładnikowym klejem epoksydowym o konsystencji miękkiej pasty, odpornym na długotrwałe działanie temperatury do 2000°C i dobrych parametrach dielektrycznych. Wśród jego zastosowań wymienić należy odprowadzanie ciepła przy montażu półprzewodników, w optoelektronice, technice lotniczej i przemyśle elektronicznym.

TC 430 to również dwuskładnikowy klej epoksydowy, ale o dłuższym czasie żelowania niż TC 420. Jest on stosowany do „bondingu" struktur

Fotografia 5. Dwuskładnikowa żywica epoksydowa typu TBS

Fotografia 6. ELASTOSIL RT675 – dwuskładnikowa, termoprzewodząca żywica silikonowa w kolorze czerwonobrązowym

półprzewodnikowych w mikroelektronice i wytrzymuje długotrwałe działanie temperatury do 2500°C.

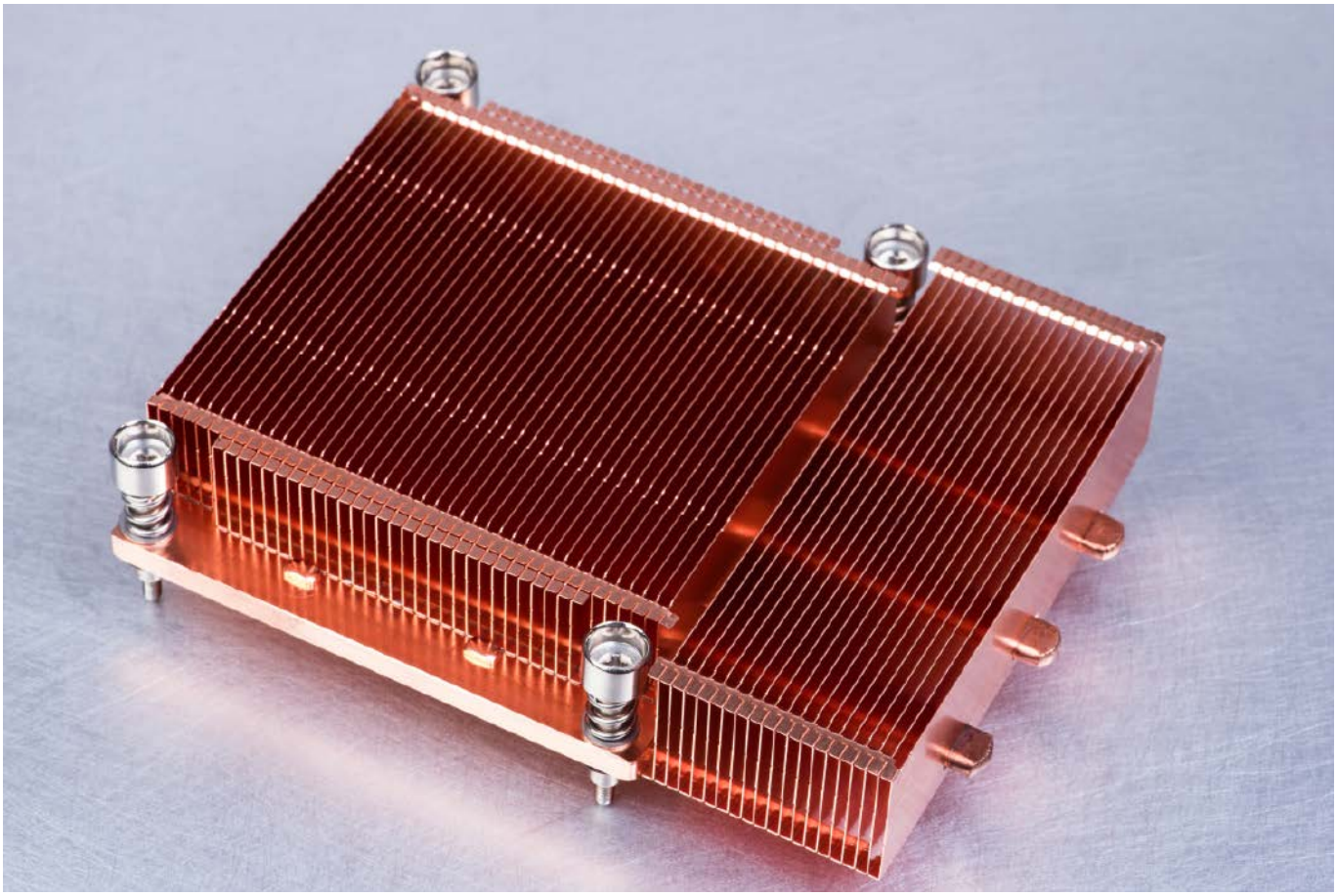
Przy omawianiu materiałów termoprzewodzących nie sposób nie wspomnieć o ceramicznych klejach wysokotemperaturowych firmy Cotronics, które – oprócz wytrzymałości na temperatury powyżej 1000°C – w niektórych przypadkach wykazują także doskonałą termoprzewodność. Przykładem tutaj może być Resbond 903HP – jednoskładnikowy klej na bazie trójtlenku glinu, odporny na temperatury do 1800°C. Może być nakładany przy użyciu pędzla, szpachli i urządzeń natryskowych. Przeznaczony jest do łączenia nieporowatych materiałów ceramicznych, szkła i niereaktywnych metali. Resbond 908 jest natomiast dwuskładnikowym klejem (także na bazie trójtlenku glinu), odpornym na temperatury do 1650°C oraz wilgoć. Stanowi doskonały izolator elektryczny, stosowany do łączenia, zalewania i hermetyzacji delikatnych elementów elektronicznych, czujników oraz urządzeń pracujących w wysokich temperaturach.

Resbond 920 z kolei to dwuskładnikowy klej ceramiczny na bazie trójtlenku glinu, który oprócz doskonałej izolacji elektrycznej oferuje również odporność na temperatury do 1650°C oraz większość chemikaliów i rozpuszczalników. Żeluje w temperaturze pokojowej i jest przeznaczony do klejenia rezystorów wysokotemperaturowych, pirometrów, elementów pieców grzewczych itp.

Do innych, bardzo specjalistycznych metod kontroli termicznej, należą chłodzenie cieczą oraz chłodzenie modułami Peltiera. Pierwsza z wymienionych metod polega na cyrkulacji płynu chłodniczego w bliskiej odległości od nagrzewającego się elementu; ciecz znacznie łatwiej przewodzi bowiem ciepło niż powietrze, a ponadto może być rozprowadzana poprzez sieć przewodów. Pary chłodziwa, powstające w pobliżu komponentu, przemieszczają się do chłodnych obszarów zamkniętego układu, gdzie ulegają kondensacji. Zaletą tej metody jest fakt, że utajone ciepło parowania cieczy okazuje się bardzo efektywne przy chłodzeniu komponentów – ta sama zasada fizyki jest zresztą powszechnie wykorzystywana w lodówkach.

Zjawisko Peltiera natomiast obserwuje się przy przepływie prądu stałego przez złącze dwóch różnych półprzewodników – jeżeli prąd płynie przez moduł w określonym kierunku, jedna strona elementu ogrzewa się, zaś druga ulega ochłodzeniu. Wszystkie takie podzespoły wymagają użycia materiałów termoprzewodzących na powierzchni styku z chłodzonym elementem, co pozwala w pełni wykorzystać efektywność modułów Peltiera. Gorąca strona systemu chłodniczego może także wymagać użycia radiatora w celu rozproszenia odprowadzonego ciepła do otoczenia urządzenia.

eprasa.pl 65e3eb482b



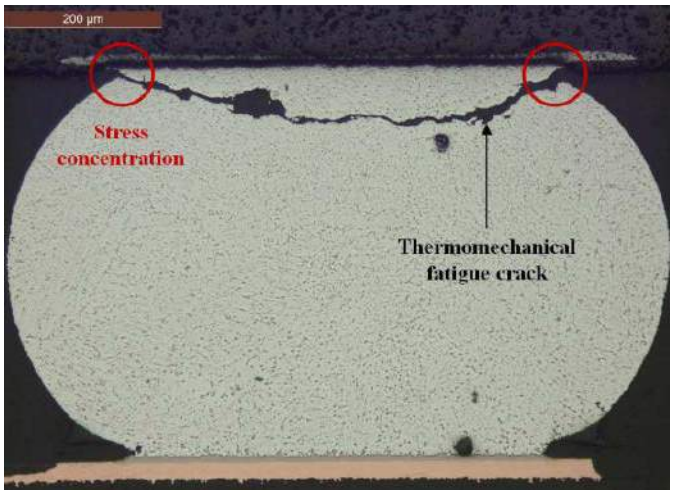
Chłodzenie w urządzeniach elektronicznych

– przewodnik projektanta

Odprowadzanie ciepła z komponentów elektronicznych ma fundamentalne znaczenie dla niezawodności urządzeń oraz bezpieczeństwa – tak samego sprzętu, jak i korzystających z niego użytkowników. W artykule omawiamy szeroki wachlarz dostępnych na rynku rozwiązań, zwracając przy tym uwagę na rozmaite aspekty związane z procesem projektowania systemów chłodzenia.

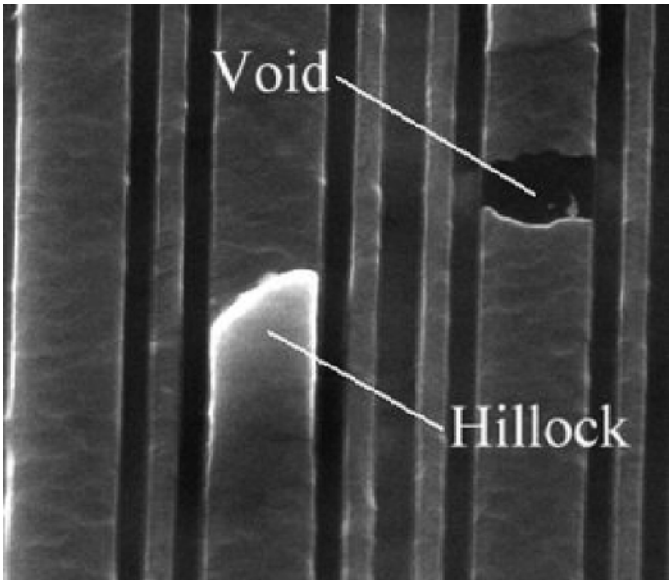
Zanim przejdziemy do opisu konkretnych zagadnień fizycznych, technologicznych oraz projektowych, warto na początek nakreślić praktyczne znaczenie chłodzenia we współczesnej elektronice. Dokonajmy zatem ekspresowego przeglądu negatywnych efektów nadmiernego wzrostu temperatury elementów i urządzeń elektronicznych.

1. **Wpływ temperatury na niezawodność połączeń lutowanych** – liczne badania materiałowe udowodniły wpływ narażeń termomechanicznych (tj. naprężeń mechanicznych „wspieranych" przez zmieniającą się w szerokim zakresie temperaturę) na powstawanie uszkodzeń połączeń lutowanych (**fotografia 1**). O ile bowiem temperatura otoczenia sama w sobie jest kluczowa dla niezawodności urządzeń elektronicznych, to silne, lokalne przegrzewanie układów scalonych czy też dyskretnych podzespołów półprzewodnikowych może dodatkowo zwiększać wahania temperatury, prowadząc do szybszego powstawania pęknięć w objętości lutowia.



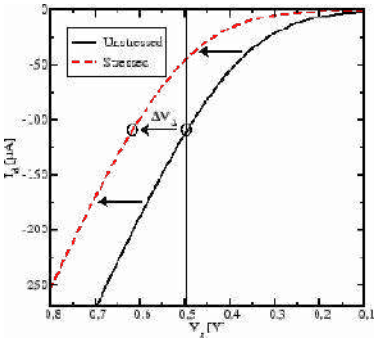
Fotografia 1. Lut układu BGA uszkodzony w wyniku zmęczenia termomechanicznego (http://t.ly/8XsT8)

eprasa.pl 65e3eb482b



Fotografia 2. Przykładowe uszkodzenia ścieżek przewodzących w strukturze cyfrowego układu scalonego: akumulacja materiału (po lewej) oraz przerwa w obwodzie (po prawej) – http://t.ly/xfRKw

2. **Termiczne przyspieszanie starzenia półprzewodników** – za awarie komponentów elektronicznych powstałe w wyniku ich nadmiernego nagrzewania się podczas pracy odpowiada wiele mechanizmów. Sporo zjawisk, zachodzących w samym półprzewodniku bądź w metalowych połączeniach struktur krzemowych (tzw. *bonding*), jest niejako „katalizowanych" przez temperaturę, a największy udział w starzeniu komponentów ma elektromigracja. Proces ten polega na przemieszczaniu atomów materiału, które są niejako „porywane" przez ruch elektronów, co prowadzi do lokalnych dysproporcji objętościowych – gdy w jednym miejscu przewodnika ubywa, w innym powstaje jego nadmiar, wynikający z akumulacji przemieszczonych atomów. W ten sposób w strukturze lub połączeniach powstają zwężenia i poszerzenia, a w skrajnych przypadkach – przerwy bądź zwarcia (**fotografia 2**). Co gorsza, proces ten znacząco przyspiesza zarówno wraz z lokalnym wzrostem gęstości prądu (czemu sprzyja postępująca miniaturyzacja, zwłaszcza w zakresie komponentów dużej mocy), jak i ze wzrostem temperatury pracy (**rysunek 1**). Podwyższona temperatura przyspiesza także wystąpienie uszkodzenia w wyniku tzw. przebicia starzeniowego dielektryka (TDDB, ang. *Time-Dependent Dielectric Breakdown*). Jakby tego było mało, przegrzewanie struktur komponentów półprzewodnikowych powoduje również zmiany w ich parametrach elektrycznych – przykładem może być efekt starzeniowy, określany mianem NBTI (ang. *Negative Bias Temperature Instability*) i polegający na stopniowym podwyższaniu napięcia progowego bramek tranzystorów MOSFET-P (a ściślej rzecz ujmując – przesuwaniu całej charakterystyki w stronę bardziej ujemnych potencjałów – patrz **rysunek 2**). NBTI prowadzi do zmniejszania transkonduktancji, a w efekcie do obniżenia prądu drenu tranzystora, pracującego w tych samych warunkach układowych (tj. przy określonych napięciach V_{GS} oraz V_D). Co gorsza, powyższa lista wciąż nie wyczerpuje potencjalnych konsekwencji pracy półprzewodników w podwyższonych temperaturach.

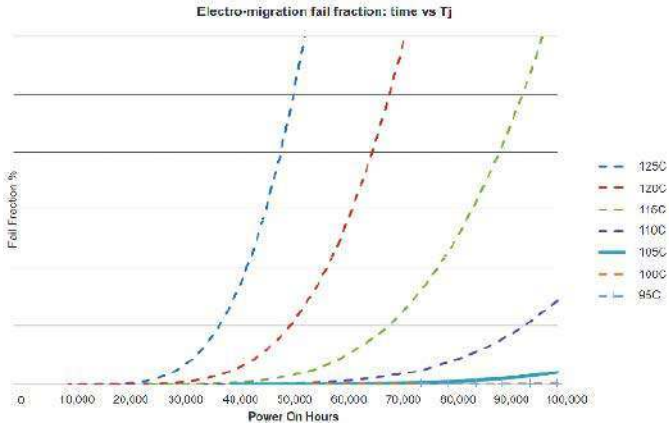


Rysunek 2. Ilustracja efektu Negative Bias Temperature Instability (NBTI) – przesunięcie charakterystyki prądu drenu w funkcji napięcia bramka-źródło (http://t.ly/f5yTg)

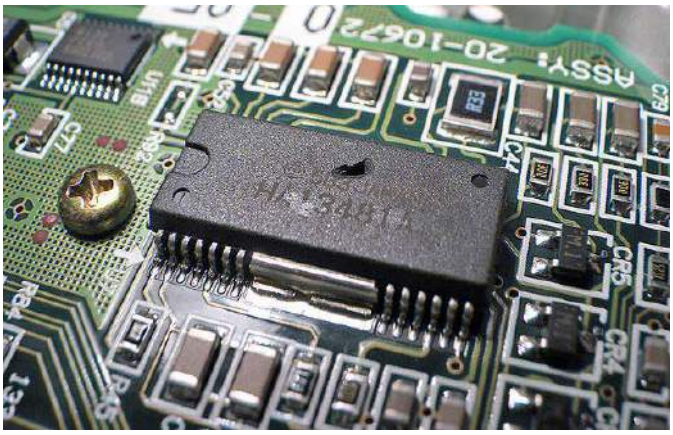
3. **Krytyczne uszkodzenia termiczne** – przegrzanie elementów elektronicznych podczas pracy w temperaturach przekraczających dopuszczalne przez producenta warunki (**tabela 1**) może w prosty sposób doprowadzić do „efektownych" oraz – niestety – fatalnych w skutkach uszkodzeń i to zarówno samych podzespołów, jak i płytek drukowanych, na których się one znajdują. Chyba każdemu elektronikowi zdarzyło się w swojej karierze trafić na układ scalony, tranzystor lub inny element półprzewodnikowy, w którego obudowie widoczny był głęboki otwór, świadczący o „wybuchu" korpusu pod wpływem nadmiernego wzrostu ciśnienia (**fotografia 3**). W przypadku płytek drukowanych awaria może przebiegać na drodze delaminacji, zarówno warstw

Tabela 1. Definicje temperaturowych klas elementów elektronicznych określone przez firmę Renesas (http://t.ly/C59mH)

Klasa temperaturowa komponentu	Zakres temperatur [°C]
Comm (komercyjny)	0...+70
Hi-Temp Comm (komercyjny wysokotemperaturowy)	–10...+100
Extended Comm (komercyjny rozszerzony)	–20...+85
Ind (przemysłowy)	–40...+85
Automotive (motoryzacyjny)	–40...+105
Full-Range Ind (przemysłowy pełnozakresowy)	–40...+125
Power Supply (przeznaczony do układów zasilania)	–40...+130
Mil (wojskowy)	–55...+125

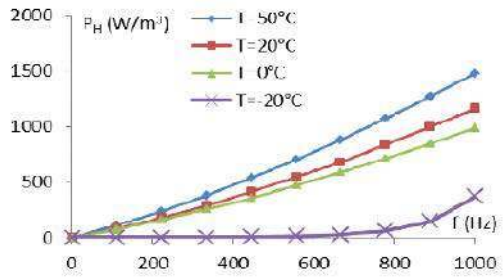


Rysunek 1. Zależność awaryjności procesorów marki Texas Instruments od czasu pracy – dane wykreślone dla siedmiu różnych temperatur struktury (http://t.ly/kcbhD)

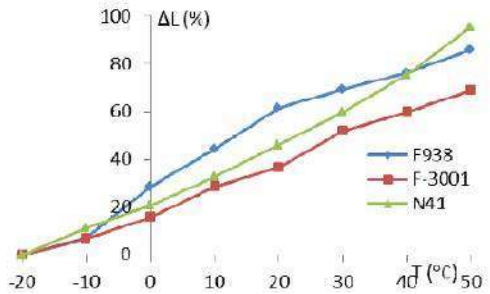


Fotografia 3. Przykład uszkodzenia obudowy układu scalonego na skutek intensywnego przegrzania struktury półprzewodnikowej (http://t.ly/2LCcZ)

eprasa.pl 65e3eb482b



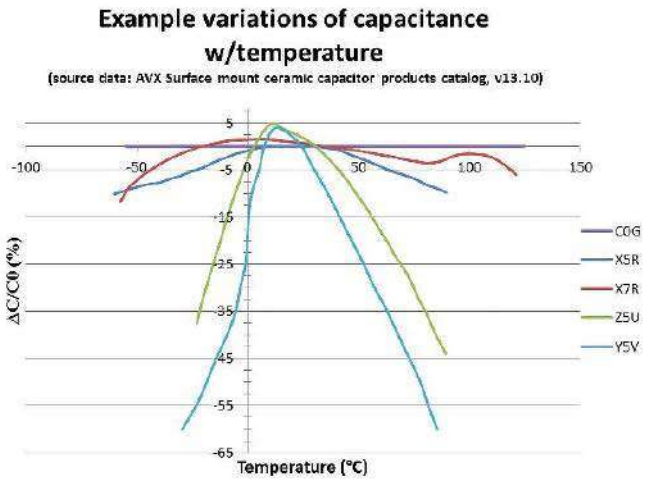
Rysunek 3. Wpływ temperatury na straty mocy w rdzeniu ferrytowym typu F938 (http://t.ly/qTLG5)



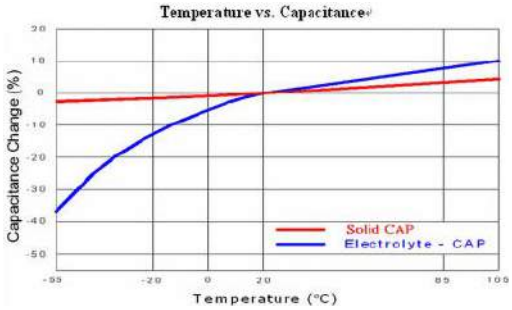
Rysunek 4. Wpływ temperatury na indukcyjność dławików ferrytowych z rdzeniami typu F938, F-3001 oraz N41 (http://t.ly/qTLG5)

zewnętrznych (oderwanie padu lub padu ze ścieżką, uszkodzenie soldermaski, etc.), jak i w głębszych strukturach PCB. I choć większość tego typu problemów wynika zwykle z poważnych awarii funkcjonalnych (np. zwarcia lub przeciążenia w obwodach dużej mocy) bądź nieprawidłowości w procesie lutowania rozpływowego, to niedostatecznie chłodzony komponent (a w szczególności ten całkowicie pozbawiony niezbędnego radiatora) może w skrajnych przypadkach doprowadzić do podobnych uszkodzeń, choć zwykle na nieco mniejszą skalę.

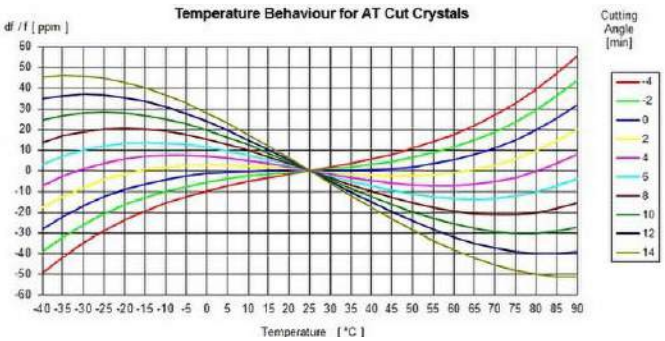
4. **Wpływ temperatury na parametry komponentów** – opisane powyżej zmiany starzeniowe, zachodzące w półprzewodnikach pod wpływem nadmiernej temperatury pracy, są niestety nieodwracalne i stopniowo degradują struktury komponentów. W codziennej praktyce znacznie częściej bierzemy jednak pod uwagę efekty przejściowe, obecne tylko w podwyższonych warunkach termicznych. Najprostszy przykład stanowią rezystory, dla których producenci zawsze określają temperaturowy współczynnik rezystancji (TCR), będący miarą dryfu oporności w funkcji temperatury. Temperatura wpływa także na komponenty indukcyjne,



Rysunek 5. Charakterystyki temperaturowe kondensatorów ceramicznych MLCC wykonanych w oparciu o pięć popularnych klas dielektryków (http://t.ly/6yxfr)



Rysunek 6. Charakterystyki temperaturowe kondensatorów z elektrolitem stałym i ciekłym (http://t.ly/UFIEc)

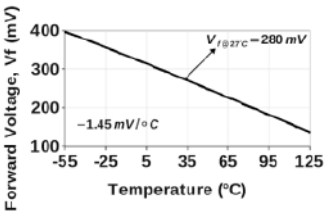


Rysunek 7. Pęk charakterystyk temperaturowych rezonatorów kwarcowych. Parametrem jest kąt cięcia wyrażony w minutach kątowych (http://t.ly/sD_JB)

zwłaszcza na poziom strat w rdzeniu (**rysunek 3**) oraz indukcyjność (**rysunek 4**).

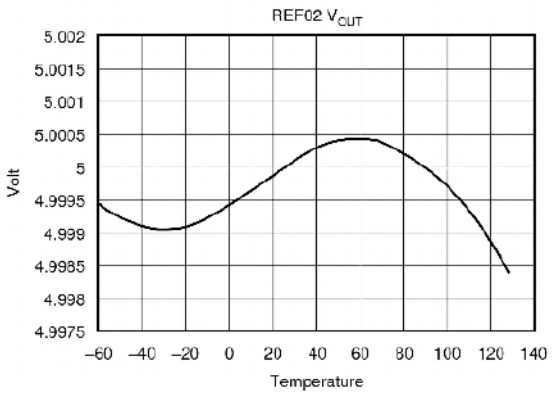
W świecie kondensatorów zróżnicowanie efektów termicznych jest jeszcze większe. Co ciekawe, w przypadku kondensatorów ceramicznych (oczywiście z wyjątkiem tych z dielektrykiem C0G, tj. o niemal zerowej wrażliwości na zmiany temperatury), charakterystyki nie są nawet monotoniczne w pełnym zakresie dopuszczalnych warunków termicznych (patrz **rysunek 5**). Zależność pojemności od temperatury staje się widoczna także wśród kondensatorów elektrolitycznych – znacznie stabilniejsze termicznie od klasycznych kondensatorów z elektrolitem ciekłym są elementy wykonane na bazie elektrolitu stałego (**rysunek 6**). Na marginesie warto też dodać, że efekty starzenia termicznego tych pierwszych należą do najważniejszych czynników wpływających na ich niezawodność, a wynika to wprost z odparowywania, czy nawet wycieków elektrolitu w warunkach podwyższonej temperatury otoczenia.

Wymienione przykłady to zaledwie początek długiej listy odwracalnych zmian, jakie zachodzą w komponentach elektronicznych pod wpływem podwyższonej temperatury pracy. Powszechnie znana jest chociażby zależność dewiacji częstotliwości rezonatorów kwarcowych od temperatury, co ciekawe – silnie związana z zastosowanym kątem cięcia kryształu (**rysunek 7**). Sytuacja jest znacznie bardziej złożona w przypadku elementów półprzewodnikowych – te wykazują bowiem nie tylko zależność szeregu parametrów elektrycznych od temperatury, ale także różnią się kształtem charakterystyki. O ile prostsze elementy cechują się raczej monotoniczną (a nawet liniową) zależnością od temperatury (jako przykład można podać napięcie przewodzenia diody Schottky'ego – patrz **rysunek 8**), to już bardziej

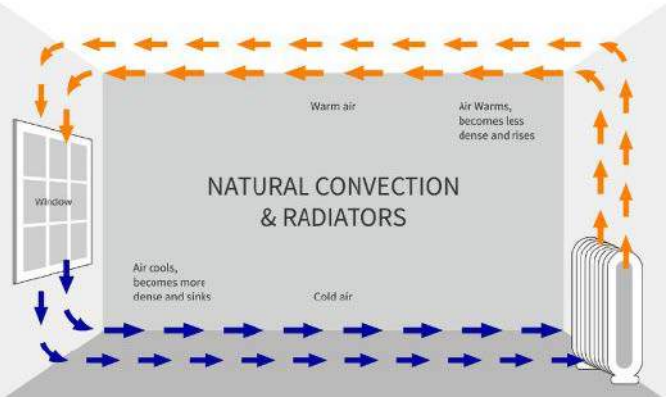


Rysunek 8. Charakterystyka napięcia przewodzenia diody Schottky'ego w funkcji temperatury złącza (http://t.ly/ayQB5)

eprasa.pl 65e3eb482b



Rysunek 9. Charakterystyka termiczna skompensowanego, scalonego źródła napięcia odniesienia typu REF02 (http://t.ly/rQCBA)



Rysunek 10. Ilustracja zjawiska konwekcji swobodnej na przykładzie pokoju ogrzewanego za pomocą kaloryfera (http://t.ly/36iaM)

złożone układy scalone, wyposażone w obwody kompensacyjne, mogą wykazywać dość złożone przebiegi charakterystyki termicznej, wynikające wprost z zastosowanej konstrukcji układu (**rysunek 9**).

5. **Zagrożenia dla użytkowników** – niedostateczne chłodzenie elektroniki w urządzeniach mobilnych (np. laptopach czy smartfonach) może w skrajnych przypadkach prowadzić do oparzeń i podrażnień skóry u użytkowników. Problem ten, nawet jeśli nie występuje w natężeniu istotnym z medycznego punktu widzenia, niemal zawsze wywołuje irytację – każdy, kto miał okazję przez dłuższy czas pracować na laptopie o obudowie zbyt silnie nagrzewającej się w rejonie otoczenia touchpada bądź klawiatury doskonale wie, że na dłuższą metę taka praca jest po prostu nieprzyjemna. Konsekwencją rozgrzewania się obudowy urządzenia jest zwykle pogorszenie wrażeń z użytkowania, co z kolei rzutuje negatywnie na opinie o samym producencie sprzętu. Ze znacznie poważniejszymi konsekwencjami mogą natomiast wiązać się istotne klinicznie obrażenia, wynikające np. z przypadkowego dotknięcia gorącego radiatora bądź obudowy wzmacniacza dużej mocy. W skrajnych przypadkach przegrzanie urządzenia może doprowadzić do pożaru, a jeśli w wewnątrz obudowy znajduje się akumulator (w szczególności litowy) – mamy gotowy przepis na technologiczną katastrofę.

Powróćmy do fizyki, czyli o metodach transferu ciepła

Każdą z metod chłodzenia urządzeń elektronicznych można rozpatrywać pod kątem praktycznego zastosowania podstawowych zjawisk, związanych z drogami transportu energii. Fundamentalne znaczenie praktycznie we wszystkich przypadkach ma **przewodzenie**, polegające na przepływie ciepła od ciała (lub ośrodka) o wyższej temperaturze do ciała/ośrodka chłodniejszego. To właśnie przewodzenie jest podstawą działania radiatorów, materiałów termoprzewodzących, czy też rozmaitych „termicznych" struktur wspomagających chłodzenie na poziomie PCB. Mało tego – można bez zbędnej przesady stwierdzić, że opisywany sposób przekazu ciepła toruje drogę pozostałym efektom – konwekcji oraz promieniowaniu.

Konwekcja swobodna polega na przemieszczaniu się mas gazu (zwykle powietrza) lub cieczy (np. w systemach chłodzenia wodnego), ogrzanych przez kontakt z gorącym elementem, ku górze (**rysunek 10**). Za proces ten odpowiadają subtelne zmiany w lokalnej gęstości danego ośrodka. Unoszenie ogrzanych warstw gazu (cieczy) niejako „przygotowuje miejsce" do uzupełnienia przez otaczające masy o niższej temperaturze. Proces ten zachodzi samoczynnie i tym intensywniej, im wyższa jest różnica temperatur pomiędzy ośrodkiem a źródłem ciepła.

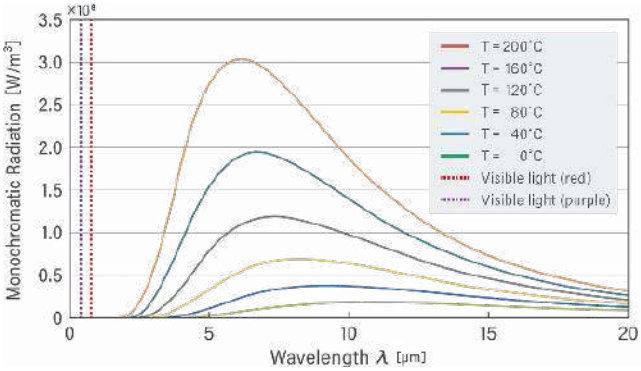
Konwekcja wymuszona to w istocie „sztucznie podkręcona" konwekcja swobodna – w tym przypadku, zamiast zdać się na działanie praw fizyki i grę różnic w gęstości ośrodka, niejako pomagamy w ewakuacji warstw ogrzanego medium do innych obszarów obudowy urządzenia lub po prostu do otoczenia. Podstawowym elementem konstrukcyjnym systemów chłodzenia z konwekcją wymuszoną jest wentylator, pozwalający na efektywne wydmuchiwanie gorącego powietrza z wnętrza obudowy lub wtłaczanie powietrza chłodnego z jej otoczenia. Zastosowanie konwekcji wymuszonej wielokrotnie redukuje wymaganą objętość obudowy w porównaniu do konstrukcji o tej samej mocy strat, ale wyposażonej tylko w chłodzenie pasywne (radiatory) – korzystając z pomocy wentylatora możemy bowiem uniknąć gromadzenia się wewnątrz urządzenia gorącego powietrza, zdolnego w skrajnym przypadku do niemal całkowitego zahamowania konwekcji swobodnej (na skutek wyrównania temperatur).

Promieniowanie termiczne to utrata energii na skutek emisji podczerwieni, głównie w jej średnim oraz bliskim (w przypadku ciał o bardzo wysokiej temperaturze) przedziale widma. Rozkład spektralny promieniowania zależy od temperatury ciała i to zarówno pod względem amplitudy, jak i położenia piku charakterystyki widmowej na osi poziomej, reprezentującej długość fali (**rysunek 11**). Matematyczną reprezentację tego przesunięcia opisuje prawo Wiena: długość emitowanej fali λ_{max} w piku amplitudy jest odwrotnie proporcjonalna do temperatury T ciała:

$$\lambda_{max} = \frac{b}{T}$$

gdzie b – stała przesunięć Wiena równa w przybliżeniu $2{,}89777 \cdot 10^{-3}$ [m·K]

W praktyce na ilość ciepła wytracanego drogą promieniowania wpływ ma także względna zdolność emisyjna ciała, będąca ilorazem tzw. zdolności emisyjnej danego obiektu oraz zdolności emisyjnej ciała doskonale czarnego (w codziennych zastosowaniach, np. w pirometrach cyfrowych, tę bezwymiarową wielkość zwykło się oznaczać grecką literą ε). Nie wchodząc w zawiłości aparatu matematycznego opisującego zjawisko promieniowania termicznego powiemy tylko, że w przypadku ciała doskonale czarnego ε miałby wartość równą 1, zaś wszelkie rzeczywiste obiekty mają ten parametr niższy. O ile? To zależy (i to w bardzo dużym stopniu) od rodzaju powierzchni. Przykładowo – czyste, błyszczące aluminium

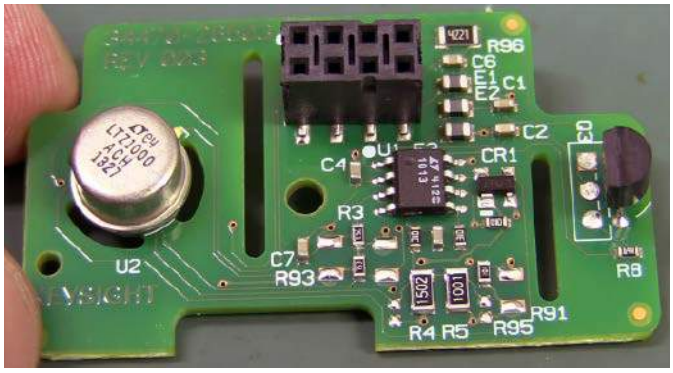


Rysunek 11. Wykres promieniowania termicznego ciała doskonale czarnego (ilustracja prawa przesunięć Wiena) – http://t.ly/toFfR

eprasa.pl 65e3eb482b

ma emisyjność na poziomie zaledwie 0,04, podczas gdy w przypadku tego samego metalu poddanego anodowaniu wartość ta rośnie do 0,9. W dalszej części artykułu przyjrzymy się praktycznemu zastosowaniu różnych wykończeń powierzchni w produkcji radiatorów.

Wszystkie cztery opisane powyżej sposoby przekazu ciepła są najczęściej wymieniane jednym tchem w opracowaniach dotyczących metod chłodzenia urządzeń elektronicznych. Warto jednak wspomnieć o jeszcze jednym zjawisku, które – choć w praktyce wiąże się także z przewodzeniem oraz konwekcją – w rzeczywistości stanowi niejako „nową jakość" we współczesnych systemach odprowadzania ciepła. Mowa o metodach opartych na **przemianie fazowej**, a dokładnie na dwóch jej rodzajach: parowaniu oraz skraplaniu. Jeżeli na powierzchni nagrzewającego się komponentu (lub połączonego z nim elementu chłodzącego o małej rezystancji termicznej) umieścimy warstwę cieczy, to w procesie jej odparowywania zostanie „zużyta" spora ilość wytraconej przez podzespół energii. Chłodzenie tego typu sprawdza się w urządzeniach, w których duża ilość ciepła jest generowana w niewielkiej przestrzeni (np. w laptopach czy komputerach stacjonarnych).



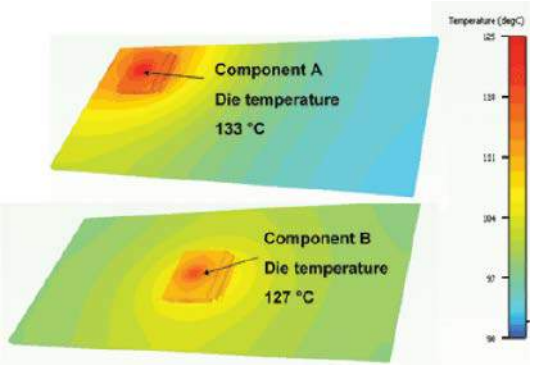
Fotografia 5. Moduł generatora napięcia odniesienia z precyzyjnym źródłem LTZ1000ACH oraz obwodami peryferyjnymi. Widoczne są wycięcia (sloty) wokół głównego układu scalonego oraz na drodze pomiędzy nim, a resztą układu (w szczególności złączem rastrowym). Wycięcia mają na celu odizolowanie czułego układu źródeł ciepła przewodzonego przez ścieżki oraz (częściowo) sam laminat (http://t.ly/O6jpk)

Ograniczenia w projektowaniu systemów chłodzenia

W projektowaniu systemu chłodzenia urządzenia elektronicznego należy uwzględnić szereg ograniczeń, które w największym stopniu wpływają na zakres możliwych do zastosowania rozwiązań konstrukcyjnych.

- **Hermetyczność obudowy** wymagana może być zarówno od niewielkich urządzeń przenośnych i ubieralnych, jak i sprzętu stacjonarnego (np. sensorów IoT przeznaczonych do pracy w otwartej przestrzeni). W znakomitej większości takich przypadków możliwości konstruktorów ograniczone zostają do rozwiązań pasywnych – wewnętrznych radiatorów, połączeń termicznych z obudową oraz ciepłowodów (omówimy je dokładniej w dalszej części artykułu). Problemy związane z chłodzeniem urządzeń wodoszczelnych wynikają oczywiście z faktu, że ciepło może być odprowadzone z wnętrza sprzętu wyłącznie za pomocą przewodzenia oraz promieniowania (zaś to ostatnie dotyczy tylko obudowy).
- **Uwarunkowania wymiarowe** – ciasne przestrzenie wewnątrz obudowy znacznie redukują swobodę projektantów w zakresie wyboru metodologii chłodzenia. Najlepiej wiedzą o tym projektanci smartfonów, którzy nieustannie zmagają się z problemem za „krótkiej kołdry" – podczas gdy wydajność procesorów stosowanych w telefonach cały czas rośnie, to niezmiennie konieczne okazuje się utrzymywanie znikomej grubości całkowitej urządzeń. Na domiar złego, od nowoczesnych urządzeń mobilnych wymaga się zwykle pewnego poziomu wodoszczelności, niewielkiej masy własnej, a także gęstego upakowania dziesiątków podzespołów – modułów, przewodów FPC, anten, cewek ładowania bezprzewodowego, kamer, głośników i wielu innych. Producenci smartfonów uciekają się więc do rozmaitych rozwiązań – termicznej optymalizacji obudowy, stosowania materiałów termoprzewodzących, czy nawet… rurek ciepła wypełnionych cieczą (**fotografia 4**).
- **Uwarunkowania geometryczne** – w wielu przypadkach odprowadzenie umiarkowanej ilości ciepła jest możliwe przy sporym udziale naturalnego przewodzenia nadmiaru energii z nagrzewającego się komponentu do najbliższego otoczenia konstrukcyjnego. Dobrym przykładem będzie tutaj płytka drukowana, której warstwy miedzi stanowią całkiem wydajny radiator dla komponentów o niezbyt wysokich wymaganiach w zakresie chłodzenia. Należy jednak pamiętać, że w przypadku precyzyjnych układów pomiarowych takie „udzielone" ciepło, pochodzące z dalszych obszarów PCB, może zaburzać pracę wzmacniaczy, oscylatorów i innych układów relatywnie wrażliwych na temperaturę. Stąd też w najwyższej klasy urządzeniach pomiarowych można znaleźć rozmaite wycięcia w PCB (sloty), umieszczone wokół czułego źródła napięcia odniesienia – przykład pokazano na **fotografii 5**. Warto w tym miejscu dodać, że w kwestii efektywności chłodzenia znaczenie ma nawet umiejscowienie elementu na PCB – dobra praktyka nakazuje ustawienie nagrzewającego się komponentu możliwie jak najbliżej środka płytki, gdyż im bliżej brzegu, tym trudniej jest uzyskać odpowiednie rozproszenie ciepła w strukturach PCB (**rysunek 12**).
- **Praca w próżni** – aparatura pracująca w otwartej przestrzeni kosmicznej jest narażona nie tylko na nadmiar ciepła wynikający z własnych strat mocy – równie istotnym zagrożeniem okazują się wahania temperatury, zwłaszcza w przypadku satelitów krążących relatywnie blisko Słońca (a więc m.in. wszystkich orbiterów ziemskich i księżycowych). Odprowadzanie ciepła z wnętrza urządzeń jest niemożliwe przy użyciu konwekcji, stanowiącej główną metodę wytracania nadmiaru energii w warunkach ziemskich. Do dyspozycji pozostaje więc tylko przewodzenie (od źródeł ciepła

Fotografia 4. Wnętrze smartfona Samsung Galaxy S7 wraz z zastosowaną w nim rurką ciepła (http://t.ly/TKLoC)



Rysunek 12. Wpływ umiejscowienia nagrzewającego się układu scalonego na maksymalną temperaturę jego struktury (http://t.ly/aBGL0)

eprasa.pl 65e3eb482b

Fotografia 6. Przykładowy system chłodzenia sprzętu kosmicznego (http://t.ly/AU2Jd)

do radiatorów i innych elementów chłodzących) oraz promieniowanie (z zewnętrznych powierzchni). W aplikacjach kosmicznych można zatem znaleźć głównie płyty chłodzące, specjalne systemy na bazie tzw. oscylacyjnych rurek ciepła (**fotografia 6**), a także moduły termoelektryczne (Peltiera). W każdym przypadku jednak usunięcie nadmiaru ciepła poza urządzenie może opierać się już tylko na promieniowaniu termicznym.

- **Koszty** – konwencjonalne systemy chłodzenia oparte na konwekcji swobodnej (radiatory, obudowy metalowe z interfejsem termicznym, np. przekładką termoprzewodzącą) oraz wymuszonej (radiatory + wentylatory) są przeważnie rozwiązaniami najtańszymi spośród wszystkich dostępnych. Nieco wyżej pod względem ceny plasują się np. rurki ciepła (ciepłowody), najwyżej zaś w cenowej hierarchii znajdują się rozbudowane, aktywne chłodnice z wymuszonym obiegiem cieczy oraz systemy termoelektryczne.

Podstawowy model termiczny typowego układu chłodzenia

Fundamentalne znaczenie dla chłodzenia elementów elektronicznych ma minimalizacja **rezystancji termicznej** – parametr ten można najprościej scharakteryzować jako iloraz różnicy temperatur ΔT (którą dany obiekt, np. radiator czy przekładka termoprzewodząca, „wprowadza" do układu podczas przepływu przezeń strumienia ciepła) oraz mocy (P), która ów przepływ ciepła wywołuje. Stąd też wynika praktyczna jednostka rezystancji termicznej (R_{th}), czyli [°C/W] lub [K/W]:

$$R_{th} = \frac{\Delta T}{P}$$

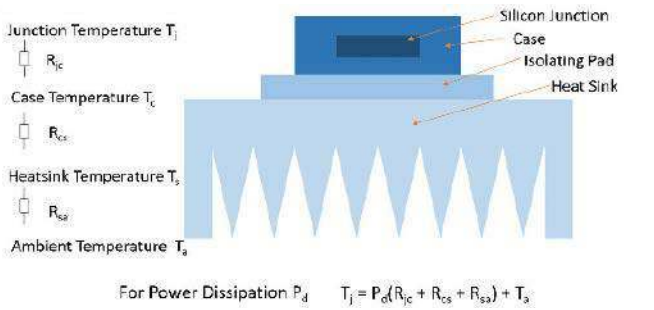
Z pojęciem rezystancji termicznej wiąże się bezpośrednio także przewodność cieplna (k). Spójrzmy na następujące równanie:

$$R_{th} = \frac{l}{k \cdot A}$$

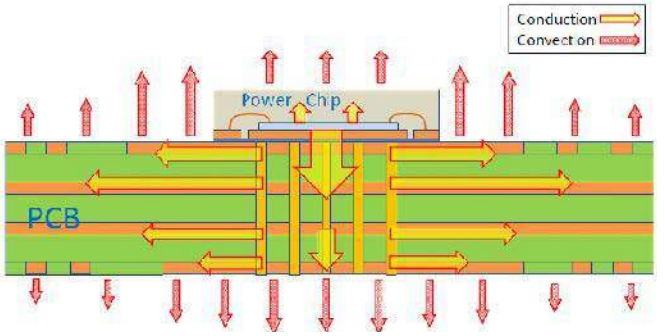
gdzie:
l – długość drogi przepływu ciepła w próbce materiału o przewodności cieplnej k [W/(m·K)],
A – pole przekroju obiektu, przez które przepływa strumień ciepła.

Już pobieżna analiza powyższych zależności jasno wskazuje, że rezystancja termiczna jest w istocie parametrem konkretnego **obiektu**, wykonanego z określonego typu materiału, zaś przewodność cieplna to parametr typowy dla samego **materiału** – a więc niezależny od jego wymiarów i geometrii. Nic zatem dziwnego, że komponenty służące do chłodzenia urządzeń i elementów elektronicznych (np. radiatory) mają określoną przez producenta rezystancję termiczną, zaś w przypadku materiałów (np. past termoprzewodzących) najważniejszym parametrem jest właśnie przewodność cieplna.

Warto zwrócić uwagę na fakt, że rezystancja termiczna stanowi swego rodzaju odpowiednik rezystancji elektrycznej, zaś strumień



Rysunek 13. Uproszczony model termiczny układu komponent-przekładka termoprzewodząca-radiator (http://t.ly/CAx-N)

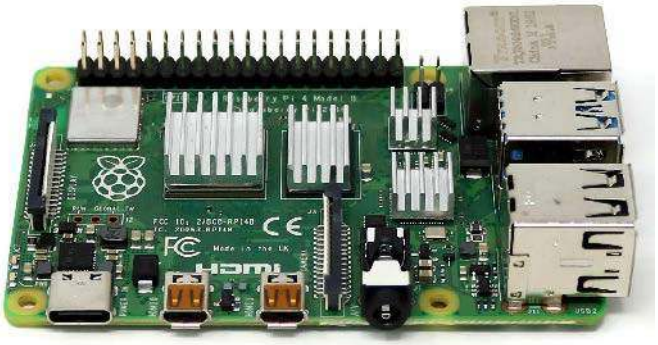


Rysunek 14. Schematyczne przedstawienie dróg odbioru ciepła z komponentu SMD zamontowanego na płytce drukowanej przy użyciu padu z przelotkami termicznymi (http://t.ly/_npKR)

ciepła odpowiada… natężeniu prądu. Z tej analogii, tak intuicyjnej zwłaszcza dla elektroników, wynika uproszczona metoda obliczania temperatury struktury półprzewodnikowej na podstawie znajomości mocy strat w chłodzonym komponencie oraz rezystancji termicznych następujących „warstw" układu, tj:

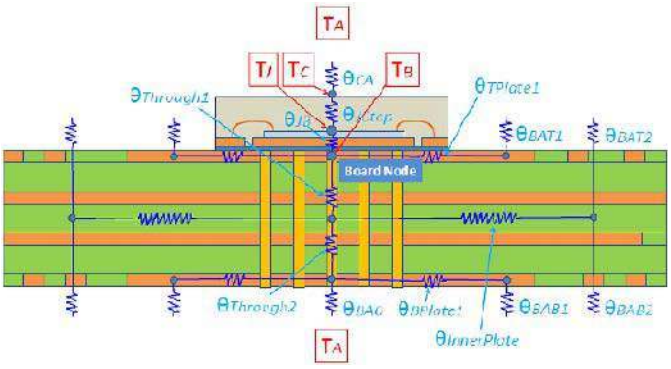
- rezystancji termicznej struktura-obudowa, czyli $R_{th(j-c)}$,
- rezystancji termicznej obudowa-radiator (co w praktyce jest równoważne z rezystancją warstwy materiału termoprzewodzącego), $R_{th(c-s)}$,
- rezystancji termicznej radiatora, $R_{th(s-a)}$.

Klasyczny model komponentu wyposażonego w radiator i warstwę termoprzewodzącą zaprezentowano na **rysunku 13**. Ten sposób liczenia rezystancji termicznych lub temperatur (w zależności od tego, która wielkość jest niewiadomą w danym projekcie) najlepiej sprawdza się w przypadku... komponentów przewlekanych. W technologii SMT do głosu dochodzą już bowiem zupełnie inne drogi przewodzenia ciepła. Opisana powyżej metoda obliczeń termicznych dobrze nadaje się np. do oszacowania temperatury struktury stabilizatora 78xx w obudowie TO-220, przykręconej do radiatora o znanej rezystancji $R_{th(s-a)}$, zaś poboczne drogi „ucieczki" ciepła (np. przez wyprowadzenia komponentu, pady lutownicze i ścieżki na PCB) można w zupełności zaniedbać, gdyż ich udział w obniżaniu temperatury będzie



Fotografia 7. Minikomputer Raspberry Pi z radiatorami zamontowanymi na układach scalonych metodą klejenia (http://t.ly/F1F-S)

eprasa.pl 65e3eb482b



Rysunek 15. Szczegółowy model termiczny układu z rysunku 14, tym razem pokazany z oznaczeniem ważniejszych rezystancji składowych (http://t.ly/_npKR)

Fotografia 8. Radiator kompatybilny z obudową TO-220. Widoczne otwory do przykręcenia komponentu oraz piny do mocowania w otworach PCB metodą lutowania przewlekanego (http://t.ly/R5hgX)

Fotografia 9. Radiatory z serii P marki Ohmite wyposażone w zintegrowany docisk do mocowania komponentu (http://t.ly/JdmMI)

Fotografia 10. Masywny radiator do układów BGA wyposażony w zintegrowane dociski sprężynowe (http://t.ly/6cgbA)

pomijalny. Z diametralnie innymi zagadnieniami mierzymy się natomiast rozpatrując parametry termiczne układów scalonych i komponentów dyskretnych montowanych powierzchniowo. Rolę radiatora pełnią najczęściej przede wszystkim przelotki, płaszczyzny masy i zasilania, a także ścieżki podłączone do indywidualnych padów (**rysunek 14**). Dodatkowe komponenty chłodzące – o ile zostaną w ogóle zaimplementowane – należy rozpatrywać biorąc pod uwagę sposób ich mocowania, np. przez przyklejenie radiatora do górnej powierzchni obudowy układu (**fotografia 7**). Przeprowadzenie obliczeń termicznych z uwzględnieniem wszystkich najważniejszych dróg przepływu ciepła wymaga opracowania możliwie szczegółowego modelu oraz określenia połączeń pomiędzy „wirtualnymi rezystorami" – wszak rezystancję termiczną obowiązują te same metody łączenia, co rezystancję elektryczną (możliwe są połączenia szeregowe, równoległe oraz mieszane – patrz **rysunek 15**). Na szczęście, w doskonale opracowanych materiałach marki onsemi znajdujemy tabelę z wartościami przewodności cieplnej dla wszystkich rozpatrywanych materiałów (**tabela 2**). Rzecz jasna same wartości to nie wszystko – trzeba bowiem jeszcze uwzględnić nietrywialne zależności geometryczne oraz (co chyba najważniejsze w każdym zadaniu związanym z modelowaniem matematycznym) dokonać niezbędnych uproszczeń poprzez pominięcie elementów o znaczeniu marginalnym dla wyniku obliczeń i zidentyfikowanie tych składowych rezystancji termicznej, które w praktyce okażą się najważniejsze.

Konstrukcje i sposoby montażu radiatorów

Istnieje wiele sposobów mocowania radiatorów, przy czym wybór optymalnego rozwiązania zależy przede wszystkim od rodzaju obudowy komponentu oraz wielkości radiatora. Lekkie, niskoprofilowe radiatory można z powodzeniem kleić bezpośrednio do obudów komponentów przy pomocy klejów lub samoprzylepnych podkładek termoprzewodzących. Większe modele są zwykle lutowane do płytki drukowanej przy pomocy wbudowanych pinów i łączone z obudową komponentu poprzez przykręcenie (**fotografia 8**) albo docisk (**fotografia 9**). Radiatory przeznaczone głównie do układów BGA (np. procesorów) są przeważnie mocowane specjalnymi, sprężynowymi dociskami szybkomocującymi (**fotografia 10**).

Niemal wszystkie stosowane w praktyce radiatory mają system użebrowań, którego celem jest zwiększenie powierzchni wymiany ciepła z otaczającym powietrzem. Oprócz konstrukcji z prostymi żebrami (fotografia 10), stosowane są także znacznie bardziej rozbudowane geometrie, których celem wprawdzie może być ułatwienie przepływu powietrza w chłodzeniu metodą konwekcji wymuszonej (niezwykle ciekawy przykład przedstawiono na **fotografii 11**), ale często (zwłaszcza w przypadku komputerów stacjonarnych) wymyślna konstrukcja radiatora stanowi po prostu istotny czynnik decydujący o estetyce całości (**fotografia 12**).

Fotografia 11. Interesujący przykład radiatora, który stanowi jednocześnie... wirnik wentylatora (http://t.ly/g6HZR)

Znaczenie dla efektywności radiatora ma – oprócz geometrii – także wykończenie powierzchni. Tańsze radiatory o naturalnej, metalicznej powierzchni znacznie gorzej oddają bowiem ciepło drogą promieniowania termicznego. Zniwelowanie tego problemu może być zrealizowane na drodze anodowania bądź pokrywania specjalnymi powłokami, nanoszonymi na powierzchnię radiatora metodą określaną mianem *E-Coating* – drugie z wymienionych rozwiązań daje lepszy efekt estetyczny (**fotografia 13**), choć nieznacznie pogarsza skuteczność chłodzenia przy użyciu tak wykończonego radiatora (**rysunek 16**).

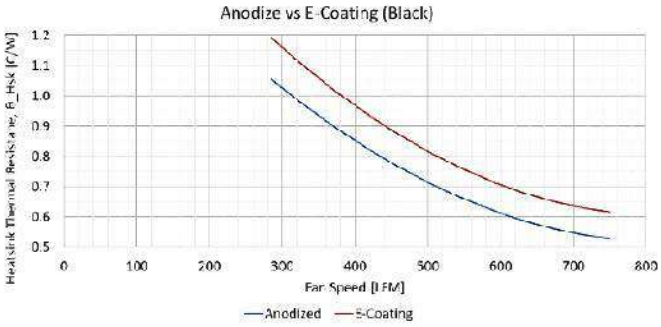
Fotografia 12. „Dizajnerski" model radiatora przeznaczonego do współpracy z dyskami M.2 (http://t.ly/S66LH)

Tabela 2. Przewodność cieplna materiałów uwzględnianych w modelu termicznym układu SMD (na podstawie: http://t.ly/_npKR)

Materiał	Przewodność cieplna [W/m·K]
Krzem	145
Kompozyt obudowy IC	0,7
Ramka konstrukcyjna wewnątrz IC	277
Żywica epoksydowa mocująca strukturę do podłoża	2,4
Miedź	388
Laminat FR4	0,35
Lutowie (stop SnAgCu)	57,3
Lutowie (stop Sn63Pb37)	50

eprasa.pl 65e3eb482b

Fotografia 13. Porównanie powłok radiatorów wykonanych metodą E-Coating (po lewej) oraz przy pomocy zwykłego anodowania (po prawej). Źródło: http://t.ly/Q1OBg

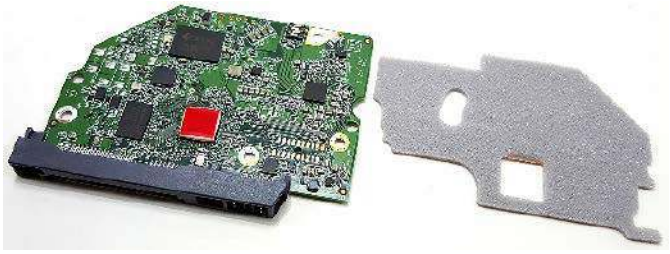


Rysunek 16. Wpływ anodowania oraz nanoszenia powłok typu E-Coating na osiągi termiczne radiatorów aluminiowych (http://t.ly/Q1OBg)

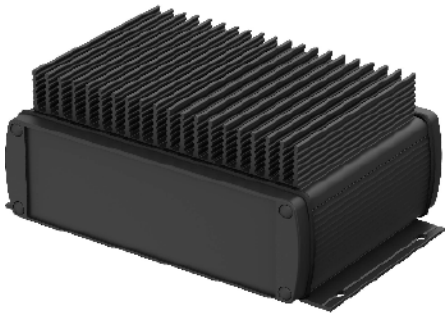
Obudowa jako radiator, czy może radiator w roli obudowy?

Jak już wspomnieliśmy we wcześniejszej części artykułu, obudowa może być niezwykle skutecznym elementem systemu chłodzenia urządzenia elektronicznego. Warto w tym miejscu wskazać trzy główne metody praktycznej realizacji takiej konstrukcji.

- **Zastosowanie obudowy jako radiatora** – metalową obudowę, nawet jeżeli swoją budową nie przypomina ona typowego, bogato użebrowanego radiatora, można z powodzeniem zastosować w roli elementu rozpraszającego ciepło do otoczenia. Dobrym przykładem mogą być obudowy dysków HDD, które – dzięki zastosowaniu przekładek termoprzewodzących – łączone są powierzchnią PCB lub obudową komponentu wymagającego chłodzenia (**fotografia 14**).
- **Obudowa zintegrowana z radiatorem** – w portfolio producentów obudów uniwersalnych do elektroniki można znaleźć aluminiowe obudowy, których jedna z powierzchni jest zaopatrzona w masywny radiator (**fotografia 15**). Takie rozwiązanie zapewnia bardzo efektywny odbiór ciepła ze znajdujących się wewnątrz układów, choć w tym przypadku oczywiście także trzeba zapewnić właściwy kontakt nagrzewających się podzespołów z wewnętrzną powierzchnią ścianki obudowy (zwykle poprzez zastosowanie pasty lub przekładki termoprzewodzącej).
- **Radiator w formie obudowy** – trzecie rozwiązanie może okazać się najefektywniejsze ze wszystkich opisanych do tej pory, ma jednak pewną wadę – obudowę trzeba zaprojektować oraz wyprodukować z myślą o konkretnym modelu urządzenia i nie da się zastosować jej do jakiegokolwiek innego sprzętu. Ekspansja minikomputerów jednopłytkowych upowszechniła to rozwiązanie jako zapewniające doskonałe warunki chłodzenia za sprawą dużej powierzchni oddawania ciepła – przykład można zobaczyć na **fotografii 16**.



Fotografia 14. Kontroler dysku głównego z widocznym padem termicznym (czerwony kwadrat) zamontowanym na obudowie układu scalonego. Widoczny po prawej stronie szary element to pianka ochronna (http://t.ly/rulh_)

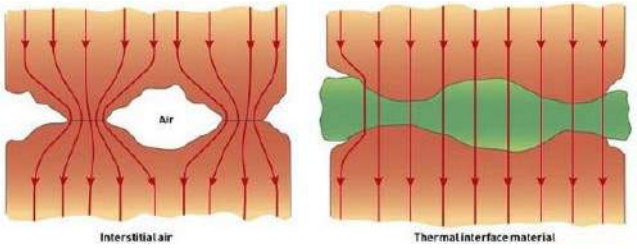


Fotografia 15. Aluminiowa obudowa uniwersalna ze zintegrowanym radiatorem (http://t.ly/EZRUL)

Fotografia 16. Aluminiowy radiator pełniący rolę obudowy minikomputera Raspberry Pi 4 Model A (http://t.ly/AOkh9)

Materiały termoprzewodzące

Do tej pory niejednokrotnie wspomnieliśmy już o konieczności stosowania materiałów termoprzewodzących pomiędzy powierzchnią obudowy komponentu a radiatorem. Konieczność ta wynika wprost z niedoskonałości obydwu obiektów w ujęciu mikroskopowym – pomiędzy rzeczywistymi powierzchniami znajdują się bowiem liczne, drobne wnęki powietrzne. Przekaz ciepła może zatem zachodzić praktycznie tylko w tych miejscach, w których obydwa elementy faktycznie stykają się ze sobą bezpośrednio (powietrze jest doskonałym izolatorem cieplnym, więc zaledwie niewielka część energii może być przekazywana przez wspomniane luki). Lokalne „zagęszczanie" strumienia ciepła (**rysunek 17**) w praktyce manifestuje się jako wzrost rezystancji termicznej całego układu. Zadaniem pasty,



Rysunek 17. Ilustracja wpływu pasty termoprzewodzącej na sposób przepływu strumienia ciepła przez interfejs komponent-radiator (http://t.ly/42Np_)

eprasa.pl 65e3eb482b

Fotografia 17. Zestaw kilkunastu różnych past termoprzewodzących (http://t.ly/xparf)

Fotografia 21. Zastosowanie automatycznego dyspensera do nakładania pasty termoprzewodzącej na wewnętrzną powierzchnię metalowej obudowy (http://t.ly/dJH62)

kleju lub podkładki termoprzewodzącej jest zatem wypełnienie drobnych nierówności obu powierzchni i wyparcie znajdującego się w nich powietrza.

Fotografia 18. Dwuskładnikowy, epoksydowy klej termoprzewodzący (http://t.ly/dTy4p)

W produkcji materiałów termoprzewodzących istotne znaczenie mają bazy silikonowe, stosowane zarówno do wytwarzania past (**fotografia 17**), jak i klejów (**fotografia 18**) oraz przekładek samoprzylepnych (**fotografia 19**). W celu zwiększenia wydajności chłodzenia materiał bazowy jest często uzupełniany domieszkami obniżającymi wynikową rezystancję termiczną – wiele past termoprzewodzących ma w swoim składzie proszki metaliczne (srebro, miedź, złoto lub aluminium), bądź wypełniacze ceramiczne.

Fotografia 19. Zaletą przekładek termoprzewodzących jest łatwy montaż i możliwość dostosowania kształtu do dowolnej geometrii współpracujących komponentów (http://t.ly/Opt9A)

Fotografia 20. Sześć różnych sposobów dozowania pasty termoprzewodzącej na powierzchni obudowy procesora komputerowego (http://t.ly/dQeog)

Sposób nakładania pasty termoprzewodzącej na powierzchnię komponentu (np. procesora komputerowego) oraz jego wpływ na wynikową efektywność chłodzenia jest już od wielu lat tematem internetowych dyskusji. W praktyce sam kształt ścieżki dozowania (**fotografia 20**) ma jednak nieporównanie mniejsze znaczenie niż szereg innych aspektów – skład pasty, termin przydatności do użycia oraz ilość naniesionego na element materiału znacznie silniej wpływają na osiągi systemu chłodzenia. Warto w tym miejscu dodać, że w produkcji seryjnej zastosowanie znajdują automatyczne dyspensery materiałów termoprzewodzących (past i klejów) – przykład można zobaczyć na **fotografii 21**.

Otwory wentylacyjne i wentylatory

Otwory wentylacyjne obudowy stanowią jeden z najistotniejszych elementów wspierających chłodzenie metodami konwekcji swobodnej oraz wymuszonej. Wybór liczby, kształtu oraz rozmieszczenia otworów zależy od szeregu czynników.

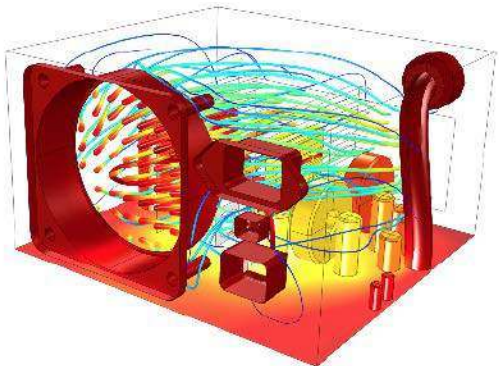
- **Wymagana skuteczność chłodzenia** – w urządzeniach wytracających niewielką moc wystarczające okazuje się często wykonanie zaledwie kilku otworów lub slotów wentylacyjnych (**fotografia 22**), podczas gdy zasilacze średniej i dużej mocy są nierzadko produkowane w obudowach z blachy perforowanej, w której – najprościej rzecz ujmując – jest „więcej otworów niż materiału" (**fotografia 23**).
- **Stopień ochrony IP** – kształt i rozmiar otworów wentylacyjnych mogą pomóc w spełnieniu przez urządzenie wymogów przyjętego przez producenta stopnia ochrony IP. Przykładowo – urządzenie klasy IP3x ma uniemożliwiać dostęp do wnętrza obudowy za pomocą pręta lub

Fotografia 22. Obudowa uniwersalna z trzema otworami wentylacyjnymi (http://t.ly/rYkme)

Fotografia 23. Zasilacz modułowy w obudowie z blachy perforowanej (http://t.ly/DuktJ)

Fotografia 24. Zewnętrzna szafa telekomunikacyjna z widocznymi otworami wentylacyjnymi daszkowymi, wykonanymi jako przetłoczenia blachy (http://t.ly/jlduR)

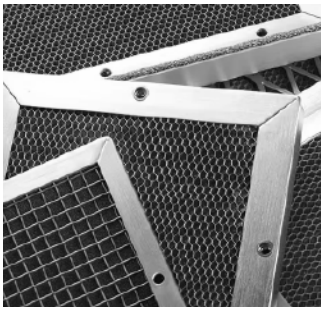
eprasa.pl 65e3eb482b



Rysunek 18. Symulacja numeryczna prądów powietrza w obudowie zasilacza komputerowego (http://t.ly/jcbl8)

innego ciała stałego o średnicy >=2,5 mm, zaś w przypadku klasy IP4x wartość ta maleje do 1 mm. Producent urządzenia musi zatem zapewnić taki kształt i rozmiar otworów, by wprowadzenie przez nie odpowiedniego narzędzia testowego było niemożliwe. Mało tego – tzw. otwory daszkowe, produkowane poprzez przetłoczenia w arkuszu blachy, pozwalają nawet uzyskać pewien stopień ochrony przed opadami atmosferycznymi lub wodą innego pochodzenia (IPx1...IPx3), co jest chętnie wykorzystywane przez projektantów urządzeń pracujących w otwartej przestrzeni (**fotografia 24**).

- **Rozmieszczenie komponentów w obudowie urządzenia** – prawidłowe podejście do projektowania sprzętu chłodzonego konwekcją (w szczególności wymuszoną) wymaga skrupulatnego przemyślenia rozmieszczenia otworów wentylacyjnych względem położenia wentylatora oraz wszystkich pozostałych bloków urządzenia. Przepływ powietrza nie może bowiem powodować nagrzewania czułych obwodów (np. analogowych) gorącym strumieniem gazu, ogrzanego przez radiatory współpracujące z zasilaczem czy też końcówką mocy. Bezcenne jest w tym przypadku doświadczenie wsparte pewną dozą inżynierskiej intuicji – w bardziej złożonych przypadkach trzeba jednak posiłkować się zaawansowanym oprogramowaniem do symulacji ruchu powietrza (**rysunek 18**)
- **Ochrona otworów wentylacyjnych przed RFI** – każdy otwór w metalowej obudowie urządzenia (odgrywającej rolę ekranu) stanowi „wyłom" w ochronie przed emisją i odbiorem zakłóceń elektromagnetycznych. Dlatego też w urządzeniach, w których skuteczne niwelowanie zakłóceń radiowych ma znaczenie pierwszorzędne, powszechnie stosowane są rozmaite osłony w postaci siatek ortogonalnych bądź „plastrów miodu" (**fotografia 25**) – za cenę pewnego oporu stawianego przepływowi powietrza, elementy te są w stanie skutecznie „uszczelnić" system ekranowania.
- **Aspekty estetyczno-artystyczne** – umiejętne opracowanie rozmieszczenia i kształtu otworów wentylacyjnych pozwala zastosować ten element projektu obudowy w roli ozdobnej. Niektórzy producenci traktują obecność otworów jako okazję do uwidocznienia logotypu bądź innych elementów identyfikacji wizualnej marki (**fotografia 26**).



Fotografia 25. Osłony RFI przeznaczone do wentylatorów i otworów wentylacyjnych (http://t.ly/NqmPT)

Fotografia 26. Przykład zastosowania otworów wentylacyjnych jako elementu identyfikacji wizualnej marki producenta (http://t.ly/_Zww0)

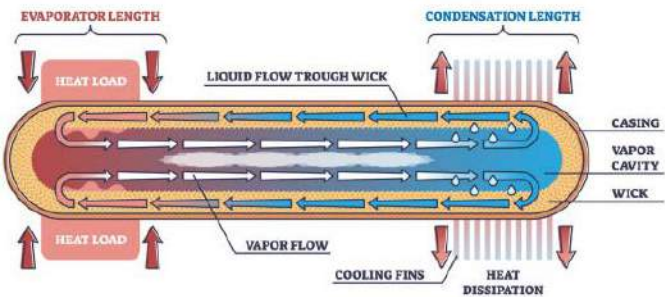
Fotografia 27. Zastosowanie chłodzenia cieczą w komputerze o dużej mocy obliczeniowej (http://t.ly/FcyMT)

Systemy chłodzenia cieczą

Oprócz rozwiązań wymuszających przepływ powietrza we wnętrzu obudowy urządzenia, do dyspozycji projektantów są także znacznie droższe, ale za to nieporównanie efektywniejsze systemy chłodzenia cieczą. W powszechnej świadomości społeczeństwa XXI wieku technologia ta łączy się głównie z designerskimi komputerami dla graczy – nie należy jednak zapominać, że jako wydajna technika odbioru ciepła jest stosowana także w dużo poważniejszych zastosowaniach, np. potężnych komputerach pracujących w centrach danych (**fotografia 27**). Niewątpliwą zaletą tej technologii – oprócz doskonałych parametrów termicznych – jest także możliwość odsunięcia większości elementów systemu chłodzenia od właściwego źródła ciepła: miedziane przewody transportujące chłodziwo można bowiem prowadzić niemal dowolnie, co niebywale ułatwia pracę projektantom mechaniki urządzeń.

Chłodzenie wspomagane przemianą fazową

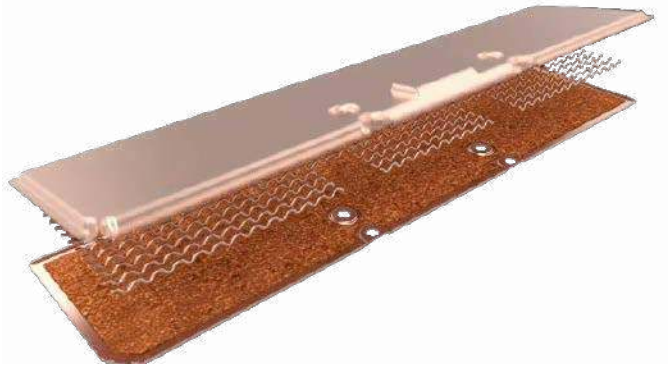
Klasycznego chłodzenia cieczą nie należy mylić z (podobnymi wizualnie, ale działającymi na diametralnie innej zasadzie) technikami opartymi na zjawiskach przemiany fazowej. Podstawy fizyczne tej grupy metod opisaliśmy już w telegraficznym skrócie na początku artykułu, teraz skupimy się zatem na realizacjach technicznych. Dwie najczęściej stosowane topologie to tzw. rurki ciepła (ciepłowody) oraz komory parowe. Te pierwsze składają się, jak sama nazwa wskazuje, z cienkościennych, metalowych (zwykle miedzianych) rurek, wewnątrz których znajduje się niewielka ilość cieczy. Wewnętrzne powierzchnie rurek są natomiast wyłożone plecionką z cienkich drutów (**rysunek 19**, **fotografia 28**). Ogrzewanie jednego końca rurki (połączonego z radiatorem zamontowanym na obudowie chłodzonego komponentu) powoduje odparowywanie cieczy, która – przemieszczając się w stronę



Rysunek 19. Zasada działania rurki ciepła (http://t.ly/FZUnJ)

eprasa.pl 65e3eb482b

Fotografia 28. Przekroje poprzeczne rurek ciepła wykonanych w dwóch różnych technologiach (http://t.ly/g-3Da)

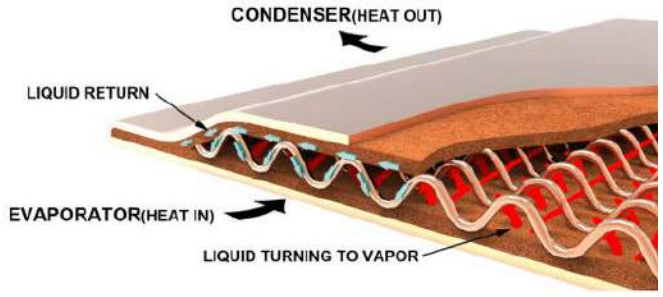


Rysunek 21. Ogólna budowa dwuelementowej komory parowej (http://t.ly/vObJ6)

przeciwległego, „zimnego" końca rurki – ulega schłodzeniu i skropleniu. Tam jest „przechwytywana" przez plecionkę i samoczynnie doprowadzana z powrotem do obszaru odbioru ciepła. Ten „termodynamiczny samograj" znalazł szerokie zastosowania przede wszystkim w budowie komputerów przenośnych, w których umożliwia elastyczne projektowanie ułożenia elementów komputera tak, by ciepło odebrane z procesora odprowadzić w inne miejsce obudowy i tam rozproszyć je za pomocą przewodzenia (do pokrywy) bądź za pośrednictwem konwekcji wymuszonej (przy pomocy niskoprofilowego wentylatora promieniowego – **fotografia 29**).

Na nieco innych założeniach konstrukcyjnych opierają się tzw. komory parowe (**rysunek 20**). Choć w tym przypadku także u podstaw działania leży zastosowanie przemian fazowych cieczy w obiegu zamkniętym, to budowa komór parowych różni się od rurek ciepła przede wszystkim geometrią i wymiarami. Komory są płaskimi pojemnikami o stosunkowo dużej powierzchni i kontaktują się z chłodzonymi elementami bezpośrednio (tj. bez zastosowania metalowej płytki bazowej w roli interfejsu, jak ma to miejsce w przypadku systemów opartych na ciepłowodach). I właśnie ta różnica sprawia, że skuteczność komór parowych jest wyraźnie wyższa w porównaniu do rurek ciepła – ściana komory wprowadza do układu znacznie mniejszą rezystancję termiczną, niż metalowy blok połączony z ciepłowodami i mocowany na procesorze lub innym elemencie elektronicznym. Obecnie w użyciu są dwie technologie produkcji komór parowych: tradycyjne, dwuelementowe (czyli złożone z dwóch, odpowiednio uformowanych i połączonych szczelnie na obwodzie arkuszy blachy – **rysunek 21**) oraz jednoelementowe (hybrydowe). Te drugie są wytwarzane z silnie spłaszczonej, cienkościennej rury, wewnątrz której umieszcza się odpowiednie elementy nośne oraz warstwę plecionki. Zaletą komór jednoelementowych jest dużo niższy koszt produkcji, wynikający z prostszego procesu technologicznego, wadą zaś – mniejsza elastyczność pod względem wytwarzania nietypowych kształtów.

Fotografia 29. Wnętrze laptopa z widocznym systemem rurek ciepła oraz dwoma wentylatorami promieniowymi (http://t.ly/XSYea)



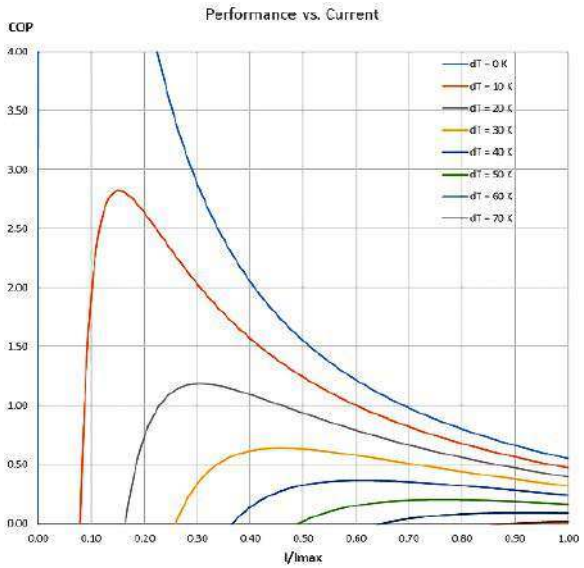
Rysunek 20. Przekrój komory parowej z zaznaczonymi kierunkami transportu cieczy (http://t.ly/QICoP)

Chłodzenie ogniwami Peltiera

Termoelektryczne systemy chłodzenia pozwalają na uzyskanie istotnego obniżenia temperatury za cenę znacznego zwiększenia ilości ciepła, które trzeba rozproszyć do otoczenia. O ile bowiem pozostałe opisane w niniejszym artykule rozwiązania oddają do atmosfery tylko ciepło wygenerowane przez chłodzony element, to w przypadku ogniw Peltiera trzeba jeszcze wytracić dodatkową energię, co wymusza niejako… chłodzenie systemu chłodzenia (**fotografia 30**). Co gorsza, rzeczywista wydajność znacząco maleje wraz ze wzrostem natężenia prądu zasilania oraz różnicy temperatur pomiędzy stroną gorącą a stroną zimną. Podstawowym parametrem rozważanym w systemach chłodzenia termoelektrycznego jest współczynnik wydajności (COP), będący ilorazem ciepła zaabsorbowanego po stronie zimnej (czyli odebranego z chłodzonego elementu) oraz mocy dostarczonej do modułu Peltiera z systemu zasilania. Stosowny wykres można zobaczyć na **rysunku 22**.

Fotografia 30. Termoelektryczny moduł chłodzący (http://t.ly/2jnyL)

eprasa.pl 65e3eb482b

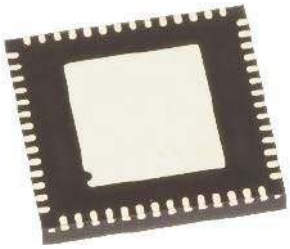


Rysunek 22. Wykres współczynnika COP dla ośmiu różnych wartości różnicy temperatury pomiędzy stroną gorącą, a stroną zimną modułu Peltiera (http://t.ly/1mkua)

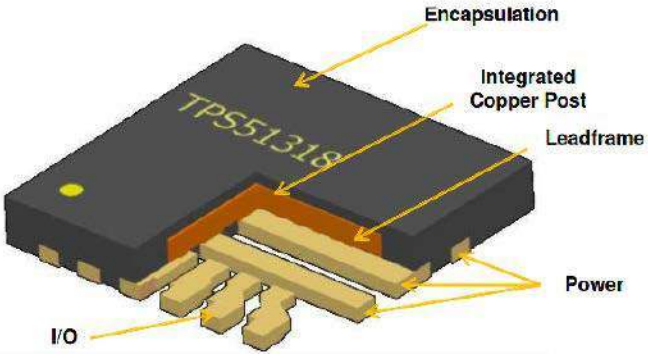
Chłodzenie na poziomie komponentów i PCB

Do tej pory omawialiśmy zagadnienia chłodzenia urządzeń na poziomie systemowym oraz lokalnym. Warto przyjrzeć się kilku rozwiązaniom, które implementowane są przez producentów półprzewodników w samych komponentach – techniki te są o tyle ważne, że interfejs pomiędzy strukturą krzemową a zewnętrznymi radiatorami bądź padem termicznym na PCB, wpływa bezpośrednio na jakość drogi odprowadzania ciepła i nierzadko stanowi wąskie gardło całej tej ścieżki.

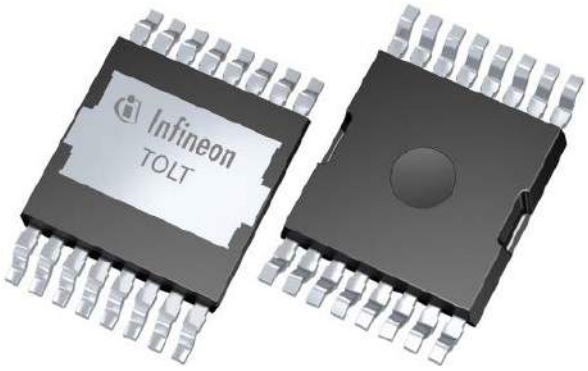
Jednym z najpopularniejszych rozwiązań, z którym spotkał się w swojej praktyce każdy elektronik, jest zastosowanie padu termicznego pod obudową układu scalonego (np. QFN – **fotografia 31**). Pad współpracuje z odpowiednim polem lutowniczym na płytce drukowanej – rozwiązanie takie zyskuje jednak realny sens tylko wtedy, gdy w produkcji przewidziano wytworzenie przelotek termicznych. Zatykanie otworów specjalną pastą na bazie proszku miedzianego przynosi fenomenalne efekty zarówno z punku widzenia jakości lutów, jak i efektywności transferu ciepła – niestety jest to technika dość kosztowna. Z tego względu w praktyce stosowane jest często obejście problemu poprzez zastosowanie grupy zwykłych przelotek o bardzo małej średnicy (zwykle poniżej 0,3 mm) – choć ryzyko penetrowania ich przez pastę lutowniczą nadal istnieje, to dzięki pomniejszeniu otworów jest ono znacznie mniejsze.



Fotografia 31. Przykładowa obudowa QFN z padem termicznym (http://t.ly/eOPPL)



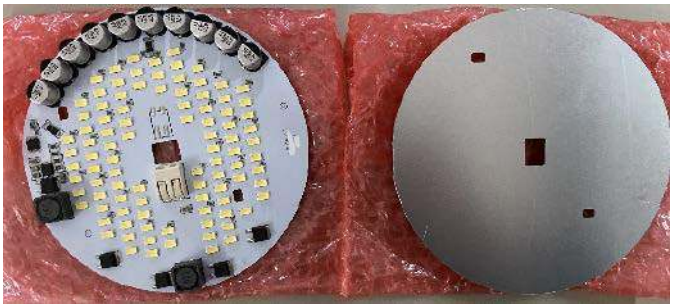
Rysunek 23. Budowa układu QFN w technologii HotRod (http://t.ly/Jk0f7)



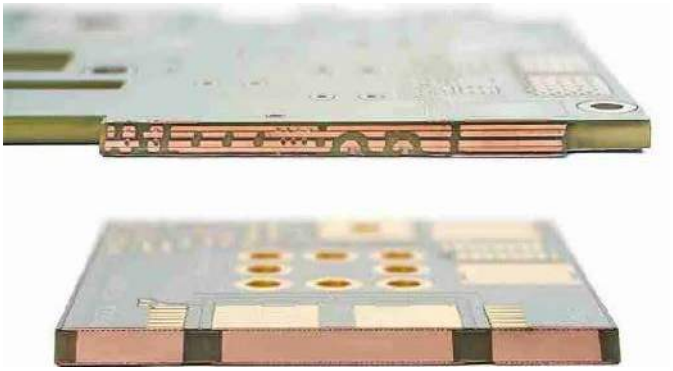
Rysunek 24. Obudowa układu scalonego typu TOLT z radiatorem wyprowadzonym na stronę górną (http://t.ly/ytZwI)

Modyfikacją standardowej konstrukcji układów QFN, bazującej na typowej ramce (*lead frame*), jest technologia HotRod. Polega ona na bezpośrednim wyprowadzeniu metalowych elementów ramki na brzegi obudowy układu w taki sposób, że stają się one zarazem pinami lutowniczymi, połączonymi z masą i szynami zasilania układu (**rysunek 23**). Takie rozwiązanie ma dwie główne zalety: po pierwsze znacząco redukuje rezystancję połączeń (dzięki eliminacji cienkich drucików stosowanych do bondingu), co istotnie podnosi obciążalność prądową. Po drugie – możliwość przylutowania elementów ramki do padów na PCB znakomicie obniża wynikową rezystancję termiczną.

Kolejnym ulepszeniem, stosowanym w przypadku obudów typu SOP, jest przeniesienie zintegrowanego radiatora na… górną stronę obudowy układu scalonego (**rysunek 24**). Tak powstała konstrukcja, określana przez markę Infineon mianem TOLT, pozwala na łatwe i efektywne dołączenie zewnętrznego radiatora, eliminuje bowiem palący problem standardowych obudów układów scalonych. Problem ten wynika z faktu, iż „pokrywa" obudowy (wykonana z kompozytu) stanowi poważną przeszkodę dla przepływu ciepła ze struktury – nic więc dziwnego, że w literaturze można spotkać się ze stwierdzeniem, iż większość ciepła traconego przez układy SMD jest odprowadzana przez… wyprowadzenia lutownicze (rzecz jasna tylko wtedy,



Fotografia 32. Przykładowy moduł oświetlenia LED wykonany na bazie PCB z podłożem aluminiowym (http://t.ly/RhqNR)



Fotografia 33. Przekroje obwodów drukowanych wykonanych w technologii *heavy copper* (http://t.ly/aO4jE)

eprasa.pl 65e3eb482b

Fotografia 34. PCB z wkładką miedzianą (*copper coin*) umieszczoną we wnęce PCB (*cavity*) (http://t.ly/0Z9xW)

gdy obudowa nie została wyposażona w centralny pad termiczny, który z natury rzeczy stanowi preferencyjną drogę chłodzenia).

Na koniec warto przypomnieć, że także konstrukcja płytki drukowanej może istotnie pomóc w zadaniach związanych z chłodzeniem. Niezwykle efektywne i zarazem stosunkowo tanie w produkcji są płytki na podłożach aluminiowych (**fotografia 32**), stąd ich niezmierna popularność w systemach oświetlenia LED.

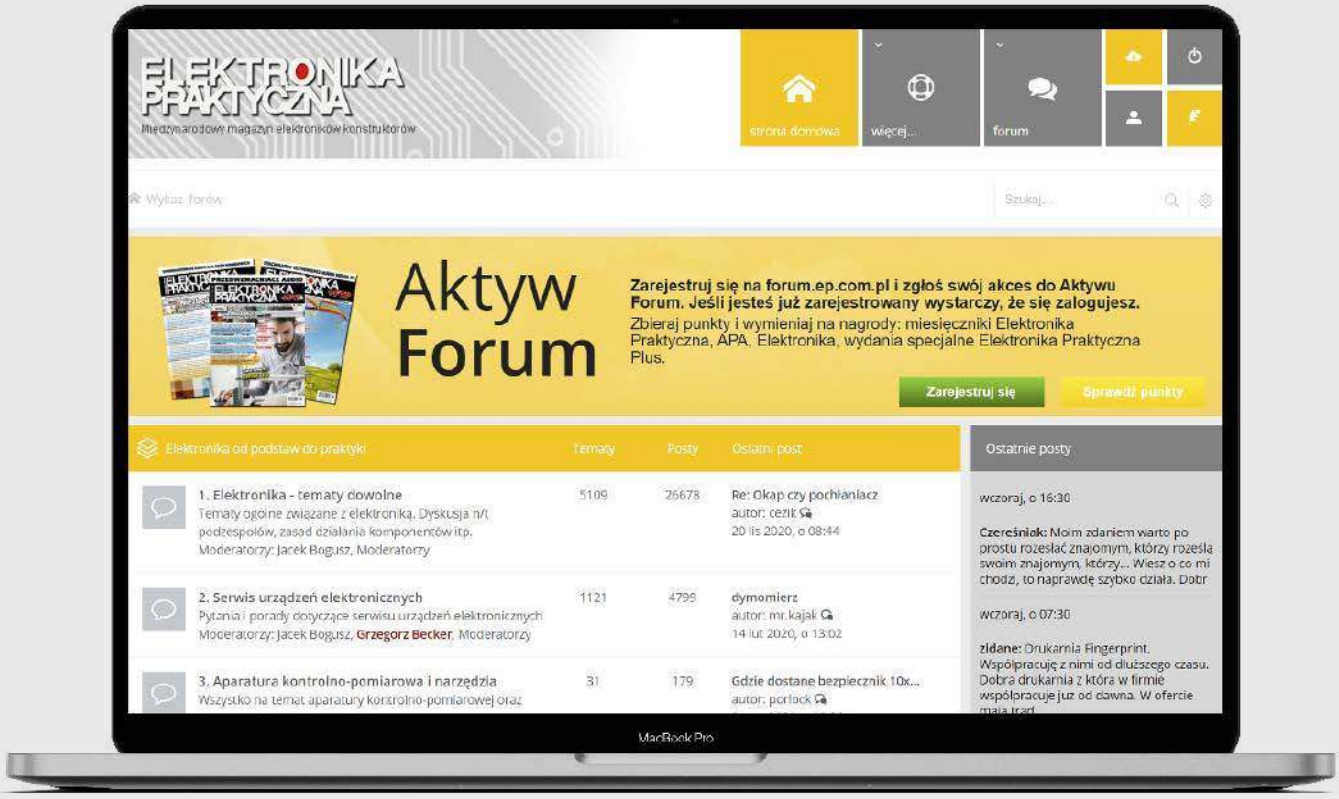
Najbardziej zaawansowani wytwórcy PCB potrafią ponadto wykonywać znacznie trudniejsze w produkcji, ale niezwykle skuteczne termicznie obwody typu *heavy copper* (tj. z ponadprzeciętnie grubymi warstwami miedzi – **fotografia 33**) oraz obwody z wbudowanymi radiatorami miedzianymi, tzw. *copper coin* (**fotografia 34**).

Podsumowanie

W artykule zaprezentowaliśmy szerokie spektrum rozwiązań stosowanych do chłodzenia urządzeń elektronicznych. Jak widać, technologie odprowadzania ciepła dalece wykraczają poza metody w pierwszej chwili kojarzące się z chłodzeniem elektroniki, a oparte na radiatorach i wentylatorach. Umyślnie pominęliśmy natomiast szereg metod chłodzenia spotykanych niemal wyłącznie w literaturze lub nielicznych doniesieniach prasowych – przykładowo: ultra-nowoczesne chłodnice wbudowane w strukturę PCB, systemy z bezpośrednim natryskiem chłodziwa, czy też opatentowane kilka lat temu miniaturowe komory parowe (także wbudowane we wnętrze płytki drukowanej) prawdopodobnie jeszcze przez wiele lat nie wejdą do powszechnego użycia, stanowiąc jedynie „ciekawostkę przyrodniczą". Na rynku istnieje jednak szereg rozwiązań spotykanych dość rzadko, ale jak najbardziej praktycznych – jako przykład możemy wskazać oscylacyjne rurki ciepła, które za sprawą uniezależnienia od grawitacji doskonale sprawdzają się w zaawansowanej aparaturze kosmicznej. Zainteresowanych Czytelników zachęcamy do samodzielnych poszukiwań dalszych informacji na temat rozmaitych, „egzotycznych" systemów chłodzenia urządzeń elektronicznych.

inż. Przemysław Musz, EP

REKLAMA



O projektach, miniprojektach, projektach soft i na wiele innych tematów dyskutuj na https://forum.ep.com.pl

eprasa.pl 65e3eb482b

Chłodzenie w urządzeniach elektronicznych

Miniaturyzacja urządzeń elektronicznych (oraz wynikający z niej wzrost gęstości mocy w elementach półprzewodnikowych) wymusza dziś stosowanie optymalnych materiałów termoprzewodzących.

Projektując odpowiednio wydajne i odporne na czynniki środowiskowe układy chłodzące, należy wziąć pod uwagę szereg parametrów, do których należą:

- specjalne wymagania bezpieczeństwa, m.in. izolacja i rezystancja,
- odporność mechaniczna,
- średnia i maksymalna moc wydzielana przez elementy,
- temperatura otoczenia, maksymalna bezpieczna temperatura chłodzonych elementów,
- sprawność zewnętrznego układu chłodzenia,
- gładkość powierzchni chłodzącej, zakładany sposób montażu,
- rodzaj materiału przewodzącego ciepło.

Systemy chłodzenia można podzielić na dwie główne kategorie: aktywne oraz pasywne.

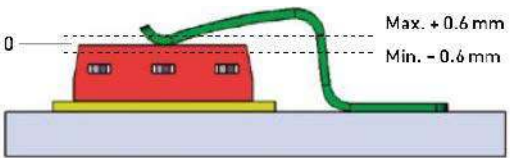
Układ pasywny może być prosty, tani (bez dodatkowych wentylatorów itp.), niezawodny (elementy odporne na zużycie), a przy tym stosunkowo łatwy do analizy. Należy jednak pamiętać, że w celu uzyskania stabilnego, niezakłóconego transportu ciepła należy spełnić pewne warunki.

Chłodzenie pasywne wykorzystuje następujące zjawiska:

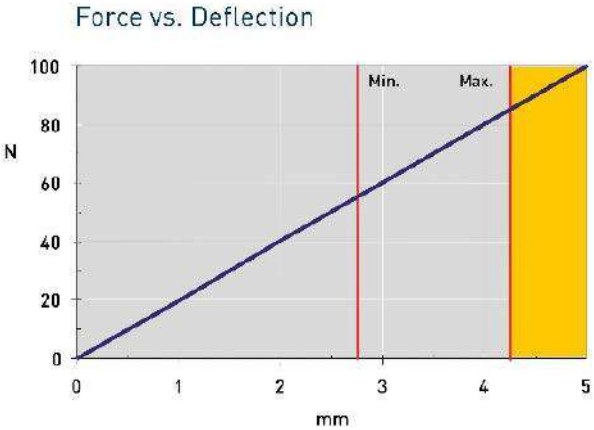
- **Promieniowanie termiczne**, czyli emisja fal w paśmie podczerwieni. Przykładem może być żarówka wydzielająca ciepło do otoczenia. W większości przypadków promieniowanie rozprasza niewielką część wytworzonego podczas pracy ciepła.
- **Przewodzenie**, gdy ciepło przepływa przez materiał stały, zwykle o dużej przewodności cieplnej (np. miedź czy aluminium),
- **Konwekcja**, podczas której ciepło jest odprowadzane przez powietrze lub (rzadziej) ciecz, np. wodę.

Większość projektów opiera się na kombinacji chłodzenia na drodze przewodzenia i konwekcji oraz – w mniejszym stopniu – promieniowania. Przewodzenie pomaga odprowadzić ciepło z komponentu, natomiast konwekcja umożliwia rozproszenie ciepła przetransportowanego ze źródła do radiatora. Powierzchnia styku radiatora musi być płaska i gładka, jeśli ma zapewnić optymalny kontakt termiczny z chłodzonym obiektem. Aby zminimalizować problem niedoskonałości powierzchni stosuje się materiały poprawiające kontakt termiczny. Firma BLelektronik oferuje szeroki asortyment takich produktów, w tym:

- miękkie podkładki termoprzewodzące gap filler (z silikonem lub bez),
- dwuskładnikowe dozowane gap fillery,
- folie i kapturki silikonowe,
- folie grafitowe,
- materiały typu Phase Change,



Rysunek 1. Przekrój poprzeczny układu chłodzenia elementu w obudowie TO-220, złożonego z radiatora, podkładki termoprzewodzącej oraz klipsa dociskowego



Rysunek 2. Zależność siły docisku od odchylenia klipsa

- kleje i taśmy termoprzewodzące,
- zalewy termoprzewodzące (silikonowe, poliuretanowe, epoksydowe).

Z uwagi na niestabilność i problem z zapewnieniem powtarzalnych parametrów termicznych przy zastosowaniu pasty termoprzewodzącej, coraz powszechniej używa się folii, gap fillerów oraz materiałów Phase Change. Do wzrostu ich popularności przyczyniają się koszt i łatwość montażu bez użycia dodatkowych narzędzi. W przypadku dużych serii produkcyjnych stosowanie materiałów dozowanych maszynowo lub drukowanych może być bardziej opłacalne i powtarzalne.

Właściwe zaprojektowanie układu chłodzącego wymaga określenia mocy źródła ciepła i maksymalnych temperatur chłodzonego elementu oraz otoczenia. Na tej podstawie można określić wartości parametrów termicznych i mechanicznych radiatorów oraz przekładek termoprzewodzących.

Podstawowe parametry charakteryzujące materiały termoprzewodzące to: przewodnictwo cieplne, impedancja termiczna, odporność na przebicie izolacji oraz przenikalność elektryczna.

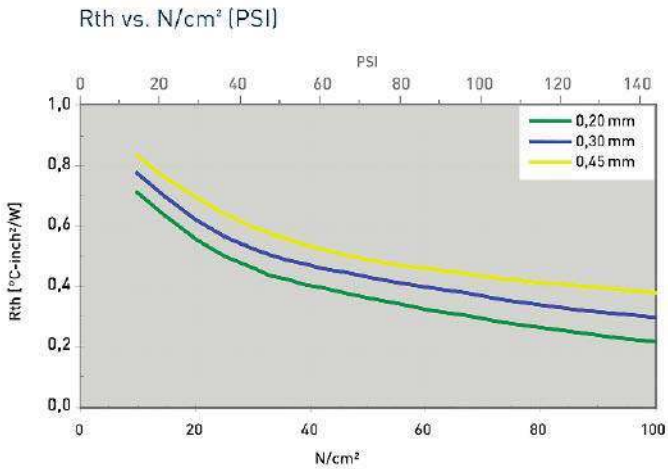
Przewodność cieplna [$W \cdot m^{-1} \cdot K^{-1}$] – parametr zdefiniowany dla prostopadłościennego ciała przewodzącego ciepło w warunkach stanu ustalonego – jest to ilość przekazanej energii w czasie, proporcjonalna do powierzchni przekroju poprzecznego przegrody oraz różnicy temperatur, a odwrotnie proporcjonalna do grubości przegrody.

$$\lambda = \frac{Q}{t} \cdot \frac{d}{S\Delta T} [W \cdot m^{-1} \cdot Km^{-1}]$$

gdzie:
Q – ilość ciepła przepływającego przez ciało,
t – czas przepływu,
S – pole przekroju, przez który przepływa ciepło,
ΔT – różnica temperatur w kierunku przewodzenia ciepła,
d – grubość przegrody,
[$W \cdot m^{-1} \cdot K^{-1}$] – współczynnik przewodnictwa cieplnego.

Fotografia 1. Przykładowy klips dociskowy

eprasa.pl 65e3eb482b



Rysunek 3. Zależność rezystancji termicznej folii termoprzewodzącej od nacisku dla folii TFO-O300-SI firmy HALA

Impedancja/rezystancja termiczna [°C/W] – iloraz różnicy temperatur pomiędzy elementem wydzielającym ciepło, a elementem chłodzącym oraz wartości mocy wywołującej tę różnicę. Wartość impedancji (często podawana dla określonej powierzchni przekładki) zależy od siły docisku oraz wielkości i jakości łączonych powierzchni.

W przypadku elementu o znanej powierzchni oraz określonych stratach mocy rezystancję termiczną podkładki termoprzewodzącej można określić na bazie wykresów z katalogu producenta. Najprostszym sposobem zapewnienia stałej siły docisku jest zastosowanie sprężystego elementu o dokładnie zdefiniowanych parametrach mechanicznych w funkcji wielkości odkształcenia, np. Hala Clip TO 220-1.

Wymaganą rezystancję termiczną R_{th} materiału termoprzewodzącego można obliczyć ze wzoru:

$$R_{th}=(T_j-T_{hsink})/P-R_{thj\text{-}c}[°C/W]$$

T_j – dopuszczalna temperatura złącza półprzewodnikowego,
P – moc strat w elemencie,
T_{hsink} – temperatura zewnętrznego radiatora/powierzchni chłodzącej,
$R_{thj\text{-}c}$ – rezystancja termiczna od złącza półprzewodnikowego do obudowy elementu.

Należy pamiętać o tym, że kluczowy parametr to nie temperatura obudowy elementu T_c, ale wewnętrzna temperatura złącza T_j.

Przykładowo: dla tranzystora MOSFET w obudowie TO-220 o mocy strat 25 W, przymocowanego do radiatora o rezystancji termicznej R_{thsink}=1°C/W i temperaturze 50°C, możemy obliczyć maksymalną rezystancję termiczną podkładki termoprzewodzącej:

$$R_{th}=<(150°C-50°C)/25\ W-1°C/W=3°C/W$$

dla $R_{thj\text{-}c}$=1°C/W i założonej maksymalnej temperatury złącza T_j=150°C

Zakładając, że tranzystor mocowany jest sprężyną Hala Clip TO 220-1 o sile docisku 55...85 N (co odpowiada naciskowi 35...55 N/cm² lub 50...80 PSI dla powierzchni TO-220 równej 160 mm²), właściwym materiałem może być TFO-O300-SI o rezystancji R_{th}(cal²)=<0,6°C/W(cal²) dla siły nacisku >30 PSI.

Wartość rezystancji termicznej podkładki jest zazwyczaj podana w katalogu dla powierzchni 1 cal² (645 mm²), stąd obliczoną wartość 3°C/W należy pomnożyć przez 0,25 (160 mm²/645 mm²).

W obliczeniach należy uwzględnić zarówno moc średnią, jak i szczytową. Wszystkie analizy powinny brać pod uwagę pojemność cieplną, a pomiary temperatury należy wykonywać po termicznej stabilizacji systemu (co może trwać nawet kilka godzin).

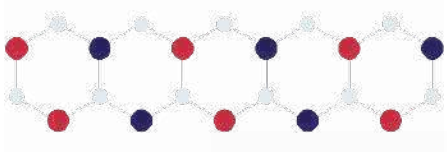
Przedstawiony przykład opisuje najprostszy przypadek użycia materiałów termoprzewodzących. W odniesieniu do bardziej złożonych systemów do analizy termicznej można użyć oprogramowania symulacyjnego, korzystającego z metody elementów skończonych MES.

BLelektronik
www.blelektronik.com.pl
tel. 12 357 63 78

REKLAMA

BL elektronik

Optymalne rozwiązania dla elektroniki i energetyki



Klejenie

Lakierowanie

Hermetyzacja

Odprowadzanie ciepła

CHT SMART CHEMISTRY WITH CHARACTER. Robnor ResinLab OTTO CHEMIE Sealants • Adhesives NovaChem Kisling HALA

FHU BL elektronik

ul. J. Conrada 65, 31-357 Kraków
tel. +48 12 357 63 78, +48 696 483 020
kontakt@blelektronik.com.pl
sklep.blelektronik.com.pl e-sklep

eprasa.pl 65e3eb482b

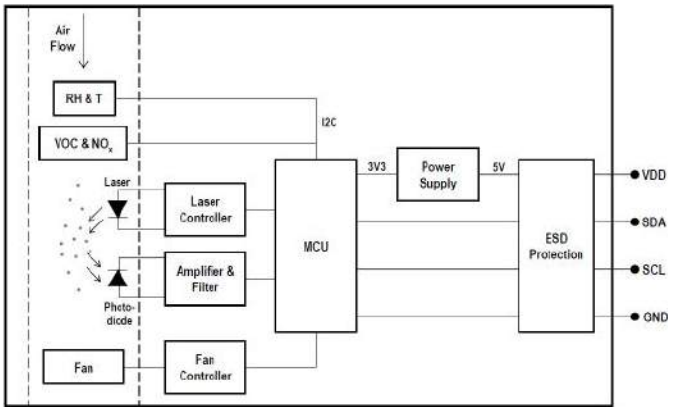
Internet Rzeczy w pomiarach środowiskowych (3)

Moduł czujników środowiska SEN5x firmy Sensirion

Moduł czujników środowiska SEN5x firmy Sensirion to platforma sensoryczna przeznaczona do dokładnego pomiaru różnych parametrów środowiskowych, takich jak cząstki stałe, lotne związki organiczne (LZO), tlenki azotu (NO_x), a także wilgotność i temperatura powietrza. Dzięki autorskim algorytmom moduł umożliwia łatwą integrację z różnymi aplikacjami. Dane pomiarowe są w pełni skalibrowane.

Warunki środowiskowe mają ogromny wpływ na nasze samopoczucie, komfort i produktywność. Rozwiązania czujnikowe firmy Sensirion dostarczają szczegółowych i wiarygodnych danych na temat kluczowych parametrów środowiskowych, takich jak wilgotność, temperatura, lotne związki organiczne (VOC, Volatile Organic Compounds), cząstki stałe (PM2,5), formaldehyd, NO_x i CO_2. Monitorowanie środowiska otwiera nowe możliwości tworzenia inteligentniejszych urządzeń, które poprawiają nasz komfort i samopoczucie, a także zwiększają efektywność energetyczną w szerokiej gamie zastosowań.

Rodzina modułów SEN5x składa się z czujników identycznych pod względem wymiarowym, układowym i interfejsowym, ale różniących się liczbą informacji wyjściowych. SEN50 wykonuje tylko pomiar cząstek stałych. SEN54 realizuje pomiary stężenia cząstek stałych, VOC, wilgotności i temperatury. SEN55 mierzy dodatkowo stężenie NO_x. Konstrukcja modułów jest wynikiem rozległego doświadczenia firmy Sensirion w zakresie badania środowiska i oferuje najlepszą możliwą wydajność pomiarową, wyjątkową trwałość, a także kompaktową obudowę. Czujniki SEN5x dostarczają w pełni skalibrowane dane za pomocą interfejsu cyfrowego. Zastrzeżone algorytmy umożliwiają prostą integrację z aplikacjami HVAC i urządzeniami do pomiaru jakości powietrza.



Rysunek 1. Schemat blokowy modułu czujników środowiska SEN55 [2]

Poprzednie części artykułu znajdują się pod adresem: **https://ulubionykiosk.pl/media**

Schemat blokowy modułu czujników środowiska SEN55 został pokazany na **rysunku 1** [2]. W module zintegrowane zostały rozwiązania z czujnika pyłów typu SPS30, opisanego w artykule „Czujnik pomiaru pyłów SPS30 firmy Sensirion" [9]. Szersze omówienie czujników pyłów Czytelnicy znajdą natomiast w artykule „Cyfrowe czujniki pyłów" [7]. Drugie zastosowane w module rozwiązanie pomiarowe to czujnik VOC typu SGP41. Szerszy opis czujników gazu znalazł się w artykule „Cyfrowe czujniki gazu" [8].

Parametry modułu SEN55 [1]:

- **Pomiar pyłów (PM)**
  - Dokładność pomiaru masowego: ±10%
  - Zakres pomiaru masowego: 0...1000 μg/m³
  - Czas pracy: > 10 lat
  - Rozmiary cząstek: PM1,0, PM2,5, PM4, PM10
- **Temperatura (T)**
  - Dokładność: 0,45°C
  - Zakres: –10...+50°C
  - Czas odpowiedzi (τ63%): < 60 s
- **Wilgotność względna (RH)**
  - Dokładność: 4,5% RH
  - Zakres: 0…90% RH
  - Czas odpowiedzi: (τ63%) 20 s
- **Lotne związki organiczne (VOC)**
  - Indeks VOC: 0...500
  - Zakres pomiaru (TVOC): 0...1000 ppm
  - Czas odpowiedzi (τ63%): < 10 s
  - Czas detekcji (po włączeniu): < 60 s

eprasa.pl 65e3eb482b

- Czas osiągnięcia pełnej dokładności (po włączeniu): < 1 h
- Kompensacja wilgotności

- **Parametry ogólne**
  - Zasilanie (min./typ./max.): 4,5/5,0/5,5 V
  - Pobór prądu (średni): 63 mA
  - Pobór prądu (szczytowy): 110 mA
  - Interfejs komunikacyjny: I²C
  - Rozmiar (DxSxW): 52,8×43,6×22,3 mm

Pomiar poziomu pyłów (PM)

Materiał w postaci pyłu zawieszonego, w skrócie PM (Particulate Matter), jest mieszaniną unoszących się w powietrzu cząstek stałych i kropelek cieczy, które – wdychane przez człowieka – mogą powodować poważne problemy zdrowotne. Kategoria PM obejmuje cząstki o różnych właściwościach – tj. kształcie, właściwościach optycznych, wielkości i składzie – ale najczęściej dzieli się je na podkategorie na podstawie rozmiaru. Różne kategorie składowych są zwykle określane zgodnie ze wspólną nomenklaturą PM.x, gdzie „x" określa maksymalną średnicę cząstek w ich mieszance w powietrzu lub „aerozolu" [7].

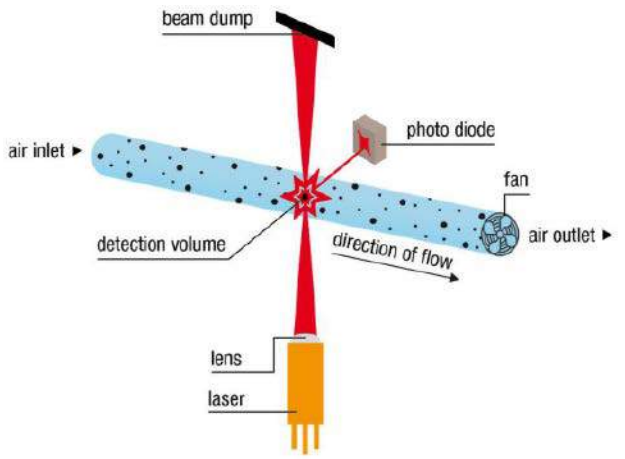
Typowa definicja PM obejmuje cząstki o wielkości od 0,1 do 10 mikrometrów. Najczęściej spotykaną klasyfikacją rozmiarową jest podział na dwie frakcje [2]:

- PM2,5 – wszystkie aerozole atmosferyczne o wielkości cząstek 2,5 mikrometra lub mniejszej, w skład których wchodzą zwykle stosunkowo reaktywne związki organiczne i nieorganiczne (np. azotan amonu), smog, bakterie, drobny kurz;
- PM10 – wszystkie cząstki o wielkości 10 mikrometrów lub mniejszej, w skład których wchodzą zwykle stosunkowo obojętne chemicznie związki, takie jak krzemionka i tlenki metali, kurz, pyłki roślinne, zarodniki pleśni.

Cząstki PM10 i PM2,5 są niewidoczne gołym okiem.

Czujniki pyłów firmy Sensirion to optyczne liczniki cząstek w czasie rzeczywistym (OPC) działające w oparciu o rozpraszanie laserowe. Wszystkie OPC kierują zawieszone pyły z otoczenia do komory pomiarowej (zawierającej źródło światła – np. laser – i fotodetektor) wewnątrz urządzenia (rysunek 1). W wyniku interakcji cząstek i fotonów część przychodzącego światła ulega rozproszeniu w kierunku fotodetektora (**rysunek 2**).

Zebrany sygnał jest przeliczany na aktualne wartości liczby cząstek i stężenia masowego, podawane odpowiednio w #/cm³ i µg/m³. Chociaż OPC są dość porównywalne pod względem wydajności przy zliczaniu cząstek, algorytm przekształcający zmierzony sygnał na stężenie masowe jest prawdziwym czynnikiem różnicującym wydajność urządzeń dostarczanych przez różnych producentów. Dzieje się tak dlatego, że parametry optyczne pyłów (takie jak współczynnik załamania światła i kształt) mają duży wpływ na oszacowanie masy cząstki. Z tego też względu pośredni pomiar optyczny (laserowy) charakteryzuje się pewną rozbieżnością w porównaniu z dokładniejszymi metodami grawimetrycznymi (opartymi na masie) [7]. Jeśli jednak czujniki PM mają być stosowane w produktach komercyjnych przy rozsądnych kosztach, najwygodniejszą i najbardziej praktyczną implementacją technologii pomiaru okażą się OPC.



Rysunek 2. Sposób pracy czujnika poziomu pyłów [6]

VOC – lotne związki organiczne

Gazy redukujące to związki, które reagują z tlenem atmosferycznym katalizowanym na nagrzanych powierzchniach, takich jak warstwa tlenku metalu czujników SGP4x. Przykładami gazów redukujących są wodór (H_2), lotne związki organiczne (LZO), tlenek węgla (CO) i metan (CH_4). Typowe czujniki VOC typu MOX (jak układy SGP4x) reagują tylko na H_2 i LZO.

VOC (lotne związki organiczne) to zbiorcze określenie substancji zawierających wiązania węgiel-wodór, które odparowują w temperaturze pokojowej lub wyższej. Pochodzą one z bardzo różnych źródeł i pod wieloma względami nie zostały dokładnie zdefiniowane. VOC są emitowane przez istoty żywe, źródła naturalne, chemikalia lub procesy spalania. Należy również pamiętać, że ludzie także wydychają gazowy H_2, więc typowy czujnik VOC MOX będzie reagował również na wodór, nawet jeśli stężenie VOC jest niskie. Chociaż ludzki nos stanowi doskonały detektor gazów, nie jest w stanie wykryć gazów bezwonnych lub tych, które występują w niskich stężeniach.

NO_x – tlenki azotu

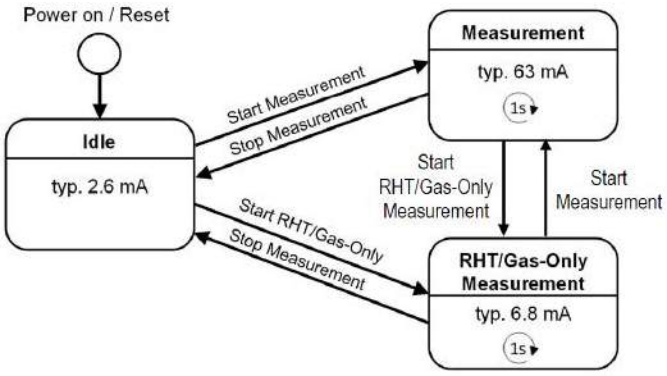
NO_x to – obok VOC – główne zanieczyszczenia gazowe w pomieszczeniach zamkniętych. Mówiąc najprościej, gazy utleniające to wysoce reaktywne gazy, które dostarczają więcej tlenu do procesów spalania niż zwykłe powietrze.

Najczęstszymi przykładami są tlenki azotu (NO_x), ozon (O_3) i sam tlen (O_2). NO_x odnosi się głównie do tlenku azotu (NO) i dwutlenku azotu (NO_2), które współistnieją. Ich wysoka reaktywność oznacza, że są niestabilne w kontakcie z powierzchnią. Dlatego też gazów utleniających w pomieszczeniach zamkniętych jest zwykle znacznie mniej w porównaniu z VOC. Jeśli jednak istnieje stałe źródło, stężenia NO_x we wnętrzach budynków mogą osiągnąć szkodliwy poziom. Tlenki azotu powstają w wyniku spalania paliw kopalnych i przyczyniają się do zwiększania zanieczyszczeń cząstkami stałymi. Gotowanie na kuchenkach gazowych lub zanieczyszczone powietrze zewnętrzne, dostające się z ruchliwych dróg, wpływa na jakość powietrza w pomieszczeniach. Jednakże tlenki azotu powstają również w wyniku naturalnych procesów atmosferycznych. Związki te – zwłaszcza dwutlenek azotu – podrażniają i uszkadzają układ oddechowy. Podwyższone ich stężenia w powietrzu, którym oddychamy, mają negatywny wpływ na czynność płuc.

Sensirion Gas Index Algorithm

Oprogramowanie Sensirion Gas Index Algorithm wylicza indeks VOC i NO_x na podstawie surowych sygnałów wejściowych. Algorytm ten umożliwia niezawodne wykrywanie zmian VOC i NO_x przy minimalnych różnicach pomiędzy danymi z różnych czujników. Algorytm opiera się na statystycznej normalizacji wzmocnienia oraz przesunięcia i stale dostosowuje oba parametry, stosując wykładniczo malejącą funkcję wyuczonych parametrów, aby móc dostosować się do zmieniających się warunków środowiskowych. Ważne jest, aby surowe sygnały podawać do odpowiedniego algorytmu ze stałym interwałem próbkowania, który musi pokrywać się z okresem odczytu surowych sygnałów z czujnika. Domyślny interwał próbkowania zastosowany w algorytmie wynosi 1 s. Algorytm oblicza sygnały indeksu VOC i NO_x rekurencyjnie na podstawie pojedynczej wartości mierzonej przez czujnik w każdym kroku czasowym, a także aktualizowanych synchronicznie zmiennych stanu. Te stany wewnętrzne to przede wszystkim rekurencyjnie oszacowana średnia i wariancja odpowiedniego sygnału, a także dodatkowe zmienne, takie jak czas pracy urządzenia. Po oszacowaniu stanów algorytm konwertuje surowe sygnały

eprasa.pl 65e3eb482b



Rysunek 3. Tryby pracy modułu czujników środowiska SEN5x [2]

w znacznikach – odpowiednio – na indeks VOC lub NOx i stosuje adaptacyjny filtr dolnoprzepustowy.

Algorytm domyślnie odwzorowuje odczyty VOC na skalę VOC od 1 do 500, gdzie 100 odnosi się do średniego stanu, a wartości NOx są odwzorowywane na skalę NO_x od 1 do 500, w której wszystkie wartości powyżej 1 są nieprawidłowe [5].

Tryby pracy modułu czujników środowiska

Moduł czujników SEN5x oferuje trzy różne tryby pracy: uśpienia, pomiaru pełnego i pomiaru RHT/Gas-Only. Po włączeniu moduł SEN5x przechodzi automatycznie w tryb uśpienia. Z tego trybu czujnik można przełączyć w jeden z pozostałych stanów. **Rysunek 3** prezentuje tryby pracy modułu SEN5x:

Tryb uśpienia

- Moduł znajduje się w trybie uśpienia po włączeniu zasilania lub zresetowaniu.
- Większość wewnętrznej elektroniki zostaje wyłączona, co redukuje zużycie energii.
- Wentylator i laser pozostają wyłączone.
- Moduł jest gotowy do przyjęcia i realizacji dowolnego polecenia.

Tryb pomiaru pełnego

- Cała elektronika włączona (maksymalny pobór prądu).
- Pomiar trwa i moduł w sposób ciągły przetwarza dane pomiarowe.
- Nowe odczyty są dostępne co sekundę.

Tryb pomiaru RHT/Gas-Only

- Wentylator i laser są wyłączone, aby zmniejszyć zużycie energii.
- RHT i czujnik gazu są włączone. Trwa pomiar RHT oraz gazu, a moduł w sposób ciągły przetwarza dane pomiarowe.
- Nowe odczyty są dostępne co sekundę.
- Do trybu pomiaru RHT/Gaz można wejść z trybów uśpienia i pomiaru pełnego.
- Pomiar pyłu zwraca stałą wartość równą 0xFFFF.

W trybie pomiaru SEN5x dostarcza nowe dane z interwałem próbkowania wynoszącym jedną sekundę. W niektórych zastosowaniach podawanie wartości PM z taką częstotliwością nie jest konieczne, warto więc wtedy przełączyć moduł w tryb RHT/Gas-Only, w którym wentylator i laser są wyłączone, ale włączone pozostają czujniki VOC, NO_x i RHT (jeśli występują w rozważanym wariancie urządzenia). W przypadku SEN50 tryb obniżonego poboru mocy uzyskuje się, przełączając pomiędzy trybem pomiarowym a trybem uśpienia.

Skrócenie czasu pomiaru PM pozwala znacznie zaoszczędzić zużycie energii. Kiedy czujnik pyłów znajduje się w trybie pomiarowym (po pierwszych 60 sekundach), typowy średni prąd zasilania wynosi 63 mA, natomiast maksymalnie sięga 70 mA. Biorąc pod uwagę tryb rozruchu i inne tryby pracy, maksymalny prąd szczytowy wynosi 110 mA. W trybie pomiaru RHT/Gas-Only pobór prądu zostaje zmniejszony do 6,8 mA [4].

Właściwe, naprzemienne używanie tych trybów pracy, w sposób pokazany na **rysunku 4**, może zmniejszyć zużycie energii 7...9 razy przy minimalnym uszczerbku dla wydajności systemu czujników.

Tryb pomiarowy podzielony jest na dwa etapy: charakterystyczne dla pierwszej fazy w trybie pomiarowym jest uruchomienie wentylatora i gromadzenie danych statystycznych przez algorytm PM. Na tym etapie dane pomiarowe SEN5x nie są stabilne i należy je odrzucić. W drugiej fazie dane PM, VOC, NO_x i RHT są aktualizowane co sekundę. Z tego trybu moduł można przełączyć na tryb pomiaru RHT/Gas-Only, w którym dane VOC, NO_x i RHT również są aktualizowane co sekundę. W celu ponownego uzyskania PM cykl jest powtarzany.

Optymalizacja czasu pracy w trybie pomiaru

Gdy czujnik pyłów zostanie wprowadzony w tryb pomiarowy, laser i wentylator ulegną automatycznemu włączeniu. Aby oszczędzać energię, można ustawić sensor na tryb pomiaru RHT/Gas-Only. Aby jednak zachować wysoką dokładność ogólnego pomiaru, należy wziąć pod uwagę różne czynniki ograniczające minimalny czas pracy w trybie pełnym.

Ze względu na bezwładność wentylatora, osiągnięcie docelowej prędkości obrotowej zajmuje kilka sekund. Efekt ten można zaobserwować także na wyjściu czujnika w ciągu pierwszych sekund pracy. Ponieważ do wykrywania cząstek stałych wymagane są statystyki, czas do osiągnięcia typowej stabilnej wartości wyjściowej zależy również od stężenia cząstek w próbkowanym powietrzu. Przy niższych stężeniach czujnik potrzebuje więcej czasu niż przy stężeniach wyższych.

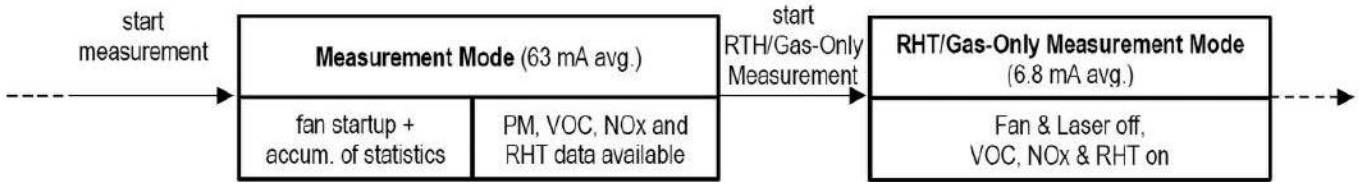
Dalsza optymalizacja jest możliwa poprzez dostosowanie czasu rozruchu do faktycznie zmierzonego stężenia – zgodnie z typowymi czasami rozruchu podanymi w arkuszu danych SEN5x. Należy pamiętać, że typowe wartości tego parametru są wartościami średnimi, rzeczywisty czas rozruchu bywa różny dla poszczególnych modeli czujników i różnych typów aerozoli.

Dwie główne zmienne wpływające na całkowite zużycie energii, które należy uwzględnić w wydajności systemu, to czasy spędzone w trybach pełnego pomiaru oraz w trybie RTH/Gas-Only.

Można zastosować następujący przykładowy algorytm w celu dostosowania czasu rozruchu do rzeczywistego stężenia cząstek stałych [5]:

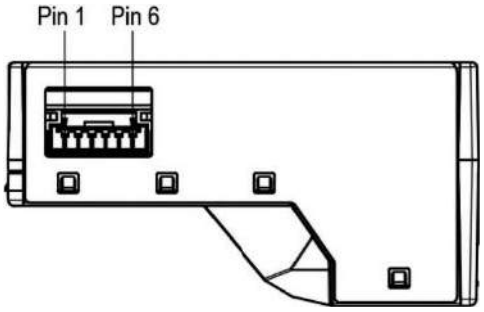
- Pomiar po 16 sekundach
  - Jeśli stężenie >100 #/cm³ → dane pomiarowe nadają się do przetworzenia.
  - Jeżeli stężenie ≤100 #/cm³ → rozruch nie został zakończony, należy dokonać ponownego pomiaru po 30 sekundach.
- Pomiar po 30 sekundach → dane pomiarowe są gotowe do przetworzenia.

Po okresie rozruchowym zaleca się pobranie kilku punktów danych dla pomiaru PM i uśrednienie ich w celu uzyskania stabilnego pomiaru. Dobrym punktem wyjścia dla osiągnięcia najlepszej dokładności jest uśrednienie wartości pomiarów z kolejnych 30 sekund po okresie uruchamiania, co oznacza, że całkowity czas pracy w trybie pomiarowym wynosi 60 sekund. Dokument [5] pokazuje przykład kodu implementacji tej procedury. Krótszy okres uśredniania pozwoli zaoszczędzić energię, ale może skutkować zmniejszoną powtarzalnością pomiaru.



Rysunek 4. Jeden pełny cykl pracy ze zmniejszoną mocą [4]

eprasa.pl 65e3eb482b



Rysunek 5. Usytuowanie złącza komunikacyjnego modułu czujników SEN55 [2]

Tabela 1. Rozmieszczenie sygnałów na złączu komunikacyjnym modułu SEN55 [2]

Nr pinu	Nazwa	Opis
1	VCC	Zasilanie 5 V ±10%
2	GND	Masa
3	SDA	I²C: We/wy komunikacyjne, kompatybilne: TTL 5 V, LVTTL 3,3 V
4	SCL	I²C: We komunikacyjne, kompatybilne: TTL 5 V, LVTTL 3,3 V
5	SEL	Wybór protokołu komunikacji, GND – I²C
6	NC	Niepodłączone

Automatyczne czyszczenie wentylatora

Gdy moduł pracuje w trybie pomiaru, uruchamiana jest procedura automatycznego czyszczenia wentylatora po upływie określonego czasu pracy. Powoduje to przyspieszenie wentylatora do maksymalnej prędkości na 10 sekund, aby wydmuchać kurz nagromadzony wewnątrz niego.

- Wartości pomiarów nie są aktualizowane, gdy trwa czyszczenie wentylatora.
- Interwał czyszczenia jest ustawiony na 604 800 sekund (tj. 168 godzin lub 1 tydzień) z tolerancją ±3%.
- Interwał można skonfigurować za pomocą polecenia *Ustaw interwał automatycznego czyszczenia*.
- Ustaw interwał na 0, aby wyłączyć automatyczne czyszczenie.
- Reset czujnika pyłów przestawia częstotliwość czyszczenia do wartości domyślnej.
- Jeśli czujnik pyłów zostanie wyłączony, licznik czasu ulegnie zresetowaniu do 0. Należy uruchomić cykl czyszczenia przynajmniej raz w tygodniu, jeśli czujnik jest okresowo wyłączany i włączany (np. raz dziennie).
- Procedurę czyszczenia można także rozpocząć ręcznie za pomocą polecenia *Rozpocznij czyszczenie*.

Kompensacja temperatury

Domyślnie sygnały wyjściowe temperatury i wilgotności z czujnika są kompensowane ze względu na samonagrzewanie się modułu. Jeśli moduł został zamontowany w urządzeniu, może zaistnieć potrzeba skorygowania kompensacji temperatury, aby uwzględnić zmianę sprzężenia termicznego i wpływ temperatury innych elementów urządzenia [2].

Podłączanie modułu SEN5x

Moduł ma małą obudowę (o rozmiarach zewnętrznych 52,5×43,3×22,3 mm) o dosyć nieregularnym kształcie (**rysunek 5**) [2]. Wiatraczek modułu pracuje bardzo cicho (24 dB(A)). Sygnały i zasilanie dostępne są na nietypowym złączu 6-pinowym z rozstawem 1,25 mm (rysunek 5). Rozmieszczenie sygnałów na złączu komunikacyjnym modułu SEN55 pokazane zostało w **tabeli 1**. Obudowa nie ma połączenia z zaciskiem masy (2) i powinna pozostać odizolowana lub dołączona do uziemienia.

Razem z modułem SEN5x w zestawie dostarczane są dwa kable z wtyczkami do gniazdka złącza komunikacyjnego (**fotografia tytułowa** [1]). Z drugiej strony pierwszego kabla znajduje się wtyczka RJ-45 przeznaczona do dołączenia do mostka SEK-SensorBridge. Natomiast z drugiej strony kabla drugiego zamocowane zostały wtyczki BLS, kompatybilne z gniazdkami typu goldpin.

Interfejs I²C

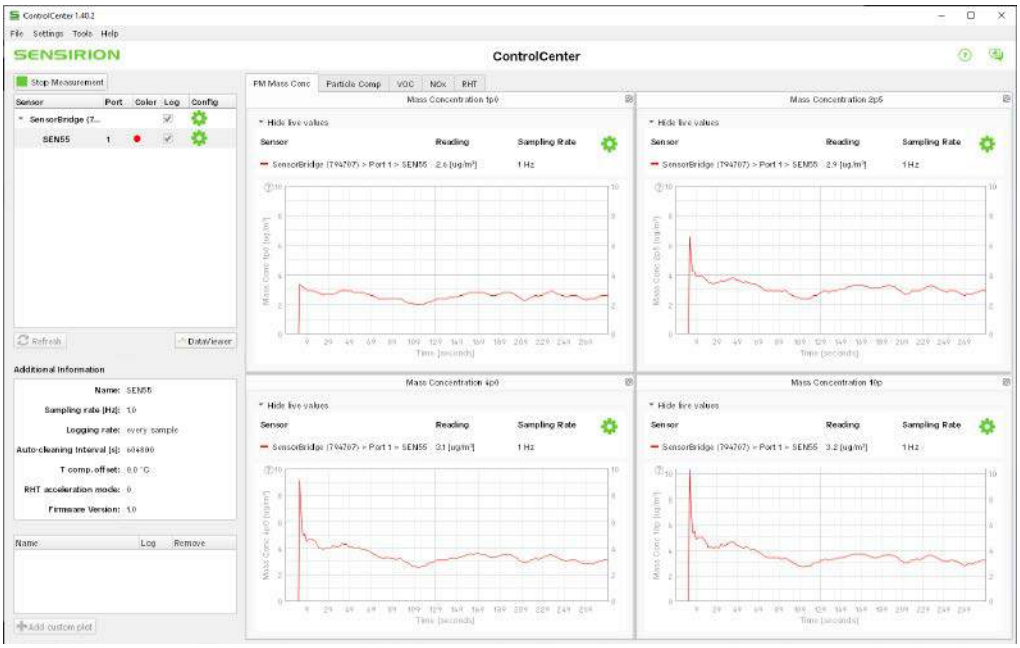
Moduł obsługuje transmisję protokołu I²C z prędkością 100 kbit/s. Aby prawidłowo wybrać interfejs I²C, pin wyboru interfejsu (SEL) trzeba dołączyć do GND przed lub w tym samym czasie, gdy moduł otrzymuje zasilanie. Obie linie SCL i SDA są wejściami/wyjściami z otwartym drenem i należy podłączyć je do zewnętrznych rezystorów podciągających (np. R=10 kΩ). Moduł ma stały adres I²C: 0x69. Polecenia odczytu i zapisu przesyłają dane w 2-bajtowych pakietach, po których następuje 8-bitowa suma kontrolna, obliczana w sposób podany w dokumencie [2].

Moduł SEN5x można łatwo zastosować we własnej aplikacji dzięki firmowym bibliotekom programowym. Dostępne są uniwersalne drivery w językach C i Python, a także biblioteki dla platform Raspberry Pi (język C) oraz Arduino [1].

Mostek SEK-SensorBridge

Mostek SEK-SensorBridge firmy Sensirion jest przeznaczony do jednoczesnej obsługi dwóch portów w celu dołączenia dwóch zestawów ewaluacyjnych z czujnikami marki Sensirion [10].

Mostek może być wpięty do komputera poprzez gniazdko microUSB (zestaw staje się widoczny w systemie operacyjnym jako wirtualny port szeregowy) albo poprzez łącze transmisji szeregowej w standardzie RS485 (**fotografia 1**, po lewej). Nie ma ograniczeń co do liczby mostków SensorBridge, podłączonych do tego samego komputera.



Rysunek 6. Zakładka *PM Mass Concentration* aplikacji ControlCenter z podłączonym modułem SEN55

eprasa.pl 65e3eb482b

Fotografia 1. Mostek SEK-SensorBridge, widok z tyłu (z lewej) i z przodu (po prawej) [12]

Każde z dwóch złączy pomiarowych RJ45 może zostać użyte do podłączenia jednego zestawu ewaluacyjnego czujnika za pomocą dostarczonego kabla (**fotografia 1**, po prawej). Na złączu znajduje się masa, zasilanie (ustawiane w zakresie 1,2 V...5,5 V) oraz linie komunikacyjne w standardach I²C (do 1 MHz), UART i SPI.

SensorBridge zachowuje kompatybilność z oprogramowaniem ControlCenter firmy Sensirion. Czujniki kompatybilne z płytką SensorBridge pracują od razu po podłączeniu do mostka. Firma Sensirion udostępnia też bibliotekę w języku Python, która umożliwia połączenie z czujnikiem poprzez mostek SensorBridge.

Oprogramowanie ControlCenter

ControlCenter firmy Sensirion to aplikacja umożliwiająca natychmiastowe rozpoczęcie pracy z czujnikami producenta [11]. Sensoryczne moduły ewaluacyjne mogą być dołączone do komputera poprzez mostek SensorBridge lub bezpośrednio poprzez konwerter USB-UART. Główną funkcjonalnością ControlCenter jest wykonywanie i wizualizacja pomiarów na żywo z różnych czujników. Po pobraniu najnowszej wersji aplikacji ze strony ControlCenter [11] należy ją zainstalować razem z driverami.

Zestaw ewaluacyjny SEK-SEN5x

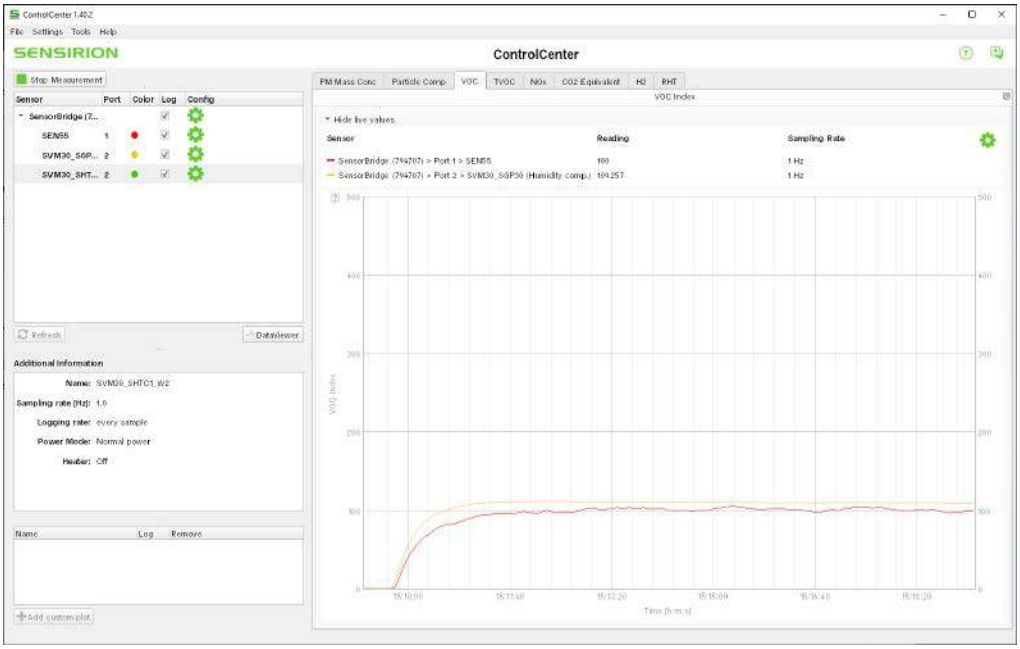
Najszybszy sposób na rozpoczęcie eksperymentu z modułem SEN5x to użycie zestawu ewaluacyjnego SEK-SEN5x firmy Sensirion [3] w połączeniu z mostkiem SEK-SensorBridge i oprogramowaniem Sensirion ControlCenter. W zestawie dostarczane są dwa kable z wtyczkami do gniazdka złącza komunikacyjnego modułu SEN5x, opisane we wcześniejszej części artykułu.

1. Dedykowanym kablem podłącz zestaw ewaluacyjny SEK-SEN5x (z modułem SEN55) do gniazdka złącza pomiarowego RJ45 mostka SEK-SensorBridge.
2. Kablem microUSB podłącz mostek SEK-SensorBridge do komputera.
3. Uruchom aplikację ControlCenter.
4. Kliknij *Refresh* (jeśli potrzeba). Przy kolejnych uruchomieniach aplikacji czujnik SEN55 będzie już automatycznie identyfikowany, a dodatkowo wygenerowany zostanie zestaw predefiniowanych wykresów.

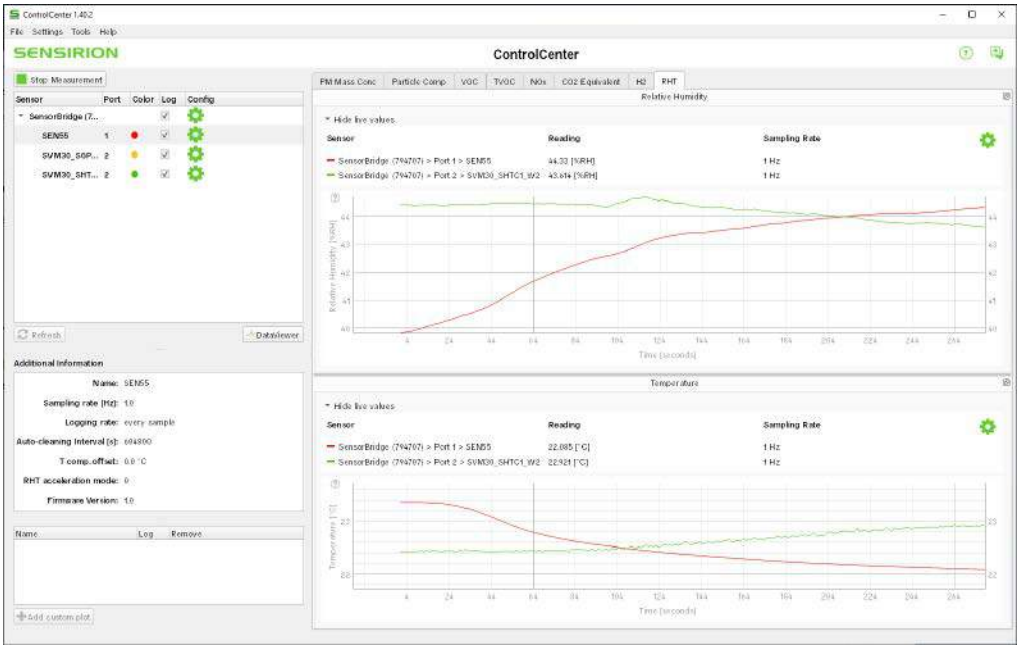
Częstotliwość wykonywania pomiarów jest ustawiana domyślnie na 1 Hz. Można pobierać każdą próbkę – lub co 2/5/10/100/1000 (decymacja). Można też wykonywać uśrednianie na próbkach z ustawionego zakresu. Automatyczne czyszczenie czujnika pyłów ustawione jest domyślnie na okres co 7 dni, ale istnieje również opcja natychmiastowego wykonania czyszczenia na żądanie użytkownika.

Oś czasu wykresów ustawiona została na czas upływający od uruchomienia pomiaru. Druga możliwość to czas astronomiczny. Domyślne ustawienie długości pokazywanego przebiegu to 300 s. Zakres wartości osi pionowej jest ustalany automatycznie i ma predefiniowany poziom minimalny.

Na rysunku 6 widnieje ekran aplikacji ControlCenter po podłączeniu modułu SEN55. Wykresy pokazują policzone koncentracje



Rysunek 7. Zakładka *VOC* aplikacji ControlCenter z podłączonym modułem SEN55 oraz SVM30



Rysunek 8. Zakładka *RHT* aplikacji ControlCenter z podłączonym modułem SEN55 oraz SVM30

eprasa.pl 65e3eb482b

w czerech zakresach PM1,0, PM2,5, PM4 oraz PM10. Pomiar rozpoczęto w pomieszczeniu o małej zawartości pyłów. Zwraca uwagę długi czas (ok. 60 s) potrzebny na dojście krzywych do prawdziwej wartości pomiarowej. W zakładce *Particle Computation* można zobaczyć pomiary zliczania cząstek. Zakładka NO_x pokazuje indeks NO_x, jednak ze względu na brak istotnego stężenia takich związków w badanym powietrzu wyświetlana jest stała wartość „1".

Nic nie stoi na przeszkodzie, by do mostka SEK-SensorBridge dołączyć jednocześnie drugi moduł czujników. Na **rysunku 7** można zobaczyć zakładkę *VOC* aplikacji ControlCenter po podłączeniu modułu SEN55 oraz SVM30 (po rozpoczęciu pomiarów). Widoczny jest długi czas ustalania prawidłowego odczytu VOC mierzonego przez SEN55 – w porównaniu z odczytem z czujnika SGP30 modułu SVM30. Jednak w stanie stabilnym obie wartości okazują się zbliżone.

Na **rysunku 8** pokazana została zakładka *RHT* aplikacji ControlCenter z podłączonym modułem SEN55 oraz SVM30 (po rozpoczęciu pomiarów). Uwagę zwraca bardzo długi czas ustalania odczytu temperatury (T) oraz wilgotności względnej (RH) mierzonej przez SEN55 – w porównaniu z precyzyjnym czujnikiem SHTC1 modułu SVM30. Aplikacja ControlCenter umożliwia zastosowanie trzech trybów przyspieszania korygowania pomiarów RHT modułu SEN55. W pokazanym przykładzie zastosowano tryb *Low Acceleration*.

Na **rysunku 9** uwidoczniono wykres RHT aplikacji ControlCenter po dłuższym czasie pracy czujników w stabilnych warunkach. Pomiary z modułu SEN55 znacznie różnią się od pomiarów referencyjnych z czujnika SHTC1 modułu SVM30.

Dodatkowe pomiary RHT wykonywane przez moduł Smart Gadget firmy Sensirion z czujnikiem SHTC31 umieszczonym obok też przynoszą różniące się nieco wyniki. Wymiana modułu SEN55 na inny egzemplarz pokazała znacznie szybsze zmiany wartości pomiarów po włączeniu zasilania oraz większą zgodność z pomiarami zewnętrznymi. Zastosowanie trybu *High Acceleration* w celu przyspieszenia korekcji pomiarów RHT modułu SEN55 powoduje wolniejsze zmiany wartości pomiarów, ale lepszą zgodność z pomiarami referencyjnymi.

Wymiana modułu SVM30 skutkowała dużo mniejszymi zmianami wartości jego pomiarów po włączeniu zasilania przy zbliżonych poziomach mierzonych wielkości. Dokładność pomiarów RHT wykonywanych poza warunkami laboratoryjnymi stanowi bardzo duży problem.

Na **rysunku 10** pokazany jest wykres *Particle Comp* aplikacji ControlCenter po rozpaleniu kominka w sąsiednim pomieszczeniu. Wyraźnie wyższy poziom stężenia mniejszych cząstek wskazuje na obecność dymu [7].

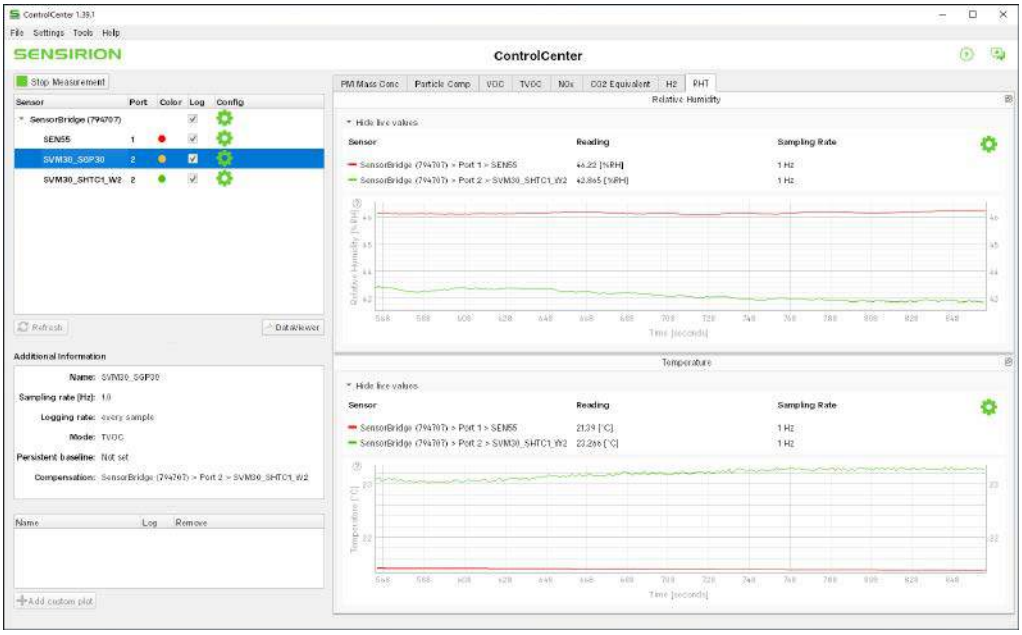
DataViewer

Oprócz wizualizacji wykresów na żywo, inną funkcjonalnością ControlCenter jako narzędzia ewaluacyjnego jest ułatwianie rejestrowania pomiarów wykonywanych w ramach różnych eksperymentów.

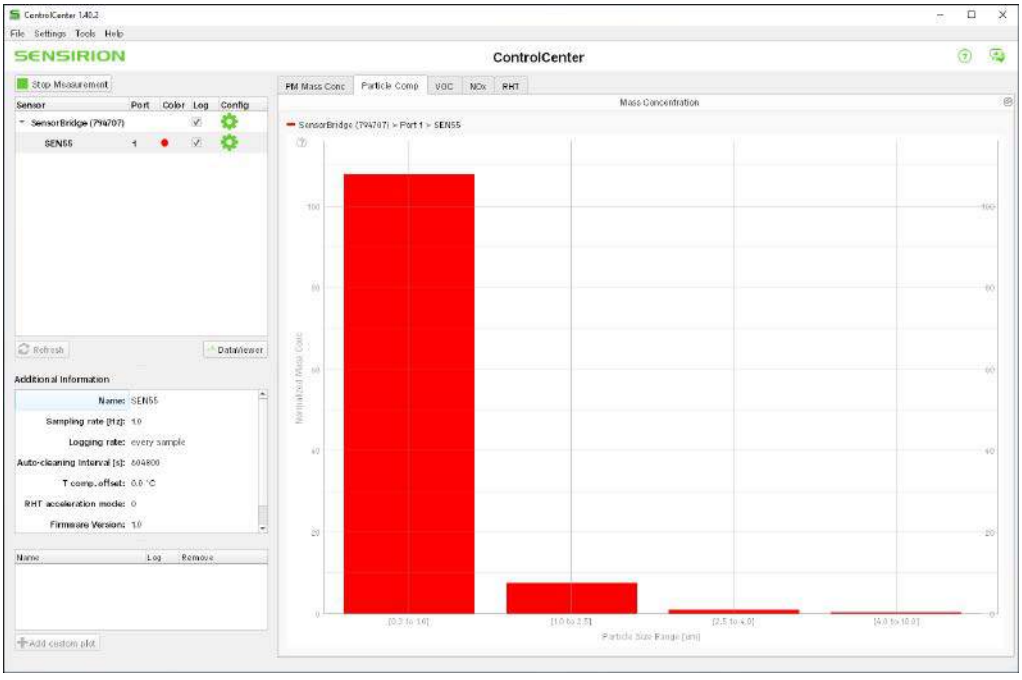
Domyślnie rejestracja do pliku zostaje włączona dla wszystkich czujników i rozpoczyna się wraz z pomiarem.

Logi zapisywane są w plikach EDF. Plik EDF stanowi zastrzeżony format Sensirion wzorowany na formacie CSV. Dlatego jest kompatybilny z Microsoft Excel i innymi edytorami obsługującymi pliki CSV.

Aplikacja DataViewer – siostrzane narzędzie programowe aplikacji ControlCenter – umożliwia wizualizację zarejestrowanych pomiarów. Oprogramowanie rozpoznaje pliki EDF i wizualizuje pomiary w formie wykresów. Aplikacja DataViewer to oddzielny program, który można uruchomić w poprzez aplikację ControlCenter (najprościej: klikając przycisk *DataViewer* po lewej stronie okna aplikacji



Rysunek 9. Wykres *RHT* aplikacji ControlCenter po dłuższym czasie pracy

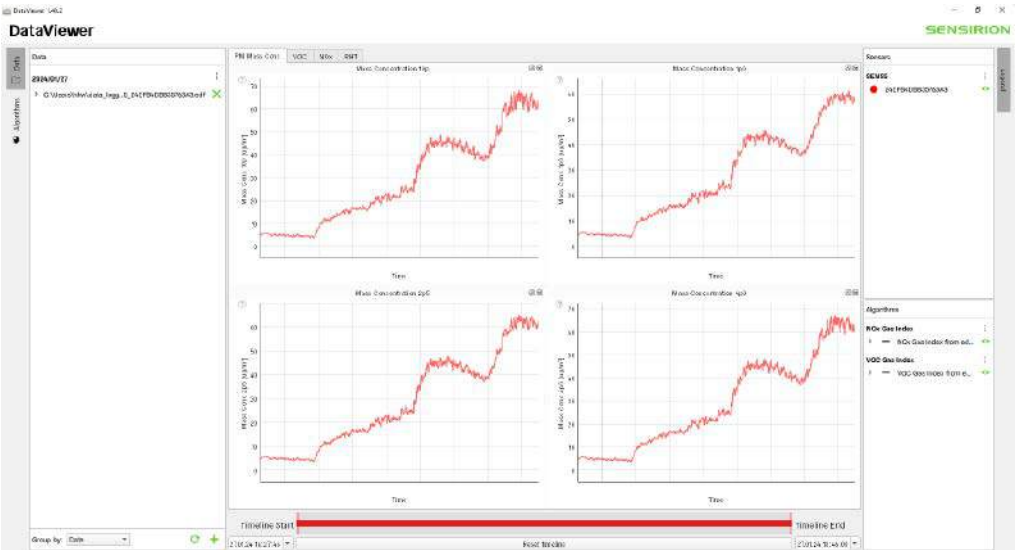


Rysunek 10. Wykres *Particle Comp* aplikacji ControlCenter

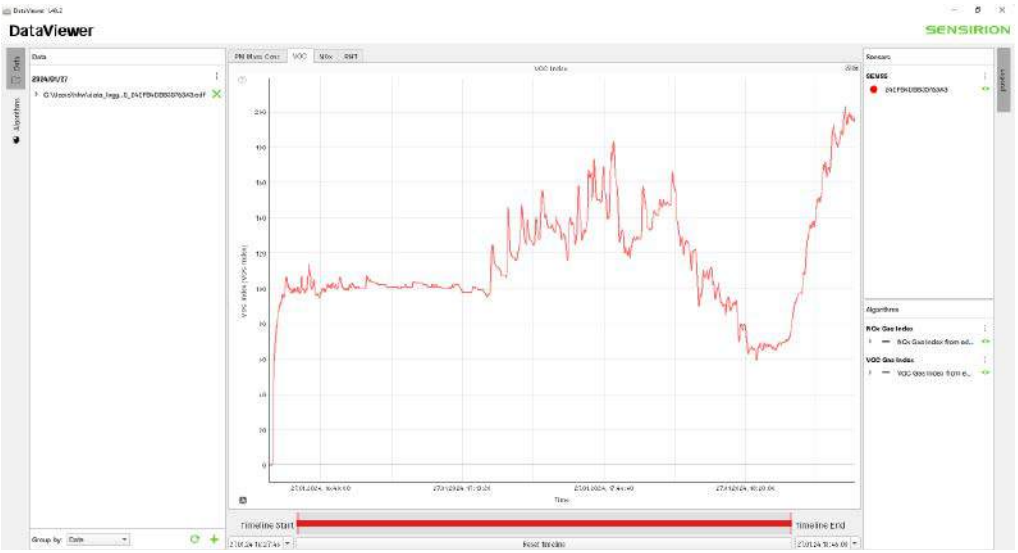
eprasa.pl 65e3eb482b

ControlCenter). Program, po uruchomieniu, automatycznie ładuje bieżący dziennik pomiarów, także w trakcie jego rejestrowania. Drugi sposób uruchomienia aplikacji DataViewer z menu Tools → Launch DataViewer umożliwia wskazanie pliku EDF do wyświetlenia.

Na **rysunku 11** widoczny jest zapis koncentracji pyłów, zarejestrowany przed rozpaleniem kominka w sąsiednim pomieszczeniu – i po nim. **Rysunek 12** prezentuje zapis pomiaru indeksu VOC z tym samym czasem rejestracji. Znaczne obniżenie poziomu pyłów i indeksu VOC w około godziny 18.20 spowodowane było przez uchylenie okna w tym pomieszczeniu.



Rysunek 11. Zakładka *PM Mass Concentration* aplikacji DataViewer



Rysunek 12. Zakładka VOC aplikacji DataViewer

Podsumowanie

Moduł czujników środowiskowych SEN5x firmy Sensirion to znaczący postęp w stosunku do czujnika pyłów SPS30 tego samego producenta. A już poprzednik był zdecydowanie najlepszym czujnikiem w swojej dziedzinie. Nowy moduł – dużo mniejszy i wygodniejszy w użyciu – dodatkowo mierzy VOC i NO_x. Zwraca uwagę zupełnie cicha praca modułu SEN5x, pomimo zastosowanego wiatraczka.

Wykrycie skutków uszkodzenia uszczelki drzwiczek kominka, podczas prób z modułem czujników środowiska SEN55, pokazuje jego wielką przydatność, nawet w warunkach domowych.

Na połowę roku 2024 zapowiadana jest premiera modułów czujników środowiska z nowej serii SEN6x firmy Sensirion. Mają to być dwa warianty; oprócz mierzonych przez obecną serię SEN5x parametrów otoczenia, będą one (w zależności od wersji) mierzyć poziom albo dwutlenku węgla (CO_2) albo formaldehydu (HCHO). Całość zamknięta zostanie w bardzo małej obudowie o wymiarach 54×24×22 mm.

Henryk A. Kowalski
Instytut Informatyki
Politechnika Warszawska

Literatura

[1] SEN55 Environmental sensor node for PM, RH/T, VOC, NO_x measurements, https://tiny.pl/dq6j8
[2] Datasheet SEN5x, 01/2022, https://tiny.pl/dq6js
[3] SEK-SEN5x Evaluation kit SEN5x series, https://tiny.pl/dq6jv
[4] Reduced Power Operation SEN5x, Application note, 08/2022, https://tiny.pl/dq6ph
[5] Engineering guidelines SEN5x, User guide, 01/2022, https://tiny.pl/dq6px
[6] Sensor specification statement Particulate Matter Sensors, Application note, 03/2020, https://tiny.pl/dq6jl
[7] *Cyfrowe czujniki pyłów*, Henryk A. Kowalski, „Elektronika Praktyczna", 6/2020, https://tiny.pl/dq6jj
[8] *Cyfrowe czujniki gazu*, Henryk A. Kowalski, „Elektronika Praktyczna", 3/2020, https://tiny.pl/czkh1
[9] *Czujnik pomiaru pyłów SPS30 firmy Sensirion*, Henryk A. Kowalski, „Elektronika Praktyczna", 9/2020, https://tiny.pl/dq6j5
[10] SEK-SensorBridge, https://tiny.pl/dq6j9
[11] ControlCenter, https://tiny.pl/dq6jr
[12] ControlCenter Manual, https://tiny.pl/dq6jt

REKLAMA

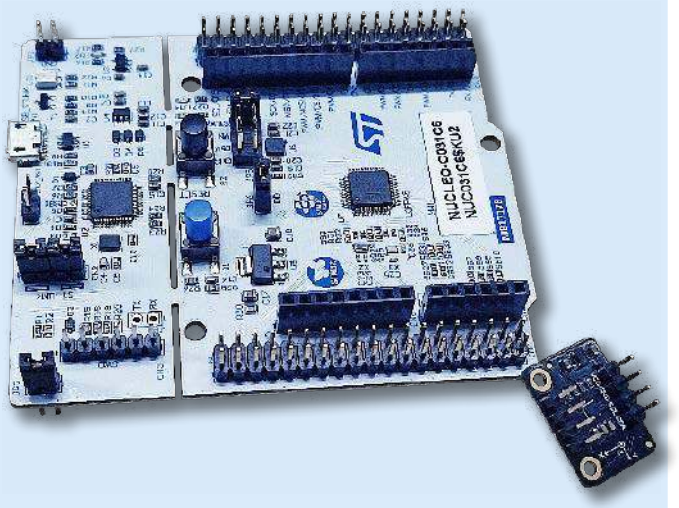
http://www.ep.com.pl/EPwtoku

eprasa.pl 65e3eb482b

Crash Course STM32C0

– programowanie mikrokontrolerów ARM w rejestrach

Programowanie prostszych mikrokontrolerów (np. AVR, PIC, MSP430 czy też przestarzałych już 8051 bądź HCS08) bez użycia bibliotek, tj. przy wykorzystaniu samych tylko plików nagłówkowych z definicjami rejestrów i zawartych w nich bitów, jest raczej naturalną konsekwencją nieskomplikowanej architektury tych procesorów. Bardziej rozbudowane układy – w szczególności te oparte na rdzeniach ARM – są zwykle nieporównanie trudniejsze do opanowania na niskim poziomie abstrakcji, stąd większość programistów systemów wbudowanych korzysta w swojej codziennej pracy z bibliotek. Niniejszy kurs ma na celu pokazanie innej ścieżki rozwoju i – mamy nadzieję – przekona przynajmniej część spośród naszych Czytelników do zaprzyjaźnienia się z wymagającą, ale niezwykle wartościową metodą programowania układów STM32.



Wprowadzenie

Pamiętam, jak kilkanaście lat temu pewien doświadczony elektronik-programista powiedział mi, że – cytuję – „w przypadku STM32 wystawienie jedynki na port wymaga ukończenia wyższych studiów technicznych". Choć niewątpliwie w zdaniu tym jest pewna przesada, to jednak wbrew pozorom nie jest ono pozbawione ziarnka prawdy. Faktycznie: skonfigurowanie nawet najprostszych, podstawowych peryferiów nowoczesnego procesora ARM jest zadaniem dalece bardziej złożonym niż wykonanie tego samego zadania np. na poczciwym układzie ATtiny czy MSP430. Niesłusznie byłoby jednak zrzucać całą winę za złożoność programowania 32-bitowców na karb samego tylko rdzenia. Dużo większe znaczenie ma bowiem po prostu konstrukcja wbudowanych bloków peryferyjnych, które znajdujemy w „dużych" mikrokontrolerach – a ta zależy już tylko od decyzji poszczególnych producentów, opracowujących procesory z rdzeniami na licencji ARM.

Wśród programistów pokutuje zatem przekonanie, że programowanie 32-bitowców w rejestrach okazuje się niezwykle trudne, a sam kiedyś spotkałem się z niedowierzaniem, że w ogóle jest to możliwe. Do pisania kodu na najniższym możliwym poziomie (zaraz po asemblerze) zniechęca także powszechne stwierdzenie, że taki sposób pracy byłby zwyczajnie nieopłacalny, ponieważ kod napisany w rejestrach nieszczególnie nadaje się do przenoszenia pomiędzy różnymi rodzinami procesorów tego samego producenta. Po części istotnie tak jest – osoby znające procesory STM32 od podszewki z pewnością spotkały się już z (nierzadko zaskakującymi) różnicami pomiędzy określonymi grupami układów, np. STM32L1, STM32F4, STM32F0 czy też STM32F3. Co ciekawe, bliższe zapoznanie się z obszerną dokumentacją pokazuje dobitnie, że o ile niektóre peryferia są niemal identyczne w większości linii produktowych, o tyle dywersyfikacja nazewnictwa i funkcjonalności rejestrów w innych blokach sprzętowych potrafi przyprawić o ból głowy. Mało tego – producentowi zdarza się (choć rzecz jasna nie w złej wierze) zastawiać na programistów niskopoziomowych pewne pułapki, np. poprzez przesunięcie bitów lub nawet fragmentów pól bitowych w ramach rejestrów o tej samej nazwie i (w przybliżeniu) tożsamym przeznaczeniu.

Czy pisanie kodu w rejestrach ma zatem jakiś sens? Odpowiedź jak najbardziej brzmi TAK! W przypadku rodziny STM32C0 nie należy bowiem spodziewać się szczególnego zróżnicowania modeli układów, gdyż ich znacząca rozbudowa zanadto zbliżyłaby je do „mainstreamowych" linii produktowych, takich jak STM32F0 czy STM32L0. Jeżeli zatem ograniczymy nasz nowy sposób pracy nawet tylko do tej jednej grupy układów, to problemy z przenośnością kodu przestaną mieć znaczenie, zyskamy natomiast szereg innych benefitów. Jakich?

Oto najważniejsze zalety niskopoziomowego podejścia do programowania STM32 (i nie tylko).

1. **Wydajność obliczeniowa** – kod pisany z użyciem gołych rejestrów może być wysoce zoptymalizowany pod względem szybkości wykonywania, co ma znaczenie szczególnie w przypadku procedur realizujących pewne zadania na granicy maksymalnej szybkości sprzętu. Przykładowo – jeżeli procesor musi z zawrotną prędkością obsługiwać krótkie sygnały impulsowe lub np. generować pewne sekwencje sygnałów wyjściowych, to niskopoziomowe programowanie będzie w takim przypadku optymalnym rozwiązaniem: łatwiejszym niż pisanie w asemblerze, a zarazem szybszym i bardziej przewidywalnym od tworzenia kodu np. z użyciem bibliotek HAL.
2. **Oszczędność pamięci** – patrząc na współczesne portfolio procesorów ARM, można odnieść wrażenie, że ograniczenia pod względem dostępnej pamięci Flash stają się coraz mniej dotkliwe, nawet w przypadku rozbudowanych aplikacji. Nie należy jednak zapominać, że wśród mikrokontrolerów 32-bitowych zdarzają się małe, lekkie układy o pamięci na poziomie 16 kB, a nawet 8 kB (czyli zbliżonym do mniejszych przedstawicieli rodzin AVR bądź PIC). Przy tak ograniczonych zasobach sprzętowych oszczędność pamięci jest zaletą niepodważalną.
3. **Doskonała znajomość procesora** – programując wyłącznie z użyciem bibliotek, można dość łatwo wpaść w pułapkę pewnych uproszczeń – biblioteki wiele zadań wykonują niejako za programistę, co sprawia, że często nie ma on potrzeby zaglądania do tego,

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie
https://ulubionykiosk.pl/media

eprasa.pl 65e3eb482b

co dzieje się pod maską wysokopoziomowych funkcji i struktur biblioteki HAL. Tymczasem pisanie kodu niskopoziomowego wymusza dokładne poznanie struktury określonych peryferiów (a także samego rdzenia) na poziomie nie tylko rejestrów, ale wręcz poszczególnych ich bitów. Programiści wyposażeni w tę cenną wiedzę potrafią znacznie lepiej radzić sobie z problemami, wynikającymi np. z braków w dokumentacji bibliotek czy też rozmaitych błędów – i to zarówno w strukturze API, jak i samego sprzętu (tzw. błędy w krzemie).

4. **Stuprocentowa przejrzystość kodu** – kod napisany w rejestrach być może sam w sobie nie należy do najbardziej intuicyjnych dzieł programistycznych (chyba że zostanie odpowiednio opatrzony komentarzami, do czego gorąco zachęcamy!). Ma jednak pewną ogromną zaletę – nie ukrywa żadnych kruczków (np. potencjalnych błędów) pod kolejnymi warstwami abstrakcji i wywołaniami wielokrotnie zagnieżdżonych hierarchicznie funkcji, ułatwia zatem pełną kontrolę przepływu danych i sygnałów sterujących. Może to być istotny atut w przypadku systemów o krytycznym znaczeniu dla bezpieczeństwa, nie musimy bowiem bazować na bibliotekach, w przypadku których (z natury rzeczy) możemy nie mieć 100-procentowego przekonania co do ich stabilności oraz niezawodności.

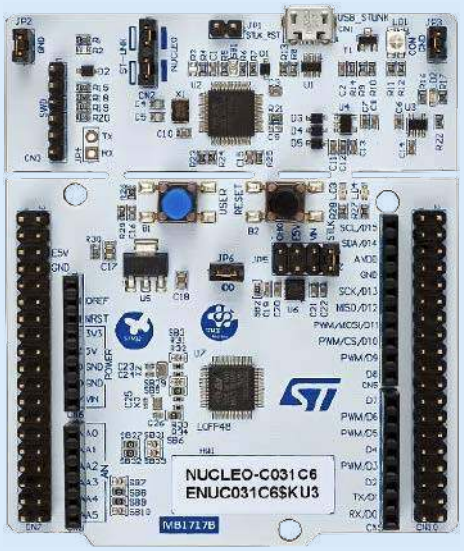
Fotografia 1. Dowód na to, jak firma ST Microelectronics próbuje podbić rynek 8-bitowców za pomocą nowej rodziny STM32C0 – moduł STMC0316-DK w roli zamiennika oryginalnego procesora ATmega328 na płytce Arduino Uno (źródło: nota aplikacyjna AN5780)

Niniejszy kurs oparty został na procesorze z najnowszej serii STM32C0, dumnie promowanej przez firmę ST Microelectronics jako produkt, mający na celu zastąpienie (żeby nie powiedzieć – wyparcie z rynku) mikrokontrolerów 8-bitowych. I tutaj ciekawa dygresja – kilkanaście lat temu, gdy pierwsze układy STM32F1 dopiero rozkręcały się na rynku MCU, niemal równolegle z nimi marka ST promowała także własną rodzinę 8-bitowców – STM8S. Nie da się ukryć, że grupa ta nie zrobiła szczególnej furory w świecie systemów embedded, a dziś jej status handlowy to NRND (nierekomendowane do nowych projektów). W międzyczasie ST solidnie zainwestował w rozwój procesorów STM32, stając się jednym z potentatów na tym obszarze. Strategiczne posunięcie, jakim było opracowanie serii lekkich, niewielkich procesorów ARM w niewiarygodnie wręcz niskim przedziale cenowym (najtańsze układy mają kosztować przy zamówieniach hurtowych tylko 24 centy!), wydaje się więc strzałem w dziesiątkę. Firma ST Microelectronics nie kryje się zresztą za bardzo z chęcią wyparcia z rynku kultowych już mikrokontrolerów ATmega328 (stanowiących fundament ekosystemu Arduino), o czym świadczy oficjalny moduł, przeznaczony do… wpinania do podstawki DIL28 w miejsce oryginalnego procesora na Arduino Uno (**fotografia 1**).

Trudno w tym momencie przewidzieć, czy wspomniane posunięcie marketingowe odniesie sukces – z dużą dozą prawdopodobieństwa można jednak stwierdzić, że nie odbędzie się to w takim zakresie, jakiego prawdopodobnie oczekują menedżerowie ST (najnowsze moduły z serii Arduino – np. Arduino Uno R4 – także bazują już wszak na ARM-ach, więc trudno będzie zdominować ten segment rynku). Na pewno jednak najnowsza gałąź drzewa genealogicznego rodziny STM32 znajdzie swoje miejsce w tysiącach aplikacji komercyjnych, zarówno tych prostych i niskobudżetowych (jako główne kontrolery), jak i w rozbudowanych systemach wieloprocesorowych (jako peryferyjne układy do realizacji prostszych zadań). Tym bardziej więc warto poznać bliżej te niezwykle ciekawe procesory.

Niniejszy kurs jest przeznaczony przede wszystkim dla:

- **programistów doświadczonych w pracy z STM32** z użyciem bibliotek HAL lub LL, zainteresowanych rozwojem w zakresie niskopoziomowego tworzenia kodu,



Fotografia 2. Widok płytki ewaluacyjnej NUCLEO-C031C6 (http://t.ly/QLIj7)

- **programistów mikrokontrolerów 8-bitowych**, którzy dzięki naszym materiałom będą mogli w szybki i przystępny sposób przejść na nową, znacznie wydajniejszą platformę (nie zmieniając przy tym zanadto swoich przyzwyczajeń – tym bardziej że styl nazewnictwa funkcji i struktur, zastosowany zwłaszcza w bibliotekach HAL, może naprawdę przytłoczyć zwolenników czystego, schludnego i zwartego kodu źródłowego).

Sprzęt oraz potrzebne akcesoria

Kurs został zaplanowany z myślą o możliwie najniższych nakładach finansowych, związanych z zakupem sprzętu niezbędnego do realizacji opisanych zadań. Dlatego też wybór padł na jedną z nielicznych (ale łatwo dostępną także w Polsce) płytek ewaluacyjnych dla nowej rodziny procesorów STM32. Mowa o zestawie NUCLEO-C031C6 (**fotografia 2**), który – wbrew przyjętej przez ST nomenklaturze – wprawdzie bazuje na rozmiarze płytki znanym dotąd z układów w obudowach 64-pinowych, ale w istocie jest wyposażony w układ 48-nóżkowy. Niestety, wiążą się z tym pewne ograniczenia pod względem kompatybilności z niektórymi nakładkami funkcjonalnymi z serii Nucleo.

Drugim elementem niezbędnym do realizacji zadań kursowych będzie moduł akcelerometru MEMS, zawierający układ LIS2DW12. Programy były testowane za pomocą modułu z serii Fermion marki DFRobot (kod handlowy producenta: SEN0405 – patrz **fotografia tytułowa**), ale na rynku można znaleźć także kilka innych

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie
https://ulubionykiosk.pl/media

eprasa.pl 65e3eb482b

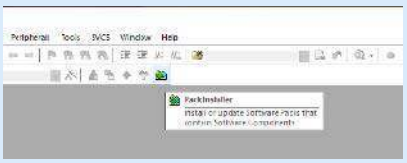
modułów o zbliżonej funkcjonalności. W razie potrzeby zaprezentowane przykłady da się także łatwo zmodyfikować pod kątem zastosowania z innymi modelami akcelerometrów (lub zupełnie innymi czujnikami) – przygotowane listingi zostały dokładnie opisane komentarzami, więc nic nie stoi na przeszkodzie, by zamiast czujnika MEMS użyć np. sensora wilgotności, temperatury bądź dowolnego innego modułu I²C/SPI o napięciu linii I/O równym 3,3 V. Oprócz płytki Nucleo oraz modułu z czujnikiem do wykonania zadań kursowych będzie rzecz jasna potrzebny kabel micro USB, kilka przewodów z żeńskimi wtykami BLS oraz komputer z systemem Windows (co nie oznacza, że użytkownicy Linuksa bądź macOS nie mogą wykonać opisanych ćwiczeń – informacje na ten temat znajdują się w dalszej części artykułu).

Instalacja oprogramowania

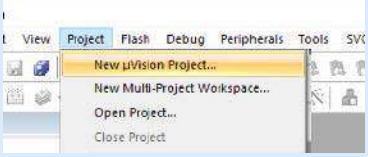
Krótki czas, który upłynął od rynkowej premiery STM32C0, sprawia, że nie wszystkie środowiska IDE zdołano zaadaptować do współpracy z naszym bohaterem. W chwili pisania niniejszego kursu (luty 2024 r.) najlepszym wyborem zdaje się być zainstalowanie oprogramowania Keil µVision 5 i temu właśnie zadaniu poświęcimy tę część artykułu.

Produktem potrzebnym do realizacji zadań kursowych jest środowisko IDE w wersji specjalnej, kompatybilnej jedynie z procesorami ARM Cortex-M0 oraz M0+ marki ST Microelectronics: Keil MDK for STMicroelectronics Edition. Pełna instrukcja instalacji wraz z linkami i kodem PSN, niezbędnym do uzyskania bezpłatnej licencji, znajduje się w dokumentacji dostępnej pod adresem:
https://developer.arm.com/documentation/kan344/latest

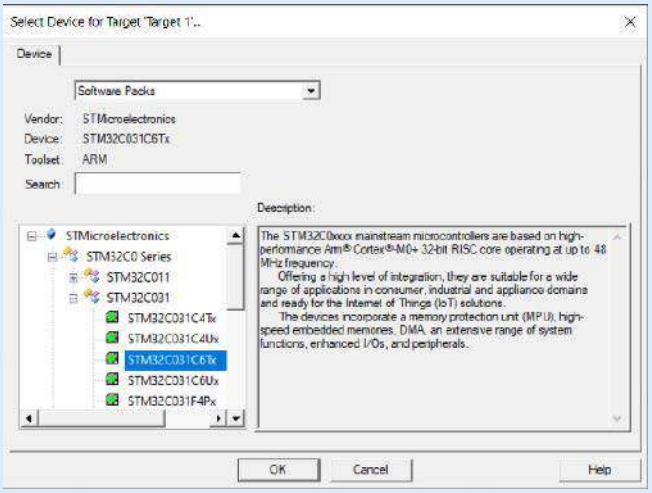
Po pobraniu pliku wykonywalnego należy go uruchomić i przeprowadzić (w sposób typowy dla środowiska Windows) proces instalacji. Gdy instalacja dobiegnie końca, trzeba włączyć program i wybrać opcję *Pack Installer* z jednego z górnych pasków narzędzi (**rysunek 1**), co spowoduje otwarcie okna z obszerną listą układów (zakładka *Devices* – **rysunek 2**). Należy teraz wybrać rodzinę STM32C0, choć nic nie stoi na przeszkodzie, by przy okazji doinstalować także pakiety dla STM32F0, STM32G0 oraz STM32L0 (a nawet dla układów Bluetooth



Rysunek 1. Przycisk uruchamiający okno instalacji pakietów



Rysunek 3. Opcja utworzenia nowego projektu



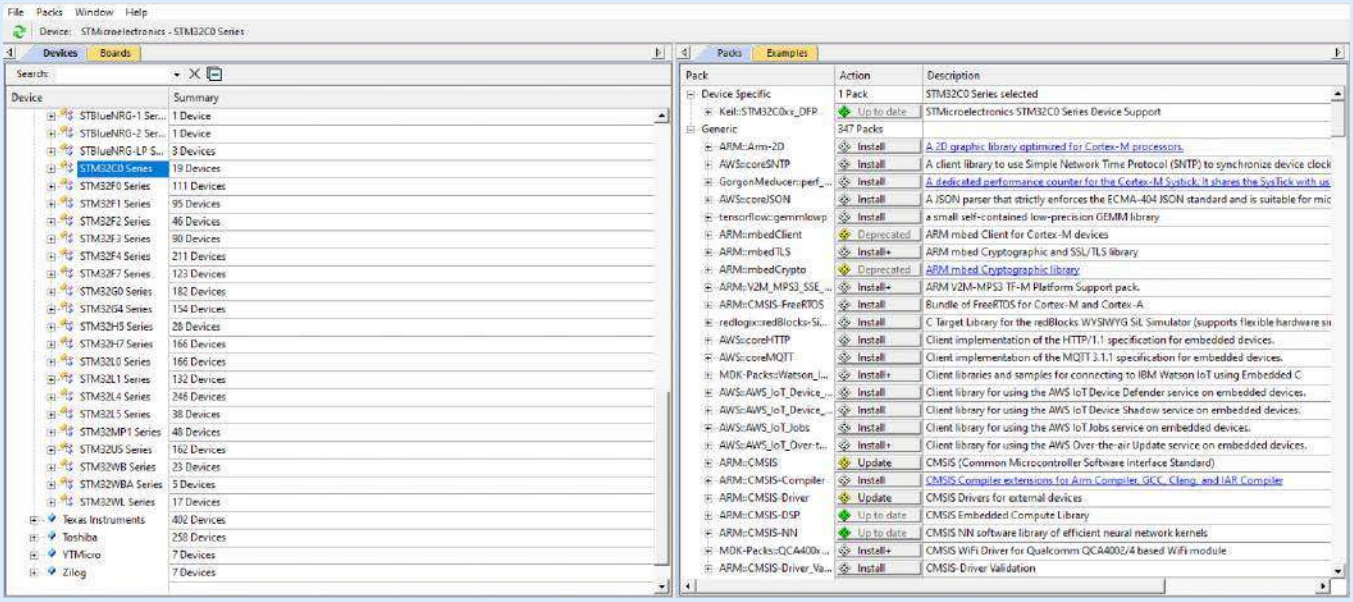
Rysunek 4. Okno wyboru docelowego mikrokontrolera

z serii BlueNRG). W prawym panelu, w zakładce *Packs*, należy wybrać do instalacji pakiety Keil::STM32C0xx_DFP oraz ARM::CMSIS.

Kolejnym etapem przygotowania środowiska programistycznego jest uzyskanie licencji od firmy ARM. Proces ten został dokładnie opisany w instrukcji producenta i nie będziemy go szczegółowo omawiać – dość powiedzieć, że aby otrzymać kod aktywacyjny, trzeba zastosować wspomniany powyżej kod PSN, wpisując go w oknie *License Management*, uruchomionym z menu *File* (program należy przy tym włączyć z uprawnieniami administracyjnymi). Do uzyskania licencji konieczne będzie jeszcze wypełnienie formularza online, zaś po otrzymaniu wiadomości e-mail z kodem LIC – wpisanie go w oknie *License Management* i zaakceptowanie.

Utworzenie i konfiguracja projektu

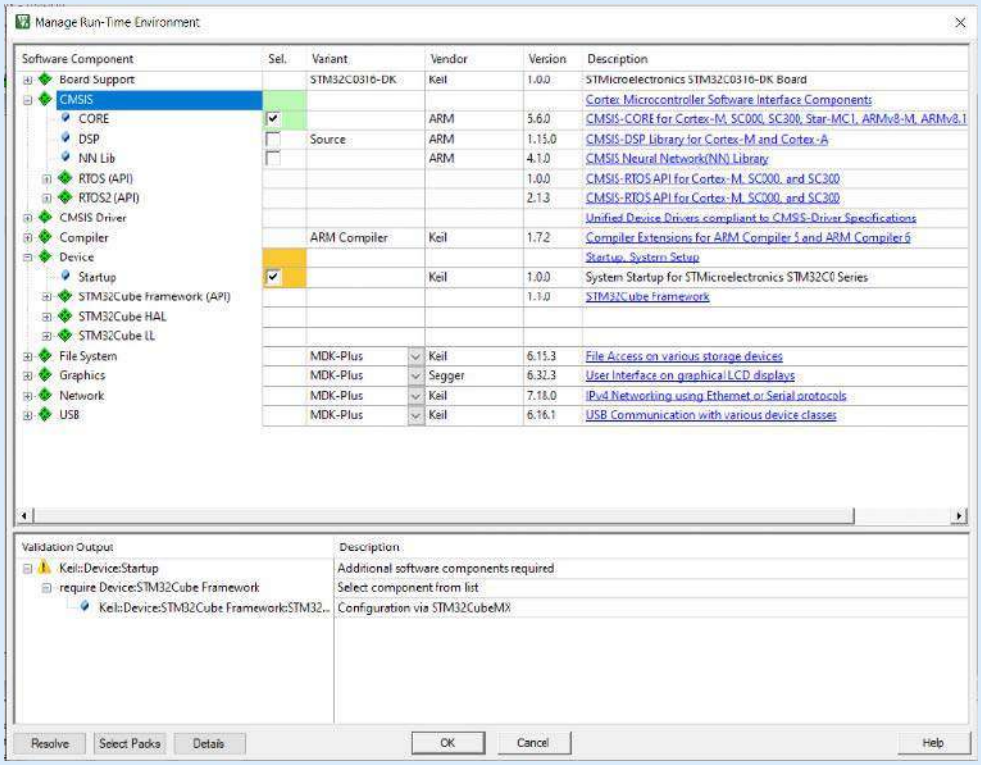
W tak przygotowanym środowisku klikamy pozycję *New µVision Project* z menu *Project* (**rysunek 3**), co spowoduje otwarcie okna widocznego na **rysunku 4** i umożliwiającego wybór



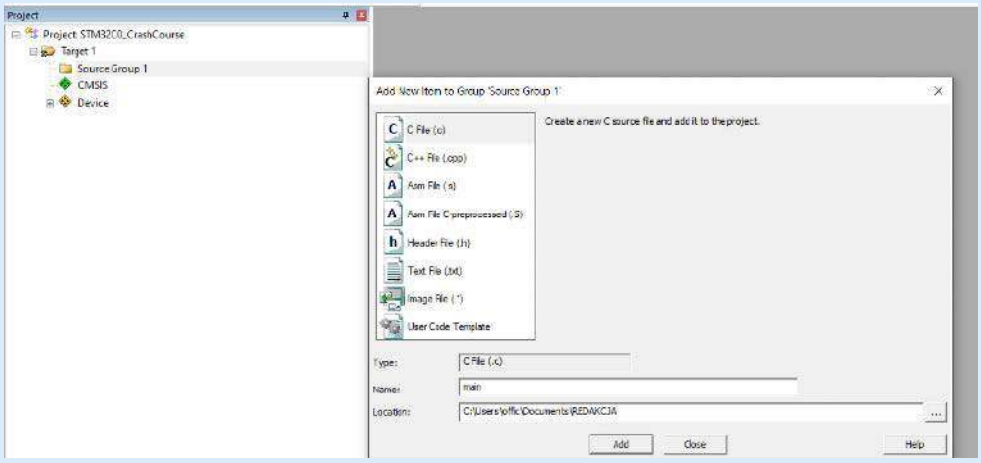
Rysunek 2. Okno instalacji pakietów

eprasa.pl 65e3eb482b

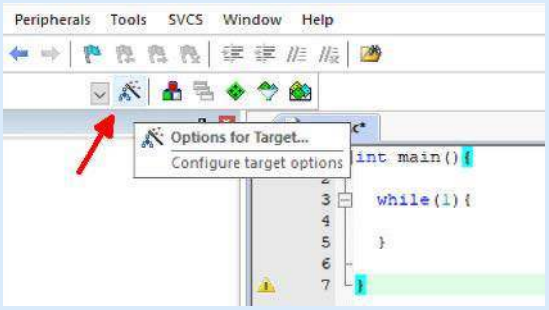
Rysunek 5. Przycisk uruchamiający okno Manage Run-Time Environment



Rysunek 6. Okno Manage Run-Time Environment



Rysunek 7. Okno wyboru rodzaju i nazwy pliku dodawanego do projektu



Rysunek 8. Przycisk otwierający okno opcji sprzętowych

docelowego procesora (w tej roli STM32C031C6Tx). W pustym IDE klikamy następnie przycisk *Manage Run-Time Environment* (**rysunek 5**), zaś w otwartym w ten sposób oknie wybieramy potrzebne pakiety: *CMSIS->CORE* oraz *Device->Startup* (**rysunek 6**). Program będzie starał się przekonać nas, że konieczny jest jeszcze komponent STM32CubeMX, jednak jako rasowi niskopoziomowcy nie będziemy korzystać z graficznego generowania konfiguracji peryferiów. Zatwierdzamy wybór przyciskiem *OK*, a następnie klikamy prawym klawiszem myszy pozycję *Source Group 1* w lewym panelu *Project*, co spowoduje otwarcie menu kontekstowego. Klikamy opcję *Add New Item to Group (...)*, otwierając w ten sposób okno umożliwiające wybór rodzaju pliku. Wpisujemy nazwę main oraz wybieramy typ pliku *C File (.c)* (**rysunek 7**) i klikamy *Add*. Otwarty zostanie pusty plik tekstowy, w którym będziemy mogli rozpocząć pisanie naszego kodu.

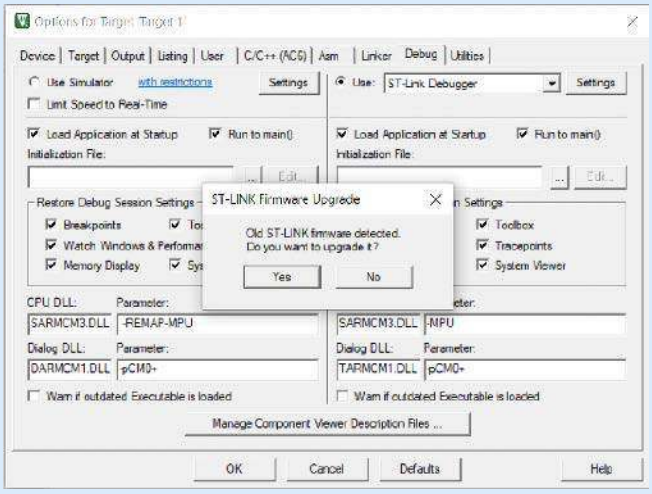
Kolejnym etapem konfiguracji jest ustawienie programatora/debuggera – w tym celu wybieramy przycisk *Options for Target* (**rysunek 8**), a w otwartym oknie, w zakładce *Debug*, ustalamy rodzaj interfejsu jako *ST-Link Debugger* (**rysunek 9**) i klikamy znajdujący się obok przycisk *Settings*. Program zapyta nas prawdopodobnie o chęć aktualizacji firmware'u programatora – potwierdzamy wybór przyciskiem *Yes*, przechodząc w ten sposób do okna aktualizacji (**rysunek 10**). Tutaj również wyrażamy zgodę na aktualizację, uprzednio wydając programowi polecenie połączenia się ze sprzętem (*Device Connect*, a następnie *Yes* >>>>).

Tak przygotowane środowisko jest gotowe do rozpoczęcia pracy.

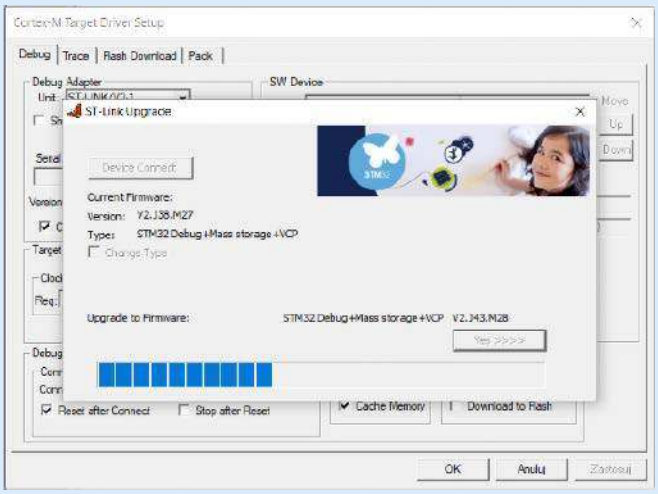
Na marginesie warto dodać, że użytkownicy środowiska STM32CubeIDE (dostępnego na systemy Windows, Linux oraz macOS) także mogą z powodzeniem wykonać ćwiczenia opisane w niniejszym kursie – w tym celu należy jednak zapoznać się z odpowiednimi instrukcjami, które można dość łatwo znaleźć w Internecie (na blogach poświęconych mikrokontrolerom oraz w tutorialach na portalu YouTube). Wszystkie pliki źródłowe udostępnione na serwerze ep.com.pl były testowane w obydwu środowiskach – zarówno Keil, jak i w najnowszej wersji STM32CubeIDE. Niestety, oficjalne IDE dla STM32 niezbyt wspiera działania programistów niskopoziomowych, dlatego trzeba skorzystać z pewnego obejścia domyślnych ustawień. Dodajmy więc tylko, że istnieją dwie drogi: można albo utworzyć normalny projekt, a następnie zignorować otwarty domyślnie graficzny generator kodu STM32CubeMX (i podmienić pliki źródłowe *main.c*, *stm32c0xx_it.c* oraz *stm32c0xx_it.h* na własne), albo utworzyć pusty projekt i dodać do niego odpowiednie zależności. Druga ścieżka jest znacznie lepsza, ale także bardziej pracochłonna – zainteresowanych

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie **https://ulubionykiosk.pl/media**

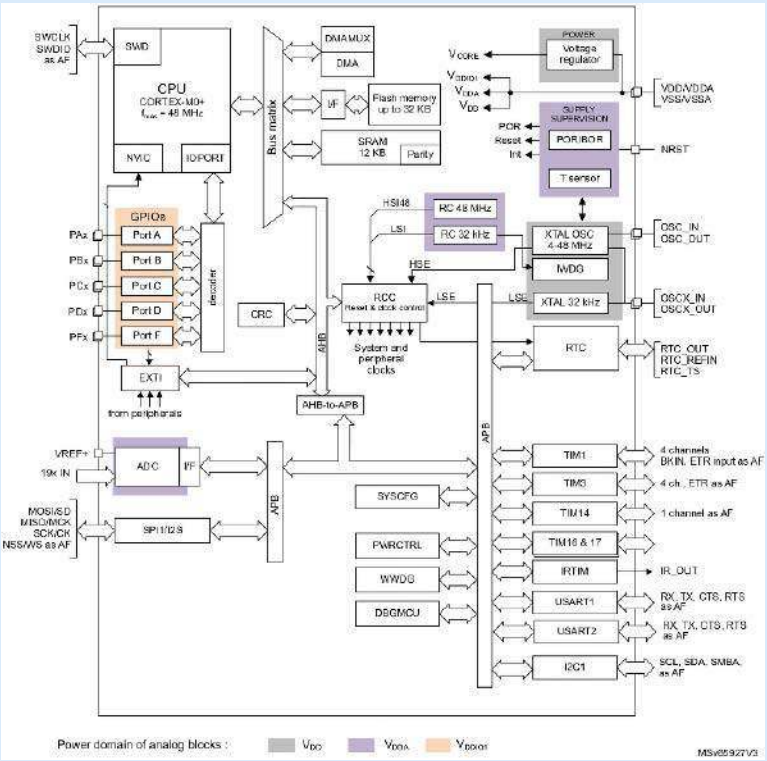
eprasa.pl 65e3eb482b



Rysunek 9. Okno opcji sprzętowych z otwartym alertem dot. aktualizacji firmware'u programatora ST-Link



Rysunek 10. Okno służące do aktualizacji firmware'u programatora ST-Link



Rysunek 11. Schemat blokowy mikrokontrolera STM32C031C6Tx

Czytelników zachęcam do podjęcia poszukiwań odpowiednich poradników w Internecie.

Garść informacji o mikrokontrolerze STM32C031C6T6

Schemat blokowy naszego bohatera można zobaczyć na **rysunku 11**. Układ zawiera 32 kB pamięci Flash, 12 kB pamięci RAM i może być taktowany sygnałem o częstotliwości do 48 MHz. Zakres napięć zasilania to 2,0...3,6 V, mikrokontroler jest produkowany w obudowie LQFP48. Na pokładzie znajduje się bogaty zestaw peryferiów sprzętowych:

- układy zarządzania zasilaniem (PWR) oraz taktowaniem i obwodami resetu (RCC),
- obwody zabezpieczające POR, BOR,
- wbudowane oscylatory 48 MHz oraz 32 kHz,
- oscylatory współpracujące ze zewnętrznymi rezonatorami kwarcowymi do 48 MHz (główny) i 32 kHz (pomocniczy dla RTC),
- 3-kanałowy kontroler DMA,
- 12-bitowy przetwornik ADC,
- timery/liczniki: 16-bitowy (zaawansowany) oraz cztery prostsze, także 16-bitowe,
- timer systemowy SysTick,
- watchdogi: okienkowy oraz niezależny,
- zegar czasu rzeczywistego z kalendarzem,
- interfejs I²C,
- dwa interfejsy USART,
- jeden interfejs SPI ze wsparciem I²S,
- sprzętowy blok obliczeń CRC,
- obwody wspierające generowanie sygnałów sterujących diodami nadawczymi IR (IRTIM).

Podczas realizacji zadań kursowych zdecydowanie warto mieć pod ręką dwa dokumenty: RM0490 (Reference manual. STM32C0x1 advanced Arm-based 32-bit MCUs) oraz notę katalogową naszego procesora (STM32C031x4/x6 datasheet). W pierwszym z nich znajduje się komplet informacji o budowie procesora i jego bloków peryferyjnych, sposobach konfiguracji oraz znaczeniu poszczególnych rejestrów i pól bitowych. W nocie katalogowej można znaleźć natomiast (oprócz szczegółowych parametrów elektrycznych, termicznych itp.) także dane dotyczące m.in. maksymalnych częstotliwości taktowania poszczególnych peryferiów, mapę pamięci, układ wyprowadzeń, a także tabele funkcji alternatywnych poszczególnych linii GPIO. W celu skrócenia opisu nie będziemy zatem zamieszczać w artykule map rejestrów, tabel z ustawieniami bitów, etc. – zachęcamy Czytelników do samodzielnego realizowania kursu równolegle z czytaniem oryginalnej dokumentacji.

Przykład 1. Blinky

Nasz pierwszy program – tradycyjnie w przypadku kursów programowania mikrokontrolerów – będzie miał za zadanie migać diodą, znajdującą się na płytce Nucleo. Realizację tej funkcjonalności zaczniemy od skonfigurowania rejestru RCC (Reset and Clock Control), czyli bloku peryferyjnego odpowiedzialnego za dostarczanie sygnałów taktujących i obsługę różnych źródeł resetu mikrokontrolera. Schemat bloku RCC (a dokładniej – drzewo połączeń sygnałów taktowania) można zobaczyć na **rysunku 12**. Osoby zaznajomione z STM32 błyskawicznie zauważą, że w tym przypadku nie mamy (spotykanego w innych rodzinach procesorów ST) bloku syntezera częstotliwości z pętlą PLL i dzielnikami – układy STM32C0 mogą przyjąć sygnał z zewnętrznego rezonatora kwarcowego, pracującego z częstotliwością maksymalną równą 48 MHz (taki też rezonator znajduje się na płytce Nucleo).

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie
https://ulubionykiosk.pl/media

eprasa.pl 65e3eb482b

Na początek jednak nie będziemy korzystać z zewnętrznego wzorca częstotliwości, użyjemy jedynie rejestru IOPENR, odpowiedzialnego za włączenie taktowania poszczególnych portów I/O. Do rejestru odwołamy się następująco:

```
RCC->IOPENR;
```

i przypiszemy do niego bity włączające (dla ułatwienia) od razu wszystkie porty dostępne na pokładzie naszego procesora. Według przyjętej konwencji bity te mają nazwy:

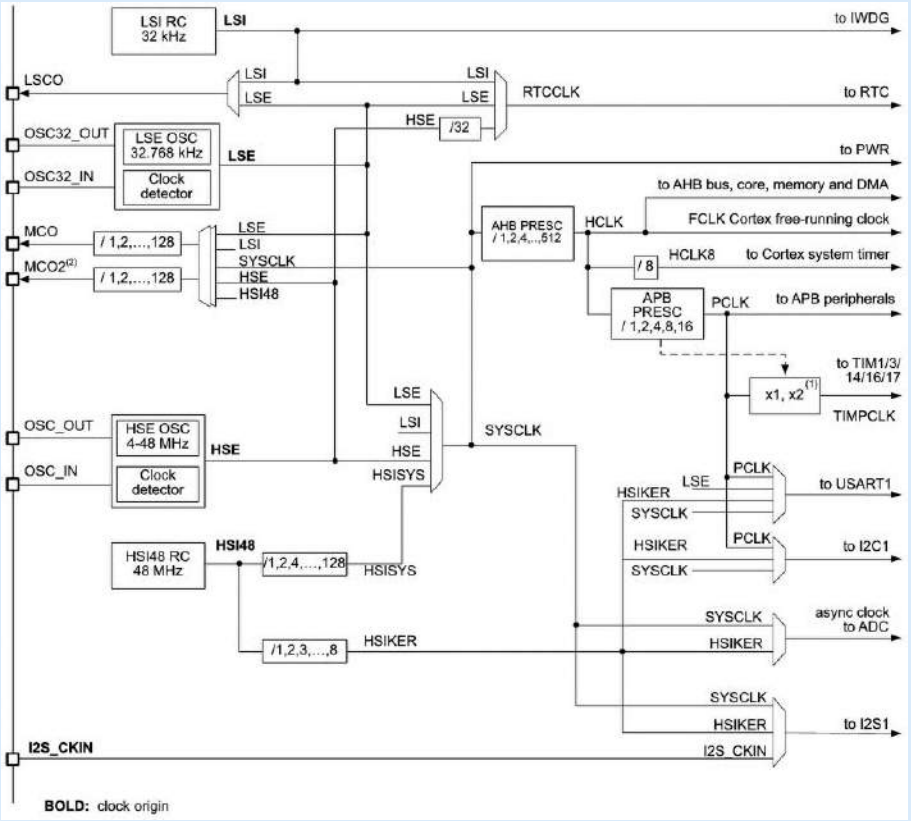
```
RCC_IOPENR_GPIOxEN
```

gdzie x to oznaczenie literowe portu A, B, C, D).

Następnie musimy skonfigurować bit 5 portu A, gdyż do niego właśnie podłączona jest dioda LED. Każda linia portu może pracować w jednej z czterech ról:

- wejście cyfrowe [00],
- wyjście cyfrowe [01],
- funkcja alternatywna [10],
- wejście funkcji analogowej [11].

Wyboru dokonujemy w 32-bitowym rejestrze GPIOx_MODER, w którym – kolejno, począwszy od najmłodszego bitu – znajdują się dwubitowe pola, pozwalające na ustawienie określonego trybu pracy. Dostępne ustawienia tychże pól podano powyżej w nawiasach kwadratowych.



Rysunek 12. Schemat blokowy RCC (drzewo sygnałów zegarowych)

W przypadku rodziny STM32C0 większość linii jest domyślnie ustawiona po resecie jako wejście funkcji analogowej, stąd aby ustawić jakikolwiek inny tryb, musimy najpierw wyzerować obydwa bity. Wykonamy to za pomocą maski, obejmującej całe pole bitowe przypisane do danego wyprowadzenia I/O:

```
GPIOA->MODER &= ~GPIO_MODER_MODE5;
```

Następnie ustawimy młodszy bit w polu odpowiadającym linii 5:

```
GPIOA->MODER |= GPIO_MODER_MODE5_0;
```

W typowej aplikacji wybralibyśmy teraz jeszcze rodzaj wyjścia (rejestr OTYPER), konfigurację wewnętrznych rezystorów podciągających (PUPDR) oraz szybkość linii GPIO (OSPEEDR) – na szczęście, domyślne ustawienia po resecie są odpowiednie w tym przypadku (wyjście typu push-pull, rezystory nieaktywne, szybkość minimalna), stąd możemy dla uproszczenia opisu pominąć na razie znaczenie tych rejestrów.

Aby zamigać diodą, w pętli głównej programu trzeba naprzemiennie włączać i wyłączać wyjście. W przypadku STM32 można to zrobić na dwa sposoby: albo z wykorzystaniem rejestru wyjściowego (ODR), poddawanego działaniu funkcji logicznej & lub | z odpowiednią maską, albo za pośrednictwem specjalnego rejestru, umożliwiającego dostęp w trybie atomowym (BSRR). Skorzystamy z drugiej, wygodniejszej opcji. Zaświecenie diody nastąpi po ustawieniu (**S**et) linii w stan w wysoki:

```
GPIOA->BSRR = GPIO_BSRR_BS5;
```

zaś aby ją zgasić, trzeba zmienić stan na niski (**R**eset):

```
GPIOA->BSRR = GPIO_BSRR_BR5;
```

Pomiędzy obydwiema tymi operacjami trzeba jeszcze wprowadzić pewne arbitralne opóźnienie – odpowiednią funkcję

```
void delayMilisec(uint32_t n)
```

przygotowano dobierając empirycznie zawartość pętli `for()` oraz liczbę znajdujących się w niej wywołań makra asemblera (`__NOP()`) w taki sposób, by w przybliżeniu uzyskać opóźnienie o zadanej za pomocą parametru n liczbie milisekund. Kompletny kod można zobaczyć na **listingu 1**.

```
/*
 * Crash Course STM32C0
 * Elektronika Praktyczna
 *
 * Przyklad 1. Dzien dobry w jezyku MCU, czyli Blinky na STM32C0
 * plik main.c
 *
 */

#include "stm32c031xx.h"

void delayMilisec(uint32_t n);

int main()
{

    /* wlaczenie taktowania bloków GPIO */
    RCC->IOPENR |= RCC_IOPENR_GPIOAEN
                 | RCC_IOPENR_GPIOBEN
                 | RCC_IOPENR_GPIOCEN
                 | RCC_IOPENR_GPIODEN;

    /* wyzerowanie bitow konfig. linii PA5 */
    GPIOA->MODER &= ~GPIO_MODER_MODE5;

    /* ustawienie PA5 jako wyjscie */
    GPIOA->MODER |= GPIO_MODER_MODE5_0;

    while(1)
    {
        /* stan wysoki na PA5 */
        GPIOA->BSRR = GPIO_BSRR_BS5;

        delayMilisec(100);

        /* stan niski na PA5 */
        GPIOA->BSRR = GPIO_BSRR_BR5;

        delayMilisec(100);

    }
}

void delayMilisec(uint32_t n){

    for(uint32_t i = 0; i < 400*n; i++)
    {
        __NOP();
        __NOP();
        __NOP();
        __NOP();
        __NOP();
        __NOP();
        __NOP();
        __NOP();
        __NOP();
        __NOP();
    }

}
```

Listing 1. Miganie diodą LED

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie **https://ulubionykiosk.pl/media**

eprasa.pl 65e3eb482b

Prawidłowo napisany program należy skompilować (*Build* lub klawisz F7), a następnie wgrać do procesora (przycisk *Download* lub klawisz F8).

Przykład 2. Obsługa przycisku

Tym razem dodamy do naszego kodu „turbodoładowanie" – naciskając niebieski przycisk USER, spowodujemy przyspieszenie migania diody. W tym celu musimy ustawić połączoną z przyciskiem linię portu C o numerze 13 w tryb wejścia cyfrowego:

```
GPIOC->MODER &= ~GPIO_MODER_MODE13;
```

Na płytce znajduje się dyskretny rezystor podciągający, stąd nie ma potrzeby włączania wbudowanych „podciągów" w rejestrze PUPDR.

W celu sprawdzenia, czy przycisk jest naciśnięty, odwołamy się do rejestru wejściowego, czyli IDR:

```
if((GPIOC->IDR & GPIO_IDR_ID13) == GPIO_IDR_ID13)
```

Kompletny kod ciała funkcji **`main()`** programu można zobaczyć na **listingu 2**.

```
int main()
{

    uint32_t delay;

    RCC->IOPENR |= RCC_IOPENR_GPIOAEN
                 | RCC_IOPENR_GPIOBEN
                 | RCC_IOPENR_GPIOCEN
                 | RCC_IOPENR_GPIODEN;

    GPIOA->MODER &= ~GPIO_MODER_MODE5;
    GPIOA->MODER |= GPIO_MODER_MODE5_0;

    /* PC13 jako wejscie cyfrowe */
    GPIOC->MODER &= ~GPIO_MODER_MODE13;


    while(1)
    {

        /* sprawdzenie stanu linii wejsciowej w rejestrze GPIO_IDR*/
        if((GPIOC->IDR & GPIO_IDR_ID13) ==  GPIO_IDR_ID13)
            delay = 200;
        else
            delay = 50;

        GPIOA->BSRR = GPIO_BSRR_BS5;

        delayMilisec(delay);

        GPIOA->BSRR = GPIO_BSRR_BR5;

        delayMilisec(delay);

    }
}
```

Listing 2. Funkcja main() z listingu 1 wzbogacona o obsługę przycisku zmieniającego szybkość migania diody

Przykład 3. Konfiguracja bloku RCC. Obsługa wyjścia MCO

Do tej pory nasz mikrokontroler był taktowany sygnałem z wewnętrznego oscylatora RC o częstotliwości nominalnej ok. 48 MHz (podzielonym przez 4 za pomocą preskalera HSIDIV w rejestrze RCC_CR). Docelowo jednak chcemy skorzystać z nieporównanie dokładniejszego rezonatora kwarcowego, warto więc zapoznać się ze sposobem konfiguracji bloku RCC w celu ustawienia odpowiednich częstotliwości taktowania rdzenia i magistral. Napiszemy w tym celu funkcję **`void RCC_configClockSource(void)`**, której ciało zaprezentowano na **listingu 3**.

W rejestrze kontrolnym (CR) włączymy najpierw bit odpowiedzialny za uruchomienie oscylatora HSE współpracującego z rezonatorem kwarcowym.

```
RCC->CR |= RCC_CR_HSEON;
```

Następnie musimy odczekać do momentu prawidłowego rozruchu, wykonując przez ten czas polling bitu RCC_CR_HSERDY:

```
if((RCC->CR & RCC_CR_HSERDY) == RCC_CR_HSERDY)
```

Jeżeli oscylator działa poprawnie, należy skonfigurować preskalery poszczególnych magistral, które w przypadku STM32C0 są dwie: AHB oraz APB. Dla pierwszej z nich ustawimy częstotliwość równą połowie SYSCLK, czyli głównego sygnału zegarowego:

```
RCC->CFGR |= RCC_CFGR_HPRE_3;
```

Ta zostanie znów podzielona przez dwa, co da w efekcie taktowanie 12 MHz dla większości peryferiów mikrokontrolera (ten zabieg ma na celu jedynie zobrazowanie sposobu konfiguracji – rzecz jasna, w realnej aplikacji można uzyskać zarówno częstotliwości wyższe, jak i niższe od wybranych do tego przykładu).

```
RCC->CFGR |= RCC_CFGR_PPRE_2;
```

Ciekawą opcją jest możliwość wyprowadzenia jednego z kilku wewnętrznych sygnałów zegarowych procesora na port GPIO, a dokładniej – linię wyposażoną w funkcję alternatywną MCO1 lub MCO2. W naszym przypadku wybierzemy tę drugą, ustawiając jako źródło główny sygnał taktowania (SYSCLK):

```
RCC->CFGR |= RCC_CFGR_MCO2SEL_0;
```

oraz dzieląc ów sygnał (o docelowej częstotliwości równej 48 MHz) przez 64, co da w efekcie przebieg 750-kilohercowy:

```
RCC->CFGR |= RCC_CFGR_MCO2PRE_2 | RCC_CFGR_MCO2PRE_1;
```

Na koniec możemy nareszcie przełączyć blok RCC naszego procesora z domyślnego oscylatora HSI na generator HSE:

```
RCC->CFGR |= RCC_CFGR_SW_0;
```

W kodzie funkcji **`void RCC_configClockSource(void)`** przewidziano także miejsce na obsługę ewentualnych błędów inicjalizacji HSE lub przełączania SYSCLK na ów sygnał – dla uproszczenia przyjęliśmy, że nasz program wpadnie w którymkolwiek z wymienionych przypadków w pętlę nieskończoną. Rzecz jasna, w rzeczywistej aplikacji takie działanie byłoby niedopuszczalne i należałoby odpowiednio obsłużyć te błędy, powracając np. na HSI oraz informując

```
void RCC_configClockSource(void)
{

    uint16_t i = 0;

    /* wlaczenie oscylatora HSE */
    RCC->CR |= RCC_CR_HSEON;

    while(i < 1000)
    {

        i++;

        /* test flagi gotowosci HSE */
        if((RCC->CR & RCC_CR_HSERDY) == RCC_CR_HSERDY)
            break;

    }

    if(i < 1000)
    {

        /* preskaler HCLK = 2 => HCLK = SYSCLK/2 = 24 MHz */
        RCC->CFGR |= RCC_CFGR_HPRE_3;

        /* preskaler PCLK = 2 => PCLK = HCLK/2 = 12 MHz */
        RCC->CFGR |= RCC_CFGR_PPRE_2;

        /* preskaler MCO2 = 64 => MCO2 = SYSCLK/64 = 750 kHz */
        RCC->CFGR |= RCC_CFGR_MCO2PRE_2 | RCC_CFGR_MCO2PRE_1;

        /* zrodlo sygnalu dla MCO2: SYSCLK */
        RCC->CFGR |= RCC_CFGR_MCO2SEL_0;

        /* zrodlo sygnalu dla SYSCLK: HSE */
        RCC->CFGR |= RCC_CFGR_SW_0;

        /* sprawdzenie czy SYSCLK przelaczyl sie na HSE */
        while(i < 10)
        {

            /* test flagi gotowosci HSE */
            if((RCC->CFGR & RCC_CFGR_SWS_0) == RCC_CFGR_SWS_0)
                break;

            i++;

        }

        /* miejsce na obsluge bledu SWS */
        if(i == 10)
            while(1);

    } else {

        /* miejsce na obsluge bledu uruchomienia HSE */
        while(1);

    }

}
```

Listing 3. Funkcja konfiguracji RCC

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie https://ulubionykiosk.pl/media

eprasa.pl 65e3eb482b

Rysunek 13. Przebieg na wyjściu MCO2

użytkownika lub nadrzędne urządzenie o awarii systemu taktowania. Przedstawiona metoda może jednak bardzo ułatwić debugowanie kodu na początkowych etapach pracy.

W kodzie z przykładu 1 trzeba wykonać jeszcze jedną zmianę – tym razem potrzebujemy bowiem odpowiednio skonfigurować linię PA10, pod którą podepniemy funkcję MCO2. W tym celu ustawiamy „starszy" z dwóch rejestrów, odpowiedzialnych za ustalenie numeru funkcji alternatywnej (AF):

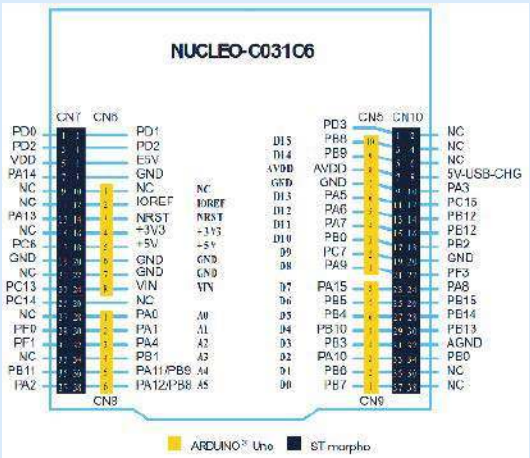
```
GPIOA->AFR[1] |= GPIO_AFRH_AFSEL10_1 | GPIO_AFRH_AFSEL10_0;
```

Jak widać, przyjęta przez producenta konwencja nazewnicza jest w tym przypadku nieco mniej spójna niż w innych peryferiach. Obydwa rejestry AFR występują tutaj bowiem jako… tablica o rozmiarze 2, przy czym młodszy rejestr przechowuje bity konfigurujące linie danego portu o numerach 0...7, zaś starszy obejmuje linie 8...15. W nazwach bitów ma to swoje odzwierciedlenie w postaci litery H lub L (w powyższym przykładzie korzystamy z rejestru starszego). Jak widać, wybraliśmy bity na pozycjach 0 i 1, co daje sumę równą 3 – i tego dokładnie potrzebujemy, gdyż w tabeli AF, dostępnej w nocie katalogowej mikrokontrolera, funkcja MCO2 linii PA10 figuruje właśnie pod numerem AF3.

Wyprowadzając sygnał zegarowy (nawet wstępnie podzielony w preskalerze), warto także zmienić szybkość maksymalną danej linii GPIO – w przypadku procesorów STM32 dostępne są cztery poziomy:

- domyślny (Very Low Speed [00]),
- powolny (Low Speed [01]),
- szybki (High Speed [10]),
- bardzo szybki (Very High Speed [11]).

Taka konfiguracja ma na celu umożliwienie optymalizacji poziomu zakłóceń RFI generowanych przez mikrokontroler podczas przełączania linii GPIO. Każde z ustawień pola bitowego ma swoje



Rysunek 14. Układ wyprowadzeń złączy goldpin na płytce Nucleo-C031C6

odzwierciedlenie w konkretnej, maksymalnej częstotliwości przełączania danego pinu – szczegóły można znaleźć w nocie katalogowej procesora. W naszym przypadku, choć sygnał jest stosunkowo wolny (tylko 750 kHz), ustawimy dla przykładu najwyższą prędkość pracy linii 10 GPIO:

```
GPIOA->OSPEEDR |= GPIO_OSPEEDR_OSPEED10_1 |
GPIO_OSPEEDR_OSPEED10_0;
```

Oscylogram przebiegu zarejestrowanego na wyprowadzeniu PA10 można zobaczyć na **rysunku 13**. Dla ułatwienia pracy z tym oraz kolejnymi przykładami pokażemy w tym miejscu także rozkład wyprowadzeń na listwach szpilkowych płytki Nucelo (**rysunek 14**) – niestety, producent nie zdecydował się na opisanie ich bezpośrednio na PCB, stąd każdorazowo przy podłączaniu pojedynczych przewodów do złączy goldpin trzeba zdać się na… ręczne liczenie pinów. Ot, drobne utrudnienie (niestety niejedyne w przypadku tej rodziny zestawów ewaluacyjnych).

Przykład 4. Nadawanie danych przez UART

Tym razem wskoczymy od razu w bardziej zaawansowane zagadnienia – dla ułatwienia realizacji kolejnych ćwiczeń przygotujemy teraz podstawową obsługę nadawania znaków poprzez interfejs UART. Warto dodać, że do sprawdzenia działania naszego programu nie potrzebujemy żadnego zewnętrznego konwertera USB-UART – zaimplementowany na płytce Nucleo programator/debugger ST-Link w standardzie udostępnia nam bowiem wygodny w użyciu, wirtualny port szeregowy, współpracujący z interfejsem USART2 naszego procesora.

Dla ułatwienia orientacji w plikach źródłowych tym razem wydzielimy funkcje odpowiedzialne za obsługę peryferiów sprzętowych do plików hardware.c/hardware.h, zaś funkcje pomocnicze (opóźniające) – do plików utils.c/utils.h. W środowisku Keil można je dodać w sposób analogiczny do opisanego na początku, przy okazji tworzenia pliku main.c.

Linie portów USART2 (PA2 i PA3) ustawiamy w tryb funkcji AF1:

```
GPIOA->MODER &= ~(GPIO_MODER_MODE2 | GPIO_MODER_MODE3);
GPIOA->MODER |= GPIO_MODER_MODE2_1 | GPIO_MODER_MODE3_1;
GPIOA->AFR[0] |= GPIO_AFRL_AFSEL2_0 | GPIO_AFRL_AFSEL3_0;
```

W funkcji UART_config(void) konfigurujemy kolejno:

- bit zezwolenia na taktowanie USART2:
  `RCC->APBENR1 |= RCC_APBENR1_USART2EN;`
- częstotliwość taktowania (preskaler) w celu uzyskania prędkości transmisji równej 9600 bps:
  `USART2->BRR = 120000/96;`

Następnie musimy jeszcze włączyć nadajnik:

```
USART2->CR1 = USART_CR1_TE;
```

oraz ustawić główny bit uruchamiający interfejs USART2:

```
USART2->CR1 |= USART_CR1_UE;
```

Kod funkcji wysyłającej ciąg znaków (string) za pomocą USART2 można zobaczyć na **listingu 4**. W celu zachowania zgodności z wymogami procesora, przed nadaniem znaków musimy sprawdzić, czy bufor nadawczy jest pusty, odczytując w tym celu bit **USART_ISR_TXE_TXFNF**, znajdujący się w rejestrze statusu (ISR):

```
if ((USART2->ISR & USART_ISR_TXE_TXFNF) == USART_ISR_TXE_
TXFNF) [1]
```

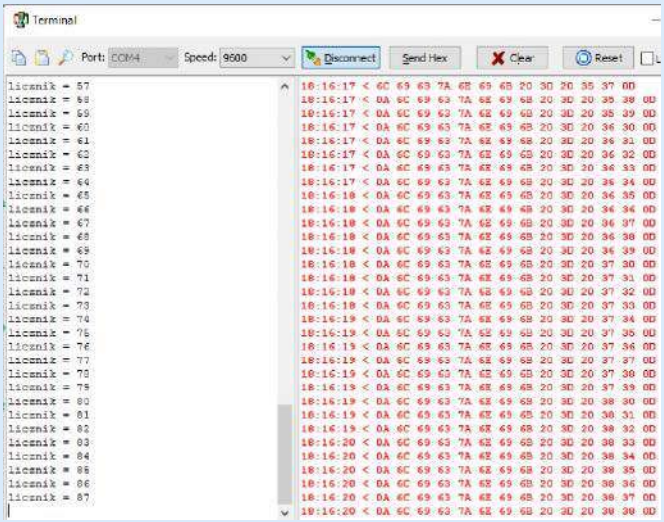
Jeżeli otrzymamy zielone światło, możemy w pętli **while()** nadać wszystkie znaki, aż do spełnienia warunku wyjścia z pętli (ostatni znak stringu w języku C ma wartość 0x00). Każdorazowo wysłanie bajtu odbywa się poprzez zapis znaku do rejestru nadawczego (TDR), przy czym wskaźnik jest od razu postinkrementowany:

```
USART2->TDR = *(str++);
```

Po wysłaniu odczekujemy na zwolnienie bufora, ponownie odpytując bit **TXE_TXFNF** rejestru ISR. Nieco inny test znajduje się natomiast na końcu opisywanej funkcji – tutaj sprawdzamy bowiem bit TC rejestru ISR, informujący nie tyle o zwolnieniu bufora, co o fizycznym

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie https://ulubionykiosk.pl/media

eprasa.pl 65e3eb482b



Rysunek 15. Okno terminalu portu szeregowego z widocznymi danymi, wysyłanymi przez procesor STM32C0 za pomocą interfejsu UART1

zakończeniu transmisji. Warto pamiętać o tej istotnej różnicy, gdyż ma ona duże znaczenie m.in. podczas obsługi UART-a za pomocą kontrolera DMA.

Tym razem ewentualne błędy wykryte podczas działania funkcji są sygnalizowane zaświeceniem diody LED oraz (podobnie jak poprzednio) wejściem w pętlę nieskończoną. Ten rodzaj obsługi błędu, niedopuszczalny w jakimkolwiek rzeczywistym zastosowaniu, może jednak oddać nieocenione zasługi podczas początkowego debugowania kodu.

W głównej pętli **while()** programu wywołujemy funkcję **UART_print_chars()** co około 100 ms, wysyłając na UART napis „licznik=xx", gdzie xx to kolejny numer z zakresu 0...99 (**rysunek 15**). Sama obsługa wysyłki znaków jest bardzo prosta i intuicyjna, więc nie będziemy jej szerzej opisywać – oryginalne pliki źródłowe można znaleźć w materiałach dodatkowych do niniejszego kursu, znajdujących się na serwerze ep.com.pl w dziale Archiwum.

Przykład 5

Zostawmy na chwilę obsługę modułu UART i zajmijmy się zadaniem prostszym, ale o fundamentalnym znaczeniu dla aplikacji mikrokontrolerów. Tym razem skonfigurujemy timer sprzętowy TIM1 i użyjemy go do płynnego sterowania jasnością diody LED, oczywiście za pomocą sygnału PWM.

Wyjście PWM timera jest dostępne jako funkcja AF5 linii PA5. Stosowną konfigurację tego wyprowadzenia można znaleźć poniżej.

```
GPIOA->MODER &= ~GPIO_MODER_MODE5;
GPIOA->MODER |= GPIO_MODER_MODE5_1;
GPIOA->AFR[0] |= GPIO_AFRL_AFSEL5_2 |
GPIO_AFRL_AFSEL5_0;
```

Ciało funkcji **void PWM_config(void)**, odpowiedzialnej za konfigurację wyjścia kanału 1 timera TIM1, można znaleźć na **listingu 5**. Obsługa timera nie różni się w swoich podstawowych założeniach od tego samego zadania, zrealizowanego np. za pomocą mikrokontrolera AVR. Uwagę warto zwrócić jednak na kilka aspektów.

Po pierwsze, w linii konfigurującej zawartość rejestru PSC (preskaler) znalazło się mnożenie przez 2:

```
TIM1->PSC |= 2 * PCLK_FREQ/( PWM_FREQ *  PWM_RES )  - 1;
```

W żadnym wypadku nie jest to błąd – w dokumentacji procesora znajdujemy bowiem informację, że sygnał zegarowy dostarczany z RCC do timerów jest 2-krotnie szybszy dla wszystkich ustawień preskalera magistrali APB > 1. W naszym przypadku preskaler APB został ustawiony na wartość 2, zatem aby uzyskać docelową częstotliwość przebiegu równą 1 kHz, należy zastosować wspomniany mnożnik. Makro **PWM_FREQ** to właśnie częstotliwość przebiegu (wyrażona w hercach), **PCLK_FREQ** to częstotliwość szyny APB (12 MHz, także zapisana w hercach), zaś PWM_RES to rozdzielczość, arbitralnie ustawiona na wartość równą 100. Taki zabieg pozwala na proste przestrajanie współczynnika wypełnienia w zakresie od 0 do 100% z podziałką co 1%, bez konieczności dokonywania dodatkowych obliczeń. Wartość **PWM_RES** (pomniejszona o 1) trafia zatem do rejestru automatycznego przeładowywania licznika (ARR). Linia:

```
TIM1->CCMR1 |= TIM_CCMR1_OC1M_2 | TIM_CCMR1_OC1M_1;
```

odpowiada za wybór jednego z wielu udostępnionych przez timer trybów pracy – w naszym przypadku stan aktywny na wyjściu (czyli stan wysoki) będzie trwał od momentu wyzerowania licznika (liczymy w górę) do momentu zrównania się stanu licznika z zawartością rejestru CCR1. Ten ostatni posłuży nam do przestrajania współczynnika wypełnienia przebiegu PWM i dla ułatwienia opakujemy go w funkcję **void PWM_set_duty(uint8_t pwm)**.

W celu uruchomienia wyjścia PWM musimy jeszcze ustawić trzy różne bity, odpowiedzialne za – kolejno – włączenie kanału 1, globalne zezwolenie na generowanie sygnałów oraz włączenie timera jako takiego:

```
TIM1->CCER |= TIM_CCER_CC1E;
TIM1->BDTR |= TIM_BDTR_MOE;
TIM1->CR1 |= TIM_CR1_CEN;
```

Ostatnia linia funkcji konfiguracyjnej wymusza wstępne przeładowanie rejestrów timera, co umożliwia prawidłowy rozruch od zera:

```
TIM1->EGR |= TIM_EGR_UG;
```

```
void UART_print_chars(char *str)
{

    uint8_t timeout;

    /* sprawdzenie gotowosci bufora do wysylki */
    if ((USART2->ISR & USART_ISR_TXE_TXFNF) == USART_ISR_TXE_TXFNF){

        /* wysylka znakow az do osiagniecia konca stringu */
        /* warunek konca: zawartosc znaku rowna 0x00 */
        while (*str){

            /* zapis bajtu do rejestru nadawczego */
            /* nastepnie: postinkrementacja wskaznika "w locie" */
            USART2->TDR = *(str++);

            /* timeout ~100 us * 12 = ~1,2 ms */
            timeout = 12;

            /*  oczekiwanie na zwolnienie bufora TDR */
            while (!(USART2->ISR & USART_ISR_TXE_TXFNF)){

                if(timeout > 0){

                    delayMicrosec(100);
                    timeout--;

                }else{

                    /* "brutalny" sposob obslugi bledu */
                    /* wlaczenie LED i wejscie w petle nieskonczona */
                    GPIOA->BSRR = GPIO_BSRR_BS5;
                    while(1);

                }
            }

        }

        /* kolejny timeout ~1,2 ms */
        timeout = 12;

        /* oczekiwanie na ustawienie flagi zakonczenia transmisji */
        while (!(USART2->ISR & USART_ISR_TC)){

            if(timeout > 0){

                delayMicrosec(100);
                timeout--;

            }else{

                /* "brutalny" sposob obslugi bledu */
                GPIOA->BSRR = GPIO_BSRR_BS5;
                while(1);
            }
        }
    }
}
```

Listing 4. Ciało funkcji wysyłającej znaki przez USART2

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie
https://ulubionykiosk.pl/media

eprasa.pl 65e3eb482b

Rysunek 16. Prawidłowy przebieg sygnału PWM i jego statystyki (zakładka po prawej stronie ekranu oscyloskopu)

Przebieg uzyskany za pomocą timera TIM1 można zobaczyć na **rysunku 16**. Oryginalny kod programu zmienia wypełnienie impulsu na przemian w górę (w pełnym zakresie) i w dół – efekt będzie widoczny w postaci płynnego rozjaśniania oraz przygaszania diody LED.

Przykład 6. Obsługa SPI full-duplex. Komunikacja z akcelerometrem

Okablowanie płytki NUCLEO w tym przykładzie:

- LIS2DW12 (VCC) – 3V3
- LIS2DW12 (GND) – GND
- LIS2DW12 (SCL) – PA5
- LIS2DW12 (SDO) –PA6
- LIS2DW12 (SDA) – PA7
- LIS2DW12 (CS) – PB0

W kolejnym przykładzie zajmiemy się zagadnieniem znacznie bardziej złożonym – uruchomimy interfejs SPI w trybie standardowej wymiany danych (full duplex) i skomunikujemy nasz procesor z akcelerometrem LIS2DW12, z którego będziemy pobierać wyniki pomiarów, a następnie przesyłać je w formie ramki ASCII/CSV do terminalu portu szeregowego.

Maksymalna częstotliwość sygnału SCLK akcelerometru wynosi 10 MHz. W naszych zastosowaniach nie potrzebujemy zbyt szybkiego taktowania, dlatego możemy bez problemu obniżyć częstotliwość np. do 1,5 MHz – w tym celu 12-megahercowy zegar PCLK podzielimy przez 8. Za takie ustawienie preskalera SPI odpowiada linia:

```
SPI1->CR1 = SPI_CR1_BR_1;
```

Zgodnie z dokumentacją czujnika trzeba ustawić tryb SPI na wartość 3 (CPOL=1, CPHA=1), przy czym dane mają być przesyłane, począwszy od najstarszego bitu:

```
SPI1->CR1 |= SPI_CR1_CPHA | SPI_CR1_CPOL;
SPI1->CR1 &= ~SPI_CR1_LSBFIRST;
```

W kolejnych liniach wybieramy tryb obsługi linii zezwolenia (CE/NSS) – chcemy programowo sterować tym wyprowadzeniem poprzez normalny port wyjściowy GPIO, dlatego wyłączamy automatyczne przełączanie stanu NSS:

```
SPI1->CR1 |= SPI_CR1_SSM | SPI_CR1_SSI;
```

Następnie konfigurujemy tryb pracy interfejsu (master), rozmiar danych (8 bitów) oraz moment wyzwalania sygnału RXNE (powinien być dobrany odpowiednio do długości słowa danych, którą w mikrokontrolerach STM32 można regulować w szerokim zakresie):

```
SPI1->CR1 |= SPI_CR1_MSTR;
SPI1->CR2 |= SPI_CR2_DS_2 | SPI_CR2_DS_1 | SPI_CR2_DS_0;
SPI1->CR2 |= SPI_CR2_FRXTH;
```

Na koniec włączamy interfejs SPI:

```
SPI1->CR1 |= SPI_CR1_SPE;
```

Funkcja odpowiedzialna za dwukierunkowy przesył danych z wykorzystaniem 4-liniowego interfejsu SPI (CE, SCLK, MOSI, MISO) rozpoczyna się oczywiście od ustawienia wspomnianego wyjścia obsługującego linię CE czujnika:

```
GPIOB->BSRR = GPIO_BSRR_BR0;
```

Następnie w pętli `for()` dokonujemy wysyłki kolejnych bajtów danych, oczekując za każdym razem na zwolnienie bufora nadawczego:

```
while (!(SPI1->SR & SPI_SR_TXE)){};
```

Rzecz jasna, tutaj także powinniśmy zastosować odpowiednią obsługę błędu i timeout programowy, zaś sposób reakcji na zajętość bufora podczas próby nadania znaków należałoby oczywiście dobrać odpowiednio do potrzeb aplikacji (metod może być wiele, np. odczekanie i ponowienie próby, zresetowanie czujnika, sygnał alarmowy do układu nadrzędnego czy nawet awaryjne zatrzymanie urządzenia – zależnie od znaczenia takiego błędu dla funkcjonalności oraz bezpieczeństwa całego systemu).

Należy zwrócić uwagę na sposób uzyskania dostępu do rejestru danych:

```
*(uint8_t *)&(SPI1->DR) = out[i];
```

Zastosowanie jawnego rzutowania i odwołania do rejestru DR przez wskaźnik jest tutaj nieprzypadkowe – niestety (o czym dokumentacja wspomina jedynie dość lakonicznie) zwykłe odwołanie w postaci:

```
SPI1->DR = out[i];
```

byłoby przez kompilator potraktowane jako dostęp 32-bitowy, co doprowadziłoby do nieprawidłowego działania interfejsu.

Po wysłaniu każdego bajtu należy odczekać na zakończenie odbioru danych z linii MISO – w tym celu sprawdzamy stan bitu RXNE, znajdującego się w rejestrze SR. Tym razem zastosujemy ograniczenie liczby wykonań pętli:

```
while(((SPI1->SR & SPI_SR_RXNE) != SPI_SR_RXNE) && (tmt > 0))
```

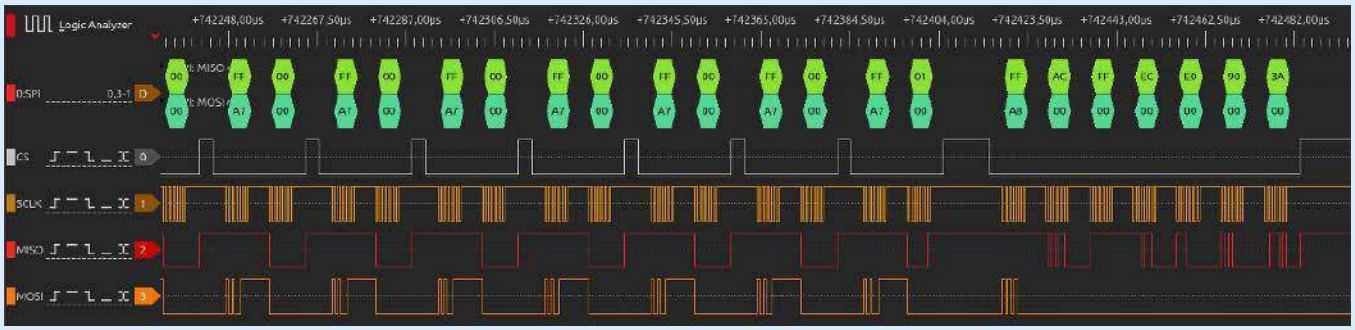
Na koniec odczytujemy odebrany bajt z tego samego rejestru danych (DR) i zapisujemy go do bufora odbiorczego:

```
in[i] = (uint8_t)(SPI1->DR);
```

Do obsługi modułu akcelerometru służy sterownik, którego kod umieszczono w plikach lis2dw12_driver.c/lis2dw12_driver.h. Zawiera on zaledwie trzy proste funkcje:

- `void LIS2DW12_startup(void)`

  Funkcja wykonuje podstawową konfigurację akcelerometru – na początek odczytywany jest rejestr WHO_AM_I, co pozwala (przy



Rysunek 17. Przebiegi na liniach interfejsu SPI w trybie odczytu danych z akcelerometru (przykład 6)

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie https://ulubionykiosk.pl/media

eprasa.pl 65e3eb482b

użyciu analizatora stanów logicznych lub debuggera) sprawdzić poprawność komunikacji z procesorem. Po otrzymaniu komendy sensor powinien zwrócić wartość 0x44.

- `void LIS2DW12_read_XYZ(int16_t *x, int16_t *y, int16_t *z){`

  Funkcja odczytuje zawartość rejestrów wyjściowych 3-osiowej struktury MEMS. Wartości są przesyłane w formacie uzupełnienia do 2, stąd po odebraniu sześciu bajtów należy z nich poskładać 16-bitowe wartości surowych odczytów, które umieszczone zostaną w zmiennych x, y, z za pomocą wskaźników, podanych jako parametry funkcji.
- `uint8_t LIS2DW12_check_DRDY(void){`

  Funkcja sprawdza stan bitu DRDY w rejestrze STATUS akcelerometru – w kodzie funkcji **`main()`** wykorzystujemy ją do pollingu w oczekiwaniu na nowe wyniki pomiaru przyspieszenia.

W ramach artykułu nie będziemy szczegółowo omawiać wszystkich operacji wykonywanych przez poszczególne funkcje sterownika akcelerometru – całość jest szczegółowo opisana za pomocą komentarzy, zatem ponownie zachęcamy naszych Czytelników, by zajrzeli do materiałów dodatkowych powiązanych z kursem. Dla lepszego zobrazowania przebiegu komunikacji zamieszczamy natomiast **rysunek 17**, prezentujący stosowny zrzut ekranu z analizatora stanów logicznych.

Przykład 7. Obsługa przerwań. Bloki EXTI i NVIC

Okablowanie płytki NUCLEO w tym przykładzie:

- LIS2DW12 (VCC) – 3V3
- LIS2DW12 (GND) – GND
- LIS2DW12 (SCL) – PA5
- LIS2DW12 (SDO) –PA6
- LIS2DW12 (SDA) – PA7
- LIS2DW12 (CS) – PB0
- **LIS2DW12 (INT1) – PA10**

Tym razem wzbogacimy nasz przykład o zastosowanie przerwań od linii GPIO, unikając tym samym konieczności ciągłego odpytywania akcelerometru pod kątem dostępności nowych danych pomiarowych. Skorzystamy z jednej z dwóch linii przerwań układu LIS2DW12 (INT1), konfigurując ją uprzednio w taki sposób, by po zakończeniu każdego kolejnego pomiaru na INT1 pojawiał się impuls dodatni (L→H→L), informujący nasz procesor o konieczności dokonania pełnego odczytu danych X, Y, Z.

Konfiguracja linii GPIO do współpracy z blokiem przerwań nie wymaga specjalnych zabiegów – wyprowadzenie PA10 będzie po prostu zwykłym wejściem cyfrowym (bez funkcji alternatywnej). W przypadku inicjalizacji bloku NVIC (kontrolera przerwań) użyjemy wyjątkowo nie definicji rejestrów, ale gotowych, prostych funkcji należących do biblioteki CMSIS. Ciała obydwu funkcji konfigurujących NVIC oraz EXTI zaprezentowano na **listingu 7**. Najpierw ustawimy globalne zezwolenie obsługi przerwań:

```
__enable_irq();
```

Następnie ustawimy priorytet przerwania od linii bloku EXTI – czyli kontrolera przerwań i zdarzeń, obsługującego w naszym procesorze zarówno przerwania od linii GPIO, jak i szereg sygnałów wewnętrznych z różnych peryferiów. Ponieważ korzystamy z linii

```
void PWM_config(void)
{

    /* wlaczenie taktowania dla TIM1 */
    RCC->APBENR2 |= RCC_APBENR2_TIM1EN;

    /* preskaler = 239 */
    TIM1->PSC |= 2 * PCLK_FREQ/( PWM_FREQ *  PWM_RES )  - 1;

    /* ustawienie wartosci automatycznego przeladowania licznika */
    TIM1->ARR = PWM_RES - 1;

    /* startujemy z wypelnieniem = 0 */
    TIM1->CCR1 = 0;

    /* PWM tryb 1 - wyjscie aktywne jezeli licznik < CCR1 */
    TIM1->CCMR1 |= TIM_CCMR1_OC1M_2 | TIM_CCMR1_OC1M_1;

    /* wlaczenie funkcji pre-ladowania rejestru CCR1 */
    TIM1->CCMR1 |= TIM_CCMR1_OC1PE;

    /* wlaczenie kanalu 1 */
    TIM1->CCER |= TIM_CCER_CC1E;

    /* Main Output Enable - globalne zezwolenie na generowanie sygnalow PWM */
    TIM1->BDTR |= TIM_BDTR_MOE;

    /* wlaczenie timera */
    TIM1->CR1 |= TIM_CR1_CEN;

    /* wymuszenie wstepnego przeladowania rejestrow timera */
    TIM1->EGR |= TIM_EGR_UG;

}

void PWM_set_duty(uint8_t pwm)
{
    /* zmiana rejestru porownywanego z wartoscia licznika */
    TIM1->CCR1 = pwm;
}
```

Listing 5. Funkcje do obsługi timera w trybie generatora PWM

```
void SPI_config()
{

    /* wlaczenie taktowania SPI1 */
    RCC->APBENR2 |= RCC_APBENR2_SPI1EN;

    /* baudrate: ~1,5 MHz */
    SPI1->CR1 = SPI_CR1_BR_1;

    /* Tryb SPI: 3 (1,1) */
    SPI1->CR1 |= SPI_CR1_CPHA | SPI_CR1_CPOL;

    /* upewniamy sie, ze dane beda wysylane od MSB do LSB */
    SPI1->CR1 &= ~SPI_CR1_LSBFIRST;

    /* programowa obsluga linii CE (NSS) */
    SPI1->CR1 |= SPI_CR1_SSM | SPI_CR1_SSI;

    /* SPI w trybie master */
    SPI1->CR1 |= SPI_CR1_MSTR;

    /* transmisja slow 8-bitowych */
    SPI1->CR2 |= SPI_CR2_DS_2 | SPI_CR2_DS_1 | SPI_CR2_DS_0;

    /* ustawienie momentu wyzwalania zdarzenia RXNE (8 bitow)*/
    SPI1->CR2 |= SPI_CR2_FRXTH;

    /* wlaczenie bloku SPI */
    SPI1->CR1 |= SPI_CR1_SPE;
}

void SPI_duplex_transaction(uint8_t *out, uint8_t *in, uint8_t len)
{

    /* CS = 0 */
    GPIOB->BSRR = GPIO_BSRR_BR0;

    uint16_t tmt = 500;

    for(uint8_t i = 0; i < len; i++)
    {

        /* oczekiwanie na pusty bufor nadawczy */
        while (!(SPI1->SR & SPI_SR_TXE))
        {};

        /* zapis bajtu do wysylki */
        *(uint8_t *)&(SPI1->DR) = out[i];

        /* oczekiwanie na zakonczenie odbioru */
        while(((SPI1->SR & SPI_SR_RXNE) != SPI_SR_RXNE) && (tmt > 0)){
            tmt--;
        }

        /* odczyt odebranego bajtu z bufora */
        in[i] = (uint8_t)(SPI1->DR);

    }

    /* CS = 1 */
    GPIOB->BSRR = GPIO_BSRR_BS0;

}
```

Listing 6. Funkcje przeznaczone o konfiguracji i wysyłania danych przez SPI

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie https://ulubionykiosk.pl/media

eprasa.pl 65e3eb482b

PA10, do jej obsługi musimy zastosować przerwanie wspólne dla wyprowadzeń o numerach od 4 do 15:

```
NVIC_SetPriority(EXTI4_15_IRQn, 1);
```

Na koniec „odmaskowujemy” przerwanie, umożliwiając jego wywołanie po wykryciu aktywnego zbocza na linii PA10:

```
NVIC_EnableIRQ(EXTI4_15_IRQn);
```

Funkcja konfiguracji samego bloku EXTI także nie jest zbyt skomplikowana i składa się zaledwie z trzech linii kodu. W pierwszej z nich zezwalamy na aktywację zdarzenia od linii 10 portu (na razie jeszcze bez wskazania, którego portu dotyczy):

```
EXTI->IMR1 |= EXTI_IMR1_IM10;
```

Następnie wybieramy rodzaj zbocza aktywnego (rosnące), ustawiając bit w rejestrze RTSR1 (gdyby konieczne było ustawienie aktywnego zbocza niskiego, wykonalibyśmy analogiczną operację na siostrzanym rejestrze FTSR1):

```
EXTI->RTSR1 = EXTI_RTSR1_RT10;
```

Na koniec musimy jeszcze wskazać kontrolerowi EXTI, którego portu ma dotyczyć obsługa przerwania. Do tego celu służą obszerne pola bitowe (o długości aż 8 bitów), umieszczone w czterech rejestrach EXTICR[0]... EXTICR[3]. Konwencja nazewnicza oraz sposób odwołania do rejestru jako komórki tablicy jest tutaj analogiczny do tego, z którym mieliśmy już do czynienia przy okazji omawiania rejestru AFR bloku GPIO. Linie portów o numerach 0...3 są obsługiwane przez EXTICR[0], linie 4...7 – EXTICR[1], 8...11 – przez EXTICR[2], zaś 12...15 – przez EXTICR[3]. Linia 10 „wypada” więc w zakresie EXTICR[2], a wybór portu A wymaga po prostu wyzerowania wszystkich ośmiu bitów w tej części rejestru – i tak też czynimy w poniższej linii kodu:

```
EXTI->EXTICR[2] &=
~EXTI_EXTICR3_EXTI10;
```

W celu obsługi przerwania skorzystamy z handlera o ściśle określonej nazwie, zdefiniowanej w pliku startup_stm32c031xx.s (patrz **listing 8**). Na **listingu 9** można zobaczyć ciało funkcji obsługującej przerwanie EXTI4_15IRQn – jak widać, jest ona bardzo prosta. Całość rozpoczyna się sprawdzeniem, czy obsługiwane przerwanie faktycznie pochodzi od linii o interesującym nas numerze:

```
if(EXTI->RPR1 & EXTI_RPR1_RPIF10)
```

Jeżeli tak, musimy od razu zresetować flagę przerwania poprzez... ustawienie tego samego bitu:

```
EXTI->RPR1 |= EXTI_RPR1_RPIF10;
```

Na koniec wpisujemy nową wartość (1) do zmiennej globalnej *drdy_flag*, zadeklarowanej w pliku *main.c*. Teraz wystarczy już tylko proste sprawdzenie zawartości tej zmiennej w pętli głównej programu – jeżeli będzie ona ustawiona, program przeprowadzi odczyt danych z akcelerometru, resetując uprzednio flagę *drdy_flag*

```
void NVIC_Config(void){

    /* wywolanie funkcji CMSIS - globalne zezwolenie przerwan */
    __enable_irq();

    /* ustawienie priorytetu przerwania EXTI4...15 */
    NVIC_SetPriority(EXTI4_15_IRQn, 1);

    /* "odmaskowanie" przerwania EXTI4...15 */
    NVIC_EnableIRQ(EXTI4_15_IRQn);

}

void EXTI_config(void){

    /* "odmaskowanie" przerwania od EXTI */
    EXTI->IMR1 |= EXTI_IMR1_IM10;

    /* przewanie generowane na zboczu narastajacym */
    EXTI->RTSR1 = EXTI_RTSR1_RT10;

    /* wyzerowanie bitow EXTICR => podpiecie linii PA10 */
    EXTI->EXTICR[2] &= ~EXTI_EXTICR3_EXTI10;

}
```

Listing 7. Konfiguracja kontrolera przerwań NVIC i bloku EXTI

```
__Vectors       DCD     __initial_sp                    ; Top of Stack
                DCD     Reset_Handler                   ; Reset Handler
                DCD     NMI_Handler                     ; NMI Handler
                DCD     HardFault_Handler               ; Hard Fault Handler
                DCD     0                               ; Reserved
                DCD     0                               ; Reserved
                DCD     0                               ; Reserved
                DCD     0                               ; Reserved
                DCD     0                               ; Reserved
                DCD     0                               ; Reserved
                DCD     0                               ; Reserved
                DCD     SVC_Handler                     ; SVCall Handler
                DCD     0                               ; Reserved
                DCD     0                               ; Reserved
                DCD     PendSV_Handler                  ; PendSV Handler
                DCD     SysTick_Handler                 ; SysTick Handler

                ; External Interrupts
                DCD     WWDG_IRQHandler                 ; Window Watchdog
                DCD     0                               ; Reserved
                DCD     RTC_IRQHandler                  ; RTC through EXTI Line
                DCD     FLASH_IRQHandler                ; FLASH
                DCD     RCC_IRQHandler                  ; RCC
                DCD     EXTI0_1_IRQHandler              ; EXTI Line 0 and 1
                DCD     EXTI2_3_IRQHandler              ; EXTI Line 2 and 3
                DCD     EXTI4_15_IRQHandler             ; EXTI Line 4 to 15
                DCD     0                               ; Reserved
                DCD     DMA1_Channel1_IRQHandler        ; DMA1 Channel 1
                DCD     DMA1_Channel2_3_IRQHandler      ; DMA1 Channel 2 and Channel 3
                DCD     DMAMUX1_IRQHandler              ; DMAMUX
                DCD     ADC1_IRQHandler                 ; ADC1
                DCD     TIM1_BRK_UP_TRG_COM_IRQHandler  ; TIM1 Break, Update, Trigger and Commutation
                DCD     TIM1_CC_IRQHandler              ; TIM1 Capture Compare
                DCD     0                               ; Reserved
                DCD     TIM3_IRQHandler                 ; TIM3
                DCD     0                               ; Reserved
                DCD     0                               ; Reserved
                DCD     TIM14_IRQHandler                ; TIM14
                DCD     0                               ; Reserved
                DCD     TIM16_IRQHandler                ; TIM16
                DCD     TIM17_IRQHandler                ; TIM17
                DCD     I2C1_IRQHandler                 ; I2C1
                DCD     0                               ; Reserved
                DCD     SPI1_IRQHandler                 ; SPI1
                DCD     0                               ; Reserved
                DCD     USART1_IRQHandler               ; USART1
                DCD     USART2_IRQHandler               ; USART2
                DCD     0                               ; Reserved
                DCD     0                               ; Reserved
                DCD     0                               ; Reserved

__Vectors_End
```

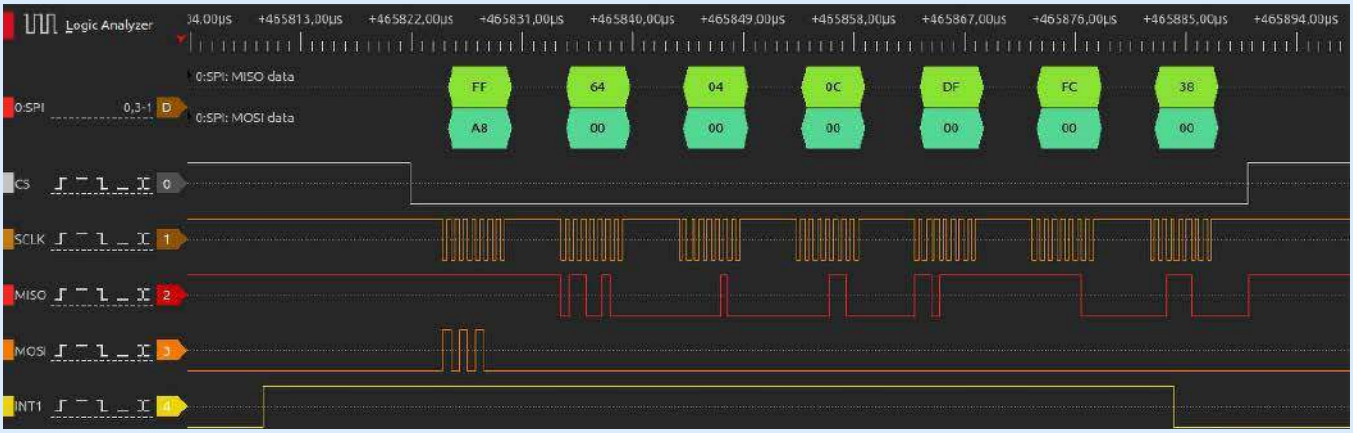
Listing 8. Fragment pliku startup_stm32c031xx.s, prezentujący tablicę wektorów przerwań z domyślnymi nazwami poszczególnych procedur obsługi

```
void EXTI4_15_IRQHandler(void)
{

    extern volatile uint8_t drdy_flag;

    /* sprawdzenie czy zrodlem przerwania byla linia PA10 */
    if(EXTI->RPR1 & EXTI_RPR1_RPIF10){

        /* resetowanie flagi przerwania */
        EXTI->RPR1 |= EXTI_RPR1_RPIF10;

        /* ustawienie flagi programowej, ktora zostanie odczytana w main() */
        drdy_flag = 1;

    }
}
```

Listing 9. Ciało procedury obsługi przerwania od EXTI4...15

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie https://ulubionykiosk.pl/media

eprasa.pl 65e3eb482b



Rysunek 18. Przebiegi na liniach interfejsu SPI w trybie odczytu danych z akcelerometru, tym razem z obsługą przerwania od linii INT1 czujnika (przykład 7)

do wartości spoczynkowej równej 0. Na **rysunku 18** można zobaczyć przebiegi, zarejestrowane za pomocą analizatora stanów logicznych – tym razem z dodatkowym, piątym kanałem, podłączonym do linii PA10.

Przykład 8. Obsługa I²C

Okablowanie płytki NUCLEO w tym przykładzie:

- LIS2DW12 (VCC) – 3V3
- LIS2DW12 (GND) – GND
- **LIS2DW12 (SCL) – PB8**
- **LIS2DW12 (SDA) – PB9**
- LIS2DW12 (INT1) – PA10
- **LIS2DW12 (SDO) – GND (ustawienie adresu I²C)**
- **LIS2DW12 (CS) – 3V3 (wyłączenie SPI, zezwolenie na tryb I²C)**

W ostatnim ćwiczeniu niniejszego kursu zaprezentujemy program o działaniu analogicznym do zaprezentowanego w poprzednim przykładzie, tym razem jednak skorzystamy z interfejsu I²C. W opisie pominiemy już konfigurację linii interfejsu SCL/SDA, przejdziemy bowiem od razu do właściwej inicjalizacji bloku I²C. Teraz, oprócz włączenia taktowania w rejestrze APBENR1, musimy dodatkowo wybrać źródło głównego sygnału zegarowego dla I²C – dokonujemy tego, ustawiając bit I2C1SEL_0 w rejestrze CCIPR bloku RCC:

```
RCC->CCIPR |= RCC_CCIPR_I2C1SEL_0;
```

Nieprzypadkowo korzystamy właśnie z tego źródła – mikrokontrolery STM32 mają dość specyficzny, silnie rozbudowany zestaw opcji służących do konfiguracji najdrobniejszych nawet aspektów timingu sygnałów na liniach interfejsu I²C. Ręczne ustawienie tych parametrów może nastręczać pewnych trudności, dlatego producent opracował trzy metody uproszczenia pracy programistów:

1. W dokumencie Reference Manual dostępne są tabele, prezentujące przykładowe ustawienia poszczególnych pól bitowych rejestru TIMINGR przy różnych prędkościach transmisji i różnych częstotliwościach taktowania.
2. Dla mikrokontrolerów STM32F dostępny jest arkusz kalkulacyjny, umożliwiający zautomatyzowane wyliczenie potrzebnych nastaw po podaniu szeregu parametrów wejściowych.
3. Narzędzie do graficznej konfiguracji peryferiów STM32 (STM32CubeMX) zawiera użyteczny kalkulator, pozwalający na wyklikanie odpowiednich wartości za pomocą intuicyjnego interfejsu GUI.

```
void I2C_config(uint8_t addr){

    /* wybieramy SYSCLK = 48 MHz jako zrodlo sygnalu taktowania I2C1 */
    RCC->CCIPR |= RCC_CCIPR_I2C1SEL_0;

    /* wlaczenie taktowania interfejsu od strony szyny systemowej */
    RCC->APBENR1 |= RCC_APBENR1_I2C1EN;

    /* konfiguracja timingu dla f_SCLK = 100 kHz @ I2CCLK = 48 MHz */
    I2C1->TIMINGR = (uint32_t)(0x0B << I2C_TIMINGR_PRESC_Pos) |
                    (uint32_t)(0x13 << I2C_TIMINGR_SCLL_Pos) |
                    (uint32_t)(0x0F << I2C_TIMINGR_SCLH_Pos) |
                    (uint32_t)(0x02 << I2C_TIMINGR_SDADEL_Pos) |
                    (uint32_t)(0x04 << I2C_TIMINGR_SCLDEL_Pos);

    /* wlaczenie peryferium i zezwolenie na przerwanie od RX */
    I2C1->CR1 = I2C_CR1_PE | I2C_CR1_RXIE;

    /* wlaczenie opcji autmatycznego konczenia transmisji */
    /* ustawienie adresu slave'a */
    I2C1->CR2 = I2C_CR2_AUTOEND | (addr << I2C_CR2_SADD_Pos);

}
```

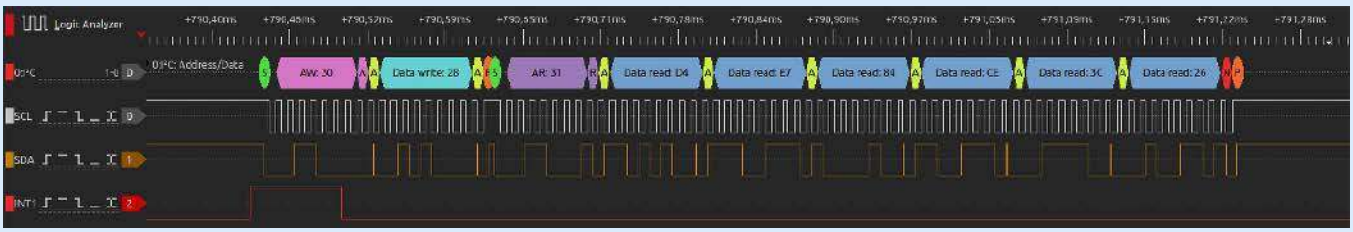
Listing 10. Funkcja inicjalizująca interfejs I2C

My wybierzemy opcję pierwszą – we wspomnianym dokumencie są dostępne dane dla taktowania 48 MHz i docelowej częstotliwości SCL równej 100 kHz – takie też wartości wpisujemy do naszego rejestru:

```
I2C1->TIMINGR = (uint32_t)(0x0B << I2C_TIMINGR_PRESC_Pos) |
                (uint32_t)(0x13 << I2C_TIMINGR_SCLL_Pos) |
                (uint32_t)(0x0F << I2C_TIMINGR_SCLH_Pos) |
                (uint32_t)(0x02 << I2C_TIMINGR_SDADEL_Pos) |
                (uint32_t)(0x04 << I2C_TIMINGR_SCLDEL_Pos);
```

Następnie włączamy blok I²C, zezwalamy na obsługę przerwań odbiorczych oraz uruchamiamy funkcję automatycznego zakończenia transmisji. W tym miejscu wpiszemy także adres slave (na szynie I²C znajduje się tylko jeden czujnik, więc jego adres możemy przypisać raz, w czasie inicjalizacji).

```
I2C1->CR1 = I2C_CR1_PE | I2C_CR1_RXIE;
I2C1->CR2 = I2C_CR2_AUTOEND | (addr << I2C_CR2_SADD_Pos);
```



Rysunek 19. Przebiegi na liniach interfejsu I²C w trybie odczytu danych z akcelerometru z obsługą przerwania od linii INT1 czujnika (przykład 8)

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie **https://ulubionykiosk.pl/media**

eprasa.pl 65e3eb482b

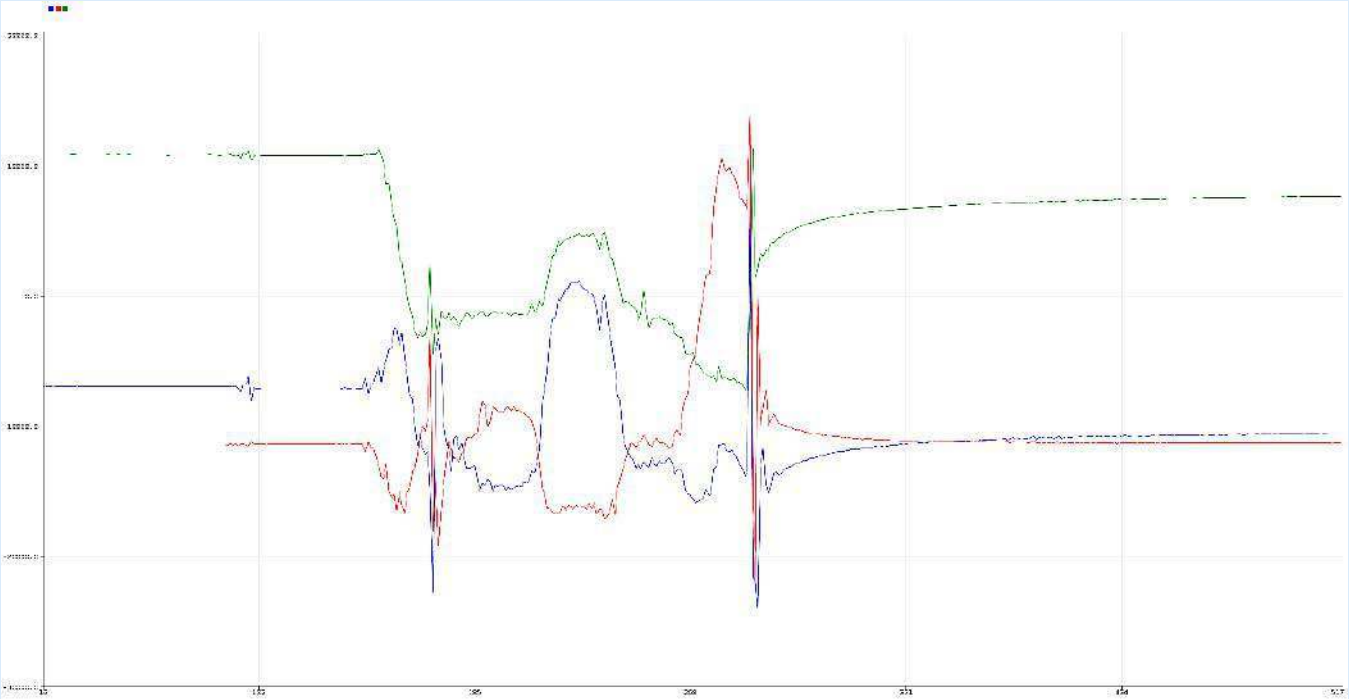
```
void I2C_write_bytes(uint8_t *ptr, uint8_t n){

    uint16_t tout = 0;

    /* sprawdzenie warunku rozpoczecia kolejnej transmisji */
    while((I2C1->CR2 & I2C_CR2_START) == I2C_CR2_START){

        if(tout < 1000){
            delayMicrosec(1);
        }else{
            break;
        }
        tout++;

    }

    /* ponowne sprawdzenie bitu START */
    if((I2C1->CR2 & I2C_CR2_START) != I2C_CR2_START){

        /* zerowanie bitu R/W (bedziemy nadawac dane) */
        I2C1->CR2 &= ~I2C_CR2_RD_WRN;

        /* ustawienie automatycznego zakonczenia transmisji */
        I2C1->CR2 |= I2C_CR2_AUTOEND;

        /* ustawienie dlugosci transmisji */
        I2C1->CR2 &= ~I2C_CR2_NBYTES;
        I2C1->CR2 |= (n << 16);

        /* sprawdzenie czy bufor nadawczy jest pusty */
        if((I2C1->ISR & I2C_ISR_TXE) == I2C_ISR_TXE){

            /* zapis pierwszego bajtu do rejestru wyjsciowego */
            I2C1->TXDR = ptr[0];

            /* nadanie warunku startu */
            I2C1->CR2 |= I2C_CR2_START;

        }

        /* wysylka pozostalych bajtow */
        for(uint8_t i= 1; i< n; i++){

            tout = 0;

            /* oczekiwanie na pusty bufor nadawczy */
            while((I2C1->ISR & I2C_ISR_TXE) != I2C_ISR_TXE){

                if(tout < 1000){

                    delayMicrosec(1);

                }else{

                    break;
                }

                tout++;

            }

            /* jezeli bufor pusty */
            if((I2C1->ISR & I2C_ISR_TXE) == I2C_ISR_TXE){

                /* nadawanie kolejnych bajtow */
                I2C1->TXDR = ptr[i];

                tout = 0;

                /* oczekiwanie na pusty bufor nadawczy */
                while(!(I2C1->ISR & I2C_ISR_TXE)){

                    if(tout < 1000){

                        delayMicrosec(1);

                    }else{

                        break;
                    }

                    tout++;
                }

            }else{
                /* miejsce na obsluge bledu zajetosci bufora */
            }
        }
    }else{
        /* miejsce na obsluge bledu bitu START */
    }
}
```

Listing 11. Funkcja zapisu danych przez I2C

Funkcje przeznaczone do zapisu i odczytu danych poprzez I²C są dość obszerne, choć w istocie stosunkowo łatwe do przeanalizowania. Tym razem analizę poszczególnych zapisów pozostawiamy więc naszym Czytelnikom jako swego rodzaju zadanie domowe – pomocą będą komentarze, opisujące szczegółowo działanie programu. Ciała funkcji inicjalizacji oraz zapisu i odczytu bajtów można zobaczyć na **listingach 10...12**. **Rysunek 19** pokazuje zrzut ekranu z analizatora stanów logicznych, zaś **rysunek 20** – przykładowe odczyty z akcelerometru, zobrazowane za pomocą narzędzia Serial Plotter, dostępnego w środowisku Arduino IDE.

Podsumowanie

W niniejszym artykule przekazaliśmy naszym Czytelnikom solidne podstawy programowania mikrokontrolerów STM32C0 za pomocą rejestrów i zaledwie trzech funkcji biblioteki CMSIS. Mamy nadzieję, że zaprezentowany materiał nie tylko zachęci do samodzielnego eksplorowania dokumentacji procesorów STM32, ale także zmotywuje do głębszego poznania tych interesujących procesorów. W pojedynczym artykule nie sposób rzecz jasna opisać wszystkich bloków peryferyjnych – na pokładzie najnowszych układów znajdują się wszak peryferia takie, jak kontroler DMA, przetwornik ADC, układy watchdogów,



Rysunek 20. Przykładowy zapis danych z akcelerometru w formie graficznej

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie **https://ulubionykiosk.pl/media**

eprasa.pl 65e3eb482b

```
void I2C_read_bytes(uint8_t reg, uint8_t *ptr, uint8_t n){

    uint16_t tout = 0;

    /* sprawdzenie warunku rozpoczecia kolejnej transmisji */
    while((I2C1->CR2 & I2C_CR2_START) == I2C_CR2_START){

        if(tout < 1000){

            delayMicrosec(1);

        }else{

            break;

        }
        tout++;

    }

    /* ponowne sprawdzenie bitu START */
    if((I2C1->CR2 & I2C_CR2_START) != I2C_CR2_START){

        I2C1->CR2 &= ~I2C_CR2_RD_WRN;

        I2C1->CR2 &= ~I2C_CR2_AUTOEND;
        I2C1->CR2 &= ~I2C_CR2_NBYTES;
        I2C1->CR2 |= (1 << 16);

        if((I2C1->ISR & I2C_ISR_TXE) == I2C_ISR_TXE){

            I2C1->TXDR = reg;
            I2C1->CR2 |= I2C_CR2_START;
        }

        tout = 0;

        while((I2C1->CR2 & I2C_CR2_START) == I2C_CR2_START){

            if(tout < 1000){

                delayMicrosec(1);

            }else{

                break;

            }

            tout++;
        }

        if((I2C1->CR2 & I2C_CR2_START) != I2C_CR2_START){

            I2C1->CR2 |= I2C_CR2_RD_WRN | I2C_CR2_AUTOEND;
            I2C1->CR2 &= ~I2C_CR2_NBYTES;
            I2C1->CR2 |= (n << 16);

            /* powtorzony start */
            I2C1->CR2 |= I2C_CR2_START;

            for(uint8_t i = 0; i < n; i++){

                tout = 0;

                /* oczekiwanie na dane w buforze odbiorczym */
                while((I2C1->ISR & I2C_ISR_RXNE) != I2C_ISR_RXNE){

                    if(tout < 1000){
                        delayMicrosec(1);
                    }else{
                        break;
                    }

                    tout++;

                }

                /* ponowne sprawdzenie bufora odbiorczego */
                if((I2C1->ISR & I2C_ISR_RXNE) == I2C_ISR_RXNE){

                    /* odczyt bajtu z bufora */
                    ptr[i] = I2C1->RXDR;

                }else{
                    /* przykladowa obsluga przypadku braku */
                    /* danych w buforze */
                    ptr[i] = 0;

                }
            }
        }else{
            /* miejsce na obsluge bledu zajetosci bufora */
        }
    }else{
        /* miejsce na obsluge bledu zajetosci bufora */
    }
}
```

Listing 12. Funkcja odczytu danych przez I2C

timer systemowy, zegar RTC, akcelerator obliczeń CRC i kilka innych. Mało tego – każdy z nich, podobnie jak opisane w kursie bloki GPIO, RCC, UART, SPI, I²C oraz timery – ma jeszcze szereg funkcji, o których nie zdołaliśmy nawet wspomnieć na łamach naszego opracowania. Zachęcamy zatem do samodzielnego kontynuowania przygody z niskopoziomowym programowaniem układów ARM.

Mamy także nadzieję, że formuła kursu ekspresowego (crash course) przypadnie naszym Czytelnikom do gustu. A może nasi Czytelnicy sami zaproponują tematy kolejnych „szkoleń w pigułce”? Jeśli tak, serdecznie zapraszamy do kontaktu z Redakcją poprzez e-mail bądź chat na facebookowym profilu „Elektroniki Praktycznej”.

inż. Przemysław Musz, EP

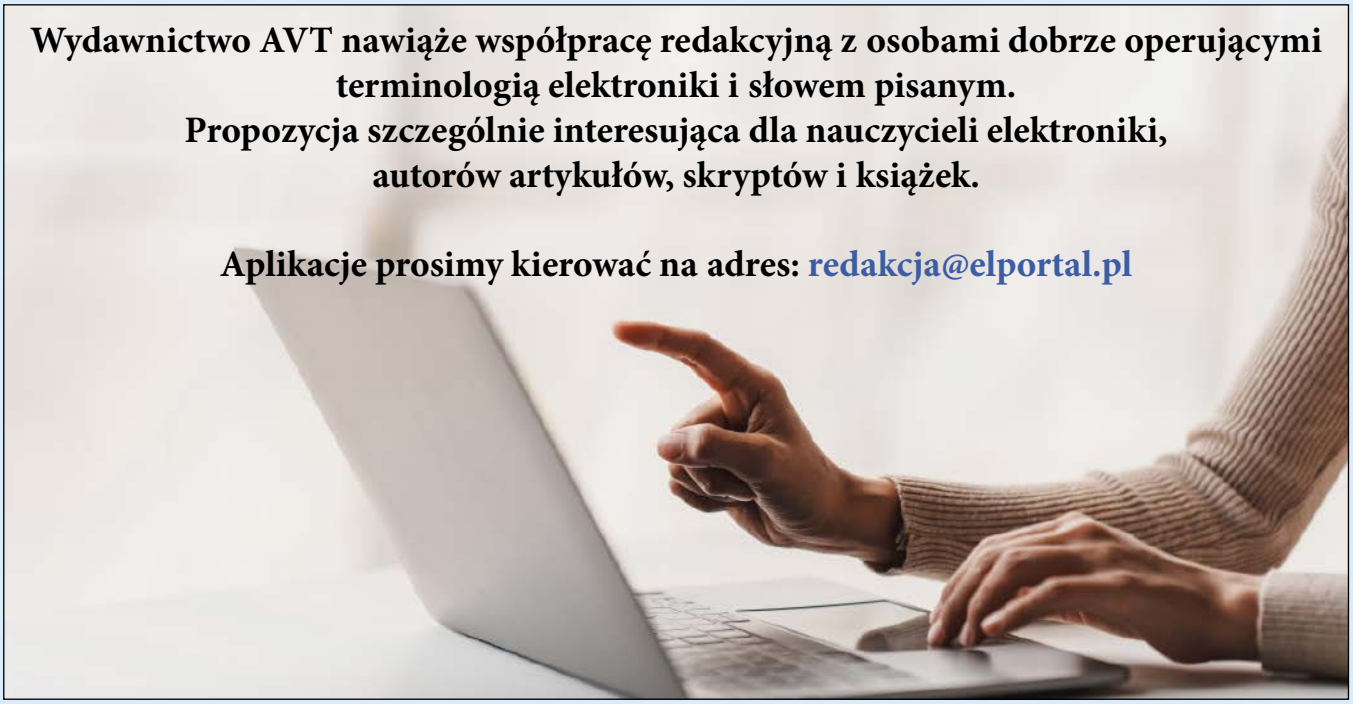
[1] Zapis ten można także skrócić do postaci:

```
if (USART2->ISR & USART_ISR_TXE_TXFNF)
```

REKLAMA

Wydawnictwo AVT nawiąże współpracę redakcyjną z osobami dobrze operującymi terminologią elektroniki i słowem pisanym.
Propozycja szczególnie interesująca dla nauczycieli elektroniki, autorów artykułów, skryptów i książek.

Aplikacje prosimy kierować na adres: redakcja@elportal.pl



Poprzednie części kursu i dodatkowe materiały dostępne są na stronie **https://ulubionykiosk.pl/media**

eprasa.pl 65e3eb482b

Kurs FPGA Lattice (17)

Odtwarzacz melodii

W poprzednim odcinku stworzyliśmy moduł, który odtwarza dźwięk o żądanej długości i częstotliwości. Teraz rozbudujemy ten projekt, dodając pamięć ROM z melodyjkami oraz moduł, który będzie odczytywał nuty z pamięci i przekazywał je do modułu generującego dźwięki.

Poprzednie odcinki znajdują się pod adresem: **https://ulubionykiosk.pl/media**

Zobacz więcej, materiały dodatkowe:

- Repozytorium modułów wykorzystywanych w kursie: https://github.com/leonow32/verilog-fpga
- MachXO2 Family Datasheet: https://tiny.pl/cgwbb
- Memory Usage Guide for MachXO2 Devices: https://tiny.pl/c61gc
- Projekt w programie Diamond: https://tiny.pl/dmmj3
- Film demonstrujący działanie kodu z tego odcinka kursu: https://www.youtube.com/embed/qpO5nwuU3do
- Artykuł dostepny on-line: https://tiny.pl/dmmjn

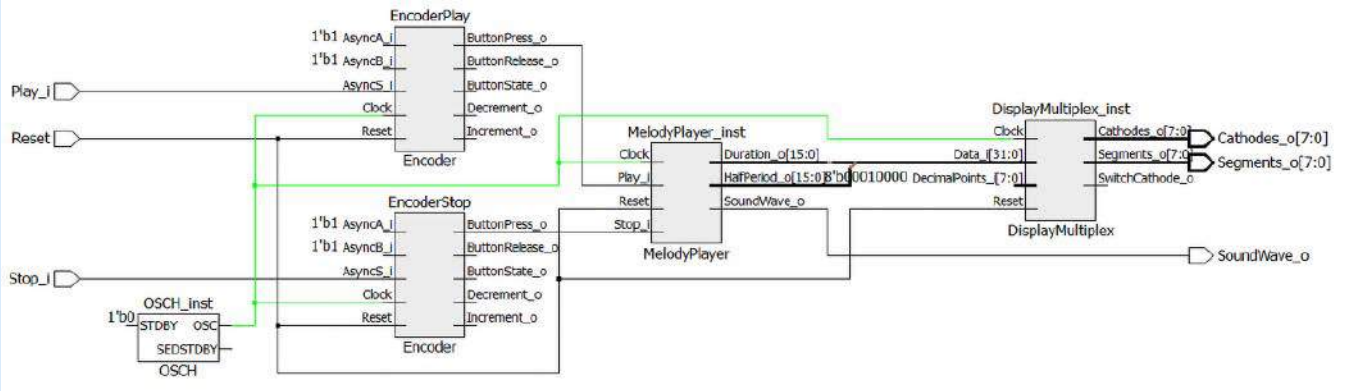
Projekt, którym zajmiemy się w tym odcinku kursu, będzie trochę bardziej skomplikowany niż konstrukcje stworzone przez nas wcześniej. Z tego powodu zaczniemy od zapoznania się ze schematem projektu, zaprezentowanym na **rysunku 1** oraz hierarchią modułów, którą widać na **rysunku 2**.

Utwórz nowy projekt w Lattice Diamond i dodaj do niego pliki widoczne na **rysunku 3**. Wszystkie moduły, z wyjątkiem **top** i **MelodyPlayer**, omawialiśmy już w poprzednich odcinkach kursu. Każdy z nich możesz pobrać z repozytorium na GitHubie, a także pobrać gotowy projekt dla programu Diamond. Linki do źródeł znajdziesz w ramce.

Moduł top

Przeanalizujmy plik **top.v**, którego kod prezentuje **listing 1**. Na liście portów mamy następujące wejścia i wyjścia:

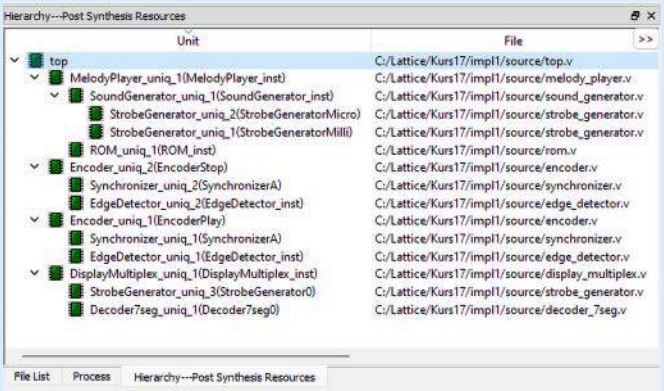
- **Reset** – standardowo, jak w każdym poprzednim projekcie, reset asynchroniczny jest wyzwalany przyciskiem K0 na płytce MachXO2 Mega.
- **Play_i** oraz **Stop_i** – są to wejścia podłączone do przycisków, w enkoderach E41 (górny) i E42 (dolny). Te przyciski będą służyć do uruchomienia odtwarzania melodii oraz do jej zatrzymania.



Rysunek 1. Schemat modułu top

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie **https://ulubionykiosk.pl/media**

eprasa.pl 65e3eb482b

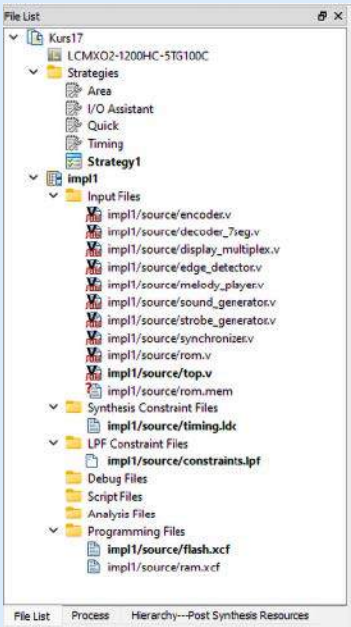


Rysunek 2. Hierarchia modułów w projekcie

- **SoundWave_o** – wyjście podłączone do tranzystora sterującego głośniczkiem.
- **Segments_o[7:0]** – 8-bitowe wyjście sterujące segmentami wyświetlacza multipleksowanego.
- **Cathodes_o[7:0]** – 8-bitowe wyjście sterujące katodami wyświetlacza multipleksowanego.

W pierwszej kolejności budujemy generator sygnału zegarowego, tworząc instancję modułu **OSCH** – dokładnie tak samo, jak w poprzednich odcinkach kursu.

Następnie musimy zająć się wykrywaniem wciśnięcia gałek enkoderów. Na płytce User Interface Board zastosowano już odpowiednie rezystory pull-up oraz filtry RC do odszumiania drgań styków, więc nie potrzebujemy żadnego dodatkowego kodu w tym celu. Wystarczą nam dwie instancje modułów **Encoder** – po jednej na każdy przycisk. Omówimy tylko jedną z nich, ponieważ obie działają w identyczny sposób.



Rysunek 3. Drzewko projektu z widoczną listą plików źródłowych

W linii 2 tworzymy instancję modułu enkodera, który obsługiwać będzie enkoder E41 – przycisk E41 używany jest z kolei do uruchamiania odtwarzacza. Interesuje nas tylko wykrywanie wciśnięcia gałki enkodera, więc do jego wejścia **AsyncS_i** doprowadzamy sygnał **Play_i**. Moduł enkodera zapewnia synchronizację sygnałów wejściowych z domeną zegarową, a także wykrywanie zmian badanego sygnału. W tym przypadku używać będziemy tylko wyjścia **ButtonPress_o**, informującego o wykryciu wciśnięcia przycisku (linia 6). Łączymy je z sygnałem Play typu wire, który został utworzony w linii 1.

Nie interesuje nas wykrywanie obrotu gałki, zatem do wejść **AsyncA_i** oraz **AsyncB_i** doprowadzamy na stałe stan wysoki (linie 3 i 4), a wszystkie pozostałe wyjścia modułu enkodera pozostawiamy niepołączone.

Dalej, w linii 7 i kolejnej tworzymy dwie 16-bitowe zmienne wire **Duration_ms** oraz **HalfPeriod_us**. Posłużą one do przekazywania informacji o aktualnie odtwarzanym dźwięku z modułu **MelodyPlayer** (linie 11 i 12) do modułu sterującego wyświetlaczem **DisplayMultiplex** (linia 14).

```
// Plik top.v

`default_nettype none
module top(
    input  wire Reset,              // Pin 17
    input  wire Play_i,             // Pin 66
    input  wire Stop_i,             // Pin 69
    output wire SoundWave_o,        // Pin 28
    output wire [7:0] Segments_o,
    output wire [7:0] Cathodes_o
);

    // Generator sygnału zegarowego
    parameter CLOCK_HZ = 14_000_000;
    wire Clock;
    OSCH #(
        .NOM_FREQ("14.00")
    ) OSCH_inst(
        .STDBY(1'b0),
        .OSC(Clock),
        .SEDSTDBY()
    );

    // Wykrywanie zbocza opadającego na przycisku enkodera E41
    // W stanie spoczynkowym jest stan 1
    wire Play;                                          // 1
    Encoder EncoderPlay(                                // 2
        .Clock(Clock),
        .Reset(Reset),
        .AsyncA_i(1'b1),                                // 3
        .AsyncB_i(1'b1),                                // 4
        .AsyncS_i(Play_i),                              // 5
        .Increment_o(),
        .Decrement_o(),
        .ButtonPress_o(Play),                           // 6
        .ButtonRelease_o(),
        .ButtonState_o()
    );

    // Wykrywanie zbocza opadającego na przycisku enkodera E42
    wire Stop;
    Encoder EncoderStop(
        .Clock(Clock),
        .Reset(Reset),
        .AsyncA_i(1'b1),
        .AsyncB_i(1'b1),
        .AsyncS_i(Stop_i),
        .Increment_o(),
        .Decrement_o(),
        .ButtonPress_o(Stop),
        .ButtonRelease_o(),
        .ButtonState_o()
    );

    // Zmienne do pokazywania dźwięków na wyświetlaczu
    wire [15:0] Duration_ms;                            // 7
    wire [15:0] HalfPeriod_us;

    // Instancja odtwarzacza melodii
    MelodyPlayer #(                                     // 8
        .CLOCK_HZ(CLOCK_HZ)
    ) MelodyPlayer_inst(
        .Clock(Clock),
        .Reset(Reset),
        .Play_i(Play),                                  // 9
        .Stop_i(Stop),                                  // 10
        .SoundWave_o(SoundWave_o),                      // 11
        .Duration_o(Duration_ms),                       // 12
        .HalfPeriod_o(HalfPeriod_us)                    // 13
    );

    // Instancja sterownika wyświetlacza
    DisplayMultiplex #(
        .CLOCK_HZ(CLOCK_HZ),
        .SWITCH_PERIOD_US(1000),
        .DIGITS(8)
    ) DisplayMultiplex_inst(
        .Clock(Clock),
        .Reset(Reset),
        .Data_i({Duration_ms, HalfPeriod_us}),          // 14
        .DecimalPoints_i(8'b00010000),                  // 15
        .Cathodes_o(Cathodes_o),
        .Segments_o(Segments_o),
        .SwitchCathode_o()
    );

endmodule

`default_nettype wire
```

Listing 1. Kod pliku top.v

Przechodzimy wreszcie do instancji odtwarzacza melodyjek (linia 8). W liniach 9 i 10 mamy wejścia uruchamiające i przerywające odtwarzanie melodii – są to sygnały wychodzące ze sterowników enkoderów. W linii 11 znajduje się wyjście sygnału dźwiękowego, które wyprowadzone jest na pin układu FPGA. Linie 12 i 13 to wyjścia informujące o długości oraz półokresie aktualnie odtwarzanego dźwięku, które służą tylko do celów informacyjnych. Można te wyjścia pozostawić niepodłączone.

Na końcu mamy instancję sterownika wyświetlacza LED, który już wielokrotnie stosowaliśmy w poprzednich odcinkach kursu.

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie https://ulubionykiosk.pl/media

eprasa.pl 65e3eb482b

```
// Plik melody_player.v

`default_nettype none
module MelodyPlayer #(
    parameter  CLOCK_HZ = 10_000_000
)(
    input wire Clock,
    input wire Reset,
    input wire Play_i,
    input wire Stop_i,
    output wire SoundWave_o,

    output wire [15:0] Duration_o,
    output wire [15:0] HalfPeriod_o
);

    // Pamięć melodii
    reg  [11:0] Address;                                    // 1
    wire [ 7:0] Data;                                       // 2
    ROM #(
        .ADDRESS_WIDTH(12),                                 // 3
        .DATA_WIDTH(8),                                     // 4
        .MEMORY_FILE("rom.mem")                             // 5
    ) ROM_inst(
        .Clock(Clock),
        .Reset(Reset),
        .ReadEnable_i(1'b1),                                // 6
        .Address_i(Address),                                // 7
        .Data_o(Data)                                       // 8
    );

    // Zmienne
    reg [15:0] Duration_ms;                                 // 9
    reg [15:0] HalfPeriod_us;                               // 10
    reg Request;                                            // 11
    wire SoundGeneratorDone;                                // 12

    // Rejestr stanów i możliwe stany
    reg [2:0] State;                                        // 13
    localparam IDLE             = 3'd0;
    localparam DUMMY            = 3'd1;
    localparam READ_DURATION_H  = 3'd2;
    localparam READ_DURATION_L  = 3'd3;
    localparam READ_HPERIOD_H   = 3'd4;
    localparam READ_HPERIOD_L   = 3'd5;
    localparam PLAYING          = 3'd6;

    // Maszyna stanów, która odczytuje kolejne nuty z pamięci
    // i przekazuje do generatora dźwięków
    always @(posedge Clock, negedge Reset) begin            // 14
        if(!Reset) begin                                    // 15
            State         <= IDLE;
            Request       <= 0;
            Duration_ms   <= 0;
            HalfPeriod_us <= 0;
            Address       <= 0;
        end else begin
            case(State)                                     // 16
                IDLE: begin
                    if(Play_i) begin
                        Address         <= 0;
                        State           <= DUMMY;
                    end
                end

                DUMMY: begin                                // 17
                    Address             <= Address + 1'b1;
                    State               <= READ_DURATION_H;
                end

                READ_DURATION_H: begin                      // 18
                    Duration_ms[15:8]   <= Data;
                    Address             <= Address + 1'b1;
                    State               <= READ_DURATION_L;
                end

                READ_DURATION_L: begin                      // 19
                    Duration_ms[7:0]    <= Data;
                    Address             <= Address + 1'b1;
                    State               <= READ_HPERIOD_H;
                end

                READ_HPERIOD_H: begin                       // 20
                    HalfPeriod_us[15:8] <= Data;
                    Address             <= Address + 1'b1;
                    State               <= READ_HPERIOD_L;
                end

                READ_HPERIOD_L: begin                       // 21
                    HalfPeriod_us[7:0]  <= Data;
                    Request             <= 1'b1;
                    State               <= PLAYING;
                end

                PLAYING: begin                              // 22
                    Request <= 1'b0;
                    if(Duration_ms == 16'd0) begin          // 23
                        State           <= IDLE;
                    end else if(Stop_i) begin               // 24
                        State           <= IDLE;
                        Duration_ms     <= 16'd0;
                        HalfPeriod_us   <= 16'd0;
                    end else if(SoundGeneratorDone) begin   // 25
                        State           <= READ_DURATION_H;
                        Address         <= Address + 1'b1;
                    end
                end
            endcase
        end
    end

    // Instancja generatora dźwięków
    SoundGenerator #(                                       // 26
        .CLOCK_HZ(CLOCK_HZ)
    ) SoundGenerator_inst(
        .Clock(Clock),
        .Reset(Reset),
        .Start_i(Request),                                  // 27
        .Finish_i(Stop_i),
        .Duration_ms_i(Duration_ms),                        // 28
        .HalfPeriod_us_i(HalfPeriod_us),                    // 29
        .SoundWave_o(SoundWave_o),
        .Busy_o(),
        .Done_o(SoundGeneratorDone)                         // 30
    );

    // Tylko na potrzeby debugowania
    // aby pokazać czas trwania i półokres nuty na wyświetlaczu
    assign Duration_o   = Duration_ms;
    assign HalfPeriod_o = HalfPeriod_us;

endmodule
`default_nettype wire
```

Listing 2. Kod pliku melody_player.v

Zwróć uwagę, że wejście **Data_i** modułu wyświetlacza jest 32-bitowe, ponieważ wyświetlacz ma osiem cyfr wyświetlających 4-bitową szesnastkową cyfrę od 0 do 9 i od A do F. Z tego powodu dwie 16-bitowe zmienne **Duration_ms** oraz **HalfPeriod_us** sklejamy ze sobą za pomocą operatora konkatenacji {}. W ten sposób cztery cyfry wyświetlacza z prawej strony będą pokazywać półokres odtwarzanego dźwięku, a pozostałe cztery cyfry z lewej będą pokazywały jego długość (linia 14). W celu zwiększenia czytelności ustawimy wyświetlanie kropki pomiędzy tymi dwiema liczbami (linia 15).

Moduł MelodyPlayer

Moduł **MelodyPlayer** zawiera w sobie dwa moduły podrzędne – pamięć **ROM**, omawianą w 15 odcinku kursu (w niej zapisana jest melodia) oraz moduł **SoundGenerator**, opracowany przez nas w 16 odcinku. Kod tego modułu pokazano na **listingu 2**.

Praca niniejszego modułu jest bardzo prosta. Rozpoczyna ją wystąpienie stanu wysokiego na wejściu **Play_i**. Stan wysoki powinien trwać jeden cykl sygnału zegarowego, a następnie wejście to powinno przejść w stan niski (kolejne próby uruchomienia pracującego już modułu będą ignorowane). Moduł odczytuje informacje o dźwięku, który ma być odtworzony. Kopiuje te dane z pamięci ROM do zmiennych roboczych, po czym uruchamia odtwarzanie dźwięku w module **SoundGenerator**. Następnie czeka, aż odtwarzanie nuty zostanie zakończone i wtedy odczytuje kolejny dźwięk z pamięci. Odbywa się to tak długo, aż moduł odczyta nutę o zerowej długości lub aż pojawi się stan wysoki na wejściu **Stop_i**.

Każda nuta definiowana jest dwiema 16-bitowymi liczbami. Są to: czas trwania w milisekundach (**Duration_ms**) oraz półokres sygnału dźwiękowego, podany w mikrosekundach (**HalfPeriod_us**). Zatem na każdą nutę przypadają cztery bajty. Jeżeli półokres jest równy zeru, mamy pauzę, której długość określona została w **Duration_ms**. Natomiast jeżeli **Duration_ms** jest równe zero, to znaczy, że nie ma już więcej nut do odtworzenia i dotarliśmy do końca melodii.

Portów wejścia i wyjścia nie będziemy opisywać, ponieważ nie ma tutaj nic, czego nie omówiliśmy już wcześniej.

Kod modułu zaczynamy od instancji pamięci ROM, w której przechowywane będą melodie. W liniach 1 i 2 tworzymy dwie kluczowe zmienne do obsługi pamięci, tzn. **Addess** typu reg oraz **Data** typu wire. Szerokość zmiennej **Data** to osiem bitów, czyli jeden bajt, natomiast zmienna Address jest 12-bitowa. Oznacza to, że możemy zaadresować 2^{12}, czyli 4096 bajtów. Każda nuta zajmuje 4 bajty, więc sumarycznie będziemy mogli zapisać 1024 nuty.

Następnie, w liniach 3 i 4 konfigurujemy instancję modułu za pomocą parametrów **ADDRESS_WIDTH** oraz **DATA_WIDTH**,

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie https://ulubionykiosk.pl/media

eprasa.pl 65e3eb482b

w których trzeba podać liczbę bitów zmiennych **Addess** oraz **Data**. Dalej podajemy nazwę pliku z zawartością pamięci (linia 5). W kolejnej części kursu zobaczymy, jak ustworzyć taki plik.

Teraz podłączamy wejścia i wyjścia. W linii 6 wejście **ReadEnable** ustawiamy na stałe w stan wysoki. Jest to funkcjonalność wbudowana w blok EBR polegająca na tym, że pamięć ignoruje zegar i wszelkie zmiany na wejściu adresowym, jeżeli wejście **ReadEnable** jest w stanie niskim. W linii 7 łączymy wejście adresowe ze zmienną **Address**, sterowaną przez maszynę stanów, którą omówimy za chwilę. W linii 8 łączymy wyjście danych ze zmienną **Data**, które odczytywane jest przez logikę tej samej maszyny stanów.

Tworzymy dwie 16-bitowe zmienne **Duration_ms** (linia 9) oraz **HalfPeriod_us** (linia 10) typu reg. Te zmienne będziemy zapisywać w maszynie stanów, którą za chwilę omówimy, danymi pobranymi z pamięci ROM. Będziemy odczytywać – bajt po bajcie – dane z pamięci ROM i kopiować je do tych zmiennych.

Celem zmiennej **Request** (linia 11) jest uruchomienie modułu, w chwili gdy zakończone zostanie odczytywanie z pamięci kolejnej nuty i będzie ona gotowa do odczytu w zmiennych **Duration_ms** oraz **HalfPeriod_us**. Zmienna ta będzie ustawiana w stan wysoki na jeden cykl sygnału zegarowego.

Pozostaje nam już tylko utworzenie zmiennej **SoundGeneratorDone**, typu wire (linia 12). Służy ona do przekazywania sygnału zakończenia odtwarzania nuty z modułu **SoundGenerator** (linia 30) do maszyny stanów, która uruchamia odtwarzanie kolejnej nuty (linia 25).

Przejdźmy do maszyny stanów. Jej rejestr tworzymy w linii 13 i następnie definiujemy wszystkie możliwe stany. Cała logika maszyny stanów zawarta jest w jednym synchronicznym bloku always (linia 14). Maszyna używa kilka zmiennych, które już wcześniej omówiliśmy, a w momencie resetu układu inicjalizujemy je w linii 15 i kilku kolejnych.

Maszyna stanów opisana została w postaci instrukcji warunkowej **case**, wykonującej różne zadania w zależności aktualnej wartości od zmiennej **State** (linia 16).

Stan **IDLE** to stan bezczynności, kiedy nie jest odtwarzana żadna melodia. W tej sytuacji sprawdzamy jedynie, czy sygnał **Play_i** jest w stanie wysokim i jeżeli tak, to zerujemy licznik adresu, aby odtwarzać melodię od początku i jednocześnie zmieniamy stan na **DUMMY**.

W kolejnym cyklu zegara wchodzimy do stanu **DUMMY**, w którym jedynie zwiększamy licznik adresu o 1 i przełączamy stan na **READ_DURATION_H** (linia 17). Można zadać pytanie, dlaczego zwiększyliśmy adres, a nie odczytaliśmy danych z adresu zerowego? Otóż bloki pamięci EBR, w przeciwieństwie do tradycyjnych pamięci, działają synchronicznie z sygnałem zegarowym. W jednym takcie zegarowym moduł sterujący pamięcią ustawia żądany adres. W kolejnym takcie – pamięć udostępnia dane z tego adresu na swoim wyjściu **Data_o**. Dopiero w następnym takcie można te dane odczytać.

Stany **READ_DURATION_H**, **READ_DURATION_L** oraz **READ_HPERIOD_H** (linie 18-20) wyglądają bardzo podobnie. Odczytujemy kolejny bajt danych i kopiujemy go do odpowiednich połówek 16-bitowych zmiennych **Duration_ms** oraz **HalfPeriod_us**. Każdy z tych stanów trwa tylko jeden takt zegarowy.

Dochodzimy wreszcie do stanu **READ_HPERIOD_L** (linia 21). W tym momencie kopiujemy ostatni z czterech odczytanych bajtów do **HalfPeriod_us**. Dalej ustawiamy zmienną **Request** w stan wysoki, aby poinformować moduł **SoundGenerator**, że przy kolejnym takcie zegara ma odczytać dane na swoich wejściach **Duration_ms_i** oraz **HalfPeriod_us_i**, a następnie ma rozpocząć odtwarzanie dźwięku. Moduł SoundGenerator odczytuje zmienną **Request** poprzez wejście **Start_i** (linia 27). Następnie stan maszyny jest zmieniany na **PLAYING**.

W linii 22 obsługujemy stan **PLAYING.** Niezależnie od wszystkich innych rzeczy, zerujemy zmienną **Request**, ponieważ moduł SoundGenerator rozpoczyna w tym momencie odtwarzanie dźwięku, a ta zmienna powinna mieć wartość 1 tylko przez jeden takt sygnału zegarowego. Jednocześnie, w tym samym takcie zegarowym, możliwe są do wykonania trzy operacje:

- Jeżeli odczytaliśmy z pamięci nutę o zerowej długości, tzn. zmienna **Duration_ms** jest równa zero (linia 23), to znaczy, że dotarliśmy do końca melodii. W związku z tym stan maszyny jest ustawiany na IDLE, w którym czekać będzie ona na kolejne wciśnięcie przycisku start.
- Jeżeli warunek 1 nie został spełniony, to następnie sprawdzamy, czy został wciśnięty przycisk stop (linia 24). W takiej sytuacji zmieniamy stan maszyny na IDLE oraz zerujemy zmienne tymczasowe.
- Jeżeli warunki 1 i 2 nie zostały spełnione, sprawdzamy, czy został ustawiony **SoundGeneratorDone** (linia 25) przez moduł **SoundGenerator**. Sygnał ten jest ustawiany na wyjściu tego modułu w linii 30. Informuje on, że odtwarzanie nuty zostało zakończone. Wtedy zmieniamy stan maszyny na **READ_DURATION_H**, aby odczytać kolejną nutę oraz inkrementujemy licznik adresu.

Jeżeli żaden z tych warunków nie został spełniony – to maszyna stanów nie robi nic. Trwa odtwarzanie dźwięku. W tym

```
// Plik melody_player_tb.v

`timescale 1ns/1ns
`default_nettype none
module MelodyPlayer_tb();

    parameter CLOCK_HZ       = 10_000_000;
    parameter HALF_PERIOD_NS = 1_000_000_000/(2 * CLOCK_HZ);

    // Generator sygnału zegarowego
    reg Clock = 1'b1;
    always begin
        #HALF_PERIOD_NS;
        Clock = !Clock;
    end

    // Zmienne
    reg Reset   = 1'b0;
    reg Play    = 1'b0;
    reg Stop    = 1'b0;
    wire SoundWave;

    // Instancja testowanego modułu
    MelodyPlayer #(
        .CLOCK_HZ(CLOCK_HZ)
    ) DUT(
        .Clock(Clock),
        .Reset(Reset),
        .Play_i(Play),
        .Stop_i(Stop),
        .SoundWave_o(SoundWave)
    );

    // Zapisywanie zmiennych w trakcie symulacji
    initial begin
        $dumpfile("melody_player.vcd");
        $dumpvars(0, MelodyPlayer_tb);
    end

    // Sekwencja testowa
    initial begin
        $timeformat(-6, 3, "us", 12);
        $display("===== START =====");
        $display("        Time Durat HaPer        Freq");  // 1

        #1 Reset <= 1'b1;

        repeat(9) @(posedge Clock);                        // 2

        @(posedge Clock)                                   // 3
        Play <= 1'b1;
        @(posedge Clock)
        Play <= 1'b0;

        repeat(10) @(posedge Clock);                       // 4

        wait(DUT.State == DUT.IDLE);                       // 5
        repeat(10) @(posedge Clock);                       // 6

        $display("====== END ======");
        $finish;
    end

endmodule
`default_nettype wire
```

Listing 3. Kod pliku melody_player_tb.v

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie https://ulubionykiosk.pl/media

eprasa.pl 65e3eb482b

```
// Plik rom.mem
00 20 07 77 // 64c1
00 20 03 BB // 64c2
00 40 01 DD // 32c3
00 00 00 00 // koniec
```

Listing 4. Kod pliku rom.mem

czasie czekamy tak długo, aż któryś z trzech opisanych warunków stanie się prawdziwy.

Pozostaje już tylko utworzyć instancję modułu **SoundGenerator** w linii 26. Moduł ten omawialiśmy szczegółowo w poprzednim odcinku kursu.

Testbench modułu MelodyPlayer

Zanim wgramy bitstream do FPGA, przetestujmy, jak działa nasz nowy odtwarzacz melodyjek. W testbenchu utworzymy instancję modułu **MelodyPlayer** i podamy mu plik, w którym ma się znajdować testowa melodyjka, składająca się tylko z kilku prostych dźwięków. Przeanalizujmy kod testbencha, który pokazano na **listingu 3**.

Testbench zaczyna się tak samo, jak każdy inny omawiany w tym kursie, więc przeskoczymy od razu do sekwencji testowej. W gruncie rzeczy jest ona bardzo prosta. W linii 1 printujemy nagłówek tabeli, w której pojawiać się będą informacje o aktualnie odtwarzanym dźwięku. Następnie odczekujemy kilka taktów zegarowych tylko po to, by wykres w GTKWave ładniej wyglądał (linia 2) – dzięki temu uzyskamy odstęp między krawędzią okna wykresu a interesującymi zmianami sygnałów.

W linii 3 i kolejnych oczekujemy na zbocze rosnące sygnału zegarowego. Następnie ustawiamy zmienną **Play** w stan wysoki. Jest ona połączona z wejściem **Play_i** testowanego modułu, a stan wysoki powoduje rozpoczęcie odtwarzania melodii. Później znów czekamy na zbocze rosnące sygnału zegarowego, po czym przywracamy tę zmienną w stan niski. Rozpoczyna się odtwarzanie dźwięku. Moduł **MelodyPlayer** zaczyna odczytywać z pamięci pierwszą nutę i przekazuje ją do modułu **SoundGenerator**.

Maszyna stanów modułu **MelodyPlayer** pracuje i kopiuje kolejne nuty do generatora dźwięków tak długo, aż się skończą, tzn. zostanie odczytany dźwięk o zerowym półokresie i zerowej częstotliwości. Wtedy maszyna stanów przejdzie ponownie w stan IDLE. W linii 5 umieściliśmy instrukcję **wait**, która zawiesza wykonywanie instrukcji sekwencji testowej aż do wystąpienia opisanej sytuacji. Taka metoda testowania jest wygodna – daje nam możliwość sprawdzenia dowolnej melodii, niezależnie od tego, jak długo trwa, ile dźwięków jest zapisanych w pamięci, czy też jaka jest częstotliwość sygnału zegarowego w symulacji (nie musimy bawić się w odliczanie cykli zegarowych).

Dalej pozostaje już tylko poczekać kilka taktów zegarowych, aby w przeglądarce GTKWave widoczny stał się odstęp między końcem odtwarzania dźwięku a końcem symulacji (linia 6). Następnie kończymy symulację za pomocą instrukcji **$finish**.

Pamięć melodii musimy wypełnić jakąś melodią. Na potrzeby testbencha utworzymy plik rom.mem, którego zawartość zaprezentowano na **listingu 4**. Zawiera on tylko trzy krótkie dźwięki, aby symulacja wykonywała się w miarę szybko. Oczywiście moglibyśmy zasymulować jakąś rzeczywistą melodię, ale wtedy symulator pracowałby bardzo długo, a plik z wynikami symulacji miałby wiele gigabajtów danych.

Jak należy stworzyć plik z zawartością pamięci, zobaczymy w dalszej części artykułu.

Pozostaje już tylko uruchomić symulator skryptem, którego kod znajduje się na **listingu 5**.

Po przeprowadzeniu symulacji powinniśmy otrzymać na konsoli komunikaty pokazane na **listingu 6**. Warning w pierwszej linii jest zjawiskiem prawidłowym i wynika z tego, że pamięć zainicjalizowaliśmy plikiem zawierającym zaledwie 16 bajtów. Pamięć ma 4096 bajtów, więc reszta komórek pozostaje niezainicjalizowana (przy próbie odczytania niezainicjalizowanych komórek pamięci dostaniemy wartość nieokreśloną X).

W następnej tabeli widzimy czas symulacji, w którym rozpoczęło się odtwarzanie poszczególnych dźwięków. Kolumna Durat to skrót od Duration, czyli czas trwania dźwięku w milisekundach. HaPer to półokres, a Freq to oczywiście częstotliwość w hercach.

Zobaczmy screenshoty z symulacji. Pierwszy z nich, ukazujący całą symulację od początku do końca, pokazano na **rysunku 4**. Sygnał wyjściowy, doprowadzony do głośniczka, w celu poprawy czytelności zaznaczono kolorem pomarańczowym. Widać wyraźnie, że podczas symulacji zostały odtworzone trzy dźwięki o różnych częstotliwościach, przy czym pierwszy i drugi dźwięk były tak samo krótkie, a trzeci był dwukrotnie dłuższy.

Zwróć uwagę na **DurationTimer** oraz **HalfPeriodTimer**. Te dwa liczniki celowo zostały przedstawione w formie analogowej. Widzimy wyraźnie, że na początku odtwarzania dźwięku licznik **DurationTimer** jest ładowany wartością początkową i odlicza w dół, aż osiągnie wartość zerową, co kończy odtwarzanie dźwięku. Natomiast **HalfPeriodTimer** podczas odtwarzania nuty wielokrotnie liczy od wartości początkowej do zera. Kiedy osiągnie zero, wówczas odwracany jest stan wyjścia **SoundWave**.

Przybliżmy widok na początek pracy, co prezentuje **rysunek 5**. Wszystko zaczyna się od stanu wysokiego na wejściu **Play_i**, co trwa przez jeden takt zegarowy. Dla zwiększenia czytelności zaznaczono ten sygnał kolorem żółtym. Skutkuje to uruchomieniem maszyny stanów. Zwróć uwagę, że inkrementuje się licznik adresu pamięci. Pamięć podaje różne dane, które są przepisywane do zmiennych **Duration_ms** i **HalfPeriod_us**. Finalnie maszyna stanów przechodzi w stan **PLAYING** i ustawia wejście **Start_i** modułu **SoundGenerator** w stan wysoki na jeden cykl zegara. To sprawia, że w następnym cyklu zegarowym liczniki odpowiedzialne za liczenie długości i półokresu dźwięku zaczynają pracować. Sygnał **Busy_o** przechodzi w stan wysoki, ponieważ zaczyna się

```
@echo off
iverilog -o melody_player.o melody_player.v melody_player_tb.v sound_generator.v strobe_generator.v rom.v
vvp melody_player.o
del melody_player.o
```

Listing 5. Kod pliku melody_player.bat

```
WARNING: ../rom/rom.v:17: $readmemh(rom.mem): Not enough words in the file for the requested range [0:4095].
VCD info: dumpfile melody_player.vcd opened for output.
===== START =====
        Time Durat HaPer        Freq
     1.700us    32  1911         261
 32002.300us    32   955         523
 64002.900us    64   477        1046
128003.500us     0     0           0
====== END ======
melody_player_tb.v:60: $finish called at 128004500 (1ns)
```

Listing 6. Wynik symulacji na konsoli

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie https://ulubionykiosk.pl/media

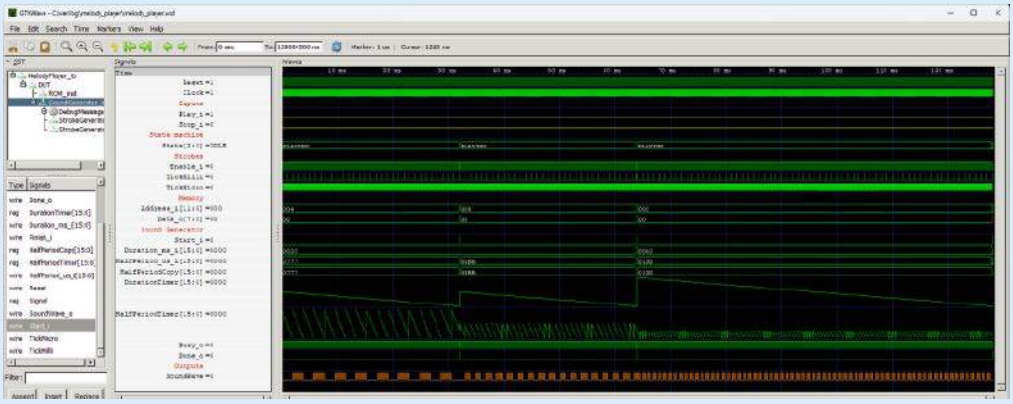
eprasa.pl 65e3eb482b

właściwa praca modułu odtwarzającego dźwięki.

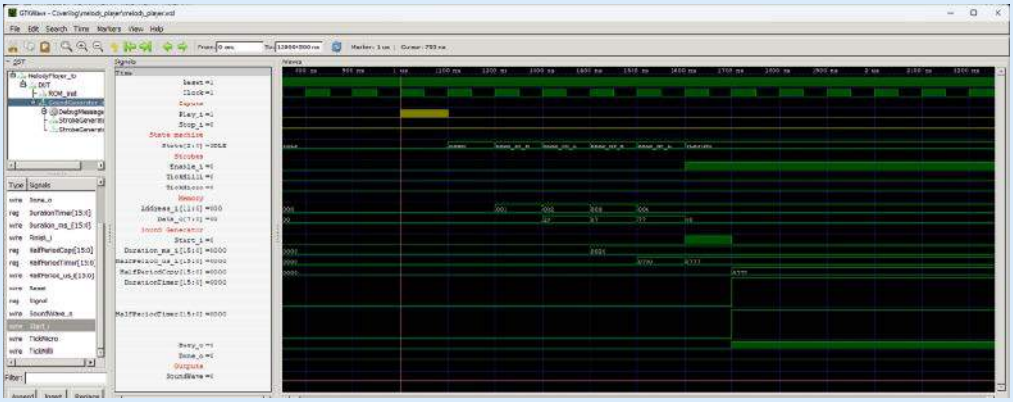
Przyjrzyjmy się, jak działa odczytywanie kolejnej nuty po zakończeniu odtwarzania aktualnego dźwięku (**rysunek 6**). Kiedy w module **SoundGenerator** licznik **DurationTimer** jest równy zero i jednocześnie **TickMilli** jest w stanie wysokim, to znaczy, że upłynął czas odtwarzania dźwięku. Z tego powodu wyjście **Busy_o** generatora dźwięków przechodzi w stan niski do czasu, aż zostanie on ponownie uruchomiony z nowymi danymi. Ponadto jego wyjście **Done_o** zostaje ustawione w stan wysoki na jeden takt zegarowy. To powoduje ponowne uruchomienie maszyny stanów modułu **MelodyPlayer**, przez co zostają odczytane kolejne bajty i załadowane do liczników. Kiedy liczniki są już załadowane nowymi danymi, ustawiamy wejście **Start_i** modułu **SoundGenerator** w stan wysoki na jeden takt zegarowy i rozpoczyna się odtwarzanie kolejnej nuty.

Plik z melodyjkami

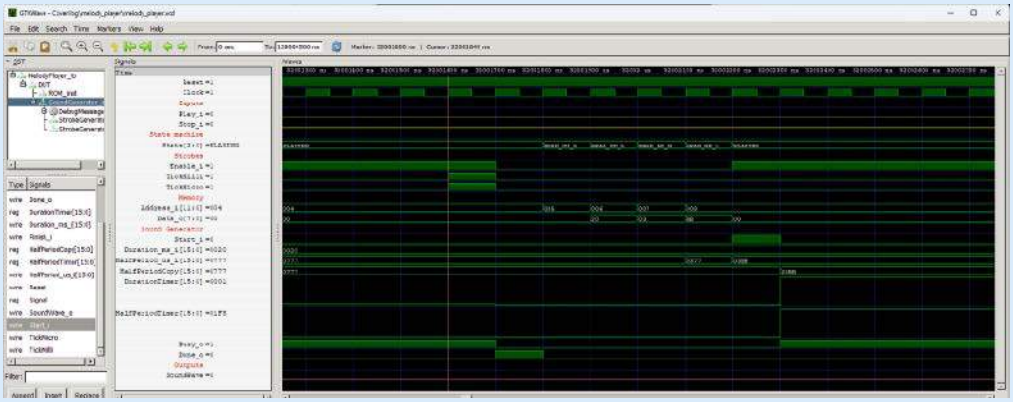
Mamy już gotowy moduł odtwarzający dźwięk, odczytujący nuty z pamięci, mamy także moduł pamięci ROM, lecz trzeba jeszcze zapełnić tę pamięć jakimiś melodyjkami. Aby ułatwić sobie zadanie przekształcania nut na częstotliwość dźwięku i długość w milisekundach, stwierdziłem, że dobrze będzie



Rysunek 4. Przebiegi uzyskane podczas symulacji



Rysunek 5. Zbliżenie na odczytywanie pierwszej nuty z pamięci



Rysunek 6. Zbliżenie na moment odczytywania kolejnej nuty

Start Page | Reports | Spreadsheet View

	Name	Group By	Pin	BANK	BANK_VCC	VREF	IO_TYPE	PULLMODE	DRIVE	SLEWRATE	CLAMP	OPENDRAIN	DIFFRESISTOR	DIFFDRIVE	HYSTERESIS
1	All Ports	N/A	N/A	N/A	N/A	N/A			N/A	N/A					
1.1	Input	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.1.1	Play_i	N/A	66	1	Auto	N/A	LVCMOS33	NONE	NA	NA	ON	OFF	OFF	NA	LARGE
1.1.2	Reset	N/A	17	3	Auto	N/A	LVCMOS33	UP	NA	NA	ON	OFF	OFF	NA	SMALL
1.1.3	Stop_i	N/A	69	1	Auto	N/A	LVCMOS33	NONE	NA	NA	ON	OFF	OFF	NA	LARGE
1.2	Output	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1.2.1	Cathodes_o[0]	N/A	25	3	Auto	N/A	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.2	Cathodes_o[1]	N/A	51	1	Auto	N/A	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.3	Cathodes_o[2]	N/A	47	2	Auto	N/A	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.4	Cathodes_o[3]	N/A	45	2	Auto	N/A	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.5	Cathodes_o[4]	N/A	43	2	Auto	N/A	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.6	Cathodes_o[5]	N/A	42	2	Auto	N/A	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.7	Cathodes_o[6]	N/A	41	2	Auto	N/A	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.8	Cathodes_o[7]	N/A	40	2	Auto	N/A	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.9	Segments_o[0]	N/A	29	2	Auto	N/A	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.10	Segments_o[1]	N/A	30	2	Auto	N/A	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.11	Segments_o[2]	N/A	34	2	Auto	N/A	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.12	Segments_o[3]	N/A	35	2	Auto	N/A	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.13	Segments_o[4]	N/A	36	2	Auto	N/A	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.14	Segments_o[5]	N/A	37	2	Auto	N/A	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.15	Segments_o[6]	N/A	38	2	Auto	N/A	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.16	Segments_o[7]	N/A	39	2	Auto	N/A	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.17	SoundWave_o	N/A	28	2	Auto	N/A	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA

Rysunek 7. Konfiguracja pinów w Spreadsheet

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie
https://ulubionykiosk.pl/media

eprasa.pl 65e3eb482b

```
melody = ""

# Debug
#melody += "64c1 64c2 32c3 "

# Axel F
melody += "4g2 8.#a2 16g2 16- 16g2 8c3 8g2 8f2 4g2 8.d3 16g2 16- 16g2 8#d3 8d3 8#a2 8g2 8d3 8g3 16g2 16f2 16- 16f2 8d2 8a2 2g2 2- "

# Star wars
melody += "4a1 4a1 4a1 4f1 16c2 4a1 4f1 16c2 2a1 4e2 4e2 4e2 4f2 16c2 4#g1 4f1 16c2 2a1 4a2 4a1 16a1 4a2 4#g2 16g2 16#f2 16f2 4#f2 8#a1
4#d2 4d2 16#c2 16c2 16b1 4c2 8f1 4#g1 4f1 16#g1 4c2 4a1 16c2 2e2 2- "

# Harry Potter
melody += "8b1 8.e2 16g2 8#f2 4e2 8b2 4.a2 4.#f2 8.e2 16g2 8#f2 4d2 8f2 2b1 8- 8b1 8.e2 16g2 8#f2 4e2 8b2 4d3 8#c3 4c3 8#g2 8.c3 16b2
8#a2 4#f2 8g2 2e2 8- 8g2 4b2 8g2 4b2 8g2 4c3 8b2 4#a2 8#f2 8.g2 16b2 8#a2 4#a1 2- "

# Pink Panther
melody += "8#g1 2a1 8b1 2c2 8#g1 8a1 8b1 8c2 8f2 8e2 8a1 8c2 8e2 2#d2 16d2 16c2 16a1 8g1 1a1 8#g1 2a1 8b1 2c2 8#g1 8a1 8b1 8c2 8f2 8e2
8c2 8e2 8a2 1#g2 8#g1 2a1 8b1 2c2 16#g1 8a1 8b1 8c2 8f2 8e2 8a1 8c2 8e2 2#d2 8d2 16c2 16a1 2- "

# Star Wars 2
melody += "8#c1 8#c1 16#c1 2#f1 2#c2 8b1 16#a1 8#g1 2#f2 4#c2 8b1 16#a1 8#g1 2#f2 4#c2 8b1 16#a1 8b1 2#g1 8#c1 8#c1 16#c1 2#f1 2#c2 8b1
16#a1 8#g1 2#f2 4#c2 8b1 16#a1 8#g1 2#f2 4#c2 8b1 16#a1 8b1 2#g1 4#c1 16#c1 2#d1 8#c2 8b1 8#a1 8#g1 8#f1 16#f1 8#g1 16#a1 4#g1 2- "

# Final countdown
melody += "4- 8- 16b2 16a2 4b2 4e2 4- 8- 16c3 16b2 8c3 8b2 4a2 4- 8- 16c3 16b2 4c3 4e2 4- 8- 16a2 16g2 8a2 8g2 8#f2 8a2 4g2 8- 16#f2
16g2 4a2 8- 16g2 16a2 8b2 8a2 8g2 8#f2 4e2 4c3 2b2 4- 16b2 16c3 16b2 16a2 1b2 2- "

# Phanthom of the Opera
melody += "4e1 4a1 4e1 4g1 8f1 2f1 4d1 4g1 8d1 1e1 4e1 4a1 4e1 4g1 8f1 2f1 4d1 4g1 8d1 1e1 4e1 4a1 4c2 4e2 8d2 2d2 4d2 4g2 8d2 1e2 4e2
1a2 8g2 8f2 8e2 8d2 8c2 8b1 8a1 1#g1 4f1 4f1 8e1 1e1 2- "

duration_dict = {
    "64.": b'\x00\x30',
    "64":  b'\x00\x20',
    "32.": b'\x00\x60',
    "32":  b'\x00\x40',
    "16.": b'\x00\xBB',
    "16":  b'\x00\x7D',
    "8.":  b'\x01\x77',
    "8":   b'\x00\xFA',
    "4.":  b'\x02\xEE',
    "4":   b'\x01\xF4',
    "2.":  b'\x05\xDC',
    "2":   b'\x03\xE8',
    "1.":  b'\x0B\xB8',
    "1":   b'\x07\xD0',
}

frequency_dict = {
    "#c1": b'\x07\x0B',
    "c1":  b'\x07\x77',
    "#d1": b'\x06\x47',
    "d1":  b'\x06\xA6',
    "e1":  b'\x05\xEC',
    "#f1": b'\x05\x47',
    "f1":  b'\x05\x97',
    "#g1": b'\x04\xB3',
    "g1":  b'\x04\xFB',
    "#a1": b'\x04\x30',
    "a1":  b'\x04\x70',
    "b1":  b'\x03\xF4',
    "#c2": b'\x03\x85',
    "c2":  b'\x03\xBB',
    "#d2": b'\x03\x23',
    "d2":  b'\x03\x53',
    "e2":  b'\x02\xF6',
    "#f2": b'\x02\xA3',
    "f2":  b'\x02\xCB',
    "#g2": b'\x02\x59',
    "g2":  b'\x02\x7D',
    "#a2": b'\x02\x18',
    "a2":  b'\x02\x38',
    "b2":  b'\x01\xFA',
    "#c3": b'\x01\xC2',
    "c3":  b'\x01\xDD',
    "#d3": b'\x01\x91',
    "d3":  b'\x01\xA9',
    "e3":  b'\x01\x7B',
    "#f3": b'\x01\x51',
    "f3":  b'\x01\x65',
    "#g3": b'\x01\x2C',
    "g3":  b'\x01\x3E',
    "#a3": b'\x01\x0C',
    "a3":  b'\x01\x1C',
    "b3":  b'\x00\xFD',
    "-":   b'\x00\x00',
}

melody = melody.strip()
notes = melody.split(" ")
counter = 0

with open("rom.mem", "w") as file:

    for note in notes:

        print(f"{counter}\t{note:6s}\t", end="")
        counter += 1

        half_period_hex = None;
        duration_hex = None;

        for frequency in frequency_dict:
            if frequency in note:
                note_without_frequency = note.replace(frequency, "")
                half_period_hex = frequency_dict[frequency]
                break;

        for duration in duration_dict:
            if duration in note_without_frequency:
```


Poprzednie części kursu i dodatkowe materiały dostępne są na stronie
https://ulubionykiosk.pl/media

eprasa.pl 65e3eb482b

```
                duration_hex = duration_dict[duration]
                break;

        print(f"{duration_hex[0]:02X}{duration_hex[1]:02X}
                {half_period_hex[0]:02X}{half_period_hex[1]:02X}")

        file.write(f"{duration_hex[0]:02X} ")
        file.write(f"{duration_hex[1]:02X} ")
        file.write(f"{half_period_hex[0]:02X} ")
        file.write(f"{half_period_hex[1]:02X} ")
        file.write(f"// {note:6s}\n")

    file.write("00 00 00 00 // end\n")
```

Listing 7. Kod pliku converter.py

Tabela 1. Lista wszystkich możliwych długości nut

Notacja	Czas [ms]	HEX
64	32	0020
64.	48	0030
32	64	0040
32.	96	0060
16.	187	00BB
16	125	007D
8	250	00FA
8.	375	0177
4	500	01F4
4.	750	02EE
2	1000	03E8
2.	1500	05DC
1	2000	07D0
1.	3000	0BB8

zaadaptować format zapisu stosowany w starych telefonach Nokia. Każdy mógł wówczas skomponować sobie dowolną melodię, wpisując nuty z klawiatury w postaci liter i cyfr, a na szczęście, w internecie wciąż dostępna jest cała masa stron z mnóstwem melodii gotowych do użytku.

Najpierw jednak zobaczmy, jak zapisywane są poszczególne nuty. Notacja zapisu jest bardzo prosta i składa się z dwóch części. Pierwsza to czas trwania dźwięku. Wszystkie możliwe czasy zebrano w **tabeli 1**. Druga część to częstotliwość dźwięku dla wszystkich nut i ich półtonów z trzech oktaw. Aby nie wchodzić zbytnio w zawiłości muzyczne, częstotliwości wszystkich obsługiwanych dźwięków pokazano w **tabeli 2**. Łącząc dowolne dwie części z tych tabel, możemy uzyskać dowolne nuty, a poszczególne nuty w melodii powinny być oddzielone spacją. W ten sposób melodię mamy zapisaną za pomocą znaków ASCII. Wystarczy tylko opracować jakiś skrypt, który przekonwertuje zapisy nut na bitstream zrozumiały dla naszego modułu.

Zobaczmy kod zaprezentowany na **listingu 7**. Jest to skrypt w Pythonie, który konwertuje melodyjki zapisane w notacji z telefonów Nokia na plik pamięci rom.mem, inicjalizujący pamięć ROM. Ponieważ jest to kurs języka Verilog, a nie Python, nie będziemy szczegółowo analizować tego kodu (zresztą jest on bardzo trywialny, nie ma żadnej obsługi błędów ani możliwości konfiguracji).

Melodyjki dodajemy na początku pliku do zmiennej **melody**. Umieściłem tam kilka przykładowych melodii, które będą rozpoznawane przez większość Czytelników. Możemy dodać ich dowolną liczbę, byle tylko sumaryczna liczba wszystkich nut nie przekroczyła 1023. W tym miejscu można dodać melodie, których mnóstwo znajdą Czytelnicy w internecie. Warto tylko zwrócić uwagę, że po każdej nucie – łącznie z ostatnią – trzeba umieścić spację.

Następnie mamy dwa słowniki dla każdej możliwej opcji długości i częstotliwości dźwięku. W tych słownikach etykietom tekstowym

Tabela 2. Lista wszystkich nut możliwych do odtworzenia

Notacja	Częstotliwość [Hz]	HalfPeriod [us]	HEX
c1	261,63	1911	0777
#c1	277,18	1803	070B
d1	293,66	1702	06A6
#d1	311,13	1607	0647
e1	329,63	1516	05EC
f1	349,23	1431	0597
#f1	369,99	1351	0547
g1	392,00	1275	04FB
#g1	415,30	1203	04B3
a1	440,00	1136	0470
#a1	466,16	1072	0430
b1	493,88	1012	03F4
c2	523,25	955	03BB
#c2	554,37	901	0385
d2	587,33	851	0353
#d2	622,25	803	0323
e2	659,26	758	02F6
f2	698,46	715	02CB
#f2	739,99	675	02A3
g2	783,99	637	027D
#g2	830,61	601	0259
a2	880,00	568	0238
#a2	932,33	536	0218
b2	987,77	506	01FA
c3	1046,50	477	01DD
#c3	1108,73	450	01C2
d3	1174,66	425	01A9
#d3	1244,51	401	0191
e3	1318,51	379	017B
f3	1396,91	357	0165
#f3	1479,98	337	0151
g3	1567,98	318	013E
#g3	1661,22	300	012C
a3	1760,00	284	011C
#a3	1864,66	268	010C
b3	1975,53	253	00FD
–	pauza	0	0000

przyporządkowano odpowiadający im bitstream. Dalej znajduje się pętla iterująca po wszystkich nutach po kolei, która wyszukuje fragmenty tekstu w słownikach i dopasowuje do nich odpowiednie bajty. Wszystko to jest później zapisywane w pliku **rom.mem**.

Aby uruchomić ten skrypt, otwieramy konsolę i wpisujemy polecenie **python converter.py**.

Mamy już wszystko, czego potrzebujemy. Syntezujemy, konfigurujemy piny IO w taki sposób, jak uwidoczniono na **rysunku 7** – i wgrywamy bitstream do FPGA. Po naciśnięciu górnego enkodera usłyszymy melodię. Naciśnięcie drugiego enkodera powoduje przerwanie odtwarzania.

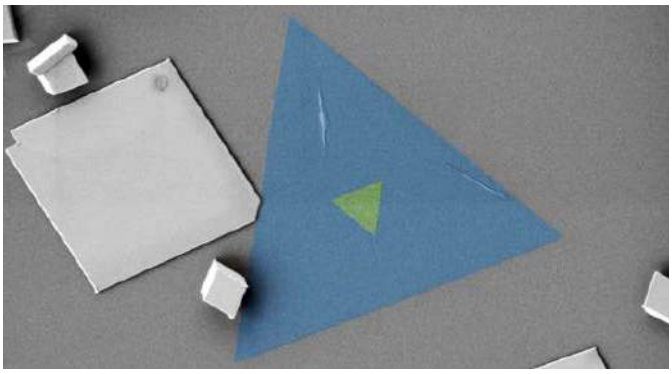
W następnym odcinku prześledzimy, jak przesyłać dane z FPGA do innego układu poprzez popularny interfejs UART.

Dominik Bieczyński
leonow32@gmail.com

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie **https://ulubionykiosk.pl/media**

eprasa.pl 65e3eb482b

koktajl niusów



W poszukiwaniu nowych technologii materiałowych dla elektroniki

Pod takim hasłem przedstawiamy projekt „Skalowalna, niskotemperaturowa synteza dwuwymiarowych półprzewodników do nowatorskich zastosowań w elektronice", który ma rozwiązać problem syntezy materiałów elektronicznych 2D w dość znacznych temperaturach. Według założeń projektu, realizowanego przez Politechnikę Warszawską (PW), przygotowana zostanie niskotemperaturowa technologia bazująca na technice MOCVD, związanej z osadzaniem cienkich warstw na powierzchni różnych materiałów poprzez użycie związków metaloorganicznych w fazie pary. Metoda ta powinna dostarczać dwuwymiarowe struktury półprzewodników, na podłożach zgodnych z wymogami przemysłu (w tym elastycznych), o jakości dorównującej podłożom wytwarzanym w tradycyjnych procesach wysokotemperaturowych. Zrozumienie dokładnych mechanizmów przyrostu struktur pozwoli lepiej kontrolować te procesy, jak również udoskonalić protokoły syntezy oraz jakość materiałów. Jest to krok w stronę budowy wydajniejszych urządzeń elektronicznych, odblokowania obecnie niedostępnych aplikacji i redukcji poboru mocy. Opisywany projekt zakończy się demonstracją praktycznego zastosowania i jest podzielony na 2 części: technologiczną oraz naukową.

https://tiny.pl/dqcnt

Kompaktowy dalmierz laserowy COOLSHOT 20 GIII firmy Nikon do gry w golfa

Od pierwszego uderzenia piłki, aż po końcowy dołek – najnowszy dalmierz COOLSHOT 20 GIII pomaga każdemu uzyskać pomiary najważniejszych odległości podczas rozgrywki. Kompaktowe urządzenie polecane jest graczom, którzy pragną szybkich, prostych oraz dokładnych pomiarów. Dostępne w dalmierzu laserowym COOLSHOT 20 GIII funkcje priorytetu pierwszego planu oraz LOCKED-ON QUAKE pozwalają wyeliminować jakąkolwiek niepewność. Samo urządzenie delikatnie wibruje, potwierdzając precyzyjny pomiar odległości do drzewca chorągiewki, a nie obecnych za nią drzew. Przewidziany zakres pomiarowy dalmierza, wynoszący 5...730 m, sprawia, że COOLSHOT 20 GIII zdolny jest do wspierania każdego uderzenia zawodnika. W prosty sposób można wybierać preferowany typ pomiaru – wystarczy nacisnąć przycisk trybu w celu wygodnego przełączania się pomiędzy jardami i metrami. Realizowana przez dalmierz funkcja pomiaru ciągłego ułatwia również skanowanie terenu, kiedy zachodzi taka potrzeba. Wystarczy przytrzymać przycisk zasilania przez maksymalnie 8 s, żeby szybko, wręcz płynnie mierzyć odległości do przeszkód. Nie trzeba przy tym zdejmować okularów, by móc namierzać cel. Dalmierz laserowy COOLSHOT 20 GIII umożliwia znaczne odsunięcie źrenicy wyjściowej, dzięki czemu jest wygodny w użyciu dla każdego. Monokular o powiększeniu 6×, z wielowarstwowymi powłokami, zapewnia jasny oraz wyraźny obraz, a regulacja dioptrażu umożliwia precyzyjne ustawianie ostrości. Wszystko to przy rewelacyjnie prostej konfiguracji, a także obsłudze. Urządzenie jest niewiele większe niż smartfon i doskonale leży w dłoni podczas gry, dobrze znosząc nawet lekki deszcz.

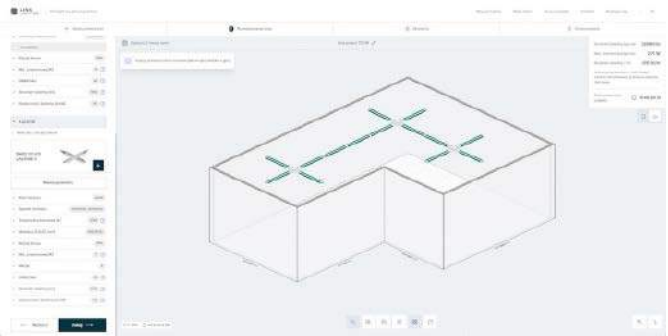
https://tiny.pl/dqcnr

Firma Bosch prezentuje system automatycznego ładowania pojazdów elektrycznych

Na parkingu wyposażonym w automatyczne systemy parkowania pojazdy elektryczne mogą łatwo znaleźć wolne miejsce postojowe z dostępem do ładowania. Kierowca ustawia pojazd w specjalnej strefie przejściowej w pobliżu wjazdu do garażu i – za pomocą aplikacji – uruchamia zautomatyzowaną usługę parkowania autonomicznego. Związana z systemem automatycznego ładowania pojazdów elektrycznych infrastruktura zaprowadza pojazd do miejsca parkingowego, na którym robot otwiera klapkę ładowania oraz mocuje kabel ładujący. Kiedy baterie „elektryka" zostaną w pełni naładowane, wspomniany robot demontuje kabel. Następnie samochód odjeżdża z powrotem na klasyczne miejsce parkingowe, zwalniając miejsce dla kolejnego pojazdu. Cały proces realizowany jest bez udziału człowieka. Firma Bosch przetestowała opisywany system w 2 lokalizacjach. Technologię wdrożenia funkcji automatycznego ładowania zaprezentowała także w czasie targów CES 2024. Opracowany przez niemiecką markę system doskonale poprawia komfort użytkowników pojazdów elektrycznych. Przynosi on znaczące korzyści dla producentów tych pojazdów, operatorów stacji ładowania oraz dostawców energii. Z punktu widzenia środowiska samochody elektryczne oznaczają natomiast redukcję emisji dwutlenku węgla, która ściśle wiąże się z ruchem drogowym.

https://tiny.pl/dqcnd

eprasa.pl 65e3eb482b



Innowacyjny konfigurator oświetlenia Lena Lighting już dostępny

Wprowadzony do użytku konfigurator oświetlenia pozwala na odpowiednie rozplanowanie instalacji produktów Lena Lighting w pomieszczeniach. Tym sposobem klienci otrzymują opcję projektowania wnętrz z myślą o właściwym rozlokowaniu lamp – tak, żeby idealnie wkomponowywały się w aranżację dowolnej przestrzeni. Oferowany przez Lena Lighting konfigurator błyskawicznie dopasowuje akcesoria kompatybilne z określonymi modelami źródeł światła. Jest to kolejny istotny krok w stronę spersonalizowanego podejścia do oświetlenia, które upraszcza proces konfiguracji i czyni go bardziej intuicyjnym. Użytkownicy konfiguratora mogą tworzyć wizualizacje wnętrz, określać parametry lampy oraz projektować linie świetlne, używając do tego celu łączników. Aplikacja przygotowuje informacje o dostępnościach produktów oraz ich cenach. Po zakończeniu konfiguracji użytkownicy mogą skontaktować się z obsługą klienta, przesyłając wygenerowane w ten sposób zapytanie ofertowe.

https://tiny.pl/dqcn5

Wyjątkowy gadżet: reklamowy, plastikowy pendrive DS-1409 od MemoRabbit

Model DS-1409 plastikowego, reklamowego pendrive'a z pewnością zaskoczy każdego rodzica. Łączy on w sobie pamięć USB, kartę kredytową oraz puzzle, które stanowią nie lada gratkę dla każdego malucha. Jest to przede wszystkim oryginalny i atrakcyjny sposób na reklamę firm, ponieważ to właśnie ich logotypy mogą być układane z przesuwanych elementów. Dostarczany przez MemoRabbit pendrive DS-1409 o wymiarach 8,5×5,5×0,3 cm pozwala na wygodne przechowywanie i przenoszenie danych. Budowa pendrive'a umożliwia noszenie akcesorium w bezpieczny sposób w portfelu, a jednocześnie sprzyja rozrywce. Produkt skierowany jest nie tylko do dzieci, ponieważ układanie puzzli może okazać się świetnym sposobem na zabicie czasu również w przypadku dorosłych użytkowników. Jest to zarazem niezwykle skuteczny chwyt marketingowy – bawiąc się oraz układając przesuwane elementy, klienci o wiele szybciej zapamiętują daną markę.

https://memorabbit.pl/product/ds-1409

Bariery w zastosowaniu energii odnawialnej można pokonać dzięki zastosowaniu nowoczesnych technologii. Najnowsze badanie Farnell

Najnowsze badanie opinii przeprowadzone przez firmę Farnell dowiodło, że – choć poziom świadomości konsumentów, związany z potrzebą zwiększenia wykorzystania źródeł energii odnawialnej, jest wysoki – wciąż nierozstrzygnięte pozostają pytania, jak zmniejszyć utrzymującą się zależność od paliw kopalnych. Wielu respondentów wskazało, że barierami w przejściu na źródła energii odnawialnej są: utrzymujący się na rynku wpływ firm zajmujących się paliwami kopalnymi, brak właściwej technologii zasilania akumulatorowego, znaczące koszty modernizacji infrastruktury, a także sama

„technologia odnawialna". Jeden z ankietowanych odpowiedział, że wybrane źródła energii odnawialnej, np. energia wiatrowa i słoneczna, to źródła nieciągłe, co może prowadzić do wzrostu kosztów powiązanych z wdrażaniem rozwiązań przeciwdziałających okresowym brakom w zaopatrzeniu w energię oraz wykluczających ryzyko tego typu. Oprócz tego część badanych podziela przekonanie, jakoby istniejące od lat uzależnienie od infrastruktury opartej na paliwach kopalnych utrudniało przejście na źródła energii odnawialnej (OZE) oraz czyniło taką zmianę bardzo kosztowną. Inni respondenci podkreślili, że bariery w przejściu na OZE mogą się różnić w zależności od miejsca na świecie. Powodem tego są różnice m.in. w polityce rządowej, poziomie społecznego zaangażowania w inicjatywy ekologiczne czy zróżnicowanie poziomów planowania określonych projektów infrastruktury oraz inwestowania w te projekty. Niektórzy badani stwierdzili również, że powszechne wdrożenie technologii pojazdów z napędem elektrycznym oraz ogrzewania elektrycznego wymaga sporego nakładu prac inżynieryjnych. Przykładowo – liczba stacji ładowania samochodów elektrycznych nieustannie wzrasta, jednak wiele spośród nich jest niezdatnych do używania, ponieważ nie mogą zostać przyłączone do sieci elektrycznej z powodu braku odpowiedniego zasilania. Ten problem może wpłynąć negatywnie na poziom zainteresowania inwestowaniem w rozwiązania z branży pojazdów elektrycznych (EV).

https://tiny.pl/dqcnn

Firma Asseco usprawnia wymianę danych rozpoznawczych w ramach koalicji NATO

Asseco – jako pierwsza firma w Polsce – wprowadziła do oferty rozwiązanie, które umożliwia przesyłanie danych na wojskowych serwerach CSD oraz korzystanie z nich – za pomocą oprogramowania analitycznego. Rozwiązanie CSD_eP, oparte na platformie ArcGIS Pro, zapewnia gromadzenie, przechowywanie i udostępnianie danych rozpoznawczych w ramach koalicji NATO. Systemy CSD (ang. Coalition Shared Data) pozwalają na udostępnianie między stronami danych rozpoznawczych, np. zdjęć satelitarnych, raportów, map oraz materiałów wideo z dronów. Rozwiązanie Asseco zintegrowane jest z innowacyjnym oprogramowaniem Esri ArcGIS pro. Już z poziomu jednej aplikacji specjaliści wojskowi mogą dostać się do serwera CSD, wyszukać czy pobrać określone materiały, wykonać pracę analityczną, generować raporty i przesyłać je z powrotem na serwer. Takie rozwiązanie udoskonala pracę i pozwala na bieżącą wymianę strategicznych informacji w ramach paktu NATO. Rozwiązanie Asseco było

eprasa.pl 65e3eb482b

testowane w czasie ćwiczeń NATO CWIX i otrzymało akredytację do działań w sieciach tajnych. Była to niepowtarzalna okazja do doskonalenia sposobów wymiany istotnych informacji, a wypracowane w czasie ćwiczeń innowacje pozwalają obecnie udoskonalać dostępne technologie i budować kompletnie nowe systemy, zmieniające sposób działania na polu walki.

https://tiny.pl/dqcnk

983 oprawy drogowe VIA rozświetlają Komprachcice

Firma Miloo-Electronics wsparła gminę Komprachcice (woj. opolskie) w zadaniu modernizacji oświetlenia drogowego. Chodzi tutaj o montaże 983 nowoczesnych opraw drogowych VIA, których jakość oraz efektywność potwierdzają certyfikaty: ENEC i ENEC+. Oprawa drogowa VIA powstała jako produkt przeznaczony dla miast i gmin, które borykają się z rosnącym problemem lawinowego przyrostu cen energii elektrycznej. Wyróżnia się ona układem optycznym niwelującym olśnienie, a także zastosowaniem wytrzymałej szyby hartowanej. Znamiennym elementem oprawy drogowej VIA są wysokowydajne moduły LED o skuteczności świetlnej 155 lm/W, które zapewniają żywotność powyżej 100 tysięcy godzin pracy. Gwarantowany jest bardzo prosty montaż oprawy oraz pełna jej kompatybilność z obecnymi na rynku systemami inteligentnego sterowania – tzw. Smart City. Produkt spełnia wszelakie wymogi dotyczące bezpieczeństwa energetycznego. Wspomniany wcześniej certyfikat ENEC+ oznacza wiarygodność parametrów funkcjonalnych, deklarowanych w chwili wprowadzania produktów na rynek. Są to w szczególności: trwałość, skuteczność świetlna, rozsył światłości, prąd i napięcie zasilania, strumień świetlny, maksymalna temperatura pracy, temperatura barwowa, wskaźnik oddawania barw oraz moc.

https://tiny.pl/dqcn6

Złośliwy kod najczęściej spotykany w fałszywych grach, m.in. Roblox

Specjaliści ds. bezpieczeństwa firmy ESET wykryli złośliwy kod w różnych fałszywych wersjach gier mobilnych. Komunikują oni, że może to być strategia cyberprzestępców zagrażająca wszystkim – w tym dzieciom. Wśród zauważonych zagrożeń znalazł się szereg fałszywych aplikacji na platformę Android, za pośrednictwem których atakujący przesyłają adware, a także złośliwe oprogramowanie. Zgodnie z ustaleniami ekspertów z firmy ESET, do ataków najczęściej stosowane są właśnie fałszywe wersje znanych gier, a przestępcy stosują tego rodzaju strategię w okresach, kiedy dzieci częściej sięgają po mobilną rozrywkę, czyli w wakacje i święta. Eksperci zwrócili uwagę m.in. na obecność downloadera Agent.CZB w zmodyfikowanej wersji popularnego Robloxa oraz złośliwy kod Spy.SpinOk (z funkcjami szpiegowskimi) w zmodyfikowanych aplikacjach do pobierania muzyki i filmów oraz w aplikacjach do przesyłania plików. Celem downloadera Agent.CZB jest pobieranie oraz uruchamianie występujących w internecie złośliwych kodów, jak również przejmowanie kontroli nad smartfonem czy kradzieże informacji z urządzenia. Ze względu na środki zaradcze, podejmowane przez internautów świadomych zagadnień związanych ze złośliwym oprogramowaniem, atakujący przymuszani są do ciągłych modyfikacji złośliwych wersji gier, co z kolei przyczynia się do stałych zmian w krajobrazie cyberbezpieczeństwa platformy Android.

https://tiny.pl/dqck7

Firma ASTOR i jej myśl inżynierska oraz menedżerska docenione w Japonii

Współpraca ASTOR z japońskim przemysłem ma własną historię. Jej początki sięgają 2007 roku, kiedy na mocy podpisanej umowy dystrybucyjnej, na polski rynek wprowadzone zostały pierwsze roboty Kawasaki Robotics. Zawarcie umowy z lokalnym dystrybutorem było częścią strategii europejskiej ekspansji Kawasaki Robotics i miało na celu wzmocnienie marki w Europie Środkowej oraz Wschodniej. Współpraca firmy ASTOR z Kawasaki Robotics na przestrzeni wielu lat ewoluowała, czego potwierdzeniem są m.in. nagrody na szczeblu europejskim. W 2020 roku w Berlinie podpisany został kontrakt, na podstawie którego firma ASTOR stała się operatorem Kawasaki Robotics CEE Hub, obejmującego 12 państw z regionu CEE. Nie tak dawna wizyta przedstawicieli marki w Japonii była doskonałą okazją do tego, żeby odwiedzić targi iREX 2023, uznawane za największe na świecie pokazy robotyki i rozwiązań autonomicznych. Spora liczba wystawców (654) związanych z sektorem automatyzacji, robotyzacji oraz inżynierii przemysłowej i imponująca frekwencja (aż 150 000 odwiedzających tylko w trzy pierwsze dni) podkreślają znaczenie tego wydarzenia. Japońscy producenci odpowiadają za ok. 40% sprzedaży robotów na całym świecie. Aby zrozumieć globalny rynek robotyki, niezbędne jest zatem spojrzenie na Japonię, gdyż tamtejszy rynek jest kolebką innowacji w zakresie kierunków prac badawczo-rozwojowych w zakresie robotyki.

https://tiny.pl/dqck9

Udoskonalona wersja modułu I/O-Link SLIO od Yaskawa

Od teraz moduł SLIO I/O-Link umożliwia dogodną komunikację pomiędzy procesorami a urządzeniami obiektowymi, które wspierają I/O-Link. Moduł można uruchamiać na dwa sposoby: bezpośrednio z procesora CPU bądź jako moduł dołączony do wyspy zdalnej. Moduł SLIO I/O-Link pracuje jako master I/O-Link – pozwala na podłączenie maksymalnie 6 masterów, a przez to – nawet do 24 urządzeń I/O-Link na moduł główny. Od strony sprzętowej urządzenie obejmuje 4 parametryzowalne, 64-bajtowe kanały, które mogą pracować w trybie standardowego wejścia/wyjścia (SIO) lub w trybie I/O-Link. Diody LED stanu w module wskazują bieżący tryb, a także wszelkie występujące błędy. Porty modułu są izolowane galwanicznie od magistrali płyty bazowej, a moduły główne I/O-Link są konfigurowane, kontrolowane i aktualizowane za pomocą oprogramowania I/O-Link Manager. Pliki IODD ładowane są poprzez serwer I/O-Link i nie mają kodów wykonywalnych. Oprogramowanie komunikuje się z jednym lub większą liczbą podłączonych modułów głównych I/O-Link za pośrednictwem sieci Ethernet, może też odczytywać ich konfigurację. Co więcej, użytkownicy mają możliwość modyfikowania konfiguracji poprzez interfejs GUI, zapisywania jej na nośniku danych i ponownego zapisywania do modułu poprzez Ethernet.

https://tiny.pl/dqckl

Jakub Tyburski
jakub.tyburski@elportal.pl

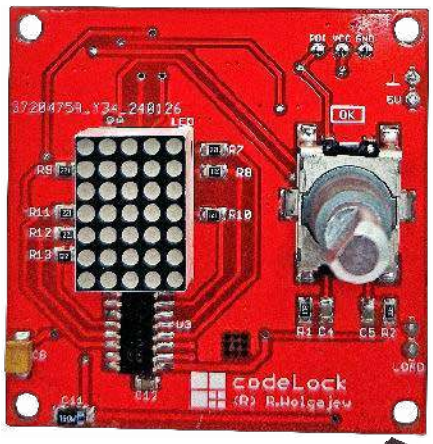
eprasa.pl 65e3eb482b

Hity następnego numeru • „Elektronika Praktyczna” 4/2024

codeLock – efektowny zamek kodowy

Zamki kodowe – podobnie jak termometry, termostaty i miniaturowe radyjka – to niewątpliwie „elementarz” każdego elektronika amatora. Któż z nas nie ma w swoim portfolio urządzeń tego typu, które mimo oczywistej prostoty dają dużo radości z własnoręcznego wykonania użytecznej konstrukcji? W artykule opisujemy prosty zamek kodowy, który dzięki zastosowanym rozwiązaniom programowym oraz sprzętowym przywodzi na myśl pokrętła sejfów bankowych, znanych z produkcji filmowych.

Sercem układu jest niewielki mikrokontroler z rodziny AVR ATtiny, wspomagany przez garść elementów dyskretnych i zaledwie trzy dodatkowe układy scalone. Wbudowana matryca LED 5×7 pozwala na wyświetlanie cyfr przesuwających się synchronicznie z obrotami enkodera inkrementalnego, co w prosty sposób symuluje działanie bębna zamka sejfowego, zaś niewielki głośniczek SMD generuje dźwięki „tykania”, jeszcze wyraźniej podkreślające źródło inspiracji Autora projektu.

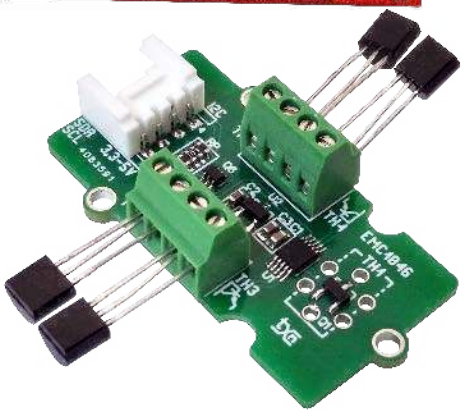


Zaawansowany selektor audio

Typowe selektory sygnału audio posiadają zazwyczaj zestaw wejść i tylko jedno wyjście, przełączane pomiędzy poszczególnymi kanałami wejściowymi. Sytuacja komplikuje się, kiedy potencjalnych adresatów owego sygnału również jest kilku. W takich właśnie przypadkach może przydać się opisany w artykule układ.

Najważniejsze parametry urządzenia:

- cztery wejścia stereo na złączach żeńskich RCA (Cinch),
- cztery wyjścia stereo na złączach żeńskich RCA (Cinch),
- możliwość swobodnego wyboru pary wejście-wyjście,
- obsługa przy pomocy wyświetlacza LCD 2×16 oraz czterech przycisków.



Pięciokanałowy termometr I²C

W niektórych aplikacjach przydatny okazuje się niezależny pomiar temperatury w kilku punktach urządzenia. Na rynku dostępne są wprawdzie czujniki temperatury z interfejsem 1-Wire czy I²C, ułatwiające prostą realizację małej sieci termometrów, problemem może być jednak ich koszt i – nierzadko – obecne w sprzedaży podróbki oryginalnych układów.

Przedstawione urządzenie pozwala zastosować w roli czujników „najzwyczajniejsze” tranzystory krzemowe NPN typu 2N3904, dzięki czemu nawet kilka kanałów pomiarowych można zrealizować niewielkim kosztem i to praktycznie z elementów zalegających w przepastnych szufladach warsztatu elektronicznego.

Temat numeru: wyświetlacze EPD

Wyświetlacze typu E-Paper (E-Ink, EPD) wyróżniają się na tle wszystkich konkurencyjnych technologii obrazowych przede wszystkim pod względem fenomenalnych osiągów w zakresie oszczędności energii. Wady pierwszych generacji wyświetlaczy EPD stopniowo odchodzą do lamusa – powolne ekrany o częstotliwości odświeżania na poziomie nie przekraczającym kilku herców ustępują znacznie nowocześniejszym modułom, które oferują już szybkość reakcji umożliwiającą nawet wyświetlanie prostych animacji. Monochromatyczne ekrany o przeciętnej jakości obrazu są szybko wypierane przez wyświetlacze oferujące kilka barw, a rynkową atmosferę podgrzewają kolejne doniesienia o panelach oferujących fenomenalną wręcz jakość wyświetlania pełnokolorowych fotografii oraz elementów interfejsu użytkownika.

Elektronika w Praktyce: instalacje nisko- i wysokoprądowe

W kwietniowym numerze „Elektroniki Praktycznej” przyjrzymy się zagadnieniom leżącym na styku elektroniki oraz elektryki. Szeroko omówimy rodzaje przewodów do różnych typów instalacji oraz stosowane w nich materiały izolacyjne, podamy także szereg przydatnych informacji dotyczących doboru okablowania zasilającego w różnych aplikacjach. Zaprezentujemy ponadto zagadnienia złączy zaciskanych i przeznaczonych do nich narzędzi instalatorskich, zaś dopełnieniem materiału będą prezentacje rozmaitych rozwiązań w zakresie osłon oraz uchwytów do przewodów i wiązek kablowych.

Wykaz firm ogłaszających się w tym numerze „Elektroniki Praktycznej”

AKSOTRONIK 21
BL ELEKTRONIK 80, 81
BORNICO 9
COMPUTER CONTROLS 13
ELMAX 57
ELTY 48, 49
FAULHABER 63
FERYSTER 29
GLYN 45, 54
HAMMOND 7
LASTENIC LASER 23
MICROCHIP 11, 52, 116
MICROS 5, 33
SEMICON 64, 65
TME 36, 37
WG ELECTRONICS 43, 50

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. Leszczynowa 11
tel. 22 257 84 99, e-mail: avt@avt.pl

Wydawca:
Wiesław Marciniak

Adres redakcji:
03-197 Warszawa, ul. Leszczynowa 11
e-mail: redakcja@ep.com.pl, www.ep.com.pl

Redaktor Naczelny:
Przemysław Musz

Redaktor Programowy, Przewodniczący Rady Programowej:
Piotr Zbysiński

Menedżer Magazynu:
Katarzyna Gugała, tel. 22 257 84 64

Szef Pracowni Konstrukcyjnej:
Jakub Sobański

Zespół marketingu i reklamy:
Katarzyna Gugała, Bożena Krzykawska, Grzegorz Krzykawski, Grzegorz Lalak

Stali współpracownicy:
Lucjan Bryndza, Nikodem Czechowski, Jarosław Doliński, Andrzej Gawryluk, Krzysztof Górski, Tomasz Jabłoński, Paweł Kowalczyk, Henryk Kowalski, Rafał Kozik, Michał Kurzela, Szymon Panecki, Damian Sosnowski, Ryszard Szymaniak, Adam Tatuś, Jakub Tyburski, Robert Wołgajew

Uwaga!
Kontakt z wymienionymi osobami jest możliwy via e-mail, według schematu: imię.nazwisko@ep.com.pl

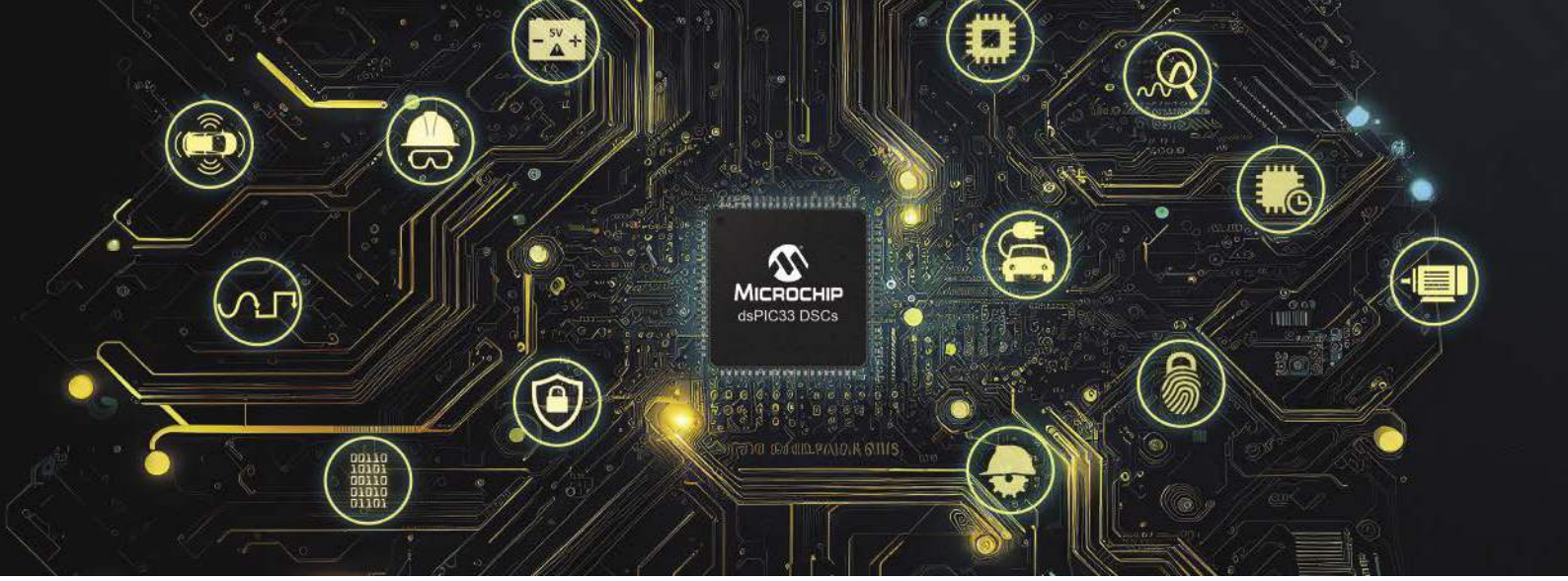
DTP, okładka, redakcja strony internetowej www.ep.com.pl:
MAD Sp. z o.o.

Prenumerata w Wydawnictwie AVT
www.ulubionykiosk.pl lub tel. 22 257 84 22 (godz. 10.00–14.00)
e-mail: prenumerata@avt.pl

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

Copyright AVT-Korporacja Sp. z o.o.
03-197 Warszawa, ul. Leszczynowa 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”.

eprasa.pl 65e3eb482b



dsPIC® DSC: właściwy wybór do systemów sterowania w czasie rzeczywistym

Moc DSP i prostota MCU w jednym

Kontrolery sygnałowe dsPIC® (DSC) zostały zaprojektowane tak, aby zapewnić mikrokontrolerom (MCU) wydajność i funkcje charakterystyczne dla procesora sygnałowego (DSP), wykorzystując wyspecjalizowaną architekturę i zestaw instrukcji.

Z biegiem czasu DSC ewoluowały, stając się wysoce wydajnymi i elastycznymi kontrolerami do zastosowań związanych ze sterowaniem w czasie rzeczywistym, dzięki czemu układy te nadają się do szerokiego zakresu zastosowań w motoryzacji, przemyśle, energetyce, sterowaniu silnikami i zaawansowanym interfejsach, a także zastosowaniach dotykowych. Najnowsze procesory DSC dsPIC33 zawierają ulepszone funkcje i wysokowydajne układy peryferyjne wspierające nowe technologie i projekty, co czyni je idealnym wyborem dla projektantów zaawansowane aplikacji czasu rzeczywistego o następujących cechach:

- **Małe opóźnienia:** aplikacje czasu rzeczywistego wymagają procesorów DSC dsPIC dsPIC o małych opóźnieniach i szybkiej odpowiedzi deterministycznej.
- **Moc przetwarzania:** procesory DSC dsPIC pracują z częstotliwością do 100 MHz z akceleracją operacji matematycznych i dużą przepustowością.
- **Wysokowydajne układy peryferyjne:** zoptymalizowana wydajność dzięki bogatemu zestawowi wydajnych aplikacyjnych układów peryferyjnych.
- **Bezpieczeństwo:** Szerokie portfolio układów klasy safety-ready i functional safety-compliant zgodnych z bezpieczeństwem funkcjonalnym i z bibliotekami oprogramowania diagnostycznego.
- **Ochrona:** bezpieczne rozwiązania wykorzystujące moduły DSC dsPIC umożliwiają ochronę systemu przed wieloma złośliwymi atakami.
- **Środowisko:** płytki ewaluacyjne, płytki rozwojowe, pakiety oprogramowania, konfigurator urządzeń i kodu, IDE, debugger i projekty referencyjne.

MICROCHIP

microchip.com/dsPIC-DSCs

Nazwa i logo Microchip oraz logo Microchip są zastrzeżonymi znakami towarowymi firmy Microchip Technology Incorporated w USA i innych krajach. Wszystkie pozostałe znaki towarowe są własnością ich zarejestrowanych właścicieli.
© 2024 Microchip Technology Inc.
Wszelkie prawa zastrzeżone. MEC2540A-POL-02-24

eprasa.pl 65e3eb482b