Wydanie elektroniczne dostępne na www.ulubionykiosk.pl

ELEKTRONIKA PRAKTYCZNA

EP.com.pl

● Międzynarodowy magazyn elektroników konstruktorów ● sierpień ● 8/2024 ●

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 źródowe projektów na naszym serwerze **FTP** www.ulubionykiosk.pl/media

inspirujące, użyteczne projekty

• Sonda logarytmiczna z układem AD8307 • Termostat do pieca gazowego • Podwójny konwerter napięcia z 0...5 V na 0...10 V • Minimoduł z transceiverem CAN • Moduł przełącznika obciążenia typu low-side do Raspberry Pi • Termometr – minutnik

podzespoły, sprzęt, aplikacje

• Elektroniczne komponenty zabezpieczające – podzespoły, aplikacje i normy • Złącza pływające ze znaczną kompensacją tolerancji do szybkiej transmisji danych • Perspektywy rozwoju układów scalonych open source we współpracy z IHP Microelectronics • „Bezpieczeństwo przede wszystkim" w projektowaniu urządzeń IoT • Testowanie układów scalonych • Badania EMC w ofercie Centralnego Laboratorium Aparatury Medycznej CELAMED

tutoriale

• Internet Rzeczy w pomiarach środowiskowych. Czujnik wielospektralny AS7341 firmy ams-OSRAM • Uwaga na diody zabezpieczające! • Oszczędzanie energii w teorii i w praktyce • Amatorskie pomiary wzmacniaczy audio • EMC – checklista projektanta, czyli jak nie popełnić błędów? • Zagadkowy mikrokontroler TW32F003

kursy

• Kurs Nordic nRF z BT. Przyciski i wielowątkowość • Środowisko ESP-IDF. Wi-Fi i podstawowe tryby pracy • Kurs FPGA Lattice. Sekwencyjny algorytm Double Dabble

ISSN 1230-3526 Indeks 357677
9 771230 352245 08
18,90 zł (w tym 8% VAT) • PRICE: 8 EUR

EMC – JAK NIE POPEŁNIĆ BŁĘDÓW

eprasa.pl 4202fe84f2

prenumerata

–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 | AVTKorporacja sp. z o.o. ul. Leszczynowa 11, 03-197 Warszawa
rachunek bankowy: 18 1050 1012 1000 0024 3173 1013

Nie ma fal?

O kompatybilności elektromagnetycznej napisano już chyba wszystko. Trudno wskazać tu jakiś obszar, który jeszcze nie został poddany dogłębnej eksploracji, nie doczekał się modeli matematycznych i symulacji komputerowych albo też nie został opisany odpowiednimi normami branżowymi. A jednak wciąż powstają kolejne książki, kursy internetowe oraz stacjonarne, artykuły i… nowe wersje międzynarodowych standardów. Dlaczego?

Przyczyn tego stanu rzeczy jest, jak zwykle, kilka. Kompatybilność elektromagnetyczna podąża (bo musi!) za postępem technologicznym. Jeszcze kilkanaście lat temu mikrokontroler taktowany częstotliwością na poziomie 80 MHz był wręcz demonem wydajności w swojej klasie. Dziś kilkaset megaherców staje się standardem, a więc pasmo zaburzeń generowanych przez sam tylko procesor wkroczyło w zakres gigaherców (trzeba wszak uwzględnić nie tylko częstotliwość podstawową, ale także przynajmniej kilka pierwszych harmonicznych – transformacja Fouriera wiecznie żywa!).

Jeszcze szybciej niż pasmo częstotliwości, w którym pracują współczesne urządzenia elektroniczne, rośnie liczba tychże sprzętów. Dziś większość z nas nosi przy sobie (lub na sobie) przynajmniej kilka urządzeń – smartwatch, smartfon, smartband, słuchawki Bluetooth… każde z nich wytwarza i odbiera fale radiowe, a tłok w mitycznym eterze zaczyna przypominać tłum podróżnych na wielkomiejskim dworcu kolejowym w samym środku wakacji. Analogia nie jest przypadkowa, bo im więcej urządzeń jednocześnie nadaje, tym silniejszy generują one smog elektromagnetyczny. Duże skupiska ludzi tylko potęgują ten efekt – a przecież coraz większa liczba obiektów w infrastrukturze urbanistycznej także „nadaje". Inteligentne systemy oświetlenia, sieci czujników do monitorowania pogody lub ruchu drogowego, śmietniki informujące nadrzędny system o przepełnieniu, zdalnie odczytywane mierniki zużycia mediów – to tylko niektóre spośród setek przykładów urządzeń tworzących współczesny obraz Internetu Rzeczy.

Kolejny powód rosnącego znaczenia (oraz zainteresowania) tematyką EMC jest tyleż zaskakujący, co prozaiczny. Analizując materiały dostępne w czasopismach, notach aplikacyjnych, tutorialach i wielu innych, można dość szybko dojść do wniosku, że świadomość zagadnień kompatybilności elektromagnetycznej jest wyraźnie niższa niż w przypadku problematyki systemów wbudowanych (nawet tych wysoce zaawansowanych), oprogramowania, standardów telekomunikacyjnych itp. Taki stan rzeczy wynika zapewne po części z tego, że operując na polu EMC, trzeba przestawić się na inny sposób myślenia o elektronice – tutaj niepodzielnie rządzi bowiem analiza w dziedzinie częstotliwości, a nie czasu. W ślad za tym idzie kolejny prawdopodobny powód trudności, jakie inżynierom sprawia kompatybilność elektromagnetyczna – o ile bowiem każdy praktykujący elektronik może dziś z powodzeniem badać sygnały analogowe i cyfrowe w domenie czasu (doskonałe oscyloskopy cyfrowe czy analizatory stanów logicznych są przecież dostępne w cenach, jakie jeszcze dwie lub trzy dekady temu nikomu nawet się nie śniły), o tyle już z analizą widmową sprawa nie wygląda tak różowo. Oscyloskopy są wprawdzie wyposażone w funkcję FFT, ale jej przydatność jest (w przypadku dużej części obecnego na rynku sprzętu pomiarowego) mocno dyskusyjna, a i pasmo oscyloskopów zazwyczaj kończy się gdzieś hen, daleko przed początkiem najczęściej eksploatowanego spektrum fal radiowych (mowa tutaj głównie o paśmie S i zawartym w nim wszechobecnym zakresie ~2,4 GHz). Możliwości bezpośredniej obserwacji tego, na ile „kulturalnie" zachowuje się projektowane przez nas urządzenie w eterze, są zatem ograniczone – mogą sobie na to pozwolić firmy, które zainwestują w odpowiednie wyposażenie: analizator widma (najlepiej czasu rzeczywistego), sondy pola bliskiego, komory TEM, generatory RF, wzmacniacze, oprogramowanie analityczne i wiele innych potrzebnych akcesoriów. Na rynku funkcjonują oczywiście firmy na własne potrzeby budujące pełnoprawne komory SAC, ale milionowe koszty, które należy utopić w tego typu inwestycji, muszą się jakoś zwrócić. Dlatego też ci spośród producentów urządzeń elektronicznych, którzy prowadzą od lat intensywną działalność B+R, rozszerzają zwykle zakres usług B2B o badania prowadzone na rzecz zewnętrznych podmiotów.

Na koniec wypadałoby wskazać jeszcze jeden powód trudności, jakie stawia przed konstruktorami świat EMC. Przyczyny należy bowiem szukać także w… naszej psychice i doświadczeniach, które większość inżynierów wynosi ze studiów. Świat elektromagnetyzmu kojarzy się zwykle ze skomplikowanymi zagadnieniami na styku fizyki pól EM i teorii obwodów – stąd wielu praktyków podchodzi do sprawy niczym przysłowiowy pies do jeża. Trudno się temu dziwić – EMC potrafi zaskoczyć nawet wytrawnych graczy, a na dodatek pracownicy laboratoriów kompatybilności elektromagnetycznej lubują się w opowieściach o dziwnych zakłóceniach i rezonansach, przez które twórcy badanych urządzeń rwali sobie garściami włosy z głowy w trakcie badań precompliance lub (co gorsza!) właściwych badań certyfikacyjnych.

Jakie jest zatem remedium na opisane problemy? I tutaj znów można wskazać kilka ścieżek. Po pierwsze: warto czerpać praktyczne doświadczenie z wszelkiego rodzaju badań EMC, gdyż wyrobienie sobie pewnej intuicji inżynierskiej pozwala zidentyfikować potencjalne źródła problemów i zaradzić im na wcześniejszym etapie projektu, jeszcze przed dostarczeniem próbek urządzenia do laboratorium. Po drugie: rozwój, rozwój i jeszcze raz rozwój. W Internecie można znaleźć naprawdę wiele interesujących materiałów (zarówno płatnych, jak i bezpłatnych), które pozwolą poszerzyć wiedzę i uporządkować już posiadane informacje. Po trzecie: warto pamiętać własne błędy i obserwować pomyłki popełniane przez innych, gdyż to właśnie identyfikacja oraz rozwiązywanie problemów dają najlepszy wgląd w arkana „sztuk technicznych".

W sierpniowym wydaniu „Elektroniki Praktycznej" poświęcamy tematyce EMC wyjątkowo dużo miejsca – oprócz przeglądowego artykułu w formie checklisty projektanta, mamy także sporo materiału poświęconego ochronie przeciwprzepięciowej oraz metodom testowania komponentów i układów pod kątem odporności na wyładowania ESD. Do rubryki Notatnik Konstruktora trafiły tym razem trzy kolejne artykuły, a dla Czytelników spragnionych interesujących konstrukcji mamy aż sześć różnorodnych projektów i miniprojektów. Osoby zainteresowane niszowymi komponentami także znajdą coś dla siebie – w dziewięciostronicowym artykule rozwiązujemy zagadkę niezwykle tanich mikrokontrolerów ARM. Choć same układy z łatwością można nabyć na popularnych portalach handlowych, to już pozyskanie o nich jakichkolwiek bardziej szczegółowych informacji okazuje się nie lada wyczynem. Jak zawsze kontynuujemy także rozpoczęte wcześniej cykle i kursy poświęcone czujnikom w aplikacjach IoT, programowaniu modułów NRF i ESP32 oraz macierzy FPGA.

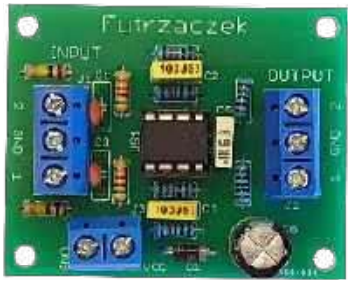
Życzę naszym Czytelnikom interesującej lektury i… udanej końcówki wakacji (lub „bezbolesnego" powrotu do pracy)!

Przemysław Musz

eprasa.pl 4202fe84f2

ELEKTRONIKA PRAKTYCZNA

Nr 8 (379) • Sierpień 2024



Nie przeocz
Nowe podzespoły 5
Dodaj do obserwowanych 10
Koktajl niusów 96

Projekty
Sonda logarytmiczna z układem AD8307 (1) 12
Termostat do pieca gazowego (2) 17

Miniprojekty
Podwójny konwerter napięcia z 0...5 V na 0...10 V 22
Minimoduł z transceiverem CAN 24
Moduł przełącznika obciążenia typu low-side do Raspberry Pi 26

Projekty czytelników
Termometr – minutnik 28

Temat numeru: Elektroniczne komponenty zabezpieczające
Elektroniczne komponenty zabezpieczające – podzespoły, aplikacje i normy 32

Prezentacje
Złącza pływające ze znaczną kompensacją tolerancji do szybkiej transmisji danych 40
Perspektywy rozwoju układów scalonych open source we współpracy z IHP Microelectronics 42
„Bezpieczeństwo przede wszystkim" w projektowaniu urządzeń IoT 44
Testowanie układów scalonych 60
Badania EMC w ofercie Centralnego Laboratorium Aparatury Medycznej CELAMED 64

Moduły w aplikacjach
Internet Rzeczy w pomiarach środowiskowych (8). Czujnik wielospektralny AS7341 firmy ams-OSRAM 46

Notatnik konstruktora
Uwaga na diody zabezpieczające! 51
Oszczędzanie energii w teorii i w praktyce (3) 66
Amatorskie pomiary wzmacniaczy audio (2) 70

Elektronika w praktyce
EMC – checklista projektanta, czyli jak nie popełnić błędów? 54

Podzespoły
Zagadkowy mikrokontroler TW32F003 74

Kursy
Kurs Nordic nRF z BT (3). Przyciski i wielowątkowość 82
Środowisko ESP-IDF (3). Wi-Fi i podstawowe tryby pracy 85
Kurs FPGA Lattice (22). Sekwencyjny algorytm Double Dabble 89

Prenumerata 2
Od wydawcy 5
Hity następnego numeru 99

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

eprasa.pl 4202fe84f2

Szczegóły na stronie 16

nowe podzespoły

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

Wzmacniacze mocy i niskoszumowe do samochodowych kompensatorów sygnału 4G/5G

Guerrilla RF wprowadza na rynek nowe wzmacniacze mocy i wzmacniacze niskoszumowe do samochodowych kompensatorów sygnału sieci komórkowej 4G/5G (tytułem wyjaśnienia: mowa o wzmacniaczach przeznaczonych do poprawy jakości połączenia w kabinie pojazdu). Są to układy z kwalifikacją AEC-Q100 Class 2, wyróżniające się wysoką niezawodnością i przystosowane do pracy w temperaturze otoczenia od –40 do +105°C.

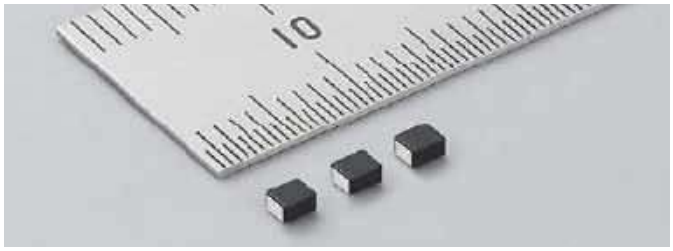
Nowa oferta obejmuje wzmacniacze mocy GRF5507W i GRF5517W o doskonałej liniowości, uzupełniające wcześniejsze wersje GRF5526W i GRF5536W. Pokrywają one zakres częstotliwości od 700 MHz do 4200 MHz i charakteryzują się maksymalną mocą wyjściową 23 dBm przy współczynniku ACLR lepszym niż –45 dBc.

Z kolei GRF2106W i GRF2133W to nowe wzmacniacze niskoszumowe na zakres częstotliwości pracy odpowiednio 2700...5000 MH z i 1800...5000 MHz. GRF2106W charakteryzuje się wzmocnieniem nominalnym 21,5 dB i współczynnikiem szumu 0,8 dB. W przypadku GRF2133 parametry te wynoszą odpowiednio 28 dB i 0,6 dB.

GRF5507W i GRF5517W produkowane są w obudowach QFN-16 o wymiarach 3×3 mm. Ich ceny hurtowe zaczynają się od 1,54 USD przy zamówieniach 10 tys. sztuk. Natomiast GRF2106W i GRF2133W są produkowane w obudowach DFN-6 (1,5×1,5 mm), a ich ceny hurtowe to odpowiednio 0,62 USD i 0,83 USD.

Do wszystkich 4 układów producent oferuje płytki ewaluacyjne.

www.guerrilla-rf.com



Chipowe cewki indukcyjne o małej rezystancji DC i wysokim napięciu znamionowym

Murata rozszerza ofertę cewek indukcyjnych do aplikacji motoryzacyjnych o dwa pierwsze warianty – o indukcyjności 0,33 μH i 0,47 μH – należące do nowej serii DFE2MCPH_JL. W porównaniu z odpowiednikami z wcześniejszej serii DFE2MCAH_J0, cewka DFE2MCPHR33MJLLQ (0,33 μH) charakteryzuje się zredukowaną o 38% rezystancją DC, wynoszącą 18 mΩ oraz zwiększonym

REKLAMA

HAMMOND®

1590 odlew aluminiowy IP54/IP65

Dowiedz się więcej:
hammondmfg.com/1590

eusales@hammondmfg.com • + 44 1256 812812

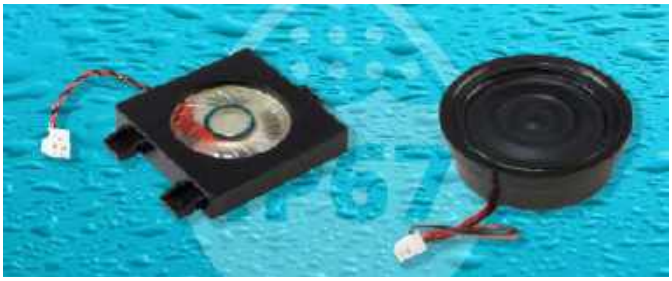
eprasa.pl 4202fe84f2

	L_0	Rezystancja DC		Prąd znamionowy			
		Rezystancja DC		dla spadku indukcyjności		dla wzrostu temperatury	
		maks.	typ.	min.	typ.	min.	typ.
DFE2MCPHR33MJLLQ	0,33 µH	18 mΩ	14 mΩ	6,9 A	8,1 A	5,1 A	6,4 A
DFE2MCPHR47MJLLQ	0,47 µH	22 mΩ	19 mΩ	5,4 A	6,3 A	4,4 A	5,5 A

o 31% dopuszczalnym prądem przewodzenia, sięgającym 5,1 A przy tym samym wzroście temperatury. Prąd znamionowy przy identycznym spadku indukcyjności jest większy o 64% i wynosi 6,9 A. Ponadto cewka charakteryzuje się napięciem znamionowym 40 V przy relatywnie małych gabarytach (rozmiar 0806) – oraz szerokim zakresem temperatury roboczej (od –40 do +155°C).

W najbliższym czasie oferta cewek serii DFE2MCPH_JL ma zostać uzupełniona o kolejne wersje, produkowane na zakres indukcyjności od 0,1 do 4,7 µH.

www.murata.com



Wodoodporne głośniki zamknięte firmy CUI Devices w wersjach o stopniu ochrony IPX7 i IP67

Oddział Audio firmy CUI Devices obwieścił, że do rodziny głośników zamkniętych CES dołączają nowe, wodoodporne modele o stopniu ochrony IPX7 i IP67. Wśród najważniejszych parametrów nowych urządzeń wymienić należy poziom ciśnienia akustycznego (SPL) od 95 do 109 dB w odległości 10 cm, zakres częstotliwości rezonansowej od 750 do 1300 Hz i moc znamionową od 1 do 3 W. Dzięki zamkniętej konstrukcji modele te doskonale nadają się do pracy w miejscach narażonych na wilgoć i pył.

Nowa oferta obejmuje głośniki w obudowach owalnych, prostokątnych, okrągłych lub kwadratowych ze stożkami PEEK, PEI lub PET, o wymiarach od 24×15×6,55 mm do 61,09×61,09×23,35 mm. W zależności od wersji ich impedancja znamionowa wynosi 4 lub 8 Ω, a one same oferowane są ze złączem lub kablem. Wybrane modele wyposażono również w otwory do montażu panelowego.

Ceny nowych głośników z oferty CUI Devices zaczynają się od 2,20 USD przy zamówieniach 100 sztuk.

www.cuidevices.com

Miniaturowe przekaźniki kontaktronowe o napięciu przebicia 5 kV

Firma Pickering Electronics wprowadza do oferty najmniejsze na rynku przekaźniki kontaktronowe o napięciu przebicia 5 kV, produkowane w obudowach SIP o wymiarach 24×8,1×6,2 mm (W×H×D), zajmujących około 1/6 powierzchni poprzednich wersji. Nowe przekaźniki mogą przełączać napięcia do 1,5 kV – producent poleca je zatem do zastosowań w testerach podzespołów półprzewodnikowych i kabli, elektronice medycznej i wszelkiego typu urządzeniach wysokonapięciowych.

Przekaźniki z serii 104 występują w konfiguracji 1 Form A z cewkami o napięciu znamionowym 5, 12 lub 24 VDC. Opcjonalnie mogą zawierać wewnętrzną diodę zabezpieczającą. Charakteryzują się mocą znamionową 25 W i maksymalnym prądem przełączanym 1 A. Zakres dopuszczalnej temperatury ich pracy rozciąga się od –40 do +125°C, przy czym na zamówienie mogą zostać dostarczone również warianty o temperaturze pracy zwiększonej do +150°C. Przekaźniki serii 104 zapewniają odporność na udary i wibracje do – odpowiednio – 50 g i 20 g (10...2000 Hz).

www.pickeringrelay.com

Szybkie mikrokontrolery z rdzeniem ARM Cortex-M85 do układów napędowych

Oferta mikrokontrolerów do układów napędowych firmy Renesas powiększyła się o nową serię energooszczędnych modeli RA8T1, opartych na rdzeniu ARM Cortex-M85 i wyposażonych w zestaw specjalnie dobranych bloków analogowych. Korzystają one z technologii bezpieczeństwa ARM TrustZone oraz technologii ARM Helium, zapewniającej 4-krotny wzrost szybkości cyfrowego przetwarzania sygnałów i algorytmów uczenia maszynowego – w porównaniu z mikrokontrolerami opartymi na jednostce ARM Cortex-M7. W niektórych aplikacjach pozwala to wyeliminować konieczność stosowania zewnętrznego akceleratora sprzętowego.

Mikrokontrolery RA8T1 są obsługiwane przez pakiet FSP (Flexible Software Package) firmy Renesas. Umożliwia on szybsze tworzenie aplikacji, ponieważ dostarcza całe potrzebne oprogramowanie infrastrukturalne, w tym: RTOS, BSP, sterowniki urządzeń peryferyjnych, oprogramowanie pośredniczące (middleware), stosy protokołów bezpieczeństwa oraz oprogramowanie referencyjne do tworzenia złożonych aplikacji AI, sterowania silnikami i komunikacji z chmurą. Ułatwia to klientom integrację własnego, opracowanego wcześniej kodu i zapewnia pełną elastyczność w tworzeniu aplikacji.

Pozostałe cechy mikrokontrolerów RA8T1:

- moc obliczeniowa: 6,39 CoreMark/MHz,
- do 2 MB pamięci Flash dual-bank + 1 MB pamięci SRAM (w tym 128 KB TCM),
- akcelerator kryptograficzny Renesas Security IP (RSIP-E51A),
- funkcja bezpiecznego rozruchu (secure boot),
- 18-bitowe timery PWM,
- bloki analogowe: 12-bitowe przetworniki A/C i C/A, szybkie komparatory do pomiaru napięcia/prądu, czujnik temperatury,
- interfejsy: SCI, SPI, I^2C/I^3C, CAN/CAN-FD, Ethernet i USB 2.0 full-speed,
- obudowy: BGA-224, LQFP-100, LQFP-144 i LQFP-176.

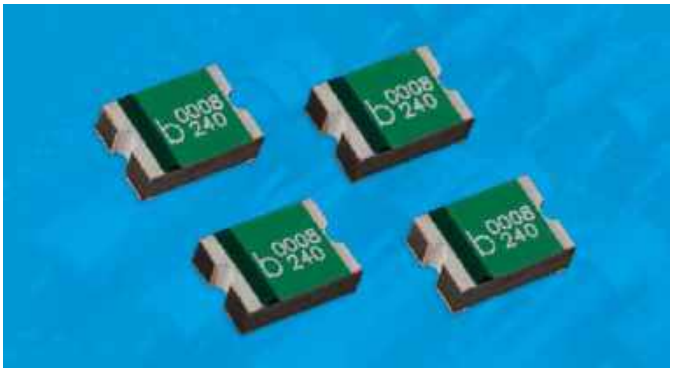
www.renesas.com

Resetowalny bezpiecznik SMD o napięciu znamionowym 240 VAC

Do oferty firmy Bel Fuse wchodzi seria resetowalnych bezpieczników PPTC do montażu powierzchniowego. To pierwsze tego typu elementy dostępne na rynku, których napięcie znamionowe wynosi 240 VAC. Zapewniają one ochronę przed przeciążeniami i zwarciami w sektorach: IoT, przemysłowym oraz medycznym. Typowe zastosowania obejmują przemysłowe systemy sterowania, aparaturę pomiarową, robotykę i instalacje zabezpieczające.

Seria 0ZAF obejmuje obecnie 5 typów bezpieczników o prądzie znamionowym (trzymania) od 50 do 160 mA, przystosowanych

eprasa.pl 4202fe84f2



do pracy w przemysłowym zakresie temperatury otoczenia od –40 do +85°C. Są one produkowane w obudowach rozmiaru 2920 (maks. 8,0×5,4×2,6 mm). Charakteryzują się zdolnością wyłączania od 1,2 do 2,0 A – w zależności od modelu – i mocą znamionową 1,5 W.

Bezpieczniki 0ZAF są dostępne w ofercie m.in. Digikey, Mouser czy Newark.

www.belfuse.com

Nowa generacja nadajników Bluetooth z zasilaniem typu *energy harvest* o dwukrotnie większym zasięgu transmisji

Nowe moduły radiowe drugiej generacji, wprowadzone na rynek przez markę EnOcean, oferują większy zasięg w porównaniu do oryginalnego modelu PTM 215B. W nadajnikach PTM 216B zastosowano nowo opracowany konwerter energii kinetycznej EnOcean ECO 260, pozwalający na ponad dwukrotne zwiększenie zasięgu transmisji (75 m na zewnątrz i 15 m wewnątrz budynków). Pozwala to na poszerzenie zakresu zastosowań nadajnika o systemy oświetleniowe m.in. dużych obiektów magazynowych oraz wszelkiego typu budowli z wysokim sklepieniem.

PTM 216B jest konfigurowany bezprzewodowo przez NFC (ISO 14443). Umożliwia to przeprowadzanie konfiguracji systemu nawet przez początkujących użytkowników, bez konieczności ponownej wizyty w celu ewentualnego korygowania parametrów. Moduł ma wymiary 40×40×11,2 mm i obsługuje maksymalnie 4 przełączniki. Może pracować w temperaturze otoczenia od –25 do +65°C. Producent deklaruje jego niezawodność na poziomie 100 tys. cykli mechanicznych.

Pozostałe właściwości:

- standard komunikacji: Bluetooth LE (Advertising Mode) 2,4 GHz,
- obsługiwane kanały BLE: 37, 38, 39,
- maks. moc nadajnika: +4 dBm,
- bezpieczeństwo: AES128,
- wbudowana antena PCB,
- unikalny 48-bitowy numer ID każdego modułu.

www.enocean.com

REKLAMA

eprasa.pl 4202fe84f2

Wzmacniacz operacyjny zero-drift o wejściowym napięciu offsetu 9 µV i dryfcie 0,05 µV/°C

LMR1002F-LB jest pierwszym w ofercie firmy Rohm precyzyjnym wzmacniaczem zero-drift, który korzysta z techniki chopper do redukcji wpływu dryftu i szumów niskoczęstotliwościowych (stanowiących problem szczególnie w precyzyjnych aplikacjach pomiarowych). W odróżnieniu od konwencjonalnych wzmacniaczy o małym dryfcie, których wejściowe napięcie offsetu wynosi maksymalnie 150 µV, w LMR1002F-LB wartość tę ograniczono o 94% do zaledwie 9 µV. Eliminuje to konieczność stosowania komponentów zewnętrznych i oprogramowania do korekcji napięcia niezrównoważenia, ułatwiając projektowanie precyzyjnych układów pomiarowych i obniżając koszty implementacji. Dodatkowo maksymalny dryft temperaturowy napięcia offsetu wynosi zaledwie 0,05 µV/°C w zakresie temperatury roboczej od –40 do +125°C, co jest najlepszym wynikiem spośród wszystkich wzmacniaczy operacyjnych dostępnych obecnie w ofercie firmy Rohm.

Precyzyjne wzmacnianie sygnałów z czujników – przy jednoczesnej minimalizacji błędów związanych z wpływem temperatury – zwiększa dokładność sterowania procesami. W połączeniu z szerokim zakresem napięcia zasilania (od 2,7 do 5,5 V) oraz wejściem i wyjściem rail-to-rail taka konstrukcja czyni układ idealnym do zastosowań w wielu aplikacjach przemysłowych, np. falownikach, detektorach gazu, kontrolerach temperatury, przepływomierzach i miernikach ciśnienia.

LMR1002F-LB jest produkowany w obudowie SOP-8 o wymiarach 6,2×5,0×1,7 mm.

www.rohm.com

Miniaturowe cewki wysokoprądowe do układów zasilania urządzeń przenośnych

Taiyo Yuden powiększa ofertę wielowarstwowych, metalowych cewek indukcyjnych o nowy wariant LSCND1412HKTR24ME o indukcyjności 0,24 µH, zaprojektowany pod kątem zastosowań w układach zasilania urządzeń przenośnych. Jest to cewka o objętości mniejszej o 30% od wcześniejszego odpowiednika (LSCNE2012HKTR24MD) – wymiary obudowy to 1,4×1,2×0,8 mm. Charakteryzuje się wyższym o około 8% prądem nasycenia (6,5 A vs. 6,0 A). Do oferty wchodzi też jej odpowiednik o symbolu LSCND1412HKTR33MF, o takich samych gabarytach, lecz o większej indukcyjności (0,33 µH) oraz model LSCND2012HKTR24MF (0,24 µH), w obudowie o rozmiarach 2,0×1,2 mm. W porównaniu z wcześniejszą wersją wykazuje on większy o 17% prąd nasycenia (7,0 A vs. 6,0 A).

	Powierzchnia	Grubość	Indukcyjność	Tolerancja	Prąd znamionowy		Rezystancja DC
					Prąd nasycenia	dla ΔT	
LSCND1412HKTR24ME	1,4×1,2 mm	0,8 mm	0,24 µH	±20%	6,5 A	4,4 A	22 mΩ
LSCND1412HKTR33MF			0,33 µH		5,4 A	4,0 A	27 mΩ
LSCND2012HKTR24MF	2,0×1,2 mm	0,8 mm	0,24 µH		7,0 A	5,4 A	19 mΩ

www.yuden.co.jp

Generatory sygnałów zegarowych PCIe 6.0 do zastosowań w elektronice samochodowej

Firma Diodes wprowadza na rynek serię generatorów sygnałów zegarowych PI6CG33xQ z wyjściami HCSL (high-speed current-steering logic), które mogą znaleźć zastosowanie w motoryzacji. Umożliwiają one taktowanie interfejsów PCIe 6.0/5.0/4.0 przy błędzie jitteru poniżej 40 fs (rms). Występują w wersjach 2-, 4-, 6- i 8-wyjściowych, kompatybilnych pod względem rozkładu wyprowadzeń z wcześniejszymi odpowiednikami.

Generatory z serii PI6CG33xQ mogą współpracować z rezonatorem 25 MHz lub zewnętrznym sygnałem referencyjnym CMOS i zawierają zestaw wyjść 100 MHz z wewnętrzną siecią dopasowującą, co eliminuje część zewnętrznych rezystorów i zmniejsza powierzchnię zajmowaną przez układ na płytce drukowanej. Ponadto wyposażono je w wyjścia Output Enable do systemu zarządzania zasilaniem. Amplituda i współczynnik slew-rate sygnałów wyjściowych są programowane za pomocą interfejsu SMBus, co ułatwia optymalizowanie parametrów pracy w aplikacjach docelowych.

Układy serii PI6CG33xQ występują w obudowach QFN z wyprowadzeniami typu *wettable flank*, ułatwiającymi inspekcję optyczną połączeń. Uzyskały kwalifikację AEC-Q100 Grade 2. Ich zakres zastosowań obejmuje samochodowe systemy informacyjno-rozrywkowe, nawigacyjne, wspomagania kierowcy itp.

www.diodes.com

Moduły komunikacyjne LTE-M z wbudowanym odbiornikiem GNSS

SARA-R52 i LEXI-R52 to najnowsze moduły komunikacyjne LTE-M z oferty firmy u-blox, wyposażone dodatkowo w odbiornik GNSS. Zostały zaprojektowane do zastosowań przemysłowych i aplikacji IoT, w tym mierników zużycia mediów i systemów śledzenia zasobów. Bazują na tanim, scalonym modemie UBX-R5 2. generacji.

Zestaw nowych funkcji modułów pozwolił na zmniejszenie liczby elementów peryferyjnych. SpotNow – nowa, unikalna funkcja pozycjonowania opracowana przez u-blox – umożliwia dostarczenie danych o położeniu z dokładnością do 10 m w ciągu kilku

eprasa.pl 4202fe84f2

sekund. Pozwala ona na zastosowanie układu w aplikacjach sporadycznie wymagających lokalizacji, takich jak kontenery do recyklingu, urządzenia do śledzenia osób starszych lub maszyny czyszczące. Funkcja nCPU pozwala użytkownikowi na uruchamianie własnego oprogramowania, bez potrzeby korzystania z zewnętrznego mikrokontrolera. Z kolei uSCM (Smart Connection Manager) to funkcja automatycznego zarządzania połączeniem, pozwalająca balansować pomiędzy małym poborem mocy a szybkością transmisji, np. podczas ponownego nawiązywania połączenia po jego zerwaniu.

Moduły nowej serii R52 charakteryzują się maksymalną mocą wyjściową 23 dBm, dzięki czemu zapewniają stabilną łączność w trudnych warunkach pracy – dla porównania warto dodać, że wiele modułów LTE-M oferuje moc wyjściową jedynie na poziomie 20...21 dBm. LEXI-R52 udostępnia te same funkcje, co SARA-R52, ale ma mniejszą obudowę o wymiarach 16×16×2 mm.

www.ublox.com

Tranzystor GaN HEMT 650 V w obudowie CCPAK copper-clip o małej indukcyjności

Nexperia informuje o rozpoczęciu produkcji nowej rodziny tranzystorów GaN FET, opartych na wysokonapięciowej technologii GaN HEMT. Elementy są oferowane w obudowach SMD typu CCPAK o wymiarach 12×9,4×2,5 mm, odpowiednich do zastosowań np. w spawarkach, falownikach fotowoltaicznych, wzmacniaczach mocy klasy D, serwonapędach, zasilaczach serwerowych klasy Titanium, pompach ciepła itp. W odróżnieniu od tradycyjnych obudów, zastosowana tu technika copper-clip pozwoliła wyeliminować wewnętrzne druciki łączące strukturę z wyprowadzeniami – taki zabieg poprawia parametry termiczne i redukuje wartość strat wynikających z wpływu indukcyjności pasożytniczej, której wartość zmniejszono 3-krotnie. W celu zapewnienia maksymalnej elastyczności producent udostępnia obudowy CCPAK w wariantach z odprowadzaniem ciepła przez dolną lub górną powierzchnię (bottom-cooled lub top-cooled).

W ofercie firmy Nexperia jest już dostępny pierwszy model o symbolu GAN039-650NTB, o napięciu przebicia 650 V i rezystancji RDS(ON) równej 33 mΩ w temperaturze +25°C. Jest on produkowany w obudowie CCPAK top-cooled, a w najbliższym czasie do dystrybucji ma trafić również jego odpowiednik bottom-cooled (ozn. GAN039-650NBB) – o identycznej rezystancji RDS(ON). Tranzystor może pracować przy maksymalnej temperaturze złącza +150°C i charakteryzuje się bardzo małą rezystancją termiczną Rth(j-mb), poniżej 0,5 K/W. Jego maksymalny prąd przewodzenia wynosi 58,5 A, a dopuszczalny zakres napięcia VGS to ±20 V.

www.nexperia.com

Tanie mikrokontrolery do układów napędowych i współpracy z czujnikami przemysłowymi

NXP Semiconductors zapowiada wprowadzenie do sprzedaży dwóch nowych serii tanich mikrokontrolerów: MCX A14x i MCX A15x, wyposażonych w zestaw podstawowych funkcji do współpracy m.in. z czujnikami przemysłowymi, układami napędowymi silników BLDC/PMSM i układami zarządzania zasilaniem urządzeń bateryjnych. Są one wspierane przez środowisko projektowe MCUXpresso Developer Experience i nowe płytki prototypowe FRDM. Mogą pracować pod kontrolą systemów FreeRTOS i Zephyr.

Mikrokontrolery MCX A14x i MCX A15x zostały oparte na jednostkach obliczeniowych ARM Cortex-M33, przy czym różnią się maksymalną częstotliwością taktowania, wynoszącą odpowiednio 48 MHz i 96 MHz. Zawierają zestaw energooszczędnych funkcji peryferyjnych, interfejsy 3×UART, 2×SPI, I²C, I³C i USB 2.0 oraz do 128 MB pamięci Flash i do 32 kB pamięci SRAM z korekcją ECC. Inteligentne bloki peryferyjne mogą pracować w trybie autonomicznym, bez angażowania jednostki CPU, co zmniejsza pobór mocy. Do bloków tych należą szeregowe porty komunikacyjne z wbudowanymi buforami, układ DMA, zestaw timerów wielofunkcyjnych, przetworniki A/C i C/A, komparatory, inteligentny wzmacniacz operacyjny z funkcjami uśredniania i pomiaru wartości szczytowej sygnału oraz blok FlexPWM z kontrolą czasu martwego i enkoderem do układów sterowania silnikami. Mikrokontrolery z obu serii są produkowane w trzech wariantach obudów: QFN-32 (5×5 mm), QFN-48 (7×7 mm) i LQFP-64 (10×10 mm).

www.nxp.com

Wodoszczelny czujnik ciśnienia barometrycznego o wymiarach 3,0×3,0×2,1 mm

ZPA4756 to czujnik ciśnienia barometrycznego do montażu SMD o wymiarach zaledwie 3,0×3,0×2,1 mm, zaprojektowany do zastosowań w smartfonach i laptopach. Oferuje zakres pomiarowy 300...1100 hPa i szeroki zakres temperatury roboczej: od –40 do +85°C. Zawiera cyfrowy interfejs SPI/I²C, pracujący z częstotliwością próbkowana do 256 Hz i wyprowadzający wyniki pomiaru w formacie 24-bitowym.

Jako układ zaprojektowany do urządzeń bateryjnych, ZPA4756 wyróżnia się małym poborem mocy. Przy napięciu zasilania z zakresu 1,7...3,6 V pobiera typowo 1,3 μA prądu, wykonując pomiary z interwałem 1 s. Bezwzględna dokładność pomiaru ciśnienia wynosi typowo ±0,4 hPa w całym zakresie pomiaru w temperaturze –20...+65°C. Dokładność względna to typowo ±0,020 hPa dla ΔP=25 hPa.

www.murata.com

eprasa.pl 4202fe84f2

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.

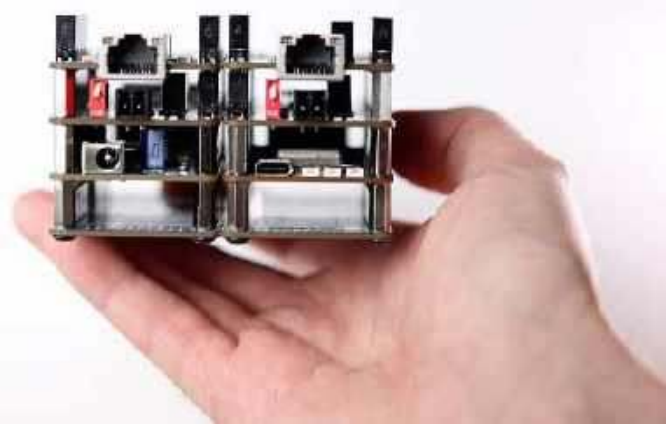
next.module – elastyczna i kompaktowa platforma do prototypowania

Prezentowany projekt obejmuje serię kompaktowych modułów, które można w niemalże dowolny sposób łączyć ze sobą, tworząc wysoce wyrafinowane i złożone, a jednocześnie dość kompaktowe urządzenie. Platforma next.module przeznaczona jest do prototypowania systemów wbudowanych.

Jak zapewnia sam autor konstrukcji, korzystanie z niej okaże się łatwe nawet dla początkujących entuzjastów elektroniki.

Każdy moduł realizuje jedną lub więcej podstawowych funkcji: pamięci, czujnika, rozszerzeń wejścia i/lub wyjścia, interfejsu. Dostępne są ponadto „klocki" z wyświetlaczem, przyciskami itp.

Moduły prezentowane przez autora zostały gruntownie przetestowane i działają niezawodnie. Ich użytkownicy mają dostęp do kompletnych schematów, a także lekkiego pakietu SDK z prostym interfejsem API do interakcji z każdym modułem. Co więcej można podłączyć tu standardową płytkę stykową, by stworzyć własny blok, oferujący dowolną potrzebną nam funkcjonalność.



„Zawsze chciałem używać małych modułów. W miarę możliwości i rozsądku zmieścić na nich wszystkie elementy w pełni funkcjonalnego zestawu. Idealne byłoby 50×50 mm lub nawet mniej", pisze projektant systemu o platformie next.module. „Połączenia elektryczne pomiędzy modułami muszą być niezawodne, moduły muszą być ze sobą trwale połączone, aby konstrukcja nie rozpadła się przy nieostrożnym ruchu na stole (...)".

Moduły można rozszerzać w pionie – to spory walor praktyczny – ale w dalszym ciągu konieczne jest rozszerzanie również w poziomie. W przeciwnym razie, przy dużej liczbie modułów, powstanie wąska i niestabilna wieża. Warto pamiętać również o opcji rozbudowy poziomej przy użyciu płyt HUB – oznacza to, że w jednym z rzędów umieszczona jest większa płytka HUB, rozmiarami odpowiadająca wielokrotności wymiarów modułu (na przykład 2×1, 1×2, 2×2, 3×2 itd.) i wyposażona w takie same złącza systemowe na górze i na dole oraz otwory montażowe, jak zwykłe moduły. Tego typu HUB-y umożliwiają równoległe podłączenie do sterowania wielu modułów i fizyczne spięcie ich w stabilny mechanicznie system.

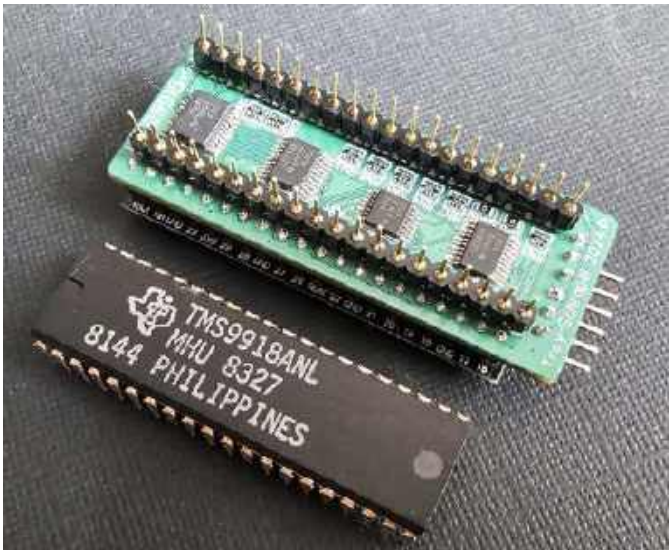
Główną wadę opisanego rozwiązania stanowi fakt, że HUB zajmuje jeden poziom w montażu, a połączenie mechaniczne pomiędzy modułami jest słabsze niż w przypadku konstrukcji bez HUB-ów.

Złącza do połączeń zewnętrznych spozycjonowano z 2-milimetrowym występem poza płytkę. Jest to konieczne, aby złącza te nie były wpuszczone w obudowę i nie wystawały z niej. Na drukarce 3D można wydrukować obudowę dla systemu o grubości ścianki równej 1,6 mm. Pozostała szczelina 0,4 mm to przestrzeń pomiędzy płytką a ścianką obudowy. Warto wspomnieć, że autor na stronie projektu udostępnia również stosowny projekt obudowy.

next.module jest obecnie w fazie aktywnego rozwoju. Jeśli spodobał Ci się ten pomysł, zacznij obserwować stronę projektu i udostępnij go swoim znajomym, którzy również mogą być zainteresowani. Wszystkie aktualności na temat działań realizowanych przez autora możesz znaleźć w sekcji „Logi projektu", w której krok po kroku opisuje on prowadzone prace.

https://hackaday.io/project/189481-nextmodule

eprasa.pl 4202fe84f2



Zamiennik układu VDP (TMS9918A) oparty na Raspberry Pi Pico

Celem projektu było stworzenie zamiennika układu VDP. Klasyczny układ scalony TMS9918A to kontroler wyświetlacza wideo (VDC) wyprodukowany przez firmę Texas Instruments, w dokumentacji często określany również mianem „procesora wideo" (VDP). Wprowadzony na rynek w 1979 roku, zdobył od tego czasu status komponentu niemalże kultowego – co wynikało z faktu, że był on używany w całej generacji minikomputerów osobistych z lat 80. XX wieku. Niestety obecnie nie jest już produkowany, a to utrudnia nawet naprawy maszyn retro, nie mówiąc już o planach konstruowania ich klonów.

TMS9918A generuje zarówno grafikę znakową opartą na siatce (używaną do wyświetlania tekstu lub obrazów tła), jak i sprite'y przeznaczone do przesuwania obiektów na pierwszym planie.

Zamiennik opisywanego układu – PICO9918 – używa Raspberry Pi Pico do emulacji działania VDP. W tej chwili jest on na wczesnym etapie prototypowania, ale (jak pisze autor) efekty prac wyglądają obiecująco, a nowa konstrukcja działa doskonale w oryginalnym Texas Instruments TI-99/4A.

Najnowsza wersja projektu oznaczona została numerem v0.3. Kolejnym krokiem planowanym przez autora jest zmniejszenie rozmiarów systemu poprzez bezpośrednie dołączenie RP2040, zamiast korzystania z oryginalnego modułu RPi Pico.

Schemat obecnej wersji jest już dostępny, jednakże – jak zastrzega autor – nie został jeszcze w pełni przetestowany.

Uwaga: Ze względu na fakt, że linie GROMCLK i CPUCLK korzystają z GPIO23 i GPIO29, nie można używać w systemie „prawdziwego" Raspberry Pi Pico. Model v0.3 jest przeznaczony do modułu DWEII RP2040 USB-C, który udostępnia te dodatkowe linie GPIO.

Kolejna wersja PICO9918 będzie korzystać bezpośrednio z RP2040, dlatego opisane zjawisko nie będzie stanowić problemu.

„Mógłbym zmniejszyć głębię bitową VGA do 9 lub 10 bitów, aby umożliwić użycie oryginalnej płytki Raspberry Pi Pico, ale biorąc pod uwagę plan długoterminowy, polegający na bezpośrednim użyciu układu RP2040, zdecydowałem się pójść tą drogą" – uzasadnia autor.

Na stronie projektu autor zamieścił film z działania modułu w wersji v0.2, z zewnętrznym Pi Pico dostarczającym wymagany sygnał GROMCLK do TI-99. Sygnał ten został dodany do wersji v0.3.

https://hackaday.io/project/196478-pico9918

Wzmacniacz słuchawkowy Porter dla gitarzystów

Porter to wzmacniacz słuchawkowy dla gitarzystów i basistów, którzy chcą ćwiczyć przy użyciu własnych efektów, bez denerwowania sąsiadów. Ma charakter raczej narzędzia niż zabawki i dostarcza bezpośredniej odpowiedzi na istniejący problem. Ponadto można podłączyć go bezpośrednio do wyjścia z gitary lub efektów gitarowych.

Dźwięk z Portera został dostrojony tak, aby zapewnić brzmienie przypominające wzmacniacz gitarowy, więc przesterowane dźwięki z efektów brzmią tak, jak powinny. To możliwie prosty, w pełni analogowy projekt. Wzmacniacz został zredukowany do najważniejszych elementów, niezbędnych do efektywnego działania; dostępne są tryby gitarowe i basowe, które odpowiednio zmieniają brzmienie systemu. Aby zaoszczędzić miejsce, autor rozwiązania potraktował port micro-USB również jako wejście liniowe. Bateria może być zarówno ładowana, jak i wymieniana, a jej wymiary eliminują konieczności częstego ładowania.

„Zacząłem projektować Portera w sierpniu 2022 roku, kiedy posiadanie współlokatorów i moja skłonność do solowych występów późno w nocy popchnęły mnie do poszukiwania sposobu na cichsze ćwiczenia. Mój wzmacniacz lampowy nie miał wyjścia słuchawkowego, a na wzmacniaczu półprzewodnikowym instrument nie brzmiał zbyt dobrze. Wszystkie opcje dostępne na rynku podłączane były bezpośrednio do gitary i wykluczały użycie efektów zewnętrznych, co było dla mnie bardzo ważne (...) podczas ćwiczenia dowolnych utworów wymagających różnych efektów w różnych momentach. Pomyślałem, że fajnie byłoby spróbować stworzyć własny wzmacniacz słuchawkowy" – opisuje autor.

W założeniu wymiary urządzenia miały odpowiadać mniej więcej wielkości paczki gumy do żucia. Początkowa wersja wzmacniacza była minimalistyczna, pozbawiona jakichkolwiek elementów sterujących, nawet wyłącznika (włączał się on po podłączeniu kabla – trick powszechnie stosowany w efektach i gitarach z wyjściami aktywnymi). Projekt ewoluował jednak na przestrzeni 6 wersji i prawie 2 lat – w tym czasie uzupełniony został o różne funkcje, które były niezbędne, aby potraktować go jako naprawdę skuteczne narzędzie do ćwiczeń. Rozmiar zwiększył się, ale dzięki temu projekt pasował do gotowej obudowy Hammonda.

https://hackaday.io/project/196611-porter

eprasa.pl 4202fe84f2

AVT sklep

W ofercie AVT*

AVT5991

Najważniejsze parametry:
- konstrukcja oparta na scalonym układzie AD8307,
- wbudowana przetwornica DC/DC o topologii podwajacza pojemnościowego,
- blok wejściowy z przełączanymi tłumikami 0/–20/–40 dB,
- impedancja wejściowa: 1 kΩ/10 kΩ/100 kΩ,
- impedancja wyjściowa: 50 Ω (złącze BNC),
- wzmocnienie regulowane potencjometrem (około 24,9 mV/dB ±9%),
- punkt przecięcia charakterystyki regulowany potencjometrem (±3dB),
- zasilanie: dwa ogniwa typu AAA.

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

AVT3257	Izolowana galwanicznie sonda oscyloskopowa 2,5 kV (EdW 10/2019)
AVT3241	Różnicowa sonda oscyloskopowa 20 MHz (EdW 5/2019)
AVT1962	Ekonomiczna sonda logarytmiczna (EP 8/2017)
AVT5378	Różnicowa sonda pomiarowa z izolacją galwaniczną (EP 1/2013)

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

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

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

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

Sonda logarytmiczna z układem AD8307 (1)

Szerokopasmowe sondy pomiarowe należą do przyrządów, których nie powinno zabraknąć w pracowni wszechstronnego elektronika-konstruktora, szczególnie jeśli jest on radioelektronikiem. Sondy logarytmiczne przydają się zwłaszcza do mierzenia napięć, a pośrednio – także prądów i mocy w obwodach prądów zmiennych. Do niezaprzeczalnych zalet tych przyrządów należy możliwość wykonywania dość precyzyjnych pomiarów sygnałów o bardzo dużej dynamice, w szerokim zakresie częstotliwości, przy czym – gdy w pomiarach zastosuje się generator-wobulator i oscyloskop – można prześledzić także charakterystyki częstotliwościowe badanego podzespołu czy urządzenia.

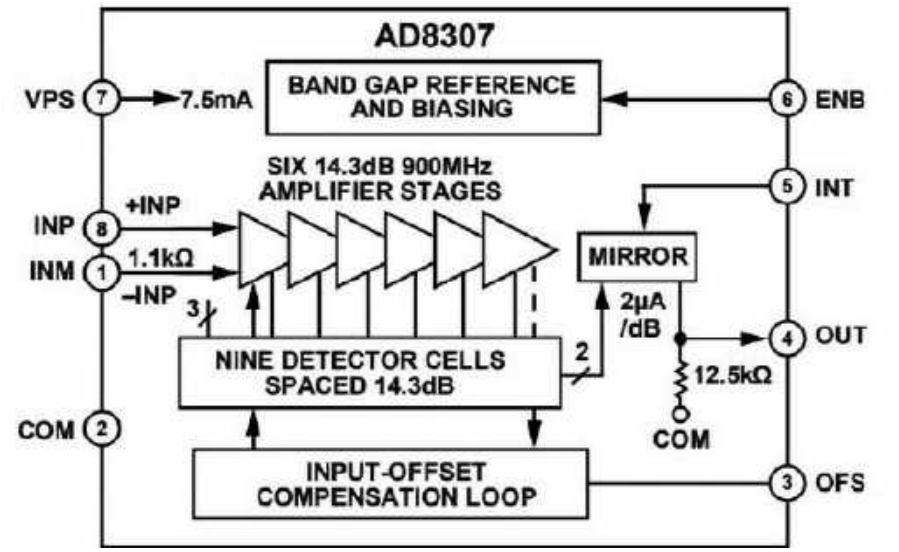
Prezentowany tutaj projekt sondy logarytmicznej jest rozwiniętą i znacznie ulepszoną wersją projektu prostej (ekonomicznej) poprzedniczki, zaprezentowanej w „Elektronice Praktycznej" 8/2017 (kit AVT1962), w której niemal całe przetwarzanie w torze sygnałowym zostało oparte na tanich elementach dyskretnych. Nowa odsłona sondy bazuje na specjalistycznym układzie scalonym AD8307 produkcji firmy Analog Devices i jest przystosowana do współpracy z generatorem-wobulatorem: AVT5580 (EP 2…4/2017) oraz jego nowszą wersją AVT5980 (EP 4…6/2023).

Zastosowanie i właściwości przyrządu

Jak wspomniano we wstępie do artykułu, opisana tutaj sonda logarytmiczna jest przyrządem przeznaczonym do bezpośredniego pomiaru napięć oraz pośredniego pomiaru prądów i mocy w obwodach prądów zmiennych w szerokim zakresie częstotliwości i o znacznej dynamice. W szczególności, przy włączeniu do pomiarów dodatkowo generatora-wobulatora i oscyloskopu z funkcjonalnością pracy w trybie XY można również analizować charakterystyki częstotliwościowe badanych czwórników. Motywacją do realizacji nowego projektu była chęć uzyskania parametrów użytkowych istotnie lepszych niż w przypadku stosunkowo nieskomplikowanej sondy logarytmicznej AVT1962. Efekt ten osiągnięto przede wszystkim dzięki zastosowaniu specjalizowanego układu AD8307, zapewniającego znacznie szerszy zakres dynamiki i pasma częstotliwości dla mierzonych sygnałów, przy całkiem niezłej dokładności przetwarzania logarytmicznego.

Obok głównego pomiarowego bloku przyrządu, opartego o układ AD8307 wraz z dwoma potencjometrami przeznaczonymi do regulacji charakterystyki przetwarzania, w urządzeniu można wyróżnić: blok zasilania (z multiwibratorem astabilnym oraz z przetwornicą pojemnościową w układzie podwajającym napięcie), blok precyzyjnego stabilizatora regulowanego, blok wejściowy z dwoma opcjonalnymi tłumikami (wybieranymi zworkami) oraz blok wzmacniaczy wyjściowych. Zanim jednak przejdziemy do szczegółowego omówienia wymienionych fragmentów układu, warto zapoznać się bliżej z właściwościami układu AD8307.

eprasa.pl 4202fe84f2



Rysunek 1. Schemat blokowy układu AD8307

Właściwości i aplikacje układu AD8307

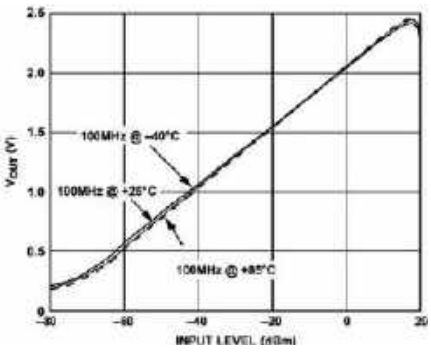
W tym rozdziale skorzystano przede wszystkim z informacji zawartych w karcie katalogowej układu AD8307. Blokowy schemat funkcjonalny układu pokazano na **rysunku 1**, natomiast opis i przeznaczenie jego poszczególnych wyprowadzeń zebrano w **tabeli 1**. Producent zdefiniował tę kostkę jako niskokosztowy, kompletny, wielostopniowy wzmacniacz logarytmiczny o zakresie dynamiki przetwarzania równym 92 dB, tj. od –75 dBm do +17 dBm – z możliwością obniżenia progu czułości aż do –90 dBm przy zastosowaniu odpowiedniego układu dopasowującego (przedwzmacniacza wstępnego). Układ AD8307 zawiera w pełni różnicową ścieżkę sygnału ze sprzężeniem prądu stałego. Dzięki temu, ze względu na symetryczny charakter odpowiedzi układu, piny INM (nr 1) oraz INP (nr 8) mogą być używane zamiennie: jako gorący – albo podłączony do masy (dla składowej zmiennej). Rezystancja wejściowa Rin, mierzona pomiędzy pinami INM oraz INP dla składowej stałej, wynosi Rin=1,1 kΩ. Typowe aplikacje układu AD8307, jako konwertera poziomu sygnału na postać decybelową, to m. in.: pomiar mocy emitowanej do anteny przez nadajnik radiokomunikacyjny, wskaźnik siły sygnału odbieranego przez odbiornik (RSSI), niedrogi przetwornik sygnałów radarowych i sonarowych, a także przetwornik logarytmiczny w wektorowych analizatorach obwodów zmiennoprądowych (VNA) i w analizatorach widma (o zakresie pomiaru do 120 dB). Warto podkreślić także możliwość zastosowania układu AD8307 do określania poziomu sygnałów zmiennych o częstotliwościach poniżej 20 Hz, co predestynuje tę kostkę do aplikacji w multimetrach z pomiarem napięć i prądów zmiennych w trybie „True RMS".

Układ scalony AD8307 jest jednym z pierwszych monolitycznych wzmacniaczy logarytmicznych dostępnych na rynku. Przetwarzanie logarytmiczne jest w nim oparte o technikę wieloetapowej, progresywnej kompresji z sukcesywnym wykrywaniem poziomu sygnału, zapewniającym podstawowy zakres dynamiczny aż do 92 dB (przy błędzie przetwarzania nieprzekraczającym ±3 dB) lub do 88 dB (przy błędzie przetwarzania nieprzekraczającym – zaledwie – ±1 dB), przy częstotliwościach mierzonych sygnałów sięgających do 100 MHz. Ten fakt – w zestawieniu z niewygórowaną ceną w sprzedaży detalicznej – sprawia, że układ doskonale wpasowuje się we wszelkie zastosowania radioamatorskie, szczególnie krótkofalarskie. Układ AD8307 uchodzi za bardzo stabilny i łatwy w implementacji – w podstawowym zastosowaniu wymaga tylko kilku biernych elementów zewnętrznych, bez jakichkolwiek podzespołów regulacyjnych. Jest dostępny zarówno w obudowie DIP-8, przeznaczonej do ręcznego montażu THT, jak i w obudowie SOIC-8, przeznaczonej do montażu powierzchniowego SMT.

Układ scalony AD8307 jest zasilany pojedynczym napięciem w zakresie od 2,7 V do 5,5 V (7,5 V max.), przy typowym natężeniu prądu zasilania równym 7,5 mA, a dopuszczalny zakres temperatur jego pracy mieści się w przedziale od –40 do +85°C. Deklarowany czas włączenia zasilania układu po przejściu pinu ENB (nr 6) ze stanu niskiego do wysokiego wynosi 100 ns, natomiast prąd uśpienia jest rzędu 150 μA. Zgodnie z deklaracjami producenta, omawiany przetwornik logarytmiczny umożliwia pracę w pasmie od 0 Hz (sygnały stałoprądowe) aż do 500 MHz z zachowaniem liniowości przetwarzania nie gorszej niż ±1 dB, co w większości popularnych (a zwłaszcza amatorskich) zastosowań pomiarowych wydaje się wysoko satysfakcjonującą wartością. Nominalne wartości (tzn. bez zewnętrznej regulacji dołączonymi potencjometrami) parametrów charakterystyki przetwarzania: nachylenia oraz lokalizacji teoretycznego punktu przecięcia z osią OX (tzn. nieosiąganego w praktyce, gdyż Uwy przetwornika nigdy nie wynosi dokładnie 0 V) wynoszą odpowiednio: 25 mV/dB oraz –84 dBm. Producent podkreśla znaczną odporność na zmiany temperatury otoczenia, a także dobrą stabilność temperaturową charakterystyk

Tabela 1. Opisy funkcji pinów układu AD8307

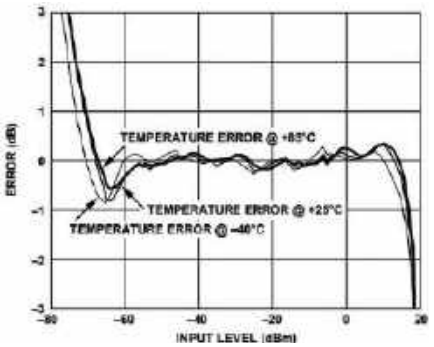
Nr pinu	Oznaczenie	Przeznaczenie pinu
1	INM	Wejście sygnału o polaryzacji ujemnej (wewnętrznie spolaryzowane do Uvps/2)
2	COM	Masa układu
3	OFS	Regulacja przesunięcia charakterystyki przenoszenia (rekomendowane podłączenie zewnętrznego kondensatora)
4	OUT	Wyjście napięcia reprezentującego zlogarytmowany sygnał wejściowy (RSSI). Rout=12,5 kΩ
5	INT	Regulacja lokalizacji teoretycznego punktu przecięcia charakterystyki przetwarzania z osią OX (zakres max.: ±3 dB)
6	ENB	Załączanie (aktywacja) układu poziomem wysokim (kompatybilna z poziomami CMOS, tzn. 0 V/Uvps)
7	VPS	Zasilanie dodatnie w zakresie Uvps=2,7...5,5 V
8	INP	Wejście sygnału o polaryzacji dodatniej (wewnętrznie spolaryzowane do Uvps/2)



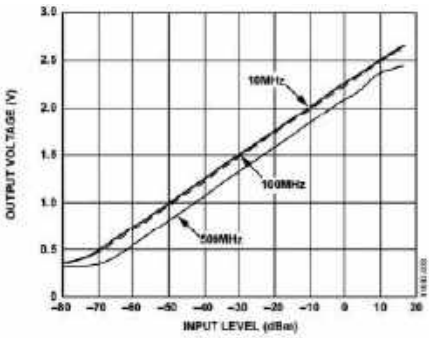
Rysunek 2. Wykres Vout w funkcji poziomu wejściowego w temperaturach –40°C, +25°C oraz +85°C

eprasa.pl 4202fe84f2

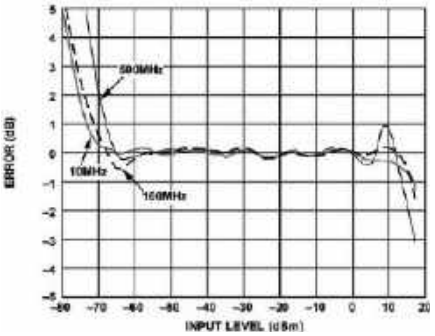
przetwarzania: Vout w funkcji poziomu wejściowego (**rysunek 2**) oraz logarytmicznej dokładności przetwarzania w funkcji poziomu wejściowego [dBm] (**rysunek 3**) – w temperaturach −40, +25 oraz +85°C.



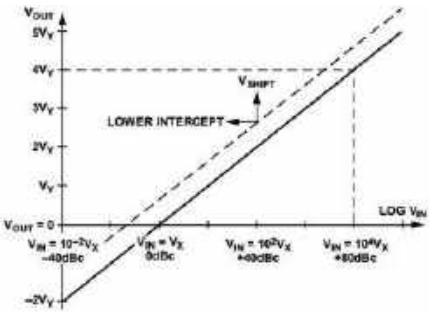
Rysunek 3. Wykres logarytmicznej dokładności przetwarzania w funkcji poziomu wejściowego [dBm] w temperaturach −40°C, +25°C oraz +85°C



Rysunek 4. Wykres Vout w funkcji poziomu wejściowego dla częstotliwości 10, 100 oraz 500 MHz



Rysunek 5. Wykres logarytmicznej dokładności przetwarzania w funkcji poziomu wejściowego [dBm] dla częstotliwości 10, 100 oraz 500 MHz



Rysunek 6. Funkcja przenoszenia idealnego wzmacniacza logarytmicznego

Podobnie jest w przypadku wrażliwości charakterystyk przetwarzania na częstotliwość mierzonego sygnału: Vout w funkcji poziomu wejściowego (**rysunek 4)** oraz logarytmicznej dokładności przetwarzania w funkcji poziomu wejściowego [dBm] (**rysunek 5)** przy częstotliwości 10, 100 oraz 500 MHz. Jest ona znikoma praktycznie aż do około 100 MHz (producent deklaruje użyteczność aplikacyjną układu aż do częstotliwości rzędu 500 MHz, jednak należy mieć pełną świadomość faktu, że jej uzyskanie będzie możliwe jedynie przy bardzo przemyślanej realizacji projektu PCB i to najlepiej wyłącznie w technologii SMD). Wymienione właściwości, szczególnie w przypadku niedrogiego, popularnego układu scalonego, obecnego na rynku już od około 20 lat, należy uznać za istotne zalety tego produktu.

Wróćmy na chwilę do rysunku 1, czyli blokowego schematu funkcjonalnego układu AD8307. Jego „sercem" jest sześć stopni wzmacniaczy-ograniczników oraz powiązanych z nimi detektorów poziomów i sumator wyjściowego sygnału prądowego. Każdy z wymienionych wzmacniaczy ma wzmocnienie 14,3 dB i pasmo przenoszenia 900 MHz, natomiast całkowite wzmocnienie tego bloku wynosi około 86 dB przy szerokości pasma –3 dB równym 500 MHz. Różnicowe wyjścia prądowe detektorów są sumowane, a następnie przekształcane na wyjście (także prądowe), nominalnie wyskalowane na 2 μA/dB. Logarytmiczne napięcie wyjściowe jest uzyskiwane na drodze przeniesienia prądu z wyjścia sumatora, poprzez wbudowane lustro prądowe, do wewnętrznego rezystora 12,5 kΩ, co daje logarytmiczne nachylenie charakterystyki przenoszenia równe około 25 mV/dB, czyli 500 mV na dekadę. Napięcie to jest wyprowadzone na pinie OUT (nr 4) układu AD8307 – warto dodać, że nie jest ono buforowane, co pozwala na zastosowanie różnych interfejsów wyjściowych, w tym rezystancji korygującej nachylenie charakterystyki przenoszenia, a także filtru podemodulacyjnego i/lub wzmacniacza wyjściowego. Ostatni stopień detektora pełni także funkcję korekcji temperaturowej charakterystyki przetwarzania logarytmicznego.

Cały blok przetwarzania logarytmicznego powinien realizować idealną funkcję przenoszenia, określaną jako:

$$V_{OUT}=V_Y\times\log(V_{IN}/V_X), \quad (1)$$

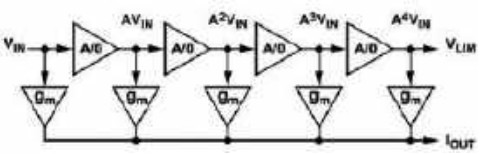
gdzie:

V_{OUT} – napięcie wyjściowe [V],

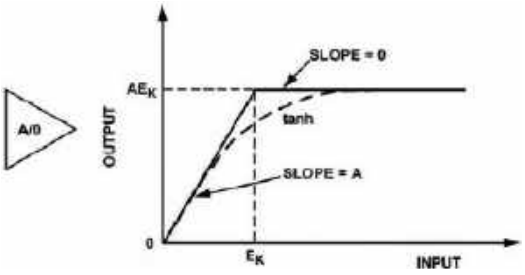
V_Y – napięciowy współczynnik nachylenia [V],

V_{IN} – napięcie wejściowe [V],

V_X – napięcie dla punktu przecięcia [V].



Rysunek 7. Wzmacniacz logarytmiczny korzystający ze stopni A_0 i pomocniczych komórek sumujących

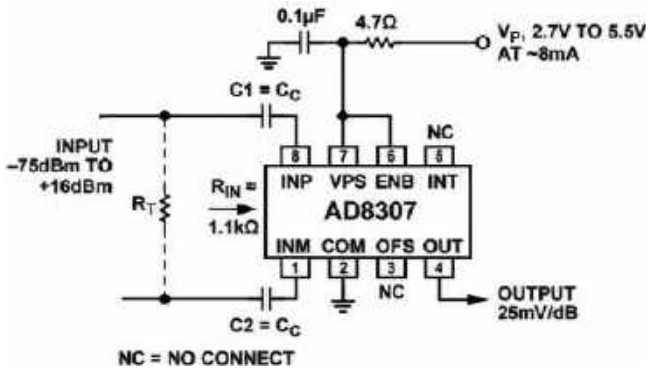


Rysunek 8. Charakterystyki przejściowe wzmacniacza A_0 (idealna oraz tanh(x))

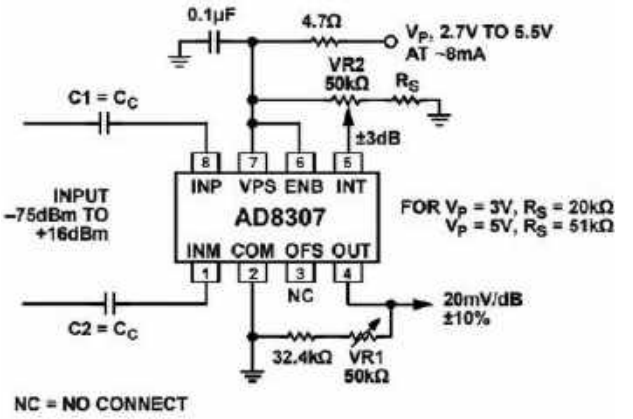
Na **rysunku 6** pokazano wykres takiej właśnie funkcji przenoszenia w skali półlogarytmicznej. Na tym samym wykresie uwidoczniono także, w jaki sposób w aplikacjach układu AD8307 możliwa jest regulacja tzw. punktu przecięcia. W praktyce realizowana jest ona za pomocą pinu INT kości (nr 5), poprzez który jest dodawany lub odejmowany niewielki prąd do lustra prądowego odpowiedzialnego za wytwarzanie napięciowego sygnału wyjściowego. Na **rysunku 7** przybliżone zostały szczegóły realizacji bloku przetwornika logarytmicznego, złożonego z połączonych szeregowo wzmacniaczy-ograniczników oraz włączonych przed, pomiędzy (jako odczepy) oraz za nimi wzmacniaczy transkonduktancyjnych, których prądy wyjściowe są sumowane. Każdy ze wzmacniaczy-ograniczników ma charakterystykę przejściową określaną skrótowo jako „A/0" (**rysunek 8**), z uwagi na teoretycznie odcinkowo-liniowy charakter przetwarzania. Jednak w rzeczywistości poszczególne wzmacniacze-ograniczniki realizowane są za pomocą tranzystorowych par różnicowych, co sprawia, że ich charakterystyki przejściowe są bliższe przebiegowi funkcji tanh(x), który – m. in. ze względu na swą gładkość – znacznie lepiej nadaje się do aproksymacji funkcji logarytmicznej w porównaniu do prostej realizacji odcinkowo-liniowej.

Na **rysunku 9** pokazano podstawową aplikację układu AD8307. Mierzony sygnał jest podawany symetrycznie na wejścia INP oraz INM poprzez jednakowe pojemności sprzęgające C1 i C2. Oprócz tych pojemności jedynymi elementami towarzyszącymi są tutaj: prosty dolnoprzepustowy filtr zasilania RC (4,7 Ω/100 nF) oraz równoległa do wejścia ustroju pomiarowego rezystancja Rt, której zadaniem jest dopasowanie impedancji wejściowej układu do przyjętych założeń pomiarowych.

eprasa.pl 4202fe84f2



Rysunek 9. Podstawowa aplikacja układu AD8307



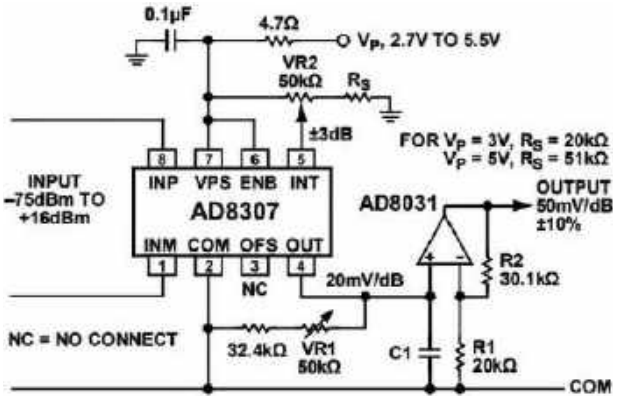
Rysunek 10. Aplikacja układu AD8307 z regulacją nachylenia i przecięcia logarytmicznej funkcji przenoszenia

W tej prostej implementacji ustrój pomiarowy jest załączony na stałe (pin ENB podpięty do wyprowadzenia zasilania VPS), natomiast wyprowadzenia pozwalające na korektę offsetu (OFS) i punktu przecięcia charakterystyki przenoszenia (INT) nie zostają użyte – nachylenie wspomnianej charakterystyki przyjmuje nominalną wartość 25 mV/dB, zatem żaden z tych trzech parametrów nie może być w tej aplikacji modyfikowany wprost.

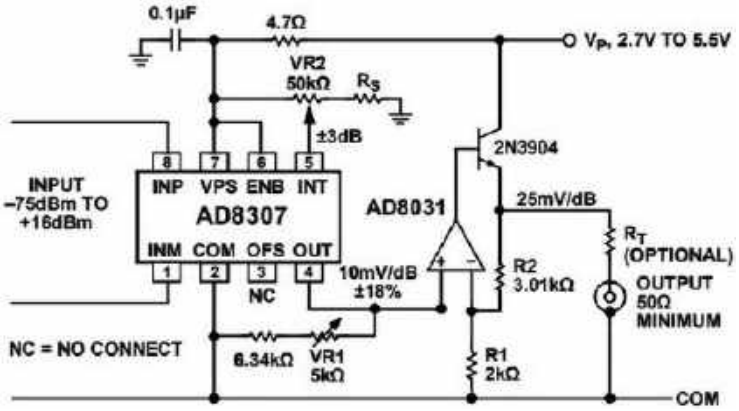
Z kolei na **rysunku 10** zaprezentowano realizację układu pomiarowego na bazie AD8307, w której możliwa jest regulacja nachylenia oraz punktu przecięcia logarytmicznej funkcji przenoszenia, przy czym pierwszy z tych parametrów może przyjmować wartości 20 mV/dB ±10%, a drugi ±3 dB względem wartości nominalnej. Uzyskanie tych dodatkowych stopni swobody w regulacji parametrów układu pomiarowego wymagało jedynie dodania dwóch potencjometrów (VR1 i VR2) oraz dwóch rezystorów włączonych z nimi w szereg do masy układu.

Rysunek 11 pokazuje aplikację układu scalonego AD8307 z rysunku 10 (czyli z możliwościami regulacyjnymi), jednak rozbudowaną o prosty filtr dolnoprzepustowy z pojemnością C1 oraz o wzmacniacz wyjściowy z układem AD8031, który zapewnia wzmocnienie (także dla przebiegów wolnozmiennych i napięć stałych) bardzo zbliżone do 2,5×. Dzięki temu nachylenie logarytmicznej funkcji przenoszenia układu pomiarowego zostało zwiększone do 50 mV/dB ±10%. Takie rozwiązanie może być użyteczne np. w przypadkach, gdy zmiany poziomu sygnału mierzonego są względnie małe, a rozdzielczość i/lub dokładność końcowego ustroju pomiarowego (np. przetwornika analogowo-cyfrowego ADC) okazuje się niewystarczająca do ich poprawnej interpretacji.

Na **rysunku 12** widzimy kolejne rozwiązanie układowe z zastosowaniem AD8307. Koncepcyjnie jest ono bardzo zbliżone do implementacji z rysunku 11.



Rysunek 11. Aplikacja układu AD8307 z dodatkowym wzmacniaczem-buforem wyjściowym



Rysunek 12. Aplikacja układu AD8307 z dodatkowym wzmacniaczem do obsługi wyjścia koncentrycznego 50 Ω

Jednak wyjście wzmacniacza końcowego (także ze wzmacniaczem operacyjnym AD8031) wysterowuje wtórnik emiterowy (z tranzystorem 2N3904), którego zadaniem jest zwiększyć wydajność prądową wyjścia całego ustroju pomiarowego i – tym samym – umożliwić zastosowanie go do współpracy z innymi, nieco oddalonymi urządzeniami poprzez połączenie kablem koncentrycznym o impedancji 50 Ω. Uzyskane w tym rozwiązaniu nachylenie logarytmicznej funkcji przenoszenia wynosi 25 mV/dB i również może być precyzyjnie regulowane w podanym na rysunku zakresie.

Na koniec tej części materiału warto odesłać wnikliwego Czytelnika do pełnej noty katalogowej układu scalonego AD8307. Podano w niej znacznie więcej zarówno szczegółów technicznych, dotyczących tego ciekawego podzespołu, jak i możliwych rozwiązań układowych, dotyczących jego implementacji. Jakkolwiek, z uwagi na ograniczony rozmiar tej publikacji, przedstawione zostały w niej wyłącznie aspekty techniczne oraz rozwiązania układowe, które należy uznać za najbardziej przydatne w kontekście przedmiotowego projektu sondy logarytmicznej.

Uzbrojeni w niezbędną wiedzę na temat układu AD8307 możemy przystąpić do opisu samego urządzenia – już za miesiąc omówimy zasadę działania, budowę układu oraz zagadnienia związane z montażem i uruchomieniem urządzenia.

Adam Sobczyk, EP

eprasa.pl 4202fe84f2

Ulubiony Kiosk

Pobierz bezpłatnie multimedialne dodatki do tego wydania Elektroniki Praktycznej

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

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

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

Szczegóły na **UlubionyKiosk.pl/media**

eprasa.pl 4202ted4f2

AVT sklep

W ofercie AVT*
AVT6047

Najważniejsze parametry:
- konstrukcja oparta na mikrokontrolerze ATmega4808
- pomiar temperatury otoczenia: czujnik DS18B20
- dokładność pomiaru temperatury: ±0,5°C
- sterowanie: enkoder obrotowy
- wyświetlacz: LCD TFT 128×160 px
- wyjścia przekaźnikowe: 2×DPDT 250 V/8 A
- wbudowany zegar czasu rzeczywistego (DS3231) z podtrzymaniem (CR2032)

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

AVT6007	Termostat do elektrozaworu z silnikiem DC (EP 10/2023)
AVT5996	Miniaturowy termostat cyfrowy (EP 8/2023)
AVT5970	Termostat MIN-MAX (EP 2/2023)
–––––	Timer z termostatem (EP 12/2022)
AVT5822	Termostat warsztatowy (EP 12/2020)
AVT5732	uniTherm – termostat z wyświetlaczem OLED (EP 12/2019)
AVT5620	Wielozadaniowy termostat (EP 1/2018)
AVT3220	Termostat z wyświetlaczem LED (EdW 1/2018)

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

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

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

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

Termostat do pieca gazowego (2)

Prezentowany w artykule projekt to uniwersalny, mikroprocesorowy termostat do pieca gazowego. W poprzedniej części opisu omówiliśmy konstrukcję i sposób montażu urządzenia. Tym razem zajmiemy się kwestiami związanymi z uruchomieniem oraz konfiguracją termostatu, przybliżymy także Czytelnikom najważniejsze zagadnienia programistyczne.

Pierwsza część artykułu znajduje się pod adresem: **https://ulubionykiosk.pl/media**

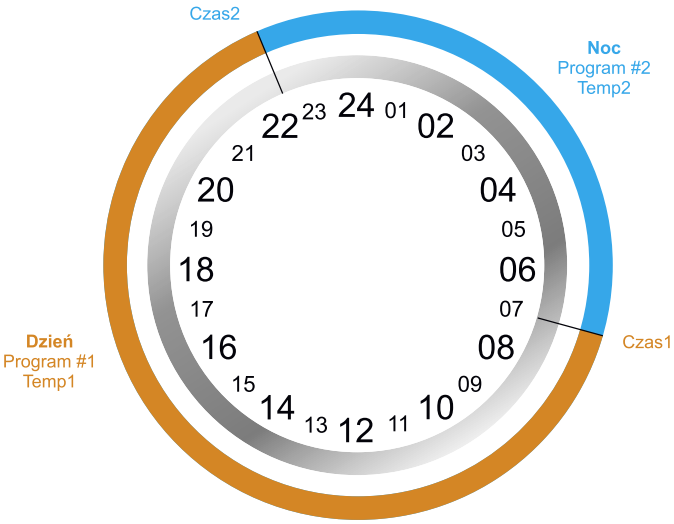
Uruchomienie i działanie układu

Uruchomienie układu zaczynamy od płytki zasilacza. Po podłączeniu sieci 230 V sprawdzamy obecność napięcia wyjściowego +5 V (wyprowadzenia 1 i 2 złącza P2) względem masy (wyprowadzenia 3 i 4 złącza P2). Jeżeli napięcie to jest prawidłowe, łączymy obie płytki złączami IDC. Napięcie na wyjściu stabilizatora U2 – umieszczonego na płytce sterownika – powinno wynosić +3,3 V. W następnym kroku do złącza J2 wpinamy programator umożliwiający zaprogramowanie mikrokontrolera ATmega4808 za pomocą jednej linii UPDI. Ja użyłem programatora MPLAB PICkit4 współpracującego ze środowiskiem projektowym MPLABX IDE. Po podłączeniu zewnętrznego czujnika DS18B20 do złącza P4 układ jest gotowy do weryfikacji działania oraz rozpoczęcia programowania ustawień.

Po włączeniu zasilania pojawia się ekran główny podzielony na dwie strefy. W pierwszej z nich wyświetla się temperatura mierzona przez czujnik DS18B20. Wynik jest aktualizowany co 1 sekundę (co wynika z czasu potrzebnego na dokonanie pomiaru). W drugiej strefie wyświetlane są natomiast: rzeczywisty czas w godzinach, minutach i sekundach, data, a także bieżący program termostatu.

Ten ostatni działa w dwóch przedziałach czasowych, umownie nazwanych: „dzień" i „noc". Do każdego z tych przedziałów przypisana została wartość temperatury. Z założenia w czasie „dzień" ustawiana jest wyższa temperatura, właściwa dla aktywności domowników (może to być na przykład +21°C). W czasie „noc" ustawiamy temperaturę niższą, na przykład +18°C. Przekłada się to na mniejsze zużycie gazu, a dodatkowo w niższej temperaturze lepiej się śpi.

Fotografia 4. Ekran główny termostatu



Rysunek 5. Zasada działania programu termostatu

eprasa.pl 4202fe84f2

Fotografia 5. Menu ustawień

Fotografia 6. Przykładowy ekran ustawień termostatu

Przedział czasowy „dzień" zaczyna się o godzinie określonej przez Czas1 i kończy o godzinie określonej przez Czas2. W tym przedziale wykonywany jest program o nazwie „Program #1" z przypisaną temperaturą regulacji Temp1.

Przedział czasowy „noc" zaczyna się o godzinie Czas2, a kończy o godzinie Czas1. W tym przedziale z kolei wykonywany jest czas regulacji „Program #2" z przypisaną Temp2. Wszystkie parametry programu: Czas1, Czas2, Temp1 i Temp2, są programowane. Jedyne ograniczenie bieżącej wersji oprogramowania stanowi warunek, że Czas1 musi być mniejszy od Czas2. Na przykład Czas1=07.00, Czas2=22.15, ale Czas2 nie powinien mieć na przykład wartości 0.30. Wtedy program regulacji będzie działał, ale po zaniku zasilania może błędnie określić, w którym przedziale czasowym się znajduje, aż do momentu osiągnięcia jednego z czasów: Czas1 lub Czas2. Na **rysunku 5** pokazano zasadę czasowego podziału doby na dwa programy Program #1 i Program #2. W czasie wykonywania Programu #1 termostat utrzymuje temperaturę Temp1, a w czasie trwania Programu #2 – temperaturę Temp2.

Temperatura jest regulowana przy zastosowaniu pętli histerezy. Załóżmy, że temperatura otoczenia okazuje się wyższa od temperatury Temp1. Termostat wyłącza wówczas grzanie pieca. Temperatura otoczenia spada, osiąga wartość Temp1. Termostat nadal nie włączy pieca – tak długo, aż temperatura otoczenia spadnie do wartości Temp1 – Hist, gdzie Hist jest wartością programowaną w zakresie od 0,1°C do 2°C z krokiem 0,1°C. Jeżeli temperatura otoczenia jest niższa od Temp1 – Hist, to termostat włącza grzanie. Temperatura w pomieszczeniu rośnie, aż osiągnie wartość wyższą od Temp1. Wtedy termostat wyłącza grzanie i cykl się powtarza. Takie rozwiązanie zapobiega częstym cyklom załącz-wyłącz, kiedy temperatura jest bliska wartości Temp1. Temperatura w pomieszczeniu będzie się w przybliżeniu wahać od wartości Temp1-Hist do Temp1. Wartość histerezy należy dobrać w zależności od warunków w ogrzewanych pomieszczeniach. Zbyt duża powoduje duże wahania temperatury, zbyt mała natomiast – częste cykle załącz-wyłącz. Prototyp termostatu pracował z histerezą 0,8°C. Histereza pozostaje wspólna dla obu programów.

Menu programowania jest dostępne po przyciśnięciu osi impulsatora. Można w nim ustawić czas, datę i termostat – **fotografia 5**.

Wybór jednej z funkcji w menu ustawień (czas, data, termostat) realizowany jest przez obrót gałki. Wybrana funkcja wyświetlana się na biało. Przyciśnięcie osi impulsatora sprawia, że rozpoczyna się wykonywanie wybranej funkcji.

Najbardziej rozbudowana jest funkcja ustawień termostatu. Można w niej zaprogramować:

- histerezę w zakresie od 0,1°C do 2,0°C z krokiem 0,1°C,
- temperaturę Temp1 w zakresie od 0°C do 31°C z krokiem 0,1°C,
- temperaturę Temp2 w zakresie od 0°C do 31°C z krokiem 0,1°C,
- czas 1 (godziny, minuty),
- czas 2 (godziny, minuty).

Nastawy zmienia się przez obrót osi impulsatora, a zatwierdza jej przyciśnięciem. Ustawiana wartość wyświetla się na biało, a po zatwierdzeniu zmienia kolor na niebieski – **fotografia 6.**

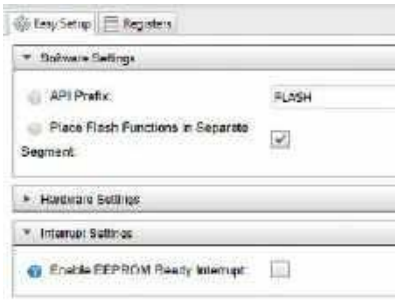
Po wejściu do menu ustawień termostatu użytkownik nie ma możliwości wybrania tylko jednej z nastaw. Musi „przeklikać" wszystkie pozycje. Sterownik pamięta każdą nastawę i jeżeli jakiejś nie chcemy modyfikować, powinniśmy sekwencyjnie przyciskać oś impulsatora, żeby przejść do kolejnej nastawy bez wprowadzania zmian. To uproszczenie nie wpływa znacząco na komfort użytkowania, ponieważ programowania termostatu nie wykonuje się często. Wszystkie nastawy są zachowywane w pamięci nieulotnej mikrokontrolera i odtwarzane po każdym zaniku napięcia zasilania. Wraz z podtrzymaniem bateryjnym zegara RTC zapewnia to w miarę niezakłócony cykl sterowania przy chwilowych zanikach napięcia sieciowego. Sterownik – po ponownym włączeniu zasilania – na podstawie ustawień Czas1 i Czas2 oraz bieżącego wskazania zegara oblicza, czy aktywny jest Program #1, czy Program #2 i zaczyna regulować nastawione temperatury właściwe aktywnemu programowi.

Ustawienia czasu oraz daty nie wymagają szerszego komentarza. W trybie ustawiania czasu programujemy kolejno: godziny i minuty. Po ustawieniu minut oraz przyciśnięciu osi impulsatora zerowany jest licznik sekund – i wartości godzin, minut i sekund są wpisywane do rejestrów układu DS3231, a zegar zaczyna odliczać nowy czas. W funkcji ustawienia daty programowane są: dzień miesiąca, miesiąc, a także rok.

Jak już wspomniałem, oprogramowanie zostało napisane w środowisku MPLAB firmy Microchip. Oprócz oczywistych elementów – takich jak środowisko IDE i kompilator C – istotnym wsparciem dla programisty jest wtyczka konfiguracyjna MCC. Oprócz konfiguracji urządzeń peryferyjnych dostarcza kody źródłowe procedur do ich obsługi, co znacznie ułatwia i skraca pisanie programu. Trzeba przy tym pamiętać, że większość procedur komunikacji sprawdza warunek zakończenia transmisji w nieskończonych pętlach (procedury blokujące). Warto je nieco zmodyfikować, tak by – po określonym czasie oczekiwania w pętli na spełnienie warunku – tę pętlę przerwać. Jeżeli tego nie zrobimy, ryzykujemy, że program wcześniej czy później trwale się zawiesi. Można również pozostawić go tak jak jest – i skonfigurować moduł watchdoga.

eprasa.pl 4202fe84f2

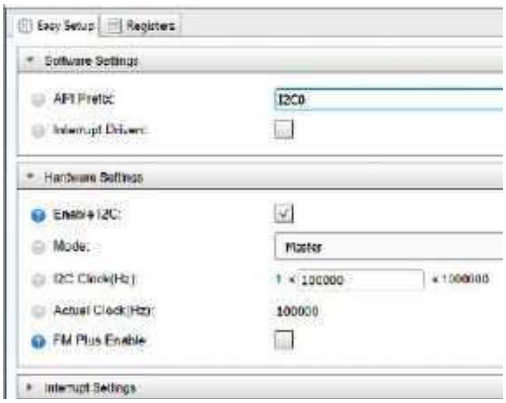
Rysunek 6. Konfiguracja układu taktowania



Rysunek 7. Moduł NVMCTRL

Mikrokontroler jest taktowany wewnętrznym oscylatorem o częstotliwości 20 MHz. Przy zasilaniu +3,3 V nominalna częstotliwość taktowania powinna być ograniczona do 10 MHz, ale w praktyce – w temperaturach pokojowych – mikrokontroler pracuje poprawnie przy 20 MHz (okazuje się to istotne dla w miarę płynnej pracy wyświetlacza). Układ taktowania jest konfigurowany w oknie Clock Control wtyczki konfiguratora MCC – **rysunek 6**.

Mikrokontroler ma wbudowane 256 bajtów nieulotnej pamięci EEPROM przeznaczonej do przechowywania danych systemowych. Dostęp do pamięci nieulotnej – w tym EEPROM (ale też i Flash) – zapewnia specjalny moduł NVMCTRL (Nonvolatile Memory Controller). Po zaznaczeniu *Place Flash Functions in Separate* w oknie NVMCTRL (**rysunek 7**) wtyczka MCC wygeneruje funkcje inicjalizujące moduł NVMCTRL oraz funkcje zapisu i odczytu bajtu z pamięci EEPROM (funkcje te używane są do zapisywania i odczytywania ustawień termostatu – **listing 1**).



Rysunek 8. Konfiguracja TWI (I²C master)

Zegar RTC DS3231 komunikuje się z mikrokontrolerem za pomocą magistrali I²C. Transmisję obsługuje sprzętowy moduł TWI konfigurowany w MCC – **rysunek 8**. Pracuje on jako master I²C z zegarem o częstotliwości 100 kHz.

W przypadku interfejsu TWI, MCC dostarcza plik twi0_master_example.c z działającymi funkcjami transferu danych po magistrali I²C. Na **listingu 2** pokazano funkcję przesyłania danych do zegara RTC DS3231, korzystającą z „przykładowej" funkcji I2C0_example_write1ByteRegister.

Mikrokontroler przesyła dane do sterownika wyświetlacza poprzez magistralę SPI. Wyświetlacz ma rozmiar 160×128 pikseli – żeby określić jasność i kolor każdego z pikseli, mikrokontroler musi przesłać do sterownika 2 bajty. Zapisanie całego wyświetlacza wymaga zatem transferu 128×160×2=40960 bajtów. Ponadto trzeba często wysłać sekwencje komend sterujących, na przykład do ustalenia adresu piksela. Wszystko to sprawia, że transfer danych powinien być szybki, co z kolei wymaga wydajnego mikrokontrolera oraz – oczywiście – szybkiego interfejsu SPI. W naszym przypadku zegar taktujący przesyłaniem bitów ma częstotliwość 5 MHz przy

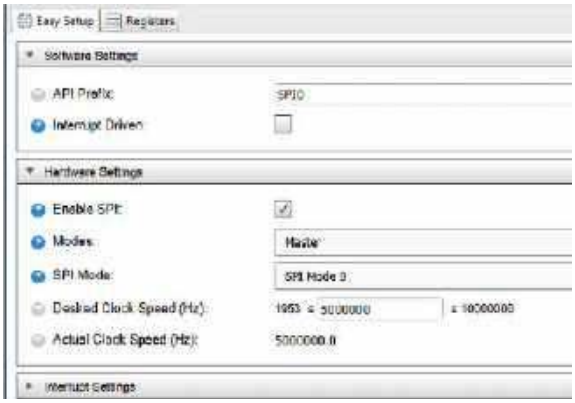
```
/**************************************************
zapisanie bajtu do pamięci EEPROM
**************************************************/
uint8_t HMI_WriteEEPROM(uint8_t address,uint8_t data)
{
    uint8_t status;
    status = FLASH_WriteEepromByte(address, data);
    return status;
}
/**************************************************
zapisanie bajtu do pamięci EEPROM
-procedura wygenerowana przez MMC
**************************************************/
/**
* \brief Write a byte to eeprom
*
* \param[in] eeprom_adr The byte-address in eeprom to write to
* \param[in] data The byte to write
*
* \return Status of write operation
*/
nvmctrl_status_t FLASH_WriteEepromByte(eeprom_adr_t eeprom_adr, uint8_t data)
{
/* Wait for completion of previous write */
while (NVMCTRL.STATUS & NVMCTRL_EEBUSY_bm)
;

/* Clear page buffer */
ccp_write_spm((void *)&NVMCTRL.CTRLA, NVMCTRL_CMD_PAGEBUFCLR_gc);

/* Write byte to page buffer */
*(uint8_t *)(EEPROM_START + eeprom_adr) = data;

/* Erase byte and program it with desired value */
ccp_write_spm((void *)&NVMCTRL.CTRLA, NVMCTRL_CMD_PAGEERASEWRITE_gc);

        if (NVMCTRL.STATUS & NVMCTRL_WRERROR_bm)
        {
            return NVM_ERROR;
        }
        else
        {
            return NVM_OK;
        }
}
```

Listing 1. Zapisanie danej do pamięci EEPROM

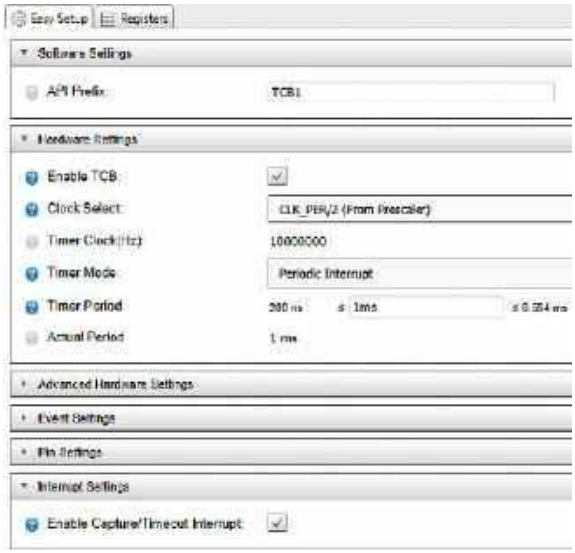
```
**************************************************
zapisz danej do rejestru o adresie reg
**************************************************/
void RTC_SendData(uint8_t reg, uint8_t data)
{
    I2C0_example_write1ByteRegister(RTC_I2C_ADD, reg, data);
}
void I2C0_example_write1ByteRegister(twi0_address_t address, uint8_t reg, uint8_t data)
{
    while(!I2C0_Open(address));                                          //sit here until we get the bus.
    I2C0_SetDataCompleteCallback(wr1RegCompleteHandler_example,&data);
    I2C0_SetBuffer(&reg,1);
    I2C0_SetAddressNackCallback(I2C0_SetRestartWriteCallback,NULL);      //NACK polling?
    I2C0_MasterWrite();
    while(I2C0_BUSY == I2C0_Close());                                    //sit here until finished.
}
```

Listing 2. Funkcja przesłania danej do zegara RTC

eprasa.pl 4202fe84f2



Rysunek 9. Konfiguracja interfejsu SPI



Rysunek 10. Konfiguracja Timera TCB

taktowaniu mikrokontrolera częstotliwością 20 MHz. Konfiguracja interfejsu SPI została pokazana na **rysunku 9**.

Procedury przesłania danych i komend wymagają również ustawienia linii CD wyboru interfejsu SPI oraz linii wyboru rodzaju przesyłanych danych DC (data/command). Na **listingu 3** pokazano procedurę przesyłającą 2 bajty dotyczące jednego piksela do pamięci obrazu wyświetlacza. MMC wygenerowała procedurę wysyłania jednego bajtu przez interfejs SPI – **listing 4**.

Przy dużej ilości przesyłanych danych zdarza się, że procedura oczekiwania na wysłanie bajtów ulega zawieszeniu, choć teoretycznie przy zastosowaniu SPI nie powinna. Drobna modyfikacja powoduje, że problem ten praktycznie znika. W pętli oczekiwania inkrementowany jest licznik. Jeżeli nie zostanie sprzętowo ustawiona flaga SPI_RXCIF oznaczająca, że bajt został wysłany (i odebrany), to wykonywana jest ponowna inicjalizacja modułu SPI SPI0_Initialize () i pętla kończy działanie – **listing 5**.

Ostatni moduł sprzętowy używany przez program to timer TCB (**rysunek 10**). Został on skonfigurowany tak, by zgłaszać cykliczne przerwania co jedną milisekundę. W procedurze obsługi tego przerwania jest realizowana obsługa impulsatora.

```
static void LCD_WriteData_NLen16Bit(uint16_t Data,uint32_t DataLen)
{
uint32_t i;
PORTC.OUT |= (1 << DC);                          //zapis danej
    PORTC.OUT &= ~(1 << CS);                     //SPI aktywny
for(i = 0; i < DataLen; i++){
SPI0_ExchangeByte( (uint8_t)(Data >> 8) );
SPI0_ExchangeByte( (uint8_t)(Data & 0XFF) );
}
PORTC.OUT |= (1 << CS);                          //SPI nieaktywny
}
```

Listing 3. Przesłanie dwu bajtów do pamięci wyświetlacza

```
uint8_t SPI0_ExchangeByte(uint8_t data)
{
    SPI0.DATA = data;
    while (!(SPI0.INTFLAGS & SPI_RXCIF_bm));
    return SPI0.DATA;
}
```

Listing 4. Funkcja wysyłająca jeden bajt przez SPI

```
#define SPI_FRAME 0xD00
uint8_t SPI0_ExchangeByte(uint8_t data)
{
uint16_t frame = 0;
    SPI0.DATA = data;
    while (1)
   {
     if!(SPI0.INTFLAGS & SPI_RXCIF_bm)
return SPI0.DATA;                                //dana prawidłowo wysłana

    ++ frame;
    if (frame >= SPI_FRAME)
{
SPI SPI0_Initialize ();
return(0);
    }
    }
    return SPI0.DATA;
}
```

Listing 5. Zmodyfikowana procedura wysyłania bajtu

```
void Termostat(double temperature)
{
double hist, temp1, temp2;
      HMI_GetTermostat();                        //pobierz dane ustawień
                                                 //termostatu
   hist = (double)termo.hist/100;
   temp1 = (double)termo.temp1/10;
   temp2 = (double)termo.temp2/10;


   Termostat_Check_Time ();                      //sprawdź, czy nie zmienił
                                                 //się program
    Termostat_Check_Temperature (temperature);
}
```

Listing 6. Funkcja termostatu

Główne procedury regulacji nie są zbyt skomplikowane. Po każdym pomiarze temperatury – wykonywanym co sekundę – wywołaniu ulega procedura Termostat, której argumentem jest temperatura w formacie zmiennoprzecinkowym – **listing 6**.

Funkcja HMI_GetTermostat() odczytuje z pamięci EEPROM wszystkie ustawienia

```
//odczytanie czasu i sprawdzenie, który program mam być aktywny
//przy zmianie progów czasowych

void Termostat_Check_Time (void)
{
    uint8_t min, hour;
    RTC_GetTime ();                                  //pobierz czas z rejestrów DS3231
    hour = HMI_ConvHexDec (time.hour);
    min =  HMI_ConvHexDec (time.min);
    if (hour == termo.t1_hour && min == termo.t1_min)
    {
        termo.program = PR1;
        return ;
    }
    else
    if (hour >termo.t1_hour && hour < termo.t2_hour)
    {
        termo.program = PR1;
        return ;
    }
    else
    if (hour == termo.t2_hour && min < termo.t2_min)
    {
        termo.program = PR1;
        return ;
    termo.program = PR2
}
```

Listing 7. Identyfikacja aktywnego programu czasowego

eprasa.pl 4202fe84f2

```
void Termostat_Check_Temperature (double temperature)
{
   if(termo.program == PR1)
   {
       if(temperature < (temp1 - hist))
       {
           Heater = 1; //wlacz grzanie
           if(termo.dpr1p)
           {
               LCD_DisplayString(20,70,"Prog#1",&Font20,COL6,WHITE);
               LCD_DisplayString(108,70,"ON ",&Font20,COL6,BLACK);
               termo.dpr1p = 0;
               termo.dpr1np = 1;
           }

       }
       if(temperature > temp1)
       {
           Heater = 0;
       if(termo.dpr1np)
           {
               LCD_DisplayString(20,70,"Prog#1",&Font20,COL6,WHITE);
               LCD_DisplayString(108,70,"OFF",&Font20,COL6,GREEN);
               termo.dpr1np = 0;
               termo.dpr1p = 1;
           }
       }
   }
   if (termo.program == PR2)
   {
        if(temperature < (temp2 - hist))
        {
           Heater = 1; //wlacz grzanie
         if(termo.dpr2p)
           {
               LCD_DisplayString(20,70,"Prog#2",&Font20,COL6,WHITE);
               LCD_DisplayString(108,70,"ON ",&Font20,COL6,BLACK);
               termo.dpr2p = 0;
               termo.dpr2np = 1;
           }
        }
       if(temperature > temp2)
       {
           Heater = 0;
        if(termo.dpr2np)
           {
               LCD_DisplayString(20,70,"Prog#2",&Font20,COL6,WHITE);
               LCD_DisplayString(108,70,"OFF",&Font20,COL6,GREEN);
               termo.dpr2np = 0;
               termo.dpr2p = 1;
           }
       }

   }

}
```

Listing 8. Funkcja termostatu

termostatu: histerezę, temperatury programów temp1 i temp2 oraz oba czasy. Następnie wywoływana jest funkcja Termostat_Check_Time () – **listing 7**. Ma ona za zadanie porównać bieżący czas (odczytany z rejestrów zegara RTC) z nastawami Czas1 i Czas2, a na podstawie wyniku tego porównania – określić, który program czasowy ma się wykonywać.

Na podstawie wartości zmiennej termo.program funkcja Termostat_Check_Temperature (temperature) identyfikuje, która z temperatur ma być użyta w procesie regulacji. Procedura ta została pokazana na **listingu 8**. Opisana funkcja załącza lub wyłącza przekaźnik sterujący piecem oraz wyświetla na ekranie głównym informację o aktywnym programie, a także wskazuje, czy grzanie jest włączone, czy wyłączone.

Program termostatu zajmuje 69% z 48 kB dostępnej pamięci Flash, zaś zdecydowana większość tej przestrzeni przypada na procedury związane z obsługą graficznego wyświetlacza, w tym także wzorce znaków alfanumerycznych. Również podczas pisania programu oprogramowanie interfejsu użytkownika HMI okazało się najbardziej pracochłonne. Można zadać sobie pytanie, czy warto poświęcać tyle czasu na bardziej lub mniej atrakcyjny interfejs użytkownika. Skoro jednak mamy dzisiaj możliwości używania tanich, dobrej jakości wyświetlaczy, wydajnych małych mikrokontrolerów czy dobrych bezpłatnych narzędzi programistycznych, szkoda byłoby tego nie wykorzystać.

Tomasz Jabłoński, EP

REKLAMA

eprasa.pl 4202fe84f2

AVT sklep

W ofercie AVT*

AVT6051

Najważniejsze parametry:
- konwersja nieujemnego napięcia stałego w przedziale 0...5 V na napięcie stałe 0...10 V,
- rezystancja wejściowa 100 kΩ, wyjściowa ok. 50 Ω,
- zasilanie napięciem stałym (standardowo 15 V, maksymalnie 32 V),
- pobór prądu do 10 mA.

Dodatkowe materiały do pobrania ze strony www.ulubionykiosk.pl/media
AVT6041 Konwerter napięcia stałego na pętlę prądową 4...20 mA (EP 5/2024)
AVT3284 Precyzyjne źródło prądowe 0...25 mA (EdW 7/2020)

* **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**.

Podwójny konwerter napięcia z 0...5 V na 0...10 V

Pomimo wszechobecnej cyfryzacji, w systemach przemysłowych wciąż są obecne analogowe metody transmisji. Jedną z nich jest standard 0...5 V, stosowany również w urządzeniach z mikrokontrolerami. Co zrobić w sytuacji, gdy wejście naszego sterownika używa typowego standardu 0...10 V? Zastosować niniejszy moduł!

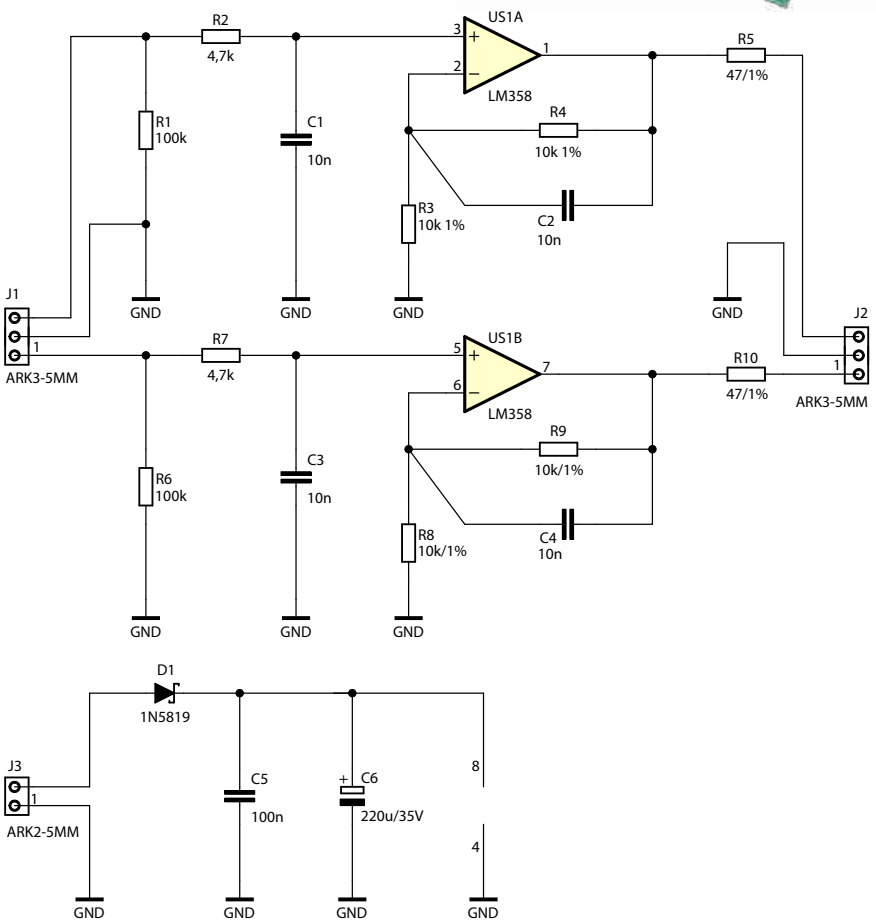
Analogowe sposoby transmisji w systemach przemysłowych nie trafiły do lamusa. Nadal mają zastosowanie tam, gdzie liczy się szybkość, zaś niekoniecznie duża dokładność. W przypadku pętli prądowej do zalet dołącza również odporność na zakłócenia. Łatwo o zamienniki czujników – nie trzeba przejmować się ich adresowaniem, wystarczy zachować zgodność poziomów napięć. Ano właśnie...

Powszechniejszy standard obejmuje zakres napięć z przedziału 0...10 V. Jednak w handlu jest dostępna również całkiem spora gama urządzeń z wyjściami pracującymi w standardzie 0...5 V. Jeżeli dotychczasowy system używał czujnika 0...10 V, zaś teraz trzeba (z różnych względów) w jego miejsce zastosować taki, który pracuje w zakresie 0...5 V, to rozwiązania są dwa: przeprogramować PLC i pogodzić się z pracą w połowie dopuszczalnego zakresu – lub między wyjście nowego czujnika a wejście PLC włączyć ten prosty układ. I gotowe, bez jakichkolwiek przeróbek w programie!

Budowa

Schemat ideowy omawianego układu znajduje się na **rysunku 1**. Zawiera dwa wzmacniacze operacyjne zintegrowane w jednym, podwójnym układzie scalonym US1A. Owe wzmacniacze operacyjne skonfigurowane są do pracy w topologii nieodwracającej o wzmocnieniu 2 V/V, co wynika z zależności:

$$k=1+(R4/R3)=1+(R9/R8)$$
$$k=1+(10\ k\Omega/10\ k\Omega)$$
$$k=1+1=2\ [V/V]$$



Rysunek 1. Schemat ideowy układu konwertera napięć

Wykaz elementów:

Rezystory: (THT o mocy 0,25 W, jeżeli nie napisano inaczej)
R1, R6: 100 kΩ
R2, R7: 4,7 kΩ
R3, R4, R8, R9: 10 kΩ 1% 0,6 W
R5, R10: 47 Ω 1% 0,6 W

Kondensatory:
C1...C4: 10 nF raster 5 mm MKT
C5: 100 nF raster 5 mm MKT
C6: 220 µF 35 V raster 3,5 mm

Półprzewodniki:
D1: 1N5819
US1: LM358 DIP8

Pozostałe:
J1, J2: ARK3/500
J3: ARK2/500
Jedna podstawka DIP8

eprasa.pl 4202fe84f2

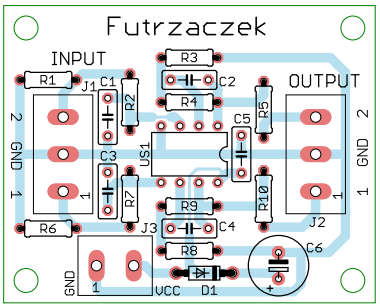
Kondensatory C2 i C4 ograniczają pasmo przenoszenia tak skonstruowanych wzmacniaczy, tworząc filtry dolnoprzepustowe o teoretycznej częstotliwości odcięcia wynoszącej ok. 1,6 kHz, co istotnie zawęża pasmo szumowe. Taka szerokość pasma przenoszenia, obejmująca swym przedziałem również składową stałą, okaże się wystarczająca w zdecydowanej większości przypadków.

Wejścia układu są polaryzowane przez rezystory R1 i R6 o rezystancji 100 kΩ, co jest zgodne z rezystancją wejściową większości spotykanych na rynku sterowników PLC. Rolą rezystorów R2 i R7 jest skompensowanie wpływu wejściowych prądów polaryzujących bazy tranzystorów wejściowych wzmacniaczy operacyjnych. Ich rezystancja pozostaje nieco niższa niż wypadkowa rezystancja polaryzująca wejścia odwracające wzmacniaczy operacyjnych (która wynosi 5 kΩ – jest równa połączeniu równoległemu rezystorów R3 i R4 bądź R8 i R9). Jednak szeregowo z rezystorami R2 i R7 należy uwzględnić rezystancję przewodów łączących wyjście czujnika z płytką oraz rezystancję wyjściową samego czujnika. Dlatego – po złożeniu tych wszystkich trzech rezystancji – ostateczny wynik będzie bliski właśnie wartości 5 kΩ. Skompensowanie wpływu przytoczonych wcześniej prądów jest niezbędne w układzie, który ma prawidłowo przenosić składową stałą, bowiem brak staranności w tym zakresie wywoła offset napięciowy na wyjściu opisywanego wzmacniacza, na dodatek o trudnej do przewidzenia wartości. Kondensatory C1 i C3 tworzą wraz z rezystancją wejściową obu torów dodatkowe filtry dolnoprzepustowe, ograniczając wypadkowe pasmo przenoszenia całego układu do zaledwie 700 Hz.

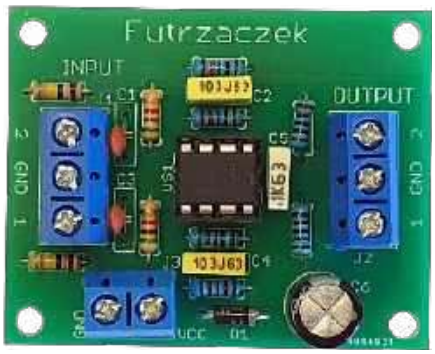
Układ typu LM358 wykonany został w oparciu na tranzystorach bipolarnych, nie ma zatem konieczności szczególnej ochrony jego wejść przed wyładowaniami elektrostatycznymi – złącza tranzystorów wejściowych mogą tę funkcję pełnić bez uszczerbku dla swojego zdrowia. Przy okazji rezystory R2 i R7 ograniczają ich prąd w razie podania na wejście zbyt wysokiego napięcia wejściowego, zaś kondensatory C1 i C3 wydłużają okres narastania napięcia, dając tymże złączom czas na otwarcie się.

Od strony wyjść wzmacniaczy nieodwracających układ zawiera włączone szeregowo rezystory o wartości 47 Ω każdy. Po pierwsze pełnią one funkcję dopasowania impedancyjnego przy podłączeniu do wyjść długich przewodów ekranowanych. Po drugie – co równie ważne, jeżeli nie ważniejsze w tym relatywnie wolnym układzie – zapobiegają wzbudzaniu się wzmacniaczy operacyjnych przy obciążeniu ich wyjść pojemnością wynikającą chociażby z użycia kabli ekranowanych. Tak czy siak, warto je w tym miejscu pozostawić oraz uwzględnić ich wpływ, jeżeli następny stopień układu ma zauważalnie małą impedancję wejściową.

Użyte w prezentowanym układzie wzmacniacze operacyjne to układy tanie i popularne, a do tego zapewniające prawidłową pracę już od 0 V napięcia na wejściach. Dlatego w projekcie nie były konieczne dodatkowe zabiegi, takie jak uwzględnianie pompy ładunkowej wytwarzającej napięcie ujemne dla obwodów wejściowych tychże wzmacniaczy. Nie są to jednak układy rail-to-rail i trzeba mieć na uwadze, że od dodatniej linii zasilającej wymagają kilkuwoltowego marginesu, zarówno na swoim wyjściu, jak i na swoich wejściach.



Rysunek 2. Schemat montażowy i wzór ścieżek płytki



Fotografia 1. Widok zmontowanego układu

Montaż i uruchomienie

Układ został zmontowany na jednostronnej płytce drukowanej o wymiarach 50 mm × 40 mm. Wzór jej 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 diody. Pod układ scalony US1 proponuję zastosować podstawkę, aby ułatwić jego wymianę w razie ewentualnego uszkodzenia. W pełni zmontowany układ można zobaczyć na **fotografii 1**.

Poprawnie złożone urządzenie jest gotowe do działania po podaniu zasilania na zaciski GND i VCC złącza J3. Do zasilania powinno służyć napięcie stałe o wartości nie mniejszej niż 15 V – z uwagi na spadek napięcia na diodzie D1 oraz stopniu wyjściowym wzmacniacza operacyjnego. Nic nie stoi jednak na przeszkodzie, aby owo napięcie było dużo wyższe, nawet do 32 V. W systemach ze sterownikami PLC typowo spotyka się napięcie 24 V, które idealnie nada się do zasilania tegoż układu.

Napięcie z przedziału 0...5 V podane na zaciski oznaczone jako 1 lub 2 złącza J1 (INPUT) zostanie wzmocnione i przekazane na analogicznie opisane zaciski złącza J2 (OUTPUT). Pomiary przeprowadzone na układzie prototypowym wykazały, że – przy zaciskach wejściowych pozostawionych bez jakiegokolwiek podłączenia – na wyjściach panuje napięcie rzędu 6...8 mV, co ma związek zarówno z offsetem napięciowym wzmacniaczy operacyjnych, jak i brakiem opisanej wcześniej kompensacji wejściowych prądów tychże wzmacniaczy. Po zwarciu wejścia z masą, czyli po zasymulowaniu źródła napięcia o zerowej rezystancji wewnętrznej, wartość ta spadła do 2...3 mV i ma niewątpliwie związek z offsetem napięciowym, na który nie mamy wpływu. Dlatego trzeba przyjąć, że właśnie takiego rzędu błąd wprowadza ten układ, co przy zakresie napięcia wyjściowego 0...10 V stanowi i tak bardzo dobry wynik w zdecydowanej większości zastosowań.

Prąd pobierany przez opisane wyżej urządzenie nie przekracza wartości 10 mA, pod warunkiem braku obciążenia jego wyjść. Jeżeli takowe się pojawi, trzeba ów pobór uwzględnić w bilansie energetycznym. Nieużywane wejście układu można zostawić niepodłączone, gdyż rezystor wejściowy zapewni dostateczną polaryzację. Układ jest podatny na modyfikacje, można w szerokim zakresie zmieniać zarówno jego rezystancję wejściową, jak i pasmo przenoszenia czy wzmocnienie, stosując proste wzory znane z teorii obwodów.

Michał Kurzela, EP

REKLAMA

http://ulubionykiosk.pl

eprasa.pl 4202fe84f2

AVT sklep

W ofercie AVT*

AVT6052

Najważniejsze parametry:
- konstrukcja oparta na scalonym transceiverze CAN typu SN65HVD230,
- wbudowana ochrona przeciwprzepięciowa i przeciwzwarciowa,
- wbudowane terminatory magistrali CAN,
- wybór szybkości interfejsu za pomocą zworki,
- kompatybilność z systemami zasilanymi napięciem od 2,7 V do 5 V,
- wbudowana przetwornica DC/DC typu buck-boost oraz translatory poziomów.

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

AVT6039	Nakładka z transceiverem CAN do AVTduino UNO R4 Plus (EP 5/2024)
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)
AVT5738	Wyświetlacz 7-segmentowy dla Arduino MKR Zero (EP 1/2020)
----	Sterownik silników prądu stałego dla Arduino MKR (EP 3/2019)

* **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**.

Minimoduł z transceiverem CAN

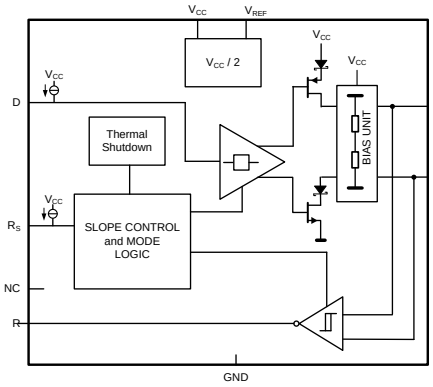
Niewielki moduł transceivera magistrali CAN szczególnie przydatny w rozmaitych układach sterowania, m.in. w automatyce budynkowej czy elektronice samochodowej. Okaże się szczególnie użyteczny, gdy chcemy poeksperymentować z Arduino UNO R4 lub innymi płytkami wyposażonymi w procesory z wbudowanym kontrolerem CAN – a pozbawionymi układu transceivera (drivera) magistrali.

Moduł korzysta z układu transceivera CAN typu SN65HVD230, którego budowę pokazano na **rysunku 1**.

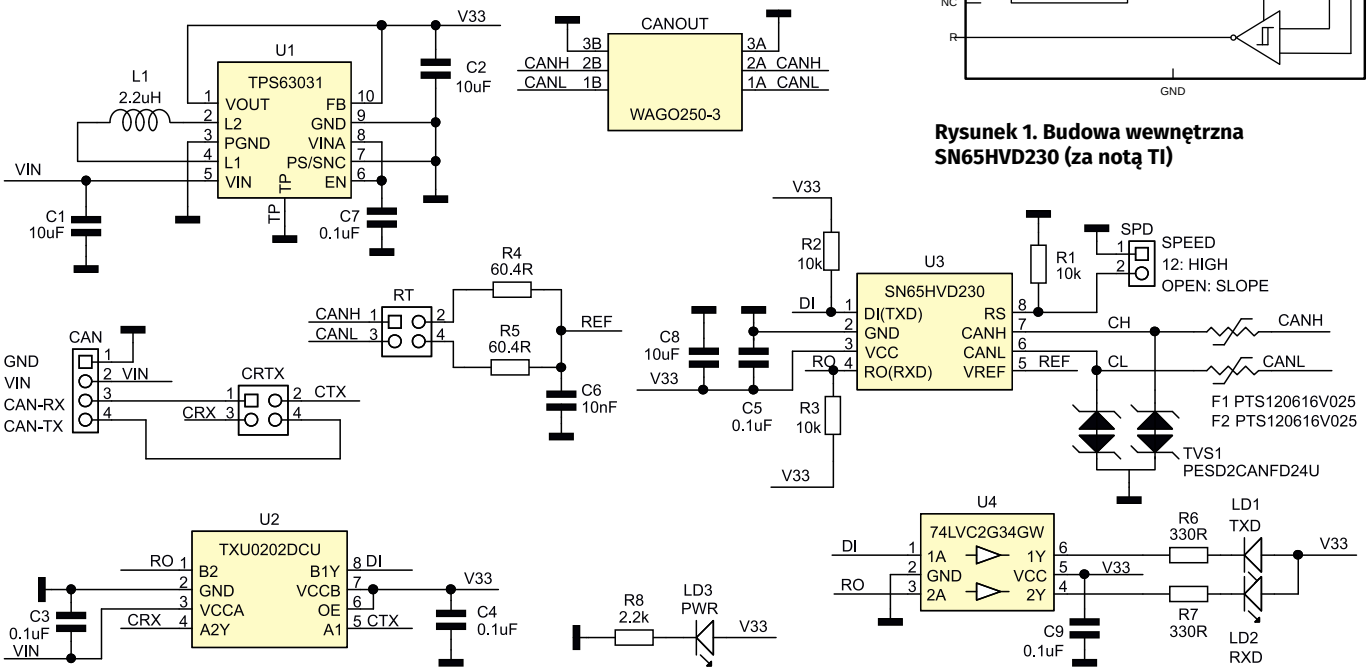
Układ zawiera wszystko, co jest konieczne do spełnienia wymogów standardów CAN zgodnie z normą ISO11898-2, a jego aplikacja ogranicza się do zaledwie kilku elementów zewnętrznych.

Schemat nakładki zaprezentowano na **rysunku 2**. Aplikację SN65HVD230 uzupełniają elementy zabezpieczające driver U3 oraz terminujące magistralę CAN. Dioda TVS1 chroni transceiver przed skutkami wyładowań ESD i przepięć, a bezpieczniki polimerowe F1, F2 – przed skutkami zwarć linii magistrali. Przełącznik RT umożliwia odłączenie rezystorów terminujących R4, R5 w aplikacjach, w których nakładka nie jest „skrajnym" urządzeniem magistrali CAN. W układzie zastosowano schemat terminacji dzielonej z użyciem elementów R4, R5 i C6, korzystający z potencjału VREF=VCC/2 układu U3 w celu stabilizacji napięcia trybu wspólnego (CM). Magistrala CAN wyprowadzona została z transceivera na złącze sprężynowe CANOUT typu WAGO250.

Zasilanie i sygnały sterujące CAN-RT/TX doprowadzone zostały do złącza CAN. Aby zapewnić uniwersalność modułu i umożliwić współpracę z systemami 2,7...5 V, płytka uzupełniona została o przetwornicę DC/DC typu buck-boost na bazie układu U1 typu TPS63031, dostarczającą napięcia



Rysunek 1. Budowa wewnętrzna SN65HVD230 (za notą TI)



Rysunek 2. Schemat nakładki

eprasa.pl 4202fe84f2

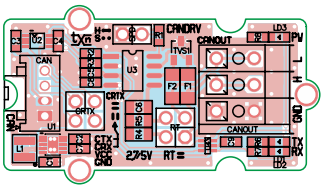
Wykaz elementów:

Półprzewodniki:
LD1: dioda LED czerwona (SMD 0603)
LD2: dioda LED żółta (SMD 0603)
LD1: dioda LED zielona (SMD 0603)
TVS1: dioda zabezpieczająca PESD-2CANFD24U (SOT-23)
U1: TPS63031DSK (PWSON10)
U2: TXU0202DCU (VSSOP8_050)
U3: SN65HVD230 (SO8)
U4: 74LVC2G34GW.125 (SC88)

Rezystory:
R1...R3: 10 kΩ (SMD 0603, 1%)
R4, R5: 60,4 Ω (SMD 0805, 1%, typ ERJP06F60R4V)
R6, R7: 330 Ω (SMD 0603, 1%)
R8: 2,2 kΩ (SMD 0603, 1%)

Pozostałe:
L1: 2,2 μH (typ DJNR3015-2R2)
CAN: złącze Grove proste
CANOUT: złącze sprężynowe Wago (WAGO250-3)
CRTX, RT: złącze IDC4 + zwory 2 szt.
F1, F2: bezpiecznik polimerowy 16 V 250 mA (SMD 1206, typ PTS120616V025)
SPD: listwa SIP2 + zwora

Kondensatory:
C1, C2, C8: 10 μF (SMD 0603, X7R, 10 V)
C3...C5, C7, C9: 100 nF (SMD 0603, X7R, 10 V)
C6: 10 nF (SMD 0805, X7R, 50 V)



Rysunek 3. Rozmieszczenie elementów na płytce modułu

3,3 V (niezależnie od wartości napięcia wejściowego) do zasilania układów U2, U3. Zwora CRTX umożliwia skrosowanie sygnałów CANRX/TX w zależności od sposobu ich wyprowadzenia z układu nadrzędnego, bez konieczności modyfikowania przewodów Grove. Diody LED RXD/TXD (buforowane układem U4) sygnalizują aktywność transmisji w obydwu kierunkach. Układ U2 (TXU0202) jest translatorem poziomów z dwoma kanałami o ustalonym kierunku (1:1), dopuszczającym zasilanie napięciem 1,1...5,5 V każdej ze stron A/B układu. W module zastosowano go do dopasowania poziomów transceivera U3 (pracującego ze stałym zasilaniem 3,3 V), niezależnie od standardu i zasilania sygnału na złączu CAN. Ułatwia to prototypowanie bez względu na zastosowany standard napięciowy czy rodzaj wybranej platformy.

Zwora SPD (domyślnie zwarta) umożliwia sterowanie trybem pracy transceivera U3. Ustawienie stanu niskiego poprzez połączenie zwory SPD (GND na wyprowadzeniu RS) aktywuje tryb High-Speed i pełną funkcjonalność U3. Włączenie pomiędzy wyprowadzenie RS a masę GND rezystora 10…100 kΩ, (w opisywanym modelu R1 ma wartość 10 kΩ) umożliwia określenie szybkości narastania zboczy na wyjściu transceivera. Z rezystorem 10 kΩ szybkość narastania wynosi ok. 15 V/ μs, natomiast z rezystorem 100 kΩ – ok. 2 V/ μs. Wartość rezystora można dobrać w zależności od akceptowalnych zakłóceń EMI – niższa wartość R1 i dłuższe czasy narastania redukują poziom emitowanych zaburzeń. Konstrukcja modelu nie implementuje natomiast trybu obniżonego poboru mocy, w którym aktywny jest tylko odbiornik CAN. Tryb ten można uruchomić, doprowadzając do wyprowadzenia RS stan wysoki (3,3 V).

Układ zmontowany został na dwustronnej płytce drukowanej o formacie zgodnym ze standardem Grove. Rozmieszczenie elementów pokazano na **rysunku 3.** Sposób montażu jest tradycyjny i nie wymaga dokładniejszego opisu.

Gotowy moduł zaprezentowano na **fotografii tytułowej**.

Moduł działa od razu po podłączeniu zasilania i nie wymaga specjalnego uruchamiania. Szybkiego sprawdzenia nakładki można dokonać za pomocą Arduino UNO R4, przy użyciu szkiców dostępnych w środowisku *ArduinoFile* → *Examples* → *Arduino_CAN* → *CANWrite/CANRead* zgodnie z opisem dostępnym na stronie *https://tiny.pl/d4zxd*. W tym celu potrzebne są dwie płytki UNO R4 w dowolnej wersji. Na płytkach Grove_CANDRV aktywujemy zworą RT (obie pozycje włączone) rezystory terminujące, łączymy odcinkiem skrętki wyprowadzenia CANH i CANL złączy CANOUT w obu nakładkach. Sygnały złącza CAN łączymy zaś z wyprowadzeniami CANRX/TX Arduino i napięciem zasilania 5 V. W przypadku R4 Minima są to wyprowadzenia D4:CANTX0 i D5:CANRX0, a w przypadku wersji Wi-Fi – piny D10:CANTX0 i D13:CANRX0. Ustawiamy zwory CRTX w pozycji poziomej, a zworę SPD zwieramy na obu modułach. Na jedną z płytek R4 ładujemy szkic CANWrite, na drugą – CANRead i sprawdzamy poprawność działania (w razie potrzeby zmieniamy ustawienie zwory CRTX). 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 płytek z identycznym VID/PID. Po resecie płytek szkice powinny nadawać i odbierać testową transmisję.

Jeżeli wszystko działa poprawnie, modułu można użyć we własnej aplikacji.

Adam Tatuś

REKLAMA

eprasa.pl 4202fe84f2

AVT sklep

W ofercie AVT*

AVT6053

Najważniejsze parametry:
- osiem kanałów wyjściowych: 24 V/500 mA,
- rezystancja kluczy w stanie otwarcia: 0,55 Ω,
- sterowanie: ekspander MCP23008 z interfejsem I²C,
- napięcie zasilania logiki sterującej: 2,7...5 V.

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

AVT6049	Moduł czterokanałowego przekaźnika mocy do RPi Pico (EP 7/2024)
AVT6046	Moduł komunikacji szeregowej RS232/RS485 do RPi Pico (EP 6/2024)
AVT6030	Dwuportowy mikro HUB USB-C dla systemów wbudowanych (EP 3/2024)
–––	Dwukanałowy port szeregowy z pełnym interfejsem UART (EP 1/2024)
–––	Moduł czterech wyjść HighSide do RPi Pico (EP 11/2023)

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

- **wersja [C]** – zmontowany, uruchomiony i przetestowany zestaw **[B]** (elementy wlutowane w płytkę PCB),
- **wersja [A]** – płytka drukowana bez elementów i dokumentacji.

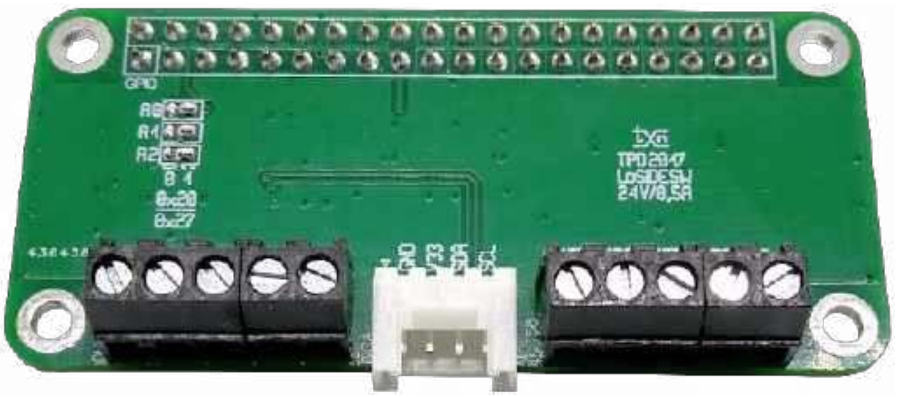
Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- **wersja [A+]** – płytka drukowana **[A]** + zaprogramowany układ **[UK]** i dokumentacja,
- **wersja [UK]** – zaprogramowany układ.

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

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

Moduł przełącznika obciążenia typu low-side do Raspberry Pi

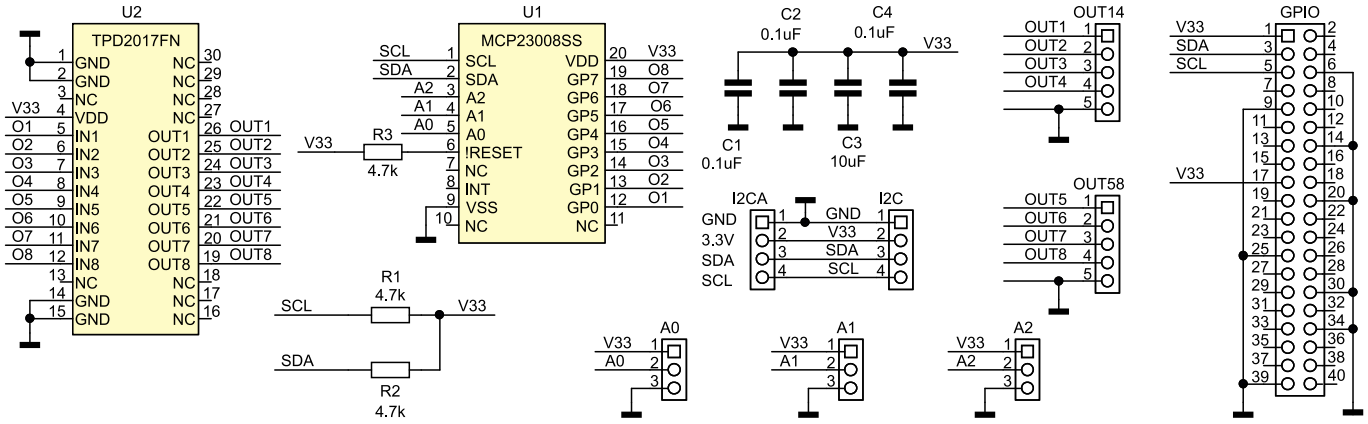
Prezentowana płytka rozszerza funkcjonalność Raspberry Pi Zero (i nie tylko) o możliwość niezależnego sterowania ośmioma obciążeniami o napięciu do 24 V i poborze prądu do 0,5 A.



Schemat modułu pokazano na **rysunku 1**. Jako element kluczujący obciążenia zastosowano specjalizowany ośmiokanałowy driver TPD2017FN, który jest funkcjonalnym odpowiednikiem popularnego ULN2803. Strukturę wewnętrzną układu można zobaczyć na **rysunku 2**.

TPD2017FN w swojej strukturze zawiera osiem identycznych kanałów kluczujących opartych na tranzystorach MOSFET. Kluczowanie odbywa się po stronie masy zasilania obciążenia, czyli Low Side. Układ ma wbudowane zabezpieczenia termiczne, a każdy kanał – zabezpieczenie przed przeciążeniem klucza w przypadku zawarcia wyjścia (aktywowane, gdy prąd przekroczy 1 A). Każdy z tranzystorów kluczujących wyposażony został ponadto w aktywny układ tłumienia przepięć występujących podczas współpracy z obciążeniami indukcyjnymi. W porównaniu z ULN2803 – dzięki wbudowanym zabezpieczeniom – poprawiona została niezawodność sterowania, natomiast dzięki małej rezystancji klucza (max. 0,55 Ω) obniżono moc strat w układzie. TPD2017FN przystosowany jest do bezpośredniego sterowania obciążeniami rezystancyjnymi i indukcyjnymi, dopuszczalne jest równolegle łączenie wyjść w celu zwiększenia prądu obciążenia. Napięcie zasilania i sterowanie układu powinno zawierać się w zakresie 2,7...5 V.

Do sterowania TPD2017FN zastosowano układ ośmiowyjściowego ekspandera GPIO typu MCP23008 współpracującego z magistralą I²C. Za pomocą zwór



Rysunek 1. Schemat modułu

Wykaz elementów:

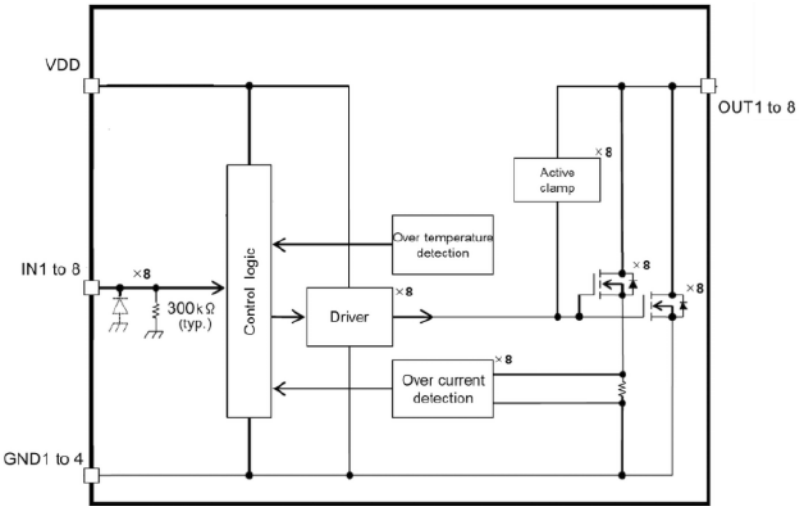
Rezystory: (SMD 0603, 1%)
R1...R3: 4,7 kΩ

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

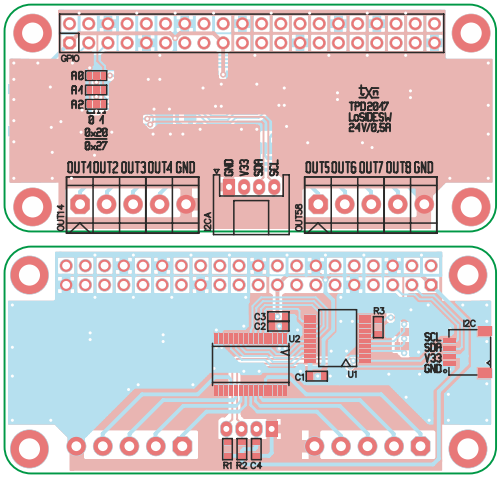
Półprzewodniki:
U1: MCP23008SS (SSOP20_230)
U2: TPD2017FN (SSOP30_300)

Pozostałe:
GPIO: złącze goldpin 2×20
I²C: złącze DG JST 2 mm (opcja)
I²CA: złącze Grove kątowe
OUT14, OUT58: złącze DG 5pin

eprasa.pl 4202fe84f2



Rysunek 2. Struktura wewnętrzna TPD2017FN (za notą Toshiba)



Rysunek 3. Rozmieszczenie elementów modułu: a – warstwa TOP, b – warstwa BOTTOM

A0,1,2 możliwe jest ustawienie jednego z ośmiu adresów (0x20...0x27) magistrali, co dopuszcza podłączenie do ośmiu modułów na jednej magistrali i wygodne sterowanie zespołem nawet 64 wyjść. Całym światem raczej nie posterujemy, ale najbliższą okolicą – na 100%. Wyjścia TPS2017FD, wraz ze wspólną masą doprowadzone są do złączy OUT14 i OUT58. Moduł uzupełniają opcjonalne złącza magistrali I²C (I²C, I²CA). Zasilanie 3,3 V jest pobierane z listwy GPIO Raspberry. Dopuszczalne napięcie zasilania obciążenia to 24 V, maksymalny prąd każdego z kanałów wynosi 0,5 A, a dopuszczalna całkowita moc strat TPD2017FD to 1,8 W. Jeżeli układ będzie pracował w wysokiej temperaturze lub z dużym obciążeniem, warto wyposażyć go w niewielki, naklejany radiator z blaszki miedzianej lub zapewnić chłodzenie wymuszone, aby zapobiec aktywacji zabezpieczenia termicznego.

Układ zmontowano na niewielkiej dwustronnej płytce drukowanej zgodnej z formatem Pi Zero. Rozmieszczenie elementów zaprezentowano na **rysunkach 3a** i **3b**.

Montaż nie wymaga dokładniejszego opisu, należy jedynie pamiętać o zalutowaniu w odpowiednich położeniach zwór adresowych A0..2. Gotową płytkę pokazano na **fotografii tytułowej**.

Moduł nie wymaga uruchamiania – po podłączeniu do Raspberry, poleceniem:

```
i2cdetect -y 1
```

sprawdzamy, czy układ MCP23008 zostanie poprawnie wykryty na magistrali. W modelu ustalony jest adres 0x27 (wszystkie zwory w położeniu „1"). Konfiguracja układu wymaga jedynie zapisu wartości 0x00 do rejestru IODIR (subadres 0x00), ustawiającej funkcję wyjścia GPIO poleceniem:

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

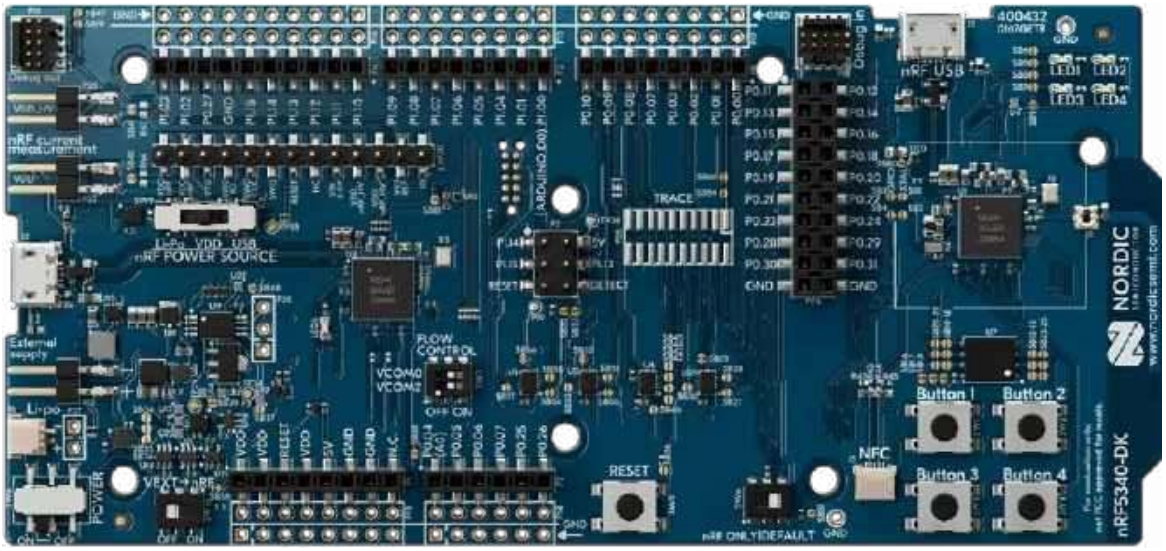
oraz wpisania wartości sterującej stanem wyjść (0xYY) do rejestru OLAT (subadres 0x0A) poleceniem:

```
i2cset -y 1 0x27 0x0A 0xYY
```

Ustawienie poszczególnych bitów zwiera odpowiadające im obciążenia do masy. Jeżeli wszystko działa poprawnie, można moduł zastosować we własnej aplikacji.

Adam Tatuś

REKLAMA



Kurs Nordic nRF z BT

Zanurzymy się w konfigurację środowiska z nRF Connect SDK i przyjrzymy się, co sprawia, że płyta deweloperska nRF5340 DK jest tak wszechstronna. Przygotuj się na ekscytującą podróż przez konfigurację, programowanie oraz testowanie, które otworzą przed Tobą nowe możliwości w technologii Bluetooth Low Energy i systemie Zephyr.

ulubionykiosk.pl

eprasa.pl 4202fe84f2

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

PROJEKT NR 262

Termometr – minutnik

Często w kuchni podczas gotowania zachodzi potrzeba monitorowania temperatury potraw – tak aby nie przekroczyła określonej wartości. Na przykład klarowny rosół nie powinien się gwałtownie gotować, kucharz musi zatem pilnować garnka, dopóki nie osiągnie on żądanej temperatury. Wymóg ten skłonił mnie do zbudowania termometru, który na bieżąco będzie dokonywał pomiarów i po przekroczeniu zadanego progu zasygnalizuje ten fakt. Prezentowane rozwiązanie może stać się pomysłem na praktyczny prezent – a jednocześnie dowodem, że „dłubanina” elektroników czasem się przydaje…

Budowa układu

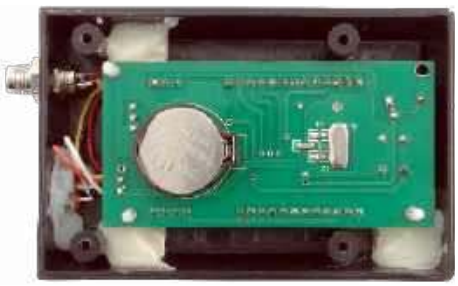
Układ zbudowany został na baziemikrokontrolera jednoukładowego PIC16F876A, którego większość wyprowadzeń steruje siedmiosegmentowym wyświetlaczem LCD 3 i 1/2 cyfry. Schemat układu znajduje się na **rysunku 1**.

Przy założeniu, że ostatnia pozycja wyświetla znak C (stopnie Celsjusza), taka architektura wyświetlacza pozwala pokazać maksymalnie liczbę 199 (tylko liczby całkowite). To w zupełności wystarcza, jeśli weźmie się pod uwagę fakt, że użyty czujnik DS18B20 mierzy temperaturę maksymalnie do +125°C. Oczywiście zastosowanie graficznego wyświetlacza LCD umożliwiłoby stworzenie grafiki nieporównanie estetyczniejszej, ale układ z założenia miał być zasilany baterią 3 V typu CR2032 i pobór prądu okazał się czynnikiem decydującym przy wyborze wyświetlacza. Pod tym względem siedmiosegmentowy panel LCD jest nie do pokonania.

Poza wyświetlaczem do mikrokontrolera podłączony jest przycisk push-button, w którym jako rezystor podciągający użyty został wewnętrzny pull-up mikrokontrolera. Ponadto zastosowano sygnalizator piezoelektryczny bez wbudowanego generatora. Równolegle do sygnalizatora podłączony został natomiast rezystor, który „łagodzi” charakter pojemnościowy tego obciążenia i stanowi element opcjonalny – przy niektórych typach sygnalizatorów nie jest konieczny.

Osobnego omówienia wymaga linia resetująca układ – obwód RC podłączony do portu MCLR mikrokontrolera wykonany wg noty katalogowej Microchipa. Z tym wejściem wiąże się pewien problem: przy zasilaniu napięciem 3 V nie działa poprawnie wewnętrzny układ POR (Power On Reset) mikrokontrolera i dopiero dołączenie R2, R3 oraz C6 zapewnia poprawny start układu w chwili włączenia napięcia zasilania.

Zewnętrzny hermetyczny czujnik temperatury typu DS18B20 podłączony jest przez gniazdo tak, aby w łatwy sposób można go było odłączyć np. do mycia. Mikrokontroler ma zaimplementowaną programowo obsługę szyny komunikacyjnej 1-Wire na pinie 5. Jeśli ktoś korzystał jedynie z gotowych bibliotek do Arduino, dla takiego czujnika temperatury, proponuję pochylić się nieco nad jego protokołem transmisji danych, co pozwoli bardziej docenić pracę wykonywaną przez „anonimowych” twórców bibliotek. Jak to zwykle bywa: fakt, że coś wydaje się proste, może okazać się dowodem, jak bardzo nie doceniamy danego zagadnienia. Dość ścisłe wymagania czasowe obsługi portu 1-Wire oraz różnorodność wyboru instrukcji sterujących czujnikiem DS18B20 mogą



Wykaz elementów:

Rezystory: (SMD 0805)
R1: 4,7 kΩ
R2: 1 kΩ
R3: 10 kΩ
R4: 220 Ω

Półprzewodniki:
T1: BSS84 (SOT-23)
IC1: PIC16876A (SO-28)
IC2: DS18B20 (opis w tekście)

Kondensatory: (SMD 1206, jeżeli nie podano inaczej)
C1, C2: 33 pF
C3, C4: 100 nF
C5: 1 µF
C6: 1 µF / 10 V (THT)

Pozostałe:
BAT: bateria CR2023 wraz z podstawką DS1092-04-B6P lub podobną
BUZ: buzzer THT typu CFD-02
KEY: przycisk TACTS60H150B250 lub podobny
LCD: wyświetlacz LCD typu DE114
SW1: przełącznik suwakowy SW-S12N20E06 lub podobny
Q1: kwarc 4,000 MHz HC49S
Podstawka pod LCD (DIP40) – opis w tekście

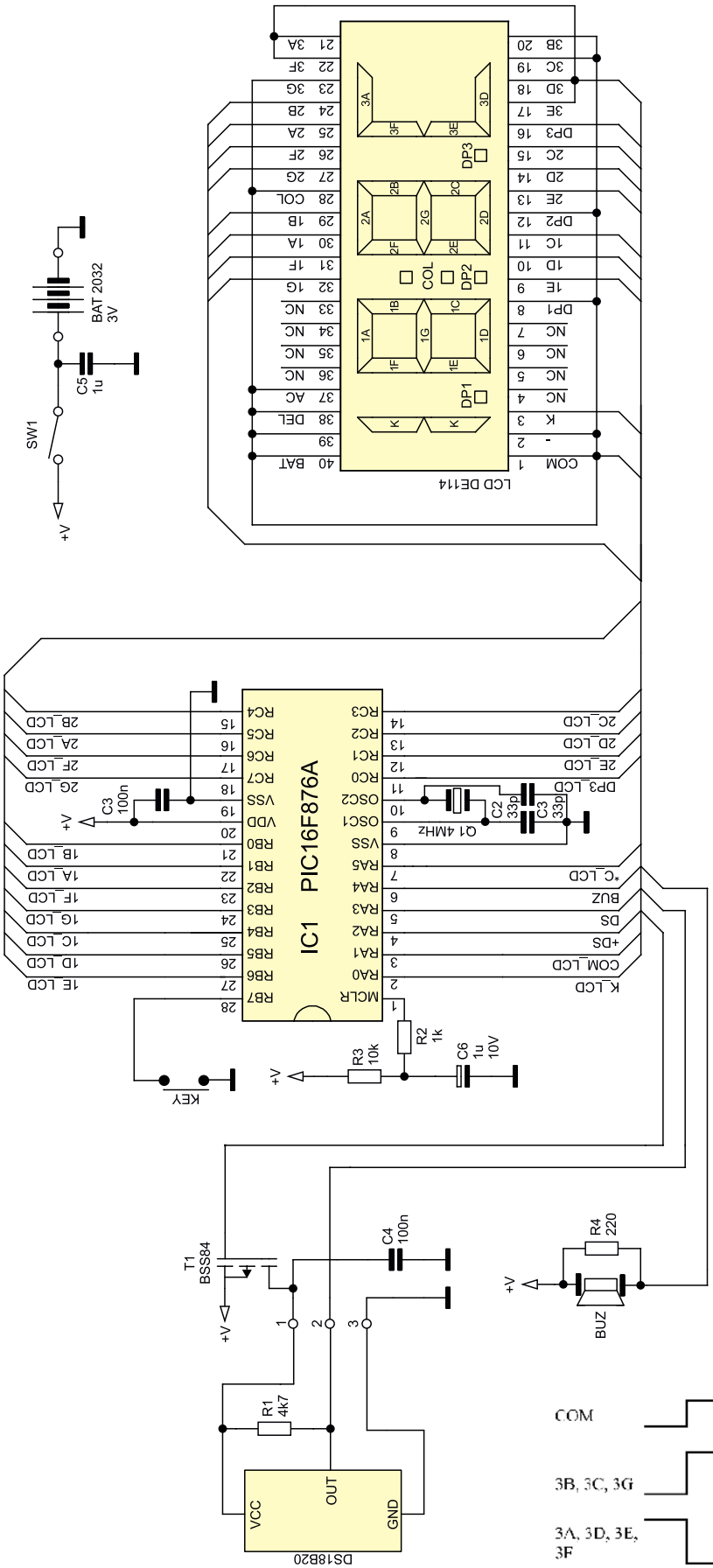
eprasa.pl 4202fe84f2

stanowić wyzwanie dla wielu początkujących programistów. Przy budowaniu własnych systemów warto się z tymi tematami zapoznać.

Linia danych czujnika podciągnięta została do poziomu wysokiego przez rezystor R1 (4,7 kΩ). Na płytce PCB znalazło się miejsce na ten rezystor, ale ja proponuję zamontować go we wtyczce czujnika, tak aby mikrokontroler w łatwy sposób mógł wykrywać obecność czujnika temperatury. Zasilanie wspomnianego sensora podłącza się przez tranzystor T1, który z kolei jest sterowany z poziomu mikrokontrolera przez port RA2. Chodzi o to, aby nie tracić energii z baterii. Czujnik zasilany jest periodycznie. Tuż przed pomiarem temperatury zasilanie ulega załączeniu, natomiast po odczytaniu zmierzonej wartości mikrokontroler ponownie odcina je od czujnika.

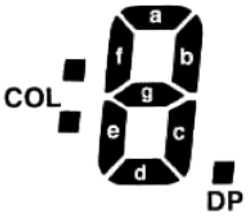
Wyświetlacz LCD (**rysunek 2**) podłączony jest do mikrokontrolera, który steruje jego pracą. Należy pamiętać, że jest to wyświetlacz bez wbudowanego sterownika, stąd trzeba zapewnić na jego wyprowadzeniach odpowiednie sygnały. W przypadku tego typu wyświetlaczy są to przebiegi prostokątne występujące zarówno na poszczególnych elektrodach segmentowych (**rysunek 3**), jak i na wyprowadzeniu wspólnym wszystkich segmentów (COM – pin nr 1 wyświetlacza).

Na segmentach, które nie są aktywne, powinien znajdować się taki sam przebieg, jak na wejściu COM (w fazie), a na segmentach aktywnych (np. segmentach 3A, 3D, 3E, 3F tworzących literę C – stopnie Celsjusza) musi znaleźć się przebieg z portu COM w odwróconej fazie (**rysunek 4**). Nieco „ekwilibrystyki" wymaga zapewnienie, aby statycznie wyświetlana wartość na wyświetlaczu miała odpowiednie przebiegi o częstotliwości „niewidzialnej" dla oka ludzkiego. Wszystkie nieużywane segmenty wyświetlacza (dwukropek, AC itp.) są podłączone

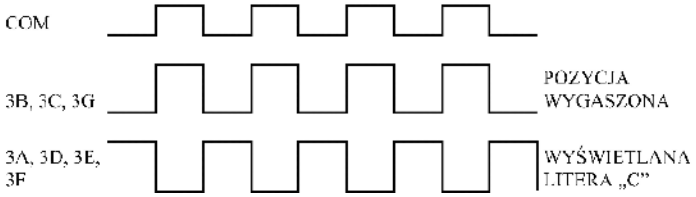


Rysunek 1. Schemat ideowy układu

Rysunek 2. Ułożenie segmentów wyświetlacza LCD



Rysunek 3. Numeracja segmentów LCD



Rysunek 4. Przykładowe przebiegi sterujące wyświetlaczem

eprasa.pl 4202fe84f2

wspólnie do pinu COM. Pozostawienie ich „w powietrzu” mogłoby spowodować aktywowanie segmentu na skutek gromadzenia się ładunku elektrostatycznego.

Układ włączany jest przełącznikiem bistabilnym SW1, odcinającym całkowicie baterię od układu. Kondensator C5, podłączony do baterii przed włącznikiem, przejmuje na siebie „udar” prądowy włączanego urządzenia.

Układ pracujący w trybie minutnika pobiera z baterii około 620 μA, co – przy założeniu, że bateria ma około 200 mAh pojemności – powinno wystarczyć na 322 godziny pracy układu. W trybie pracy termometru sytuacja nieco się komplikuje z uwagi na periodyczne zmiany pobieranego prądu, związane z włączaniem czujnika temperatury i transmisją danych. Wystarczy spojrzeć na przebieg prądu z **rysunku 5**. Zadając sobie nieco trudu, można wyliczyć, że średni prąd wynosi w tym przypadku około 900 μA, co wystarcza na 222 godziny pracy przy zastosowaniu takiej samej baterii.

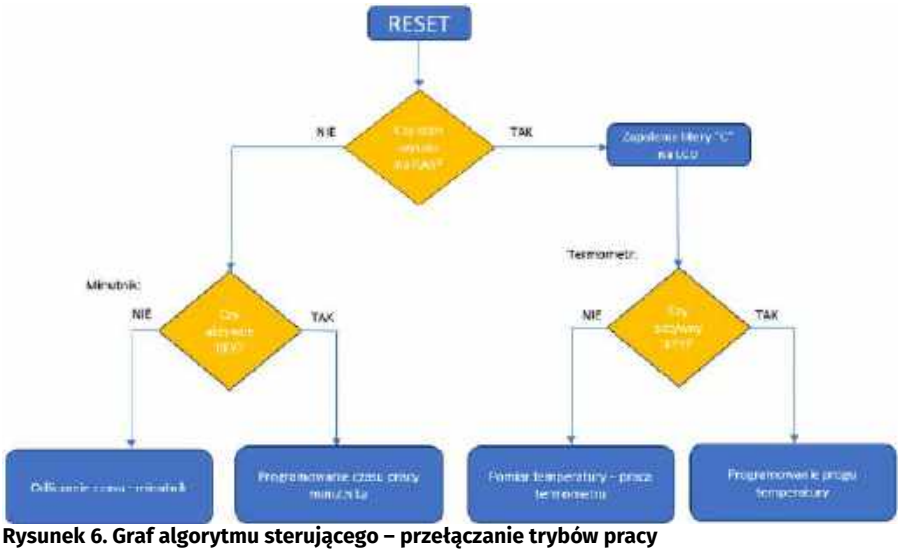
Program

Program napisany został w asemblerze. Używa on przerwań od timera, w których generuje przebiegi niezbędne do odświeżania wyświetlacza LCD – zgodnie z dokonanym pomiarem temperatury. Przerwanie od timera występuje co 40 ms. Jest to wartość graniczna uwzględniająca bezwładność oka ludzkiego i powinna być nieco wyższa, jednak wybór właśnie tej częstotliwości podyktowany był poszukiwaniem wartości wygodnej do obliczeń wskazań minutnika oraz najniższą z możliwych częstotliwości. Obniżenie częstotliwości przerwań sprzyja bowiem redukcji poboru prądu z baterii.

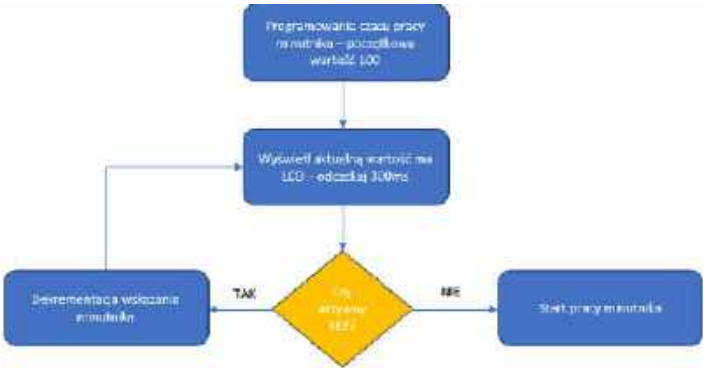
Po włączeniu zasilania i restarcie mikrokontrolera testowany jest port RA3 (szyna 1-Wire). Jeśli wykryty zostaje stan niski (brak podłączonego czujnika temperatury) – układ przechodzi do trybu pracy minutnika. Jeśli natomiast na porcie RA3 procesor wykryje stan wysoki – układ będzie pracował w trybie termometru. Wystarczy więc odpiąć od gniazda czujnik temperatury wraz ze znajdującym się we wtyczce rezystorem R1 i układ „wstanie” jako minutnik.

Następnie testowany jest przycisk KEY (push-button). Jeśli po włączeniu układu zostanie on przyciśnięty, urządzenie przejdzie do trybu programowania ustawień. Wtedy wyświetli się wartość 100, która będzie zmniejszać się do chwili zwolnienia przycisku. Urządzenie zapamięta wówczas tę wartość i zacznie pracę. Jeśli wybraliśmy tryb pracy termometru i zaprogramowaliśmy np. 90°C, to zacznie on wyświetlać bieżącą temperaturę czujnika (odświeżaną co 1 sekundę), a w chwili przekroczenia zaprogramowanego progu (90°C) włączy

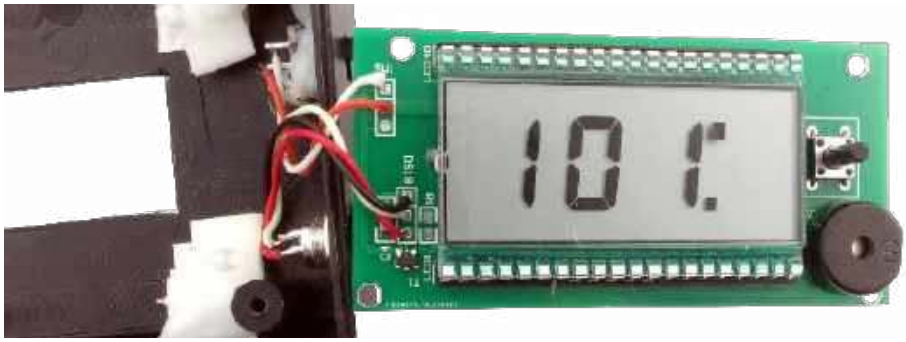
Rysunek 5. Oscylogram pokazujący przebieg prądu zasilania w trybie termometru



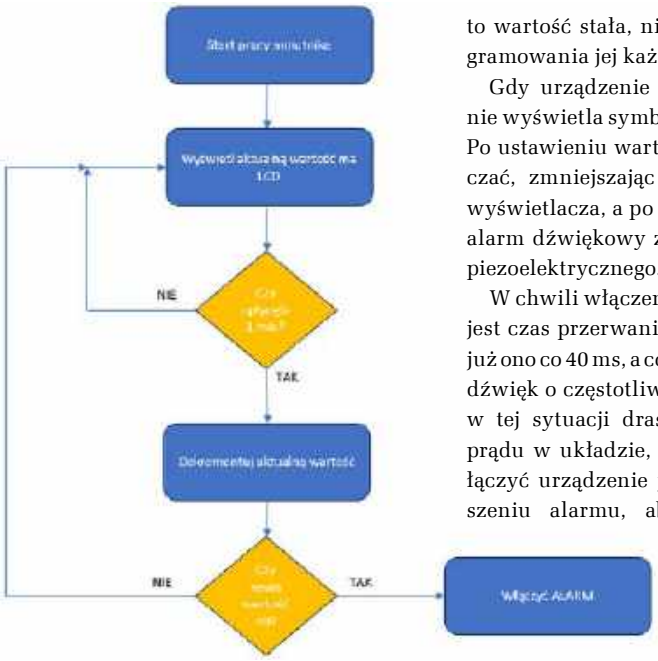
Rysunek 6. Graf algorytmu sterującego – przełączanie trybów pracy



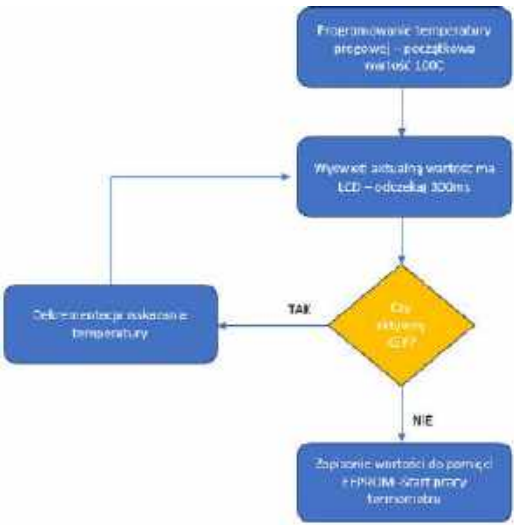
Rysunek 7. Graf algorytmu sterującego – programowanie minutnika



eprasa.pl 4202fe84f2



Rysunek 8. Graf algorytmu sterującego – obsługa minutnika



Rysunek 9. Graf algorytmu sterującego – programowanie termometru

sygnalizator dźwiękowy BUZ. Po wyłączeniu napięcia zasilania i ponownym włączeniu bez programowania (tzn. gdy przycisk jest zwolniony w chwili włączenia zasilania) temperatura progowa wywołująca włączenie alarmu będzie wartością ostatnio zaprogramowaną, pobraną z pamięci EEPROM. Jeśli więc zazwyczaj jest to wartość stała, nie ma konieczności programowania jej każdorazowo.

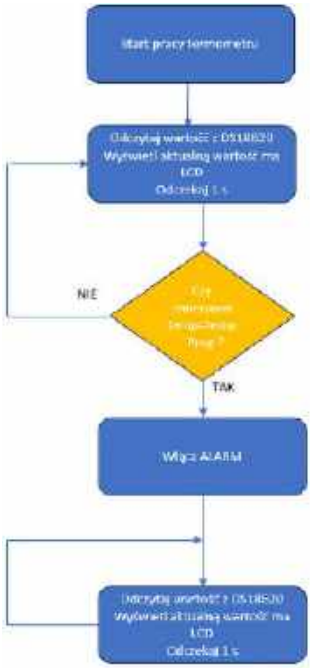
Gdy urządzenie pracuje jako minutnik, nie wyświetla symbolu stopni Celsjusza (C). Po ustawieniu wartości czasu zacznie odliczać, zmniejszając co minutę, wskazanie wyświetlacza, a po osiągnięciu zera włączy alarm dźwiękowy z użyciem sygnalizatora piezoelektrycznego.

W chwili włączenia alarmu przestawiany jest czas przerwania timera i nie następuje już ono co 40 ms, a co 500 µs, co w efekcie daje dźwięk o częstotliwości 2 kHz. Oczywiście w tej sytuacji drastycznie wzrasta pobór prądu w układzie, dlatego dobrze jest wyłączyć urządzenie jak najszybciej po zgłoszeniu alarmu, aby zaoszczędzić energię baterii.

Graf działania programu pokazano na **rysunkach 6...10.**

Montaż

Układ zmontowany został na dwustronnej płytce PCB o wymiarach 80 mm × 40 mm, przy użyciu elementów SMD i THT. Najwygodniej byłoby wgrać program do mikrokontrolera przed montażem, ale to wymaga specjalistycznej, drogiej podstawki. Można więc wlutować niezaprogramowany układ na płytkę PCB, a następnie czterema przewodami podłączyć nasz mikroprocesor do programatora, używając pól lutowniczych ścieżek podłączonych do mikrokontrolera (GND, MCLR – pin rezystora R2, RB6 i RB7 – pola lutownicze przeznaczone do przycisku i pinu 9 wyświetlacza). Po zaprogramowaniu mikrokontrolera należy odlutować przewody, wlutować pozostałe elementy SMD, a następnie THT. Do podłączenia wyświetlacza dobrze jest użyć podstawki – można w tym celu zastosować podstawkę do układów scalonych w obudowie DIP40 (odcinając dwa rzędy wyprowadzeń od łączących je poprzeczek), tak aby dało się ją wlutować do PCB. Nieużywane piny wyświetlacza – oznaczone jako NC

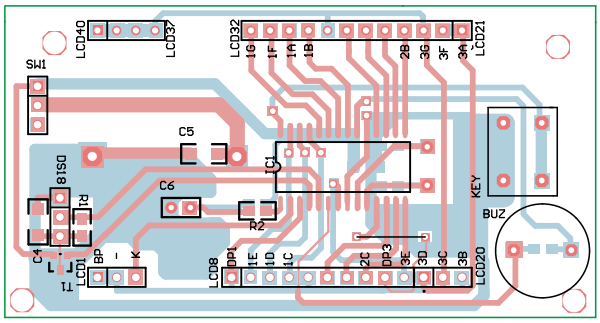


Rysunek 10. Graf algorytmu sterującego – obsługa termometru

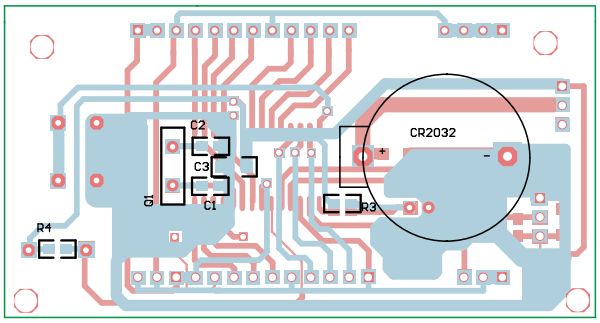
– nie są podłączone do układu. Należy spod tych pinów usunąć złącza podstawki (brak odpowiednich otworów w PCB wynika z umieszczenia w tym obszarze innych ścieżek).

Gniazdo do podłączenia czujnika temperatury podłączyłem za pomocą przewodów. Zastosowałem 3-pinowe, okrągłe gniazdo M8, ale można wykazać się tu inwencją. Wygodnym rozwiązaniem byłoby też złącze minijack stereo, choć należałoby pamiętać o tym, aby wyjmować i wkładać wtyk przy wyłączonym zasilaniu. Na końcu, po sprawdzeniu poprawności montażu (mikrokontroler wymaga zwrócenia szczególnej uwagi na ewentualne zwarcia wyprowadzeń), należy włożyć wyświetlacz w podstawkę, zachowując przy tym ostrożność: jest to szklany element i łatwo go uszkodzić. Jeśli z jakiegoś powodu zajdzie konieczność zdemontowania wyświetlacza, to w żadnym wypadku nie należy podważać go śrubokrętem – metalowy przedmiot łatwo uszczerbi szkło. Lepsza do tego celu okaże się drewniana wykałaczka lub patyczek po lodach.

Grzegorz Mazur



Rysunek 11. Schemat montażowy PCB – warstwa TOP



Rysunek 12. Schemat montażowy PCB – warstwa BOTTOM

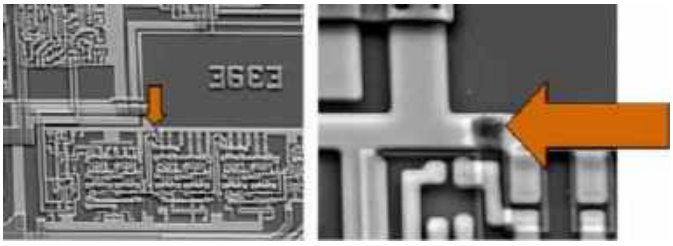
eprasa.pl 4202fe84f2

Elektroniczne komponenty zabezpieczające

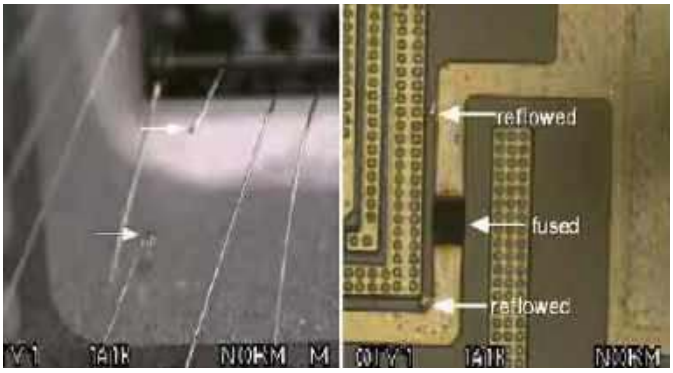
– podzespoły, aplikacje i normy

*Współczesne elementy i układy elektroniczne potrafią przetrwać naprawdę zaskakująco trudne warunki eksploatacji. Paradoksalnie, czasem nawet pozornie błahe zdarzenie jest w stanie z łatwością uszkodzić wrażliwe komponenty, które nie zostały odpowiednio zabezpieczone przed wyładowaniami ESD, krótkimi przepięciami, o wyładowaniach atmosferycznych nie wspominając. Tymczasem oferta elementów chroniących układy elektroniczne przed zniszczeniem jest dziś tak szeroka, że bez problemu można znaleźć odpowiedni produkt do każdej aplikacji. Mało tego – w obecnych czasach ani wysoka cena, ani nawet spore rozmiary komponentu (**fotografia 1**), nie są już argumentami przemawiającymi za rezygnacją ze stosowania elementów ochrony obwodów. Niemal od ręki można bowiem wskazać kompaktowe, niedrogie komponenty przeciwprzepięciowe – trzeba jedynie zwrócić uwagę na istotne parametry i związane z nimi ograniczenia, które determinują możliwość użycia danego podzespołu w określonych obszarach aplikacyjnych.*

Fotografia 1. Dioda TVS o wymiarach 0,6×0,3 mm marki Infineon w porównaniu z głową mrówki (http://t.ly/SOk6I)



Fotografia 2. Uszkodzenie ścieżki przewodzącej w strukturze układu scalonego spowodowane wyładowaniem ESD (http://t.ly/zBfqX)



Fotografia 3. Uszkodzenie bondingu procesora spowodowane przepływem prądu o chwilowej wartości 1 A (http://t.ly/hFS-m)

Znaczenie komponentów zabezpieczających we współczesnej elektronice

Wzrost znaczenia ochrony elementów i układów elektronicznych przed wyładowaniami elektrostatycznymi (ang. ESD – ElectroStatic Discharge) ma swoje źródło przede wszystkim w powszechnej obecności tworzyw sztucznych w naszym codziennym życiu. Choć materiały naturalne (organiczne) także mogą ulegać elektryzowaniu, to jednak głównym winowajcą awarii elektroniki spowodowanych przez elektryczność statyczną są wszelkiej maści sztuczne tkaniny, folie, taśmy samoprzylepne, opakowania tworzywowe i inne obiekty wykonane z polimerów. Nie jest tajemnicą, że zdecydowanie najpowszechniejszym powodem powstawania uszkodzeń jest… poruszanie się ludzi ubranych w odzież z tkanin syntetycznych i/lub przemieszczanie się po podłogach pokrytych wykładziną albo (jeszcze gorzej) dywanem. Chyba każdy miał okazję zetknąć się z efektownymi iskrami przeskakującymi podczas zakładania swetra lub składania koca – jeżeli zjawisku temu towarzyszą także efekty dźwiękowe (trzask), możemy być pewni, że mamy do czynienia z uwolnieniem naprawdę solidnego ładunku statycznego, który z łatwością byłby w stanie „zabić" większość delikatnych komponentów czy też słabiej zabezpieczonych układów elektronicznych.

Fascynujące postępy elektroniki na przestrzeni ostatnich kilkudziesięciu lat przejawiają się w dwóch aspektach: coraz większej miniaturyzacji struktur układów scalonych oraz upowszechnieniu technologii CMOS. Jak na złość, obydwa wymienione procesy (zachodzące jednocześnie i wzajemnie się przeplatające) również działają zdecydowanie na niekorzyść samych półprzewodników, obniżając ich odporność na wyładowania elektrostatyczne. Technologia oparta na mikroskopijnych, ultracienkich tranzystorach MOSFET jest bowiem szczególnie wrażliwa na elektryczność statyczną m.in. ze względu na łatwość, z jaką ładunki takie mogą przebić subtelne warstwy izolujące bramki tranzystorów. Rzecz jasna, nie jest to jedyny mechanizm awarii struktur krzemowych – uszkodzeniu mogą ulec także inne części samego półprzewodnika (**fotografia 2**), a nawet cienkie połączenia drutowe (tzw. bonding), znajdujące się pomiędzy polami kontaktowymi krzemowej płytki a wewnętrznymi zakończeniami wyprowadzeń układu scalonego (**fotografia 3**). To, jaki mechanizm awarii wystąpi w danym przypadku, zależy rzecz jasna od rodzaju elementu, ale w jeszcze większym stopniu

eprasa.pl 4202fe84f2

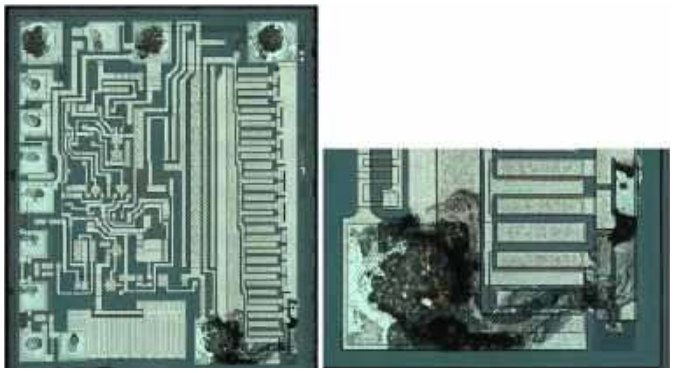
Fotografia 4. Symulator ESD z końcówką do testów metodą wyładowań kontaktowych (http://t.ly/oEv4v)

Fotografia 5. Przebieg testu wyładowania powietrznego z użyciem pistoletowego symulatora ESD (http://t.ly/E0-20)

wpływają nań warunki, w których doszło do wyładowania elektrostatycznego – a te są niezwykle trudne bądź niemożliwe do przewidzenia z uwagi na losową naturę „zdarzenia" ESD (ang. ESD event).

Wspomnieliśmy, że ze względów konstrukcyjnych układy CMOS są szczególnie wrażliwe na ładunki statyczne. Nie oznacza to jednak, że innych komponentów ów problem nie dotyczy – powszechnie wiadomo, że małe diody LED (a zwłaszcza diody laserowe) również wykazują sporą podatność na wyładowania ESD, ale w rzeczywistości nawet prosty tranzystor bipolarny lub inny element może zostać uszkodzony, jeżeli wyładowanie będzie odpowiednio intensywne. Wszystko zależy bowiem od sposobu „połączenia" wyprowadzeń elementu z obiektem będącym nośnikiem ładunków. Najbardziej niebezpieczne są wyładowania kontaktowe, które zachodzą poprzez bezpośredni, galwaniczny kontakt obiektu (np. igły testowej symulatora ESD – **fotografia 4**) ze ścieżkami na PCB, stykami złącza czy też wyprowadzeniami podzespołu. W takich warunkach przepływ impulsowego prądu o zaskakująco dużym natężeniu (i w niezwykle krótkim czasie) może z łatwością uszkodzić nawet te obwody lub komponenty, które mają już zapewniony pewien poziom ochrony ESD. Drugim rodzajem niszczycielskich zjawisk jest wyładowanie przez powietrze – w warunkach testowych realizowane poprzez zbliżenie końcówki pistoletu ESD (generatora wysokonapięciowego) do danego elementu (**fotografia 5**).

Warto podkreślić, że – choć w wielu przypadkach uszkodzenie spowodowane wyładowaniem ESD (np. podczas montażu) może dać o sobie znać już podczas pierwszej próby uruchomienia urządzenia – to znacznie większe ryzyko niosą ze sobą defekty mniej widoczne lub wręcz całkowicie niewykrywalne przez wiele godzin, dni, a nawet miesięcy. Tego typu sytuacje są określane angloję zycznym zwrotem „walking wounded" (**fotografia 6**), co dosłownie oznacza „chodzący ranny". Określenie to, zaadaptowane z medycyny ratunkowej, w tym przypadku dotyczy „pacjentów"



Fotografia 6. Przykład uszkodzenia struktury półprzewodnikowej typu „walking wounded" – stabilizator poddany wyładowaniu ESD pracował jeszcze przez godzinę po narażeniu na impuls wysokiego napięcia (http://t.ly/5ZFfS)

półprzewodnikowych – pewne rodzaje awarii mogą utrzymywać się, ba – pogłębiać! – przez dłuższy czas, by w najmniej oczekiwanym momencie ujawnić się z pełną mocą. Nietrudno się domyślić, że im większe znaczenie dla bezpieczeństwa ludzi, środowiska lub mienia ma dane urządzenie, tym istotniejsze ryzyko niesie ze sobą awaria typu „walking wounded" – dość wspomnieć chociażby o wyposażeniu lotniczym, aparaturze medycznej, a nawet instalacjach motoryzacyjnych. Autor sam zetknął się kiedyś z interesującym uszkodzeniem procesora ARM z serii Stellaris firmy Texas Instruments – choć układ działał poprawnie (procesor cały czas poprawnie realizował wgrany do pamięci program), to przypadkowe uszkodzenie ESD doprowadziło do bardzo silnego przegrzewania mikrokontrolera już w pierwszych sekundach po włączeniu zasilania. Jest to dobry przykład defektu, który choć sam w sobie nie był w stanie „zabić" układu scalonego, prowadził prostą drogą do innego scenariusza awarii (w tym przypadku – przegrzania struktury).

Normy dotyczące ochrony ESD

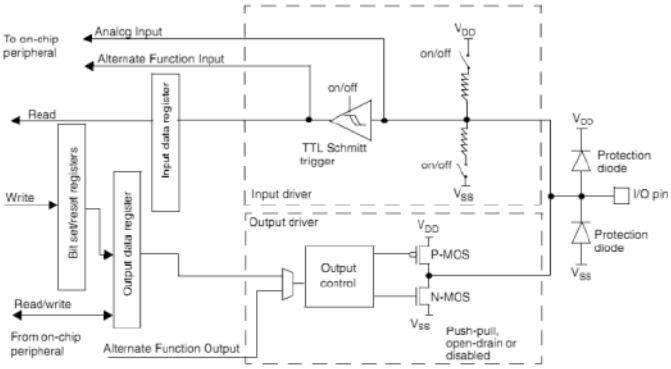
Wspomnieliśmy już, że jednym z najczęściej rozpatrywanych źródeł elektryczności statycznej jesteśmy my sami, poruszający się w naturalnym dla nas (ale syntetycznym z materiałowego punktu widzenia) środowisku. Dość powiedzieć, że przejście po podłodze wyłożonej dywanem może doprowadzić do wytworzenia różnicy potencjałów na poziomie… 35 kV! Orientacyjne wartości napięć wytwarzanych w różnych warunkach wilgotności powietrza i różnych sytuacjach generowania ładunków zebrano w **tabeli 1**.

Właśnie dlatego normy branżowe dotyczące wymogów oraz metod badania odporności na wyładowania ESD odnoszą się bardzo często do tzw. Human Body Model (HBM), czyli „elektrostatycznego" modelu ludzkiego ciała. Należy jednak wyraźnie odgraniczyć HBM od innych modeli zjawisk ESD, gdyż zamienne stosowanie różnych standardów stanowi często popełniany błąd w środowisku konstruktorów. Mało tego – producenci układów scalonych nagminnie

Tabela 1. Różnice potencjałów wytwarzanych przez ładunki statyczne, gromadzone w sytuacjach znanych z codziennego życia (http://t.ly/wOyd9)

Scenariusz wytwarzania ładunków statycznych	Różnica potencjałów [V]	
	@ 10...25% RH	@ 60...90% RH
Chodzenie po dywanie	35 000	1 500
Chodzenie po płytkach PVC	12 000	250
Praca przy biurku	6 000	100
Podniesienie torebki foliowej z biurka	20 000	1 200
Siedzenie na krześle wypełnionym pianką poliuretanową	18 000	1 500

eprasa.pl 4202fe84f2

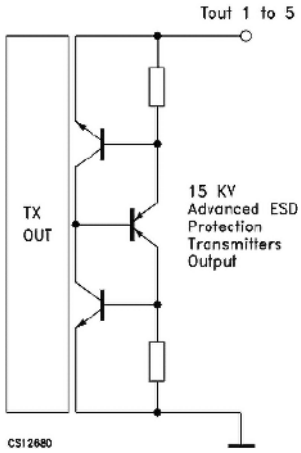


Rysunek 1. Schemat blokowy linii portu GPIO mikrokontrolera STM32 z widocznymi diodami zabezpieczającymi (po prawej stronie) – http://t.ly/xiaXd

piszą noty katalogowe w sposób, który dodatkowo sprzyja wpadaniu w pułapkę nadmiernego uproszczenia.

Aby dogłębnie zrozumieć zagadnienia ochrony elektrostatycznej, należy w pierwszej kolejności rozgraniczyć dwa obszary:

- **Ochrona ESD na poziomie komponentu** dotyczy zabezpieczeń umieszczanych bezpośrednio w strukturach krzemowych (na wejściach oraz wyjściach układów scalonych) i przyjmujących zwykle postać dwóch szeregowo połączonych diod, włączonych pomiędzy szynę zasilana a masę w kierunku zaporowym (**rysunek 1**), czasem uzupełnionych o dodatkowe elementy (**rysunek 2**).
- **Ochrona ESD na poziomie systemowym** jest najczęściej realizowana za pomocą dodatkowych komponentów dyskretnych lub prostych macierzy scalonych, zawierających przede wszystkim diody TVS (**rysunek 3**).



Rysunek 2. Rozbudowany układ ochronników ESD w strukturze transceivera RS-232 typu ST3237EB (źródło: datasheet ST3237EB)

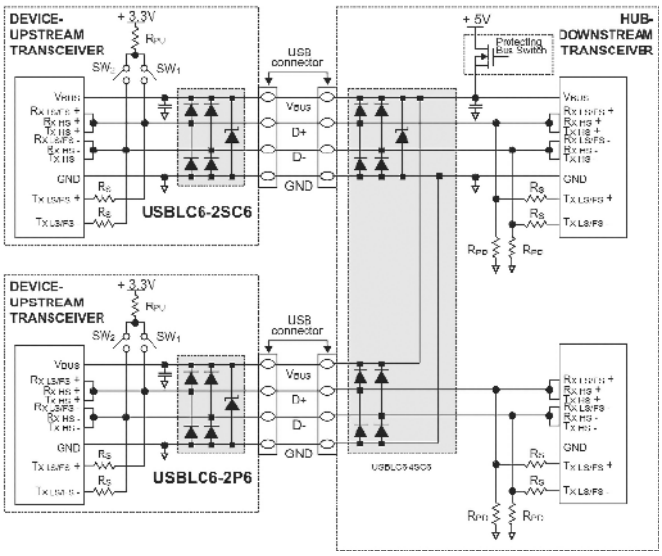
Jeżeli zechcielibyśmy porównać opisywane rodzaje ochrony do sytuacji znanych z wojskowości, to zabezpieczenia na poziomie komponentu odgrywałyby rolę kamizelki kuloodpornej, stosowanej niezależnie przez każdego żołnierza i mającej na celu ochronę przed bezpośrednim atakiem. Ochronę na poziomie systemowym w naszej militarnej analogii stanowiłyby natomiast… stalowe płyty pancerza czołgu bądź innego pojazdu bojowego. Ich zadaniem byłaby bowiem osłona wszystkich układów scalonych (żołnierzy) znajdujących się wewnątrz pojazdu (urządzenia).

Już to proste porównanie doskonale obrazuje różnice pomiędzy dwoma opisanymi poziomami ochrony ESD. W przypadku elektroniki wytrzymałość naszego „pancerza" jest determinowana przez trzy parametry:

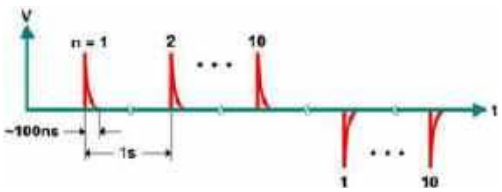
1. **Czas narastania** – im krótszy impuls, tym trudniej jest zbudować skuteczną ochronę, której zadaniem będzie efektywne stłumienie go do poziomu bezpiecznego dla układów scalonych i innych delikatnych elementów półprzewodnikowych. Kluczowy jest tutaj bowiem czas reakcji na impuls lub – innymi słowy – szybkość załączania ochronnika (np. diody TVS).
2. **Wartość szczytowa prądu** – jak już wspomnieliśmy, impuls prądowy wywołany napięciem na poziomie kilku…kilkudziesięciu kilowoltów jest w stanie wywołać przepływ prądu o zaskakująco wysokim natężeniu – i to nawet przez relatywnie dużą rezystancję. Dlatego właśnie kluczowe znaczenie dla ochrony ESD ma impedancja źródła testowego, które stosowane jest do badania odporności. Takie samo napięcie może z łatwością „zabić" element, ale równie dobrze może „nie zrobić na nim większego wrażenia" – wszystko zależy od tego, jak wysoka jest impedancja źródła widziana od strony wejścia badanego układu.
3. **Liczba impulsów** – w codziennej mowie często używamy zwrotu „do trzech razy sztuka". W ochronie ESD liczba powtórzeń także ma istotne znaczenie, choć nie zawsze impuls testowy spowoduje uszkodzenie układu lub urządzenia po 3. „strzale" z rzędu. W zależności od stosowanego standardu, liczba impulsów może wahać się od 1 do nawet 10, a im jest ich więcej, tym trudniej będzie ochronnikom ESD przetrwać powtarzalne narażenie (**rysunek 4**).

W zakresie ochrony na poziomie komponentu pod uwagę bierze się trzy scenariusze, z których każdy doczekał się własnego modelu teoretycznego oraz odpowiadających mu wymogów normalizacyjnych:

- **HBM (Human Body Model)** – jak to już wspomnieliśmy wcześniej, model najczęściej występujący w notach katalogowych producentów. I nie ma w tym nic dziwnego, gdyż jest on nie tylko dobrym odwzorowaniem warunków naszej inżynierskiej codzienności (występujących np. podczas dotykania wyprowadzeń komponentu w czasie jego przenoszenia lub montażu), ale zarazem… pozostaje najłagodniejszy (!) dla badanych komponentów. HBM zakłada bowiem, że rozładowanie pojemności 100 pF (naładowanej uprzednio ze źródła HV) następuje przez rezystancję szeregową o wartości 1,5 kΩ – a zatem całkiem sporą (**rysunek 5**). Z tego też względu krzywa przebiegu napięcia w czasie zdarzenia ESD ma dość długi czas narastania (na poziomie 25 ns). Zabezpieczenie ma zatem dość sporo czasu na zadziałanie, a i sam prąd szczytowy przy tak wysokiej rezystancji osiąga relatywnie małe wartości (do 5,33 A przy napięciu testowym 8 kV). Takie parametry testu są implementowane zarówno przez standard JEDEC/ESDA JS-001, jak i wojskową normę

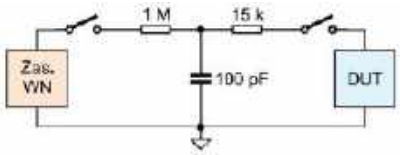


Rysunek 3. Przykład zastosowania scalonych transili USBLC6-2P6 oraz USBLC6-4SC6 do ochrony linii zasilania i danych portów USB po obydwu stronach połączenia (device oraz host) – źródło: datasheet USBLC6-2



Rysunek 4. Przebieg czasowy sekwencji testowej wg normy 61000-4-2 (http://t.ly/OHLs0)

eprasa.pl 4202fe84f2



Rysunek 5. Obwód testowy wg modelu HBM (http://t.ly/Kjwqo)

MIL-STD-883 (przy czym ta druga zakłada konieczność progresywnego testowania napięciami 1 kV, 2 kV oraz 4 kV, zaś pierwsza 2 kV). Poziomy klasyfikacji komponentów wg JEDEC JS-001 zebrano w **tabeli 2**.

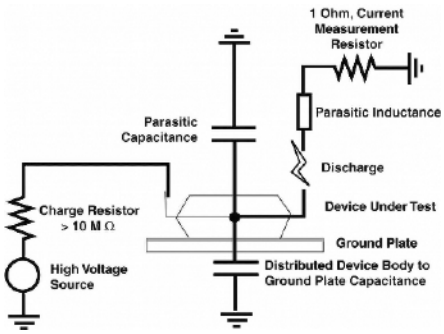
Tabela 2. Klasyfikacja odporności komponentów wg JEDEC JS-001 (http://t.ly/5CYe8)

Klasyfikacja	Napięcie [V]
0Z	< 50
0A	50...125
0B	125...250
1A	250...500
1B	500...1000
1C	1000...2000
2	2000...4000
3A	4000...8000
3B	>= 8000

- **CDM (Charged Device Model)**, – czyli w dosłownym tłumaczeniu „model urządzenia naładowanego" odnosi się do sytuacji, w której komponent, na którym zgromadził się pewien ładunek statyczny, rozładowuje się w wyniku nagłego kontaktu z powierzchnią o niższym potencjale. Przykładów nie trzeba szukać daleko – wystarczy wyobrazić sobie układ scalony przesuwający się w podajniku wibracyjnym automatu pick & place (elektryzowanie przez tarcie!), a następnie nagle stykający się wyprowadzeniami z metalową powierzchnią (np. padami na PCB, głowicą montażową czy też metalową płytą połączoną z obudową maszyny). Co ciekawe choć maksymalne napięcia stosowane w teście CDM są kilkakrotnie niższe niż w przypadku HBM, nie oznacza to wcale, że badanie CDM jest „przyjemniejsze" dla struktury półprzewodnikowej – pamiętajmy bowiem, że w tym przypadku nie mamy już rezystancji szeregowej rzędu 1,5 kΩ, ale zaledwie… 1 Ω – a to wpływa na wartość maksymalnego prądu szczytowego w czasie rozładowania. Schemat układu testowego można zobaczyć na **rysunku 6**, zaś klasyfikację wg modelu CDM – w **tabeli 3**.
- **MM (Machine Model)** – ten rodzaj badania opiera się na rozładowaniu pojemności testowej poprzez niewielką indukcyjność szeregową (z pominięciem rezystancji szeregowej, stosowanej we wszystkich innych metodach testów), co sprawia, że przebieg prądowy uzyskiwany w ramach MM ma silnie oscylacyjny

Tabela 3. Klasyfikacja odporności komponentów wg JEDEC JS-002 (http://t.ly/g6TPw)

Klasyfikacja	Napięcie [V]
C0A	< 125
C0B	125...250
C1	250...500
C2A	500...750
C2B	750...1000
C3	>= 1000



Rysunek 6. Schemat układu testowego wg modelu CDM (http://t.ly/YrrRp)

Idealne połączenia między płytkami

Niezawodne złącza serii FINEPITCH

- złącza w rastrach od 0,635 mm do 2,54 mm
- wykonanie ekranowane lub nieekranowane
- gwarancja bezawaryjnego łączenia płytek w różnych konfiguracjach orientacji w przestrzeni, odległości między laminatami lub ilości styków
- transmisja z dużą prędkością do 52 Gb/s

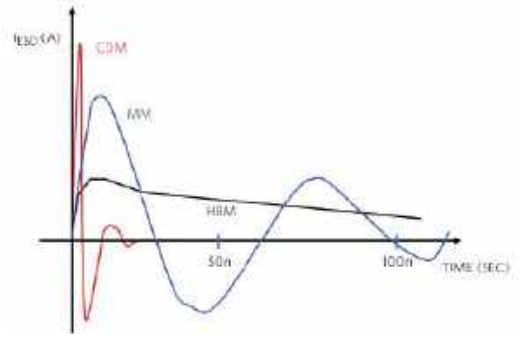
Odwiedź naszą stronę internetową i dowiedz się więcej!

PHOENIX CONTACT
INSPIRING INNOVATIONS

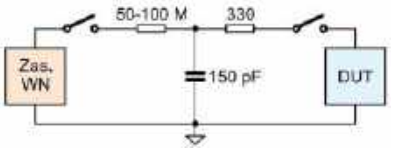
REKLAMA

© Phoenix Contact 2024

eprasa.pl 4202fe0f6



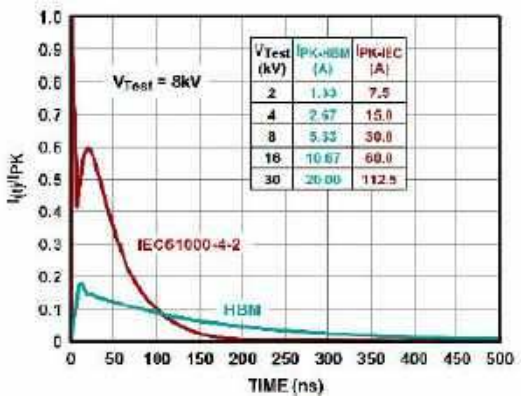
Rysunek 7. Porównanie przebiegów prądu testowego w badaniach wg modeli HBM, CDM oraz MM (http://t.ly/S_Idr)



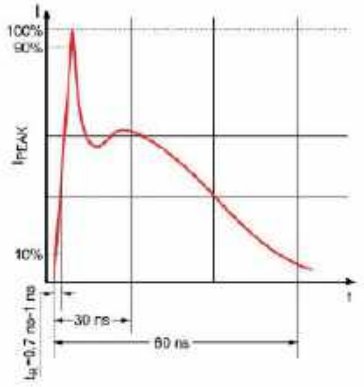
Rysunek 8. Schemat układu testowego wg normy 61000-4-2 (http://t.ly/Kjwqo)

charakter. Metoda MM nie przyjęła się w przemyśle z uwagi na niewielką powtarzalność – testy wg HBM miały oceniać dokładnie te same scenariusze uszkodzeń co CDM, dlatego w praktyce spotyka się dziś właściwie tylko zalecenia dotyczące badań wg HBM oraz CDM.

Znaczne różnice pomiędzy modelami stosowanymi w opisanych powyżej metodach testowania komponentów wpływają na diametralnie różne kształty i amplitudy przebiegów prądu – porównanie można zobaczyć na **rysunku 7**. Niestety, zagadnienie jest jeszcze bardziej złożone, przez co wymienione modele (a zwłaszcza HBM) są często mylone z innym standardem – 61000-4-2 – dotyczącym ochrony



Rysunek 9. Porównanie krzywych prądu testowego wg HBM oraz normy 61000-4-2 (http://t.ly/QettF)



Rysunek 10. Zależności czasowe krzywej impulsu prądowego wg normy 61000-4-2 (http://t.ly/Kjwqo)

Tabela 4. Poziomy odporności na wyładowania ESD wg normy 61000-4-2 (http://t.ly/lgs3l)

Rodzaj wyładowania			
Kontaktowe		Powietrzne	
Poziom	Napięcie testowe [kV]	Poziom	Napięcie testowe [kV]
1	2	1	2
2	4	2	4
3	6	3	8
4	8	4	15
X	*	X	*

* – poziom specjalny, określony w wymaganiach danego urządzenia

na poziomie systemowym. W tym przypadku wyższe są nie tylko napięcia testowe, ale także (i to aż kilkakrotnie) – wartości prądów szczytowych. Wynika to z relatywnie niskiej rezystancji, włączanej w szereg z pojemnością (od strony rozładowania do badanego obiektu): w tym przypadku jest to 330 Ω (**rysunek 8**). Sama krzywa testowa też okazuje się dość złożona, a wynika to z konieczności zasymulowania rzeczywistych warunków wyładowań w tzw. środowiskach niekontrolowanych. Porównanie przebiegów prądu wg HBM oraz IEC 61000-4-2 można zobaczyć na **rysunku 9**, zaś szczegóły „anatomiczne" pojedynczego wyładowania zobrazowano na **rysunku 10**. Należy pamiętać, że w przypadku badań wg HBM wykonuje się zwykle pojedyncze „strzały" w obu polaryzacjach, zaś norma 61000-4-2 zakłada po 10 impulsów w każdej z polaryzacji – nie ma w tym nic dziwnego, jeżeli zwrócimy uwagę na fakt, że każde urządzenie stosowane na co dzień (np. smartfon, aparat fotograficzny czy laptop) jest narażone na niezliczone wyładowania ESD podczas obsługi oraz przenoszenia. I właśnie dlatego to ta norma powinna być uwzględniana przy projektowaniu ochrony ESD większości urządzeń, w tym przede wszystkim podczas dobierania transili, scalonych macierzy TVS i innych komponentów stojących na straży wrażliwego wnętrza urządzeń elektronicznych.

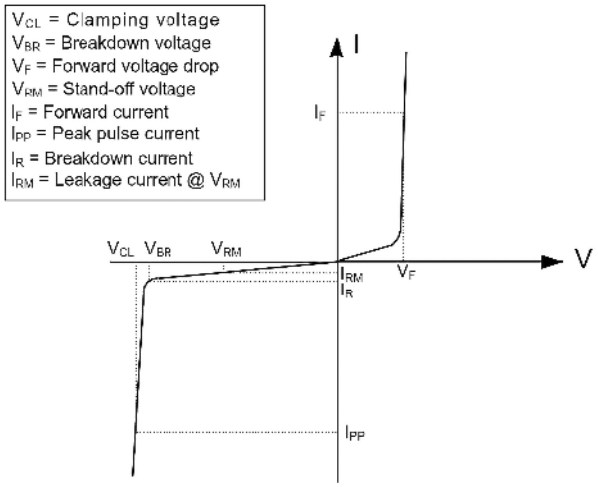
Omawiając normę 61000-4-2, nie sposób nie wspomnieć także o stosownej klasyfikacji odporności urządzeń na zdarzenia ESD. W tym przypadku mamy już tylko cztery poziomy, ale każdy z nich – ze względu na specyfikę testów diametralnie inną niż w przypadku HBM/CDM/MM – jest określony osobno dla wyładowań kontaktowych i powietrznych (**tabela 4**). Zasada jest prosta: jeżeli wykonanie badania kontaktowego na wymaganym od danego urządzenia poziomie byłoby niemożliwe, to końcówkę symulatora ESD zbliża się do danego elementu (np. styku złącza), aż do momentu przeskoku iskry. I tak, w przypadku poziomu 4 wg 61000-4-2, brak możliwości wykonania testu kontaktowego z napięciem 8 kV skutkuje wymogiem „przełączenia się" na badanie z wyładowaniem powietrznym, ale już na znacznie wyższym poziomie 15 kV. Wybór poziomu zależy oczywiście od rodzaju aplikacji – przykładowo, wg 4. edycji normy IEC/EN 60601-1-2 dotyczącej kompatybilności elektromagnetycznej medycznych urządzeń elektrycznych, do badań aparatury należy stosować najwyższy, czwarty poziom (czyli właśnie 8 kV kontaktowo/15 kV przez powietrze).

Rzecz jasna, IEC 61000-4-2 to niejedyny standard używany do oceny odporności produktów końcowych (oraz elementów ochronnych) przed skutkami wyładowań elektrostatycznych. Poszczególne branże rządzą się swoimi prawami i – jak to zwykle bywa – dobry przykład może stanowić tutaj sektor motoryzacyjny, w którym obowiązującym standardem ESD pozostaje norma ISO 10605.

Przegląd komponentów do ochrony przeciwprzepięciowej

Wspomnieliśmy już, że wyładowania ESD wiążą się nie tylko z występowaniem impulsów wysokiego napięcia (zdolnego np. do przebicia warstwy izolatora bramki tranzystora MOSFET), ale także prądu

eprasa.pl 4202fe84f2



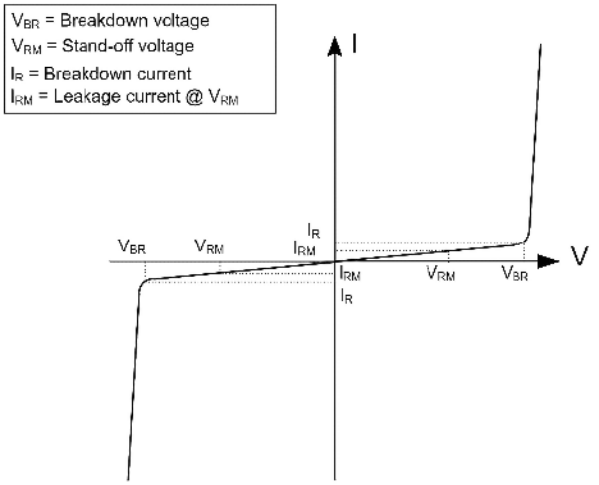
Rysunek 11. Charakterystyka I(U) jednokierunkowej diody TVS
Źródło: nota aplikacyjna ST Microelectronics, nr AN5612

o dużym natężeniu. W połączeniu ze zróżnicowanymi czasami narastania impulsów otrzymujemy zatem mieszankę wybuchową, przed którą musimy skutecznie ochronić nasze urządzenie. Złożona natura zjawisk ESD sprawia, że w praktyce stosowany jest szereg rozwiązań mających na celu ochronę linii I/O oraz zasilania przed skutkami wyładowań.

Techniki pasywne

Najprostszymi elementami chroniącymi (ale tylko w bardzo ograniczonym zakresie!) przed uszkodzeniami ESD są rezystory (włączone szeregowo) oraz kondensatory (włączone równolegle między masę a wejście lub wyjście urządzenia). Te pierwsze pozwalają skutecznie ograniczyć wartość prądu szczytowego, a dodatkowo chronią przed przepływem zbyt dużego prądu stałego, np. w przypadku współpracy linii I/O mikrokontrolera z układem zasilanym napięciem znacznie przewyższającym napięcie samego procesora. Kondensatory z kolei są w stanie „pochłonąć" nawet relatywnie spory ładunek, przez co przyczyniają się do ograniczenia efektywnego skoku napięcia podczas wyładowania – ograniczenie to jest tym skuteczniejsze, im większą mamy wartość pojemności kondensatora w stosunku do pojemności źródła.

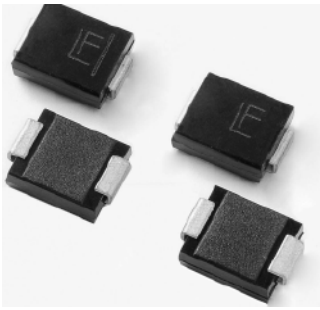
Niestety, obydwie opisane techniki znacząco wydłużają czasy narastania i opadania sygnałów, przez co zupełnie nie nadają się do stosowania np. w szybkich interfejsach komunikacyjnych. Z tego też względu ich praktyczna przydatność jest ograniczona głównie do wejść analogowych bądź linii cyfrowych o wybitnie niskiej częstotliwości przełączania.



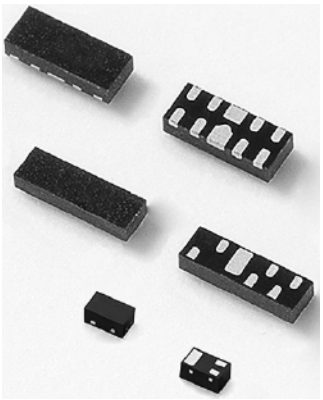
Rysunek 12. Charakterystyka I(U) dwukierunkowej diody TVS
Źródło: nota aplikacyjna ST Microelectronics, nr AN5612

Diody przeciwprzepięciowe (TVS)

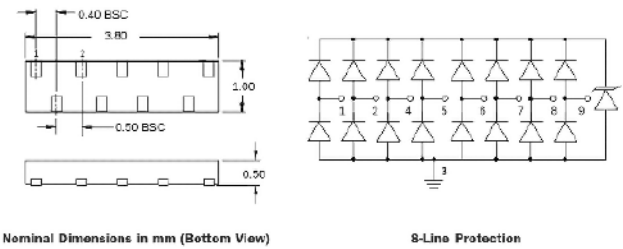
O transilach, zwanych w skrócie TVS (ang. Transient Voltage Suppressor), pisaliśmy już w tym artykule wielokrotnie – i nie jest to przypadek, wszak elementy te stanowią najpowszechniej stosowane ochronniki przeciwprzepięciowe we współczesnej elektronice. W zależności od potrzeb można używać zarówno transili jedno-, jak i dwukierunkowych. Te pierwsze znacznie lepiej nadają się do ochrony linii w układach zasilanych napięciem niesymetrycznym, gdyż – włączone w kierunku zaporowym – ograniczają potencjał do określonej wartości w przypadku polaryzacji zgodnej z normalną polaryzacją napięć w układzie (zwykle do kilku… kilkudziesięciu woltów, zależnie od napięcia zasilania), zaś w przypadku pojawienia się sygnału polaryzacji przeciwnej (np. napięcia ujemnego na wejściu cyfrowym) zwierają sygnał już na poziomie napięcia przewodzenia (czyli zwykle około 1…2 V) – patrz **rysunek 11**. Transile dwukierunkowe mają natomiast charakterystykę, której kształt nazwalibyśmy środkowosymetrycznym – w obydwu kierunkach krzywa I(U) ma ten sam przebieg (**rysunek 12**).



Fotografia 7. Przykładowe dyskretne diody TVS z serii 3.0SMCJ (http://t.ly/LnvTx)



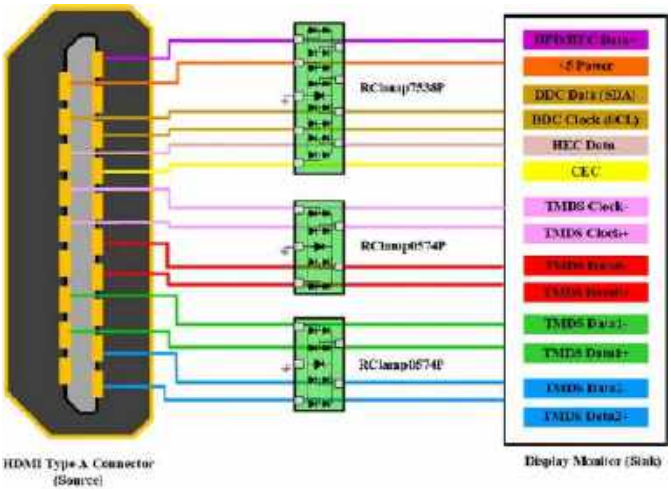
Fotografia 8. Przykładowe scalone macierze diod TVS (http://t.ly/BfF_R)



Rysunek 13. Wymiary oraz schemat wyprowadzeń układu RClamp7538P (http://t.ly/kHPrQ)

REKLAMA

eprasa.pl 4202fe84f2

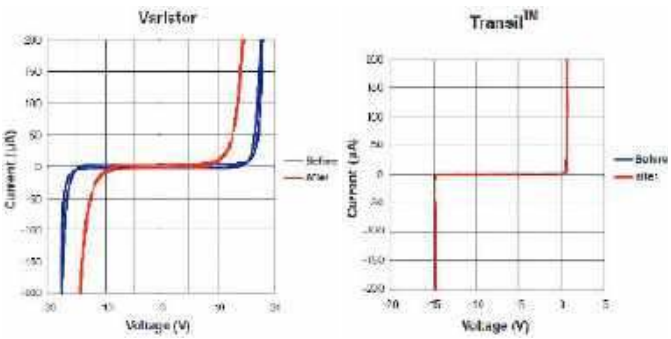


Rysunek 14. Zastosowanie układu RClamp7538P do ochrony portu HDMI (http://t.ly/CpTFW)

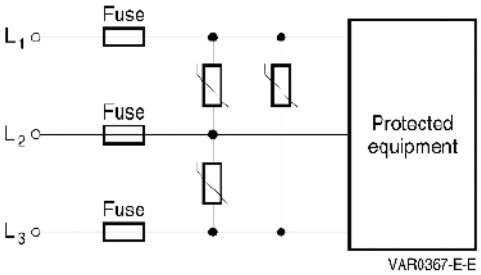
Transile są dostępne w niezwykle szerokiej gamie form i odmian. Oprócz elementów dyskretnych (**fotografia 7**), na rynku można znaleźć także rozmaite rozwiązania scalone (**fotografia 8**), w tym niektóre dostosowane specjalnie do potrzeb określonych aplikacji. Dobrym przykładem może być układ RClamp7538P (**rysunek 13**) opracowany z myślą o ochronie portów HDMI – nietypowe rozmieszczenie wyprowadzeń pozwala z łatwością poprowadzić ścieżki od złącza do ochronników TVS oraz od TVS do procesora (**rysunek 14**).

Warystory

Warystory to proste elementy półprzewodnikowe o nieliniowej charakterystyce prądowo-napięciowej – ich rezystancja gwałtownie spada po osiągnięciu określonego progu napięcia, choć nachylenie charakterystyki nie jest już tak duże, jak ma to miejsce w przypadku diod TVS. I to jest jeden z głównych powodów, dla których warystory niezbyt dobrze nadają się do ochrony linii sygnałowych. Elementy te mają także większy prąd upływu i znacznie wyższą pojemność w porównaniu z nowoczesnymi ochronnikami TVS, zaś w stanie „otwarcia" spadek napięcia pomiędzy ich zaciskami jest większy niż w przypadku analogicznych transili. Mało tego – warystory znacznie szybciej ulegają efektom starzenia, przez co ich przydatność w warunkach powtarzalnych przepięć jest istotnie ograniczona (**rysunek 15**). Zaletą warystorów (szczególnie tych w obudowach przewlekanych – **fotografia 9**) pozostaje zdolność przetrwania ogromnych przeciążeń (na poziomie kilku tysięcy amperów), stąd elementy te znajdują zastosowania przede wszystkim w obwodach sieciowych, w których pełnią funkcje ochrony przed udarami (**rysunek 16**). W handlu dostępne są jednak



Rysunek 15. Porównanie starzenia warystorów i diod TVS w warunkach powtarzalnych przepięć (niebieski – krzywe I(U) przed narażeniem, czerwony – krzywe I(U) po serii 100 impulsów 15 kV wg IEC61000-4-2 metodą kontaktową). Źródło: nota aplikacyjna ST Microelectronics, nr AN1826



Rysunek 16. Typowy układ przeciwprzepięciowy na bazie warystorów – wersja stosowana w urządzeniach trójfazowych (http://t.ly/D230O)

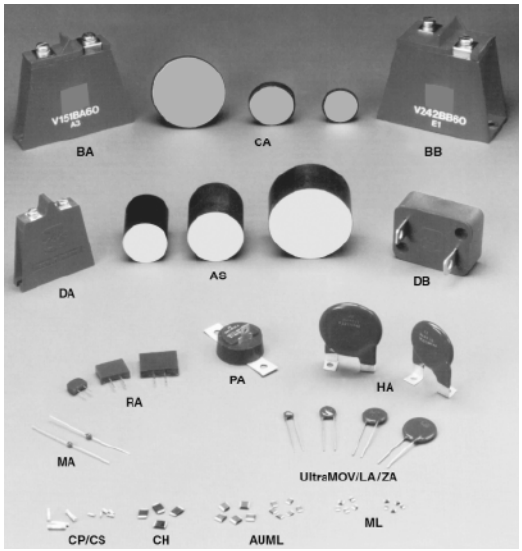
także mniejsze warystory do montażu powierzchniowego (**fotografia 10**), choć w zastosowaniach dużej mocy można znaleźć rozmaite wykonania w formie dysków lub modułów z zaciskami śrubowymi (**fotografia 11**) – największe z nich mogą znieść udary prądowe na poziomie 100 kA, przechwytując energię rzędu 12 kJ.

Fotografia 9. Przykładowy warystor do montażu przewlekanego (http://t.ly/VzGUP)

Gazowe elementy wyładowcze (GDT)

Ostatnia kategoria komponentów przeciwprzepięciowych, którą koniecznie należy przywołać, to gazowe elementy wyładowcze (ang. Gas Discharge Tube, GDT), znane także pod nazwą surge arrestors. Elementy te, jak sama nazwa wskazuje, są przeznaczone

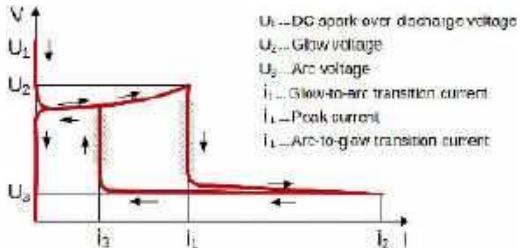
Fotografia 10. Miniaturowe warystory do montażu SMT (http://t.ly/grtoR)



Fotografia 11. Różne rozdaje warystorów (http://t.ly/y4egV)

eprasa.pl 4202fe84f2

Fotografia 12. Różne rozdaje elementów GDT (http://t.ly/ZyyY-)

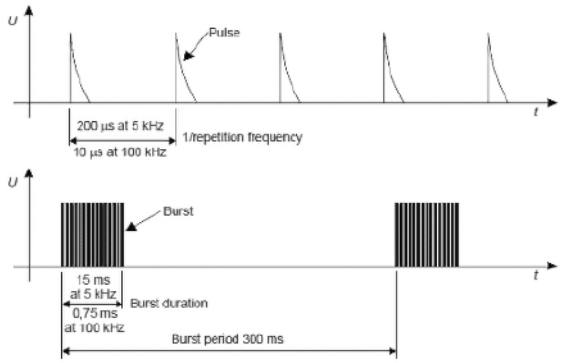


Rysunek 17. Charakterystyka napięciowo-prądowa elementu GDT (http://t.ly/ZyyY-)

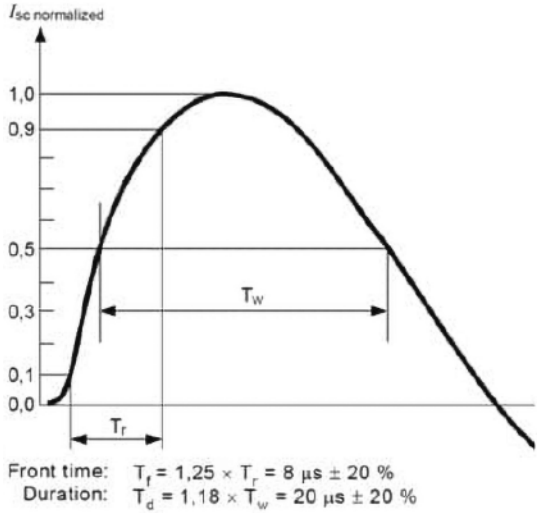
do tłumienia dużych impulsów napięciowych, a ich działanie opiera się na łuku elektrycznym powstającym pomiędzy elektrodami, umieszczonymi w hermetycznie zamkniętej komorze wypełnionej odpowiednim gazem. Forma konstrukcyjna może (ale nie musi) przypominać puszkę z osiowo wyprowadzonymi elektrodami (czasem budowa przypomina do złudzenia zwykły kondensator SMD – przykłady można zobaczyć na **fotografii 12**). Charakterystyka napięciowo-prądowa GDT jest dość rozbudowana (**rysunek 17**), co wynika z natury zjawiska łuku elektrycznego – niestety, duża zdolność gaszenia przepięć okazuje się w niektórych przypadkach przekleństwem, gdyż raz włączony element GDT może pozostawać w stanie niskiej rezystancji przez czas dłuższy, niż powinien – co niechybnie doprowadzi do zniszczenia jego samego, a nierzadko także innych części urządzenia. Stąd też gazowe elementy wyładowcze współpracują często z innymi komponentami, np. warystorami lub transilami. Niezwykle istotną zaletą GDT, którą zdecydowanie należy podkreślić, jest ich mała pojemność (znacznie poniżej 1 pF) i mała upływność, a co za tym idzie – praktycznie zerowy wpływ na normalne działanie większości układów, nawet w zakresie częstotliwości radiowych.

ESD a inne normy „przepięciowe"

Tematyka ESD przeplata się z ochroną przed innymi zdarzeniami natury przepięciowej, a te zależą już od konkretnej aplikacji. Doskonałym przykładem mogą być systemy automatyki przemysłowej (ale nie tylko), w których uwzględnić należy ryzyko wystąpienia przepięć oraz powtarzalnych serii szybkich elektrycznych stanów przejściowych, tzw. EFT (Electrical Fast Transient) – jeżeli dana linia wejściowa, wyjściowa lub zasilająca ma spełniać wymogi odporności na tego typu zdarzenia, to dobór transila bądź innego ochronnika musi uwzględniać diametralnie inną specyfikę przebiegów występujących podczas testu. Poświęcona zdarzeniom EFT norma o numerze 61000-4-4 także określa cztery poziomy odporności, przy czym napięcie szczytowe waha się od 250 V do 4 kV, zaś częstotliwość powtarzania impulsów wynosi przeważnie 5 kHz lub 100 kHz – przebiegi stosowane w badaniach wg 61000-4-4 pokazano na **rysunku 18**. Z kolei norma 61000-4-5 dotyczy ochrony przed skutkami udarów występujących głównie na skutek przepięć łączeniowych lub wyładowań atmosferycznych – impulsy charakteryzują się znacznie dłuższym czasem trwania (8/20 µs – patrz **rysunek 19**) i wiążą się z transferem o wiele większej energii. Stąd też w notach katalogowych transili, warystorów oraz elementów GDT można znaleźć liczne odwołania do impulsów typu 8/20 µs, zaś niczym zaskakującym nie są moce szczytowe na poziomie setek watów lub pojedynczych kilowatów (nawet w przypadku kompaktowych komponentów SMD).



Rysunek 18. Specyfikacja przebiegów typu burst wg normy 61000-4-4 (http://t.ly/sLiqq)



Rysunek 19. Specyfikacja impulsu testowego 8/20 wg normy 61000-4-5 (edycja 3) – http://t.ly/xGW3H

Podsumowanie

W artykule przyjrzeliśmy się bardzo istotnym zagadnieniom z zakresu ochrony przeciwprzepięciowej komponentów i układów elektronicznych. Dogłębne zrozumienie tematyki normalizacyjnej, metod testowania oraz wymogów branżowych pozwala uniknąć wielu błędów, które (na szczęście dla użytkowników, a zarazem na nieszczęście dla producentów) – wychodzą na jaw najczęściej już na etapie pierwszych „strzałów" z pistoletu ESD w trakcie badań kompatybilności elektromagnetycznej bądź podczas próby udaru napięciowego. Artykuł nie wyczerpuje rzecz jasna tej tematyki od strony aplikacyjnej – wiele aspektów projektowych, istotnych z punktu widzenia konstruktora-praktyka, omówimy na łamach „Elektroniki Praktycznej" przy innej okazji.

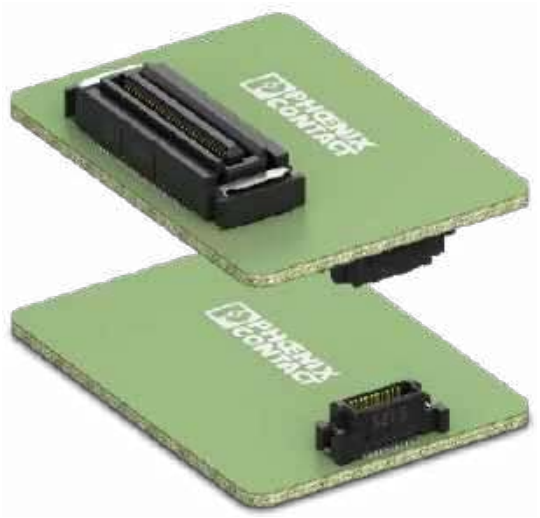
inż. Przemysław Musz, EP

REKLAMA

eprasa.pl 4202fe84f2

Złącza pływające ze znaczną kompensacją tolerancji do szybkiej transmisji danych

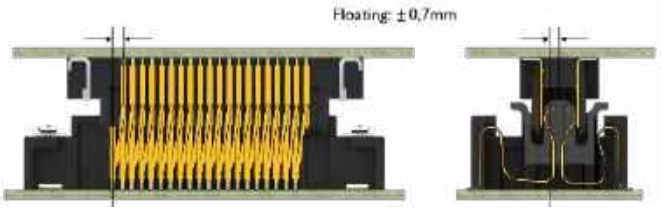
Złącza typu płytka-płytka z serii FINEPITCH firmy Phoenix Contact to ekranowane i nieekranowane rozwiązania przeznaczone do przesyłania sygnałów oraz danych. Pozwalają na realizację połączeń w indywidualnych orientacjach płytek PCB, we wszystkich kierunkach i przy zastosowaniu różnych konstrukcji, wysokości piętrowania oraz liczby pinów w rastrach od 0,635 mm do 2,54 mm. Możliwe są połączenia prostopadłe, równoległe oraz karty typu matka-córka. Złącza z serii FINEPITCH FS 0,635 o właściwościach pływających są nowym dodatkiem do portfolio Phoenix Contact. Co sprawia, że złącza pływające płytka-płytka są tak wyjątkowe? Gdzie są stosowane? I czy szybka transmisja danych stoi w opozycji do konstrukcji pływającej?



Cechy złącza pływającego

Złącza płytka-płytka składają się ze styków (które tworzą połączenie elektryczne) oraz obudowy (która je mocuje i chroni). Przy zastosowaniu tradycyjnego połączenia dwóch odpowiadających sobie złączy otrzymujemy sztywne połączenie płytka-płytka. Z kolei złącza pływające składają się z dwóch części obudowy, połączonych ze sobą za pomocą metalowych styków. Wewnętrzna część obudowy zawiera zestaw kontaktów i jest otoczona zewnętrzną obudową, która chroni powierzchnie lutownicze i styki. Pływające styki są długie i zakrzywione. Ze względu na swój kształt pozwalają one wewnętrznej części obudowy ze stykami poruszać się w granicach określonych przez zewnętrzną część obudowy. Ruchome styki działają jak elementy sprężynujące: możliwe jest ich przesuwanie i przechylanie w złączu i właśnie ta funkcja nazywana jest „pływającą".

Rysunek 1 pokazuje przekrój połączenia pływającego. Dolna płytka drukowana wyposażona jest w pływające gniazdo żeńskie z zakrzywionymi i ruchomymi stykami. Sztywny terminal męski przylutowany został do górnej płytki drukowanej. Złącze męskie jest przesunięte w prawo, co powoduje odchylenie styków pływających. Powierzchnie lutowania obu złączy podlegają niewielkim naprężeniom, gdyż pozostają nieruchome względem PCB – a wszystko dzięki specjalnej geometrii styków, umożliwiającej względny ruch złącza w przód i w tył w bardzo dużym zakresie tolerancji. Możliwe przesunięcie to aż ±0,7 mm w kierunku X i Y (**rysunek 2**). Oznacza to, że odległość tolerancji złącza jest większa niż jego raster.



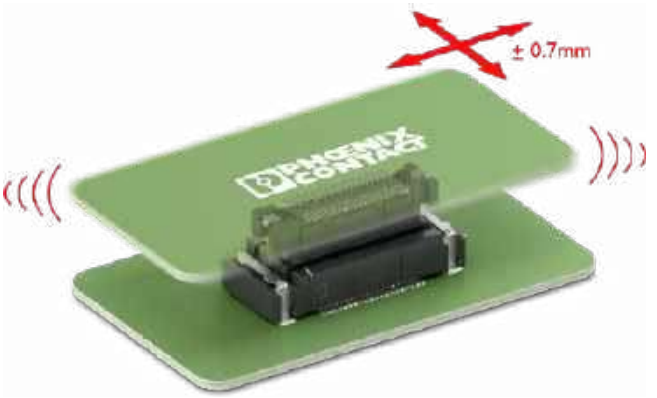
Rysunek 1. Pływające FS 0.635. Widok przekroju

Ponadto seria FS 0,635 zapewnia również tolerancję 0,6 mm w kierunku wtykania.

Pływające żeńskie gniazdo FS 0,635 można łączyć z trzema męskimi złączami o różnych wysokościach, otrzymując różne wysokości stosu – wynoszą one od 8 do 8,6 mm, a także od 11 do 11,6 mm i od 12 do 12,6 mm. Takie wartości ułatwiają znalezienie odpowiedniego rozstawu płytek drukowanych do danego zastosowania. Jeśli użyta zostanie sztywna kombinacja serii FS 0,635, można uzyskać wysokość stosu od 6 do 16,6 mm.

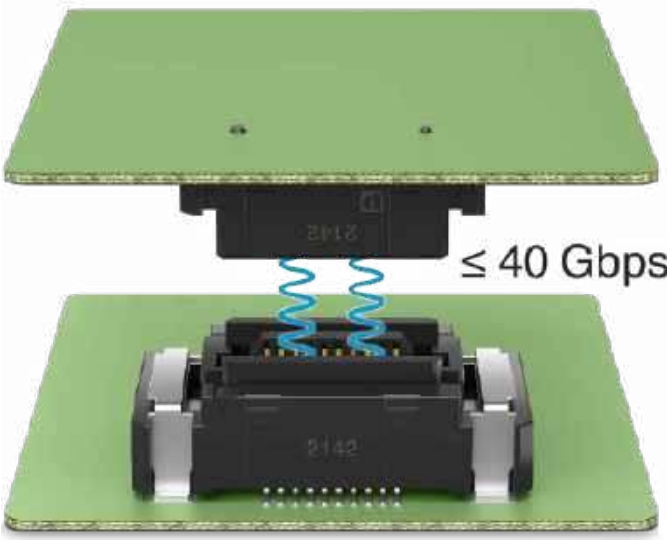
Zalety i zastosowania rozwiązań pływających

Ruchoma część gniazda gwarantuje bardzo wysoką kompensację tolerancji i pozwala na zminimalizowanie naprężeń mechanicznych w połączeniu. Jest to zaleta zarówno podczas łączenia, jak już po podłączeniu, w warunkach docelowej eksploatacji. Siły działające na złącze oraz połączenia lutowane są redukowane przez sprężynujące działanie styków pływających – co skutkuje wyjątkowo niezawodnym i długotrwałym, stabilnym połączeniem płytka-płytka. Pływające styki umożliwiają kompensację tolerancji produkcyjnych, które występują nieuchronnie podczas montażu płytek



Rysunek 2. Zakres ruchu złącza pływającego

eprasa.pl 4202fe84f2



Rysunek 3. Unikalne rozwiązania konstrukcyjne zapewniają szybkość transmisji danych do 40 Gb/s

drukowanych. Urządzenia, w których w procesie łączenia płytki drukowane używają dodatkowego prowadzenia, korzystają z dużych tolerancji kompensujących niewspółosiowość podczas zestawiania złączy. To z kolei przekłada się na mniejszą liczbę awarii w procesie montażu i umożliwia optymalizację szybkości podłączania. Oznacza to, że pływające złącze ma bezpośredni wpływ na wydajność procesu instalacji.

Złącze pływające pozwala ponadto na stosowanie wielu połączeń płytka-płytka pomiędzy dwiema płytkami drukowanymi. Właściwości pływające zapewniają wystarczającą swobodę, aby skompensować tolerancje położenia występujące podczas pozycjonowania i lutowania złączy. Sprężynowy styk minimalizuje siłę wywieraną na połączenia lutowane, co pozwala na niezawodne korzystanie z wielu złączy w ramach tego samego obwodu drukowanego.

Dodatkową zaletą złączy pływających jest ich odporność zarówno na wibracje lub wstrząsy – spotykane w urządzeniach przemysłowych narażonych na stałe naprężenia mechaniczne – jak i względne ruchy złączy spowodowane efektami termicznymi.

Szybka transmisja danych

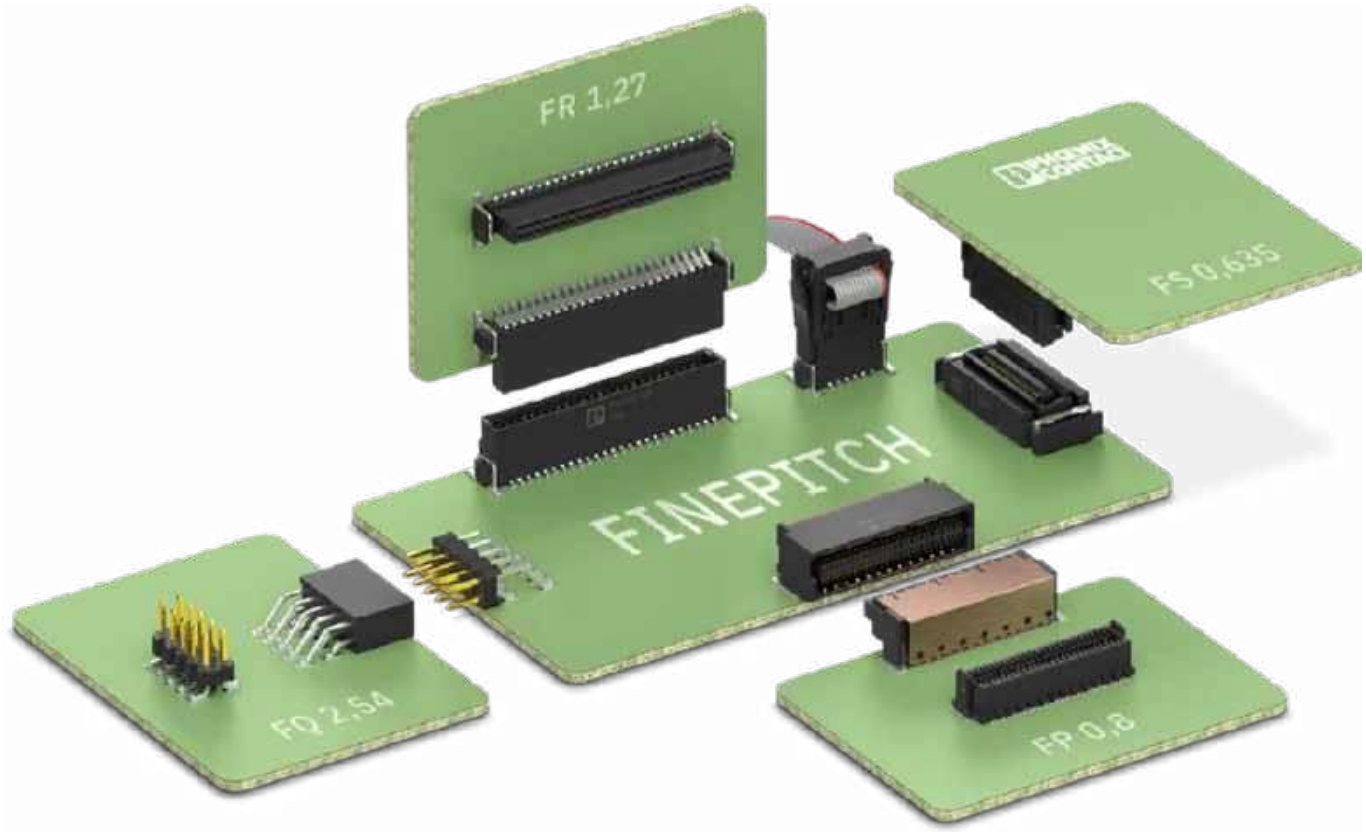
Integralność sygnału przesyłanego przez złącza płytka-płytka zależy od wielu czynników – kluczowym spośród nich staje się geometria styków. Mogłoby się wydawać, że zakrzywione, sprężynowe styki pływających złączy płytka-płytka nie są w stanie zapewnić dobrych wyników w zakresie szybkiego transferu danych. Jednak w serii FS 0,635 styk pływający został zoptymalizowany tak, aby gwarantował bardzo dobre właściwości przesyłu sygnałów (**rysunek 3**) – pozwala on osiągnąć prędkość aż do 40 Gb/s, co stanowi prawdziwą innowację w tym segmencie rynku.

Podsumowanie

Złącza pływające to niezawodny i wydajny sposób łączenia dwóch płytek drukowanych. Są solidne i trwałe, dzięki czemu idealnie nadają się do użytku w trudnych warunkach. Tworzą niezawodne połączenie między płytkami drukowanymi, kompensując niewielkie ruchy względne, a tym samym zmniejszając siłę wywieraną na połączenia lutowane. Z jednej strony funkcja ta jest pomocna w przypadku korzystania z wielu złączy na tej samej płytce drukowanej, a z drugiej strony – ułatwia automatyczne i ręczne zestawianie połączenia płytka-płytka.

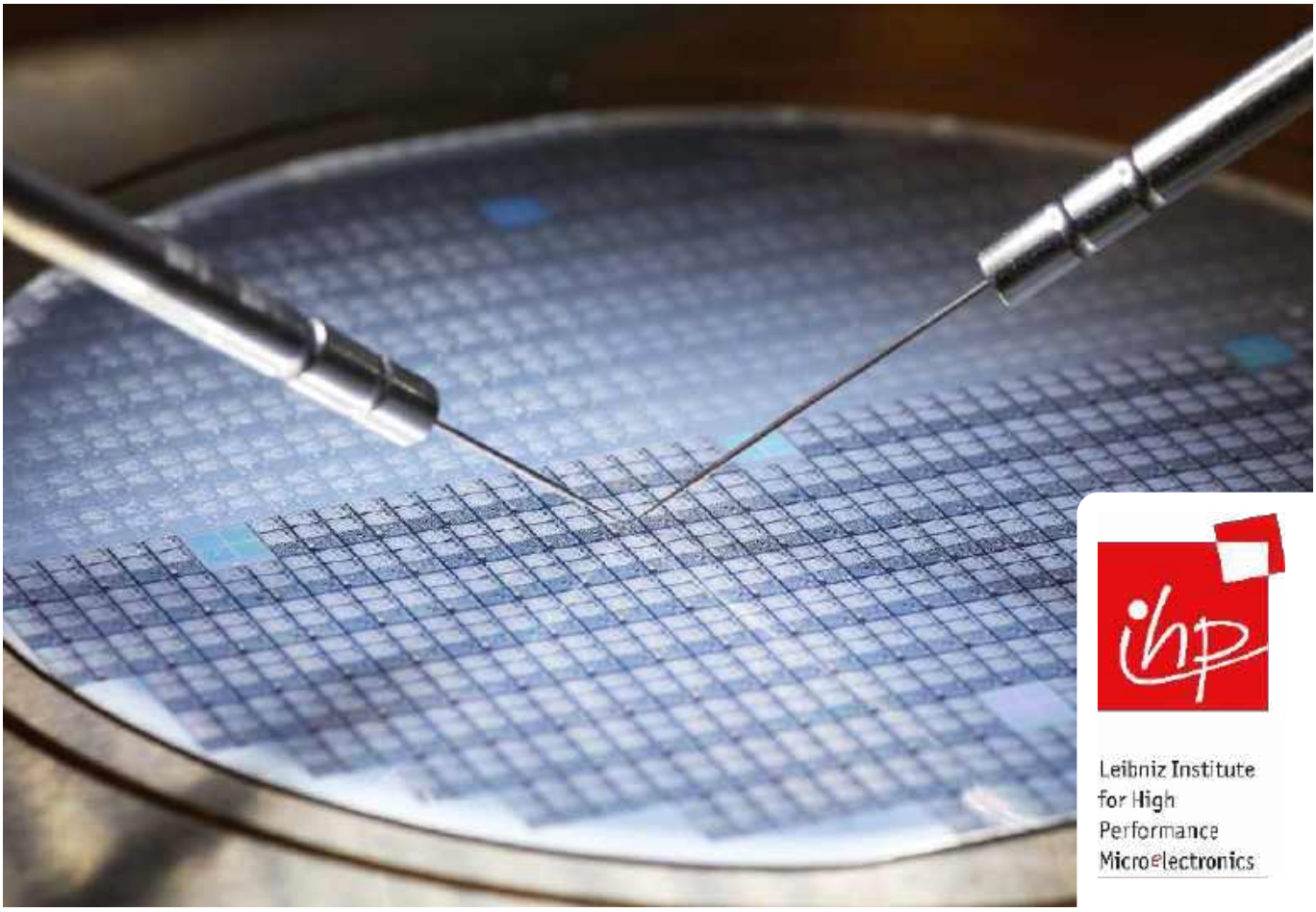
Szczególnie imponujące właściwości transmisji danych złącza pływającego serii FS 0,635 są unikalne na rynku. Pływający styk został zoptymalizowany do tego celu, dzięki czemu osiąga wyjątkową przepustowość na poziomie 40 Gbps, a to otwiera szeroki zakres nowych zastosowań złączy pływających (**rysunek 4**).

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



Rysunek 4. Seria wielopinowych złączy FINEPITCH to idealne rozwiązanie do przemysłowych połączeń PCB

eprasa.pl 4202fe84f2



Perspektywy rozwoju układów scalonych open source we współpracy z IHP Microelectronics

Po miesięcznej przerwie wracamy do Czytelników „Elektroniki Praktycznej" z tematyką open source silicon. Przerwa podyktowana była między innymi udziałem autorów w konferencji Free Silicon Conference [5], o której wspomnimy w dalszej części artykułu. W tym opracowaniu chcielibyśmy skupić się na perspektywach udziału zainteresowanych Czytelników w rozwoju otwartego sprzętu oraz możliwościach, jakie pojawiają się na gruncie europejskim w dziedzinie open source silicon.

Przypomnijmy, że nadrzędnym celem open source silicon jest obniżenie bariery wejścia na rynek dla projektantych firm projektowych, które chciałyby projektować, a także testować swoje układy mikroelektroniczne. Osiągnięcie tego celu może przełożyć się na wymierne efekty, w tym między innymi: lepszą edukację poprzez otwarty dostęp do wiedzy eksperckiej, zwiększenie liczby specjalistów w dziedzinie mikroelektroniki, wzrost liczby oraz znaczenia małych i średnich przedsiębiorstw czy też wzrost regionalnej niezależności technologicznej. Dla osiągnięcia wyżej wymienionego celu kluczowe okazują się takie aspekty, jak:

1. dostępność otwartych narzędzi EDA (Electronics Desing Automation),
2. upublicznienie bibliotek projektowych PDK (Process Design Kit) związanych z procesami technologicznymi,
3. opracowanie otwartych bibliotek projektów układów IP (Intellectual Property),
4. dostępność środków produkcji oraz laboratoriów pomiarowych na preferencyjnych warunkach,
5. zapisy licencyjne, które dają możliwość osiągania korzyści finansowych,

aczkolwiek warto nadmienić, że ta lista nie wyczerpuje wszystkich istotnych kwestii.

Od momentu upublicznienia IHP-Open-PDK [1] (luty 2023), IHP Micrelectronics [2] bierze aktywny udział w ewolucji każdego z wymienionych czynników.

W naszym poprzednim artykule – opublikowanym na łamach czerwcowego wydania EP – zaprezentowaliśmy kalendarz organizacji programów OpenMPW (Open Multi Project Wafer), które zapewniają darmowy dostęp do technologii SG13G2 każdemu, kto jest zainteresowany projektowaniem układów mikroelektronicznych i ich weryfikacją – oraz kto zaakceptuje warunki licencji open source (co wiąże się z upublicznieniem projektu użytkownika). Zwiększenie

eprasa.pl 4202fe84f2

liczby partycypantów programów OpenMPW przyczynia się bezpośrednio do rozwoju bibliotek projektowych i pozwala na stworzenie katalogu przetestowanych oraz udokumentowanych układów open source, które w przyszłości mogą być używane w bardziej złożonych rozwiązaniach. Takie podejście łamie całkowicie paradygmat projektowania układów scalonych stosowany w przemyśle. W przypadku nabycia narzędzi EDA, dostępu do komercyjnych PDK oraz bibliotek IP, zapisy licencyjne wykluczają możliwość upublicznienia rozwiązania (dokumentacji projektowej), a jeśliby nawet taki przypadek miał miejsce, to jego zastosowanie wiązałoby się z pokryciem kosztów dostępu do kompatybilnych programów, PDK oraz IP.

Nie zapominajmy, że w przypadku technologii SG13G2 biblioteki te po dziś dzień praktycznie nie istnieją. IHP dostarcza co prawda bibliotekę do syntezy układów cyfrowych, niemniej – jeśli chodzi o rozwiązania analogowe lub RF – pierwszych kilka układów scalonych jest w trakcie produkcji. W takiej sytuacji nawet mało doświadczony projektant jest w stanie dostarczyć w pełni funkcjonalny scalak, który może stanowić realizację projektu z podręcznika akademickiego lub innego powszechnie dostępnego źródła. Oprócz funkcjonalności rozwiązania (dowiedzionej zaprezentowanymi i udostępnionymi testami), na poziomie symulacji kluczowa jest także dokumentacja projektu, która w idealnym przypadku opisuje proces decyzyjny na etapie projektowania, dodając do niego walor edukacyjny. Nie mniej ważne jest dołączenie do dokumentacji wyników pomiarów przeprowadzonych w ramach testowania układu. Zawsze mile widziane są rozwiązania, które dostarczają możliwość adaptacji parametrów układu, tak aby użytkownik mógł je dostosować do potrzeb własnego projektu.

Jeden z potencjalnych oraz oczekiwanych rezultatów obniżenia bariery wejścia na szeroko rozumiany rynek półprzewodników stanowi zwiększenie liczby oraz znaczenia małych i średnich przedsiębiorstw w ujęciu globalnym. Historycznie rzecz ujmując, Europa nigdy nie specjalizowała się w masowej produkcji półprzewodników, a raczej wprost przeciwnie, dostarczała rozwiązania niszowe do partykularnych zastosowań. IHP ze swoją technologią SiGe:C jest tego doskonałym przykładem. Biorąc pod uwagę powyższe dwa stwierdzenia, można dojść do wniosku, że przyszłe małe i średnie europejskie firmy mogą dostarczać zaawansowane rozwiązania na rynek światowy. Niezbędne jest jednak stworzenie otoczenia prawnego, a konkretnie odpowiednich zapisów licencyjnych, które umożliwią czerpanie korzyści finansowych także w domenie open source. Za dobry przykład może posłużyć tu model podwójnego licencjonowania – warunki wyboru licencji są w nim zależne od celu, do którego używany będzie produkt. W przypadku celów niekomercyjnych aplikowana jest licencja typu open source np. APACHE 2.0, GPL v3, MIT, a w przypadku zastosowań komercyjnych – stosuje się tradycyjne metody licencjonowania produktu. W związku z tym, że biblioteki projektowe oraz zasady ich udostępniania są kluczowym aspektem rozwoju open source silicon, aktualnie toczy się szeroka debata na temat ww. zasad licencjonowania. Wiele dyskusji na temat przyszłości open source silicon odbyło się na konferencji FSiC w Paryżu pod koniec czerwca 2024 r.

Sama konferencja poprzedzona została jednodniowymi warsztatami, których celem było rozpoczęcie procesu opracowywania mapy rozwoju open source hardware na gruncie europejskim. Dyskutowano między innymi na tematy techniczne, biznesowe oraz organizacyjne, a dopracowanie szczegółów i kontynuację dyskusji wydelegowano do otwartych grup roboczych. Czytelnicy pragnący dowiedzieć się więcej na temat ostatnich nowinek ze świata open source silicon mogą zapoznać się ze wszystkimi materiałami z warsztatów oraz konferencji, łącznie z nagraniami wideo, na stronie [5].

Wszelkie dyskusje byłyby ograniczone w przypadku braku finansowania projektów open source hardware. Podczas jednego z warsztatów online prof. Boris Murmann – należący do grona ambasadorów open source silicon – stwierdził, że zjawisko to znajduje się aktualnie w okresie miesiąca miodowego, co przekłada się na wzbudzanie zainteresowania wynikające z efektu nowości. Powyższa sytuacja ma dość duże przełożenie na finansowanie, bowiem europejskie agencje finansujące projekty badawcze szerokim gestem wspierają projekty związane z open source hardware.

W bieżącym roku w samym IHP wystartowały projekty, takie jak: FLOWSPACE (opracowanie open-sourcowej i odpornej na promieniowanie wersji procesu SG13G2 do zastosowań w kosmosie), DEMICO (opracowanie zaawansowanych i otwartych projektów w paśmie fal milimetrowych z użyciem narzędzi open source oraz IHP-Open-PDK),

Di-Sign-HEP (opracowanie otwartych modułów kryptograficznych w domenie open source), OCDCPro (opracowanie materiałów dydaktycznych w dziedzinie mikroelektroniki, skoncentrowanych wokół open source silicon). W dokumencie [6] programu Unii Europejskiej „Chips Joint Undertaking" – opisującym perspektywy alokacji środków finansowych w centra kompetencji, platformy projektowe, linie pilotażowe, układy fotoniczne – stwierdzenie „open source" pojawia się kilkanaście razy w kontekście otwartych narzędzi EDA, otwartych PDK oraz bibliotek IP, co można uznać za informację o tym, że organy decyzyjne na poziomie unijnym biorą pod uwagę open source silicon jako obszar, który warto rozwijać. Kolejną ciekawą opcję finansowania stanowi fundacja NlNet [7], żywo zainteresowana rozwojem open source silicon. Oferta NlNet adresowana jest głównie do osób prywatnych oraz małych firm i ma za zadanie wspierać rozwój innowacji w jej początkowym stadium. Mimo że środki przyznawane przez fundację nie są wygórowane, to wielką zaletą ich modelu okazuje się minimalizacja wysiłku, który należy włożyć w procedury administracyjne. Projekty realizowane wspólnie z IHP, dotyczące wszystkich wymienionych w tym artykule aspektów, są jak najbardziej pożądane.

Nasza oferta współpracy kierowana jest także do studentów wszystkich stopni kształcenia. IHP zapewnia trzymiesięczne płatne praktyki, które można łączyć z programem Erasmus. Co więcej, również studenci zainteresowani realizacją swoich prac licencjackich, magisterskich lub doktorskich mogą skorzystać z możliwości rozwoju we współpracy z IHP. Dodatkowym atutem jest fakt położenia IHP przy granicy z Polską oraz świetnego skomunikowania z większością ośrodków akademickich, takich jak: Warszawa, Kraków, Wrocław, Poznań, Łódź, Katowice, Zielona Góra Rzeszów oraz Gdańsk.

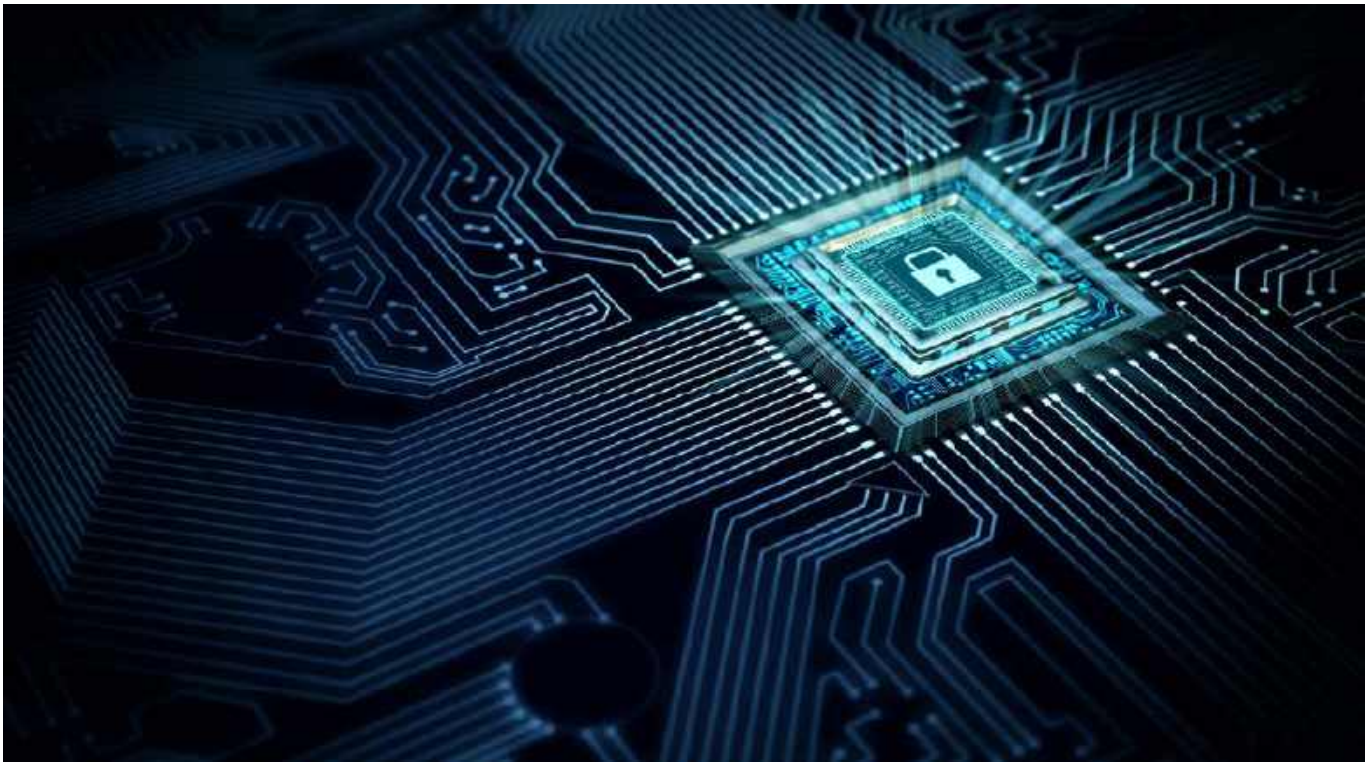
Nawiązując do powyższej inicjatywy, zapraszamy wszystkie osoby zainteresowane współpracą do zapoznania się z harmonogramem naszych programów OpenMPW [3] oraz do kontaktu z nami poprzez e-mail lub kanał slack [4].

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:

1. https://www.ihp-microelectronics.com/
2. https://github.com/IHP-GmbH/IHP-Open-PDK
3. https://ihp-open-ip.readthedocs.io/
4. https://open-source-silicon.slack.com
5. https://wiki.f-si.org/index.php?title=Main_Page
6. Chips Joint Undertaking Appendix 4 for the initiative part
7. https://nlnet.nl

eprasa.pl 4202fe84f2



„Bezpieczeństwo przede wszystkim" w projektowaniu urządzeń IoT

Urządzenia IoT stają się coraz częstszym celem cyberataków. Na szczęście istnieje rozwiązanie tego problemu – PSA Certified IoT Security Framework pomaga projektantom tworzyć produkty wyposażone w silniejsze mechanizmy obronne.

Współczesna technologia IoT nie jest tak bezpieczna, jak mogłaby być – a wynika to z faktu, że wiele urządzeń opracowywanych jest bez uwzględnienia (już na samym początku procesu projektowego) zagadnień ochrony cybernetycznej. Zabezpieczenia są często wprowadzane pod sam koniec rozwoju produktu, a takie podejście typu „last minute" prowadzi do nieprzewidzianych luk w bezpieczeństwie, które z kolei mogą zostać łatwo wykorzystane przez atakujących. Zgodnie z raportem pt. *PSA Certified 2023 Security Report*, liczba cyberataków dokonanych na rozmaite urządzenia typu „smart" potroiła się względem roku 2021.

Naruszenia bezpieczeństwa IoT obejmują szeroki zakres zdarzeń – od utraty danych osobowych pozyskanych przez atakujących ze zhakowanego urządzenia noszonego, aż po zagrożenie zdrowia lub życia w przypadku zaatakowania urządzeń medycznych, np. rozruszników serca czy pomp insulinowych. Na szczęście branża elektroniczna nie stoi w miejscu. Kolejne przepisy prawne oraz normy są adaptowane do potrzeb rynku elektroniki, co tworzy solidną podstawę bezpieczeństwa systemów IoT. W wielu obszarach rynku ten fundament stanie się obowiązkowy. Podczas gdy z globalnego punktu widzenia podejście do bezpieczeństwa IoT jest niejako „rozdrobnione", to coraz częściej pojawiają się także ujednolicone wymagania. Przykładowo: istnieją podobieństwa między dyrektywą UE dot. urządzeń radiowych i unijnymi ustawami o cyberbezpieczeństwie (Cyber Security Act) oraz odporności cybernetycznej (Cyber Resilience Act) – a ich odpowiednikami obowiązującymi na terenie USA, Wielkiej Brytanii, Singapuru, Finlandii czy Australii.

Wymaganiem leżącym u podstaw wielu tych przepisów jest uwzględnianie bezpieczeństwa już na najwcześniejszych etapach projektowania. Jeden ze standardów, które pomagają programistom osiągnąć ten cel, stanowi certyfikat PSA. PSA Certified, na czele którego stoi dostawca IP w zakresie rdzeni procesorów – Arm – oraz kilka globalnych, niezależnych laboratoriów testowych oferuje ramy bezpieczeństwa IoT, które wdrażają zaufaną ochronę dla układów scalonych, oprogramowania, usług i produktów końcowych. Układy nRF5340 SoC i nRF9160 SiP firmy Nordic uzyskały certyfikat PSA Level 2, a nRF52840 – PSA Level 1.

> *Kluczowym wymaganiem leżącym u podstaw wielu międzynarodowych przepisów i norm jest uwzględnienie zagadnień bezpieczeństwa już na najwcześniejszych etapach projektowania produktu IoT.*

Bezpieczeństwo to równowaga

Ochrona kosztuje – a nawet przy wysokim budżecie nie sposób zagwarantować 100-procentowego bezpieczeństwa produktu. Jeśli cyberprzestępcy dysponują wystarczającą ilością czasu, pieniędzy i motywacji, ochrona każdego systemu IoT może zostać przez nich naruszona. Właśnie dlatego bezpieczeństwo to w istocie równowaga między kosztami a ryzykiem. Wdrożenie zabezpieczeń aplikacji

eprasa.pl 4202fe84f2

IoT wymaga specyficznych umiejętności inżynieryjnych, dodatkowych zasobów w zakresie wyspecjalizowanych układów scalonych oraz bezpiecznego środowiska produkcyjnego. Wszystkie te elementy generują kolejne koszty. W praktyce opisywane wydatki muszą być zrównoważone w odniesieniu do realnego ryzyka (przy czym ryzyko stanowi iloczyn wpływu udanego ataku na działanie systemu oraz prawdopodobieństwa, że zostanie on przeprowadzony). Przykładowo: atak będzie miał istotny wpływ, jeśli wiąże się z utratą poufnych danych osobowych, ale może wiązać się z nim również duże prawdopodobieństwo, jeżeli dane te są interesujące, a urządzenie – z natury łatwe do złamania. W takim przypadku należy rozważyć implementację solidnych środków bezpieczeństwa, aby zmniejszyć ryzyko, ponieważ koszt bezpieczeństwa okaże się w istocie niższy niż koszt udanego ataku.

Ale nawet produkty o niskim ryzyku nie powinny pozostawać bezbronne. Podstawowa ochrona, taka jak bezpieczny rozruch i aktualizacja przeprowadzana z użyciem mechanizmu anti-rollback, to dobry początek. Bezpieczny rozruch (secure boot) zapewnia, że urządzenie weryfikuje, czy jego oryginalne oprogramowanie i wszelkie aktualizacje są autoryzowane oraz bezpieczne do uruchomienia. Funkcja anti-rollback zapobiega przywróceniu starszej (i potencjalnie podatnej) wersji oprogramowania sprzętowego.

Wyższe bezpieczeństwo można osiągnąć, izolując obszary, które muszą być silniej zabezpieczone, od tych, które zawierają mniej wrażliwe informacje. Pozostawienie mniej krytycznych obszarów ze stosunkowo słabą ochroną zmniejsza ogólny koszt zabezpieczeń bez nadmiernego zwiększania ryzyka generowanego przez system. Informacje z bezpiecznego obszaru można pozyskać za pośrednictwem interfejsu API, ale wszystkie krytyczne informacje pozostaną niedostępne nawet dla API. Przykładowo: niezabezpieczone środowisko przetwarzania (NSPE) może korzystać z API do szyfrowania danych bez dostępu do kluczy szyfrujących lub podstawowych zasobów implementacji kryptograficznej bezpiecznego środowiska przetwarzania (SPE). Urządzenie może również zawierać bezpieczną pamięć do przechowywania krytycznych danych i zasobów.

Każdy produkt IoT powinien umożliwiać jednoznaczną identyfikację i dostarczyć dowody, że faktycznie jest urządzeniem, za które się podaje. Taka identyfikacja zapobiega dołączaniu nieautoryzowanych urządzeń do sieci i otwieraniu w ten sposób luk w zabezpieczeniach. Co więcej, każde urządzenie powinno implementować odpowiednie algorytmy kryptograficzne niezbędne do zapewnienia bezpiecznej komunikacji, bez ujawniania współpracującej z nim aplikacji takich danych, które mogłyby zostać wykorzystane przez atakującego. I wreszcie – ważne jest, aby bezpieczeństwo było brane pod uwagę na wszystkich etapach cyklu życia produktu, a nie dopiero wtedy, gdy jest już on eksploatowany. Równie ważne okazuje się zatem odpowiednie podejście do debugowania, uruchamiania czy konfiguracji.

Cztery kroki na drodze do bezpieczeństwa cybernetycznego

Z perspektywy projektanta bezpieczeństwo może być postrzegane jako „zabawa" skomplikowana i kosztowna. Dołączenie kwestii bezpieczeństwa do harmonogramu projektu jest w stanie nawet zagrozić wydłużeniem czasu wprowadzania produktu na rynek. Certyfikowany przez PSA framework bezpieczeństwa IoT rozwiązuje te problemy, oferując ujednolicone podejście do bezpieczeństwa IoT na wszystkich etapach – od projektu do certyfikacji.

Framework PSA upraszcza strategię bezpieczeństwa poprzez podzielenie jej na cztery etapy:

1. modelowanie zagrożeń i analizę bezpieczeństwa („analiza"),
2. specyfikację sprzętu i oprogramowania układowego („architektura"),
3. implementację kodu źródłowego oprogramowania układowego („implementacja"),
4. niezależne testowanie („certyfikacja").

Aby zilustrować działanie opisywanej struktury, rozważmy proces modelowania zagrożeń. Pierwszy krok stanowi zidentyfikowanie zasobów, które wymagają ochrony – są to takie elementy, jak klucze szyfrowania, certyfikaty identyfikacyjne i niektóre dane. Następnie programista powinien zastanowić się, w jaki sposób dane zasoby mogą zostać zaatakowane, na przykład poprzez atak na sieć lub bezpośrednio przez interfejs układu scalonego. Trzecim krokiem jest analiza ryzyka (iloczyn wpływu oraz prawdopodobieństwa). Elementy lub dane, które są najbardziej narażone na atak, to te, którym należy przyjrzeć się z największą uwagą. Ostatni krok to właściwa implementacja – czyli zastanowienie się, co dokładnie musi zrobić programista, aby zabezpieczyć produkt przed atakiem.

Projektant może na przykład uznać, że jego aktywami są: oprogramowanie układowe, dane uwierzytelniające oraz logi systemowe. W przypadku oprogramowania układowego wymaganiem bezpieczeństwa jest integralność, a zagrożeniem – manipulacja. Słabym punktem może być złośliwe oprogramowanie, błąd w firmware, luki w pamięci masowej, niezabezpieczony interfejs JTAG czy wreszcie brak odpowiednich zabezpieczeń podczas aktualizacji oprogramowania układowego urządzenia. Efektem wykorzystania którejkolwiek z tych luk jest możliwość zapisania w pamięci złośliwego oprogramowania, a następnie uruchomienia rozproszonego ataku typu DDoS. Jednym ze sposobów łagodzenia ryzyka takiego ataku byłoby wdrożenie bezpiecznego bootloadera. Takie oprogramowanie zapewnia, że nawet jeśli złośliwe oprogramowanie zostanie zainstalowane za pośrednictwem luki, to i tak nie będzie ono uruchomione.

Po zakończeniu modelowania zagrożeń deweloper musi określić sprzęt i oprogramowanie, które pozwolą zaimplementować obsługę pożądanych mechanizmów obronnych przed zagrożeniami zidentyfikowanymi na etapie wcześniejszym. Jeśli zatem modelowanie zagrożeń ujawni potrzebę zastosowania akceleratora kryptograficznego, bezpiecznego przechowywania i izolacji, odpowiednim rozwiązaniem może okazać się oparcie produktu na układach typu nRF9160 lub nRF5340 firmy Nordic, mających certyfikację PSA i skutecznie wdrażających najlepsze praktyki w tym zakresie.

Uzyskiwanie certyfikacji

Tworzenie produktów bazujących na układach z certyfikatem PSA ułatwia uzyskanie certyfikacji produktu końcowego. Istnieją trzy poziomy certyfikacji, obejmujące układy scalone, oprogramowanie systemowe i urządzenia końcowe. nRF9160 i nRF5340 firmy Nordic mają certyfikat PSA poziomu 2, a nRF52840 – poziomu 1, co pokazuje, że implementują one tzw. PSA-Root of Trust (RoT). Oprogramowanie systemowe (na przykład system operacyjny czasu rzeczywistego, RTOS) używa PSA-RoT do implementacji własnych zabezpieczeń. Programista może następnie opracować produkt końcowy przy użyciu certyfikowanego układu scalonego i oprogramowania systemowego.

Produkty końcowe zbudowane w oparciu o układy i oprogramowanie z certyfikacją PSA mogą dziedziczyć owe certyfikaty. Warto też dodać, że certyfikat PSA jest zgodny ze standardami i przepisami wymaganymi przez rynki międzynarodowe, w tym ETSI EN 303645 i NIST 8259A. Spełnienie norm branżowych stanowi nieocenioną pomoc podczas ubiegania się o certyfikację wyrobu końcowego, zmniejsza narzut inżynieryjny i skraca czas wprowadzania produktów na rynek.

Podsumowanie

Ochronę urządzeń IoT przed złośliwymi atakami można zagwarantować tylko poprzez uwzględnienie zabezpieczeń na najwcześniejszych etapach projektowania. Certyfikat PSA zapewnia, że układy marki Nordic zawierają sprawdzone i zaufane mechanizmy ochronne, dzięki czemu projektanci mogą łatwiej zapewnić niezbędny poziom bezpieczeństwa swoich urządzeń.

www.nordicsemi.com

eprasa.pl 4202fe84f2

Internet Rzeczy w pomiarach środowiskowych (8)

Czujnik wielospektralny AS7341 firmy ams-OSRAM

Firma ams-OSRAM oferuje czujniki spektralne do aplikacji z zakresu wykrywania kolorów oraz analizy widmowej. Układy scalone z tej grupy znajdują wiele zastosowań, takich jak wybieranie kolorów, uwierzytelnianie i analiza widmowa substancji, materiałów, żywności czy płynów. Systemy te są używane w wielu obszarach, w szczególności w sektorach: konsumenckim, przemysłowym i medycznym.

Na przestrzeni lat proste czujniki światła otoczenia (ALS, Ambient Light Sensor) otrzymały najpierw kanał bliskiej podczerwieni (NIR, Near Infrared), a następnie zostały wyposażone w filtry zapewniające podział światła widzialnego na trzy kanały: czerwony (R, red), zielony (G, green) i niebieski (B, blue). Kolejnym etapem rozwoju tych komponentów było opracowanie filtrów interferencyjnych i skonstruowanie czujników XYZ oraz wielospektralnych. Zostały one omówione w artykułach: „Systemy dla Internetu Rzeczy (45) – Czujniki koloru" [2], a także „Czujniki optyczne (4). Sensory koloru oraz czujniki i moduły multispektralne" [8].

Spektroskopia optyczna jest zwykle kojarzona z drogimi urządzeniami spektrometrycznymi, których cena sięga od dziesiątek do nawet setek tysięcy dolarów. Tymczasem firma ams-OSRAM, lider technologii czujników analogowych i optycznych, sprowadziła spektroskopię do skali... układu scalonego. Rodzina czujników wielospektralnych tej firmy obejmuje szeroką gamę sensorów zapewniających od 6 do 18 kanałów w widmie widzialnym i bliskiej podczerwieni. Wszystkie układy wyposażone zostały w filtry wąskopasmowe umieszczone bezpośrednio na fotodiodach, co pozwala na uzyskanie precyzyjnej charakterystyki widmowej. W większości z nich uwzględniono ponadto bezpośrednie sterowniki LED do synchronizacji i sterowania źródłami światła, dzięki czemu mogą one sprawić, że zewnętrzne źródło światła będzie pracowało wydajnie [1].

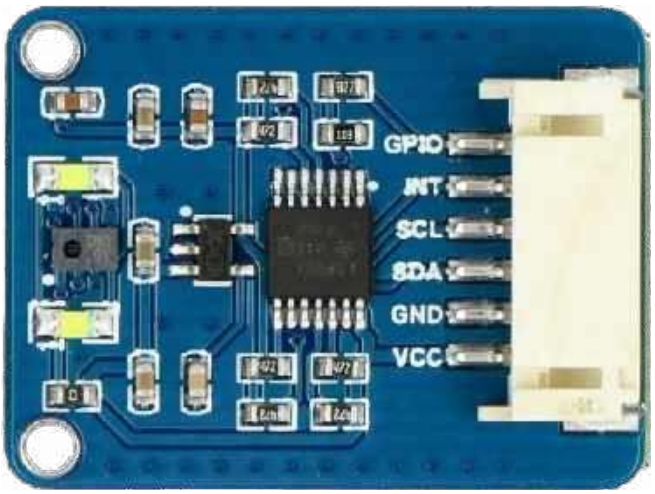
AS7341 to 11-kanałowy czujnik wielospektralny do zastosowań związanych z wykrywaniem kolorów i analizą widmową. Układ AS7343 ma bardzo podobną budowę oraz dodatkowe kanały XYZ. Oferowane są też wersje AS7341L i AS7343L zaprojektowane specjalnie do pomiarów światła odbitego w blokach pomiarowych aparatury medycznej. Rodzina AS7265x zawiera 3 układy scalone, każdy po 6 kanałów pomiarowych – zastosowanie ich w ramach pojedynczego systemu umożliwia pomiar w 18 kanałach, obejmujących długości fal od bliskiego nadfioletu do bliskiej podczerwieni (410 nm do 940 nm), z szerokością kanałów FWHM równą 20 nm. Dokładniejszy opis znajduje się w artykule [2].

Rozwiązania ams-OSRAM w zakresie czujników spektralnych w skali chipowej umożliwiają projektantom wdrażanie szerokiej gamy nowych aplikacji, takich jak: zaawansowane wykrywanie koloru, wykrywanie rodzaju dymu, optymalizacja w ogrodnictwie czy też uwierzytelnianie oraz analiza kolorów i widma rozmaitych materiałów, w tym płynów.

Układ AS7341 firmy ams-OSRAM

Główne cechy układu AS7341 [4]:

- 8 kanałów optycznych rozmieszczonych w zakresie widzialnym,



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

- 3 dodatkowe kanały: Clear, NIR oraz Flicker,
- 6 przetworników ADC do przetwarzania sygnału,
- napięcie zasilania (min./typ./maks.): 1,7/1,8/2,0 V,
- napięcie na wyprowadzeniu LDR (abs. maks.) 3,6 V,
- prąd zasilania w stanie aktywnym (typ./maks.): 210/300 μA,
- prąd zasilania w stanie oczekiwania (typ./maks.): 35/60 μA,
- prąd zasilania w stanie uśpienia (typ./maks.): 0,7/5 μA,
- prąd upływu wyprowadzeń SCL, SDA, INT (min./maks.): –5/5 μA,
- zakres ustawień prądu na wyprowadzeniu LDR: 2…258 mA co 2 mA,
- interfejs: I²C,
- adres I²C: 0x39,
- zakres temperatur pracy: od –30°C do 85°C,
- zakres wilgotności roboczej: od 5% RH do 85% RH,
- wymiary: 3,1×2×1 mm.

Układ jest umieszczony w 8-nóżkowej obudowie. Oprócz zasilania i masy (dwie nóżki) ma wyprowadzenia szyny I²C (SCL i SDA), wejście GPIO, wyjście INT (otwarty dren) oraz analogowe wyprowadzenie LDR do sterowania prądem zewnętrznej diody LED.

Sterowanie i monitorowanie układu odbywa się za pomocą ośmiobitowych rejestrów dostępnych poprzez interfejs szeregowy I²C. Rejestry te zapewniają funkcje sterowania układem, można je również odczytywać w celu określenia stanu układu i pobrania z niego danych.

Wyprowadzenia GPIO można użyć jako wejścia do podłączenia zewnętrznych fotodiod lub jako wejścia synchronizacji do rozpoczęcia/zatrzymania pomiaru widma (tryb SYNS/SYND). Wyjście INT zgłasza przerwanie, ale może być również użyte do wskazania stanu (READY/BUSY) pomiaru widma w trybach SYNS i SYND.

Architektura układu

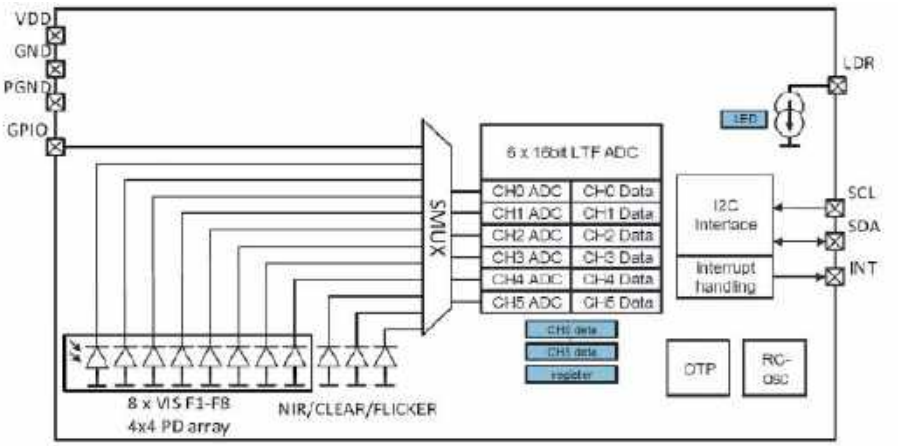
Układ AS7341 zawiera 6 niezależnych kanałów optycznych – każdy ze specjalnym, 16-bitowym przetwornikiem światło-częstotliwość. Wzmocnienie i czas integracji wszystkich sześciu

eprasa.pl 4202fe84f2

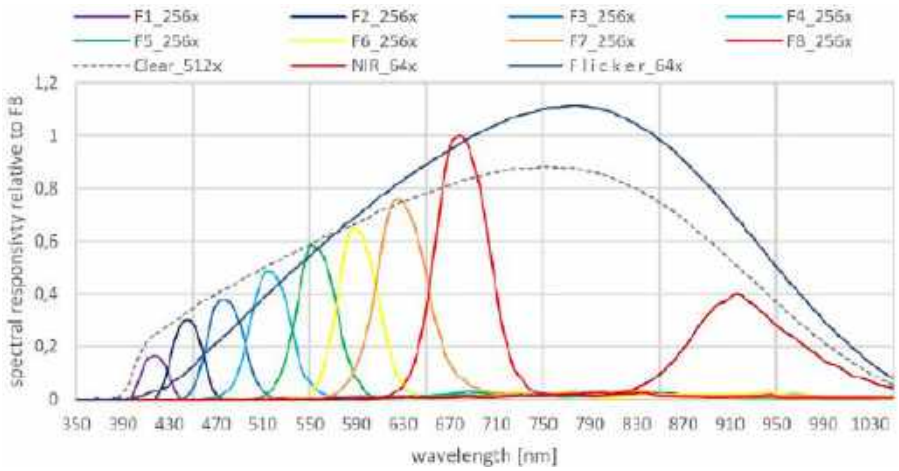
kanałów regulowane są za pomocą interfejsu I²C. Możliwe jest również zaprogramowanie czasu oczekiwania, aby automatycznie ustawić opóźnienie pomiędzy dwoma kolejnymi pomiarami widma i tym samym zmniejszyć całkowite zużycie energii. Dostęp do pozostałych kanałów można uzyskać za pośrednictwem multipleksera (SMUX) łączącego je z jednym z wewnętrznych przetworników ADC (**rysunek 1**).

Układ wyposażony został w matrycę fotodiod 4×4 z parami fotodiod kolorów [2]. Powyżej i poniżej matrycy znajdują się dwie fotodiody z wyspecjalizowanymi funkcjami, takimi jak wykrywanie migotania (FLICKER) i detekcja w bliskiej podczerwieni (NIR). W lewym i prawym dolnym rogu znajduje się przezroczysty kanał (CLEAR) – fotodiody bez filtra. Każdą parę fotodiod można przypisać do jednego z sześciu wewnętrznych przetworników ADC (CH0...CH5).

Jak już wspomniano, układ AS7341 zawiera multiplekser (SMUX) umożliwiający bardzo elastyczne mapowanie kanałów fotodiod do sześciu dostępnych przetworników ADC. Limit 6 ADC wymaga, aby każdy pomiar obejmujący więcej niż sześć z grupy obejmującej 8 kanałów VIS+3 kanały specjalne był wykonywany w dwóch cyklach integracji. Po włączeniu zasilania SMUX musi zostać skonfigurowany przed rozpoczęciem jakichkolwiek pomiarów widmowych. Producent układu udostępnia kod referencyjny i notę aplikacyjną dotyczącą konfiguracji SMUX [7]. Gdy używana jest detekcja migotania, odpowiednią fotodiodę należy skonfigurować na ADC5.



Rysunek 1. Uproszczony schemat blokowy układu AS7341 [4]



Rysunek 2. Odpowiedź spektralna kanałów pomiarowych układu AS7341 odniesiona do F8 [4]

8 kanałów optycznych (VIS) obejmuje widmo widzialne, jeden kanał może służyć do pomiaru bliskiej podczerwieni, a jeden kanał to fotodioda bez filtra („czysta"). Układ ma również osobny kanał do wykrywania migotania światła otoczenia o częstotliwości 50 Hz lub 60 Hz.

Czujnik jest zdolny również do mierzenia widmowych wielkości charakteryzujących światło, takich jak współczynnik oddawania barw (CRI), jasność czy też CCT. W zakresie widzialnym pomiar za pomocą sensora wielospektralnego odbywa się na poziomie radiometrycznym, a nie kolorymetrycznym – oznacza to, że czujnik mierzy widmowy rozkład mocy próbki. AS7341 integruje filtry ze standardowym krzemem CMOS za pomocą technologii nanooptycznych filtrów interferencyjnych, a jego obudowa zapewnia osłonę potrzebną do kontrolowania światła wpadającego do matrycy czujnika.

Kanały optyczne układu AS7341:

- F1 (405...425 nm),
- F2 (435...455 nm),
- F3 (470...490 nm),
- F4 (505...525 nm),
- F5 (545...565 nm),
- F6 (580...600 nm),
- F7 (620...640 nm),
- F8 (670...690 nm),
- NIR (910 nm),
- Clear (charakterystyka odpowiedzi „gołej" diody Si),
- FD – Flicker Detection (jak w kanale Clear).

Jak pokazano na **rysunku 2**, pasma fotodiod czujnika są ułożone w taki sposób, że ich graniczne zakresy ulegają wyrównaniu, nie pozostawiając prawie żadnych przerw w wybranym widmie. Czułości w kanałach są różne. Teoretycznie filtr wąskopasmowy w czujnikach spektralnych powinien przepuszczać jedynie unikalną częstotliwość światła odpowiadającą widmu transmisyjnemu filtra (jak spektrometr). W praktyce, szczególnie w przypadku czujnika AS7341, transmisja i blokowanie poza nim nie są idealne, co skutkuje zakłóceniami optycznymi. Filtry nie są doskonałe – kanał odpowiada też na oświetlenie poza pasmem przepustowym – jednak współczynniki skalujące, macierze lub specjalne algorytmy podczas procesu kalibracji i korekcji zmniejszą takie efekty. Warto podkreślić, że czujnik udostępnia surowe dane, w związku z czym uzyskanie wyników w bardziej użytecznej postaci wymaga wykonania wielu obliczeń. Udostępniany przez producenta arkusz Excel pokazuje sposób obliczenia „od wartości ADC do xyz, uv i poziomu w luksach" [3]. W pliku znalazły się gotowe, olbrzymie tablice potrzebne do obliczeń macierzowych, które umożliwiają wykonanie skalowania i konwersji dla typowych scenariuszy aplikacyjnych.

Tryby integracji

AS7341 obsługuje trzy tryby integracji umożliwiające wykonanie pomiaru spektralnego. Tryb integracji (INT_MODE) konfiguruje się w rejestrze 0x70 (CONFIG). Czas całkowania w trybach INT_MODE = „00" i „01" (SPM i SYNS) ustawia się natomiast za pomocą rejestrów ATIME (0x81) i ASTEP (0xCB:0xCA). Czas całkowania w milisekundach jest równy:

$$tint=(ATIME+1)\times(ASTEP+1)\times 2{,}78\ \mu F$$

Zawartość domyślna rejestru ASTEP wynosi 999 (2,78 ms), a zalecana konfiguracja tych dwóch rejestrów to ASTEP = 599 i ATIME = 29, co daje czas całkowania 50 ms. Czas całkowania określa również wartość ADC w pełnej skali, która jest równa:

$$ADCfullscale=(ATIME+1)\times(ASTEP+1)$$

eprasa.pl 4202fe84f2

Jeśli włączone będzie odczekiwanie pomiędzy kolejnymi pomiarami (WEN = „1”, 0x80), to każdy nowy pomiar rozpocznie się po czasie WTIME (0x83). Należy zwrócić uwagę, by WTIME był wystarczająco długi – tak aby całkowanie widmowe i wszelkie inne funkcje mogły zostać zakończone w tym okresie.

Specjalne rejestry zapewniają konfigurację 6 zintegrowanych przetworników ADC (CH0...CH5). Istnieje możliwość regulacji wzmocnienia, skonfigurowania i włączenia automatycznej kontroli wzmocnienia (AGC) oraz ustawienia automatycznej kompensacji zera. Wzmocnienie jest ustawiane w zakresie 0,5×…512×.

W odniesieniu do detekcji migotania istnieje możliwość skonfigurowania czasu całkowania i wzmocnienia CH5 ADC niezależnie od innych przetworników ADC.

Tryb SPM (spectral measurement, no sync) INT_MODE = 0x0

Ustawienie domyślne. Całkowanie rozpoczyna się od ustawienia bitu SP_EN = „1” w rejestrze ENABLE (0x80). Czas całkowania ustalany jest przez rejestry ATIME (0x81) i ASTEP (0xCB:0xCA). Pomiar przebiega bez synchronizacji (**rysunek 3**).

Rejestry: SP_EN = „1”, INT_MODE = 0x0, ATIME [7:0], ASTEP [15:0], WTIME [7:0]

Tryb SYNS (spectral measurement, start sync) INT_MODE = 0x1

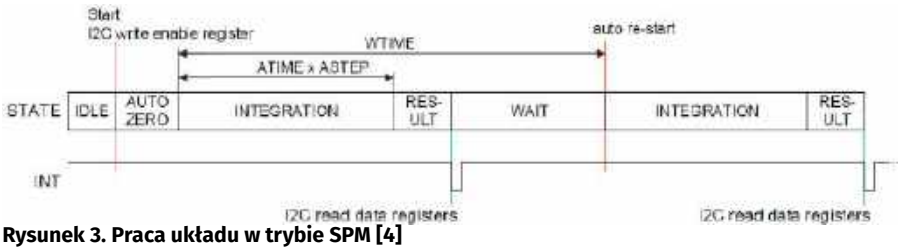
Całkowanie ze startem określanym przez zewnętrzny sygnał. Integracja rozpoczyna się zboczem narastającym/opadającym sygnału na pinie GPIO. Czas całkowania ustalany jest przez rejestry ATIME i ASTEP (**rysunek 4**).

Rejestry: SP_EN = „1”, INT_MODE = 0x1, ATIME [7:0], ASTEP [15:0], WTIME [7:0]

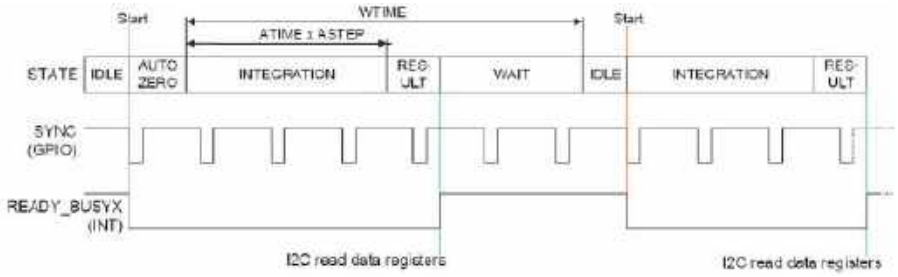
Tryb SYND (spectral measurement, start/stop sync) INT_MODE = 0x3

Całkowanie ze startem i stopem określanym przez zewnętrzny sygnał, a kontrolowane poprzez narastające/opadające zbocze na pinie GPIO. Czas integracji jest definiowany przez rejestr EDGE (0x72) zawierający liczbę opadających zboczy SYNC pomiędzy początkiem i zakończeniem całkowania. Układ przeprowadza integrację do momentu osiągnięcia liczby zboczy opadających na pinie GPIO (**rysunek 5**). Rzeczywisty czas całkowania można odczytać w rejestrze ITIME (0x65:0x64:0x63).

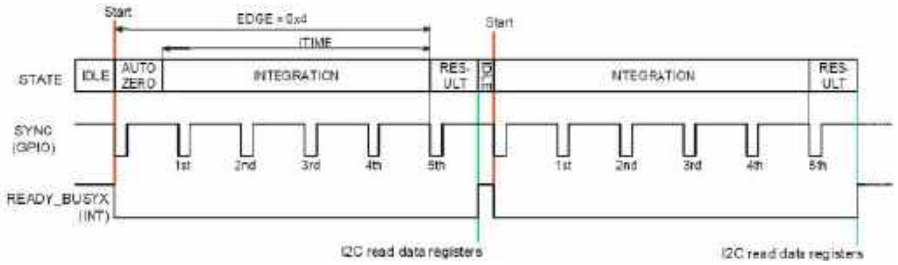
Rejestry: SP_EN = „1”, INT_MODE = 0x3, EDGE[7:0], ITIME[23:0]



Rysunek 3. Praca układu w trybie SPM [4]



Rysunek 4. Praca układu w trybie SYNS [4]



Rysunek 5. Praca układu w trybie SYND [4]

Praca układu AS7341

Układ pracuje w jednym z trzech stanów (**rysunek 6**):

- **Stan aktywny** występuje podczas aktywnej integracji (PON = „1”; SP_EN = „1”). Włączenie oczekiwania pomiędzy dwoma kolejnymi pomiarami (WEN = „1”) pozwala zredukować prąd zasilania.
- **Stan bezczynności** (IDLE) występuje, gdy PON = „1” i wszystkie funkcje pozostają wyłączone.
- **Stan uśpienia** (SLEEP) występuje, gdy PON = „0” i magistrala I²C jest bezczynna. Jeśli występuje ruch na szynie I²C, układ automatycznie przechodzi w tryb bezczynności.

Po włączeniu zasilania, układ inicjalizuje się (POR), wysyłając (zwykle przez 200 µs) sygnał NAK na szynę I²C i nie akceptując transmisji I²C. Po zakończeniu inicjalizacji układ przechodzi w stan uśpienia (SLEEP), w którym wewnętrzny oscylator i inne obwody pozostają nieaktywne, co skutkuje znacznym obniżeniem zużycia energii. Jeśli w tym stanie nastąpi transmisja I²C, rdzeń I²C tymczasowo się budzi, aby obsłużyć komunikację.

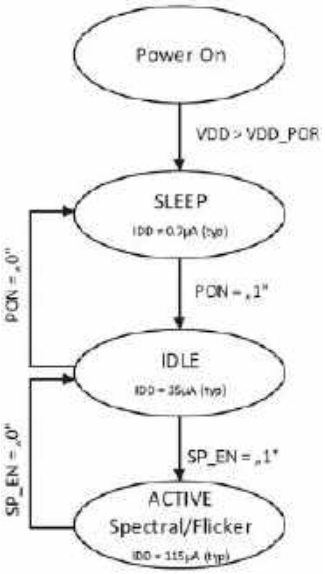
Po włączeniu bitu zasilania PON (rejestr ENABLE) układ przechodzi w stan oczekiwania (IDLE), w którym wewnętrzny oscylator i obwody towarzyszące są aktywne, ale zużycie energii pozostaje niskie.

Po włączeniu pomiaru spektralnego bitem SP_EN = 1 (rejestr ENABLE) układ przechodzi w stan aktywny. Jeżeli pomiar spektralny jest wyłączony (SP_EN = 0), układ powraca do stanu IDLE.

Jeśli włączony jest bit SAI = „1” (rejestr CFG3 – „uśpienie po przerwaniu”), maszyna stanu przejdzie w stan SLEEP, gdy wystąpi przerwanie. Nie powoduje to jednak automatycznej zmiany żadnego z ustawień rejestru (np. bit PON jest nadal wysoki, ale normalny stan pracy zostaje zastąpiony stanem SLEEP). Stan SLEEP kończy się w momencie wyczyszczenia bitu stanu SAI_ACTIVE (rejestr STATUS6).

Rejestr stanu STATUS (0x93) pokazuje szczegóły dotyczące nasycenia, przerwań, nadmiernej temperatury, działania układu i detekcji migotania światła otoczenia.

Odczyt rejestru ASTATUS (0x60 oraz 0x94) powoduje zablokowanie wszystkich 12 bajtów danych widmowych w ramach odczytu stanu. Odczyt tych bajtów po kolei (0x60 do 0x6F lub 0x94 do 0xA0)



Rysunek 6. Uproszczony diagram stanów [4]

eprasa.pl 4202fe84f2

gwarantuje integralność danych. Wszystkie dane widmowe są przechowywane jako wartości 16-bitowe. Jeśli włączona jest detekcja migotania, kanał widmowy piąty (CH5 ADC) będzie używany do realizacji detekcji migotania, a wartość CH5_DATA wyniesie „0”.

Rejestry progowe przerwań P_TH_L_LSB/MSB oraz SP_TH_H_LSB/MSB zapewniają 16-bitowe wartości, które można zastosować jako górne i dolne progi, w celu porównania z 16-bitowymi wartościami kanału CH0_DATA (ADC CH0). Jeżeli bit zezwolenia na przerwanie SP_IEN (rejestr INTENAB) jest włączony i CH0_DATA wykracza poza zakres wyznaczony progami w serii kolejnych pomiarów (długość serii jest określona w polu APERS rejestru PERS), wówczas zgłaszane jest przerwanie.

Bufor FIFO, służący do odczytywania danych widmowych przy mniejszej liczbie transakcji odczytu i zapisu I²C, to 256 bajtów pamięci RAM składającej się z 128 słów dwubajtowych. Host pozyskuje dane z bufora poprzez odczyt adresów: 0xFE...0xFF. Wskaźnik adresu rejestru automatycznie zawija się od 0xFF do 0xFE w miarę odczytywania danych.

Interfejs I²C

Układ korzysta z interfejsu I²C obsługującego adresowanie 7-bitowe i może pracować ze standardową lub pełną prędkością. Transakcje odczytu oraz zapisu są zgodne ze standardem ustanowionym przez firmę Philips (obecnie NXP). 8-bitowy bufor przechowuje adres bajtu wybranego do odczytu lub zapisu. Bufor ten zwiększa się automatycznie po przesłaniu każdego bajtu i jest zachowywany pomiędzy zdarzeniami transakcyjnymi (tj. pozostaje ważny nawet po wydaniu przez układ nadrzędny polecenia STOP i zwolnieniu magistrali I²C). Podczas kolejnych transakcji odczytu, przyszła/powtarzana transakcja odczytu I²C może zwykle pomijać bajt adresu pamięci następujący po adresie układu; bufor zachowuje adres ostatniego rejestru +1. Wszystkie pola 16-bitowe mają mechanizm zatrzaskowy do odczytu i zapisu – podczas odczytywania tych pól najpierw należy odczytać młodszy bajt, co uruchamia 16-bitowy zatrzask. Zaraz potem należy odczytać starszy bajt. Z kolei podczas zapisywania w tych polach najpierw należy zapisać młodszy bajt, a bezpośrednio po nim – starszy bajt. Odczyt lub zapis do tych rejestrów bez przestrzegania opisanych wymagań spowoduje błędy transmisji.

Moduły ewaluacyjne firmy ams-OSRAM

Zestaw ewaluacyjny (EVK) firmy ams-OSRAM stanowi platformę ewaluacyjną przeznaczoną do testowania czujnika AS7341 i demonstrowania różnych scenariuszy użycia [3].

Dostępne są dwa zestawy ewaluacyjne, różniące się zastosowaniem i adapterami optycznymi:

- Standardowy zestaw ALS przeznaczony jest do bezdotykowego pomiaru oświetlenia otoczenia lub badania poszczególnych źródeł światła – zawiera wersję sprzętową bez wbudowanych diod LED, ale z dyfuzorem o maksymalnym polu widzenia dostosowanym do możliwości samego czujnika.
- Zestaw rozszerzony (pomiar światła odbitego) służy do pomiaru kontaktowego kolorowych powierzchni i składa się z czujnika z fabrycznie zamontowaną diodą LED oraz specjalnym adapterem zainstalowanym przed czujnikiem.

W zestawie znalazł się konwerter magistrali I²C do USB (do podłączenia płytki do komputera) oraz rozbudowana aplikacja firmowa.

Moduł czujnika „AS7341 Spectral Color Sensor” firmy Waveshare

Moduł czujnika światła i koloru o nazwie „AS7341 Spectral Color Sensor” (SKU:19564) firmy Waveshare [5] zawiera czujnik AS7341 oraz 6-pinowe gniazdo o rastrze 2,54 mm (**fotografia tytułowa**). Do zestawu dołączono przewód zakończony z jednej strony wtyczką 6-pinową, a z drugiej – pojedynczymi wtykami IDC.

Firma Waveshare udostępnia również stronę Wiki [6] ze schematem, opisem modułu oraz oprogramowaniem (z dosyć dokładnym opisem) na platformy Arduino, Raspberry Pi (C oraz Python) i STM32.

Moduł czujnika AS7341 jest zasilany napięciem VCC (2,5...5,5 V) doprowadzonym z gniazdka do regulatora LDO typu RT9193-18 firmy Richtek. Ten szybki stabilizator – z niskim poziomem szumów i niewielkim spadkiem napięcia (220 mV @ 300 mA) – dostarcza napięcie 1,8 V do zasilania układu AS7341.

Wyprowadzenia SDA, SCL oraz INT układu AS7341 zostały podciągnięte do szyny 1,8 V za pomocą rezystorów 4,7 kΩ oraz podłączone do dwukierunkowego translatora poziomów logicznych LFS0204 firmy Texas Instruments. Natomiast linie SDA i SCL z drugiej strony translatora wyprowadzono na piny gniazdka i wyposażono w rezystory podciągające 4,7 kΩ dołączone do VCC.

Wyprowadzenie GPIO jest podłączone do gniazdka z zastosowaniem dzielnika rezystancyjnego 8,2 kΩ/10 kΩ. Umożliwia to obniżenie zewnętrznego napięcia podawanego na wyprowadzenie (przy pracy w roli wejścia) oraz wystawianie pełnego poziomu 1,8 V (przy pracy w trybie wyjścia).

Do wyprowadzenia LDS układu AS7341 dołączono dwie białe diody LED zasilane napięciem VCC.

MicroPython dla AS7341

Oprogramowanie w języku Python (dostępne na stronie Wiki [6]) zostało zaadaptowane na potrzeby języka MicroPython przez Roba Hammerlinga [7]. Repozytorium GitHub zawiera dokumentację (w tym notę aplikacyjną AN000660), driver czujnika oraz wiele gotowych przykładów jego użytkowania.

Dosyć tajemniczo kształtuje się sprawa noty aplikacyjnej AN000660 „AS7341 Demo for Fast Measurement Using Unicom Board” firmy ams-OSRAM. Jest ona konieczna do poprawnego konfigurowania multiplekserów SMUX służących do wyboru kanałów spektralnych czujnika AS7341 – nie można jej jednak nigdzie znaleźć, nawet za pomocą Google. Po dłuższym poszukiwaniu okazuje się, że została ona zamieszczona w pakiecie instalacyjnym zip modułu ewaluacyjnego firmy ams-OSRAM [7].

Praca modułu czujnika AS7341 z płytką Enviro Weather

Pomiary zostały wykonane z zastosowaniem modułu czujnika AS7341 „AS7341 Spectral Color Sensor” (SKU:19564)

REKLAMA

eprasa.pl 4202fe84f2

firmy Waveshare [5], dołączonego do płytki Enviro Weather firmy Pimoroni [9], drivera języka MicroPython Roba Hammerlinga [7] oraz zaadaptowanego stamtąd przykładu.

Płytka modułu czujnika AS7341 została lekko zmodyfikowana, bowiem linię zasilania VCC dołączono do wyjścia dodatkowego regulatora LDO 3,3 V, zasilanego z linii VSYS płytki Raspberry Pi Pico W. Wyprowadzenia SCL i SDA płytki czujnika (z rezystorami podciągającymi 4,7 kΩ do linii VCC) dołączono do linii SCL (GP5) i SDA (GP4) płytki Enviro Weather, gdzie są już umieszczone rezystory podciągające 10 kΩ do linii 3V3. W opisanej konstrukcji występuje problem z podciąganiem do dwóch różnych linii zasilania oraz z trochę zbyt dużym obciążeniem prądowym. Dlatego z płytki czujnika wylutowano dwa rezystory 4,7 kΩ z oznaczeniami 472 (umieszczone bezpośrednio obok napisu VCC – patrz fotografia tytułowa).

Kod programu do inicjalizacji czujnika AS7341 został pokazany na **listingu 1**. Mamy tu standardowe ustawienia. Pomiary są wykonywane w trybie SPM. Ponieważ układ ma tylko 6 przetworników ADC, odczyt kanałów spektralnych przeprowadzany jest w dwóch krokach: najpierw F1...F4 + Clear + NIR, a potem F5... F8 + Clear + NIR. Odczyt kanału pomiaru migotania wykonywany jest osobno (w kanale CH5 `ADC). Dodatkowo – za pomocą czujnika LTR-559 firmy LiteOn na płytce Enviro Weather – realizowany jest pomiar światła ALS [9].

```
sensor.set_measure_mode(AS7341_MODE_SPM)    # started by SP_EN bit = 1
sensor.set_atime(29)                        # 30× ASTEPS = 50,04 ms
sensor.set_astep(599)                       # 2.78 µs × 600 = 1.668 ms
sensor.set_again(4)                         # factor 8× (with pretty much light)

sensor.start_measure("F1F4CN")
f1,f2,f3,f4,clr,nir = sensor.get_spectral_data()
sensor.start_measure("F5F8CN")
f5,f6,f7,f8,clr,nir = sensor.get_spectral_data()
flicker_freq = sensor.get_flicker_frequency()
enviro_readings = enviro.get_sensor_readings()
print("Ambient light: " + str(enviro_readings["luminance"]) + " LUX")
```

Listing 1. Kod programu do inicjalizacji czujnika AS7341

```
Detected devices at I2C-addresses:
['0x23', '0x29', '0x39', '0x48', '0x51', '0x77']
------------------------
Toggle LED on pin 6, with every interrupt falling edge on pin 22 (INT signal)
2000-01-01 00:29:52 [info     / 113kB]   - seconds since last reading: 206
PINint Pin(GPIO22, mode=IN, pull=PULL_UP) 0
F1 (405-425nm): 5
F2 (435-455nm): 51
F3 (470-490nm): 30
F4 (505-525nm): 38

PINint Pin(GPIO22, mode=IN, pull=PULL_UP) 0
F5 (545-565nm): 71
F6 (580-600nm): 72
F7 (620-640nm): 40
F8 (670-690nm): 35
Clear: 66
NIR: 6

PINint Pin(GPIO22, mode=IN, pull=PULL_UP) 0
Flicker frequency: 100 Hz
Ambient light: 60.28 LUX
```

Rysunek 7. Pomiar widma światła żarówki LED

Zastosowanie płytki Enviro Weather wymaga wpisania do niej najpierw aktualnej wersji firmowego pliku obrazu (uf2), zawierającego MicroPythona. Następnie należy wpisać folder projektu najnowszej aplikacji Enviro (czynność ta została dokładnie omówiona w poprzednim artykule „Stacja pogodowa Enviro Weather firmy Pimoroni" [9]). Potem trzeba dodać pliki drivera czujnika: as7341.py oraz as7341_smux_select.py pobrane z GitHuba [7]. Na koniec pozostaje dodanie pliku as7341_EnviroWeather.py (https://tiny.pl/dnplh).

W projekcie zaadaptowano przykład as7341_all.py z repozytorium [7]. Dodatkowo zastosowane zostały w nim fragmenty realizujące detekcję migotania, a także obsługę przerwań oraz sterowanie zewnętrzną diodą LED.

Biała dioda LED włącza się na początku cyklu pomiarowego. W procedurze obsługi przerwania stan diody użytkownika (na płytce Enviro Weather) zmieniany jest na przeciwny. Jednocześnie wypisany zostaje komunikat. Rezultat działania oprogramowania uruchamianego w środowisku Thonny pokazano na **rysunku 7**. Wykonano pomiar typowej żarówki energetycznej LED. Należy pamiętać, że dane z czujnika (wartości poziomów ADC) pokazywane są w formie surowej, tj. bez normalizacji.

Podsumowanie

Pomiar światła – a szczególnie pomiar koloru – nie jest łatwy. Czujnik spektralny lub „spektrometr na chipie", taki jak AS7341 firmy ams-OSRAM, zapewnia dokładne pomiary widmowego rozkładu mocy, zarówno w pomiarach emisyjnych, jak i odbiciowych. AS7341, będący niewielkim chipem do montażu powierzchniowego, można zintegrować z małymi produktami – choćby czujnikiem dymu czy ręcznym kolorymetrem. Czujnik spektralny w czujniku dymu potrafi rozróżnić źródła dymu, aby dokładniej wskazać rodzaj pożaru, a tym samym podnieść poziom bezpieczeństwa, a nawet uratować ludzkie życie. Możliwość odróżnienia dymu papierosowego od emisji z waporyzatorów (vape) otwiera drogę do opracowania nowych możliwości, które stanowią wartość dodaną czujnika dymu. Z kolei wielokanałowy sensor spektralny zapewnia dokładne równoważenie punktu bieli w najszerszym zakresie warunków oświetleniowych. Czujniki takie pomagają smartfonom w uzyskaniu doskonałego odwzorowania barw przez aparat fotograficzny, nawet w scenach, w których nie działają czujniki RGB lub XYZ. A wszystko to dostępne jest za rozsądną cenę.

Henryk A. Kowalski
Instytut Informatyki
Politechnika Warszawska

Literatura

[1] Spectral sensing, ams-Osram, https://tiny.pl/dnpjx
[2] „Systemy dla Internetu Rzeczy (45) Czujniki koloru", Henryk A. Kowalski, EP 2/2021, https://tiny.pl/dnpjw
[3] ams AS7341 – 11-Channel Spectral Color Sensor, ams-Osram, https://tiny.pl/dnpjc
[4] AS7341 11-Channel Multi-Spectral Digital Sensor, Datasheet DS000504, v3-00, 2020-Jun-25, https://tiny.pl/dnpjd
[5] AS7341 Spectral Color Sensor, SKU: 19564, Waveshare, https://tiny.pl/dnpjj
[6] AS7341 Spectral Color Sensor, Wiki, Waveshare, https://tiny.pl/dnpjp
[7] micropython-as7341, Version 0.3, Rob Hammerling, September 2023, https://tiny.pl/dnpph
[8] „Czujniki optyczne (4). Sensory koloru oraz czujniki i moduły multispektralne", Przemysław Musz, EP 10/2023, https://tiny.pl/dnpjk
[9] „Stacja pogodowa Enviro Weather firmy Pimoroni", EP 4/2024, https://tiny.pl/d93r1

REKLAMA
https://www.facebook.com/ElektronikaPraktyczna

eprasa.pl 4202fe84f2

Uwaga na diody zabezpieczające!

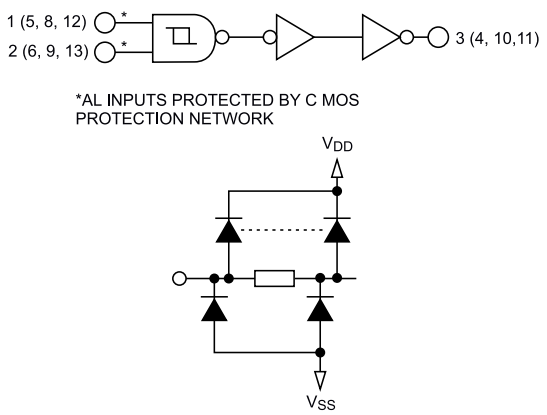
Wejście niemal każdego nowoczesnego układu cyfrowego zabezpieczają dwie diody półprzewodnikowe. Często wręcz dodajemy je sami, by ograniczyć wartość napięcia wejściowego. Z reguły mają zbawienny wpływ na układ, jednak są takie sytuacje, w których potrafią napsuć nam krwi. Kiedy zatem trzeba im się przyjrzeć ze szczególną uwagą?

Dwie połączone szeregowo diody stały się znakiem rozpoznawczym wejść (a często też wyjść) układów scalonych wykonanych w technologii CMOS. Jedna jest włączona anodą do masy zasilania i katodą do linii sygnałowej, druga zaś katodą do linii zasilającej, natomiast anodą do chronionego wyprowadzenia cyfrowego. W notach katalogowych można spotkać bardziej rozbudowane układy (**rysunek 1**), zawierające rezystor między dwoma zespołami takich diod. W literaturze określa się je mianem „clamping diodes", co jest o tyle mylące, że każdy inny diodowy ogranicznik napięcia bywa nazywany w ten sam sposób.

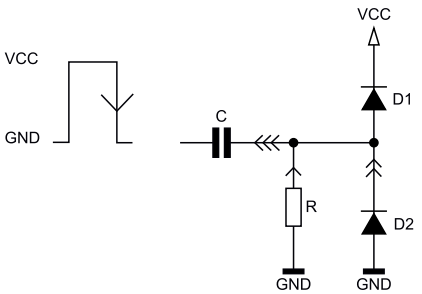
Powodem, dla którego stosuje się takie obwody, jest ochrona przed wyładowaniami elektrostatycznymi, znanymi szerzej pod anglojęzycznym skrótkowcem ESD. Bez tej ochrony każdy układ scalony wyjęty z przewodzącej gąbki niemal natychmiast ulegałby „uśmierceniu" wskutek przebicia delikatnego izolatora podbramkowego w tranzystorach CMOS. Ładunek, który zgromadzi się na wyprowadzeniu, może zostać szybko i bezboleśnie rozprowadzony przy użyciu tychże diod.

Wspomniane diody znajdują również zastosowanie jako pomocniczy limiter napięcia na wyprowadzeniach. Jeżeli jego wartość, mierzona względem masy, stałaby się ujemna, wówczas otwarciu uległaby dolna dioda, co spowodowałoby „dolanie" prądu do węzła o niskim potencjale. Ta sytuacja z reguły rzadko ma znaczenie, ponieważ spora część naszych układów jest zasilana asymetrycznie – choć istnieją oczywiście wyjątki od tej zasady, na przykład podłączenie do wejścia obwodu różniczkującego RC, choćby takiego z **rysunku 2**. Kondensator C, który był naładowany po dłużej utrzymującym się stanie wysokim na wejściu układu, nagle wymusza napięcie ujemne na wejściu układu scalonego, ponieważ w chwili nadejścia zbocza ujemnego nie zdąży się przeładować i zadziała jak źródło napięciowe, sprowadzając potencjał swojej prawej okładki poniżej zera. Ot, teoria stanów nieustalonych, nic więcej.

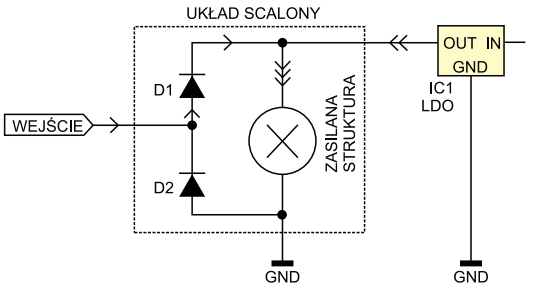
Jednak częściej doprowadzamy do otwierania się diody górnej, która ogranicza napięcie na danym wyprowadzeniu – a jakże!



Rysunek 1. Układ diod zabezpieczających w układach CMOS (na przykładzie noty katalogowej układu CD4093B firmy Texas Instruments)



Rysunek 2. Przykładowa sytuacja wymuszająca krótkotrwałe otwarcie dolnej diody

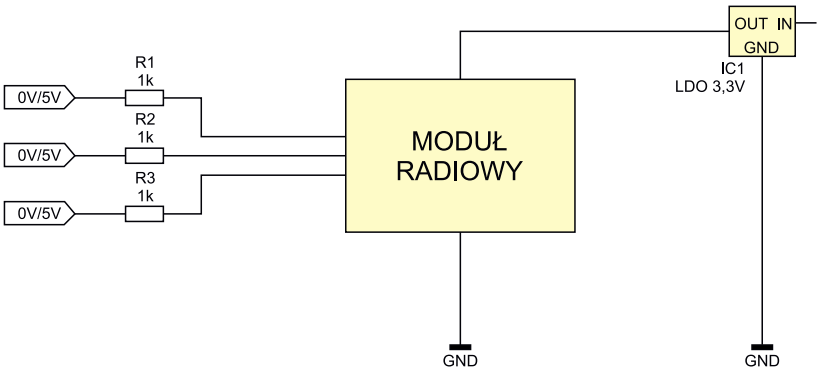


Rysunek 3. Zmiana rozpływu prądów spowodowana otwarciem górnej diody zabezpieczającej

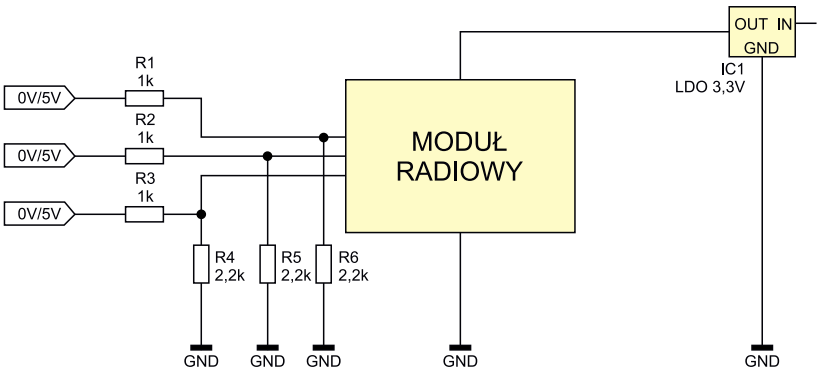
eprasa.pl 4202fe84f2

– od góry. Jeżeli potencjał wzrośnie powyżej napięcia zasilającego, do którego jest owa dioda podłączona, wówczas otworzy się ona i odprowadzi prąd do szyny zasilającej, zapobiegając dalszemu wzrostowi potencjału. I w tym miejscu większość podręczników zamyka temat: koniec, nie da się więcej, przecież tak działa zwykły limiter diodowy! Owszem, może działać w ten sposób, ale z uwzględnieniem kilku zastrzeżeń, o których już się zbyt często nie mówi.

Według teorii prąd przepływający przez otwartą górną diodę wpływa do linii zasilającej. Jednak nie może on tak sobie zniknąć – zasada zachowania ładunku elektrycznego nadal nas obowiązuje. Wspomniany prąd rozpływa się więc po całym układzie i uczestniczy w jego zasilaniu. Skoro pojawia się dodatkowe źródło zasilania, to stabilizator przeznaczony do zasilania tego obwodu wypuszcza z siebie prąd o mniejszym natężeniu, ponieważ zależy mu na utrzymaniu stałego potencjału linii zasilającej – **rysunek 3**. Jeżeli źródłem energii jest bateria lub akumulator, wówczas wypływa z nich prąd o niższym natężeniu. Musi zostać spełnione prądowe prawo Kirchhoffa: skoro prąd pobierany przez układ nie zmienia się, a w węźle pojawia się jego dodatkowe źródło, to prąd wypływający z pierwotnego źródła musi zmaleć.



Rysunek 4. Fragment schematu układu, który wpędził mnie w kłopoty



Rysunek 5. Najprostszy sposób rozwiązania problemu

Wszystko gra, dopóki mamy układ pobierający relatywnie duży prąd (kilkadziesiąt czy kilkaset miliamperów), zaś prąd – dopływający okazjonalnie przez którąś z diod zabezpieczających pracujący w nim układ scalony – ma natężenie rzędu kilku miliamperów. Sytuacja jednak mocno się komplikuje, kiedy prąd pobierany przez układ również jest niewielki, a nawet mniejszy od tego, który może wpłynąć przez opisane diody. Wówczas napięcie zasilające układ jest w stanie wzrosnąć nawet powyżej dopuszczalnej dlań wartości, co w skrajnym przypadku może skończyć się jego uszkodzeniem.

Taka sytuacja przydarzyła mi się dawno temu w prototypowej serii układów zawierających moduł radiowy RFM69. Moduł ten był zasilany napięciem 3,3 V ze stabilizatora LDO (maksymalne napięcie zasilające 3,9 V, rekomendowane do 3,6 V), zaś mikrokontroler z nim się komunikujący – napięciem 5 V. Między nimi znajdowały się trzy linie sygnałowe, które wchodziły do układu radiowego, więc wartość szczytowa na nich była wyższa od napięcia zasilającego. Pamiętając o różnicy poziomów, dodałem w szereg rezystory 1 kΩ – mogłem sobie na to pozwolić, ponieważ komunikacja nie była zbyt szybka, choć około 100 kHz częstotliwości zegara musiałem mieć zapewnione. Jednocześnie nie chciałem wprowadzać dzielników napięciowych na każdej z linii sygnałowych, ponieważ zależało mi na odzyskaniu miejsca na płytce – przecież diody zabezpieczające wejścia modułu radiowego ograniczą napięcie. Ich prąd również jest ograniczony przez szeregowe rezystory, zatem kilka miliamperów rozpłynie się po układzie i niczym nie trzeba się martwić... **Rysunek 4** zawiera uproszczony schemat tej części układu.

Partia testowa urządzeń liczyła dziesięć sztuk. W kilku moduł radiowy przestał reagować niemal natychmiast po włączeniu zasilania. Niektóre działały dłużej, ale był to raczej czas rzędu kilku minut. Część z nich z kolei wydawała się działać poprawnie. Kiepskie wyniki jak na urządzenie, które docelowo miało zostać wyprodukowane w setkach sztuk. Rozpoczęły się wielkie poszukiwania możliwej przyczyny, zaś zmiana dostawcy modułów radiowych nastąpiła kilkukrotnie. Próbowałem częściej zerować te moduły, zmniejszyć częstotliwość sygnału zegarowego podczas komunikacji z nimi (choć rozwiązania tego nie mogłem docelowo pozostawić w projekcie), ale wszystko na marne. Pojawiła się frustracja wywołana brakiem jednoznacznej przyczyny – i zniecierpliwienie klienta.

Dopiero po czasie skojarzyłem, że w stanie gotowości (Standby mode) moduł pobiera prąd o natężeniu nie większym niż 1,5 mA. Tymczasem ja „pchałem" do niego, za pośrednictwem diod zabezpieczających, prawie trzy razy tyle. Oczywiście stabilizator LDO nie jest przystosowany do tego, by „wciągać" prąd, może go jedynie z siebie „wyrzucać". Kiedy więc dochodziło do pojawienia się stanu wysokiego na którejkolwiek z linii sterowanych przez mikrokontroler, prąd wychodzący ze stabilizatora spadał niemal do zera. Natomiast przy dwóch liniach napięcie na pinie zasilającym moduł RFM69 wzrastało powyżej 4 V...

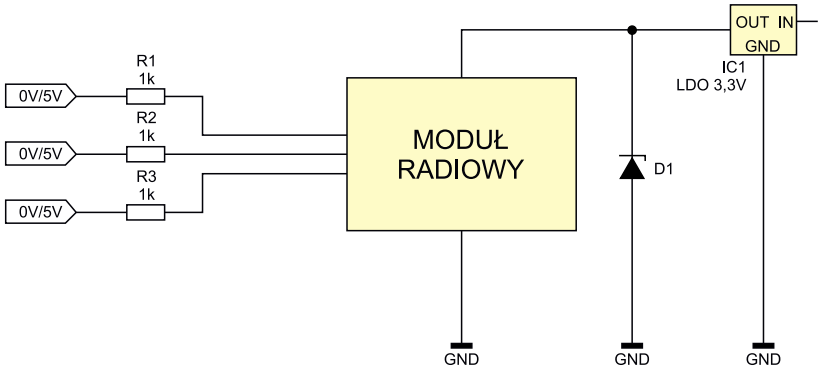
Chcąc nie chcąc, przeprosiłem się z dzielnikami napięcia i problem zniknął, jak ręką odjął. Wystarczyło dołożenie rezystorów o wartości 2,2 kΩ za każdym z rezystorów szeregowych do masy, by napięcie nie wzrastało już powyżej niebezpiecznej dla modułu wartości (szczegóły obrazuje **rysunek 5**). Przy okazji zmalała rezystancja sterująca pojemnością wejściową układu RFM69. Problem z głowy? Owszem, ale to nie koniec moich poszukiwań na ten temat. Ciekawiło mnie, w jaki inny sposób można byłoby podobny problem rozwiązać.

Dodanie diody typu transil między masę a zasilanie takiego układu (**rysunek 6**) w teorii powinno zadziałać poprawnie, ponieważ – przy próbie zwiększenia napięcia zasilającego powyżej napięcia jego przebicia – otworzy się on i przejmie na siebie nadmiar prądu. Jest to jakieś rozwiązanie, lecz trzeba dobrze dobrać wspomniany element – tak by w stanie prawidłowej pracy był zatkany (przy napięciu 3,3 V), zaś już kilkaset miliwoltów powyżej tej granicy (przy napięciu 3,9 V lub niżej) był zdolny do przewodzenia prądu o natężeniu kilku miliamperów. Nie jest to proste, zważywszy na fakt, że napięcie przebicia zależy od temperatury. Popularne diody TVS, które służą do zabezpieczania linii 3,3 V, przewodzą prąd o natężeniu 5 mA przy znacznie wyższym napięciu. Przykładowo, dioda PESD3V3U1UA.115 ma prąd

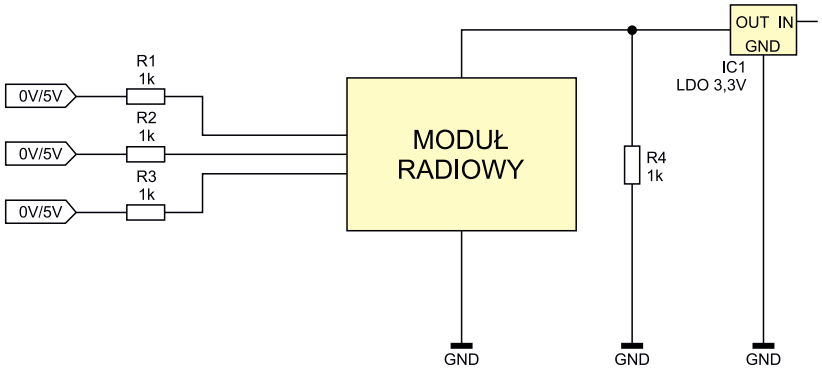
eprasa.pl 4202fe84f2

upływu nieprzekraczający 1 μA przy 3,3 V, ale prąd 5 mA osiąga dopiero przy napięciu 4,5...6,8 V, czyli – w opisywanym zastosowaniu – nieakceptowalnie wysokim. Można byłoby ją zastąpić bardziej precyzyjnym ogranicznikiem, stosując chociażby układ TL431, lecz wtedy cały ów dodatkowy obwód staje się większy, droższy i bardziej złożony niż proste rezystory tworzące dzielniki napięcia na liniach sygnałowych.

Wpadłem na jeszcze jeden pomysł, zdecydowanie prostszy do wdrożenia. Skoro problemem jest fakt, że odbiornik prądu z węzła pobiera go zbyt mało, to trzeba dodać jeszcze jedną gałąź, którą ten prąd mógłby sobie odpływać do masy, najlepiej stale. Na szczęście w omawianym układzie energooszczędność nie była dla mnie priorytetem. I tak oto doszedłem do... zwykłego rezystora o wartości 1 kΩ, który włączyłem między wyjście stabilizatora 3,3 V a masę – co ilustruje **rysunek 7**. Wprawdzie przez większość czasu pobierał prąd o natężeniu 3,3 mA, lecz – przy załączeniu się wszystkich trzech linii sygnałowych w stan wysoki – gwarantował odprowadzenie nadmiarowego prądu.



Rysunek 6. Modyfikacja z diodą transil



Rysunek 7. Modyfikacja z rezystorem odprowadzającym prąd

W tym układzie usunąłem zaledwie dwa rezystory (w porównaniu z kanonicznym rozwiązaniem), lecz przy magistrali prowadzącej większą liczbę sygnałów ta oszczędność na elementach byłaby jeszcze większa. Warto również rozważyć dobranie rezystorów szeregowych o tak dużej rezystancji, aby przepływający przez nie prąd był mniejszy niż pobierany przez niskonapięciowy układ – lecz to z kolei odbija się negatywnie na szybkości transmisji. Nie próbuję namawiać Czytelników do skąpstwa, raczej prezentuję alternatywne rozwiązania w sytuacji, kiedy na płytce pozostało ekstremalnie mało miejsca.

Michał Kurzela, EP

REKLAMA

Kurs programowania mikrokontrolerów Megawin

Na łamach „Elektroniki Praktycznej" publikujemy pierwszy na świecie kurs podstaw programowania interesujących, budżetowych mikrokontrolerów z rdzeniem ARM Cortex-M3 firmy Megawin.

ulubionykiosk.pl

eprasa.pl 4202fe84f2

EMC – checklista projektanta, czyli jak nie popełnić błędów?

Kompatybilność elektromagnetyczna to nie tylko zestaw formalnych wymogów, które należy spełnić, by móc legalnie wprowadzić produkt do masowej sprzedaży. W rzeczywistości zagadnienia EMC mają ogromny wpływ na funkcjonalność, bezpieczeństwo i jakość użytkową urządzeń elektronicznych we wszystkich branżach – od elektroniki konsumenckiej, poprzez automatykę przemysłową i aparaturę medyczną, aż do systemów lotniczych, wojskowych czy kosmicznych. Mnogość zjawisk elektromagnetycznych, zachodzących tylko w pewnych szczególnych warunkach, wyrobiła w wielu konstruktorach przeświadczenie, że świat EMC ma sporo wspólnego z czarną magią. Nic bardziej mylnego – za wszystkimi „dziwnymi objawami" rejestrowanymi w urządzeniach podczas testów laboratoryjnych (bądź zwykłej eksploatacji) stoi bowiem tylko i wyłącznie czysta fizyka. W sierpniowym numerze EP bierzemy na warsztat trudny, ale zarazem niezwykle ciekawy temat kompatybilności elektromagnetycznej, widzianej z punktu widzenia konstruktora.

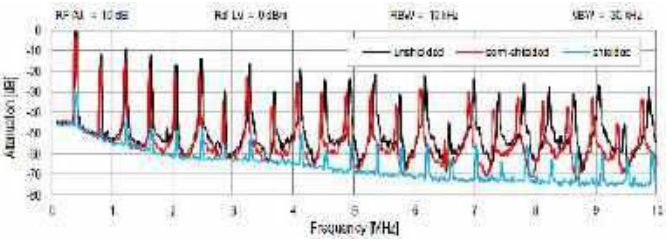
Tematyka kompatybilności elektromagnetycznej pojawiała się łamach „Elektroniki Praktycznej" już wielokrotnie. Nic dziwnego – wszak zachowanie zgodności z wymogami normalizacyjnymi nie tylko otwiera drogę do komercjalizacji wyrobu, ale także istotnie wpływa na jego użyteczność i niezawodność.

Na temat EMC powstały dziesiątki (a nawet setki) książek, nieprzebrana ilość artykułów, wpisów blogowych i kursów. Oczywiste jest, że w ramach jednego artykułu dałoby się szeroko zaprezentować zaledwie jedno, może dwa szczegółowe zagadnienia z tego zakresu. Dlatego – podchodząc do sprawy realistycznie, ale i praktycznie – postanowiliśmy tym razem zaproponować naszym Czytelnikom nieco inną formę materiału: zamiast klasycznego artykułu, przygotowaliśmy przystępną checklistę EMC, czyli zestaw 14 wybranych zasad, które warto spełnić, by zwiększyć szanse na powodzenie badań projektowanego urządzenia elektronicznego w laboratorium kompatybilności elektromagnetycznej.

Rysunek 1. Przykładowy wykres z normy EN55015 – limity emisji zakłóceń promieniowanych w zakresie 9 kHz...30 MHz (http://t.ly/nebzA)

1. Normy nie gryzą

Dobrym punktem wyjścia do zapewnienia zgodności z normami jest... uwzględnienie ich już na samym początku projektu. Choć wskazówka ta może brzmieć jak banał, w praktyce nader często spotkać się można z zupełnie odwrotnym podejściem: wielu konstruktorów najpierw projektuje urządzenie tak, by w spełniało założenia funkcjonalne, a dopiero przed etapem komercjalizacji zaczyna się zastanawiać nad tym, które normy owego sprzętu dotyczą. Co gorsza, w niektórych firmach podobne podejście praktykuje się nawet podczas opracowywania urządzeń medycznych. I choć skupienie się wyłącznie na funkcjonalności jest istotnie ogromnie ważne na etapie prototypu czy też MVP, to jednak proces projektowania wersji komercyjnej zdecydowanie powinien uwzględniać zagadnienia normalizacyjne (zarówno w zakresie odporności, jak i emisji – patrz **rysunek 1**) już na samym początku. Co ważne, opisana zasada dotyczy nie tylko samej kompatybilności elektromagnetycznej, ale także norm z zakresu bezpieczeństwa elektrycznego, użyteczności bądź aspektów natury mechanicznej lub materiałowej – każda branża rządzi się swoimi prawami, dlatego warto przynajmniej pobieżnie zorientować się, z czym będziemy mieli do czynienia po przekroczeniu drzwi laboratorium.



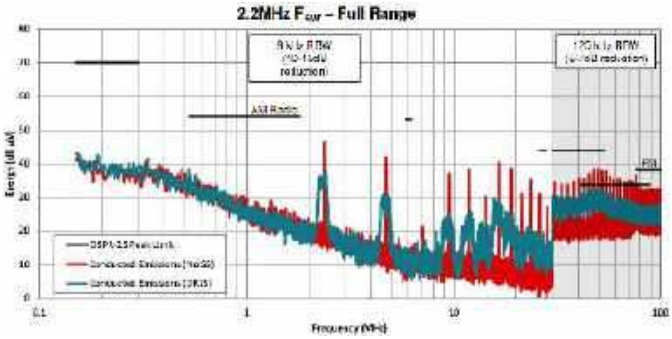
Rysunek 2. Porównanie poziomów zakłóceń w paśmie 0...10 MHz emitowanych przez przetwornicę DC/DC z zastosowaniem dławika nieekranowanego (czarna linia), częściowo ekranowanego (czerwona linia) i w pełni ekranowanego (niebieska linia) – http://t.ly/hFE-8

2. Wybieramy klocki

Na wczesnym etapie projektu nie sposób rzecz jasna przewidzieć, jaka będzie rzeczywista emisja zakłóceń oraz w jakim stopniu sprzęt będzie podatny na zaburzenia odbierane z zewnątrz. Konkretne wartości liczbowe „ujawniają" się zwykle dopiero na etapie badań precompliance, ale znajomość: pasm częstotliwości podlegających ocenie, wartości narażeń testowych generowanych przez symulator ESD, czy też napięć stosowanych w testach udarowych pozwala od razu dobrać komponenty, które dadzą najwyższe prawdopodobieństwo sukcesu w pomiarach laboratoryjnych. Właściwa selekcja elementów ochrony przeciwprzepięciowej (np. transili), niskoszumnego konwertera DC/DC, a nawet odpowiednich komponentów biernych potrafi zdziałać cuda. Dobrym przykładem będzie kwestia dławika współpracującego z przetwornicą impulsową – modele ekranowane są wprawdzie droższe od ich nieekranowanych odpowiedników o zbliżonych parametrach (indukcyjności, prądzie nasycenia, rezystancji DC itp.), ale zastosowanie ekranu magnetycznego pozwala znacznie ograniczyć emisję zaburzeń promieniowanych (**rysunek 2**). Mało tego – może się okazać, że oszczędności pozyskane dzięki zastosowaniu „gorszego" dławika okazałyby się niewspółmiernie małe, w porównaniu do kosztów procesu przeprojektowywania PCB oraz liczby iteracyjnie wykonywanych badań

eprasa.pl 4202fe84f2

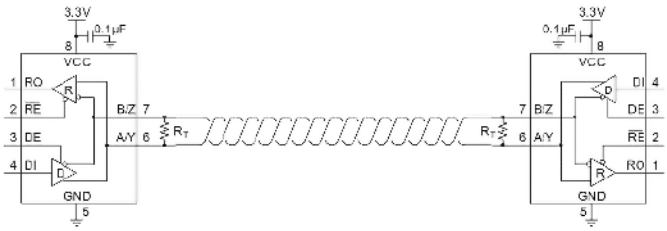
inżynierskich. Zasada „right the first time” w odniesieniu do EMC jest naprawdę warta rozważenia – a widać to jak na dłoni po zapoznaniu się z przykładowym cennikiem usług dowolnego laboratorium...



Rysunek 3. Porównanie zaburzeń emitowanych w przypadku przetwornicy DC/DC pracującej ze stałą częstotliwością przełączania oraz kontrolera ze spektrum rozproszonym (http://t.ly/bzWFx)

3. Dziel i rządź

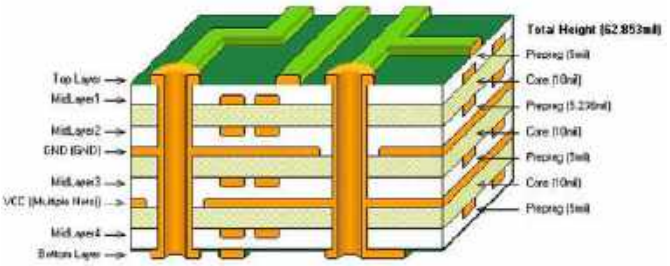
Skoro jesteśmy już przy tematyce konwerterów DC/DC, to warto podać ciekawy przykład, związany z doborem właściwego rodzaju kontrolera pozwalającego na obniżenie poziomu zakłóceń emitowanych przez układ zasilania. Klasyczne przetwornice impulsowe pracują zwykle na jednej, ustalonej fabrycznie (lub regulowanej przez użytkownika) częstotliwości. Takie rozwiązanie jest relatywnie proste konstrukcyjnie, ale ma dość istotną wadę: całość energii emitowanej w postaci zaburzeń RFI skupia się w wąskich pikach widma, odpowiadających częstotliwości podstawowej oraz szeregowi jej kolejnych harmonicznych. Zastosowanie techniki widma rozproszonego (ang. spread spectrum), polegającej na losowym lub okresowym odstrajaniu częstotliwości kluczowania w niewielkim zakresie, powoduje „rozlanie” pików do postaci szerszych pasm o niższej wartości szczytowej (**rysunek 3**). W ten sposób konstruktor może „uciec” nawet o kilkanaście decybeli od limitów emisji określonych normą, co w wielu sytuacjach okaże się zbawienne dla wyniku badań laboratoryjnych.



Rysunek 4. Schemat aplikacyjny transceiverów RS485 marki Renesas, wyposażonych we wbudowane obwody ochronne z ratingiem ±3 kV (61000-4-4), ±8 kV (61000-4-2, metoda kontaktowa) oraz ±15 kV (HBM) – http://t.ly/VYdOJ

4. Łowienie na przynętę

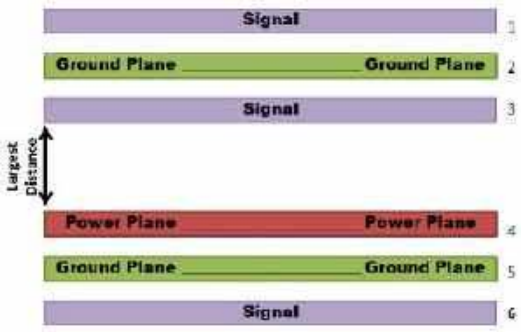
Jak powszechnie wiadomo, reklama jest dźwignią handlu. Niestety, opisy rozmaitych produktów nagminnie tuszują ich niedoskonałości, wydobywając zarazem „na wierzch” tylko zalety. Technika ta jest maksymalnie eksploatowana np. przez deweloperów, którzy kuszą klientów mieszkaniami w dobrze skomunikowanym i zielonym otoczeniu, zapominając przy tym wspomnieć o planowanej tuż obok linii tramwajowej lub obwodnicy. Producenci komponentów elektronicznych nie pozwalają sobie wprawdzie na aż tak wyraźne nadużycia zaufania odbiorców, jednak także nie pozostają „kryształowo czyści” – dobrym przykładem mogą tutaj być noty katalogowe układów scalonych z wbudowanymi wzmocnionymi zabezpieczeniami ESD. Należy bowiem pamiętać, że nie każdy układ opisywany np. jako „15 kV ESD-protected” faktycznie spełni wymogi normy dotyczącej systemowej ochrony ESD, tj. 61000-4-2. Często zapis ten będzie się bowiem odnosił do badań wg modelu HBM (Human Body Model), a rzeczywista wytrzymałość przepięciowa okaże się znacznie niższa. Jeżeli zatem chcemy zaoszczędzić złotówkę i kilka...kilkanaście mm² powierzchni PCB poprzez rezygnację z dodatkowego transila – upewnijmy się najpierw, że wybrany konwerter RS485 (przykład można zobaczyć na **rysunku 4**), interfejs USB lub inny układ faktycznie przetrwa „strzały” z symulatora ESD. Więcej informacji na ten temat można znaleźć w artykule pt. „Elektroniczne komponenty zabezpieczające – podzespoły, aplikacje i normy”, który także publikujemy w tym numerze „Elektroniki Praktycznej”.



Rysunek 5. Przykładowy stos 6-warstwowej płytki drukowanej (http://t.ly/wznMo)

5. Sklep z kanapkami

Kolejnym ważnym założeniem projektowym, które ma ogromny wpływ na optymalizację osiągów urządzenia w zakresie EMC, jest wybór optymalnego stosu wielowarstwowej płytki drukowanej. Warto w tym przypadku posiłkować się dokumentacją udostępnianą przez wybraną fabrykę – przyjęcie nierealistycznych lub nieekonomicznych założeń drastycznie zwiększy koszty produkcji, zaś podążanie ścieżką standardową (tj. zastosowanie jednego ze typowych stosów) pozwoli zachować relatywnie niski koszt i skrócić czas produkcji. Powszechnie wiadomo, że obwody wielowarstwowe dają projektantom znacznie większe możliwości m.in. w zakresie ekranowania krytycznych ścieżek w celu redukcji zakłóceń emitowanych i odbieranych z otoczenia. Równie ważne, a nawet jeszcze ważniejsze od grubości i ułożenia poszczególnych warstw dielektryka oraz miedzi (**rysunek 5**), będzie właściwe użycie dostępnych płaszczyzn. Kolejność ułożenia płaszczyzn masy i zasilania oraz warstw sygnałowych determinuje bowiem nie tylko ścieżki prądów powrotnych i pojemności międzypłaszczyznowe, ale także... istotnie zmienia warunki kontroli impedancji.



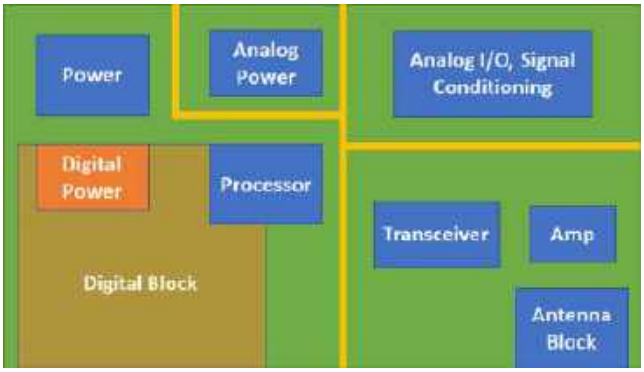
Rysunek 6. Jeden z najlepszych (pod względem integralności sygnałów i mocy) układów stosu obwodu 6-warstwowego (http://t.ly/CSgLP)

6. Pary idealne

W internetowych dyskusjach bardzo często pojawia się następujące pytanie: „jaki jest optymalny układ stosu warstw w przypadku płytki N-warstwowej”, gdzie N to dowolna parzysta liczba wskazana przez autora pytania. Jak to zwykle bywa w przypadku

eprasa.pl 4202fe84f2

elektroniki, istnieje całkiem pokaźna liczba odpowiedzi, zaś każda z nich ma swoje grono zwolenników i przeciwników. Prowadzenie ścieżek sygnałowych tylko na warstwach wewnętrznych oraz zastosowanie jednolitych płaszczyzn miedzi na warstwach TOP i BOTTOM jest wprawdzie rozwiązaniem doskonałym pod względem EMC (zapewnia bowiem świetne ekranowanie całości mozaiki), jednak z oczywistych przyczyn stosowane bywa dość rzadko. Świetnym wyjściem jest także użycie stosu pokazanego schematycznie na **rysunku 6** – wszystkie warstwy sygnałowe znajdują się blisko odpowiedniej płaszczyzny masy (łatwiejsza kontrola impedancji, jasno określone ścieżki prądów powrotnych), dolna płaszczyzna masy (5) jest oddzielona od płaszczyzny zasilania (4) jedynie cienką warstwą rdzenia (duża pojemność rozproszona wspomagająca odsprzęganie), a odległość między zasilaniem (4), a wewnętrzną warstwą sygnałową (3) pozostaje możliwie duża (słabe sprzęganie zakłóceń). Rzecz jasna, istnieje także wiele innych kombinacji – w każdym przypadku odpowiednia analiza stosunków grubości i kolejności par warstw miedzi okaże się niezwykle przydatnym narzędziem optymalizacji PCB pod względem integralności sygnałów i zasilania oraz minimalizacji zakłóceń promieniowanych.



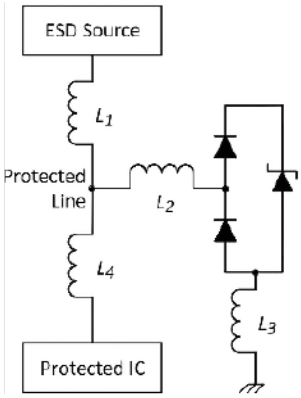
Rysunek 7. Schematyczne przedstawienie ułożenia bloków funkcjonalnych urządzenia typu mixed-signal z sekcją RF (http://t.ly/fwHe2)

7. Zasada dobrego sąsiedztwa

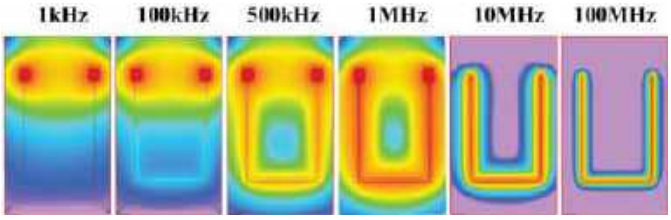
O prawidłowym (z punktu widzenia EMC) ułożeniu bloków urządzenia na obszarze PCB napisano już dziesiątki artykułów i rozdziałów w specjalistycznych monografiach. Niestety, w praktyce nad wyraz często zdarza się, że spełnienie oczekiwań teoretycznych okazałoby się niemożliwe lub nieopłacalnie trudne. Podczas codziennej pracy dość rzadko mamy możliwość całkowicie dowolnego rozłożenia podsystemów na PCB, gdyż każdy z nich współpracuje z jakimiś złączami, elementami kontrolnymi, czy nawet mechaniczną lub optyczną częścią złożonego urządzenia. O ile bowiem na dużej płytce drukowanej możemy często dość daleko odsunąć czułe układy pomiarowe od modułu radiowego lub przetwornicy DC/DC (**rysunek 7**), o tyle przypadku PCB przeznaczonej np. do urządzenia ubieralnego każdy blok „będzie miał blisko" do wszystkich pozostałych. I tutaj w grę wchodzi doświadczenie projektanta – to, co można, należy od siebie poodsuwać, zaś na dalszym etapie projektowania trzeba sięgnąć po inne narzędzia – optymalizację płaszczyzn masy i zasilania, filtry i obwody odsprzęgające, a nawet… techniki programowe.

8. Długość ma znaczenie

Każdy projektant PCB doskonale wie, jak ważne jest ułożenie kondensatorów odsprzęgających (oraz części pozostałych komponentów dyskretnych) możliwie jak najbliżej wyprowadzeń układu scalonego. Producenci półprzewodników nie pozwalają nam zresztą o tym zapomnieć i profilaktycznie umieszczają stosowne „przypominajki" w każdej nocie katalogowej lub aplikacyjnej. Niestety, wielu konstruktorów wciąż popełnia błędy w zakresie routingu innych komponentów, chociażby transili – a przecież w tym przypadku każdy, najkrótszy nawet odcinek ścieżki, także odgrywa rolę pasożytniczej indukcyjności (**rysunek 8**), która może skutecznie odseparować chroniony komponent od źródła ryzykownych przepięć lub (w przypadku błędnego ustalenia proporcji poszczególnych odcinków mozaiki) – utrudnić ładunkom podążanie do transila i przekierować sporą ich część do wrażliwego układu scalonego. Dlatego też przed wysłaniem „gerberów" do fabryki warto sumiennie prześledzić (i w razie potrzeby skorygować) layout wejść oraz wyjść urządzenia (choć innym ścieżkom także się to należy).



Rysunek 8. Model zastępczy połączenia ochronnika TVS z układem scalonym (http://t.ly/C9IQ1)



Rysunek 9. Zależność geometrycznego rozkładu gęstości prądu powrotnego od częstotliwości sygnału (http://t.ly/EZ4Ql)

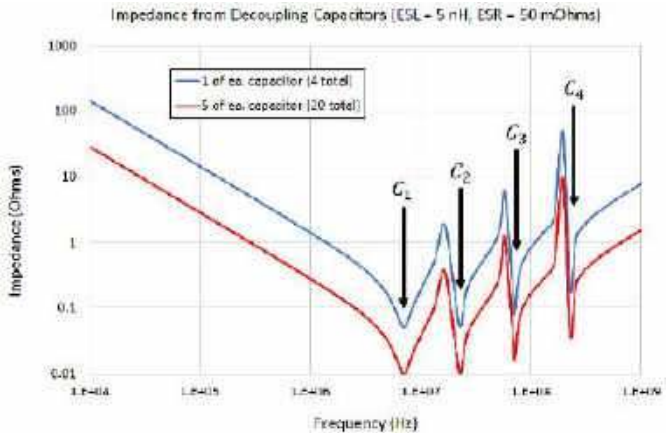
9. Tam i z powrotem

Ważnym zagadnieniem (o którym już zresztą wspomnieliśmy w niniejszym opracowaniu) jest optymalizacja projektu PCB pod względem zapewnienia właściwych ścieżek dla prądów powrotnych. W uproszczeniu możemy zdefiniować to zjawisko następująco: każdy przepływ ładunków z punktu A do punktu B na płytce drukowanej wiąże się z powstaniem prądu powrotnego, płynącego przez masę w kierunku przeciwnym (tj. z powrotem do źródła). Ścieżka powrotna zależy jednak od częstotliwości, gdyż „leniwy" prąd „wybierze" sobie drogę o najniższej możliwej impedancji. W zakresie częstotliwości niskich (do około 100 kHz) znaczna większość prądu powróci najkrótszą ścieżką, jednak w wyższych pasmach zacznie dominować składowa urojona impedancji – prąd powrotny, przez sprzężenie indukcyjne, będzie próbował wrócić do źródła tuż pod ścieżką główną (lub obok niej). Im więcej przeszkód na drodze prądu powrotnego, tym większe ryzyko nieprzewidzianych sprzężeń i emisji w różnych częściach obwodu drukowanego. To jest właśnie główny powód, dla którego prawidłowe zaprojektowanie PCB z zastosowaniem dzielonych płaszczyzn masy (zalecane w wielu materiałach dotyczących układów mixed-signal) jest tak trudne, a błędy na tym polu często stają się przyczyną jeszcze większych problemów.

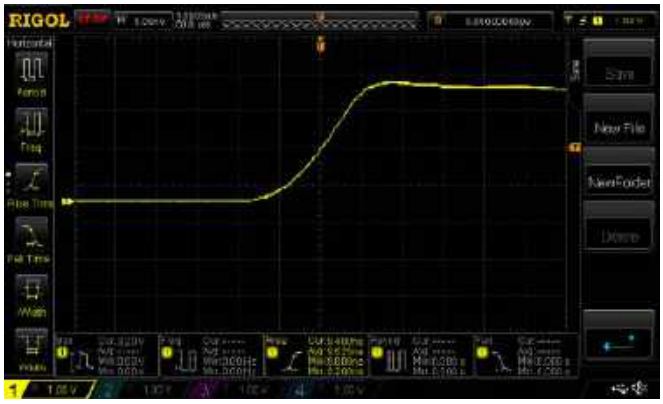
10. Wespół w zespół

Bądźmy szczerzy – zdarza nam się lenistwo. My, elektronicy, mamy w swojej pracy bardzo często do czynienia z podobnymi lub wręcz identycznymi rozwiązaniami układowymi, powtarzanymi w ramach różnych projektów. W samym zastosowaniu sprawdzonych „w boju" rozwiązań nie ma oczywiście nic złego – co innego jednak, jeżeli bezrefleksyjnie powtarzamy je w każdej sytuacji. Dobrym przykładem będzie tutaj dobór wartości kondensatorów odsprzęgających. W znakomitej większości przypadków do tego celu stosuje się pojemności 100 nF + 1...10 μF, ale przedmiotem głębszego zastanowienia bywa zwykle tylko wartość większego kondensatora, odpowiedzialnego w układzie za filtrację tętnień i zaburzeń przewodzonych w niskim paśmie częstotliwości. W szybkich układach cyfrowych i RF musimy jednak operować w pełnym spektrum

eprasa.pl 4202fe84f2



Rysunek 10. Wpływ zastosowania równoległych kondensatorów o różnych pojemnościach (C1: 100 nF, C2: 10 nF, C3: 1 nF, C4: 100 pF) na wypadkową impedancję obwodu odsprzęgającego zasilanie (http://t.ly/djEqG)



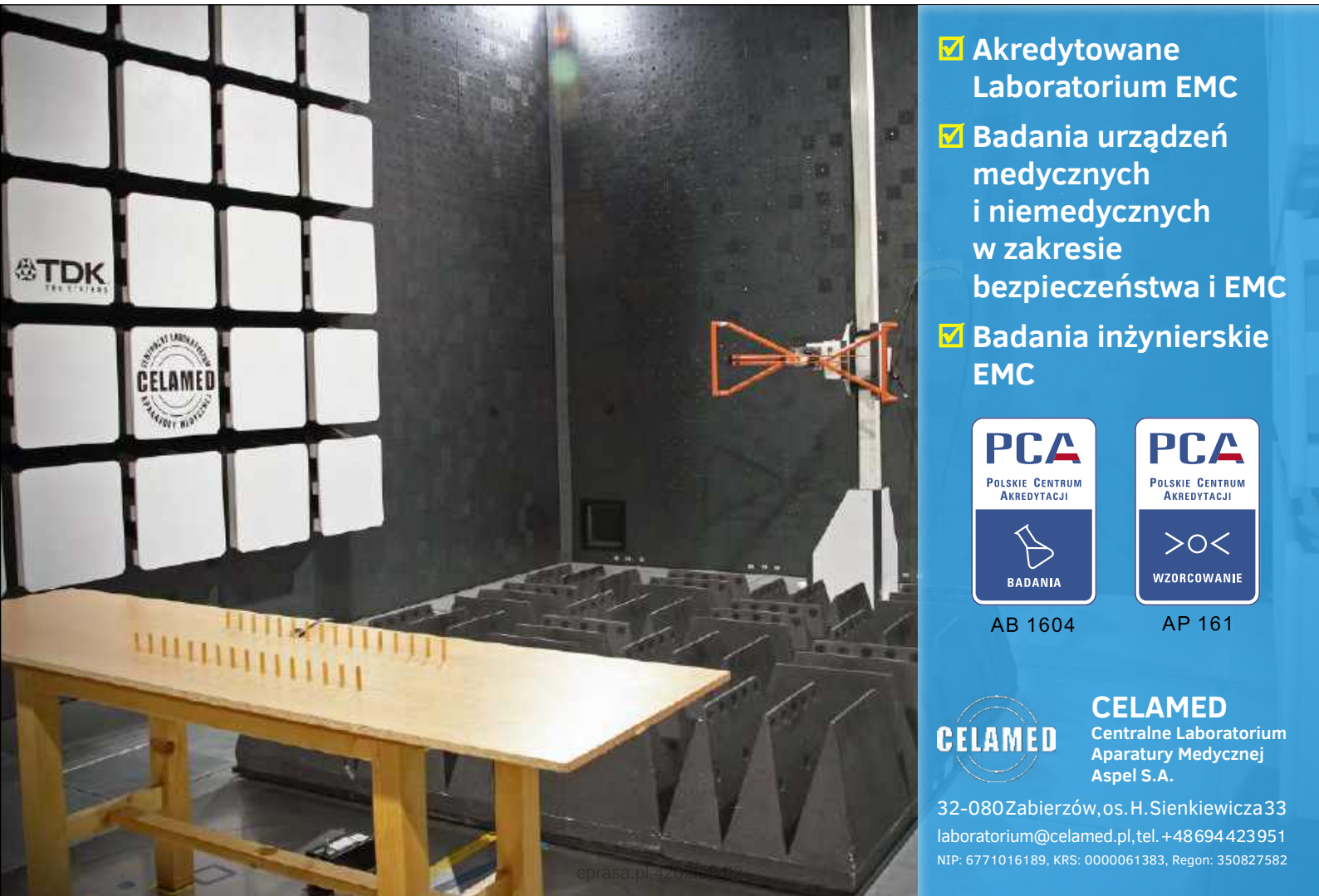
Rysunek 11. Przebieg na linii GPIO mikrokontrolera STM32 ustawionej w trybie low speed (czas narastania wynosi 9,5 ns) – http://t.ly/njNCm

używanym w danym urządzeniu – dlatego też często spotykane jest rozwiązanie, w którym łączymy równolegle całą baterię kondensatorów, np. 3 lub 4 elementy o pojemnościach różniących się pomiędzy sobą kolejno o rząd wielkości. Takie rozwiązanie pozwala znacząco obniżyć impedancję i efektywniej stłumić także zaburzenia o najwyższej częstotliwości (w tym harmoniczne wyższych rzędów).

11. Spiesz się powoli

W świecie komputerów dość popularną techniką jest podkręcanie częstotliwości taktowania procesora w celu uzyskania wyższej wydajności obliczeniowej. Na gruncie projektowania elektroniki także mamy do czynienia ze stopniowym zwiększaniem maksymalnej częstotliwości zegara mikrokontrolerów i procesorów aplikacyjnych – nie oznacza to jednak, że zawsze należy „jechać po bandzie". Z punktu widzenia kompatybilności elektromagnetycznej (ale także pod względem zużycia energii) wskazane jest bowiem możliwie jak najmocniejsze obniżenie częstotliwości taktowania, co pozwala wydatnie zredukować pasmo emitowanych zaburzeń radiowych. Podobnie dzieje się także w przypadku linii sygnałowych – przykładowo, jeżeli do wiązki kablowej lub przewodu FPC/FFC wprowadzamy sygnały cyfrowe, warto lekko złagodzić ich zbocza, ponieważ dłuższe czasy narastania i opadania to także niższe pasmo zaburzeń. Współczesne mikrokontrolery ułatwiają zresztą to zadanie – przykładowo układy STM32 oferują możliwość programowego spowolnienia dynamiki przełączania portów GPIO poprzez zapis odpowiedniego rejestru konfiguracyjnego (**rysunek 11**).

REKLAMA



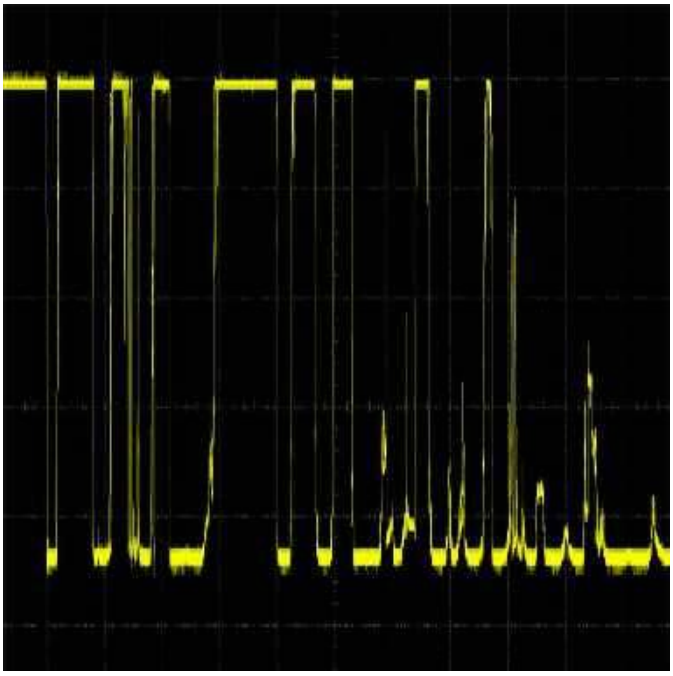
Fotografia 1. Ekranowanie obudowy tworzywowej za pomocą napylonej warstwy lakieru przewodzącego na bazie miedzi (http://t.ly/c_6SV)

12. Zasłona milczenia

Nawet najlepiej zaprojektowana płytka drukowana potrzebuje czasem dodatkowego wsparcia. Jeżeli dane urządzenie jest wyjątkowo wrażliwe na zakłócenia (jak np. czułe obwody pomiarowe lub tory RF) bądź też samo generuje silne zaburzenia, niezbędne może okazać się zastosowanie ekranowania. Oprócz użycia metalowych obudów, można także posiłkować się napylaniem warstw lakieru na bazie proszków metalicznych na wewnętrzną powierzchnię tworzywowej obudowy (**fotografia 1**). Szczególnie dużo uwagi należy poświęcić wszelkim otworom – niezależnie od tego, czy są one wykonane pod złącza, wyświetlacze, czy też wentylatory, zawsze będą stanowiły otwarte wrota dla zaburzeń propagujących do/z wnętrza urządzenia. W niektórych przypadkach sensowne będzie także rozważenie ekranów lokalnych, wykonanych w formie metalowych puszek lutowanych lub zatrzaskiwanych w specjalnych zaciskach na samej płytce drukowanej. Współczesna elektronika dysponuje ponadto pełnym arsenałem dodatkowego wyposażenia ekranującego, do którego należą: dławnice przewodzące, metaliczne siatki do osłony otworów wentylacyjnych, panele do ekranowania wyświetlaczy, czy też specjalne zaciski łączące oploty kabli z szyną DIN lub metalową obudową. Warto pamiętać, że sama obecność ekranu nie wystarczy – trzeba jeszcze zadbać o odpowiednie, niskoimpedancyjne połączenie go z masą urządzenia lub uziemieniem, gdyż bez tego nawet najlepszy ekran nie zapewni wymaganego poziomu tłumienia zaburzeń RFI.

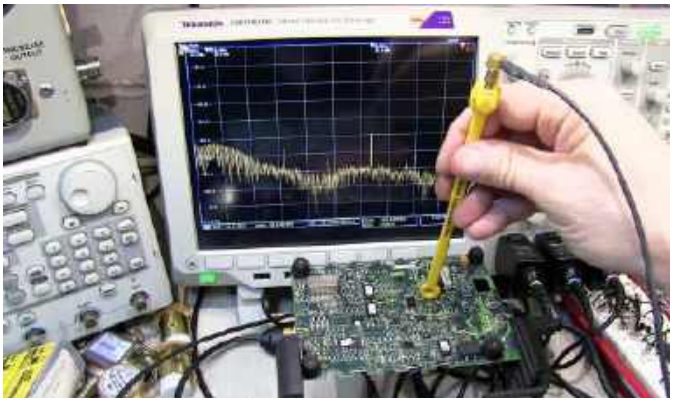
13. Software (też) ma tę moc

Zagadnienia EMC tylko z pozoru mogą stanowić pole do popisu wyłącznie dla projektantów hardware'u. W rzeczywistości jednak sporo do zdziałania mają tutaj także programiści systemów wbudowanych. O ile bowiem ograniczenie emisji zaburzeń po stronie firmware'u sprowadza się głównie do opisanych wcześniej sposobów (zastosowanie wolniejszego taktowania i mniej stromych zboczy sygnałów cyfrowych), o tyle już w zakresie odporności istnieje dużo więcej zadań dla programistów. Należy bowiem pamiętać, że testy EMC dotyczą nie tylko tego, czy dany sprzęt „przeżyje" wyładowanie z symulatora ESD bądź generatora zaburzeń EFT – niebagatelne znaczenie, zwłaszcza w przypadku urządzeń krytycznych z punktu widzenia bezpieczeństwa (np. aparatury medycznej, elektroniki pokładowej w branżach motoryzacyjnej i lotniczej, czy też sprzętu militarnego) ma także odpowiednia reakcja na silne zakłócenia, które często mogą całkowicie zdezorientować procesor bądź nasycić wzmacniacze w czułym torze pomiarowym. Najprostszy



Rysunek 12. Przykładowy oscylogram prezentujący drgania styków przycisku (http://t.ly/QKsHf)

przykład to zastosowanie odpowiednio solidnych procedur debouncingu, czyli eliminacji drgań styków. Jeżeli program zostanie odpowiednio zabezpieczony, to poradzi sobie nie tylko z mechanicznymi wibracjami spowodowanymi sprężystością blaszek kontaktowych przycisku, ale także z zaburzeniami przewodzonymi lub indukowanymi w okablowaniu wewnętrznym. Inne techniki obejmują m.in. zastosowanie watchdogów, układów POR i BOR, zewnętrznych obwodów nadzorczych, czy też obszarów pamięci podtrzymywanych bateryjnie (backup domain).



Fotografia 2. Zastosowanie sondy pola bliskiego (H) do lokalizacji źródła zakłóceń (http://t.ly/xUDMQ)

14. Dzwoni... ale w którym kościele?

Niezwykle wartościowym narzędziem podczas wstępnego „przeszukiwania" PCB w celu określenia źródeł niepożądanych emisji są sondy pola bliskiego (**fotografia 2**). Pozwalają one nie tylko na ustalenie przybliżonego położenia źródła (np. układu scalonego odpowiedzialnego za generowanie zaburzeń), ale wręcz na zlokalizowanie najbardziej problematycznych ścieżek. Duże znaczenie ma tutaj kierunkowy charakter sond pola magnetycznego (H-field), których czułość jest największa tylko w jednej orientacji względem badanego przewodnika. Pewnym problemem okazuje się silny wpływ odległości sonda-PCB na amplitudę mierzonego sygnału – przy odrobinie zachodu da się jednak tak przygotować stanowisko pomiarowe, by możliwe było prowadzenie pomiarów porównawczych (np. przed i po wymianie dławika lub kondensatora odsprzęgającego).

eprasa.pl 4202fe84f2

Jeszcze bardziej powtarzalne warunki stwarzają komory TEM – umożliwiają one ponadto przekonwertowanie wyników pomiaru na postać porównywalną (w pewnym ograniczonym zakresie) z wynikami otrzymywanymi podczas testów w pełnowymiarowych komorach SAC. I choć konwersja z pola bliskiego na pole dalekie nie jest zagadnieniem prostym pod względem matematycznym, to mimo wszystko zastosowanie TEM w testach inżynierskich okazuje się niezwykle cennym narzędziem, daje bowiem pewną zgrubną aproksymację wyników, których należy spodziewać się po pełnych badaniach EMC w profesjonalnym laboratorium. Mało tego – przy zastosowaniu odpowiedniego wzmacniacza mocy oraz generatora RF lub wyjścia TG (generatora śledzącego) dostępnego w analizatorze widma, możliwe jest wstępne testowanie odporności prototypu na zakłócenia propagowane drogą radiową.

Podsumowanie

W artykule zebraliśmy zestaw czternastu wybranych porad dotyczących prawidłowego podejścia do projektowania urządzeń elektronicznych w zgodzie z normami EMC. Staraliśmy się przy tym objąć możliwie szeroki zakres zagadnień – począwszy od wstępnej fazy procesu projektowego, aż po wykonywanie badań inżynierskich. Czytelników zainteresowanych omówioną tematyką gorąco zachęcamy do samodzielnych poszukiwań oraz… do studiowania oryginalnych norm, gdyż tylko takie podejście pozwala dogłębnie zrozumieć złożony, ale także bardzo interesujący świat kompatybilności elektromagnetycznej.

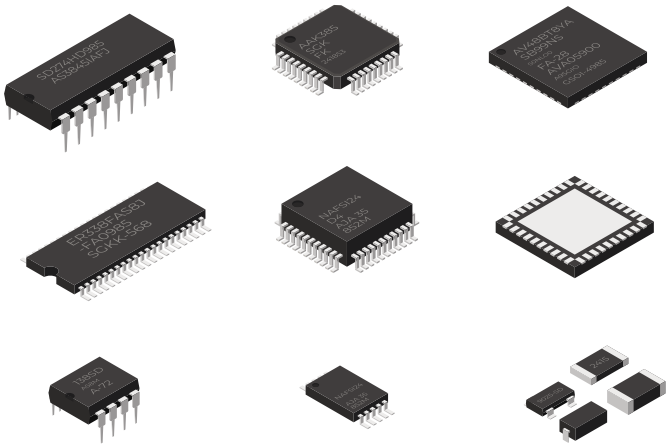
inż. Przemysław Musz, EP

REKLAMA

eprasa.pl 4202fe84f2

Testowanie układów scalonych

Główną zaletą analizy odporności urządzenia na poziomie układu scalonego jest fakt, że takie badanie nie wymaga uwzględniania wpływu konstrukcji owego układu na kompatybilność elektromagnetyczną (EMC). Wspomniana analiza obejmuje przykładowo projekt płytki drukowanej, charakter i dostępność złącza oraz obudowę. W artykule opisano zależność między testami na poziomie urządzenia oraz układu scalonego (IC).



Wprowadzenie

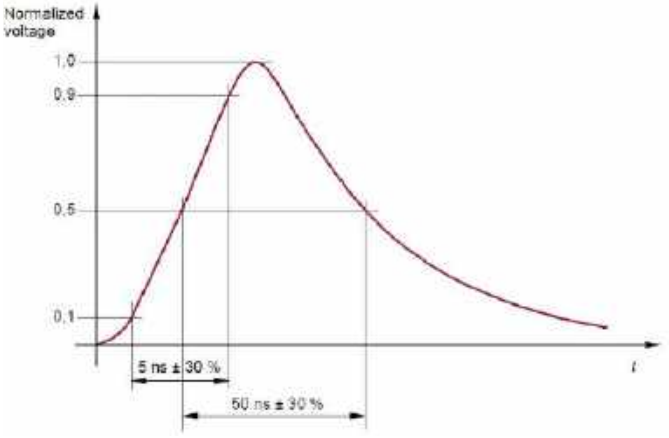
Zapewnienie zgodności z normami EMC staje się coraz bardziej wymagającym zagadnieniem. Postęp technologiczny pozwolił na zmniejszenie rozmiaru elementów, jednak wykreował on również szczególne wyzwanie, jakim jest kontrolowanie odporności urządzenia na zakłócenia elektromagnetyczne.

Aktualne wymagania dotyczące odporności zwiększają znacznie koszty projektu i produkcji urządzeń. Testując odporność na poziomie komponentów (tj. układów scalonych), można je jednak łatwiej zidentyfikować i podjąć działania zaradcze. Wyniki testów odporności pozwalają na wstępną selekcję określonych układów scalonych (w tym układów ASIC) do dalszego rozwoju wyrobu. Dodatkowo mogą one zostać włączone do analizy układów scalonych oraz przyczynić się do optymalizacji komponentów.

Wprawdzie w branży istnieją już procedury testowe, które umożliwiły zgromadzenie bogatego doświadczenia w ocenie odporności układów scalonych, ale koncepcja stojąca za obecną metodą testowania wprowadza tu pewną zmianę. Polega ona na przykładaniu impulsów zakłócających bezpośrednio do pinów testowanych układów scalonych. Kształt i amplituda zastosowanych zakłóceń są specjalnie tak dobrane, aby naśladować typowe zjawiska, na jakie narażony byłby układ scalony podczas standardowego testu odporności urządzenia zawierającego ten układ lub podczas pracy w zakłóconym środowisku.

W czasie pracy sprzęt techniczny, obiekty i urządzenia są zazwyczaj zasilane przez pulsujące sygnały zakłócające. W związku z tym standardowe testy urządzeń symulują na przykład generowanie iskry na styku przełącznika (burst) lub wyładowanie elektrostatyczne (ESD).

W przypadku standardowej metody testowania obserwuje się odporność w trybie aktywnym (tj. zachowanie działających układów scalonych zasilanych napięciem). Kryterium zaliczenia testu stanowi niezakłócone funkcjonowanie układu scalonego.



Rysunek 1. Krzywa czasowa impulsu burst (szybki test odporności na stany przejściowe) zgodna z IEC 61000-4-4

Wpływy środowiskowe lub sam test mogą powodować powstawanie napięć i prądów znacznie wyższych niż określone maksymalne wartości dla układów scalonych.

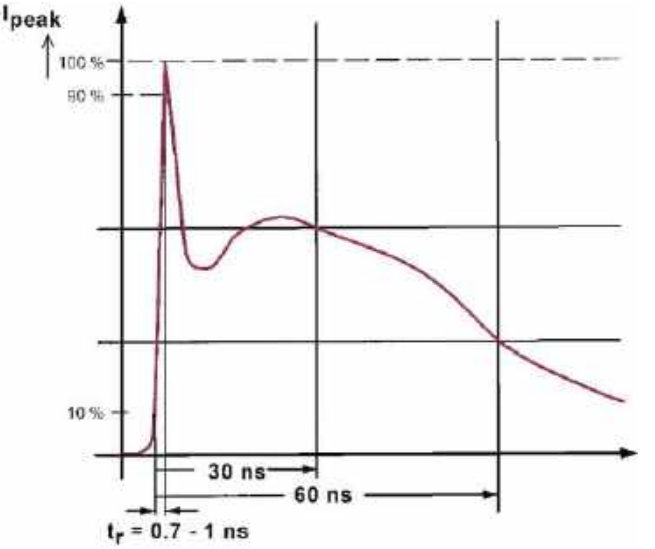
Analiza odporności na poziomie układu scalonego ma tę zaletę, że nie wymaga uwzględnienia wpływu konstrukcji urządzenia na EMC. Obejmuje to na przykład projekt płytki drukowanej, rodzaj i dostępność złącza lub konstrukcję obudowy.

Co więcej, podczas testowania odporności układu scalonego efekty zakłóceń są mniej wyraźne niż podczas testowania całego urządzenia – skutkuje to lepszą powtarzalnością wyników testu. W niniejszym artykule opisano połączenie między testem urządzenia a testem ziarnistości pinów układu scalonego.

Testowanie urządzeń

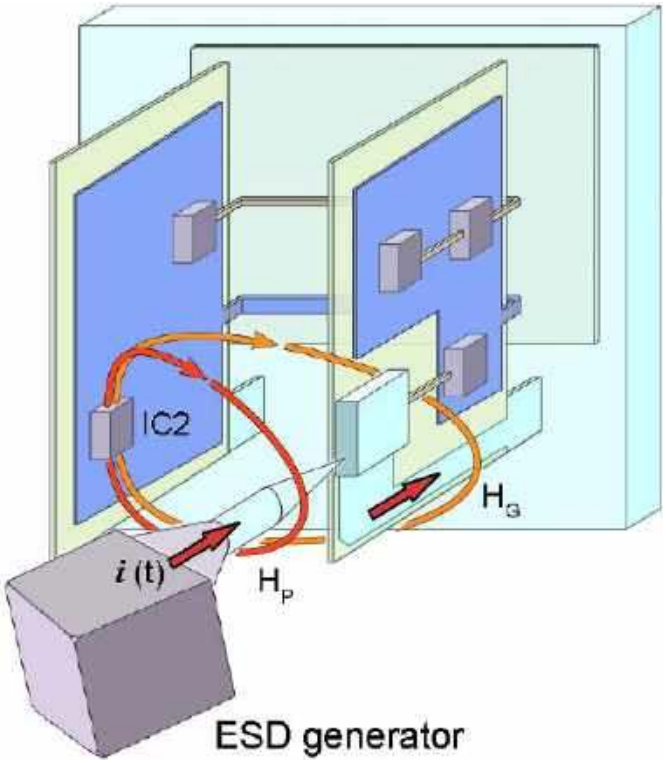
Ze względu na ciągły wzrost cyfryzacji i powszechne stosowanie podzespołów elektronicznych wiele urządzeń wymaga wysokiego stopnia odporności. Szczególnie ważne jest przeprowadzanie testów odporności na zakłócenia impulsowe podczas badania obiektów technicznych i urządzeń. Wynika to z faktu, że będą one narażone na zakłócenia impulsowe również podczas pracy i standardowego użytkowania.

Odpowiednie impulsy zakłócające (burst i ESD) w testach odporności opisano w następujących normach: IEC 61000-4-4 „Szybki test

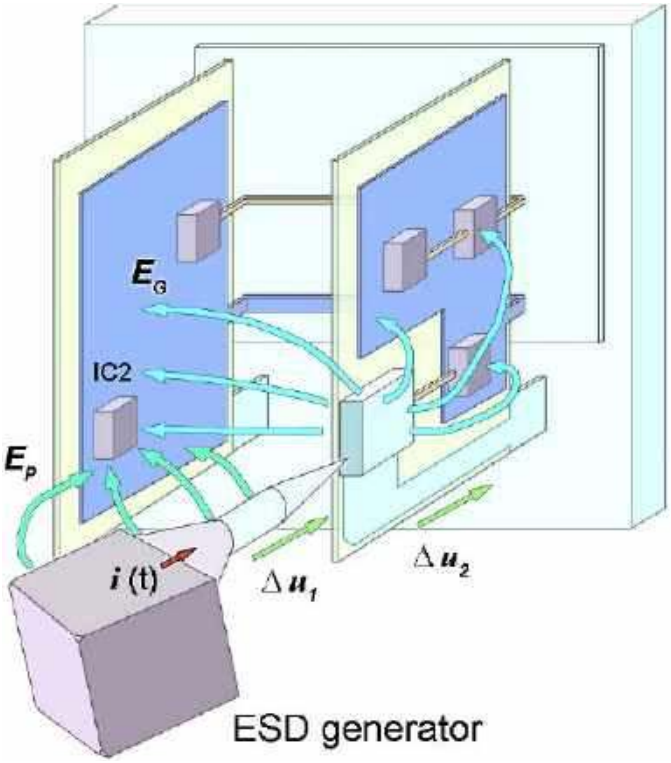


Rysunek 2. Krzywa czasowa impulsu ESD (test odporności na wyładowania elektrostatyczne) zgodna z IEC 61000-4-2

eprasa.pl 4202fe84f2



Rysunek 3. Mechanizmy sprzężenia pola magnetycznego w urządzeniu elektrycznym



Rysunek 4. Mechanizmy sprzężenia pola E w urządzeniu elektrycznym

odporności na zakłócenia przejściowe (burst)" oraz IEC 61000-4-2 „Odporność na wyładowania elektrostatyczne (ESD)":

Przy impedancji źródła wynoszącej 50 Ω do urządzenia przykładane są impulsy o minimalnym napięciu ±2 kV, ze zboczem narastającym trwającym 5 ns i zboczem opadającym trwającym 50 ns (połowa amplitudy – patrz **rysunek 1**). To właśnie z tych impulsów składają się impulsy BURST. Przy impedancji źródła wynoszącej 330 Ω do urządzenia przykładane są impulsy o napięciu co najmniej ±6 kV ze zboczem narastającym 0,7 ns i zboczem opadającym 5 ns (połowa amplitudy – **rysunek 2**).

Podstawowe zasady testowania urządzeń pod kątem zakłóceń

W celu badania odporności do urządzenia przykładane jest zakłócenie w kształcie impulsu zarówno w jego zamierzonej lokalizacji, jak i niezależnie od niej. Impuls pierwotnego zakłócenia (ESD lub burst) napięcia u(t) przyłożony do urządzenia powoduje przepływ impulsowego prądu zakłócającego i(t) przez urządzenie (**rysunek 3**). Dwa podstawowe mechanizmy – sprzężenie magnetyczne i sprzężenie pola elektrycznego (pole E) – choć występują jednocześnie, powinny być rozpatrywane indywidualnie.

1. Sprzężenie pola magnetycznego H (indukcyjne)

Interferencja pola magnetycznego H(t) tworzy się wokół przewodnika przenoszącego prąd interferencyjny i(t). Im niższa impedancja obwodu pierwotnego, tym większy przepływ prądu, a tym samym silniejsze pole magnetyczne H.

W idealnym przypadku natężenie pola wokół prostego przewodnika jest określone przez wzór (1):

$$H(t) = \frac{1}{2 \cdot \pi \cdot r} \quad (1)$$

Zakłócenia pola magnetycznego przenikają przez urządzenie i otaczające je komponenty oraz przez płytki drukowane. Pętle przewodników znajdują się na płytkach drukowanych lub w komponentach (np. układ scalony, rysunek 3). Zakłócenia pola magnetycznego generują wtórne napięcie zakłócające u_sec(t) na indukcyjności L – patrz wzór (2):

$$u\_sec(t) = -L \cdot \frac{di}{dt} \quad (2)$$

To napięcie zakłócające potencjalnie występuje na przykład w obwodach wewnętrznych układu scalonego, co może spowodować nieprawidłowości w działaniu tego komponentu.

2. Sprzężenie pola elektrycznego

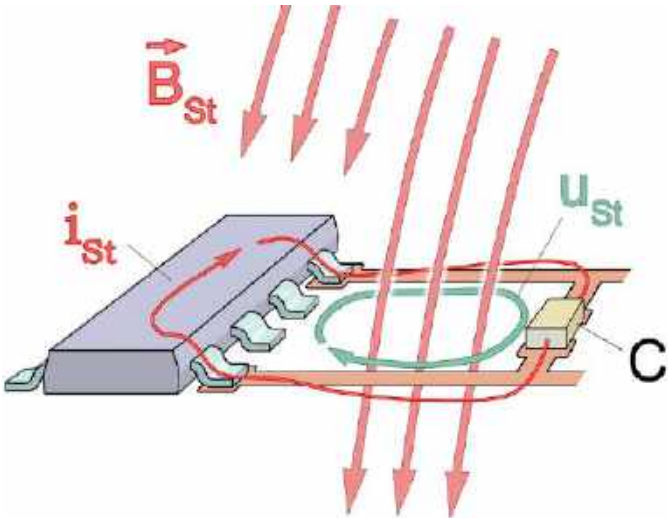
Podstawowy impuls zakłócający (ESD lub burst) napięcia u(t) przyłożonego do urządzenia powoduje spadek napięcia na zespole (**rysunek 4**). Pole elektryczne E(t) uzyskuje się z różnicy napięć wzdłuż obserwowanej geometrii. Im wyższa impedancja pierwotnej ścieżki prądu, tym wyższa różnica napięć, a tym samym wyższe wynikowe pole E. Pole E przenosi zakłócenia jako prąd przesunięcia i(t) w pętlach wtórnych, takich jak kable sygnałowe i/lub styki układu scalonego, poprzez sprzężenie pojemnościowe. Proces zachodzi za pośrednictwem pojemności sprzęgającej C, która mieści się w zakresie fF – patrz wzór (3). Napięcie interferencyjne jest wytwarzane na rezystancji wewnętrznej pętli wtórnej przez sprzężony pojemnościowo prąd i(t). To napięcie zakłócające może być obecne na pinie układu scalonego i wyzwalać zakłócenia w jego funkcjonowaniu.

$$Displacement\ current\ i(t) = C \cdot \frac{du}{dt} \quad (3)$$

W obu przypadkach (sprzężenie spowodowane przez pole magnetyczne lub pole E) zakłócenie jest zmniejszane o współczynnik jeden (współczynnik urządzenia) podczas przejścia z pierwotnego obwodu zakłócającego do pętli wtórnej i jest zlokalizowane na aktywnym układzie elektronicznym (np. na układzie scalonym). Napięcia zakłócające 6 kV, które zwykle występują podczas testu urządzenia ESD, zmniejszają się do wartości leżącej w przedziale od 0,1 V do kilkuset woltów. Spadek ten zależy od impedancji w pierwotnym i wtórnym obwodzie zakłócającym.

Układy scalone zamontowane na płytce drukowanej są narażone na działanie pól magnetycznych H(t) i elektrycznych E(t).

eprasa.pl 4202fe84f2



Rysunek 5. Model sprzężenia z układem scalonym poprzez pole magnetyczne

Sprzężenie magnetyczno-polowe (indukcyjne) w układach scalonych

Gęstość strumienia magnetycznego Bst(t) przenika przez najmniejsze pętle przewodnika (np. między układem scalonym a dołączonym kondensatorem odsprzęgającym – **rysunek 5**). Napięcie Ust jest indukowane w pętli obwodu przez strumień magnetyczny Φ, jak opisano w (4) i (5):

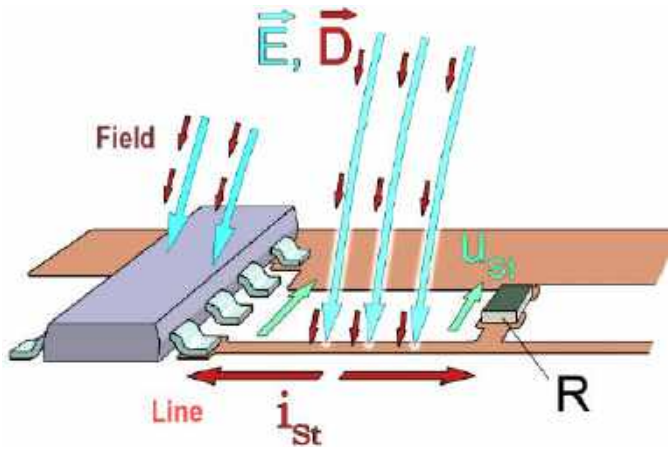
$$\Phi = \int BdA \quad (4)$$

$$U_{st} = \frac{d\Phi}{dt} \quad (5)$$

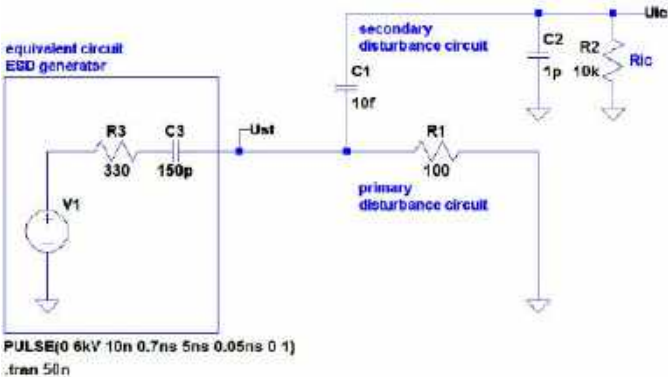
Napięcie Ust doprowadza prąd zakłócający do układu scalonego. Impedancja tego źródła zakłóceń jest niska ze względu na formowanie się sygnału w pętli przewodnika. Może to prowadzić do powstawania prądów Ist (t) o wysokim natężeniu.

Sprzężenie pola elektrycznego E (pojemnościowe) w układach scalonych

Natężenie pola elektrycznego E(t) lub związany z powierzchnią przewodzącą prąd przemieszczenia D(t) generuje zakłócenia Ist(t) w przewodniku (**rysunek 6**). Zakłócenia te powodują wzrost napięcia Ust(t) na powierzchni przewodzącej, który może zniekształcać sygnały logiczne, przesyłane przez ścieżki. Prąd przesunięcia Ist(t) może również propagować do układów scalonych i wyzwalać dalsze zakłócenia. Źródło zakłóceń „pola elektrycznego" ma wysoką impedancję.



Rysunek 6. Model sprzężenia z układem scalonym poprzez pole E



Rysunek 7. Przykład szacowania napięcia zakłócającego na styku układu scalonego podczas sprzęgania pola zakłóceń elektrycznych

Symulacja

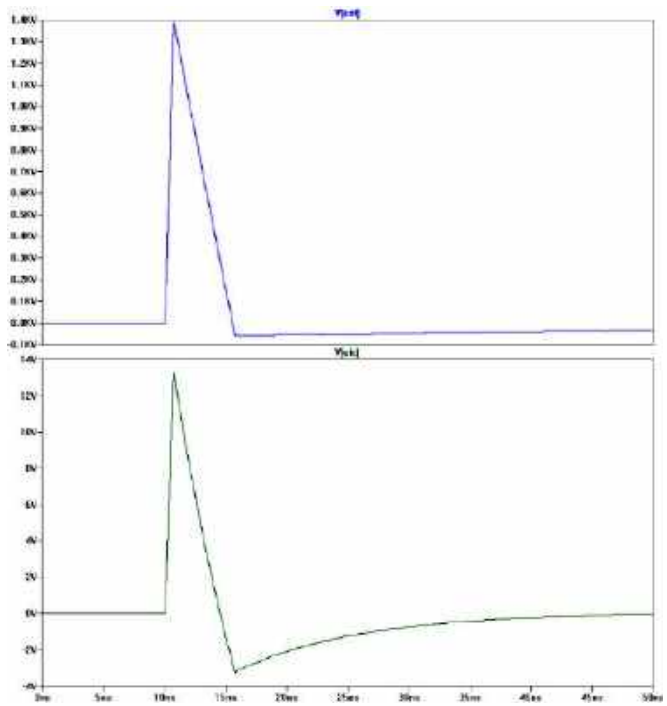
Poniższe symulacje (**rysunki 7...11**) bazują na pewnych uproszczeniach. Na rysunku 7 generowanie impulsu ESD zostało znacznie uproszczone. Obwód równoważny opiera się na zasadach sprzężenia pojemnościowego, pokazanych na rysunku 4.

Szacowanie napięcia zakłócającego na układzie scalonym

Dodatni impuls zakłócający o amplitudzie 6 kV jest wstrzykiwany jako wyładowanie kontaktowe w pierwotnym obwodzie zakłócającym (rysunek 7). Pik napięcia zakłócającego (Ust) o wartości 1,4 kV powstaje na impedancji pierwotnej R1 (rysunek 8). Napięcie zakłócające 13,5 V pozostaje w kontakcie ze stykiem układu scalonego o wysokiej impedancji poprzez sprzężenie pojemnościowe (sprzężenie pola E) przez C1. Impedancja pierwotnego obwodu zakłócającego może być znacznie większa (1 kΩ), przez co napięcie zakłócające na pinie układu scalonego może przekraczać 100 V, co grozi znacznym przekroczeniem specyfikacji maksymalnych wartości znamionowych układów scalonych.

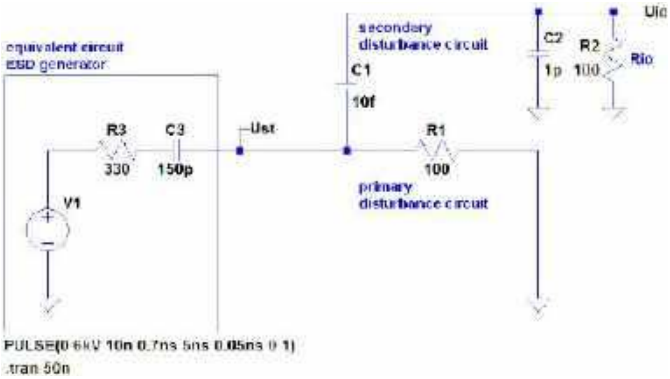
Tryby sprzęgania

Rodzaj sprzężenia zależy również od stosunku impedancji źródła do impedancji obciążenia, tj. do impedancji wejściowej układu scalonego.

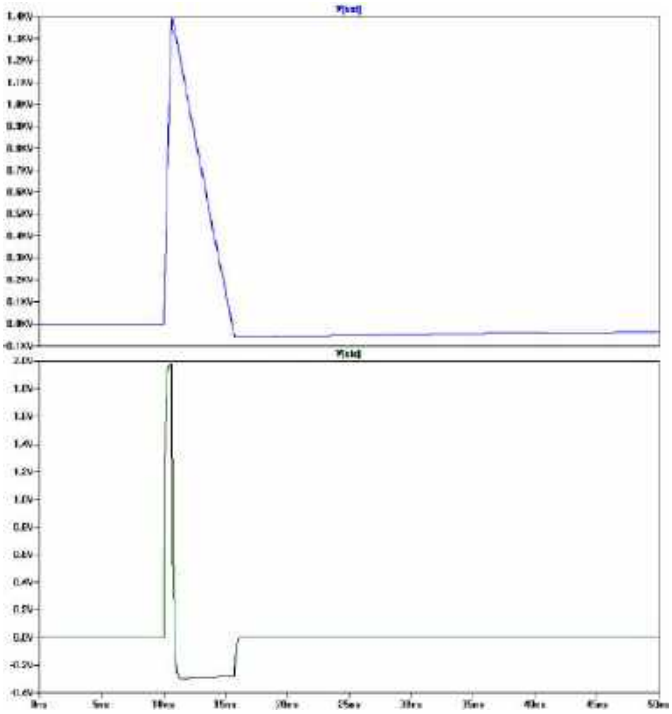


Rysunek 8. Krzywa i wartości szczytowe sprzężonego napięcia zakłócającego w obwodzie pierwotnym i wtórnym

eprasa.pl 4202fe84f2



Rysunek 9. Obwód równoważny różnicujących efektów sprzężenia interferencyjnego przez pole elektryczne



Rysunek 10. Krzywa różnicująca efekty sprzężenia interferencyjnego przez pole elektryczne

Sprzężenie pola elektrycznego E (pojemnościowe)

Przy nachyleniu 1 ns skutkującym maksymalną szybkością transferu 1 GHz zakłada się, że pojemność sprzęgająca C1 wynosi 1 pF. Impedancja X tejże pojemności przyjmuje wtedy wartość 159 Ω. Gdy rezystancja wejściowa układu scalonego wynosi 10 kΩ, jest ona znacznie większa niż rezystancja źródła (impedancja C1). W związku z tym sąsiedni impuls interferencyjny na R2 (na układzie scalonym) ma taki sam kształt fali, jak pierwotny impuls interferencyjny. Powoduje to proporcjonalny podział napięcia z dzielnikiem pojemnościowym C1, C2.

$$Ri_{IC} \gg X_{C1}$$

Zupełnie inne warunki uzyskuje się, gdy impedancja obciążenia jest mniejsza niż impedancja źródła. Poniższy przykład zakłada R2=100 Ω, C1=10 fF.

$$Ri_{IC} \ll XC1$$

W tych warunkach główny impuls zakłócający jest rozróżniany w układzie scalonym (rysunki 9 i 10).

Sprzężenie pola H (magnetycznego)

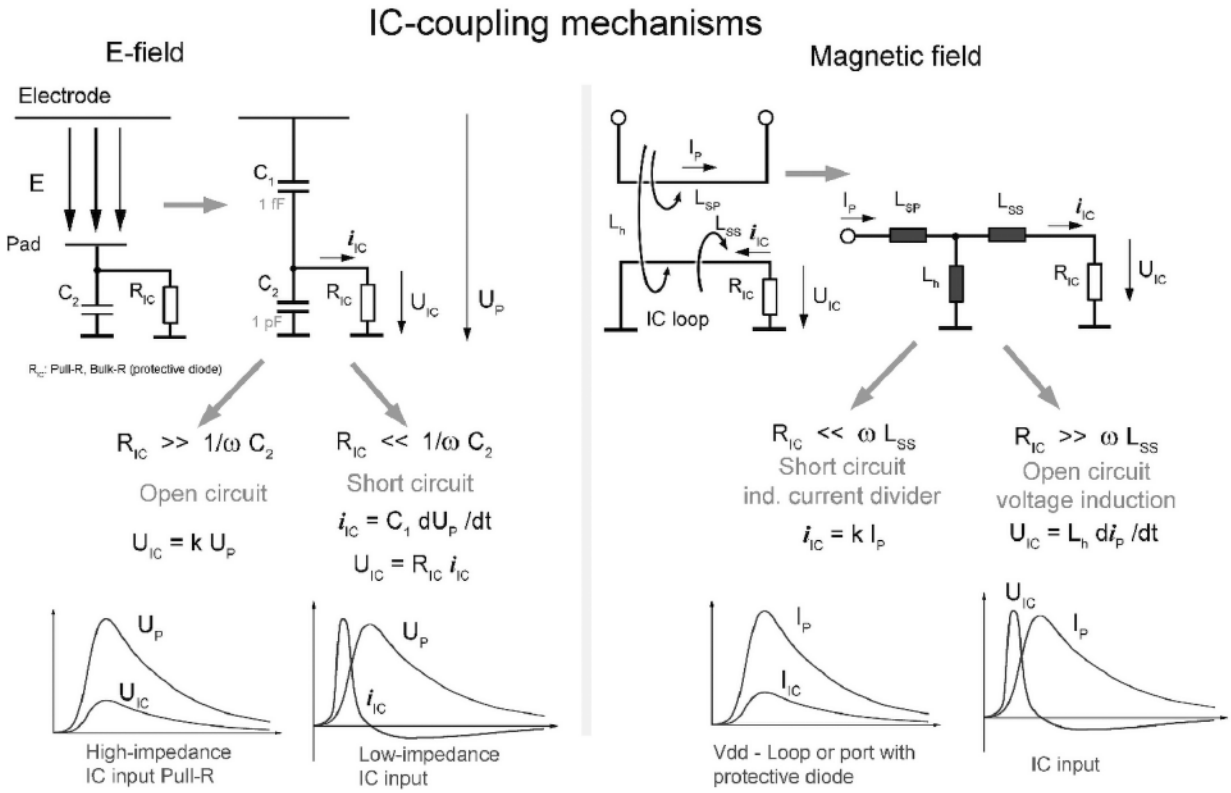
W przypadku sprzężenia z polem H warunki są odwrotne. W stanie spoczynku:

Ric » Xss jest zróżnicowany,

Ric ‹ Xss jest podzielony prądowo.

Podsumowanie mechanizmów sprzęgania pokazano na rysunku 11. W przypadku sprzężenia pola H przyjmuje się schemat zastępczy obwodu transformatora z indukcyjnością główną (Lh) oraz indukcyjnością upływu (Ls).

DACPOL Sp. z o.o.
www.dacpol.pl



Rysunek 11. Obszary robocze mechanizmów sprzęgania układów scalonych: podział/różnicowanie prądów i napięć

eprasa.pl 4202fe84f2

Badania EMC w ofercie Centralnego Laboratorium Aparatury Medycznej CELAMED

Akredytowane przez Polskie Centrum Akredytacji Centralne Laboratorium Aparatury Medycznej CELAMED zajmuje się działalnością pomiarowo-badawczą głównie w zakresie bezpieczeństwa elektrycznego, funkcjonowania zasadniczego i kompatybilności elektromagnetycznej medycznych urządzeń elektrycznych i medycznych systemów elektrycznych oraz wszelkiego rodzaju urządzeń elektronicznych i elektrycznych.

Drugim obszarem działalności Laboratorium jest wykonywanie wzorcowań urządzeń w dziedzinie wielkości elektrycznych (prąd, napięcie DC i AC, rezystancja) oraz w dziedzinie czasu i częstotliwości. Wzorcowania obejmują m.in. generatory oraz kalibratory napięć i prądów DC i AC, rezystory stałe oraz regulowane, mierniki napięć, prądów DC i AC, mierniki rezystancji, multimetry i oscyloskopy oraz generatory czasu i częstotliwości, a także mierniki czasu i częstotliwości.

Laboratorium ma ponad 30-letnie doświadczenie w zakresie badań medycznych i niemedycznych urządzeń elektrycznych oraz pomiarów elektrycznych, wzorcowania i nadzorowania sprzętu kontrolno-pomiarowego. Pierwsze pomiary wykonywane były w 1986 r.

CELAMED posiada dwa certyfikaty akredytacji:

- nr AB 1604 dla laboratorium badawczego,
- nr AP 161 dla laboratorium wzorcującego,

nadane przez Polskie Centrum Akredytacji.

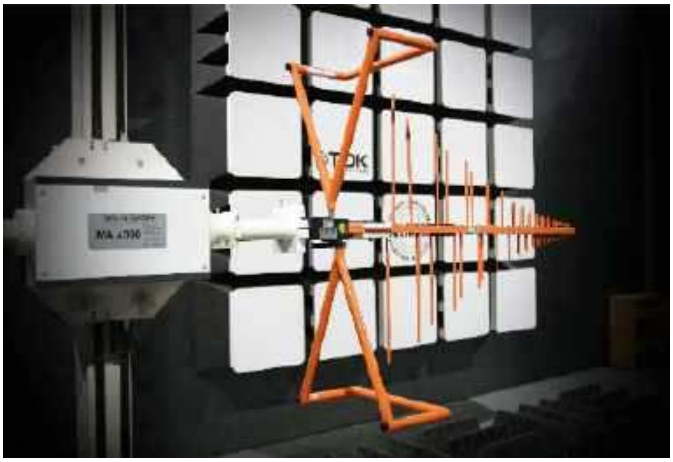
Wykonywane badania

Prowadzone badania dotyczą sprawdzenia zgodności urządzeń / wyrobów z rozporządzeniami i dyrektywami unijnymi, normami międzynarodowymi oraz krajowymi. Ich celem jest spełnienie wymagań w zakresie:

- bezpieczeństwa użytkowania,
- funkcjonowania zasadniczego,
- kompatybilności elektromagnetycznej (badania odporności i emisyjności),
- odporności na narażenia mechaniczne i klimatyczne.

Fotografia 1. Komora SAC do badań w zakresie kompatybilności elektromagnetycznej (widok z zewnątrz)

Więcej informacji:
CELAMED Centralne Laboratorium Aparatury Medycznej Aspel S.A.
32-080 Zabierzów, os. H. Sienkiewicza 33
laboratorium@celamed.pl, tel. +48 694 423 951



Fotografia 2. Wnętrze komory SAC – widoczna antena pomiarowa na kolumnie zmotoryzowanej oraz absorbery elektromagnetyczne (w tle)

Testy wykonywane przez Laboratorium dotyczą wyrobów i wyposażenia elektrycznego, telekomunikacyjnego oraz elektronicznego, a także wyposażenia medycznego, w tym medycznych urządzeń elektrycznych, medycznych systemów elektrycznych czy stacjonarnego sprzętu treningowego.

Zakres badań kompatybilności elektromagnetycznej w zakresie odporności obejmuje:

- wyładowania elektrostatyczne (ESD) według normy EN 61000-4-2,
- serie szybkich elektrycznych stanów przejściowych (BURST) według EN 61000-4-4,
- udary (SURGE) według normy EN 61000-4-5,
- zapady napięcia, krótkie przerwy i zmiany napięcia (PQT) według normy EN 61000-4-11,
- pole magnetyczne o częstotliwości sieciowej (POWERM) według normy EN 61000-4-8,
- odporność na impulsowe pole magnetyczne (PULSEM) według normy EN 61000-4-9,
- odporność na zaburzenia przewodzone indukowane przez pola o częstotliwości radiowej, metoda badań i pomiarów według normy EN 61000-4-6,
- odporność urządzeń na zaburzenia promieniowane o częstotliwości radiowej, metoda badań i pomiarów według normy EN 61000-4-3.

Zakres badań kompatybilności elektromagnetycznej w zakresie emisji częstotliwości radiowych obejmuje:

- pomiary emisji promieniowanej i przewodzonej, metoda badań i pomiarów według EN 55016-2-3, EN 55016-2-3, EN 55011 oraz EN 55032,

eprasa.pl 4202fe84f2

Fotografia 3. Zestaw aparatury do badań emisji i odporności urządzeń

Fotografia 4. Zespół wzmacniaczy mocy do generowania narażeń podczas testów podatności na zakłócenia elektromagnetyczne

Fotografia 5. Częstościomierz wysokiej klasy o paśmie 0,1...3 GHz

- pomiary harmonicznych prądu (fazowy prąd zasilający ≤ 16 A) zgodnie z normą EN 61000-3-2,
- pomiary zmian, wahań napięcia i migotania światła w publicznych sieciach zasilających niskiego napięcia, powodowanych przez odbiorniki o fazowym prądzie znamionowym ≤ 16 A zgodnie z normą EN 61000-3-3.

Powyższe badania odporności i emisji są wykonywane dla wyrobów:

- medycznych,
- powszechnego użytku, narzędzi elektrycznych,
- informatycznych i urządzeń multimedialnych,
- środowiska mieszkalnego, handlowego i lekko przemysłowego,
- środowiska przemysłowego,
- pozostałych, których wielkości probiercze mogą być osiągnięte przez posiadane wyposażenie pomiarowe.

Zakres wykonywanych wzorcowań

Laboratorium wykonuje wzorcowania przyrządów pomiarowych w zakresie wielkości elektrycznych:

- napięcia (AC, DC),
- prądu (AC, DC),
- rezystancji,
- oraz czasu i częstotliwości.

Wzorcowania dotyczą następujących przyrządów pomiarowych:

- kalibratory napięcia, prądu i oporu,
- źródła prądu i napięcia,
- amperomierze, woltomierze, omomierze, multimetry,
- amperomierze cęgowe,
- generatory, częstotliwościomierze,
- oporniki dekadowe, wzorcowe,
- testery wytrzymałości izolacji,
- mierniki impedancji/ciągłości PE,
- defibrylatory,
- testery defibrylatorów,
- generatory przebiegów EKG,
- sieci sprzęgająco-odsprzęgające.

Przy wyznaczaniu niepewności stosowane są metody zgodne z wytycznymi dokumentu EA-4/02 M: 2022 wydanego przez Europejską Współpracę w Dziedzinie Akredytacji oraz inne dokumenty opublikowane w EA-01/01.

Laboratorium, wydając świadectwa z wzorcowań, zapewnia i dokumentuje spójność pomiarową stosowanego wyposażenia.

PCA Polskie Centrum Akredytacji BADANIA
AB 1604

PCA Polskie Centrum Akredytacji WZORCOWANIE
AP 161

eprasa.pl 4202fe84f2

Oszczędzanie energii w teorii i w praktyce (3)

W poprzedniej części cyklu przyjrzeliśmy się możliwościom ograniczenia poboru prądu przez współczesne mikrokontrolery. Choć skupiliśmy się głównie na produktach firmy Microchip, inni producenci oferują równie oszczędne układy.

W wielu aplikacjach peryferia wbudowane w mikrokontroler mogą okazać się niewystarczające – lub nie oferować odpowiednich parametrów. W takich sytuacjach zachodzi potrzeba sięgnięcia po układy zewnętrzne. Za dobry przykład niech posłużą tu czujniki położenia i przyspieszenia używane w smartwatchach czy opaskach sportowych do monitorowania aktywności fizycznej. Kolejną grupę stanowią układy monitorujące tętno i natlenienie krwi użytkownika takiej opaski sportowej czy monitora kondycji. Innym przykładem mogą być czujniki środowiskowe do monitorowania pogody. W aplikacjach zasilanych bateryjnie układy takie nie tylko muszą wykazać się ekstremalną oszczędnością energii, ale także – zależnie od kontekstu – powinny pozwalać mikrokontrolerowi na wyłączenie bądź też wykazywać zdolność, by wspomniany mikrokontroler w razie potrzeby obudzić. Oczywiście, wybór dostępnych komponentów z omawianej kategorii jest równie szeroki, jak oferta mikrokontrolerów, warto więc przyjrzeć się choćby kilku z nich, by poznać ich parametry i możliwości – oraz ograniczenia.

Układy IMU

Na początek weźmy pod lupę prawdopodobnie najciekawsze komponenty stworzone dzięki zaawansowaniu technologii produkcji półprzewodników – sensory mierzące zmiany prędkości i orientację w przestrzeni. Technologia MEMS (ang. microelectromechanical system) pozwoliła na stworzenie akcelerometrów i żyroskopów mieszczących się w strukturze układu scalonego. Dzięki temu duże i kosztowne dotychczas urządzenia elektromechaniczne stały się integralną częścią m.in. smartfonów, smartwatchy i opasek sportowych. Ważną (i pożądaną) cechą układów IMU jest ich energooszczędność – w końcu pracują głównie w urządzeniach zasilanych bateryjnie.

Typowy układ IMU ma akcelerometr oraz żyroskop. Opcjonalnie może mieć wbudowany również magnetometr, choć on akurat w większości układów jest zewnętrznym komponentem dołączonym do IMU z użyciem pomocniczego interfejsu. Magnetometr pozwala na wskazanie bezwzględnej orientacji czujnika względem bieguna magnetycznego Ziemi – oraz zmian tej orientacji. Dodatkowo zapobiega on powolnemu dryftowi czujnika, spowodowanemu efektem Coriolisa. Nie zawsze jest to potrzebne, dlatego na rynku nie brakuje sensorów IMU pozbawionych magnetometru. Omawiane czujniki dostarczają zazwyczaj surowych danych o przyspieszeniu, zmianach prędkości kątowej i – opcjonalnie – intensywności pola magnetycznego. Często wyposażane są też we własny, wewnętrzny procesor, który zajmuje się przetwarzaniem surowych danych i generowaniem różnych przerwań lub rozpoznawaniem gestów czy aktywności, przy założeniu, że czujnik umocowany jest na nadgarstku (w opasce sportowej bądź w smartwatchu). Układ będzie generował wybrane przez programistę przerwania, które mogą wybudzać główny mikrokontroler celem podjęcia jakiejś akcji. Oznacza to, że układ IMU będzie musiał pracować cały czas.

Spójrzmy zatem na pobór prądu typowego przedstawiciela omawianej grupy czujników: BMI270 firmy Bosch. Wyposażony jest on jedynie w akcelerometr i żyroskop, ale udostępnia też dodatkowe funkcje, takie jak wykrywanie gestów nadgarstka, liczenie kroków czy rozpoznawanie podstawowych aktywności: brak ruchu/chodzenie/bieganie/inne. Układ został zoptymalizowany pod kątem pracy bateryjnej, dlatego w nocie podano pobór prądu dla napięcia zasilania 1,8 V. Maksymalnie układ pobiera 970 μA w trybie wysokiej wydajności akcelerometru i żyroskopu. W trybie niskiego poboru prądu z użyciem obu czujników MEMS pobór wynosi 420 μA, jeśli jednak zrezygnujemy z użycia żyroskopu, w trybie oszczędzania energii możemy osiągnąć pobór zaledwie 10 μA. W tym ostatnim trybie dane są zresztą aktualizowane 25 razy na sekundę, co wystarcza do wykrycia ruchu. Zaawansowane funkcje pobierają dodatkowo 3 μA dzięki optymalizacji wewnętrznego procesora pod konkretne zastosowania.

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

Inny przedstawiciel układów IMU to model LSM6DS3TR-C firmy STMicroelectronics. Układ oferuje podobne funkcje, co opisany wyżej, przy czym jest kompatybilny z urządzeniami opartymi na systemie operacyjnym Android, co pozwala na łatwą integrację ze smartfonami i smartwatchami bazującymi na tym systemie. Ciekawą funkcją odróżniającą opisywany układ od BMI270 jest opcja automatycznego wykrywania upadku – jednym z jej zastosowań może być sygnalizacja przewrócenia się osoby starszej. Nas jednak interesuje przede wszystkim zapotrzebowanie na energię. Maksymalny pobór prądu w trybie wysokiej wydajności wynosi 900 μA, w trybie normalnym 450 μA, a w trybie niskiego poboru prądu 290 μA. W ostatnim z wymienionych trybów dane z akcelerometru i żyroskopu odczytywane są z częstotliwością 52 Hz, czyli ponad dwa razy częściej niż w przypadku układu firmy Bosch. Rezygnacja z żyroskopu i ograniczenie częstotliwości próbkowania akcelerometru do 12,5 Hz obniża pobór prądu do zaledwie 9 μA.

Układ ICM-42670-P jest nowym produktem w ofercie TDK InvenSense. Nazwa ta może być znana hobbystom ze względu na niegdyś bardzo popularne moduły z układami MPU6050, wciąż dostępne za około 10 złotych. Układ ICM-42670-P oferuje akcelerometr i żyroskop MEMS o niskim poziomie szumów własnych, programowalne filtry cyfrowe oraz rozwiązanie o nazwie APEX Engine, które realizuje funkcje rozpoznawania gestów, liczenia kroków, itp. Pobór prądu jest dość niski: gdy pracuje akcelerometr i żyroskop, wynosi on 550 μA przy 1,8 V. Sam akcelerometr pobiera zaś 200 μA.

Wspomniany układ MPU6050 nie jest rekomendowany do nowych projektów, co może stanowić pewien problem, gdyż poza nim nie spotkamy na rynku tanich układów IMU zawierających wbudowany magnetometr. Podobnie sytuacja wygląda z układem ICM-20948 firmy TDK InvenSense. Producent chwali się technologią DMP występującą w wielu układach IMU tej firmy, czyli wbudowanym mikroprocesorem, do którego przygotowano firmware pozwalający przetwarzać dane lokalnie, bez udziału głównego mikrokontrolera. W efekcie układ rozpoznaje różne gesty, może automatycznie liczyć kroki i inne aktywności, wykrywać upadek czy też realizować inne funkcje, w tym automatyczną kalibrację w tle. DMP nie może jednak integrować surowych danych z sensorów – to zadanie wciąż należy do głównego mikrokontrolera. Co więcej, by używać DMP, trzeba przy każdym włączeniu zasilania załadować do niego firmware z nadrzędnego procesora. Producent chwali się też najniższym poborem prądu na świecie wśród układów IMU z magnetometrem. Przy użyciu wszystkich trzech sensorów i napięciu 1,8 V układ

eprasa.pl 4202fe84f2

pobiera 3,11 mA. Sam żyroskop, przy próbkowaniu 102,3 Hz, pobiera 1,23 mA. Akcelerometr przy tej samej częstotliwości próbkowania w trybie oszczędności energii pobiera 68,9 µA, sam magnetometr zaś – przy próbkowaniu 8 Hz – pobiera 90 µA.

Kolejnym układem IMU z magnetometrem jest BNO055 firmy Bosch, wyposażony w akcelerometr, żyroskop i magnetometr, a także automatycznie integrujący pozyskiwane z nich dane i dostarczający bezwzględnych wartości położenia i kierunku. Zwalnia to główny mikrokontroler z konieczności przeprowadzania wymienionych obliczeń, ale sam układ IMU nie jest już tak wysoko energooszczędny. Przy napięciu 3 V i częstotliwości aktualizacji wartości 100 Hz układ pobiera do 12,3 mA. W trybie oszczędzania energii pobór wynosi 2,72 mA, dopiero uśpienie sensorów ogranicza go do wciąż wysokiego poziomu 40 µA.

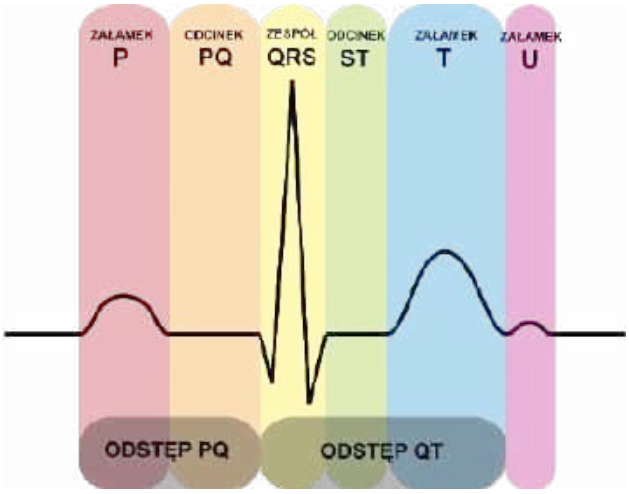
Zaprezentowane modele to zaledwie część dostępnych na rynku układów IMU, ale łatwo zauważyć, że są to układy relatywnie „prądożerne", zwłaszcza jeśli chcemy uzyskać wysoką częstotliwość próbkowania i dużą precyzję. W praktycznych zastosowaniach można jednak ograniczyć się do sporo niższego taktowania (lub nawet używać tylko samego akcelerometru) celem redukcji poboru prądu. Układy te mają też w większości przypadków kolejkę FIFO o pojemności rzędu 1 kB, co pozwala przetwarzać dane pakietami, zamiast na bieżąco. Jeszcze bardziej energooszczędną opcją jest utrzymywanie układu IMU w stanie uśpienia i wybudzanie go 2...5 razy na sekundę celem dokonania pojedynczego pomiaru przyspieszenia, orientacji i zmian prędkości kątowej. W większości zastosowań nawet takie zgrubne pomiary powinny pozwolić zorientować się w położeniu czujnika. Oczywiście będzie to adekwatne, gdy mówimy o urządzeniu noszonym bądź używanym przez człowieka. Takie urządzenia, jak kontrolery ruchowe, zestawy AR/VR, drony czy stabilizatory kamer, wymagają jednak wysokiej częstotliwości próbkowania, a więc i wyższego poboru prądu. W tych aplikacjach jednak układ IMU czy główny mikrokontroler odpowiadają jedynie za niewielką część zapotrzebowania na energię.

Czujniki do pomiaru tętna i natlenienia krwi

W ostatniej dekadzie zarówno wzrost popularności opasek fitnessowych, jak i pandemia COVID-19 przyczyniły się do rozwoju tanich, scalonych czujników biometrycznych. Sensory te przede wszystkim mierzą puls i poziom natlenienia krwi. Pomiar odbywa się metodą optyczną – diody LED przezskórnie oświetlają naczynia krwionośne, a fotodioda (bądź fotodiody) rejestrują niewielkie zmiany jasności światła odbitego i zaabsorbowanego przez krew. Te nieznaczne zmiany intensywności promieniowania pozwalają na pomiar nie tylko pulsacji krwi w naczyniu, ale też stopnia jej natlenienia. Konstrukcyjnie najczęściej spotyka się rozwiązanie, w którym układ scalony sterujący diodami i dokonujący pomiarów jest oddzielony od układu zawierającego diody oraz fotodiody. Istnieją jednak układy, które łączą część sterująco-przetwarzającą z zestawem diod i fotodiod w jednej obudowie.

W celu ograniczenia poboru prądu diody LED nie są aktywne cały czas, lecz sterowane krótkimi impulsami, trwającymi od kilkunastu do kilkuset mikrosekund. Okazuje się, że to wystarcza do dokonania adekwatnego pomiaru. W przypadku urządzeń zasilanych bateryjnie, jak opaski fitnessowe, dodatkowo pomiar tętna i natlenienia nie jest realizowany w czasie rzeczywistym, lecz – zależnie od potrzeby – co minutę lub rzadziej. Ze względu na używanie diod LED – nawet przy pracy impulsowej – czujnik tętna i natlenienia pobiera sporo energii, dlatego redukcja częstotliwości pomiaru do kilku razy dziennie może wydłużyć pracę opaski nawet dziesięciokrotnie.

Porównajmy zatem dwa układy scalone, oba z oferty Analog Devices/Maxim: MAX30101 i MAX30112. MAX30101 integruje w swojej strukturze diody LED i fotodiodę, podczas gdy MAX30112 używa zewnętrznych komponentów, zazwyczaj zintegrowanych w gotowy moduł, jak na przykład NJL5513R firmy Nisshinbo Micro Devices. Układ MAX30101 w trakcie pomiaru tętna i natlenienia z częstotliwością próbkowania wynoszącą 50 sps pobiera typowo 600 µA, maksymalnie

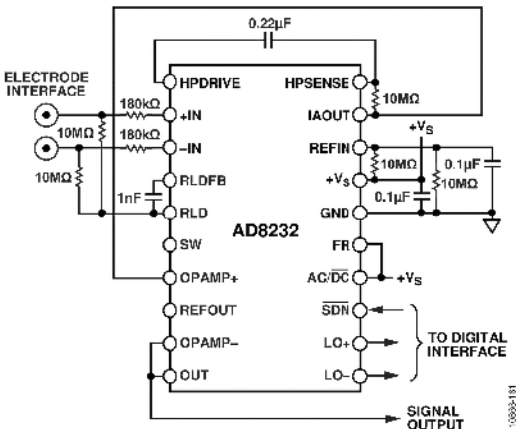


Rysunek 1. Składowe elementy typowego przebiegu EKG

zaś 1,1 mA. MAX30112 przy podobnej konfiguracji pobiera 304 µA na potrzeby diod LED i dodatkowe 40 µA do samego pomiaru. Tak duża różnica w poborze prądu wynika z faktu, że układ MAX30101 używa czterech diod LED, podczas gdy MAX30112 – tylko dwóch. Podane wartości uśredniono, gdyż w rzeczywistości diody LED są zasilane krótkimi impulsami prądowymi o wartości nawet 200 mA. W dobrych warunkach (sensor dobrze dociśnięty do skóry, brak silnych źródeł światła) można jeszcze wyraźniej obniżyć prąd diod, a tym samym średni pobór prądu przez układy.

Istnieje jednak inna metoda pomiaru tętna, stosowana niegdyś m.in. przez firmę Garmin. Opiera się ona na bezpośrednim pomiarze EKG. Użytkownik takiego rozwiązania zakłada specjalny pasek na klatkę piersiową. Przednia część paska ma dwa klipsy do przypięcia zasilanego bateryjnie modułu pomiarowego – klipsy połączone są z dwiema elektrodami, wykonanymi z przewodzącej gumy. Sam moduł realizuje funkcję dwuelektrodowego pomiaru EKG, konwertuje pomiar na tętno, a na końcu przesyła jego wartości do odbiornika za pomocą protokołu ANT+. Choć na pierwszy rzut oka pomiar wydaje się prosty, to w rzeczywistości trzeba wziąć pod uwagę amplitudę mierzonego sygnału bioelektrycznej aktywności serca. Amplituda ta wynosi od 0,5 mV do 5 mV – i nakłada się na nią relatywnie duże napięcie niezrównoważenia ±300 mV, powstające na styku elektrod ze skórą. Standardowe pasmo przenoszenia aparatów EKG stosowanych w służbie zdrowia wynosi 0,05…100 Hz. Na **rysunku 1** pokazano typowy przebieg sygnału EKG. Do pomiaru tętna wystarczy zmierzyć odstęp między kolejnymi załamkami R w zespole QRS.

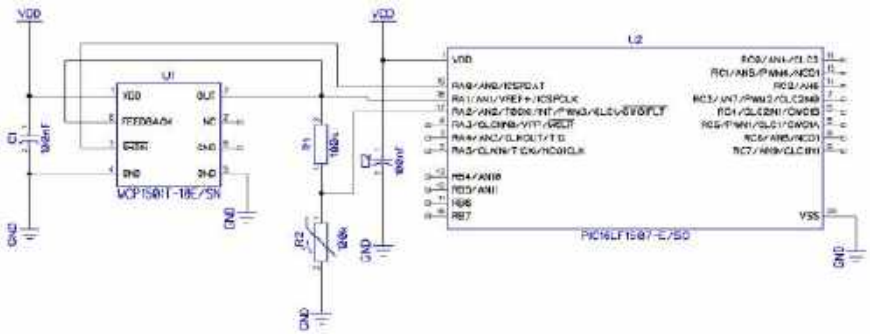
Zaprojektowanie analogowego wzmacniacza do pomiarów EKG nie powinno nastręczać dużych problemów. Można jednak skorzystać z zaprojektowanego specjalnie do tego celu układu, jakim jest AD8232 firmy Analog Devices. Układ ten pozwala na pomiar tętna z użyciem



Rysunek 2. Standardowa aplikacja AD8232 w konfiguracji dwuelektrodowej

eprasa.pl 4202fe84f2

dwóch lub trzech elektrod. Nota katalogowa prezentuje rozwiązanie układowe przeznaczone do użycia z opisanym wyżej paskiem, co uwidocznione zostało na schemacie na **rysunku 2**. Sygnał wyjściowy z układu można mierzyć za pomocą wewnętrznego przetwornika ADC mikrokontrolera lub zastosować komparator do wytwarzania impulsów odpowiadających załamkom R, które mikrokontroler może zliczać celem pomiaru częstotliwości uderzeń serca. Jedną z użytecznych funkcji jest wykrywanie, czy elektrody zostały prawidłowo umocowane na skórze – została ona zrealizowana za pomocą dwóch wewnętrznych komparatorów. Sam układ AD8232 pobiera typowo 170 µA, maksymalnie zaś 230 µA. W stanie uśpienia natomiast maksymalny pobór prądu nie przekracza 500 nA. Na znanym portalu aukcyjnym dostępne są gotowe, chińskie moduły, które do pomiaru EKG używają jednorazowych elektrod samoprzylepnych.



Rysunek 3. Przykładowy układ pomiaru temperatury z użyciem termistora i źródła napięcia odniesienia

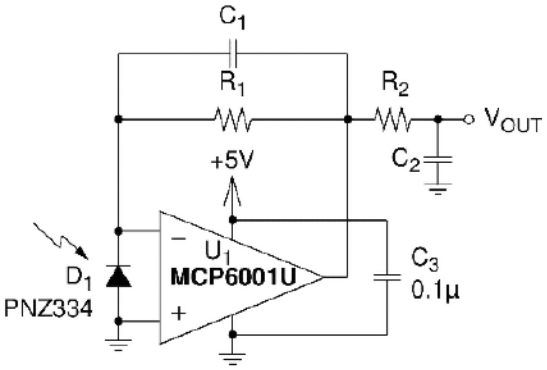
Czujniki środowiskowe, cyfrowe i analogowe

Nie tylko opaski fitnessowe muszą być energooszczędne. Istnieje szeroka gama urządzeń monitorujących środowisko, którym stawia się wymóg energooszczędności. Najprostszy przykład to zwykła stacja pogodowa. W takim wypadku optymalnym rozwiązaniem może okazać się zastosowanie scalonego czujnika temperatury i wilgotności – na przykład układu SHT40 firmy Sensirion. Model ten pozwala mierzyć temperaturę z dokładnością do ±0,2°C, a wilgotność względną z dokładnością do ±1% RH. Poza samymi sensorami, układ ma wszystkie niezbędne komponenty, w tym przetwornik ADC i interfejs I²C. Nota katalogowa podaje od razu uśredniony pobór prądu przy pomiarze raz na sekundę i różnych ustawieniach dokładności pomiaru. Przy największej dokładności układ pobiera średnio 2,2 µA, przy najmniejszej 0,4 µA. Typowy prąd w stanie czuwania wynosi jedynie 80 nA. Za to w czasie pomiaru czujnik pobiera aż 320 µA. Układ ma też niewielki element grzejny, którego zadaniem jest osuszanie czujnika w razie zalania lub w środowisku o wysokiej wilgotności. Element ten włączany jest na maksymalnie sekundę, ale pobiera wtedy relatywnie duży prąd, nawet 60 mA.

Gdy interesuje nas wyłącznie pomiar temperatury, rozwiązaniem może okazać się użycie popularnego wśród hobbystów czujnika DS18B20 firmy Maxim. Oryginalny układ jest dość kosztowny, dlatego hobbyści najczęściej używają chińskich kopii kosztujących grosze. Układ pozwala na pomiar temperatury z rozdzielczością od 0,5°C (9 bitów) do 0,0625°C (12 bitów), przy czym dokładność samego czujnika wynosi ±0,5°C. Pobór prądu w trakcie pomiaru przy napięciu zasilania 5 V wynosi 1 mA, ale przy pomiarze 9-bitowym uśredniony pobór prądu wynosi tylko około 94 µA. Gdy czujnik czuwa, pobiera tylko 750 nA. Ogromną wadą opisywanego układu jest minimalne napięcie zasilania wynoszące 3 V, co ogranicza opcje dostępnych źródeł zasilania urządzenia do akumulatorów litowo-jonowych lub minimum trzech zwykłych baterii, przy czym w tym drugim wypadku nie zostaną one w pełni wykorzystane. Alternatywnie można dodać niewielką przetwornicę typu Boost, ale jest to dodatkowy koszt i dodatkowa komplikacja układu.

Tanią, choć nie do końca dokładną metodą pomiaru temperatury byłoby zastosowanie analogowego rozwiązania, czyli pomiar napięcia przewodzenia diody krzemowej lub pomiar z użyciem termistora. Rozwiązanie z diodą często jest integrowane w mikrokontrolerach, ale ciepło generowane przez sam mikrokontroler może nieznacznie zawyżyć wynik. Z kolei użycie termistora okazuje się nieco problematyczne, gdyż trzeba uwzględnić jego nieliniową charakterystykę zależności oporu od temperatury, jednocześnie ograniczając też zjawisko samonagrzewania się termistora. Osobną kwestię stanowi bardzo szeroki zakres zmian rezystancji termistora w funkcji temperatury – na przykład termistor NTC 100 kΩ w temperaturze –55°C ma opór ponad 11 MΩ, a w temperaturze 125°C już tylko 3 kΩ. Tak duży przedział wartości może być problematyczny, lecz istnieje praktyczne rozwiązanie pozwalające na dość dokładny pomiar, zwłaszcza jeśli do interpretacji użyje się tabeli wyników skopiowanej z noty katalogowej termistora. **Rysunek 3** pokazuje przykładowe rozwiązanie układowe korzystające ze źródła napięcia odniesienia MCP1501T-10E. Źródło to dostarcza napięcie 1,024 V, które jest używane zarówno przez moduł ADC mikrokontrolera PIC16LF1507, jak i przez dzielnik składający się z rezystora R1 o wartości 100 kΩ oraz termistora NTC, również 100 kΩ. Układ może zawierać tyle dzielników, ile dostępnych wejść analogowych ma mikrokontroler. Przy temperaturze –55°C na termistorze będzie panować napięcie 1,015 V, a przy 125°C – tylko 0,030 V. Maksymalny pobór prądu wyniesie niecałe 10 µA na dzielnik, średni zaś połowę tej wartości. Na schemacie uwzględniono opcję wyłączenia źródła napięcia odniesienia, co ogranicza pobór prądu ze 140 µA do 205 nA. Przy wykonywaniu pomiaru z użyciem ADC należy pamiętać o tym, że – ze względu na ładowanie wewnętrznego kondensatora S/H o typowej pojemności 15...22 pF – przetwornik oczekuje źródła o impedancji 10 kΩ, a w tym rozwiązaniu impedancja jest 10 razy większa. By pomiar był dokładny, należy zredukować zegar przetwornika ADC tak, by cykl ładowania kondensatora trwał przynajmniej 2,5 µs.

Układ z rysunku 3 można też z powodzeniem zastosować do pomiarów jasności otoczenia, używając fotodiody PIN zamiast termistora. Wartość rezystora dobiera się wówczas dla maksymalnego prądu wstecznego oświetlonej diody – tak, by napięcie na niej wynosiło kilka miliwoltów. Następnie oblicza się wartość napięcia na diodzie dla prądu ciemnego. Takie rozwiązanie okaże się wystarczające do pomiaru jasności otoczenia fotodiody, gdyż – zależnie od wartości rezystora – będzie ona potrzebować setek mikrosekund na reakcję po pojawieniu się światła. Podobnie potrzebny będzie pewien czas od pojawienia się napięcia zasilania obwodu do ustalenia się napięcia na fotodiodzie. Jeśli zatem chcemy mierzyć szybkie



Rysunek 4. Fotodioda PIN współpracująca ze wzmacniaczem transimpedancyjnym w celu uzyskania lepszego czasu reakcji i filtracji szumów

eprasa.pl 4202fe84f2

zmiany światła, fotodioda PIN musi współpracować ze wzmacniaczem transimpedancyjnym. **Rysunek 4** pokazuje przykładowe rozwiązanie z noty aplikacyjnej AN951 Microchip. Rozwiązanie takie pozwala fotodiodzie PIN znacznie szybciej reagować na zmiany jasności. Czytelnik mógłby zapytać, dlaczego by nie użyć zwykłego fotorezystora? Można – i byłoby to rozwiązanie bardzo dobre. Ograniczeniem technicznym zastosowania fotorezystorów jest czas reakcji, typowo 20...50 milisekund, przy czym fotorezystory zwykle wolniej reagują na spadek jasności. Drugi problem stanowi mocno nieliniowa charakterystyka pracy i duży rozrzut parametrów (dla przykładu: fotorezystor PGM5526 firmy Token ma rezystancję w mroku 1 MΩ, ale przy 10 lx spada ona do 8...20 kΩ, czyli 50...125 razy). Problemem długofalowym w użytkowaniu fotorezystorów okazuje się dodatkowo dyrektywa RoHS – komponenty te używają związków kadmu, selenu i ołowiu, przez co podlegają restrykcjom w stosowaniu na terenie UE. Z tego względu zasadniczo odradza się uwzględnianie ich w nowych projektach.

Czujniki ciśnienia MEMS są w teorii prostymi komponentami. Używając materiału piezorezystywnego (którego rezystancja zależy od naprężeń nań działających), naniesionego na membranę, tworzy się mostek Wheatstone'a. Zależnie od zastosowania obie strony membrany mogą być wyprowadzone na zewnątrz za pomocą przyłączy do wężyków lub otworów w obudowie – może też być tylko jeden otwór, pozwalający na oddziaływanie ciśnienia na membranę, podczas gdy po jej drugiej stronie panuje próżnia.

Innym powszechnie stosowanym rozwiązaniem technologicznym jest pomiar pojemności kondensatora, którego jedną z elektrod stanowi membrana uginającą się pod ciśnieniem. Dostępne są zarówno czujniki analogowe, wymagające zewnętrznych układów zasilania mostka i wzmacniania sygnału różnicowego, jak i warianty wyposażone w swojej strukturze w analogowy front-end – a często też przetwornik ADC i układ sterujący, komunikujący się szeregowo z mikrokontrolerem. Absolutne czujniki ciśnienia nie tylko mogą mierzyć powolne jego zmiany zależne od warunków pogodowych, ale także mają wystarczającą czułość i prędkość działania, by reagować na zmianę wysokości czujnika, mierzyć prędkość i kierunek ruchu w pionie, a nawet określać piętro budynku na podstawie nieznacznych różnic ciśnienia atmosferycznego. Przykładem takiego czujnika może być BMP581 firmy Bosch: mierzy on bezwzględne ciśnienie atmosferyczne w zakresie od 300 hPa do 1250 hPa. Dokładność bezwzględna wynosi ±0,5 hPa, ale przy pomiarach względnych zmian ciśnienia osiąga ona wartość ±0,08 hPa. Wbudowany czujnik temperatury ma dokładność ±0,5°C, a rozdzielczość 1°C. Jest on używany do linearyzacji i kompensacji temperaturowej czujnika ciśnienia. Nota podaje maksymalny pobór prądu czujnika 260 μA, ale uśredniony pobór prądu w trybie low power – przy częstotliwości pomiaru 1 Hz – wynosi tylko 1,3 μA, w trybie wysokiej rozdzielczości – przy 30 Hz – rośnie on do 75 μA. Pobór między pomiarami, przy napięciu zasilania 1,8 V, wynosi 1 μA, a w trybie głębokiego uśpienia spada do 550 nA.

Dla porównania spójrzmy na układ ICP-10101 firmy TDK InvenSense. Pozwala on na pomiar ciśnienia w zakresie od 300 hPa do 1100 hPa oraz względnych jego zmian z dokładnością do ±1 Pa, co wg producenta pozwala zmierzyć zmianę wysokości czujnika o 8,5 cm, czyli mniej niż wynosi wysokość pojedynczego stopnia schodów. Dla pomiarów bezwzględnych ciśnienia atmosferycznego w typowych warunkach dokładność wynosi ±1 hPa. Zintegrowany czujnik temperatury ma dokładność ±0,4°C. Pobór prądu w czasie pomiaru wynosi 210 μA przy napięciu 1,8 V, uśredniony pobór w trybie low power wynosi 1,3 μA, a w trybie low noise 5,2 μA – w obu przypadkach częstotliwość pomiaru wynosi 1 Hz. W stanie czuwania układ pobiera 1 μA. Układu nie można uśpić.

Firma NXP Semiconductors oferuje zgoła odmienny czujnik pomiaru ciśnienia, nastawiony raczej na monitorowanie warunków pogodowych niż na obserwowanie zmiany wysokości nad poziomem morza. MPL3150A2S, bo o nim mowa, jest czujnikiem piezorezystywnym, w pełni skalibrowanym w zakresie 700...1500 hPa. Mimo to bezwzględna dokładność wynosi ledwo ±4 hPa, co oznacza konieczność kalibracji każdego czujnika (za to dokładność względna wynosi ±0,5 hPa). Wbudowany czujnik temperatury ma dokładność w pełnym zakresie pomiaru równą ±3°C. Pobór prądu też nie prezentuje się najlepiej: przy napięciu 2,5 V i częstotliwości próbkowania 1 Hz w trybie szybkim układ pobiera średnio 8,5 μA, w standardowym z oversamplingiem 16×40 μA, a w trybie wysokiej rozdzielczości – aż 285 μA. W czasie pojedynczego pomiaru układ pobierze 2 mA, zaś w trybie czuwania układ pobiera 2 μA, czyli też więcej niż konkurencja. Te liczne wady kompensuje jednak dość rozbudowany układ kontrolny. MPL3150A2S potrafi pracować autonomicznie, dokonując pomiarów w interwałach od jednej sekundy do dziewięciu godzin. Wewnętrzna kolejka FIFO pozwala zachować 32 pomiary, co przekłada się nawet na 12 dni autonomicznej pracy, zaś rozbudowany system przerwań z dwoma niezależnymi wyjściami pozwala sygnalizować przekroczenie któregoś z progów ciśnienia lub temperatury, osiągnięcie wartości w zadanym zakresie, konkretnej wartości czy w końcu zmiany o zadany krok. Możliwość długotrwałej pracy bez nadzoru mikrokontrolera, który sam może pobierać więcej prądu w trakcie wykonywania programu, czyni ten układ atrakcyjnym dla projektantów zdalnych monitorów pogody.

Ostatnim omawianym czujnikiem ciśnienia jest układ ENS220 firmy ScioSense. Producent chwali się bardzo niskim poborem prądu i wysoką dokładnością bezwzględną pomiaru ciśnienia i temperatury. Układ dokonuje pomiaru metodą pojemnościową i również może pracować autonomicznie. Zakres pomiaru ciśnienia sięga od 300 hPa do 1200 hPa, dokładność bezwzględna zaś wynosi ±0,5 hPa w zakresie 500...1100 hPa i w temperaturze od –5°C do 55°C. W pełnym zakresie pracy dokładność bezwzględna wynosi ±3 hPa. Względna dokładność pomiaru to ±2,5 Pa, rozdzielczość zaś 1/64 Pa. Dokładność pomiaru temperatury w pełnym zakresie oscyluje w graniach ±0,5°C, ale w zakresie od –5°C do 55°C wynosi typowo ±0,1°C, maksymalnie ±0,2°C. Przy napięciu zasilania 1,8 V układ pobiera 105 nA w trybie bezczynności, w czasie konwersji zaś pobór ten rośnie do 75 μA (przy oversamplingu 128×). Jeśli zrezygnować z oversamplingu, średni pobór prądu spadnie do ok. 0,6 μA, kosztem wyższego poziomu szumów czujnika, co ogranicza dokładność. Nota podaje, że przy jednej konwersji na minutę układ pobiera średnio tylko 0,8 μA. Podobnie jak poprzednik, ENS220 ma bufor FIFO o długości 32 pomiarów, możliwość pracy autonomicznej z maksymalnym interwałem wynoszącym 10 minut – oraz rozbudowany system przerwań, choć układ udostępnia tylko jeden pin przerwania, używany też jako wyjście SDO w trybie pracy SPI. Produkt ScioSense wydaje się zatem bardzo dobrym wyborem, gdy istotniejsze jest ograniczenie poboru prądu niż częsta akwizycja pomiarów. Ponownie możliwość dokonywania pomiarów nawet przez ponad pięć godzin, bez przekazywania danych do mikrokontrolera oraz programowalne przerwania czynią ten układ jeszcze atrakcyjniejszym z punktu widzenia monitorowania pogody, zwłaszcza przy uwzględnieniu bardzo niskiego poboru prądu w stanie czuwania.

Zakończenie

Komponenty, którym przyjrzeliśmy się w artykule, to tylko znikoma część spośród dostępnych na rynku czujników i układów pomiarowych możliwych do zastosowania w urządzeniu zasilanym bateryjnie. Oczywiście nie wyczerpaliśmy jeszcze tematu układów peryferyjnych. W następnej części cyklu spojrzymy na układy watchdog, nadzorujące zasilanie, RTCC, czy układy i rozwiązania analogowe zastępujące wewnętrzne peryferia mikrokontrolerów. Rozważymy też sensowność ich stosowania.

Paweł Kowalczyk, EP

eprasa.pl 4202fe84f2

Amatorskie pomiary wzmacniaczy audio (2)

W poprzedniej części naszego cyklu zapoznaliśmy się z potrzebnym wyposażeniem oraz omówiliśmy podstawowe zagadnienia związane z pomiarami zniekształceń harmonicznych i szumów. Tym razem przejdziemy do opisu oprogramowania analizatora, dokonamy najważniejszych ustawień i kalibracji oraz przygotujemy się do wykonania pierwszych praktycznych pomiarów zniekształceń wzmacniacza audio.

Pierwsza część artykułu znajduje się pod adresem: **https://ulubionykiosk.pl/media**

Oprogramowanie analizatora

W sieci można znaleźć kilka bezpłatnych programów, które potrafią mierzyć między innymi zniekształcenia nieliniowe THD. Wszystkie używają kart dźwiękowych jako interfejsu sprzętowego, pozwalającego na realizowanie funkcji generatora sygnału i analizatora – a jednym z najbardziej znanych jest bezpłatna wersja RigtMark Audio Analyzer RMAA. To program o dość szerokim zakresie funkcjonalności, który oprócz pomiarów THD potrafi zmierzyć pasmo przenoszenia, zniekształcenia intermodulacyjne, przesłuchy międzykanałowe itp. Sam używałem tego programu wielokrotnie. Niestety ma on też wady: wersja bezpłatna jest ograniczona funkcjonalnie, a co ważniejsze – ma limitowaną dokładność analizy FFT. Poza tym program nie jest od wielu lat rozwijany: ostatnia wersja powstała ok. 6 lat temu, co nie oznacza, że nie jest w pełni przydatna. My jednak będziemy używać innego oprogramowania: świetnego i nieustannie rozwijanego programu REW autorstwa Johna Mulcahy.

REW, czyli Room EQ Wizard, jest darmowym programem do… pomiarów akustyki pomieszczeń, a także parametrów głośników i urządzeń audio. Funkcje pomiaru oraz analizy dźwięku REW pomagają zoptymalizować akustykę pomieszczenia odsłuchowego, studia lub kina domowego i pozwalają na znalezienie najlepszego miejsca dla głośników oraz subwooferów, a także optymalnej lokalizacji pozycji odsłuchowej. Ten opis mógłby sugerować, że opisywane oprogramowanie nie przyda się do realizacji naszych celów – to jednak tylko pozory. REW ma wbudowany bardzo dobrej jakości programowy analizator, potrafiący dokładnie zmierzyć między innymi parametr THD+N i IMD z analizą ilościową poszczególnych harmonicznych. Nie zmierzymy nim przesłuchów międzykanałowych, ale statyczne zniekształcenia nieliniowe THD, intermodulacyjne IMD – już tak i (według mnie) lepiej niż przy użyciu RMAA.

Z tego powodu pomiary wzmacniaczy za pomocą REW stały się popularne wśród pasjonatów projektujących i budujących wzmacniacze, przedwzmacniacze, ale także na przykład przetworniki cyfrowo-analogowe (DAC). Program można pobrać ze strony domowej autora: https://www.roomeqwizard.com/. Do wyboru mamy wersje przeznaczone do systemów Windows 64-bit, Windows 32-bit, macOS i Linux. Ja używałem wersji 64-bitowej na system Windows 10.

Po standardowym zainstalowaniu i uruchomieniu programu pojawia się strona główna – **rysunek 9**.

Jak już wiemy, będziemy korzystać tylko z niewielkiego fragmentu potężnych możliwości REW. Dlatego skupimy się wyłącznie na wybranych narzędziach używanych do pomiarów wzmacniaczy. Na **rysunku 10** pokazano pasek narzędzi, spośród których będziemy używali:

- Generatora – spełnia funkcje zaawansowanego cyfrowego generatora przebiegów sinusoidalnych,
- RTA – analizatora widma do pomiaru zniekształceń harmonicznych, intermodulacyjnych oraz szumów.

Przed użyciem program wymaga skonfigurowania wejść i wyjść karty dźwiękowej – oraz wykonania czynności kalibracyjnej. Każda zmiana parametrów pracy dołączonej karty dźwiękowej wymaga ponownej kalibracji: trzeba wykonać tę czynność poprawnie, aby uzyskać wiarygodne i dokładne pomiary.

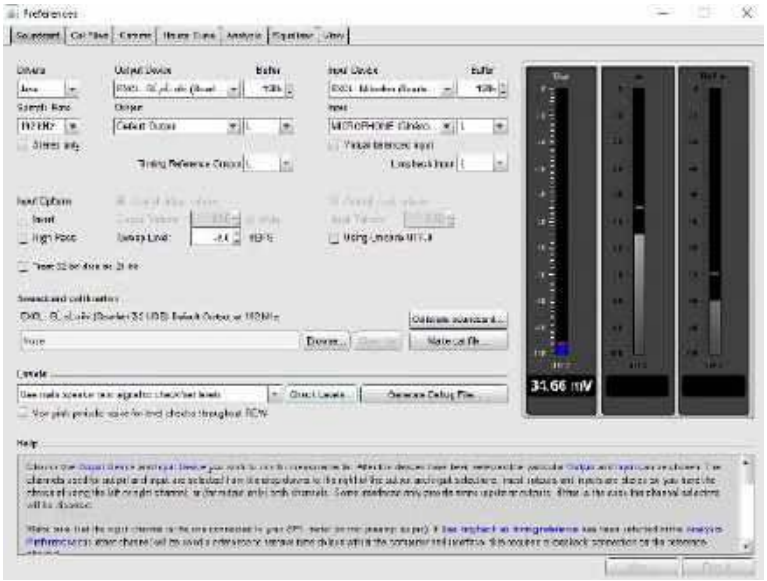
Na początku musimy podłączyć kartę dźwiękową przez port USB do komputera. W prawym górnym rogu ekranu startowego umieszczony jest przycisk Preferences. Po jego przyciśnięciu otworzy się okno o tej samej nazwie – **rysunek 11**.

Nas będzie interesowała pierwsza zakładka: Soundcard. Zaczynamy od skonfigurowania driverów i wejść-wyjść karty. W systemie Windows REW może używać sterowników audio Java lub

Rysunek 9. Okno startowe REW

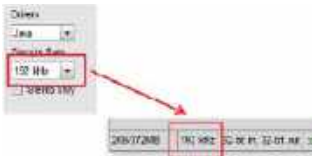
Rysunek 10. Pasek narzędziowy REW z przyciskiem Preferences

eprasa.pl 4202fe84f2



Rysunek 11. Okno Preferences, zakładka Soundcard

sterowników ASIO (jeżeli są zainstalowane) – wyboru dokonamy w oknie Drivers. Oprócz typu sterownika określa się tutaj również częstotliwość próbkowania. Wybrana częstotliwość i maksymalna długość słowa wyświetlane są w lewym dolnym rogu okna startowego – **rysunek 12**.



Rysunek 12. Okno Drivers

Na innych platformach obsługiwane są tylko sterowniki Java. W systemie Windows drivery te oferują dwa rodzaje dostępu do interfejsu: WASAPI lub wbudowany JavaSound (który obsługuje częstotliwości próbkowania do 192 kHz i tylko dane 16-bitowe). Dostęp WASAPI – niezależnie od częstotliwości próbkowania – wspiera każdą rozdzielczość bitową, jaką zapewnia karta lub inne urządzenie zewnętrzne. Urządzenia korzystające z dostępu WASAPI są poprzedzone prefiksem EXCL. Sterowniki ASIO obsługują jedno urządzenie ASIO, które musi być używane zarówno jako wejście, jak i wyjście. REW nie ma wtedy kontroli nad regulacją poziomów.

W oknie Sample ustawiamy częstotliwość próbkowania udostępnianą przez naszą kartę dźwiękową. Scarlett 2i2 gen 3 może pracować z maksymalną częstotliwością 192 kHz.

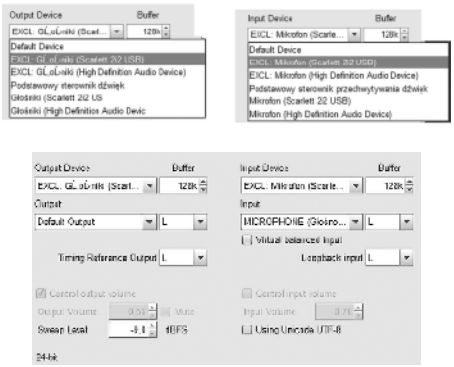
W kolejnym kroku trzeba wybrać urządzenia wyjściowe i wejściowe, które zostaną użyte do pomiarów. Wyboru dokonujemy z list rozwijanych w oknach Output Device oraz Input Device – **rysunek 13a**.

Zgodnie z tym, co powiedzieliśmy wcześniej, możemy wybrać dostęp WASAPI i urządzenia wyjściowe EXCL: Głośniki (Scarlett 2i2 USB) oraz EXCL Mikrofon (Scarlett 2i2 USB) – **rysunek 13b**. Jeśli korzystamy ze sterowników Java, kanał (lewy-prawy) można ustawić po wybraniu wejścia lub wyjścia. Interfejsy zazwyczaj zapewniają stereofoniczne porty I/O, więc zarówno wejście pomiarowe, jak i wyjście może być lewe lub prawe. Ja w trakcie pomiarów używałem kanału lewego wejściowego i wyjściowego.

Jeżeli wejścia i wyjścia pracują prawidłowo, można ustawić długość bufora na 128 kSa. Zwiększanie długości bufora podnosi wartość opóźnienia pomiaru, ale nie wpływa na jego jakość. Jest natomiast wskazane, kiedy wstępują zakłócenia lub przerwy w badanych sygnałach.

a)

b)



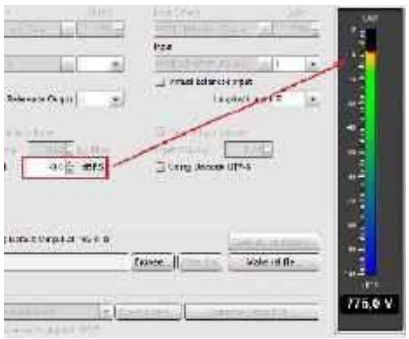
Rysunek 13. Wybór urządzenia wyjściowego i urządzenia wejściowego

Kalibracja karty

Kalibracja karty wiąże się z pomiarami wykonywanymi przez układ elektroniczny karty i oprogramowanie REW. Aby móc w ogóle myśleć o dobrych jakościowo pomiarach, musimy zapewnić jak najmniej zakłócone środowisko testowe. Do pomiarów w zasadzie nie nadają się komputery stacjonarne, a to ze względu na silne zakłócenia od zasilacza (w tym zakłócenia z sieci energetycznej). Najlepszym rozwiązaniem okazuje się zastosowanie laptopa, zasilanego w trakcie pomiarów tylko z akumulatora. Podczas badania dobrze jest również wyłączyć sieci Wi-Fi i Bluetooth oraz pozamykać wszystkie inne programy, łącznie z przeglądarką. Najlepiej, aby w pobliżu nie pracowały urządzenia wytwarzające silne pola elektromagnetyczne czy magnetyczne. Dodatkowo wszystkie połączenia elektryczne przesyłające sygnały pomiarowe powinny być dobrej jakości i niezawodne.

Po ustawieniu drivera oraz urządzeń wejścia i wyjścia przechodzimy do właściwej kalibracji karty dźwiękowej. Podczas tego procesu wykonuje się pomiar parametrów karty: pasma przenoszenia i charakterystyki fazowej. Na podstawie danych kalibracyjnych REW kompensuje niedoskonałości karty, dzięki czemu późniejszy pomiar nie jest obciążony wprowadzanymi przez nią błędami przetwarzania. Przed rozpoczęciem kalibracji upewniamy się, czy ustawienia próbkowania oraz wejścia i wyjścia są prawidłowe. W moim przypadku ustawienia wyglądają tak, jak na rysunku 13. Następnie trzeba połączyć kablami wyjście opisane jako LEFT z wejściem 1 (w przypadku karty Scarlett 2i2), tworząc zapętlenie (loopback) wejścia z wyjściem. Pozwala to na zmierzenie charakterystyki częstotliwościowej i zapisanie jej jako plik kalibracyjny. Kiedy wszystko jest gotowe, możemy przystąpić do kalibracji, naciskając przycisk Calibrate Soundcard w oknie preferencji karty (rysunek 11). Kalibracja przebiega w kilku krokach.

Pierwszy etap to użycie sygnału o częstotliwości 1 kHz do ustawienia poziomu sygnału wyjściowego generatora (sygnału pomiarowego) i poziomu sygnału wejściowego analizatora. W ramach pierwszej z tych operacji ustawiamy parametr Sweep Level w zakresie od –12 do –6 dBFS; –9 dBFS oznacza poziom wyjściowy o 9 dBFS niższy od maksymalnego 0 dBFS – **rysunek 14**. Teraz odtwarzany jest sygnał 1 kHz z wyjścia karty i jest on jednocześnie podawany na jej wejście poprzez zewnętrzne (elektryczne) zapętlenie sygnałów. Za pomocą elementów regulacyjnych karty (pokrętła GAIN wejścia 1) ustawiamy poziom na wejściu. Różnica pomiędzy poziomem sygnału wyjściowego (w naszym przypadku –9 dBFS) nie



Rysunek 14. Ustawianie sygnału OUT

eprasa.pl 4202fe84f2

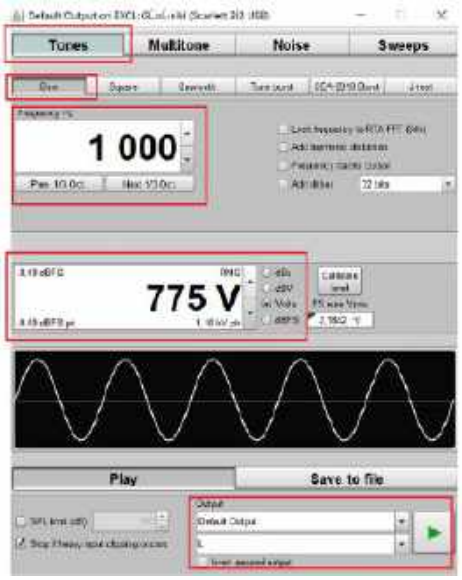
powinna być większa niż 6 dBFS. Poziom wejściowy powinien być niższy od poziomu szczytowego (czerwona linia) o przynajmniej 1 dBFS, co pokazane zostało na **rysunku 15**. W karcie Scarlett 2i2 można taką wartość ustawić bez problemu, natomiast w innych urządzeniach być może trzeba będzie skorzystać z odmiennych regulatorów (na przykład poziomu wyjściowego lub miksera systemowego).

Po ustawieniu sygnału wejściowego w kolejnym kroku przechodzimy do wykonywania automatycznego pomiaru – **rysunek 16**.

Po wykonaniu pomiaru kalibracji generowana jest charakterystyka amplitudowo-fazowa karty i plik konfiguracyjny. Na **rysunku 17** pokazano zmierzoną charakterystykę mojej karty. Jak widać, w paśmie 20 Hz...20 kHz nierównomierność pasma wynosi ±0,1 dB, a pasmo przenoszenia określone dla spadku –3 dB wynosi od 2,2 Hz do 56,9 kHz. Pamiętajmy, że takie pasmo przenoszenia jest możliwe przy próbkowaniu 192 kHz, przy którym częstotliwość Nyquista ma wartość 96 kHz.

Na tym etapie mamy już skalibrowaną kartę i możemy zmierzyć jej parametry – czyli przede wszystkim współczynnik zniekształceń harmonicznych i szumy THD+N – oraz zniekształcenia intermodulacyjne IMD.

Rysunek 15. Ustawienie poziomu wejściowego w stosunku do poziomu wyjściowego



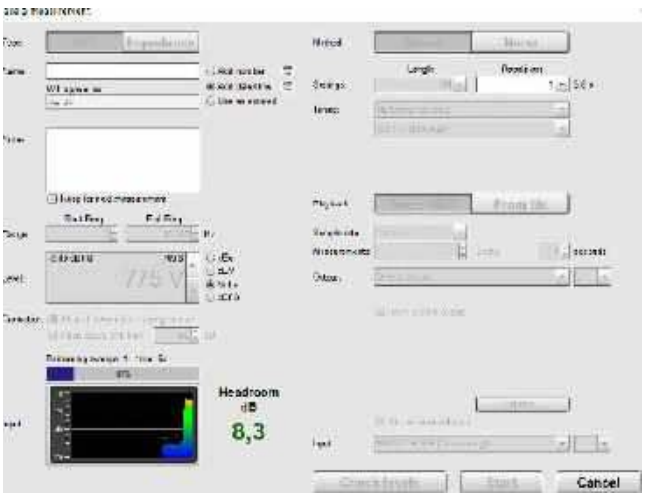
Rysunek 18. Okno generatora

Nim jednak wykonamy pomiar parametrów karty, powiemy parę słów o generatorze uruchamianym z paska narzędziowego. To dość zaawansowane narzędzie, zdolne do generowania przebiegów sinusoidalnych, prostokątnych i piłokształtnych. Ponadto potrafi ono generować różne rodzaje szumów, sygnały wielotonowe i specjalne sygnały testowe, na przykład do badania poziomu jittera. Okno generatora pokazano na **rysunku 18**.

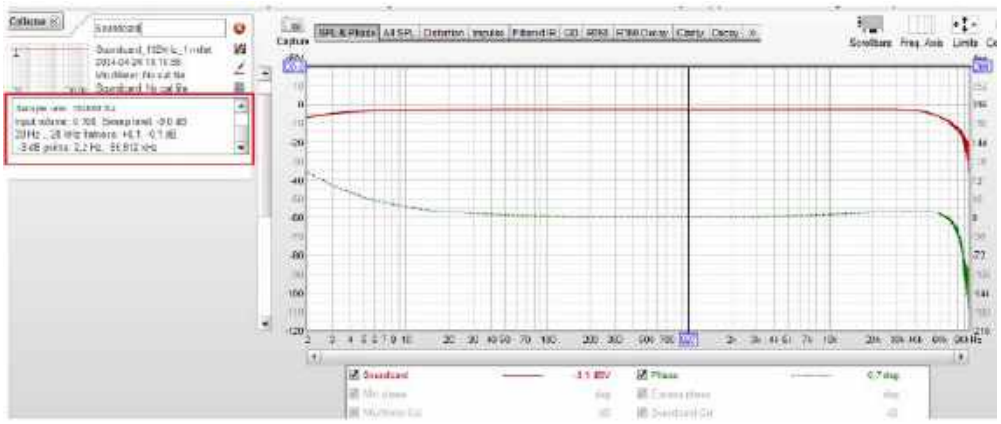
Do badania THD będziemy potrzebowali na początek czystego przebiegu sinusoidalnego o standardowej częstotliwości 1 kHz. Wybieramy zakładkę Tones (pojedynczy przebieg) i opcję Sine (kształt sinusoidalny). W oknie Frequency wpisujemy częstotliwość w Hz. W kolejnym oknie ustalana jest amplituda sygnału generatora (**rysunek 19**). Program generatora „nie wie", jaką amplitudę ma sygnał wyjściowy: zależy to od typu karty i położenia jej regulatorów. W przypadku karty Scarlett 2i2 poziom sygnału wyjściowego zależy od położenia pokrętła MONITOR, dlatego potrzebna jest kalibracja poziomu wyjściowego. Polega ona na zmierzeniu wartości RMS sygnału wyjściowego z karty i wpisaniu go w oknie Calibrate Level. Wartość tego napięcia można zmierzyć oscyloskopem cyfrowym lub miernikiem z funkcją true RMS. W przypadku oscyloskopu cyfrowego zadanie to jest proste, bo urządzenia z tej grupy mają wbudowaną funkcję pomiaru napięcia RMS (co pokazane zostało na **rysunku 20**).

Podczas pomiaru multimetrem należy zachować ostrożność, ponieważ przy wyższych częstotliwościach może on okazać się niedokładny. Ja równolegle mierzyłem miernikiem Fluke 177 i zauważyłem, że przy częstotliwości 1 kHz dokładność katalogowa pomiaru wynosi 2%, ale przy częstotliwościach poniżej 400 Hz jest to już 1%. Część mierników ma ograniczoną częstotliwość pomiaru napięć RMS do zaledwie 100 Hz. Można temu zaradzić, ustawiając do pomiaru napięcia generatora na przykład 50 Hz, a po pomiarze wrócić do 1 kHz. Amplituda sygnału nie zmienia się w funkcji częstotliwości.

Na dole okna generatora w oknie Output ustawiamy kanał, na który wyprowadzony zostanie sygnał z generatora. Po przyciśnięciu zielonego przycisku Play, na wyjściu kanału lewego karty powinien się pojawić sygnał o częstotliwości 1 kHz i amplitudzie 775 mV.

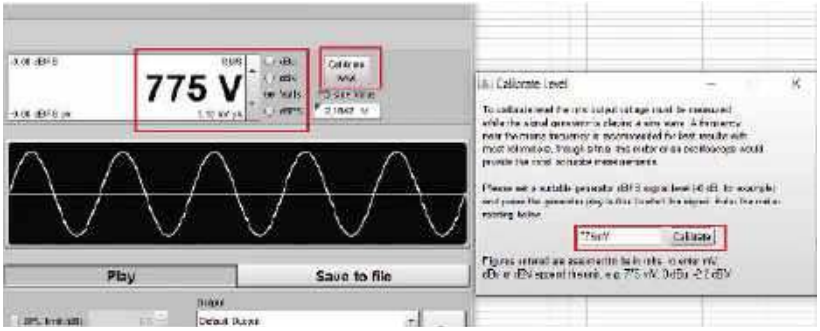


Rysunek 16. Wykonywanie pomiaru charakterystyki amplitudowo-fazowej w trakcie kalibracji

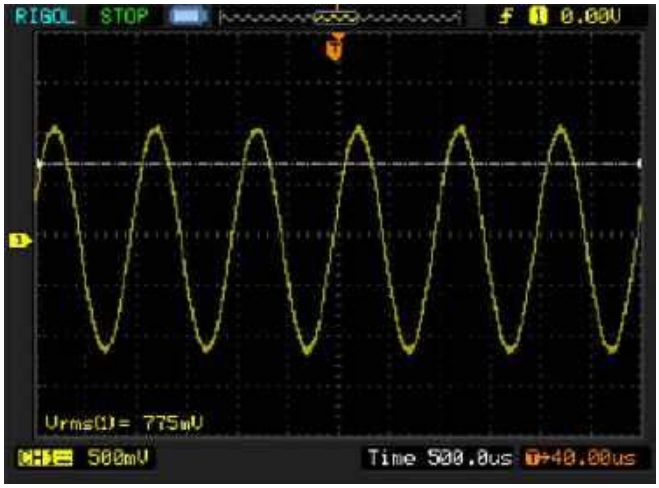


Rysunek 17. Charakterystyka karty

eprasa.pl 4202fe84f2



Rysunek 19. Kalibracja poziomu sygnału generatora



Rysunek 20. Pomiar napięcia RMS za pomocą oscyloskopu

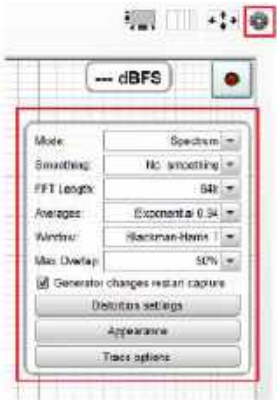
Pomiar zniekształceń karty

Po kalibracji karty i skonfigurowaniu generatora możemy przystąpić do pomiaru zniekształceń samej karty w konfiguracji z zapętlonym wejściem z wyjściem. Jest to dość istotne badanie, ponieważ korzystając z niego, można oszacować, jaki wpływ na pomiar wzmacniacza (czy innego elementu toru) będą miały zniekształcenia i szumy własne karty. Pomiar jest wykonywany w oknie RTA, a przed jego uruchomieniem trzeba ustawić kilka parametrów. Klikamy w prawym górnym rogu na ikonkę kółka zębatego i wprowadzamy najpierw parametry FFT:

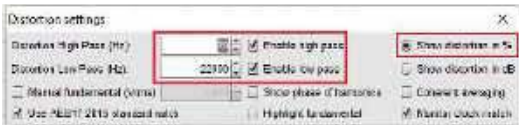
- Tryb (Mode): Spectrum,
- Wygładzanie (Smoothing): wyłączone,
- Długość bufora próbek FFT: 64 kSa,
- Uśrednianie (Averages): Expotential 0,94,
- Okno: Blackman Harris 7,
- Max Overlap: 50%

– tak jak pokazane to zostało na **rysunku 21**.

Po ustawieniu parametrów FFT przechodzimy do zakładki Distortion Settings. Jak wiemy, próbkowanie sygnału wejściowego odbywa się z szybkością 192 kHz. Z twierdzenia o próbkowaniu wynika, że można w ten sposób digitalizować sygnały o maksymalnej częstotliwości 96 kHz. Jeżeli przyjmiemy, że pomiar THD+N będzie obejmował przebiegi do 96 kHz, to REW zmierzy szumy i THD w takim paśmie. Wiemy już, że THD+N trzeba mierzyć w odniesieniu do pasma użytecznego. Jeżeli nawet w paśmie od 20 kHz do 96 kHz nie będzie wielu harmonicznych



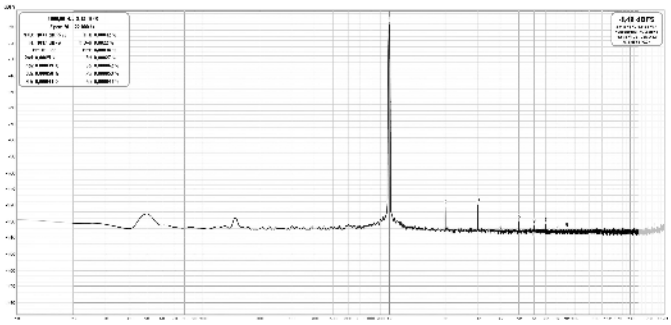
Rysunek 21. Ustawienia pomiaru FFT



Rysunek 22. Ustawienia Distortion settings

zawyżających pomiar THD, to na pewno pojawią się tam szumy także wpływające na wynik THD+N. Nas interesują parametry w paśmie akustycznym 20 Hz...20 kHz i dlatego ustawimy pomiar w zakładce Distortion Settings dla tego pasma. Dokonujemy tego przez zaznaczenie użycia filtrów dolno- i górnoprzepustowego – **rysunek 22**. Częstotliwości filtrów można zmieniać – ja ustaliłem pasmo pomiaru THD od 20 Hz do 22 kHz; dodatkowo wybrałem sposób wyświetlania wyników pomiarów w procentach.

Teraz – przy założeniu, że wcześniej wykonaliśmy kalibrację – możemy przeprowadzić pomiar THD+N samej karty. Upewniamy się, czy generator jest włączony (zielony przycisk Play) i naciskamy czerwony przycisk w prawym górnym rogu okna RTA. Pomiar jest wykonywany automatycznie i cyklicznie. Na **rysunku 23** pokazano wynik pomiaru mojej karty.



Rysunek 23. Charakterystyka zniekształceń karty Scarlett 2i2 gen. 3

Jeżeli jest przyciśnięty przycisk Show distortions, to w lewym górnym rogu pojawia się okno z wynikami pomiarów w procentach, tak jak zostało to ustawione – **rysunek 24**.

Pomiar wykonaliśmy za pomocą tonu o częstotliwości 1 kHz, poziomie –8,49 dBFS i w paśmie 20 Hz...22 kHz. Zniekształcenia harmoniczne ciągu pomiarowego generator–analizator wyniosły 0,00038%, a zniekształcenia plus szum THD+N 0,0021%. Przy okazji wspomnijmy, że THD+N bez ograniczania pasma do 20 Hz...22 kHz wyniósł 0,27%, czyli dość sporo jak na urządzenie pomiarowe.

1 000,00 Hz, -8,49 dBFS	
Span: 20 .. 22 000 Hz	
N+D: -104,3 dBFS A	THD: 0,00038 %
N: -102,0 dBFS	THD+N: 0,0021 %
HH 10 .. 22	HHD: 0,00014 %
2nd: 0,00022 %	3rd: 0,00025 %
4th: 0,000040 %	5th: 0,000061 %
6th: 0,000046 %	7th: 0,000048 %
8th: 0,000044 %	9th: 0,000044 %

Rysunek 24. Wyniki pomiarów zniekształceń (w procentach) karty Scarlett 2i2 gen. 3

Poniżej tych dwóch zasadniczych wartości podane są procentowe wyniki obliczeń poszczególnych harmonicznych. To też istotny pomiar w analizie pracy wzmacniaczy.

Teraz możemy określić, jak mają się zmierzone parametry do danych podanych przez producenta karty. W przypadku wejścia i wyjścia liniowego karty Scarlett 2i2 THD+N <0,002% z filtrem pasmowym 22 Hz...22 kHz, u nas wyszło THD+N równe 0,0021%, czyli praktycznie tyle, ile w karcie katalogowej w takich samych warunkach pomiaru. Z tego wynika, że nasz układ pomiarowy składający się z karty i oprogramowania działa prawidłowo i możemy przejść do pomiaru parametrów THD+N wzmacniacza mocy – tematem tym zajmiemy się w kolejnym odcinku naszego cyklu.

Tomasz Jabłoński, EP

eprasa.pl 4202fe84f2

Zagadkowy mikrokontroler TW32F003

W artykule zaprezentowano mikrokontroler typu TW32F003 (vel PY32F003F18P6) oraz omówiono metodę programowania jego pamięci Flash i ustawień konfiguracyjnych za pomocą programatora – debuggera ST-Link.

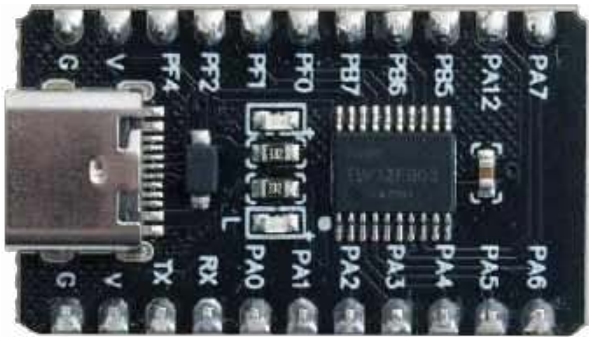
Przeglądając budżetowe mikrokontrolery z rdzeniem ARM Cortex M0/M0+, autor natrafił na sygnowany przez firmę Twen układ typu TW32F003. Mikrokontroler ten jest oferowany na internetowych portalach sprzedażowych głównie w postaci płytki uruchomieniowej o symbolu TWen32F003. Jej wygląd prezentuje **fotografia 1**.

Płytka uruchomieniowa

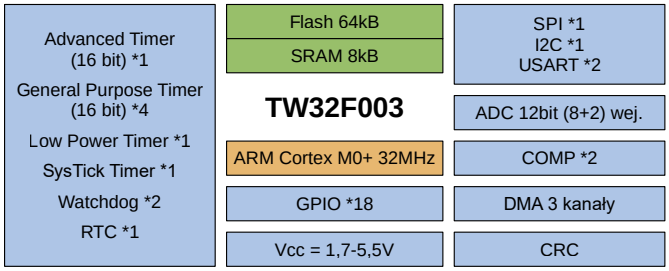
Płytka TWen32F003 jest jednym z wielu modułów opracowanych na potrzeby projektów z internetowej platformy edukacyjno-szkoleniowej http://haohaodada.com. Platforma ta koncepcyjnie zbliżona jest do znanego wśród programistów mikrokontrolerów ARM środowiska Mbed Online Compiler (obecnie już nierozwijanego i zastąpionego przez Keil Studio Cloud). Programy pisze się tu w oknie przeglądarki www, kompiluje w chmurze, a program wynikowy jest następnie przesyłany do płytki użytkownika w celu wykonania. Platforma http://haohaodada.com obsługuje wiele różnych dalekowschodnich mikrokontrolerów, zarówno 8-, jak i 32-bitowych, do których oferuje specjalne płytki uruchomieniowe. O ile zazwyczaj zainstalowane w nich mikrokontrolery są dość dobrze udokumentowane, o tyle w przypadku płytki TWen32F003 większość dostawców ogranicza się do stwierdzenia, że użyty w niej mikrokontroler jest wyposażony w 32-bitowy rdzeń ARM, nie określając nawet dokładnego typu rdzenia. Poszukiwania producenta TW32F003 w Internecie również nie przynoszą żadnych rezultatów. Można jedynie natrafić na schemat blokowy tego układu, obrazujący zaimplementowane w mikrokontrolerze peryferia. Schemat ten widnieje na **rysunku 1**.

Jak widać, mikrokontroler TW32F003 wyposażony został w rdzeń ARM Cortex M0+, który może być taktowany z maksymalną częstotliwością 32 MHz – oraz w stosunkowo bogaty, jak na budżetowy układ, zestaw peryferiów. Wbudowane pamięci Flash i SRAM mają odpowiednio rozmiar 64 kB i 8 kB. Znalazły się tu też typowe układy czasowo-licznikowe, interfejsy komunikacyjne SPI, I²C i UART, 2 watchdogi, 12-bitowy przetwornik analogowo-cyfrowy, 2 komparatory, sprzętowy blok obliczeniowy CRC, a nawet 3-kanałowy kontroler DMA, który rzadko występuje w mikrokontrolerach z tej półki cenowej. Na uwagę zasługuje także bardzo szeroki zakres napięć zasilania układu, rozciągający się od 1,7 V aż do 5,5 V.

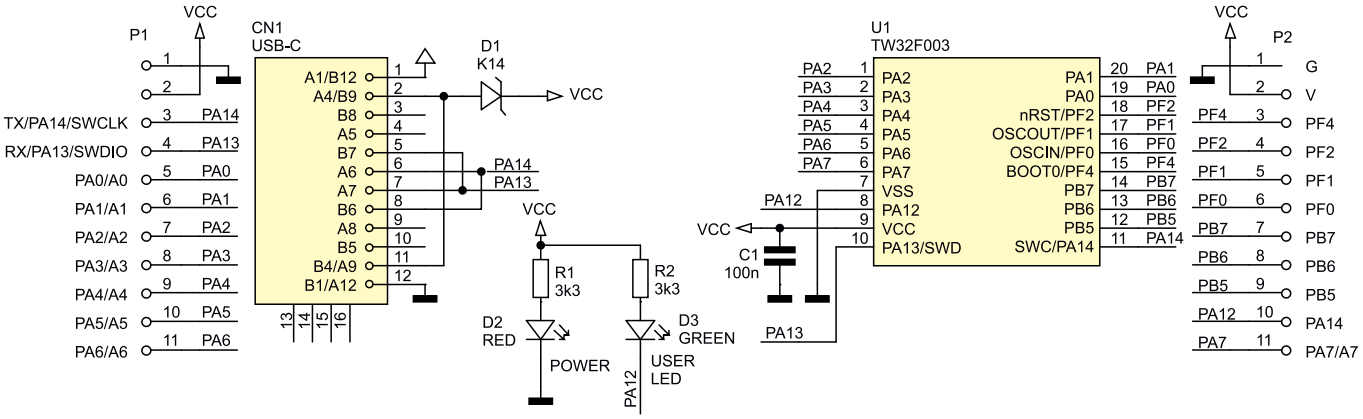
Schemat ideowy płytki uruchomieniowej, na której zainstalowano omawiany mikrokontroler, prezentuje **rysunek 2**. Trudno o prostszy



Fotografia 1. Płytka uruchomieniowa TWen32F003



Rysunek 1. Zasoby mikrokontrolera TW32F003



Rysunek 2. Schemat ideowy płytki TWen32F003

eprasa.pl 4202fe84f2

układ. Poza mikrokontrolerem i dwiema diodami LED, z których jedna sygnalizuje obecność napięcia zasilania, a druga jest przeznaczona do sterowania przez program użytkownika, na płytce nie znajduje się w zasadzie nic więcej. Nie ma tu nawet stabilizatora napięcia zasilania +3,3 V, typowo spotykanego w podobnych modułach.

Zgodnie z aktualnym trendem płytka jest wyposażona w złącze standardu USB C. Pewne zdziwienie może budzić fakt podłączenia portów PA13 i PA14 mikrokontrolera do linii danych DN i DP w gnieździe USB, ponieważ mikrokontroler TW32F003 nie ma wbudowanego interfejsu USB. Jak jednak wynika z diagramu na stronie platformy szkoleniowej, płytki uruchomieniowe nie są podłączane bezpośrednio do portu USB komputera PC, lecz do gniazda USB w programatorze – debuggerze typu STC Link. Kabel USB C łączący płytkę uruchomieniową z programatorem debuggerem zastępuje w tym przypadku zwykły przewód taśmowy interfejsów SWD i RS. Drugą opcją jest podłączenie płytki TWen32F003 do komputera PC za pośrednictwem modułu LILYGO T-U2T. Jest to – wykonany w formie „pendrive'a", zakończonego z obu stron gniazdami USB C – konwerter USB/UART-TTL. Używa się go w płytkach systemu LILYGO. Bezpośrednie podłączenie płytki uruchomieniowej TWen32F003 do portu USB w komputerze PC pozwala jedynie na jej zasilanie – układ nie jest wykrywany przez system operacyjny MS Windows jako urządzenie peryferyjne.

```
1fff0d40  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
1fff0d50  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
1fff0d60  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
1fff0d70  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ................
1fff0d80  32 33 59 50 33 30 30 46 50 38 31 46 30 30 30 36  23YP300FP81F0006
1fff0d90  ff 70 e0 00 00 00 00 00 00 90 17 08 00 00 00 00  .p..............
1fff0da0  59 87 00 01 00 0d 09 16 ff ff ff ff ff ff ff ff  Y...............
1fff0db0  ff ff ff ff ff ff ff ff aa aa aa aa aa aa aa aa  ................
```

Rysunek 3. Kod identyfikacyjny układu zapisany w pamięci systemowej mikrokontrolera TW32F003

Tabela 1. Mikrokontrolery rodziny PY32F0

Zasób	Typ		
	PY32F002A	PY32F003	PY32F030
Pamięć Flash	20 kB	16 kB/32 kB/64 kB	8 kB/16 kB/32 kB/48 kB/64 kB
Pamięć SRAM	3 kB	2 kB/4 kB/8 kB	2 kB/4 kB/6 kB/8 kB
Układ czasowo-licznikowy Advanced Timer	1 (16-bit)	1 (16-bit)	1 (16-bit)
Układ czasowo-licznikowy General Purpose Timer	1 (16-bit)	4 (16-bit)	4 (16-bit)
Układ czasowo-licznikowy Low Power Timer	1	1	1
Układ czasowo-licznikowy SysTick Timer	1	1	1
Układ czasowo-licznikowy Watchdog Timer	1	2	2
Układ komunikacyjny SPI	1	1	2
Układ komunikacyjny I²C	1	1	1
Układ komunikacyjny USART	1	2	2
Zegar RTC	0	1	1
Moduł CRC-32	1	1	1
Kontroler DMA	0	1 (3 kanały)	1 (3 kanały)
Sterownik wyświetlacza LED	0	0	1 (4 cyfry × 8 segmentów)
Porty IO	6/8/14/15/18	7/8/14/18	7/17/18/22/28/30
Przetwornik ADC	1 (12-bit)	1 (12-bit)	1 (12-bit)
Komparator analogowy	2	2	2
Max. częstotliwość zegara	24 MHz	32 MHz	48 MHz
Napięcie zasilania	1,7...5,5 V	1,7...5,5 V	1,7...5,5 V

TW32F003 vel PY32F003F18P6

Analiza listy zasobów sprzętowych mikrokontrolera TW32F003, jego zakresu napięcia zasilania oraz rozkładu wyprowadzeń pozwoliła na identyfikację układu. Okazuje się, że mikrokontroler TW32F003 to rebrandowany układ typu PY32F003F18P6 produkcji Puya Semiconductor, do którego pamięci Flash został wgrany dodatkowy bootloader obsługujący platformę http://haohaodada.com, natomiast wyprowadzenie nRST/PF2 zostało skonfigurowane jako port I/O. Potwierdza to kod identyfikacyjny, znajdujący się w pamięci systemowej mikrokontrolera pod adresem 0x1FFF 0D80 – 0x1FFF 0D8F (**rysunek 3**). Przestawienie znaków w kodzie identyfikacyjnym spowodowane jest zapisaniem ich w pamięci w postaci 32-bitowych słów w notacji Little Endian.

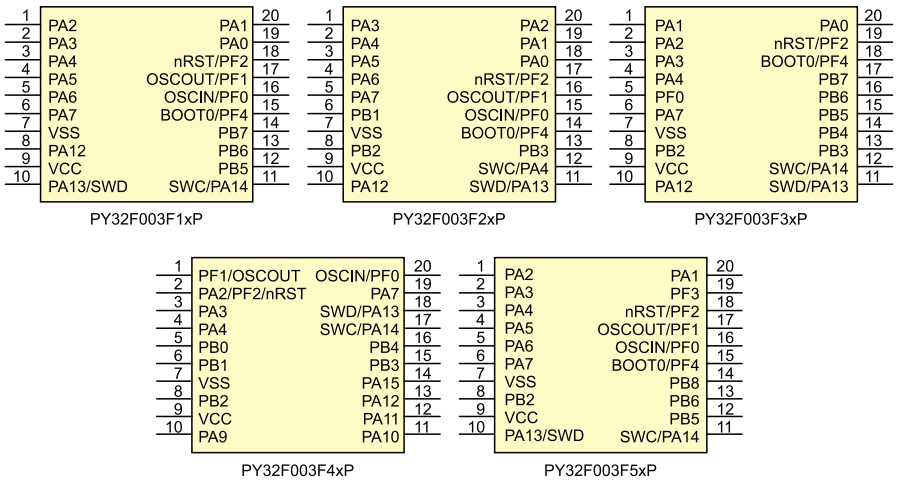
Mikrokontrolery PY32F0

Mikrokontrolery firmy Puya są dość znane. Zdobyły one rozgłos jako najtańsze na rynku procesory z rdzeniem ARM, dostępne w cenie już od 8 centów za sztukę [1]. Rodzina układów PY32F0 obejmuje trzy modele: PY32F002A, PY32F003 oraz PY32F030, przy czym układ typu PY32F002 okazuje się najprostszy spośród nich. Jest on oferowany tylko z jedną opcją rozmiaru pamięci Flash i SRAM, obniżoną częstotliwością taktowania rdzenia wynoszącą 24 MHz oraz zredukowanym zestawem peryferiów. Z kolei układ typu PY32F030 jest najbardziej rozbudowany: jego rdzeń może być taktowany z częstotliwością równą 48 MHz, obecny w nim generator sygnału zegarowego wyposażono w pętlę PLL, a sam układ ma dodatkowy interfejs SPI oraz sterownik wyświetlacza LED o organizacji 4 znaki po 8 segmentów. Wbrew temu, co sugeruje oznaczenie, układ PY32F030 nie stanowi klona mikrokontrolera STM32F030. Mimo wielu podobieństw, nie jest z nim zgodny programowo. Na przykład rejestry kontrolujące porty wejścia/wyjścia w obu układach znajdują się pod innymi adresami. Krótkie porównanie poszczególnych członków rodziny PY32F0 zawiera **tabela 1**. Pełna dokumentacja mikrokontrolerów PY32F0 – obejmująca karty katalogowe, podręczniki programowania, erraty oraz noty aplikacyjne – jest dostępna zarówno na stronie producenta [2], jak i na portalach prowadzonych przez entuzjastów mikrokontrolerów firmy Puya [3]. Dokumentacja ta dostarczona została w większości w języku chińskim, jednak podstawowe dokumenty (takie jak karty katalogowe i podręczniki programowania) są też dostępne w języku angielskim.

Mikrokontrolery PY32F0 można nabyć w wielu typach obudów, poczynając od najmniejszych z 8 wyprowadzeniami, aż do największych o 32 wyprowadzeniach. Są to standardowe obudowy typu SOP, MSOP, TSSOP, DFN, QFN i LQFP. Pojawiła się też jedna nietypowa obudowa ESSOP10 – o 10 wyprowadzeniach w rastrze 1 mm. Kolejną rzadko spotykaną cechą tych układów okazuje się

eprasa.pl 4202fe84f2

fakt, że mikrokontroler w ramach danej obudowy jest oferowany z różnymi wariantami układu wyprowadzeń. Przykładowe zestawienie dostępnych odmian mikrokontrolera typu PY32F003 w obudowie TSSOP20 pokazuje **rysunek 4**.



Rysunek 4. Warianty układu wyprowadzeń mikrokontrolera PY32F003 w obudowie TSSOP20

Metody programowania

Programy na mikrokontrolery PY32F0 (a więc i układ TW32F003) mogą być pisane w dowolnym środowisku programistycznym obsługującym układy ARM z rdzeniem Cortex M0/M0+. Firma Puya dostarcza biblioteki programistyczne oraz przykładowe programy. Biblioteki obsługi peryferiów obejmują zarówno wysokopoziomową wersję bibliotek HAL, jak i wersję niskopoziomową LL. Jak ma to zazwyczaj miejsce w przypadku układów produkcji chińskiej, dostarczane biblioteki i przykładowe programy są dostosowane do kompilatorów firm Keil oraz IAR. W przypadku kompilatora gcc konieczna jest modyfikacja kodów startowych w celu ich dostosowania do wymagań tego narzędzia. Nie ma jednak potrzeby modyfikowania bibliotek osobiście, ponieważ w Internecie można znaleźć biblioteki przygotowane do zastosowania z kompilatorem gcc, np. autorstwa Jaya Carlsona [1].

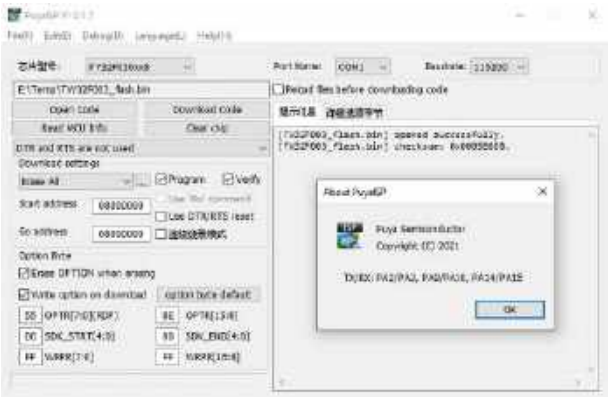
Zapis programu wynikowego do pamięci Flash mikrokontrolera TW32F003 można wykonać za pomocą:

- bootloadera platformy http://haohaodada.com,
- fabrycznego bootloadera,
- programatora SWD.

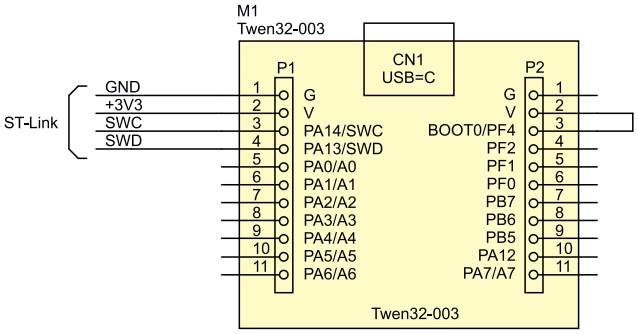
Pierwszy sposób byłby w naszych warunkach mało przydatny. Poza tym bootloader ten zajmuje początkowe 2 kB pamięci Flash, ograniczając tym samym maksymalny dopuszczalny rozmiar aplikacji użytkownika. Najlepiej więc skasować go, zwalniając całą pamięć Flash na potrzeby własnego programu.

Programowanie pamięci Flash za pomocą fabrycznego bootloadera wymaga użycia programu PuyaISP (**rysunek 5**). Bootloader systemowy jest aktywowany po podaniu na linię BOOT0 (PF4) stanu wysokiego i zrestartowaniu mikrokontrolera. Programowanie odbywa się przez interfejs szeregowy UART przy użyciu linii PA2 (sygnał TxD) i PA3 (sygnał RxD). Do podłączenia płytki TWen32F003 do komputera PC wymagany jest konwerter UART-TTL/USB. Program sterujący, oprócz kasowania i zapisu pamięci Flash, pozwala też na odczyt i zapis ustawień konfiguracyjnych mikrokontrolera (tzw. Option Bytes). Ponieważ program PuyaISP pracuje tylko w wersji okienkowej w 64-bitowym systemie MS Windows, problematyczna okazuje się jego integracja z używanym środowiskiem IDE. Poza tym nie pozwala on na debugowanie wykonywanego przez mikrokontroler programu.

Z tego też powodu autor do programowania mikrokontrolera TW32F003 zastosował programator-debugger typu ST-Link oraz program sterujący pyStlink, pozwalające na programowanie układu przez interfejs SWD. Schemat połączenia płytki TWen32F003 z programatorem STLink pokazuje **rysunek 6**, natomiast na **fotografii tytułowej** pokazano rzeczywisty układ używający programatora-debuggera ST-Link wchodzącego w skład płytki STM32F0 Discovery. Programator-debugger jest podłączony do portów PA13 (linia SWDIO) i PA14 (linia SWCLK) mikrokontrolera. Dodatkowo port PF4 (linia BOOT0) mikrokontrolera dołączono do szyny zasilania, co wymusza wykonywanie przez mikrokontroler kodu fabrycznego bootloadera. Okazało się to konieczne, ponieważ bootloader platformy http://haohaodada.com, z którym dostarczana jest płytka TWen32F003, zmienia przy starcie konfigurację portów PA13 i PA14, odłączając je od układu SWD i blokując tym samym możliwość połączenia z programatorem-debuggerem.



Rysunek 5. Okno programatora ISP dla mikrokontrolerów firmy Puya



Rysunek 6. Schemat podłączenia programatora ST-Link do płytki Twen32F003

Obsługa układów PY32F0 i TW32F003 w programie pyStlink

Użycie programatora-debuggera ST-Link wraz z programem pyStlink do programowania innych mikrokontrolerów niż układy STM32 było już szczegółowo opisywane na łamach „Elektroniki Praktycznej" [4], dlatego też informacje o samym pyStlink, jego obsłudze, jak również sposobie dodawania nowych mikrokontrolerów nie będą tutaj powtarzane – zainteresowanych Czytelników zachęcamy do lektury wspomnianego artykułu. Niniejszy opis skupia się na implementacji w programie pyStlink obsługi układu TW32F003, która okazała się nieco bardziej złożona niż w przypadku omówionych we wspomnianym artykule układów typu HC32F003/HC32F005. Gwoli przypomnienia, dodanie w programie pyStlink obsługi nowego mikrokontrolera wymaga przygotowania opisu tego układu oraz napisania sterownika obsługi jego pamięci Flash.

Pierwszym problemem przy tworzeniu opisu mikrokontrolera TW32F003 okazała się automatyczna detekcja przez program pyStlink typu podłączonego układu. W dokumentacji mikro-

eprasa.pl 4202fe84f2

kontrolerów PY32F0 podawana jest co prawda lokalizacja rejestru IDCODE zawierającego kod identyfikacyjny układu (oraz skojarzonej z nim komórki w pamięci systemowej zlokalizowanej pod adresem 0x1FFF 0FF8), jednak bez podania identyfikatorów przypisanych poszczególnym typom mikrokontrolerów rodziny PY32F0. Z tego też powodu rozpoznawanie układu oparto na wspomnianym wcześniej tekstowym identyfikatorze typu mikrokontrolera, zlokalizowanym w pamięci systemowej pod adresem 0x1FFF 0D80 – 0x1FFF 0D8F (rysunek 3). Pierwsze cztery bajty tego identyfikatora zostały potraktowane tak, jakby były one rejestrem IDCODE w mikrokontrolerze STM32. Przy takim podejściu kod identyfikacyjny mikrokontrolera TW32F003 odczytywany przez program pyStlink jest równy 0x332.

```
'part_no': 0xc60,
'core': 'CortexM0+',
'idcode_reg': [0x40015800, 0x00100C60, 0x1FFF0D80],
'devices': [
    ..........................................

    {
        'dev_id': 0x332,  # Puya Semiconductor
        'flash_size_reg': 0,
        'flash_driver': 'PY32F0',
        'erase_sizes': (128, ),
        'devices': [
            {'type': 'PY32F002A',   'flash_size': 20, 'sram_size': 3, 'eeprom_size': 0,
             'freq': 24},
            {'type': 'PY32F003XX4', 'flash_size': 16, 'sram_size': 2, 'eeprom_size': 0,
             'freq': 32},
            {'type': 'PY32F003XX6', 'flash_size': 32, 'sram_size': 4, 'eeprom_size': 0,
             'freq': 32},
            {'type': 'PY32F003XX8', 'flash_size': 64, 'sram_size': 8, 'eeprom_size': 0,
             'freq': 32},
            {'type': 'PY32F030XX3', 'flash_size': 8,  'sram_size': 2, 'eeprom_size': 0,
             'freq': 48},
            {'type': 'PY32F030XX4', 'flash_size': 16, 'sram_size': 2, 'eeprom_size': 0,
             'freq': 48},
            {'type': 'PY32F030XX6', 'flash_size': 32, 'sram_size': 4, 'eeprom_size': 0,
             'freq': 48},
            {'type': 'PY32F030XX7', 'flash_size': 48, 'sram_size': 6, 'eeprom_size': 0,
             'freq': 48},
            {'type': 'PY32F030XX8', 'flash_size': 64, 'sram_size': 8, 'eeprom_size': 0,
             'freq': 48},
            {'type': 'TW32F003',    'flash_size': 64, 'sram_size': 8, 'eeprom_size': 0,
             'freq': 32},
        ],
    },
],
```

Listing 1. Opis mikrokontrolerów PY32F0 i TW32F003 w programie pyStlink (plik devices.py)

Drugi problem stanowiła identyfikacja rozmiaru pamięci Flash wbudowanej w mikrokontroler. Dokumentacja układu PY32F003 nie wymienia rejestru zawierającego informację o rozmiarze zaimplementowanej w układzie pamięci. Co prawda w dostarczanych przez producenta plikach nagłówkowych *py32f003xx.h*, definiujących zasoby mikrokontrolerów PY32F003, znajduje się deklaracja:

```
#define FLASHSIZE_BASE  0x1FFF0FFCUL  /*!< FLASH Size register base address */
```

określająca lokalizację rozmiaru pamięci Flash w pamięci systemowej, jednak zawartość tej komórki pamięci nie odpowiada rzeczywistemu rozmiarowi pamięci Flash w mikrokontrolerze. Problem ten rozwiązano przez modyfikację w programie pyStlink algorytmu wyszukiwania typu podłączonego układu. W programie dodano warunek, że w przypadku braku odczytu kodu identyfikacyjnego mikrokontrolera lub też braku odczytu rozmiaru pamięci Flash, program pyStlink posiłkuje się podanym w linii komend typem układu (parametr `-c` wywołania programu), wyszukując w słowniku obsługiwanych układów mikrokontroler wskazanego typu. W przypadku znalezienia zgodnego układu, rozmiar pamięci i nazwa sterownika obsługującego pamięć Flash w mikrokontrolerze są pobierane ze znalezionego opisu. Dodany w słowniku obsługiwanych przez program pyStlink układów opis mikrokontrolerów PY32F0 i TW32F003 pokazuje **listing 1**. Wynik identyfikacji przez program pyStlink tak zdefiniowanego układu TW32F003 prezentuje natomiast **rysunek 7**.

Zgodnie z konwencją przyjętą w programie pyStlink, sterownik obsługi pamięci Flash mikrokontrolerów PY32F0 i TW32F003 zlokalizowano w oddzielnym pliku o nazwie *py32f0.py*. Sterownik ten jest tworzony jako klasa pochodna klasy stm32. Dzięki temu wymaga on jedynie napisania metody inicjalizacji mikrokontrolera, metod kasowania całej pamięci Flash i pojedynczych sektorów oraz metody zapisu pamięci Flash. Pozostałe funkcje sterownika są dziedziczone z klasy stm32 i nie wymagają modyfikacji.

```
\python.exe pyStlink.py -c tw32f003
DEVICE: ST-Link/V2 V2J40S0
SUPPLY: 2.91V
CORE:   CortexM0+
MCU:    TW32F003
FLASH:  64kB
SRAM:   8kB
```

Rysunek 7. Identyfikacja mikrokontrolera TW32F003 przez program pyStlink

Uproszczona funkcja inicjalizacji mikrokontrolerów PY32F0 i TW32F003, pozbawiona nadmiarowego kodu odpowiedzialnego za debugowanie, zaprezentowana została na **listingu 2**. Inicjalizacja układu polega na włączeniu w rejestrze RCC_CR oscylatora HSI (tj. wewnętrznego generatora RC), ustawieniu w rejestrze RCC_ICSCR w polu HSI_FS częstotliwości pracy bloku

```
# HSI trim data addr
  TRIM_HSI_24MHZ     = 0x1FFF0F10

# Flash controller timers settings at SYSCLK=24MHz (reg. addr, size, trim data addr)
FLASH_TIM_CFG_24MHZ = (
    {'reg': FLASH_TS0,    'size': 'Byte', 'trim_data': 0x1FFF0F6C},
    {'reg': FLASH_TS1,    'size': 'Word', 'trim_data': 0x1FFF0F6E},
    {'reg': FLASH_TS2P,   'size': 'Byte', 'trim_data': 0x1FFF0F70},
    {'reg': FLASH_TPS3,   'size': 'Word', 'trim_data': 0x1FFF0F72},
    {'reg': FLASH_TS3,    'size': 'Byte', 'trim_data': 0x1FFF0F6D},
    {'reg': FLASH_PERTPE, 'size': 'Long', 'trim_data': 0x1FFF0F74},
    {'reg': FLASH_SMERTPE,'size': 'Long', 'trim_data': 0x1FFF0F78},
    {'reg': FLASH_PRGTPE, 'size': 'Word', 'trim_data': 0x1FFF0F7C},
    {'reg': FLASH_PRETPE, 'size': 'Word', 'trim_data': 0x1FFF0F7E},
)


def nvm_init(self):
    self._driver.core_reset_halt()

    # set SYSCLK to HSI=24MHz
    clk = self.RCC_CR_HSIDIV_1 | self.RCC_CR_HSION
    self._stlink.set_debugreg32(self.RCC_CR, clk)

    clk_trim = self._stlink.get_debugreg16(self.TRIM_HSI_24MHZ)
    clk_trim |= self.RCC_HSI_FS_24MHZ
    self._stlink.set_mem16(self.RCC_ICSCR,
                           list(clk_trim.to_bytes(2, byteorder='little')))

    clk = self.RCC_CFGR_PPRE_1 | self.RCC_CFGR_HPRE_1 | self.RCC_CFGR_SW_HSISYS
    self._stlink.set_debugreg32(self.RCC_CFGR, clk)

    # init Flash controller for SYSCLK=24MHz
    self.unlock_nvm()
    self._stlink.set_debugreg32(self.FLASH_ACR, self.FLASH_LATENCY_0)

    # init Flash controller timers
    for tim_cfg in self.FLASH_TIM_CFG_24MHZ:
      reg = tim_cfg['reg']
      if tim_cfg['size'] == 'Byte':
        data = self._stlink.get_debugreg8(tim_cfg['trim_data'])
      elif tim_cfg['size'] == 'Word':
        data = self._stlink.get_debugreg16(tim_cfg['trim_data'])
      else:
        data = self._stlink.get_debugreg32(tim_cfg['trim_data'])
      self._stlink.set_debugreg32(reg, data)

      self.lock_nvm()
```

Listing 2. Uproszczona funkcja inicjalizacji mikrokontrolerów rodziny PY32F0

eprasa.pl 4202fe84f2

Tabela 2. Mapa rejestrów kontrolera pamięci Flash (adres bazowy 0x4002 2000)

L.p.	Offset	Rejestr	Realizowana funkcja
1	0x00	FLASH_ACR	Ustawienie cykli opóźnienia dostępu do pamięci Flash
2	0x08	FLASH_KEYR	Odblokowanie modyfikacji rejestrów kontrolera
3	0x0C	FLASH_OPTKEYR	Odblokowanie modyfikacji ustawień konfiguracyjnych układu (tzw. Option Bytes)
4	0x10	FLASH_SR	Rejestr stanu operacji
5	0x14	FLASH_CR	Rejestr kontrolny
6	0x20	FLASH_OPTR	Ustawienia konfiguracyjne układu (Option Byte OPT)
7	0x24	FLASH_SDKR	Ochrona dostępu do obszaru pamięci Flash z kodem SDK (Option byte SDK)
8	0x2C	FLASH_WRPR	Blokada przed zapisem sektorów pamięci Flash (Option byte WRP)
9	0x90	FLASH_STCR	Rejestr kontrolny trybu uśpienia
10	0x100	FLASH_TS0	Ustawienie czasu TS0
11	0x104	FLASH_TS1	Ustawienie czasu TS1
12	0x108	FLASH_TS2P	Ustawienie czasu TS2P
13	0x10C	FLASH_TPS3	Ustawienie czasu TPS3
14	0x110	FLASH_TS3	Ustawienie czasu TS3
15	0x114	FLASH_PERTPE	Ustawienie czasu TPE kasowania strony pamięci
16	0x118	FLASH_SMERTPE	Ustawienie czasu TPE kasowania sektora/kasowania całej pamięci
17	0x11C	FLASH_PRGTPE	Ustawienie czasu TPE zapisu strony pamięci
18	0x120	FLASH_PRETPE	Ustawienie czasu TPE poprzedzającego zapis strony

Rejestr FLASH_CR
Adres: 0x4002 2014

b31	b30	b29 b28	b27	b26	b25	b24	b23 b20	b19	b18	b17	b16 b12	b11	b10 b3	b2	b1	b0
LOCK	OPTLOCK	Res	OBL_LAUNCH	Res	ERRIE	EOPIE	Res	PGSTRT	Res	OPTSTRT	Res	SER	Res	MER	PER	PG

b31 – ochrona rejestrów kontrolera pamięci Flash przed zapisem
b30 – ochrona rejestrów ustawień konfiguracyjnych Option Bytes przed zapisem
b29:28 – bity zarezerwowane
b27 – żądanie załadowania ustawień konfiguracyjnych Option Bytes
b26 – bit zarezerwowany
b25 – odblokowanie przerwania błędu operacji INT_ERR
b24 – odblokowanie przerwania zakończenia operacji INT_EOP
b23:20 – bity zarezerwowane
b19 – start zapisu pamięci Flash
b18 – bit zarezerwowany
b17 – start zapisu ustawień konfiguracyjnych Option Bytes
b16:12 – bity zarezerwowane
b11 – operacja kasowania sektora pamięci Flash
b10:3 – bity zarezerwowane
b2 – operacja kasowania całej pamięci Flash
b1 – operacja kasowania strony pamięci Flash
b0 – operacja zapisu pamięci Flash

Rysunek 8. Rejestr kontrolny FLASH_CR

HSI i przepisaniu do pola HSI_TRIM tego rejestru ustawień kalibracyjnych układu HSI (dla wybranej częstotliwości pracy) pobranych z pamięci systemowej. Na koniec trzeba jeszcze ustawić w rejestrze kontrolnym RCC_CFGR oscylator HSI jako źródło sygnału zegarowego mikrokontrolera. W układzie TW32F003 blok HSI może generować sygnał zegarowy o częstotliwości 4 MHz, 8 MHz, 16 MHz, 22,12 MHz i 24 MHz. Jednak w przypadku modelu PY32F002A dane kalibracyjne są zdefiniowane tylko dla częstotliwości 8 MHz i 24 MHz. Aby więc sterownik obsługiwał prawidłowo wszystkie układy z rodziny PY32F0, jako częstotliwość sygnału zegarowego HSI została wybrana wartość 24 MHz. Następnie do rejestrów kontrolera pamięci Flash przepisywane zostają z pamięci systemowej skalibrowane ustawienia timerów, ustalające cykle programowania i kasowania pamięci Flash przy zaprogramowanej częstotliwości sygnału zegarowego (tj. 24 MHz). Dostęp do rejestrów kontrolera Flash jest chroniony – z tego też względu przed ich modyfikacją konieczne jest odblokowanie dostępu przez wpisanie do rejestru FLASH_KEYR sekwencji kodów 0x4567 0123 i 0xCDEF 89AB, co realizuje funkcja unlock_nvm(). Ponowne włączenie ochrony rejestrów kontrolera pamięci Flash odbywa się przez ustawienie bitu LOCK w rejestrze kontrolnym FLASH_CR. Zadanie to wykonuje funkcja lock_nvm(), która kończy inicjalizację układu.

Wszystkie operacje na pamięci programu mikrokontrolera (tj. zapis pamięci, jej kasowanie, jak również kasowanie sektorów i kasowanie stron) są wykonywane przez kontroler pamięci Flash. Mapę jego rejestrów zawiera **tabela 2**. Poszczególne operacje kontrolera Flash są wywoływane przez ustawienie odpowiedniego bitu w rejestrze kontrolnym FLASH_CR, którego strukturę pokazuje **rysunek 8**.

Pamięć Flash w mikrokontrolerach PY32F0 może być kasowana całkowicie pojedynczymi sektorami o rozmiarze 4 kB oraz pojedynczymi stronami o rozmiarze 128 B. Kasowanie zawartości pamięci Flash polega na ustawieniu bitu operacji w rejestrze kontrolnym FLASH_CR (odpowiednio bitu MER – w przypadku kasowania pamięci, bitu SER – w przypadku kasowania sektora i bitu PER – w przypadku kasowania strony), a następnie zapisie 32-bitowego słowa pod adres znajdujący się odpowiednio: w obszarze pamięci Flash, kasowanego sektora lub kasowanej strony. Uproszczony kod funkcji kasowania całej pamięci Flash i wybranych stron pokazuje **listing 3**. W sterowniku nie zaimplementowano obsługi kasowania sektorów, ponieważ nie było takiej potrzeby.

Programowanie pamięci Flash w układach PY32F0 odbywa się całymi stronami o rozmiarze 128 B. W związku z tym adres

eprasa.pl 4202fe84f2

początkowy zapisywanych danych musi ulec wyrównaniu do adresu strony. Warunek ten kontroluje funkcja zapisu pamięci Flash. Następnie rozmiar zapisywanych danych jest dopełniany do pełnej wielokrotności rozmiaru strony pamięci Flash.

Zapis pamięci jest wykonywany tylko słowami o rozmiarze 32 bitów. Polega on na ustawieniu bitu operacji PG w rejestrze kontrolnym FLASH_CR, następnie zapisie pod kolejnymi adresami na programowanej stronie pamięci Flash 31 słów danych, każde o rozmiarze 32 bitów, inicjalizacji operacji zapisu przez ustawienie bitu PGSTART w rejestrze kontrolnym FLASH_CR i zapisie ostatniego 32-bitowego słowa danych pod końcowy adres na programowanej stronie. Wysłanie trzydziestego drugiego 32-bitowego słowa uruchamia zapis strony pamięci Flash. Zakończenie procesu jest sygnalizowane przez bit końca operacji (EOP) w rejestrze stanu FLASH_SR. W przypadku, gdy zapis strony się nie powiódł, bit ten pozostaje wyzerowany. Jeżeli błąd zapisu jest spowodowany ustawionym zabezpieczeniem sektora pamięci Flash przed zapisem, w rejestrze stanu dodatkowo ustawiony zostaje bit WRPERR. Uproszczony kod funkcji zapisu pamięci Flash opisuje **listing 4**. **Rysunek 9** pokazuje natomiast wynik procesu programowania pamięci Flash w mikrokontrolerze TW32F003.

```
def erase_all(self):
    self.unlock_nvm()
    self.set_operation(self.FLASH_CR_EOPIE | self.FLASH_CR_MER)
    self._stlink.set_mem32(self._driver.FLASH_START, [0xFF, 0xFF, 0xFF, 0xFF])
    self.wait_busy(1, 'Erasing FLASH')
        if self.get_operation_result() == self.OPR_ERROR:
            raise lib.stlinkex.StlinkException('Error erasing chip.'
                                               'Check write protection!')
    self.lock_nvm()


def erase_pages(self, addr, size):
    self.unlock_nvm()
    page_addr = addr & ~(self._driver.FLASH_PAGE_SIZE - 1)
    end_addr = addr + size
    while page_addr < end_addr:
        self.set_operation(self.FLASH_CR_EOPIE | self.FLASH_CR_PER)
        self._stlink.set_mem32(page_addr, [0xFF, 0xFF, 0xFF, 0xFF])
        self.wait_busy(1, 'Erasing FLASH page No %d'
             %((page_addr - self._driver.FLASH_START) // self._driver.FLASH_PAGE_SIZE))
        if self.get_operation_result() == self.OPR_ERROR :
            raise lib.stlinkex.StlinkException('Error erasing FLASH page No %d.'
                                               'Check write protection!'
             %((page_addr - self._driver.FLASH_START) // self._driver.FLASH_PAGE_SIZE))
        page_addr += self._driver.FLASH_PAGE_SIZE
    self.lock_nvm()
```

Listing 3. Uproszczone funkcje kasowania całej pamięci Flash oraz kasowania wybranych stron

```
def flash_write(self, addr, data, erase=False, erase_sizes=None):
    if addr is None:
        addr = self.FLASH_START
    if addr % self.FLASH_PAGE_SIZE:
        # Flash is programmed in full pages only
        raise lib.stlinkex.StlinkException('Start address is not aligned '
                                           'to FLASH page!')
    if len(data) % self.FLASH_PAGE_SIZE:
        # Pad data to page size
        data.extend([0xff] * (self.FLASH_PAGE_SIZE - len(data) % self.FLASH_PAGE_SIZE))

    flash = Flash(self, self._stlink, self._dbg)

    if erase:
        if erase_sizes:
            flash.erase_pages(addr, len(data))
        else:
            flash.erase_all()

    flash.unlock_nvm()
    flash.set_operation(flash.FLASH_CR_EOPIE | flash.FLASH_CR_PG)
    while data:
        data_block = data[:self.FLASH_PAGE_SIZE]
        data_addr = addr
        self._stlink.set_mem32(data_addr, data_block[:self.FLASH_PAGE_SIZE-4])
        flash.set_operation(flash.FLASH_CR_PGSTRT)
        self._stlink.set_mem32(data_addr + self.FLASH_PAGE_SIZE-4,
                               data_block[self.FLASH_PAGE_SIZE-4:])
        flash.wait_busy(0.005)
        if flash.get_operation_result() == flash.OPR_ERROR:
           raise lib.stlinkex.StlinkException('Writing FLASH page addr 0x%08x failed. '
                                              'Check page write protection!' %(data_addr))
        data = data[self.FLASH_PAGE_SIZE:]
        addr += self.FLASH_PAGE_SIZE
    flash.clear_operation(flash.FLASH_CR_PG)
    flash.lock_nvm()
```

Listing 4. Uproszczona funkcja zapisu pamięci Flash

Ustawienia konfiguracyjne Option Bytes

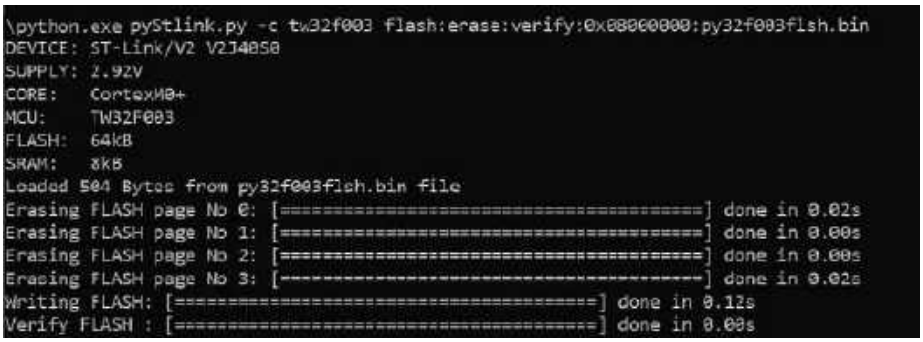
Niezależne od realizowanego programu ustawienia konfiguracyjne mikrokontrolera są w układach PY32F0 i TW32F003 zgrupowane w tzw. Option Bytes. Są to:

- ustawienia użytkownika (User option),
- adres obszaru chronionego SDK (SDK option),
- blokady sektorów pamięci Flash przed zapisem (WRP address).

Parametry te przechowuje się na odpowiedniej stronie systemowej pamięci Flash w postaci trzech 32-bitowych słów, przy czym poszczególne ustawienia są przypisane do bitów b15...b0 w danym słowie, natomiast bity b31..b16 pełnią funkcję kontroli błędów. W nieuszkodzonym słowie są one równe zanegowanym wartościom bitów b15...b0. Przy starcie mikrokontrolera bity b15...b0 są kopiowane do specjalnych rejestrów kontrolera pamięci Flash, odpowiednio FLASH_OPTR, FLASH_SDKR i FLASH_WRPR. **Tabela 3** zawiera zestawienie adresów słów konfiguracyjnych w systemowej pamięci Flash oraz odpowiadających im rejestrów kontrolera pamięci Flash.

Strukturę rejestru z ustawieniami użytkownika prezentuje **rysunek 10**. Ustawienia te obejmują konfigurację sygnału BOOT1, funkcję wejścia nRST, automatyczne uruchamianie timerów watchdog, próg zadziałania układu kontroli napięcia zasilania (BOR) oraz ochronę pamięci Flash przed odczytem.

Strukturę rejestru z konfiguracją obszaru chronionego SDK w pamięci Flash pokazuje **rysunek 11**. Ustawienia te definiują obszar pamięci, z której kod może być wykonywany, ale nie może być odczytywany przez instrukcje wykonywane poza tym obszarem (tj. zlokalizowane w pozostałej części pamięci Flash lub w pamięci SRAM) przez kontroler DMA ani też przez interfejs SWD. Obszar ten jest przeznaczony do przechowywania kodu (np. bibliotek),



Rysunek 9. Zapis pamięci Flash mikrokontrolera TW32F003 przez program pyStlink

eprasa.pl 4202fe84f2

Rejestr FLASH_OPTR
Adres: 0x4002 2020

b31	b16	b15	b14	b13	b12	b11	b9	b8	b7	b0
Res	nBOOT1	NRST_MODE	WWDG_SW	IWDG_SW	BOR_LEV [2:0]			BOR_EN	RDP[7:0]	

b31:16 – bity zarezerwowane
b15 – stan linii nBOOT1
b14 – konfiguracja pinu NRST (0 – wejście RESET, 1 – port GPIO)
b13 – tryb pracy układu WWDG (0 – uruchamiany sprzętowo, 1 – uruchamiany przez program)
b12 – tryb pracy układu IWDG (0 – uruchamiany sprzętowo, 1 – uruchamiany przez program)
b11:9 – próg zadziałania układu BOR
000 – zbocze narastające: 1,8 V, zbocze opadające: 1,7 V
001 – zbocze narastające: 2,0 V, zbocze opadające: 1,9 V
010 – zbocze narastające: 2,2 V, zbocze opadające: 2,1 V
011 – zbocze narastające: 2,4 V, zbocze opadające: 2,3 V
100 – zbocze narastające: 2,6 V, zbocze opadające: 2,5 V
101 – zbocze narastające: 2,8 V, zbocze opadające: 2,7 V
110 – zbocze narastające: 3,0 V, zbocze opadające: 2,9 V
111 – zbocze narastające: 3,2 V, zbocze opadające: 3,1 V
b8 – aktywacja układu BOR (0 – wyłączony, 1 – włączony)
b7:0 – ochrona pamięci Flash przed odczytem
0xAA – poziom 0 (ochrona wyłączona)
≠ 0xAA – poziom 1 (ochrona włączona)

Rysunek 10. Rejestr ustawień konfiguracyjnych FLASH_OPTR

Uwaga:
Przy definiowaniu konfiguracji sygnału BOOT1 i wejścia nRST należy zachować szczególną ostrożność, ponieważ potencjalnie można na stałe i nieodwracalnie zablokować dostęp do programowania pamięci Flash oraz debugowania układu. Po wyzerowaniu bitu nBOOT1 ustawienie pinu PF4/BOOT0 w stanie wysokim nie aktywuje już bowiem fabrycznego bootloadera. Jeżeli jednocześnie wejście nRST zostało skonfigurowane jako port I/O oraz program użytkownika zmienia konfigurację fabryczną portów PA13 i PA14 odłączając je od układu SWD (co zazwyczaj ma miejsce przy stosowaniu mikrokontrolera z małą liczbą wyprowadzeń), to programator-debugger SWD nie będzie już w stanie podłączyć się do układu i tym samym nie będzie możliwe ani programowanie pamięci Flash, ani też zmiana ustawień konfiguracyjnych mikrokontrolera. Aby uniknąć takiej sytuacji, dobrze jest konfigurować porty PA13 i PA14 w programie z pewnym opóźnieniem, a nie natychmiast po starcie mikrokontrolera, co powinno dać czas programatorowi-debuggerowi na zestawienie połączenia z układem.

którego dostawca nie chce udostępniać użytkownikowi nawet w postaci binarnej. Po wgraniu go do obszaru chronionego SDK przez dostawcę i włączeniu ochrony użytkownik końcowy może rozwijać swoją aplikację, traktując zapisany kod w obszarze SDK jako czarną skrzynkę o zdefiniowanych przez dostawcę wejściach i wyjściach. Ochrona obszaru SDK jest aktywna, gdy adres początkowy SDK_STRT pozostaje równy lub mniejszy od adresu końcowego SDK_END. W przeciwnym przypadku ochrona jest wyłączona. Adresy: początkowy i końcowy obszaru SDK, są definiowane z krokiem 2 kB.

Strukturę rejestru z ustawieniami blokad sektorów pamięci Flash przed zapisem pokazuje **rysunek 12**. Bity b15...b0 odpowiadają stanowi ochrony przed zapisem sektorów pamięci Flash (każdy o rozmiarze 4 kB), odpowiednio o numerach S15...S0.

Oryginalny program pyStlink nie obsługuje programowania ustawień konfiguracyjnych mikrokontrolera. Aby uzyskać tę funkcjonalność dla układów PY32F003 i TW32F003, w klasie stm32 zdefiniowano dwie dodatkowe metody: *ob_get_all()* i *ob_write()*, realizujące odpowiednio odczyt ustawień konfiguracyjnych i zapis wybranego słowa konfiguracyjnego. Implementację tych metod w sterowniku obsługi pamięci Flash PY32F0 w programie pyStlink zawiera **listing 5**.

Odczyt ustawień konfiguracyjnych układu ogranicza się do pobrania zawartości rejestrów kontrolera pamięci Flash (zawierających kopie danych konfiguracyjnych) – i ich wyświetlenia. Funkcja ta w programie pyStlink została podpięta pod parametr wywołania programu:

dump:ob

Wynik jej działania pokazuje **rysunek 13**.

Zapis nowych ustawień konfiguracyjnych mikrokontrolera polega na ich wpisaniu do odpowiedniego rejestru Option Bytes w kontrolerze pamięci Flash (tj. rejestru FLASH_OPTR – w przypadku modyfikacji ustawień użytkownika, rejestru FLASH_SDKR – w przypadku modyfikacji obszaru chronionego SKD i rejestru FLASH_WRPR – w przypadku modyfikacji ochrony sektorów przed zapisem), a następnie ustawieniu bitu operacji OPTSTRT w rejestrze kontrolnym FLASH_CR i zapisie 32-bitowego słowa pod adres 0x4002 2080. Operacja ta inicjuje skopiowanie bitów b15...b0 z rejestrów Option Bytes w kontrolerze Flash do powiązanych z nimi komórek pamięci systemowej Flash oraz ustawienie w tych komórkach zanegowanych wartości bitów na pozycjach b31...b16. Zakończenie procesu jest sygnalizowane przez bit końca operacji (EOP) w rejestrze stanu FLASH_SR. Nowe ustawienia zostają zapisane w komórkach pamięci systemowej Flash, ale pozostają nieaktywne.

Rejestr FLASH_SDKR
Adres: 0x4002 2024

b31	b13	b12	b8	b7	b5	b4	b0
Res	SA_END[4:0]			Res	SA_STRT[4:0]		

b31:13 – bity zarezerwowane
b12:8 – adres końca obszaru chronionego SDK (x2kB)
b7:5 – bity zarezerwowane
b4:0 – adres początku obszaru chronionego SDK (x2kB)

Rysunek 11. Rejestr ustawień konfiguracyjnych FLASH_SDKR

Są one uwzględniane w pracy mikrokontrolera dopiero po ponownym włączeniu zasilania układu (tj. resecie typu POR lub BOR). Drugim sposobem aktywacji wprowadzonych zmian w ustawieniach konfiguracyjnych układu jest ustawienie bitu OBL_LAUNCH

eprasa.pl 4202fe84f2

Tabela 3. Ustawienia konfiguracyjne mikrokontrolerów PY32F0

L.p.	Parametr	Adres	Ustawienie fabryczne	Skojarzony rejestr kontrolera Flash
1	Ustawienia użytkownika (User option)	0x1FFF 0E80	0x0455 BEAA	FLASH_OPTR
2	Adres obszaru chronionego SDK (SDK Option)	0x1FFF 0E84	0xFF00 00FF	FLASH_SDKR
3	Blokady sektorów pamięci Flash przed zapisem (WRP address)	0x1FFF 0E8C	0x0000 FFFF	FLASH_WRPR

Rejestr FLASH_WRPR
Adres: 0x4002 202C

b31	b16b15	b0
Res	WRP[15:0]	

b31:16 – bity zarezerwowane
b15:0 – bN: ochrona przed zapisem sektora nr N (N=0..15) (0 – zapis i kasowanie zablokowane, 1 – ochrona wyłączona)

Rysunek 12. Rejestr ustawień konfiguracyjnych FLASH_WRPR

```
\python.exe pyStlink.py -c tw32f003 dump:ob
DEVICE: ST-Link/V2 V2J40S0
SUPPLY: 2.93V
CORE:   CortexM0+
MCU:    TW32F003
FLASH:  64kB
SRAM:   8kB
  OPTR: 0xfeaa
  SDKR: 0x001f
  WRPR: 0xffff
```

Rysunek 13. Odczyt ustawień konfiguracyjnych mikrokontrolera TW32F003 przez program pyStlink

```
# option bytes (name, reg. address)
OPT_BYTES = (
    {'name': 'OPTR', 'reg': Flash.FLASH_OPTR},
    {'name': 'SDKR', 'reg': Flash.FLASH_SDKR},
    {'name': 'WRPR', 'reg': Flash.FLASH_WRPR},
)


def ob_get_all(self):
    self._dbg.debug('PY32F0.ob_get_all')
    opt_data = []
    for opt_byte in self.OPT_BYTES:
        opt_data += [[opt_byte['name']]+[self._stlink.get_debugreg32(opt_byte['reg'])]]
    return opt_data


def ob_write(self, name, data):
    self._dbg.debug('PY32F0.ob_write()')
    for opt_byte in self.OPT_BYTES:
        if opt_byte['name'] == name:
            reg = opt_byte['reg']
            self._dbg.debug('Programming Option Byte %s (addr:0x%08x, data:0x%04x)'
                            %(name, reg, data))
            flash = Flash(self, self._stlink, self._dbg)
            flash.unlock_nvm()
            flash.unlock_ob()
            self._stlink.set_debugreg32(reg, data)
            flash.set_operation(flash.FLASH_CR_EOPIE | flash.FLASH_CR_OPTSTRT)
            self._stlink.set_mem32(flash.FLASH_OPT_WRITE, [0xFF, 0xFF, 0xFF, 0xFF])
            flash.wait_busy(0.005)
            if flash.get_operation_result() == flash.OPR_ERROR:
                raise lib.stlinkex.StlinkException('Writing Option Byte %s failed'
                                                   %(name))
            flash.lock_nvm()
            print('Option Byte %s programmed successfully. Reseting MCU' %(name))
            flash.set_operation(flash.FLASH_CR_OBL_LAUNCH)
            return
    raise lib.stlinkex.StlinkException('Programming Option Byte %s is not implemented '
                                       'for this MCU' %(name))
```

Listing 5. Funkcje odczytu i zapisu ustawień konfiguracyjnych Option Bytes

w rejestrze kontrolnym FLASH_CR. Wymusza to restart mikrokontrolera z załadowaniem nowej konfiguracji.

Rejestry Option Bytes w kontrolerze pamięci Flash są dodatkowo chronione przed modyfikacją. Dlatego też przed ich zapisem konieczne jest najpierw odblokowanie dostępu do modyfikacji rejestrów kontrolera pamięci Flash, jak zostało to zaprezentowane przy zapisie i kasowaniu pamięci, a następnie odblokowanie dostępu do modyfikacji rejestrów Option Bytes przez wpisanie do rejestru FLASH_OPTKEYR sekwencji kodów 0x0819 2A3B i 0x4C5D 6E7F, co w sterowniku realizuje funkcja *unlock_ob()*.

Do programowania ustawień konfiguracyjnych mikrokontrolera lista opcji wywołania programu pyStlink została rozszerzona o dodatkowy parametr:

ob:nazwa_option_byte:wartość

Wynik operacji zmiany ochrony sektorów pamięci Flash przed zapisem pokazuje **rysunek 14**, natomiast w ramce zebrano pełną listę opcji wywołania programu pyStlink, uwzględniającą opisane w artykule modyfikacje (https://tiny.pl/dn39s).

```
\python.exe pyStlink.py -c tw32f003 ob:wrpr:0xfff0
DEVICE: ST-Link/V2 V2J40S0
SUPPLY: 2.93V
CORE:   CortexM0+
MCU:    TW32F003
FLASH:  64kB
SRAM:   8kB
Option Byte WRPR programmed successfully. Reseting MCU
```

Rysunek 14. Włączenie ochrony sektorów S0..S3 przed zapisem w mikrokontrolerze TW32F003 przez program pyStlink

Podsumowanie

Egzotyczny na pierwszy rzut oka mikrokontroler TW32F003 okazał się dość dobrze znanym i udokumentowanym układem typu PY32F003F1P6 firmy Puya Semiconductor. A dzięki programatorowi-debuggerowi STLink, obsługiwanemu przez zmodyfikowany program pyStlink, można przetestować ten interesujący układ (oraz inne mikrokontrolery z rodziny PY32F0) bez konieczności inwestowania w dodatkowe narzędzia.

Aleksander Borysiuk
alex_priv@wp.pl

Źródła:
1. Jay Carlson, „The Cheapest Flash Microcontroller You Can Buy Is Actually an ARM Cortex M0+", https://jaycarlson.net
2. Mikrokontrolery PY32F002/PY32F003/PY32F030: https://www.puyasemi.com/py32_xilie.html#common
3. Mikrokontrolery PY32: https://py32.org
4. Aleksander Borysiuk, „Programowanie mikrokontrolerów HC32F003/HC32F005 za pomocą programatora ST-Link", „Elektronika Praktyczna" 9/2022, https://tiny.pl/dn39b
5. Program pyStlink: https://github.com/pavelrevak/pyStlink

eprasa.pl 4202fe84f2

Kurs Nordic nRF z BT (3)

Przyciski i wielowątkowość

W ostatniej części kursu pokazaliśmy, jak środowisko nRF Connect wspiera użytkownika w analizowaniu działania aplikacji. Tym razem skoncentrujemy się na systemie Zephyr, zintegrowanym z zainstalowanym SDK. Pokażemy, jak utworzyć własny wątek i przypisać mu konkretne zadania do wykonania.

Przyciski

Do tej pory w naszym kursie omówiliśmy interfejsy wyjściowe, takie jak klasyczny LED oraz rozbudowany moduł logowania. Zanim przejdziemy do bardziej zaawansowanych zagadnień, uruchomimy prosty interfejs wejściowy. Będzie to zestaw 4 przycisków zamontowanych na płytce nRF5340 DK, z których każdy jest podłączony do osobnego pinu.

Zaczynamy od utworzenia nowego, pustego projektu i sprawdzenia konfiguracji przycisków w devicetree (**listing 1**).

```
buttons {
   compatible = "gpio-keys";
   button0: button_0 {
       gpios = <&gpio0 23 (GPIO_PULL_UP | GPIO_ACTIVE_LOW)>;
       label = "Push button 1";
       zephyr,code = <INPUT_KEY_0>;
   };
   button1: button_1 {
       gpios = <&gpio0 24 (GPIO_PULL_UP | GPIO_ACTIVE_LOW)>;
       label = "Push button 2";
       zephyr,code = <INPUT_KEY_1>;
   };
   button2: button_2 {
       gpios = <&gpio0 8 (GPIO_PULL_UP | GPIO_ACTIVE_LOW)>;
       label = "Push button 3";
       zephyr,code = <INPUT_KEY_2>;
   };
   button3: button_3 {
       gpios = <&gpio0 9 (GPIO_PULL_UP | GPIO_ACTIVE_LOW)>;
       label = "Push button 4";
       zephyr,code = <INPUT_KEY_3>;
   };
};
```

Listing 1. Konfiguracja przycisków w devicetree

```
#include <zephyr/kernel.h>
#include <zephyr/drivers/gpio.h>
#include <zephyr/logging/log.h>

LOG_MODULE_REGISTER(main, LOG_LEVEL_DBG);

static const struct gpio_dt_spec buttons[] = {
   GPIO_DT_SPEC_GET(DT_NODELABEL(button0), gpios),
   GPIO_DT_SPEC_GET(DT_NODELABEL(button1), gpios),
   GPIO_DT_SPEC_GET(DT_NODELABEL(button2), gpios),
   GPIO_DT_SPEC_GET(DT_NODELABEL(button3), gpios)
};

static struct gpio_callback cb_data;

static void button_callback(const struct device *dev,
   struct gpio_callback *cb, uint32_t pins)
{
   LOG_INF("Pin mask 0x%08x", pins);
}

int main(void) {
   gpio_init_callback(&cb_data, button_callback,
       BIT(buttons[0].pin) | BIT(buttons[1].pin) |
       BIT(buttons[2].pin) | BIT(buttons[3].pin));

   for (int i = 0; i < ARRAY_SIZE(buttons); ++i) {
       gpio_pin_configure_dt(&buttons[i], GPIO_INPUT);
       gpio_pin_interrupt_configure_dt(&buttons[i],
           GPIO_INT_EDGE_FALLING);
       gpio_add_callback(buttons[i].port, &cb_data);
   }

   int counter = 0;
   while (1) {
       LOG_INF("Tick %d", counter++);
       k_msleep(1000);
   }
   return 0;
}
```

Listing 2. Plik main.c

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

Konfiguracja składa się z listy przycisków, umieszczonych w bloku kompatybilnym ze wskazaniem „gpio-keys". Każdy przycisk ma swoje pole gpios, w którym określamy pin oraz dodatkowe parametry.

Następnie wypełniamy plik *main.c* zgodnie z **listingiem 2**. Kod do obsługi przycisków korzysta ze znanego nam już modułu *gpio*. Nowością jest sposób konfiguracji przerwań oraz funkcja callback, która wykonuje się przy każdym naciśnięciu przycisku.

Po skompilowaniu i wgraniu aplikacji na płytkę możemy zaobserwować w konsoli logów, że każde naciśnięcie przycisku generuje log informacyjny z maską bitową identyfikującą przycisk.

Warto wspomnieć, ile pracy wykonuje za nas Zephyr. Dzięki odpowiedniej konfiguracji opadające zbocze na dowolnym z pinów przypisanych do przycisków wywołuje przerwanie, które – obsługiwane przez Zephyra – uruchamia naszą funkcję *button_callback*, podając pin wywołujący przerwanie jako parametr *pins*. Nie musimy konfigurować żadnych rejestrów przerwań. Pamiętajmy jednak, że funkcja *button_callback* jest wywoływana bezpośrednio z przerwania, więc nie powinna zajmować zbyt wiele czasu. W naszym przypadku zlecamy tylko wysłanie loga wątkowi logowania poprzez makro *LOG_INF* i kończymy pracę w kontekście przerwania.

Podczas debugowania programu proponujemy ustawić breakpoint w funkcji *button_callback*. Po zatrzymaniu programu zwróć uwagę na stos wywołań (call stack).

Nowy wątek

Uznaliśmy, że jednym z tematów obowiązkowych do omówienia w naszym kursie są wątki (task, thread). W poprzednim artykule, opisując moduł logowania, wspomnieliśmy o osobnym wątku, który Zephyr automatycznie tworzy do delegowania obsługi wysyłanych wiadomości (poprzez *LOG_INF*, *LOG_ERR* itp.). Teraz pora utworzyć nasz własny wątek. Jego zadaniem będzie obsługa komend do sterowania diodą LED oraz przełączanie go w odpowiednim czasie.

Chcielibyśmy włączać i wyłączać naszego LED-a, a także ustawiać go w tryb migania z określoną częstotliwością. Przygotujmy więc interfejs, którego działanie sprawdzimy za pomocą wcześniej

```
#ifndef LED_CONTROL_H
#define LED_CONTROL_H

typedef struct {
   enum {
       LED_STATE, // param: 0 (off) or 1 (on)
       LED_BLINK  // param: time [ms] for every toggle
   } command;
   unsigned int param;
} led_msg;

void led_send(led_msg msg);

#endif
```

Listing 3. Plik led_control.h

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie **https://ulubionykiosk.pl/media**

eprasa.pl 4202fe84f2

```
#include "led_control.h"
#include <zephyr/kernel.h>
#include <zephyr/drivers/gpio.h>
#include <zephyr/logging/log.h>

LOG_MODULE_REGISTER(led_control, LOG_LEVEL_DBG);

static const struct gpio_dt_spec led =
   GPIO_DT_SPEC_GET(DT_NODELABEL(led0), gpios);

K_MSGQ_DEFINE(led_queue, sizeof(led_msg), 10, 1);

void led_send(led_msg msg) {
   k_msgq_put(&led_queue, &msg, K_NO_WAIT);
}

static void led_worker(void *a, void *b, void *c) {
   gpio_pin_configure_dt(&led, GPIO_OUTPUT_ACTIVE);

   led_msg msg;
   k_timeout_t timeout = K_FOREVER;

   while (1) {
       if (k_msgq_get(&led_queue, &msg, timeout) == 0) {
           if (msg.command == LED_STATE) {
               LOG_INF("LED_STATE %d received", msg.param);
               gpio_pin_set_dt(&led, msg.param);
               timeout = K_FOREVER;
           } else if (msg.command == LED_BLINK) {
               LOG_INF("LED_BLINK %dms received", msg.param);
               timeout = K_MSEC(msg.param);
           }
       } else {
           gpio_pin_toggle_dt(&led);
       }
   }
}

K_THREAD_DEFINE(led_thread, 500, led_worker, NULL, NULL, NULL,
   K_HIGHEST_THREAD_PRIO, 0, 0);
```

Listing 4. Plik led_control.c

przygotowanych przycisków. Przyda nam się on również w późniejszej części kursu. Plik *led_control.h* (**listing 3**) opisuje nasz interfejs.

Plik *led_control.c* (**listing 4**) zawiera implementację obsługi diody LED. Nowy wątek tworzymy statycznie za pomocą makra *K_THREAD_DEFINE*. Warto wspomnieć, że Zephyr przewiduje również opcję tworzenia wątków w trakcie działania programu, funkcją *k_thread_create* (więcej szczegółów w dokumentacji [1]).

A oto, jak działa nasza kontrola LED-a w dużym skrócie i uproszczeniu. Zephyr będzie wywoływał *led_worker* niezależnie od *main* czy innych wątków. Można powiedzieć, że wątki działają równolegle, choć technicznie nie jest to możliwe, gdyż ich kod wykonuje jeden procesor. Wątki nie są „świadome" siebie nawzajem, zatem z punktu widzenia naszego *led_worker* jest on jedyną funkcją w systemie. Tak też należy go interpretować.

Funkcja *led_worker* zostanie wywołana przez Zephyra i na początku włączy LED-a. Następnie, w niekończącej się pętli, będzie oczekiwać na wiadomość podaną poprzez kolejkę *led_queue* w czasie określonym w zmiennej *timeout*. Jeśli wiadomość przyjdzie przed wyznaczonym czasem, *k_msgq_get* zwróci „0" i rozpocznie się dekodowanie odebranej wiadomości. *Timeout* jest inny niż „wieczność" tylko w trybie migania, i po jego upływie po prostu zmienia stan diody.

Wiadomości do kolejki mogą być wysyłane z dowolnego wątku, a także mogą mieć dowolną strukturę. Są bardzo popularnym, jeśli nie najpopularniejszym, sposobem sterowania wątkami i komunikacji między nimi w systemie **Zephyr**.

Skoro mamy już przygotowany wątek do obsługi diody LED, pozostaje nam podłączyć sterowanie nią do przycisków w funkcji *button_callback* (**listing 5**). Oczywiście należy pamiętać o dołączeniu nagłówka *led_control.h*.

Dołączmy również plik źródłowy *led_control.c* do naszego projektu w *CMakeLists.txt* (**listing 6**). Po skompilowaniu i wgraniu

```
static void button_callback(const struct device *dev,
   struct gpio_callback *cb, uint32_t pins)
{
   LOG_INF("Pin mask 0x%08x", pins);

   if (pins & BIT(buttons[0].pin)) {
       led_send((led_msg){ LED_STATE, 0 });
   } else if (pins & BIT(buttons[1].pin)) {
       led_send((led_msg){ LED_STATE, 1 });
   } else if (pins & BIT(buttons[2].pin)) {
       led_send((led_msg){ LED_BLINK, 500 });
   } else if (pins & BIT(buttons[3].pin)) {
       led_send((led_msg){ LED_BLINK, 100 });
   }
}
```

Listing 5. Plik button_callback.c

```
cmake_minimum_required(VERSION 3.20.0)
find_package(Zephyr REQUIRED HINTS $ENV{ZEPHYR_BASE})

project(thread_test)

target_sources(app PRIVATE
   src/main.c
   src/led_control.c
)
```

Listing 6. Plik CMakelist.txt

projektu możemy kontrolować diodę LED za pomocą czterech przycisków znajdujących się na płytce.

Shell

Istnieje jeszcze jedna kwestia, warta dodania do arsenału programisty przy pracy z Zephyrem, a mianowicie shell. To wspaniałe narzędzie do testowania i konfigurowania naszych projektów. Najlepiej zobaczyć go w akcji.

Włączamy shell z obsługą UART w *Kconfig* poprzez dodanie do *prj.conf* dwóch ustawień, jak w **listingu 7**.

Mówimy Zephyrowi, który UART ma być skojarzony z shellem, uzupełniając wpis w *led_pcb.overlay* (**listing 8**).

Nie przejmujmy się na razie tym, że zarówno logi, jak i shell używają tego samego portu. Dodajemy nasze własne komendy powłoki Zephyra na końcu pliku *led_control.c* (jak w **listingu 9**). Będą one wywoływać nasze funkcje sterujące LED-em. Budujemy i testujemy nasz interfejs do LED-a poprzez komendy wpisywane w tym samym

```
CONFIG_SHELL=y
CONFIG_SHELL_BACKEND_SERIAL=y
```

Listing 7. Ustawienia w prj.conf

```
   chosen {
       zephyr,console = &uart0;
       zephyr,shell-uart = &uart0;
   };
```

Listing 8. Wpis w led_pcb.overlay

REKLAMA

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie
https://ulubionykiosk.pl/media

eprasa.pl 4202fe84f2

```
[00:00:09.540,496] <inf> main: Tick 6
[00:00:10.540,537] <inf> main: Tick 7
[00:00:11.540,618] <inf> main: Tick 8
[00:00:12.540,679] <inf> main: Tick 9
[00:00:13.540,740] <inf> main: Tick 10
[00:00:14.540,882] <inf> main: Tick 11
[00:00:15.540,063] <inf> main: Tick 12
[00:00:16.540,924] <inf> main: Tick 13
[00:00:17.540,985] <inf> main: Tick 14
uart:~$ led blink 200
LED blinking period 200 ms:
[00:00:17.919,525] <inf> led_control: LED_BLINK 200ms received
[00:00:18.541,046] <inf> main: Tick 15
[00:00:19.541,107] <inf> main: Tick 16
[00:00:20.541,168] <inf> main: Tick 17
[00:00:21.541,229] <inf> main: Tick 18
[00:00:22.541,290] <inf> main: Tick 19
[00:00:23.541,351] <inf> main: Tick 20
uart:~$
```

Rysunek 1. Logi działającej aplikacji

```
#ifdef CONFIG_SHELL

#include <zephyr/shell/shell.h>
#include <stdlib.h>

static int cmd_led_on(const struct shell *shell, size_t argc, char **argv) {
   led_send((led_msg){ LED_STATE, 1 });
   shell_print(shell, "LED on");
   return 0;
}

static int cmd_led_off(const struct shell *shell, size_t argc, char **argv) {
   led_send((led_msg){ LED_STATE, 0 });
   shell_print(shell, "LED off");
   return 0;
}

static int cmd_led_blink(const struct shell *shell, size_t argc, char **argv) {
   if (argc != 2) {
       shell_print(shell, "led blink <time in ms>");
       return -EINVAL;
   }

   uint32_t period = strtoul(argv[1], NULL, 10);
   led_send((led_msg){ LED_BLINK, period });
   shell_print(shell, "LED blinking period %d ms:",period);
   return 0;
}

SHELL_STATIC_SUBCMD_SET_CREATE(led_menu,
   SHELL_CMD(on, NULL, "LED on", cmd_led_on),
   SHELL_CMD(off, NULL, "LED off", cmd_led_off),
   SHELL_CMD(blink, NULL, "LED blink with period [ms]", cmd_led_blink),
   SHELL_SUBCMD_SET_END
);

SHELL_CMD_REGISTER(led, &led_menu, "LED", NULL);
#endif
```

Listing 9. Komendy shella w led_control.c

terminalu, w którym oglądaliśmy (i nadal oglądamy) logi (**rysunek 1**).

Wiele modułów Zephyra ma wbudowane komendy shellowe. Wystarczy je włączyć. Tak w ramach ćwiczeń pobawmy się shellem modułu *gpio*. Dodajemy ostatni już wpis do *prj.conf*: *CONFIG_GPIO_SHELL=y* – włączy on komendy shella wkompilowane w moduł *gpio*. Po wgraniu projektu będziemy mogli sterować naszym LED-em bezpośrednio, sterując pinem w module *gpio* Zephyra niejako poza „wiedzą" naszej aplikacji.

Komenda wyłączająca LED to: *„gpio set gpio@842500 28 0"*. Dane dotyczące portu i pinu możemy uzyskać, analizując konfigurację *led0* w pliku *led_pcb.dts*.

Podsumowanie

W tej części kursu nauczyliśmy się konfigurować i obsługiwać przyciski, tworzyć wątki oraz zarządzać nimi, a także korzystać z narzędzia shell. Dzięki tym umiejętnościom jesteśmy gotowi do dalszej pracy z systemem Zephyr i realizacji bardziej zaawansowanych zagadnień związanych z Bluetooth.

Krzysztof Kierys
Paweł Jachimowski

Odnośniki w tekście:

1. https://tiny.pl/dn1mm

REKLAMA

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

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

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

KLUB AVT ELEKTRONIKA

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

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

eprasa.pl 4202fe84f2

Środowisko ESP-IDF (3)

Wi-Fi i podstawowe tryby pracy

Wszystkie układy ESP32 są obecnie wyposażone w sprzętowy interfejs Wi-Fi. Pozwala to na budowanie aplikacji wymagających łączności z siecią i przekazywania dużych ilości danych. Środowisko ESP-IDF oczywiście wspiera ten interfejs. Udostępnione biblioteki programistyczne umożliwiają pracę układu w różnych trybach Wi-Fi, a także ułatwiają konfigurację.

Zmiana wersji lub uaktualnienie ESP-IDF

Rodzina układów ESP32 wciąż się rozbudowuje, a za zmianami podąża również środowisko ESP-IDF. W kolejnych wersjach uwzględniana jest obsługa nowszych modeli ESP32, oferujących często całkiem nowe możliwości. Także samo środowisko podlega zmianom – poprzez modernizowanie wersji bibliotek: usuwanie błędów i optymalizację procedur. Nowe aplikacje użytkownika warto zatem pisać, używając najświeższych stabilnych wersji środowiska ESP-IDF, z kolei drobne poprawki starszych aplikacji mogą wymagać tej wersji ESP-IDF, w której zostały napisane. W przypadku pracy z edytorem opartym o Eclipse pobranie nowej wersji środowiska lub przełączenie na inną wersję można wykonać, klikając opcję *Espressif* → *Download and Configure ESP-IDF*. Wyświetli się plansza podobna do tej z **rysunku 1**. Zaznaczając opcję *Use an existing ESP-IDF directory from file system*, możemy wskazać katalog z wcześniej pobranymi wersjami ESP-IDF. Jedną z nich będzie można ustawić jako aktywną i używaną w procesie kompilacji programów użytkownika. Jeżeli opcja pozostanie niezaznaczona, można wybrać wersję środowiska do pobrania z sieci spod adresu https://dl.espressif.com/dl/idf-eclipse-plugin/updates/latest/. Pliki wersji po pobraniu zostaną rozpakowane do wskazanego folderu i mogą być użyte jako aktywna wersja ESP-IDF.

Standard Wi-Fi w modułach rodziny ESP32

Starsze warianty ESP32 mogą pracować jako Wi-Fi (standard 802.11 b/g/n) o następujących parametrach:

- prędkość do 150 Mb/s,
- obsługa wielu połączeń (do 255),
- tryby pracy: stacja, punkt dostępowy, mieszany.

Nowe układy (np. ESP32-S3) są przystosowane do standardu Wi-Fi 6 802.11 b/g/n/ax, który od starszej wersji różni się:

- prędkością do 300 Mb/s,
- standardem szyfrowania WPA3,
- mniejszym zużyciem energii.

Nowy standard Wi-Fi 6 jest kompatybilny ze starszymi wersjami Wi-Fi.

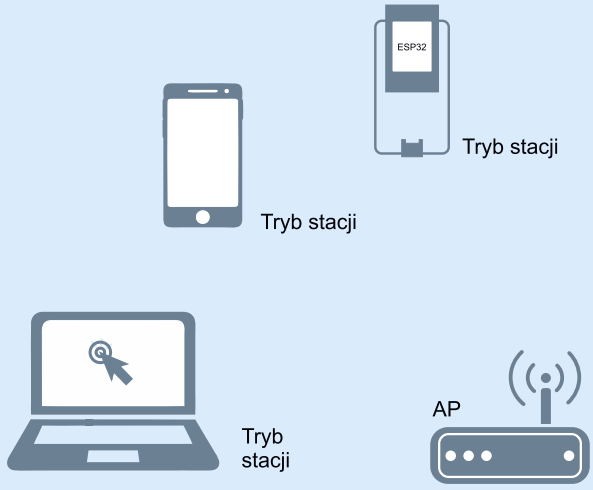
Tryb stacji

Praca w trybie stacji oznacza, że moduł ESP32 może po zalogowaniu łączyć się z innymi uczestnikami lokalnej sieci Wi-Fi. Jest to sytuacja pokazana na **rysunku 2**. Lokalną sieć tworzy w tym przykładzie router oraz dwa inne urządzenia działające w trybie stacji: komputer PC i telefon. Moduł ESP32, po uwierzytelnieniu się właściwym hasłem, otrzymuje swój numer IP przydzielony przez router. Od tego momentu staje się widoczny w ramach sieci lokalnej – może za pośrednictwem routera łączyć się z pozostałymi uczestnikami sieci i wymieniać dane. Prosty program pokazany na kolejnych listingach zademonstruje kroki niezbędne do tego, aby moduł – pracując w trybie stacji – stał się osiągalny w ramach sieci.

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

Na początku, co widać na **listingu 1**, włączamy wszystkie pliki nagłówkowe bibliotek, które będą używane w dalszej części programu. Zamiast „twój ssid" i „twoje hasło" trzeba wpisać

Rysunek 1. Okno wyboru wersji środowiska ESP-IDF



Rysunek 2. Topologia sieci z modułem ESP32 w trybie stacji

```
#include <stdio.h>                          //obsługa komendy printf
#include <string.h>                         //obsługa ciągów znaków
#include "freertos/FreeRTOS.h"              //dla operacji rtos: delay,mutexs,semphrs
#include "esp_system.h"                     //dla funkcji esp_err_t
#include "esp_wifi.h"                       //dla funkcji i operacji Wi-Fi
#include "esp_log.h"                        //dla wyświetlania logów
#include "esp_event.h"                      //dla obsługi zdarzeń Wi-Fi
#include "nvs_flash.h"                      //dla obsługi pamięci nieulotnej
#include "lwip/err.h"                       //obsługa błędów lwip
#include "lwip/sys.h"                       //aplikacje systemowe dla lwip
const char *your_ssid = "twój ssid";
const char *your_pass = "twoje hasło";
int retry_num=0;
```

Listing 1. Pliki nagłówkowe oraz najważniejsze stałe i zmienne używane do obsługi Wi-Fi

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie **https://ulubionykiosk.pl/media**

eprasa.pl 4202fe84f2

```
static void wifi_event_handler ( void *event_handler_arg, esp_event_base_t event_base, int32_t event_id, void *event_data) {
 if (event_id == WIFI_EVENT_STA_START)
 {
  printf ( "ŁĄCZENIE Z WIFI....\n" );
 }
 else if (event_id == WIFI_EVENT_STA_CONNECTED)
 {
  printf ( "Wi-Fi POŁĄCZONE\n" );
 }
 else if (event_id == WIFI_EVENT_STA_DISCONNECTED)
 {
  printf ( "Utracono połączenie Wi-Fi\n" );
  if (retry_num< 5 ){esp_wifi_connect();retry_num++; printf ( "Ponowna próba połączenia...\n" );}
 }
 else if (event_id == IP_EVENT_STA_GOT_IP)
 {
  printf ( "Wi-Fi ma adres IP...\n\n" );
 }
}
```

Listing 2. Procedura obsługi zdarzeń

```
void wifi_connection(){
 esp_netif_init();                                                          //inicjalizacja interfejsu sieciowego
 esp_event_loop_create_default();                                           //włączenie obsługi zdarzeń
 esp_netif_create_default_wifi_sta();                                       //konfiguracja struktury danych dla interfejsu
                                                                            //stacji Wi-Fi
 wifi_init_config_t wifi_initiation = WIFI_INIT_CONFIG_DEFAULT();           //struktura wifi_init_config
                                                                            //z wartościami domyślnymi
 esp_wifi_init(&wifi_initiation);                                           //inicjacja wifi warościami domyślnymi
 esp_event_handler_register(WIFI_EVENT, ESP_EVENT_ANY_ID, wifi_event_handler, NULL);   //tworzenie rejestru obsługi zdarzeń dla Wi-Fi
 esp_event_handler_register(IP_EVENT, IP_EVENT_STA_GOT_IP, wifi_event_handler,NULL);   //tworzenie rejestru obsługi zdarzeń dla
                                                                            //zdarzenia IP
 wifi_config_t wifi_configuration ={                                        //struktura wifi_configuration dla nazwy sieci
                                                                            //i hasła
 .sta= {
  .ssid = "",
  .password= "",
  }
 };
 strcpy((char*)wifi_configuration.sta.ssid,your_ssid);                      //kopiowanie do struktury łańcuchów znaków
 strcpy((char*)wifi_configuration.sta.password,your_pass);
 esp_wifi_set_config(ESP_IF_WIFI_STA, &wifi_configuration);                 //konfiguracja po zdarzeniu //ESP_IF_WIFI_STA
 esp_wifi_start();                                                          //start połączenia
 esp_wifi_set_mode(WIFI_MODE_STA);                                          //wybranie trybu pracy stacja
 esp_wifi_connect();                                                        //łączenie z wykorzystaniem podanych ssid
                                                                            //i pass
 printf( "wifi_init_softap finished. SSID:%s password:%s",your_ssid,your_pass);
}
```

Listing 3. Główna procedura realizująca podłączenie modułu ESP32 do sieci Wi-Fi w trybie stacji

rzeczywistą nazwę sieci – do której moduł ma się podłączyć – oraz hasło.

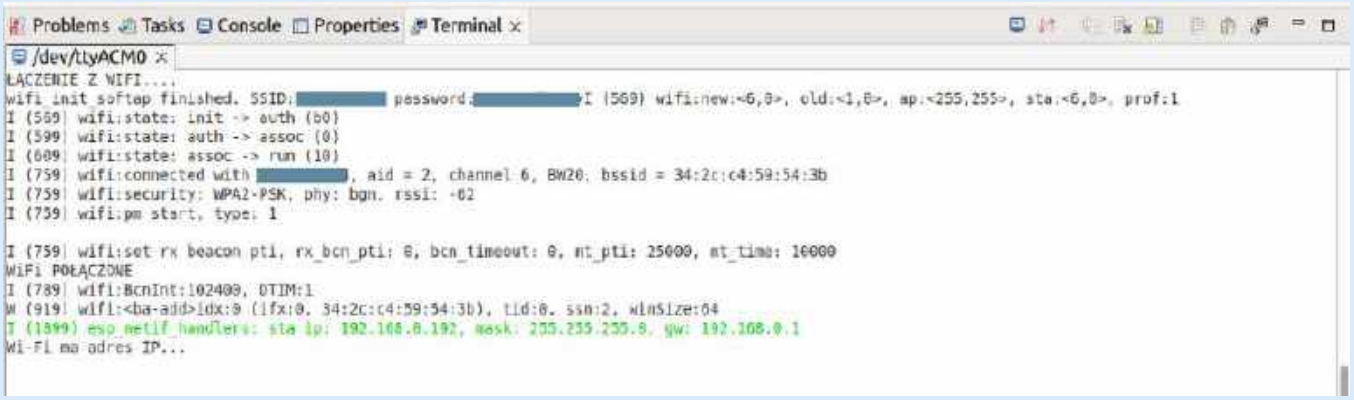
Na **listingu 2** pokazano, jak utworzyć procedurę obsługi zdarzeń (są to różne sytuacje, w tym przypadku występujące podczas włączania modułu ESP32 do sieci). Przykładem zdarzenia może być start trybu stacji, podłączenie do sieci, uzyskanie numeru IP czy też odłączenie od sieci. Gdy któraś z wymienionych sytuacji wystąpi, wywołana zostanie funkcja obsługi z przekazanym w parametrze event_id identyfikatorem zdarzenia. Dzięki identyfikatorowi wiadomo, które zdarzenie wymaga obsługi w danym momencie. W analizowanym przykładzie najczęściej

```
void app_main(void){
 nvs_flash_init();        //inicjacja pamięci do przechowywania konfiguracji Wi-Fi
 wifi_connection();
}
```

Listing 4. Funkcja main programu

```
#include <string.h>                //obsługa ciągów znaków
#include "freertos/FreeRTOS.h"     //dla operacji rtos: delay,mutexs,semphrs
#include "freertos/task.h"         //dla operacji rtos: task
#include "esp_system.h"            //dla funkcji esp_err_t
#include "esp_wifi.h"              //dla funkcji i operacji Wi-Fi
#include "esp_event.h"             //dla obsługi zdarzeń Wi-Fi
#include "esp_log.h"               //dla wyświetlania logów
#include "nvs_flash.h"             //dla obsługi pamięci nieulotnej

#include "lwip/err.h"              //lwip kody błędów
#include "lwip/sys.h"              //lwip system
```

Listing 5. Pliki nagłówkowe używane w drugim programie testowym



Rysunek 3. Komunikaty wyświetlane w terminalu szeregowym podczas nawiązywania połączenia z siecią przez moduł ESP32

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie **https://ulubionykiosk.pl/media**

eprasa.pl 4202fe84f2

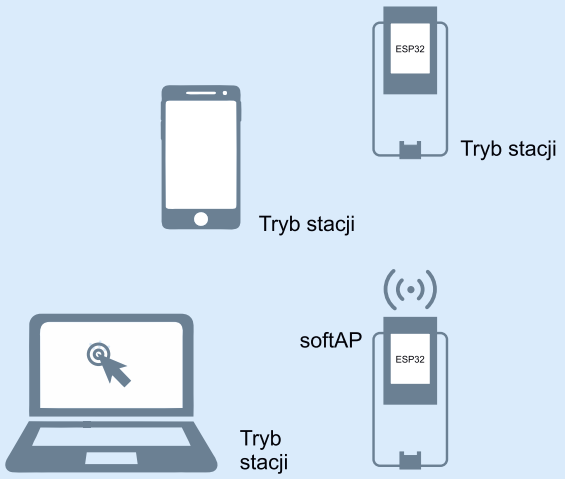
```
#define EXAMPLE_ESP_WIFI_SSID      "twoja_nazwa_sieci"
#define EXAMPLE_ESP_WIFI_PASS      "hasło_do_twojej_sieci"
#define EXAMPLE_ESP_WIFI_CHANNEL   1                          //kanał radiowy twojej sieci
#define EXAMPLE_MAX_STA_CONN       4                          //maksymalna ilość stacji w twojej sieci Wi-Fi

static const char *TAG = "wifi softAP";  //znacznik logu
```

Listing 6. Najważniejsze stałe i zmienne używane do obsługi Wi-Fi w trybie softAP

```
static void wifi_event_handler(void* arg, esp_event_base_t event_base,
                               int32_t event_id, void* event_data)
{
    if (event_id == WIFI_EVENT_AP_STACONNECTED) {
        wifi_event_ap_staconnected_t* event = (wifi_event_ap_staconnected_t*) event_data;
        ESP_LOGI(TAG, "station "MACSTR" join, AID=%d",
                 MAC2STR(event->mac), event->aid);
    } else if (event_id == WIFI_EVENT_AP_STADISCONNECTED) {
        wifi_event_ap_stadisconnected_t* event = (wifi_event_ap_stadisconnected_t*) event_data;
        ESP_LOGI(TAG, "station "MACSTR" leave, AID=%d",
                 MAC2STR(event->mac), event->aid);
    }
}
```

Listing 7. Przykładowa procedura obsługi zdarzeń Wi-Fi w trybie softAP

będzie to tylko wysłanie komunikatu, który wyświetli się na monitorze portu szeregowego przez polecenie `printf`. W przypadku zdarzenia `WIFI_EVENT_STA_DISCONNECTED` będzie to podejmowana 5 razy próba nawiązania połączenia.



Rysunek 4. Topologia sieci z modułem ESP32 trybie softAP

Listing 3 zawiera główną procedurę realizującą podłączenie modułu ESP32 – pracującego w trybie stacji – do wybranej sieci Wi-Fi. Poszczególne kroki opatrzono zwięzłymi opisami w komentarzach. Najpierw przeprowadzane są niezbędne konfiguracje, włączane są obsługi zdarzeń, tworzona jest struktura zadeklarowanych wcześniej stałych zawierających nazwę sieci i hasło dostępu. Na końcu funkcji `wifi_connection()` wywoływane są procedury podłączeniowe. **Listing 4** pokazuje funkcję `main()` łączącą wszystko razem.

Po kompilacji programu i wgraniu do pamięci ESP32 układ zaczyna szukać sieci o podanej w stałej `your_ssid` nazwie i próbuje się do niej zalogować. W przypadku sukcesu nadawany jest modułowi numer IP. W monitorze można podejrzeć komunikaty generowane podczas tego procesu, podobne do tych widocznych na **rysunku 3**. Z komunikatu logu wynika, że router dodał moduł ESP32 pracujący w trybie stacji do sieci – i nadał mu numer IP 192.168.0.192.

Przykładowa procedura oparta została na materiale dostępnym w [1].

Tryb punktu dostępowego

Tryb punktu dostępowego w dokumentacji nosi nazwę soft Access Point (softAP). Pracując w tym trybie, ESP32 tworzy lokalną sieć Wi-Fi, do której mogą się podłączać urządzenia będące stacjami

```
void wifi_init_softap(void)
{
    ESP_ERROR_CHECK(esp_netif_init());                  //inicjalizacja interfejsu sieciowego
    ESP_ERROR_CHECK(esp_event_loop_create_default());   //włączenie obsługi zdarzeń
    esp_netif_create_default_wifi_ap();                 //konfiguracja struktury danych dla interfejsu AP Wi-Fi

    wifi_init_config_t cfg = WIFI_INIT_CONFIG_DEFAULT(); //struktura cfg z wartościami domyślnymi
    ESP_ERROR_CHECK(esp_wifi_init(&cfg));               //inicjacja wifi wartościami domyślnymi
    //tworzenie rejestru obsługi zdarzeń
    ESP_ERROR_CHECK(esp_event_handler_instance_register(WIFI_EVENT,
                                                        ESP_EVENT_ANY_ID,
                                                        &wifi_event_handler,
                                                        NULL,
                                                        NULL));

    wifi_config_t wifi_config = {
        .ap = {
            .ssid = EXAMPLE_ESP_WIFI_SSID,
            .ssid_len = strlen(EXAMPLE_ESP_WIFI_SSID),
            .channel = EXAMPLE_ESP_WIFI_CHANNEL,
            .password = EXAMPLE_ESP_WIFI_PASS,
            .max_connection = EXAMPLE_MAX_STA_CONN,
            .authmode = WIFI_AUTH_WPA_WPA2_PSK
        },
    };
    if (strlen(EXAMPLE_ESP_WIFI_PASS) == 0) {
        wifi_config.ap.authmode = WIFI_AUTH_OPEN;
    }
    //inicjacja trybu Wi-Fi softAP
    ESP_ERROR_CHECK(esp_wifi_set_mode(WIFI_MODE_AP));
    ESP_ERROR_CHECK(esp_wifi_set_config(WIFI_IF_AP, &wifi_config));
    ESP_ERROR_CHECK(esp_wifi_start());

    ESP_LOGI(TAG, "wifi_init_softap finished. SSID:%s password:%s channel:%d",
             EXAMPLE_ESP_WIFI_SSID, EXAMPLE_ESP_WIFI_PASS, EXAMPLE_ESP_WIFI_CHANNEL);
}
```

Listing 8. Ciało funkcji inicjalizującej moduł Wi-Fi do pracy w trybie softAP

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie https://ulubionykiosk.pl/media

eprasa.pl 4202fe84f2

```
Problems  Tasks  Console  Properties  Terminal ×
/dev/ttyACM0 ×
I (374) wifi softAP: wifi_init_softap finished. SSID:ESP32-softAP password:1234567890ab channel:1
I (44784) wifi:new:<1,1>, old:<1,1>, ap:<1,1>, sta:<255,255>, prof:1
I (44784) wifi:station: 10:0b:a9:47:2e:ac join, AID=1, bgn, 40U
I (44814) wifi softAP: station 10:0b:a9:47:2e:ac join, AID=1
I (44874) esp_netif_lwip: DHCP server assigned IP to a station, IP is: 192.168.4.2
W (46044) wifi:<ba-add>idx:2 (ifx:1, 10:0b:a9:47:2e:ac), tid:0, ssn:84, winSize:64
W (181644) wifi:<ba-add>idx:3 (ifx:1, 10:0b:a9:47:2e:ac), tid:5, ssn:33, winSize:64
```

Rysunek 5. Komunikaty wyświetlane w terminalu szeregowym podczas nawiązywania dołączania nowych urządzeń do sieci utworzonej przez moduł ESP32

Wi-Fi. Utworzona sieć umożliwia podłączonym stacjom komunikację i wymianę danych pomiędzy sobą, przy skorzystaniu z pośrednictwa softAP. Taką konfigurację zaprezentowano na **rysunku 4**. Trzy stacje: komputer PC, telefon oraz moduł ESP32, podłączone są do sieci utworzonej przez inny moduł ESP32, pracujący jako softAP. Na kolejnych listingach pokazano składowe programu demonstrującego, jak za pomocą modułu ESP32 utworzyć w pełni funkcjonalny softAP.

```
void app_main(void)
{
    //Initialize NVS
    esp_err_t ret = nvs_flash_init();
    if (ret == ESP_ERR_NVS_NO_FREE_PAGES || ret == ESP_ERR_NVS_NEW_VERSION_FOUND) {
      ESP_ERROR_CHECK(nvs_flash_erase());
      ret = nvs_flash_init();
    }
    ESP_ERROR_CHECK(ret);

    ESP_LOGI(TAG, "ESP_WIFI_MODE_AP");
    wifi_init_softap();
}
```

Listing 9. Główna funkcja programu obsługującego tryb softAP

Na **listingu 5** wyliczone zostały wszystkie potrzebne pliki nagłówkowe. **Listing 6** zawiera deklaracje kluczowych ustawień tworzonej sieci Wi-Fi. Do `EXAMPLE_ESP_WIFI_SSID` przypisujemy nazwę sieci, np. „esp32_moja_siec", a do `EXAMPLE_ESP_WIFI_PASS` – hasło logowania.

Na **listingu 7** można obejrzeć przykładową procdurę obsługi zdarzeń Wi-Fi. Obsługiwane są dwa zdarzenia: `WIFI_EVENT_AP_STACONNECTED` (zdarzenie podłączenia do sieci nowej stacji), a także `WIFI_EVENT_AP_STADISCONNECTED` (zdarzenie odłączenia stacji od sieci). Na **listingu 8** pokazana została funkcja `wifi_init_softap()`, inicjalizująca Wi-Fi ESP32 w trybie softAP. Ostatni **listing 9** zawiera główną funkcję `main()` programu.

Na **rysunku 5** pokazano zrzut ekranu z komunikatami wysyłanymi do terminalu przez moduł ESP32 z uruchomionym przykładowym oprogramowaniem. Moduł działa w trybie softAP i tworzy sieć o nazwie „ESP32-softAP" zabezpieczoną hasłem „1234567890ab". Do sieci podłączona jest jedna stacja, której przydzielono numer IP: 192.168.4.2.

Przykładowa procedura oparta została na materiale dostępnym w [2].

Ryszard Szymaniak, EP

[1] https://tiny.pl/d4vct
[2] https://tiny.pl/d4vc7

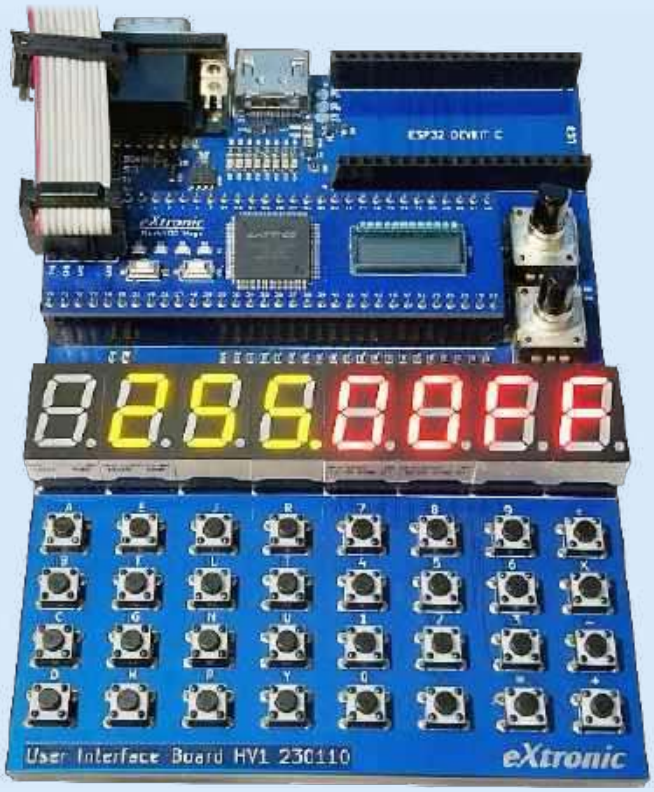
REKLAMA

eprasa.pl 4202fe84f2

Kurs FPGA Lattice (22)

Sekwencyjny algorytm Double Dabble

Wyświetlacz cyfrowy stosowaliśmy już nieraz, ale zawsze pokazywaliśmy na nim wartości w postaci szesnastkowej. Wygodniej byłoby widzieć liczby w formacie dziesiętnym. W zwykłych językach programowania mamy gotowe funkcje, które wyświetlają liczby w różnych formatach, ale w FPGA konwersję musimy niestety przeprowadzić samodzielnie.



Najwygodniejszym sposobem na przekonwertowanie liczby zapisanej binarnie na format dziesiętny jest użycie kodu BCD (binary coded decimal). Każdą cyfrę dziesiętną zapisujemy jako 4-bitową liczbę binarną. Cztery bity dają możliwość utworzenia 16 kombinacji, ale w kodzie BCD stosujemy tylko kombinacje od 0 do 9, a pozostałe są nieużywane. Zobacz **rysunek 1**.

Przykładowo: liczba dziesiętna 1234 zapisana w kodzie BCD będzie mieć 16 bitów i przyjmie postać 0001_0010_0011_0100. W kodzie binarnym byłoby to 100_1101_0010, czyli 11 bitów. Widzimy więc, że kod BCD nie jest optymalny, jeśli chodzi o zużycie zasobów, jednak pozwala w dość łatwy sposób operować na liczbach dziesiętnych.

Jak działa algorytm Double Dabble?

Najprościej będzie to wytłumaczyć na przykładzie. Zobacz **rysunek 2**. Chcemy przekształcić 8-bitową liczbę 113 na kod BCD. Jej binarna reprezentacja to 0b01110001. Umieszczamy ją w tabeli w kolumnie „Binarnie" i jest to stan początkowy algorytmu. Liczba ta jest 3-cyfrowa, zatem musimy przygotować trzy kolumny w tabeli – dla cyfr setek, dziesiątek i jedności – a każda z nich będzie miała 4 bity. Z tych kolumn odczytamy wynik w kodzie BCD.

W celu zwiększenia czytelności, w tabeli pominąłem wszystkie bity, które są nieistotne. We wszystkich pustych komórkach należy wpisać zero.

Algorytm Double Dabble polega na wykonywaniu w pętli dwóch operacji. Pierwszą z nich jest przesunięcie wszystkich rejestrów o jeden bit w lewo. Takie przesuwanie liczb binarnych jest równoznaczne z pomnożeniem przez 2 – stąd angielska nazwa *double*. Następnie sprawdzamy każdą z cyfr. Jeżeli jest większa lub równa 5, to dodajemy do niej 3 – jest to faza *dabble*. Przesunięć musi być tyle, ile bitów ma liczba na wejściu.

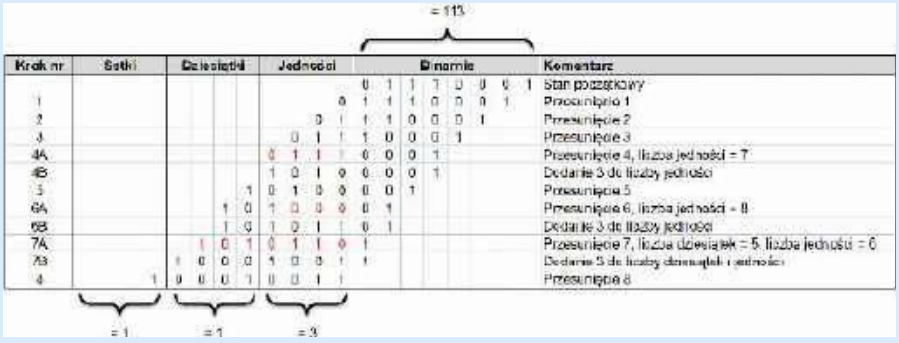
Zobaczmy, jak to działa:

- W kroku 1 przesuwamy w lewo liczbę, którą mieliśmy na wejściu. Cyfry setek, dziesiątek i jedności wciąż są zerami.
- Przechodzimy do kroku 2. Ponownie przesuwamy wszystkie rejestry. Cyfra jedności stała się jedynką. Jest to mniej niż 5, więc idziemy dalej.
- Następnie, w kroku 3, po raz kolejny przesuwamy wszystko w lewo. Cyfra jedności to 0b0011, czyli dziesiętnie 3, a więc wciąż mniej niż 5.
- W kroku 4A ponownie przesuwamy rejestry. Okazuje się, że po przesunięciu cyfra jedności jest równa 0b0111 (zaznaczone

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

Cyfra	Kod BCD			
	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
niezdefiniowane	1	0	1	0
niezdefiniowane	1	0	1	1
niezdefiniowane	1	1	0	0
niezdefiniowane	1	1	0	1
niezdefiniowane	1	1	1	0
niezdefiniowane	1	1	1	1

Rysunek 1. Kod BCD



Rysunek 2. Opis algorytmu Double Dabble krok po kroku

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie **https://ulubionykiosk.pl/media**

eprasa.pl 4202fe84f2

na czerwono), czyli dziesiętnie 7. Zatem w kroku 4B musimy dodać do niej 3, co w rezultacie daje 10, czyli binarnie 0b1010.

- W kroku 5 ponownie przesuwamy. Wszystkie cyfry są mniejsze od 5.
- Podczas kroku 6A przesuwamy i sprawdzamy. Okazuje się, że cyfra jedności to 0b1000, czyli dziesiętnie 8. Zatem w kroku 6B dodajemy 3 i dostajemy 11, czyli binarnie 0b1011.
- Następnie znów przesuwamy wszystkie rejestry w kroku 7A. Sprawdzamy wszystkie cyfry. Okazuje się, że jednocześnie cyfra dziesiątek i jedności są większe lub równe 5. Zatem w kroku 7B do obu tych cyfr dodajemy 3. Dzieje się to w tej samej chwili.
- Krok 8 to ósme i ostatnie przesunięcie, ponieważ liczba na wejściu była 8-bitowa. Algorytm kończy pracę i możemy odczytać wynik. Cyfra setek w kodzie BCD to 0b0001, dziesiątek 0b0001, a jedności 0b0011, co daje nam liczbę 113 zapisaną w BCD.

Dochodzimy do wniosku, że dwa elementy sprzętowe, które potrzebne są do realizacji algorytmu Double Dabble w FPGA, to rejestr przesuwający, sumatory dla każdej cyfry i trochę prostych elementów logicznych, sklejających to wszystko w jedną całość (tak zwane *glue logic*).

Moduł Double Dabble w wersji sekwencyjnej

Moduł realizujący algorytm Double Dabble w wersji sekwencyjnej zaprezentowano na **listingu 1**. Zacznijmy od omówienia parametrów, wejść i wyjść tego modułu:

- **INPUT_BITS** – ile bitów ma zmienna wejściowa,
- **OUTPUT_DIGITS** – ile cyfr ma mieć wynik,
- **OUTPUT_BITS** – ile bitów ma mieć wyjście; ten parametr ustawia się automatycznie w wyniku pomnożenia parametru OUTPUT_DIGITS przez 4.

Moduł wyposażony jest w następujące porty wejścia i wyjścia (**Clock** i **Reset** nie omawiamy, bo działają tak, jak we wszystkich dotychczas omawianych modułach):

- **Binary_i** – wejście danych w formacie binarnym,
- **BCD_o** – wyjście danych w formacie BCD,
- **Start_i** – wejście informujące moduł o żądaniu rozpoczęcia pracy; aby uruchomić moduł, należy ustawić to wejście w stan wysoki na jeden takt zegarowy,
- **Busy_o** – stan wysoki na tym wyjściu informuje, że moduł jest w trakcie pracy,
- **Done_o** – wyjście informujące (poprzez ustawienie stanu wysokiego na jeden takt zegarowy), że moduł zakończył pracę, a wynik jest dostępny do odczytu na wyjściu BCD_o.

Ogólnie rzecz biorąc, moduł musi zawierać maszynę stanów, która będzie naprzemiennie wykonywać operacje double i dabble. Muszą one zostać wykonane tyle razy, ile jest bitów w zmiennej wejściowej, zatem potrzebujemy także licznika.

Przejdźmy do deklaracji zmiennych, używanych w module. W linii 1 tworzymy iterator pętli for (zwyczajowo nazywany **i**) będący zmienną typu integer. Następnie, w linii 2, mamy rejestr **Binary**, którego celem jest wykonanie kopii wejścia **Binary_i**. Taka potrzeba wiąże z się faktem, że stan wejścia może zmienić się podczas pracy, co skutkowałoby nieprawidłowym wynikiem. Aby temu zapobiec, musimy wykonać kopię danych wejściowych w dedykowanym rejestrze. Potrzebujemy jeszcze jednego rejestru, o nazwie **BCD** (linia 3). Posłuży on jako rejestr roboczy, w którym będziemy przesuwać bity, a kiedy moduł

```
// Plik double_dabble.v

`default_nettype none

module DoubleDabble #(
    parameter INPUT_BITS    = 16,
    parameter OUTPUT_DIGITS = 5,
    parameter OUTPUT_BITS   = OUTPUT_DIGITS * 4
)(
    input wire Clock,
    input wire Reset,
    input wire Start_i,
    output reg Busy_o,
    output reg Done_o,
    input wire [ INPUT_BITS-1:0] Binary_i,
    output reg [OUTPUT_BITS-1:0] BCD_o
);

    // Zmienne
    integer i;                                                          // 1
    reg [ INPUT_BITS-1:0] Binary;                                       // 2
    reg [OUTPUT_BITS-1:0] BCD;                                          // 3

    // Maszyna stanów
    reg State;                                                          // 4
    localparam DOUBLE = 1'b0;
    localparam DABBLE = 1'b1;

    // Licznik bitów
    localparam MAX_VALUE = INPUT_BITS - 1;                              // 5
    localparam WIDTH = $clog2(MAX_VALUE + 1);                           // 6
    reg [WIDTH-1:0] Counter;                                            // 7

    // Blok sekwencyjny
    always @(posedge Clock, negedge Reset) begin                        // 8
        if(!Reset) begin                                                // 9
            Counter <= 0;
            Busy_o  <= 0;
            Done_o  <= 0;
            Binary  <= 0;
            BCD     <= 0;
            BCD_o   <= 0;
            State   <= DOUBLE;
        end

        // Jeżeli jest żądanie rozpoczęcia pracy
        else if(Start_i) begin                                          // 10
            Busy_o  <= 1'b1;
            Done_o  <= 1'b0;
            Binary  <= Binary_i;
            Counter <= MAX_VALUE;
            BCD     <= 0;
            State   <= DOUBLE;
        end

        // Jeżeli proces jest w trakcie
        else if(Busy_o) begin                                           // 11

            // Faza Double
            if(State == DOUBLE) begin                                   // 12
                BCD    <= {BCD[OUTPUT_BITS-2:0], Binary[INPUT_BITS-1]}
                Binary <= {Binary[INPUT_BITS-2:0], 1'b0};
                State  <= DABBLE;                                       // 13
            end

            // Faza Dabble
            else begin                                                  // 14

                // Sprawdź każdą cyfrę
                // Jeżeli cyfra >= 5 to dodaj do niej 3
                for(i=3; i<OUTPUT_BITS; i=i+4) begin                    // 15
                    if(BCD[i-:4] >= 4'd5)                               // 16
                        BCD[i-:4] <= BCD[i-:4] + 4'd3;                  // 17
                end

                // Jeżeli trwa odliczanie bitów
                if(Counter) begin                                       // 18
                    Counter <= Counter - 1'b1;
                end

                // Jeżeli odliczanie zakończone
                else begin                                              // 19
                    Busy_o <= 0;
                    Done_o <= 1'b1;
                    BCD_o  <= BCD;
                end

                // Zmiana stanu
                State <= DOUBLE;                                        // 20
            end
        end

        // Kasowanie flagi Done_o
        else if(Done_o) begin                                           // 21
            Done_o <= 1'b0;
        end
    end

endmodule

`default_nettype wire
```

Listing 1. Kod pliku double_dabble.v

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie https://ulubionykiosk.pl/media

eprasa.pl 4202fe84f2

zakończy pracę, wówczas jego zawartość zostanie skopiowana do **BCD_o**.

Kluczową zmienną pozostaje rejestr maszyny stanów **State**, utworzony w linii 4. Maszyna ma tylko dwa stany, więc jest to rejestr 1-bitowy. Stany nazywamy za pomocą parametrów lokalnych **DOUBLE** i **DABBLE** w dwóch kolejnych liniach.

Następnie trzeba przygotować licznik, który będzie odliczał przesunięcia bitów. Musimy uwzględnić tyle przesunięć, ile jest bitów w zmiennej wejściowej (co określone zostało parametrem **INPUT_BITS).** Wygodnie będzie liczyć od maksymalnej wartości do zera – takie podejście jest optymalne z uwagi na wykorzystanie zasobów. Aby sprawdzić, czy stan licznika jest większy od zera, wystarczy nam bramka OR, sprawdzająca stan wszystkich jego bitów. Wyjście takiej bramki znajduje się w stanie niskim tylko wtedy, kiedy wszystkie jej wejścia również są w stanie niskim.

Jako pomoc przyda się nam zmienna przechowująca maksymalną wartość licznika. Tworzymy zatem parametr lokalny **MAX_VALUE**, do którego wpisujemy liczbę bitów wejścia pomniejszoną o jeden – jeżeli wejście jest 8-bitowe, to licznik będzie liczył od 7 do 0 (linia 5). Następnie musimy określić, ilu bitów potrzebuje licznik, aby „pomieścić" wartość maksymalną. W tym celu posługujemy się funkcją $clog2(*x*), która oblicza logarytm naturalny przy podstawie 2 z *x* – tą metodą tworzymy parametr lokalny **WIDTH** (linia 6). Jeżeli nie wiesz, dlaczego argument funkcji został powiększony o 1, zajrzyj do 6 odcinka kursu (EP 04/2023), w którym funkcja $clog2() została szczegółowo omówiona na przykładach. Finalnie tworzymy właściwy licznik **Counter** o odpowiedniej liczbie bitów w linii 7.

Przechodzimy wreszcie do sekwencyjnego bloku always w linii 8, reagującego – jak zwykle – na rosnące zbocze sygnału zegarowego lub opadające zbocze sygnału resetującego. Kiedy **Reset** jest w stanie niskim, wówczas zerujemy wszystkie zmienne typu reg (linia 9).

Blok always wykonuje różne operacje w zależności od tego, czy w stanie wysokim znajdują się sygnały **Start_i**, **Busy_o** lub **Done_o**. Przejdźmy do linii 10 – sprawdzamy w niej stan wejścia **Start_i**, które jest żądaniem rozpoczęcia pracy. W takiej sytuacji ustawiamy wszystkie zmienne, by w kolejnym takcie wykonywać właściwą pracę modułu, a w szczególności kopiujemy wartość z wejścia **Binary_i** do rejestru **Binary**, ustawiamy zmienną Busy_o w stan wysoki, wartość licznika na maksymalną, a rejestr stanu maszyny State na DOUBLE, ponieważ ta operacja ma się wykonać w następnym takcie zegara.

Kiedy moduł jest w trakcie wykonywania pracy, tzn. kiedy zmienna **Busy_o** znajduje się w stanie wysokim (linia 11), wykonujemy różne działania w zależności od stanu zmiennej

```
// Plik double_dabble_tb.v

`timescale 1ns/1ns
`default_nettype none

module DoubleDabble_tb();

    // Konfiguracja
    parameter INPUT_BITS    = 8;                                          // 1
    parameter OUTPUT_DIGITS = 3;                                          // 2
    parameter OUTPUT_BITS   = OUTPUT_DIGITS * 4;                          // 3

    // Generator sygnału zegarowego
    parameter CLOCK_HZ            = 1_000_000;
    parameter real HALF_PERIOD_NS = 1_000_000_000.0 / (2 * CLOCK_HZ);

    reg Clock = 1'b1;
    always begin
        #HALF_PERIOD_NS;
        Clock = !Clock;
    end

    // Zmienne
    reg Reset = 0;
    reg Start = 0;                                                        // 4
    wire Done;                                                            // 5
    reg  [ INPUT_BITS-1:0] Binary = {INPUT_BITS{1'bX}};                   // 6
    wire [OUTPUT_BITS-1:0] BCD;                                           // 7

    integer MaxInput = 2**INPUT_BITS - 1;                                 // 8
    integer i;                                                            // 9

    // Eksport wyników symulacji
    initial begin
        $dumpfile("double_dabble.vcd");
        $dumpvars(0, DoubleDabble_tb);
    end

    // Instancja testowanego modułu
    DoubleDabble #(                                                       // 10
        .INPUT_BITS(INPUT_BITS),
        .OUTPUT_DIGITS(OUTPUT_DIGITS),
        .OUTPUT_BITS(OUTPUT_BITS)
    ) DUT(
        .Clock(Clock),
        .Reset(Reset),
        .Start_i(Start),
        .Busy_o(),
        .Done_o(Done),
        .Binary_i(Binary),
        .BCD_o(BCD)
    );

    // Weryfikacja wyników symulacji
    integer PassCounter = 0;                                              // 11
    integer FailCounter = 0;                                              // 12

    task Verify(input [INPUT_BITS-1:0] Binary, input [OUTPUT_BITS-1:0] BCD);   // 13
        integer Digit;                                                    // 14

        begin: VerifyBlock                                                // 15
            reg [OUTPUT_BITS-1:0] Result;                                 // 16
            Result = 0;                                                   // 17

            // Konwersja BCD na format binarny
            for(Digit=0; Digit<OUTPUT_DIGITS; Digit=Digit+1) begin       // 18
                Result = Result + BCD[Digit*4+3-:4] * 10**Digit;
            end

            // Porównanie wyników
            if(Result === Binary)                                         // 19
                PassCounter = PassCounter + 1;
            else begin
                FailCounter = FailCounter + 1;
                $display("Result: %h, Binary %h, BCD: %h", Result, Binary, BCD); // 20
                // lub $fatal(0, ...) aby zakończyć natychmiast
            end
        end
    endtask

    // Sekwencja testowa
    initial begin
        $timeformat(-9, 3, "ns", 10);
        $display("===== START =====");
        $display("INPUT_BITS:    %9d", INPUT_BITS);
        $display("OUTPUT_BITS:   %9d", DUT.OUTPUT_BITS);
        $display("OUTPUT_DIGITS: %9d", OUTPUT_DIGITS);
        $display("MaxInput:      %9d", MaxInput);
        $display("Counter WIDTH: %9d", DUT.WIDTH);

        @(posedge Clock);
        Reset = 1'b1;

        // Testuj wszystkie wartości od zera do maksimum
        for(i=0; i<=MaxInput; i=i+1) begin                                // 21
            @(posedge Clock);                                             // 22
            Binary <= i;
            Start  <= 1'b1;

            @(posedge Clock);                                             // 23
            Binary <= {INPUT_BITS{1'bX}};
            Start  <= 1'b0;

            @(posedge Done);                                              // 24
            Verify(i, BCD);
        end
```

Listing 2. Kod pliku double_dabble_tb.v

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie
https://ulubionykiosk.pl/media

eprasa.pl 4202fe84f2

State (linia 12). Kiedy jest ona równa **DOUBLE**, czyli 1'b0, przesuwamy bity w rejestrach **BCD** i **Binary** oraz ustawiamy State na **DABBLE** (linia 13).

W stanie **DABBLE** (linia 14) musimy sprawdzić, czy któraś z cyfr BCD jest większa lub równa 5 – i jeżeli tak, to dodajemy do niej 3. Najprościej byłoby wykonać tę procedurę w poniższy sposób:

```
// Cyfra jedności
if(BCD[3:0] >= 4'd5)
    BCD[3:0] <= BCD[3:0] + 4'd3;

// Cyfra dziesiątek
if(BCD[7:4] >= 4'd5)
    BCD[7:4] <= BCD[7:4] + 4'd3;

// Cyfra setek
if(BCD[11:8] >= 4'd5)
    BCD[11:8] <= BCD[11:8] + 4'd3;
```

...i tak dalej, dla każdej cyfry. Moduł jest napisany w taki sposób, by umożliwić jego parametryzację i obsługę dowolnie długich liczb. Z tego powodu musimy przejść na nieco wyższy poziom abstrakcji i sprawdzać każdą cyfrę w pętli for.

Rozwiązanie to pokazano w linii 15. Iteratorem pętli for jest zmienna **i** typu integer, którą początkowo ustawiamy na 3 i w każdym obiegu pętli zwiększamy o 4. W ten sposób dostaniemy liczby 3, 7, 11, itd., które są indeksami najstarszych bitów w każdej z liczb BCD.

Aby wybrać poszczególne bity ze zmiennej wielobitowej, najczęściej stosujemy operator [MSB:LSB] – podajemy w nim indeks najstarszego oraz najmłodszego bitu, które chcemy odczytać lub zapisać. W tym przypadku musimy posłużyć się operatorem [MSB-:RANGE] i wskazać w nim: indeks najstarszego bitu oraz ile bitów w dół nas interesuje. Wszystkie cyfry BCD są 4-bitowe, zatem posłużymy się konstrukcją [i-:4], gdzie **i** jest iteratorem pętli for opisanym wyżej. W ten sposób sprawdzamy, czy każda z cyfr BCD jest większa lub równa 5 (linia 16) i jeżeli tak, to tę liczbę inkrementujemy o 3 (linia 17).

Jednocześnie, w tym samym takcie zegarowym, wykonujemy kilka innych operacji. Sprawdzamy, czy stan licznika **Counter** jest większy od zera (linia 18) – jeżeli tak, to zmniejszamy go o 1, a jeżeli nie – to znaczy, że przetworzyliśmy już wszystkie bity. Zatem zmienną **Busy_o** zerujemy, **Done_o** ustawiamy w stan wysoki i kopiujemy stan rejestru roboczego **BCD** na wyjście **BCD_o**.

Niezależnie od wszystkiego zmieniamy stan rejestru maszyny stanów na **DOUBLE** (linia 20). Gdy algorytm zakończy pracę, działanie takie właściwie nie ma sensu, ale stwierdziłem, że również w niczym nie przeszkadza, a znacznie ułatwiło napisanie kodu.

Przechodzimy do ostatniego warunku w bloku always w linii 21. Jeżeli **Done_o** jest w stanie wysokim, to zerujemy tę zmienną.

Testbench modułu Double Dabble

Zgodnie z naszym zwyczajem opracujemy testbench, by przeprowadzić symulację nowego modułu. Jednak tym razem dodamy ciekawą funkcjonalność – testbench automatycznie przetestuje, czy moduł działa prawidłowo.

W przypadku konwersji liczby z formatu binarnego na BCD mamy stosunkowo proste zadanie, ponieważ moduł konwertera ma tylko jedno wejście danych **Binary_i** (pozostałe wejścia istnieją w gruncie rzeczy tylko po to, by wykonanie algorytmu było możliwe) – i jedno wyjście danych **BCD_o**. Dzięki temu zyskujemy możliwość przetestowania wszystkich kombinacji – wystarczy tylko, by w pętli wrzucać do modułu wszystkie liczby od zera do maksimum. Weryfikacja polegać będzie na porównaniu odpowiedzi testowanego modułu z odpowiedzią obliczoną przez testbench podczas symulacji.

```
        // Testuj wszystkie wartości od maksimum do zera
        for(i=MaxInput; i>=0; i=i-1) begin                          // 25
            @(posedge Clock);
            Binary <= i;
            Start  <= 1'b1;

            @(posedge Clock);
            Binary <= {INPUT_BITS{1'bX}};
            Start  <= 1'b0;

            @(posedge Done);
            Verify(i, BCD);
        end

        @(posedge Clock);

        // Pokaż wyniki weryfikacji
        $display("Pass: %d", PassCounter);                          // 26
        $display("Fail: %d", FailCounter);                          // 27
        $display("====== END ======");
        $finish;
    end

endmodule
```

Listing 2. Kod pliku double_dabble_tb.v – cd.

Przejdźmy do omówienia kodu testbencha, który pokazano na **listingu 2**. Tym razem testbench rozpoczniemy od kilku parametrów, dzięki czemu będziemy mogli łatwo i szybko zmieniać konfigurację testowanego modułu. W linii 1 ustawiamy parametr **INPUT_BITS** informujący o tym, ile bitów ma mieć liczba w formacie binarnym. W linii 2 za pomocą parametru **OUTPUT_DIGITS** ustawiamy, ile cyfr ma mieć liczba wyjściowa w kodzie BCD. Parametr **OUTPUT_BITS** określa (linia 3), ile bitów ma mieć wyjście modułu. W normalnych warunkach jest to czterokrotność liczby cyfr i nie ma potrzeby zmieniać tej praktyki, ale wspomniany parametr może zostać użyty do celowego wprowadzania błędów, by sprawdzić, co się stanie, jeżeli liczba bitów wyjścia okaże się niewystarczająca.

Generator sygnału zegarowego skonstruowany jest tak samo, jak we wszystkich poprzednich odcinkach kursu.

Na kolejnym etapie deklarujemy różne zmienne. Przejdźmy od razu do linii 4, w której tworzymy zmienną **Start** typu reg. Zmienna ta służyć będzie do sterowania wejściem **Start_i** testowanego modułu, a stan wysoki spowoduje uruchomienie konwersji. Linię niżej deklarujemy zmienną **Done** typu wire, która służyć będzie do oczekiwania na zakończenie konwersji. W linii 6 tworzymy zmienną **Binary** – jej celem będzie wysterowanie wejścia **Binary_i**. Zmienna inicjalizowana jest w ciekawy sposób: za pomocą operatora konkatenacji inicjalizujemy każdy z jej bitów wartością X, czyli wartością niezdefiniowaną. Chodzi o to, by testbench ustawiał jakąś konkretną wartość tej zmiennej tylko wtedy, kiedy będzie ona odczytywana przez testowany moduł. W linii 7 znajduje się zmienna **BCD** typu wire, która będzie przekazywać wynik z wyjścia **BCD_i** modułu DoubleDabble do tasku, który ma zweryfikować, czy wynik jest prawidłowy.

Przydadzą się jeszcze dwie zmienne typu integer, które posłużą do ułatwienia obliczeń. Zmienna **MaxInput** z linii 8 określa maksymalną liczbę, jaką można przekonwertować. Wynika ona z liczby bitów wejścia, określonej parametrem **INPUT_BITS**. W linii 9 tworzymy iterator pętli for.

W linii 10 tworzymy instancję testowanego modułu. Łączymy jej wejścia i wyjścia ze zmiennymi zdefiniowanymi wcześniej. Omińmy na razie task weryfikujący działanie modułu i przeskoczmy do sekwencji testowej.

W linii 21 mamy pętlę for, której zadaniem jest przetestowanie wszystkich liczb możliwych do konwersji (od zera do wartości maksymalnej). Praca wewnątrz pętli podzielona została na trzy etapy.

W pierwszym (linia 22) kopiujemy wartość iteratora pętli **i** do zmiennej **Binary**, połączonej z wejściem

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie https://ulubionykiosk.pl/media

eprasa.pl 4202fe84f2

testowanego modułu. Wartością tą będzie liczba, którą moduł ma przekonwertować z formatu binarnego na BCD. Następnie ustawiamy zmienną **Start** w stan wysoki, aby poinformować moduł, że ma odczytać swoje wejście danych i rozpocząć pracę.

W kolejnym takcie zegarowym (linia 23) ustawiamy zmienną Binary w stan nieistotny, podobnie jak to robiliśmy podczas jej deklarowania w linii 6. Musimy także wyzerować zmienną Start, bo w przeciwnym razie moduł rozpoczynałby pracę w każdym kolejnym takcie zegarowym.

Musimy poczekać, aż moduł skończy pracę (linia 24) i udostępni wynik na swoim wyjściu. Czekamy na pojawienie się stanu wysokiego na wyjściu **Done**. Wtedy uruchamiamy task weryfikujący, przekazując mu poprzez argument iterator pętli, czyli liczbę do konwersji, a także wynik obliczeń.

W linii 25 mamy kolejną, bardzo podobnie działającą pętlę. Wykonuje ona te same czynności, lecz iterator pętli zmniejsza się od wartości maksymalnej do zera.

Omówimy teraz sposób, w jaki testbench będzie sprawdzać, czy wynik zwrócony przez testowany moduł jest prawidłowy. Podczas symulacji moduł wykona ileś konwersji, a weryfikacja każdej z nich może dać wynik pozytywny lub negatywny. Potrzebujemy więc dwóch liczników do zliczania wyników każdej weryfikacji. W linii 11 tworzymy zmienną **PassCounter**, która będzie zwiększana o 1 po każdej pozytywnej weryfikacji, a linię niżej tworzymy zmienną **FailCounter**, która będzie inkrementowana w przypadku, gdy moduł zwróci wynik inny niż oczekiwany. Chcemy oczywiście, by po zakończeniu symulacji zawartość licznika **FailCounter** była równa zeru.

Task **Verify** rozpoczynamy w linii 13. Ma on dwa wejścia: **Binary** – które jest liczbą binarną do przekonwertowania na kod BCD – oraz wejście **BCD**, do którego powinna zostać doprowadzona ta sama liczba, lecz oczywiście w formacie BCD.

Metoda weryfikacji prezentuje się następująco: task ma za zadanie przeliczyć otrzymaną liczbę w formacie BCD na format binarny i porównać z liczbą, otrzymaną na wejściu **Binary**. Przeliczanie odbywa się w pętli for (linia 18). Iteratorem pętli jest zmienna Digit, która zmienia się od zera do OUTPUT_DIGITS. W każdym obiegu pętli odczytujemy kolejną cyfrę BCD, po czym mnożymy ją przez 10 podniesione do potęgi równej iteratorowi pętli. Tak otrzymany wynik dodajemy do obecnej wartości zmiennej **Result**. Mówiąc konkretniej, jeżeli na wejściu BCD mamy liczbę 123, to pętla wykonuje działanie $3 \cdot 10^0 + 2 \cdot 10^1 + 1 \cdot 10^2$ i tak obliczony wynik zapisuje do zmiennej **Result**.

Zwróć uwagę, gdzie zadeklarowane są zmienne **Digit** oraz **Result**. Pierwsza z nich powstaje przed blokiem begin-end (linia 14), a druga na początku tego bloku (linia 16), po czym jest inicjalizowana zerem (linia 17). Zrobiłem to specjalnie, by pokazać, że zmienne wewnątrz tasku można tworzyć na dwa sposoby. Zapamiętaj, że jeżeli chcesz utworzyć zmienną wewnątrz bloku begin-end, to ten blok musi być jakoś nazwany, nawet jeżeli ta nazwa nigdzie nie będzie używana. W przykładowym kodzie blok nazwałem **VerifyTask** (linia 15).

```
@echo off
iverilog -o double_dabble.o ^
            double_dabble.v ^
            double_dabble_tb.v
vvp double_dabble.o
del double_dabble.o
```

Listing 3. Kod pliku double_dabble.bat

```
VCD info: dumpfile double_dabble.vcd opened for output.
===== START =====
INPUT_BITS:            8
OUTPUT_BITS:          12
OUTPUT_DIGITS:         3
MaxInput:            255
Counter WIDTH:         3
Pass:          512
Fail:            0
====== END ======
double_dabble_tb.v:127: $finish called at 9218000 (1ns)
```

Listing 4. Log z konsoli po przeprowadzeniu symulacji

W linii 18 mamy wspomnianą wcześniej pętlę for, a w linii 19 – porównujemy wynik otrzymany na wejściu **BCD** z wynikiem obliczonym przez task. Porównania dokonujemy za pomocą operatora === a nie == (linia 19). Jest to istotne z tego powodu, że w języku Verilog istnieje wartość nieokreślona x. Taką wartość mają między innymi zmienne, którym nie przypisano wartości początkowej. Mogłoby się zdarzyć, że w wyniku błędnie napisanego kodu, na którymś wejściu tasku pojawią się wartości nieokreślone i zostaną one przekazane do porównania, a w rezultacie dostaniemy nieokreślony wynik. Operator === porównuje dokładnie wszystkie bity, uwzględniając wartości nieokreślone, a nawet stan wysokiej impedancji. Jeżeli zostaną wykryte jakiekolwiek różnice między porównywanymi zmiennymi, wówczas operator zwróci fałsz.

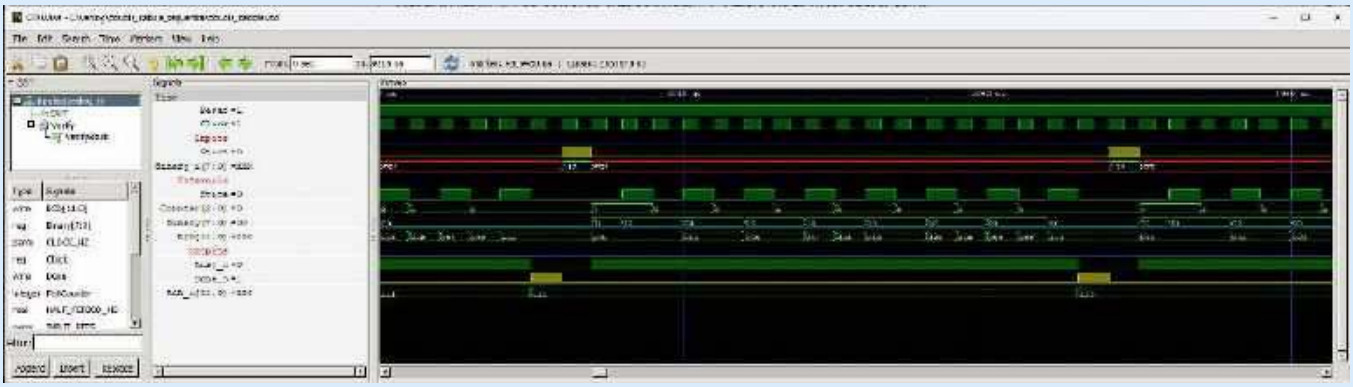
W zależności od wyniku porównania zwiększamy zmienną PassCounter lub FailCounter. W przypadku negatywnego wyniku weryfikacji wyświetlamy komunikat na konsoli za pomocą funkcji **$display()**, aby ułatwić szukanie błędu (linia 20), po czym symulacja wykonywana jest dalej. Jeżeli wykrycie błędu ma natychmiast zakończyć symulację, można posłużyć się funkcją **$fatal()**.

Na samym końcu symulacji wyświetlamy informację, ile testów zakończyło się pozytywną weryfikacją (linia 26), a ile razy uzyskano wynik negatywny (linia 27).

Aby wykonać symulację w symulatorze Icarus Verilog, uruchom skrypt, który zaprezentowano na **listingu 3** lub wpisz opisane polecenia ręcznie w konsoli systemowej.

Po przeprowadzeniu symulacji powinieneś zobaczyć zapisy podobne do tych widniejących na **listingu 4**. Symulator przetestował 512 różnych wariantów i nie znalazł ani jednego błędu.

Otwórz plik wynikowy w przeglądarce GTKWave i skonfiguruj ją tak, by uzyskać efekt widoczny na **rysunku 3**. Przybliżyłem



Rysunek 3. Okno symulacji

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie https://ulubionykiosk.pl/media

eprasa.pl 4202fe84f2

obraz na proces konwersji liczby 113 – właśnie taką konwertowaliśmy w przykładzie na początku tego odcinka. Zerknij ponownie na rysunek 2 i spróbuj znaleźć podobieństwa.

Moduł top

Przetestujmy nasz nowy moduł w praktyce na prawdziwym FPGA, za pomocą płytek MachXO2 Mega oraz User Interface Board, które zostały zaprezentowane w EP 09/2023 i można je nabyć w sklepie AVT. Opracujemy aplikację testową, w której zastosujemy enkoder obrotowy do sterowania licznikiem. Obrót w prawo będzie zwiększał licznik o 1, a w lewo będzie zmniejszał o 1. Licznik będzie miał możliwość liczenia od 0 do 9999. Zastosujemy również sterownik 8-cyfrowego wyświetlacza 7-segmentowego. Cztery cyfry po prawej stronie wyświetlacza będą pokazywać stan licznika w formacie szesnastkowym, tak jak to robiliśmy dotychczas. Natomiast cztery cyfry po lewej będą pokazywać stan licznika w zapisie dziesiętnym, po przekonwertowaniu przez moduł Double Dabble opracowany w tym odcinku kursu.

Utwórz nowy projekt i dodaj do niego pliki, które widać w drzewku projektowym na **rysunku 4**. Wszystkie pliki omawialiśmy już w poprzednich odcinkach kursu. Możesz je pobrać z repozytorium na GitHubie, dostępnego pod adresem [1], a cały projekt znajdziesz w programie Diamond pod adresem [2].

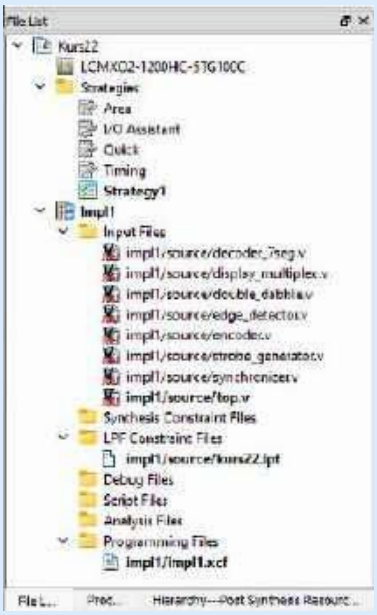
Weźmy na warsztat kod modułu top, pokazany na **listingu 5**. Oprócz wejścia zegara i resetu, moduł zawiera wejścia, które odczytują piny enkodera obrotowego – oraz wyjścia sterujące wyświetlaczem. Elementy te omówione zostały w odcinkach numer 9 i 14.

Obsługą enkodera zajmuje się moduł **Encoder**, którego instancję tworzymy w linii 2. Zmienne sygnalizujące wykrycie obrotu w prawo i w lewo to **Increment oraz Decrement** typu wire, które tworzymy w linii 1. Wykrycie takiego zdarzenia powoduje ustawienie stanu wysokiego na długość jednego taktu sygnału zegarowego.

W linii 3 tworzymy 16-bitowy licznik **Counter** – jego celem jest zliczenie impulsów z enkodera. Poniżej mamy blok always, w którym zaimplementowano logikę licznika. Licznik ma zmieniać swoją wartość od 0 do 9999, zatem jeżeli ustawiony jest na wartość maksymalną i wtedy nadejdzie żądanie inkrementacji, musimy ustawić go w stan 0 (linia 4). Podobnie w drugą stronę: jeżeli obecna wartość licznika to 0 i sygnał dekrementacji jest w stanie wysokim, to trzeba go ustawić na wartość maksymalną (linia 5).

Powinniśmy wprowadzić prostą linię opóźniającą o jeden takt zegarowy dla sygnału wyzwalającego konwersję. Pomyślmy. Sygnał inkrementacji lub dekrementacji jest ustawiany na jeden cykl zegara. W następnym cyklu licznik ulega zwiększeniu lub zmniejszeniu. Konwersję należy zatem uruchomić dopiero w kolejnym takcie po zmianie stanu licznika.

Najprościej ten problem rozwiązać, wprowadzając opóźnienie w postaci zwyczajnego przerzutnika D. Takąrolęodgrywa1-bitowa zmienna **ConversionStart** typu reg, którą tworzymy w linii 6. Logikę przerzutnika D opisujemy w osobnym bloku always poniżej. W każdym cyklu zegarowym wpisujemy do niego stan sygnałów **Increment** lub **Decrement** połączonych bramką OR (linia 7).



Rysunek 4. Drzewko projektu

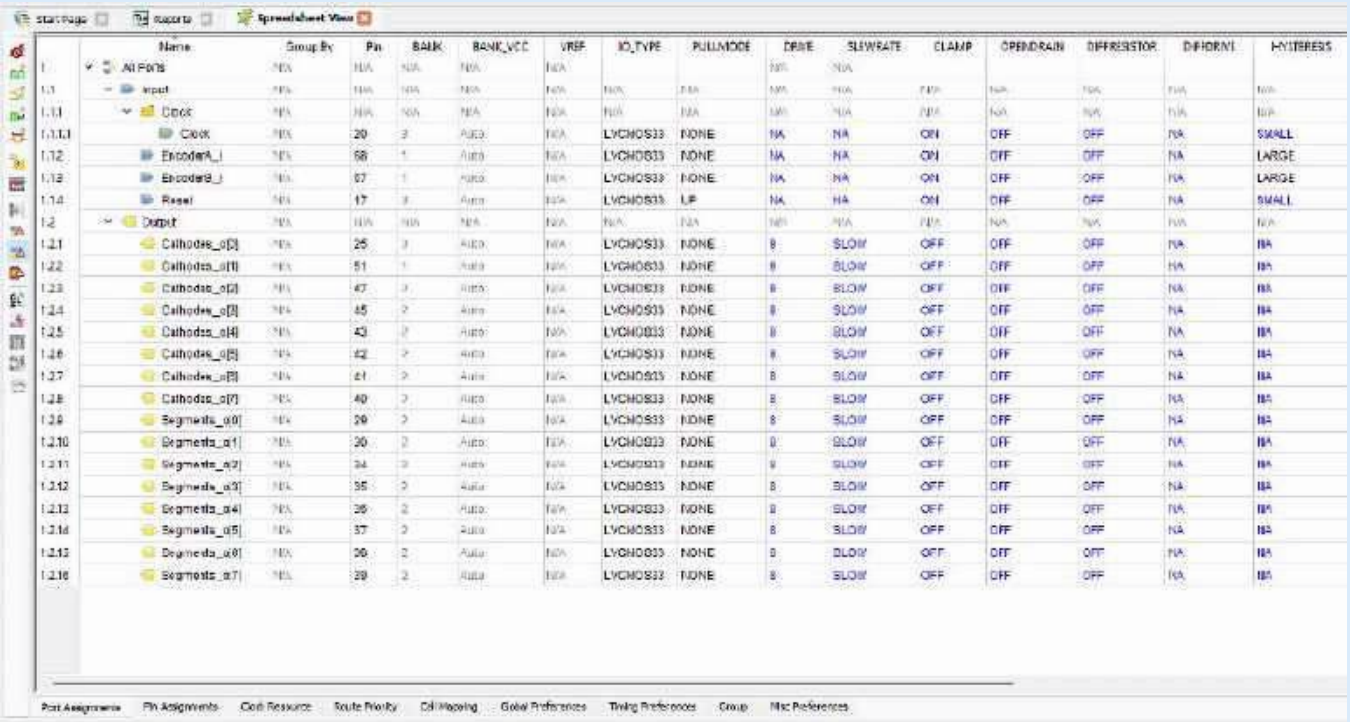
Przejdźmy do instancji modułu konwertera **DoubleDabble** w linii 9. Kiedy wejście Start_i – do którego doprowadzono zmienną

```
// Plik top.v

`default_nettype none

module top #(
    parameter CLOCK_HZ = 25_000_000
)(
    input wire Clock,                // Pin 20
    input wire Reset,                // Pin 17
    input wire EncoderA_i,           // Pin 68
    input wire EncoderB_i,           // Pin 67
    output wire [7:0] Cathodes_o,    // Pin 40 41 42 43 45 47 51 52
    output wire [7:0] Segments_o     // Pin 39 38 37 36 35 34 30 29
);

    // Zmienne dla enkodera obrotowego
    wire Increment;                                     // 1
    wire Decrement;

    // Instancja enkodera obrotowego
    Encoder Encoder_inst(                               // 2
        .Clock(Clock),
        .Reset(Reset),
        .AsyncA_i(EncoderA_i),
        .AsyncB_i(EncoderB_i),
        .AsyncS_i(1'b1),
        .Increment_o(Increment),
        .Decrement_o(Decrement),
        .ButtonPress_o(),
        .ButtonRelease_o(),
        .ButtonState_o()
    );

    // Licznik w górę i w dół o zakresie 0...9999
    reg [15:0] Counter;                                 // 3

    // Logika licznika
    always @(posedge Clock, negedge Reset) begin
        if(!Reset) begin
            Counter <= 0;
        end else if(Increment) begin                    // 4
            if(Counter == 16'd9999)
                Counter <= 16'd0;
            else
                Counter <= Counter + 1'b1;
        end else if(Decrement) begin                    // 5
            if(Counter == 16'd0)
                Counter <= 16'd9999;
            else
                Counter <= Counter - 1'b1;
        end
    end

    // Wyzwalanie konwersji
    reg ConversionStart;                                // 6

    always @(posedge Clock, negedge Reset) begin
        if(!Reset) begin
            ConversionStart <= 0;
        end else begin
            ConversionStart <= Increment | Decrement;   // 7
        end
    end

    // Zmienna dla liczby w formacie dziesiętnym
    wire [15:0] Decimal;                                // 8

    // Instancja konwertera BIN/BCD
    DoubleDabble #(                                     // 9
        .INPUT_BITS(16),
        .OUTPUT_DIGITS(4)
    ) DoubleDabble_inst(
        .Clock(Clock),
        .Reset(Reset),
        .Start_i(ConversionStart),
        .Busy_o(),
        .Done_o(),
        .Binary_i(Counter),
        .BCD_o(Decimal)
    );

    // Instancja wyświetlacza
    DisplayMultiplex #(                                 // 10
        .CLOCK_HZ(CLOCK_HZ),
        .SWITCH_PERIOD_US(1000),
        .DIGITS(8)
    ) DisplayMultiplex_inst(
        .Clock(Clock),
        .Reset(Reset),
        .Data_i({Decimal, Counter}),                    // 11
        .DecimalPoints_i(8'b00010000),
        .Cathodes_o(Cathodes_o),
        .Segments_o(Segments_o),
        .SwitchCathode_o()
    );

endmodule

`default_nettype wire
```

Listing 5. Kod pliku top.v

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie **https://ulubionykiosk.pl/media**

eprasa.pl 4202fe84f2



	Name	Group By	Pin	BANK	BANK_VCC	VREF	IO_TYPE	PULLMODE	DRIVE	SLEWRATE	CLAMP	OPENDRAIN	DIFFRESISTOR	DIFFOUT	HYSTERESIS
1	All Ports	NA	NA	NA	NA	NA			NA	NA					
1.1	Input	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1.1.1	Clock	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1.1.1.1	Clock	NA	20	3	Auto	NA	LVCMOS33	NONE	NA	NA	ON	OFF	OFF	NA	SMALL
1.1.2	Encoder0_i	NA	68	1	Auto	NA	LVCMOS33	NONE	NA	NA	ON	OFF	OFF	NA	LARGE
1.1.3	Encoder1_i	NA	67	1	Auto	NA	LVCMOS33	NONE	NA	NA	ON	OFF	OFF	NA	LARGE
1.1.4	Reset	NA	17	3	Auto	NA	LVCMOS33	UP	NA	NA	ON	OFF	OFF	NA	SMALL
1.2	Output	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1.2.1	Cathodes_o[0]	NA	25	3	Auto	NA	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.2	Cathodes_o[1]	NA	51	1	Auto	NA	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.3	Cathodes_o[2]	NA	47	2	Auto	NA	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.4	Cathodes_o[3]	NA	45	2	Auto	NA	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.5	Cathodes_o[4]	NA	43	2	Auto	NA	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.6	Cathodes_o[5]	NA	42	2	Auto	NA	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.7	Cathodes_o[6]	NA	41	2	Auto	NA	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.8	Cathodes_o[7]	NA	40	2	Auto	NA	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.9	Segments_o[0]	NA	29	2	Auto	NA	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.10	Segments_o[1]	NA	30	2	Auto	NA	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.11	Segments_o[2]	NA	34	2	Auto	NA	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.12	Segments_o[3]	NA	35	2	Auto	NA	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.13	Segments_o[4]	NA	36	2	Auto	NA	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.14	Segments_o[5]	NA	37	2	Auto	NA	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.15	Segments_o[6]	NA	38	2	Auto	NA	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA
1.2.16	Segments_o[7]	NA	39	2	Auto	NA	LVCMOS33	NONE	8	SLOW	OFF	OFF	OFF	NA	NA

Rysunek 5. Konfiguracja pinów w Spreadsheet

ConversionStart – znajduje się w stanie wysokim, rozpoczyna się konwersja danych na wejściu **Binary_i**, do którego doprowadzono licznik **Counter**. Po jakimś czasie wynik jest udostępniany na wyjściu **BCD_o**, skąd przesłany zostanie do kolejnych modułów za pośrednictwem 16-bitowej zmiennej **Decimal** typu wire, utworzonej w linii 8. Nie korzystamy z wyjść **Busy_o** ani **Done_o**, ponieważ moduł wyświetlacza działa ciągle i nie trzeba go informować, że stan jego wejść się zmienił.

Ostatni moduł, którego instancję tworzymy w linii 10, to moduł sterownika wyświetlacza **DisplayMultiplex**. Wielokrotnie był on już używany w poprzednich odcinkach. Na jego 32-bitowe wejście **Data_i** (linia 11) dostarczamy 16-bitowy stan licznika **Counter** oraz 16-bitowy wynik konwersji **Decimal**. Obie te zmienne sklejamy w jedną za pomocą operatora konkatenacji {}.

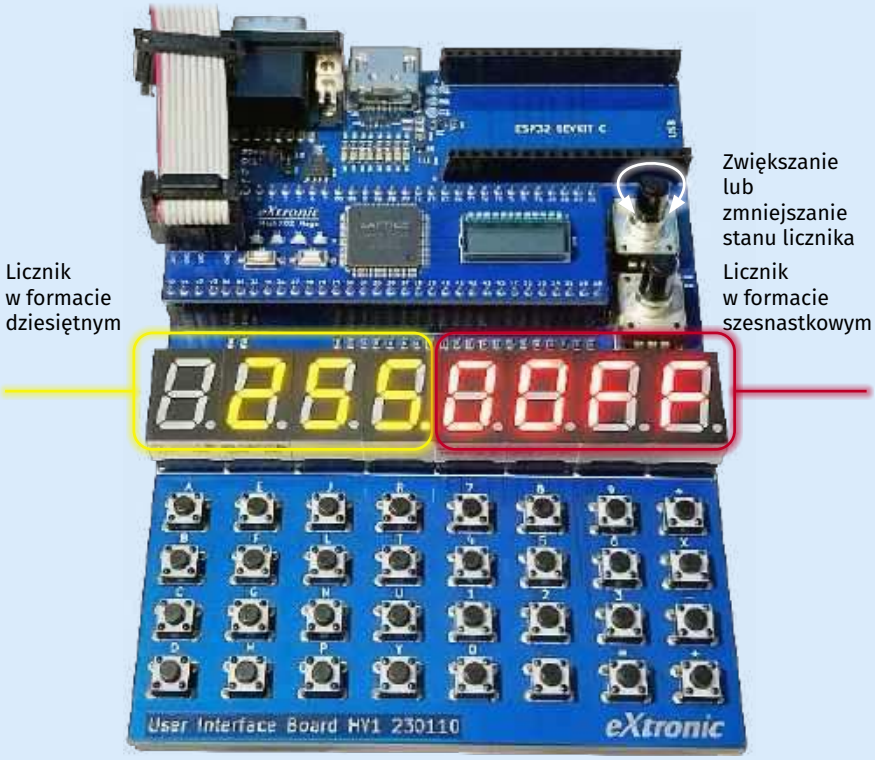
Uruchom syntezę, a następnie skonfiguruj piny w narzędziu Spreadsheet w taki sposób, jak to pokazano na **rysunku 5**. Pamiętaj, by na dole okna kliknąć zakładkę Timing Preferences i ustawić częstotliwość 25 MHz dla wejścia Clock.

Po wygenerowaniu bitstreamu i wgraniu go do FPGA powinieneś zobaczyć cyfrę 0 po prawej stronie wyświetlacza (moduł DisplayMultiplex wygasza nieistotne zera i traktuje dane na wejściu jako jedną liczbę). Pokręć dekoderem E41 w prawo lub w lewo, a stan licznika na wyświetlaczu będzie się zmieniał.

Przykład operacji pokazano na **fotografii 1**. Po lewej stronie widzimy liczbę dziesiętną 255, co w zapisie szesnastkowym ma postać: FF – i właśnie taka liczba jest wyświetlana po prawej stronie.

Algorytm Double Dabble w wersji sekwencyjnej jest łatwy i wymaga niewiele zasobów sprzętowych. Jego wadę stanowi fakt, że musimy czekać pewną liczbę taktów zegarowych na uzyskanie wyniku, a im większe liczby chcemy przekonwertować, tym dłużej potrwa oczekiwanie. Opisany algorytm da się zaimplementować także w wersji kombinacyjnej – czyli bez żadnego sygnału zegarowego. Tym zajmiemy się w kolejnym odcinku.

Dominik Bieczyński
leonow32@gmail.com



Fotografia 1. Płytka testowa podczas pracy

Zobacz więcej:
- Repozytorium modułów używanych w kursie https://github.com/leonow32/verilog-fpga
- Projekt w programie Diamond: https://tiny.pl/d4v75
- Film demonstrujący działanie aplikacji: https://youtu.be/96zKycZBVxI

Poprzednie części kursu i dodatkowe materiały dostępne są na stronie **https://ulubionykiosk.pl/media**

eprasa.pl 4202fe84f2

koktajl niusów

Nowoczesny system zdalnego odczytu liczników wody z użyciem sieci LoRaWAN – jedna z kluczowych inwestycji firmy Emitel we Wrocławiu

Jest to pierwszy tego rodzaju system w Polsce, który korzysta z sieci LoRaWAN w celu bezprzewodowej wymiany danych. Zaprojektowane od podstaw i zbudowane przez firmę Emitel rozwiązanie obejmuje swoim zasięgiem cały Wrocław, pozwalając na łatwy i niezawodny odczyt danych z liczników wody na terenie całej metropolii. Wdrożenie opisywanego systemu niewątpliwie przyczyniło się do znacznego wzrostu efektywności zarządzania infrastrukturą wodociągową na obszarze Wrocławia, dzięki czemu Emitel dołączył do grona pionierów technologii IoT do zarządzania infrastrukturą w miastach. Specjalne nakładki zainstalowane przez specjalistów firmy na wodomierzach pozwalają na bezprzewodową komunikację LoRaWAN i zdalny odczyt danych bez udziału człowieka. Jednym z kluczowych wyzwań, z którym należało się zmierzyć projektując system zdalnego odczytu liczników wody, było pokonanie szeregu przeszkód wynikających w głównej mierze z charakterystyki zabudowy Wrocławia – grubych murów budynków oraz głębokich studzienek i trudno dostępnych miejsc. Co ważne: system umożliwia stabilne i niezawodne przesyłanie danych, nawet z najtrudniej dostępnych obszarów, gwarantując dobrą dostępność wskaźników odczytu, a ten – bez względu na okoliczności – oczekiwany jest zawsze przez odbiorcę końcowego.

https://tiny.pl/d46q1

Historyczne zwycięstwo zespołu AGH Space Systems w zawodach University Rover Challenge 2024

W znakomitym stylu, różnicą jedynie 0,96 punktu, zespół AGH Space Systems zdołał pokonać obrońców tytułu – zespół Team Mountaineers z West Virginia University (USA). Jest to wprost fantastyczny wynik zespołu, któremu jeszcze nigdy nie udało się zdobyć wysokiej nagrody na amerykańskiej ziemi. Zanim jednak do tego doszło, zawodnicy musieli pokonać 37 zespołów z całego świata zakwalifikowanych do finału University Rover Challenge 2024. Były to zespoły wybrane spośród aż 102 teamów uczestniczących, i śmiało rywalizujących, w tegorocznej edycji wspomnianych zawodów. O osiągnięciu historycznego wyniku przez zespół krakowskiej uczelni zadecydował w sporej mierze pomyślny, świetny występ łazika Kalmana we wszystkich 4 konkurencjach zawodów. Wart odnotowania jest fakt otrzymania wysokich ocen głównie

za dokumentację techniczną projektu i konkurencję autonomicznej jazdy po trudnym, marsjańskim terenie. Działający na co dzień w ramach koła naukowego zespół AGH Space Systems zajmuje się ulepszaniem elektroniki pokładowej, obsługującej m.in. łazik Kalman. Na 2024 rok uczelniany team zaplanował prace m.in. nad systemem komunikacyjnym (dzięki któremu możliwe ma się stać sterowanie robotem na odległość), a także nad integracją wizji cyfrowej, umożliwiającej operatorom rozpoznawanie obiektów i otoczenia w wysokiej jakości obrazu – bez opóźnień.

https://tiny.pl/d46q4

Telewizor The Frame firmy Samsung w unikatowej ramie Crystals

Znany polski designer, Oskar Zięta, stworzył ramę Crystals do telewizora The Frame. Projekt ten jest owocem współpracy firm Samsung oraz Zieta Studio – i stanowi połączenie sztuki, technologii, a także inżynierii. Idealnie odbijająca światło rama Crystals podkreśla matową powierzchnię telewizora The Frame oraz – jak komunikuje marka Samsung, prezentując rynkową nowość – łączy w sobie technologiczne wątki opowieści o deformacji metalu, a także maksymalizuje potencjał odbić światła na polerowanych powierzchniach. Producent wskazuje też, że optymalnie dopracowane kąty oprawy multiplikują i zakrzywiają obrazy dostępne na ekranie telewizora, a na metalowej powierzchni ramy Crystals splata się multum płaszczyzn interpretacji. Jest to pierwsze tak oryginalne akcesorium do telewizora The Frame, które to otwiera nową erę designu w tej kategorii rozwiązań, a jednocześnie w wysokim stopniu traktuje o technologii, hipnotyzując potęgą obiektów stałych i odbitych. Warto zaznaczyć, że Oskar Zięta to światowej sławy polski projektant i architekt, który nadzwyczaj chętnie eksperymentuje

eprasa.pl 4202fe84f2

ze skalą oraz formą, a symbolem jego kreacji jest metal (wliczając w to stal i wynikający z niej potencjał widoczny w obiektach z niej wykonanych).

https://tiny.pl/d46q8

Narzędzie Diagnostics firmy Apple do samoobsługowej naprawy dostępne również w Europie

Od niedawna narzędzie Diagnostics firmy Apple do samoobsługowej naprawy dostępne jest również w 32 krajach Europy. Dostarcza ono użytkownikom możliwości, które wcześniej zarezerwowane były wyłącznie dla autoryzowanych serwisów Apple oraz niezależnych dostawców usług w zakresie napraw sprzętu elektronicznego. Klienci mogą więc sprawdzać produkty pod kątem pracy poszczególnych podzespołów, a także identyfikować komponenty wymagające natychmiastowej naprawy. Od kwietnia 2022 roku – w ramach programu samoobsługowej naprawy – każda osoba, która posiada doświadczenie w serwisie urządzeń elektronicznych, w praktyce ma zapewniony przez amerykańskiego producenta dostęp do instrukcji, oryginalnych części i narzędzi używanych w sklepach Apple Store lub przez autoryzowane serwisy Apple. Obecnie program ten przewidziany jest dla 42 produktów marki, w tym MacBooka Air z układem scalonym M3. Zapewniane przez firmę narzędzie Diagnostics stanowi efekt ciągłego dążenia do wydłużania cyklu życia produktów. Decydentom z Apple zależy na tym, żeby oferować bezpieczne i ekonomiczne opcje naprawy, lecz jednocześnie priorytetem dla niej pozostaje, w dużej mierze, projektowanie i tworzenie wytrzymałych rozwiązań. Zarówno z punktu widzenia klientów, jak i planety najlepiej, żeby naprawa w ogóle nie była potrzebna. Dzięki narzędziu Diagnostics użytkownikom urządzeń jest łatwiej stwierdzić, czy ich produkt faktycznie potrzebuje serwisu. Podczas sesji uruchomionej na innym urządzeniu mogą sprawdzić stan oraz sprawność produktu. Po rozpoczęciu sesji oraz przebrnięciu przez wyświetlane na ekranie komunikaty użytkownik dowiaduje się nie tylko, czy jego urządzenie z miejsca wymaga naprawy, ale także, które jego elementy należy wymienić (we własnym zakresie).

https://tiny.pl/d46xm

Minęło 70 lat od debiutu pierwszej lampy projekcyjnej do kin z rodziny XBO OSRAM

Wprowadzona na rynek w 1954 roku kinowa lampa projekcyjna z rodziny XBO szybko i skutecznie zastąpiła stosowaną przedtem lampę węglową, odnosząc sukces i stając się wkrótce międzynarodowym standardem. Okazała się tak wielką rewelacją, że w 1983 roku otrzymała Oscara za wysoką wydajność świetlną oraz innowacyjność techniczną. Jeszcze w wielu miejscach ksenonowe lampy wyładowcze (a do tej grupy należy m.in. konstrukcja przywołana na początku tego wpisu) stanowią niezwykle pożądane rozwiązanie. Dzieje się tak za sprawą wysokiego poziomu luminancji i dużej wartości współczynnika oddawania barw (CRI) – razem zapewniających ostry jak brzytwa obraz o wysokiej intensywności kolorów. Produkowane do dzisiaj kinowe lampy projekcyjne z rodziny XBO wyróżnia m.in. dość długa żywotność oraz stabilny strumień światła. Przez ostatnie lata ich unikatowa konstrukcja była – i wciąż jest rozwijana, przy nieustannej optymalizacji obejmującej także proces produkcji tych lamp. Pomimo upływu czasu rozwiązania te cieszą się dużym uznaniem operatorów kinowych i tego stanu rzeczy nie zmienia nawet podbijająca kina od lat projekcja laserowa. Współcześnie rodzina kinowych lamp projekcyjnych XBO obejmuje produkty przeznaczone w całości do cyfrowych projektorów filmowych. Są to lampy stale dostosowywane do rosnących i wciąż zmieniających się wymagań branżowych, za sprawą których widzowie mogą faktycznie zachwycać się technologią projekcji cyfrowej. Jak wyjaśnia Product Manager Cinema Lamps w firmie ams OSRAM, André Köhler:

„Z nieskrywaną dumą spoglądamy na liczącą ok. 70 lat historię kinowych lamp projekcyjnych z rodziny XBO – historię, w której zarówno tradycja, jak i innowacja przeplatają się ze sobą, i to w nieoczekiwany sposób. Są to w istocie 2 rzeczy, które wniosły znaczący wkład w świat kina, sprawiając, że staliśmy się cenionym partnerem wielu operatorów kin na całym świecie".

https://tiny.pl/d46xw

Ponieważ puste nie zawsze jest puste – najnowszy tester baterii produkcji VARTA

Każdego roku pozbywamy się wielu ton zaledwie połowicznie rozładowanych baterii. Średnio tylko 50% wszystkich wyrzucanych baterii faktycznie dobrnęło do końca swojej przydatności. By realnie przyczynić się do minimalizacji tego niepokojąco wysokiego odsetka, firma VARTA wprowadza nową wersję swojego testera baterii. Nowe funkcje, dobra kompatybilność i nowoczesny design sprawiają, że urządzenie jest niezbędnym produktem w nowoczesnych gospodarstwach domowych. Umożliwia ono sprawdzenie rzeczywistego stanu naładowania ogniw, a tym samym zapewnia ich bardziej zrównoważone użytkowanie. Jest to wyjątkowo wszechstronny produkt, imponujący łatwością obsługi, nadający się do wszystkich popularnych baterii jednorazowych i akumulatorów, m.in. AA, AAA, C, D i 9 V oraz alkalicznych ogniw guzikowych. Miernik w przystępny sposób wskazuje na wbudowanym wyświetlaczu LCD poziomy napięć, w szczególności ogniw Li-Ion czy akumulatorów NiMH. Wyświetlana jest dokładna wartość napięcia w woltach oraz trzystopniowa skala w formie podobizny baterii (3 paski – pełne ogniwo, 2 paski – ogniwo nadal nadające się do użytku, a także 1 pasek – ogniwo do wymiany).

https://tiny.pl/d46xf

Znakomitej czystości węglan winylenu (VC) do baterii litowo-jonowych (Li-Ion) – jeden z flagowych produktów Grupy PCC

Węglan winylenu ma szczególne znaczenie dla dynamicznie rozwijającego się sektora baterii litowo-jonowych (Li-Ion). Jako specjalistyczny dodatek do elektrolitów pełni on funkcję środka błonotwórczego w procesie formowania filmu. Tego typu domieszki

eprasa.pl 4202fe84f2

zapobiegają m.in. tworzeniu się dendrytów oraz znaczącej degradacji roztworu elektrolitu. Wytwarzany przez Grupę PCC węglan winylenu (VC) to bezbarwna, klarowna ciecz, która jest stabilizowana przez dodatek antyutleniaczy i środków antyrodnikujących w celu uniknięcia polimeryzacji. Produkt ten występuje w funkcji dodatku błonotwórczego lub prekursora polimeryzacji, który doskonale rozpuszcza się w etanolu, tetrahydrofuranie, węglanie etylenu, węglanie propylenu, węglanie dimetylu, a także węglanie dietylu (są to rozpuszczalniki elektrolityczne stosowane w bateriach Li-Ion). Węglan winylenu pakowany jest w atmosferze ochronnej azotu do bębnów ze stali nierdzewnej, które zawierają specjalne szybkozłącza. Próbki produktu pakuje się w butelki fluorowane oraz w butelki ze stali nierdzewnej lub aluminium, owinięte na zewnątrz workiem z folii aluminiowej. Opisywany produkt należy chronić przed działaniem światła słonecznego i przechowywać w suchym, chłodnym miejscu – w temperaturze otoczenia poniżej 25°C.

https://tiny.pl/d46tp

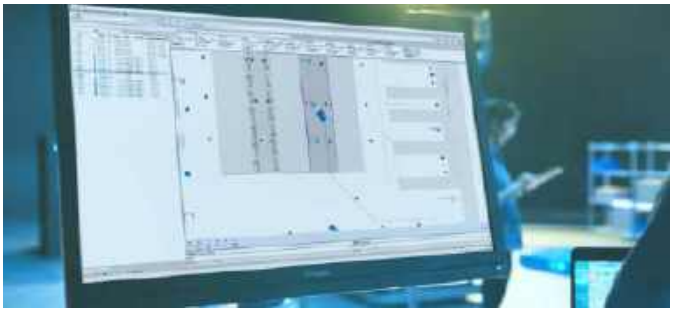
Oparty na sztucznej inteligencji robot CLOi ServeBot firmy LG Electronics dla branży hotelarskiej i opieki zdrowotnej

Zaprojektowany z myślą o hotelarstwie i służbie zdrowia robot CLOi ServeBot może bez trudu korzystać z windy w celu przemieszczania się pomiędzy piętrami. Każde z 6 kół robota ma niezależne zawieszenie, co nadaje stabilność, a także umożliwia transport napojów po nierównych powierzchniach, bez rozlewania. Z myślą o jeszcze większej wygodzie użytkownika drzwi CLOi ServeBot można ustawić tak, aby otwierały się lub zamykały automatycznie podczas pracy. Opisywany robot samoczynnie stwierdza, kiedy dostawa transportowanego ładunku została zakończona, ponieważ czujniki i kamery wbudowane w jego przedziały niezawodnie wykrywają moment, w którym przedmioty są usuwane. Dzięki algorytmom sztucznej inteligencji robot z łatwością może omijać różne przeszkody, określać optymalną trasę do miejsca docelowego, a także komunikować się z maksymalnie 20 innymi robotami CLOi ServeBot (w celu maksymalizacji wydajności zadań lub uniknięcia potencjalnych kolizji). Przedni wyświetlacz o przekątnej 10,1" działa jak mobilna platforma reklamowa, a treści można bez zakłóceń przesyłać w oparciu o stację LG CLOi Station (przez smartfon lub internet) – bądź aplikację mobilną CMS. Jakby tego było mało, robot CLOi ServeBot może również eskortować klientów do zadanego miejsca docelowego i zapewnić nieskomplikowane wskazówki głosowe dla osób niedowidzących. Dodatkowo użytkownicy mogą monitorować status robota w czasie rzeczywistym za pomocą stacji LG CLOi Station, która zapewnia wygodne w obsłudze funkcje przywoływania oraz planowania.

https://tiny.pl/d46tk

Wprowadzenie oprogramowania FLOW Core 4.0 firmy OMRON umożliwiającego skalowalne wdrażanie robotów AMR

Firma OMRON ogłosiła, że ma zaszczyt zaprezentować oprogramowanie Fleet Operations Workspace (FLOW) Core 4.0 – najnowszej generacji platformę znacząco rozszerzającą możliwości autonomicznych robotów mobilnych (tj. AMR) z serii: LD, MD i HD, która upraszcza proces integracji i usprawnia zarządzanie flotami. Jest to pojedynczy pakiet oprogramowania służący m.in. do koordynowania



zadań i harmonogramów ładowania, obsługi zaawansowanej nawigacji i kontroli ruchu oraz zapewnienia bezproblemowej, a przy tym błyskawicznej integracji z systemami IT i OT, takimi jak: MES, ERP i WMS. Zaktualizowane oprogramowanie FLOW Core 4.0 oferuje, w pierwszej kolejności, zwirtualizowaną wersję Fleet Manager – rozwiązania OMRON, które w praktyce służy do obsługi floty robotów AMR. Opisywane oprogramowanie sprawnie obsługuje lidar 3D, występujący we wszystkich modelach robotów AMR. Przewidziano w nim różne narzędzia pozwalające na dostosowywanie strategii dokowania do procedur ładowania. Zapewniono również pakiet analizy danych FLOW iQ, który umożliwia wizualizację i monitorowanie układu infrastruktury bezprzewodowej, widzianego przez robota. Wprowadzenie tzw. dynamicznej diagnostyki sieci bezprzewodowej umożliwia osiągnięcie istotnej konwergencji IT i OT, przyspieszając wdrażanie floty robotów AMR. Z kolei za sprawą interfejsu użytkownika MobilePlanner 8 można zwiększyć wydajność automatyzacji – w szczególności korzystając z edytora kształtów 2D, który oferuje m.in. możliwość obiektowego zdefiniowania kształtów docelowych CAPS i robotów, bez konieczności pisania kodu.

https://tiny.pl/d46ts

Niezwykle przydatne złącza Connectivity Heroes firmy Murrelektronik

Każdy superbohater ma specjalną moc – złącza Connectivity Heroes od Murrelektronika również! Część z nich dba o dane, a reszta o sygnały oraz zasilanie. W ofercie firmy Murrelektronik znajdują się m.in. prefabrykowane złącza typów M8 i M12 oraz wtyki zaworowe. Wiele z nich zintegrowane zostało z diodami LED bądź tłumikami przepięć, a także uzupełnione o akcesoria i narzędzia, w tym: adaptery, trójniki i klucze dynamometryczne. Oprócz tego dostępne są złącza sieciowe M8, M12 i RJ45 (obok M12 Power oraz MQ15). W przypadku modelu M12 Power mamy do czynienia z kompaktowymi i wytrzymałymi elementami. Z kolei model MQ15 z systemem szybkiego łączenia pozwala na sprawne podłączanie silników. Każde ze złączy z serii Connectivity Heroes zostało skrupulatnie przetestowane i jest sprawdzane przed wysyłką do klienta. W szczególności przeprowadza się testy: zwarciowe, odwracania przewodów oraz wysokonapięciowe, testowana jest również funkcjonalność złączy, które dodatkowo poddaje się końcowej kontroli wizualnej. Tym sposobem firma Murrelektronik zawsze daje klientom pewność, że otrzymują oni rozwiązania najwyższej klasy.

https://tiny.pl/d46tb

Jakub Tyburski
jakub.tyburski@elportal.pl

eprasa.pl 4202fe84f2

Hity następnego numeru • „Elektronika Praktyczna” 9/2024

Temat numeru: Nowoczesne zasilacze impulsowe

Zasilacze impulsowe niezaprzeczalnie górują nad poczciwymi konstrukcjami transformatorowymi pod wieloma względami. Oferują doskonałą sprawność na poziomie przekraczającym 90% oraz wysoką gęstość mocy (a co za tym idzie – także kompaktowe wymiary). Te istotne zalety to niestety tylko jedna strona medalu – wysoki poziom zakłóceń przewodzonych i promieniowanych oraz problemy związane z nieciągłym charakterem poboru prądu z sieci energetycznej wciąż pozostają problemem wymagającym licznych ulepszeń i optymalizacji. We wrześniowym numerze naszego czasopisma prezentujemy szereg informacji praktycznych, bardzo istotnych z punktu widzenia konstruktorów urządzeń i systemów elektronicznych. Poruszymy m.in. tematykę topologii układowych (w tym przetwornic rezonansowych), znaczenia układów PFC, prostowników synchronicznych, a także zasilaczy przeznaczonych do aplikacji specjalistycznych (ze szczególnym naciskiem na urządzenia medyczne).

Elektronika w Praktyce: Zestawy edukacyjne i ewaluacyjne

Zestawy ewaluacyjne pełnią niezwykle ważną funkcję w arsenale narzędzi współczesnego elektronika: pozwalają bardzo szybko i łatwo zapoznać się z funkcjonalnością układu scalonego bądź modułu, bez konieczności wykonywania jakichkolwiek prac projektowych lub montażowych. Rynek płytek uruchomieniowych to zarówno proste układy zawierające kilka lub kilkanaście elementów, jak i niezwykle bogato wyposażone, kilkunastowarstwowe płyty drukowane przeznaczone do pracy z najnowocześniejszymi macierzami FPGA, procesorami aplikacyjnymi czy też układami mixed-signal o najwyższej wydajności. Nie sposób nie wspomnieć też o ekosystemach elektroniki modułowej, które z równym powodzeniem znajdują swoich odbiorców wśród początkujących amatorów, jak i profesjonalistów opracowujących prototypy nowych urządzeń. We wrześniowej odsłonie naszej stałej rubryki „Elektronika w Praktyce” przyglądamy się aktualnym trendom na rynku zestawów ewaluacyjnych oraz edukacyjnych, prezentujemy też subiektywny wybór najciekawszych modeli dostępnych w sprzedaży.

Sonda logarytmiczna (2)

Szerokopasmowe sondy pomiarowe należą do przyrządów, których nie powinno zabraknąć w pracowni wszechstronnego elektronika-konstruktora, szczególnie jeśli jest on radioelektronikiem. Sondy logarytmiczne przydają się zwłaszcza do mierzenia napięć, a pośrednio – także prądów i mocy w obwodach prądów zmiennych. Do niezaprzeczalnych zalet tych przyrządów należy możliwość wykonywania dość precyzyjnych pomiarów sygnałów o bardzo dużej dynamice, w szerokim zakresie częstotliwości, przy czym – gdy w pomiarach zastosuje się generator-wobulator i oscyloskop – można prześledzić także charakterystyki częstotliwościowe badanego podzespołu czy urządzenia. We wrześniowym numerze „Elektroniki Praktycznej” kontynuujemy opis takiego interesującego i niezwykle użytecznego przyrządu pomiarowego – tym razem zaprezentujemy szczegóły konstrukcyjne urządzenia, opiszemy sposób jego montażu oraz uruchomienia.

aRGB controller (1)

Komputery klasy PC już dawno przestały być tylko bezdusznymi maszynami liczącymi, stosowanymi do nauki czy rozrywki. Stały się elementami wyposażenia wnętrz, a w tym obszarze funkcjonalność musi iść w parze z estetyką (wszak chcemy otaczać się przedmiotami pięknymi). Tę naturalną kolej rzeczy zauważyli już dawno producenci z branży komputerowej i to właśnie dzięki nim coraz to nowsze urządzenia wznoszą wzornictwo przemysłowe na jeszcze wyższy poziom. Jednym z istotnych elementów stylistyki współczesnych urządzeń są wszelkiego rodzaju elementy świetlne, które w dobie oświetlenia LED mogą przybierać bardzo wysmakowane formy. Prezentowany projekt to praktyczna implementacja (udokumentowanego metodami inżynierii wstecznej) protokołu ARGB Gen2, stosowanego do sterowania oświetleniem elementów obudowy komputerowej. Niewielki moduł jest przystosowany do współpracy z dostępnym na rynku sprzętem (np. wentylatorami z wbudowanym podświetleniem kompatybilnym ze standardem ARGB Gen2) – efekty świetlne zależą od wartości temperatury, za której pomiar odpowiada popularny termometr scalony DS18B20.

Energooszczędna strzałka LED

Jak wskazać gościom naszego lokalu prawidłowy kierunek? Strzałkami, napisami lub innymi oznaczeniami, przy pomocy porozklejanych kartek lub bardziej elegancko – na przykład ozdobnymi tabliczkami. A gdyby tak te kierunkowskazy mogły się uruchamiać na życzenie, po wciśnięciu przycisku i pokazywać kierunek ruchu za pomocą animacji? A dodatkowo pobierałyby znikomy prąd w stanie spoczynku? Prezentowany układ pokazuje animowaną strzałkę jednorazowo lub przez dłuższy czas, zależnie od wybranego wejścia. A po całej tej akcji usypia się, dzięki czemu pobiera prąd na poziomie zaledwie 0,4 µA. Można go zabudować w interaktywnej tablicy bez doprowadzania do niej zasilania, ponieważ nawet niewielka bateria CR2032 jest w stanie zapewnić jego wielokrotne uruchamianie.

Wykaz firm ogłaszających się w tym numerze „Elektroniki Praktycznej”

AKSOTRONIK....83
ASPEL....57, 64
BORNICO....37
COMPUTER CONTROLS....7
DACPOL....60
ELMAX....49
HAMMOND....5
IHP....42
NORDIC SEMICONDUCTOR....44
PHOENIX CONTACT....34, 40
SEMICON....25

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

Copyright AVTKorporacja Sp. z o.o.
03-197 Warszawa, ul. Leszczynowa 11

Projekty publikowane w „Elektronice Praktycznej”. mogą być wykorzystywane wyłącznie do własnych potrzeb. Korzystanie z tych projektów do innych celów, zwłaszcza do działalności zarobkowej, wymaga zgody redakcji „Elektroniki Praktycznej”. Przedruk oraz umieszczanie na stronach internetowych całości lub fragmentów publikacji zamieszczanych w „Elektronice Praktycznej”. jest dozwolone wyłącznie po uzyskaniu zgody redakcji. Redakcja nie odpowiada za treść reklam i ogłoszeń zamieszczanych w „Elektronice Praktycznej”.

eprasa.pl 4202fe84f2

Arrow MultiSoulution Day Poland

Warsaw,
September 10th, 2024

ARROW | arrow.com

Time	Track 1	Track 2
08.30–09.00	Registration	
09.15–10.00	Opening – Realising AI at the Edge	
10.00–10.45	Infineon	Zephyr OS.
10.45–11.00	Coffee Break	
11.00–11.45	Silabs.	FPGA
11.45–12.00	Coffee Break	
12.00–12.45	TE	NXP MCX
12.45–13.45	Lunch Break	
13.45–14.30	Wolfspeed	NVidia AI
14.30–14.45	Coffee Break	
14.45–15.30	Onsemi	BMS
15.30	Grill	

AIm the Power

Już 10 września od 9.00 zapraszamy na tegoroczne wydarzenie AMS, podczas którego:

- **2× ścieżka szkoleniowa**
- **showcase 40 producentów**
- **grill dla gości i producentów**

Info z pełną agendą wydarzenia + rejestracja:

eprasa.pl 4202fe84f2