ELEKTRONIKA PRAKTYCZNA

EP.com.pl

● Międzynarodowy magazyn elektroników konstruktorów ● maj ● 5/2024 ●

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

• Nakładka z transceiverem CAN do AVTduino UNO R4 Plus • cMeter – przenośny miernik pojemności • Konwerter napięcia stałego na pętlę prądową 4...20 mA • Moduł precyzyjnego zegara RTC z interfejsem Grove • 24-bitowy sprzętowy licznik impulsów z interfejsem I²C • Ministerownik silnika komutatorowego małej mocy • Radioodbiornik internetowy z dekoderem VS1053

podzespoły, sprzęt, aplikacje

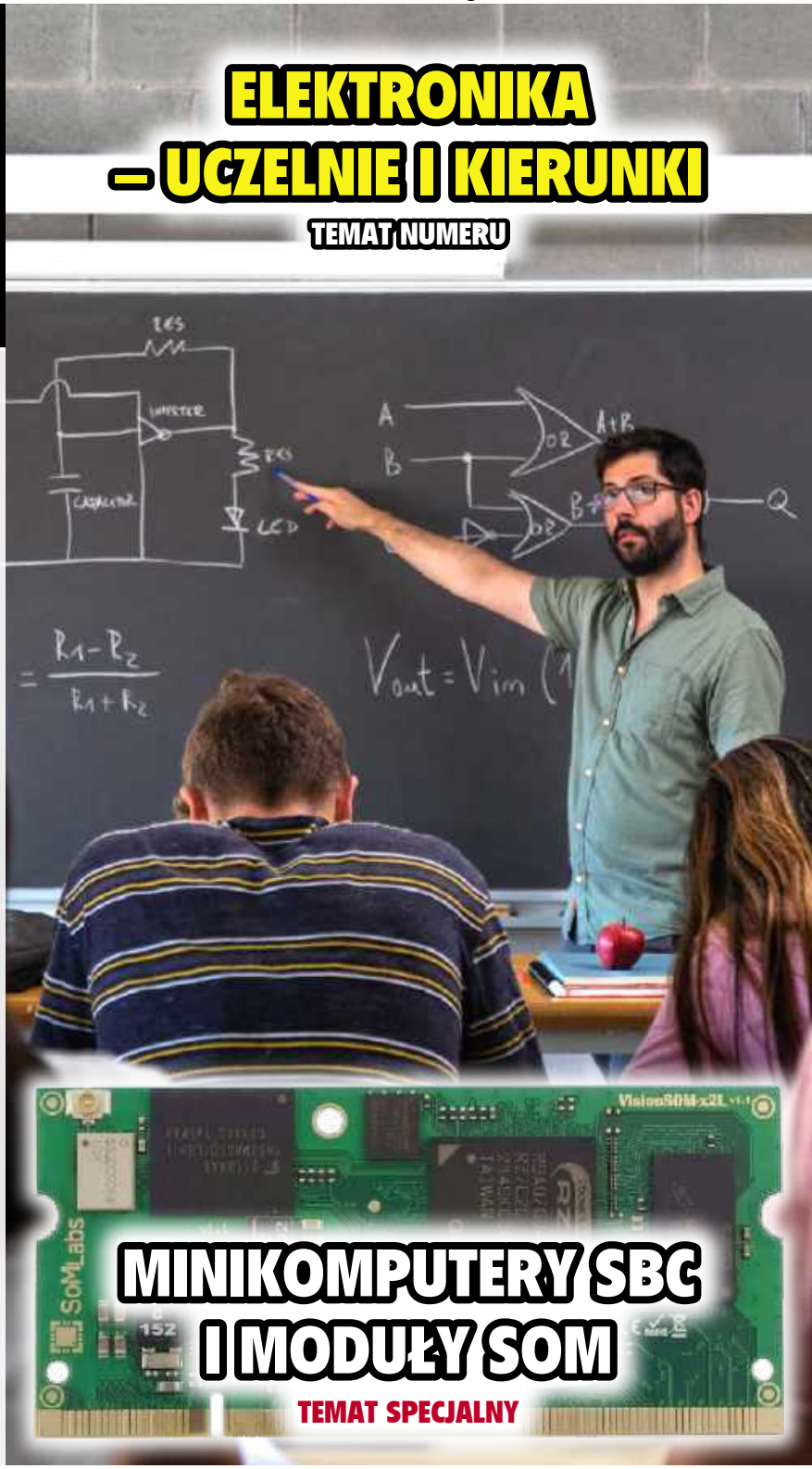
• Półprzewodnikowa rewolucja w domenie open source • Elastyczne i sztywno-giętkie obwody drukowane w ofercie PCBWay • Nowa wersja oprogramowania: Arm Keil MDK v6 • Single Pair Ethernet (SPE). Kluczowa technologia w cyfryzacji naszego świata • Optoelektronika do najbardziej wymagających aplikacji • Inżynieria Internetu Rzeczy: innowacyjne podejście do kształcenia inżynierów

tutoriale

• Druk 3D w służbie elektroniki • Pamięć w modułach SoM i SBC • Internet Rzeczy w pomiarach środowiskowych. Optymalizacja poboru mocy urządzenia IoT z płytką Raspberry Pi Pico W • Czujniki obrazu i przestrzeni

kursy

• Kurs programowania mikrokontrolerów Megawin • Kurs FPGA Lattice. Odbiornik UART



ELEKTRONIKA – UCZELNIE I KIERUNKI

TEMAT NUMERU

MINIKOMPUTERY SBC I MODUŁY SOM

TEMAT SPECJALNY

ISSN 1230-3526 Indeks 357677 05 9 771230 352245 18,90 zł (w tym 8% VAT) • PRICE 8 EUR

CZUJNIKI OBRAZU I PRZESTRZENI

prenumerata

Oferta prenumeraty ważna od 1.06.2024

–20%
NA START
181,40 zł

–30%
po pierwszym roku prenumeraty
158,80 zł

–40%
po drugim roku prenumeraty
136,10 zł

–50%
po trzecim roku nieprzerwanej prenumeraty
113,40 zł

Odkryj korzyści z prenumeraty drukowanej – większe oszczędności z każdym rokiem!

Rozpocznij swoją przygodę z *Elektroniką Praktyczną*. Decydując się teraz na roczną prenumeratę drukowaną, otrzymasz nie tylko dostęp do najnowszych wydań, ale i **znakomity start dzięki zniżce 20%** na pierwsze zamówienie!

Prenumerata to nie tylko wygoda dostępu do treści, ale także sposób na znaczące oszczędności. Dołącz do grona naszych stałych czytelników i ciesz się coraz lepszymi warunkami.

Im dłużej jesteś z nami, tym więcej oszczędzasz:
- po roku nieprzerwanej prenumeraty zapewnimy Ci **30% rabatu** na kolejny rok,
- po dwóch latach wierności zaoferujemy **40% rabatu**,
- po trzech latach lojalności osiągniesz **najwyższy poziom rabatu – 50%!**

Jak otrzymać rabat za lojalność?
Zaloguj się na swoje konto prenumeratora na **www.UlubionyKiosk.pl** i zamów prenumeratę, korzystając z przycisku PRZEDŁUŻ w zakładce „Prenumeraty".

Przeglądaj wcześniej, płać mniej – postaw na e-prenumeratę!

Wybierz prenumeratę cyfrową PDF i ciesz się dostępem do czasopisma nawet 7 dni przed oficjalną premierą w kioskach. Oszczędzaj czas i pieniądze – skorzystaj z **rabatu 30%** na roczną e-prenumeratę w cenie 126,90 zł.

Dodatkowa oferta dla prenumeratorów wersji drukowanej: jeśli już subskrybujesz wersję papierową, możesz dokupić równoległe e-wydania w cenie 36,20 zł/rok – z niesamowitym **rabatem 80%**.

Zyskaj nieograniczony dostęp do zasobów dla pasjonatów elektroniki!

Tylko prenumeratorzy mają pełny dostęp do:
- artykułów przed ich publikacją w *Elektronice Praktycznej* na www.ep.com.pl – EP W TOKU
- materiałów dodatkowych (takich jak pliki źródłowe projektów) na www.UlubionyKiosk.pl/media

Zamów prenumeratę drukowaną lub e-prenumeratę na www.UlubionyKiosk.pl lub przez przelew na konto Wydawnictwa AVT, a po zaksięgowaniu wpłaty wyślemy Ci mailowo kod dostępu do portalu.

Zacznij korzystać z pełnych zasobów już dziś!

Zamów prenumeratę lub e-prenumeratę na www.UlubionyKiosk.pl/prenumerata
tel. 22 257 84 22 (godz. 10–14) | prenumerata@avt.pl | AVT-Korporacja sp. z o.o. ul. Leszczynowa 11, 03-197 Warszawa
rachunek bankowy: 18 1050 1012 1000 0024 3173 1013

Elektroniczne szkiełko i oko

Stare porzekadło głosi, że jeden obraz jest wart więcej niż tysiąc słów. W XXI wieku koszt sprzętu, który pozwala ów obraz zarejestrować w doskonałej jakości, stał się niewiarygodnie niski. Pamiętam, jak jeszcze kilkanaście lat temu na internetowych forach dla amatorów elektroniki można było trafić na pojedyncze opisy uruchomienia niewielkich modułów kamer, pozyskanych z pierwszych wyposażonych w nie telefonów komórkowych. Każdy taki projekt rozgrzewał atmosferę i działał na wyobraźnię osób, które marzyły np. o budowie małego robota mobilnego, zdolnego do obserwacji otoczenia i transmisji obrazu na odległość. Ba! Niektórzy Czytelnicy być może pamiętają jeszcze ciekawy gadżet: kamery z serii MCA-10 (a także MCA-20 i MCA-25) miały postać zewnętrznej przystawki o kubaturze zbliżonej do 1/3...1/2 objętości samego telefonu. Rozdzielczość MCA-10 wynosiła 352×288 px – przy dzisiejszych standardach takie parametry mogą wywoływać pełen sentymentu uśmiech, ale u wielu amatorów zdobycie fragmentów dokumentacji interfejsu owej kamery (w celu jej podłączenia i pobrania obrazu) było swego czasu szczytem marzeń.

Od tamtych lat upłynęły ponad dwie dekady. Dziś niemal każdy nosi w kieszeni smartfona nie z jedną, ale trzema lub czterema kamerami, spośród których główny aparat ma rozdzielczość mierzoną w dziesiątkach bądź nawet pojedynczych setkach megapikseli. Moduły kamer o publicznie dostępnej, szczegółowej dokumentacji i naprawdę dobrych parametrach zalały rynek, a ich cena detaliczna zaczyna się od kilku...kilkunastu złotych. Rozbudowa urządzenia o funkcjonalność pozyskiwania i analizy obrazu staje się tak prosta, jak użycie dowolnego innego czujnika – w skrajnych przypadkach wizyjne rozpoznawanie obiektów można zrealizować poprzez prosty odczyt z użyciem interfejsu I²C lub UART, gdyż całość algorytmów przetwarzania i analizy siedzi we wbudowanym procesorze ML. Współczesne systemy embedded coraz silniej wspierają implementacje kamer, co da się zauważyć chociażby podczas przeglądania oferty nowoczesnych mikrokontrolerów i procesorów aplikacyjnych – standardem staje się wyposażanie ich w bloki peryferyjne wyspecjalizowane w komunikacji z czujnikami obrazu.

Pomimo zawrotnego tempa rozwoju technologii matryc, optyki i procesorów obrazu, wciąż nie sprawdziły się jednak prognozy niektórych „znawców", którzy wieszczyli szybką śmierć segmentu konwencjonalnych aparatów cyfrowych (głównie lustrzanek) na rzecz kamer dostępnych w smartfonach. Profesjonalne aparaty najwyższej klasy wciąż pojawiają się na rynku (i mają się doskonale), co więcej: są nieustannie zastępowane przez nowsze modele o doskonałych parametrach. Warto odnotować, że body lustrzanek czy bezlusterkowców z najwyższej półki wcale nie zostały zdominowane przez matryce o rozdzielczości porównywalnej choćby z aparatami montowanymi w przeciętnych modelach smartfonów – przykładowo Canon EOS-1D X Mark III ma czujnik obrazu o rozdzielczości „zaledwie" 20 Mpx, bezlusterkowy EOS R6 Mark II – 24,2 Mpx, topowy Nikon „z lustrem" (model D6) – 20,8 Mpx… podobne przykłady można by mnożyć. Skąd ta dysproporcja? Powód jest prosty – liczba megapikseli doskonale się sprzedaje na rynku elektroniki konsumenckiej. Podczas gdy profesjonalistów (bardziej niż rozdzielczość) interesuje szereg innych parametrów: poziom szumów, zakres dynamiki, szybkość zdjęć seryjnych, wsparcie wielopolowego autofokusa, zakres czułości czy wreszcie wierność odwzorowania barw, amatorom często w zupełności wystarczają zdjęcia o dość przeciętnej jakości, ale... skutecznie podrasowane przez odpowiednie filtry AI, z których działania często nawet nie zdają sobie sprawy. I nie ma w tym nic złego – każde urządzenie nadaje się do określonych zastosowań i lepiej lub gorzej spełnia oczekiwania użytkowników znajdujących się na różnych poziomach zaawansowania.

Można nawet pokusić się o stwierdzenie, że podczas gdy producenci czołowych modeli aparatów profesjonalnych walczą o każdy szczegół matrycy (i nie tylko – to samo dotyczy także współpracującej z nią elektroniki, jakości obudowy czy wreszcie najwyższej klasy optyki), wytwórcy smartfonów stosują rozwiązanie przypominające nieco… naturalny zmysł wzroku. Ludzkie oko ma bowiem – wbrew pozorom – raczej dość fatalne parametry optyczne. Tym, co stanowi o doskonałości naszego widzenia, jest nic innego, jak tylko sieć neuronowa: skomplikowany, wielowarstwowy system połączeń ma początek już na poziomie siatkówki i rozciąga się aż do kory wzrokowej włącznie. To właśnie dzięki zabiegom „naturalnej inteligencji" uzyskujemy taki, a nie inny obraz świata. W przypadku smartfonów jest nieco podobnie – matryce światłoczułe o pikselach w rozmiarze na poziomie mikrometra (lub mniej) nie są w stanie dostarczyć idealnego obrazu o niskim poziomie szumów, z odwzorowaniem kolorów także może być krucho, zaś subminiaturowa optyka siłą rzeczy wiąże się z problemami z uzyskaniem oczekiwanej przez użytkowników ostrości czy też zniekształceniami geometrii. Za finalną jakością stoją bowiem silne algorytmy przetwarzania obrazu, które czuwają nad każdym jego aspektem – odszumianiem, korekcją barwną, kompensacją cieni i świateł, odpowiednim wyostrzeniem cyfrowym, redukcją dystorsji czy wreszcie (tak popularnym w ostatnich latach) przetwarzaniem HDR. A zatem… po raz kolejny w historii zatoczyliśmy wielkie, technologiczne koło, by powrócić do naturalnych wzorców. Ewolucja po raz kolejny miała rację.

A co jeszcze – oprócz materiału o czujnikach obrazu – czeka na naszych Czytelników w majowym numerze „Elektroniki Praktycznej"? Oprócz kilku użytecznych projektów, interesującego radia internetowego na bazie ESP32 oraz artykułu o pamięciach stosowanych w minikomputerach, mamy także kolejne odcinki cyklów nt. programowania FPGA i pomiarów środowiskowych. Ponadto zaczynamy zapowiadany w kwietniu kurs programowania mikrokontrolerów Megawin, a jakby tego było mało – otwieramy nową rubrykę stałą, zatytułowaną „Technologie wokół elektroniki". Na pierwszy ogień bierzemy niezwykle ważną tematykę – druk 3D, rzecz jasna… z punktu widzenia elektronika.

Zapraszam do lektury!

Przemysław Musz

ELEKTRONIKA PRAKTYCZNA

Nr 5 (376) • Maj 2024

nowe podzespoły

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

Miniaturowe akcelerometry MEMS do aparatów słuchowych i urządzeń noszonych na ciele

Bosch Sensortec wprowadza do sprzedaży dwa miniaturowe akcelerometry MEMS o symbolach BMA530 i BMA580, zaprojektowane do zastosowań w aparatach słuchowych i innych urządzeniach noszonych na ciele użytkownika. Są to obecnie najmniejsze tego typu podzespoły, zamykane w obudowach WLCSP o powierzchni zaledwie 1,2×0,8 mm. W porównaniu z wcześniejszą wersją BMA253, zajmują mniejszą o 76% powierzchnię na płytce drukowanej, a ich wysokość zredukowano z 0,95 mm do 0,55 mm.

Oprócz urządzeń przenośnych, model BMA530 idealnie nadaje się do rozpoznawania gestów w zabawkach, wykrywania upadków w laptopach i innych urządzeniach oraz do aktywowania funkcji zarządzania energią, takich jak uśpienie smartfona, gdy pozostaje on w bezruchu.

BMA580 umożliwia wykrywanie aktywności głosowej i oferuje zaawansowany zestaw funkcji do urządzeń audio. Zwykle mikrofony w aparatach słuchowych muszą być zawsze włączone, aby nasłuchiwać aktywności głosowej. Zamiast tego, BMA580 korzysta z przewodnictwa kostnego do wykrywania wibracji głosu użytkownika, a następnie wybudza mikrofon ze stanu uśpienia, co pozwala ograniczyć pobór mocy. Żaden inny czujnik dostępny na rynku nie łączy obu tych zalet: praktycznego zastosowania zjawiska przewodnictwa kostnego z tak małymi wymiarami. BMA580 umożliwia interakcję z użytkownikiem w odpowiedzi na dotknięcie urządzenia, na przykład w celu odebrania lub zakończenia połączenia. Jego oprogramowanie zawiera algorytm, który potrafi rozróżnić pojedyncze, podwójne i potrójne dotknięcia.

	BMA530	BMA580
Zakresy pomiarowe	±2, ±4, ±8, ±16 g	
Rozdzielczość	16 bitów	
Częstotliwość pracy interfejsu	1,56 Hz – 6,4 kHz	
Offset	±75 mg	±50 mg
TCO	±0,75 mg/K	±0,2 mg/K
Błąd czułości	1%	0,5%
Gęstość szumów napięciowych	120 µg/√Hz	
Pobór prądu (tryb ciągły)	125 µA	
Pobór prądu (tryb low-power @ 100 Hz)	18 µA	
Pobór prądu (tryb suspend)	4,75 µA	
Interfejsy	I^3C, I^2C, SPI	
Pamięć FIFO	1 kB	

www.bosch-sensortec.com

6-osiowy czujnik inercyjny MEMS o zakresie dopuszczalnej temperatury pracy od –40 do +110°C

SCH16T-K01 to najnowszy 6-osiowy czujnik inercyjny firmy Murata, zrealizowany w technologii pojemnościowej 3D-MEMS, wyposażony w 3-osiowy żyroskop i 3-osiowy czujnik przyspieszenia. Żyroskop charakteryzuje się stabilnością 0,5 dph i gęstością szumu

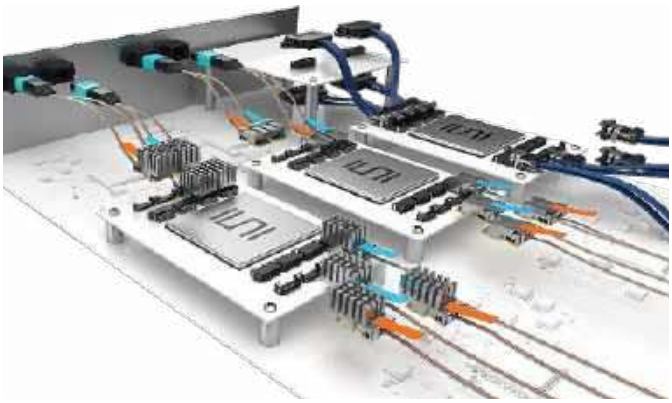
0,3 mdps/√Hz. Akcelerometr oferuje szeroki zakres dynamiczny do 26 g, co zapewnia ochronę przed wibracjami. SCH16T-K01 wykazuje bardzo dobrą liniowość i stabilność offsetu w całym zakresie dopuszczalnej temperatury pracy od –40 do +110°C. Jego wyjście zostało skalibrowane fabrycznie, co ułatwia implementację w urządzeniach docelowych, a wbudowane funkcje diagnostyczne pozwalają na zastosowania w systemach safety-critical. Dodatkowym atutem są funkcje synchronizacji czasowej (wyjście Data Ready, indeks czasu, wejście SYNC).

SCH16T-K01 może znaleźć zastosowanie w maszynach budowlanych i rolniczych, systemach transportu materiałów oraz oprzyrządowaniu morskim. Jest zamykany w obudowie SOIC-24 o wymiarach 14×12×3 mm. Uzyskał kwalifikację AEC-Q100, co pozwala na jego użycie również w branży motoryzacyjnej.

Pozostałe parametry:

- zakres pomiarowy żyroskopu: ±300°/s,
- zakres pomiarowy akcelerometru: ±8 g,
- zakres napięcia zasilania: 3,0...3,6 V (1,7...3,6 V w przypadku linii I/O),
- format wyprowadzania danych: 16– i 20-bitowy,
- interfejs: SafeSPI v2.0.

www.murata.com



Złącza FireFly Micro Flyover do interfejsów optycznych i miedzianych o przepustowości do 28 Gbps

Samtec wprowadza na rynek pierwszy system połączeniowy, umożliwiający wymienne stosowanie mikrozłączy optycznych i miedzianych – FireFly Micro Flyover. Składa się on z transceivera, systemu złączy dwuczęściowych oraz kabla. Może być używany w systemach o szybkości transmisji 14, 16, 25 i 28 Gbps, pracujących w konfiguracjach ×4, ×8 i ×12. Producent oferuje do niego modele 3D, kartę adaptera PCI Express-over-Fiber oraz zestawy ewaluacyjne, dostępne na stronie internetowej Samtec jako część usługi Samtec Sudden Service.

System FireFly Micro Flyover może znaleźć zastosowanie w szybkich systemach komputerowych, medycynie, aparaturze pomiarowej i aplikacjach FPGA. Model ECUO obsługuje transmisję danych z szybkością do 56 Gbps, zaś model ETUO, przeznaczony do pracy w rozszerzonym zakresie temperatury otoczenia od –40 do +85°C, jest polecany do zastosowań wojskowych, lotniczych i przemysłowych. Zapewnia bezbłędną transmisję danych podczas wstrząsów i wibracji, określonych w normie MIL-STD-810.

Model PCUO, idealny do zastosowań w systemach o dużej gęstości (testery ATE, dystrybucja sygnałów wideo, automatyka przemysłowa), umożliwia transmisję danych PCIe 3.0/4.0 na odległość do 100 m. Rozszerzona wersja temperaturowa, PTUO, działa w zakresie od –40°C do +85°C przy współczynniku BER lepszym niż 1E-12.

Złącza systemu FireFly Micro Flyover zapewniają szybkość transmisji do 28 Gbps przy minimalnej powierzchni footprintu wynoszącej zaledwie 4 cm². W razie potrzeby mogą być chłodzone radiatorem. Wszystkie modele są wymienne z kablami miedzianymi i optycznymi FireFly. Małe gabaryty umożliwiają umieszczanie ich tuż obok modułów ASIC.

Samtec obecnie oferuje trzy zestawy ewaluacyjne wspierające system FireFly Micro Flyover: 14 Gbps FireFly FMC Development Kit, 25/28 Gbps FireFly FMC+ Development Kit oraz 28 Gbps FireFly Evaluation Kit.

www.samtec.com

Bezpieczniki resetowalne PPTC serii MF-NSMF w wersjach o prądzie i napięciu roboczym do 2,0 A/60 V

Firma Bourns powiększa ofertę resetowalnych bezpieczników PPTC o ponad 20 nowych modeli, stanowiących rozszerzenie serii MF-NSMF. Obecnie w ramach tej serii dostępne są bezpieczniki o prądzie znamionowym (I_{hold}) do 2,0 A i napięciu roboczym ($V_{maks.}$) do 60 V_{DC}, które mogą znaleźć zastosowanie w ochronie termicznej i nadprądowej pakietów akumulatorowych oraz portów USB i HDMI w komputerach stacjonarnych, laptopach, odbiornikach TV, a także innych urządzeniach konsumenckich. Są one produkowane w obudowach chipowych 1206 (3,2×1,6 mm) o grubości od 0,1 do 1,6 mm. Spełniają wymogi norm UL/CSA/TUV w zakresie bezpieczeństwa użytkowania (IEC 62319-1, IEC 60738-1, IEC 60730-1:2013). Niektóre modele uzyskały kwalifikację AEC-Q200.

www.bourns.com

Generatory dźwięku i powiadomień głosowych do współpracy z tanimi buzzerami

Seiko Epson rozpoczyna dystrybucję wersji próbnych dwóch nowych układów LSI do generowania komunikatów głosowych i powiadomień. S1V3F351 i S1V3F352, zawierające generator dźwięków, pamięć Flash do przechowywania danych audio oraz oscylator, umożliwiają łatwe dodawanie funkcji audio do urządzeń konsumenckich i medycznych oraz systemów automatyki budynkowej i domowej.

Tradycyjnie osiągnięcie wysokiej jakości sygnału audio w urządzeniach konsumenckich wymagało stosowania głośników. Tanie buzzery zwykle nie zapewniały zadowalającej jakości dźwięku

	S1V3F351	S1V3F352
Pojemność pamięci Flash	około 30 s nagrania	około 80 s nagrania
Pojemność zewnętrznej pamięci QSPI-Flash I/F	do 16 MB	
Liczba kanałów	2	
Algorytm generowania dźwięku	– EOV (format Epson o dużej kompresji danych) – PCM (16 bitów)	
Częstotliwość próbkowania	15,625 kHz	
Bitrate	16/24 kbps	16/24/32/40 kbps
Regulacja głośności	–63 dB...0 dB w krokach co 0,5 dB	
Liczba powtórzeń	1...254 lub do komendy Stop	
Regulacja szybkości odtwarzania	75%...125% w krokach co 5%	
Zmiana wysokości tonu (pitch conversion)	75%...125% w krokach co 5%	–
Interfejsy do systemu host	SPI, UART, I²C; tryb autonomiczny	
Zegar	wewnętrzny lub zewnętrzny	
Napięcie zasilania	1,8...5,5 V	
Napięcie programowania pamięci Flash	2,2...5,5 V	2,4...5,5 V
Pobór prądu w trybie uśpienia	0,34 µA	0,46 µA
Obudowa	P-TQFP048-0707-0.50	

i odpowiedniego ciśnienia akustycznego. Aby wyeliminować ten problem, firma Epson opracowała algorytm umożliwiający uzyskanie wysokiej jakości dźwięku nawet w prostych buzzerach.

Wyjątkową cechą modelu S1V3F351 jest funkcja regulacji prędkości i tonu dźwięku w czasie rzeczywistym, pozwalająca użytkownikom końcowym modyfikować dźwięk zgodnie z ich preferencjami. Dodatkowo funkcja mieszania sygnałów w dwóch kanałach umożliwia jednoczesne odtwarzanie muzyki w tle i dźwięku powiadomienia. Dzięki bardzo skutecznej kompresji S1V3F351 i S1V3F352 wymagają jedynie niewielkiej pojemności pamięci do przechowywania danych audio, umożliwiając generowanie powiadomień w wielu językach i wybór dźwięku tła spośród wielu dostępnych wariantów.

Dodatkowo Epson dostarcza aplikację na komputery PC, która pozwala na generowanie powiadomień głosowych w 12 językach, bez potrzeby realizacji profesjonalnych nagrań studyjnych. Narzędzie to umożliwia również import istniejących danych audio w formacie WAV.

www.epson-electronics.de

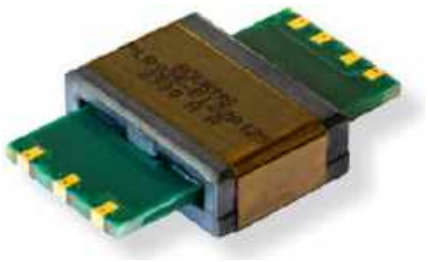
10-amperowy przekaźnik SPDT o napięciu izolacji do 10 kV

G6RN-1 to nowy 10-amperowy przekaźnik SPDT NO z oferty firmy Omron, produkowany w obudowie o wysokości 15 mm, co stanowi około 60% wysokości G2R, czyli wcześniejszego odpowiednika tego komponentu. G6RN-1 wykazuje dużą czułość cewki przy małym poborze mocy (wynoszącym 220 mW). Charakteryzuje się odstępem izolacyjnym i drogą upływu po 8 mm, wystarczającymi do zapewnienia izolacji na poziomie 10 kV pomiędzy cewką i kontaktami. Zakres dopuszczalnej temperatury pracy wynosi od –40 do +85°C.

Przekaźnik G6RN-1 występuje w wersjach o napięciu znamionowym cewki 5, 6, 12 i 24 V_{DC}. Spełnia wymogi norm IEC/EN 60335-1 i IEC60079-15 w zakresie bezpieczeństwa użytkowania i możliwości pracy w strefach zagrożonych wybuchem. Dopuszczalny prąd przełączany wynosi (w zależności od wersji) 8 A lub 10 A, a napięcie znamionowe to 250 V_{AC}/30 V_{DC}.

www.components.omron.com

Transformatory planarne do aplikacji PoE, zasilaczy i systemów oświetleniowych



Bourns wprowadza na rynek dwa transformatory planarne z serii PLR, zaprojektowane do zastosowań w konwerterach DC-DC o topologii forward w zasilaczach impulsowych, systemach oświetleniowych i aplikacjach PoE o mocy wyjściowej do 25 W. Są one przystosowane do montażu SMT (wymiary budowy wynoszą 34×23×7 mm). Występują w wersjach z uzwojeniem wyjściowym 5 V/5 A (model PLR025-EI22A05S) i 12 V/2,1 A (PLR025-EI22A12S). Uzwojenie wejściowe jest przystosowane do pracy z napięciem 35...57 V.

Oba modele charakteryzują się izolacją na poziomie 1500 V_{DC} @ 1 s, indukcyjnością uzwojenia pierwotnego min. 85 µH i małą indukcyjnością rozproszenia (maks. 0,216 µH). Mogą pracować w zakresie temperatury otoczenia od –40 do +125°C.

www.bourns.com

Energooszczędny moduł komunikacyjny LoRa o zasięgu transmisji do 15 km w otwartym terenie

W ofercie firmy Quectel pojawił się nowy moduł komunikacyjny KG200Z, pracujący w standardzie LoRa, a zaprojektowany do aplikacji IoT wymagających niezawodnej, bezprzewodowej komunikacji długodystansowej przy małym poborze mocy. Oferuje on zasięg transmisji 2...5 km w środowisku miejskim i 10...15 km w otwartym terenie oraz zapewnia stabilne połączenie, a także dużą odporność na zaburzenia elektromagnetyczne. Jest tanim modułem, opartym na mikrokontrolerze STM32WLEx firmy STMicroelectronics, kompatybilnym z protokołem LoRaWAN. Zawiera jednostkę zarządzania zasilaniem, wzmacniacz mocy, wzmacniacz niskoszumny i przełącznik transceivera RF. Obsługuje zakresy częstotliwości 470...510 MHz i 862...928 MHz, modulacje LoRa, (G) FSK, (G)MSK i BPSK oraz szyfrowanie danych w standardzie AES. Jeśli chodzi o komunikację lokalną, został wyposażony w interfejsy UART, SPI, I²C i SWD.

KG200Z jest dostarczany w postaci modułu o wymiarach 12×12×1,8 mm. Może znaleźć zastosowanie w inteligentnych zamkach i czujnikach zamknięcia drzwi, systemach wykrywania wycieków gazu i wody, czujnikach jakości powietrza, instalacjach HVAC, miernikach zużycia mediów, do monitorowania ruchu drogowego i dostępności miejsc parkingowych oraz we wszelkiego typu aplikacjach, w których urządzenia muszą wysyłać okresowo wiadomości na długie dystanse przy małym poborze mocy.

Ważniejsze dane techniczne:

- zasięg transmisji: 2...5 km w miastach i 10...15 km w otwartym terenie,
- pobór mocy: 1,7 µA w trybie uśpienia,
- czułość odbiornika: –138 dBm,
- interfejsy: UART, SPI, I²C i SWD,
- wymiary i masa: 12×12×1,8 mm, 0,56 g,
- zakres temperatur pracy: od –40°C do +85°C.

www.quectel.com

Bezkontaktowy czujnik temperatury IR do termometrów dousznych i bramek pomiarowych

ZTP-148SRC1 to tani, bezkontaktowy czujnik temperatury z oferty firmy Amphenol, charakteryzujący się małymi gabarytami, dużą czułością i krótkim czasem odpowiedzi. Zawiera termoelement o powierzchni aktywnej 0,7×0,7 mm, filtr IR o płaskiej charakterystyce spektralnej i termistor do kompensacji zmian temperatury otoczenia, zamknięte w hermetycznej obudowie TO-46 o wymiarach ∅ 5,4×2,8 mm. Może znaleźć zastosowanie w termometrach dousznych i czołowych oraz bezdotykowych bramkach do pomiaru temperatury. Wewnętrzny sensor ZTP-148SRC1 charakteryzuje się rezystancją wynoszącą typowo 85 kΩ @ +25°C, współczynnikiem temperaturowym maks. 0,12%/°C, napięciem szumu 37 nW/√Hz i stałą czasową 32 ms. Rezystancja wewnętrznego termistora wynosi 100 kΩ (±3%) @ 25°C, a jego współczynnik beta to 3950 K.

www.amphenol-sensors.com

Enegooszczędny moduł komunikacyjny Bluetooth LE z 256 kB wbudowanej pamięci Flash

Renesas wprowadza do oferty nowy, energooszczędny moduł komunikacyjny Bluetooth LE z jednostką obliczeniową ARM Cortex-M33 i wbudowaną pamięcią Flash, oznaczony symbolem DA14592. Pracuje z napięciem zasilania od 1,7 do 3,6 V, pobierając w stanie aktywnym

od 1,1 do 9,5 mA prądu – w zależności od trybu pracy i mocy nadawania. W stanie uśpienia pobór prądu jest ograniczony do 2,0...4,8 µA, a w trybie hibernacji – bez podtrzymania pamięci RAM i przy wyłączonych wewnętrznych zegarach – zmniejsza się do 90 nA. Układ pracuje z maksymalną mocą nadajnika +6 dBm i charakteryzuje się czułością odbiornika –97 dBm.

Dzięki przyjęciu optymalnego kompromisu pomiędzy pojemnością wbudowanych pamięci RAM, ROM i Flash (odpowiednio 96 kB, 288 kB i 256 kB) oraz rozmiarem chipa DA14592 nadaje się do szerokiej gamy aplikacji, w tym urządzeń medycznych, śledzenia zasobów, systemów pomiarowych, skanerów PoS oraz systemów lokalizacyjnych CSL (crowd-sourced location), opartych na danych od wielu użytkowników. Wymaga tylko 6 zewnętrznych komponentów pasywnych i występuje w wersjach z dwoma typami obudów: WLCSP (3,32×2,48 mm) i FCQFN (5,1×4,3 mm). W ofercie Renesas jest też dostępny gotowy do implementacji wariant DA14592MOD – ze wszystkimi elementami współpracującymi i anteną – którego wymiary wynoszą 20,0×15,9×2,5 mm. Pozwala on skrócić czas wprowadzania produktów na rynek i zredukować ogólne koszty projektu.

www.renesas.com

Miniaturowe przełączniki kontaktronowe o mocy znamionowej 10 W i dużej żywotności

Littelfuse po raz kolejny rozszerza ofertę miniaturowych przełączników kontaktronowych. Tym razem wprowadza do sprzedaży 7-milimetrowe przełączniki z serii MITI-7L o dłuższym czasie bezawaryjnej pracy (w stosunku do wcześniejszych odpowiedników). Mogą one znaleźć zastosowanie m.in. w instalacjach alarmowych i testerach ATE do badania parametrów półprzewodników dużej mocy.

W ramach serii MITI-7L producent oferuje przełączniki NO (normalnie otwarte), produkowane w hermetycznych, szklanych korpusach o wymiarach ∅ 1,8×7 mm. Są one w stanie przełączać moc do 10 W. Charakteryzują się rezystancją izolacji minimum 10^{11} Ω i rezystancją styków poniżej 150 mΩ. W ofercie Littelfuse znalazły się też ich odpowiedniki do montażu powierzchniowego (seria MISM-7L).

Pozostałe parametry:

- napięcie przełączane: 170 V_{DC}/120 V_{AC},
- prąd przełączany: 0,25 A_{DC}/0,18 A_{AC},
- pojemność styków: 0,65 pF,
- czas włączania/wyłączania: maks. 0,4 ms/0,5 ms,
- odporność na udary/wibracje: 100 g/30 g,
- zakres temperatury pracy: –40...+125°C,
- żywotność: min. 100 milionów cykli (5 V_{DC}/20 mA, 100 Hz).

www.littelfuse.com

Magnetyczny czujnik przemieszczenia o dokładności 10 µm

Infineon prezentuje nowy magnetyczny czujnik przemieszczenia, oznaczony symbolem XENSIV TLI5590-A6W, wyprodukowany w technologii TMR (tunnel magnetoresistance). Struktura

wewnętrzna układu obejmuje dwa mostki Wheatstone'a, których rezystancja TMR zależy od kierunku i natężenia zewnętrznego pola magnetycznego. W wyniku oddziaływania pola zewnętrznego magnesu wielobiegunowego oba mostki generują przebiegi wyjściowe (sin, cos), używane do pomiaru położenia względnego. Układ zapewnia bardzo dobre dopasowanie parametrów kanałów (<5% w pełnym wejściowym zakresie indukcji magnetycznej ±5 mT) i pozwala uzyskać dokładność pomiaru rzędu 10 µm.

TLI5590 jest polecany do zastosowań zwłaszcza w produkcji wielkoseryjnej. Może być wdrażany jako zamiennik enkoderów optycznych i resolwerów. Ze względu na duży stosunek sygnału do szumu i mały pobór mocy doskonale nadaje się do urządzeń bateryjnych. Jest zamykany w miniaturowej, 6-pinowej obudowie SG-WFWLB-6-3 o wymiarach 1,27×0,93×0,4 mm. Może pracować w szerokim zakresie temperatury otoczenia od –40 do +125°C, co pozwala na zastosowanie go przy pomiarze ruchu liniowego i obrotowego, zarówno w aplikacjach konsumenckich, jak i przemysłowych.

Pozostałe parametry:

- czułość: 9/18 mV/V/mT,
- pasmo: 5 kHz,
- pobór prądu: 1 mA @ 3,3 V,
- zakres temperatury pracy: –40...+125°C,
- odporność na wyładowania ESD: ±2 kV (HBM).

www.infineon.com

Chipset do realizacji adapterów sieciowych USB-C PD pracujących w topologii ZVS flyback

Infineon wprowadza na rynek nowy chipset EZ-PD PAG2. Seria obejmuje 4 układy scalone ułatwiające realizację ładowarek sieciowych USB-C power delivery (PD) o małych gabarytach i dużej mocy wyjściowej.

EZ-PD PAG2S-AC i EZ-PD PAG2S-QZ to kontrolery obsługujące topologie – odpowiednio – ACF (active-clamp flyback) i QR-ZVS (quasi-resonant flyback with zero voltage switching), zapewniające obsługę protokołu USB-PD oraz wyposażone w funkcje synchronicznego prostownika i kontrolera PWM. Zapewniają one komunikację oraz izolację pomiędzy sekcją pierwotną i wtórną za pośrednictwem transformatora impulsowego PET (pulse-edge transformer) CYPET121.

EZ-PD PAG2P jest kontrolerem pracującym po stronie pierwotnej transformatora, wyposażonym m.in. w wysokonapięciowy układ startowy, odbiornik sygnału z transformatora impulsowego, sterowniki bramek, obwody zabezpieczające, układ rozładowania kondensatora przeciwzakłóceniowego i konwerter DC-DC boost.

EZ-PD PAG2S-PS – zintegrowany kontroler USB PD i SR (synchronous-rectification) – może współpracować z kontrolerami PWM innych producentów. Charakteryzuje się krótkimi czasami włączania/wyłączania. Zawiera regulator napięcia do kontroli stałego napięcia i stałego prądu przy użyciu sprzęgaczy optycznych.

Kontrolery rodziny EZ-PD PAG2 pozwalają uzyskać dużą sprawność energetyczną w aplikacjach o szerokim zakresie mocy wyjściowej. Zapewniają kompatybilność ze standardami BC v1.2, AFC, Apple charging i QC5.0 oraz obsługują szeroki zakres napięcia wejściowego od 90 do 265 V_{AC}, co pozwala na pracę w różnych regionach geograficznych.

EZ-PD PAG2S-QZ i EZ-PD PAG2S-AC produkowane są w obudowach QFN-32, a EZ-PD PAG2S-PS – w obudowach QFN-32 i SOIC-24. EZ-PD PAG2P występuje w dwóch wariantach: z funkcją rozładowania kondensatora przeciwzakłóceniowego lub bez niej. Oba występują w obudowach typu SOIC-14.

www.infineon.com

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.

Carpenter Tau – Programowalna klawiatura DIY wykonana z drewna

W dziale obserwowanych projektów konstrukcje klawiatur DIY pojawiają się regularnie. Prezentowany projekt autorstwa Bo Yao wyróżnia się materiałem – wykonany został bowiem z drewna czarnego orzecha i różowego palisandru. Z tego względu klawiatura Carpenter Tau stanowi połączenie tradycyjnego rzemiosła i nowoczesnej technologii. Starannie wyrzeźbiona z drewna, mechaniczna perełka oferuje niepowtarzalne doświadczenia podczas pisania, a konfigurowalne układy i skróty pomagają zwiększyć wydajność pracy.

Do kontroli klawiatury Carpenter Tau autor wybrał odpowiedni mikrokontroler – SoC – który integruje w sobie Bluetooth i protokół RF o częstotliwości 2,4 GHz. Po analizie różnych opcji zdecydował się na układ WCH CH582 (ze względu na jego moc obliczeniową, bezpośredni dostęp do protokołu RF 2,4 GHz i niewielkie rozmiary).

Kolejny etap procesu projektowego stanowiło skonstruowanie kompletnego modelu 3D klawiatury. Autor korzystał z oprogramowania do projektowania parametrycznego – takiego jak Creo Parametric – do stworzenia modelu klawiszy, płyty klawiatury, obudowy i opcjonalnego podparcia dłoni. Parametryczne projektowanie umożliwiło mu dostosowanie grubości krawędzi klawiszy i innych elementów, tak by klawiatura lepiej spełniała wymagania użytkownika. Po zakończeniu projektu 3D pliki zostały zaimportowane do oprogramowania do renderowania (takiego jak KeyShot) w celu wygenerowania realistycznej wizualizacji klawiatury.

Autor, jak sam pisze, zdecydował się na drewno jako materiał do budowy klawiatury z kilku powodów. Nowoczesnemu pracownikowi, który każdego dnia spędza wiele godzin przy klawiaturze, znacznie przyjemniej jest z niej korzystać, gdy czuje naturalny dotyk drewna. Twarde drewniane obudowy i klawisze – podobnie jak metalowe – są solidne i precyzyjne, ale drewno jest lżejsze od metalu. Trwałość naturalnego surowca można zwiększyć dzięki zastosowaniu olejnego impregnatu, zabezpieczającego przed pleśnią, wilgocią i gniciem. Wreszcie, drewno to materiał biodegradowalny – nawet gdy w końcu się zużyje i wymaga utylizacji, nie szkodzi środowisku.

Prezentowana klawiatura jest w pełni programowalna. Zapewnia ona łączność bezprzewodową w trzech stanach – z użyciem interfejsu 2,4 GHz, Bluetooth lub po kablu, zależnie od wymagań. Na ogół klawiatury open source oferują jedynie łączność przewodową i Bluetooth, pomijając opcję 2,4 GHz. Kluczowa różnica tkwi w opóźnieniu Bluetooth ma obszerniejszy stos, co skutkuje minimalnie większym opóźnieniem (wynoszącym co najmniej 10 ms), nawet w przypadku najszybszych klawiatur Bluetooth. Projekty hobbystyczne open source mogą mieć nawet wyższe opóźnienia. Różnica ta staje się zauważalna zwłaszcza w przypadku gier. Dzięki najszybszym klawiaturom, które korzystają z interfejsu 2,4 GHz, opóźnienie można zredukować do 1 ms, co sprawia, że realna staje się nawet częstotliwość rzędu 1000 naciśnięć klawisza na sekundę. To prędkość porównywalna z parametrami klawiatury przewodowej.

Za łącznością 2,4 GHz przemawia nie tylko chęć zminimalizowania opóźnień, ale także inne czynniki, takie jak żywotność baterii czy brak odbiornika Bluetooth w niektórych komputerach stacjonarnych. Większość klawiatur DIY nie zapewnia takiej komunikacji, ale autor zdecydował się na jej zaimplementowanie. Wskazał również, że implementacja szybkiego protokołu przy użyciu modułu przesyłającego „surowe dane" 2,4 GHz nie jest tak trudna, jak mogłoby się wydawać. Wymaga pracy porównywalnej z opracowaniem efektywnego protokołu komunikacyjnego na podstawie UDP czy QUIC. W prototypowym eksperymencie autorowi udało się osiągnąć stabilne opóźnienie transmisji wynoszące 1 ms (przy zastosowaniu dwóch tanich układów do transmisji 2,4 GHz, bez korzystania z jakiegokolwiek standardowego protokołu RF). Zdaniem autora, w przypadku konstrukcji klawiatury wysokiej jakości ważne jest, aby nie szukać oszczędności również w tym aspekcie.

https://hackaday.io/project/193751-carpenter-tau-keyboard
https://github.com/taukeyboard/carpenter-tau-keyboard

Aparat fotograficzny oparty na Raspberry Pi Zero HQ

Projekt ten, choć wygląda na gotowy, nadal jest na etapie tworzenia i daleko mu do finalnej formy – w pełni funkcjonalnego cyfrowego aparatu, opartego na module Raspberry Pi Zero HQ.

Budowane urządzenie korzysta z kompaktowego komputera jednopłytkowego Raspberry Pi Zero do obsługi kamery i wyświetlacza. Całość funkcjonować ma jak zwykły aparat kompaktowy.

Na obecnym etapie autor musi dokończyć tworzenie kodu i wydrukować jeszcze m.in. okładki beczkowe dla pierścieni przysłony/ostrzenia.

System oferuje bezpośredni przekaz obrazu kamery w czasie rzeczywistym z funkcjonalnościami zoom-crop i panning, ułatwiającymi manualne wyostrzenie obrazu przed zrobieniem zdjęcia.

Głównym mankamentem tego projektu okazuje się niska częstotliwość odświeżania wyświetlacza OLED. Autor jeszcze nie znalazł na to sposobu, ale potencjalnym rozwiązaniem wspomnianego problemu mogłoby okazać się przepisanie programu w szybszym języku programowania – np. w C.

https://tiny.pl/d9vc1

W ofercie AVT*
AVT6039

Najważniejsze parametry:
- konstrukcja oparta o scalony transceiver CAN typu SN65HVD1050,
- dwie opcje terminacji magistrali,
- wbudowane zabezpieczenia ESD i bezpieczniki polimerowe,
- zasilanie: 6...24 V,
- automatyczny wybór linii danych w przypadku współpracy z modułami UNO R4 Minima i UNO R4 Wi-Fi.

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

AVT6034	Nakładka z wyświetlaczem OLED do AVTDuino UNO R4 (EP 4/2024)
AVT6028	Sterownik silników do AVTDuino UNO R4 (EP 3/2024)
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)

* **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**.

Nakładka z transceiverem CAN do AVTduino UNO R4 Plus

Procesor R7FA4M1 – zastosowany w nowym Arduino UNO R4 – ma wbudowany kontroler CAN. Aby skorzystać z możliwości komunikacji opartej o tę magistralę we własnych aplikacjach, wystarczy tylko dodać odpowiedni transceiver, na przykład w postaci zaprezentowanej nakładki.

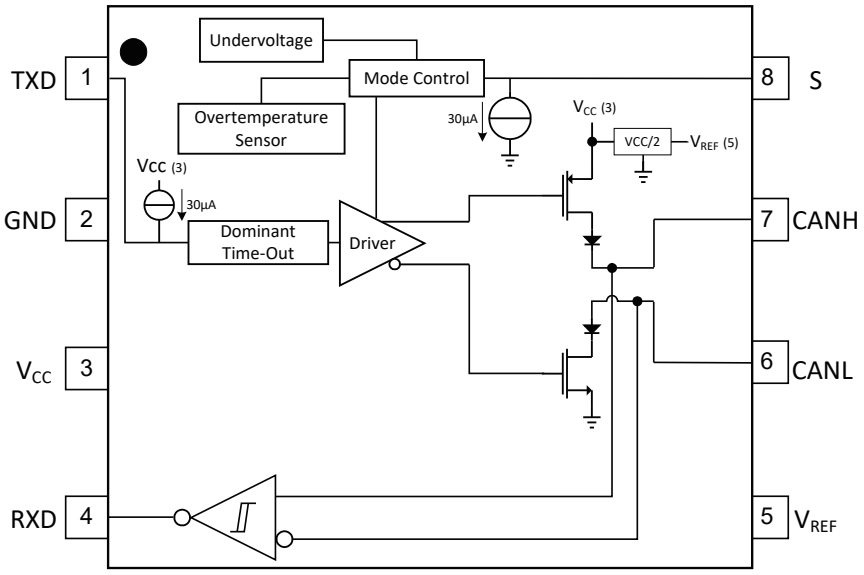
Nakładka korzysta z układu transceivera CAN typu SN65HVD1050, którego budowę pokazano na **rysunku 1**.

Układ zawiera wszystko, co jest konieczne do spełnienia wymogów standardów CAN zgodnie z normą ISO11898-2, a jego aplikacja ogranicza się do zaledwie kilku elementów zewnętrznych.

Schemat nakładki pokazano na **rysunku 2**. Aplikację SN65HVD1050 uzupełniają elementy zabezpieczające U1 oraz terminujące magistralę CAN. Dioda TVS1 zabezpiecza transceiver przed skutkami przepięć, a bezpieczniki polimerowe F1, F2 – przed skutkami zwarć magistrali. Przełącznik RT umożliwia odłączenie rezystorów terminujących R4, R5, gdy nakładka jest nie jest „skrajnym" urządzeniem magistrali CAN. Sposób terminacji i aktywację napięcia VREF można dostosować do wymogów aplikacji. Domyślnie przy użyciu rezystorów R4, R5 i kondensatora C3 realizowany jest schemat terminacji dzielonej, z możliwością zastosowania napięcia VREF=VCC/2 układu U1 w celu stabilizacji punktu pracy trybu współbieżnego (CM) po zwarciu zwory REF. Usuwając kondensator C3 i zworę REF wdrażamy schemat terminacji standardowej z obciążeniem magistrali wyłącznie rezystancją R4+R5≈120 Ω.

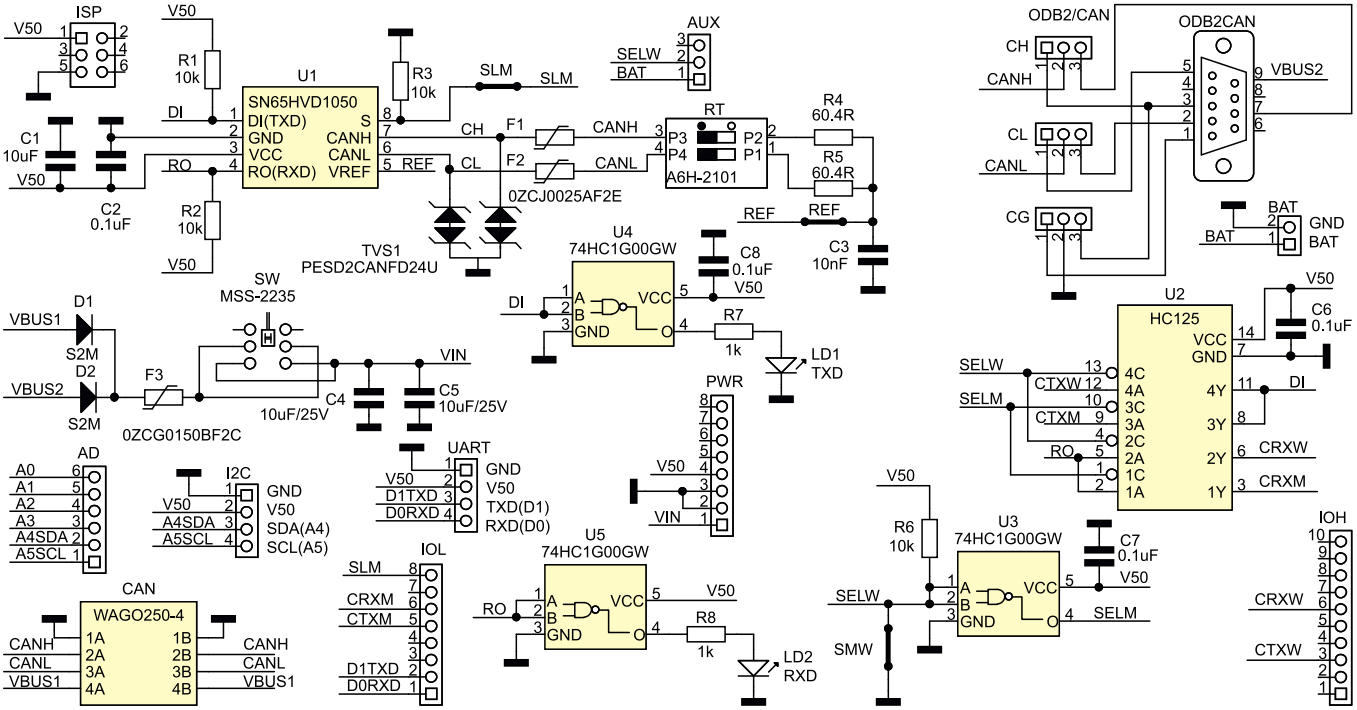
Z transceivera magistrala CAN wyprowadzona jest na złącze sprężynowe CAN typu WAGO250 oraz na złącze DB9 ODB2CAN. W przypadku tego ostatniego za pomocą zwór możemy ustalić sposób wyprowadzenia sygnałów na zgodność z ODB2 lub CAN. Każdorazowo jednak przed podłączeniem należy sprawdzić poprawność instalacji przewodu i zastosowanego gniazda ODB/CAN, gdyż zdarzają się modyfikacje. Każde z zastosowanych w nakładce gniazd CAN umożliwia doprowadzenie zasilania do płytki bazowej ARDUINO. Napięcie powinno spełniać wymogi UNO R4, czyli mieścić się w zakresie 6...24 V z dodatkowym uwzględnieniem spadku na diodach D1, D2 zabezpieczających płytkę przed odwrotnym podłączeniem zasilania. Napięcie zewnętrzne doprowadzone jest do złącza PWR na wyprowadzenie VIN. Dodatkowy przełącznik SW umożliwia wyłączenie zasilania bez wypinania przewodów magistrali, bezpiecznik F3 zabezpiecza natomiast Arduino i zasilanie CAN przed skutkami ewentualnych zwarć podczas prototypowania.

Pewną niekonsekwencją projektową komplikującą schemat nakładki okazuje się różne przyporządkowanie wyprowadzeń kontrolera CAN na płytce UNO R4 Minima i UNO R4 Wi-Fi. W przypadku R4 Minima są to wyprowadzenia D4:CANTX0 i D5:CANRX0, a w przypadku Wi-Fi wyprowadzenia D10:CANTX0 i D13:CANRX0. Aby uniknąć kolejnych wymagających ręcznej konfiguracji elementów oraz zachować zgodność z dwiema wersjami, do automatycznego przełączania przyporządkowania wyprowadzeń użyłem istniejącego w wersji Wi-Fi złącza baterii. Jeżeli nakładka współpracuje z R4 Wi-Fi, na środkowym wyprowadzeniu złącza występuje potencjał masy – ustala on stan niski sygnału SELW (Select Wi-Fi). Inwerter U3 generuje wówczas sygnał zanegowany SELM (Select Minima), natomiast sygnały doprowadzone do bramek trójstanowych układu U2 kluczują odpowiednio wyprowadzenia interfejsu CANRX/CANTX pomiędzy wyprowadzeniami płytki bazowej (D10, D13) a transceiverem U1. Wersja Minima nie ma złącza baterii, więc stan sygnału SELW jest wysoki i przyporządkowanie wybierane przez U2 zmienia się na zgodność z R4 Minima (D4, D5). Sygnały z kontrolera



Rysunek 1. Budowa wewnętrzna SN65HVD1050 (za notą TI)

CAN, po odpowiednim kluczowaniu, doprowadzone są do wejść transceivera CAN U1 i dodatkowo – po buforowaniu przez U4, U5 – są sygnalizowane diodami TXD/RXD. Zwora SLM, domyślnie rozwarta, umożliwia sterowanie stanem transceivera U1. Ustawienie stanu niskiego na wyprowadzeniu D7 aktywuje tryb High-Speed i pełną funkcjonalność U1, a stan wysoki D7 ustala tryb nasłuchu, w którym aktywny jest tylko odbiornik transceivera. W przypadku współpracy z R4 Wi-Fi na złącze BAT wyprowadzone jest napięcie baterii, umożliwiające podłączenie zewnętrznej baterii podtrzymującej



Rysunek 2. Schemat nakładki

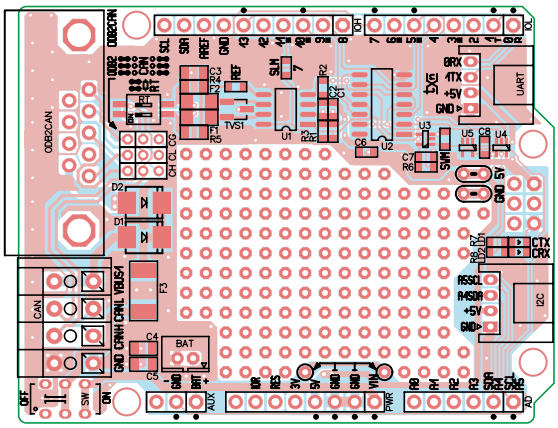
Wykaz elementów:

Rezystory:
R1...R3, R6: 10 kΩ (SMD 0603, 1%)
R4, R5: 60,4 Ω (SMD 0805, 1%, typ ERJP06F60R4V)
R7, R8: 1 kΩ (SMD 0603, 1%)

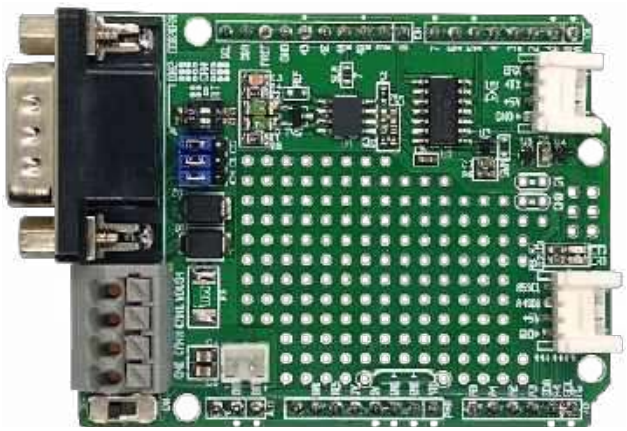
Półprzewodniki:
D1, D2: dioda Schottky'ego S2M (SMB)
LD1, LD2: dioda LED czerwona/żółta (SMD 0603)
TVS1: dioda zabezpieczająca PESD2CANFD24U (SOT-23)
U1: SN65HVD1050 (SO8)
U2: HC125 (SO14)
U3, U4, U5: 74HC1G00GW (TSSOP5)

Kondensatory:
C1: 10 µF (SMD 0603, X7R 10 V)
C2, C6...C8: 100 nF (SMD 0603, X7R 10 V)
C3: 10 nF (SMD 0805, X7R 50 V)
C4, C5: 10 µF (SMD 0805, X7R 25 V)

Pozostałe:
AD: złącze szpilkowe SIP6 2,54 mm (13...15 mm)
AUX: złącze męskie SIP3
BAT: złącze JST 2 pin. 2 mm (opcja)
CAN: złącze sprężynowe Wago (WAGO250-4)
CG, CH, CL: złącze szpilkowe SIP3 2 mm + zwory
F1, F2: bezpiecznik polimerowy 24 V/0,5 A 0,55 Ω (typ 0ZCJ0025AF2E, SMD 1206)
F3: bezpiecznik polimerowy 24 V/1,5 A 0,04 Ω (typ 0ZCG0150BF2C, SMD 1812)
I²C, UART: złącze Grove kątowe
IOH: złącze szpilkowe SIP10 2,54 mm (13...15 mm)
IOL, PWR: złącze szpilkowe SIP8 2,54 mm (13...15 mm)
ODB2CAN: gniazdo DB9 męskie (DB9RA/M)
RT: przełącznik SDIP OMRON (typ A6H-2101)
SW: przełącznik suwakowy (typ MSS-2235)



Rysunek 3. Rozmieszczenie elementów na płytce modułu



Fotografia 1. Zmontowany moduł

działanie RTC (o ile jego użycie zostanie wreszcie poprawione w oprogramowaniu). W celu ułatwienia zastosowania nakładki na złącza Grove wyprowadzono magistralę I²C (wyprowadzenia A4, A5) oraz UART (D0, D1) umożliwiające podłączenie czujników zasilanych napięciem 5 V.

Układ zmontowany został na dwustronnej płytce drukowanej zgodnej z Arduino Shield Rev3. Rozmieszczenie elementów pokazano na **rysunku 3.** Sposób montażu nie wymaga opisu. W zależności od przewidywanego zastosowania przedłużane złącza szpilkowe PWR, AD, IOL, IOH (wysokość 13...15 mm) można zastąpić „stackowalnymi" złączami Arduino, umożliwiającymi montowanie modułów w kanapki. Niewykorzystana powierzchnia płytki została przeznaczona na pole prototypowe, a oznaczone pady umożliwiają wyprowadzenie napięć zasilania (5 V) i masy (GND).

Zmontowany moduł konwertera zaprezentowano na **fotografii 1**.

Moduł nie wymaga uruchamiania. Szybkiego sprawdzenia działania nakładki można dokonać przy użyciu szkiców dostępnych w środowisku *ArduinoFile>Examples>Arduino_CAN >CANWrite/CANRead* z opisem wg *https://docs.arduino.cc/tutorials/uno-r4-minima/can*. Do tego celu potrzebne są dwa moduły UNO R4 w dowolnej wersji: na płytkach rozszerzeń aktywujemy przełącznikiem RT (obie pozycje włączone) rezystory terminujące, a następnie łączymy odcinkiem skrętki wyprowadzenia CANH i CANL złączy CAN w obu nakładkach. Na jedną z płytek R4 ładujemy szkic CANWrite, na drugą CANRead i sprawdzamy poprawność działania. Ze względu na identyczne identyfikatory USB Minima i Wi-Fi ładowanie szkicu najlepiej wykonać sekwencyjnie, podłączając najpierw jedną, później drugą płytkę, bo środowisko nie jest w stanie poprawnie wykryć dwóch modułów z identycznym VID/PID. Po resecie płytek szkice powinny nadawać i odbierać testową transmisję.

Jeżeli wszystko działa poprawnie, moduł można zastosować we własnej aplikacji.

Adam Tatuś, EP

W ofercie AVT*
AVT6040

Najważniejsze parametry:
- dopuszczalny zakres mierzonych pojemności: 0,1 pF...999 µF,
- zakresy pomiarowe: 100 pF, 1000 pF, 100 nF, 1000 nF, 100 µF, 1000 µF,
- dokładność pomiarów: 2%,
- maksymalna częstotliwość odczytów: 2 Hz,
- czas pracy na baterii AAA(1): 6 miesięcy.

(1) Przy zachowaniu założonych warunków użytkowania (szczegóły w tekście artykułu).

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

AVT5570	Przyrząd do formowania kondensatorów elektrolitycznych (EP 1/2017)
AVT614	Warsztatowy tester kondensatorów (EdW 10/2004)
AVT5003	Tester elementów elektronicznych (EP 3/2001)
AVT2404	Miernik rezystancji kondensatorów (EdW 3/2001)

* **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**.

cMeter – przenośny miernik pojemności

W praktyce każdego elektronika amatora czy profesjonalisty nieodzownym elementem codziennej rutyny są wszelkiego rodzaju naprawy, a te – jak łatwo się domyślić – wiążą się z nieustającą koniecznością testowania „podejrzanych" elementów. Jednymi z najczęściej spotykanych komponentów w sprzęcie elektronicznym są rezystory i kondensatory, przy czym o ile dość łatwym zadaniem okazuje się organoleptyczne zdiagnozowanie uszkodzenia rezystora, to już nie tak oczywiste jest postawienie poprawnej diagnozy uszkodzenia kondensatora (li tylko na podstawie wyglądu „osobnika").

Co prawda większość współczesnych multimetrów wyposażonych jest w funkcjonalność pomiaru pojemności, lecz nie wszystkie realizują ją dobrze (jeśli weźmiemy pod uwagę zarówno dokładność pomiaru, jak i funkcjonalność). Użycie przyrządu wielofunkcyjnego w tym celu wyklucza ponadto jego zastosowanie do innych zadań w tym samym czasie (co z kolei okazuje się często pożądane w praktyce). Właśnie tego typu dylematy stały się przyczynkiem do opracowania niniejszego projektu miniaturowego, przenośnego miernika pojemności o nazwie cMeter. Przyznać muszę, że na początku prac konstrukcyjnych skupiłem się na poszukiwaniu optymalnej metody pomiaru pojemności, zakładając dość duży (w gruncie rzeczy wręcz ekstremalny) zakres wartości pojemności badanych elementów. Jak się wkrótce okazało – najlepszą (a może najprostszą?) metodą jest książkowy pomiar czasu ładowania kondensatora mierzonego, który to czas jest proporcjonalny do pojemności kondensatora oraz do rezystancji szeregowej obwodu pomiarowego. Dość szybko przypomniałem sobie, że dobrym źródłem wiedzy praktycznej z wielu tematów z pogranicza elektroniki i programowania jest strona http://elm-chan.org/, czyli serwis dobrze znanego twórcy pakietów FatFS i PetitFS. Dokładnie tak było i tym razem. To właśnie we wspomnianym serwisie znalazłem ciekawą, choć bardzo prostą implementację

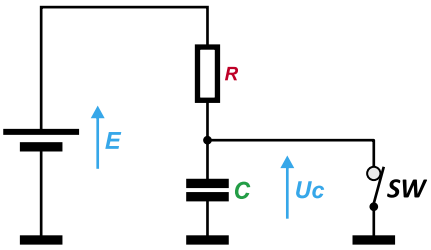
Wykaz elementów:

Rezystory:
R1: 976 kΩ 1% (SMD 0603)
R2: 562 kΩ 1% (SMD 0603)
R3: 10 kΩ (SMD 0603)
R4: 10 kΩ 1% (metalizowany, mini-MELF 0204)
R5: 100 Ω 1% (metalizowany, mini-MELF 0204)
R6, R7: 39 kΩ 1% (metalizowany, mini-MELF 0204)
R8: 680 Ω 1% (metalizowany, mini-MELF 0204)
R9: 3,3 MΩ 1% (metalizowany, mini-MELF 0204)
R10...R17: 330 Ω (SMD 0603)

Kondensatory:
C1, C2: ceramiczny X7R 100 nF (SMD 0603)
C3, C4: ceramiczny X7R 10 µF (SMD 0603)

Półprzewodniki:
U1: ATtiny44 (SO14)
U2: MCP1640T-I/CHY (SOT23-6)
U3: M74HC4094 (SO16)
LED: wyświetlacz 7-segmentowy LED typu OSK4039A-IG lub podobny w wybranym kolorze
T1: BC807 (SOT23)
T2...T5: MMBTRA105SS (SOT23)

Pozostałe:
L1: dławik drutowy SMD 4,7 µH typu WLPN303015M4R7PB (SMD 3×3 mm)
PWR: mikroprzełącznik TACT SMD typu TACTM-67N-F NINIGI (wysokość 7...9 mm)
BATT: koszyczek baterii AAA typu 1021 KEYSTONE



Rysunek 1. Typowy, szeregowy obwód RC, który odpowiada za ładowanie kondensatora

wspomnianego wcześniej mechanizmu pomiaru pojemności – i to właśnie ona stała się inspiracją do stworzenia niniejszego projektu. Zacznijmy jednak od podstaw. Na **rysunku 1** pokazano typowy, szeregowy obwód RC, odpowiadający za ładowanie kondensatora.

W obwodzie z rysunku 1 otwarcie przełącznika SW rozpoczyna proces ładowania kondensatora, podczas którego napięcie na zaciskach kondensatora wyraża się następującym wzorem:

$$U_c = E \cdot (1 - e^{\frac{-t}{R \cdot C}})$$

gdzie
U_c – napięcie na zaciskach kondensatora [V],
E – napięcie idealnego źródła zasilania [V],
e – liczba Eulera (zwana również liczbą Nepera),
t – czas ładowania kondensatora [s],
R – rezystancja rezystora szeregowego [Ω],
C – pojemność kondensatora [F].

Jak widać, jeśli przyjmiemy stałe wartości E oraz R, napięcie na zaciskach mierzonego kondensatora zależne będzie **wyłącznie** od jego pojemności (C) oraz czasu ładowania (t). Z powyższego wzoru możemy wyznaczyć czas ładowania dla konkretnego napięcia U_c, który wyraża się następującym wzorem:

$$t = R \cdot C \cdot ln(1 - \frac{U_c}{E})$$

Wyrażając Uc jako ułamek E (ułamek napięcia zasilania), czyli przyjmując

$$U_c = THRESH \cdot E$$

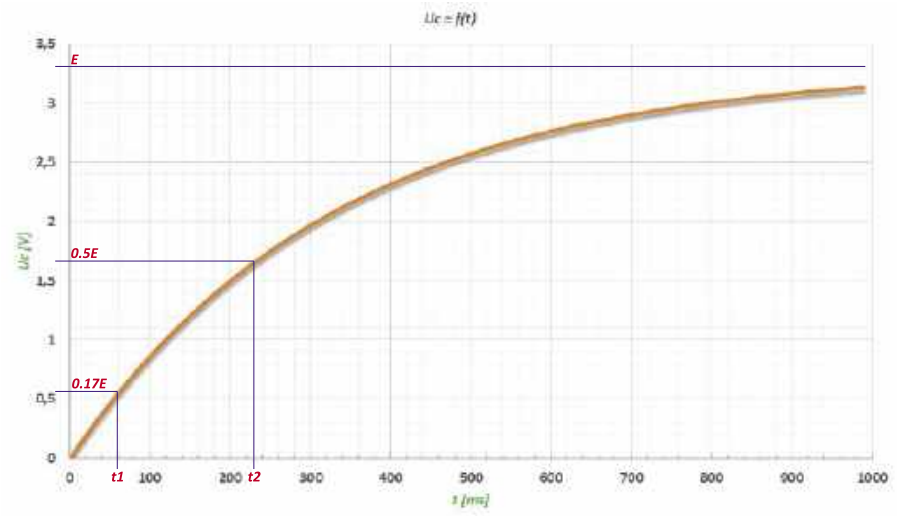
otrzymujemy:

$$t = -R \cdot C \cdot ln(1 - THRESH)$$

Przekształcając powyższy wzór w celu otrzymania pojemności, dochodzimy do poniższego wyrażenia:

$$C = \frac{-t}{R \cdot ln(1 - THRESH)}$$

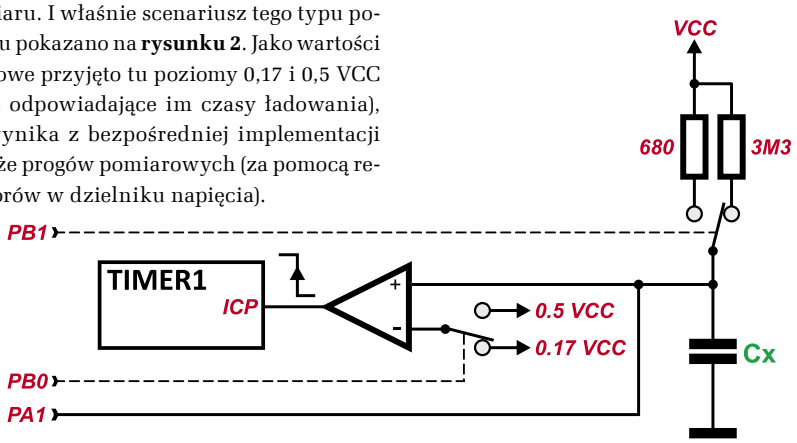
Niezbędna do ustalenia pojemność kondensatora mierzonego wynika zatem wyłącznie ze znanej wartości rezystora szeregowego, napięcia progowego (również znanego) i czasu ładowania (łatwego do zmierzenia), czyli – zakładając stałe wartości R i *THRESH* – proporcjonalna jest **wyłącznie** do czasu ładowania. I to jest clou mechanizmu pomiarowego – które pozornie mogłoby wydawać się banalne, ale nie wyczerpuje jeszcze treści naszych rozważań. Proponowane powyżej



Rysunek 2. Wykres ładowania kondensatora

mechanizmy zakładają bowiem, iż początkowe napięcie na zaciskach kondensatora (którego pojemność chcemy zmierzyć) równe jest zero. Takie zjawisko jednak **NIGDY** nie występuje w rzeczywistych układach, gdyż po pierwsze – każdy rzeczywisty kondensator ma pewną rezystancję szeregową, która „spowalnia" proces jego rozładowania, a po drugie – rozładowanie takiego kondensatora do wspomnianego zera lub naładowanie go do wartości napięcia zasilania wymagałoby sporo czasu. Oczekiwanie z kolei byłoby niepożądane w trakcie procesu pomiarowego, gdyż – jak łatwo się domyślić – chcemy mierzoną wartość otrzymać niemalże natychmiast a nie np. po 10 sekundach, prawda? Tak naprawdę zresztą, kto gwarantuje nam, że mierzony kondensator został rozładowany do ZERA? Jak sobie poradzić z tym problemem? Dość łatwo! Należy zmierzyć czasy ładowania mierzonego kondensatora dla pewnych dwóch wartości progowych napięcia (*THRESH*) i na podstawie różnicy tych czasów obliczyć wartość mierzonej pojemności. Same wartości progowe (*THRESH*) nie są tutaj krytyczne, lecz powinny być od siebie na tyle „oddalone", by, po pierwsze, zapewnić odpowiednią dokładność pomiaru, a po drugie – maksymalnie skrócić czas tegoż pomiaru. I właśnie scenariusz tego typu pomiaru pokazano na **rysunku 2**. Jako wartości progowe przyjęto tu poziomy 0,17 i 0,5 VCC (oraz odpowiadające im czasy ładowania), co wynika z bezpośredniej implementacji tychże progów pomiarowych (za pomocą rezystorów w dzielniku napięcia).

A skoro mowa o wartościach *progowych*… czy coś Wam „świta" w głowach? Najłatwiej jest użyć komparatora analogowego z odpowiednio ustawionym napięciem odniesienia (V_{REF}), by stosowny pomiar mógł być wykonany w najprostszy sposób. A jeśli mowa o komparatorze analogowym, to nie sposób nie użyć tego wbudowanego w mikrokontroler, zwłaszcza że dość często potrafi on wyzwalać zdarzenie przechwycenia pracującego licznika (Timer1), a co za tym idzie – może w prosty sposób „generować" znaczniki czasowe. Tak, właśnie tym tropem pójdziemy! Ale istnieje jeszcze jeden aspekt omawianego mechanizmu, nie mniej istotny. Skoro chcemy mierzyć pojemność kondensatorów o dość dużym zakresie (powiedzmy – od pojedynczych pF do setek µF), to czas pomiaru dla tych największych (przy ustalonej wartości rezystancji szeregowej R) może okazać się bardzo długi (rzędu dziesiątek sekund), czego z pewnością byśmy nie chcieli. Jak rozwiązać ten problem? Znowu, w najprostszy sposób. Należy zmieniać wartość rezystancji szeregowej (czyli de facto zakres czasów ładowania kondensatora), tak by nawet dla „dużych" kondensatorów czas ten był na akceptowalnym poziomie. Wniosek z naszych rozważań jest taki, że musimy mieć możliwość



Rysunek 3. Funkcjonalna wizualizacja pełnego mechanizmu pomiarowego urządzenia cMeter

po pierwsze – przełączania wartości rezystora szeregowego, a po drugie – zmiany napięcia referencyjnego, aby móc dokonać pomiaru wspomnianej pojemności, przy zachowaniu odpowiedniej dokładności tychże pomiarów. Funkcjonalną wizualizację pełnego mechanizmu pomiarowego, o którym napisałem powyżej, pokazano na **rysunku 3**.

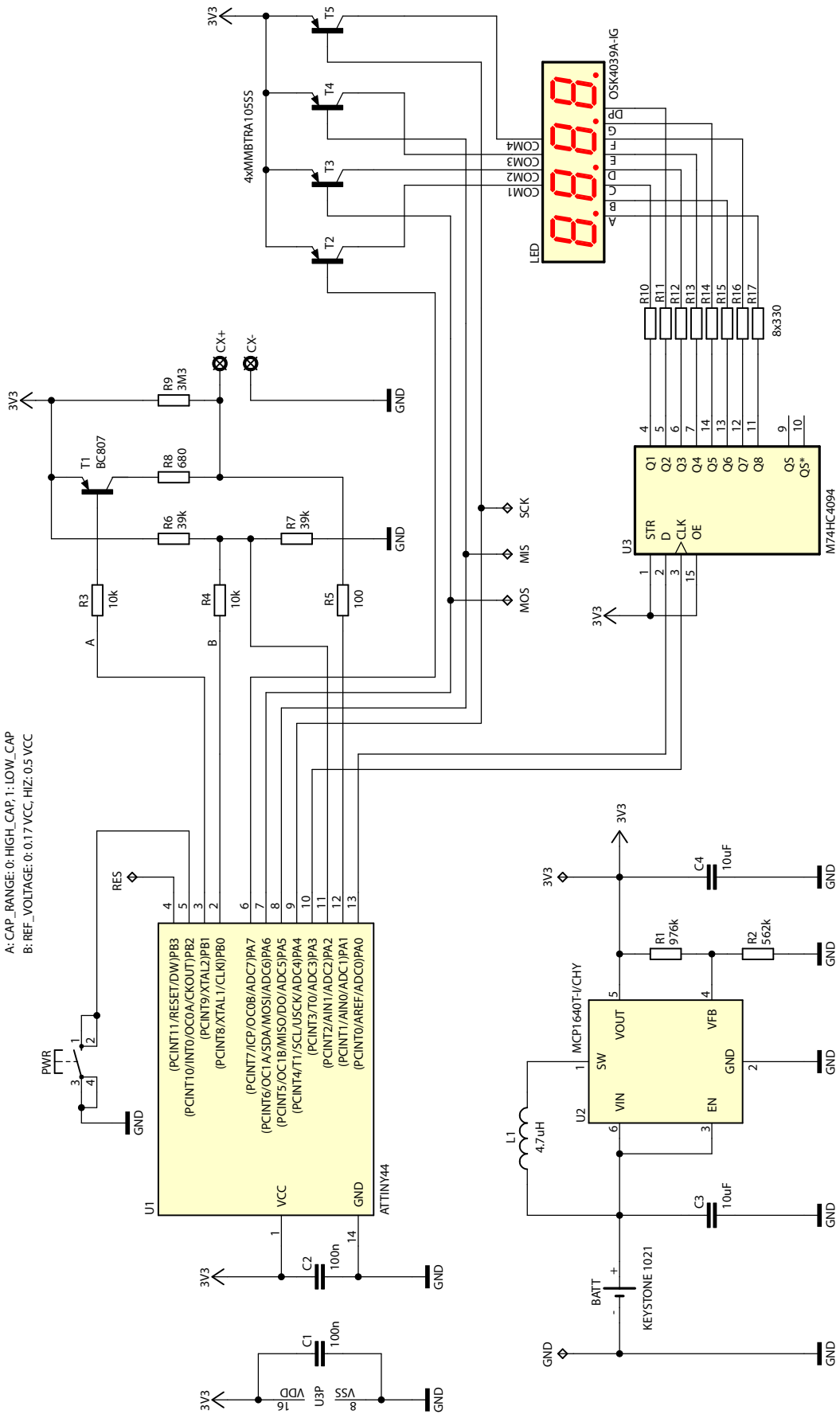
I właśnie na bazie powyższych założeń powstał projekt urządzenia cMeter, którego schemat ideowy znalazł się na **rysunku 4**.

Jak widać, zaprojektowano bardzo prosty system mikroprocesorowy. Jego serce stanowi niewielki mikrokontroler ATtiny44 firmy Microchip (dawniej Atmel), taktowany wewnętrznym oscylatorem RC o częstotliwości 8 MHz i realizujący całą założoną funkcjonalność urządzenia. Mikrokontroler nasz realizuje następujące zadania:

- steruje pracą szeregowego rejestru przesuwnego 74HC4094 (wyprowadzenia PA3→Clock, PA0→Data), za pomocą którego obsługuje 7-segmentowy wyświetlacz LED o organizacji 4 znaków w konfiguracji wspólnej anody (wyprowadzenia PA7…PA4 mikrokontrolera),
- steruje zewnętrznymi obwodami pomiarowymi (wyprowadzenie PB1 jest odpowiedzialne za wybór wartości rezystora szeregowego [3,3 MΩ lub 680 Ω], zaś PB0 umożliwia wybór napięcia referencyjnego [0,17 VCC lub 0,5 VCC]),
- steruje pracą wbudowanego komparatora analogowego, stanowiącego podstawę mechanizmu pomiarowego (wyprowadzenia PA2, PA1), inicjując tym samym proces rozładowania i następującego po nim ładowania kondensatora mierzonego (PA1=1/0).

Warto już w tym momencie podkreślić, iż wszystkie rezystory w torze pomiarowym przewidziano jako rezystory metalizowane w obudowach mini-MELF 0204, by do minimum zredukować ewentualne szumy wprowadzane przez typowe i tanie elementy cienko– i grubowarstwowe (thin film, thick film).

Wybór mikrokontrolera ATtiny44 oraz podłączonego do jego wyprowadzeń rejestru



Rysunek 4. Schemat ideowy urządzenia cMeter

szeregowego 74HC4094 mógłby się wydawać dość wątpliwy, jeśli bierzemy pod uwagę, że bez problemu można byłoby wybrać mikrokontroler o odpowiedniej liczbie portów wyjściowych, zamiast stosować przedmiotowy mikrokontroler i dodatkowy rejestr przesuwny. To jednak tylko pozory! Nie chciałem bowiem stosować mikrokontrolera o większej liczbie (niepotrzebnych) wyprowadzeń, a co za tym idzie – o niewygodnej do lutowania dla amatora obudowie (TQFP32). Istnieje jednak drugi, ważniejszy powód: zastosowanie rejestru przesuwnego 74HC4094 zdecydowanie upraszczało projekt obwodu drukowanego, a układ tego rodzaju kosztuje około 1,50 zł. Wspomniane wcześniej wspólne anody wyświetlacza LED sterowane są poprzez proste klucze tranzystorowe T2… T5 z uwagi na dość duże prądy o wartościach rzędu 32 mA (8×4 mA na segment). Z kolei katody naszych elementów LED obsługiwane są przez wyprowadzenia rejestru przesuwnego i, jak już można się domyślić, do ich obsługi (podobnie jak wspólnych anod) zastosowano doskonale znany mechanizm multipleksowania. Jest to typowe rozwiązanie problemu tego typu, a polega na sekwencyjnym sterowaniu kolejnych cyfr wyświetlacza LED. Wspomniana procedura, wykonywana dostatecznie szybko (w naszym wypadku 60 razy na sekundę dla każdej wspólnej anody), pozwala na obsłużenie 32 segmentów wyświetlacza LED przy udziale wyłącznie 6 wyprowadzeń mikrokontrolera. Już teraz nadmienię, że użyjemy w tym celu układu czasowo-licznikowego Timer0 wbudowanego w strukturę mikrokontrolera, który pracował będzie w trybie CTC i wywoływał stosowne przerwanie systemowe (od porównania) 240 razy na sekundę (czyli 60 razy dla każdej wspólnej anody), obsługując właściwy mechanizm multipleksowania. Wróćmy jednak do schematu ideowego naszego urządzenia, gdyż kilku niezbędnych słów wyjaśnienia wymaga blok zasilający. Zdecydowałem się na zastosowanie popularnej baterii typu AAA o pojemności w granicach 1300 mAh (w przypadku dobrych baterii alkalicznych lub litowych), a wybór ten podyktowany był łatwą dostępnością ogniw i ich niską ceną. Z uwagi na to konieczne stało się użycie prostej, acz nowoczesnej przetwornicy step-up pod postacią układu scalonego MCP1640 firmy Microchip, która dostarcza napięcie wyjściowe rzędu 3,3 V (regulowane dzielnikiem rezystancyjnym) już przy napięciu wejściowym na poziomie 0,65 V i zapewnia niski prąd spoczynkowy w granicach 1 µA oraz wysoką sprawność, co nie jest bez znaczenia dla typu zastosowanego źródła napięcia zasilającego. Aby ocenić, jak długo urządzenie cMeter pracować będzie na pojedynczej baterii AAA, należy zastanowić się, z jakich etapów składa się cykl jego pracy i jakiej wielkości prądy pobiera wtedy ze źródła zasilania. Przystępując do obliczeń, przyjąłem następujący podział cyklu pracy urządzenia:

- czas trybu Power Down (uśpienia), który trwa z dużym przybliżeniem 24 h/dobę i podczas którego pobierany jest prąd rzędu 200 µA. Dodajmy, że „większość” tego prądu to prąd spoczynkowy przetwornicy, który w przypadku dużej różnicy między napięciem wejściowym a wyjściowym, co ma miejsce w naszym przykładzie, znacznie przekracza (o rząd wielkości) deklarowany przez producenta, minimalny prąd spoczynkowy oraz prąd pobierany przez obwód pomiarowy – rezystorowy dzielnik napięcia przyłączony na stałe pomiędzy bieguny zasilania.
- czas wykonywania i wyświetlania pomiarów, który zakładamy na poziomie 30 s i podczas którego pobierany jest średni prąd na poziomie 20 mA.

Założono ponadto, iż wybudzanie urządzenia i następujący po nim pomiar ma miejsce 20 razy na dobę, co oznacza, że urządzenie używane jest maksymalnie 10 minut dziennie. Przy założeniach jak wyżej otrzymano teoretycznie, niespełna 6 miesięcy pracy na pojedynczej baterii AAA, co wydaje się wartością co najmniej satysfakcjonującą. I na koniec, wyłącznie dla porządku dodam,

Ustawienia Fuse-bitów:
CKSEL3:0: 0010
SUT1:0: 10
CKDIV8: 1
CKOUT: 1
DWEN: 1
EESAVE: 0

```
//Definicje portów rejestru przesuwnego
#define SERIAL_PORT PORTA
#define SERIAL_DDR DDRA
#define SERIAL_DAT_NR PA0
#define SERIAL_CLK_NR PA3

//Porty rejestru przesuwnego (DAT, CLK), jako wyjściowe ze stanem 0
#define SERIAL_PORT_AS_OUTPUT SERIAL_DDR |= (1<<SERIAL_DAT_NR)|(1<<SERIAL_CLK_NR)

#define SERIAL_DAT_SET SERIAL_PORT |= (1<<SERIAL_DAT_NR)
#define SERIAL_DAT_RESET SERIAL_PORT &= ~(1<<SERIAL_DAT_NR)
#define SERIAL_CLK_SET SERIAL_PORT |= (1<<SERIAL_CLK_NR)
#define SERIAL_CLK_RESET SERIAL_PORT &= ~(1<<SERIAL_CLK_NR)
#define SERIAL_CLK_TICK SERIAL_CLK_SET; SERIAL_CLK_RESET

//Definicje konfiguracji poszczególnych segmentów (katod) rejestru przesuwnego
#define SEG_A (1<<7)
#define SEG_B (1<<5)
#define SEG_C (1<<0)
#define SEG_D (1<<2)
#define SEG_E (1<<3)
#define SEG_F (1<<6)
#define SEG_G (1<<4)
#define SEG_DP (1<<1)

//Definicje portu wspólnych anod – tranzystory sterujące
#define COM_PORT PORTA
#define COM_DDR DDRA

//Definicje konfiguracji poszczególnych wspólnych anod
#define COM_DIG1 PA7
#define COM_DIG2 PA6
#define COM_DIG3 PA5
#define COM_DIG4 PA4

//Port wspólnych anod jako port wyjściowy
#define COM_AS_OUTPUT COM_DDR |= (1<<COM_DIG4)|(1<<COM_DIG3)|(1<<COM_DIG2)|(1<<COM_DIG1)
//Wszystkie wspólne anody wyłączone ("1", gdyż sterujemy bazami tranzystorów PNP)
#define COM_BLANK COM_PORT |= (1<<COM_DIG4)|(1<<COM_DIG3)|(1<<COM_DIG2)|(1<<COM_DIG1)

//Definicje dla Timera0 odpowiedzialnego za multipleksowanie wyświetlacza LED
#define START_MUX_TIMER TCCR0B = (1<<CS02)      //Preskaler = 256
#define STOP_MUX_TIMER TCCR0B = 0

//Definicja bitu odpowiedzialnego za kropkę dziesiętną
#define DP_BIT 0b10000000

//Indexy znaków specjalnych
#define CHAR_space 10
#define CHAR_p 11
#define CHAR_n 12
#define CHAR_u 13
#define CHAR_b 14
#define CHAR_A 15
#define CHAR_t 16
#define CHAR_h 17
#define CHAR_i 18
#define CHAR_g 19
#define CHAR_C 20
#define CHAR_L 21
#define CHAR_r 22
#define CHAR_o 23
#define CHAR_d 24
#define CHAR_E 25
#define CHAR_F 26

//Deklaracje zmiennych globalnych
extern volatile uint8_t Digit[4];              //Zmienna przechowująca wartość wyświetlaną na wyświetlaczu LED
extern volatile uint8_t readyForUpdate;        //Zezwolenie na atomową zmianę zmiennych
```

Listing 1. Plik nagłówkowy mechanizmu multipleksowania

że mikrokontroler nasz korzysta również z wbudowanego w swoją strukturę przetwornika analogowo-cyfrowego, dzięki któremu dokonuje ustawicznego pomiaru napięcia zasilania systemu mikroprocesorowego i w przypadku jego nieodpowiedniej wartości podejmuje stosowne czynności. Tyle w kwestiach funkcjonalnych – przejdźmy zatem do zagadnień implementacyjnych.

Mam świadomość, że prezentowana koncepcja nie stanowi być może *rocket science* ani rozwiązania na wskroś uniwersalnego, jednak chciałem pokazać Wam, jak w efektywny i efektowny sposób ogarnąć tego rodzaju zagadnienie programistyczne, czyniąc sam proces programowania niezmiernie przyjemnym. Nieskromnie powiem, że w moim przekonaniu właśnie w ten przejrzysty sposób powinno się konstruować moduły obsługi danych peryferiów, gdyż jakakolwiek modyfikacja sprowadza się wtedy do kosmetycznych i prostych do wykonania zmian. Na początek przeanalizujmy plik nagłówkowy mechanizmu multipleksowania, który pokazano na **listingu 1**, a dzięki któremu porządkujemy późniejszy kod źródłowy, czyniąc go bardzo czytelnym i jednocześnie upraszczając proces wprowadzania zmian. Plik ten zarówno definiuje główne ustawienia sprzętowe, jak i wprowadza niezbędne zmienne – zadeklarowano w nim szereg zmiennych globalnych (typu volatile z uwagi na ich użycie tak w programie głównym, jak i funkcji ISR), przechowujących ustawienia poszczególnych elementów wyświetlacza LED. Już na tym etapie musimy zdefiniować kilka stałych opisujących wzorce znaków czy też upraszczających dostęp do portów procesora, gdyż zależy nam, by nasza procedura obsługi przerwania multipleksująca wyświetlacz była jak najkrótsza. Definicje, o których mowa, pokazano na **listingu 2**.

Jak widać, wspomniane definicje zostały umieszczone w pamięci RAM mikrokontrolera. Jest to pewnego rodzaju „marnotrawstwo”, gdyż stałe te z powodzeniem można (a nawet wypada) umieścić w pamięci Flash mikrokontrolera, aby nie marnować cennej pamięci RAM (zwłaszcza że wartości tych stałych nasz kompilator i tak musi umieścić, a następnie odczytać z tejże pamięci Flash na starcie działania programu). Dokładnie tak postępowałem dotychczas, pisząc oprogramowanie embedded – pamiętajmy jednak, że dostęp do pamięci Flash jest nieco wolniejszy niż odczyt stałych z pamięci RAM (dokładnie 5 taktów zegara zamiast 2). W związku z tym zdecydowałem się na powyższe rozwiązanie, zwłaszcza że wykorzystanie pamięci RAM w naszej aplikacji utrzymuje się na poziomie 20%. Wartość ta stanowi pozornie niewielki przyrost szybkości, ale zawsze coś. Skądinąd idea taka jest zgodna z podejściem twórców Androida, charakteryzowanym przez frazę: „dlaczego nieużywana pamięć RAM ma leżeć odłogiem”? Abstrahując już od celowości i sensowności takiego postępowania, brnijmy dalej we wskazanym kierunku. Pora na zaprezentowanie funkcji konfigurującej zarówno mechanizm multipleksowania, jak i niezbędne ustawienia sprzętowe, której ciało znajduje się na **listingu 3**.

Dalej, na **listingu 4** uwidoczniono funkcję obsługi przerwania od porównania wartości licznika Timer0 z rejestrem OCR0A,

```
//Definicje znaków wyświetlacza LED (aktywny stan "0", gdyż sterujemy katodami diod LED)
const uint8_t digPattern[] =
{
    (uint8_t) ~(SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F),      //0
    (uint8_t) ~(SEG_B|SEG_C),                              //1
    (uint8_t) ~(SEG_A|SEG_B|SEG_D|SEG_E|SEG_G),            //2
    (uint8_t) ~(SEG_A|SEG_B|SEG_C|SEG_D|SEG_G),            //3
    (uint8_t) ~(SEG_B|SEG_C|SEG_F|SEG_G),                  //4
    (uint8_t) ~(SEG_A|SEG_C|SEG_D|SEG_F|SEG_G),            //5
    (uint8_t) ~(SEG_A|SEG_C|SEG_D|SEG_E|SEG_F|SEG_G),      //6
    (uint8_t) ~(SEG_A|SEG_B|SEG_C),  //7
    (uint8_t) ~(SEG_A|SEG_B|SEG_C|SEG_D|SEG_E|SEG_F|SEG_G),//8
    (uint8_t) ~(SEG_A|SEG_B|SEG_C|SEG_D|SEG_F|SEG_G),      //9
    0xFF,                                                  //Wyświetlacz wygaszony (10)
    (uint8_t) ~(SEG_A|SEG_B|SEG_G|SEG_F|SEG_E),            //p - 11
    (uint8_t) ~(SEG_E|SEG_G|SEG_C),                        //n - 12
    (uint8_t) ~(SEG_C|SEG_D|SEG_E),                        //u - 13
    (uint8_t) ~(SEG_G|SEG_F|SEG_E|SEG_D|SEG_C),            //b - 14
    (uint8_t) ~(SEG_A|SEG_B|SEG_C|SEG_E|SEG_F|SEG_G),      //A - 15
    (uint8_t) ~(SEG_C|SEG_D|SEG_E|SEG_F|SEG_G),            //t - 16
    (uint8_t) ~(SEG_C|SEG_E|SEG_F|SEG_G),                  //h - 17
    (uint8_t) ~(SEG_E),                                    //i - 18
    (uint8_t) ~(SEG_A|SEG_B|SEG_C|SEG_D|SEG_F|SEG_G),      //g - 19
    (uint8_t) ~(SEG_A|SEG_D|SEG_E|SEG_F),                  //C - 20
    (uint8_t) ~(SEG_D|SEG_E|SEG_F),                        //L - 21
    (uint8_t) ~(SEG_E|SEG_G),                              //r - 22
    (uint8_t) ~(SEG_C|SEG_D|SEG_E|SEG_G),                  //o - 23
    (uint8_t) ~(SEG_B|SEG_C|SEG_D|SEG_E|SEG_G),            //d - 24
    (uint8_t) ~(SEG_A|SEG_D|SEG_E|SEG_F|SEG_G),            //E - 25
    (uint8_t) ~(SEG_A|SEG_E|SEG_F|SEG_G)                   //F - 26
};

//Definicje dla portu sterującego wspólnymi anodami wyświetlaczy LED
//(aktywny stan "0", gdyż sterujemy bazami tranzystorów PNP)
const uint8_t comPattern[] =
{
    (uint8_t) ~(1<<COM_DIG1),              //Wspólna anoda cyfry 1 (pierwsza z lewej)
    (uint8_t) ~(1<<COM_DIG2),              //Wspólna anoda cyfry 2
    (uint8_t) ~(1<<COM_DIG3),              //Wspólna anoda cyfry 3
    (uint8_t) ~(1<<COM_DIG4),              //Wspólna anoda cyfry 4 (pierwsza z prawej)
};
```

Listing 2. Definicje niezbędnych stałych mechanizmu multipleksowania

```
void initMultiplex(void)
{
    //Porty wspólnych anod i interfejsu szeregowego jako wyjściowe ze stanami
    //nieaktywnymi na wyjściach
    SERIAL_PORT_AS_OUTPUT;
    COM_BLANK;
    COM_AS_OUTPUT;

    //Konfiguracja Timera0 w celu generowania przerwania do obsługi multipleksowania
    //wyświetlacza LED (240 Hz)
    TCCR0A = (1<<WGM01);                    //Tryb CTC
    OCR0A = 129;                            //240 Hz (ISR co 4.17 ms)
    START_MUX_TIMER;                        //Uruchomienie Timera0
    TIMSK0 = (1<<OCIE0A);                   //Uruchomienie przerwania Timer0 Output
                                            //Compare Match A
}
```

Listing 3. Funkcja konfigurująca mechanizm multipleksowania

```
ISR(TIM0_COMPA_vect)
{
    static uint8_t Nr;                                  //Numer kolejnej cyfry przeznaczonej do wyświetlenia
    uint8_t Dig = Digit[Nr];                            //Optymalizacja volatile


    //Wyłączenie wspólnych anod wyświetlaczy LED
    COM_BLANK;
    //Wystawienie wzoru na port katod (rejestr przesuwny) uwzględniające bit kropki dziesiętnej (bit 7 -> DP_BIT)
    if(Dig & DP_BIT) serialSend(digPattern[Dig & (~DP_BIT)] & (~SEG_DP)); else serialSend(digPattern[Dig]);
    //Włączenie odpowiedniej wspólnej anody (aktywny stan "0")
    COM_PORT &= comPattern[Nr];
    //Wybranie kolejnej wspólnej anody
    Nr = (Nr+1) & 0x03;
    //Zezwolenie na atomową zmianę zmienej Digit[] w funkcji Main
    if(Nr == 0) readyForUpdate = 1; else readyForUpdate = 0;
}
```

Listing 4. Funkcja obsługi przerwania realizująca mechanizm multipleksowania

```
inline void serialSend(uint8_t Byte)
{
    //Wysyłamy bajt do rejestru przesuwnego (zbocze rosnące na CLK) przesuwając kolejne
    //bity na wyjście DAT począwszy od bitu MSB
    if(Byte & 0x80) SERIAL_DAT_SET; else SERIAL_DAT_RESET; SERIAL_CLK_TICK;
    if(Byte & 0x40) SERIAL_DAT_SET; else SERIAL_DAT_RESET; SERIAL_CLK_TICK;
    if(Byte & 0x20) SERIAL_DAT_SET; else SERIAL_DAT_RESET; SERIAL_CLK_TICK;
    if(Byte & 0x10) SERIAL_DAT_SET; else SERIAL_DAT_RESET; SERIAL_CLK_TICK;
    if(Byte & 0x08) SERIAL_DAT_SET; else SERIAL_DAT_RESET; SERIAL_CLK_TICK;
    if(Byte & 0x04) SERIAL_DAT_SET; else SERIAL_DAT_RESET; SERIAL_CLK_TICK;
    if(Byte & 0x02) SERIAL_DAT_SET; else SERIAL_DAT_RESET; SERIAL_CLK_TICK;
    if(Byte & 0x01) SERIAL_DAT_SET; else SERIAL_DAT_RESET; SERIAL_CLK_TICK;
}
```

Listing 5. Funkcja odpowiedzialna za przesłanie bajtu danych do rejestru przesuwnego

odpowiedzialną za realizację mechanizmu multipleksowania wyświetlacza. W implementacji tej wzięto pod uwagę obsługę kropki dziesiętnej wyświetlacza LED.

I na sam koniec, na **listingu 5**, pokazano funkcję odpowiedzialną za przesłanie bajtu danych do rejestru przesuwnego.

Warto przez chwilę zastanowić się nad znaczeniem nieopisanej wcześniej zmiennej readyForUpdate. Jest to zmienna, która funkcji głównej aplikacji użytkownika wskazuje

```
//Ustawienia portów odpowiedzialnych za proces pomiaru pojemności
#define MEAS_PORT PORTB
#define MEAS_DDR DDRB
#define MEAS_CAP_RNG_NR PB1
#define MEAS_REF_VLG_NR PB0

#define MEAS_SELECT_CAP_AS_OUTPUT MEAS_DDR |= (1<<MEAS_CAP_RNG_NR)
#define MEAS_SELECT_LOW_CAP MEAS_PORT |= (1<<MEAS_CAP_RNG_NR)
#define MEAS_SELECT_HIGH_CAP MEAS_PORT &= ~(1<<MEAS_CAP_RNG_NR)
#define MEAS_LOW_CAP_IS_SELECTED (MEAS_PORT & (1<<MEAS_CAP_RNG_NR))

#define MEAS_SELECT_LOW_REF MEAS_DDR |= (1<<MEAS_REF_VLG_NR)          //Port PB0, jako wyjściowy ze stanem 0 (0.17 VCC)
#define MEAS_SELECT_HIGH_REF MEAS_DDR &= ~(1<<MEAS_REF_VLG_NR)        //Port PB0, jako HiZ (0.5 VCC)
#define MEAS_LOW_REF_IS_SELECTED (MEAS_DDR & (1<<MEAS_REF_VLG_NR))
#define MEAS_REF_PULL_UP_ON MEAS_PORT |= (1<<MEAS_REF_VLG_NR)         //Podciągnięcie portu referencji pod VCC (w Power Down)
#define MEAS_REF_PULL_UP_OFF MEAS_PORT &= ~(1<<MEAS_REF_VLG_NR)

//Port rozładowania mierzonego kondensatora (tym samym port komparatora AIN0)
#define DISCHARGE_DDR DDRA
#define DISCHARGE_PORT PORTA
#define DISCHARGE_NR PA1
#define MEAS_DISCHARGE_CAP DISCHARGE_DDR |= (1<<DISCHARGE_NR)          //Port PA1, jako wyjściowy ze stanem 0
                                                                       //(rozładowanie kondensatora mierzonego)
#define MEAS_CHARGE_CAP DISCHARGE_DDR &= ~(1<<DISCHARGE_NR)            //Port PA1, jako HiZ (ładowanie kondensatora mierzonego)
#define MEAS_DISCHARGE_PULL_UP_ON DISCHARGE_PORT |= (1<<DISCHARGE_NR)  //Podciągnięcie portu rozładowania pod VCC (w Power Down)
#define MEAS_DISCHARGE_PULL_UP_OFF DISCHARGE_PORT &= ~(1<<DISCHARGE_NR)

//Ustawienia Timera1 odpowiedzialnego za pomiar pojemności
#define MEAS_ENABLE_ISRS TIMSK1 = (1<<ICIE1)|(1<<TOIE1)                //Zezwolenie na przerwania od przechwycenia i przepełnienia licznika
                                                                       //Timer1
#define MEAS_DISABLE_ISRS TIMSK1 = 0                                   //Wyłączenie wszystkich przerwań licznika Timer1
#define MEAS_CLEAR_ISRS_FLAGS TIFR1 = (1<<ICF1)|(1<<TOV1)              //Wyczyszczenie flag przerwań od przechwycenia i przepełnienia
                                                                       //licznika Timer1
#define MEAS_CLEAR_COUNTER TCNT1 = 0                                   //Wyzerowanie licznika Timer1
#define MEAS_START_COUNTER TCCR1B = (1<<ICES1)|(1<<CS10)               //Uruchomienie i konfiguracja licznika Timer1: Preskaler = 1,
                                                                       //przechwycenie na zboczu rosnącym
#define MEAS_STOP_COUNTER TCCR1B = (1<<ICES1)                          //Zatrzymanie licznika Timer1

//Ustawienia komparatora analogowego ACO
#define ANALOG_COM_INPUT_CAP_ENABLE ACSR |= (1<<ACIC)

//Flagi procesu pomiarowego
#define FLAG_IDLE 0                                                    //Flaga bezczynności procesu pomiarowego
#define FLAG_RDY_LOW 1                                                 //Flaga gotowości danych pomiarowych dla mniejszego kondensatora
#define FLAG_RDY_HIGH 2                                                //Flaga gotowości danych pomiarowych dla mniejszego kondensatora
#define FLAG_IN_PROGRESS 3                                             //Flaga trwającego procesu pomiarowego
#define FLAG_OVF 4                                                     //Flaga przekroczenia zakresu pomiarowego
#define FLAG_ADD_DELAY 5                                               //Flaga konieczności uwzględnienia dodatkowego czasu na rozładowanie
                                                                       //kondensatora (dla dużych jego wartości)

//Maksymalna liczba przepełnień licznika Timer1 dla obu wartości rezystancji ładowania kondensatora mierzonego
#define LOW_CAP_MAX_OVFS 27                                            //100 nF
#define HIGH_CAP_MAX_OVFS 57                                           //1000 uF

//Definicje stałych kondensatora zerowego 25pF (dopuszczalnej pojemności pasożytniczej)
#define ZERO_CAP_PF 25                                                 //Jednostka pF
#define ZERO_CAP_PULSES ((ZERO_CAP_PF*1339UL)/100)                     //Inaczej: CpF/0.074653

//Definicje stałych kondensatora referencyjnego 47nF
#define REF_CAP_NF 47                                                  //Jednostka nF
#define REF_CAP_PULSES (REF_CAP_NF*13395UL)                            //Inaczej: CnF/0.000074653

//Definicja stałej kondensatora pierwszego zakresu pomiarowego 99.9nF (liczba impulsów do naładowania do 0.5 VCC)
#define FIRST_STAGE_CAP 999                                            //Jednostka 0.1 nF
#define FIRST_STAGE_CAP_PULSES (1830UL*FIRST_STAGE_CAP)

//Definicje stałych maksymalnego kondensatora, dla którego nie trzeba uwzględniać dodatkowego czasu na jego rozładowanie
#define HIGH_CAP_UF 440                                                //Jednostka uF
#define HIGH_CAP_PULSES (HIGH_CAP_UF*2757UL)                           //Inaczej: CuF/0.00036229

//Flagi funkcji startującej pomiar (wybór zakresu pomiarowego)
#define LOW_CAP 0
#define HIGH_CAP 1


//Prototypy funkcji
void initMeasurement(void);
void startMeasurement(uint8_t capType);
void stopMeasurement(void);


//Zmienna modułu
extern volatile uint8_t measFlag;                                      //Flaga procesu pomiarowego
extern volatile uint8_t Overflows;                                     //Liczba przepełnień licznika Timer1
extern volatile uint32_t Counts;                                       //Liczba zliczonych impulsów
```

Listing 6. Plik nagłówkowy mechanizmu pomiarowego

moment atomowej aktualizacji zmiennych volatile procedury obsługi przerwania mechanizmu multipleksowania. Potrzeba wprowadzenia takiej zmiennej wynikała z konieczności synchronizacji aktualizacji zmiennych (dokonywanej w aplikacji głównej) z pracą funkcji multipleksującej wyświetlacz LED, tak by nie występowało zjawisko „mieszania" zawartości zmiennych w kolejnych „przebiegach" funkcji multipleksującej. Aktualizacja, o której mowa powyżej, następuje po pełnym cyklu multipleksu dla całego wyświetlacza LED. To wszystko, jeśli chodzi o obsługę wyświetlacza – przejdźmy zatem do grupy funkcji odpowiedzialnych za mechanizm pomiarowy. Tradycyjnie, na początek plik nagłówkowy, dzięki któremu porządkujemy późniejszy kod źródłowy czyniąc go bardzo czytelnym, a jednocześnie upraszczając proces ewentualnego wprowadzania zmian. Plik ten, pokazany na **listingu 6**, definiuje główne ustawienia sprzętowe, a także wprowadza niezbędne zmienne. Dalej, na **listingu 7**, znalazło się ciało funkcji odpowiedzialnej za inicjalizację mechanizmu pomiarowego. Kolejne funkcje, których ciała widoczne są na **listingach 8** i **9**, uruchamiają i zatrzymują ów mechanizm.

I na sam koniec dwie funkcje obsługi przerwań: od przechwycenia zawartości licznika (Timer1) oraz od przepełnienia licznika (Timer1), które realizują właściwy mechanizm pomiarowy. Ciała tychże funkcji pokazano na **listingach 10** i **11**.

Powyższa informacja zamyka kwestie implementacyjne. Przejdźmy zatem do schematu montażowego naszego urządzenia, widocznego na **rysunkach 5a** i **5b**.

```
void initMeasurement(void)
{
    //Wybór dolnego zakresu pomiarowego: 100 nF
    //(de facto wielkości rezystora szeregowego)
    MEAS_SELECT_LOW_CAP;
    //Port wyboru zakresu pomiarowego, jako wyjściowy
    MEAS_SELECT_CAP_AS_OUTPUT;
    //Wybór napięcia odniesienia dla komparatora analogowego: 0.17 Vcc
    MEAS_SELECT_LOW_REF;
    //Rozładowanie kondensatora
    MEAS_DISCHARGE_CAP;
    //Podłączenie wyjścia komparatora analogowego do wejścia ICP (Capture) Timera1
    ANALOG_COM_INPUT_CAP_ENABLE;
    //Domyślna wartość flagi pomiarowej
    measFlag = FLAG_IDLE;
}
```

Listing 7. Funkcja odpowiedzialna za inicjalizację mechanizmu pomiarowego

```
inline void startMeasurement(uint8_t capType)
{
    //Ustawienie domyślnych wartości zmiennych procesu pomiarowego
    Overflows = 0;
    measFlag = FLAG_IN_PROGRESS;
    //Wybór zakresu pomiarowego: 100 nF lub 1000 uF
    //(de facto wielkości rezystora szeregowego)
    if(capType == LOW_CAP) MEAS_SELECT_LOW_CAP; else MEAS_SELECT_HIGH_CAP;
    //Wybór napięcia odniesienia dla komparatora analogowego: 0.17 Vcc
    //(plus krótki delay na ustabilizowanie się napięcia)
    MEAS_SELECT_LOW_REF;
    _delay_us(5);
    //Wyzerowanie licznika Timer1
    MEAS_CLEAR_COUNTER;
    //Skasowanie flag przerwań licznika Timer1
    MEAS_CLEAR_ISRS_FLAGS;
    //Uruchomienie przerwań licznika Timer1 (OVF i ICP)
    MEAS_ENABLE_ISRS;
    //Start licznika Timer1 (@8 MHz)
    MEAS_START_COUNTER;
    //Rozpoczęcie ładowania kondensatora
    MEAS_CHARGE_CAP;
}
```

Listing 8. Funkcja odpowiedzialna za uruchomienie mechanizmu pomiarowego

```
void stopMeasurement(void)
{
    //Zatrzymanie licznika Timer1
    MEAS_STOP_COUNTER;
    //Wyłączenie przerwań licznika Timer1 (OVF i ICP)
    MEAS_DISABLE_ISRS;
    //Rozpoczęcie rozładowania kondensatora
    MEAS_DISCHARGE_CAP;
}
```

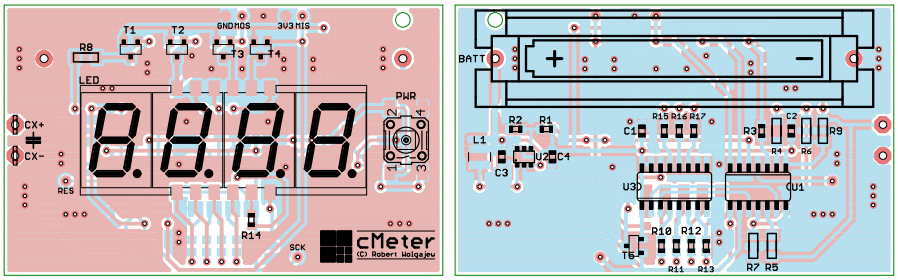
Listing 9. Funkcja odpowiedzialna za zatrzymanie mechanizmu pomiarowego

```
//Funkcja odpowiedzialna za przechwycenie zawartości licznika Timer1 po naładowaniu kondensatora mierzonego do wartości 0.17 lub 0.5 VCC

ISR(TIM1_CAPT_vect)
{
    static uint32_t T1, T2;  //Liczba zliczonych impulsów dla obu progów napięcia odniesienia: 0.17 i 0.5 Vcc

    //Nastąpiło przechwycenie zawartości licznika Timer1, więc sprawdzamy dla jakiego poziomu napięcia: 0.17 lub 0.5 Vcc
    if(MEAS_LOW_REF_IS_SELECTED)
    {
        //Przechwycenie nastąpiło dla niższego napięcia Vref (0.17 Vcc) więc zapisujemy liczbę zliczonych impulsów
        T1 = ICR1 + (Overflows * 65536);
        //Zmieniamy Vref na 0.5 Vcc (plus krótki delay na ustabilizowanie się napięcia)
        MEAS_SELECT_HIGH_REF;
        _delay_us(5);
        //Skasowanie ewentualnych flag przerwań Timera1
        MEAS_CLEAR_ISRS_FLAGS;
    }
    else
    {
        //Zatrzymujemy pomiar co powoduje zatrzymanie zegara Timer1, wyłączenie przerwań zegara oraz rozładowanie kondensatora
        stopMeasurement();
        //Przechwycenie nastąpiło dla wyższego napięcia Vref (0.5 Vcc) więc zapisujemy liczbę zliczonych impulsów
        T2 = ICR1 + (Overflows * 65536);
        //Obliczenie liczby zarejestrowanych imlulsów licznika Timer1
        Counts = T2-T1;

        //Sprawdzenie, czy zarejestrowana liczba impulsów przekracza zakres dla 99.9 nF i jeśli tak to startujemy proces pomiarowy
        //od nowa ze mniejszym rezystorem pomiarowym
        if(T2 > FIRST_STAGE_CAP_PULSES)
        {
            //Maksymalny, dodatkowy czas na rozładowanie naładowanego w poprzednim kroku kondensatora (rozładowanie do 0.45 V,
            //czyli poniżej 0.17 VCC)
            _delay_us(15);
            //Wznowienie procesu pomiarowego dla nowego zakresu mierzonych pojemności
            startMeasurement(HIGH_CAP);
        }
        //Ustawiamy odpowiednią flagę w zależności od bieżącego zakresu pomiarowego (wielkości mierzonego kondensatora)
        else measFlag = MEAS_LOW_CAP_IS_SELECTED? FLAG_RDY_LOW : FLAG_RDY_HIGH;
    }
}
```

Listing 10. Funkcja obsługi przerwania od przechwycenia zawartości licznika Timer1 realizująca mechanizm pomiarowy

```
//Funkcja odpowiedzialna za policzenie liczby przepełnień licznika Timer1 podczas pomiaru pojemności kondensatora mierzonego

ISR(TIM1_OVF_vect)
{
    ++Overflows;

    if(MEAS_LOW_CAP_IS_SELECTED)
    {
        //Przekroczono dolny zakres pomiarowy (99.9 nF), więc wybieramy mniejszy rezystor ładowania kondensatora
        //i rozpoczynamy proces od nowa
        if(Overflows > LOW_CAP_MAX_OVFS)
        {
            //Zatrzymanie procesu pomiarowego (w tym rozpoczęcie rozładowania kondensatora)
            stopMeasurement();
            //Maksymalny, dodatkowy czas na rozładowanie naładowanego w poprzednim kroku kondensatora (rozładowanie do 0.45 V,
            //czyli poniżej 0.17 VCC)
            _delay_us(15);
            //Wznowienie procesu pomiarowego dla nowego zakresu mierzonych pojemności
            startMeasurement(HIGH_CAP);
        }
    }
    else
    {
        //Przekroczono zakres pomiarowy urządzenia, więc wyświetlamy stosowny komunikat ustawiając bity zmiennej measFlag
        if(Overflows > HIGH_CAP_MAX_OVFS)
        {
            //Zatrzymanie procesu pomiarowego (w tym rozpoczęcie rozładowania kondensatora)
            stopMeasurement();
            //Ustawienie flagi Overflow
            measFlag = FLAG_OVF;
        }
    }
}
```

Listing 11. Funkcja obsługi przerwania od przepełnienia licznika Timer1 realizująca mechanizm pomiarowy



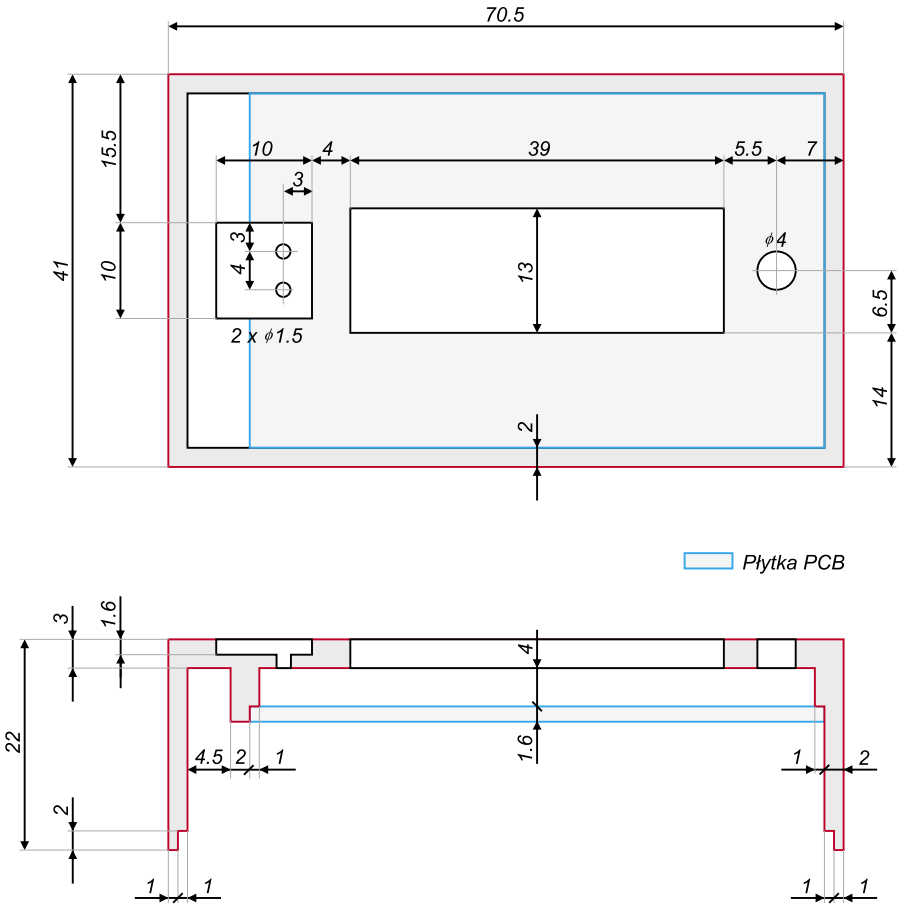
Rysunek 5. Schemat montażowy urządzenia cMeter (a – strona TOP, b – strona BOTTOM)

Jak widać, zaprojektowano bardzo zgrabną, dwustronną, niewielką płytkę drukowaną ze zdecydowaną przewagą elementów SMD lutowanych po obu stronach laminatu. Montaż urządzenia rozpoczynamy od warstwy TOP, na której w pierwszej kolejności przylutowujemy wszystkie półprzewodniki, w tym wyświetlacz LED. Proces ten najłatwiej wykonać przy użyciu stacji lutowniczej na gorące powietrze (tzw. Hot-Air) i odpowiednich stopów lutowniczych. Jeśli jednak nie dysponujemy tego rodzaju sprzętem, można również zastosować metodę z użyciem typowej stacji lutowniczej. Najprostszym sposobem montażu elementów o tak dużym zagęszczeniu wyprowadzeń, niewymagającym jednocześnie posiadania specjalistycznego sprzętu, jest zastosowanie zwykłej stacji lutowniczej, dobrej jakości cyny z odpowiednią ilością topnika oraz dość cienkiej plecionki rozlutowniczej, która umożliwi usunięcie nadmiaru cyny spomiędzy wyprowadzeń układów. Należy przy tym uważać, by nie uszkodzić termicznie tego rodzaju elementów. Następnie lutujemy elementy bierne, po czym przechodzimy na warstwę BOTTOM. Tutaj, podobnie jak poprzednio, montaż rozpoczynamy od przylutowania wszystkich półprzewodników (w tym układów scalonych), po czym montujemy pozostałe elementy bierne oraz gniazdo baterii AAA. W tym momencie wracamy na warstwę TOP, gdzie przylutowujemy przycisk PWR. Na tym etapie urządzenie gotowe jest do uruchomienia. Na **fotografii 1** pokazano zmontowane urządzenie cMeter od strony warstwy TOP, zaś na **fotografii 2** – to samo urządzenie od strony warstwy BOTTOM.

Przejdźmy zatem do opisu obsługi naszego urządzenia. Tu sprawa jest niezmiernie prosta: podczas pierwszego uruchomienia urządzenia, czyli po włożeniu baterii zasilającej do koszyczka, następuje włączenie urządzenia oraz kalibracja wartości zerowej pojemności, co sprowadza się w tym przypadku do pomiaru pojemności pasożytniczej, powinno zatem zostać wykonane przy niepodłączonych do jakiegokolwiek kondensatora stykach pomiarowych. Uzyskana wartość zapisywana jest w pamięci RAM i używana podczas późniejszego procesu pomiarowego. W tym momencie miernik gotowy jest do pracy. Pomiary dokonywane są maksymalnie 2 razy na sekundę, zaś wartość mierzona wyświetlana jest łącznie z jej jednostką (cyfra pierwsza od prawej) na wyświetlaczu LED, przy czym zakres pomiarowy wraz z symbolem wyświetlanej jednostki (p, n, u) jest wybierany automatycznie przez urządzenie. W przypadku kondensatorów o pojemności powyżej 999 µF wyświetlony zostanie napis **high**. Co ważne, przewidziano również możliwość kalibracji naszego miernika, by wyświetlane wartości jak najbardziej odzwierciedlały rzeczywistą wartość mierzonej

Fotografia 1. Zmontowane urządzenie cMeter od strony warstwy TOP

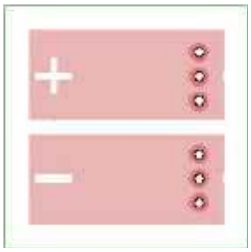
Fotografia 2. Zmontowane urządzenie cMeter od strony warstwy BOTTOM



Rysunek 6. Widok 2D obudowy urządzenia cMeter z zaznaczeniem kluczowych wymiarów

pojemności. W tym celu do styków pomiarowych urządzenia należy dołączyć kondensator wzorcowy o pojemności 47 nF, po czym nacisnąć i przytrzymać przycisk PWR. Miernik wyświetli wówczas napis **CAL** i dokona pomiaru wartości kondensatora wzorcowego – zakładając, że ma on pojemność o dokładnej wartości 47 nF, po czym obliczy stosowny współczynnik korekcyjny i zapisze go w pamięci RAM oraz EEPROM. Dokona tego jednak tylko wtedy, gdy mierzona wartość pojemności znajdzie się w zakresie ±10% wartości 47 nF. Obliczony współczynnik korekcyjny brany będzie następnie pod uwagę przy obliczaniu wartości pojemności mierzonych kondensatorów. Zakończenie pomiaru kondensatora wzorcowego zostanie następnie potwierdzone wyświetleniem napisu **donE**. Z kolei w celu wyłączania i ponownego włączenia urządzenia należy nacisnąć krótko przycisk **PWR** – wyświetli się wówczas napis powitalny **robi**. Warto również podkreślić, że cMeter wyposażono w mechanizm automatycznego wyłączenia (ze względu na potrzebę oszczędzania energii), który aktywuje się po upływie 30 sekund od włączenia urządzenia. Dla porządku dodam, iż urządzenie cały czas monitoruje wartość napięcia baterii zasilającej i przy jego nieodpowiedniej wielkości wyświetla napis **bAtt**, po czym przechodzi do stanu uśpienia.

Omówiliśmy zagadnienia związane z obsługą, przejdźmy zatem do nie mniej ciekawych kwestii dotyczących obudowy urządzenia. Przystępując do prac projektowych, przygotowałem stosowny projekt 2D, którego rysunek z zaznaczeniem wszystkich niezbędnych wymiarów pokazano na **rysunku 6**.



Rysunek 7. Wygląd obwodu drukowanego (od strony warstwy TOP) płytki styków pomiarowych urządzenia cMeter

Rysunek 8. Widok 3D obudowy urządzenia cMeter

Jak widać, w płycie czołowej obudowy (jej ściance górnej) o grubości 3 mm przewidziano „specjalne" wyżłobienie, w które należy wkleić niewielką płytkę drukowaną o wymiarach 10×10 mm, integrującą w sobie styki pomiarowe (pola miedzi). Do płytki tej, od spodu, należy przylutować przewody, które przeciągnąć trzeba do środka obudowy i podłączyć do styków Cx± na płytce drukowanej urządzenia cMeter, zachowując odpowiednią polaryzację. Na **rysunku 7** pokazano wygląd obwodu drukowanego (od strony warstwy TOP) płytki styków pomiarowych, o której mowa powyżej.

Przeanalizujmy zatem konkrety dotyczące projektu 3D obudowy. W tym zakresie, jak to ostatnio u mnie bywa, poprosiłem mojego kolegę **Bartłomieja Wawrzyszko**, zajmującego się hobbystycznie projektami tego rodzaju, o przygotowanie stosownej obudowy, zgodnej z moim projektem 2D, w środowisku pozwalającym na wydruk na drukarce 3D. W efekcie powstał projekt obudowy, którego widok 3D pokazano na **rysunku 8**.

Wspomniana obudowa składa się tak naprawdę z dwóch elementów:

- części górnej (pokazanej na rysunku 8), w której umieszczono otwory na elementy interfejsu użytkownika (okienko wyświetlacza LED i otwór na przycisk PWR) oraz wyżłobienie na płytkę styków pomiarowych
- części dolnej pełniącej funkcję klapki, za pomocą której zamykamy obudowę od dołu (unieruchamiając ją dodatkowo stosownym wkrętem blokującym).

W części górnej naszej obudowy, we wspomnianym powyżej okienku na wyświetlacz LED, należy umieścić przydymioną pleksi o wymiarach 13×39 mm i grubości 3 mm, stanowiącą ochronę wyświetlacza i zapewniającą atrakcyjność wizualną. Dociekliwym Czytelnikom podpowiem, że kupiłem taki kawałek pleksi na znanym portalu aukcyjnym (łącznie z usługą jej wycięcia) za przysłowiowe 3 zł. Wracając do tematu obudowy warto podkreślić, że stosowne pliki z jej projektem (do wydrukowania na drukarce 3D) zostaną zamieszczone na serwerze http://ep.com.pl. Dodatkowo podpowiem, że jeśli nie dysponujecie odpowiednim urządzeniem umożliwiającym wydrukowanie obudowy według załączonych plików, to z powodzeniem możecie to zlecić (i to na naprawdę atrakcyjnych warunkach) jednej z chińskich firm, która znana jest z produkcji obwodów drukowanych w bardzo przystępnych cenach. Na **fotografii 3** pokazano wygląd wspomnianej obudowy wykonanej na drukarce 3D pracującej w technologii FDM (*Fused Deposition Modeling*), w której półpłynne tworzywo sztuczne spaja się pod wpływem wysokiej temperatury i szybko zastyga, tworząc (niemalże) jednolitą strukturę. Niemalże,

Fotografia 3. Widok obudowy urządzenia cMeter wydrukowanej w technologii FDM

Fotografia 4. Widok dolnej części obudowy (klapki) urządzenia cMeter wydrukowanej w technologii SLA

gdyż (co widać na fotografii 3) obudowa wydrukowana w ten sposób ma pewną, nie zawsze akceptowalną, strukturę (poniekąd zależną od jakości samej drukarki).

W celu zobrazowania kontrastu, na zdjęciu tytułowym pokazano wygląd obudowy urządzenia cMeter wydrukowanej na drukarce 3D pracującej w technologii MJF (*Multi Jet Fusion*), polegającej na druku 3D ze sproszkowanych tworzyw sztucznych (poliamidów), poprzez selektywne natryskiwanie na nie lepiszcza (sklejającego ze sobą poszczególne warstwy modelu) i zgrzewania ich w wysokiej temperaturze, co skutkuje ich trwałym zespojeniem. Atutem wydruków 3D z zastosowaniem technologii MJF jest wysoka wytrzymałość mechaniczna produkowanych części. Uzyskiwana jest ona dzięki jednolitej strukturze, która ma 100-procentowe wypełnienie. Takie drukowanie 3D sprawia, że można uzyskać dowolne części, nawet o wysokim stopniu skomplikowania elementów, a koszt wydruku uzależniony jest wyłącznie od ilości zużytego materiału. Należy przy tym podkreślić, że powstające produkty mają wysoką powtarzalność, z dokładnością wymiarową do 0,2 mm. Metoda MJF w drukowaniu 3D pozwala na wytwarzanie funkcjonalnych części, o gładkiej powierzchni, niskiej porowatości i dowolnej geometrii. Wspomniana powyżej obudowa została wydrukowana dzięki usłudze jednej z dalekowschodnich firm produkujących obwody drukowane, o czym wspomniałem już wcześniej. Szczerze polecam tego typu rozwiązanie. Na **fotografii 4** pokazano wygląd dolnej części obudowy (klapki) wydrukowanej w SLA, czyli technologii, która bazuje na procesie fotopolimeryzacji (podobnie jak technologia DLP – ang. *Direct Light Processing* – z którą ma wiele wspólnych cech). Budowany obiekt powstaje w tym przypadku wskutek selektywnego utwardzania żywicy fotopolimerowej światłem lasera. Metoda ta (nazywana pierwotnie stereolitografią) jest najstarszą metodą druku 3D, uznawaną przez wielu za jedną z najbardziej ekonomicznych metod przyrostowych. Drukowane elementy łączą w sobie wysoką dokładność z dużą gładkością powierzchni. Niestety są mało odporne na wysokie temperatury i dość często już przy temperaturach rzędu 50°C stają się ciągliwe.

Robert Wołgajew, EP

W ofercie AVT*
AVT6041

Najważniejsze parametry:
- konwersja nieujemnego napięcia stałego na wyjście pętli prądowej 4...20 mA
- zasilanie napięciem stałym nie wyższym niż 32 V, typ. 24 V,
- regulowana potencjometrem czułość na wejściu: możliwość pracy z napięciami wejściowymi 0...5 V, 0...10 V i wyższymi,
- pobór prądu do 30 mA (przy zasilaniu 24 V),
- rezystancja wejściowa 500 kΩ,
- regulowane potencjometrem minimalne natężenie wypływającego prądu (typowo 4 mA).

* **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.

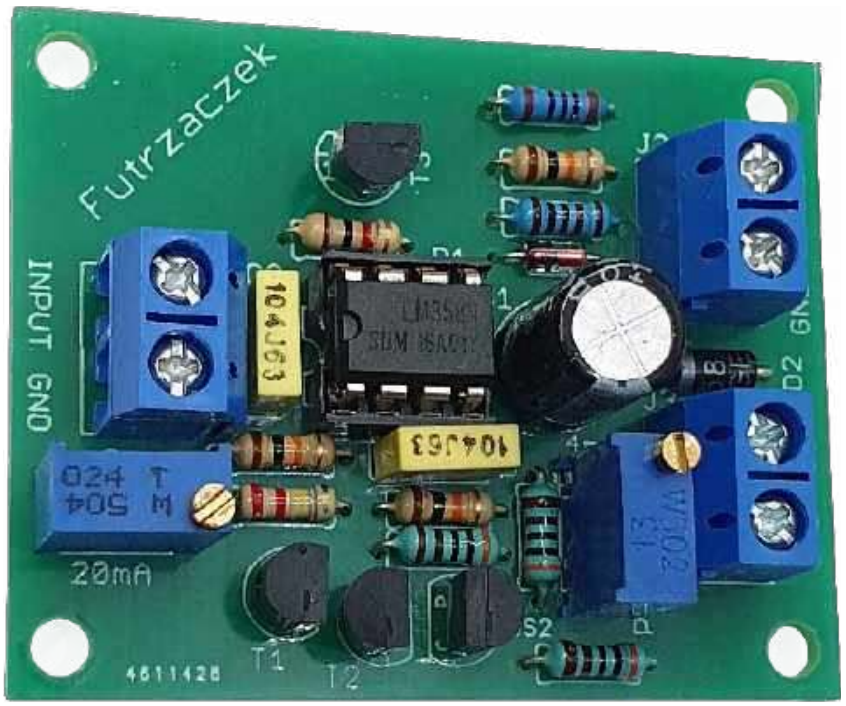
Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- **wersja [A+]** – płytka drukowana **[A]** + zaprogramowany układ **[UK]** i dokumentacja,
- **wersja [UK]** – zaprogramowany układ.

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**.

Konwerter napięcia stałego na pętlę prądową 4...20 mA

Pętla prądowa 4...20 mA to przemysłowy standard transmisji analogowej, który ma wiele zalet. Odporność na zakłócenia, niewrażliwość na długość połączeń oraz łatwa detekcja uszkodzeń to tylko niektóre z nich. Profesjonalne, certyfikowane moduły do konwersji napięcia stałego na pętlę prądową potrafią kosztować krocie. Czy da się problem rozwiązać prościej i – co najważniejsze – taniej? Jasne!



Prosta sytuacja z życia wzięta: sterownik przemysłowy ma wejścia analogowe w postaci pętli prądowych 4...20 mA. Można przyjąć taki parametr za standard. Niestety, oryginalny czujnik temperatury, który współpracował z tym sterownikiem, uległ zniszczeniu i trzeba zastąpić go innym. Tak się składa, że model ten został już z rynku wycofany – są dostępne zamienniki od innych producentów, ale ich cena szokuje. Znacznie prościej jest znaleźć czujnik temperatury z wyjściem napięciowym 0...5 V. Trzeba tylko pogodzić wyjście czujnika z wejściem sterownika.

Tym właśnie może zająć się prezentowany układ. Owszem, można też z łatwością kupić gotowe moduły renomowanych producentów, z powodzeniem realizujące tę funkcję. Jednak ich cena okazuje się wysoka. Jeżeli więc konwertowana wielkość nie należy do krytycznych, można posłużyć się tym właśnie układem – prostym i znacznie tańszym.

Budowa

Schemat ideowy omawianego układu znajduje się na **rysunku 1**. Wejściowe napięcie stałe podaje się na zaciski złącza J1. Przy użyciu potencjometru P1 jest ono dzielone w takim stopniu, aby jego wartość zawierała

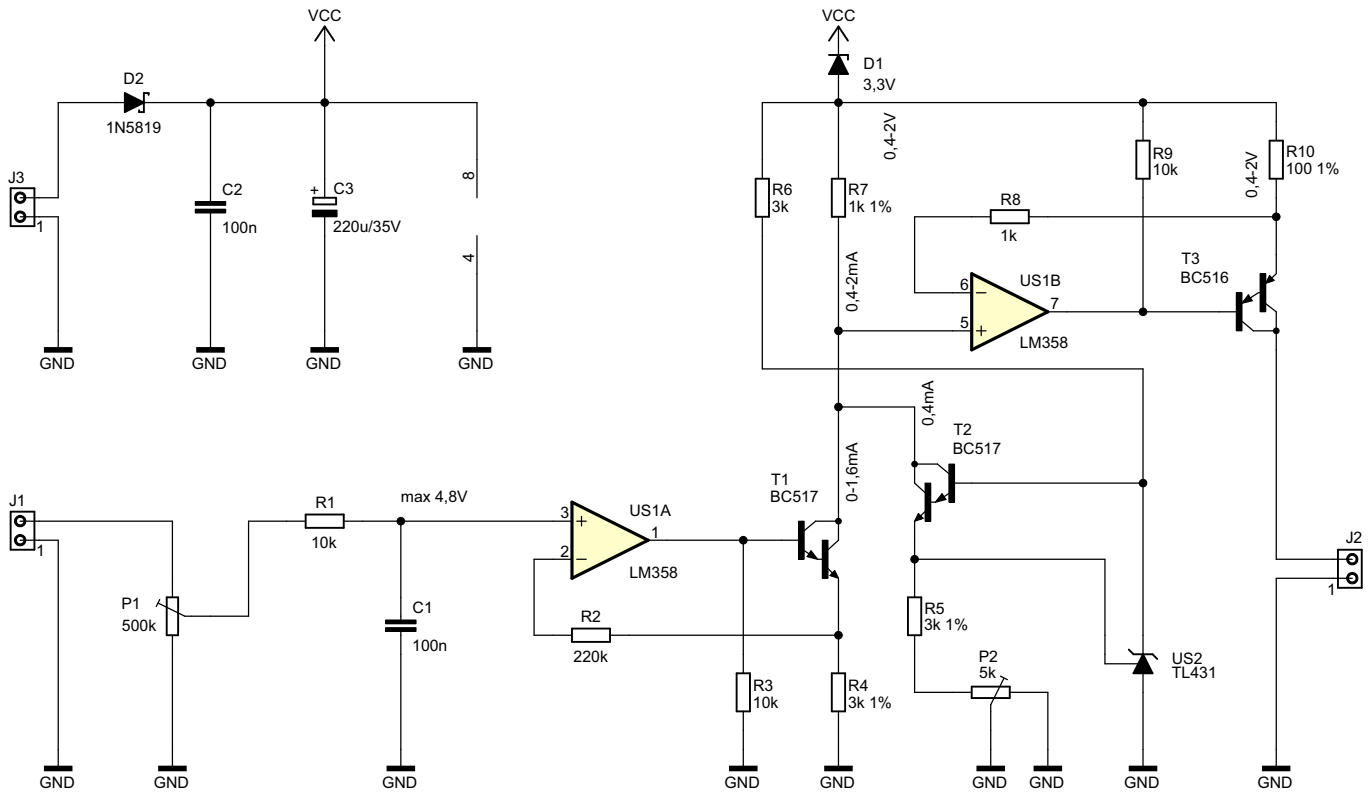
Wykaz elementów:

Rezystory: (THT o mocy 0,25 W)
R1, R3, R9: 10 kΩ
R2: 220 kΩ
R4, R5: 3 kΩ 1%
R6: 3 kΩ
R7: 1 kΩ 1%
R8: 1 kΩ
R10: 100 Ω 1%
P1: 500 kΩ montażowy pionowy 3296W
P2: 5 kΩ montażowy pionowy 3296W

Kondensatory:
C1, C2: 100 nF raster 5 mm MKT
C3: 220 µF 35 V raster 3,5 mm

Półprzewodniki:
D1: dioda Zenera 3,3 V 0,5 W
D2: 1N5819
T1, T2: BC517
T3: BC516
US1: LM358 (DIP8)
US2: TL431CLP (TO92)

Inne:
J1...J3: złącze śrubowe ARK2/500
Jedna podstawka DIP8



Rysunek 1. Schemat ideowy układu konwertera napięcia stałego na pętlę prądową

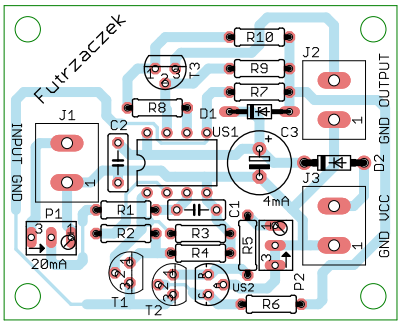
się w przedziale 0...4,8 V. W ten sposób układ może współpracować ze źródłem napięcia 5 V lub wyższym. Musi być to napięcie nieujemne, wartości niższe od zera nie będą prawidłowo obsłużone.

Odpowiednio podzielone napięcie jest poddawane filtracji dolnoprzepustowej z użyciem kondensatora C1 oraz wypadkowej rezystancji połączenia rezystora R1 z rezystancją wyjściową potencjometru P1. Jak nietrudno obliczyć, rezystancja współpracująca z ową pojemnością może wynosić maksymalnie 260 kΩ – będzie to wynik uzyskany przy ustawieniu potencjometru P1 w połowie dostępnego zakresu regulacji. Ta filtracja zmniejsza wpływ zakłóceń na konwertowany sygnał oraz obniża wartość skuteczną szumów.

Następnym blokiem jest precyzyjne źródło prądowe, zrealizowane z użyciem wzmacniacza operacyjnego US1A. Wejście układu LM358 obsługuje potencjał już od 0 V (a nawet nieco poniżej), więc niepotrzebne było dodawanie zasilacza napięcia ujemnego. Jego wyjście tak steruje bazą tranzystora T1, by uzyskać spadek napięcia na rezystorze R4 równy podzielonemu napięciu wejściowemu. Przy zastosowanych wartościach elementów prąd płynący przez R4 będzie zawierał się w przedziale 0...1,6 mA. Ponieważ tranzystor T1 ma bardzo wysokie wzmocnienie prądowe (jest typu Darlingtona), prąd jego kolektora jest równy prądowi emitera z naprawdę niewielkim błędem. Rezystor R3 obciąża wyjście wzmacniacza operacyjnego, linearyzując pracę jego stopnia wyjściowego oraz ułatwiając zatkanie tranzystora T1. Z kolei rezystor R2 stanowi kompensację prądu płynącego przez wejście odwracające wzmacniacza US1A. Jego rezystancja powinna być równa tej, która steruje wejściem nieodwracającym, lecz ona może się zmienić w szerokich granicach, dlatego postawiono pod tym względem na pewien kompromis.

Czemu w roli T1 nie został użyty tranzystor MOSFET z kanałem typu N? Prąd jego bramki jest niemal zerowy, zatem prąd rezystora R4 w rzeczywistości byłby równy prądowi jego drenu. Jednak własne doświadczenia wskazały, że tranzystory polowe potrafią się wzbudzać w układzie precyzyjnego źródła prądowego, nawet po zastosowaniu zewnętrznych elementów realizujących kompensację częstotliwościową. Bywa tak, że dobranie elementów do układu powoduje, że w wykonanej większej partii znajdzie się kilka sztuk, które będą miały skłonność do wzbudzania się. Tranzystory bipolarne nie przejawiają tego typu zachowań, a wpływ prądu bazy jest naprawdę pomijalny, ponieważ wzmocnienie prądowe takiego tranzystora wynosi wiele tysięcy. Większy błąd wprowadzają w układzie inne czynniki, jak tolerancja rezystorów i offset wzmacniaczy operacyjnych.

Oprócz tranzystora T1, do tego samego węzła z rezystorem R7 podłączony jest także kolektor tranzystora T2, „wciągający" dodatkowy prąd o stałej wartości. Natężenie tego prądu powinno wynosić 0,4 mA, a za jego ustalenie odpowiada układ US2, który tak steruje potencjałem bazy T2, aby na szeregowym połączeniu rezystora R5 i potencjometru P2

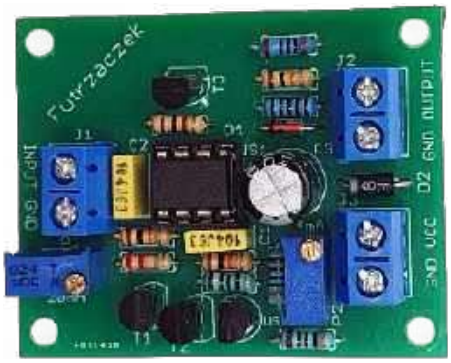


Rysunek 2. Schemat montażowy i wzór ścieżek płytki

odkładało się napięcie równe 2,5 V – tyle, ile wynosi napięcie referencyjne układu TL431.

Prądy kolektorów obydwu tranzystorów wywołują spadek napięcia na rezystorze R7. Mając na uwadze ich sumę (0...1,6 mA z kolektora T1 i 0,4 mA z kolektora T2), jesteśmy w stanie prześledzić, że przez R1 może płynąć prąd w przedziale 0,4...2 mA. Rezystancja R7 wynosi 1 kΩ, więc spadek napięcia na tym elemencie wyniesie 0,4...2 V. Tym napięciem sterowane jest drugie precyzyjne źródło prądowe, zrealizowane z użyciem wzmacniacza operacyjnego US1B i tranzystora T3. Na rezystorze R10 (100 Ω) musi odkładać się napięcie równe temu, które występuje na zaciskach R7, więc prąd przezeń płynący będzie w przedziale 4...20 mA. Prąd o takim właśnie natężeniu będzie wypływał z kolektora tranzystora T3 wprost do zacisku złącza J2, będącego wyjściem pętli prądowej.

Rezystor R8 stanowi podobną kompensację dla US1B, jak rezystor R2 dla US1A – z tą różnicą, że rezystancja „widziana" przez wejście nieodwracające układu US1B jest bardzo dobrze określona i wynosi 1 kΩ. R9 stanowi obciążenie wyjścia US1B. Na omówienie zasługuje również dioda Zenera D1, zasilająca tę część układu napięciem niższym o około 3,3 V od zasilającego. W ten sposób układ US1, zasilany wprost z zasilacza, może prawidłowo działać. Dotyczy to w szczególności US1B, który operuje na potencjałach bliskich napięciu zasilającemu, a który musi mieć zachowany margines od dodatniego potencjału zasilania nie mniejszy niż 2 V. Użycie diody D1 daje tę gwarancję i to z pewnym naddatkiem. Aby jednak przez D1 zawsze płynął prąd o natężeniu minimum 5 mA, wymagany do jej prawidłowej pracy, katoda układu US2 jest polaryzowana właśnie za pośrednictwem tej diody. Prąd katody US2 jest ograniczany przez rezystor R6.



Fotografia 1. Widok zmontowanego układu

Montaż i uruchomienie

Układ został zmontowany na jednostronnej płytce drukowanej o wymiarach 50 mm × 40 mm. Jej wzór ścieżek oraz schemat montażowy pokazuje **rysunek 2**. W odległości 3 mm od krawędzi płytki znalazły się cztery otwory montażowe, każdy o średnicy 3,2 mm.

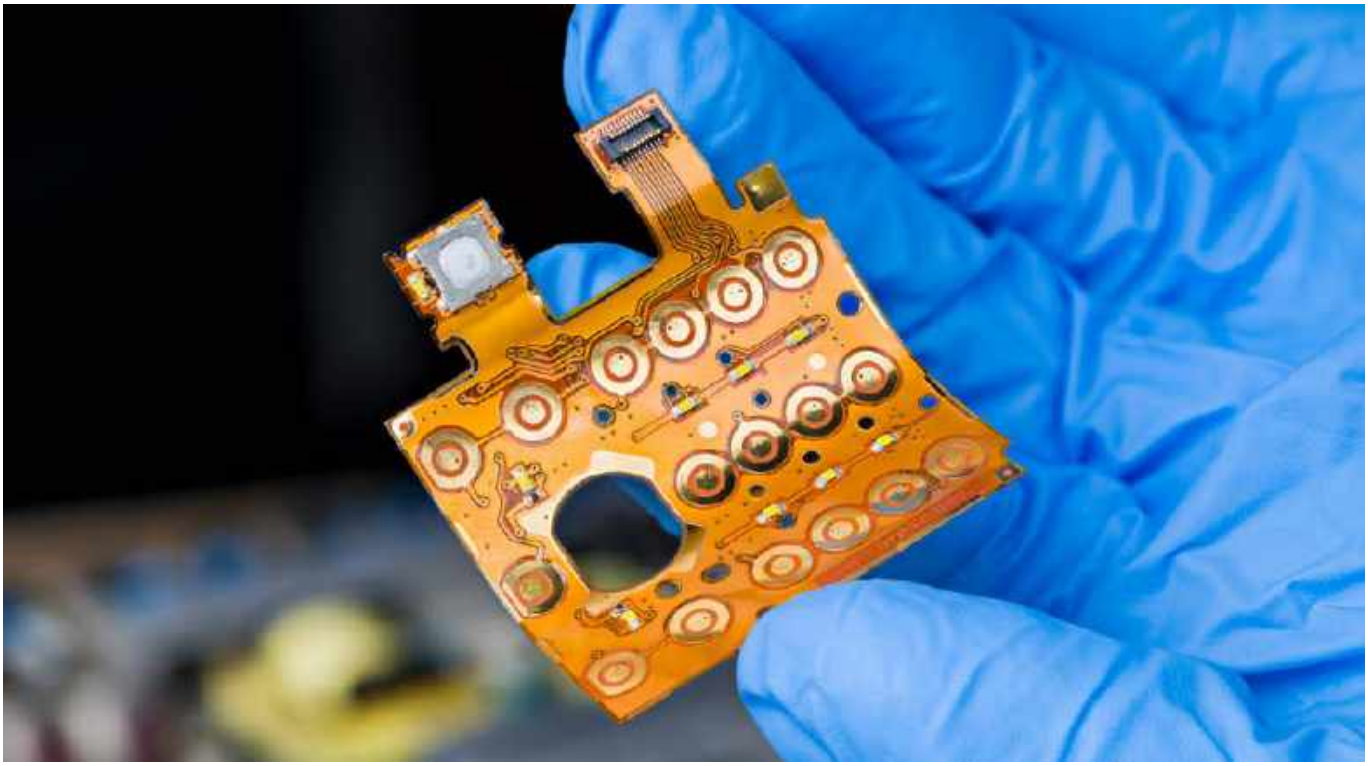
Montaż proponuję przeprowadzić w jak najbardziej typowy sposób, czyli zaczynając od elementów o najniższych obudowach – rezystorów i diod. Pod układ scalony US1 warto zastosować podstawkę, aby ułatwić jego wymianę w razie ewentualnego uszkodzenia. Gotowy układ można zobaczyć na **fotografii 1**.

Poprawnie zmontowane urządzenie jest gotowe do działania po podaniu zasilania na zaciski GND i VCC złącza J3. Do zasilania powinno służyć napięcie stałe o wartości 24 V, nie wyższe niż 32 V z uwagi na wytrzymałość podzespołów. Dolne ograniczenie wynika z konieczności zapewnienia na wyjściu pętli prądowej napięcia o dostatecznie wysokiej wartości, aby rezystancja nawet bardzo długich przewodów połączeniowych nie była w stanie zmusić tranzystora T3 do wejścia w stan nasycenia. Pobór prądu przy 24 V jest wyższy o około 10 mA od aktualnego prądu wyjściowego, zatem nie przekracza 30 mA.

Do poprawnej pracy układ wymaga dwóch regulacji. Pierwszej z nich dokonuje się przy braku podłączonego sygnału wejściowego, zaciski złącza J1 powinny być wolne. Potencjometrem P2 należy ustawić prąd wyjściowy równy 4 mA. Następnie należy na wejście układu (złącze J1) podać napięcie odpowiadające 100% dostępnej skali – czyli 5 V w standzie 0...5 V, 10 V w standzie 0...10 V lub inne, jednak nie więcej niż 250 V z uwagi na możliwość wystąpienia przebicia w potencjometrze P1 lub izolacji złącza J1 (przy czym regulację należy zacząć od najniższego ustawienia suwaka P1, tj. od pozycji odpowiadającej zwarciu go z masą). Potencjometrem P1 należy wtedy ustawić prąd wyjściowy równy 20 mA. Im dokładniej zostanie przeprowadzona regulacja, tym wierniej napięcie wejściowe będzie konwertowane na prąd.

Układ nie był optymalizowany pod kątem szybkości działania. Założono, że konwertowane sygnały będą z natury wolnozmienne jak z czujników temperatury, poziomu czy masy. Główne ograniczenie pasma układu stanowi wejściowy filtr dolnoprzepustowy, zawężający pasmo przenoszenia do około 160 Hz lub mniej, zależnie od położenia ślizgacza potencjometru P1. W ten sposób ograniczone jest również pasmo szumowe oraz wartość szczytowa ewentualnych zakłóceń, które znalazłyby się w wejściowym sygnale napięciowym.

Michał Kurzela, EP



Elastyczne i sztywno-giętkie obwody drukowane w ofercie PCBWay

Obwody elastyczne oraz sztywno-giętkie (ang. rigid-flex) są nieodłącznym elementem współczesnej technologii elektronicznej. W niektórych zastosowaniach użycie tego rodzaju PCB jest podyktowane koniecznością upakowania dużej liczby modułów i złączy rozproszonych w wyjątkowo ciasnej przestrzeni (np. w obudowie smartfona), w innych pierwszorzędne znaczenie ma natomiast możliwość pracy obwodów elastycznych w warunkach powtarzalnego zginania. Niezależnie od motywacji stojącej za wyborem technologii flex/rigid-flex, jedno jest pewne – wyprodukowanie niezawodnych płytek na podłożach polimerowych jest nie lada wyzwaniem technologicznym. Dla doświadczonego zespołu PCBWay takie zlecenia są jednak chlebem powszednim.

Elastyczne obwody drukowane spotykamy obecnie praktycznie we wszystkich gałęziach współczesnej elektroniki. Płytki drukowane typu flex oraz rigid-flex są bowiem „elastyczne" nie tylko w znaczeniu mechanicznym, ale także aplikacyjnym – pozwalają na użytkowanie niemal 100% przestrzeni dostępnej wewnątrz najbardziej nawet kompaktowych urządzeń, stąd tak często można je znaleźć m.in. w aparatach cyfrowych (zarówno w body lustrzanek i aparatów bezlusterkowych, jak i w obiektywach), urządzeniach mobilnych czy ubieralnych, aktywnych implantach medycznych oraz setkach

Fotografia 1. Elastyczny obwód jednowarstwowy

Rysunek 1. PCBWay oferuje produkcję płytek sztywno-giętkich z usztywnieniami na bazie FR4, poliimidu, a nawet stali nierdzewnej lub aluminium

innych produktów. Obwody elastyczne są także niezastąpione w zastosowaniach, które wymagają połączenia dwóch (lub więcej) modułów przemieszczających się względem siebie podczas pracy urządzenia – przykładem mogą być głowice drukarek atramentowych, skanery dokumentów, napędy optyczne komputerów i wiele innych.

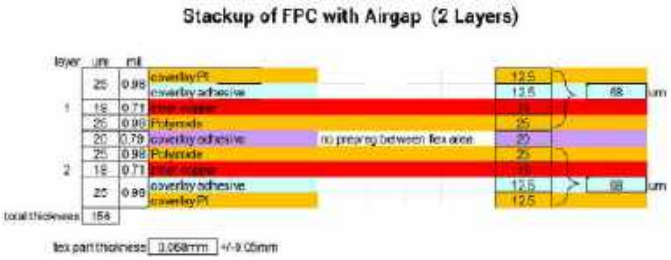
Do niedawna użycie obwodu sztywno-giętkiego lub w pełni elastycznego wiązało się z koniecznością poniesienia przez producenta urządzenia sporych kosztów wdrożeniowych – taka implementacja była więc w praktyce opłacalna tylko przy produkcji wielkoskalowej. Sytuacja uległa diametralnej zmianie w momencie wprowadzenia tego rodzaju płytek drukowanych do oferty PCBWay – czołowego producenta PCB oraz dostawcy kompleksowych usług montażowych.

Zalety obwodów drukowanych typu flex i rigid-flex

Elastyczność obwodów z omawianej grupy umożliwia niemal dowolne układanie połączeń oraz rozmieszczanie poszczególnych modułów wewnątrz obudowy urządzenia, nawet o bardzo złożonej geometrii. Bezpośrednie przejście z części sztywnej (zwykle zbudowanej w oparciu o laminat FR4, ale nie tylko) na elastyczną (na podłożu polimerowym) pozwala na całkowitą lub przynajmniej znaczną eliminację konwencjonalnych złączy i wtyków oraz wiązek kablowych, co znakomicie ułatwia integrację docelowego produktu i przynosi niemałe oszczędności związane z uproszczeniem procesu montażu. Cienkie paski obwodów elastycznych (o grubości na poziomie ułamka milimetra) zapewniają także nieporównanie większą trwałość w warunkach powtarzalnego zginania – w porównaniu do najlepszych nawet wiązek kablowych. Znacznie wyższa jest też odporność na udary, wibracje i naprężenia, co ma znaczenie w urządzeniach intensywnie eksploatowanych lub pracujących w szczególnie wymagających środowiskach (np. sprzęt militarny). Obwody elastyczne na podłożach poliimidowych pozwalają na uzyskanie promienia gięcia na poziomie 1 mm, zaś zakres temperatur pracy dochodzi nawet do 200°C. Podłoża poliestrowe (PET) zapewniają natomiast wyższą (w porównaniu do poliimidu) wytrzymałość na rozdarcie (800 g vs 500 g).

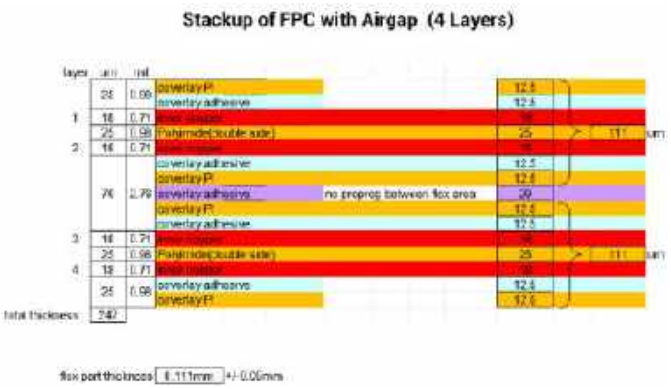
Obwody sztywno-giętkie i elastyczne – możliwości technologiczne

PCBWay świadczy kompleksowe usługi w zakresie produkcji elastycznych obwodów drukowanych o liczbie warstw od 1 do 16 (**fotografie 1...3**), przy czym całkowita grubość PCB może zawierać się w przedziale od 4 do 40 milsów (nie licząc części usztywniającej, np. w przypadku złączy krawędziowych bądź płytek rigid-flex). Tolerancja geometrii obwodów jednowarstwowych wynosi ±1 mils,



Rysunek 2. Budowa stosu 2-warstwowego obwodu elastycznego ze szczeliną powietrzną

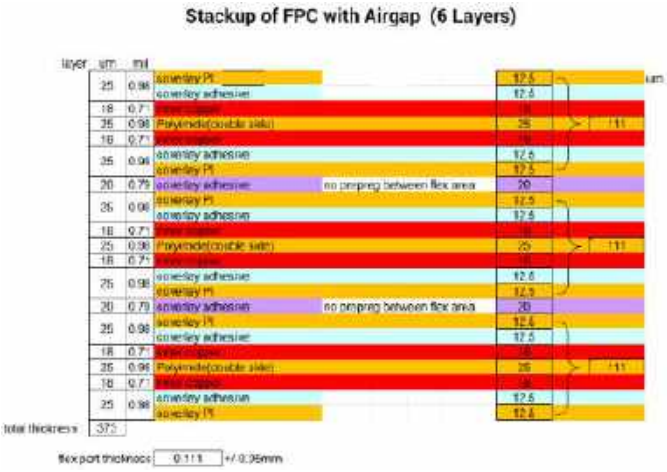
Fotografia 2. Dwuwarstwowy obwód drukowany na podłożu elastycznym



Rysunek 3. Budowa stosu 4-warstwowego obwodu elastycznego ze szczeliną powietrzną

zaś jeśli chodzi o płytki wielowarstwowe, to wartość ta jest nieznacznie większa (±1,2 milsa). Największe produkowane obwody mogą dochodzić nawet do 27,5" w najdłuższym wymiarze, co pozwala na realizację najbardziej rozbudowanych projektów.

Minimalny promień gięcia w przypadku obwodu jednowarstwowego wynosi zaledwie 3...6 grubości PCB, przy konstrukcji



Rysunek 4. Budowa stosu 6-warstwowego obwodu elastycznego ze szczeliną powietrzną

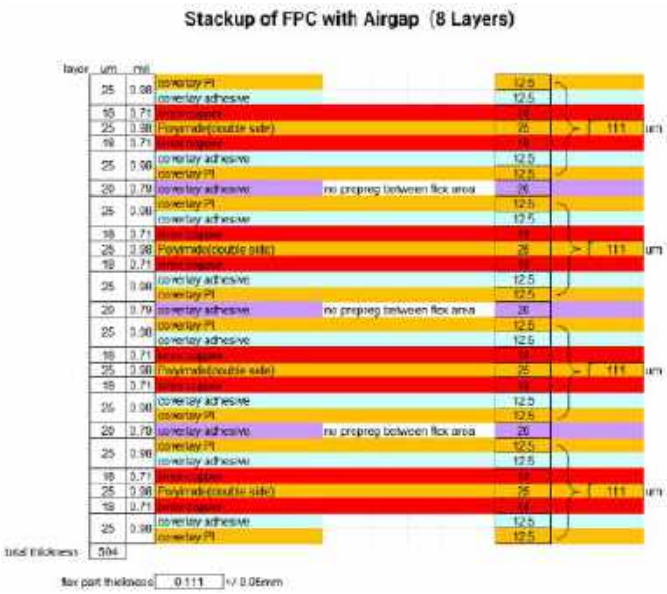
dwuwarstwowej współczynnik ten rośnie do 7...10, zaś rozbudowane stosy wielowarstwowe mogą być bezpiecznie zginane na promieniu będącym 10...15-krotnością grubości PCB.

W aplikacjach wymagających najwyższego stopnia miniaturyzacji (w szczególności w obwodach HDI) duże znaczenie mają limity produkcyjne dotyczące średnicy otworów oraz szerokości ścieżek i odstępów izolacyjnych. Tutaj także PCBWay ma czym się pochwalić – minimalna średnica otworu może wynosić zaledwie 4 milsy, zaś możliwe do wytworzenia szerokości ścieżek i odstępów zaczynają się już od 2 milsów!

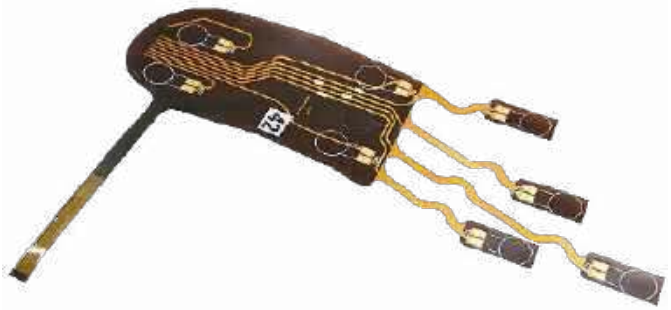
Warto dodać, że PCBWay wytwarza nie tylko obwody standardowe, lecz także płytki przeznaczone do pracy w obwodach radiowych (RF) z impedancją falową kontrolowaną z dokładnością do ±4 Ω (w zakresie do 50 Ω) lub ±7% (przy docelowej wartości impedancji charakterystycznej powyżej 50 Ω).

Na koniec tej części artykułu dodajmy jeszcze informacje na temat wykończenia obwodów elastycznych i sztywno-giętkich. Producent oferuje dwa podstawowe kolory soldermaski (zielony oraz czarny), stosowane w części usztywnionej (tj. przeznaczonej do montażu komponentów SMD). W zakresie zabezpieczenia powierzchni padów firma umożliwia wybór jednej z następujących technologii:

- HASL,
- ENIG,
- ENEPIG,



Rysunek 5. Budowa stosu 8-warstwowego obwodu elastycznego ze szczeliną powietrzną



Fotografia 3. Wielowarstwowa płytka elastyczna o nietypowej geometrii

- pokrycia elektrolityczne nikiel + złoto,
- miękkie złocenie,
- twarde złocenie,
- srebrzenie immersyjne,
- tymczasowe pokrycia organiczne (tzw. OSP),
- cynowanie immersyjne.

Szczegóły dotyczące konstrukcji stosu obwodów wielowarstwowych

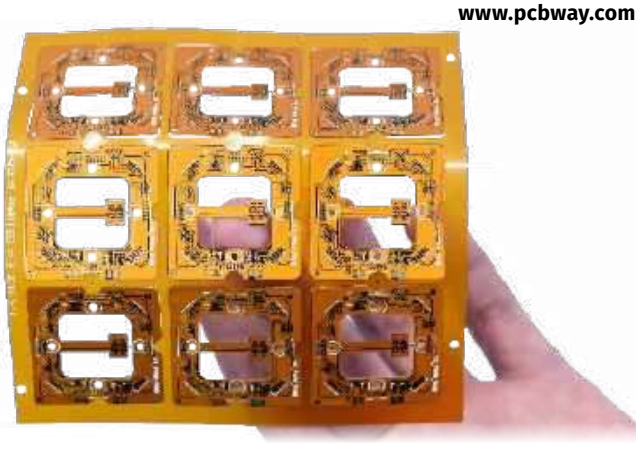
W projekcie każdej wielowarstwowej płytki drukowanej jednym z najważniejszych parametrów technologicznych jest konstrukcja stosu. PCBway oferuje szeroką gamę standardowych stosów – przykładowo: w przypadku płytek dwuwarstwowych dostępnych jest aż 19 predefiniowanych opcji o grubości całkowitej od 0,1 mm do 0,38 mm. Obwody elastyczne o 4 warstwach występują w 8 wersjach grubości całkowitej (od 0,2 mm do 0,33 mm), a wciąż mówimy tutaj tylko o płytkach bez części sztywnej – w obszarze konstrukcji hybrydowych część sztywna bazuje na laminacie szklano-epoksydowym FR4, nadającym temu rejonowi płytki grubość finalną 1,549 mm.

Szczególną odmianą wielowarstwowych obwodów elastycznych są konstrukcje ze szczeliną powietrzną (ang. airgap), tj. całkowicie pozbawione warstwy prepregu w części giętkiej. PCBWay wytwarza tego rodzaju płytki drukowane w wersjach 2-, 4– 6– i 8-warstwowych – odpowiednie konstrukcje stosu można zobaczyć na rysunkach.

Podsumowanie

Obwody elastyczne oraz sztywno-giętkie powoli tracą już pozycję symbolu nowoczesności, stając się technologią coraz szerzej rozpowszechnioną, a w wielu zastosowaniach – wręcz konieczną. Z tego względu bogata oferta usług PCBWay obejmuje nie tylko produkcję niemal wszystkich odmian obwodów sztywnych, ale także rozmaitych wykonań elastycznych 1...16-warstwowych. Wieloletnie doświadczenie zespołu PCBWay, w połączeniu z rozbudowanym parkiem maszynowym i najnowocześniejszą aparaturą kontrolno-pomiarową, pozwalają firmie na wytyczanie nowych standardów w zakresie produkcji zaawansowanych obwodów do najbardziej wymagających aplikacji.

www.pcbway.com



Najważniejsze parametry:
- konstrukcja zbudowana w oparciu o scalony zegar RTC RV-3028,
- cztery opcje zasilania rezerwowego: bateria CR1220, bateria ładowalna VL1220, superkondensator lub kondensator ceramiczny,
- fabryczna kalibracja z dokładnością ±1 ppm @ 25°C.

Dodatkowe materiały do pobrania ze strony www.ulubionykiosk.pl/media
AVT5901 Moduł z zegarem RTC i pamięcią FRAM po I²C (EP 11/2021)

* **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**

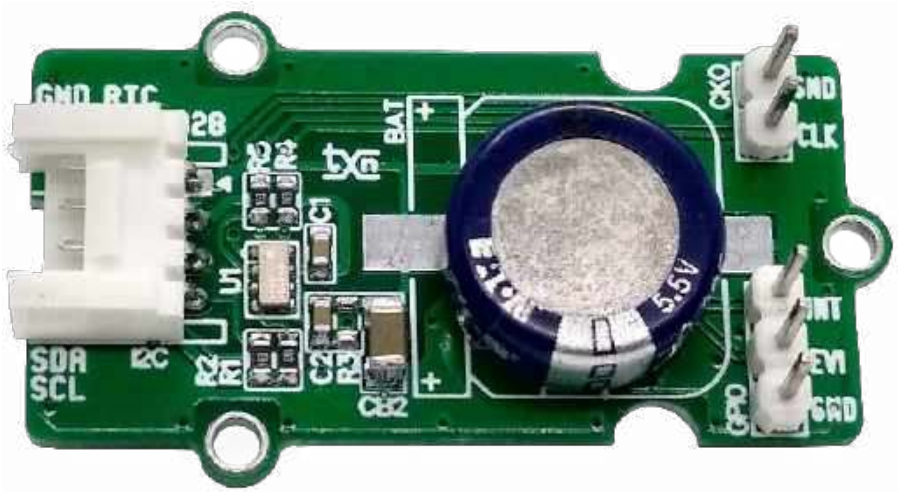
W przypadku braku dostępności na stronie sklepu osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: **kity@avt.pl**.

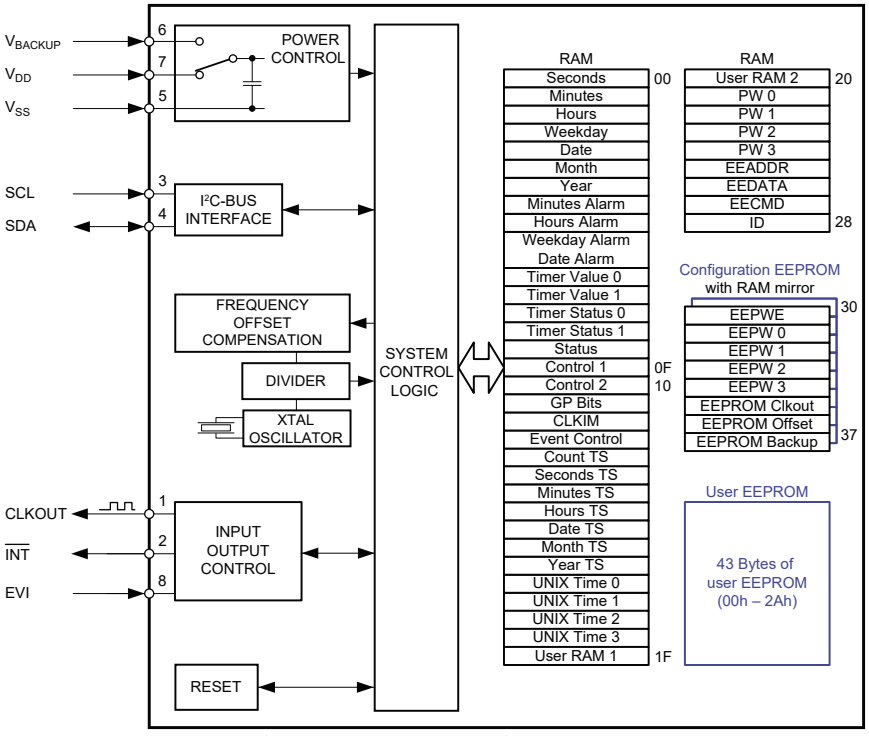
Moduł precyzyjnego zegara RTC z interfejsem Grove

Prezentowany moduł precyzyjnego zegara RTC oparty o układ RV-3028 przyda się w aplikacjach, w których zależy nam na dokładnym odmierzaniu czasu rzeczywistego.

Moduł Grove_RTC_RV3028 oparto o układ RV-3028, którego strukturę wewnętrzną pokazano na **rysunku 1**. Jest to nowoczesny, kalibrowany fabrycznie (±1 ppm, 25°C) zegar czasu rzeczywistego z obsługą podtrzymania bateryjnego oraz wyjątkowo niskim poborem prądu (45 nA, 3 V). Układ oferuje możliwość generowania alarmu na wyjściu przerwania INT i sygnału zegarowego o programowanej częstotliwości na wyjściu CLKOUT, obsługuje także przerwanie zewnętrzne na wejściu EVI. Układ podtrzymania bateryjnego współpracuje z bateriami (także ładowalnymi) oraz superkondensatorami. Wbudowany obwód Trickle-Charger z czterema ustalanymi programowo wartościami prądu ładowania upraszcza konstrukcję układu. Do dyspozycji projektanta dostępne są 43 bajty pamięci EEPROM, warto też dodać, że rejestry wewnętrzne mogą być zabezpieczone przed zmianą zawartości za pomocą hasła użytkownika. Do RV-3028 oferowane jest wsparcie w postaci bibliotek przeznaczonych min. do Arduino oraz systemu Linux. W porównaniu do popularnych zegarów RTC typu DS1307 itp., układ oferuje znaczącą poprawę parametrów i zwiększoną funkcjonalność, warto więc rozważyć jego zastosowanie w nowych projektach.

Schemat modułu zegara RTC, ukazano na **rysunku 2**. Moduł jest zgodny ze standardem Grove, zasilanie i magistrala danych doprowadzone zostały do złącza I²C. Rezystory





Rysunek 1. Budowa RV-3028 (za notą Micro Crystal AG)

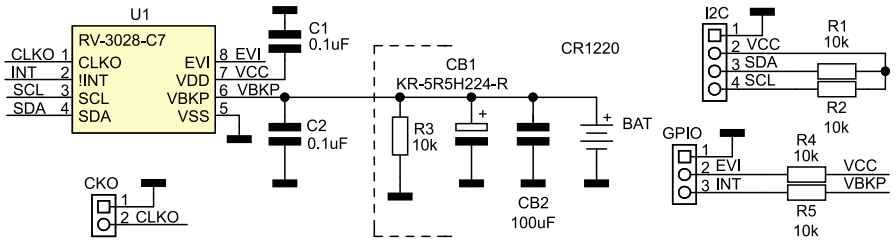
Wykaz elementów:

Rezystory: (SMD 0603, 5%)
R1...R5: 10 kΩ

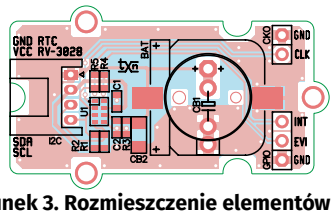
Półprzewodniki:
U1: RV-3028-C7-TAQC

Kondensatory:
C1, C2: 100 nF (SMD 0603, X7R, 10 V)
CB1: superkondensator 0,22 F/5 V (typ KR-5R5H224-R, patrz opis)
CB2: 100 µF/6,3 V (SMD 1206, patrz opis)

Pozostałe:
BAT: bateria CR1220 z podstawką (patrz opis)
CKO: złącze SIP2 2,54 mm proste
GPIO: złącze SIP3 2,54 mm proste



Rysunek 2. Schemat modułu RTC



Rysunek 3. Rozmieszczenie elementów

R1 i R2 polaryzują szynę I²C i mogą zostać pominięte, jeżeli odpowiednie pull-upy są już wbudowane we współpracujący układ. RV-3028 wyposażony został w obwody podtrzymania zasilania, przełączające automatycznie źródła napięcia podczas zaniku zasilania podstawowego (VCC). Źródło zapasowe podłączone jest do wyprowadzenia VBKP. W zależności od preferencji i wymaganego czasu podtrzymania można zastosować baterię litową lub akumulatorki. Wbudowany układ backup zawiera diodę Schottky'ego blokującą przepływ wsteczny do VCC oraz konfigurowane programowo cztery rezystory o wartościach 3, 5, 9 i 15 kΩ, umożliwiające ograniczenie prądu ładowania. Jeżeli podtrzymanie nie jest wymagane, wyprowadzenie VBKP trzeba podłączyć do masy poprzez rezystor 10 kΩ, a obwód ładowania pozostawić wyłączony. Moduł obsługuje podtrzymanie napięcia za pomocą superkondensatora CB1 0,22 F/ 5,5 V, kondensatora ceramicznego CB2 100 µF/ 6,3 V oraz typowej baterii CR1220 montowanej w podstawce SMD (Connfly DS1092-12-N8S). Oczywiście należy wlutować tylko jeden element podtrzymujący. Akceptowalne jest zastosowanie litowych baterii ładowalnych, np. VL1220, ale należy zwrócić szczególną uwagę na dopuszczalne maksymalne napięcie ładowania danego modelu. RV-3028 nie stabilizuje różnicy potencjałów na zaciskach źródła, a jedynie ogranicza prąd poprzez wybór wartości rezystora szeregowego. W przypadku zasilania RTC z napięcia 5 V, po odjęciu spadku na wewnętrznej diodzie Schottky'ego, na wyprowadzeniu VBKP napięcie będzie wynosić ok. 4,5 V, co przekracza maksymalne dopuszczalne dla VL1220 3,55 V. Z kolei przy zasilaniu 2,7 V napięcie na VBKP wynosić będzie ok. 2,2 V, co nie pozwoli doładować baterii.

Na złącze GPIO wyprowadzone są sygnały wyjścia przerwania INT oraz wejścia przerwania EVI. Linia INT pozwala na sygnalizację alarmów generowanych przez RV-3028 związanych z czasem, zasilaniem lub przerwaniem zewnętrznym. Wyjście jest typu OD i podciągnięte jest przez R5 do VBKP, co zapewnia działanie także podczas braku zasilania VCC i umożliwia – przykładowo – wybudzanie systemu ze stanu uśpienia. Wejście EVI może być używane do rejestracji przerwania zewnętrznego ze znakiem czasowym zapisywanym w rejestrach U1. Na złącze CKO wyprowadzony został sygnał z konfigurowanego generatora zegarowego; w zależności od ustawień układ jest zdolny do generowania przebiegu o częstotliwości 1, 32, 64, 1024, 8192, 32768 Hz, którego można użyć do czasowego wybudzania systemu.

Grove_RTC_RV3028 zmontowany jest na miniaturowej dwustronnej płytce drukowanej, mechanicznie zgodnej z systemem Grove. Rozmieszczenie elementów zaprezentowano na **rysunku 3**. Zmontowany został pokazany na **fotografii otwierającej artykuł**.

Montaż jest typowy i nie wymaga opisu, należy tylko określić, czy zastosowane będzie zasilanie rezerwowe. Jeżeli aplikacja nie wymaga podtrzymania RTC, montujemy tylko R3, pomijając C2, CB1, CB2, BAT. Jeżeli podtrzymanie będzie korzystać z baterii, montujemy C2 i podstawkę CR1220 wraz z baterią. Jeżeli stosujemy superkondensator, to montujemy C2 oraz – w zależności od wymaganego czasu podtrzymania – kondensator CB1 (0,22 F) lub ceramiczny CB2 (100 µF).

Sam układ nie wymaga uruchamiania; w celu sprawdzenia jego działania można skorzystać z Arduino i przeznaczonej do niego biblioteki RTC RV-3028-C7 lub Raspberry Pi i biblioteki pimoroni/rv3028-python (dostępne w serwisie GitHub: https://github.com/pimoroni/rv3028-python).

Szybkie sprawdzenie poprawności komunikacji da się w prosty sposób wykonać za pomocą Raspberry Pi. RV-3028 powinien być dostępny pod adresem 0x52, co sprawdzimy poleceniem:

```
pi@PI4:~ $ i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9
a  b  c  d  e  f
50: -- -- 52 -- -- -- -- -- -- --
-- -- -- -- -- --
```

Zawartość rejestrów oraz odliczanie czasu natomiast da się zweryfikować poprzez wykonanie zrzutu rejestrów z adresu 0x52:

```
pi@PI4:~ $ i2cdump -y -r 0x00-
0x3f 1 0x52 b
     0  1  2  3  4  5  6
7  8  9  a  b  c  d  e  f
0123456789abcdef
00: 25 46 09 01 20 08 23 80 80
80 00 00 00 00 30 00    %F??
?#???....0.
10: 00 00 00 00 00 00 00 00
00 00 00 00 84 20 96 49 00
..........? ?I.
20: 00 00 00 00 00 37 1c
00 33 00 00 0a 7a 12 19 19
.....7?.3..?z???
30: 00 00 00 00 00 c0 01
1c a9 19 00 00 00 00 00 00
.....?????.....
```

Jeżeli wszystko działa, można moduł zastosować we własnej aplikacji.

Adam Tatuś, EP

Najważniejsze parametry:
- konstrukcja oparta na sprzętowym liczniku scalonym typu S-35770,
- wejście zliczające aktywowane zboczem narastającym,
- sprzętowe wejście resetowania licznika,
- interfejs I²C do odczytu zawartości licznika i obsługi rejestru użytkownika,
- zasilanie: 1,5...5,5 V/300 µA (maks.)

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

AVT6038	Miniwyświetlacz LCD 4×10 znaków z podświetleniem i interfejsem I²C (EP 4/2024)
AVT6037	Pięciokanałowy termometr I²C (EP 4/2024)
AVT6025	Sterownik mikrosilników prądu stałego (EP 2/2024)
----	Aktywny hub I²C Grove (EP 2/2024)
----	Bufor I²C Grove (EP 1/2024)

* **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**

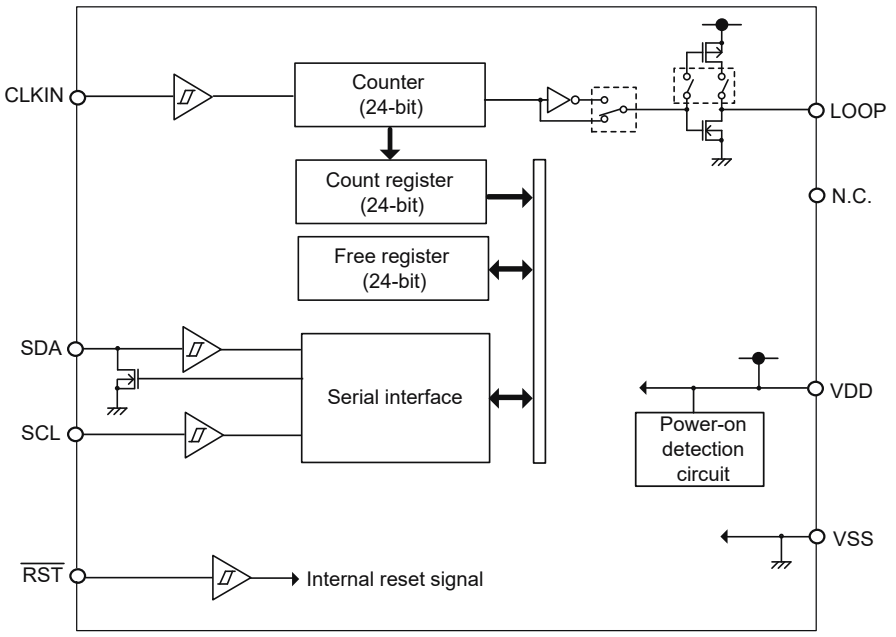
W przypadku braku dostępności na stronie sklepu osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: **kity@avt.pl**.

24-bitowy sprzętowy licznik impulsów z interfejsem I²C

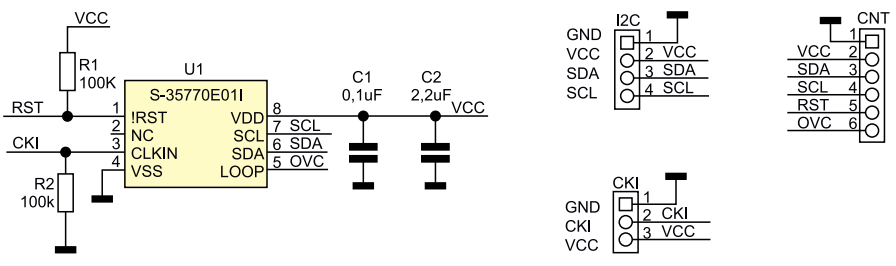
Moduł 24-bitowego licznika impulsów, przydatny w domowej automatyce, np. do zliczania sygnałów z przystawek impulsowych liczników mediów.

Moduł bazuje na układzie 24-bitowego licznika z interfejsem I²C typu S-35770 firmy Ablic, którego budowę pokazano na **rysunku 1**.

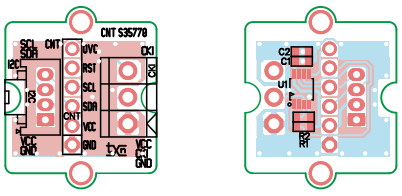
Schemat nakładki zaprezentowano na **rysunku 2**. Układ zasilany jest napięciem z zakresu 1,5...5,5 V, pobór prądu podczas aktywnej transmisji nie przekracza 300 µA (w spoczynku wynosi poniżej 0,01 µA). Zasilanie i magistrala I²C doprowadzone zostały do złącza oznaczonego I²C. Zliczane impulsy doprowadzone są do złącza CKI, pojemność licznika to 16777215 impulsów (0xFFFFFF, rozdzielczość 24-bitowa). Zmiana stanu licznika jest aktywowana narastającym zboczem sygnału CKI, po osiągnięciu wartości 0xFFFFFF rejestr ulega skasowaniu i zliczanie rozpoczyna się od wartości 0x000000. Układ ma aktywowane stanem niskim wejście RST, służące do jego sprzętowego zerowania; sygnał ten kasuje tylko zawartość licznika, nie ma wpływu na komunikację I²C. Ponadto dostępny jest sygnał przepełnienia OVC (wyprowadzenie LOOP), zmieniający stan na przeciwny po każdej zmianie wartości zliczonych impulsów z 0xFFFFFF na 0x000000. Sygnał OVC po resecie układu ustawia się na 0. Umożliwia on kaskadowe łączenie wielu modułów w celu uzyskania większej pojemności. Do złącza CKI doprowadzone jest też zasilanie, co ułatwia podłączenie zewnętrznych przetworników wymagających zasilania lub polaryzacji styku impulsatora. Sygnały RST, OVC wraz z zasilaniem oraz I²C dostępne



Rysunek 1. Budowa wewnętrzna S-35700 (za notą Ablic)



Rysunek 2. Schemat modułu



Rysunek 3. Rozmieszczenie elementów Grove_CNT_S35770 (a – warstwa TOP, b – warstwa BOTTOM)

Fotografia 1. Zmontowany moduł

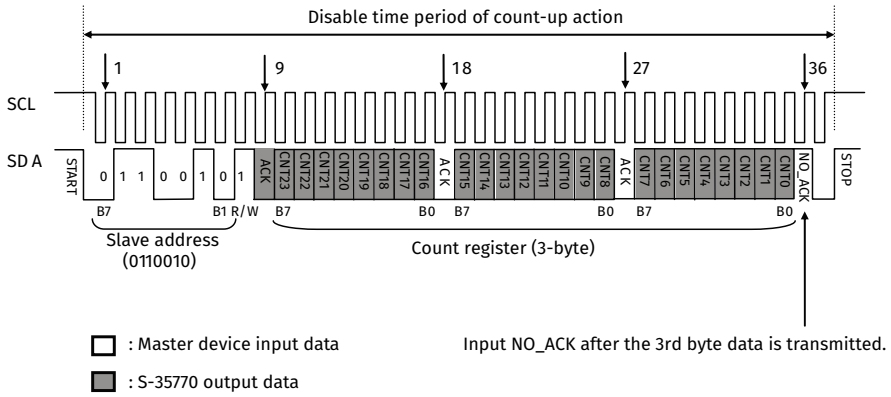
Wykaz elementów:

Rezystory:
R1, R2: 100 kΩ (SMD 0603, 1%)

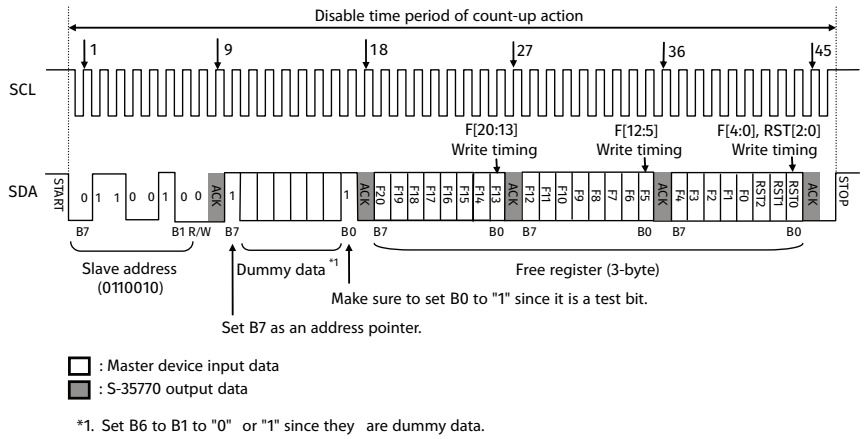
Kondensatory: (SMD 0603 X7R 10 V)
C1: 100 nF
C2: 2,2 µF

Półprzewodniki:
U1: S-35770E01I (TMSOP8)

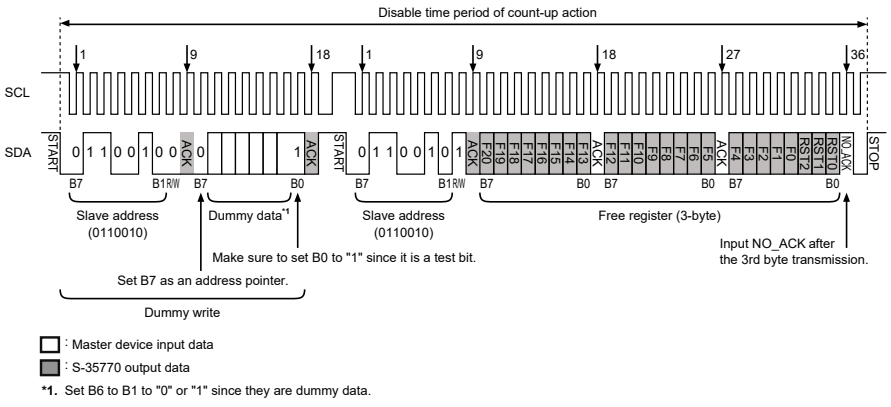
Pozostałe:
CKI: złącze śrubowe DG 3-pin. 3,5 mm (DG381-3.5-3)
CNT: listwa SIP6
I²C: złącze Grove proste (110990030)



Rysunek 4. Odczyt rejestrów licznikowych



Rysunek 5. Zapis rejestrów użytkownika



Rysunek 6. Odczyt rejestrów użytkownika

są na złączu CNT. Układ wykazuje wrażliwość na ładunki statyczne, należy więc zachować odpowiednią ostrożność.

Układ zmontowany został na dwustronnej płytce drukowanej zgodnej z Grove. Rozmieszczenie elementów zobrazowano na **rysunkach 3a** i **3b.** Sposób montażu nie wymaga opisu. Zmontowany moduł zaprezentowano na **fotografii 1**.

Moduł nie wymaga uruchamiania. Układ S-35770 dostępny jest na magistrali I²C pod adresem 0x32. Licznik ma dwie grupy rejestrów 24-bitowych (3×8 bitów): pierwsza to rejestr licznikowy, w którym zliczane są impulsy, druga to uniwersalny rejestr użytkownika, którego trzy najmłodsze bity – ustawione na 010 – używane są do programowego kasowania stanu licznika. Jeżeli korzystamy z rejestrów użytkownika, należy pamiętać, aby najmłodsze 3 bity ustawić w pozostałych przypadkach na 111. Sekwencję odczytu rejestrów licznika pokazano na **rysunku 4**, a sekwencję zapisu zaprezentowano na **rysunku 5**. Należy pamiętać, że zapis musi odbywać się zawsze w pakiecie 3 bajtów, w przeciwnym razie układ może działać nieprawidłowo. Sekwencja odczytu z rejestrów użytkownika widoczna jest na **rysunku 6**. Podczas aktywnej transmisji I²C zliczanie impulsów ulega zablokowaniu, aby zapobiec zmianie stanu licznika podczas odczytu.

Szybkiego sprawdzenia działania układu można dokonać przy użyciu Arduino i biblioteki *ABLIC_S35770_Demo*, dołączonej do materiałów dodatkowych. Szkic komunikuje się z układem S-35770 poprzez I²C oraz dwa wyprowadzenia cyfrowe: na jednym generowany jest reset licznika, na drugim – testowo – impulsy do zliczania. Po zmianie wyprowadzeń w konfiguracji:
ABLIC_S35770 myABLIC_S35770(11, 10)
gdzie:
11 – pin sygnału RST, 10 – pin sygnału CKI, możemy przetestować komunikację i cykle zliczania. Jeżeli wszystko działa poprawnie, moduł może zostać zastosowany we własnej aplikacji.

Adam Tatuś, EP

W ofercie AVT*
AVT6042

Najważniejsze parametry:
- konstrukcja oparta na scalonym mostku H typu TB67H45x,
- obciążalność mostka: 1,5 A (ciągła)/3,5 A (szczytowa),
- zakres napięć zasilania: 4...50 V,
- wejścia sterujące: IN1,2 (kierunek ruchu, hamowanie),
- możliwość sterowania prędkości obrotowej sygnałem PWM.

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

----	Mikrodriver silnika DC małej mocy (EP 1/2023)
AVT5923	Płynny regulator prędkości i kierunku obrotów silnika 12 V (EP 3/2022)
AVT5879	Monitor pracy wentylatora (EP 8/2021)
Projekt 246	Sterownik wentylatora wyciągu łazienkowego (kuchennego) (EP 10/2019)
AVT5698	Sterownik wentylatorów 12 V dużej mocy (EP 8/2019)

* **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**.

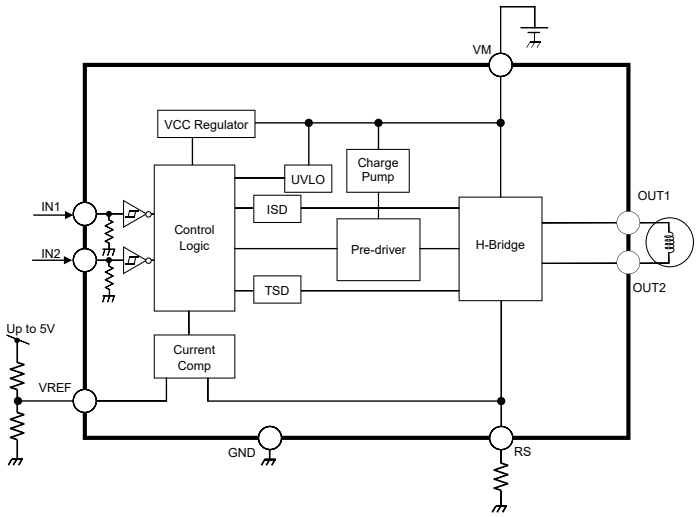
Ministerownik silnika komutatorowego małej mocy

Niewielki moduł drivera silnika komutatorowego małej mocy – przydatny w robotyce amatorskiej, oparty na układzie TB67H45x marki Toshiba.

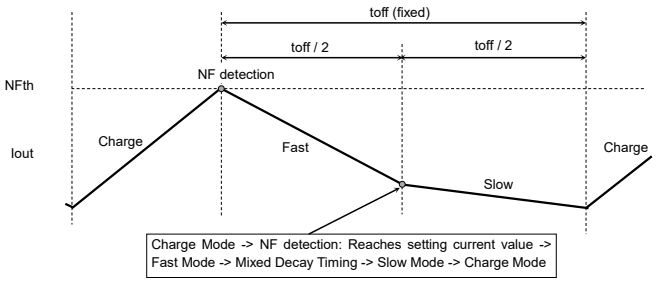
Strukturę wewnętrzną układu TB67H45x pokazano na **rysunku 1.**

TB67H450/451 zwiera jednokanałowy stopień mocy (mostek H) o obciążalności do 1,5 A (maksymalnie 3,5 A) i zalecanym napięciu pracy 5...40 V. Niska rezystancja kluczy – wynosząca maksymalnie 0,8 Ω – ogranicza moc strat. Układ wyposażony został w zabezpieczenie przeciążeniowe (ISD) i termiczne (TSD) oraz obwody wyłączające mostek po zbyt głębokim spadku napięcia zasilania (ULVO). W zależności od typu układu zabezpieczenie przeciążeniowe jest zatrzaskiwane (w przypadku TB67H450) lub samopowrotne (w modelu TB67H451).

Pełne sterowanie mostkiem umożliwia uzyskanie czterech stanów pracy: obroty w przód, tył, hamowanie i zatrzymanie wirnika. Logikę sterowania wyprowadzeń IN1,2 zaprezentowano w **tabeli 1**. Gdy oba wejścia sterujące IN1,2 są w stanie niskim przez czas dłuższy niż 1 ms, układ automatycznie przechodzi w stan obniżonego poboru mocy, pobierając prąd nie większy niż 1 µA. Wybudzanie układu trwa nie dłużej niż 30 µs, co pozwala na efektywne oszczędzanie energii, szczególnie



Rysunek 1. Schemat wewnętrzny TB67H45x (za notą Toshiba)



Rysunek 2. Fazy sterowania silnikiem w trybie Chopper

Wykaz elementów:

Rezystory:
R1: 2,2 kΩ (SMD 0805)
RM: 0,22 Ω (SMD 2512, 2W)

Kondensatory:
C1, C2: 100 nF/50 V (SMD 0805)
CE1: 47 µF/50 V (elektrolityczny D=8 mm, r=3,5 mm)

Półprzewodniki"
U1: TB67H450AFNG lub TB67H451AFNG (HSOP8)

Pozostałe:
MTR, PWR: złącze DG 2-pin 3,5mm (DG381-3.5-2)
RV: potencjometr montażowy pionowy 5 mm 10 kΩ

w układach zasilanych bateryjnie lub akumulatorowo, bez konieczności używania dodatkowych wyprowadzeń sterujących.

Regulacja obrotów może być realizowana metodą bezpośredniego PWM lub metodą przerywania ustalonego prądu maksymalnego w trybie Chopper. Tryb PWM ulega aktywacji, gdy wyprowadzenie RS podłączone jest do masy, a sygnał PWM (o częstotliwości maksymalnej 400 kHz) doprowadzony jest do jednego z wyprowadzeń IN1, IN2. Maksymalna wartość prądu w trybie Chopper definiowana jest kombinacją oporności rezystora podłączonego do wyprowadzenia RS i wartości napięcia odniesienia na wyprowadzeniu VREF. W trybie Chopper także możliwa staje się regulacja prędkości metodą PWM. Maksymalny prąd można obliczyć ze wzoru:

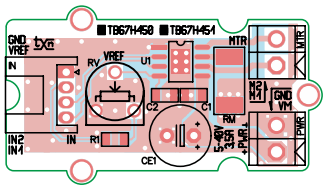
$$I_{omax}=V_{ref(gain)}\cdot V_{ref}/R_{rs}$$

gdzie

$$V_{ref(gain)}=1/10{,}0$$

Napięcie regulacyjne V_{ref} musi zawierać się w przedziale 1...4 V, wejście V_{ref} akceptuje napięcia w przedziale 0...5,5 V. W modelu przy zasilaniu 5 V, V_{ref}~=0...4 V, RS=0,22 Ω, prąd maksymalny wynosi ok. 1,85 A. Regulacja odbywa się w trzech fazach, co pokazano na **rysunku 2**.

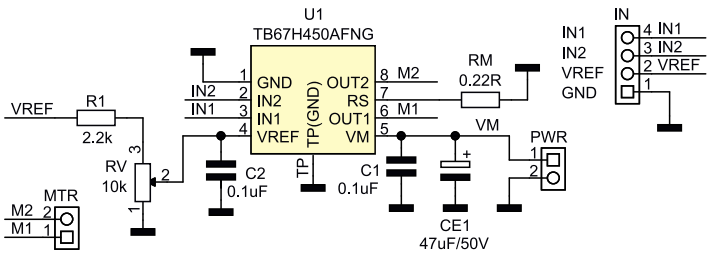
W pierwszej fazie mostek H załącza prąd na uzwojenie, aż do momentu osiągnięcia detekcji ustalonego prądu (NFth). W drugiej fazie silnik hamowany jest przeciwprądem (fast decay), natomiast na trzecim etapie



Rysunek 4. Rozmieszczenie elementów

Tabela 1. Logika sterowania TB67H45x

IN1	IN2	OUT1	OUT2	Mode
L	L	OFF (Hi-Z)	OFF (Hi-Z)	Stop
				Standby mode after tstby
H	L	H	L	Forward
L	H	L	H	Reverse
H	H	L	L	Short brake



Rysunek 3. Schemat ideowy modułu

– prądem zwarcia (slow decay). Czas trwania drugiej i trzeciej fazy sterowania jest ustalony na 25 µs z 50-procentowym podziałem pomiędzy fazę drugą i trzecią. Jeżeli podczas drugiej fazy prąd zaniknie (I=0), wyjścia mostka ustawiane są w stan wysokiej impedancji.

Schemat modułu pokazano na **rysunku 3**.

Aplikacja TB67H45x okazuje się bardzo prosta: moduł zasilany jest napięciem stałym ze złącza PWR, odsprzęganym przez CE1, C1. Silnik podłączony jest do złącza MTR, natomiast sterowanie i napięcie VREF – do złącza IN. W modelu domyślnie wybrany jest tryb Chopper, co wymaga montażu odpowiednio obliczonego RM; prąd ograniczenia ustalony został na ok. 1,8 A (Vref=5 V, RM=0,22 Ω). Potencjometr RV umożliwia zmniejszenie prądu maksymalnego.

Minimoduł zmontowany został na dwustronnej płytce drukowanej zgodnej z Grove. Rozmieszczenie elementów zobrazowano na **rysunku 4**.

Sposób montażu jest klasyczny i nie wymaga opisu, należy jedynie zwrócić uwagę na wybór odpowiedniej wersji układu U1: TB67H450/451 oraz poprawnie przylutować jego pad termiczny. W przypadku forsownej pracy na U1 warto nakleić radiator i wymusić obieg powietrza chłodzącego. Zmontowany minimoduł pokazano na **fotografii tytułowej**.

Moduł nie wymaga uruchamiania – zmontowany ze sprawnych elementów dobranych pod kątem parametrów używanego silnika, działa od razu po włączeniu zasilania i doprowadzeniu sygnałów sterujących. Moduł pracuje poprawnie zasilany napięciem 5 V, możliwa jest też praca z nieco niższym zasilaniem. Próg detekcji stanu wysokiego na wyprowadzeniach IN1, IN2 wynosi 2 V, więc po przeliczeniu dzielnika R1, RV i RM, tak aby spełnić wymogi prądu maksymalnego silnika, można go zasilać niższym napięciem np. 3,3 V.

Adam Tatuś, EP

Półprzewodnikowa rewolucja w domenie open source

Hasło „open source" kojarzy się większości osób przede wszystkim z oprogramowaniem tworzonym przez pasjonatów informatyki i udostępnianym za darmo ogólnoświatowej społeczności użytkowników. Nie mniej interesująca jest jednak grupa otwartych rozwiązań sprzętowych, czyli open source hardware – mało kto wie, że istnieją już bezpłatne pakiety oprogramowania przeznaczonego do projektowania układów scalonych, w tym nawet układów ASIC o paśmie rzędu setek GHz. Jeszcze większym zaskoczeniem może być fakt, że fabryki półprzewodników otwierają swoje podwoje także dla małych firm, partnerów akademickich, a nawet... odbiorców prywatnych!

Pomimo istnienia pewnego wsparcia open source hardware (takiego jak: obwody, układy, elementy i przyrządy elektroniczne), domena ta nie doczekała się tak szerokiego upowszechnienia efektów pracy twórców, jak miało to miejsce w przypadku open source software.

Pewną formę pośrednią stanowi oprogramowanie open source EDA (Electronic Design Automation), którego sztandarowym przykładem jest KiCAD do projektowania obwodów drukowanych. Inny przykład to kody źródłowe w języku opisu sprzętu HDL (Hardware Description Language), które – choć zbliżone w formie do kodów oprogramowania – służą do opisu oraz testowania układów/systemów elektronicznych (w zdecydowanej większości cyfrowych). Opisana wyżej dysproporcja między oprogramowaniem a sprzętem wynika głównie z ograniczenia dostępności zasobów i możliwości produkcyjnych (produkcji i montażu płytek PCB w przypadku elektroniki dyskretnej czy produkcji układów scalonych w warunkach laboratoryjnych, tzw. clean room) oraz całego środowiska testowego. Ze względu na wspomniane przeszkody trudniej jest tutaj uzyskać zamierzony efekt finalny, aby doświadczyć emocji związanych z jego użytkowaniem lub – co ważniejsze – móc zaoferować ów działający i sprawdzony projekt innym w domenie open source.

Od końca pierwszej dekady XXI wieku nastąpił dynamiczny rozwój projektów open source hardware. Choć popularność platform, takich jak Arduino czy tych opartych na rozwiązaniach firmy ST Microelectronics, pozwoliła na rozpowszechnienie wiedzy oraz projektów w obrębie open source hardware, to prawdziwy kamień milowy stanowiło pojawienie się RISC-V – wolnej architektury RISC niezależnej od rynkowych potentatów. Kluczem do sukcesu RISC-V okazała się modularność i dostępność narzędzi do tworzenia oraz testowania aplikacji, które składają się na cały ekosystem RISC-V. Wciąż nierozwiązany pozostawał jednak problem dostępu do środków produkcji, które w przypadku rozwiązań opartych na krzemie były poza zasięgiem zwykłego śmiertelnika, głównie ze względu na barierę cenową.

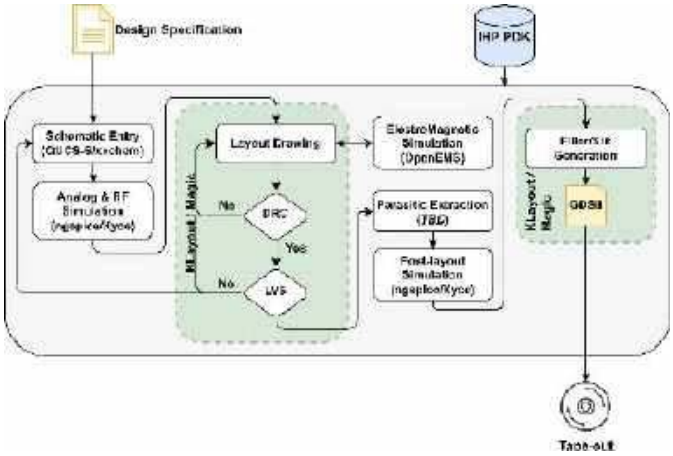
Równolegle, aby przyspieszyć innowacje w branży elektronicznej, amerykańska agencja rządowa DARPA (Defense Advanced Research Agency) ogłosiła w czerwcu 2017 roku utworzenie Electronics Resurgence Initiative (ERI). ERI ze swoim ogromnym finansowaniem wezwała w 2018 roku do opracowywania innowacyjnych podejść do materiałów, projektów i architektur mikrosystemów. Tak powstał projekt OpenROAD atakujący bariery kosztów i wiedzy specjalistycznej, które blokują wykonalność projektowania sprzętu w zaawansowanych technologiach. OpenROAD dążył do opracowania narzędzi open source umożliwiających autonomiczne, 24-godzinne generowanie układów scalonych na podstawie opisu w języku HDL oraz specyfikacji.

Sytuacja zmieniła się diametralnie w lipcu 2020 roku, kiedy to do domeny publicznej został po raz pierwszy w historii uwolniony PDK (Process Design Kit) SKY130 firmy Skywater. Na PDK składa się kompletny zestaw informacji na temat przyrządów półprzewodnikowych oraz wytycznych do projektowania w zakresie danego procesu technologicznego. Można w nim znaleźć między innymi modele urządzeń, statystyki parametrów modeli, zasady projektowania, specyfikację procesu: poziomy domieszkowania krzemu, geometrię przyrządów, zasady DRC (Design Rule Check). W przypadku bloków cyfrowych dostarczane są biblioteki standard cell, które zawierają w sobie opisy behawioralne HDL, opisy w języku SPICE, charakterystyki czasów propagacji oraz strat mocy i layouty bramek logicznych. Wraz z przeniesieniem do domeny open source informacji o procesie SKY130 amerykańska firma efabless podjęła się zadania dostarczenia użytkownikom zautomatyzowanych narzędzi open source oraz ustrukturyzowania informacji dostarczonej przez Skywater, aby była ona kompatybilna z użytymi narzędziami programowymi. Tak powstał OpenLane (bazujący w 80% na OpenROAD), czyli zestaw darmowych programów oraz skryptów konfiguracyjnych, który dostarcza użytkownikowi możliwość wygenerowania dokumentacji produkcyjnej w formacie GDSII, na podstawie behawioralnego opisu układu cyfrowego w języku Verilog. Dodatkowo efabless zajęła się zarządzaniem sponsorowanymi przez Google cyklami produkcyjnymi OpenMPW (Open Multi Project Wafer) w technologii SKY130. W ramach programu OpenMPW użytkownicy otrzymywali do dyspozycji 10 mm^2 na waflu krzemowym, w postaci modułu wewnątrz układu scalonego o wdzięcznej nazwie Caravel. Opisana metoda, oprócz dostarczenia użytkownikowi platformy mikroprocesorowej w postaci Pico RISC-V, magistrali Wishbone oraz wewnętrznego analizatora logicznego, wyręcza projektanta w żmudnej pracy projektowania padringu oraz sieci dystrybucji zasilania. Jedynym wymaganiem, aby otrzymać dostęp do krzemu, było przekazanie do domeny open source projektu zgłoszonego do produkcji w ramach OpenMPW.

W tym miejscu warto nacisnąć STOP i zastanowić się nad tym, co wydarzyło się w 2020 roku. Otóż jeden z pracowników Google, Tim Ansell, przekonał swojego pracodawcę do słuszności finansowania Open Silicon Initiative. Jednym z kluczowych argumentów była prognoza podaży specjalistów w dziedzinie mikroelektroniki na rynku pracy pod koniec dekady 2020...2030, która ujawniła bardzo duże niedobory kadrowe, a także problemy strukturalne w kształceniu nowych pracowników. Ponadto trend technologiczny zakładał zastosowanie wyspecjalizowanych obwodów obliczeniowych, które ktoś musi zaprojektować i zweryfikować, a rosnący w związku z tym nakład czasu i zasobów to kolejny powód bólu głowy menedżerów. W Google wdrożono więc pomysł, którego celem było zainteresowanie ludzi mikroelektroniką, przy jednoczesnym drastycznym obniżeniu bariery dostępu. Cała prezentacja Tima jest dostępna na kanale YouTube Fundacji FOSSi (Free Open Source Silicon) wraz z innymi interesującymi wykładami osób zaangażowanych w Open Source Silicon.

W 2021 roku na tej samej zasadzie sfinansowany został projekt przeniesienia informacji o procesie Global Foundries GF180 do domeny publicznej. Do chwili obecnej efabless ma za sobą organizację ośmiu edycji OpenMPW w technologii SKY130 oraz dwóch w technologii GF180, a publiczne repozytorium projektów na stronach efabless liczy ponad 1000 układów, zarówno analogowych, jak i cyfrowych.

W kalendarzu najważniejszych wydarzeń związanych z Open Source Silicon należy odnotować datę 20 lutego 2023 roku, kiedy to niemiecki instytut



Rysunek 1. Zestaw narzędzi open source do projektowania układów analogowych z urzględnieniem pasma RF

badawczy IHP (Leibniz Institute for High Performance Microelectronics) upublicznił informacje na temat procesu BiCMOS SG13G2, który bazuje na ultraszybkich tranzystorach HBT pracujących w zakresie setek GHz. Założeniem IHP jest rozwój Open Source PDK i narzędzi EDA głównie pod kątem projektowania układów analogowych z silnym naciskiem na RF (Radio Frequency). Obecnie IHP oferuje zestaw standardowych bramek w technologii 130 nm CMOS (za pomocą których można syntetyzować obwody cyfrowe), wsparcie edytorów schematów (takich jak xschem i Qucs-S), modele SPICE urządzeń półprzewodnikowych kompatybilne z symulatorami ngspice i Xyce, przykłady układów urządzeń półprzewodnikowych w formacie GDSII, czy też eksperymentalne i stale rozwijane wsparcie projektowania układów za pomocą Klayout.

Kompletna propozycja zestawu narzedzi Open Source do projektowania układów analogowych z urzględnieniem pasma RF jest pokazana na **rysunku 1**.

Oprócz powyższych informacji, publiczne repozytorium IHP-Open-PDK na platformie GitHub zawiera również wyniki i warunki pomiarów urządzeń półprzewodnikowych, umożliwiając rozwój i weryfikację modeli przez społeczność Open Source Silicon.

Atrakcyjność i unikalność IHP open source PDK polega na zapewnieniu pełnego i darmowego zestawu narzędzi do tworzenia projektów z możliwością produkcji na linii pilotażowej IHP. Instytut oferuje usługę produkcji układów BiCMOS w swoich laboratoriach clean room w oparciu o własne kwalifikowane technologie od 2001 roku, a technologie BiCMOS IHP są jednymi z najbardziej wydajnych na świecie. IHP umożliwia produkcję niewielkich serii układów scalonych na waflach krzemowych o średnicy 200 mm oraz szerokie możliwości pomiarów i charakteryzacji struktur półprzewodnikowych, co jest szczególnie atrakcyjne z punktu widzenia partnerów akademickich oraz przemysłowych o niskich wymaganiach ilościowych. Instytut planuje uruchomić pierwsze darmowe programy MPW z zastosowaniem IHP open source PDK od grudnia 2024 roku, a szczegółowy opis rozwiązań oraz planów rozwijanych w IHP zostanie zamieszczony w kolejnym artykule.

dr Krzysztof Herman, dr Anna Sojka-Piotrowska
herman@ihp-microelectronics.com,
sojka@ihp-microelectronics.com
IHP GmbH – Leibniz Institute for High Performance Microelectronics, Frankfurt (Oder), Niemcy
https://www.ihp-microelectronics.com

Ważne linki:
- https://github.com/IHP-GmbH/IHP-Open-PDK
- https://fossi-foundation.org/
- Free Silicon Foundation https://wiki.f-si.org/
- https://efabless.com
- https://open-source-silicon.slack.com

Inżynieria Internetu Rzeczy: innowacyjne podejście do kształcenia inżynierów

Opracowanie oraz uruchomienie na Wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej programu studiów na kierunku Inżynieria Internetu Rzeczy, będące przedmiotem artykułu, okazało się pod wieloma względami innowacyjne i unikatowe. Unikatowość ta ma dwa wymiary. Pierwszy z nich dotyczy zakresu tematycznego kształcenia – w chwili uruchamiania studiów (2020 r.) w żadnej z polskich uczelni publicznych nie były jeszcze prowadzone studia dotyczące wspomnianej tematyki (wg wiedzy autorów jest to nadal przedsięwzięcie wyjątkowe w skali kraju). Drugi wymiar natomiast stanowi sama innowacyjna struktura programu studiów – wyróżnia ją realizowany w semestrach 1–6 moduł zespołowych zajęć projektowo-warsztatowych o znacznym wymiarze godzinowym.

Inżynieria Internetu Rzeczy jako nowy kierunek studiów

Podjęta w 2019 r. decyzja o przyjęciu Internetu Rzeczy jako obszaru tematycznego nowego kierunku studiów, na potrzeby którego opracowana i wdrożona zostanie innowacyjna koncepcja kształcenia (nowa struktura programu studiów), wynikała z:

- prognoz wskazujących, że Internet Rzeczy będzie się rozwijał w bardzo szybkim tempie,
- zakresu działalności naukowej oraz dydaktycznej prowadzonej na Wydziale Elektroniki i Technik Informacyjnych PW, obejmującej wszystkie podstawowe technologie warunkujące rozwój Internetu Rzeczy (IoT enabling technologies),
- analizy licznych opracowań wskazujących, że zasadniczą barierą ograniczającą postęp w zakresie wprowadzania do praktyki gospodarczej i życia społecznego rozwiązań korzystających z Internetu Rzeczy jest niedostatek kadr o odpowiednich kompetencjach.

W tym kontekście nowy program studiów, idealnie pasujący do profilu Wydziału (kształcącego inżynierów m.in. na kierunkach: elektronika, informatyka, telekomunikacja i cyberbezpieczeństwo), stanowiący próbę choćby częściowego złagodzenia problemów wynikających ze wspomnianego deficytu kadr, należy traktować jako odpowiedź na potrzeby i oczekiwania społeczne.

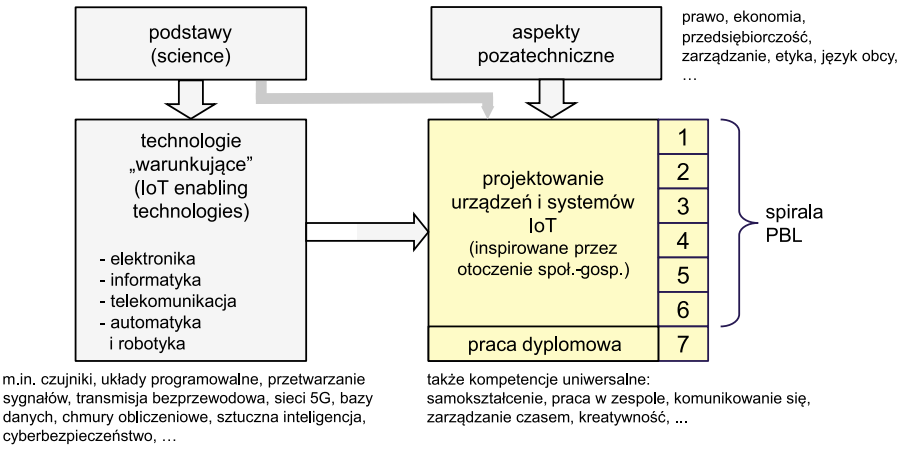
Choć w roku 2019 przyjęcie Inżynierii Internetu Rzeczy jako nazwy nowego kierunku stanowiło decyzję ryzykowną, z perspektywy czasu można stwierdzić, że była to decyzja trafna – prognozy szybkiego rozwoju IoT sprawdziły się. W roku 2023 na jednego mieszkańca naszego kraju przypadało ok. 7 podłączonych urządzeń IoT, a do roku 2028 liczba ta wzrośnie do ok. 16 [1].

Innowacyjna koncepcja kształcenia (struktura programu studiów)

Koncepcja kształcenia na kierunku Inżynieria Internetu Rzeczy odchodzi od tradycyjnego modelu kształcenia inżynierów, w którym program studiów pierwszego stopnia (studiów inżynierskich) rozpoczyna się od zestawu przedmiotów z zakresu nauk ścisłych (matematyka, fizyka, itp.), stanowiących „fundament" przedmiotów bezpośrednio związanych z kierunkiem studiów – podstawowych i bardziej specjalistycznych. Kluczowym elementem programu na pierwszym roku studiów staje się realizowany w kilkuosobowych zespołach projekt. Owo „nietradycyjne" podejście do organizacji studiów inżynierskich nie jest pomysłem ostatnich lat – realizowane było na niektórych renomowanych uczelniach badawczych,

np. Carnegie-Mellon University (USA), już we wczesnych latach 90. ubiegłego wieku. Przesłankami uzasadniającymi taką właśnie organizację kształcenia były i nadal są: zwiększenie zainteresowania studiami inżynierskimi oraz zmniejszenie „odsiewu" na pierwszym roku studiów, powodowanego trudnościami w przyswojeniu teorii bez powiązania jej z potencjalnymi obszarami zastosowań.

Przyjęta na studiach na kierunku Inżynieria Internetu Rzeczy koncepcja realizacji procesu dydaktycznego nie ogranicza się do zmian w początkowej fazie studiów – zakłada w ogólności odejście od kształcenia opartego na biernym uczestnictwie w zajęciach, narzucającym pozyskiwanie wiedzy teoretycznej i pasywne jej odtwarzanie na sprawdzianach, na rzecz kształcenia opartego na rozwiązywaniu problemów i realizacji projektów oraz innych formach zajęć aktywizujących studentów. Studenci pozyskują w ten sposób – obok wiedzy i umiejętności „technicznych", związanych z tematyką rozwiązywanych problemów – także bardzo cenione przez pracodawców kompetencje „miękkie" (umiejętność pracy w zespole, zarządzania projektem i planowania zadań, umiejętność komunikowania się, kreatywność, poczucie współodpowiedzialności itp.).



Rysunek 1. Koncepcja programu studiów pierwszego stopnia (inżynierskich) na kierunku Inżynieria Internetu Rzeczy

Koncepcja ta znajduje odzwierciedlenie w realizowanym na każdym semestrze studiów (z wyjątkiem ostatniego, przeznaczonego na pracę dyplomową) module zespołowych zajęć projektowych – PBL (project-based learning), o wymiarze najczęściej 12 punktów ECTS[2], prowadzonych w odpowiednio wyposażonych laboratoriach.

Program studiów
Cele i treści kształcenia

Celem programu jest wykształcenie wysokiej klasy specjalistów przygotowanych do tworzenia inteligentnych systemów i sieci, których podstawę funkcjonowania stanowi zastosowanie Internetu Rzeczy. Zakłada się, że absolwent będzie miał kompetencje (wiedzę, umiejętności praktyczne i kompetencje społeczne) umożliwiające tworzenie rozwiązań związanych z:

- wyposażaniem przedmiotów/urządzeń (stacjonarnych i mobilnych) w inteligentne sensory, często realizujące także wstępne przetwarzanie zbieranych danych, a także elektroniczne identyfikatory oraz elementy wykonawcze,
- tworzeniem infrastruktury sieciowej (używającej łączności przewodowej lub bezprzewodowej), która – poprzez Internet – zapewnia połączenie poszczególnych urządzeń,
- tworzeniem systemu (informatycznego) umożliwiającego gromadzenie danych zbieranych przez urządzenia oraz przetwarzanie tych danych – często z zastosowaniem metod sztucznej inteligencji,
- integrowaniem ww. elementów w sposób umożliwiający realizację rozmaitych inteligentnych produktów i usług, dostosowanych do potrzeb różnych grup użytkowników.

7-semestralny program studiów pierwszego stopnia nie może pokryć pełnego, bardzo rozległego spektrum kompetencji w zakresie inżynierii Internetu Rzeczy, określonych w literaturze i opiniach interesariuszy. W jednej z najciekawszych publikacji dotyczących kształcenia w zakresie IoT[3], zawierającej analizę kilkudziesięciu programów studiów w tym obszarze, stwierdzono wręcz: *żadna z analizowanych kilkudziesięciu ofert edukacyjnych nie pokrywa choćby „w przybliżeniu" zdefiniowanego w artykule zestawu pożądanych kompetencji* [inżyniera Internetu Rzeczy]. Do podobnych wniosków doprowadziła analiza treści programów dotyczących tej tematyki, oferowanych przez uczelnie i inne instytucje edukacyjne, przeprowadzona na Wydziale w 2022 roku, w kontekście opracowywania programu studiów drugiego stopnia na kierunku Inżynieria Internetu Rzeczy [4].

Niezbędna stała się więc selekcja treści i ich integracja w wewnętrznie spójny program. W procesie selekcji uwzględniono m.in.:

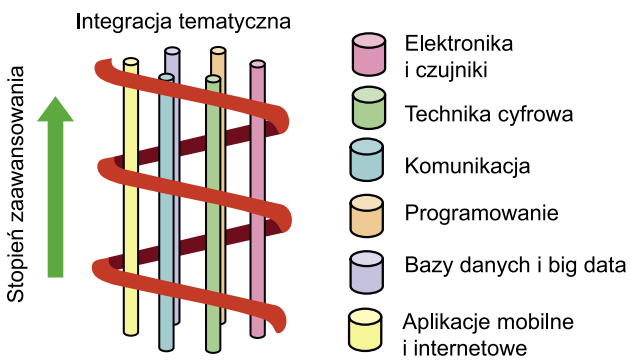
- wyniki analizy materiałów źródłowych dotyczących kształcenia w zakresie IoT (publikacje, podobne programy na innych uczelniach),
- oczekiwania i opinie pracodawców, wyrażone m.in. w badaniach ankietowych,
- konieczność zapewnienia równowagi ilości treści z zakresu różnych technologii warunkujących,
- zasoby Wydziału (kadra, infrastruktura dydaktyczna) i propozycje przyszłych realizatorów – zespołów/pracowników Wydziału.

Wybrane treści wkomponowano w przyjętą koncepcję programową, zakładającą stosowanie metod kształcenia opartego na rozwiązywaniu problemów i realizacji projektów oraz innych form prowadzenia zajęć aktywizujących studentów.

Stanowiącą wynik tych działań koncepcję programu zilustrowano na **rysunku 1**. Treści z zakresu różnych technologii warunkujących wraz z „klasycznymi" podstawami inżynierii z zakresu nauk ścisłych (matematyka, fizyka) oraz przedmiotami z zakresu nauk humanistyczno-społecznych tworzą podstawę do realizacji projektów. W każdym semestrze studiów (z wyjątkiem ostatniego) realizowany jest duży moduł zespołowych zajęć projektowych (moduł PBL).

Inne spojrzenie na program studiów na kierunku Inżynieria Internetu Rzeczy ilustruje **rysunek 2**. Termin „spirala PBL" oznacza, że na kolejnych semestrach (w ramach kolejnych modułów PBL) pogłębiane są kompetencje zdobyte na niższych semestrach w każdym z kluczowych dla systemów Internetu Rzeczy obszarów tematycznych.

W **tabeli 1** pokazano nominalny plan studiów, rozumiany jako przewidywany harmonogram realizacji programu studiów w poszczególnych semestrach cyklu kształcenia, z zaznaczeniem grup



Źródło: na postawie T. Kenyon, *PBL in Electronic Engineering*, 2002

Rysunek 2. Spirala PBL

Tabela 1. Plan studiów pierwszego stopnia (studiów inżynierskich) na kierunku Inżynieria Internetu Rzeczy

[KLASA] *przedmiot/moduł*	WCLP(Z)	ECTS	semestr						
			1	2	3	4	5	6	7
[WYCHOWANIE FIZYCZNE]		0	0	0	0				
[JĘZYKI OBCE]		12			4	4	4		
[PRZEDMIOTY EKONOMICZNO-SPOŁECZNE]		6							
Metodyczne aspekty działalności inżyniera	----(2)		2						
Przedsiębiorczość w praktyce	–2––					2			
PRZEDMIOTY EKONOMICZNO-SPOŁECZNE – przedmioty obieralne	–2––	2				2			
[MATEMATYKA]		15							
Matematyka 1 – Wstęp do matematyki	1–11		4[e]						
Matematyka 2 – Analiza	2111		6[e]						
Matematyka 3 – Algebra	12–1			5[e]					
[FIZYKA]		8							
Wstęp do fizyki	111–			4[e]					
Fizyczne podstawy elektroniki i teleinformatyki	21––				4[e]				
[PODSTAWY ELEKTRONIKI]		19							
Podstawy elektroniki i pomiarów 1	211–		4						
Sygnały i systemy	211–			5					
Elementy i układy elektroniczne	2–11				5				
PODSTAWY ELEKTRONIKI – przedmioty obieralne		5					5		
[INFORMATYKA TECHNICZNA]		23							
Podstawy programowania	2–2–		4						
Podstawy techniki cyfrowej	2–11			5[e]					
Mikrokontrolery i układy programowalne	2–2–					5			
Bazy danych i big data	2––2						5		
INFORMATYKA TECHNICZNA – przedmioty obieralne		4						4	
[TELEINFORMATYKA]		30							
Podstawy transmisji bezprzewodowej	2–2–				5[e]				
Architektura i inżynieria usług i aplikacji	2–11					5			
Podstawy cyberbezpieczeństwa	2––2						4[e]		
Metody analizy danych	212–							6	
Techniki chmur obliczeniowych	2–2–							5	
TELEINFORMATYKA – przedmioty obieralne		5							5
[PROJEKTOWANIE SYSTEMÓW I URZĄDZEŃ INTERNETU RZECZY]		69							
PBL1: Moduły i systemy Internetu Rzeczy	---2(7)		10[e]						
PBL2: Systemy wbudowane i oprogramowanie	---2(8)			11[e]					
PBL3: Komunikacja bezprzewodowa i przewodowa	---4(8)				12[e]				
PBL4: Inteligentne urządzenie Internetu Rzeczy	---4(8)					12[e]			
PBL5: Usługi i aplikacje Internetu Rzeczy	---4(8)						12[e]		
PBL6: Aplikacje rozproszone i chmury obliczeniowe Internetu Rzeczy	---4(8)							12[e]	
[PRZEDMIOTY OBIERALNE TECHNICZNE]		8							8
[DYPLOMOWANIE]		20							
PDI: Pracownia dyplomowa								3	
SDI: Seminarium dyplomowe									2
PPDI: Przygotowanie pracy dyplomowej									15
EPDI: Redakcja i edycja pracy dyplomowej									
suma		**210**	**30**	**30**	**30**	**30**	**30**	**30**	**30**
[PRAKTYKA]		4							

[e] przedmioty egzaminacyjne
w kolumnie WCLP(Z) podano tygodniowy wymiar (w godzinach) poszczególnych form zajęć: wykładów, ćwiczeń, zajęć laboratoryjnych, zajęć projektowych i zajęć zintegrowanych obejmujących różne formy prowadzenia zajęć

przedmiotów oraz przedmiotów podlegających wyborowi przez studenta. Można zauważyć, że liczba przedmiotów w semestrze została ograniczona do 5 lub 6, co ułatwiło dobrą synchronizację ich treści.

Prowadzenie zajęć oraz metody weryfikacji i oceny efektów uczenia się osiągniętych przez studenta

W zestawie przedmiotów i innych modułów zajęć tworzących program studiów, a w szczególności w ramach modułów PBL, używane są różnorodne formy prowadzenia zajęć, integrujące w przemyślany sposób rozmaite techniki kształcenia. Warto w tym kontekście zauważyć, że nastawienie na zdobywanie wiedzy poprzez samodzielne wyszukiwanie informacji oraz ich krytyczną analizę, rozwiązywanie problemów i projektowanie w naturalny sposób ogranicza wymiar wykładów; w przedmiotach obowiązkowych stanowią one jedynie 23,6% godzin zajęć – wyraźnie mniej niż w przypadku innych programów studiów prowadzonych na Wydziale i na uczelni.

Tym zróżnicowanym formom prowadzenia zajęć odpowiadają rozmaite metody weryfikacji i oceny efektów uczenia się, przy czym:

- ze względu na dominację form kształcenia aktywizujących studentów, sprawdzanie wiedzy odbywa się często pośrednio – przez weryfikowanie umiejętności zastosowania tej wiedzy (do rozwiązania problemu, realizacji projektu itp.),
- „tradycyjne" sposoby weryfikacji efektów uczenia się przybierają w niektórych przypadkach nietradycyjną formę; przykładowo, egzamin przewidziany jako jedna z form weryfikacji efektów uczenia się w niektórych modułach PBL ma w tym przypadku formę „obrony" (w pewnej mierze – publicznej) zrealizowanego projektu.

Wyróżniająca cecha programu – moduły PBL

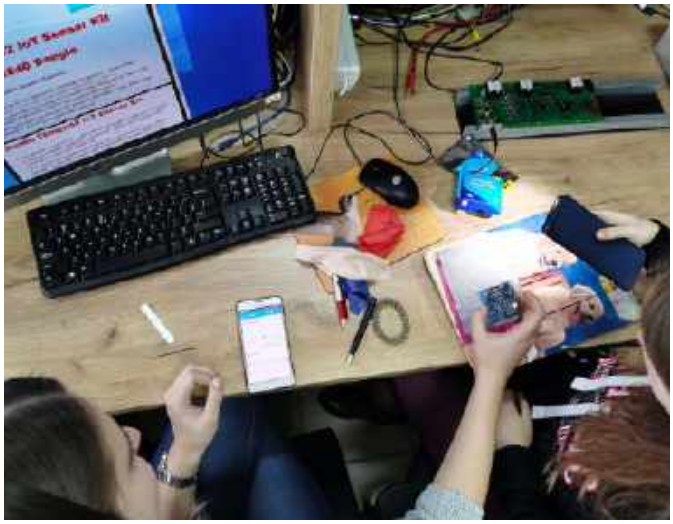
Realizacja modułów PBL odbywa się na początkowych semestrach zgodnie ze schematem CSI Google Lab Double Diamond, opartym na koncepcji Design Thinking. Schemat ten obejmuje trzy etapy:

- budowanie zespołu,
- definiowanie zadania (pierwszy komponent Double Diamond),
- projektowanie i implementacja (drugi komponent Double Diamond).

Moduły PLB są realizowane z użyciem dwóch podstawowych form zajęć (w przypadku obu tych form studenci pracują w zespołach kilkuosobowych):

- zajęcia zintegrowane – warsztaty tematyczne, odbywające się zwykle w formule dwóch 4-godzinnych spotkań w każdym tygodniu zajęć; w ramach każdego warsztatu (tematu) studenci realizują działania prowadzące do efektu, który jest mierzalny i podlega ocenie cząstkowej,
- zajęcia projektowe – tematy projektów są często inspirowane zagadnieniami zgłaszanymi przez otoczenie społeczno-gospodarcze; formuła regularnych spotkań kontrolnych jest dwojakiego rodzaju: raportowanie postępów na forum grupy studenckiej i konsultacje zespołu z opiekunem (opiekun sprawuje nadzór nad co najwyżej 3 zespołami); raport prezentujący wyniki projektu wraz z wnioskami i podsumowaniem jest podstawą do wystawienia oceny końcowej.

Sposób realizacji modułów PBL zilustrowany jest przykładem prowadzonego w semestrze 2 modułu PBL2 „Systemy Wbudowane i Oprogramowanie". Tematyka zajęć dotyczy niskopoziomowego programowania w językach C, Python i MicroPython oraz komunikacji z czujnikami, tak aby zapewnić pełną kontrolę pracy czujników w czasie rzeczywistym z wchodzeniem procesora w stan uśpienia po zakończeniu pomiarów. Zespół prowadzący zajęcia współpracuje z wiodącymi firmami w branży elektronicznej, takimi jak Nordic Semiconductor, ST Microelectronics, Infineon, Keysight i Kamami. Większość nowoczesnego sprzętu w studenckim laboratorium sprzętowym pochodzi z darowizn od tych firm. Duże znaczenie ma przy tym fakt, że studenci mogą pracować na sprzęcie z aktualnej, najnowszej oferty producentów. Daje to słuszne poczucie stosowania wiodących rozwiązań technicznych. Wymienione firmy są zapraszane



Fotografia 1. Zajęcia projektowe w ramach PBL2

do prezentacji swoich produktów i prowadzenia warsztatów, a zaproszenia te są traktowane przez nie jako swoiste wyróżnienie.

Projekt jest prowadzony w małych, najczęściej 4-osobowych zespołach studenckich. Każdy zespół ma własnego tutora, co umożliwia bieżące kontakty i bezpośrednie wspieranie studentów w realizowanych przez nich zadaniach projektowych. Typowe zajęcia projektowe ilustruje **fotografia 1**. W celu ułatwienia studentom realizacji zadań warsztatowych i projektowych zespół prowadzący przedmiot opublikował podręcznik akademicki dotyczący zagadnień będących przedmiotem zajęć [5].

W semestrze letnim roku akademickiego 2023/2024 w ramach przedmiotu PBL2 studenci realizują projekt „Stacja monitorowania jakości powietrza". Projekt jest inspirowany problemami wentylacji w pomieszczeniach laboratorium sprzętowego. Projektowane rozwiązanie składa się z dwóch modułów: monitorowania w pomieszczeniu oraz monitorowania na zewnątrz. Jako platforma sprzętowa używane są płytki Raspberry Pi 4 i Raspberry Pi Pico W. Do monitorowania jakości powietrza w pomieszczeniu używane są głównie czujniki Sensirion SEN55 oraz Bosch BME688, natomiast na zewnątrz używany jest głównie zestaw Weather Meter Kit z płytką Enviro Weather firmy Pimoroni.

Prowadzony na kolejnym, trzecim semestrze moduł PBL3 „Komunikacja przewodowa i bezprzewodowa" został szczegółowo opisany w artykule zamieszczonym w czasopiśmie „Elektronika: konstrukcje, technologie, zastosowania" [6].

Zajęcia prowadzone w ramach modułów PBL są wspomagane przez przedmioty techniczne z zakresu elektroniki, informatyki, telekomunikacji (w mniejszym stopniu – automatyki i robotyki), a także przedmioty rozwijające zrozumienie metodycznych i pozatechnicznych aspektów pracy inżyniera oraz służące kształtowaniu kompetencji

„miękkich". Taką funkcję pełni m.in. realizowany w pierwszej połowie 1 semestru przedmiot „Metodyczne aspekty działalności inżyniera", zaprojektowany z intencją wyposażenia studentów w istotne kompetencje uniwersalne, przydatne w dalszym kształceniu.

Szczegółowe informacje o programie można znaleźć na stronie https://iot.pw.edu.pl. Znajduje się na niej m.in. materiał „Inżynieria Internetu Rzeczy: uczysz się, realizując projekty, czyli krótki przewodnik po programie studiów", a także materiały multimedialne, które zostały użyte do promocji nowego programu studiów podczas Drzwi Otwartych PW.

Kontynuacja kształcenia – studia drugiego stopnia

Pierwsi absolwenci studiów pierwszego stopnia (studiów inżynierskich) na kierunku Inżynieria Internetu Rzeczy ukończyli ten etap nauki na początku 2024 roku. Z myślą o ich potrzebach, począwszy od semestru letniego roku akademickiego 2023/2024, realizowane są studia drugiego stopnia (studia magisterskie) na kierunku o tej samej nazwie. Mają one charakter „otwarty" – są skierowane także do absolwentów innych kierunków na Wydziale, na innych wydziałach PW i na innych uczelniach. W istocie, z pierwszej grupy osób przyjętych na wspomniane studia (w lutym 2024 r.), jedynie 38,9% stanowili absolwenci studiów pierwszego stopnia na tym kierunku.

W **tabeli 2** zaprezentowano nominalny plan studiów drugiego stopnia na kierunku Inżynieria Internetu Rzeczy.

Studia te, czerpiąc z pozytywnych doświadczeń związanych z opracowaniem i realizacją studiów pierwszego stopnia, a w szczególności modułów PBL, wzbogacono o elementy (treści i formy realizacji zajęć) wynikające ze specyfiki studiów magisterskich (zorientowanie na rozwiązywanie problemów o charakterze badawczym). Jednocześnie stały się one „poligonem doświadczalnym" nowych rozwiązań, takich jak:

- eksperymentalna próba wdrożenia idei kształcenia przez rozwiązywanie problemów o wysoce interdyscyplinarnym charakterze poprzez współdziałanie studentów kierunku Inżynieria Internetu Rzeczy ze studentami z innych wydziałów, kształcących się na kierunkach niezwiązanych z dyscyplinami reprezentowanymi na Wydziale Elektroniki i Technik Informacyjnych; w pierwszej edycji programu realizowana będzie współpraca ze studentami Wydziału Architektury, w szczególności w zakresie architektury informacyjnej (computational design) oraz projektowania uniwersalnego, realizującego koncepcję społeczeństwa włączającego, odpowiadającego potrzebom różnych grup społecznych (osób starszych, osób z niepełnosprawnościami itp.), a także współpraca ze studentami Wydziału Geodezji i Kartografii, w kontekście realizacji przez ten wydział inicjatyw związanych z projektami dotyczącymi inteligentnych miast (Smart City);
- ewolucyjne przechodzenie do realizacji części przedmiotów (wybranych form zajęć) z zastosowaniem metod i technik kształcenia na odległość, tak aby ułatwić studentom pogodzenie obowiązków związanych z kształceniem na studiach drugiego stopnia z powszechnie podejmowaną, zwłaszcza przez studentów kierunków szczególnie pożądanych na rynku pracy, pracą zawodową.

Inne warte odnotowania przedmioty, związane m.in. z charakterem studiów drugiego stopnia, to np.:

Tabela 2. Plan studiów drugiego stopnia (studiów magisterskich) na kierunku Inżynieria Internetu Rzeczy

[KLASA] *przedmiot/moduł*	WCLP(Z)	ECTS	semestr		
			1	2	3
[PRZEDMIOTY HUMANISTYCZNO-SPOŁECZNE]		5			
Przedsiębiorczość startupowa	1*– – 1*(1*) (10+16+14)	3	3		
Zarządzanie ryzykiem operacyjnym	1*1*1*– (10+10+10)	2		2	
[INŻYNIERIA INTERNETU RZECZY – PODSTAWY]		7			
Stosowana probabilistyka	2 – – – (1)	3	3[e]		
Uczenie maszynowe	2 – – 2	4		4[e]	
[INŻYNIERIA INTERNETU RZECZY]		10			
Sieci inteligentnych urządzeń	2 – – 2	4	4		
Kosynteza sprzętowo-programowa	2 – 1 1	4	4		
Internet Rzeczy: nauka i praktyka	1*1*– – (10+20)	2	2		
[INŻYNIERIA INTERNETU RZECZY – MODUŁY PBL]		24			
Bezpieczeństwo komunikacji bezprzewodowej	– – – 4 (8)	12	12[e]		
Inteligentne otoczenie	– – – 4 (8)	12		12[e]	
PRZEDMIOTY OBIERALNE: INŻYNIERIA INTERNETU RZECZY/TELEINFORMATYKA		8		4	4
PRZEDMIOTY OBIERALNE TECHNICZNE		4			4
[PROWADZENIE BADAŃ I DYPLOMOWANIE]		32			
Methodological and ethical issues of technoscientific research	1*1*– – (20+10)	2		2	
Pracownia problemowa		2	2		
Pracownia dyplomowa		6		6	
Seminarium dyplomowe	– 2 – –	2			2
Przygotowanie pracy dyplomowej		20			20
Redakcja i edycja pracy dyplomowej		0			0
suma		**90**	**30**	**30**	**30**

* wymiar przybliżony (łączna liczba godzin zajęć rozłożona na poszczególne formy zajęć w sposób nierównomierny)
[e] przedmioty egzaminacyjne

- specjalny przedmiot poświęcony metodologicznym podstawom prowadzenia badań naukowych, prowadzony w języku angielskim,
- specjalny przedmiot, zaplanowany wzorem programów studiów drugiego stopnia w uczelniach zagranicznych, o niesprecyzowanej a priori zawartości merytorycznej, stanowiący forum dyskusji z zaproszonymi gośćmi, reprezentującymi środowisko naukowe i otoczenie społeczno-gospodarcze uczelni, prezentującymi nowe osiągnięcia nauki i techniki związane z inżynierią Internetu Rzeczy oraz doświadczenia zdobyte w ramach przedsięwzięć biznesowych w tym obszarze,
- zestaw przedmiotów humanistyczno-społecznych ukierunkowanych na kreowanie postaw przedsiębiorczych i rozwijanie umiejętności zarządzania projektami, z uwzględnieniem ryzyka z tym związanego; taki zestaw stwarza absolwentom atrakcyjną możliwość zastosowania kompetencji zdobytych w trakcie studiów w celu utworzenia start-upu w oparciu o wyniki uzyskane w ramach jednego z dużych projektów zespołowych lub pracy dyplomowej, traktowanej jako preinkubator planowanego przedsięwzięcia o charakterze komercyjnym,
- ograniczenie liczby przedmiotów w poszczególnych semestrach i w całym cyklu studiów (priorytet ma głębokość i stopień zaawansowania zdobywanej wiedzy; projekt pojawia się w większości przedmiotów); ogólna liczba przedmiotów w 3-semestralnym cyklu kształcenia została ograniczona do 13 (liczba ta nie uwzględnia przedmiotów związanych z procesem dyplomowania), co ułatwiło dobrą synchronizację ich treści,
- zorientowanie na kształtowanie praktycznych umiejętności prowadzenia badań i rozwiązywanie problemów; w przedmiotach obowiązkowych (bez dyplomowania) 72,8% punktów ECTS jest związanych z zajęciami o charakterze praktycznym, głównie projektami,
- ograniczenie tradycyjnych metod weryfikacji osiąganych przez studentów efektów uczenia się na rzecz metod typowych dla kształcenia opartego na badaniach i realizacji projektów i – będąca pochodną tego podejścia – mała liczba egzaminów (jedynie 4 egzaminy z przedmiotów obowiązkowych programu),
- znaczna elastyczność, związana z wyborem przedmiotów oraz definiowaniem tematyki projektów.

Ocena i wnioski

Zaprezentowane w tym punkcie rozważania dotyczą studiów pierwszego stopnia, nie sposób bowiem ocenić studiów drugiego stopnia kilka miesięcy po ich uruchomieniu.

Studia cieszą się dużym zainteresowaniem kandydatów. Przykładowo, w roku 2020 zainteresowanie studiowaniem na nowym kierunku wykazało 928 kandydatów. Biorąc pod uwagę limit przyjęć (30 miejsc), oznacza to, że o jedno miejsce ubiegało się ponad 30 kandydatów. Z danych opublikowanych przez ówczesne Ministerstwo Nauki i Szkolnictwa Wyższego wynika, że były to wówczas studia najbardziej „oblegane" spośród wszystkich prowadzonych przez polskie uczelnie [7]. Zadziałał w tym przypadku zapewne „efekt nowości" i związany z tym brak informacji o progu przyjęć – liczbie punktów z egzaminu maturalnego zapewniającej przyjęcie w poprzedniej rekrutacji (dostępna w kolejnych latach informacja o wysokim progu zniechęciła prawdopodobnie pewną grupę kandydatów) oraz dość intensywna kampania informacyjno-promocyjna (specjalna strona www, aktywność podczas Drzwi Otwartych PW). Warto w tym kontekście wspomnieć, że obecnie istotną rolę w promocji studiów wśród potencjalnych kandydatów odgrywają członkowie studenckiego koła naukowego KOIoT i jego opiekunowie. Jedną z form tej promocji są zajęcia warsztatowe skierowane do uczniów ósmych klas szkół podstawowych i uczniów szkół średnich.

Ciekawych informacji dotyczących profilu studiujących dostarczają wyniki ankiet przeprowadzonych w połowie pierwszego semestru wśród studentów kolejnych roczników rozpoczynających kształcenie. Szczególnie interesujące są odpowiedzi na pytanie dotyczące innych uczelni, na które studenci zostali przyjęci, ale zrezygnowali z nich na korzyść Inżynierii Internetu Rzeczy. W tej dość licznej grupie (36% studiujących), oprócz dość oczywistych i oczekiwanych odpowiedzi (informatyka, automatyka i robotyka oraz inne kierunki prowadzone na uczelniach technicznych, informatyka i kierunki z dziedziny nauk ścisłych na uniwersytetach), zdarzają się odpowiedzi zaskakujące. Wymieniono m.in.: kierunek lekarski na WUM, ekonomię na UW oraz w SGH, prawo na UW, kognitywistykę na UW, Artes Liberales na UW, MISH na UW i UAM.

Wspomniane ankiety zawierają wysoce pozytywną ocenę programu studiów na kierunku Inżynieria Internetu Rzeczy. Ponad połowa (55%) studentów tego kierunku oceniła, że realizowane przez nich studia wyróżniają się pozytywnie spośród innych prowadzonych na Wydziale (nie było ocen negatywnych ani neutralnych; pozostali studenci uznali, że nie mają dostatecznej wiedzy, aby dokonać oceny) [8].

Równie pozytywne są opinie wyrażane przez studentów wyższych lat studiów; można je znaleźć na stronie www https://iot.pw.edu.pl.

Nie oznacza to braku możliwości doskonalenia programu i metod jego realizacji. Możliwości te dotyczą m.in.:

- szerszego współdziałania z „zewnętrzem akademickim" – wyjścia poza „teren" Wydziału i nawiązania współpracy z innymi wydziałami PW, a potencjalnie także z innymi środowiskami akademickimi: uczelniami i instytutami badawczymi;
- szerszego współdziałania ze sferą biznesu i administracji;
- zaadaptowania korzyści wynikających z możliwości realizacji kształcenia na odległość i kształcenia odwróconego [9], tak aby umożliwić studiującym pogodzenie kształcenia z obowiązkami wynikającymi z pracy zawodowej [10];
- wprowadzanie do procesu dydaktycznego metod i narzędzi sztucznej inteligencji, w szczególności – generatywnej sztucznej inteligencji; warto przy tym kontekście zwrócić uwagę, że przyjęty w modułach PBL model weryfikacji efektów uczenia się, w którym istotną rolę odgrywają prototypy fizycznych obiektów zawierających istotny komponent sprzętowy, tworzy warunki do stosowania liberalnych zasad korzystania przez studentów z narzędzi typu chatGPT – trudno oczekiwać bowiem, że narzędzia te wykonają „za studenta" projekt i prototyp systemu zawierającego rozwiązania z zakresu Internetu Rzeczy;
- nadanie kształceniu bardziej międzynarodowego charakteru, m.in. przez wprowadzanie do oferty większej liczby zajęć prowadzonych w języku angielskim, a także zachęcanie studentów do korzystania z oferty uczelni zagranicznych i innych instytucji oferujących usługi edukacyjne;
- oferowanie mikroprogramów prowadzących do uzyskania mikropoświadczeń [11], związanych m.in. z wybranymi obszarami zastosowań Internetu Rzeczy.

Zakończenie

Program studiów na kierunku Inżynieria Internetu Rzeczy jest w istotny sposób różny od pozostałych programów studiów oferowanych przez PW, a także inne polskie uczelnie. Jego unikatowość (innowacyjność) ma dwa wymiary:

- zakres tematyczny, cele i efekty uczenia się – według wiedzy autorów w żadnej z polskich uczelni publicznych nie są jeszcze prowadzone studia pierwszego stopnia dotyczące tej tematyki,
- koncepcja realizacji (innowacyjna struktura) programu studiów, którą wyróżnia realizowany na semestrach 1–6 moduł zespołowych zajęć projektowo-warsztatowych o znacznym wymiarze, prowadzonych zgodnie z koncepcją PBL – według wiedzy autorów, taki model kształcenia, stanowiący innowacyjne przedsięwzięcie w okresie tworzenia programu, nadal nie jest jeszcze realizowany w żadnej innej z polskich uczelni technicznych.

Wcześniej przytoczone fakty i opinie uzasadniają tezę, że opracowanie i realizacja tego programu stanowi udany eksperyment w sferze kształcenia, który powinien być kontynuowany i rozpowszechniony.

Inicjatywa uruchomienia studiów na kierunku Inżynieria Internetu Rzeczy stanowi ponadto istotny – także z punktu widzenia wizerunkowego – krok w kierunku unowocześnienia i uatrakcyjnienia oferty kształcenia na Politechnice Warszawskiej, a zwłaszcza poprawy stopnia jej dopasowania do potrzeb nowoczesnej gospodarki i społeczeństwa korzystającego w coraz większym stopniu z dobrodziejstw, jakie niosą rozliczne zastosowania Internetu Rzeczy.

Andrzej Kraśniewski,
Instytut Telekomunikacji, Politechnika Warszawska
Henryk A. Kowalski,
Instytut Informatyki, Politechnika Warszawska

Materiały źródłowe:

- *Rynek Internetu Rzeczy w Polsce. Analiza rynku i prognozy rozwoju na lata 2023-2028*, PMR Market Experts, https://tiny.pl/dw1tl [dostęp: 4.04.2024]
- E. Mäkiö-Marusik, „Current trends in teaching cyber physical systems engineering: A literature review”, *2017 IEEE 15th International Conference on Industrial Informatics*, Emden, Germany, 2017, pp. 518-525, doi: 10.1109/INDIN.2017.8104826
- W. B. Daszczuk, K. Gracki, H. Kowalski, G. Mazur, A. Skorupski, Z. Szymański, *Inżynieria systemów Internetu Rzeczy. Sprzęt i oprogramowanie*, Oficyna Wydawnicza Politechniki Warszawskiej, 2021, ISBN: 978-83-8156-269-0
- P. Korpas, „Wnioski z zastosowania metody nauczania projektowego na przedmiocie Komunikacja przewodowa i bezprzewodowa”, „Elektronika: konstrukcje, technologie, zastosowania”, 2022, vol. 63, nr 12, str. 17–21, DOI: 10.15199/13.2022.12

Przypisy:

[1] Rynek internetu rzeczy w Polsce. Analiza rynku i prognozy rozwoju na lata 2023–2028, PMR Market Experts, https://tiny.pl/dw1tl

[2] Nakład pracy studenta związany z tym jednym przedmiotem stanowi zatem, najczęściej 40% semestralnego nakładu pracy studenta związanego z realizacją programu studiów.

[3] E. Mäkiö-Marusik, „Current trends in teaching cyber physical systems engineering: A literature review”, 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), Emden, Germany, 2017, pp. 518–525, doi: 10.1109/INDIN.2017.8104826

[4] Przykładowo, na platformie Coursera (https://tiny.pl/dw1tp [dostęp 12.03.2023]) można znaleźć 170 programów mających w nazwie ‘IoT’, oferowanych głównie przez renomowane uczelnie amerykańskie i azjatyckie, a także komercyjnych dostawców usług edukacyjnych

[5] W. B. Daszczuk, K. Gracki, H. Kowalski, G. Mazur, A. Skorupski, Z. Szymański, *Inżynieria systemów internetu rzeczy. Sprzęt i oprogramowanie*, Oficyna Wydawnicza Politechniki Warszawskiej, 2021, ISBN: 978-83-8156-269-0

[6] P. Korpas, „Wnioski z zastosowania metody nauczania projektowego na przedmiocie Komunikacja przewodowa i bezprzewodowa”, „Elektronika: konstrukcje, technologie, zastosowania”, 2022, vol. 63, nr 12, str. 17-21, DOI: 10.15199/13.2022.12

[7] https://tiny.pl/dw1tj

[8] Ocenę tę należy rozpatrywać w kontekście faktu, że studia na wszystkich pozostałych kierunkach prowadzonych na Wydziale w rankingu kierunków technicznych miesięcznika „Perspektywy” od lat zajmują miejsca na podium, w większości przypadków na jego najwyższym stopniu

[9] Elementy kształcenia odwróconego, polegającego na przeniesieniu „do domu” procesów służących zdobywaniu przez studenta wiedzy (realizowanych tradycyjne na uczelni w formie wykładu) i przeznaczeniu zajęć na uczelni przede wszystkim na realizację – przy odpowiednim wsparciu ze strony kadry akademickiej – zajęć o charakterze warsztatowym i projektowym, są istotnym komponentem realizacji modułów PBL. Studenci kierunku IIR są więc przyzwyczajeni do tej formy prowadzenia procesu kształcenia.

[10] Badania pokazują, że większość studentów wyższych lat studiów pierwszego stopnia i podejmujących studia drugiego stopnia pracuje, często na pełnym etacie.

[11] Możliwości takie stwarza Zarządzenie Rektora PW z dnia 6 listopada 2023 r. w sprawie uzyskiwania mikropoświadczeń w wyniku realizacji mikroprogramów w Politechnice Warszawskiej

Nowa wersja oprogramowania: Arm Keil MDK v6

Arm – jako lider technologii i architektury procesorów mających zastosowanie między innymi w smartfonach, tabletach, urządzeniach biurowych, elektronice konsumenckiej i profesjonalnej – oferuje swoim użytkownikom narzędzia programistyczne polecane do konkretnych typów urządzeń.
Od wielu lat Arm dostarcza środowisko Keil Microcontroller Development Kit (MDK), stanowiące kompleksowe narzędzie do tworzenia oprogramowania na potrzeby aplikacji wbudowanych, bazujących na Arm Cortex-M. Oprogramowanie to stało się standardem w obsłudze ponad 10000 modeli mikrokontrolerów pochodzących od 45 dostawców układów scalonych i okazuje się nieodzownym elementem wielu projektów.

W związku z rosnącym zapotrzebowaniem na większe możliwości uczenia maszynowego (ML) w aplikacjach IoT, Arm wypuszcza nową wersję Keil MDK 6, która została zoptymalizowana do wsparcia całego portfolio procesorów Arm Cortex-M. Arm dostosował swój plan działania do nowych potrzeb, wprowadzając mikrokontrolery (MCU) Cortex-M55 oraz Cortex-M85. Są one zaprojektowane z uwzględnieniem wysokiej wydajności, wymaganej w aplikacjach ML i DSP. Jednocześnie urządzenia IoT stają się coraz bardziej inteligentne, co sprawia, że programowanie staje się bardziej złożone i wymaga unowocześniania również metod oraz rozwiązań deweloperskich.

Oprócz nowo wprowadzonej obsługi hostów systemów Windows, Linux i macOS, wersja 6 pakietu MDK nadal zawiera sprawdzone środowisko µVision przystosowane do systemu Windows oraz narzędzia zapewniające bezpieczeństwo funkcjonalne. Systemy wbudowane wymagają zazwyczaj wieloletniego wsparcia produktowego, a MDK zapewnia kompleksową obsługę przez cały cykl życia produktu – od momentu jego powstania, przez okres eksploatacji, aż po finalne etapy zakończenia i utrzymania.

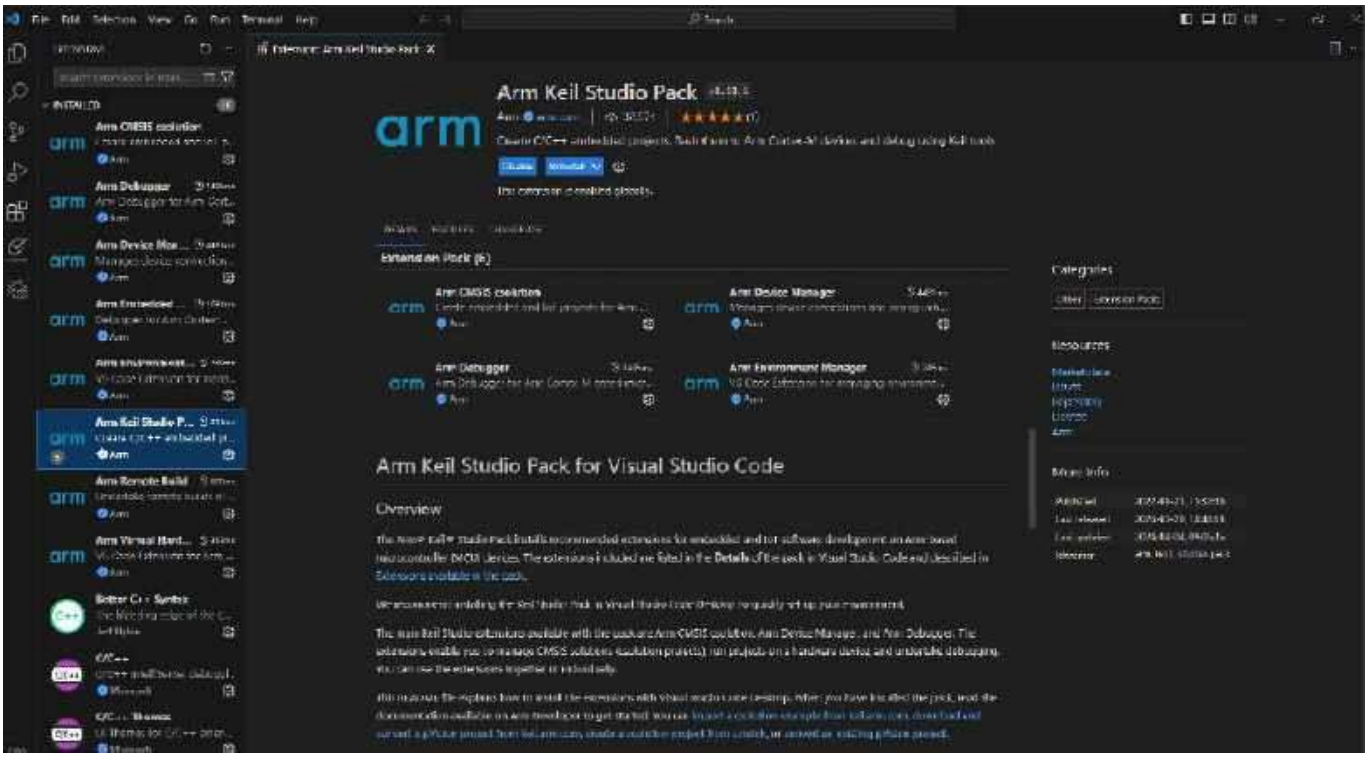
Główne cechy MDK – środowisko µVision

MDK-Core opiera się na środowisku µVision IDE, oferującym zaawansowane wsparcie urządzeń Cortex-M oraz architektury ARMv8-M. Szczegółowe informacje o obsługiwanych mikrokontrolerach można znaleźć na stronie: https://keil.arm.com/devices/.

Różnorodne wersje rdzeni ARM znajdują zastosowanie w systemach wbudowanych oraz urządzeniach o niskim zużyciu energii dzięki ich energooszczędnej konstrukcji. Arm najlepiej zna budowę swoich rdzeni, dlatego jego własne oprogramowanie MDK wyróżnia się najwyższym stopniem zaawansowania i optymalnym dostosowaniem do nowych rozwiązań dostarczanych przez tego producenta.

Program Pack Installer stanowi integralną część środowiska i służy do pobierania, instalowania oraz zarządzania pakietami oprogramowania. Można dzięki niemu aktualizować lub usuwać biblioteki, a tym samym komponenty oprogramowania µVision. Pakiety programowe, znane jako Software Packs, ułatwiają zarządzanie układami i całymi płytkami, a także bibliotekami.

Debugger µVision oferuje zintegrowane środowisko, pozwalające na kompleksowe testowanie, weryfikację oraz optymalizację kodu aplikacji. Zawiera proste i złożone punkty przerwań, okna zegarów i kontrolę wykonania programu. Daje możliwość podglądu stanów urządzeń peryferyjnych.



Rysunek 1. Widok okna środowiska Keil Studio

Rysunek 2. Struktura ekosystemu Keil MDK v6

Pakiet µVision, dzięki regularnym aktualizacjom bibliotek oraz dostępności gotowych projektów przykładowych, znacząco wspomaga i przyspiesza rozwój oprogramowania wbudowanego. Producent, wyznaczając nowe standardy, dostarcza także pełną i profesjonalną dokumentację techniczną.

Keil MDK w wersji 6 wychodzi naprzeciw potrzebom programistów, rozszerzając funkcjonalność środowiska o nowe rozwiązania:

- **Keil Studio**, tworzące nową platformę programistyczną do mikrokontrolerów bazujących na Cortex-M, oparte o edytor Visual Studio Code firmy Microsoft. Prezentowane rozwiązanie oferuje kompleksowe środowisko IDE przeznaczone do mikrokontrolerów Cortex-M i zapewnia powtarzalne kompilacje. Zawiera wsparcie przepływów pracy CMSIS i zintegrowany debugger, co umożliwia tworzenie, kompilowanie, a także testowanie aplikacji wbudowanych na różnych systemach operacyjnych, takich jak Windows, Linux i macOS. Integracja z Git oraz dostępność rozszerzeń innych firm czynią MDK w wersji 6 elastyczną platformą, idealną z punktu widzenia projektów IoT czy ML.
- **CMSIS-Toolbox** w Keil MDK wersji 6 stanowi kluczowy element procesu programowania opartego na CMSIS. Proces ten rozpoczyna się od wyboru odpowiedniego urządzenia lub płytki, co pozwala na konfigurację kompleksowego zestawu narzędzi, w tym debuggera. Użytkownik uzyskuje dostęp do szerokiej gamy komponentów oprogramowania, których można ponownie użyć, w tym jąder RTOS, sterowników urządzeń oraz oprogramowania pośredniego (Middleware). CMSIS oferuje również biblioteki obliczeniowe i uczenia maszynowego, dostosowane do portfolio procesorów Cortex-M. Nowo wprowadzony komponent CMSIS-View umożliwia weryfikację oprogramowania na podstawie zdarzeń oraz analizę czasu wykonywania kodu, a to okazuje się nieocenione przy doborze najodpowiedniejszych modeli ML do konkretnej aplikacji.
- Udoskonalona integracja z **Arm Virtual Hardware (AVH)** pozwala na rezygnację z programowania na fizycznym sprzęcie, dzięki wirtualizacji kompletnego podsystemu SoC opartego na procesorach Arm. To z kolei otwiera drogę do automatyzacji testów obciążeń oprogramowania przy użyciu dokładnych modeli symulacyjnych Cortex-M. Keil MDK wspomaga tworzenie i weryfikację przypadków testowych, zarówno w środowiskach stacjonarnych, jak i chmurowych, co pozwala programistom na zastosowanie metodologii CI/CD, DevOps i MLOps. AVH jest dostępny w różnych implementacjach, w tym na GitHub, Qeexo AutoML, Keil Studio Cloud i AWS AMI, oferując elastyczny dostęp do zasobów chmurowych.
- Dzięki dodaniu rozwiązania **FuSa RTS i biblioteki FuSa C do MDK-Professional** programiści mogą tworzyć oprogramowanie do systemów bezpieczeństwa funkcjonalnego w aplikacjach o wyższych potrzebach w tym zakresie. Wstępnie certyfikowane biblioteki z obszerną dokumentacją i praktycznymi materiałami pomogą osiągnąć cel w krótszym czasie.

Podsumowanie

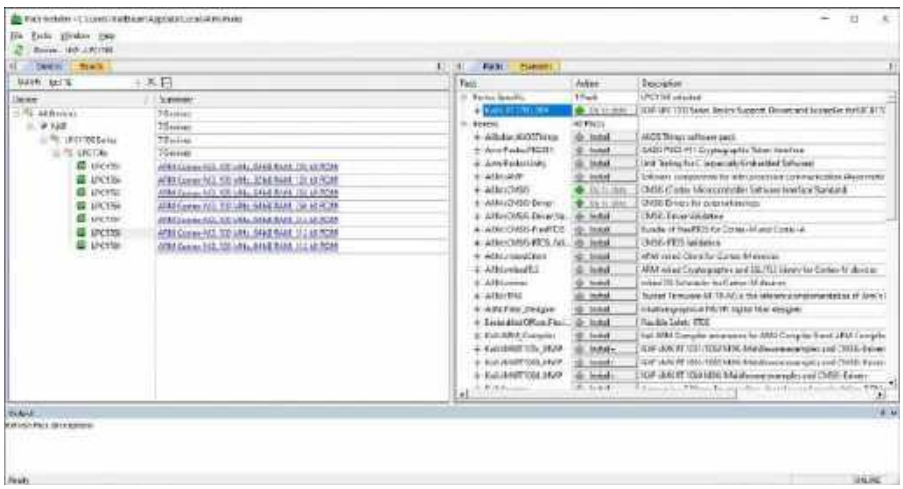
Keil MDK, stworzone przez ARM, to zaawansowane oprogramowanie wyposażone w narzędzia do obsługi szerokiej gamy popularnych rdzeni mikrokontrolerów. Jego ciągłe aktualizacje i wsparcie bibliotek µVision IDE ustanawiają nowe standardy na rynku, wyprzedzając konkurencyjne produkty, a także napędzając innowacje. Konkurując z bezpłatnymi kompilatorami Keil MDK wspiera programistów, ponieważ oferuje im funkcje ułatwiające analizę błędów, monitorowanie stanów układów peryferyjnych oraz optymalizację wydajności. Ważnym elementem jest także wsparcie oferowane przez producenta.

Arm i jego partnerzy ekosystemowi podejmują wyzwanie związane z rosnącą złożonością oprogramowania, dostarczając programistom z branży systemów wbudowanych i IoT spójne przepływy pracy. Dzięki temu, że działają one w różnorodnych zestawach narzędziowych i z ustandaryzowanymi komponentami oprogramowania, procesory ARM są doskonale przystosowane do zastosowań wbudowanych. Keil MDK w wersji 6 jest kluczowym kamieniem milowym w dążeniu do kształtowania przyszłości IoT i AI oraz ich rozwoju na platformie ARM.

Najnowsze oprogramowanie jest już dostępne do testowania w wersji próbnej na stronie producenta: https://www.keil.arm.com/.

W najbliższym kwartale planowana jest także możliwość zakupu MDK 6 do zastosowań komercyjnych.

Grzegorz Cuber
Technical Manager
Computer Controls Sp. z o.o.
www.ccontrols.pl



Rysunek 3. Widok okna Pack Installer

Druk 3D w służbie elektroniki (1)

Druk 3D znany jest w przemyśle od kilku dekad. Jednak pojawienie się ponad dekadę temu projektu RepRap sprawiło, iż ta technologia nagle stała się dostępna dla każdego, a jej cena spadła z setek tysięcy do kilkuset dolarów za urządzenie. Drukarki oferują unikalne możliwości wytwórcze w niskiej cenie, co okazuje się przydatne zarówno w produkcji, jak i w projektowaniu prototypów.

Marketingowcy reklamują drukarki 3D jako narzędzia do łatwej, domowej produkcji przedmiotów użytkowych – w roztaczanych przez nich wizjach drukarka 3D działa niczym replikator z filmów i seriali z uniwersum Star Treka. W praktyce aż tak różowo nie jest, i choć faktycznie da się wytwarzać przedmioty użytkowe oraz różne narzędzia, to sama technologia druku 3D wymaga pewnej wiedzy i nieco innego sposobu myślenia niż przy bardziej tradycyjnych metodach produkcyjnych. Jednocześnie możliwości (relatywnie) szybkiego prototypowania, wykonywania unikalnych, jednorazowych elementów i modeli czy nawet dorabiania uszkodzonych części mogą się przydać w niemal każdej branży. Dobrym przykładem z obszaru bliskiego Czytelnikowi będzie np. opcja wykonania prototypu obudowy urządzenia celem sprawdzenia jej walorów ergonomicznych i estetycznych. Przy okazji można sprawdzić w praktyce, czy przykładowe rozmieszczenie otworów wentylacyjnych lub/i użyte chłodzenie aktywne spełniają swoje zadanie. Co więcej, koszt przygotowania nowego prototypu jest minimalny, więc projekt da się wykonywać iteracyjnie, aż do uzyskania pożądanego kształtu. Druk 3D pozwala też wyprodukować przetestowane obudowy (szczególnie gdy mamy do czynienia z produkcją małoseryjną, w której przypadku zamówienie formy na wtryskarki okazałoby się zbyt drogie). Istnieją wyspecjalizowane „farmy drukarek", realizujące takie zlecenia i oferujące przy tym szeroką gamę materiałów oraz kolorów, a nawet druk używający kilku różnych materiałów naraz.

Inne, przetestowane w praktyce zastosowania druku 3D to:

- produkcja form do odlewania wkładek i zatyczek do uszu z silikonu medycznego;
- wytwarzanie narzędzi oraz przystawek (NASA testowała druk klucza nastawnego na Międzynarodowej Stacji Kosmicznej);
- dorabianie niedostępnych lub kosztownych zaślepek, gałek, a także innych elementów tworzywowych;
- wykonywanie adapterów i elementów montażowych;
- produkcja systemów do przechowywania elementów oraz narzędzi (np. GridFinity);
- wytwarzanie pojedynczych zabawek, układanek, figurek lub innych drobnych gadżetów;
- produkcja niewymienionych powyżej, a przydatnych przedmiotów użytkowych.

W tej serii artykułów skupimy się głównie na zastosowaniach technologii druku 3D w praktyce projektantów i serwisantów elektroniki, ale pojawią się też – w celach ilustracyjnych – inne przykłady potencjału tej technologii.

Technologie druku 3D

Istnieje wiele różnych technologii druku 3D. W przemyśle lotniczym stosowano metodę, w której na podłożu układa się warstwę metalowego proszku, po czym wiązka lasera zgrzewa wybrane punkty na podstawie projektu. Następnie nakłada się kolejną warstwę proszku

i powtarza proces – aż do uzyskania kompletnej części. Niewątpliwą zaletą tej metody jest produkowanie detali z metalu, lecz koszty i czas potrzebne na wykonanie jednego modelu czynią ją mało użyteczną – rzecz jasna poza produkcją pojedynczych prototypów lub komponentów. Dlatego – zamiast omawiać tak specjalistyczne rozwiązania – skupimy się na dwóch technologiach druku dostępnych dla każdego: z użyciem filamentu oraz żywicy światłoczułej.

W metodzie stereolitograficznej stosuje się światłoutwardzalną żywicę epoksydową. W podstawie zbiornika znajduje się wysokiej rozdzielczości ekran LCD podświetlony silnym źródłem światła ultrafioletowego, zaś nad ekranem – metalowa płytka, która w trakcie pracy ulega stopniowemu podnoszeniu. Zbiornik wypełnia się żywicą, po czym płytę nośną opuszcza się nad ekran, tak by zachować minimalny odstęp. Na ekranie wyświetlany jest obraz pożądanych kształtów, a światło UV utwardza żywicę tam, gdzie przenika przez ekran. Po naświetleniu płyta zostaje podniesiona, po czym naświetlaniu ulega następna warstwa materiału. Proces powtarza się aż do zakończenia druku. Gotowe wydruki są myte w alkoholu izopropylowym, a potem dodatkowo naświetlane aż do uzyskania pełnej twardości, gdyż sam proces druku tylko częściowo utwardza żywicę.

Metoda ta pozwala tworzyć bardzo precyzyjne modele z licznymi, drobnymi detalami, a czas trwania całego procesu drukowania nie zależy od złożoności kształtów. Z drugiej strony – druki zazwyczaj są kruche, żywica generuje nieprzyjemne zapachy, a ryzyko jej przypadkowego rozlania jest spore. Dlatego też ta metoda zdobyła nieco mniejszą popularność w gronie hobbystów i ma ograniczone zastosowania wśród profesjonalistów.

Drugą – najpopularniejszą – metodę druku 3D stanowi druk z użyciem filamentu. W tej technologii ekstruder przemieszcza się nad blatem, a filament, czyli drut z tworzywa termoplastycznego, jest wprowadzany do ekstrudera mechanicznie. Blok grzewczy stapia koniec tego filamentu, a nacisk świeżego materiału wypycha stopiony materiał przez dyszę na blat. Dysza rozprowadza go, a jeśli w którymś miejscu okazuje się on niepotrzebny – napęd ekstrudera cofa filament, natomiast podciśnienie zasysa część stopionego materiału do wnętrza głowicy. Wentylator(-y) przy głowicy chłodzi nałożony już materiał. Po wydrukowaniu warstwy głowica oddala się od blatu na zdefiniowaną w ustawieniach modelu odległość, po czym rozpoczyna druk kolejnej warstwy (przy czym świeżo nałożony filament częściowo wtapia się w warstwę poprzednią).

Metoda filamentowa pozwala na zastosowanie bardzo wielu różnorodnych materiałów w szerokiej palecie kolorów, dzięki czemu wydruki mogą mieć najróżniejsze właściwości. **Fotografia 1** prezentuje przykładowe wydruki (od lewej: podstawka pod płytkę rozwojową wykonana z niebieskiego PLA, duża i mała gałka wykonane z półprzezroczystego PLA w kolorze żółtym oraz duża gałka wykonana z czarnego PLA typu Silk). Czas trwania druku zależy od rozmiarów detalu, użytego filamentu, złożoności modelu i grubości warstw. Na prędkość pracy drukarki wpływ mają też jej parametry mechaniczne, zdolność systemu kontrolnego do ich kompensacji, średnica dyszy, moc grzałki głowicy drukującej i oczekiwana jakość wydruku. Mechaniczne właściwości gotowych wydruków zależą natomiast od użytego filamentu i grubości warstwy, ale w większości przypadków można założyć, iż wydruk będzie o 20...60% słabszy, jeśli działające na niego siły

Fotografia 1. Przykładowe wydruki dla elektroników: podstawka pod płytkę drukowaną i gałki do potencjometrów lub enkoderów

Fotografia 2. Dwa wydruki print-in-place: brelok z ruchomych pierścieni i przekładnia planetarna

dążą do przemieszczenia względem siebie lub rozerwania warstw – pod tym względem wydruki przypominają drewno. Oczywiście na tę zależność wpływa rodzaj użytego filamentu i dodatkowe procesy obróbki gotowego wydruku. I to właśnie mnogość dostępnych materiałów stanowi największą siłę druku 3D metodą filamentową (określaną skrótem FDM).

Druk 3D a tradycyjne metody wytwarzania prototypów

Tradycyjnie obudowy prototypowe i inne elementy wykonuje się metodami subtraktywnymi, tj. poprzez usuwanie materiału. Frezarki i tokarki, w tym CNC, pozwalają obrabiać najróżniejsze budulce, od drewna, przez tworzywa sztuczne, do metali. Detale można też obrabiać, używając narzędzi ręcznych lub elektronarzędzi, takich jak wielofunkcyjne minifrezarki. Elementy blaszane da się wycinać za pomocą przecinarek plazmowych, laserowych albo typu Waterjet, a następnie zaginać na giętarkach ręcznych lub hydraulicznych. Sklejkę, a także akryl, poliwęglan i inne tworzywa można obrabiać w podobny sposób. A to tylko część z istniejących metod, które ponadto często trzeba ze sobą łączyć. Natomiast obróbka bardziej złożonych kształtów, nawet na frezarce czy tokarce CNC, nierzadko wymaga zarówno zmian narzędzi obrabiających element, jak i orientacji czy mocowania tego elementu, a każda taka zmiana wiąże się z koniecznością uważnego pilnowania kolejności operacji i tego, by maszyna nie zgubiła orientacji względem elementu.

W porównaniu z opisanymi powyżej technologiami druk 3D wydaje się niezwykle wręcz prosty. Nie oznacza to jednak, iż metoda ta ma ograniczoną użyteczność. Jako proces addytywny pozwala bowiem uzyskać bardzo skomplikowane kształty bez ciągłej zmiany orientacji elementu czy zmiany używanego narzędzia i bez przekładania detalu między różnymi maszynami. Unikalną zaletę druku 3D stanowi też możliwość wykonywania części, które tkwią wewnątrz innych elementów – na portalach udostępniających modele do druku 3D nie brakuje takich właśnie projektów. Przykładem mogą być różne zabawki typu fidget – choćby przekładnie planetarne, których nie da się rozebrać. **Fotografia 2** ukazuje dwa modele: brelok złożony z koncentrycznych pierścieni, zdolnych do obracania się względem siebie – oraz przekładnię planetarną, która też zyskuje ruchomość (ale nie rozpada się mimo braku wspólnej bazy utrzymującej poszczególne elementy) dzięki specjalnemu profilowi zębów. **Fotografia 3** pokazuje zaś najpopularniejszy model do testowania jakości druku i filamentu: łódkę 3DBenchy oraz trzy egzemplarze zabawki „Trilobug" w różnych skalach. Te modele też są drukowane w jednym „kawałku", a zawiasy między segmentami pozostają ukryte. Jeden z „robaków" wykonany został z filamentu PLA udającego kolor i fakturę drewna, choć niezawierającego go wcale (istnieją oczywiście filamenty zawierające kompozytowe drewno w formie drobnych trocin).

Druk 3D, jak każda inna metoda wytwórcza, ma też swoje wady i ograniczenia. Jeśli pierwsza warstwa nie przylgnie dobrze do blatu w drukarce filamentowej albo odklei się w trakcie drukowania, to cały

Fotografia 3. 3DBenchy oraz trzy stworzenia o ruchomych segmentach drukowane tak, jak je pokazano

wydruk jest do wyrzucenia. Dysze czasem się zapychają, a wtedy masa stopionego tworzywa może oblepić cały blok grzewczy i obudowę głowicy, co wymaga jej częściowego lub całkowitego demontażu celem wyczyszczenia. Źle dobrane parametry bądź zbyt wilgotny filament mogą prowadzić do (szpecących wygląd i osłabiających wytrzymałość) niedoborów lub nadmiarów materiału, powstawania nici i bąbli oraz bardzo rzucających się w oczy szwów (w miejscu, gdzie rozpoczyna się druk nowej warstwy). Zastosowanie niektórych materiałów termoplastycznych wiąże się z wydzielaniem nieprzyjemnego zapachu, a w razie przegrzania prowadzi do emisji toksycznych i niebezpiecznych oparów oraz gazów. Druk 3D, wbrew reklamom producentów, nie jest wcale procesem w pełni automatycznym (bezobsługowym) i wymaga sporej wiedzy z różnych dziedzin. Podobnie reklamy kłamią, mówiąc o tym, że wystarczy załadować plik i można od razu rozpocząć drukowanie. W rzeczywistości większość drukarek wymaga pewnej dozy regulacji, konserwacji, dbałości o czystość i pilnowania pracy urządzenia – przynajmniej na początku procesu drukowania. Niejeden wydruk zmienił się w bezkształtną masę tworzywa albo w plątaninę cienkich nitek, bo pierwsza warstwa odkleiła się od blatu. Droższe drukarki 3D używają kamer i systemów AI, by rozpoznać problemy z drukiem i zatrzymać go, zanim dojdzie do zmarnowania materiału.

Podczas używania frezarek i tokarek CNC proces obróbki może się nie udać, jeśli obrabiany element wyrwie się z uchwytu albo frez zderzy się z nim lub z uchwytem montażowym. Takie zdarzenia są niemal zawsze wynikiem błędu operatora. W przypadku druku 3D natomiast przyczyna niepowodzenia nierzadko leży w utracie wypoziomowania blatu z powodu wibracji, zbyt dużej wilgotności filamentu, różnicach produkcyjnych między dwoma filamentami tego samego typu (na przykład filament PLA od jednego producenta może mieć nieco inną temperaturę topnienia i lepkość niż niemal identyczny filament PLA innego producenta), niewidocznym gołym okiem zużyciu dyszy (zwłaszcza przy druku filamentem w kolorze białym) czy choćby przesunięciu się rurki Bowdena w mocowaniu. Warto też pamiętać o tym, że niektóre materiały mają tendencję do odkształcania się z powodu spadku temperatury – duże modele mogą skurczyć się tak mocno, że ich skrajne fragmenty odklejają się od podłoża. W esktremalnych wypadkach istnieje ryzyko uderzenia głowicy o odstającą krawędź i całkowitego oderwania modelu.

Anatomia filamentowej drukarki 3D

Wszystkie drukarki 3D używające filamentu zawierają te same, podstawowe elementy. Najważniejszy z nich to głowica, do której doprowadzony zostaje filament i to w niej jest on stapiany, a następnie wyciskany przez dyszę oraz nakładany warstwa po warstwie. Głowica ma też wbudowany, który chłodzi stopione tworzywo po nałożeniu. Filament podaje się za pomocą ekstrudera, w którym jest on trzymany przez dwa (lub więcej) koła zębate – co najmniej jedno z nich napędzane przez silnik krokowy, a drugie – dociskane regulowaną sprężyną. Ekstruder może znajdować się bezpośrednio nad głowicą (Direct Drive), co ułatwia druk filamentami elastycznymi – lub na ramie drukarki, co redukuje masę głowicy (między ekstruderem a głowicą znajduje się wówczas rurka Bowdena wykonana z PTFE). Trzy silniki krokowe (lub więcej) poruszają głowicą i/lub stołem. Blat ma wbudowaną grzałkę, a z wierzchu pokrywa go zwykle zdejmowalna tafla materiału, do którego przywiera filament. Popularnymi materiałami są szkło lub blacha stalowa pokryta PEI, ale z powodzeniem można użyć też jednostronnego laminatu FR4 (ułożonego miedzią do dołu – filament przywiera do żywicy). Ostatnimi elementami (poza konstrukcją mechaniczną) są sterownik i zasilacz.

Początkowo wszystkie drukarki 3D bazowały na sterownikach zbudowanych w oparciu o ośmiobitowe mikrokontrolery Atmel AVR ze względu na to, iż pierwsze otwarte oprogramowanie do tych maszyn napisane zostało w środowisku Arduino. Obecnie nowe drukarki 3D zawierają układy ARM ze względu na ich większą moc obliczeniową potrzebną do konwersji komend G-Code na ruch maszyny. W tym tkwi różnica między drukarkami 3D a frezarkami CNC i podobnymi narzędziami – drukarki same interpretują komendy i nie potrzebują stałego podłączenia do komputera, na którym pracuje program sterujący, jak Mach3 czy bCNC. Obecnie na rynku pojawiają się drukarki z układami SoC, które używają uczenia maszynowego i prostej kamery, by wykrywać problem z wydrukiem i wstrzymać pracę drukarki bez konieczności łączenia się z zewnętrznym serwerem producenta urządzenia. Funkcjonalność taką można dodać samodzielnie, używając na przykład oprogramowania Octoprint.

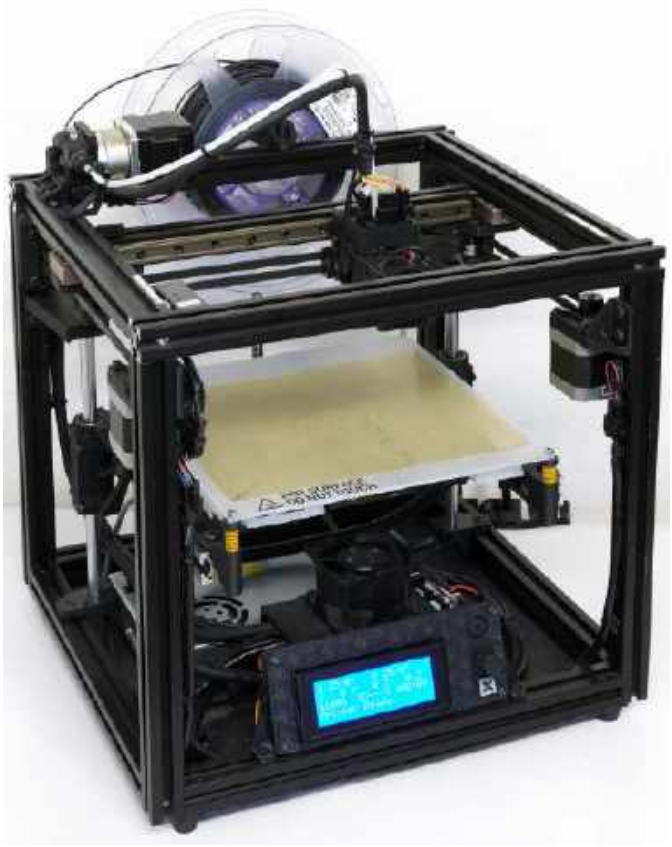
Fotografia 4 ukazuje typową drukarkę 3D o konstrukcji ramowej z ruchomym blatem (tzw. „bed slinger"), w tym przypadku jest to model Ender 3 V2 firmy Creality. Określenie „bed slinger" odnosi się do faktu, iż jedna z osi (w tym wypadku oś Y) przesuwa cały blat wraz z drukowanym przedmiotem. Oś X to pozioma belka, na której znajduje się głowica drukująca. Po lewej stronie belki umocowany jest silnik krokowy napędzający oś X za pomocą paska klinowego, zaś z drugiej strony ramy zamocowany jest ekstruder, obok niego natomiast znajduje się śruba trapezowa osi Z, która podnosi całą oś X.

Fotografia 4. Drukarka 3D typu „bed slinger"

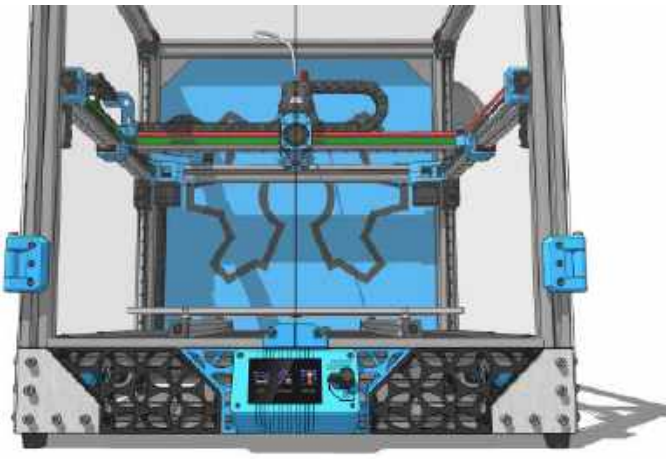
Ekstruder łączy się z głowicą za pomocą dość sztywnej rurki wykonanej z PTFE, nazywanej rurką Bowdena. W tej rurce przemieszcza się filament do głowicy. Ponieważ filament ma średnicę 1,75 mm, a rurka – średnicę wewnętrzną 2 mm, tworzy się pewien „luz", który ogranicza możliwości druku filamentami elastycznymi, a retrakcja, czyli odległość cofania filamentu, musi być dość duża. Zaletą rozwiązania z rurką Bowdena jest niższa masa głowicy, co pozwala na szybszą jej akcelerację, a tym samym na większą maksymalną prędkość druku. Innym rozwiązaniem byłoby zamontowanie ekstrudera bezpośrednio nad głowicą drukującą, czyli tzw. Direct Drive. Odległość między ekstruderem a blokiem grzewczym, w którym filament jest stapiany, wynosi w takim przypadku 2...4 cm, dzięki czemu retrakcja może być mniejsza, a filamenty elastyczne mają mniejszą „szansę" na nadmierną kompresję w rurce.

W ostatnich latach coraz częściej spotyka się drukarki „Core XY": osie X i Y tworzą w nich jeden, wspólny zespół poruszający głowicą. Cała maszyna wyposażona jest w sztywną ramę skrzyniową, do której można przymocować panele obudowy. Istnieją dwa warianty tej konstrukcji: w pierwszym wariancie (**fotografia 5**) osie X i Y nie przemieszczają się względem reszty maszyny, ale w zamian blat jest ruchomy i w pozycji początkowej znajduje się u góry, by powoli opadać w miarę drukowania kolejnych warstw. W drugim wariancie (**fotografia 6**) blat pozostaje nieruchomy, a osie X i Y stanowią część ruchomej ramy, która w trakcie drukowania kolejnych warstw jest unoszona w górę. Drukarki Core XY często mają wbudowane dodatkowe funkcjonalności, takie jak autopoziomowanie blatu, ogrzewanie wnętrza obudowy, zintegrowaną kamerę do monitorowania wydruku, oświetlenie czy wreszcie – system zmiany filamentu w trakcie druku.

Warto też wspomnieć o drukarkach typu Delta (**fotografia 7**). W ich przypadku głowica umocowana jest na trzech ramionach, których drugie końce połączone są z trzema wózkami przemieszczającymi się wzdłuż pionowych prowadnic. Kontrolując odpowiednio wysokość każdego wózka, można umieścić głowicę w dowolnym punkcie



Fotografia 5. Drukarka 3D „Core X-Y" z podnoszonym blatem



Fotografia 6. Drukarka 3D „Core X-Y" z opuszczaną ramą X-Y

nad (zwykle okrągłym) blatem. Konstrukcje tego typu kiedyś oferowały najwyższą prędkość druku i większą maksymalną wysokość obszaru roboczego, obecnie jednak zostały niemal całkowicie wyparte przez inne modele.

Proces drukowania krok po kroku

Pierwszy krok w procesie wytwarzania przedmiotu metodą druku 3D to przygotowanie modelu. Narzędzi do modelowania i projektowania znajdziemy bardzo wiele, a większość tych programów nie należy do tanich. Zagadnienie modelowania jest bardzo szerokie i wykracza poza zakres tej publikacji. Wspomnę jednak o dwóch programach

Fotografia 7. Drukarka typu Delta – głowica znajduje się na trzech sztywnych ramionach, których wysokość kontrolują trzy silniki krokowe

wartych uwagi. Pierwszy – Blender – to program przeznaczony do bardziej tradycyjnego modelowania, renderowania i animacji. Dostępny wprawdzie na licencji Open Source, ma jednak ogromne wsparcie ze strony branży filmowej, gamingowej oraz ze strony producentów sprzętu. Ze względu na przeznaczenie Blender nie nadaje się zbytnio do projektowania modeli o ściśle określonych wymiarach, za to oferuje potężne narzędzia do wirtualnego rzeźbienia i modyfikacji istniejących modeli. Drugim programem, przeznaczonym do parametrycznego projektowania, symulacji i przygotowania do produkcji, jest Autodesk Fusion 360. Środowisko to jest generalnie płatne, ale producent oferuje darmową licencję dla hobbystów i małych firm – w zamian za pewne ograniczenia co do liczby projektów otwartych do edycji i mocno limitowany dostęp do symulacji użytkownicy otrzymują potężne narzędzia CAD/CAM/CAE. We Fusion 360 można narysować schemat, zaprojektować płytkę drukowaną, a potem wymodelować dopasowaną kształtem obudowę. Program oferuje też wszystkie niezbędne narzędzia, by przygotować pliki G-Code na frezarki i tokarki CNC, wycinarki plazmowe i laserowe czy nawet drukarki 3D, choć lepiej używać do tego specjalnych slicerów.

Slicing to właśnie drugi etap w procesie drukowania. Model jest w nim „plasterkowany", a program generuje (zależnie od typu drukarki) albo mapę poszczególnych warstw do naświetlania, albo zbiór poleceń G-Code opisujących ruchy głowicy i ekstrudera oraz parametry druku. W slicerze ustala się temperatury wydruku, prędkość posuwu, metodę poprawy przylegania modelu do płyty, opcje dodawania podpór, stopień wypełnienia modelu i wiele innych. Istnieje sporo slicerów, ale z całej tej grupy wyróżnić można dwa najpopularniejsze: Ultimaker Cura oraz Prusa Slicer. Oba programy mają gotowe konfiguracje do szerokiej gamy dostępnych na rynku drukarek, gotowe konfiguracje do różnych materiałów (choć te mogą wymagać dostosowania) i ogrom różnych, mniej lub bardziej zaawansowanych opcji konfiguracyjnych, wtyczek czy też specjalnych trybów pracy. W jednym z następnych artykułów omówię dokładnie proces generowania pliku G-Code w slicerze Ultimaker Cura.

Gotowy plik przesyłany jest do drukarki 3D. Można to zrobić (zależnie od dostępnych opcji) za pomocą sieci Wi-Fi, łącząc drukarkę z komputerem kablem USB lub kopiując plik na kartę pamięci, którą wkłada się następnie do gniazda w sterowniku drukarki. Przed rozpoczęciem drukowania trzeba założyć szpulę z filamentem i wprowadzić go do ekstrudera. Drukarki z rurką Bowdena mają ekstruder umieszczony z dala od właściwej głowicy, więc trzeba go przepchnąć przez całą długość rurki ręcznie. W głowicach Direct Drive odległość między ekstruderem a gorącym końcem głowicy wynosi kilkanaście milimetrów, zatem dystans jest wielokrotnie mniejszy. Jeśli zmieniamy kolor lub rodzaj filamentu, głowicę trzeba wstępnie ogrzać, a następnie wypchnąć z niej stary filament. Czasami używa się do tego specjalnych filamentów „czyszczących". W przypadku zmiany materiału na filament kompozytowy może też być konieczna wymiana samej dyszy – ten proces również przeprowadza się „na gorąco", bo zastygły filament sięga wnętrza gardzieli, co może utrudniać odkręcanie dyszy w temperaturze pokojowej. Warto też przygotować sam blat. Zazwyczaj oznacza to przetarcie go wilgotną szmatką lub użycie alkoholu izopropylowego oraz ściereczki z mikrofibry. Niektóre materiały wymagają naklejenia taśmy samoprzylepnej albo pokrycia blatu klejem lub tworzywem rozpuszczonym w acetonie. Po wykonaniu tych czynności wybiera się plik z listy i rozpoczyna proces drukowania – ten nie wymaga zwykle dalszej interakcji, należy jednak mieć baczenie, czy pierwsza warstwa dobrze przylgnie do podłoża, a potem obserwować, czy model nie oderwie się od niego lub nie nastąpi inna awaria.

Po zakończeniu procesu wydruku model trzeba odczepić od podłoża. Konieczna może też okazać się obróbka mechaniczna, jak choćby oderwanie wsporników, odcięcie nitek, wygładzenie papierem ściernym lub pilnikiem „szwów", etc. Niektóre tworzywa warto umieścić w pojemniku z oparami rozpuszczalnika, na przykład acetonu, co pozwoli wygładzić powierzchnię i usunąć linie warstw. W przypadku druku stereolitograficznego półtwardy model trzeba ostrożnie oderwać od płytki nośnej, a następnie wypłukać 2...3 razy w alkoholu izopropylowym. Oczyszczony model albo umieszcza się w naświetlarce UV, albo wystawia na działanie słońca – dzięki temu ulega on całkowitemu utwardzeniu.

Warto też dodać, iż gotowe modele można poddać dalszej obróbce, zależnie od potrzeb. Należy jednak wziąć pod uwagę fakt, że niektóre materiały nie dają się łatwo szlifować ze względu na niską temperaturę mięknięcia. Wybrane elementy poddaje się też procesowi wyżarzania, co zmienia strukturę polimeru i usuwa naprężenia wewnątrz materiału. Proces ten może jednak zmienić wymiary elementu lub doprowadzić do deformacji, dlatego często przeprowadza się go w foremce wypełnionej drobnym proszkiem i sprasowanej (by ustabilizować kształt) lub po zatopieniu elementu w gipsie (by móc go kompletnie stopić). Wyżarzanie wymaga, by element był drukowany ze stuprocentowym wypełnieniem, w przeciwnym razie bowiem może zmienić kształt pod wpływem grawitacji.

Zakończenie

Druk 3D, wbrew reklamom, nie jest prostym procesem produkcji absolutnie wszystkiego. Drukarka 3D też nie będzie nowym sprzętem domowym dla każdego. Wytwarzanie addytywne może być równie skomplikowanym i wymagającym procesem, co każdy inny proces produkcyjny, a używanie drukarki wymaga nie tylko sporej wiedzy, ale też czasu i chęci do eksperymentowania, trudno bowiem znaleźć drukarkę, która nie wymagałaby regulacji, kalibracji, a czasami nawet modyfikacji. Ponadto trzeba wiedzieć, jaki filament będzie optymalny do danego zastosowania. I właśnie filamentami zajmiemy się w następnej części naszego cyklu.

Paweł Kowalczyk

Internet Rzeczy w pomiarach środowiskowych (5)

Podziękowania dla Pana Macieja Michny z Centrum Badań i Rozwoju Nordic Semiconductor w Krakowie za udostępnienie zestawów sprzętowych Power Profiler Kit II (PPK2).

Optymalizacja poboru mocy urządzenia IoT z płytką Raspberry Pi Pico W

Optymalizacja poboru mocy urządzeń IoT jest zagadnieniem obejmującym liczne aspekty sprzętowe i programowe. Dotyczą one wyboru mikrokontrolera, czujników, źródła zasilania, protokołu transmisji bezprzewodowej i wielu innych. Wśród różnych płytek przydatnych do budowy urządzenia IoT sporym powodzeniem cieszy się Raspberry Pi Pico W. Wyposażona została w mikrokontroler RP2040 z dwoma rdzeniami ARM Cortex-M0+ oraz układ scalony CYW43439 firmy Infineon umożliwiający łączność bezprzewodową w standardach Wi-Fi 4 i Bluetooth 5.2. Zrealizowanie optymalnego zasilania urządzenia IoT ze wspomnianą płytką nie jest jednak proste. W artykule pokazane zostało odpowiednie rozwiązanie tego problemu.

Optymalizacja poboru mocy układu IoT zaczyna się już na etapie wyboru czujników i układów scalonych towarzyszących procesorowi, takich jak przetwornice DC/DC. Należy przede wszystkim wybrać układy o małym poborze energii w stanie spoczynku. Prąd spoczynkowy IQ (Quiescent current) można zdefiniować jako prąd pobierany przez układ scalony w stanie włączonym bez obciążenia i przełączania [9]. Stanu spoczynku nie należy mylić ze stanami obniżonego poboru mocy, jak: stan oczekiwania (standby), drzemki (dormant), wybudzania (wake-up) czy uśpienia (sleep). Kolejnym warunkiem minimalizacji zużycia energii jest zapewnienie zastosowania w roli czujników odpowiednich układów małej mocy. Konstrukcja czujnika może w dużym stopniu wpływać na poziom energii pobieranej przez urządzenie [1].

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

Kluczowanie zasilania

Pierwszym sposobem ograniczania poboru mocy jest zastosowanie klucza do włączania zasilania czujnika. Na przykład w *Thingy:52 IoT Sensor Kit* firmy Nordic Semiconductor [6] układ scalony czujnika otoczenia CCS811 (26 mA podczas pomiaru) jest zasilany poprzez analogowy przełącznik scalony typu NX3DV2567 firmy NXP.

Drugi sposób kluczowania zasilania stanowi zasilanie czujnika bezpośrednio z wyprowadzenia IO procesora. Typowo z linii tej można pobierać prąd co najmniej 2 mA (RP2040 oferuje tryby obciążalności wyjść na poziomie 2, 4, 8 oraz 12 mA) [2]. Przykładowo: w module czujnikowym *BP-BASSENSORSMKII* firmy Texas Instruments [7] czujnik światła OPT3001 tej firmy pobiera tylko 3,7 µA prądu w trakcie pomiaru i jest dołączony bezpośrednio do wyprowadzenia procesora.

Trzeci sposób ograniczania poboru mocy to zastosowanie czujnika z tak małym poborem prądu w stanie oczekiwania, że można go zasilać cały czas (bez kluczowania). Przykładem jest ciągłe zasilanie czujnika ruchu ADXL362 firmy Analog Devices w *Thingy:53 IoT prototyping*

Tabela 1. Pobór zasilania układu RP2040 [2]

Tryb pracy	Pobór prądu [mA]					
	DVDD		IOVDD		USB_VDD	
	Typowy średni	Maksymalny średni	Typowy średni	Maksymalny średni	Typowy średni	Maksymalny średni
Popcorn	10,9	16,6	24,8	35,5	-	-
BOOTSEL mode Active	9,4	14,7	1,2	4,3	1,4	2,0
BOOTSEL mode Idle	9,0	14,3	1,2	4,3	0,2	0,6
Dormant	0,18	4,2	-	-	-	-
Sleep	0,39	4,5	-	-	-	-

platform firmy Nordic Semiconductor [5]. Prąd sensora w stanie czuwania sięga zaledwie 0,01 µA. Ciągła praca czujnika pozwala na wybudzanie urządzenia w przypadku detekcji ruchu.

Optymalizacja transmisji bezprzewodowej

Wybór rozwiązania łączności dla urządzenia IoT ma poważne konsekwencje dla selekcji komponentów do danej aplikacji oraz wydajności urządzenia i zużycia energii przez nie. Odległość między dwoma węzłami sieci bezprzewodowej, topologia, szybkość transmisji danych i rozmiar wiadomości wpływają na czas transmisji, co z kolei oddziałuje na budżet energetyczny.

Optymalizacja oprogramowania pod kątem niskiego zużycia energii

Zaprogramowanie urządzenia do pracy w trybach niskiego poboru mocy znacząco wpłynie na oszczędzanie energii baterii. Nowe rozwiązania w zakresie zarządzania zasilaniem wprowadziły szeroką gamę trybów uśpienia o bardzo niskim poborze mocy. Należy zaprogramować aplikację tak, aby pracowała możliwie najkrócej w aktywnym trybie MCU. Może to oznaczać uproszczenie obliczeń, operacje wsadowe lub przejście na projekt asynchroniczny i sterowany przerwaniami.

Płytka Raspberry Pi Pico W

Raspberry Pi Pico W firmy Raspberry Pi to płytka z mikrokontrolerem RP2040 wyposażonym w dwa rdzenie ARM Cortex-M0+ (pracujące z częstotliwością do 133 MHz) oraz 264 kB RAM [10]. Na płytce znajduje się również pamięć QSPI Flash o pojemności 2 MB. Mikrokontroler RP20240 udostępnia rozbudowane interfejsy komunikacyjne: 2×SPI, 2×I²C, 2×UART, 3×12-bit ADC, 16 kanałów PWM oraz obsługę trybów niskiego zużycia energii: uśpienia (sleep) i trybu drzemki (dormant). Układ może być programowany w języku C/C++ lub MicroPython.

Płytka ma zamontowane gniazdko microUSB służące do zasilania oraz przesyłania danych (USB 1.1 w trybach Host i Device). Ponadto wyposażona została w układ scalony CYW43439 firmy Infineon realizujący łączność bezprzewodową w standardzie Wi-Fi 4 (IEEE 802.11b/g/n) oraz Bluetooth 5.2 (BDR, EDR oraz BLE) z pojedynczą anteną współdzieloną.

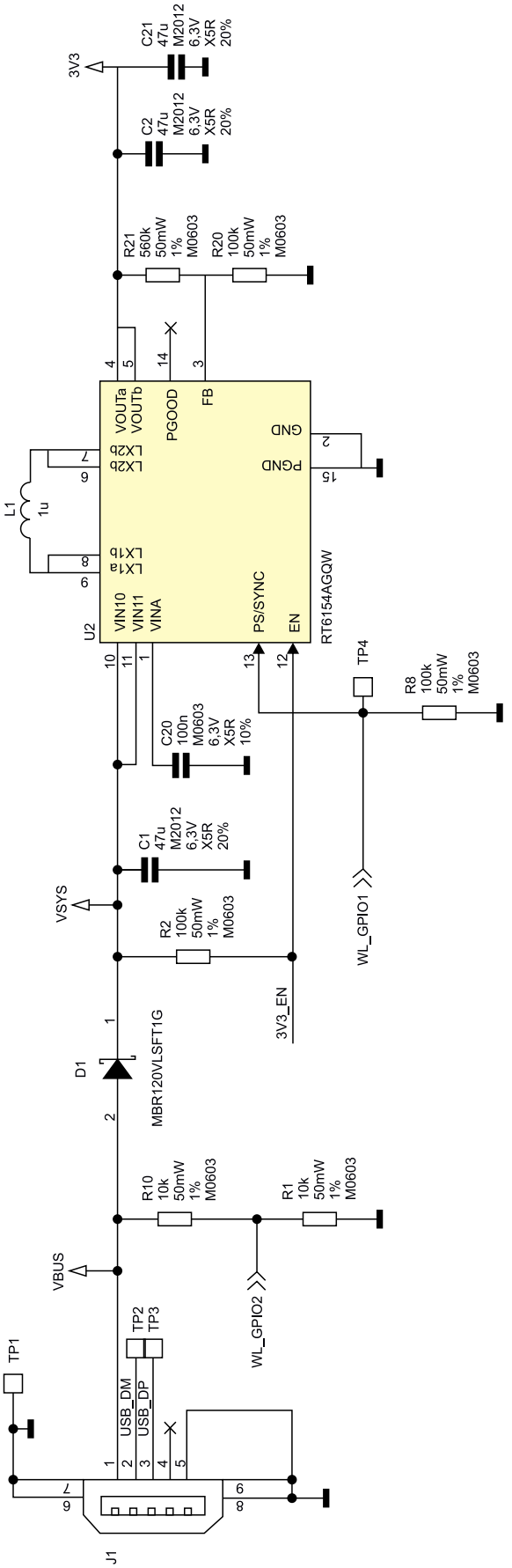
Po obu stronach płytki znajdują się pady umożliwiające wlutowanie złączy goldpin (2×20) lub bezpośrednie przylutowanie do płytki drukowanej (SMD). Udostępniają one zasilanie oraz 26 wyprowadzeń GPIO.

Procesor RP2040 zapewnia szereg opcji redukcji mocy zasilania [2]:

- Bramkowanie zegara najwyższego poziomu poszczególnych urządzeń peryferyjnych i bloków funkcjonalnych.
- Automatyczna kontrola bramek zegara najwyższego poziomu w oparciu o stan uśpienia procesora.
- Możliwość zmiany „w locie” częstotliwości lub źródła zegara systemowego (np. przełączenie na wewnętrzny oscylator pierścieniowy i wyłączenie PLL oraz oscylatora kwarcowego).
- Stan drzemki (dormant) o zerowej mocy dynamicznej, wybudzenie po zdarzeniu od GPIO lub przerwaniu od zegara RTC.
- Wprowadzanie pamięci w stan wyłączenia zasilania (z zachowaniem zawartości RAM).
- Bramkowanie zasilania urządzeń peryferyjnych, które są obsługiwane, np. ADC, czujnika temperatury.

Warto pamiętać, że jeśli zostanie zastosowany RTC w celu wybudzenia ze stanu drzemki, to musi on mieć zewnętrzne źródło zegara.

Procesor RP2040 wyposażono w dwie domeny zasilania: cyfrową rdzenia (DVDD, 1,1 V) i bloków wejścia-wyjścia (IOVDD, 3,3 V). W **tabeli 1** pokazany został pobór zasilania procesora podczas pracy typowego programu pomiarowego (Popocorn), pracy w trybie bootowania (z aktywną szyną USB i bez) oraz podczas pracy procesora w trybie drzemki (dormant) lub uśpienia (sleep) [2]. Zwraca uwagę dosyć wysoki maksymalny prąd rdzenia dla trybów drzemki i uśpienia.



Rysunek 1. Układ zasilania płytki Raspberry Pi Pico W [3]

Zasilanie płytki Raspberry Pi Pico W

Płytka Pico W została zaprojektowana z prostą, ale elastyczną architekturą obwodów zasilających i może łatwo współpracować z różnymi źródłami, takimi jak baterie lub zewnętrzny zasilacz.

Integracja Pico W z zewnętrznymi obwodami ładowania jest również prosta. **Rysunek 1** ukazuje układ zasilania płytki.

Na płytce Pico W występuje kilka napięć zasilania [3]:

- **VBUS** to napięcie wejściowe z gniazdka micro-USB. Nominalnie wynosi 5 V (lub 0 V, jeśli USB nie jest podłączone lub nie jest zasilane).
- **VSYS** to główne napięcie systemowe płytki, które może zmieniać się w zakresie od 1,8 V do 5,5 V i jest używane przez układ przetwornicy do generowania napięcia 3,3 V.
- **3V3** to główne zasilanie dla RP2040 i jego bloków GPIO, generowane przez przetwornicę DC/DC typu RT6154. Napięcie to może zostać użyte do zasilania obwodów zewnętrznych – maksymalny prąd wyjściowy zależy od obciążenia RP2040 i napięcia VSYS; zaleca się dołączanie pobierającego nie więcej niż 300 mA.

Te trzy szyny zewnętrznego obciążenia zasilania są dostępne na wyprowadzeniach (40, 39, 37) płytki Pico W.

Napięcie z portu micro-USB (VBUS) jest podawane przez diodę Schottky'ego na szynę VSYS. Dioda zwiększa elastyczność aplikacyjną, umożliwiając automatyczny wybór źródła zasilania z różnych źródeł VSYS (o ile dodatkowy zasilacz zostanie podłączony do VSYS przez zewnętrzną diodę Schottky'ego, co pozwoli zastosować sumę logiczną „na drucie").

Linia cyfrowa WL_GPIO2 (układu CYW43439) monitoruje napięcie szyny VBUS. Odczyt cyfrowy daje „0" – gdy nie ma napięcia VBUS i „1" – gdy jest ono obecne. Dzielnik R10 (10 kΩ) oraz R1 (10 kΩ) rozładowuje ładunek z szyny VBUS, jeśli nie jest ona zasilana. Dodatkowo napięcie VSYS podzielone przez trzy zostaje doprowadzone na wejście ADC3 (GP29) procesora. Wymaga to chwilowego wyłączenia układu CYW43439.

Linia cyfrowa WL_GPIO1 (układu CYW43439) steruje wejściem PS (Power Save) przetwornicy RT6154A. Jeśli na wejściu jest poziom niski (domyślnie), przetwornica pracuje w trybie PFM (pulse frequency modulation) i przy niskim obciążeniu włącza klucze tylko w celu koniecznego doładowania kondensatorów. Poziom wysoki na wejściu PS włącza tryb PWM i ciągłą pracę przetwornicy. Skutkuje to znaczącym zmniejszeniem tętnień wyjściowych przy jednoczesnym obniżeniu sprawności.

Wejście EN przetwornicy podciągnięte zostało do VSYS przez rezystor 100 kΩ (włączenie) i jest dostępne na wyprowadzeniu 37 płytki Pico W. Zwarcie tej linii do masy wprowadza układ przetwornicy w stan wyłączenia.

Trzeba pamiętać, że procesor RP2040 ma wbudowany regulator LDO dostarczający wewnętrznie napięcie 1,1 V do zasilania rdzenia.

Przetwornica Buck-Boost DC/DC typu RT6154A, ze stałym napięciem wyjściowym 3,3 V, jest przeznaczona do systemów zasilanych jednoogniwowym akumulatorem Li-Ion lub Li-Polymer o typowym napięciu od 2,5 V do 4,2 V. Można ją również stosować w systemach zasilanych dwu– lub trójogniwową baterią alkaliczną, NiCd lub NiMH. Przetwornica pracuje w szerokim zakresie napięć wejściowych od 1,8 V do 5,5 V z prądem upływu 1,0 µA (max) [11].

Obwód blokady podnapięciowej zapobiega nieprawidłowemu działaniu urządzenia przy niskich napięciach wejściowych. Uniemożliwia on włączenie przez przetwornicę kluczy zasilania w nieokreślonych warunkach oraz zapobiega głębokiemu rozładowaniu akumulatora. Aby przetwornica mogła się włączyć, napięcie VSYS musi być większe niż UVLO(H)=1,7 V. W czasie pracy, jeśli napięcie VSYS spadnie poniżej UVLO(L)=1,7 V, przetwornica zostanie wyłączona do czasu, aż zasilanie ponownie przekroczy próg narastania UVLO. Układ RT6154A uruchamia się automatycznie ponownie, jeśli napięcie wejściowe powróci do wysokiego poziomu napięcia wejściowego przekraczającego UVLO(H). Napięcie wyjściowe, po włączeniu RT6154A, wzrasta do ustawionej wartości w ciągu 1 ms. W okresie rozruchu cykl pracy oraz prąd szczytowy są ograniczone (Ilim=2,6 A), aby zredukować duży prąd szczytowy pobierany z wejścia.

Kluczowanie zasilania całego modułu

Nawet w głębokim uśpieniu procesor RP2040 pobiera typowo prąd od 180 µA (typ.) do 4,2 mA (max.) [8]. W wielu przypadkach, gdy wymagany jest minimalny pobór prądu, najlepszą opcją okazuje się całkowite wyłączenie systemu (lub części systemu z układem RP2040). Firmowe propozycje trzech takich układów zamieszczone zostały w publikacji „Power switching RP2040 for low standby current applications" [8]. W pierwszym rozwiązaniu zastosowano kluczowany układ przetwornicy DC/DC. W drugim użyto układu klucza prądowego, a w trzecim znalazło się kluczowanie zasilania tranzystorami MOSFET. W każdym przypadku potrzebny jest przycisk startu oraz podtrzymanie włączania poprzez sterowanie z wyjścia procesora. Praktyczna realizacji tych zaleceń znalazła zastosowanie w płytce Enviro Weather (PIM628) firmy Pimoroni [12].

Płytka Enviro Weather (PIM628)

Płytka Enviro Weather (PIM628) firmy Pimoroni wyposażona została w sterownik oparty na Raspberry Pi Pico W [12] oraz czujniki ciśnienia, wilgotności, temperatury i światła. Praca z płytką i jej programowanie zostało dokładnie omówione w poprzednim odcinku cyklu pt. „Stacja pogodowa Enviro Weather firmy Pimoroni" [15].

Moduł został zaprojektowany tak, aby działał dobrze przy zasilaniu bateryjnym. Na płytce Enviro Weather zastosowano scalony układ RTC (zegar czasu rzeczywistego), dzięki czemu można okresowo budzić mikrokontroler, odczytywać stany czujników (i opcjonalnie łączyć się z Wi-Fi), a następnie ponownie wyłączyć procesor, co przekłada się na miesiące nieograniczonej pracy na baterii. Zestaw przeznaczony jest do projektów stacji pogodowych.

Moduł Enviro Weather może przejść w tryb głębokiego uśpienia, w którym Pico W, czujniki pokładowe oraz czujniki podłączone przez gniazdko Qw/ST pozostają całkowicie wyłączone. Pobór prądu płytki w stanie uśpienia wynosi 20 µA – jedynym komponentem, który pozostaje uruchomiony, jest układ RTC, zdolny do ponownego obudzenia płytki po ustalonym czasie (według ustawień timera). Moduł można także wybudzić, naciskając przyciski POKE lub podłączając kabel USB.

Po podłączeniu do zasilania USB płytka Enviro Weather nie wchodzi w stan głębokiego uśpienia, chociaż oprogramowanie resetuje płytę za każdym razem, gdy nastąpi przerwanie od układu RTC.

Zegara RTC można także używać do śledzenia czasu i daty (co oznacza, że nie musimy marnować energii na wykonywanie połączeń bezprzewodowych w celu sprawdzenia godziny/daty za każdym razem, gdy rejestrujemy odczyt czujnika!). Dioda ostrzegawcza podłączona została do zegara RTC, dzięki czemu świeci nawet podczas głębokiego uśpienia płytki, powiadamiając użytkownika o ewentualnych problemach.

Enviro Weather może być zasilany napięciem od 2 V do 5,5 V. Sprawdzą się tu np. 2 lub 3 ogniwa alkaliczne AA lub AAA, 4 akumulatory NiMH lub jednoogniwowe LiPo. Moduł nie obsługuje ładowania akumulatora, więc potrzebna okaże się osobna ładowarka.

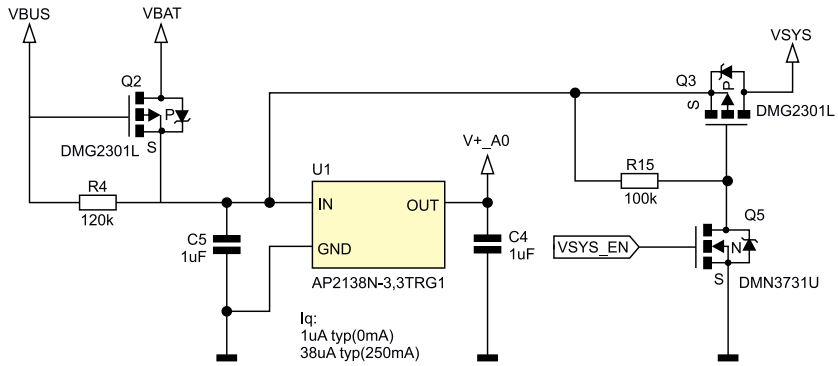
Układ scalony PCF85063A zegara RTC

Układ scalony PCF85063A firmy NXP to zegar czasu rzeczywistego (RTC) i kalendarz zoptymalizowany pod kątem niskiego zużycia energii [13]. Zasilanie układu mieści się w zakresie 0,9...5,5 V, zasilanie szyny I²C natomiast w zakresie 1,8...5,5 V. Pobór prądu wynosi 18 µA (aktywna szyna I²C) oraz 0,22 µA (szyna nieaktywna). Układ timera jest zdolny do generowania sygnału na wyjściu INT z odstępem czasu od 244 µs do 4 godz. 15 min. Na wyjściu CLKOUT może być ustawiony poziom niski lub przebieg o częstotliwości od 1 Hz do 21768 Hz (domyślnie po resecie).

Na płytce Enviro Weather układ PCF8563A jest zasilany z napięcia V+_A0 (3,3 V) i konfigurowany do generowania sygnału przerwania. Do wyjścia INT dołączono tranzystor MOSFET generujący sygnał RTC_ALARM. Wyjście CLKOUT układu dołączono natomiast do diody LED (LED_ALARM).

Zasilanie płytki Enviro Weather

Płytka Enviro Weather używa kilku napięć zasilania (**rysunek 2**) [4]. Czujniki LTR-559, BME280 i moduły wpięte do złącza rozszerzeń QW/ST zasilane są z napięcia 3,3 V dostarczanego przez płytkę Raspberry Pi Pico W. Układ RTC PCF8563A (oraz czujnik deszczu, jeśli jest podłączony) ma osobne zasilanie napięciem V+_A0 o wartości 3,3 V, wytwarzanym przez układ przetwornicy U1 typu AP2138N o bardzo niskim prądzie upływu 1,0 µA. Przetwornica U1 zapewnia „zawsze włączoną" szynę V+_A0 potrzebną płytce Enviro Weather w trybie niskiego poboru mocy (uśpienia).



Rysunek 2. Układ zasilania modułu Enviro Weather [4]

Jeśli jest obecne zasilanie na szynie VBUS, to przetwornica płytki Pico W zasila procesor RP2040. Tranzystor Q2 jest zablokowany i odcina napięcie zasilania z szyny VBAT. Dioda wewnętrzna tranzystora Q3 podaje zasilanie z szyny VSYS na wejście przetwornicy U1 (AP2138N) w sposób ciągły. Procesor pracuje cały czas i nie wchodzi w uśpienie.

Jeśli nie jest obecne zasilanie na szynie VBUS, to napięcie z gniazdka BATT jest podawane na szynę VBAT i włączany jest tranzystor Q2 poprzez rezystory R4 (120 kΩ, na płytce Enviro) oraz R10 i R1 (10 kΩ, na płytce Pico W) dołączone do masy. Włączony tranzystor Q2 doprowadza napięcie na wewnętrzną szynę zasilania, z którą połączone jest wejście przetwornicy U1 oraz tranzystor Q3 kluczujący podawanie zasilania na szynę VSYS.

Napięcie na szynie wewnętrznej może mieścić się w zakresie od ok. 1,8 do 5,5 V. Przy napięciu poniżej 3,3 V układ AP2138N powtarza na wyjściu napięcie wejściowe z małym spadkiem. Do szyny V+_A0 jest dołączony tylko układ PCF8563A zdolny do pracy już przy napięciu 1,8 V i z bardzo małym poborem prądu: 18 µA. Dołączona poprzez tranzystor Q3 przetwornica RT6154A płytki Pico W pracuje również od napięcia 1,8 V. Oznacza to poprawną pracę z wejścia BAT w szerokim zakresie napięć.

Sygnał VSYS_EN powoduje włączenie tranzystorów Q5 oraz Q3 i podanie zasilania z szyny wewnętrznej na szynę VSYS. Sygnał VSYS_EN jest tworzony jako suma (OR) czterech sygnałów: przycisku POKE, przerwania RTC (RTC_ALARM), a także wyprowadzenia GP2 procesora (HOLD_VSYS_EN) oraz czujnika deszczu. Zasilanie to włączane jest przez sygnał przerwania zegara RTC. Program Enviro Weather wystawia na nóżce GP2 procesora sygnał podtrzymania zasilania HOLD_VSYS_EN. Sygnał jest zdejmowany, gdy procesor wchodzi w stan uśpienia (lub zostaje wyłączony).

Zasilanie bateryjne płytki Enviro Weather

Do dynamicznego pomiaru prądu zasilania bardzo dobrze nadaje się zestaw Power Profiler Kit II (PPK2) firmy Nordic Semiconductor. Jest to samodzielny układ, który bez zewnętrznego sprzętu może mierzyć i dostarczać prądy od poniżej µA do 1 A. Praktyka pokazuje, że zakres pracy rozciąga się do ok. 2 A. Dokładny opis PPK2 zamieszczony został w artykule „Profilowanie mocy z zastosowaniem Power Profiler Kit II" [14].

Płytka Enviro Weather może być zasilana z jednego źródła: poprzez gniazdko microUSB albo z zewnętrznego napięcia poprzez złącze akumulatora JST-PH.

W ramach badań płytki Enviro Weather do gniazdka BAT (JST-PH) został dołączony naładowany akumulator Li-Po 3,7 V (z napięciem 4,12 V). Pomiary przeprowadzono po konfiguracji płytki [15]. Gdy włączone zostanie zasilanie, wystąpi pojedynczy pik prądowy ok. 1,69 A (**rysunek 3**, środkowy wykres). Potem przez ok. 11 s pracuje blok pomiaru i transmisji danych za pośrednictwem Wi-Fi. W tej fazie pulsuje biała dioda ACTIVITY i występują impulsy prądowe do ok. 300 mA (rysunek 3, wykres górny). Górny wykres na rysunku 3 pokazuje również fragment 10 ms przebiegu poboru prądu (wykresu środkowego) z próbkowaniem 100 kSps (rozdzielczość 10 µs). Następnie procesor wchodzi w uśpienie. Pracuje tylko LDO 3,3 V oraz zegar RTC. Pobór prądu maleje do ok. 35 µA. Szyna VSYS płytki Rasberry Pico W (rysunek 3, kanał 0 na dolnym przebiegu) jest zasilana tylko na czas aktywnej pracy procesora.

Duży pik prądowy przy włączaniu urządzenia stanowi spore wyzwanie dla źródła zasilania. Układ RTC ma zabezpieczenie Power-On Reset (POR). Wymagane jest, aby podanie zasilania VDD układu RTC zegara rozpoczynało się od zera woltów po włączeniu zasilania lub po wyłączeniu zasilania, aby zapewnić, że nie nastąpi uszkodzenie zawartości rejestrów. Czasami układ RTC po podaniu zasilania BAT zachowuje się nietypowo i w sposób ciągły słabo świeci dioda WARNING dołączona do wyjścia CLKOUT. Jest to spowodowane zadziałaniem układu POR zegara RTC. Wymusza on reset sprzętowy układu i ustawia na wyjściu CLKOUT domyślny przebieg 32768 Hz.

Przykład włączania zasilania bateryjnego płytki Enviro Weather, z wciśniętym przyciskiem POKE, przy niskim napięciu 2,607 V z gniazdka BATT, pokazany został na **rysunku 4**.

Włączanie zasilania płytki Enviro Weather z gniazdka BATT przebiega w dwóch etapach. W pierwszej kolejności włączana jest przetwornica obniżająca U1 typu AP2138N (rysunek 4, kanał CH1 – żółty), która na wyjściu dostarcza stałe napięcie na szynę V+_A0. W przypadku napięcia wejściowego od ok. 1,8 V do ok. 3,35 V przetwornica

Rysunek 3. Pobór prądu płytki Enviro Weather podczas bloku pomiaru i transmisji danych

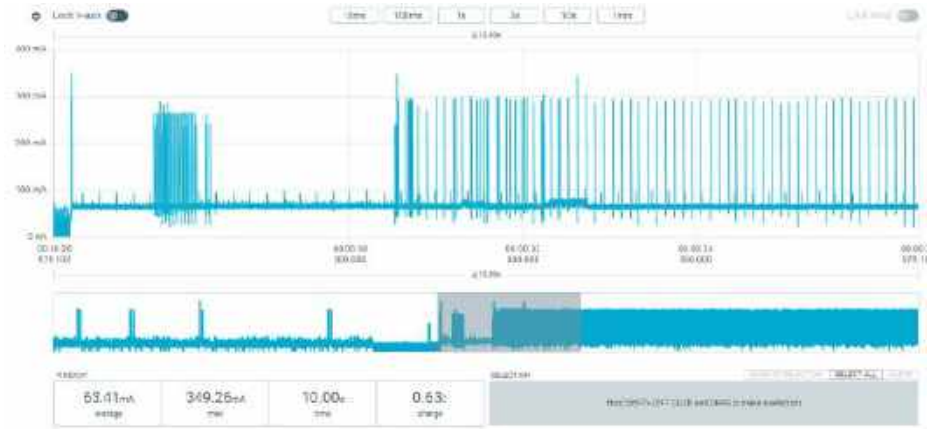
powtarza napięcie wejściowe ze spadkiem ok. 37 mV. Przy wyższych napięciach wejściowych przetwornica dostarcza regulowane napięcie 3,3 V.

Po wzroście napięcia VSYS_EN do ok. 0,6 V (w przybliżeniu po 70 µs) włącza się tranzystor Q5. Powoduje to otwarcie tranzystora Q3 i podanie napięcia z szyny wewnętrznej na szynę VSYS (rysunek 4, kanał CH3 – niebieski). Ładowany jest kondensator elektrolityczny 47 µF na wejściu przetwornicy RT6154 (U2) na płytce Pico W. Po około 200 µs ładowania startuje narastanie napięcia wyjściowego na szynie 3V3 z tej przetwornicy (rysunek 4, kanał CH2 – zielony). Całość pierwszego etapu trwa ok. 450 µs. Odpowiada to pierwszemu pikowi prądu zasilania (ok. 1,41 A) na rysunku 3 (środkowy wykres).

W drugim etapie na szynach V+_A0 oraz 3V3 ustalają się poprawne poziomy napięć. Pracujący procesor wystawia sygnał HOLD_VSYS_EN, co skutkuje trwałym włączeniem tranzystorów Q5 i Q3 i podtrzymaniem podawania pełnego napięcia zasilania na wejście przetwornicy RT6154. Próby wykazały możliwość obniżenia napięcia zasilania podawanego na gniazdko BATT do ok. 2,5 V.

W przypadku gdy napięcie na wejściu BATT jest zbyt niskie, przetwornica RT6154A (Pico W) nie dostarcza napięcia wystarczającego do rozpoczęcia pracy procesora RP2040. Czerwona dioda WARNING na płytce Enviro Weather lekko się żarzy w sposób ciągły.

Rysunek 4. Start pracy z zasilaniem z gniazdka BATT z wciśniętym przyciskiem POKE



Rysunek 5. Pobór prądu płytki Enviro Weather podczas zasilania z USB

Praca płytki Enviro Weather z zasilaniem USB

Podłączenie zasilania 5 V do szyny VBUS płytki Raspberry Pi Pico W powoduje rozpoczęcie pracy płytki Enviro Weather. Po włączeniu zasilania występuje pojedynczy pik prądowy ok. 1,4 A (**rysunek 5**). Po skończeniu bloku pomiaru i transmisji danych procesor nie wchodzi w stan uśpienia. Zamiast tego aktywnie sprawdza stan znacznika przerwania układu RTC. Średni pobór prądu spada do ok. 60 mA.

Jeśli do płytki Raspberry Pi Pico W zostały wlutowane złącza goldpin, to całość można, zamiast z szyny USB, zasilać poprzez pin VSYS napięciem od ok. 1,8 V do 5,5 V [3]. Jednak wtedy płytka Enviro Weather będzie pracować tak jak z zasilaniem USB.

Podsumowanie

Szczegółowe zrozumienie działania obwodów zasilania płytki Enviro Weather nie było łatwe. Brakuje właściwie jakiejkolwiek dokumentacji, co oznacza, że trzeba się opierać na schemacie oraz opisach w kodzie programu. Dodatkowo sprawę komplikuje dynamiczny charakter zachodzących zjawisk. Również informacja o konfigurowaniu zasilania procesora RP2040 do pracy z niskim poborem prądu jest schowana w zakładce dokumentacji na stronie producenta.

Próby pracy płytki Enviro Weather z zasilaniem akumulatorowym i wysyłaniem pomiarów do chmury Adafruit IO pokazały bardzo wysoką skuteczność oszczędzania energii. Zastosowany na płytce układ zasilania jest bardziej uniwersalny niż propozycje Raspberry Pi; nieco brakuje jedynie możliwości zastosowania transmisji Bluetooth zamiast Wi-Fi. Wydaje się, że ta kwestia mogłaby dodatkowo obniżyć wymagania energetyczne układu.

Henryk A. Kowalski
Instytut Informatyki
Politechnika Warszawska

Literatura

[1] How to Optimize Power Consumption of IoT Devices, Farnell An Avnet Company, 2024, https://tiny.pl/d937d
[2] RP2040 Datasheet, 1A microcontroller by Paspberry Pi, 4.06.2023, https://tiny.pl/d9375
[3] Raspberry Pi Pico W Datasheet – An RP2040-based microcontroller board with wireless, v2.1, 203 Mar 2023, Raspberry Pi, https://tiny.pl/d937j
[4] Enviro Weather (PIM628) schematic, enviro_weather_schematic.pdf, https://tiny.pl/d937n
[5] Nordic Thingy:53 IoT prototyping platform, Nordic Semiconductor, https://tiny.pl/d9372
[6] Nordic Thingy:52 IoT Sensor Kit, Product Page, https://tiny.pl/d937z
[7] BP-BASSENSORSMKII Sensors BoosterPack plug-in module for building automation, https://tiny.pl/d93rq
[8] RP2040 (Power switching RP2040 for low standby current applications), 2023-02-10), https://tiny.pl/d93rt
[9] What is the Essence of Quiescent Current?, Steve Taranovich, Feb. 2, 2022, Electronic Design, https://tiny.pl/d93rr
[10] Raspberry Pi Pico and Pico W, https://tiny.pl/dt49r
[11] RT6154A/B Current Mode Buck-Boost Converter, https://tiny.pl/dw1dq
[12] Enviro Weather (Pico W Aboard), Pimoroni, https://tiny.pl/dt497
[13] PCF85063A Tiny Real-Time Clock/Calendar with Alarm Function and I²C-Bus, NXP, https://tiny.pl/dt49d
[14] Profilowanie mocy z zastosowaniem Power Profiler Kit II, EP 5/2022, https://tiny.pl/d93rd
[15] Internet Rzeczy w pomiarach środowiskowych (4) Stacja pogodowa Enviro Weather firmy Pimoroni, EP 4/2024, https://tiny.pl/d93r1

Single Pair Ethernet (SPE)

Kluczowa technologia w cyfryzacji naszego świata

Cyfryzacja postępuje w wielu branżach, a Przemysłowy Internet Rzeczy (IIoT) stale się rozwija. W rezultacie rośnie liczba urządzeń komunikacyjnych i związane z tym zapotrzebowanie na bezproblemową i coraz szybszą komunikację. Ethernet od dawna sprawdza się w inteligentnym łączeniu urządzeń w różnych obszarach zastosowań. W przyszłości Single Pair Ethernet nie będzie już tylko rozszerzeniem szeregowych protokołów komunikacyjnych Fieldbus, ale całkowicie je zastąpi. Charakterystyka SPE sprawia, że technologia ta jest innowacyjna w szerokim zakresie aplikacji, a tym samym stanowi prawdziwą wartość dodaną z punktu widzenia najnowocześniejszych infrastruktur komunikacyjnych. Phoenix Contact ma obszerne doświadczenie w wielu branżach i portfolio produktów idealne do wdrożenia Ethernetu jednoparowego w szerokim zakresie zastosowań.

Rozwiązania Ethernet tradycyjnie stosowane w różnych obszarach wymagają zwykle dwóch par przewodów, ale w przypadku Gigabit Ethernet i zwiększonej szybkości transmisji danych mogą potrzebować nawet czterech par. Ethernet jednoparowy (**fotografia 1**) działa tylko z pojedynczą parą przewodów i umożliwia jednoczesną transmisję danych oraz energii. Prędkości transmisji osiągane przez tę technologię – od 10 Mb/s przy maksymalnej odległości transmisji wynoszącej 1000 metrów, do 1 Gb/s przy maksymalnej odległości

Fotografia 1. Phoenix Contact oferuje złącza IP20 zgodne z IEC 63171-2, a także złącza IP67 w konstrukcji M8 i M12 zgodne z IEC 63171-5. Ponadto obecnie trwają prace nad złączami hybrydowymi M12 do Ethernetu jednoparowego zgodnymi z IEC 63171-7

transmisji wynoszącej 40 metrów – okazują się wystarczające do realizacji nawet najbardziej wymagających zadań, takich jak aplikacje intensywnie korzystające z czujników sieciowych ze skanerami lub kamerami (**fotografia 2**). Ethernet jednoparowy okazuje się zatem odpowiedni do stosowania w wielu dziedzinach, które wcześniej zmagały się z ograniczeniami pod względem szybkości transmisji danych, zasięgu i płynności komunikacji.

Połączenia do 1000 metrów, prędkości do 1 Gb/s

Jednym z kluczowych ograniczeń standardowych rozwiązań Ethernet jest maksymalna odległość 100 metrów przy połączeniach punkt-punkt. Aby pokonać większe odległości w systemach przemysłowych, na przykład na liniach produkcyjnych czy przenośnikach

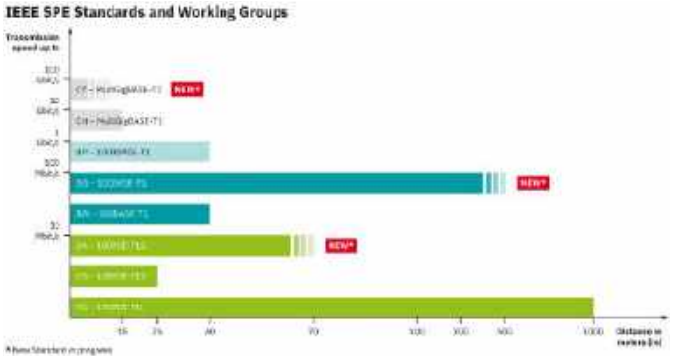
Fotografia 2. Automatyka budynkowa: SPE idealnie nadaje się do łączenia i przekazywania danych generowanych przez wiele instalacji budynkowych

taśmowych, wcześniej trzeba było instalować dodatkowe repeatery lub przełączniki – czyli interfejsy podatne na awarie, które wymagają konserwacji. Technologia SPE umożliwia podłączenie różnych urządzeń na odległość do 1000 metrów z prędkością transmisji 10 Mb/s za pomocą tylko jednego kabla, a także opcjonalne zastosowanie technologii Power over Data Line (PoDL). Oznacza to, że rozwiązania SPE będą mogły w przyszłości zastąpić określone technologie magistral obiektowych.

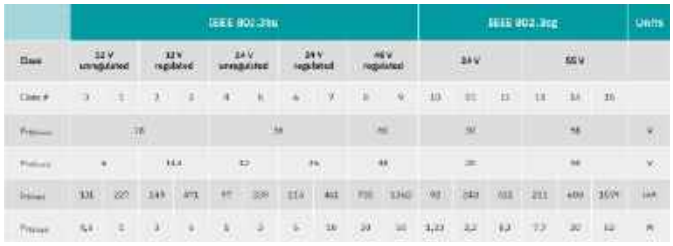
Złożone topologie sieciowe z bramami do łączenia różnych systemów mogą być spójnie konfigurowane przy użyciu Single Pair Ethernet i obsługiwane za pomocą jednolitych usług Ethernet. Dzięki prędkości transmisji od 10 Mb/s do 1 Gb/s SPE może spełnić wymagania szerokiej gamy aplikacji. Co więcej, konsorcja IEEE 802.3 dyskutują obecnie nad kolejnymi standardami SPE przy wyższych prędkościach transmisji danych: 10 Gb/s i szybszych na krótkich odległościach (<15 metrów) – oraz 100 Mb/s lub 1 Gb/s przy odległościach do 500 metrów. Te nowe standardy (**rysunek 1**) otworzą spektrum SPE na jeszcze więcej obszarów zastosowań.

Koszty automatyzacji, oszczędność miejsca i wysoka wydajność

Przeciętna fabryka generuje obecnie około jednego terabajta danych dziennie, a parametr ten stale rośnie. Niezbędna do skutecznej oceny tych danych okazuje się ciągła komunikacja. SPE może zapewnić spójne połączenie sieciowe – od czujnika aż do chmury, w której przechowywane są dane. Ponadto, w świetle rosnącej liczby czujników i inteligentnych urządzeń końcowych używanych w zastosowaniach przemysłowych, SPE oferuje idealne rozwiązanie okablowania – jest proste, bezpieczne, kompaktowe i opłacalne. Podczas tworzenia infrastruktury połączenia SPE staną się w przyszłości znacznie tańsze niż kombinacje komponentów magistrali i ethernetowych urządzeń sieciowych, które obecnie są powszechnie stosowane.



Rysunek 1. Standardy Ethernetu jednoparowego



Rysunek 2. Klasy mocy PoDL

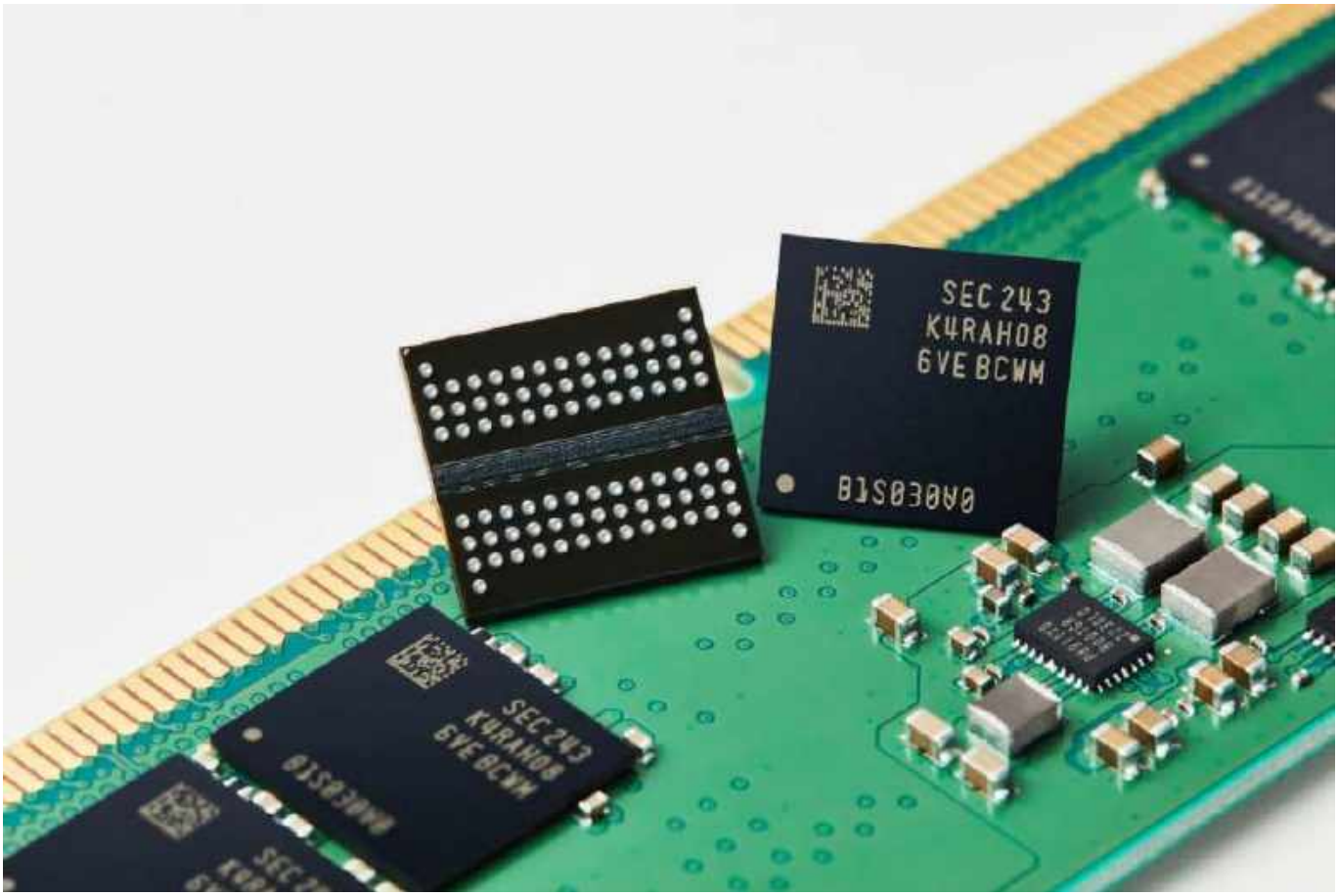
SPE oferuje wiele korzyści wszędzie tam, gdzie stosowane są roboty autonomiczne i współpracujące. Dzięki szybkości transmisji danych wyższej niż w przypadku konwencjonalnych systemów Fieldbus SPE umożliwia robotom i jednostce sterującej komunikowanie się ze sobą z wyższą częstotliwością próbkowania i większym transferem danych. Do tego dochodzi uproszczone okablowanie, z danymi i zasilaniem w jednej linii. Aby spełnić wymagania dotyczące wydajności, które w przyszłości mogą wykraczać poza zdefiniowane standardy PoDL (**rysunek 2**), udostępnione zostaną hybrydowe rozwiązania SPE, które w jednym złączu zawierają zarówno styki danych, jak i zasilania. Zmniejszenie liczby kabli i połączeń skutkuje obniżeniem ryzyka błędów, szybszym rozwiązywaniem problemów i łatwiejszym serwisowaniem. To wszystko przekłada się na znacząco niższe koszty instalacji.

Bezpieczeństwo i wydajność: Technologia APL oparta na SPE

Advanced Physical Layer (APL) jest oddzielnym standardem SPE przeznaczonym do bardzo wrażliwych obszarów automatyki. Jest to idealne rozwiązanie do przemysłu procesowego, ponieważ pozwala sprostać wysokim wymaganiom dotyczącym transmisji danych i zasilania, nawet w obszarach chronionych przed wybuchem (strefy 0, 1 i 2). APL korzysta ze standardu 10BASE-T1L z IEEE 802.3cg wraz z IEC TS 60079-47, 2021-03 (2-WISE) (2-WISE = 2-przewodowym iskrobezpiecznym Ethernetem), a tym samym implementuje metody ochrony przeciwwybuchowej, m.in. iskrobezpieczeństwo. Wśród innych korzyści należy wspomnieć fakt, że technologia ta umożliwia połączenia na dużych odległościach (długość magistrali do 1000 metrów, rozgałęzienia do 200 m), interoperacyjność urządzeń i systemów różnych producentów oraz pozyskiwanie wraz z analizą dużych ilości dodatkowych danych na potrzeby np. konserwacji. W szczególności sektorowi naftowemu, gazowemu i chemicznemu APL dostarcza nowe rozwiązania zwiększające wydajność sieci – a także podnoszące opłacalność modernizacji systemów z integracją istniejącego okablowania i protokołów Ethernet, takich jak EtherNet/IP™, HART-IP, OPC UA i PROFINET.

Istnieje wiele obszarów, w których sieci oparte na protokole IP z SPE mogą być użytecznymi dodatkami lub rozwiązaniami zastępczymi – są to wszystkie aplikacje wymagające spójnej komunikacji opartej na protokole IP na duże odległości i na ograniczonej przestrzeni. Phoenix Contact rozpoczął pracę nad potencjałem Single Pair Ethernet bardzo wcześnie. Dlatego nasi eksperci mają ogromne doświadczenie w zakresie potencjalnych zastosowań oraz wymagań SPE – i są w stanie zapewnić kompleksowe rozwiązania dostosowane do potrzeb odbiorcy. Możliwości tej kluczowej technologii okazują się ogromne i umożliwiają postęp w dziedzinie cyfryzacji. Phoenix Contact dysponuje wiedzą oraz odpowiednimi narzędziami, niezbędnymi do realizacji i integracji Ethernetu jednoparowego w szerokim zakresie dziedzin, aby sprostać wyzwaniom przyszłości.

Paweł Zientarski
Menedżer Obszaru Biznesu
– złącza i obudowy do elektroniki, rynek dystrybucji
pzientarski@phoenixcontact.pl
tel. 694 485 087



Pamięć w modułach SoM i SBC

Miniaturowe systemy jednopłytkowe określane jako SoM (System on Module) lub SBC (Single Board Computer) są oferowane w setkach, jeśli nie tysiącach różnych modeli i wersji. Zasadniczo różnią się one pod względem mocy obliczeniowej, pojemności pamięci i dostępnych interfejsów, ale nie tylko. Wiele nietypowych rozwiązań zostało opracowanych po to, aby dostosować miniaturowy system do specyficznych zadań lub żeby poprawić stabilność działania i niezawodność. W artykule omówimy rodzaje i właściwości układów pamięci stosowanych w nowych typach modułów SoM i SBC.

Każdy system wbudowany (*Embedded System*) musi zawierać procesor główny – CPU, pamięć operacyjną – RAM (*Random Access Memory*) oraz pamięć nieulotną, której zawartość można modyfikować – NVM (*Non-Volatile Memory*). W ostatnich kilku latach obserwowaliśmy intensywny rozwój w obszarze wszystkich trzech wymienionych grup komponentów. Procesory ARM błyskawicznie przeszły z architektury 32-bitowej na 64-bitową oraz z częstotliwości taktowania mierzonych w MHz do poziomu gigaherców, a dodatkowo standardem stały się układy wielordzeniowe (*Multicore Processors*). Najsłabszym ogniwem okazały się pamięci, ale producenci tych komponentów szybko nadrobili zaległości.

Pamięci w systemach embedded

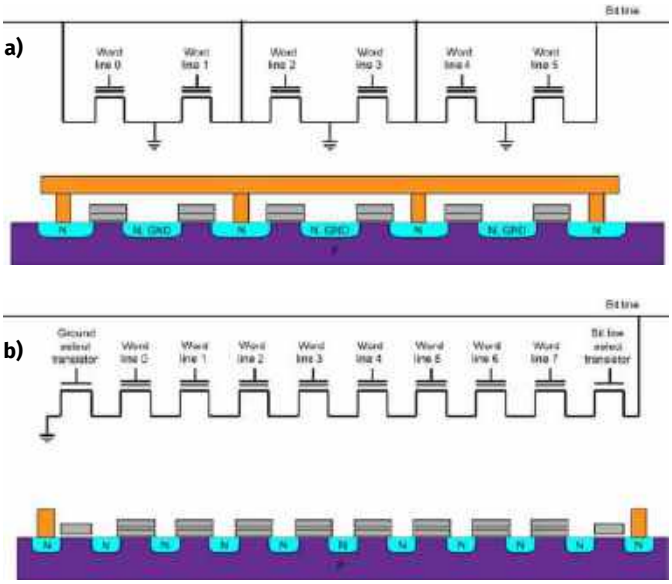
Pierwszym w pełni półprzewodnikowym rodzajem pamięci, która umożliwia budowanie złożonych systemów embedded, jest pamięć Flash. Powstała ona z rozwinięcia technologii EEPROM i – w zależności od budowy komórek pamięci oraz ich wzajemnego połączenia – dzieli się na:

- pamięć Flash typu NOR,
- pamięć Flash typu NAND.

Na **rysunku 1** pokazano uproszczone struktury obu rodzajów pamięci. W NOR Flash każda komórka jest indywidualnie połączona z linią bitową, natomiast komórki NAND Flash są połączone szeregowo z linią bitową. Struktura szeregowa zmniejsza liczbę połączeń, co skutkuje zwiększeniem gęstości układu – przy tej samej technologii procesu pamięć NAND Flash zajmuje o około 60% mniej przestrzeni niż pamięć NOR Flash. W naturalny sposób wpływa to na koszt produkcji (wytwarzanie pamięci NOR Flash jest droższe), ale ma też istotny wpływ na parametry układów.

NOR Flash

Pamięć NOR Flash charakteryzuje się dużą szybkością odczytu i zapewnia swobodny dostęp do każdej komórki pamięci. Możliwe jest bezpośrednie odwołanie do dowolnej lokalizacji w pamięci, bez konieczności adresowania na poziomie bloków. Te dwie kluczowe cechy czynią pamięć NOR Flash idealną do przechowywania kodu programu sterującego, takiego jak BIOS komputera.



Rysunek 1. Uproszczony schemat budowy pamięci Flash: a) typu NOR, b) typu NAND (http://t.ly/nzgmP)

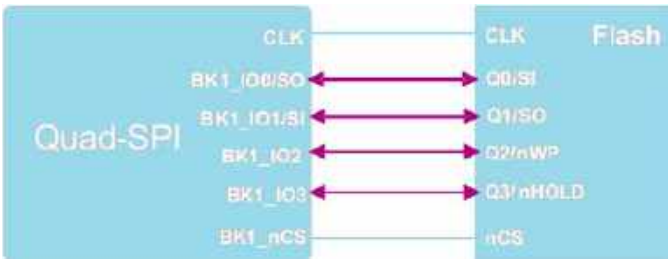
W pamięciach Flash obu typów dane mogą być zapisywane tylko wtedy, gdy komórki zostały wcześniej wykasowane. Kasowanie danych w pamięci NOR Flash przebiega relatywnie wolno – każdy pojedynczy bajt musi być „wyczyszczony". Trwa to nawet kilkaset razy dłużej niż w przypadku NAND Flash – na przykład układ Cypress NAND Flash S34ML04G2 wymaga 3,5 ms, aby usunąć blok o wielkości 128 kB, podczas gdy pamięć Flash Cypress NOR S70GL02GT potrzebuje około 520 ms, aby wykasować sektor o tym samym rozmiarze. To różnica prawie 150-krotna. Wolne kasowanie danych w pamięciach NOR wydłuża również operację zapisu, dlatego układy NOR Flash często oferują rozwiązania wyposażone w funkcję buforowania danych.

Pamięci te pobierają ponadto więcej energii w trakcie zapisywania danych, niż ma to miejsce w przypadku NAND Flash, ale są bardziej energooszczędne w trakcie samego odczytywania danych oraz w trybie standby. Zatem w urządzeniach zasilanych bateryjnie i innych zastosowaniach, w których problematyczne okazuje się zużycie energii, pamięci NOR Flash stanowią doskonałą opcję.

Pamięci z interfejsem szeregowym SPI

Pamięci NOR Flash są dziś dostępne zwłaszcza jako układy z interfejsem szeregowym SPI i oferują pojemności nawet 1 Gb (1 gigabit). Działają przy napięciu 3,3 V lub 1,8 V (**fotografia 1**), a w wersji Ultra Low Voltage – nawet 1,2 V. Zastosowanie interfejsu szeregowego pozwala zmniejszyć rozmiary tych komponentów oraz ich cenę, a jednocześnie umożliwia uproszczenie projektu PCB. Jednak przejście na mniejszą liczbę linii komunikacyjnych oznacza niższą

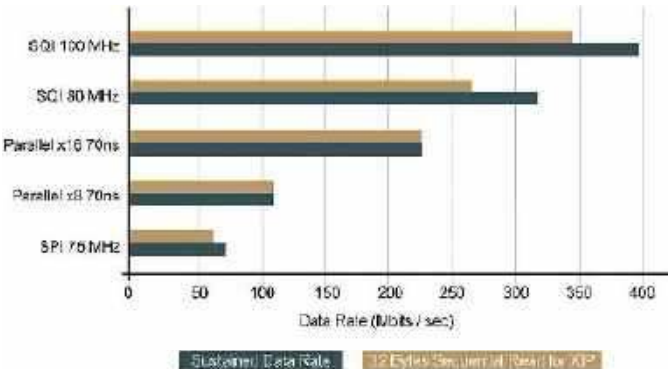
Fotografia 1. Układy pamięci NOR Flash produkcji Winbond (http://t.ly/xvq0r)



Rysunek 2. Interfejs QSPI połączony z pamięcią Flash SQI (http://t.ly/KrRCr)

przepustowość danych. Aby znaleźć kompromis pomiędzy tymi zależnościami, w najnowszych interfejsach szeregowych SPI producenci używają wersji z czterema wejściami/wyjściami danych – QSPI (*Quad SPI*) – dostosowanej do układów pamięci Flash SQI (**rysunek 2**).

Flash SQI oznacza *Flash Serial Quad I/O*, czyli pamięć z 4-bitowym synchronicznym interfejsem szeregowym, charakteryzującym się naprawdę małą liczbą pinów i dużą przepustowością – komunikacja może być synchronizowana zegarem o częstotliwości nawet 100 MHz. Komendy sterujące są bardzo podobne do poleceń SPI, ale zawierają 4-bitowe sekwencje, zamiast jednobitowego strumienia. Dlatego interfejs ten oferuje około czterokrotnie większą przepustowość danych niż klasyczny SPI. W porównaniu do równoległych interfejsów pamięci Flash, SQI Flash zapewnia bardzo dużą wydajność, bez konieczności stosowania skomplikowanych obwodów PCB – **rysunek 3**.



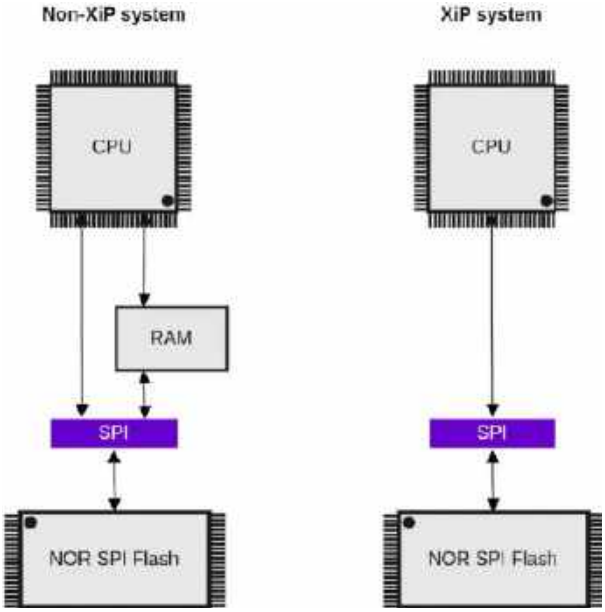
Rysunek 3. Porównanie przepustowości różnych interfejsów komunikacyjnych stosowanych w układach pamięci NOR Flash (http://t.ly/Xec5N)

NOR Flash i XIP

Pamięci NOR Flash przyczyniły się do wdrożenia aplikacji działających w trybie XIP (*Execution-in-Place*). Określenie to odnosi się do wykonywania kodu programu bezpośrednio z pamięci zewnętrznej, bez potrzeby kopiowania do pamięci RAM. Uruchamianie kodu programu z pamięci zewnętrznej umożliwia zwolnienie dodatkowej pamięci operacyjnej na dane dynamiczne. Ideę tego rozwiązania pokazuje **rysunek 4**. Aby XIP był możliwy, pamięć musi oferować dostęp swobodny i wystarczająco dużą przepustowość. NOR Flash doskonale nadaje się do rozwiązań typu XIP, natomiast NAND Flash już nie. Kod programu zawarty w pamięci NAND Flash musi zostać skopiowany do pamięci RAM przed wykonaniem.

NAND Flash

Pamięć NAND Flash przede wszystkim odznacza się większą gęstością upakowania danych, co przekłada się na korzystniejszy stosunek kosztu do pojemności niż w przypadku pamięci NOR Flash. Jednak jest to okupione nieco utrudnionym dostępem do danych – tzw. dostępem sekwencyjnym. Wszystkie operacje odczytu i zapisu wykonywane są na określonych blokach pamięci, a dane do i z bloków przekazywane są sekwencyjnie. Tylko czas kasowania i zapisu



Rysunek 4. Idea działania aplikacji typu XiP (http://t.ly/md1_x)

jest krótszy niż w NOR Flash, ale w niektórych aplikacjach to wystarczy, aby wypadkowy czas dostępu również był korzystniejszy dla NAND Flash.

Działanie sekwencyjne ogranicza zakres zastosowań pamięci tego typu. Jak już wspomniano, wykonywanie kodu programu z NAND Flash jest zwykle możliwe dopiero po wcześniejszym skopiowaniu całości lub wybranych fragmentów do pamięci RAM.

Niezawodność pamięci NAND Flash

Niezawodność zapisanych danych pozostaje ważnym aspektem każdego urządzenia. W pamięciach Flash występuje zjawisko zwane *bit-flipping* – niektóre bity mogą zostać odwrócone. Zjawisko to jest powszechniejsze w NAND Flash niż w NOR Flash.

Ze względu na wydajność procesu produkcji nawet nowe pamięci NAND Flash zawierają uszkodzone bloki (*bad blocks*), które są rozrzucone całkowicie losowo i mogą zajmować nawet 2% pojemności pamięci. Ponadto każdy blok ma określoną żywotność – zwykle jest to ok. 100 000 cykli zapisu. To duża wartość, ale w niektórych aplikacjach wykonujących intensywne działania na pamięci mogą powstawać kolejne uszkodzone bloki i należy zadbać o odpowiednie zarządzanie nimi oraz o implementację systemu korekcji błędów (ECC).

Nowe pamięci NOR Flash są dostarczane bez uszkodzonych bloków, a ze względu na zastosowania tych układów ryzyko powstawania uszkodzeń okazuje się znikome.

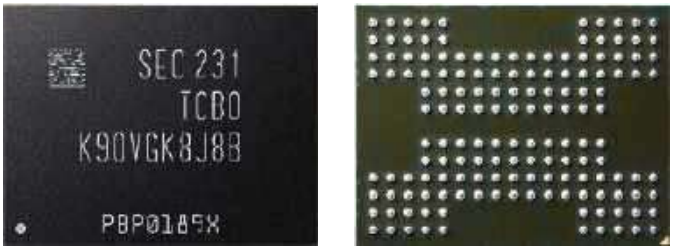
Klasyfikacja pamięci NAND Flash

Generalnie można wyróżnić trzy główne typy pamięci NAND: SLC (*Single Level Cell*), MLC (*Multi Level Cell*) i TLC (*Triple Level Cell*). Pamięci TLC mają większą pojemność (przy porównywalnych wymiarach) niż MLC, które z kolei pod tym względem przewyższają SLC. Oprócz tego każdy typ pamięci NAND ma charakterystyczne zalety i pewne ograniczenia.

W pamięciach SLC każda komórka przechowuje tylko jeden bit informacji, ale pracuje przy niższych napięciach, co wydłuża żywotność nośnika wyrażoną liczbą cykli kasowania/zapisywania. Ograniczeniem układów SLC NAND jest większy koszt w porównaniu z innymi typami pamięci, które charakteryzuje wyższa gęstość upakowania danych.

W pamięciach MLC Flash każda komórka przechowuje dwa bity informacji, ale proces odczytu i zapisu działa wolniej, w porównaniu z pamięciami typu SLC. Główną zaletą jest (nawet kilkukrotnie) niższy koszt bitu danych niż w przypadku SLC NAND.

W pamięciach TLC NAND każda komórka przechowuje trzy bity informacji, ale działa wolniej od omówionych poprzednio, jest bardziej



Fotografia 2. Pamięć Samsung V-NAND Flash 8. generacji, o 236 warstwach i pojemności 1 terabita (http://t.ly/WwjtL)

podatna na błędy i zużycie. Pamięci TLC są wytwarzane wtedy, gdy priorytetem jest koszt, a zatem głównie w elektronice użytkowej.

Ponadto dostępne są pamięci 3D NAND lub V-NAND (Vertical NAND), które mają strukturę warstwową, aby zwiększyć gęstość upakowania danych. Na rynku można spotkać układy o ponad 200 warstwach – **fotografia 2**.

Błędy przechowywania

Dane przechowywane w pamięciach Flash z czasem ulegają samoistnemu uszkodzeniu. Jest to spowodowane utratą ładunku w bramkach tranzystorów tworzących komórki pamięci i ostatecznie prowadzi do zafałszowania zapisanej informacji. Aby przeszkodę tę pokonać, należy problematyczny blok skopiować, a następnie skasować. Warto dodać, że na ten typ błędów bardziej narażone są komórki pamięci z większą liczbą cykli kasowania/programowania. Czynnikiem, który sprzyja zafałszowywaniu zapisanych danych, jest również temperatura – im wyższa, tym większe prawdopodobieństwo jego wystąpienia. Częściej tego typu błędy występują w pamięciach MLC i TLC.

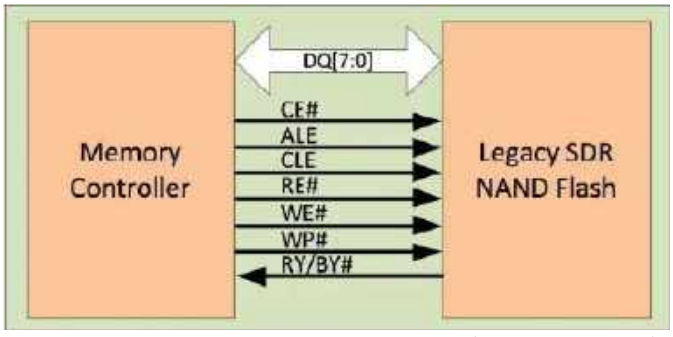
Pamięci NAND Flash zapewniają typową retencję danych na poziomie 10 lat, podczas gdy NOR Flash oferuje czas przechowywania danych na poziomie 20 lat.

Interfejsy pamięci NAND Flash

Pierwsze interfejsy pamięci NAND Flash, znane jako *Legacy*, były przystosowane do transferu asynchronicznego i zawierały sygnały sterujące WE – Write-Enable i RE – Read Enable, bez konieczności stosowania sygnału zegarowego. Interfejs Legacy obsługuje przesyłanie danych przy pojedynczej szybkości transmisji – **Legacy SDR** (*Legacy Single Data Rate*), zatem przesyłanie danych odbywa się tylko na jednym zboczu sygnału sterującego.

Typowe pamięci NAND Flash zawierają 8-bitową lub 16-bitową magistralę danych/adresów z dodatkowymi sygnałami, takimi jak: Chip Enable (CE), Write Enable (WE), Read Enable (RE), Address Latch Enable (ALE), Command Latch Enable (CLE) i Ready/Busy (RB) – **rysunek 5**. Układ sterujący musi dostarczyć polecenie (czytaj, zapisz lub usuń), a następnie adres i dane. Te wszystkie operacje sprawiają, że losowy odczyt danych z NAND Flash był bardzo powolny i maksymalna osiągalna przepustowość wynosiła około 40 MB/s.

Aby przezwyciężyć ograniczenia związane z różnorodnością działania interfejsów w układach NAND Flash, główni producenci pamięci Flash (z wyjątkiem Samsunga i Toshiby) opracowali w 2007 roku



Rysunek 5. Interfejs Legacy pamięci NAND Flash (http://t.ly/kKVPu)

standard **ONFI** (*Open NAND Flash Interface*). Początkowa wersja specyfikacji ONFI miała na celu ujednolicenie przypisania pinów i poleceń pamięci Flash NAND. Interfejs elektryczny ONFI NAND v1.0 jest podobny do Legacy SDR, ale wprowadza opcję obsługi 16-bitowej magistrali danych lub dodatkowej niezależnej 8-bitowej magistrali danych i sygnałów sterujących w celu obsługi do 4 „kości" w jednym pakiecie. Aby połączyć się ze sterownikami pracującymi z różnymi napięciami logicznymi, dodano również opcjonalną szynę napięciową (VDDQ) jako zasilanie wejściowe interfejsu we/wy. Maksymalna osiągalna przepustowość wyniosła około 50 MB/s. Obecnie grupa robocza ONFI nadal publikuje aktualizacje rozszerzające zestaw standardów, a najnowszą wersją roboczą jest ONFI 5.0 z 2021 r. Standard ten określa przepustowość komunikacji do 2400 MT/s (megatransferów na sekundę).

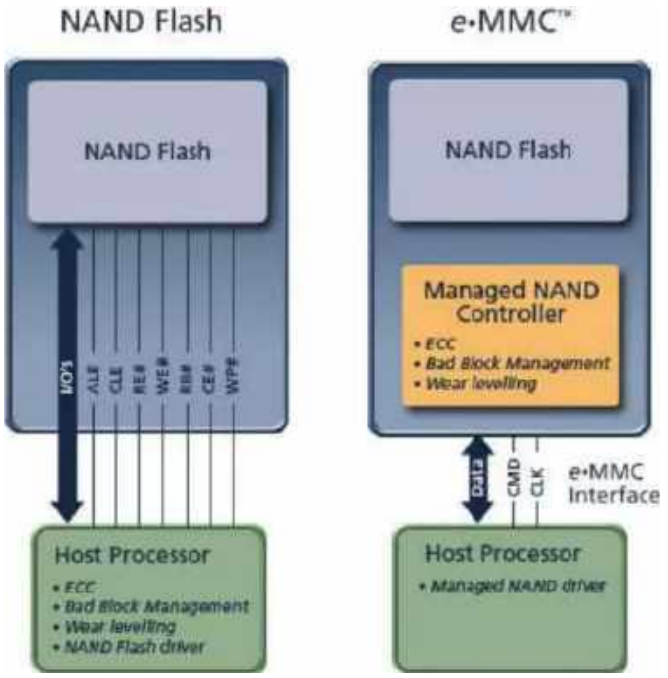
W rok po wydaniu ONFI 1.0, firmy Samsung Semiconductors i Toshiba Memory Corporation (TMC) – wiodący na świecie producenci pamięci flash NAND – wprowadzili standard interfejsu **Toggle** jako alternatywę układów pamięci Flash, których możliwości wykraczały poza standard Legacy. Ewolucja standardu Toggle nastąpiła w ramach wydań specyficznych układów NAND firm Samsung i TMC. Kontrastuje to ze standardem ONFI, który został zaktualizowany w oparciu o ujednolicone publiczne wydania opracowane przez jego grupę roboczą.

Embedded Multi Media Card – eMMC

W pewnym momencie pamięci NAND Flash dostępne były jako układy wykonane różnych technologiach (SLC, MLC, TLC, V-NAND, 3D NAND) i wyposażone w różne interfejsy (Legacy, ONFI, Toggle, QSPI). Potrzebny był ujednolicony standard komunikacyjny pomiędzy hostem a pamięcią – niezależny od właściwości zastosowanych pamięci.

Określenie eMMC (*embedded Multi-Media Card*) odnosi się do miniaturowego systemu – składającego się zarówno z samej pamięci Flash, jak i jej kontrolera, zintegrowanych na wspólnej płytce krzemowej – wyposażonego w wygodny interfejs, charakterystyczny dla kart pamięci MMC. Kontroler obsługuje funkcje korekcji błędów i zarządzania blokami, w tym logiczną alokację bloków i równoważenie zużycia, które wymagają skomplikowanych algorytmów i zależą całkowicie od technologii zastosowanych pamięci NAND Flash. Kontroler obsługuje te funkcje wewnętrznie, dzięki czemu są one niewidoczne dla procesora głównego (hosta). Schemat blokowy systemu eMMC – w porównaniu ze standardową konfiguracją z pamięcią NAND Flash – pokazano na **rysunku 6**.

Standard eMMC został opracowany w 2006 roku przez JEDEC i MultiMediaCard Association. Technologia przeznaczona jest do stosowania w urządzeniach przenośnych (takich jak telefony komórkowe), Internecie Rzeczy (IoT) i innych systemach wbudowanych. Interfejs komunikacyjny zawiera 8-bitową, równoległą magistralę danych oraz sygnał zegarowy (CLK), sygnał sterowania komendami (CMD), strobowanie danych (DS), a także linię zerującą (RST) i kilka linii zasilania – **rysunek 7**. Najnowsza specyfikacja eMMC w wersji 5.1 pozwala na przesyłanie do 400 megabajtów na sekundę (MB/s), co jest porównywalne do dysku półprzewodnikowego z interfejsem SATA. Wszystkie tryby



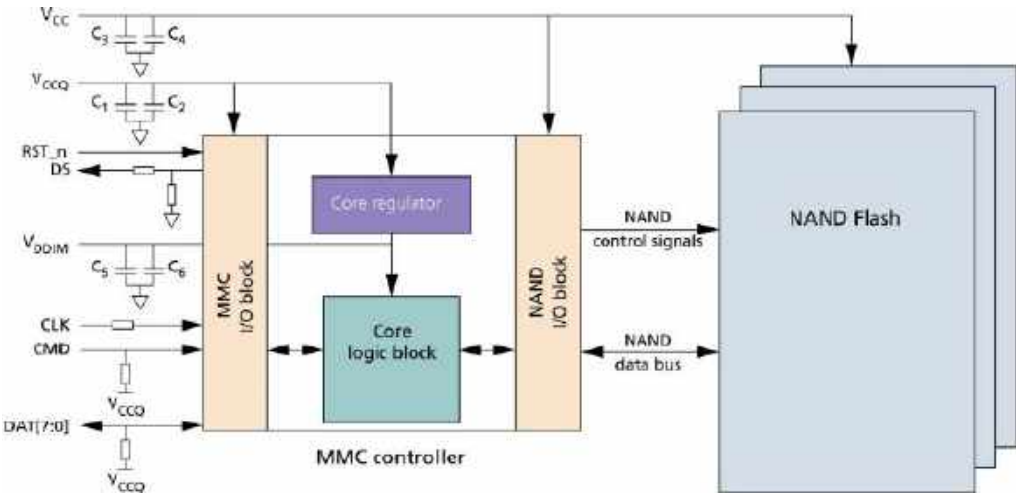
Rysunek 6. Budowa systemu eMMC w porównaniu ze standardową konfiguracją z pamięcią NAND Flash (http://t.ly/lrr0H)

pracy magistrali określone w specyfikacji eMMC 5.1 podsumowano w **tabeli 1**.

Pojemności układów eMMC wahają się od 1 GB do 512 GB (**fotografia 3**) i są dostępne w różnych klasach w zależności od docelowego zastosowania (np. w aplikacjach konsumenckich czy przemysłowych). Biorąc pod uwagę rozmiar, eMMC jest w stanie obsłużyć niezwykle duże ilości danych na tak małej powierzchni.

UFS zamiast eMMC

W 2011 roku JEDEC wydał standard UFS (*Universal Flash Storage*), mający zastąpić pamięci flash eMMC. UFS ma szeregowy interfejs



Rysunek 7. Struktura blokowa układu eMMC (http://t.ly/jUAac)

Tabela 1. Tryby pracy magistrali określone w specyfikacji eMMC 5.1

Tryb	Szybkość (data rate)	Napięcie I/O	Szerokość szyny danych	Częstotliwość CLK	Maksymalna przepustowość szyny danych
Legacy MMC	Single	3,3 V / 1,8 V	1, 4, 8	0...26 MHz	26 MB/s
High Speed SDR	Single	3,3 V / 1,8 V	4, 8	0...52 MHz	52 MB/s
High Speed DDR	Dual	3,3 V / 1,8 V	4, 8	0...52 MHz	104 MB/s
HS200	Single	1,8 V	4,8	0...200 MHz	200 MB/s
HS400	Dual	1,8 V	8	0...200 MHz	400 MB/s

Fotografia 3. Pamięć eMMC 16 GB Samsung (http://t.ly/QFDxi)

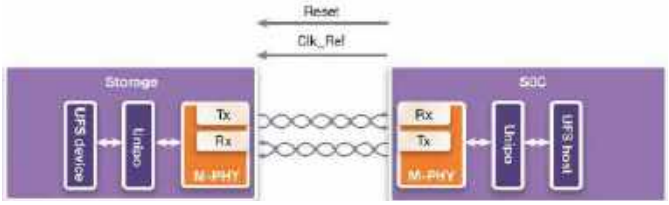
LVDS (*Low-Voltage Differential Signaling*), który zawiera oddzielnie przypisane wyprowadzenia do odczytu / zapisu. Pozwala to na pełną interakcję dwukierunkową (full duplex) – dane do odczytu i zapisu mogą być przesyłane jednocześnie. eMMC ma równoległy interfejs, który może wysyłać dane tylko w jednym kierunku w danej chwili.

Kolejnym istotnym rozwiązaniem jest kolejka poleceń (*Command Queue*), która porządkuje komendy do wykonania. W ten sposób można scalać niektóre polecenia, aby nie powtarzać zadań, a dodatkowo kolejność operacji może zostać odpowiednio zmieniona. eMMC, bez CQ, musi poczekać na proces, który zostanie zakończony, zanim przejdziemy do realizacji następnego.

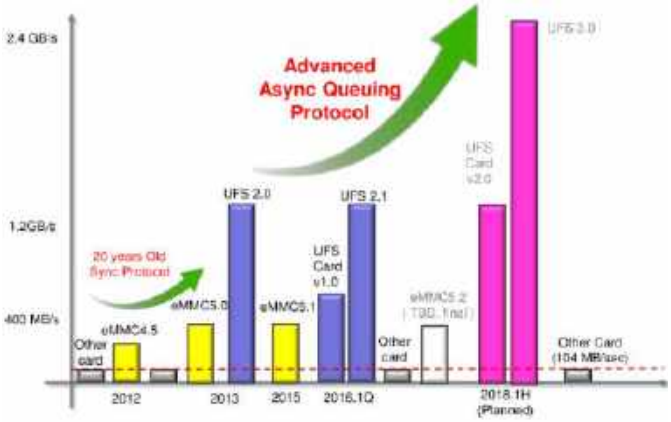
UFS w wersji 2.0 oferuje sekwencyjne prędkości odczytu/zapisu praktycznie porównywalne z dyskami SSD, jednocześnie zachowując niskie zużycie energii, charakterystyczne dla eMMC – **rysunek 9**.

Pamięć operacyjna

Określenie **SDRAM** (*Synchronous Dynamic Random Access Memory*) odnosi się do rodzaju pamięci dynamicznej wyposażonej w interfejs synchroniczny, dzięki któremu wewnętrzne sygnały taktujące



Rysunek 8. Struktura interfejsu UFS (http://t.ly/YQVwq)



Rysunek 9. Porównanie przepustowości interfejsów eMMC oraz UFS (http://t.ly/NvE97)

Fotografia 4. Moduł SoM Raspberry Pi Compute Module 3+, wyposażony w pamięć DDR2 produkcji Micron – układ U2 (http://t.ly/SYEsn)

generowane są z zegara szyny pamięci, co w efekcie pozwala na przyspieszenie transmisji danych. Obecnie jest to najpopularniejsza architektura pamięci RAM, ale produkowana niemal wyłącznie jako pamięć **DDR** (*Double Data Rate*). DDR oferuje zdwojoną przepustowość, ponieważ przesył danych następuje przy obu zboczach sygnału zegarowego – narastającym i opadającym (w przeciwieństwie do **SDR –** *Single Data Rate*).

Każda pamięć ma określoną częstotliwość pracy, czyli rzeczywistą częstotliwość taktowania wyrażoną w MHz. W przypadku pamięci DDR szczególne znaczenie zyskuje częstotliwość efektywna, której wartość odpowiada podwojonej częstotliwości rzeczywistej i wyrażona jest w MT/s (megatransferach na sekundę). Przepustowość jest z kolei wielkością określającą zdolność do przesyłania danych w jednostce czasu – MB/s (megabajtów na sekundę) i uwzględnia częstotliwość efektywną oraz szerokość magistrali danych pamięci.

Pamięć DDR od dawna używana była w komputerach stacjonarnych i przenośnych – w postaci modułów DIMM. Wzrost wymagań stawianych systemom embedded sprawił, że pamięci DDR znalazły się na płytkach modułów SoM i komputerów SBC – **fotografia 4**. Rozwiązania tego typu cechuje relatywnie nieduża przepustowość, ale w zamian gwarantują one niewielkie opóźnienie (czyli krótki czas od wysłania polecenia do pamięci, do uzyskania dostępu do danych) oraz względnie niski koszt produkcji i duże pojemności.

Parametry i generacje

Pamięci SDRAM korzystają z adresowania o strukturze siatki – przecięcie wierszy i kolumn wskazuje konkretny adres pamięci. Z takiej organizacji wynikają cztery podstawowe parametry czasowe (opóźnienia/timingi pamięci/latency), które określają szybkość działania układu:

- **tCAS** lub **CAS Latency** (*Column Address Strobe Latency*) – określa liczbę cykli pomiędzy wysłaniem adresu kolumny do pamięci a momentem otrzymania danych w odpowiedzi, przy założeniu, że właściwy wiersz jest już otwarty. Jest to dokładna wartość, która musi zostać ustawiona w kontrolerze pamięci. CL to jeden z ważniejszych parametrów określających szybkość i wydajność układu pamięci.
- **tRCD** (*RAS to CAS Delay*) – jest to minimalna liczba cykli zegara pomiędzy otwarciem wiersza pamięci a dostępem do znajdujących się w nim kolumn. Czas odczytu pierwszego bitu z pamięci bez aktywnego wiersza równy jest sumie tRCD + CL.
- **tRP** (*Row Precharge Time*) – jest to minimalna liczba cykli zegara potrzebnych do wykonania sekwencji zamknięcia aktywnego wiersza, wstępnego ładowania nowego wiersza i dostępu do nowego wiersza. Czas odczytu pierwszego bitu pamięci z pamięci DRAM z otwartym niewłaściwym wierszem wynosi zatem tRP + tRCD + CL.
- **tRAS** (*Row Active Strobe Time*) – jest to minimalna liczba cykli zegara wymagana pomiędzy aktywacją danego wiersza a jego zamknięciem – pojawieniem się komendy wstępnego ładowania nowego wiersza. Minimalna wartość będzie zawsze większa od sumy opóźnień CL i tRCD.

Rysunek 10. Fragment dokumentacji pamięci DDR produkcji Infineon, w której opóźnienia pamięci podano w formacie CL – tRCD – tRP (http://t.ly/kQCPN)

Opóźnienia dla danej pamięci podaje się zazwyczaj w formacie: CL – tRCD – tRP – tRAS, lub krócej: CL – tRCD – tRP (**rysunek 10**).

Dziś dostępne są pamięci DDR generacji 5. określane skrótem DDR5. Jednak stosuje się je tylko w najbardziej wymagających aplikacjach, podczas gdy w systemach embedded powszechne są pamięci od DDR2 do DDR4. W **tabeli 2** zestawiono podstawowe parametry kolejnych generacji pamięci DDR.

Energooszczędna pamięć DDR

Pamięci RAM przeznaczone do takich aplikacji, jak smartfony, ultrabooki i urządzenia IoT, zostały zoptymalizowane pod kątem zapewniania wysokiej wydajności przy jednoczesnym niskim zużyciu energii. Tak powstały układy LPDDR (*Low Power Double Data Rate*). Zasadniczą różnicą w stosunku do DDR jest tu niższe napięcie pracy, które może wynosić nawet 0,5 V (LPDDR5) – **tabela 3**.

Najnowsza generacja energooszczędnych pamięci – LPDDR5 – stosowana jest przede wszystkim w smartfonach, natomiast szybkie i energooszczędne pamięci LPDDR4 napotkamy w wydajniejszych komputerach SBC – na **fotografii 5** pokazano komputer jednopłytkowy typu TitanSBC wyposażony w procesor i.MX8, współpracujący z 4 GB pamięci LPDDR4.

Tabela 2. Podstawowe parametry różnych generacji pamięci DDR (http://t.ly/yLZtD)

	Rok wydania	Szybkość przesyłania danych [MT/s]	Poziom napięcia [V]
DDR1	1998	266...400	2,5/2,6 V
DDR2	2003	533...800	1,8 V
DDR3	2007	1066...1600	1,35/1,5 V
DDR4	2014	2133...3200	1,2 V
DDR5	2020	3200...6400	1,1 V

Fotografia 5. Komputer jednopłytkowy typu TitanSBC wyposażony w procesor i.MX8 współpracujący z 4 GB pamięci LPDDR4 (http://t.ly/kYJaQ)

Tabela 3. Podstawowe parametry różnych generacji pamięci LPDDR (http://t.ly/yLZtD)

	Rok wydania	Szybkość przesyłania danych [MT/s]	Poziom napięcia [V]
LPDDDR	2008	333...400	1,2 V
LPDDR2	2010	800...1066	1,2 V
LPDDR3	2012	1600...1866	1,2 V
LPDDR4	2014	3200	1,1 V
LPDDR4X	2017	3200...4267	0,6 V
LPDDR5	2020	6400	0,5 V

Dodatkowe właściwości pamięci

ECC/Non-ECC

Pamięć ECC (*Error Checking and Correcrion, Error Correction Code*) jest wyposażona w system kodowania korekcyjnego. Zasada jej działania bazuje na rozbudowie szyny danych pamięci. Dzięki poszerzeniu szyny danych powstaje możliwość przesyłania dodatkowych informacji, w tym przypadku – informacji kontrolnych. Układy ECC pozwalają na znacznie stabilniejsze działanie systemu niż w przypadku pamięci non-ECC – oferują także możliwość korekcji błędów jednobitowych oraz detekcji błędów dwubitowych. Pamięci ECC obsługiwane są jedynie pod warunkiem, że dany system został do tego odpowiednio przystosowany.

Buffered/Unbuffered

Pamięć buforowana/niebuforowana określana jest też jako pamięć *registered/unregistered*. Bufory czy też rejestry, które pojawiają się w module pamięci, służą jako układy podtrzymujące przekazywany sygnał. Ich zadaniem jest synchronizacja przekazu sygnałów, zarówno sterujących, jak i pochodzących z magistrali adresowej do pamięci. Takie działanie zwiększa stabilność systemu. Kosztem tego jest zwiększenie czasu dostępu do pamięci z uwagi na większą liczbę użytych układów.

Podsumowanie

W nowoczesnych systemach embedded, takich jak moduły SoM i SBC, w roli NVM królują układy pamięci eMMC. Pomimo zalet tej technologii, wielu producentów stosuje równolegle drugi typ pamięci, np. QSPI. Jako pamięć operacyjną znajdziemy dziś najczęściej układy DDR4, charakterystyczne dla komputerów typu PC, a procesor główny obowiązkowo powinien mieć wbudowany akcelerator AI – przykład takiego rozwiązania, w postaci modułu VisionSOM-V2L, można zobaczyć na **fotografii 6**. Takie zestawienie komponentów sprawia, że niewielki moduł SoM jest w stanie sprostać naprawdę wielkim wyzwaniom.

Damian Sosnowski, EP

Fotografia 6. Moduł SoM firmy SomLabs wyposażony w pamięć RAM 1 GB typu DDR4, pamięć eMMC o pojemności 32 GB oraz pamięć Flash QSPI (http://t.ly/CVdwB)

Obserwacyjna głowica optoelektroniczna na platformie obrotowo-uchylnej GOSK-2

Optoelektronika do najbardziej wymagających aplikacji

Etronika jest wiodącym prywatnym polskim producentem urządzeń optoelektronicznych dla wojska i służb mundurowych. Jej asortyment – taki jak np. kamery termowizyjne montowane na bezzałogowych statkach powietrznych, czyli tzw. dronach – znajdują zastosowanie również na rynku cywilnym.

Więcej informacji:
Etronika Sp. z o.o.
03-808 Warszawa, ul. Mińska 25
tel. +48 22 870 64 96, biuro@etronika.pl
www.etronika.pl

Kamery termowizyjne

Etronika specjalizuje się w projektowaniu i wytwarzaniu różnorodnych kamer termowizyjnych – zarówno w postaci samodzielnych konstrukcji, jak i w formie specjalnych modułów termowizyjnych, instalowanych wewnątrz innych urządzeń, np. celowniczych głowic optoelektronicznych czy miniaturowych głowic obserwacyjnych na platformie obrotowo-uchylnej. Bazują one na niechłodzonych matrycach bolometrycznych, a także na chłodzonych detektorach. Kamery termowizyjne produkowane przez Etronikę mogą być wyposażone w obiektywy stałoogniskowe o szerokim zakresie kątów pola widzenia oraz w obiektywy zmiennoogniskowe, pozwalające na uzyskanie nawet dziesięciokrotnego powiększenia optycznego obserwowanego obrazu. Sygnał wideo może być przesyłany w sposób analogowy i cyfrowy, np. MIPI CSI-2, SDI, itp. Wszystkie kamery są wytwarzane i testowane zgodnie z polskimi normami obronnymi, a także systemami jakości AQAP 2110:2016 oraz ISO 9001:2015.

Fotografia 1. Kamery termowizyjne KTL

Kamery telewizyjne

Oprócz kamer termowizyjnych, Etronika wytwarza również kamery telewizyjne. Urządzenia te bazują przede wszystkim na kolorowych matrycach CMOS, o przekątnych od 1/3" do 2/3" i rozdzielczości do 4K, z przesłoną typu rolling shutter oraz global shutter, a także z opcjonalnym filtrem IR. Kamery telewizyjne również mogą być wyposażone w obiektywy stało– i zmiennoogniskowe, o parametrach optycznych dostosowanych do potrzeb urządzenia, w którym będą one pracować. Sygnał wyjściowy, podobnie jak w kamerach termowizyjnych, może być wyprowadzony zarówno w postaci analogowej, jak i cyfrowej.

Fotografia 2. Moduł kamery KTL

Fotografia 3. Dalmierz laserowy DL-80

Dalmierze laserowe

W ofercie Etroniki znajdują się także impulsowe dalmierze laserowe. Są one przeznaczone przede wszystkim do optoelektronicznych głowic obserwacyjnych oraz celowniczych i budowane w oparciu o osobne tory: nadawczy oraz odbiorczy. Tor nadawczy składa się z lasera na ciele stałym pompowanym diodowo – o długości fali bezpiecznej dla oka – i z zespołu soczewek odpowiednio formujących wiązkę światła laserowego. Tor odbiorczy to z kolei odpowiedni obiektyw – wraz z półprzewodnikową diodą lawinową APD – oraz analogowo-cyfrowy blok przetwarzania i zliczania sygnałów. Dalmierze Etroniki charakteryzują się rozdzielczością pomiarową 0,5 metra, zasięgiem pomiarowym sięgającym 40 km oraz częstotliwością repetycji pomiaru do 10 Hz w wybranych modelach.

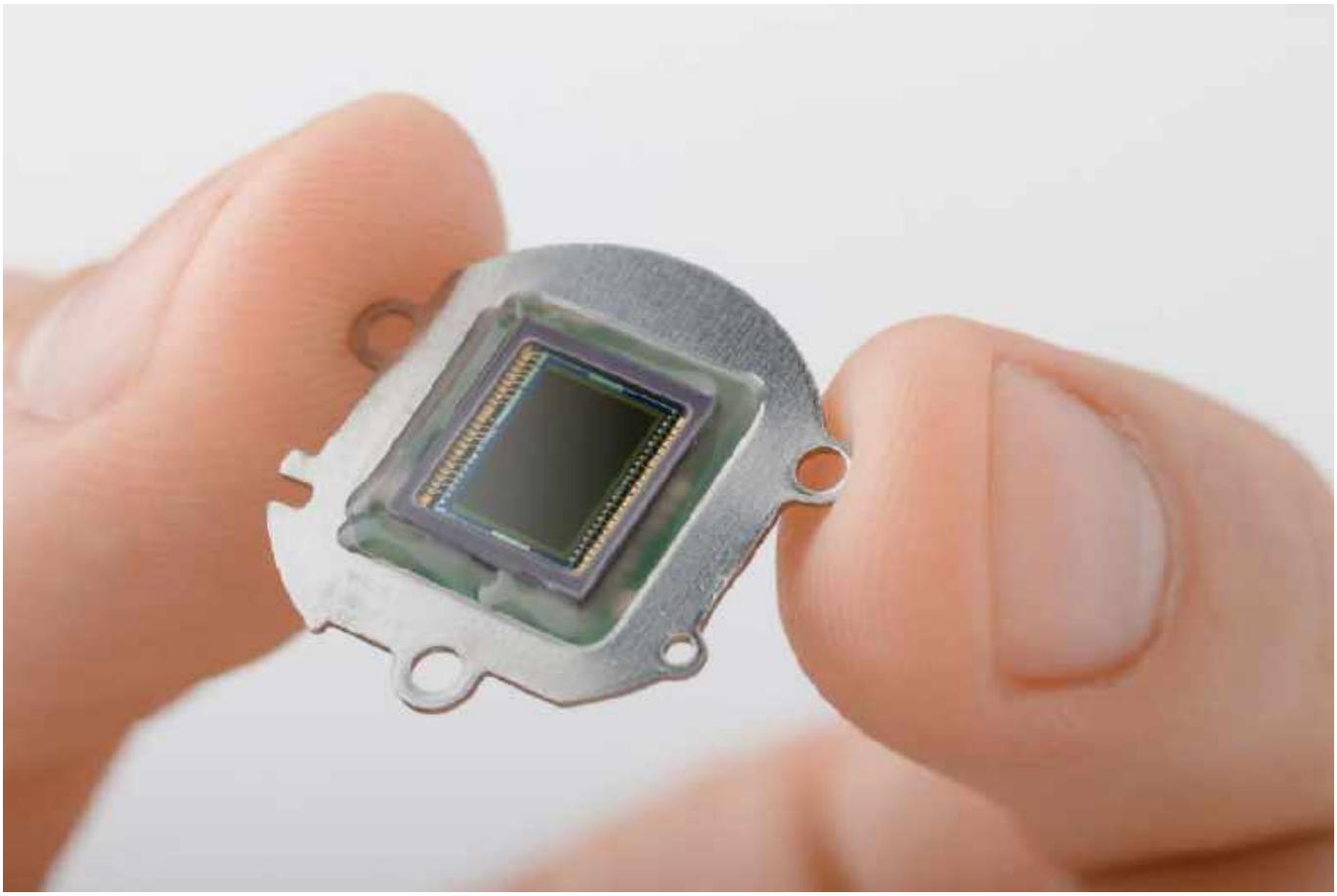
Fotografia 4. Głowica optoelektroniczna KTD-60 Kumak

Fotografia 5. Głowica optoelektroniczna KTD-90.3 Argos

Głowice obserwacyjne i celownicze

Optoelektroniczne głowice obserwacyjne i celownicze to złożone urządzenia, zawierające w sobie co najmniej dwa różne sensory. Sensory te zostają zintegrowane w hermetycznej obudowie z ogrzewanymi oknami, co pozwala na ich długotrwałe stosowanie w różnorodnych warunkach atmosferycznych. Głowice mogą być osadzone na platformach obrotowych bądź obrotowo-uchylnych, dzięki czemu znacząco zwiększa się zasięg obserwacji takiego urządzenia. Różnorodność interfejsów sterowania i wideo pozwala na dopasowanie konfiguracji danej głowicy do konkretnej aplikacji. Przykładowo, typowa konfiguracja sensorów w głowicach celowniczych z rodziny ZIG to dwie kamery telewizyjne o wąskim i szerokim kącie pola widzenia, kamera termowizyjna oraz dalmierz laserowy.

Fotografia 6. Optoelektroniczna głowica celownicza ZIG-T-1L



Czujniki obrazu i przestrzeni

Odbiór bodźców wizualnych (możliwych do zarejestrowania zmysłem wzroku) stanowi nieodłączny element naszej rzeczywistości. W elektronice rolę oczu odgrywają kamery, zaś siatkówkę zastępują matryce światłoczułe, zwane inaczej czujnikami obrazu. Porównując najnowsze modele smartfonów, z których każdy jest dzisiaj wyposażony w trzy, cztery czy nawet pięć kamer, wielu użytkowników zwraca uwagę przede wszystkim na deklarowaną przez producenta rozdzielczość aparatów, a dopiero w dalszej kolejności – na inne cechy, w tym zakres powiększeń zoomu optycznego, osiągi pod względem fotografii w słabym świetle itp. Tymczasem dla nas – elektroników – znaczenie ma szereg zagadnień konstrukcyjnych i mniej oczywistych parametrów, które determinują możliwości zastosowania danego czujnika obrazu w określonych obszarach aplikacyjnych.

Fotografia 1. Vladimir Zworykin – ojciec technologii telewizyjnej – pozuje przy galerii lamp analizujących (http://t.ly/JAYOW)

Fotografia cyfrowa zdominowała współczesny świat w każdym jego aspekcie – począwszy od fotografii amatorskiej, poprzez codzienną pracę profesjonalnych fotoreporterów, aż po badania naukowe (i to zarówno te prowadzone z użyciem mikroskopów optycznych, jak i realizowane za pomocą wartych miliony dolarów, sztucznych satelitów obserwacyjnych). Fascynująca historia urządzeń do przetwarzania obrazu na postać sygnału elektrycznego rozpoczęła się na początku ubiegłego wieku i przez przeszło sześć dekad (od lat 30. do lat 90. XX wieku) opierała się na przeróżnych konstrukcjach lamp analizujących (**fotografie 1** i **2**). Podwaliny pod fotografię cyfrową położył natomiast w roku 1975 Steve Sasson, który – zaledwie dwa lata po przyjęciu do pracy w laboratoriach firmy Kodak – opracował pierwszy na świecie „samodzielny" aparat cyfrowy (warto dodać,

Fotografia 2. 17-milimetrowa lampa analizująca typu Vidicon (http://t.ly/JAYOW)

Fotografia 3. Pierwszy na świecie aparat cyfrowy konstrukcji Steve'a Sassona (http://t.ly/b7vlQ)

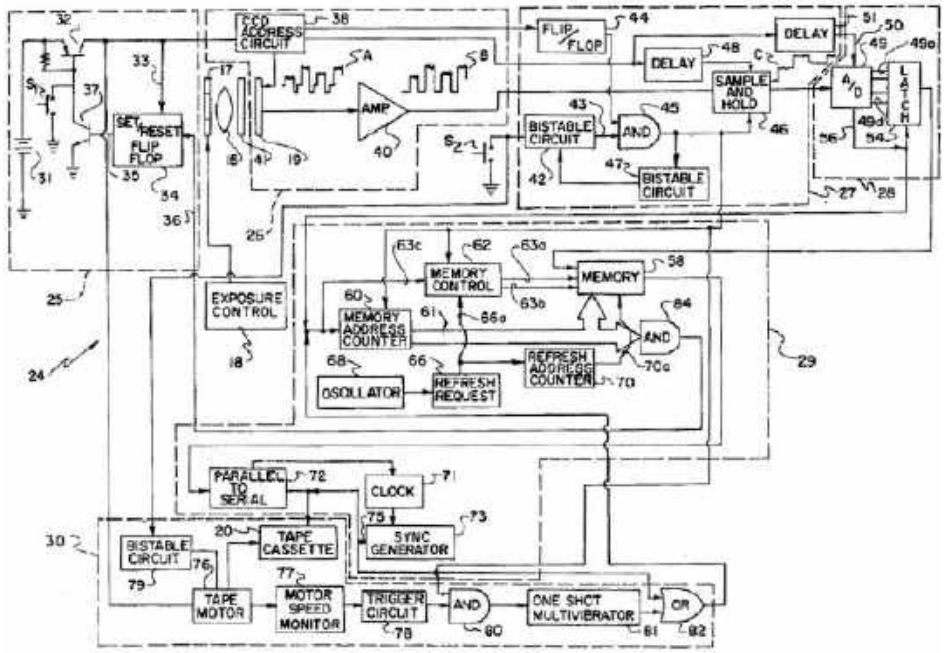
Fotografia 4. Przykładowe czujniki obrazu marki Teledyne (http://t.ly/Hb9Gh)

że sam układ CCD o rozdzielczości 8×1 px powstał zaledwie kilka lat wcześniej w Bell Labs, zaś już w 1975 roku Sasson korzystał z matrycy o rozdzielczości 100×100 px). Funkcję nośnika danych pełniła… zwyczajna kaseta magnetofonowa (**fotografia 3**) – co nie powinno dziwić z uwagi na popularność pamięci taśmowej w dawnych komputerach, choć w połączeniu z (przynajmniej wg dzisiejszej perspektywy) ogromnym aparatem dawało naprawdę uroczy efekt. Czytelników zainteresowanych tym przełomowym wynalazkiem zachęcamy do zapoznania się z patentem nr US4131919 (**rysunek 1**), zgłoszonym do amerykańskiego urzędu patentowego przez Kodaka w roku 1977.

Po niemal pięciu dekadach od powstania tego wiekopomnego wynalazku możemy korzystać z niewiarygodnie szerokiej gamy przetworników obrazu o tak zróżnicowanych parametrach, że naprawdę trudno dziś wskazać obszar aplikacyjny, do którego nie dałoby się dobrać odpowiedniej matrycy z aktualnej oferty producentów optoelektroniki. W artykule dokonamy przeglądu najważniejszych zagadnień z zakresu konstrukcji i parametrów przetworników obrazu oraz przestrzeni, pokażemy także subiektywny wybór najciekawszych komponentów dostępnych na rynku.

Rodzaje matryc obrazowych

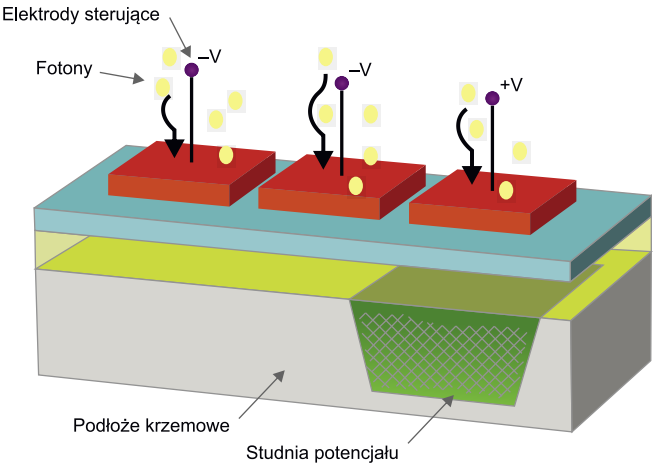
Dywersyfikacja oferty czujników obrazu jest dziś naprawdę ogromna – do grupy tej zaliczają się bowiem nie tyko matryce dwuwymiarowe, ale także liniały CCD i CMOS, stosowane m.in. w skanerach dokumentów, spektroskopach optycznych czy też laserowych czujnikach przemieszczeń. Przykładowe rozwiązania można zobaczyć na **fotografii 4**.



Rysunek 1. Schemat blokowy aparatu cyfrowego wg patentu Steve'a Sassona i Garetha A. Lloyda. Patent amerykański nr US4131919 (http://t.ly/IYXhG)

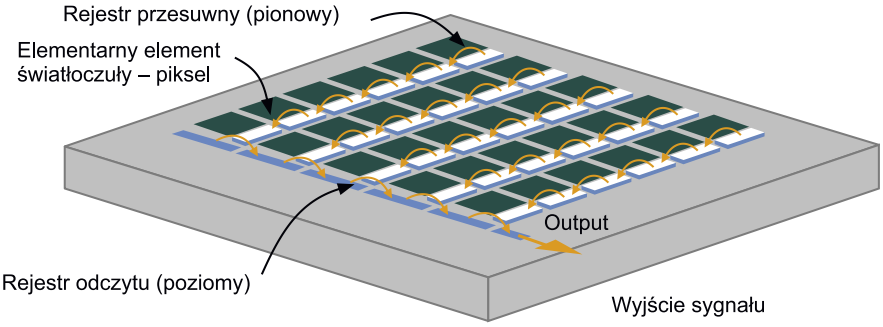
CCD i przyjaciele, czyli ekspresowy przegląd odmian matryc starszej generacji

Rozróżnienie dwóch podstawowych grup matryc światłoczułych jest – przynajmniej z grubsza – znane każdemu elektronikowi. Przetworniki CCD (ang. *charge-coupled device*) zostały opracowane jako pierwsze i przez wiele lat stanowiły fundament fotografii cyfrowej oraz wszelkich innych zastosowań związanych z rejestracją obrazów jedno- i dwuwymiarowych. Podstawa działania matrycy CCD opiera się na gromadzeniu ładunków uwolnionych ze struktury półprzewodnika na skutek zjawiska fotoelektrycznego wewnętrznego. Fotony o odpowiednio wysokiej energii są w stanie „wybić" elektrony, a te zostają następnie



Rysunek 2. Schematyczny przekrój struktury półprzewodnikowej sensora obrazu typu CCD (http://t.ly/mD8Ca)

Fotografia 5. Ilustracja artefaktu określanego jako *smearing*, występującego w matrycach CCD FT (http://t.ly/oyzPe)

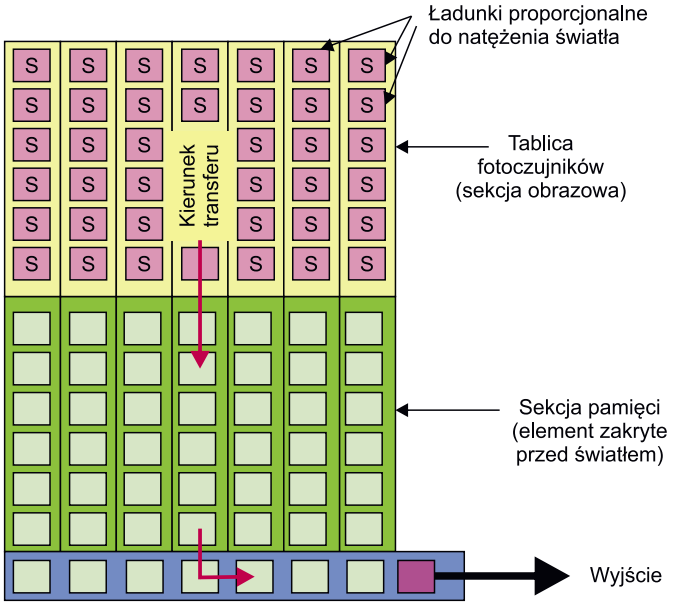


Rysunek 3. Sposób transferu ładunków od poszczególnych pikseli klasycznej matrycy CCD do jej portu wyjściowego (http://t.ly/mD8Ca)

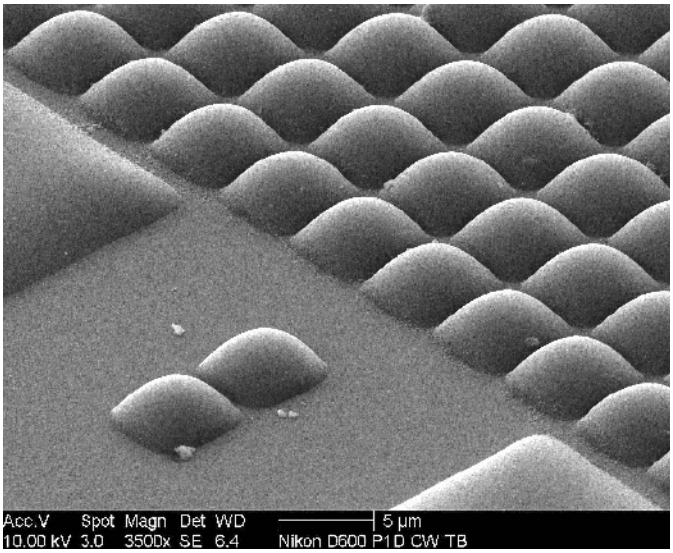
wychwycone i zmagazynowane w tzw. studni potencjału, wytworzonej w półprzewodniku przez polaryzację odpowiednim napięciem (**rysunek 2**). W istocie komórki matrycy można rozpatrywać jako miniaturowe kondensatory MOS – ładunki ze wszystkich kondensatorów w danej kolumnie są kolejno transferowane do wspólnego, przesuwnego rejestru odczytowego, skąd zostają pobrane i zmierzone z użyciem niskoszumnego wzmacniacza ładunku (**rysunek 3**). Warto dodać, że ładunki są „przepychane" przez kolejne pola kolumny dzięki zastosowaniu odpowiedniej sekwencji napięć sterujących, która może występować w wielu odmianach (różniących się liczbą faz). Ponieważ jednak podstawowa konstrukcja matrycy CCD nie sprawdzała się z powodu dość powolnego odczytu (który siłą rzeczy musiał być dokonywany piksel po pikselu, linia po linii) – równolegle ze zwiększaniem rozdzielczości macierzy – wprowadzane były kolejne innowacje, mające na celu przyspieszenie odczytu sygnałów.

Rozwiązanie pośrednie stanowił podział macierzy na kilka sektorów, wyposażonych w niezależne, przesuwne rejestry odczytowe i układy pomiarowe. Jeszcze lepszym wyjściem okazało się natomiast zastosowanie dodatkowej sekcji pamięci, która – odizolowana od wpływu oświetlenia – mogła przechowywać ładunki z poszczególnych pikseli w celu ich odczytania „offline", tj. po zakończeniu ekspozycji. W czasie odczytu właściwa macierz światłoczuła mogła już odbierać kolejną ramkę obrazu. Ten typ matrycy, określany jako FT (ang. *frame transfer* – **rysunek 4**), zapewniał lepszą dynamikę rejestracji szybkozmiennych scen, ale generował kolejne problemy, związane z powstawaniem artefaktów określanych jako *smearing* – czyli wyraźnych smug, rozchodzących się wzdłuż kolumn od punktów obrazu o wysokiej jasności (**fotografia 5**).

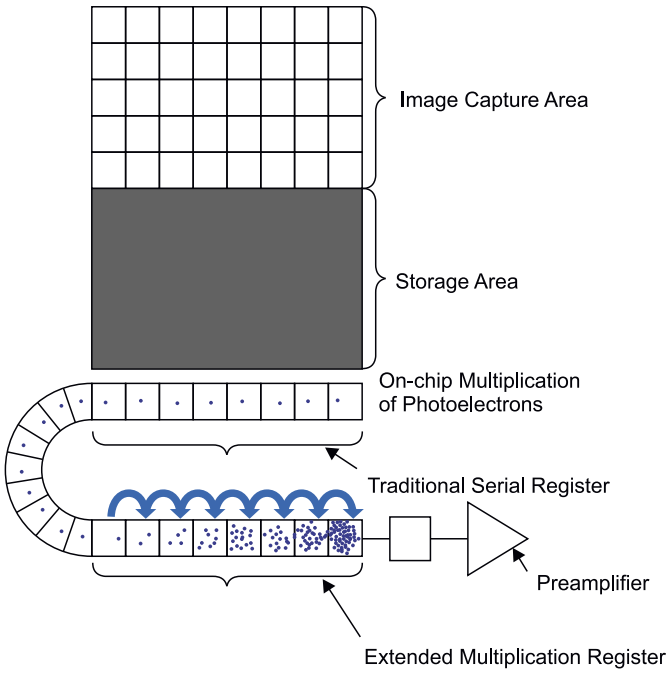
Kolejne „edycje", określane mianem CCD IT (*interline transfer*) czy CCD FIT (*frame interline transfer*), miały na celu dalsze poprawianie osiągów matryc CCD poprzez rozbudowę struktury przełączającej,



Rysunek 4. Zasada działania matrycy CCD typu FT (http://t.ly/mD8Ca)



Fotografia 6. Obraz macierzy mikrosoczewek zarejestrowany przy użyciu skaningowego mikroskopu elektronowego (http://t.ly/mD8Ca)



Rysunek 5. Schematyczna prezentacja zasady działania matrycy EMCCD (http://t.ly/mD8Ca)

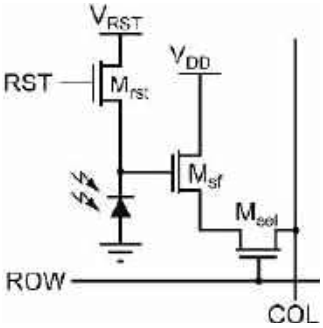
Fotografia 7. Przykładowa kamera typu EMCCD (http://t.ly/f7grW)

współpracującej z właściwymi elementami światłoczułymi. W międzyczasie wprowadzono także struktury CCD HAD (z wysoko domieszkowaną warstwą P fotodiod), których główną zaletą była znacznie lepsza czułość w porównaniu do starszych generacji matryc. Rozwój technologii optycznych umożliwił także wyposażanie matryc w zestawy mikrosoczewek (**fotografia 6**), które pozwalały zredukować wpływ zmniejszenia struktur fotoelementów na czułość detektorów – niestety, jakiekolwiek obwody elektroniczne „wciśnięte" pomiędzy piksele wymuszają redukcję wymiarów właściwych komórek światłoczułych.

Następny przełom stanowiło zatem wdrożenie wyspecjalizowanych matryc wyposażonych w półprzewodnikowy powielacz elektronowy (EMCCD – **rysunek 5**) oraz przetworników typu ICCD, w których obraz jest wzmacniany jeszcze przed trafieniem na właściwą matrycę – to ostatnie rozwiązanie znalazło zastosowanie m.in. w noktowizorach i przyrządach naukowych wymagających ekstremalnie wysokiej czułości na poziomie pojedynczych fotonów. Warto dodać, że w przypadku matryc EMCCD konieczne stało się natomiast zapewnienie chłodzenia (zwykle za pomocą ogniw termoelektrycznych), gdyż tylko w ten sposób można było drastycznie obniżyć poziom szumu, tak istotny w detekcji pojedynczych kwantów promieniowania optycznego – nie dziwią zatem charakterystyczne obudowy niektórych tego typu kamer, będące w istocie masywnymi radiatorami (**fotografia 7**).

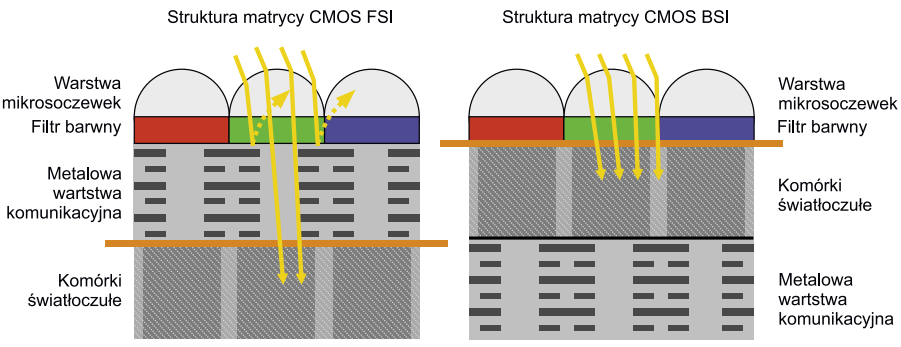
Dalszy rozwój, czyli wprowadzenie matryc CMOS

Prawdziwy przełom w obszarze cyfrowego obrazowania optycznego przyniosło opracowanie matryc CMOS, które w znakomitej większości zastosowań wyparły konwencjonalne czujniki obrazu typu CCD. Choć tutaj także mamy do czynienia z przetwarzaniem natężenia światła na ładunek elektryczny, to sposób zbierania danych z matrycy jest już diametralnie inny. Zamiast „przepychać" ładunek pomiędzy kolejnymi komórkami w poszczególnych kolumnach, czujniki obrazu CMOS korzystają z innej idei: mają bowiem – oprócz właściwych fotoelementów – także wbudowane klucze (w przypadku tzw. matryc pasywnych, tj. PPS – *passive pixel sensor*) lub zespoły złożone z mikroskopijnego wzmacniacza i klucza, choć w istocie zawierające zwykle także dodatkowe tranzystory (APS – *active pixel sensor* – **rysunek 6**). Multipleksowane są zatem napięcia wyjściowe (a nie ładunki) z poszczególnych komórek, dzięki czemu szybkość pracy sensorów CMOS może być naprawdę imponująca. Jednak i tutaj nie obyło się bez problemów, które zwyczajowo rosną w trakcie rozwoju nowej technologii niczym przysłowiowe grzyby po deszczu. Okazało się bowiem, że współczynnik wypełnienia matrycy (czyli stosunek powierzchni aktywnej fotodetektorów do całkowitej powierzchni sensora obrazu) był naprawdę niewielki i wynosił zaledwie 30...50%.

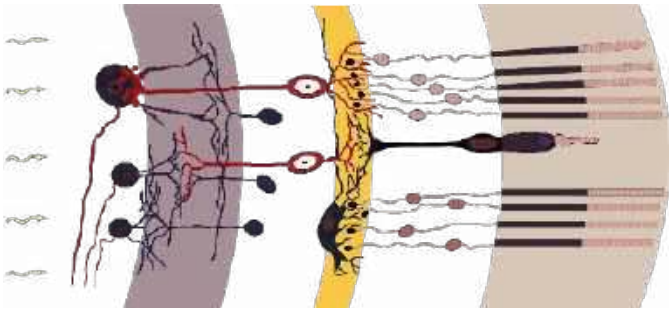


Rysunek 6. Schemat struktury tranzystorowej pojedynczej komórki czujnika obrazu CMOS. Tranzystor M_{rst} służy do „resetowania" bramki tranzystora M_{sf}, czyli zerowania piksela poprzez odprowadzenie zebranego w niej ładunku, M_{sf} pełni funkcję wtórnika napięciowego, zaś M_{sel} odpowiada za funkcję multipleksowania, łącząc wyjście wtórnika z odpowiednią kolumną macierzy (odbywa się to na skutek zasilenia bramki tranzystora M_{sel}, połączonej z linią wiersza matrycy). Źródło: http://t.ly/azAbg

Częściowym rozwiązaniem problemu ograniczonej czułości sensorów CMOS okazała się zmiana podejścia do konstrukcji czujnika – jeżeli bowiem wystawimy na kontakt ze światłem (padającym zwykle z obiektywu, choć oczywiście nie wszystkie matryce pracują w klasycznym układzie znanym z kamer czy aparatów) stronę struktury półprzewodnikowej zawierającą komórki światłoczułe, to ilość przechwyconych fotonów jest znacznie większa, niż w przypadku konstrukcji tradycyjnej (**rysunek 7**). Ze względów technologicznych pierwsze matryce CMOS miały bowiem diametralnie inną budowę, w której warstwy elektronicznego „otoczenia" komórek (czyli tranzystory współpracujące z fotodiodami) oraz metalowe oprzewodowanie macierzy znajdowały się nad właściwymi fotodiodami. Na marginesie warto dodać, że o ile w świecie optoelektroniki pierwsza z opisanych konstrukcji (określana mianem BSI od ang. *back-side illumination*) jest znacznie lepsza od tradycyjnej FSI (ang. *front-side illumination*), o tyle w ludzkim oku budowa siatkówki odpowiada właśnie strukturom FSI,

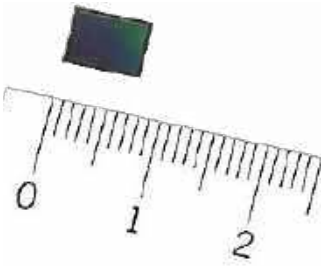


Rysunek 7. Porównanie konstrukcji sensorów CMOS typu BSI oraz FSI (http://t.ly/mD8Ca)



Rysunek 8. Schemat przekroju siatkówki ludzkiego oka jako analogia do sensorów obrazu typu FSI – światło z soczewki, zanim dotrze do właściwych komórek receptorowych (skrajna prawa warstwa), musi przejść przez szereg warstw „pomocniczych" (http://t.ly/MzTMV)

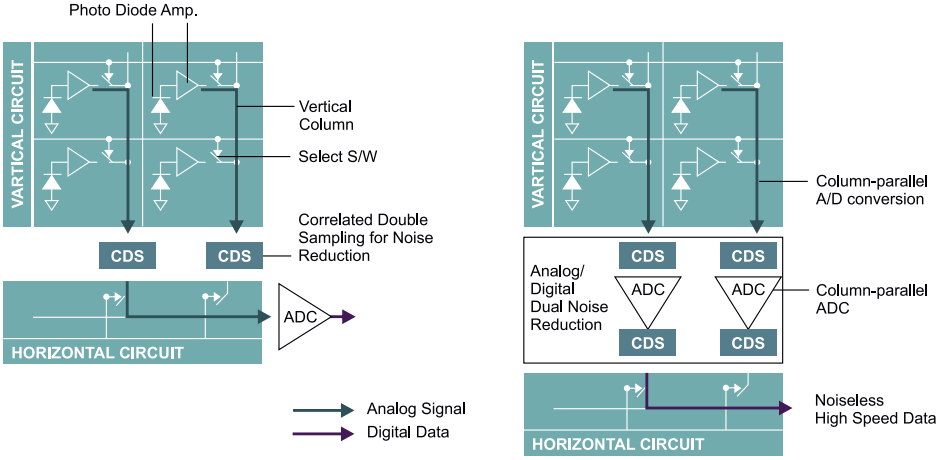
gdyż „lokalna" sieć neuronowa wraz z unaczynieniem znajduje się… nad warstwą komórek receptorowych (**rysunek 8**), co nie przeszkadza nam jednak w uzyskiwaniu doskonałego zakresu dynamiki i fantastycznej jakości widzenia w różnych warunkach oświetleniowych. Ot, kolejny przykład wyższości natury nad technologią.



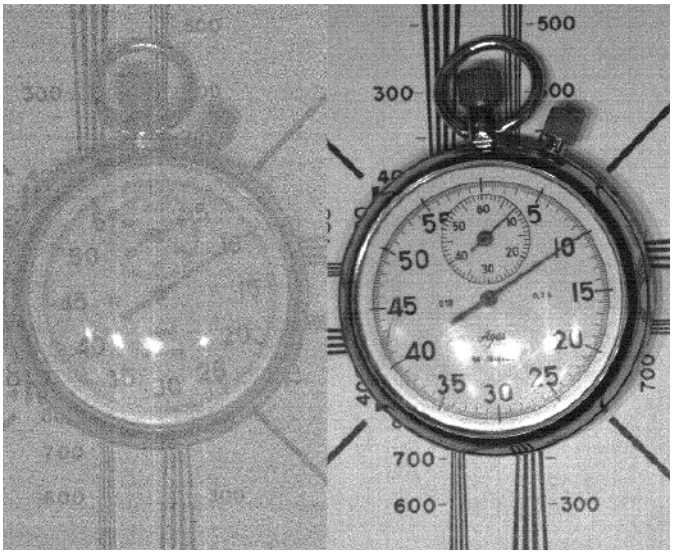
Fotografia 8. 48-megapikselowy czujnik obrazu IMX586 z serii Exmor marki Sony (http://t.ly/k9YdD)

Powróćmy jednak do meritum. Wadą starszych generacji matryc CMOS był dość wysoki poziom szumów, związany m.in. z obecnością dodatkowych układów elektronicznych w torze przetwarzania i przesyłania sygnału analogowego z pikseli do wyjścia czujnika obrazu. Inżynierowie Sony opracowali szereg odmian konstrukcji APS, określanych mianem Exmor generacji 1, 2, 3, 4 i 5. Pierwsze cztery z nich opierały się zarówno na zmianie podejścia do zagadnienia odbioru napięć z poszczególnych obszarów matrycy (innowacją było wprowadzenie niezależnych, podwójnych obwodów redukcji szumu w każdej z kolumn matrycy – **rysunek 9**), zaś 5. generacja wiązała się dodatkowo z przejściem z topologii FSI na BSI (komponenty te zyskały nazwę handlową Exmor R).

Jakość obrazu uległa jeszcze większej poprawie dzięki wdrożeniu technologii Exmor RS, w której całość obwodów „peryferyjnych" poszczególnych komórek umieszczono pod pikselami, w ramach rozbudowanej struktury 3-warstwowej, zintegrowanej dodatkowo z pamięcią DRAM. Dzięki temu współczynnik wypełnienia matrycy przekroczył 80%, co ma niebagatelne znaczenie zwłaszcza w odniesieniu do czujników obrazu o bardzo małych rozmiarach – w przypadku matryc



Rysunek 9. Porównanie klasycznej topologii matrycy CMOS (po lewej) i konstrukcji czujników obrazu Exmor marki Sony (http://t.ly/mD8Ca)



Fotografia 9. Porównanie czułości matrycy CCD (po lewej) i sCMOS (po prawej) w podobnych warunkach słabego oświetlenia fotografowanej sceny (http://t.ly/3JeWH)

Tabela 1. Parametry czujników obrazu Sony IMX586 i IMX582 (https://t.ly/wURXg)

	Sony IMX586	Sony IMX582
Rozmiar sensora	½" (8 mm)	
Rozdzielczość natywna	48 MPx: 8000×6000 px	
Rozdzielczość w trybie łączonych pikseli (Super Pixel Mode)	12 MPx: 4000×3000 px	
Rozmiar piksela (natywny)	0,8 µm	
Rozmiar piksela w trybie Super Pixel Mode	1,6 µm	
Nagrywanie wideo	4K @ 90 fps 1080p @ 240 fps 720p @ 480 fps	4K @ 30 fps 1080p @ 240 fps 720p @ 480 fps

stosowanych w kamerach nowych modeli smartfonów rozmiary pikseli są zwykle na poziomie ułamków mikrometra, zatem gra naprawdę jest warta świeczki. Warto dodać, iż wprowadzona w 2018 roku seria 48-megapikselowych sensorów w technologii Exmor RS dzierżyła podówczas zaszczytne miano rekordu wśród czujników obrazu o najwyższej na świecie rozdzielczości w tym segmencie rynku. Porównanie parametrów obydwu modeli sensora (IMX582 oraz IMX586 – **fotografia 8**) można zobaczyć w **tabeli 1**.

Podobnie jak CDD, także technologia CMOS doczekała się odmian dostosowanych do szczególnie wymagających aplikacji naukowych. Matryce sCMOS (ang. *scientific CMOS*) są swego rodzaju następcami opisanych wcześniej czujników obrazu typu EMCCD – tutaj także możliwe jest osiągnięcie ogromnej czułości (**fotografia 9**), niezwykle niskiego poziomu szumów RMS (równego 1…2 elektrony) i doskonałego zakresu dynamicznego, ale osiągnięcie takich parametrów mocno odbija się na cenie kamer sCMOS. Producenci stosują w nich bowiem rozmaite, często niezwykle zaawansowane technologie, w tym:

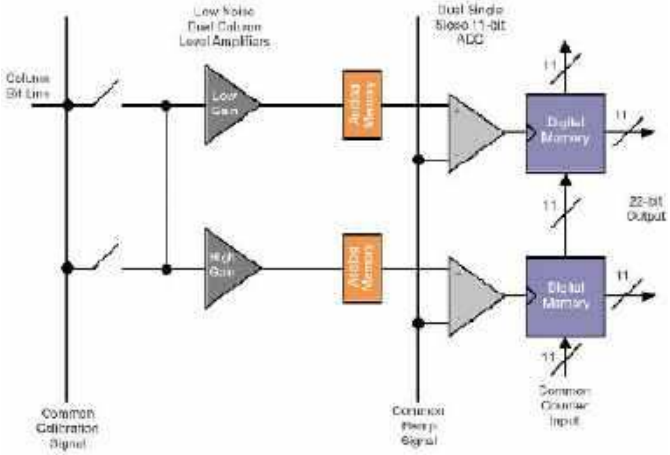
- wbudowane chłodzenie termoelektryczne, wymagające rzecz jasna efektywnego chłodzenia gorącej strony ogniw (**fotografia 10**),
- bariery próżniowe zapobiegające kondensacji pary wodnej,

Fotografia 10. Przykładowa kamera sCMOS marki Thorlabs (http://t.ly/Fy2ba)

- macierze FPGA kompensujące najsubtelniejsze różnice produkcyjne pomiędzy poszczególnymi pikselami matrycy,
- systemy podwójnych torów analogowo-cyfrowych, umożliwiających pomiar każdego piksela za pomocą dwóch niezależnych układów o małym i dużym wzmocnieniu (dane uzyskane z obydwu ramek są następnie „składane" w jedną całość – **rysunek 10**),
- wysokorozdzielcze, niskoszumne przetworniki ADC (np. 16-bitowe).

Z cech zaprezentowanych do tej pory konstrukcji czujników obrazu wynikają najważniejsze różnice użytkowe pomiędzy matrycami CCD i CMOS. Zbierzmy je zatem w jednym miejscu:

- **Szybkość** – wąskim gardłem matryc CCD jest konieczność odczytu całej macierzy (lub przynajmniej znacznej jej części) przez pojedynczy tor ze wzmacniaczem ładunku; w przypadku matryc CMOS proces ten jest realizowany sekwencyjnie przez zestaw pracujących jednocześnie torów, co znakomicie przyspiesza akwizycję ramek obrazu.
- **Szum** – poziom szumu w standardowych matrycach CMOS jest nierzadko wyższy niż w czujnikach obrazu CCD, co wynika z bardziej rozbudowanej struktury oraz innego sposobu pomiaru komórek fotoczułych.
- **Koszt produkcji** – macierze CMOS są w ogólności tańsze niż czujniki CCD o podobnych parametrach, rzecz jasna porównanie to ma sens tylko w przypadku konstrukcji standardowych (nie zaś EMCCD czy sCMOS).
- **Pobór mocy** – sensory CMOS, działające w oparciu o multipleksowanie wstępnie przygotowanych sygnałów napięciowych (zamiast



Rysunek 10. Konstrukcja dwutorowego układu pomiarowego współpracującego z komórkami macierzy sCMOS marki Andor (http://t.ly/twP3u)

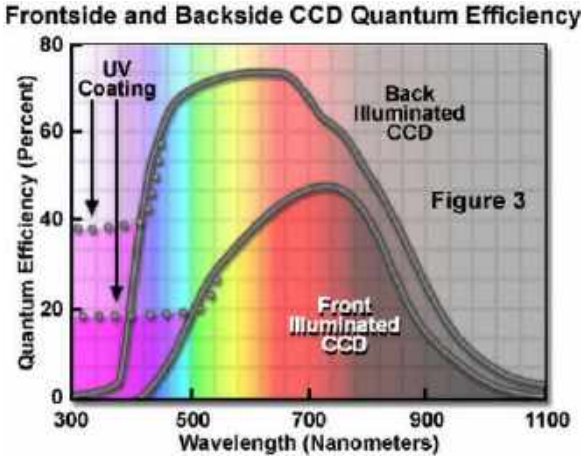
Fotografia 11. Duża, 100-megapikselowa matryca CCD marki Andanta do specjalistycznych zastosowań naukowych (http://t.ly/joBNW)

„przepychania" ładunków pomiędzy kolejnymi studniami potencjału), są znacznie bardziej energooszczędne niż porównywalne pod względem rozmiarów i rozdzielczości matryce CCD.

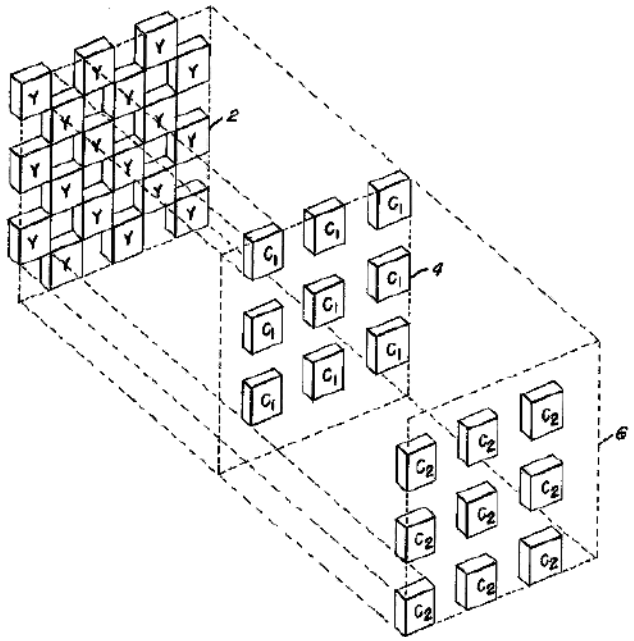
- **Rozdzielczość** – w technologii CMOS łatwiej jest wyprodukować sensory o bardzo wysokiej rozdzielczości, podczas gdy w przypadku CCD szereg ograniczeń technologicznych (w tym bodaj najpoważniejsze, czyli wspomniane „wąskie gardło sygnałowe") sprawia, że użyteczna rozdzielczość ulega zwykle ograniczeniu, choć na świecie istnieją specjalistyczne wykonania CCD o rozdzielczości dochodzącej nawet do 100 megapikseli (!) – przykład można zobaczyć na **fotografii 11**.
- **Podatność na artefakty** – jak już napisaliśmy, matryce CCD są podatne na artefakty obrazowe wynikające z efektu „przelewania" ładunku z przepełnionych studni potencjału na otaczające piksele, a także z zaburzeń powstających podczas transferu ładunków (w przypadku matryc typu FT). Problemów tych pozbawione są natomiast macierze CMOS, rzecz jasna z uwagi na niezależną pracę poszczególnych pikseli (ładunki są „mierzone" lokalnie, a nie przesuwane z piksela do piksela).
- **Zakres dynamiki** – współczesne matryce CMOS oferują nieporównanie lepszy zakres dynamiczny niż konstrukcje sprzed kilku czy kilkunastu lat. Co więcej, mają one także wysoką tolerancję na efekt saturacji, tj. lepiej niż czujniki CCD radzą sobie z bardzo jasnymi partiami obrazu – przekroczenie maksymalnej mierzalnej jasności powoduje bowiem po prostu nasycenie toru pomiarowego (tak jak w każdym przetworniku ADC), ale nie skutkuje ono „zalewaniem" otaczających pikseli nadmiarowymi fotoelektronami.

Matryce monochromatyczne i kolorowe

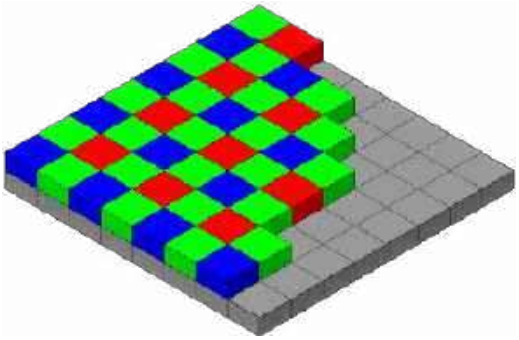
Warto zwrócić uwagę na fakt, że podstawowa struktura czujnika obrazu CMOS bądź CCD nie jest w stanie rozróżnić barwy światła. Rzecz jasna, z samej natury zjawisk kwantowych zachodzących



Rysunek 11. Porównanie wydajności kwantowej matryc CCD o konstrukcji typu BSI oraz FSI (materiały firmy Hamamatsu, http://t.ly/16TQZ)

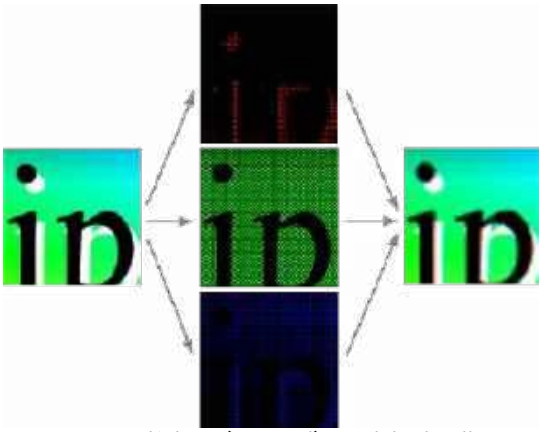


Rysunek 12. Ilustracja z opisu patentowego nr US3971065 prezentująca budowę filtru barwnego Bayera (http://t.ly/BTO6K)



Rysunek 13. Ułożenie filtrów RGB w topologii Bayera (http://t.ly/l4ELo)

w półprzewodniku poddanemu działaniu światła wynika pewna określona dla danego materiału charakterystyka widmowa (**rysunek 11**), ale manifestuje się ona przez zmienną czułość na promieniowanie o różnych długościach fali. Aby rozróżnić rzeczywistą barwę obrazu (przynajmniej w czytelnej dla nas oraz dla maszyn formie), konieczne jest zatem wyposażenie matrycy w zestaw odpowiednich filtrów barwnych, które przepuszczać będą tylko określony wycinek widma, pochłaniając przy tym fale krótsze oraz dłuższe od granic pasma przepustowego.

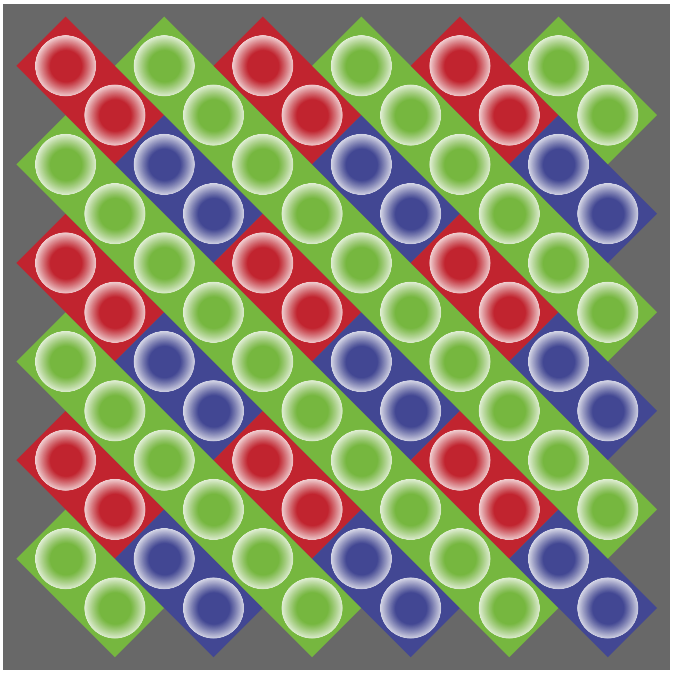


Rysunek 14. Obraz wejściowy (po lewej), po digitalizacji za pomocą sensora z filtrem Bayera, jest zapisywany w postaci trzech „niekompletnych" macierzy R, G i B (środkowe rysunki), na podstawie których algorytm demozaikowania odtwarza odpowiednio zinterpolowany obraz barwny w przestrzeni RGB (po prawej). Na podstawie http://t.ly/s-8Iv

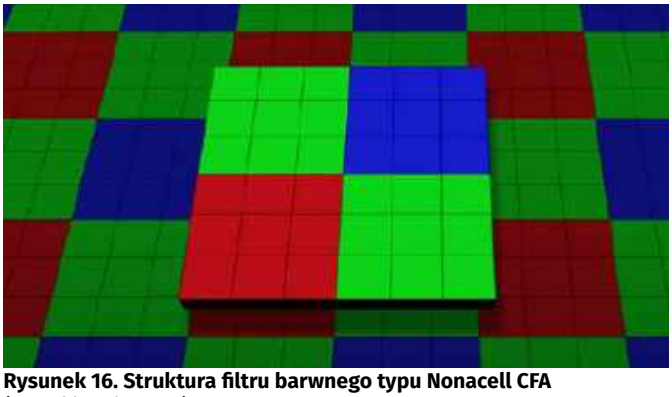
Fotografia 12. Przykłady artefaktów powstałych w wyniku demozaikowania algorytmami nieprzystosowanymi do pracy z obrazami o wysokiej częstotliwości przestrzennej szczegółów. Lewy obraz: demozaikowanie algorytmem najbliższego sąsiada – widoczne zniekształcenia chromatyczne i efekt *zippering* (kolorowego, naprzemiennego zafalowania krawędzi); środkowy obraz: efekt uzyskany przy zastosowaniu algorytmu interpolacji dwuliniowej, widoczne zniekształcenia kolorystyczne; prawy obraz: efekt uzyskany w wyniku demozaikowania algorytmem typu *adaptive homogeneity-directed* (http://t.ly/oenEQ)

I tutaj pojawia się pierwszy problem – mamy trzy barwy podstawowe, a zatem ułożenie sąsiadujących ze sobą filtrów w postaci ortogonalnej siatki musi prowadzić do pewnych dysproporcji w „gęstości" poszczególnych komórek. W amerykańskim zgłoszeniu patentowym nr US3971065 Bryce E. Bayer uwidocznił strukturę filtru barwnego (**rysunek 12**), która do dziś stanowi fundament obrazowania cyfrowego w przestrzeni RGB – charakterystyczne ułożenie filtrów czerwonych, zielonych i niebieskich zakłada, że tych drugich jest w macierzy 2-krotnie więcej niż każdego z pozostałych (**rysunek 13**), a inspiracją do takiego, a nie innego potraktowania zagadnienia filtracji optycznej była – znów – ludzka siatkówka, szczególnie czuła właśnie na światło zielone.

Drugi problem (podobnie jak i kolejne) wynika natomiast ze… sposobu rozwiązania pierwszego z nich. Zwróćmy bowiem uwagę na fakt, iż efektem działania filtru Bayera są trzy obrazy barwne, prezentujące jednak nieznacznie przesunięte widoki obrazowanej sceny (**rysunek 14**)! Mało tego, pikseli zielonych jest tyle, co wszystkich pozostałych razem wziętych… Aby uzyskać sensowny obraz z połączenia trzech powyższych, trzeba więc potraktować dane wyjściowe z matrycy odpowiednim algorytmem, określanym mianem

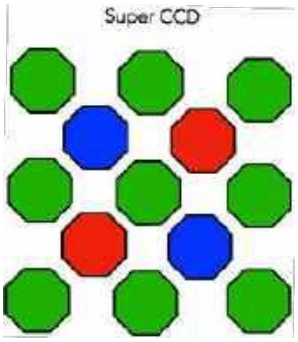


Rysunek 15. Struktura filtru barwnego typu Fujifilm EXR (http://t.ly/gsvaE)



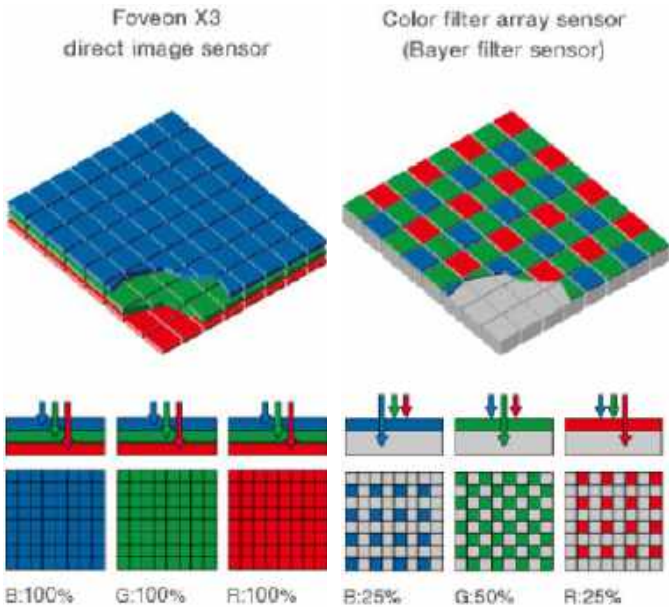
Rysunek 16. Struktura filtru barwnego typu Nonacell CFA (http://t.ly/bSTbh)

demozaikowania. W najprostszej postaci może on wykonywać zwyczajną interpolację wartości dwóch brakujących kanałów barwnych dla każdego piksela – przykładowo, aby wyznaczyć wartość określonego punktu, który w rzeczywistości współpracuje z filtrem zielonym, algorytm uzupełnia dane tegoż punktu poprzez interpolację dwóch otaczających pikseli czerwonych i wykonuje tę samą czynność w odniesieniu do najbliższych punktów niebieskich. Takie podejście sprawdza się dobrze, ale… tylko przy w miarę jednolitych obszarach barwnych, gdyż w przypadku bardziej szczegółowych rejonów sceny (np. krawędzi) może prowadzić do powstawania widocznych artefaktów (**fotografia 12**). Z tego względu właśnie oprogramowanie do obróbki plików RAW stosuje rozmaite, znacznie bardziej zaawansowane algorytmy, których celem jest poprawa jakości obrazu – zawsze jednak to, co oglądamy, jest w pewnym sensie wytworem komputera, gdyż nie istnieje ani jeden punkt na obrazie, dla którego wszystkie trzy wartości R, G i B zostałyby rzeczywiście zmierzone jednocześnie.



Rysunek 17. Struktura filtru barwnego typu Super CCD (http://t.ly/n0NJO)

W międzyczasie powstał także szereg innych systemów ułożenia filtrów barwnych, np. Fujifilm EXR (**rysunek 15**), Nonacell CFA (**rysunek 16**) czy też Super CCD (**rysunek 17**). W każdym z tych przypadków nadal istnieje rzecz jasna konieczność stosowania algorytmów demozaikujących, ale poszczególne filtry lepiej lub gorzej radzą sobie np. w kwestiach odporności na artefakty czy też pracy w słabym świetle. Dotyczy to zwłaszcza układu typu Nonacell, w którym sąsiadujące piksele czułe na tę samą barwę mogą być „sklejane" w celu uzyskania większego, a – co za tym idzie – także bardziej czułego na słabe światło punktu obrazu, co ma duże znaczenie w zdjęciach nocnych, wykonywanych za pomocą aparatów o ogromnej rozdzielczości (np. 108 Mpx).



Rysunek 18. Porównanie konstrukcji klasycznego, planarnego czujnika obrazu z filtrem Bayera oraz unikatowej topologii wielowarstwowej, zastosowanej w sensorach Foveon X3 (http://t.ly/-nk6J)

Warto także dodać, że powstała odmiana czujnika obrazu, w której wszystkie trzy fotodiody „kanałowe" są ustawione jedna na drugiej w sposób wielowarstwowy – matryca Foveon X3 (**rysunek 18**) całkowicie eliminuje potrzebę stosowania demozaikowania, zmniejsza także znacząco liczbę artefaktów na krawędziach obrazu. I znów rozwiązanie jednego problemu generuje nowe trudności: kolejne warstwy nie mogą bazować na klasycznych filtrach (w przybliżeniu) monochromatycznych, gdyby bowiem pierwszy, niebieski filtr zaabsorbował fale o długościach wszystkich innych niż zakres niebieski, to… do czujników leżących poniżej nie dotarłoby już żadne promieniowanie spoza tegoż przedziału. Inżynierowie z firmy Foveon podeszli do tematu inaczej, niejako odwrotnie – filtry optyczne mają bowiem charakterystyki pasmowo-zaporowe (tj. absorbują tylko światło – kolejno: niebieskie, zielone i czerwone), a zatem informacje uzyskane z leżących coraz głębiej fotodiod muszą być traktowane nie jako pozyskane subtraktywnie, ale addytywnie. Algorytm musi więc przeliczyć sygnały z trzech kanałów na – odpowiednią do przyjętej powszechnie w świecie fotografii cyfrowej – przestrzeń barwną RGB. Czujniki obrazu typu Foveon X3 nie przyjęły się jednak na rynku w sposób mogący jakkolwiek zagrozić klasycznym matrycom opartym na filtrach o topologii Bayera – oprócz kilku modeli lustrzanek Sigmy oraz aparatów kompaktowych produkcji Sigmy i Polaroida, próżno szukać w rynkowej ofercie konstrukcji bazujących na tym (bądź co bądź innowacyjnym) sensorze obrazu. I choć szerokim echem odbiły się w świecie fotografów plany premiery nowej konstrukcji Foveona na początku obecnej dekady, to jak na razie projekt wciąż pozostaje w domenie szumnych zapowiedzi, a nie komercyjnej oferty któregokolwiek z producentów aparatów cyfrowych.

Rodzaj migawki

Do tej pory omówiliśmy zagadnienia związane z szeregiem najważniejszych parametrów przetworników obrazu, zabrakło jednak jednej, równie istotnej cechy konstrukcyjnej, która w dużej mierze warunkuje parametry użytkowe matryc w aplikacjach wymagających krótkich czasów ekspozycji. Mowa o migawce elektronicznej.

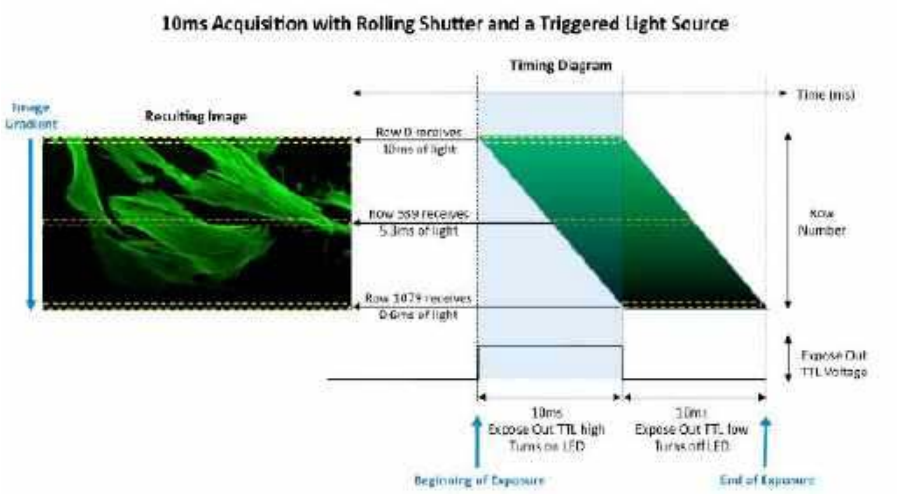
W opisach technicznych przetworników obrazu znajdziemy dwa rodzaje migawek.

- Migawka typu ***global shutter*** pozwala na zarejestrowanie obrazu ze wszystkich pikseli jednocześnie. Dzięki temu uzyskujemy cyfrową reprezentację obrazu sceny, która odpowiada faktycznemu jej wyglądowi w pewnym skończonym przedziale czasowym – rzecz jasna, im krótszy czas ekspozycji, tym mniejsza podatność na rozmazywanie konturów, ale także mniejsza liczba fotonów, które zostaną zarejestrowane przez poszczególne piksele.
- Migawka typu ***rolling shutter*** działa na drodze sukcesywnego przemiatania kolejnych linii przetwornika w czasie ekspozycji. Oznacza to, że zanim układ odczytowy „dojdzie" do końca obrazu, to szybkozmienna scena może już zdążyć ulec zmianie (np. przesunięciu).

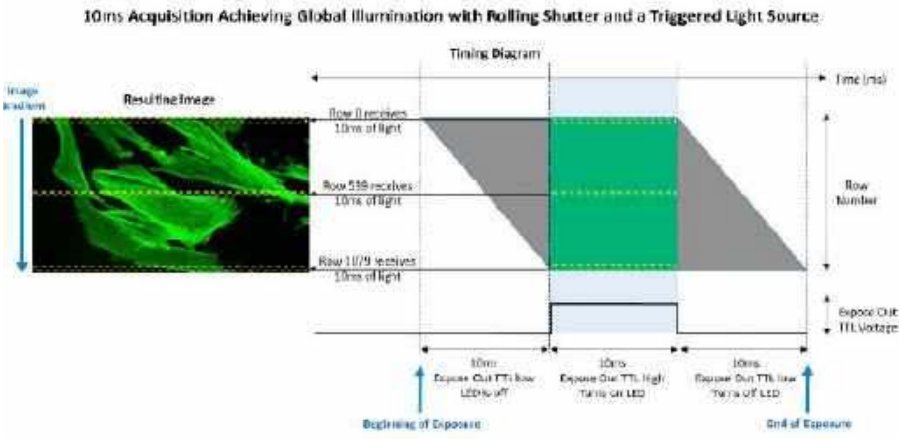
Nietrudno domyślić się, że w przypadku obrazów statycznych lub w czasie rejestracji zjawisk o małej dynamice typ migawki nie ma praktycznie żadnego znaczenia dla obrazu wynikowego. Jeżeli jednak fotografujemy lub nagrywamy zjawiska zmieniające się bardzo szybko, to zastosowanie przetwornika z migawką typu *rolling shutter* doprowadzi niechybnie do powstania artefaktów. Podręcznikowy

przykład można zobaczyć na **fotografii 13** – wirujące, sześciołopatowe śmigło samolotu zostało tu uchwycone za pomocą kamery smartfona wyposażonej w przetwornik z migawką *rolling shutter*. Jak widać, obraz został bardzo silnie zniekształcony, co wynika właśnie z rejestracji poszczególnych linii z pewnym przesunięciem czasowym względem momentu rozpoczęcia ekspozycji. Rzecz jasna, efektu tego nie należy mylić z innym artefaktem, określanym mianem *wagon-wheel effect* i widocznym np. podczas filmowania jadącego z odpowiednią prędkością samochodu – pozorny obrót kół pojazdu w kierunku przeciwnym do rzeczywistego wynika bowiem z aliasingu.

Opisane powyżej artefakty nie wyczerpują jednak tematu wpływu rodzaju migawki na warunki rejestracji obrazu. Wyobraźmy sobie, że mamy do czynienia z systemem obrazowania mikroskopowego, za pomocą którego obserwujemy np. ruchy drobnoustrojów w polu widzenia mikroskopu. Jeżeli ruchy są dostatecznie szybkie w stosunku do częstotliwości następujących po sobie ekspozycji (tzw. *framerate*), to może okazać się, że – zanim układ przemiatania matrycy zdąży dojść do ostatniego wiersza pikseli – na obrazie pojawi się organizm, którego wcześniej w owym polu nie było lub ten, który był widoczny w poprzedniej ramce, zniknie z obszaru obrazowania. Jeżeli kamera współpracuje z systemem analizy obrazu, realizującym funkcje zliczania obiektów lub ich śledzenia, pomiaru odległości pomiędzy nimi itp., to sposób realizacji migawki będzie w dużej mierze wpływał na błąd metody.



Rysunek 19. Wpływ niewłaściwej synchronizacji migawki typu *rolling shutter* i oświetlacza LED na lokalną jasność obrazu mikroskopowego (http://t.ly/6uOYZ)



Rysunek 20. Zastosowanie metody określanej mianem *pseudo-global shutter* do rejestracji obrazu w warunkach synchronizowanego oświetlenia preparatu mikroskopowego (http://t.ly/qTEn1)

W opracowaniach dotyczących klasycznych zastosowań przetworników obrazu w fotografii cyfrowej rzadko natomiast można znaleźć informacje o powiązaniu migawki z synchronicznym oświetlaczem. Tymczasem także tutaj rodzaj migawki może mieć kolosalne znaczenie dla uzyskiwanych obrazów. Spójrzmy na **rysunek 19**, na którym zaprezentowano następującą sytuację: preparat mikroskopowy zostaje oświetlony 10-milisekundowym impulsem światła z diody LED.

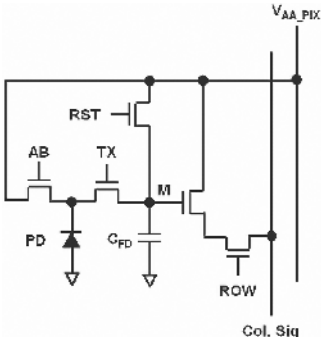
Fotografia 13. Artefakt spowodowany przez migawkę typu *rolling shutter* podczas obrazowania szybko wirującego śmigła samolotu (http://t.ly/3i2vc)

Jednocześnie z początkiem impulsu startuje także ekspozycja, która trwa jednak przez 20 milisekund. Przy takiej konfiguracji układu obrazowania pierwszy wiersz matrycy zostanie naświetlony najsilniej, zaś wszystkie kolejne otrzymają coraz mniej światła. Z punktu widzenia następnych wierszy matrycy będzie to porównywalne z „wirtualnym" zmniejszaniem współczynnika wypełnienia sygnału sterującego diodą – w skrajnym przypadku ostatni wiersz może nawet w ogóle nie zostać oświetlony w czasie, gdy przyjdzie kolej na jego ekspozycję. Efekt? Obraz będzie stopniowo ściemniał się w osi pionowej (Y) – jak gdyby na obiektyw został nałożony silny filtr połówkowy.

Co zrobić, aby uniknąć powstania takiego gradientu? Metoda jest bardzo prosta – wystarczy tak wydłużyć ekspozycję (i/lub skrócić czas oświetlenia sceny), by na okno czasowe z włączonym oświetlaczem przypadła częściowa ekspozycja wszystkich wierszy matrycy. Takie rozwiązanie, zaprezentowane na **rysunku 20**, nosi miano *pseudo-global shutter* – z jednej strony, zmiana sposobu sterowania oświetlaczem pozwala wykonać prawidłową ekspozycję całej matrycy (tak jak w przypadku klasycznych przetworników obrazu z globalną migawką), z drugiej jednak nie chroni to w żaden sposób przed artefaktami wynikającymi z dynamiki samej sceny – wszystkie opisane wcześniej problemy, związane np. z ruchem obiektu(-ów) w polu obrazowania, wystąpią tutaj z takim samym natężeniem.

Warto jeszcze dodać, że – o ile zdecydowana większość matryc CCD z natury rzeczy pracuje w trybie migawki globalnej – o tyle w przypadku czujników obrazu CMOS konstrukcja migawki może być narzucona przez odpowiedni projekt układu sterowania oraz, rzecz jasna, samych pikseli. Jak nietrudno zauważyć, klasyczna struktura matryc typu PPS oraz znacznej części czujników APS narzuca pracę w trybie *rolling shutter*.

Jeżeli jednak do każdego piksela „dorzucimy” dodatkowy tranzystor kluczujący, to możliwe jest – poprzez włączenie wszystkich tych tranzystorów jednocześnie – sterowanie zezwoleniem na ekspozycję w całej matrycy. Stąd też bardziej rozbudowane czujniki obrazu mogą pracować w trybie *global shutter*, a nawet… oferują możliwość wyboru rodzaju migawki. Wiąże się to rzecz jasna z rozbudową układu, dlatego niektóre matryce CMOS mają konstrukcję 5T czy 6T (czyli każdy piksel ma 5 lub 6 tranzystorów) – przykład można zobaczyć na **rysunku 21**. Na marginesie warto dodać, że w niektórych publikacjach naukowych można znaleźć nawet konstrukcje bazujące na 8 tranzystorach/piksel.



Rysunek 21. Topologia piksela matrycy CMOS o konstrukcji 5T (http://t.ly/06alG)

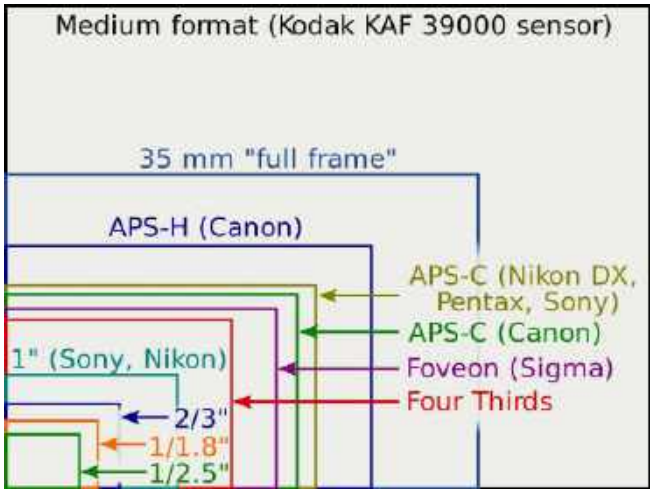
Rozdzielczość i nie tylko – parametry czujników obrazu w praktyce

We wcześniejszej części artykułu wielokrotnie wspomnieliśmy już o rozdzielczości sensorów obrazu, mierzonej najczęściej w megapikselach, czyli milionach pikseli. Warto przy tym zauważyć, że parametr ten, będący przybliżonym wynikiem mnożenia liczby pikseli w poziomie i pionie, jest w istocie sumą wszystkich pikseli z filtrami czerwonymi, zielonymi i niebieskimi – odzwierciedla zatem także docelową liczbę punktów w obrazie będącym wynikiem demozaikowania. Pewne problemy stwarzał w tym zakresie opisany wcześniej czujnik Foveon X3, którego rozdzielczość przestrzenna była wprawdzie dość przeciętna, ale „dla porządku” (a raczej – w celu dorównania konkurencyjnym czujnikom obrazu z filtrami Bayera) producent podawał liczbę pikseli przemnożoną przez trzy (względem wyniku mnożenia „pikselowej” szerokości i wysokości czujnika).

Rozdzielczość wraz z fizycznymi rozmiarami matrycy determinują natomiast wymiary poszczególnych pikseli – im mniejszy sensor, tym gęściej upakowane punkty światłoczułe, ale zarazem także wyższy poziom szumów. Nietrudno się bowiem domyślić, jak kolosalne znaczenie dla szumów podczas ekspozycji w słabym świetle ma rozmiar piksela – im jest on mniejszy, tym mniej fotonów może „wyłapać”, a co za tym idzie – udział każdego fotoelektronu w sygnale wyjściowym ma w takim wypadku dużo większy wpływ na amplitudę sygnału. Na **rysunku 22** można zobaczyć porównanie rozmiarów różnych czujników obrazu, stosowanych głównie w fotografii cyfrowej, zaś w **tabeli 2** zebrano przykładowe wymiary i oznaczenia matryc spotykanych na rynku. Warto zauważyć, że rozmiar matrycy (podany w calach) po przeliczeniu na milimetry jest wyraźnie większy niż rzeczywista długość przekątnej wyliczona na podstawie szerokości i wysokości sensora. Nie jest to jednak błąd, a… spuścizna po erze lamp analizujących, których wymiary były podawane „całościowo”, tj. z uwzględnieniem obszaru otaczającego aktywny rejon lampy.

Tabela 2. Porównanie typowych rozmiarów matryc CMOS/CCD (https://t.ly/fdZJ5)

Oznaczenie	Oznaczenie czujnika przeliczone na [mm]	Rzeczywista przekątna [mm]	Proporcje 4:3		Proporcje 3:2		Proporcje 16:10	
			Szerokość [mm]	Wysokość [mm]	Szerokość [mm]	Wysokość [mm]	Szerokość [mm]	Wysokość [mm]
35 mm pełna klatka	–	43,3	34,64	25,98	36,03	24,02	36,72	22,95
APS-C	–	30,10	24,08	18,06	25,04	16,7	25,52	15,95
4/3"	33,87	21,33	17,07	12,8	17,75	11,83	18,09	11,3
1,1"	27,94	17,6	14,08	10,56	14,64	9,76	14,92	9,33
1"	25,4	16	12,8	9,6	13,31	8,88	13,57	8,48
1/1,1"	23,09	14,55	11,64	8,73	12,10	8,07	12,33	7,71
1/1,2"	21,17	13,33	10,67	8	11,09	7,4	11,31	7,07
2/3"	16,93	10,67	8,53	6,4	8,88	5,92	9,05	5,66
1/1,6"	15,88	10	8	6	8,32	5,55	8,48	5,3
1/1,7"	14,94	9,41	7,53	5,65	7,83	5,22	7,98	4,99
1/1,8"	14,11	8,89	7,11	5,33	7,4	4,93	7,54	4,71
1/1,9"	13,37	8,42	6,74	5,05	7,01	4,67	7,14	4,46
1/2,0"	12,7	8	6,4	4,8	6,66	4,44	6,78	4,24
1/2,3"	11,04	7,83	6,26	4,7	6,51	4,34	6,64	4,15
1/2,5"	10,16	7,2	5,76	4,32	5,99	3,99	6,11	3,82
1/2,7"	9,41	6,67	5,33	4	5,55	3,7	5,65	3,53
1/2,8"	9,07	6,43	5,14	3,86	5,35	3,57	5,45	3,41
1/2,9"	8,76	6,21	4,97	3,72	5,16	3,44	5,26	3,29
1/3,0"	8,47	6	4,8	3,6	4,99	3,33	5,09	3,18
1/3,2"	7,94	5,63	4,5	3,38	4,68	3,12	4,77	2,98
1/3,4"	7,47	5,29	4,24	3,18	4,41	2,94	4,49	2,81
1/3,6"	7,06	5	4	3	4,16	2,77	4,24	2,65
1/4,0"	6,35	4,5	3,6	2,7	3,74	2,5	3,82	2,39
1/5,0"	5,08	3,6	2,88	2,16	3	2	3,05	1,91
1/6,0"	4,23	3	2,4	1,8	2,5	1,66	2,54	1,59
1/7,5"	3,39	2,4	1,92	1,44	2	1,33	2,04	1,27
1/9,0"	2,82	2	1,6	1,2	1,66	1,11	1,69	1,06



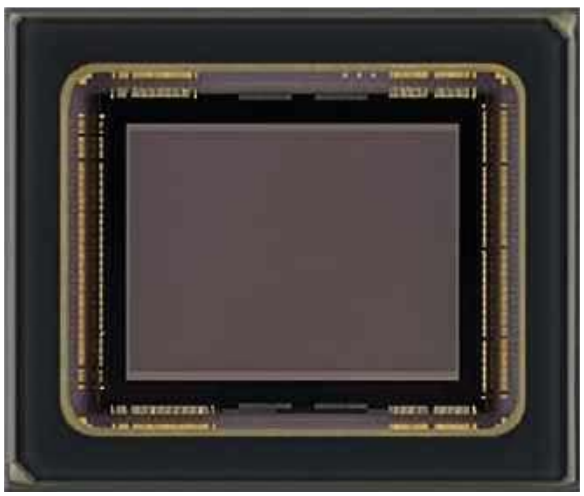
Rysunek 22. Schematyczne porównanie rozmiarów czujników obrazu różnych producentów (http://t.ly/P3RR3)

Fotografia 14. Czujnik obrazu typu FPA320x256-1.9-TS2-50 mm firmy Andanta przeznaczony do pracy w paśmie 1100...1900 nm (http://t.ly/DNJR1)

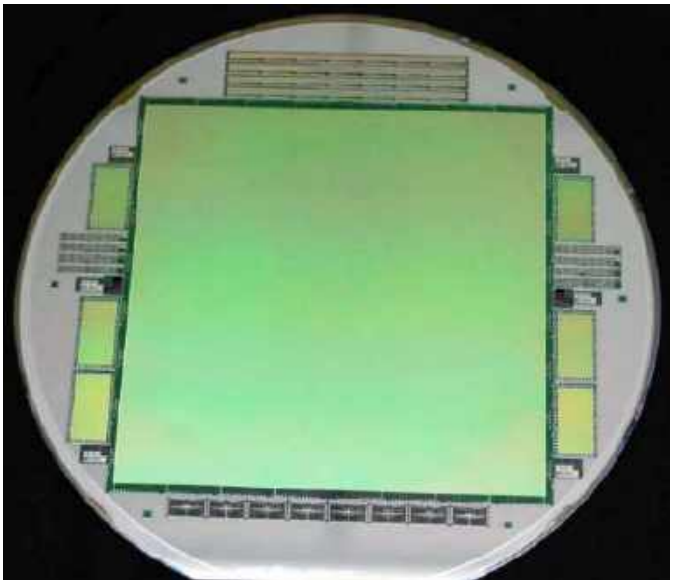
Wydajność kwantowa a widmo rejestrowanych fal elektromagnetycznych

Kolejnym istotnym parametrem jest wydajność kwantowa (QE, ang. *quantum efficiency*), określana w procentach i wyrażająca efektywność konwersji światła na sygnał elektryczny. QE to nic innego, jak stosunek liczby „wyprodukowanych" fotoelektronów do liczby fotonów, które doprowadziły do ich wybicia. Idealna wydajność kwantowa wynosiłaby więc 100%, ale w rzeczywistości najlepsze przetworniki obrazu są w stanie osiągnąć maksymalny wskaźnik QE około 95%. Zwiększaniu wydajności kwantowej w oczywisty sposób sprzyja konstrukcja BSI.

W tym miejscu warto zwrócić uwagę na fakt, że oprócz klasycznych czujników obrazu (przeznaczonych do pracy w świetle widzialnym) na rynku dostępnych jest także wiele modeli sensorów oraz gotowych kamer, które dostosowane zostały do rejestracji



Fotografia 15. Czujnik obrazu IMX487 przeznaczony do pracy w paśmie UV (http://t.ly/kxWJH)

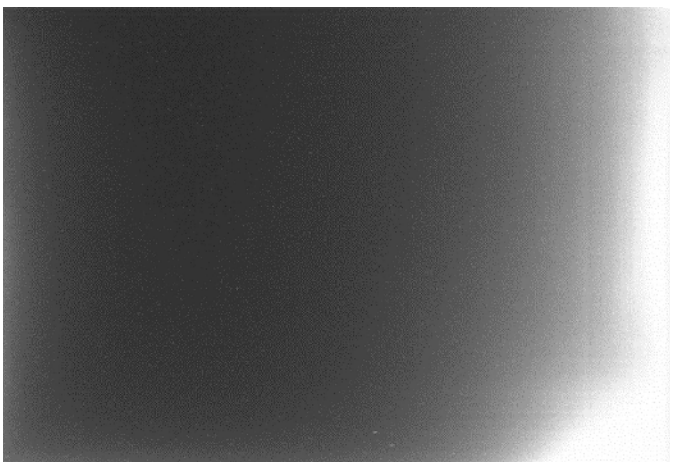


Fotografia 16. Wafel zawierający matrycę CCD do bezpośredniej rejestracji promieniowania rentgenowskiego (http://t.ly/io8BA)

scen w innych przedziałach widma elektromagnetycznego – zwykle jednak przy znacznie niższej wydajności kwantowej, na poziomie 50...60% (maks.). I nie chodzi tutaj tylko o popularne kamery zdolne do pracy w podczerwieni bliskiej (tak działają przecież niemal wszystkie współczesne kamery CCTV wyposażone w oświetlacze IR). W ofertach niektórych producentów można bowiem znaleźć także modele oparte na półprzewodnikach innych niż krzem – np. InGaAs – i dzięki temu czułe na pasmo w zakresie od 900 nm do 1700 nm, od 1100 nm do 1900 nm czy też od 1200 do 2200 nm. Takie właśnie czujniki obrazu produkuje m.in. niemiecka firma Andanta (**fotografia 14**). W sprzedaży dostępne są ponadto sensory CMOS do pracy w ultrafiolecie (np. IMX487 marki Sony – **fotografia 15**) czy nawet... matryce CCD do bezpośredniej akwizycji promieniowania rentgenowskiego (**fotografia 16**). Bezpośredniej – czyli z pominięciem konwencjonalnej techniki korzystającej ze scyntylatora przetwarzającego promieniowanie X na fotony światła widzialnego, rejestrowane następnie przez konwencjonalny czujnik obrazu.

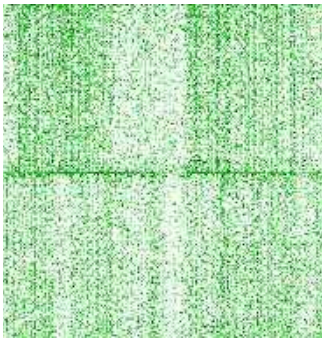
Szumy w czujnikach obrazu

O metodach obniżania szumu pisaliśmy przy okazji omawiania czujników obrazu typu EMCCD i sCMOS. Dodajmy więc zatem, że na zwiększenie szumu rejestrowanego na obrazie wpływa szereg czynników, związanych zarówno z konstrukcją sensora, jak i warunkami jego pracy, w tym:



Fotografia 17. Szum matrycy CMOS z widocznym wzmocnieniem z prawej strony, wynikającym z nierównomiernego ogrzewania czujnika obrazu podczas pracy (http://t.ly/9krz4)

- **Ustawienie czułości** – im wyższa, tym silniejszy udział szumu w obrazie wyjściowym; o wpływie czułości (określanej w standardzie ISO) na jakość obrazu wiedzą doskonale osoby zajmujące się fotografią.
- **Wielkość piksela** – jej wpływ na poziom szumów opisaliśmy już wcześniej.
- **Temperatura** – wzrost temperatury sensora obrazu (a także współpracującej elektroniki) w oczywisty sposób podnosi całkowity poziom szumu i to w sposób niezależny od natężenia rejestrowanego światła – wynika to bowiem z prądu ciemnego fotoelementów oraz losowych zaburzeń występujących w pozostałej części struktury półprzewodnika. Warto jednak zwrócić uwagę na fakt, że znaczenie zyskuje tutaj nawet zjawisko samonagrzewania się kamery i to – co gorsza – nierzadko w sposób nierównomierny (**fotografia 17**).
- **Wpływ układów odczytowych** – szum powstaje nie tylko w samej matrycy, ale także w układach przetwarzania sygnałów analogowych (wzmacniacze, multipleksery, etc.) oraz w przetwornikach ADC. Z tego względu wysokiej klasy kamery, wyposażone w niskoszumne obwody odczytowe, są znacznie droższe, niż wynikałoby tylko z ceny samego czujnika obrazu.
- **Niedoskonałości matrycy** – oprócz szumu rozłożonego (przy zachowaniu jednakowej temperatury całego sensora) w sposób równomierny na powierzchni matrycy, w obrazach rejestrowanych przy ekstremalnie niskim natężeniu światła można także zaobserwować szumy określane mianem *fixed pattern noise*, wynikające z nieidealnej konstrukcji struktury półprzewodnikowej sensora. Widać to szczególnie wyraźnie w czujnikach niższej jakości, a także w sensorach o konstrukcji dzielonej (**fotografia 18**).



Fotografia 18. Reprezentacja szumu typu *fixed pattern noise* w przypadku czujnika obrazu o konstrukcji dzielonej (starszy typ matrycy CMOS). Kolorystykę oryginalnego obrazu odwrócono w celu lepszej wizualizacji opisywanego efektu w druku (http://t.ly/CdIUW)

Subminiaturowe sensory obrazu

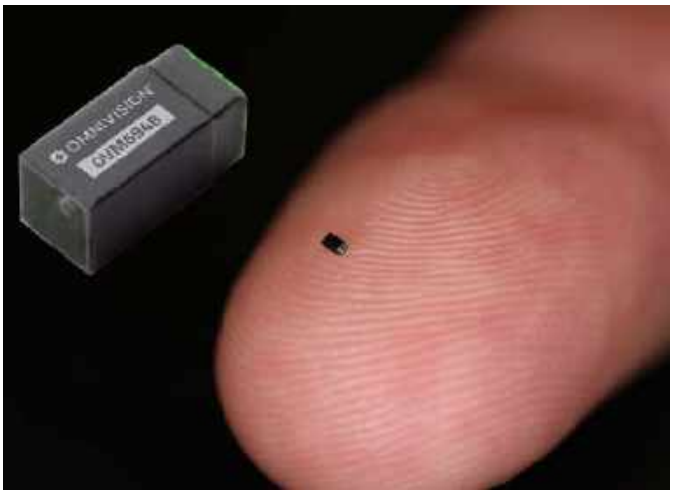
Podczas omawiania tematyki czujników obrazu nie sposób pominąć szczytowych osiągnięć w zakresie miniaturyzacji tych komponentów. Zaszczytne miano lidera należy się znanej firmie OMNIVISION, która w 2019 roku uzyskała wpis do Księgi Rekordów Guinnessa za wyprodukowanie najmniejszego na świecie, dostępnego komercyjnie czujnika obrazu. Model OVM6948 ma wprawdzie stosunkowo niewielką (jak na dzisiejsze standardy) rozdzielczość matrycy wynoszącą 40 kpx (200×200 px, akwizycja 30 fps), ale za to może poszczycić się fenomenalnie kompaktowymi rozmiarami: 0,65×0,65×1,158 mm (**fotografia 19**). Dzięki tak dalece posuniętej miniaturyzacji ta mikroskopijna kamera (mówimy tutaj bowiem o kompletnym module z wbudowaną optyką o kącie widzenia 120° i głębi ostrości 3...30 mm!) może być umieszczona w cewniku o średnicy na poziomie 1,0 mm, co daje ogromny potencjał w endoskopowych procedurach małoinwazyjnych. Tak doskonałe osiągi wymagały rzecz jasna optymalizacji każdego aspektu technologicznego – w kamerze zastosowano dwuliniowy interfejs danych (linia synchronizacji oraz analogowe wyjście obrazu), dzięki czemu obudowa modułu ma zaledwie… cztery piny.

Fotografia 20. Moduł kamery OCH2B10 marki OMNIVISION (http://t.ly/kTJoz)

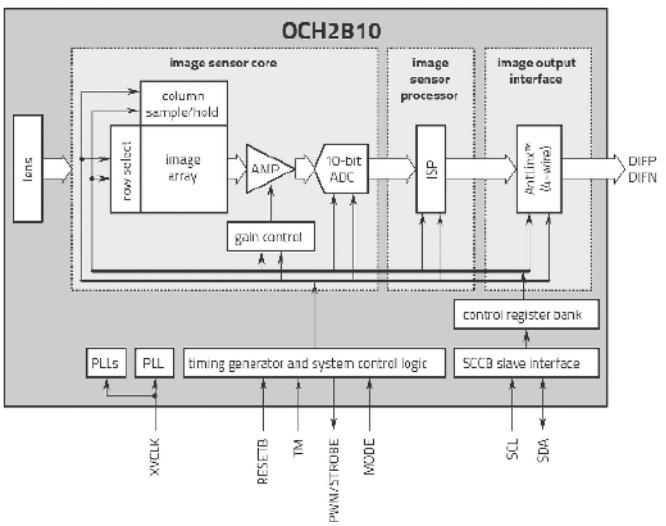
Ten sam producent wprowadził także na rynek inne moduły kamer, przeznaczone – podobnie jak OVM6948 – głównie do aplikacji medycznych. Obszerna linia produktowa CameraCubeChip obejmuje kilkanaście subminiaturowych modeli kamer, z których najlepsze osiągi w zakresie rozdzielczości oferuje OCH2B10 (**fotografia 20**). Jej 2-megapikselowa matryca (1500×1500 px) może rejestrować obraz z szybkością 60 fps i ma wymiary zaledwie 2,565×2,565×3,634 mm. Moduł został opracowany głównie z myślą o niewielkich endoskopach jednorazowych. W tym przypadku mamy już do czynienia z konstrukcją w pełni cyfrową, wyposażoną w 1-bitowy przetwornik ADC, procesor obrazu i własnościowy interfejs komunikacyjny AntLinx (**rysunek 23**).

Czujniki przestrzeni vs sensory obrazu

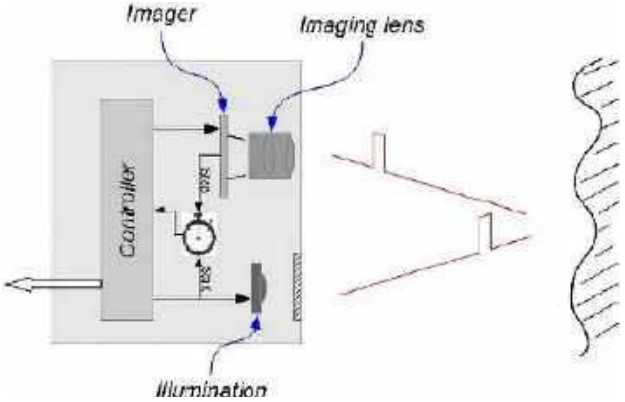
Typowe zastosowanie standardowych kamer – nawet najlepszej jakości – nie umożliwia wiarygodnej oceny odległości poszczególnych obiektów (głębi sceny). Istnieje wprawdzie szereg technik pozwalających „obejść" ograniczenia układów obrazowania bazujących na klasycznych matrycach CMOS i CCD (analiza z użyciem światła strukturalnego, stereowizja czy też mapowanie trójwymiarowe na drodze stackingu zdjęć wykonywanych z niewielką głębią ostrości), ale każda z nich ma, oprócz oczywistych zalet, także szereg bardzo istotnych wad i ograniczeń.



Fotografia 19. Najmniejsza na świecie kamera modułowa – OVM6948 marki OMNIVISION (http://t.ly/zN-wH)



Rysunek 23. Schemat blokowy kamery OCH2B10 (http://t.ly/kTJoz)



Rysunek 24. Zasada działania sensorów dToF (http://t.ly/LxQDO)

Rozwój optoelektroniki i szybkich układów przetwarzania sygnałów doprowadził do powstania nowej kategorii czujników obrazu, które należy określić raczej mianem **czujników przestrzeni**. Z klasycznymi matrycami nie mają one jednak zbyt wiele wspólnego – praktycznie jedyną cechą zbliżającą czujniki obrazowe 3D do zwykłych sensorów obrazu jest konstrukcja oparta na ortogonalnej siatce pikseli. Wszystko inne – rodzaj fotoelementów, sposób ich podłączenia oraz (a raczej przede wszystkim) metoda przetwarzania uzyskiwanych z matrycy sygnałów – jest biegunowo odległe od tego, co opisaliśmy do tej pory.

Czujniki dToF – budowa i zasada działania

Czujniki przestrzeni bazują na optycznej realizacji metody, którą w ogólności zwykło się określać mianem *time-of-flight* (ToF), czyli na pomiarze czasu przelotu wiązki światła od emitera (najczęściej lasera) do obiektu i z powrotem, po odbiciu, do matrycy fotoelementów. Zastosowanie „własnego" oświetlacza, precyzyjnie zsynchronizowanego z macierzą odbiorczą, jest warunkiem koniecznym do realizacji pomiaru ToF z oczywistego względu: układ pomiarowy musi znać dokładny moment, w którym wiązka światła wyrusza w stronę obrazowanej sceny, by móc obliczyć różnicę czasu, a następnie – na jej podstawie – odległość poszczególnych punktów od kamery.

Szeroko rozpowszechnionym rodzajem oświetlacza w tym obszarze rynku są podczerwone lasery z pionową wnęką rezonansową (VCSEL od ang. *Vertical Cavity Surface Emitting Laser*), które swoją niebywałą popularność zawdzięczają przede wszystkim niedrogim, kompaktowym czujnikom odległości. Sensory tego typu bazują na tzw. bezpośredniej odmianie metody ToF, określanej jako dToF (ang. *direct Time-of-Flight*). U podstaw tej techniki leży dokładnie to samo równanie, które stosowane jest także w przypadku impulsowych radarów mikrofalowych, sonarów czy prostych dalmierzy ultradźwiękowych (1):

$$d = \frac{c\Delta t}{2} \quad (1),$$

gdzie: d – odległość od obiektu do czujnika [m], c – prędkość fali w ośrodku, czyli w tym przypadku prędkość światła [m/s], Δt – różnica czasu pomiędzy rozpoczęciem nadawania impulsu światła laserowego a rozpoczęciem odbioru fali odbitej od przeszkody [s].

Układy sterujące czujników dToF (**rysunek 24**) wysyłają ultrakrótkie impulsy światła laserowego o czasie trwania na poziomie 0,2...5 ns. Ze względu na niską moc nadawania, krótki czas impulsu oraz nieuniknione straty (związane z rozproszeniem oraz absorpcją fotonów przez przeszkodę lub cząstki ośrodka), ilość odebranych fotonów jest znikoma. Z tego względu w roli fotoelementów stosowane są fotodiody lawinowe (APD) lub detektory jednofotonowe typu SPAD, których dodatkową zaletą jest bardzo krótki czas reakcji – konieczny, by móc zarejestrować tak wąskie impulsy światła laserowego. Redukcja szumu i wpływu zakłóceń zewnętrznych na wynik pomiaru jest realizowana poprzez wielokrotną akwizycję próbek i obróbkę ich na zasadzie analizy statystycznej – dostępne na rynku czujniki są wyposażone we wbudowane mikrokontrolery, których fabrycznie wgrane oprogramowanie przetwarza odebrane dane na drodze budowy histogramu i analizy piku jego obwiedni (**rysunek 25**).

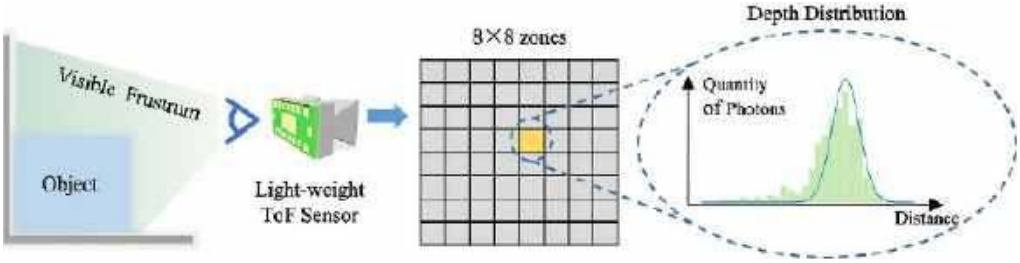
Fotografia 21. Przykładowy wielosegmentowy czujnik dToF typu VL53L8CH marki STMicroelectronics o rozdzielczości 8×8 px (http://t.ly/WQLpj)

Fotografia 22. Sensor VL53L9CA marki STMicroelectronics (http://t.ly/OMcC7)

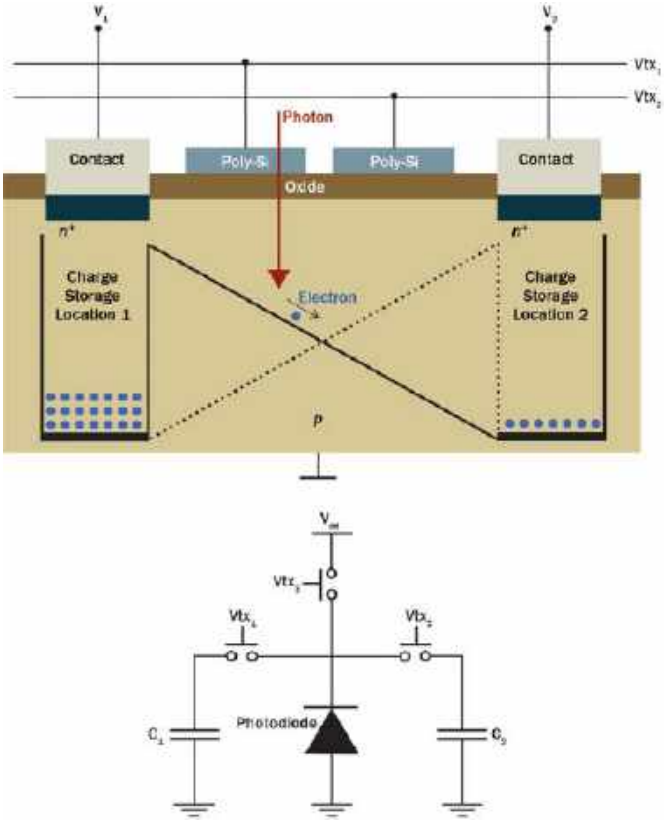
Metoda dToF oferuje doskonałą dokładność i relatywnie szeroki zakres pomiarowy, ale ma niestety sporą wadę – w praktyce nadaje się do zastosowania jedynie w prostszych czujnikach z pojedynczym polem widzenia (FOV) lub w sensorach matrycowych o relatywnie niewielkiej rozdzielczości. Jednak i tutaj dokonuje się znaczący postęp – podczas gdy do niedawna szczytowym osiągnięciem w tym zakresie były sensory wielosegmentowe o rozdzielczości na poziomie kilkudziesięciu pikseli (np. 4×8 czy 8×8 – patrz **fotografia 21**), to najnowszy sensor, który niebawem ma trafić do sprzedaży – model VL53L9CA (**fotografia 22**) – oferuje już naprawdę przyzwoitą rozdzielczość rzędu 54×42 px i może dokonywać skanowania głębi sceny w zakresie od 5 cm do nawet 10 metrów z szybkością odświeżania do 60 fps. W chwili pisania niniejszego artykułu nie są jednak dostępne szczegółowe dane dotyczące np. dokładności w poszczególnych podzakresach strefy detekcji czy też w różnych warunkach oświetleniowych (czujniki dToF niezbyt dobrze radzą sobie w silniejszym świetle słonecznym, czy nawet oświetleniu sztucznym). Można jednak być pewnym, że premiera układu VL53L9CA sporo „namiesza" na rynku czujników ToF, choć w tym segmencie pozycja firmy STMicroelectronics i tak jest już bardzo silna.

Kamery iToF – konstrukcja i metoda pomiarowa

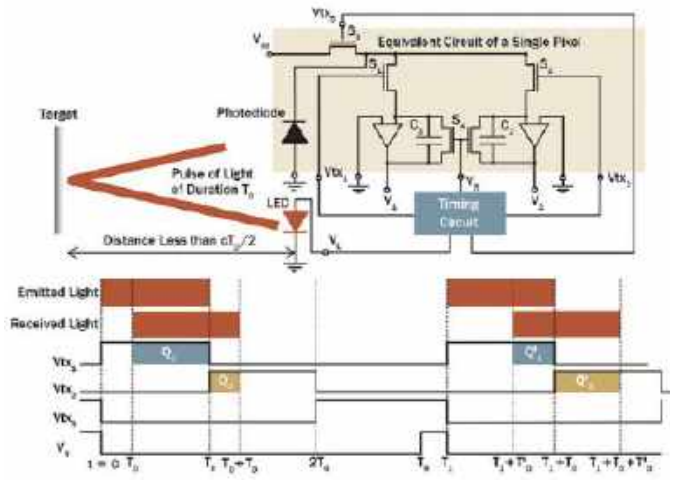
Pośrednia metoda ToF (iToF, ang. *indirect ToF*) bazuje już nie na krótkich impulsach światła laserowego, ale na oświetleniu obiektu falą ciągłą, zmodulowaną amplitudowo sygnałem o relatywnie wysokiej częstotliwości (zwykle w zakresie 20...100 MHz). Miarą odległości jest tutaj różnica fazy pomiędzy sygnałem nadanym a falą odebraną przez czuły fotoelement – na podstawie znajomości przesunięcia fazowego oraz prędkości światła wyliczana jest następnie odległość dzieląca obiekt (przeszkodę) od sensora. Metoda



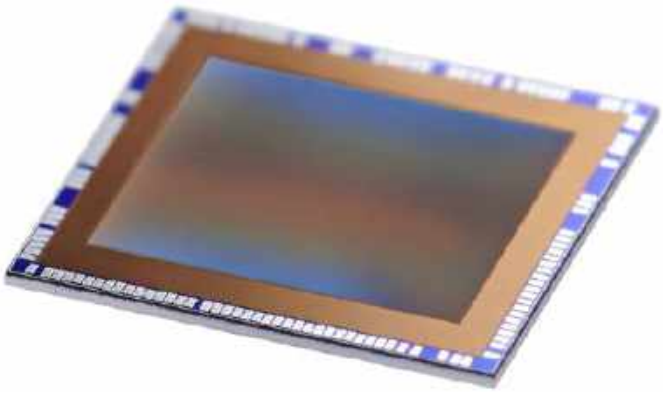
Rysunek 25. Ilustracja statystycznej metody analizy danych z matrycy czujników dToF za pomocą histogramu (http://t.ly/9ow4U)



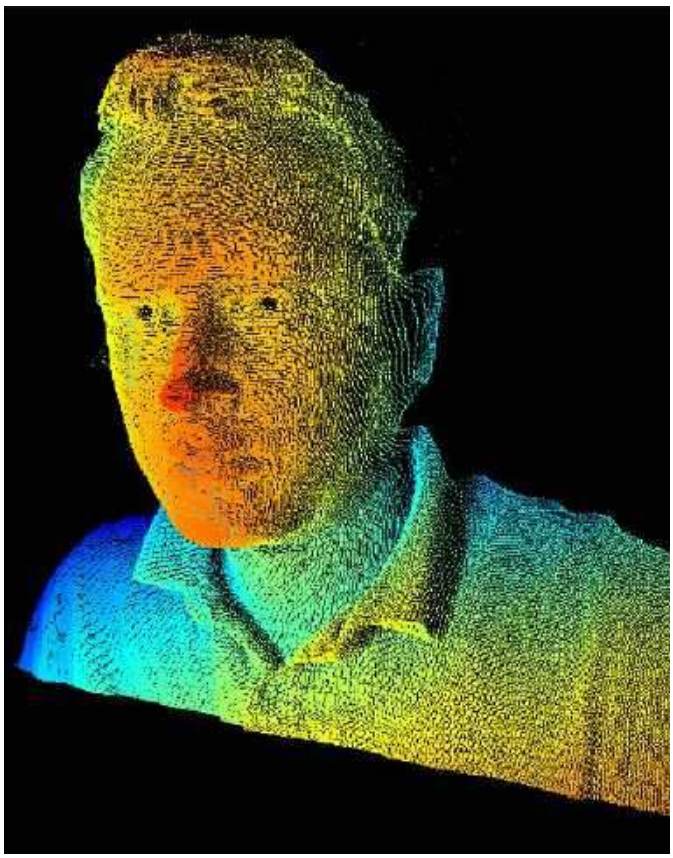
Rysunek 26. Schematyczna ilustracja struktury mieszacza fotonicznego (detektora typu PMD). Źródło: http://t.ly/RU5Pj



Rysunek 27. Zasada działania mieszacza fotonicznego (http://t.ly/QLJvy)



Fotografia 23. Matryca iToF marki STMicroelectronics – model VD55H1 (http://t.ly/RsNpY)



Rysunek 28. Mapa głębi sceny wykonana za pomocą matrycy VD55H1 (http://t.ly/RsNpY)

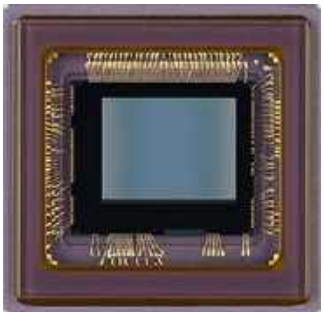
iToF pozwala znacząco zwiększyć rozdzielczość przestrzenną detektora i to do poziomu porównywalnego z przeciętnymi czujnikami obrazu typu CMOS bądź CCD.

Co ciekawe, w konstrukcjach kamer iToF można znaleźć wyspecjalizowany rodzaj detektorów, określany mianem PMD (ang. *Photonic Mixer Device*, czyli tzw. fotoniczny mieszacz – patrz **rysunek 26**). Detekcja fazy odebranej wiązki światła odbywa się tutaj na poziomie poszczególnych pikseli, tj. bez udziału zewnętrznych układów analogowych (mieszaczy, demodulatorów, etc.). Zasadę działania PMD schematycznie zilustrowano na **rysunku 27**. Foton padający na powierzchnię struktury krzemowej detektora powoduje wygenerowanie ładunku elektrycznego, który – w zależności od podanego z zewnątrz sygnału sterującego zsynchronizowanego z oświetlaczem – kierowany jest do jednego z dwóch kondensatorów. Stosunek napięć V1 i V2, będących efektem ładowania poszczególnych kondensatorów, zmienia się w zależności od przesunięcia fazowego pomiędzy sygnałem nadanym a odebranym przez dany piksel. Najważniejsza część procesu demodulacji jest więc realizowana bezpośrednio na poziomie struktury półprzewodnikowej matrycy detektorów, zatem do jej pracy konieczny okazuje się w zasadzie tylko zewnętrzny układ synchronizacji oraz przetworniki ADC, odpowiedzialne za próbkowanie napięć z poszczególnych kondensatorów.

Technologia obrazowania głębi na bazie metody iToF jest w zasadzie bardzo młoda, ale na tyle obiecująca, że szereg producentów już teraz wdrożył dostępne „z półki” matryce iToF, które z wyglądu nie różnią się zasadniczo od klasycznych czujników obrazu CMOS bądź CCD. Firma STMicroelectronics już dwa lata temu udostępniała matrycę VD55H1 (**fotografia 23**) o naprawdę imponującej rozdzielczości 0,54 Mpx (672×804 px) i szybkości odświeżania do 120 fps. Tak wysoka rozdzielczość przestrzenna pozwala generować niezwykle szczegółowe mapy głębi sceny (**rysunek 28**) w sposób nieporównanie prostszy niż przy użyciu konwencjonalnych technik (np. skanowania laserowego).

Konkurencją dla rozwiązania ST jest m.in. seria czujników przestrzeni IMX556 i IMX570 marki Sony (**fotografia 24**). Obydwa czujniki

różnią się praktycznie wszystkimi parametrami (z wyjątkiem długości fali oświetlacza oraz rodzaju interfejsu komunikacyjnego) – porównanie najważniejszych cech technicznych obydwu sensorów można znaleźć w **tabeli 3**.



Fotografia 24. Sensor obrazu iToF z serii IMX556/IMX570 marki Sony (http://t.ly/o8Stx)

Tabela 3. Porównanie podstawowych parametrów czujników iToF marki Sony (https://t.ly/5hwMG)

Model	IMX556	IMX570
Rozdzielczość	0,3 Mpx	0,3 M
Efektywna liczba pikseli	642(H)×484(V)	648(H)×488(V)
Rozmiar matrycy	1/2	1/4,5
Maksymalna szybkość odświeżania	60 fps (przy rozdzielczości VGA)	56 fps (przy rozdzielczości VGA)
Zalecana długość fali oświetlacza	850 nm lub 940 nm	850 nm lub 940 nm
Rozmiar piksela	10 µm × 10 µm	5 µm × 5 µm
Częstotliwość modulacji	4...100 MHz	2...120 MHz
Interfejs	MIPI	MIPI

Podsumowanie

Pozyskiwanie obrazu jest nieodłącznym elementem naszej codzienności. Fotografia cyfrowa, badania naukowe, diagnostyka medyczna, robotyka mobilna, rynek pojazdów autonomicznych – to tylko niektóre obszary technologii, w których czujniki obrazu odgrywają niebywale ważną rolę. Nic więc dziwnego, że przez blisko sto lat intensywnego rozwoju zdołaliśmy przejść od prostych lamp analizujących do supernowoczesnych matryc o rozdzielczości 200 milionów punktów, wyspecjalizowanych urządzeń, zdolnych do rejestracji obrazów złożonych z pojedynczych fotonów, czy też kamer iToF, które z łatwością generują obrazy głębi sceny w czasie rzeczywistym. Najmniejsze kamery – opracowane do zastosowań medycznych – mają dziś wymiary na poziomie ułamka milimetra. Pomimo tak niebywałych osiągnięć współczesnej optoelektroniki, wciąż mamy problemy z wieloma zagadnieniami, które spędzają sen z powiek naukowcom i inżynierom – walczymy z prawami fizyki (chociażby z szumem termicznym), poszukujemy nowych materiałów zdolnych do rejestracji odległych zakątków widma elektromagnetycznego, sukcesywnie „podkręcamy" tempo akwizycji ramek obrazu. Ba! Zaprzęgamy nawet sztuczną inteligencję do zadań związanych z poprawianiem niedoskonałości technologii, od której wymagamy coraz więcej i więcej. Co czeka nas w przyszłości? Jedno jest pewne – sukcesywnie zbliżamy się do granicy wykonalności, poniżej której dalsze zmniejszanie pikseli i podwyższanie rozdzielczości nie będą już po prostu możliwe, a jeśli nawet – to i tak pozbawione będą jakiegokolwiek sensu. Fizyki jeszcze nikomu nie udało się oszukać.

inż. Przemysław Musz, EP

Radioodbiornik internetowy z dekoderem VS1053

W dzisiejszych czasach większość rozgłośni radiowych prowadzi transmisje również w Internecie. Co więcej, wiele nowych stacji funkcjonuje wyłącznie w sieci. Sprawia to, że odbiorcy chętnie rozbudowują domowe systemy audio o urządzenia umożliwiające odbiór radia internetowego.

Ważne linki:
- Kod źródłowy firmware: https://github.com/ciuri/WebRadio_VS1053
- Prezentacja działania: https://www.youtube.com/watch?v=qbo4sa-MrVo

Głównym założeniem projektu było stworzenie odbiornika, który oprócz odtwarzania strumienia sieciowego, pozwoliłby także na wyszukiwanie oraz zapamiętywanie nowych stacji. Dodatkowe wymagania to zaprojektowanie intuicyjnego i prostego interfejsu użytkownika oraz możliwość sterowania urządzeniem za pomocą pilota lub enkodera obrotowego.

Wybór podzespołów

Jako centralny element wybrany został mikrokontroler ESP32-S3 na płytce developerskiej DevKitC-1. Głównym argumentem przy wyborze tego właśnie układu była wbudowana obsługa sieci Wi-Fi (stanowiąca kluczowy element projektu) oraz wystarczająca ilość pamięci i wydajność.

Kolejną kwestią wymagającą przemyślenia okazał się sposób obsługi skompresowanych danych audio. Pierwsze próby realizacji zakładały programowe dekodowanie strumienia MP3 i wysyłanie zdekodowanego sygnału do zewnętrznego przetwornika cyfrowo-analogowego (DAC) przez magistralę I^2S. Jednak z uwagi na braki w dostępnych bibliotekach (liczbę obsługiwanych formatów) oraz dużą konsumpcję zasobów mikrokontrolera, przynosiło to niezadowalające rezultaty. Ostatecznie wybór padł na dekodowanie przy użyciu wyspecjalizowanego układu, a konkretnie VS1053 firmy VLSI Solution. Jest on często stosowany w tego rodzaju aplikacjach, gdyż zapewnia obsługę praktycznie wszystkich popularnych formatów kompresji audio (MP3, WMA, OGG Vorbis, FLAC, AAC). Co ciekawe, układ obsługuje również format MIDI. Do komunikacji z mikrokontrolerem używa magistrali SPI. Bardzo istotny okazuje się fakt, że dostępne są gotowe moduły z zamontowanym układem oraz obwodami peryferyjnymi, co ułatwia zastosowanie go w prototypach.

Aby spełnić założenia dotyczące interfejsu użytkownika, urządzenie trzeba było wyposażyć w odpowiednio duży i czytelny wyświetlacz w technologii OLED. Wybór padł na model o wielkości 2,42", rozdzielczości 128×64 pikseli, sterowany układem SSD1309. Do budowy użyty został również enkoder obrotowy oraz odbiornik podczerwieni TSOP2236.

Interfejs użytkownika

Po uruchomieniu urządzenia na wyświetlaczu pojawia się ekran startowy, a w tle następuje inicjalizacja. Po jej zakończeniu odbiornik jest gotowy do obsługi i wyświetlone zostaje główne menu (**fotografia 1**). Wybraną pozycję zatwierdzamy krótkim naciśnięciem gałki enkodera. Dłuższe przyciśnięcie gałki powoduje powrót do poprzedniego menu. Do sterowania można również używać przycisków kursora

Wykaz elementów:

ESP32-S3 DevKitC-1 (1 szt.)
Rezystor 220 Ω (1 szt.)
Dioda LED (1 szt.)
Wyświetlacz OLED 2.42" SSD1309 (1 szt.)
Enkoder obrotowy z przyciskiem (1 szt.)
Odbiornik podczerwieni TSOP2236 (1 szt.)
Moduł VS1053 (1 szt.)
Kondensator 100 µF/16 V (2 szt.)
Kondensator 100 nF (2 szt.)

Fotografia 1. Główne menu

Fotografia 2. Ekran odtwarzania

na pilocie zdalnego sterowania. Istnieją ponadto nieliczne funkcje dodatkowe dostępne tylko za pośrednictwem pilota.

Favorites

Na wyświetlaczu pokazana jest lista zapisanych wcześniej „ulubionych" stacji radiowych. Użytkownik może uruchomić tutaj odtwarzanie wybranej stacji, a także usunąć wybraną pozycję z listy (usunięcie z listy dostępne jest tylko za pomocą pilota).

Select by country

Na wyświetlaczu widzimy listę krajów wraz z liczbą zarejestrowanych stacji z każdego kraju. Lista posortowana została pod kątem liczby dostępnych rozgłośni. Wybranie danej pozycji przenosi nas na listę stacji radiowych wybranego kraju (posortowaną pod kątem popularności) – wskazanie którejś z nich uruchamia odtwarzanie.

Select by tag

Po wybraniu tej pozycji otrzymujemy listę tagów, czyli słów kluczowych, które opisują rozgłośnie. Przy każdym tagu widzimy liczbę zarejestrowanych rozgłośni. Słowa kluczowe są posortowane pod kątem liczby zarejestrowanych w odniesieniu do nich stacji. Wybranie konkretnej pozycji przenosi użytkownika do listy rozgłośni dla danego tagu, z której można od razu przejść do odtwarzania wybranej stacji.

Settings->Select radio list server

Mamy tu możliwość wyboru serwera usługi sieciowej udostępniającej bazę danych z informacjami o stacjach. Usługa ta jest dostępna na kilku serwerach. W razie awarii jednego z nich możemy w tym miejscu przełączyć się na inny.

Settings->Wifi settings

Użytkownik może tutaj wybrać z listy sieć Wi-Fi, do której ma być podłączone urządzenie. Po wybraniu odpowiedniej sieci należy wprowadzić hasło dostępowe. Zatwierdzamy je długim przyciśnięciem gałki enkodera.

Settings->Restart

Wybierając tę opcję, można zrestartować urządzenie.

Turn off

Po wybraniu tej opcji odbiornik przechodzi w tryb standby.

Ekran odtwarzania

Podczas odtwarzania na wyświetlaczu (**fotografia 2**) widoczne są:

- nazwa odtwarzanej stacji,
- kraj odtwarzanej stacji,
- bitrate [kbps],
- ustawiona głośność,
- liczba bajtów w buforze odtwarzania (aktualizowana cały czas),
- napis „chunked", jeśli strumień transmitowany jest w takim właśnie trybie.

W trakcie odtwarzania użytkownik może dodać stację do listy ulubionych. Funkcja ta dostępna jest tylko za pomocą pilota. Możliwa jest również regulacja głośności odtwarzania.

Usługa radio-browser.info

Jedno z głównych założeń projektu zakłada możliwość wyszukiwania nowych stacji. Do zrealizowania tej funkcjonalności potrzebna jest baza danych zawierająca aktualne informacje i udostępniająca interfejs (API – Application Programming Interface), przez który będziemy w stanie pobrać potrzebne dane. Powyższe założenia spełnia serwis *radio-browser.info*. Jest to w pełni darmowa usługa utrzymywana przez społeczność internetową. Obecnie w serwisie zarejestrowanych jest ponad 47 tysięcy rozgłośni. Użytkownik może wyszukiwać stacje po różnych kryteriach, takich jak: kraj, popularność, powiązane tagi, język, kodek itp. Dostępna jest również opcja dodawania nowych stacji do bazy.

Komunikacja z API serwisu odbywa się przez protokół HTTP i nie jest wymagane zakładanie konta ani żadna inna metoda uwierzytelniania. Użytkownicy otrzymują rozbudowane opcje dotyczące stronicowania, sortowania i filtrowania zwracanych danych. Parametry zapytań do serwisu można przekazywać na dwa sposoby:

- W adresie URL żądania HTTP GET (przez Query String Parameters – parametry oddzielone znakiem '&').


```
[
    {
        "changeuuid": "8125d095-6397-4660-a5c4-ef0254f67e06",
        "stationuuid": "9617a958-0601-11e8-ae97-52543be04c81",
        "serveruuid": null,
        "name": "Radio Paradise (320k)",
        "url": "http://stream-uk1.radioparadise.com/aac-320",
        "url_resolved": "http://stream-uk1.radioparadise.com/aac-320",
        "homepage": "https://www.radioparadise.com/",
        "favicon": "https://www.radioparadise.com/favicon-32x32.png",
        "tags": "california,eclectic,free,internet,non-commercial,paradise,radio",
        "country": "The United States Of America",
        "countrycode": "US",
        "iso_3166_2": null,
        "state": "California",
        "language": "english",
        "languagecodes": "en",
        "votes": 186524,
        "lastchangetime": "2023-11-04 14:11:01",
        "lastchangetime_iso8601": "2023-11-04T14:11:01Z",
        "codec": "AAC",
        "bitrate": 320,
        "hls": 0,
        "lastcheckok": 1,
        "lastchecktime": "2024-04-09 08:25:52",
        "lastchecktime_iso8601": "2024-04-09T08:25:52Z",
        "lastcheckoktime": "2024-04-09 08:25:52",
        "lastcheckoktime_iso8601": "2024-04-09T08:25:52Z",
        "lastlocalchecktime": "2024-04-09 08:25:52",
        "lastlocalchecktime_iso8601": "2024-04-09T08:25:52Z",
        "clicktimestamp": "2024-04-09 20:26:41",
        "clicktimestamp_iso8601": "2024-04-09T20:26:41Z",
        "clickcount": 1570,
        "clicktrend": 28,
        "ssl_error": 0,
        "geo_lat": null,
        "geo_long": null,
        "has_extended_info": false
    }
]
```


Listing 1. Odpowiedź w formacie JSON

- W zawartości (payload) żądania HTTP POST, w którym lista parametrów musi być zapisana w formacie JSON (JavaScript Object Notation).

Przykładowy URL żądania HTTP GET (stanowiącego zapytanie o listę stacji z danego kraju) wygląda następująco:

https://nl1.api.radio-browser.info/json/stations/search?limit=10&countrycode=PL&order=clickcount&reverse=true, przy czym po znaku '?' podane są następujące parametry:

- *limit=10* (ograniczenie liczby zwróconych rekordów do 10),
- *countrycode=PL* (tylko polskie stacje),
- *order=clickcount* (sortuj po liczbie kliknięć (popularności)),
- *reverse=true* (sortuj malejąco).

Dane z serwisu zwracane są w formacie JSON, a ich przykładowa postać widnieje na **listingu 1**.

Zachęcam do zapoznania się z dokumentacją API usługi radio-browser.info, gdyż serwis ten oferuje wiele funkcjonalności i może być bardzo pomocny przy budowie podobnych projektów.

Oprogramowanie

Na początku programu wywoływana jest funkcja *Setup()*, w której następuje inicjalizacja potrzebnych zasobów oraz wczytywana jest konfiguracja przechowywana w postaci pliku JSON, w pamięci flash mikrokontrolera. Do zapisu i odczytu konfiguracji użyty został system plików SPIFFS (SPI Flash File System), zaimplementowany w mikrokontrolerach firmy Espressif.

Plik konfiguracyjny zawiera informacje, takie jak: dane dostępowe do sieci Wi-Fi, nazwa serwera usługi radio-browser.info oraz lista zapisanych stacji. Fragment struktury opisującej konfigurację pokazany jest na **listingu 2**.

```
struct DeviceConfiguration
{
    String serverName;
    String wifiName;
    String wifiPassword;
    vector<RadioStationDTO> favorites;
    //...
}
```

Listing 2. Fragment struktury opisującej konfigurację

Bardzo dużą rolę w naszym programie odgrywa obsługa formatu JSON. Zarówno plik konfiguracyjny, jak i dane z zewnętrznego serwisu są przekazywane w tym właśnie formacie. Dlatego istotne jest zaimplementowanie serializacji (zapisu obiektu do formatu JSON) oraz deserializacji (odtworzenia obiektu z postaci JSON). Z pomocą przychodzi tutaj popularna biblioteka *ArduinoJson*, która zapewnia obsługę potrzebnych operacji. Na **listingu 3** widoczne są funkcje serializujące i deserializujące dane konfiguracyjne.

```
String SerializeConfguration()
    {
        DynamicJsonDocument configJson(8192);
        configJson["serverName"] = serverName;
        configJson["wifiName"] = wifiName;
        configJson["wifiPassword"] = wifiPassword;

        JsonArray favoritesArray = configJson.createNestedArray("favorites");

        for (const auto &station : favorites)
        {
            JsonObject stationObject = favoritesArray.createNestedObject();
            stationObject["name"] = station.Name;
            stationObject["url"] = station.Url;
            stationObject["bitrate"] = station.Bitrate;
            stationObject["country"] = station.Country;
            stationObject["radiofrequency"] = station.RadioFrequency;
        }

        String jsonString;
        serializeJson(configJson, jsonString);
        return jsonString;
    }

    void DeserializeDeviceConfiguration(const char *jsonString)
    {
        const size_t capacity = JSON_OBJECT_SIZE(4) + 160000;
        DynamicJsonDocument jsonDocument(capacity);

        DeserializationError error = deserializeJson(jsonDocument, jsonString);
        if (error)
        {
            Serial.print("Json serialization errror: ");
            Serial.println(error.c_str());
            return;
        }

        serverName = jsonDocument["serverName"].as<String>();
        wifiName = jsonDocument["wifiName"].as<String>();
        wifiPassword = jsonDocument["wifiPassword"].as<String>();
        favorites.clear();
        JsonArray favoritesArray = jsonDocument["favorites"];
        for (const auto &station : favoritesArray)
        {
            RadioStationDTO favoriteStation;
            favoriteStation.Bitrate = station["bitrate"];
            favoriteStation.Country = station["country"].as<String>();
            favoriteStation.Name = station["name"].as<String>();
            favoriteStation.RadioFrequency = station["radiofrequency"].as<String>();
            favoriteStation.Url = station["url"].as<String>();
            favorites.push_back(favoriteStation);
        }
    }
```

Listing 3. Funkcja serializująca i deserializująca konfigurację

Komunikacja z serwisem *radio-browser.info* zaimplementowana została w klasie *RadioListHttpClient*. Znajdują się tam metody, takie jak: *GetTags()*, *GetCountries()*, *GetRadioURLsByCountry()*, *GetRadioURLsByTag()*. Odpowiadają one za pobieranie danych z tego serwisu. W przypadku każdej metody proces przebiega według schematu:

1. Przygotowanie odpowiedniego URL zawierającego potrzebne parametry (należy pamiętać, aby odpowiednio zakodować znaki specjalne – funkcja *urlEncode*).
2. Wysłanie zapytania HTTP GET (metoda *GET()* klasy *HTTPClient*).
3. Przekazanie strumienia HTTP do funkcji *deserializeJson*.
4. Utworzenie potrzebnych obiektów klas DTO (Data Transfer Object) na podstawie obiektu klasy *DynamicJsonDocument*.
5. Zwolnienie zasobów (*httpClient.end()* oraz *jsonDoc.clear()*).

Na **listingu 4** ukazana jest funkcja *GetCountries()*.

Wspomniane wyżej klasy DTO służą do przechowywania informacji o stacjach, krajach oraz tagach (słowach kluczowych). Mają one postać jak na **listingu 5**.

Kolejny wart omówienia fragment firmware to proces odtwarzania strumienia. Najważniejsza jego część znajduje się w pliku

```
vector<CountryDTO> RadioListHttpClient::GetCountries()
{
    vector<CountryDTO> outList;
    DynamicJsonDocument jsonDoc(8072);
    char urlText[400];
    sprintf(urlText, "https://%s/json/
countries?order=stationcount&limit=%i&reverse=true&offset=%i", _config->serverName.c_
str(), countriesPerPage, countriesPerPage * countriesPageIndex);
    httpClient.begin(urlEncode(urlText));
    httpClient.GET();
    deserializeJson(jsonDoc, httpClient.getStream());
    httpClient.end();
    for (auto item : jsonDoc.as<JsonArray>())
    {
        const char *name = item["name"];
        const char *code = item["iso_3166_1"];
        int stationsCount = item["stationcount"];
        CountryDTO c;
        c.name = name;
        c.code = code;
        c.count = stationsCount;
        outList.push_back(c);
    }
    jsonDoc.clear();
    return outList;
}
```

Listing 4. Funkcja pobierająca listę krajów

```
class RadioStationDTO
{
public:
    String Name;
    String Url;
    int Bitrate;
    String RadioFrequency;
    String Country;
};

class CountryDTO
{
public:
    String code;
    String name;
    int count;
};

class TagDTO
{
public:
    String name;
    int count;
};
```

Listing 5. Klasy DTO

```
bool TagsListState::HandleEnter()
{
    if (*_currentState != SELECT_TAG)
        return false;

    *_currentState = _stationsListState->EnterState(SELECT_TAG, "", tags[currentIndex].name);
    return true;
}

void TagsListState::HandleRight()
{
    if (*_currentState != SELECT_TAG)
        return;

    _radioListClient->SetNextTagsPage();
    GetTagsPage();
    currentIndex = 0;
}

void TagsListState::HandleLeft()
{
    if (*_currentState != SELECT_TAG)
        return;

    _radioListClient->SetPrevTagsPage();
    GetTagsPage();
    currentIndex = 0;
}
```

Listing 8. Implementacje metod wirtualnych w klasie TagsListState

HttpWebRadioClient.h. Na początku procesu odtwarzania tworzony jest task RTOS, w którym następuje podłączenie do wybranego serwera oraz wywołanie funkcji HTTP GET. Następnie odczytywane są nagłówki HTTP i pobierany jest wskaźnik do strumienia audio. Dane ze strumienia trafiają do bufora kołowego (circular buffer). Zawartość bufora kołowego (o ile nie jest pusty) jest cały czas wysyłana na magistralę SPI do układu dekodującego (VS1053). Odbywa się to w osobnym tasku RTOS, inicjalizowanym tylko raz, przy starcie urządzenia. Warto dodać, że istotna okazuje się tutaj synchronizacja dostępu do bufora kołowego, gdyż może zdarzyć się sytuacja, w której dwa zadania RTOS będą próbowały jednocześnie uzyskać do niego dostęp. W tym celu, w tzw. sekcji krytycznej użyta została klasa *mutex* i metody *lock* oraz *unlock*. Brak takiego zabezpieczenia powodował słyszalne błędy przy odtwarzaniu strumienia.

Warto zwrócić uwagę, że niektóre serwery wysyłają strumień w trybie *chunked stream*. Można to rozpoznać, gdy – po wysłaniu żądania HTTP GET – w nagłówku *Transfer-Encoding* otrzymamy wartość *chunked*. W trybie tym dane w strumieniu podzielone są na bloki. Każdy blok poprzedza informacja o jego długości (zapisana heksadecymalnie w ASCII) oraz sekwencja *[CR][LF]* (szesnastkowo [0x0D] [0x0A]). Koniec bloku danych również oznaczony jest sekwencją *[CR][LF]*. Transmisja kończy się pakietem o zerowej wielkości. Jeśli nie zinterpretujemy poprawnie takiego strumienia, w sygnale audio będą słyszalne nieprzyjemne zniekształcenia.

```
class DeviceStateBase{

public:
    virtual void HandleLoop() = 0;
    virtual void HandleUp() = 0;
    virtual void HandleDown() = 0;
    virtual void HandleLeft() = 0;
    virtual void HandleRight() = 0;
    virtual bool HandleEnter()= 0;
    virtual bool HandleBack() = 0;
};
```

Listing 6. Klasa abstrakcyjna DeviceStateBase

```
enum UIState
{
    MODE_SELECT,
    SELECT_TAG,
    SELECT_COUNTRY,
    SELECT_STATION,
    SELECT_FAVORITES,
    PLAY,
    DEVICE_START,
    SELECT_SETTINGS,
    SELECT_SERVER_SETTING,
    SELECT_WIFI_SETTING,
    ENTER_PASSWORD,
    RESTART,
    SLEEP
};
```

Listing 7. Typ wyliczeniowy UIState

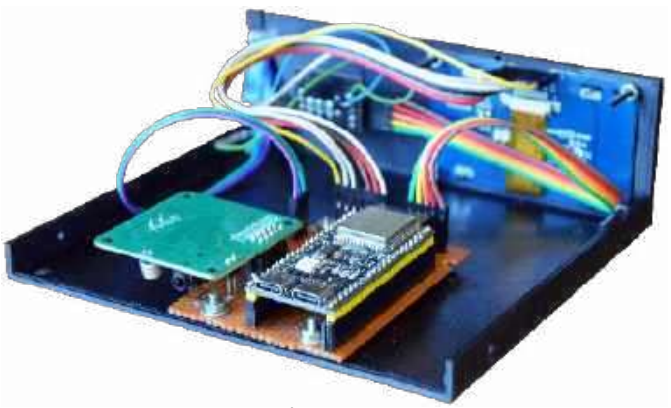
```
[env:esp32-s3-devkitc-1]
platform = espressif32
board = esp32-s3-devkitc-1
framework = arduino
lib_deps =
    baldram/ESP_VS1053_Library@^1.1.4
    bblanchon/ArduinoJson@^6.21.3
    rlogiacco/CircularBuffer@^1.3.3
    olikraus/U8g2@^2.35.4
    igorantolic/Ai Esp32 Rotary Encoder@^1.6
    z3t0/IRremote@^4.2.0
monitor_speed = 115200
```

Listing 9. Plik inicjalizacyjny platformio.ini

Interfejs użytkownika jest obsługiwany przez klasy dziedziczące po klasie abstrakcyjnej *DeviceStateBase* widocznej na **listingu 6**. Umieszczone są one w podkatalogu */lib/DeviceStates*. Znajdziemy w nich metody obsługujące zdarzenia wygenerowane przez użytkownika oraz wyświetlanie informacji na ekranie OLED. Każda z tych klas odpowiada za konkretny stan, w jakim może się znaleźć urządzenie. Listę stanów reprezentuje typ wyliczeniowy *UIState* (**listing 7**).

Przykładowe implementacje metod wirtualnych możemy znaleźć na **listingu 8**. W każdej z nich najpierw sprawdzany jest stan, w jakim znajduje się urządzenie (w tym przypadku *SELECT_TAG*), a później wykonywana jest logika danej funkcji.

Obsługa pilota zdalnego sterowania zrealizowana została za pomocą biblioteki *IRemote*, pozwalającej na dekodowanie większości protokołów używanych we współczesnych pilotach. Przy inicjalizacji wystarczy tylko podać pin mikrokontrolera, do którego podłączono odbiornik podczerwieni. Następnie należy w pętli wywoływać metodę *decode()* – zwraca ona wartość *true*, jeśli sygnał z pilota zostanie odebrany i zdekodowany. Po zdekodowaniu mamy dostęp do informacji, takich jak: typ protokołu, adres i komenda. Można również pobrać

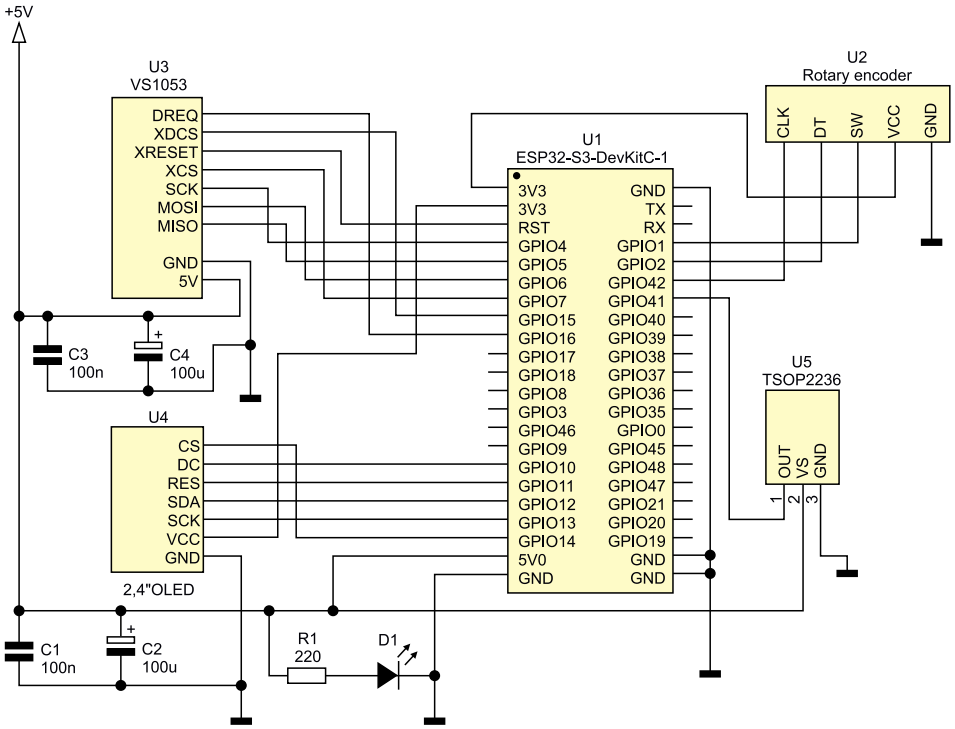


Fotografia 3. Prototyp w trakcie budowy

surowe dane odebrane przez odbiornik. Na podstawie wykrytych komend wywoływane są odpowiednie metody klas obsługujących stany urządzenia.

Wyłączenie odbiornika realizowane jest przez przejście mikrokontrolera w tryb *deep sleep*. Wybudzenie na przyciśnięcie gałki enkodera zostało skonfigurowane za pomocą funkcji *esp_sleep_enable_ext0_wakeup*.

Cały proces developmentu odbył się z użyciem edytora Visual Studio Code oraz frameworku PlatformIO. Zawartość pliku inicjalizacyjnego platformio.ini widoczna jest na **listingu 9**. Znajdują się w nim dane dotyczące modelu mikrokontrolera, lista zależności (czyli użytych zewnętrznych bibliotek) oraz inne dane konfiguracyjne. Dokumentacja dotycząca pliku platformio.ini znajduje się na stronie: https://docs.platformio.org/page/projectconf.html.



Rysunek 1. Schemat ideowy odbiornika

Montaż i uruchomienie

Schemat ideowy urządzenia pokazano na **rysunku 1**.

Odbiornik zasilany jest napięciem 5 V, filtrowanym przez kondensatory C1...C4 i doprowadzonym do płytki mikrokontrolera oraz do modułu dekodera. Wyświetlacz i enkoder obrotowy zasilane są z wyjścia 3,3 V na płytce mikrokontrolera. Wyjście audio stanowi gniazdo jack 3,5 mm umieszczone na płytce z układem dekodera. Wyświetlacz i układ VS1053 komunikują się z mikrokontrolerem za pośrednictwem dwóch oddzielnych magistral SPI. Zasilanie sygnalizuje dioda LED podłączona przez rezystor R1. Odbiornik podczerwieni TSOP2236 – zasilany napięciem 5 V – podłączony jest do pinu GPIO41 mikrokontrolera.

Prototyp został zrealizowany na uniwersalnej płytce drukowanej (**fotografia 3**), na co pozwoliła mała liczba komponentów potrzebnych do budowy. Obudowę zaprojektowano w aplikacji Fusion360, a następnie wydrukowano na drukarce 3D przy użyciu filamentu PLA.

Podsumowanie

Repozytorium projektu zostało umieszczone w serwisie GitHub i jest publicznie dostępne. Gorąco zachęcam do budowy podobnych urządzeń i mam nadzieję, że rozwiązania opisane w tym artykule okażą się pomocne.

Paweł Ciuraj

Kurs programowania mikrokontrolerów Megawin (1)

W poprzednim wydaniu „Elektroniki Praktycznej” opublikowaliśmy projekt płytki ewaluacyjnej z mikrokontrolerem MG32F103RBT6. Tym razem – zgodnie z zapowiedzią – rozpoczynamy (pierwszy na świecie!) kurs podstaw programowania tych interesujących, budżetowych układów firmy Megawin. Zaczniemy rzecz jasna od przygotowania środowiska programistycznego oraz sprawdzenia poprawności komunikacji programatora MLink – zarówno z IDE, jak i z docelowym procesorem. Następnie uruchomimy dwa proste programy umożliwiające „ożywienie” płytki i weryfikację działania niektórych znajdujących się na niej obwodów „okołoprocesorowych”.

Instalacja środowiska programistycznego

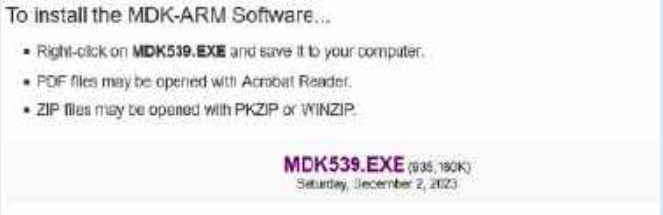
Mikrokontrolery Megawin z rdzeniem ARM Cortex-M3 mogą być programowane przy użyciu środowiska Keil MDK, wyposażonego w odpowiednie pakiety biblioteczne.

W celu pobrania środowiska Keil MDK należy wejść na stronę: keil.arm.com/mdk-community, a następnie założyć darmowe konto, używając swojego prywatnego adresu e-mail, na który wysłany zostanie kod weryfikacyjny niezbędny do przejścia do krótkiego formularza danych osobowych. Następnie, wracając jeszcze raz na stronę: https://www.keil.arm.com/mdk-community/ (**rysunek 1**), należy pobrać plik instalatora (**rysunek 2**).

Autor dziękuje firmie Micros (www.micros.com.pl) za udostępnienie programatora MLink oraz próbek układu MG32F103RBT6 na potrzeby opracowania niniejszego kursu.

Po uruchomieniu pozyskanego w opisany sposób pliku wykonywalnego, w oknie *Pack Installer* (**rysunek 3**) możemy wybrać interesujący

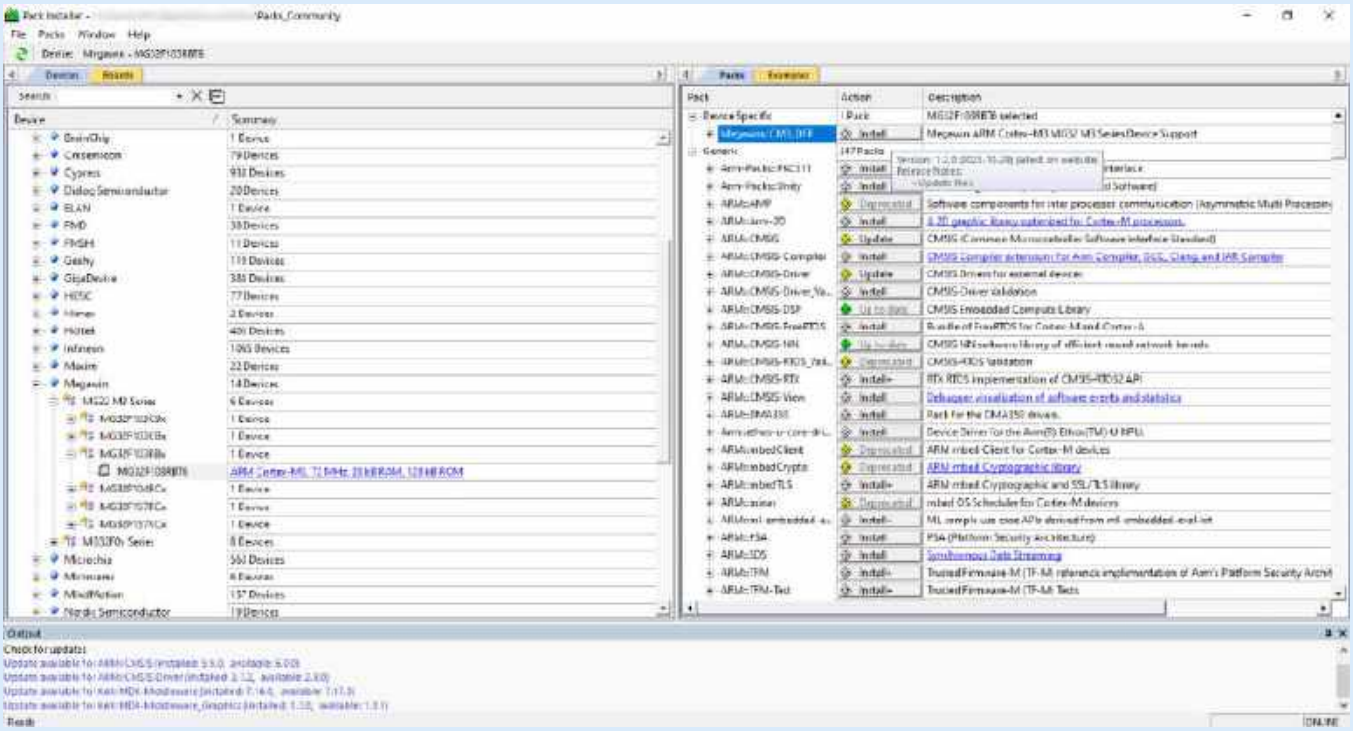
Rysunek 1. Strona internetowa umożliwiająca pobranie instalatora środowiska Keil MDK-Community edition



Rysunek 2. Link do pobrania pliku instalatora środowiska Keil MDK

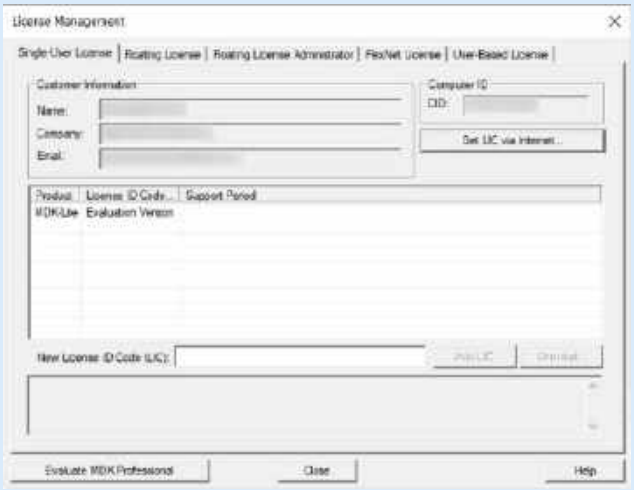
Używana w niniejszym kursie wersja środowiska Keil MDK Community Edition jest przeznaczona tylko do celów prywatnych i ewaluacyjnych – nie może być stosowana w projektach komercyjnych.

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie https://lubionykiosk.pl/media

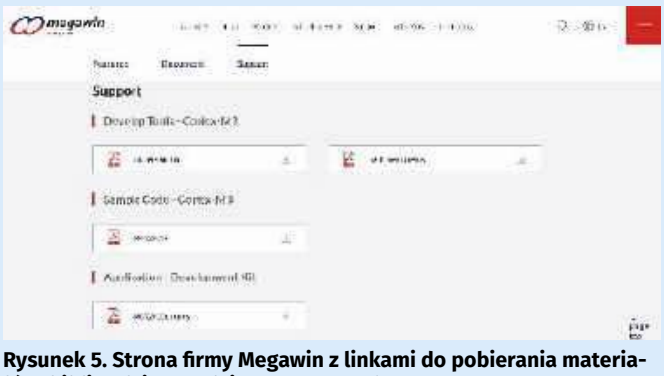


Rysunek 3. Okno Pack Installer środowiska Keil MDK

nas procesor z listy znajdującej się po lewej stronie ekranu, a następnie zainstalować odpowiedni pakiet, klikając przycisk *Install*, znajdujący się tuż obok sekcji *Device Specific* → *Megawin:CM3_DFP*. Etap ten nie jest jednak konieczny – za chwilę bowiem i tak doinstalujemy najnowszą wersję bibliotek za pomocą specjalnego programu dostarczanego przez producenta.



Rysunek 4. Okno License Management (dane wrażliwe ukryto)



Rysunek 5. Strona firmy Megawin z linkami do pobierania materiałów, bibliotek i narzędzi programowych

Na razie konieczne jest natomiast uzyskanie odpowiedniej licencji – proces ten wymaga uruchomienia środowiska Keil z uprawnieniami administratora, a następnie wybrania opcji *File* → *License Management*. W otwartym w ten sposób oknie dialogowym należy kliknąć przycisk *Get LIC via Internet* (**rysunek 4**), po czym przekierowani zostaniemy do strony internetowej z jeszcze jednym formularzem, którego zatwierdzenie wygeneruje kod nowej licencji (LIC). Ów kod zostanie przesłany mailem na adres podany w formularzu. Po przekopiowaniu otrzymanego ciągu znaków w odpowiednie pole okna *License Management* należy kliknąć przycisk *Add LIC* – po chwili powinny zostać wyświetlone dane uzyskanej licencji MDK-ARM Community. Jeżeli tak się stanie – oznacza to, że program jest gotowy do pracy.

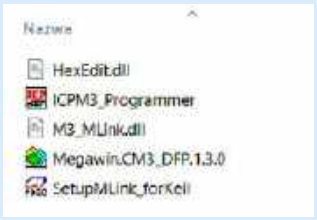
Konfiguracja środowiska Keil

Aby pobrać dodatkowe sterowniki, przechodzimy na stronę: https://www.megawin.com.tw/product/MG32F103RBT6#Support i w zakładce Support klikamy link zatytuowany *OCDM3 MLink* (**rysunek 5**). Skompresowane archiwum zawiera szereg narzędzi programowych przeznaczonych do współpracy z oficjalnym programatorem/debuggerem sprzętowym MLink (**fotografia 1**): aplikację *ICPM3_Programmer*, biblioteki dynamiczne *M3_MLink.dll i HexEdit.dll*, pakiet bibliotek kompatybilny ze środowiskiem Keil (*Megawin.CM3_DFP.1.3.0*) oraz program, który najbardziej interesuje nas w tej chwili: *SetupMLink_forKeil* (**rysunek 6**) – po rozpakowaniu archiwum uruchamiamy zatem ostatni z wymienionych plików.

W otwartym okienku dialogowym wskazujemy ścieżkę instalacji Keila (domyślnie będzie to C:\Keil_v5), a następnie klikamy *Install*. Program nie tylko automatycznie przeprowadzi instalację sterownika programatora MLink, ale także umieści pakiet DFP (zestaw

Fotografia 1. Widok programatora MLink wraz z nakładkami zabezpieczającymi (http://t.ly/JGLuk)

bibliotek, plików startowych itp.) w bazie środowiska Keil, dzięki czemu będziemy mogli od razu po zakończeniu konfiguracji przejść do tworzenia pierwszego programu. Przy okazji możemy także zaktualizować oprogramowanie układowe interfejsu MLink. Po podłączeniu urządzenia do portu USB komputera należy nacisnąć przycisk *Update MLink Firmware for M3* – o postępach aktualizacji poinformuje nas pasek statusu na dole okna programu, a na koniec zostanie wyświetlony stosowny komunikat.



Rysunek 6. Zawartość folderu po rozpakowaniu paczki OCDM3 MLink

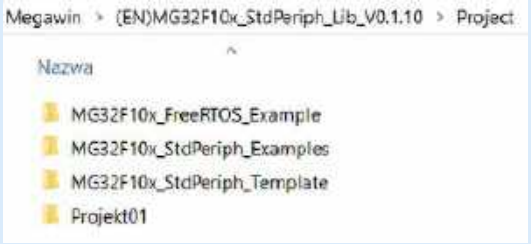
Tworzenie pierwszego projektu

Aby utworzyć projekt, można przejść do menu *Project* → *New µVision Project…*, a następnie ręcznie przeprowadzić cały proces konfiguracji opcji toolchaina, dodawania bibliotek i mozolnego ustawiania wszystkich pozostałych elementów. Na szczęście inżynierowie aplikacyjni z firmy Megawin wykonali lwią część pracy za nas, udostępniając kompletny pakiet wsparcia – w tym także gotowy szablon projektu, w pełni skonfigurowany zgodnie z wymogami środowiska Keil. Grzechem byłoby więc nie skorzystać z tego udogodnienia. Paczkę z niezbędnym repozytorium znajdziemy na podanej wcześniej stronie internetowej, tym razem jednak musimy wybrać link *Sample Code – Cortex-M3* → *MG32F10x*. Archiwum warto umieścić w folderze o możliwie krótkiej ścieżce dostępu, a nazwę samej paczki dodatkowo skrócić, np. do postaci *Megawin* – wynika to z faktu, iż system Windows w niektórych przypadkach nie radzi sobie z wypakowaniem folderu skompresowanego (z uwagi na zbyt długie ścieżki dostępu niektórych znajdujących się w nim plików – ot, drobna pułapka, którą twórca słynnego EEVBlog nazwałby zapewne mianem „a trap for young players").

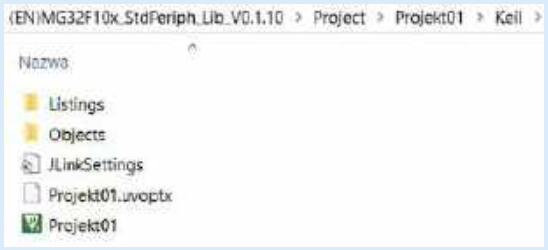
Po rozpakowaniu i otwarciu archiwum ujrzymy folder zawierający cztery główne podkatalogi:

- **Libraries** – zawierający bibliotekę CMSIS, sterownik podstawowych bloków peryferyjnych (*MG32F10x_StdPeriph_Driver*) oraz sterownik interfejsu USB (*MG32F10x_USBDevice_Driver*).
- **MG32F10x_Peripherals_Library_Manual** – przechowujący dokumentację sterownika w formacie HTML.
- **Project** – zawierający trzy kolejne podfoldery z prostym projektem na bazie systemu FreeRTOS, bogatym zestawem przykładowych projektów korzystających ze sterownika *StdPeriph* dla poszczególnych peryferiów MCU oraz to, co interesuje nas teraz najbardziej: szablon projektu (***MG32F10x_StdPeriph_Template***).
- **Utilities** – zestaw dodatkowych plików źródłowych, m.in. z funkcjami ułatwiającymi wysokopoziomową obsługę interfejsu UART.

Pozornie najprostszą metodą na zduplikowanie szablonu byłoby przekopiowanie wspomnianego wcześniej folderu *MG32F10x_StdPeriph_Template* do wybranej przez nas lokalizacji i zmiana nazwy tak utworzonej kopii, np. na *Projekt01*. Ponieważ jednak w strukturze plików konfiguracyjnych projektu (tj. **.uvprojx* oraz **.uvoptx*, przechowujących wszystkie niezbędne ustawienia), podane są ścieżki względne do folderów bibliotecznych, taka operacja spowodowałaby, że Keil miałby problem ze znalezieniem odpowiednich plików. Dlatego też – dla ułatwienia – proponuję umieścić folder-kopię tuż obok oryginału. Wtedy wystarczy zmienić nazwę folderu oraz wspomnianych dwóch plików, podmieniając oryginalny ciąg znaków np. na *Projekt01*. Po wykonaniu tej operacji możemy otworzyć projekt, klikając dwukrotnie ikonę pliku *Projekt01.uvprojx* w oknie Eksploratora. Dla pewności warto spojrzeć na widoki zawartości poszczególnych folderów, zaprezentowane na **rysunkach 7** i **8**.



Rysunek 7. Struktura folderu po rozpakowaniu paczki z biblioteką StdPeriph



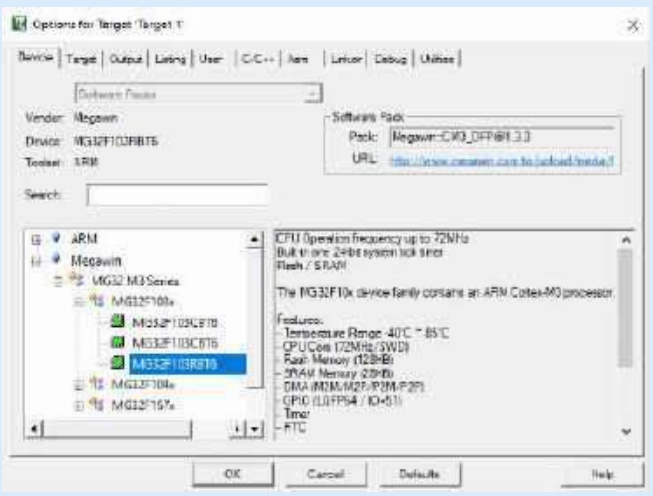
Rysunek 8. Zawartość folderu z projektem utworzonym na bazie szablonu (po zmianie domyślnych nazw plików)

Po otwarciu środowiska Keil powinniśmy skonfigurować kilka ustawień, które w domyślnej formie mogą uniemożliwić poprawną kompilację i uruchomienie projektu. Najpierw, używając przycisku znajdującego się na najważniejszym z naszego punktu widzenia pasku narzędziowym środowiska Keil (**rysunek 9**), otwieramy okno *Options for Target*, w którym wybieramy zakładkę *Device*. Tutaj musimy zmienić domyślny model procesora z *MG32F103**C9T6*** na *MG32F103**RBT6*** (**rysunek 10**).

Kolejna modyfikacja dotyczy wyboru kompilatora, którego użyjemy do zbudowania naszego projektu (zakładka *Target*). Znajdujący się w pobranym archiwum projekt został opracowany w wersji 5, zaś aktualna (w chwili pisania niniejszego artykułu) jest wersja 6 kompilatora (a dokładniej 6.21) – dlatego właśnie zmieniamy stosowne ustawienie (**rysunek 11**). Na kolejnej zakładce zatytułowanej *Output* (**rysunek 12**) możemy ponadto wpisać własną nazwę pliku wyjściowego – w celu zachowania porządku warto podać nazwę *Projekt01*.

Bardzo ważne ustawienia czekają na nas także w zakładce *C/C++ (AC6)*. Tutaj, w polu tekstowym *Define* (**rysunek 13**), możemy ustawić potrzebne wartości częstotliwości zegara systemowego oraz rezonatora kwarcowego, współpracującego z HSE. Ponieważ jednak na naszej płytce ewaluacyjnej rezonator XC1 ma częstotliwość 12 MHz,

Rysunek 9. Główny pasek narzędzi środowiska Keil MDK



Rysunek 10. Widok zakładki *Device* okna *Options for Target*

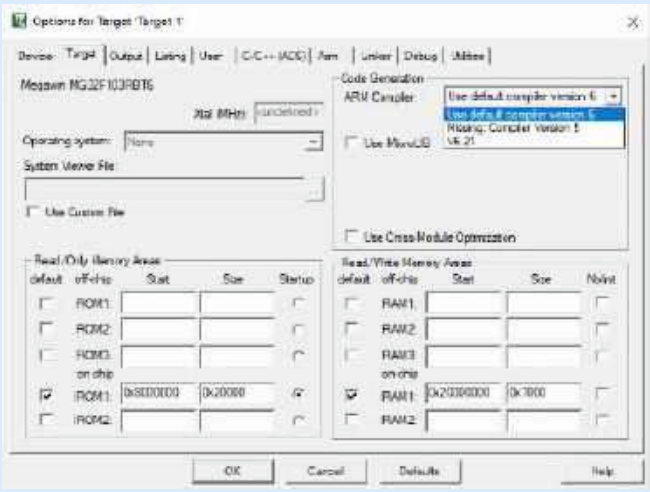
a w przypadku samego MCU chcemy pracować z domyślną, maksymalną częstotliwością 72 MHz, to możemy z czystym sumieniem pozostawić takie właśnie domyślne ustawienia szablonu. Warto natomiast przestawić poziom optymalizacji kompilatora na -O0.

Po wykonaniu wszystkich opisanych powyżej operacji należy jeszcze przejść do konfiguracji interfejsu sprzętowego. W tym celu – w zakładce *Debug* – sprawdzamy, czy programator na pewno jest ustawiony na opcję *MLink Cortex-M3 Debugger* (**rysunek 14**). Należy przy tym zwrócić uwagę, by zaznaczone były opcje *Load Application at Startup* oraz *Run to main()*.

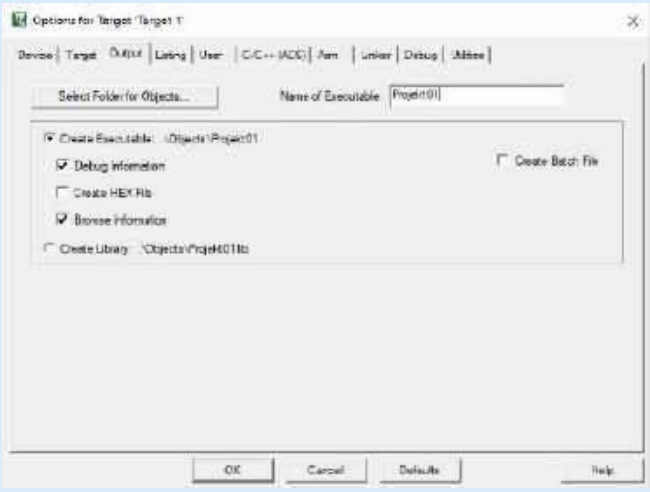
Poprawność rozpoznania sprzętu możemy zweryfikować, klikając przycisk *Settings* – w oknie po prawej stronie powinniśmy zobaczyć kod ID oraz nazwę urządzenia (**rysunek 15**).

Jeżeli wszystko działa poprawnie, przechodzimy do zakładki *Flash Download* (**rysunek 16**).

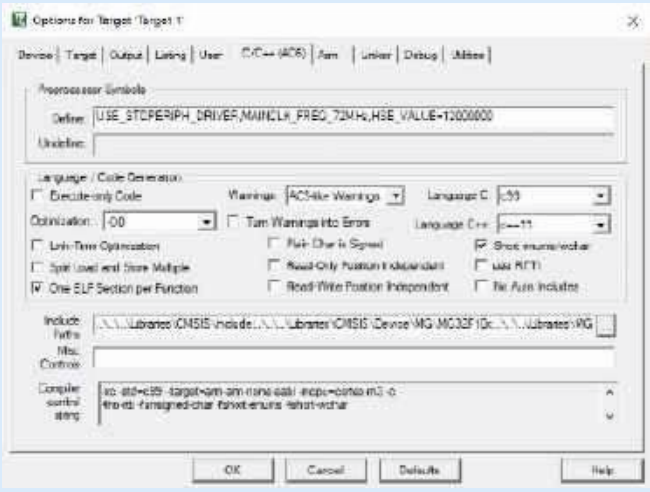
W tym momencie należy upewnić się, że zaznaczone są opcje *Erase Sectors*, *Program* oraz *Verify*, a następnie zaznaczyć domyślnie wyłączoną funkcję *Reset and Run*. Koniecznie należy także sprawdzić, czy w dolnej części okna widoczny jest algorytm wgrywania kodu maszynowego do pamięci procesora (*Programming Algorithm*).



Rysunek 11. Widok zakładki *Target* okna *Options for Target*

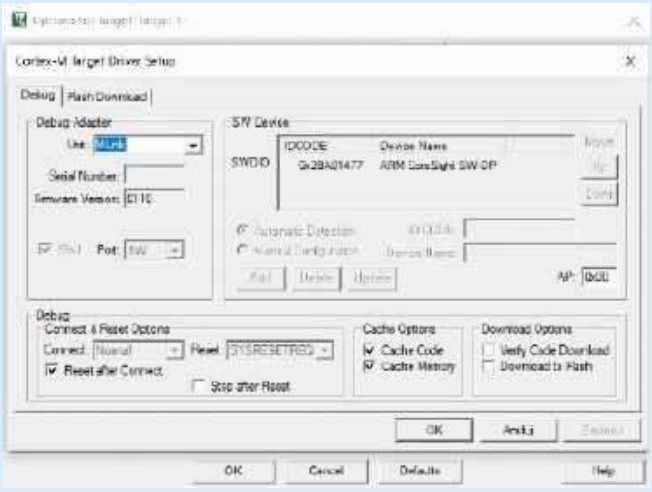


Rysunek 12. Widok zakładki *Output* okna *Options for Target*

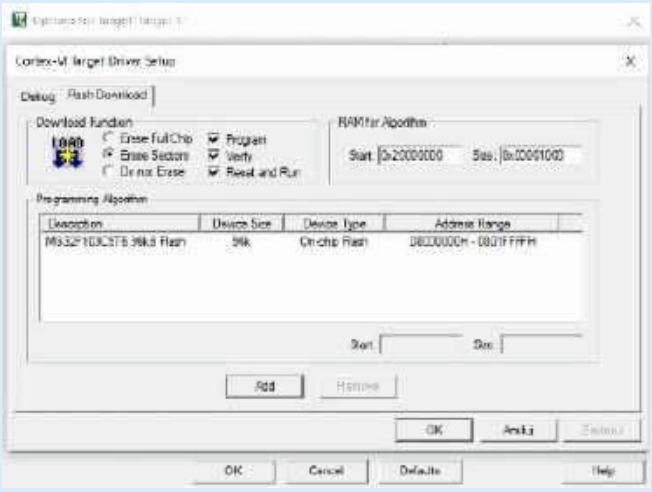


Rysunek 13. Widok zakładki *C/C++ (AC6)* okna *Options for Target*

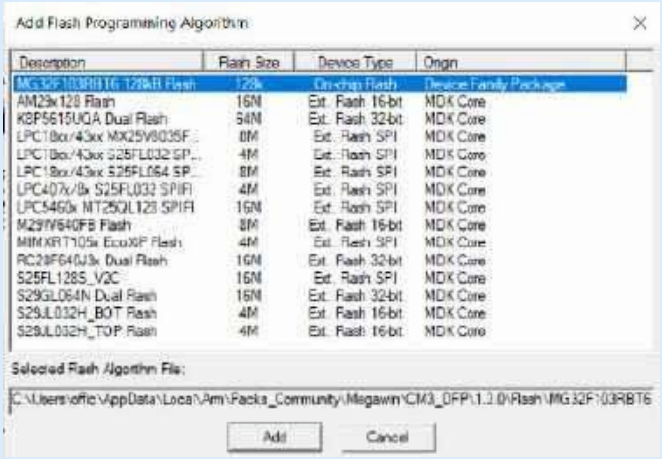
Rysunek 14. Widok zakładki *Debug* okna *Options for Target*



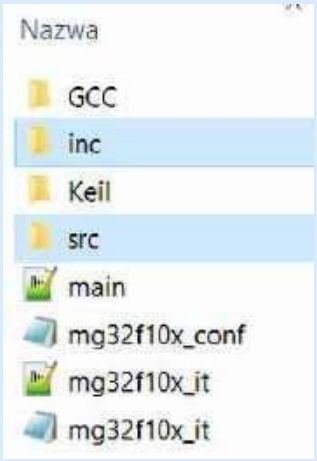
Rysunek 15. Widok okna ustawień interfejsu MLink, zakładka *Debug*



Rysunek 16. Okno ustawień interfejsu MLink, zakładka *Flash Download*



Rysunek 17. Widok okna wyboru algorytmu programowania pamięci Flash procesora



Rysunek 18. Struktura folderu projektu po dodaniu podkatalogów *inc* i *src*

Jeżeli tak nie jest – należy kliknąć przycisk *Add*, a następnie wybrać opcję odpowiadającą naszemu mikrokontrolerowi (**rysunek 17**).

Teraz możemy już przejść do edycji głównego pliku źródłowego, czyli *main.c*. W tym celu rozwijamy folder *User* w drzewie projektu, znajdującym się po lewej stronie okna środowiska Keil. Po dwukrotnym kliknięciu tej pozycji naszym oczom powinno ukazać się okno edytora z domyślnym kodem źródłowym.

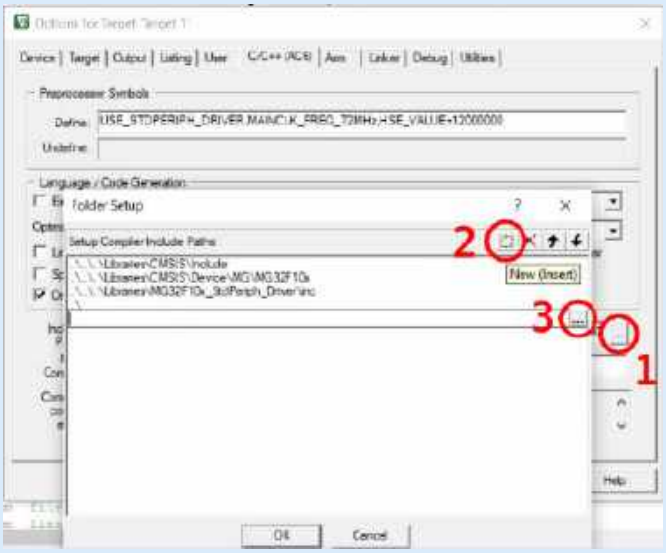
Pierwszy program

Trudno wyobrazić sobie tutorial programowania mikrokontrolerów – niezależnie od tego, której platformy dotyczy – bez napisania pierwszego programu w postaci klasycznego „blinky", czyli prostego migania diodą LED. Aby stało się zadość tradycji, w naszym kursie również zaczniemy od takiego właśnie zadania.

Zanim jednak przejdziemy do pisania właściwego kodu z użyciem instrukcji z biblioteki StdPeriph, na początek utwórzmy plik źródłowy z definicjami makr ułatwiających obsługę linii GPIO procesora. Na **listingu 1** można zobaczyć stosowne oznaczenia portów oraz numerów ich linii, które odpowiadają za obsługę poszczególnych obwodów peryferyjnych w naszym zestawie ewaluacyjnym.

Środowisko Keil nie należy niestety do najbardziej intuicyjnych pod względem sposobu dodawania plików bibliotecznych – dlatego część pracy trzeba wykonać ręcznie. Na szczęście operacji tych jest stosunkowo niewiele – na początek utwórzmy dwa dodatkowe foldery w głównym katalogu naszego projektu (**rysunek 18**) i nazwijmy je następująco: ***inc*** oraz ***src***. Jak nietrudno się domyślić, pierwszy z nich będzie przechowywał pliki nagłówkowe programu, drugi zaś – właściwe pliki z kodem źródłowym. Po utworzeniu katalogów trzeba jeszcze wskazać Keilowi, gdzie ma on szukać plików bibliotecznych – w tym celu uruchamiamy ponownie okno *Options for Target*, tym razem jednak dopisujemy nową ścieżkę dostępu w sposób opisany na **rysunku 19**. Po zatwierdzeniu możemy już utworzyć pusty plik źródłowy, klikając prawym klawiszem myszy folder wirtualny *User* (w drzewie projektu), następnie wybierając opcję *Add New Item to Group 'User'*, a w dalszej kolejności *Header File (.h)* i – na sam koniec – wpisując nazwę pliku (np. *hwconfig.h*) w polu *Name* (**rysunek 20**). Plik należy umieścić w folderze *inc*. Teraz możemy już przepisać (lub – w celu zaoszczędzenia czasu – wkleić) odpowiedni kod (zamieszczony w materiałach źródłowych do niniejszego artykułu) do otwartego w ten sposób, pustego pliku w oknie edytora IDE. Plik automatycznie pojawi się w drzewie projektu.

W podobny sposób dodajemy jeszcze dwa pliki: jeden źródłowy (*hardware.c*) – który trafi do folderu *src* – oraz odpowiadający mu nagłówek (*hardware.h*) – w katalogu *inc*. W plikach tych będziemy umieszczać funkcje obsługi peryferiów sprzętowych mikrokontrolera (oraz deklaracje tychże funkcji) – na początek



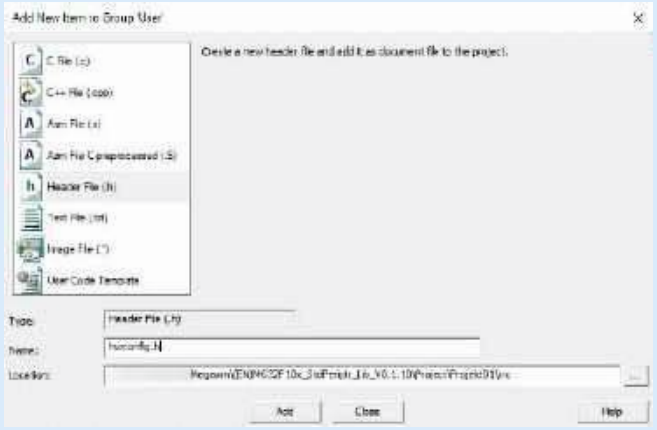
Rysunek 19. Kolejność wykonywania operacji podczas dodawania folderu z plikami nagłówkowymi: (1) kliknij przycisk z wielokropkiem, znajdujący się po prawej stronie pola tekstowego *Include* na zakładce *C/C++(AC6)* okna *Options for Target*, (2) – kliknij przycisk *New (Insert)* znajdujący się w nowo otwartym oknie *Folder Setup*, (3) – kliknij przycisk z wielokropkiem po prawej stronie ostatniej linii znajdującej się na liście folderów. Po ostatniej z wymienionych czynności zostanie otwarte standardowe okno menedżera plików Windows, pozwalające na ręczne wskazanie ścieżki do folderu z plikami nagłówkowymi

```
void initPeripherals(void){

    /* Procedura inicjalizujaca glowne peryferia sprzetowe MCU */

    /* Wlaczenie taktowania GPIO i szyny APB1 */
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_BMX1 |
                                          RCC_APB1Periph_GPIOA |
                                          RCC_APB1Periph_GPIOB |
                                          RCC_APB1Periph_GPIOC |
                                          RCC_APB1Periph_GPIOD, ENABLE);

    /* Konfiguracja linii GPIO do obslugi diod LED - wyjscia niepodciagane do VCC/GND, niska predkosc */

    GPIO_Init(LED1_port, LED1_pin, GPIO_MODE_OUT | GPIO_OTYPE_PP | GPIO_PUPD_NOPULL | GPIO_SPEED_LOW);
    GPIO_Init(LED2_port, LED2_pin, GPIO_MODE_OUT | GPIO_OTYPE_PP | GPIO_PUPD_NOPULL | GPIO_SPEED_LOW);

}
```

Listing 1.

```
#ifndef __HWCONFIG_H
#define __HWCONFIG_H

#include "mg32f10x.h"

/* diody LED1 i LED2 */

#define LED1_port    GPIOB
#define LED1_pin GPIO_Pin_13

#define LED2_port    GPIOB
#define LED2_pin GPIO_Pin_12

/* wyswietlacz LED */

#define DISP_A_port  GPIOB
#define DISP_A_pin   GPIO_Pin_7

#define DISP_B_port  GPIOC
#define DISP_B_pin   GPIO_Pin_11

#define DISP_C_port  GPIOD
#define DISP_C_pin   GPIO_Pin_2

#define DISP_D_port  GPIOB
#define DISP_D_pin   GPIO_Pin_4

#define DISP_E_port  GPIOB
#define DISP_E_pin   GPIO_Pin_5

#define DISP_F_port  GPIOB
#define DISP_F_pin   GPIO_Pin_6

#define DISP_G_port  GPIOC
#define DISP_G_pin   GPIO_Pin_12

#define DISP_DP_port GPIOB
#define DISP_DP_pin  GPIO_Pin_3

#define DISP_COM1_port   GPIOC
#define DISP_COM1_pin    GPIO_Pin_7

#define DISP_COM2_port   GPIOC
#define DISP_COM2_pin    GPIO_Pin_8

#define DISP_COM3_port   GPIOC
#define DISP_COM3_pin    GPIO_Pin_9

#define DISP_COM4_port   GPIOC
#define DISP_COM4_pin    GPIO_Pin_6

/* wyjscie potencjometru */

#define POT_port GPIOC
#define POT_pin  GPIO_Pin_4

/* interfejs UART */

#define UART_TX_port GPIOA
#define UART_TX_pin  GPIO_Pin_9

#define UART_RX_port GPIOA
#define UART_RX_pin  GPIO_Pin_10

/* interfejs USB */

#define USB_DM_port  GPIOA
#define USB_DM_pin   GPIO_Pin_11

#define USB_DP_port  GPIOA
#define USB_DP_pin   GPIO_Pin_12

/* interfejs I²C */

#define I²C_SCL_port GPIOB
#define I²C_SCL_pin  GPIO_Pin_10

#define I²C_SDA_port GPIOB
#define I²C_SDA_pin  GPIO_Pin_11

/* interfejs SPI */

#define SPI_CS_port  GPIOC
#define SPI_CS_pin   GPIO_Pin_0

#define SPI_MOSI_port    GPIOC
#define SPI_MOSI_pin GPIO_Pin_3

#define SPI_MISO_port    GPIOC
#define SPI_MISO_pin GPIO_Pin_2

#define SPI_SCLK_port    GPIOC
#define SPI_SCLK_pin GPIO_Pin_1

/* przyciski */

#define SW1_port GPIOB
#define SW1_pin  GPIO_Pin_0

#define SW2_port GPIOC
#define SW2_pin  GPIO_Pin_5

#endif
```

Listing 2.



Rysunek 20. Okno dodawania nowego pliku nagłówkowego

dodajmy zatem procedurę inicjalizującą linie GPIO odpowiedzialne za sterowanie diodami LED. Ciało funkcji pokazano na **listingu 2**. Osoby, które pracowały na mikrokontrolerach STM32 jeszcze przed erą HAL (tj. w czasie, gdy obowiązującym standardem była jeszcze biblioteka *Standard Peripherals Library*), zauważą nieprzypadkowe podobieństwo zastosowanych tutaj funkcji obsługujących bloki RCC i GPIO do tych znanych z STM32 – istotnie, ta część biblioteki przeznaczonej do mikrokontrolerów Megawin w dużej mierze bazuje właśnie na bardzo zbliżonej koncepcji. Warto jednak zwrócić uwagę, że większość pozostałych peryferiów ma częściowo lub nawet całkowicie inną konstrukcję niż odpowiadające im peryferia procesorów STM32F1, stąd bezpośrednie przeniesienie kodu projektów napisanych na STM32 będzie w znakomitej większości przypadków niemożliwe.

Garść wyjaśnień może być niezbędna osobom, które wcześniej nie miały styczności z programowaniem mikrokontrolerów STM32. Polecenie `RCC_APB1PeriphClockCmd()` z parametrem `ENABLE` ma za zadanie włączyć wewnętrzne obwody taktowania, które są niezbędne do uruchomienia bloków peryferyjnych – w tym przypadku portów od `GPIOA` do `GPIOD`, obsługiwanych przez wewnętrzną szynę APB1. Odpowiednie maski o postaci `RCC_APB1Periph_GPIOx` (gdzie x to oznaczenie portu A...D) są sumowane logicznie i tworzą pierwszy parametr funkcji `RCC_APB1PeriphClockCmd()`. Dodatkowo w sumie znalazła się także obligatoryjna flaga `RCC_APB1Periph_BMX1`, odpowiedzialna za uruchomienie taktowania szyny APB1 (**B**us **M**atri**X** **1**).

Druga zastosowana w tym fragmencie kodu funkcja z biblioteki StdPeriph ma za zadanie skonfigurować daną linię portu GPIO. W tym celu należy podać w wywołaniu funkcji `GPIO_Init()` trzy kolejne parametry:

- wskaźnik na strukturę umożliwiającą dostęp do danego portu (typu `GPIO_TypeDef*` – np. `GPIOA`),
- numer portu w postaci zgodnej z zapisem stosowanym w bibliotece (uwaga – nie stosujemy tutaj po prostu numeru określonej linii `GPIO`, ale używamy aliasu, przykładowo: aby odwołać się do PA8, podajemy parametr `GPIO_Pin_8`, który w istocie jest aliasem liczby 0x100, czyli… bitu o wartości 1 umieszczonego na pozycji 9),
- wartość konfiguracyjną, w której poszczególne elementy (zsumowane logicznie) określają koleje parametry danej linii, np.:
  - `GPIO_MODE_OUT` – pin I/O pracuje jako wyjście cyfrowe ogólnego przeznaczenia,
  - `GPIO_OTYPE_PP` – wyjście jest typu push-pull,
  - `GPIO_PUPD_NOPULL` – nie stosujemy rezystorów podciągających do VCC ani GND,
  - `GPIO_SPEED_LOW` – linia będzie pracowała ze stosunkowo małą częstotliwością przełączania (ten parametr pozwala uzyskać optymalny balans pomiędzy poziomem zakłóceń EMI

generowanych podczas przełączania a szybkością narastania i opadania zboczy sygnału).

Dla jeszcze większej wygody programowania naszego procesora napiszmy także prosty wrapper do obsługi znajdujących się na płytce ewaluacyjnej diod LED D1 i D2 oraz prostą funkcję opóźniającą – ciała obydwu procedur można zobaczyć na **listingu 3**. Uzbrojeni w najważniejsze fragmenty kodu możemy w końcu przejść do głównej funkcji programu, zaprezentowanej na **listingu 4**.

Jeżeli wszystkie dotychczasowe etapy zostały wykonane poprawnie, możemy skompilować nasz pierwszy program, klikając przycisk *Build* lub naciskając klawisz funkcyjny F7. O prawidłowym zakończeniu budowania pliku binarnego świadczyć będzie komunikat w konsoli *Build Output*:

```
„.\Objects\Projekt01.axf" – 0 Error(s), 0 Warning(s).
```

Podłączenie programatora do płytki i wgranie pierwszego programu

Prawidłowe podłączenie programatora MLink do złącza DBG płytki ewaluacyjnej pokazano na **fotografii 2**. Jak widać, skrajny „górny" pin wtyku goldpin interfejsu MLink (oznaczony literami CLK) należy pozostawić niepodłączony – dla ułatwienia w złączu DBG przewidziano odpowiadający mu styk, także niepodłączony do żadnego z sygnałów procesora. Wszystkie pozostałe linie są łączone w tej samej kolejności (w przypadku obydwu

```
void LED_onoff(uint8_t led, LED_state_t state){

    /* Wrapper do obslugi diod LED */
    /* led – numer diody (1 lub 2) */
    /* state – nowy stan diody (LED_ON lub LED_OFF) */

    if(led == 1){

            if(state == LED_ON)
                    GPIO_ResetBits(LED1_port, LED1_pin);
            else
                    GPIO_SetBits(LED1_port, LED1_pin);

    }else if(led == 2){

            if(state == LED_ON)
                    GPIO_ResetBits(LED2_port, LED2_pin);
            else
                    GPIO_SetBits(LED2_port, LED2_pin);

    }
}

void delay(uint32_t ms){

    /* Prosta funkcja zatrzymujaca procesor na okolo 1 ms */

    for(uint32_t i = 0; i < 3600; i++){
            for(uint32_t j = 0; j < ms; j++) __NOP();
    }

}
```

Listing 3.

```
int main(void)
{
    /* Inicjalizacja peryferiow */
    initPeripherals();

    /* Wylaczenie obu diod LED */
    LED_onoff(1, LED_OFF);
    LED_onoff(2, LED_OFF);

    while (1){

        LED_onoff(1, LED_ON);
        LED_onoff(2, LED_OFF);

        delay(500);

        LED_onoff(1, LED_OFF);
        LED_onoff(2, LED_ON);

        delay(500);

    }
}
```

Listing 4.

```
void display_select_pos(uint8_t pos){

    /* obsluga pozycji dziesietnych wyswietlacza (wspolne anody) */

    switch(pos){

        case 0:
            GPIO_ResetBits(DISP_COM1_port, DISP_COM1_pin);
            GPIO_SetBits(DISP_COM2_port, DISP_COM2_pin);
            GPIO_SetBits(DISP_COM3_port, DISP_COM3_pin);
            GPIO_SetBits(DISP_COM4_port, DISP_COM4_pin);
            break;

        case 1:
            GPIO_SetBits(DISP_COM1_port, DISP_COM1_pin);
            GPIO_ResetBits(DISP_COM2_port, DISP_COM2_pin);
            GPIO_SetBits(DISP_COM3_port, DISP_COM3_pin);
            GPIO_SetBits(DISP_COM4_port, DISP_COM4_pin);
            break;

        case 2:
            GPIO_SetBits(DISP_COM1_port, DISP_COM1_pin);
            GPIO_SetBits(DISP_COM2_port, DISP_COM2_pin);
            GPIO_ResetBits(DISP_COM3_port, DISP_COM3_pin);
            GPIO_SetBits(DISP_COM4_port, DISP_COM4_pin);
            break;

        case 3:
            GPIO_SetBits(DISP_COM1_port, DISP_COM1_pin);
            GPIO_SetBits(DISP_COM2_port, DISP_COM2_pin);
            GPIO_SetBits(DISP_COM3_port, DISP_COM3_pin);
            GPIO_ResetBits(DISP_COM4_port, DISP_COM4_pin);
            break;

        default:
            GPIO_SetBits(DISP_COM1_port, DISP_COM1_pin);
            GPIO_SetBits(DISP_COM2_port, DISP_COM2_pin);
            GPIO_SetBits(DISP_COM3_port, DISP_COM3_pin);
            GPIO_SetBits(DISP_COM4_port, DISP_COM4_pin);

    }
}

void display_select_digit(uint8_t dig){

    /* obsluga segmentow wyswietlacza (katody) */

    switch(dig){

        case 0:
            GPIO_SetBits(DISP_A_port, DISP_A_pin);
            GPIO_SetBits(DISP_B_port, DISP_B_pin);
            GPIO_SetBits(DISP_C_port, DISP_C_pin);
            GPIO_SetBits(DISP_D_port, DISP_D_pin);
            GPIO_SetBits(DISP_E_port, DISP_E_pin);
            GPIO_SetBits(DISP_F_port, DISP_F_pin);
            GPIO_ResetBits(DISP_G_port, DISP_G_pin);
            GPIO_ResetBits(DISP_DP_port, DISP_DP_pin);
            break;

        case 1:
            GPIO_ResetBits(DISP_A_port, DISP_A_pin);
            GPIO_SetBits(DISP_B_port, DISP_B_pin);
            GPIO_SetBits(DISP_C_port, DISP_C_pin);
            GPIO_ResetBits(DISP_D_port, DISP_D_pin);
            GPIO_ResetBits(DISP_E_port, DISP_E_pin);
            GPIO_ResetBits(DISP_F_port, DISP_F_pin);
            GPIO_ResetBits(DISP_G_port, DISP_G_pin);
            GPIO_ResetBits(DISP_DP_port, DISP_DP_pin);
            break;

        case 2:
            GPIO_SetBits(DISP_A_port, DISP_A_pin);
            GPIO_SetBits(DISP_B_port, DISP_B_pin);
            GPIO_ResetBits(DISP_C_port, DISP_C_pin);
            GPIO_SetBits(DISP_D_port, DISP_D_pin);
            GPIO_SetBits(DISP_E_port, DISP_E_pin);
            GPIO_ResetBits(DISP_F_port, DISP_F_pin);
            GPIO_SetBits(DISP_G_port, DISP_G_pin);
            GPIO_ResetBits(DISP_DP_port, DISP_DP_pin);
            break;

        case 3:
            GPIO_SetBits(DISP_A_port, DISP_A_pin);
            GPIO_SetBits(DISP_B_port, DISP_B_pin);
            GPIO_SetBits(DISP_C_port, DISP_C_pin);
            GPIO_SetBits(DISP_D_port, DISP_D_pin);
            GPIO_ResetBits(DISP_E_port, DISP_E_pin);
            GPIO_ResetBits(DISP_F_port, DISP_F_pin);
            GPIO_SetBits(DISP_G_port, DISP_G_pin);
            GPIO_ResetBits(DISP_DP_port, DISP_DP_pin);
            break;

        case 4:
            GPIO_ResetBits(DISP_A_port, DISP_A_pin);
            GPIO_SetBits(DISP_B_port, DISP_B_pin);
            GPIO_SetBits(DISP_C_port, DISP_C_pin);
            GPIO_ResetBits(DISP_D_port, DISP_D_pin);
            GPIO_ResetBits(DISP_E_port, DISP_E_pin);
            GPIO_SetBits(DISP_F_port, DISP_F_pin);
            GPIO_SetBits(DISP_G_port, DISP_G_pin);
            GPIO_ResetBits(DISP_DP_port, DISP_DP_pin);
            break;

        case 5:
            GPIO_SetBits(DISP_A_port, DISP_A_pin);
            GPIO_ResetBits(DISP_B_port, DISP_B_pin);
            GPIO_SetBits(DISP_C_port, DISP_C_pin);
```

Listing 5.

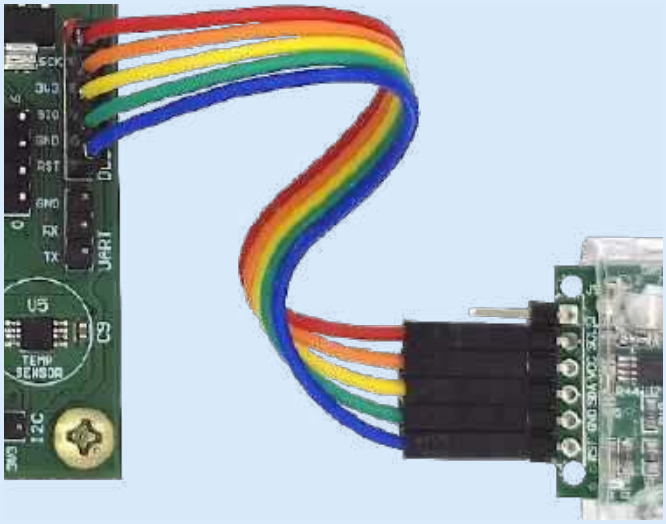
złączy). Do programowania można użyć kabla wielożyłowego (z izolacją wielokolorową lub jednobarwną) bądź zestawu pięciu przewodów żeńsko-żeńskich (z popularnymi wtykami typu DuPont/BLS).

Do zasilania płytki z powodzeniem wystarczy zastosować kabel mini USB, podłączony do jednego z wolnych portów USB komputera. Wszystkie ćwiczenia w ramach niniejszego kursu przewidziano w taki sposób, by nie było konieczne stosowanie zewnętrznego zasilacza – nawet podczas obsługi wyświetlacza LED zasilanie z portu USB, doprowadzone poprzez bezpiecznik polimerowy 100 mA, okazuje się w zupełności wystarczające. W przypadku korzystania z bardziej „prądożernych" modułów zewnętrznych (np. transceiverów GSM, Wi-Fi czy odbiornika GPS), które zasilanie pobierałyby ze złączy goldpin znajdujących się na płytce, należy oczywiście przełączyć zworkę VSEL na pozycję VEXT i podłączyć odpowiednie źródło energii do gniazda śrubowego VEXT.

Po zasileniu płytki i podłączeniu programatora jesteśmy w stanie wgrać skompilowany kod maszynowy do pamięci Flash mikrokontrolera – w tym celu klikamy przycisk *Download* lub naciskamy klawisz F8. Po chwili naszym oczom powinien ukazać się komunikat:

```
Erase Done.
Programming Done.
Verify OK.
Todo: HW Chip Reset after flash load
Application running…
Flash Load finished at (...)
```

Ponieważ w opcjach środowiska Keil zaznaczyliśmy opcję resetowania procesora po zakończeniu ładowania kodu do pamięci Flash, w tym momencie powinniśmy już obserwować naprzemienne miganie obu diod D1 i D2.



Fotografia 2. Sposób podłączenia programatora do płytki ewaluacyjnej

Obsługa wyświetlacza LED

Na koniec tej części kursu rozbudujmy jeszcze nasz program o obsługę wyświetlacza siedmiosegmentowego. Dla uproszczenia całość procedury multipleksowania zrealizujemy w nieszczególnie efektywny sposób blokujący – w tym momencie chodzi bowiem jedynie o przetestowanie poprawności montażu wyświetlacza, współpracujących z nim elementów dyskretnych oraz scalonego sterownika low-side, kontrolującego pracę segmentów (katod) wyświetlacza.

Całość obsługi 4-cyfrowego wskaźnika LED sprowadza się do cyklicznego wywoływania dwóch funkcji (**listing 5**):

```
            GPIO_SetBits(DISP_D_port, DISP_D_pin);
            GPIO_ResetBits(DISP_E_port, DISP_E_pin);
            GPIO_SetBits(DISP_F_port, DISP_F_pin);
            GPIO_SetBits(DISP_G_port, DISP_G_pin);
            GPIO_ResetBits(DISP_DP_port, DISP_DP_pin);
            break;

        case 6:
            GPIO_SetBits(DISP_A_port, DISP_A_pin);
            GPIO_ResetBits(DISP_B_port, DISP_B_pin);
            GPIO_SetBits(DISP_C_port, DISP_C_pin);
            GPIO_SetBits(DISP_D_port, DISP_D_pin);
            GPIO_SetBits(DISP_E_port, DISP_E_pin);
            GPIO_SetBits(DISP_F_port, DISP_F_pin);
            GPIO_SetBits(DISP_G_port, DISP_G_pin);
            GPIO_ResetBits(DISP_DP_port, DISP_DP_pin);
            break;

        case 7:
            GPIO_SetBits(DISP_A_port, DISP_A_pin);
            GPIO_SetBits(DISP_B_port, DISP_B_pin);
            GPIO_SetBits(DISP_C_port, DISP_C_pin);
            GPIO_ResetBits(DISP_D_port, DISP_D_pin);
            GPIO_ResetBits(DISP_E_port, DISP_E_pin);
            GPIO_ResetBits(DISP_F_port, DISP_F_pin);
            GPIO_ResetBits(DISP_G_port, DISP_G_pin);
            GPIO_ResetBits(DISP_DP_port, DISP_DP_pin);
            break;

        case 8:
            GPIO_SetBits(DISP_A_port, DISP_A_pin);
            GPIO_SetBits(DISP_B_port, DISP_B_pin);
            GPIO_SetBits(DISP_C_port, DISP_C_pin);
            GPIO_SetBits(DISP_D_port, DISP_D_pin);
            GPIO_SetBits(DISP_E_port, DISP_E_pin);
            GPIO_SetBits(DISP_F_port, DISP_F_pin);
            GPIO_SetBits(DISP_G_port, DISP_G_pin);
            GPIO_ResetBits(DISP_DP_port, DISP_DP_pin);
            break;

        case 9:
            GPIO_SetBits(DISP_A_port, DISP_A_pin);
            GPIO_SetBits(DISP_B_port, DISP_B_pin);
            GPIO_SetBits(DISP_C_port, DISP_C_pin);
            GPIO_SetBits(DISP_D_port, DISP_D_pin);
            GPIO_ResetBits(DISP_E_port, DISP_E_pin);
            GPIO_SetBits(DISP_F_port, DISP_F_pin);
            GPIO_SetBits(DISP_G_port, DISP_G_pin);
            GPIO_ResetBits(DISP_DP_port, DISP_DP_pin);
            break;

        default:
            GPIO_ResetBits(DISP_A_port, DISP_A_pin);
            GPIO_ResetBits(DISP_B_port, DISP_B_pin);
            GPIO_ResetBits(DISP_C_port, DISP_C_pin);
            GPIO_ResetBits(DISP_D_port, DISP_D_pin);
            GPIO_ResetBits(DISP_E_port, DISP_E_pin);
            GPIO_ResetBits(DISP_F_port, DISP_F_pin);
            GPIO_ResetBits(DISP_G_port, DISP_G_pin);
            GPIO_ResetBits(DISP_DP_port, DISP_DP_pin);

    }

}
```

Listing 5. cd.

```
int main(void)
{

    int16_t cnt = 0;
    int16_t number = 0;
    uint8_t pos = 0;

    initPeripherals();

    LED_onoff(1, LED_OFF);
    LED_onoff(2, LED_OFF);

    while (1)
    {
        /* odczyt stanu przycisku SW1 (1 = niewcisniety, 0 = wcisniety) */
        /* Nacisniecie SW1 zatrzymuje odliczanie */
        if(GPIO_ReadInputDataBit(SW1_port, SW1_pin))
            cnt++;

        if(cnt > 100){

            cnt = 0;

            /* Nacisniecie SW2 przyspiesza odliczanie */
            if(GPIO_ReadInputDataBit(SW2_port, SW2_pin))
                number++;
            else
                number += 10;

            /* Obsluga przepelnienia licznika */
            if(number > 9999) number = 0;

        }

        int8_t digits[4];

        /* Pozyskanie kolejnych pozycji dziesietnych */
        digits[0] = (number % 10000) / 1000;
        digits[1] = (number % 1000) / 100;
        digits[2] = (number % 100) / 10;
        digits[3] = (number % 10);

        /* Tymczasowe wylaczenie wyswietlacza */
        display_select_pos(255);

        /* Ustawienie cyfry do wyswietlenia na katodach */
        display_select_digit(digits[pos]);

        /* Wlaczenie wybranej pozycji */
        display_select_pos(pos);

        /* Obsluga licznika pozycji dziesietnych */
        pos++;
        if(pos > 3) pos = 0;

        delay(1);

    }

}
```

Listing 6.

- `display_select_pos(pos)`, która jako parametr `pos` przyjmuje numer pozycji (0 – skrajna cyfra z lewej, 3 – skrajna cyfra z prawej),
- `display_select_digit(dig)`, która w roli parametru `dig` pobiera aktualną cyfrę (0...9) do wyświetlenia na danej pozycji.

Funkcje te są wywoływane w pętli nieskończonej, w ramach której program „rozkłada" aktualizowany okresowo numer do wyświetlenia (0000...9999) na cztery cyfry, odpowiadające kolejnym pozycjom dziesiętnym wyświetlacza. Całkowite wygaszenie wyświetlacza (niezbędne do prawidłowego odświeżenia jego zawartości) następuje przed ustawieniem cyfry i jest realizowane poprzez wywołanie funkcji `display_select_pos()` z parametrem 255 – w istocie można tu zastosować dowolną liczbę spoza „legalnego" zakresu 0...3, gdyż każdy wybór nienależący do tego przedziału zostanie i tak obsłużony przez sekcję `default` instrukcji warunkowej switch.

W celu urozmaicenia (a także zobrazowania sposobu obsługi linii wejściowych GPIO) dodano ponadto obsługę dwóch znajdujących się na płytce przycisków SW1 i SW2. Naciśnięcie pierwszego z nich tymczasowo blokuje odliczanie, ale nie wpływa na pracę systemu multipleksowania segmentów wyświetlacza. Wciśnięcie SW2 powoduje natomiast przyspieszenie odliczania – zamiast domyślnej inkrementacji (`cnt++`) otrzymujemy bowiem możliwość „przeskakiwania" co 10 zliczeń. Zastosowana w tych dwóch przypadkach funkcja odczytu stanu linii I/O – `GPIO_ReadInputDataBit()` – raczej nie wymaga większych wyjaśnień. Dodajmy tylko dla porządku, że zwracana przez nią wartość uint8_t odpowiada rzeczywistemu stanowi logicznemu danego bitu (0 lub 1). Ciało głównej funkcji programu pokazano na **listingu 6**.

Podsumowanie

W pierwszym odcinku naszego kursu nauczyliśmy się, jak zainstalować i skonfigurować środowisko Keil MDK do pracy z mikrokontrolerami Megawin z serii MG32F103. Napisaliśmy także dwa pierwsze programy, pozwalające na przetestowanie najprostszych spośród znajdujących się na naszej płytce ewaluacyjnej elementów sprzętowych: diod LED, wyświetlacza 7-segmentowego oraz przycisków. W kolejnych odcinkach pokażemy tajniki obsługi interfejsów I²C, SPI, UART, a także przetwornika ADC i wbudowanego zegara czasu rzeczywistego (RTC).

inż. Przemysław Musz, EP

Komplet materiałów dodatkowych do tego odcinka kursu programowania mikrokontrolerów Megawin można pobrać z serwera „Elektroniki Praktycznej" – https://ep.com.pl/.

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie **https://ulubionykiosk.pl/media**

Kurs FPGA Lattice (19)

Odbiornik UART

W poprzednim odcinku nauczyliśmy się, jak wykonać nadajnik interfejsu UART. Czas dowiedzieć się, jak zbudować odbiornik, aby opanować komunikację dwukierunkową. Modułów opracowanych w tej i poprzedniej części będziemy używać również w kolejnych odcinkach kursu.

Informacje: co to jest UART, jak działa i jakie są jego możliwości – zostały podane w 18 części kursu. Jeżeli jej nie czytałeś, gorąco zachęcam, by zapoznać się z nią przed przystąpieniem do dzisiejszego odcinka.

Trochę teorii

Omówmy najpierw, w jaki sposób funkcjonuje odbiornik UART. Zobacz **rysunek 1**. Na zielono zaznaczono sygnał na wyjściu odbiornika. W stanie spoczynkowym linia transmisyjna ma stan wysoki.

Transmisja rozpoczyna się bitem startu, który zawsze ma stan niski. Wywołuje on zbocze opadające, a zjawisko takie ma za zadanie poinformować odbiornik, że zaczyna się ramka transmisyjna i za chwilę zostaną przetransmitowane bity danych. Zaznaczono to czerwoną strzałką. W jaki sposób będziemy wykrywać zbocze opadające? Zastosujemy tutaj moduł **EdgeDetector**, z którego korzystaliśmy już wielokrotnie w poprzednich odcinkach kursu.

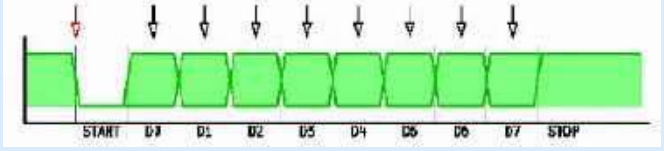
Sygnał z wykrywacza zbocza uruchomi moduł **StrobeGeneratorTicks**, opracowany w poprzednim odcinku na potrzeby nadajnika UART. Jego zadanie polega na okresowym wyznaczaniu impulsów strobe, które powodować będą odczytywanie wejścia linii transmisyjnej. Następnie stan wejścia będzie sukcesywnie zapisywany do stanu rejestru wyjściowego, skąd będzie można odczytać odebrany bajt danych.

Spójrz jeszcze raz na rysunek 1. Próbkowanie zaznaczono czarnymi strzałkami. W idealnej sytuacji badanie stanu wejścia następuje dokładnie w połowie trwania każdego bitu. Odstęp czasowy pomiędzy pobraniem poszczególnych próbek jest równy czasowi trwania każdego bitu. Jednak odstęp pomiędzy zboczem opadającym na początku bitu startu a pobraniem pierwszej próbki to czas trwania półtora bitu. Zatem proces próbkowania musimy opóźnić o pół bitu. Jak to zrobić? Możliwe są dwa rozwiązania:

1. Po wykryciu zbocza opadającego na linii transmisyjnej uruchamia się jednorazowo pierwszy timer, którego celem jest czekanie przez pół bitu. Następnie uruchamia on drugi timer, który odmierza odstępy czasu równe jednemu bitowi i pracuje do końca ramki transmisyjnej.
2. Po wykryciu zbocza opadającego na linii transmisyjnej uruchamia się timer, który pracuje przez cały czas trwania transmisji. Generuje on sygnały co pół bitu, lecz w czasie wysyłania bitów danych co drugi z tych sygnałów jest ignorowany.

W naszym module odbiornika zastosujemy rozwiązanie drugie.

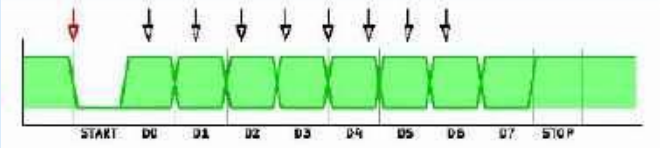
Kiedy już uzbieramy osiem próbek, odpowiadających ośmiu bitom, możemy zakończyć pracę modułu i zgłosić sygnał informujący o tym, że na wyjściu modułu odbiornika znajdują się dane gotowe do odczytania.



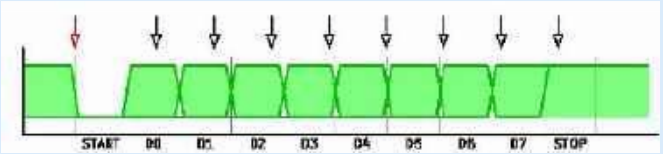
Rysunek 1. Próbkowanie poprawnie zsynchronizowane

Poprzednie odcinki znajdują się pod adresem:
https://ulubionykiosk.pl/media

Materiały do pobrania:
- Kompletny projekt w środowisku Lattice Diamond
  https://tiny.pl/dc9s2
- Repozytorium modułów stosowanych w kursie
  https://github.com/leonow32/verilog-fpga



Rysunek 2. Próbkowanie zbyt szybkie



Rysunek 3. Próbkowanie zbyt wolne

A co z bitem stopu? W gruncie rzeczy nie jest on nam do niczego potrzebny. Nie będziemy sprawdzać jego stanu ani go zapisywać (zawsze ma stan wysoki). Interesować nas będzie już tylko wykrycie zbocza opadającego, ponieważ zasygnalizuje ono rozpoczęcie transmisji kolejnej ramki danych.

Bardzo ważne jest, aby zegary nadajnika i odbiornika były dobrze zsynchronizowane. **Rysunek 2** ukazuje sytuację, w której próbkowanie następuje szybciej, niż powinno. W rezultacie bit czwarty próbkowany jest dwa razy, a bitu siódmego nie odczytujemy wcale. Prowadzi to do odebrania błędnych danych.

Rysunek 3 obrazuje odwrotną sytuację. Próbkowanie odbywa się zbyt wolno, przez co nadajnik kończy transmisję dużo wcześniej, niż odbiornik kończy odbieranie. W przykładowej sytuacji bit piąty nie jest odczytywany wcale, a bit stopu traktowany jest jako najstarszy bit danych.

Moduł UartRx

Przejdźmy teraz do omówienia kodu odbiornika UART, który pokazano na **listingu 1**. Moduł odbiornika konfigurujemy za pomocą dwóch parametrów, podobnie jak moduł nadajnika. Oprócz częstotliwości zegara, zdefiniowanej parametrem **CLOCK_HZ**, mamy także szybkość transmisji danych w bitach na sekundę, ustawianą parametrem **BAUD** (linia 1).

Moduł wyposażono w następujące porty wejściowe i wyjściowe:

- **Clock –** wejście zegara,
- **Reset –** wejście resetujące,
- **Rx_i –** wejście odbiornika, należy ten port połączyć z pinem układu FPGA,
- **Done_o –** wyjście informujące o odebraniu bajtu danych poprzez ustawienie stanu wysokiego na jeden takt zegarowy,
- **Data_o** – 8-bitowe wyjście odebranych danych.

Na początku mamy kilka zmiennych. W linii 2 tworzymy zmienną **Busy**. Stan wysoki tej zmiennej informuje, że moduł jest w trakcie odbierania danych. 9-bitowy rejestr **RxBuffer** (linia 3) posłuży do tymczasowego przechowywania odebranych bitów, zanim zostaną skopiowane na port wyjściowy po zakończeniu transmisji. Licznik **Counter** (linia 4) zliczać będzie impulsy generowane co pół bitu przez moduł **StrobeGeneratorTicks**. Licznik ten będzie liczyć od 0 do 17, co wymaga 5-bitowej rozdzielczości.

Zacznijmy od zsynchronizowania wejścia odbiornika z domeną zegarową. Należy pamiętać, że sygnały pochodzące z pinów FPGA mogą wywoływać stany metastabilne, jeżeli nie zostaną zsynchronizowane (ten temat szczegółowo omówiliśmy w 11 odcinku kursu).

W linii 6 tworzymy instancję modułu **Synchronizer**. Do jego wejścia doprowadzamy wejście **Rx_i** (linia 7), a wyjście wyprowadzamy (linia 8) za pomocą zmiennej wire **RxSync**, utworzonej w linii 5. Stan zmiennej **RxSync** jest taki sam, jak na wejściu **Rx_i**, ale z tą różnicą, że wszystkie zmiany stanu są zsynchronizowane z zegarem **Clock**. Zmienna **RxSync** będzie wielokrotnie używana w dalszej części kodu.

```
// Plik uart_rx.v

`default_nettype none
module UartRx #(
    parameter CLOCK_HZ = 10_000_000,
    parameter BAUD     = 115200                                  // 1
)(
    input wire Clock,
    input wire Reset,
    input wire Rx_i,
    output reg Done_o,
    output reg [7:0] Data_o
);

    // Zmienne
    reg Busy;                                                    // 2
    reg [8:0] RxBuffer;                                          // 3
    reg [4:0] Counter;                                           // 4

    // Synchronizacja wejścia Rx z domeną zegarową
    wire RxSync;                                                 // 5

    Synchronizer Synchronizer_Rx(                                // 6
        .Clock(Clock),
        .Reset(Reset),
        .Async_i(Rx_i),                                          // 7
        .Sync_o(RxSync)                                          // 8
    );

    // Rozpoznawanie początku ramki transmisyjnej
    // Wykrywanie bitu startu, który zawsze ma poziom 0
    wire RxFallingEdge;                                          // 9

    EdgeDetector EdgeDetector_inst(                              // 10
        .Clock(Clock),
        .Reset(Reset),
        .Signal_i(RxSync),                                       // 11
        .RisingEdge_o(),
        .FallingEdge_o(RxFallingEdge)                            // 12
    );


    // Timing
    wire Strobe;                                                 // 13
    localparam TICKS_PER_HALF_BIT = CLOCK_HZ / (BAUD * 2);       // 14

    StrobeGeneratorTicks #(                                      // 15
        .TICKS(TICKS_PER_HALF_BIT)
    ) StrobeGeneratorTicks_inst(
        .Clock(Clock),
        .Reset(Reset),
        .Enable_i(Busy || !RxSync),                              // 16
        .Strobe_o(Strobe)                                        // 17
    );

    wire SampleEnable = Strobe && !Counter[0];                   // 18

    always @(posedge Clock, negedge Reset) begin
        if(!Reset) begin                                         // 19
            Busy     <= 0;
            Counter  <= 0;
            Data_o   <= 0;
            Done_o   <= 0;
            RxBuffer <= 0;
        end else begin

            // Stan bezczynności
            if(!Busy) begin                                      // 20
                if(RxFallingEdge) begin                          // 21
                    Counter <= 5'd0;
                    Busy    <= 1'b1;
                end

                Done_o  <= 1'b0;                                 // 22
            end

            // Transmisja w trakcie
            else begin
                if(SampleEnable) begin                           // 23
                    RxBuffer <= {RxSync, RxBuffer[8:1]};         // 24
                end

                if(Counter == 5'd17) begin                       // 25
                    Data_o <= RxBuffer[8:1];
                    Done_o <= 1'b1;
                    Busy   <= 1'b0;
                end

                if(Strobe) begin                                 // 26
                    Counter <= Counter + 1'b1;
                end
            end
        end
    end

endmodule
`default_nettype wire
```

Listing 1. Kod pliku uart_rx.v

```
// Plik uart_rx_tb.v

`timescale 1ns/1ns
`default_nettype none
module UartRx_tb();

    parameter CLOCK_HZ            = 1_000_000;
    parameter real HALF_PERIOD_NS = 1_000_000_000.0 / (2 * CLOCK_HZ);
```

Listing 2. Kod pliku uart_rx_tb.v

Następnie utworzymy moduł wykrywający zbocze opadające na zsynchronizowanym sygnale **RxSync**. Aby to uczynić, w linii 10 tworzymy instancję modułu **EdgeDetector**, który również znamy z poprzednich odcinków kursu. Jego wejście łączymy z **RxSync** (linia 11), a wyjście zostanie wyprowadzone (linia 12) za pomocą zmiennej **RxFallingEdge** typu wire, utworzonej w linii 9. Po każdym wystąpieniu zbocza opadającego na linii Rx, na zmiennej **RxFallingEdge** pojawi się krótka szpilka stanu wysokiego o długości jednego cyklu zegarowego.

Kolejnym modułem będzie generator impulsów okresowych – niezbędny, by móc odczytywać stan wejścia linii transmisyjnej w ściśle określonych odstępach czasu. Użyjemy tutaj modułu **StrobeGeneratorTicks**, którego instancję tworzymy w linii 15. Moduł ten konfigurujemy za pomocą parametru **TICKS_PER_HALF_BIT** (linia 14), określającego liczbę taktów zegarowych upływających pomiędzy połówkami każdego bitu ramki transmisyjnej. Wyjście modułu (linia 17) stosuje zmienną **Strobe** typu wire, utworzoną w linii 13.

Przyjrzyjmy się bliżej wejściu **Enable_i**. Moduł pracuje tylko wtedy, kiedy na tym wejściu mamy stan wysoki. Łączymy je z dwoma sygnałami, połączonymi ze sobą bramką OR (linia 16). Pierwszy ze wspomnianych sygnałów to **Busy**, ponieważ moduł **StrobeGeneratorTicks** ma pracować, kiedy trwa odbieranie. Drugi to zanegowany sygnał **RxSync – linia Busy** ustawiana jest bowiem w stan wysoki dopiero po wykryciu zbocza opadającego na wejściu Rx. Dzięki uruchomieniu modułu **StrobeGeneratorTicks,** natychmiast po wystąpieniu stanu niskiego na Rx, pierwszy impuls będzie bliżej środka bitu startu (w przeciwnym razie byłby nieco przesunięty).

Jak już wcześniej pisałem, licznik **Counter** będziemy inkrementować co połowę czasu trwania transmitowanego bitu, ale próbki odczytywać musimy co cały bit. Aby ułatwić sobie to zadanie, w linii 18 tworzymy zmienną **SampleEnable** typu wire, przyjmującą stan wysoki na czas trwania jednego taktu zegarowego, co oznaczać będzie, że w kolejnym takcie ma zostać odczytane wejście odbiornika. W tym celu za pomocą bramki AND łączymy ze sobą dwa sygnały. Pierwszy to **Strobe**, pochodzący z wyjścia **StrobeGeneratorTicks**, który ustawiany jest w stan wysoki na jeden takt zegarowy co pół bitu. Drugi **–** to zanegowany zerowy bit licznika **Counter**. Korzystamy tutaj z faktu, że najmłodszy bit licznika po każdej inkrementacji zmienia swój stan na przeciwny. W taki sposób **SampleEnable** będzie przyjmować stan wysoki, gdy jednocześnie w stanie wysokim jest **Strobe** i kiedy licznik **Counter** ma wartość parzystą.

Przejdźmy dalej, do jedynego bloku always w module odbiornika, będącego blokiem sekwencyjnym reagującym na zbocze rosnące zegara oraz zbocze opadające sygnału resetującego. Jak zawsze, w pierwszej kolejności sprawdzamy stan **Reset –** jeżeli jest on niski, to zerujemy wszystkie zmienne typu reg (linia 19).

Logika bloku always dzieli się na dwie części, w zależności od tego, czy moduł znajduje się w stanie bezczynności, tzn. oczekuje na zbocze opadające bitu startu, czy transmisja jest w trakcie. Odróżniamy te dwa stany, sprawdzając zmienną **Busy** (linia 20).

W stanie bezczynności badamy sygnał **RxFallingEdge**, sterowany przez detektor zbocza opadającego (linia 21). Po jego wykryciu zerujemy licznik **Counter** i ustawiamy **Busy** w stan wysoki.

Niezależnie od wszystkiego, w tym samym czasie zerujemy zmienną **Done_o** (linia 22). Zmienna ta jest ustawiana w stan wysoki po zakończeniu odbierania bajtu, kiedy **Busy** zmienia się z 1 na 0, więc musimy ją wyzerować.

Następnie, podczas odbierania bajtów, wykonywane są trzy instrukcje **if**. Pierwsza z nich sprawdza (linia 23), czy w bieżącym takcie zegarowym należy odczytać wejście odbiornika. W takiej sytuacji do 9-bitowego rejestru **RxBuffer** wpisujemy wartość powstałą ze sklejenia zmiennej **RxSync** i dotychczasowych ośmiu bitów **RxBuffer** (linia 24). Inaczej mówiąc, rejestr ten przesuwamy w prawo o jeden bit, a w miejsce najstarszego bitu wstawiamy wartość **RxSync**, czyli najnowszy odczytany bit z wejścia odbiornika.

```
    // Generator sygnału zegarowego
    reg Clock = 1'b1;
    always begin
        #HALF_PERIOD_NS;
        Clock = !Clock;
    end

    // Zmienne
    reg  Reset = 1'b0;

    reg  [7:0] TxData;
    wire       TxDone;
    reg        TxRequest = 1'b0;

    wire [7:0] RxData;
    wire       RxDone;

    wire       TxRxCommon;

    // Nadajnik UART
    UartTx #(
        .CLOCK_HZ(CLOCK_HZ),
        .BAUD(100_000)
    ) UartTx_Inst(
        .Clock(Clock),
        .Reset(Reset),
        .Start_i(TxRequest),                                    // 1
        .Data_i(TxData),                                        // 2
        .Busy_o(),
        .Done_o(TxDone),                                        // 3
        .Tx_o(TxRxCommon)                                       // 4
    );

    // Odbiornik UART
    UartRx #(
        .CLOCK_HZ(CLOCK_HZ),
        .BAUD(100_000)
    ) UartRx_Inst(
        .Clock(Clock),
        .Reset(Reset),
        .Rx_i(TxRxCommon),                                      // 5
        .Done_o(RxDone),                                        // 6
        .Data_o(RxData)                                         // 7
    );

    // Eksport wyników symulacji
    initial begin
        $dumpfile("uart_rx.vcd");
        $dumpvars(0, UartRx_tb);
    end

    // Sekwencja testowa
    initial begin
        $timeformat(-6, 3, "us", 12);
        $display("===== START =====");
        $display("Ticks per half bit = %0d",
            UartRx_Inst.TICKS_PER_HALF_BIT);

        @(posedge Clock);
        Reset     <= 1'b1;
        repeat(99) @(posedge Clock);                            // 8

        // Wysyłanie pierwszego bajtu
        TxData    <= 8'hAB;                                     // 9
        TxRequest <= 1'b1;                                      // 10
        @(posedge Clock);                                       // 11
        TxData    <= 8'bxxxxxxxx;                               // 12
        TxRequest <= 1'b0;                                      // 13

        // Wysyłanie drugiego bajtu
        @(posedge TxDone);                                      // 14
        TxData    <= 8'hCD;
        TxRequest <= 1'b1;
        @(posedge Clock);
        TxData    <= 8'bxxxxxxxx;
        TxRequest <= 1'b0;

        // Czekaj na zakończenie transmisji
        @(posedge TxDone);                                      // 15
        repeat(100) @(posedge Clock);                           // 16

        $display("====== END ======");
        $finish;
    end

    // Wyświetlanie wysyłanego bajtu na początku transmisji
    always begin
        @(posedge TxRequest)                                    // 17
        $display("%t Transmitting byte: %H %s",
            $realtime,
            UartTx_Inst.Data_i,
            UartTx_Inst.Data_i
        );
    end

    // Wyświetlanie odebranego bajtu po zakończeniu transmisji
    always begin
        @(posedge RxDone)                                       // 18
        $display("%t Received byte:     %H %s",
            $realtime,
            UartRx_Inst.Data_o,
            UartRx_Inst.Data_o
        );
    end

endmodule
`default_nettype wire
```

Listing 2. Kod pliku uart_rx_tb.v – cd.

```
@echo off
iverilog -o uart_rx.o uart_rx.v uart_rx_tb.v uart_tx.v strobe_generator_ticks.v edge_detector.v synchronizer.v
vvp uart_rx.o
del uart_rx.o
```

Listing 3. Kod skryptu uart_rx.bat

Następnie sprawdzamy, czy aktualna wartość licznika **Counter** wynosi 17 (linia 25). Jeżeli tak – to znaczy, że odebraliśmy już wszystkie bity z ramki danych. Kopiujemy je zatem do wyjścia **Data_o**, ustawiamy **Done_o** w stan wysoki oraz zerujemy **Busy**.

W ostatnim warunku sprawdzamy, czy sygnał Strobe pochodzący z wyjścia modułu **StrobeGeneratorTicks** jest w stanie wysokim. Jeżeli tak, to inkrementujemy licznik.

Zwróć uwagę, że nie ma tu żadnego **else** – zatem wszystkie trzy instrukcje warunkowe wykonują się jednocześnie w każdym takcie zegarowym. Ponadto moglibyśmy opisane bloki warunkowe pozamieniać miejscami i nie miałoby to żadnego wpływu na syntezowaną logikę.

Testbench modułu UartRx

Zanim wgramy nasz kod do FPGA, przetestujemy go w symulatorze Icarus Verilog. Kod testbencha pokazano na **listingu 2**.

Podczas symulacji zastosujemy moduł nadajnika **UartTx**, który opracowaliśmy w poprzednim odcinku kursu. Moduł ten wyemituje dwa bajty 0xAB i 0xCD, a moduł odbiornika będzie miał odebrać je i wyświetlić na konsoli symulatora. Prędkość transmisji w nadajniku i odbiorniku zostanie ustawiona na 100000 bitów na sekundę. Dzięki temu każda ramka transmisyjna będzie trwała 100 µs. Zegar symulacji ustawimy na 1 MHz, aby można było łatwo zaobserwować sygnały ustawiane w stan wysoki na jeden takt sygnału zegarowego. Transmisja całej ramki potrwa 100 taktów zegarowych.

Zacznijmy od omówienia zmiennych używanych podczas symulacji:

- **TxData[7:0] –** bajt danych, który ma zostać wysłany, doprowadzony jest do wejścia **Data_i** nadajnika (linia 2).
- **TxRequest –** ustawienie tej zmiennej w stan wysoki na jeden takt zegarowy spowoduje rozpoczęcie wysyłania danych; połączona jest ona z wejściem **Start_i** nadajnika (linia 1).
- **TxDone –** sygnał sterowany przez nadajnik z wyjścia **Done_o** (linia 3), informujący o tym, że nadawanie zostało zakończone.
- **RxData[7:0] –** bajt danych odebrany przez odbiornik i dostępny na wyjściu **Data_o** odbiornika (linia 7).
- **RxDone –** sygnał sterowany przez odbiornik z wyjścia **Done_o** (linia 6), informujący o tym, że odbieranie zostało zakończone.
- **TxRxCommon –** sygnał łączący wyjście **Tx_o** nadajnika (linia 4) z wejściem **Rx_i** odbiornika (linia 5).

Następnie widzimy instancje modułów nadajnika **UartTx** i odbiornika **UartRx**. Nie znalazło się tu nic zaskakującego, zatem nie wymagają one komentarza.

```
VCD info: dumpfile uart_rx.vcd opened for output.
===== START =====
Ticks per half bit = 5
   100.000us Transmitting byte: ab «
   190.000us Received byte:     ab «
   200.000us Transmitting byte: cd Í
   290.000us Received byte:     cd Í
====== END ======
uart_rx_tb.v:93: $finish called at 400000 (1ns)
```

Listing 4. Wynik symulacji na konsoli

Przejdźmy do sekwencji testowej. Rozpoczynamy ją od ustawienia zmiennej **Reset** w stan wysoki po wystąpieniu zbocza rosnącego sygnału zegarowego, po czym czekamy kolejnych 99 taktów zegarowych (czyli razem 100 taktów (linia 8)), co trwać będzie dokładnie 100 µs. Działanie takie ma na celu utworzenie odstępu między krawędzią ekranu symulatora a rozpoczęciem transmisji pierwszego bajtu, a każdy bajt również potrwa 100 µs.

Aby rozpocząć transmitowanie bajtu, wartość żądaną do przesłania wpisujemy do zmiennej **TxData** (linia 9), a **TxRequest** ustawiamy w stan wysoki (linia 10). Czekamy na kolejne zbocze zegara (linia 11). W tym momencie nadajnik odczytuje dane z wejścia i rozpoczyna pracę. Do czasu zakończenia transmisji stan jego wejścia danych jest nieistotny – dlatego ustawiamy je w stan nieokreślony w linii 12, a zmienną **TxRequest** zerujemy (linia 13).

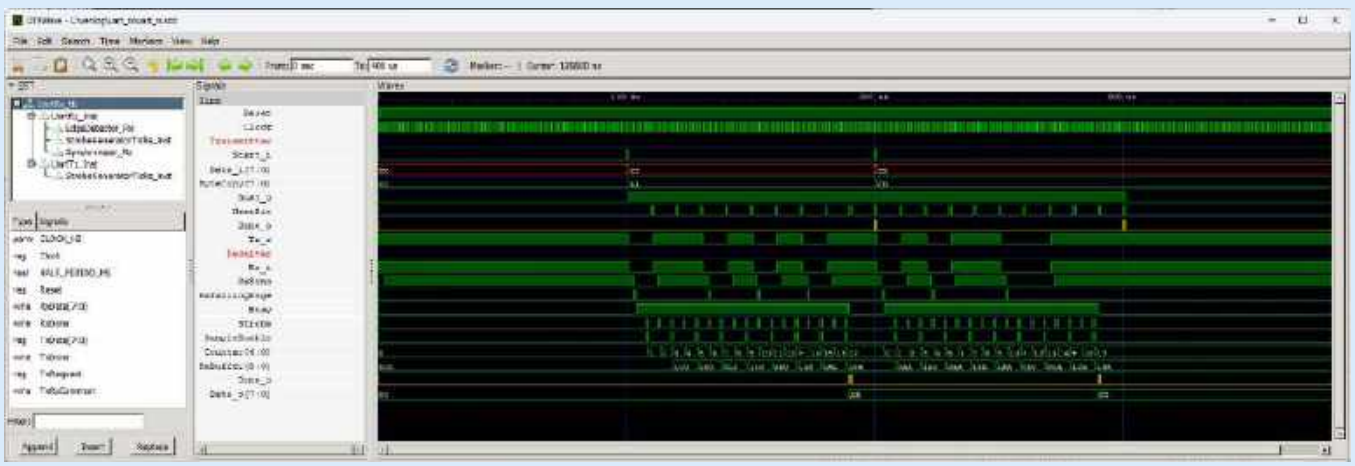
Transmisja jest w toku. Zawieszamy więc wykonywanie sekwencji testowej tak długo, aż zostanie ustawiony sygnał **TxDone** w stan wysoki (linia 14). Drugi testowy bajt wysyłany jest w dokładnie taki sam sposób, zatem nie wymaga komentarza.

Czekamy na zakończenie transmisji (linia 15) i odliczamy 100 taktów zegarowych (16).

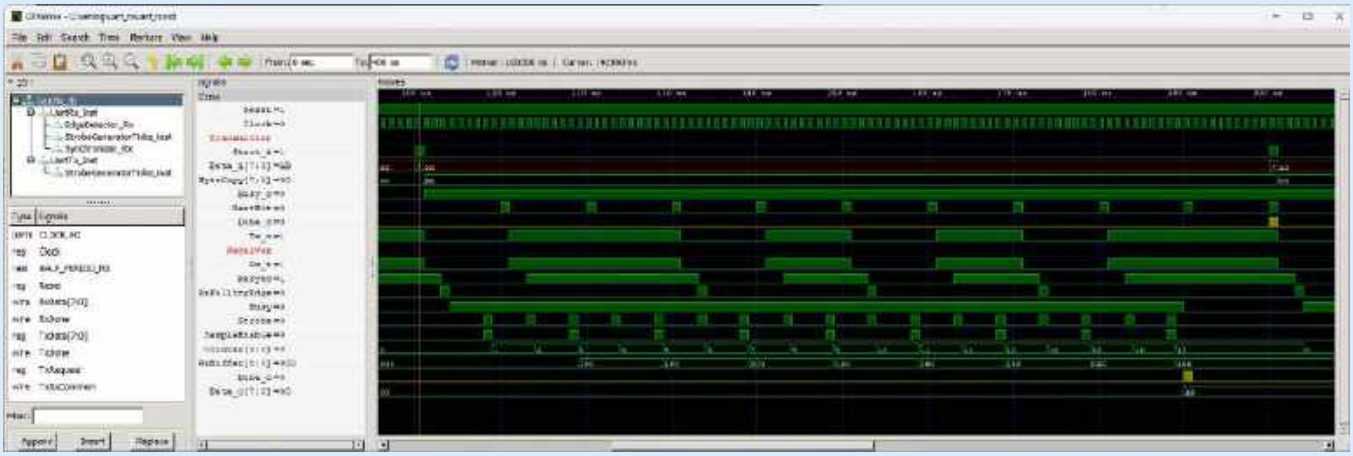
Dodatkowo widzimy dwa bloki always przeznaczone do wyświetlania komunikatów na konsoli symulatora. Oba wykonują się równolegle i jednocześnie z sekwencją testową. W linii 17 oczekujemy na zbocze rosnące sygnału **TxRequest** (co oznacza żądanie wysyłania bajtu), po czym wyświetlamy na konsoli symulatora informacje o aktualnym czasie oraz zawartości wysyłanego bajtu w postaci szesnastkowej i ASCII.

W linii 18 mamy do czynienia z bardzo podobną logiką, która różni się tylko tym, że wyświetla odebrane bajty, o czym świadczy stan wysoki na **RxDone**.

Aby przeprowadzić symulację, uruchamiamy skrypt uart_rx.bat, którego kod zaprezentowano na **listingu 3**.



Rysunek 4. Przebiegi uzyskane podczas symulacji



Rysunek 5. Zbliżenie na pierwszy transmitowany bajt

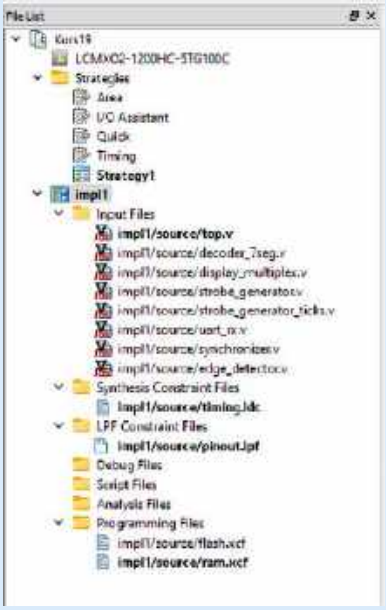
Po wykonaniu symulacji powinniśmy zobaczyć komunikaty widoczne na **listingu 4**.

Otwórzmy plik **uart_rx.vcd** za pomocą przeglądarki GTKWave i skonfigurujmy ją tak, by uzyskać rezultat widoczny na **rysunku 4**. Sygnały dotyczące nadajnika i odbiornika są celowo oznaczone czerwonymi etykietami Transmitter i Receiver.

Przybliżmy widok, aby lepiej widzieć transmisję pierwszego bajtu (zobacz **rysunek 5).**

Wszystko zaczyna się od ustawienia wartości 0xAB na wejściu **Data_i** nadajnika i ustawienia jego wejścia **Start_i** w stan wysoki (na rysunku 5 zostało to zaznaczone pomarańczowym kursorem pionowym). W następnym cyklu zegarowym bajt do nadania skopiowany zostaje do **ByteCopy**. Wciąż trwa wysyłanie poszczególnych bitów (co widzimy na **Tx_o)**, a każdy z nich trwa 10 µs.

Te same bity trafiają na wejście **Rx_i** odbiornika. Sygnał ten przechodzi przez synchronizator, który wprowadza opóźnienie o dwa takty zegarowe – przypatrz się, w jaki sposób przesunięty zostaje sygnał **RxSync** względem **Rx_i**. Następnie wykrywane jest zbocze opadające bitu startu, co widać jako krótką szpilkę stanu wysokiego na **RxFallingEdge**. Po tym w stan wysoki ustawiana jest zmienna **Busy** i na **Strobe** pojawiają się szpileczki co pół bitu. Nam potrzebna jest co druga z tych szpilek **– efekt taki** uzyskujemy na sygnale **SampleEnable**. Kiedy znajduje się on w stanie wysokim, inkrementujemy licznik **Counter** i dokładamy odebrany bit do **RxBuffer**. Gdy licznik **Counter** doliczy do 17, wówczas kopiujemy interesujące nas bity z RxBuffer na wyjście **Data_o**, a wyjście **Done_o** ustawiamy w stan wysoki na jeden takt zegarowy, by poinformować, że odebrane dane są gotowe do odczytania.



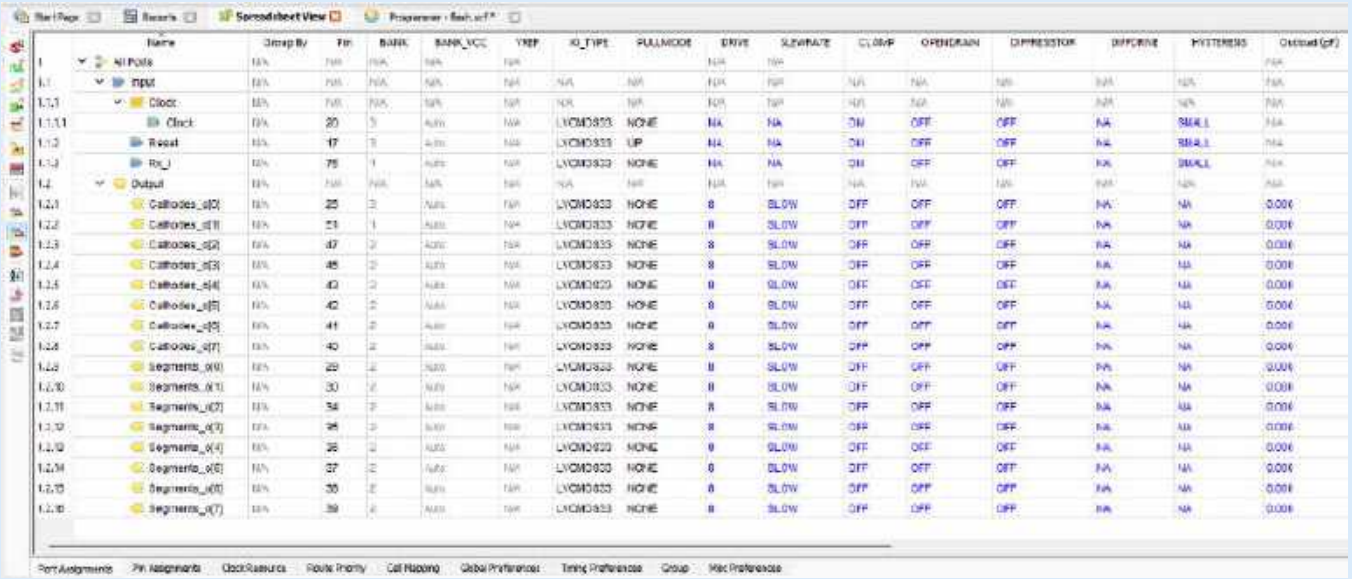
Rysunek 6. Lista plików projektu

Moduł top

Czas przetestować nasz moduł odbiornika w rzeczywistości, na prawdziwym FPGA. Stosując przejściówkę USB/UART oraz dowolny terminal, jak np. Putty czy Realterm, będziemy przesyłać

```
// Plik top.v

`default_nettype none

module top(
    input  wire Clock,              // Pin 20 (Zegar 25MHz)
    input  wire Reset,              // Pin 17 (Przycisk K0)
    input  wire Rx_i,               // Pin 75 (Oznaczenie Rx na złączu)

    output wire [7:0] Cathodes_o,
    output wire [7:0] Segments_o
);

    parameter CLOCK_HZ = 25_000_000;                                    // 1

    // Odbiornik UART
    wire RxDone;                                                        // 2
    wire [7:0] RxData;                                                  // 3

    UartRx #(                                                           // 4
        .CLOCK_HZ(CLOCK_HZ),
        .BAUD(115200)                                                   // 5
    ) UartRx_Inst(
        .Clock(Clock),
        .Reset(Reset),
        .Rx_i(Rx_i),
        .Done_o(RxDone),                                                // 6
        .Data_o(RxData)                                                 // 7
    );

    // Wyświetlanie czterech ostatnio odebranych bajtów
    reg [31:0] DataToDisplay;                                           // 8

    always @(posedge Clock, negedge Reset) begin
        if(!Reset)
            DataToDisplay <= 0;
        else if(RxDone)
            DataToDisplay[31:0] <= {DataToDisplay[23:0], RxData};       // 9
    end

    // Instancja modułu 8-cyfrowego wyświetlacza LED
    DisplayMultiplex #(                                                 // 10
        .CLOCK_HZ(CLOCK_HZ),
        .SWITCH_PERIOD_US(1000),
        .DIGITS(8)
    ) DisplayMultiplex0(
        .Clock(Clock),
        .Reset(Reset),
        .Data_i(DataToDisplay),                                         // 11
        .DecimalPoints_i(8'b01010101),                                  // 12
        .Cathodes_o(Cathodes_o),
        .Segments_o(Segments_o),
        .SwitchCathode_o()
    );

endmodule

`default_nettype wire
```

Listing 5. Kod pliku top.v



Rysunek 7. Konfiguracja pinów

z komputera do FPGA różne bajty danych. Po odebraniu każdego bajtu zostanie on wyświetlony na wyświetlaczu 7-segmentowym w formacie szesnastkowym. Na płytce User Interface Board znajduje się wyświetlacz ośmiocyfrowy, zatem możliwe będzie wyświetlenie czterech ostatnio odebranych bajtów. Każdy bajt może przyjmować wartości od 0x00 do 0xFF. Podobne rozwiązanie zastosowaliśmy w 10 odcinku kursu na temat klawiatury matrycowej, w którym na wyświetlaczu pojawiały się kody czterech ostatnio wciśniętych przycisków.

Utwórz nowy projekt w Lattice Diamond, a następnie dodaj do niego pliki z kodem źródłowym pokazane na **rysunku 6**. Wszystkie moduły, oczywiście z wyjątkiem modułu top, były omawiane już wcześniej i możesz pobrać je z adresów widocznych w ramce na początku artykułu.

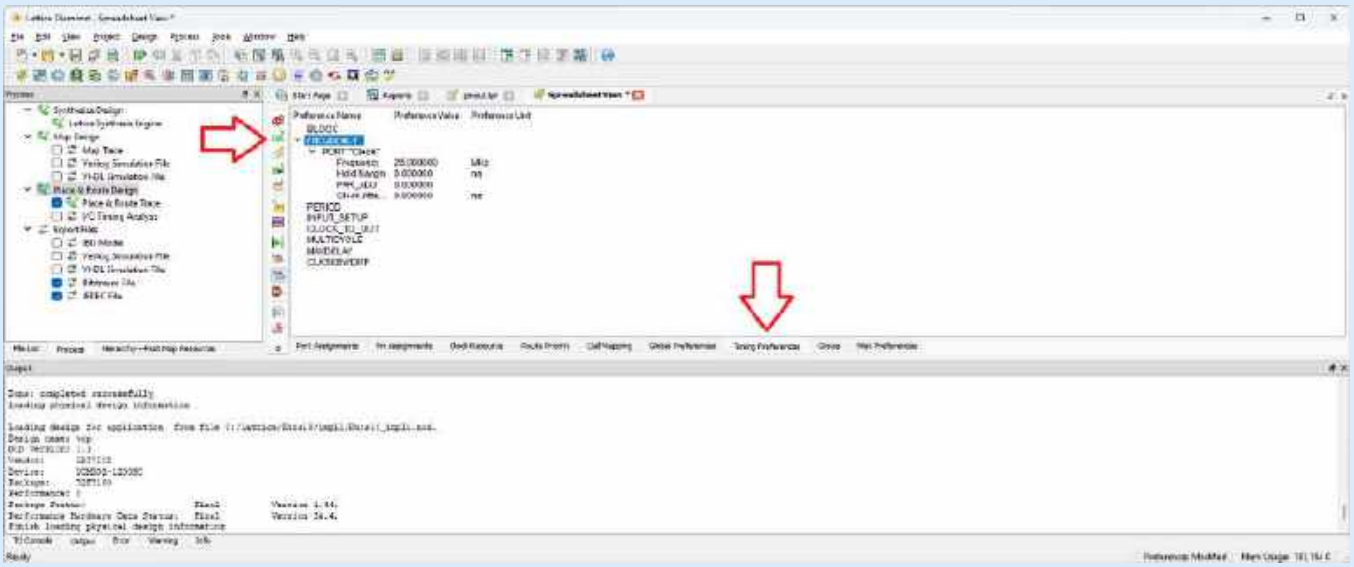
Przejdźmy teraz do omówienia modułu **top**, którego kod pokazano na **listingu 5**. Podobnie jak w poprzednim odcinku kursu na temat nadajnika UART, będziemy korzystać z zewnętrznego, kwarcowego generatora sygnału zegarowego o częstotliwości 25 MHz. Generator RC wbudowany w FPGA ma niewystarczającą dokładność, przez co transmisja UART z jego użyciem może przebiegać nieprawidłowo. Na liście portów modułu **top** dodajemy zatem wejście **Clock**. Pamiętajmy też, by odpowiednio ustawić częstotliwość zegara parametrem **CLOCK_HZ** (linia 1).

Zaczniemy od utworzenia dwóch zmiennych typu wire. Zmienna **RxDone**, zadeklarowana w linii 2, informować nas będzie o tym, że został odebrany bajt danych. Moduł odbiornika ustawi ją wtedy w stan wysoki na jeden takt sygnału zegarowego. Ostatnio odebrany bajt będzie można odczytać, korzystając z 8-bitowej zmiennej **RxData** (linia 3).

W linii 4 tworzymy instancję odbiornika UART, który ma odbierać dane z szybkością 115200 bitów na sekundę, co określamy parametrem **BAUD** w linii 5. W liniach 6 i 7 łączymy wyjścia modułu z sygnałami wire, utworzonymi wcześniej.

Następnie tworzymy 32-bitową zmienną **DataToDisplay** typu reg (linia 8). Celem tej zmiennej jest przechowywanie czterech ostatnio odebranych bajtów, aby pokazać je na wyświetlaczu. Przypomnijmy, że moduł wyświetlacza 8-cyfrowego wyświetla znaki od 0 do 9 oraz od A do F. Na każdy znak przypadają cztery bity, zatem na dwa znaki przypada osiem bitów, czyli tyle, ile ma w sobie jeden bajt.

W dalszej części listingu mamy prosty blok always, którego jedynym celem pozostaje aktualizacja zmiennej **DataToDisplay** po odebraniu nowego bajtu. W linii 9 zapisujemy do tej 32-bitowej zmiennej nową wartość, powstającą w wyniku sklejenia dotychczasowych bitów od 23 do 0 tej zmiennej (czyli 24 najmłodszych bitów) i doklejenia do nich zawartości 8-bitowej zmiennej **RxData**. Inaczej mówiąc,



Rysunek 8. Konfiguracja sygnału zegarowego

dotychczasowa zawartość **DataToDisplay** jest przesuwana w lewo o osiem bitów, a w miejsce pustych bitów wstawiana jest zawartość **RxData**.

Pozostaje już tylko utworzyć instancję modułu sterującego wyświetlaczem, co czynimy w linii 10. Moduł ten był omawiany dokładnie w 9 odcinku kursu. Do wejścia danych **Data_i** doprowadzamy zmienną **DataToDisplay** (linia 11). Aby ułatwić odróżnienie poszczególnych bajtów, zajmujących dwa znaki na wyświetlaczu, włączymy punkty dziesiętne pomiędzy nimi (linia 12).

Przeprowadź syntezę, a następnie otwórz narzędzie Spreadsheet i skonfiguruj piny układu FPGA w taki sposób, jak zaprezentowano na **rysunku 7**.

Ponieważ używamy zewnętrznego źródła sygnału zegarowego, doprowadzonego do jednego z pinów układu FPGA, musimy skonfigurować jego częstotliwość. W tym celu w Spreadsheet klikamy zakładkę **Timing Preferences**, a następnie wybieramy drugi od góry przycisk na pionowym pasku narzędzi. Dodajemy właściwość **FREQUENCY** dla pinu **Clock** i ustawiamy na 25 MHz (zobacz **rysunek 8**).

Generujemy bitstream i wgrywamy do FPGA. Pin Tx przejściówki USB/UART należy podłączyć do wejścia Rx na złączu goldpin w płytce User Interface Board. Kiedy z terminalu wyślemy jakikolwiek bajt, zostanie on wyświetlony na dwóch wyświetlaczach 7-segmentowych po prawej stronie. Jeżeli wcześniej zostały odebrane jakiekolwiek bajty, przesuną się one w lewo.

W następnym odcinku użyjemy ponownie modułu odbiornika UART, lecz odebrane bajty będziemy wyświetlać jako normalne, czytelne litery na 14-segmentowym wyświetlaczu LCD. Będzie to rozwinięcie tematu sterowników wyświetlaczy LCD, który zaczęliśmy w 13 odcinku kursu.

Dominik Bieczyński
leonow32@gmail.com

koktajl niusów

Inteligentny tłumacz językowy Vasco V4 znacząco ułatwia podróże

O dalekich podróżach marzy wielu z nas. Jednak część osób rezygnuje z tego rodzaju atrakcji z powodu nieznajomości języków obcych. To z myślą o nich powstał inteligentny tłumacz Vasco V4, dzięki któremu zacierają się wszystkie dotychczasowe granice, a komunikacja ze światem staje się czystą przyjemnością. Wyposażeni w rozwiązanie Vasco V4 użytkownicy mogą mieć cały świat w zasięgu ręki. Niedużych rozmiarów urządzenie operuje aż na 108 językach: potrafi tłumaczyć nie tylko pismo, lecz również mowę albo teksty ze zdjęć. Na tle różnych przeglądarkowych translatorów tłumacz Vasco V4 wyróżnia fakt, że występuje w nim w sumie 10 złożonych silników tłumaczeniowych. Oznacza to, że każda para językowa tłumaczona jest za pomocą optymalnie dobranego silnika, co zapewnia znakomite efekty działania. Specjaliści z Vasco Electronics doskonale zdają sobie sprawę z tego, że z prezentowanego urządzenia będą korzystać również seniorzy: w związku z tym wyposażyli je w czytelny ekran o przekątnej 5" oraz bardzo wygodny w obsłudze interfejs, który nie sprawia problemów nawet osobom nieobytym z tego rodzaju gadżetami. Długą listę zalet Vasco V4 zamyka znacząca odporność sprzętu na pyły i wilgoć. Tylko z inteligentnym tłumaczem językowym Vasco V4 nie trzeba się martwić o przypadkowe wyzwanie kogoś na pojedynek podczas podróży w odległe rejony świata. Jest to oczywiście żart, jednakże telefoniczne translatory, które korzystają tylko z jednego silnika tłumaczeniowego, stwarzają zazwyczaj spore ryzyko problemów komunikacyjnych, nierzadko poważnych w skutkach.

https://tiny.pl/dw5vp

Wyprodukowano tysięczny egzemplarz bezzałogowego statku powietrznego FlyEye

Jest to najbardziej zaawansowany na świecie system w swojej klasie, co warto podkreślić – w całości skonstruowany i wytwarzany w Polsce. Jak dotąd FlyEye został sprawdzony bojowo w najtrudniejszych warunkach – podczas konfliktu o wysokiej intensywności. W połowie marca 2024 roku linie produkcyjne opuścił tysięczny bezzałogowiec FlyEye w wersji 3.x. Model ten może działać w najtrudniejszych warunkach w dzień i w nocy, nawet przy niemal całkowicie zakłóconej łączności lub nawigacji satelitarnej. Opisywane rozwiązanie charakteryzuje się najwyższym wskaźnikiem przeżywalności – czy też ukończonych z powodzeniem misji – spośród wszystkich bezzałogowców użytych w czasie działań bojowych od 2015 roku. Zasilany jest cichym silnikiem elektrycznym, dzięki czemu może posłużyć m.in. do obserwacji pola walki, kierowania artylerią, retransmisji, patrolowania granic bądź monitoringu infrastruktury krytycznej. Składane śmigła sprawiają, że charakteryzuje się on niezwykle niskim poziomem odbicia radarowego, a możliwość startu pionowego z ręki eliminuje potrzebę stosowania dodatkowych urządzeń wspomagających start i lądowanie. Prezentowany bezzałogowiec może być transportowany w zaledwie 2 plecakach i niestraszne jest mu lądowanie w trudnym terenie, nawet o ograniczonej przestrzeni. FlyEye przenosi moduł zadaniowy – w najbardziej popularnej odmianie wyposażony jest w głowicę optoelektroniczną. Łatwa w obsłudze kamera dzienna i termowizyjna umożliwia szybką detekcję różnych celów, a także przekazywanie danych na temat ich położenia. Wszystko to ma miejsce przy zastosowaniu opcjonalnego transpondera i modułu sztucznej inteligencji EYEQ Air, który gwarantuje analizę obrazów na pokładzie bezzałogowca. Rozwiązanie takie pozwala na wykonywanie misji w środowisku, na które oddziałują różne środki walki radioelektronicznej. FlyEye może być użyty jako powietrzny retranslator z radiostacją PERAD w celu współtworzenia systemu komunikacji SILENT NETWORK.

https://tiny.pl/dw5vz

Najnowsze karty pamięci microSD Express o pojemności 256 GB od firmy Samsung

Firma Samsung jako pierwsza w branży wprowadziła na rynek karty pamięci microSD, które korzystają z interfejsu SD Express. Ich powstanie było możliwe zwłaszcza dzięki współpracy z wieloma partnerami w celu stworzenia niestandardowego rozwiązania. Stworzone przez Samsunga karty pamięci microSD Express wyróżniają się m.in. niskim poborem energii i zaawansowaną technologią oprogramowania sprzętowego. Wspominany software zaprojektowany został z myślą o zwiększeniu wydajności, a także efektywnym zarządzaniu temperaturą kart w długiej perspektywie czasowej. Dzięki tym innowacjom wydajność kart opracowanych przez Samsunga porównywalna jest z osiągami niewielkich dysków SSD. W tradycyjnych kartach microSD o interfejsie UHS-1 szybkości odczytu ograniczone są zaledwie do 104 MB/s. W przypadku wariantu microSD Express udało się je podwyższyć aż do 985 MB/s. Opisywane karty oferują sekwencyjną szybkość odczytu do 800 MB/s – czyli ok. 1,4 razy szybciej niż w dyskach SSD (do 560 MB/s) i ponad 4 razy szybciej w porównaniu

z tradycyjnymi kartami pamięci UHS-1 (do 200 MB/s). Taki transfer gwarantuje o wiele płynniejszą pracę aplikacji na komputerach i urządzeniach mobilnych. W celu zapewnienia wysokiej wydajności, zwłaszcza w trakcie intensywnego użytkowania kart microSD Express, zastosowano technologię Dynamic Thermal Guard (DTG) utrzymującą optymalną ich temperaturę.

https://tiny.pl/dw5bx

Konferencja Worldwide Developers Conference (WWDC) firmy Apple powraca 10 czerwca 2024 r.

Cała konferencja będzie dostępna online za darmo, a 10 czerwca 2024 roku planowany jest specjalny event w Apple Park, obejmujący m.in. spotkanie z zespołem Apple, prezentację i liczne atrakcje dodatkowe. Wydarzenie potrwa 4 dni – do 14 czerwca 2024 roku, a w jego trakcie firma Apple uchyli rąbka tajemnicy i zaprezentuje najnowsze udoskonalenia, które pojawią się np. w systemach operacyjnych: iOS, iPadOS, macOS, watchOS, tvOS i visionOS. Podczas prezentacji otwierającej WWDC uczestnicy poznają różne nowości ze świata oprogramowania i technologii Apple, a samą konferencję będzie można obserwować za pomocą aplikacji, strony Apple Developer, jak również portalu YouTube. W programie tegorocznej WWDC przewidziano sesje wideo i wiele innych możliwości nawiązania kontaktu z projektantami oraz inżynierami Apple czy międzynarodową społecznością deweloperów.

Dokładne informacje o rejestracji na wydarzenie można odnaleźć na stronie Apple Developer, a także w aplikacji. Dodatkowe informacje o konferencji WWDC firma Apple udostępni przed jej rozpoczęciem.

https://tiny.pl/dw5bt

Nowość w rodzinie materiałów do Zortrax Powerful Trio – żywica BASF Ultracur3D RG 3280 z wypełnieniem ceramicznym

Od teraz urządzenia Zortrax Powerful Trio oferują możliwość wytwarzania elementów 3D o właściwościach zbliżonych do ceramiki. Staje się to możliwe dzięki zastosowaniu żywicy BASF Ultracur3D RG 3280. Jej skalibrowany profil do druku 3D z wysokością warstwy 0,05 mm można znaleźć w oprogramowaniu Z-SUITE. Stworzona przez BASF żywica daje sposobność druku detali, które po obróbce w Zortrax Cleaning Station, a także Zortrax Curing Station zyskują właściwości w pełni odpowiadające ceramice. Elementy te charakteryzuje m.in. wysoka sztywność (do 10 GPa) i doskonała izolacja elektryczna, co sprawia, że opisywany materiał można zastosować w celu druku np. ceramicznych ramion manipulacyjnych. Mimo wysokiej zawartości cząstek żywicę Ultracur3D RG 3280 cechuje niska lepkość, a jej sedymentacja w zbiorniku jest ograniczona i właściwie nieistotna. W konsekwencji materiał okazuje się prosty w użytkowaniu i pozwala na druk wysokiej jakości części 3D. Za sprawą interesujących własności fizycznych (przede wszystkim wartości HDT osiągających 132°C – przy ciśnieniu 1,82 MPa i 280°C – przy ciśnieniu 0,45 MPa), Ultracur3D RG 3280 otwiera szeroki wachlarz zastosowań. Wydrukowane z niej modele 3D mogą być wykorzystywane do pracy przy wysokich temperaturach i stosowane do produkcji takich elementów, jak osłony termiczne, elementy układów silnika, komponenty turbin czy formy wtryskowe. Detale tego rodzaju odznaczają się szczególną odpornością na działanie acetonu, olejów silnikowych, hydraulicznych i przekładniowych, płynów hamulcowych czy tłuszczów uniwersalnych. Docelowo wszystkie wydruki 3D są białe i mają charakterystyczny, „ceramiczny" wygląd, a do ich obróbki nie potrzeba specjalistycznych pieców.

https://tiny.pl/dw5b7

Specjalistyczna analiza szumu fazowego i pomiary oscylatorów VCO do częstotliwości 50 GHz za pomocą urządzenia R&S FSPN50 firmy Rohde & Schwarz

Urządzenie R&S FSPN50 umożliwia wygodne badanie oscylatorów VCO, których częstotliwość wyjściowa mieści się w zakresie od 1 MHz do 50 GHz. Tym sposobem jest w stanie obsługiwać rozwiązania operujące w następujących trzech pasmach mikrofalowych: Ka (26,5...40 GHz), Q (33...50 GHz) oraz V (do 50 GHz). Jest to oryginalne rozwiązanie przeznaczone nie tylko do testów produkcyjnych, lecz także do zastosowań badawczo-rozwojowych w zakresie oscylatorów VCO. Doskonała czułość urządzenia R&S FSPN50 na szumy fazowe wynika m.in. z zastosowania 2 oscylatorów lokalnych o adekwatnych parametrach. Za precyzyjne pomiary oscylatorów VCO odpowiadają 3 źródła prądu stałego, cechujące się niezwykle niskim poziomem szumów. Zarówno szum fazowy, jak i szum amplitudowy mierzone są oddzielnie. Polecenia SCPI, które przesyła się do urządzenia R&S FSPN50, mogą być rejestrowane automatycznie, a aktualne wartości parametru „cross-correlation sensitivity gain" wyświetlane są na ekranie w trybie ciągłym. Istnieje również możliwość zwiększania dokładności pomiarowej (albo jej zmniejszania) w celu optymalizacji szybkości działania R&S FSPN50. Typowe zastosowania opisywanego urządzenia obejmują komercyjne systemy komunikacji bezprzewodowej, wojskowe systemy satelitarne i radary bliskiego zasięgu. Jest to rozwiązanie szczególnie polecane osobom pracującym w obszarze infrastruktury sieci 5G, wyróżniające się doskonałym stosunkiem ceny do wydajności.

https://tiny.pl/dw5b9

Opracowany na Politechnice Gdańskiej (PG) polski odpowiednik GPT

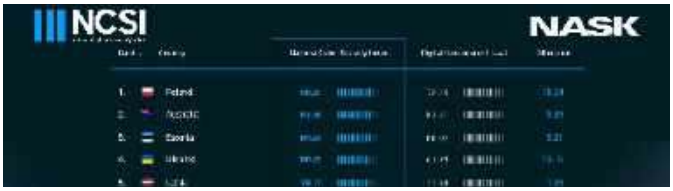
Mowa o polskojęzycznych, generatywnych, neuronowych modelach językowych Qra, które wytrenowano na bazie terabajta danych tekstowych, występujących wyłącznie w języku polskim. Zastosowany korpus liczył początkowo do 2 TB surowych danych tekstowych, które w dalszej kolejności uległy dwukrotnemu zmniejszeniu w wyniku procesów deduplikacji i oczyszczenia. Są to pierwsze modele wytrenowane na tak wielkim zasobie polskich tekstów – do ich uczenia użyto znaczących mocy obliczeniowych. Prezentowane rozwiązanie stanowi pierwszy odpowiednik GPT, który o wiele łatwiej rozumuje i przetwarza treści w języku polskim, lepiej rozumie pytania zadawane w tym języku, a także sprawniej generuje teksty po polsku. Zbudowano w sumie 3 modele Qra, które różnią się złożonością – są to: Qra 1B, Qra 7B i Qra 13B. Modele Qra 7B oraz Qra 13B uzyskują istotnie wysoki wynik perplexity, czyli zdolności do modelowania języka polskiego w zakresie jego rozumienia, warstwy leksykalnej i gramatyki. Testów pomiaru perplexity dokonano m.in. na zbiorach 10 tysięcy zdań ze zbioru testowego PolEval-2018 oraz 5 tysięcy długich, bardziej wymagających dokumentów stworzonych w 2024 roku. Zgodnie z planami modele językowe Qra będą stanowić podstawę rozwiązań informatycznych podczas obsługi spraw oraz procesów, które wymagają właściwego zrozumienia języka polskiego. Już teraz potrafią generować poprawne gramatycznie i stylistycznie odpowiedzi wyrażone w języku polskim. Generowane treści są fenomenalnie wysokiej jakości, co potwierdza m.in. właśnie wspomniana miara perplexity. Wkrótce rozpoczną się prace nad strojeniem modeli w celu sprawdzenia ich możliwości pod kątem zastosowań takich jak: klasyfikacja tekstów, dokonywanie ich streszczeń i odpowiadanie na pytania.

https://tiny.pl/dw5b5

Unikatowa opcja transkrypcji w aplikacji Podcasty Apple

Dzięki opcji transkrypcji użytkownicy aplikacji Podcasty Apple mogą m.in. przeczytać pełny zapis odcinka i wyszukać odcinek na podstawie słowa lub frazy. Co więcej, już jednym stuknięciem w określony fragment tekstu można odtworzyć podcast od wybranego momentu. W czasie odtwarzania odcinka każde słowo zostaje podkreślone, więc o wiele łatwiej jest śledzić aktualnie oglądany fragment. Udostępniona w aplikacji Podcasty Apple opcja transkrypcji może istotnie poprawić komfort korzystania z tych materiałów – i to na wiele sposobów. Przede wszystkim pomagają one wychwycić na bieżąco każde słowo prowadzących, nauczyć się nowego języka, a także sprawnie szukać informacji zasłyszanych w trakcie audycji. Użytkownicy mogą przechodzić do zapisu odcinka w lewym dolnym rogu ekranu Odtwarzane, a zapisy odcinków poprawiają dostępność treści. Kontrast między czcionką a tłem dobrano tak, żeby czytanie oraz przeglądanie długich tekstów było prostsze. Z kolei niesłyszący oraz niedosłyszący użytkownicy nie muszą odtwarzać odcinków, aby uzyskać dostęp do ich transkrypcji. Z dostępnej w aplikacji Podcasty Apple opcji transkrypcji można skorzystać w przypadku materiałów dostępnych w 4 językach: angielskim, francuskim, hiszpańskim i niemieckim – na iPhonie i iPadzie z systemami: iOS 17.4 i iPadOS 17.4. Wszystkie transkrypcje będą udostępniane automatycznie w odniesieniu do nowych odcinków po ich publikacji, natomiast zapisy wcześniejszych odcinków zostaną udostępnione stopniowo.

https://tiny.pl/dw5b4



Polska na pierwszym miejscu w rankingu cyberbezpieczeństwa NCSI

Nasz kraj zajął pierwsze miejsce w corocznym rankingu cyberbezpieczeństwa NCSI (z ang. National Cyber Security Index). To światowy indeks, który mierzy gotowość aż 32 krajów do zapobiegania zagrożeniom oraz zarządzania incydentami cybernetycznymi. Narodowy Indeks Bezpieczeństwa Cybernetycznego (NCSI) stworzony został przez estońską akademię e-Governance, aby umożliwić identyfikację kluczowych obszarów wymagających zainteresowania. Opisywany ranking zapewnia przegląd gotowości poszczególnych krajów do zapobiegania cyberatakom i przestępstwom, a także ich zwalczania. Opiera się na transparentnej metodologii. Na stronie internetowej NCSI znajdują się profile każdego z państw objętych monitoringiem, zawierające szczegółowe opisy wszystkich wskaźników i dowodów, na których opiera się wynik.

https://tiny.pl/dw5bn

Światowa premiera przenośnego głośnika LG StanbyME

Przenośny głośnik LG StanbyME to łatwy sposób na uatrakcyjnienie każdej rozrywki zarówno w domu, jak i na zewnątrz. Oferuje on nadzwyczajną jakość dźwięku i wszechstronność funkcjonalną, która uwzględnia różne style życia i preferencje użytkowników. Jest świetny np. dla osób ceniących design, które lubią słuchać muzyki, gdzie tylko tego zapragną. Zaprojektowany przez LG Electronics, przenośny głośnik bezproblemowo łączy się z ekranem StanbyME, umożliwiając sterowanie za pomocą pilota. Pozwala to użytkownikom na proste włączanie i wyłączanie głośnika oraz ekranu, a także na wygodne łączenie obydwu urządzeń poprzez interfejs Bluetooth. Żeby zapewnić jeszcze większą przyjemność z oglądania i słuchania, wbudowana w głośnik funkcja WOW Orchestra łączy się sprawnie z systemem audio w ekranie StanbyME. Znajdujący się w głośniku procesor α7 Gen 6 AI optymalizuje dźwięk odpowiednio do rodzaju treści, a podwójne głośniki wysokotonowe – o średnicy ok. 2 cm – gwarantują wyraźny dźwięk stereo w górnej części pasma. Podwójny, pasywny radiator generuje z kolei głębsze i atrakcyjniejsze basy. Centralnie umieszczony panel użytkownika pozwala na wygodne sterowanie funkcjami urządzenia, wskazuje także status urządzenia. Przenośny głośnik LG StanbyME może uzupełniać ekran StanbyME lub też być używany oddzielnie; taka elastyczność spełnia wszystkie potrzeby użytkowników związane z rozrywką audio. Dzięki specjalnemu uchwytowi opisywany głośnik można łatwo mocować, a dodatkową zaletę urządzenia stanowi gwarantowany stopień ochrony IPX5. Pojemny akumulator pozwala na odtwarzanie audio przez ok. 16 godzin. Przenośny głośnik LG StanbyME jest lekkim i kompaktowym rozwiązaniem o wymiarach: 7,8×32,6×8,7 cm oraz wadze ok. 0,9 kg. Funkcja podświetlenia krawędzi tworzy przyjemne dla oka efekty świetlne, które w subtelny sposób podkreślają nowoczesny design urządzenia.

https://tiny.pl/dw5bb

Jakub Tyburski
jakub.tyburski@elportal.pl

Hity następnego numeru • „Elektronika Praktyczna” 6/2024

Sterownik silnika krokowego z wyłącznikami krańcowymi

Silniki krokowe mają szereg zalet – ich prędkość obrotową można regulować w szerokim zakresie, w którym przez cały czas utrzymują stały moment obrotowy oraz trzymający. Jednak praktyczna aplikacja tych napędów nie jest tak prosta, jak w przypadku silników prądu stałego – wymagają one bowiem ciągłego monitorowania prądów uzwojeń, jak również cyklicznej zmiany ich kierunków. Na szczęście mamy na rynku gotowe (i niedrogie!) moduły, które potrafią zrealizować za nas większość tej złożonej procedury.

Problem w tym, że taki moduł najczęściej również wymaga odpowiedniego wysterowania, a to z kolei wymusza użycie mikrokontrolera… Zaprezentowane w tym artykule urządzenie to właśnie uniwersalny kontroler, który pozwoli na wykonywanie prostych ruchów przez dowolny bipolarny silnik krokowy. Ruch w jedną lub dwie strony, naprzemienny ruch od początku do końca zakresu, a nawet sterowanie ręczne – to wszystko potrafi prezentowany przez nas układ.

matrixClock

Zegary cyfrowe, podobnie jak termometry, termostaty czy miniaturowe radyjka, to elementarz każdego elektronika amatora. Któż z nas nie ma w swoim portfolio urządzeń tego typu, które mimo oczywistej prostoty dają dużo radości płynącej z własnoręcznego wykonania działającej konstrukcji? Inspiracją do powstania tego projektu był zakupiony na chińskim portalu sprzedażowym prosty zegar biurkowy, wyposażony w bardzo efektowny graficzny wyświetlacz VFD. Bardzo efektowny, ale... niewielki, gdyż konstrukcja dużych wyświetlaczy tego typu (zwłaszcza graficznych) jest zwykle bardzo kosztowna i w zasadzie odchodzi do lamusa. Zaproponowane urządzenie ma wprawdzie zbliżony zakres funkcjonalności, lecz wyposażone zostało w znacznie większy wyświetlacz graficzny – do roli elementów interfejsu użytkownika zostały wybrane popularne i niedrogie wyświetlacze matrycowe o rozdzielczości 5×7 punktów.

Moduł portów szeregowych do Raspberry Pi Pico

Opisany moduł rozszerza funkcjonalność Raspberry Pi Pico o drivery portów szeregowych w standardzie RS232 i RS485, przydatnych w domowej automatyce i nie tylko. Projekt bazuje na scalonych konwerterach MAX3232 i ADM3061. Poszczególne tory komunikacyjne są połączone z dwoma niezależnymi blokami UART procesora RP2040 (UART0 i UART1), dzięki czemu obydwa tory mogą pracować równocześnie bez ryzyka wystąpienia konfliktu na liniach danych. W układzie przewidziano zastosowanie czterech diod sygnalizujących aktywność linii wejścia/wyjścia, osobno dla każdego kierunku transmisji oraz interfejsu (RS232, RS485).

Przekaźniki do automatyki

Przekaźniki należą do najważniejszych komponentów stosowanych we wszystkich gałęziach automatyki. Trudno wyobrazić sobie współczesne systemy sterowania procesami przemysłowymi, dystrybucją energii elektrycznej czy inteligentnymi budynkami bez rozmaitych odmian przekaźników elektromechanicznych oraz półprzewodnikowych. W czerwcowym wydaniu „Elektroniki Praktycznej” przyjrzymy się najważniejszym trendom na rynku przekaźników do automatyki, opiszemy kluczowe parametry tych podzespołów oraz wskażemy wybrane przekaźniki do różnych zastosowań, które dostępne są w aktualnej ofercie rynkowej czołowych producentów.

Komunikacja bezprzewodowa w IoT

Internet Rzeczy to bez wątpienia jeden z wyznaczników technologii XXI wieku. Szacuje się, że na świecie pracuje obecnie 15 miliardów urządzeń połączonych z siecią w celu wymiany różnego rodzaju danych – a liczba ta z roku na rok intensywnie rośnie. Inteligentne miasta, bezpieczniejsze i bardziej ergonomiczne budynki, sprawniejsza opieka zdrowotna, przenośne i ubieralne urządzenia podnoszące jakość życia milionów użytkowników – to tylko niektóre z licznych obszarów zastosowań urządzeń IoT. W czerwcowej „Elektronice Praktycznej” sporo miejsca poświęcimy technologiom komunikacji bezprzewodowej, bez której rozwój Internetu Rzeczy byłby niemożliwy. Omówimy najważniejsze standardy transmisji danych i zwrócimy uwagę na istotne aspekty ich implementacji w różnych rodzajach aplikacji docelowych. Dla naszych Czytelników przygotowaliśmy ponadto niezwykle interesujące i użyteczne materiały szkoleniowe, które pozwolą szybko rozpocząć tworzenie własnych projektów IoT w oparciu o czołowe platformy sprzętowe i programistyczne.

Wykaz firm ogłaszających się w tym numerze „Elektroniki Praktycznej”

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, Adam Sobczyk, 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

PCBWay

Prosty sposób na prototyp PCB

Kompleksowe usługi prototypowania PCB

Nawet do 60% taniej

Elastyczne obwody drukowane

Specjalna cena - tylko

$46.74

Obwody FPC 1- i 2-warstwowe

- Ilość sztuk w zamówieniu: 1...5
- Rozmiary: do 5 × 5 cm
- Grubość: 0,1 mm
- Pokrycie: złoto immersyjne (ENIG)

O PCBWay:

PCBWay to dostawca usług w zakresie produkcji płytek drukowanych, montażu PCB oraz integracji produktów końcowych. W czasie ponad 10 lat rynkowej obecności firma osiągnęła czołową pozycję w branży.

Adres URL:
www.pcbway.com

Poczta:
service@pcbway.com