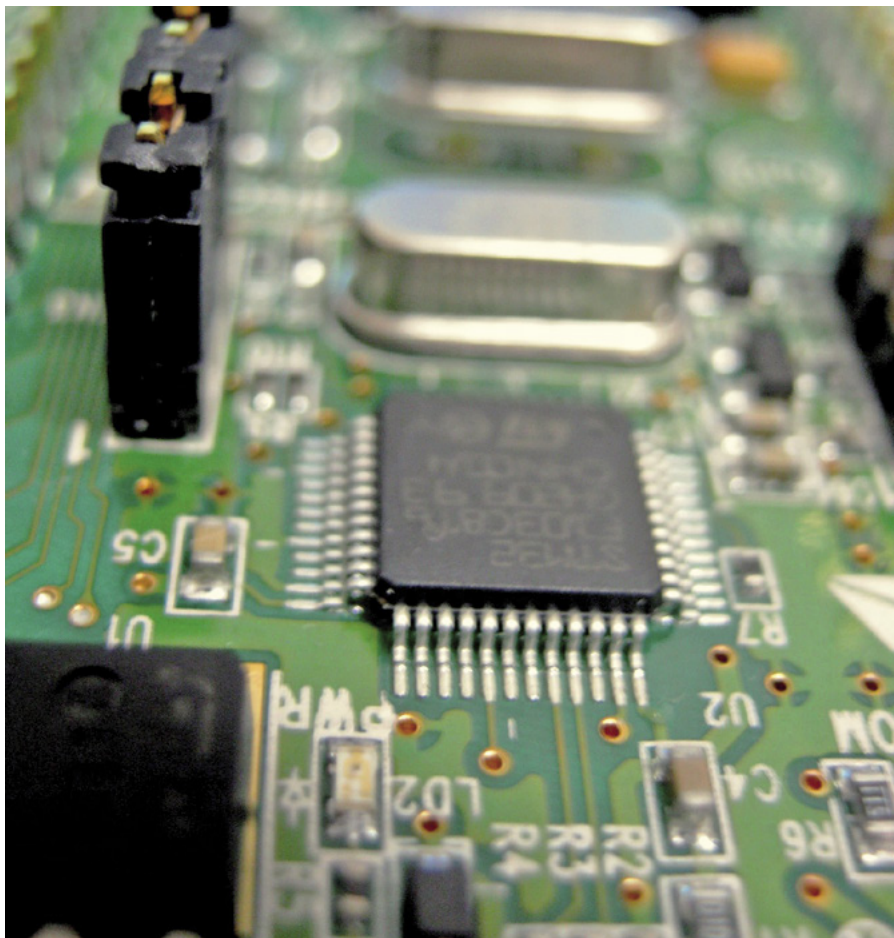


Interfejsy komunikacyjne w STM32F4

Najważniejszą nowością wdrożoną w mikrokontrolerach STM32F4 jest zastosowany w nich rdzeń: Cortex-M4F, charakteryzujący się możliwością taktowania do 168 MHz. Najlepszy i najszybszy rdzeń, żeby można było wykorzystać jego zalety w praktyce, musi mieć możliwość komunikowania się w otoczeniu. Mikrokontrolery STM32F4 w komunikacji są równie sprawne jak w DSP i obliczeniach zmiennoprzecinkowych, co wynika nie tylko z dużej liczby wbudowanych w nie wyspecjalizowanych interfejsów.

Mikrokontrolery STM32F4 są wyposażone we wszystkie popularne interfejsy komunikacyjne – jest ich łącznie 15 – uznawane obecnie za bazowy standard:

- trzy interfejsy I²C z automatyczną obsługą ramek SMBus i PMbus,
- trzy interfejsy SPI przystosowane do transmisji danych z prędkością do 37,5 Mb/s, które można skonfigurować do pracy w trybie I²S full-duplex (dostępne są dwa takie interfejsy),
- cztery interfejsy USART (z możliwością pracy w trybie klasycznych UART – wtedy użytkownik ma do dyspozycji dwa interfejsy) umożliwiające transfer danych z prędkością do 10,5 Mb/s,
- interfejs SDIO (*Secure Digital Input/Output Interface*) obsługujący karty MMC, SD oraz urządzenia z interfejsem CA-ATA,
- dwa interfejsy CAN2.0B Active,
- interfejs USB2.0 (Full Speed, obsługuje tryby: device, host oraz OTG) ze zintegrowanym interfejsem warstwy sprzętowej PHY,
- interfejs USB2.0 Full Speed/High Speed (obsługuje tryby: device, host oraz OTG) współpracujący z zewnętrznym PHY (poprzez interfejs ULPI),
- niektóre modele mikrokontrolerów STM32F4 (F417xx) wyposażono w sprzętowy interfejs warstwy MAC Ethernet 10/100 Mb/s.



I²C: wszyscy go lubią

Interfejs I²C wbudowany w STM32F4 umożliwia transmisję danych z maksymalną prędkością 400 kbit/s, może pracować w trybie *master* lub *slave*, w tym drugim z automatyczną detekcją adresu (można zadać dwa różne adresy dla lokalnego *slave*). Automatycznie jest obsługiwany tryb ramki 7- lub 10-bitowej, a także wywołanie *general call*. Logika interfejsu samoczynnie wykrywa znaki *start* i *stop*, w interfejsie zaimplementowano także system diagnostyczny wykrywający błędy transmisji na poziomie protokołu oraz przesyłanych danych (PEC – *Packet Error Checking*). Transmisję danych usprawniają: możliwość obsługi interfejsu z wykorzystaniem kanału DMA, a także synchronizacja jego obsługi z wykorzystaniem przerwań, z których jedno jest przeznaczone do sygnalizowania poprawnego zakończenia transmisji, drugie do sygnalizacji błędów.

Interfejs I²C obsługuje także specyficzne elementy protokołów transmisyjnych SMBus 2.0 oraz PMbus, łącznie z obsługą sieciowe-

go protokołu adresowania *Address Resolution Protocol* (ARP).



Rysunek 1. Bezprzewodowa karta radio-wa zgodna z SDIO



Rysunek 2. Aparat fotograficzny z interfejsem SDIO

USART: ciągle niezbędny

Interfejsy USART/UART są nadal jednymi z najczęściej stosowanych do komunikacji z modułami i urządzeniami dołączanymi do mikrokontrolerów w różnych aplikacjach. Są to m.in. podstawowe interfejsy komunikacyjne wykorzystywane przez moduły GSM, Bluetooth, GPS, ZigBee, Ethernet i wiele innych, tworząc możliwość wygodnego budowania systemów o rozproszonej inteligencji.

W mikrokontrolerach STM32F4 zastosowano dwa rodzaje interfejsów: UART (*Universal Asynchronous Receiver Transmitter* – UART4 i UART5) i USART (*Universal Synchronous-Asynchronous Receiver Transmitter* – pozostałe), które różnią się między sobą przede wszystkim tym, że UART-y nie obsługują sprzętowego trybu sterowania przepływem danych, nie obsługują kart *smartcard* i transferów synchronizowanych dodatkowym sygnałem zegarowym.

Interfejsy USART są przystosowane do obsługi transmisji w sieciach LIN, obsługują protokoły IrDA i SmartCard, umożliwiając transfer danych w kodzie NRZ (popularny w aplikacjach telekomunikacyjnych), można je wykorzystać także do budowania lokalnych sieci wieloprocesorowych z wyróżnionym urządzeniem *master*. Blok przesyłanych danych może mieć długość 8 lub 9

bitów, ramki mogą być elastycznie skonfigurowane w zależności od potrzeb i protokołu wykorzystywanego podczas transmisji. Interesującą cechą funkcjonalną jest możliwość transferu danych w trybie *half-duplex* z wykorzystaniem pojedynczej linii.

Logika interfejsów umożliwia kontrolę i wykrywanie wielu typowych błędów transmisji, co ułatwia diagnostykę komunikacji. Obsługę interfejsów ułatwia aż 10 przerwań wykorzystywanych do sygnalizacji wydarzeń związanych z komunikacją, a także możliwość blokowego transferu danych z wykorzystaniem kanałów DMA. Niebagatelnym atutem USART-ów są także zaawansowane preskalery zastosowane w torze taktującym, które pozwalają uzyskać niewielki błąd częstotliwości próbkowania bitów dla większości prędkości transmisji danych i różnych częstotliwości rezonatorów kwarcowych.

SPI: duża prędkość, łatwa obsługa

Interfejs SPI uchodzi za mniej ekonomiczną wersję I²C, co wynika z większej liczby linii niezbędnych do transmisji danych. W przypadku interfejsów wbudowanych w mikrokontrolery STM32F4 ta „wada” została zminimalizowana przez:

- możliwość simpleksowego transferu danych na dwóch liniach sygnałowych,
- dużą szybkość transmisji danych w klasycznym trybie SPI (*full-duplex* na trzech liniach) – może ona dochodzić nawet do 37,5 Mb/s!

Twórcy interfejsów SPI wbudowanych w STM32F4 przyjęli założenie, że ramka danych może mieć długość 8 lub 16 bitów, kolejność przesyłania bitów w ramce można ustalić począwszy od MSB lub LSB, zarządzanie linią selekcji nSS może się odbywać sprzętowo lub programowo, możliwe są różne konfiguracje faz zbroczy sygnału zegarowego i przesyłanych danych. Interfejsy SPI mogą spełniać rolę urządzeń *master* lub *slave*, możliwa jest także wymiana danych w trybie *multimaster*. Dzięki wbudowanemu generatorowi CRC transmisja danych (w obydwu kierunkach) może być chroniona przed błędami.



Nakładem Wydawnictwa BTC ukazała się książka pt. „Mikrokontrolery STM32 w sieci Ethernet w przykładach”, której autorem jest Marcin Peczański. Autor zawarł w niej kompletny, praktyczny wykład na temat działania interfejsu Ethernet, który został zilustrowany wieloma praktycznymi przykładami kompletnych rozwiązań sieciowych na mikrokontrolerach z rodziny STM32. Książka do nabycia w sklepie AVT: <http://sklep.avt.pl>, kod produktu: KS-110200

Interfejsy SPI można wykorzystać także – w trybie pracy I²S – w aplikacjach cyfrowego audio, na przykład do obsługi kodeków audio, wymiany danych z konsolami audio, lub współpracy z cyfrowymi mikrofonami. Interfejs I²S obsługuje częstotliwości próbkowania z zakresu 8...192 kHz, słowo danych może mieć długość 16, 24 lub 32 bitów, interfejs może pracować w trybie dupleksowym jak *master* lub *slave*. Obsługiwane są wszystkie standardowe protokoły wymiany danych: standardowy Philips I²S, *left-justified*, *right-justified* oraz PCM.

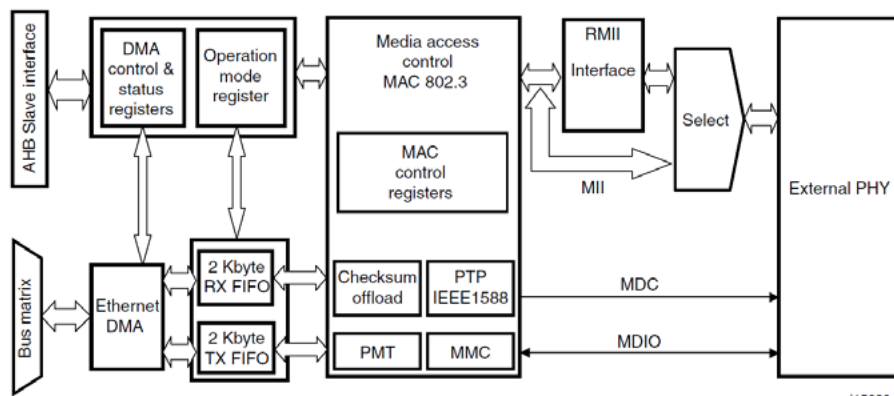
SDIO: „karciana” wygoda

Interfejs SDIO jest lekarstwem na większość problemów na jakie napotykają konstruktorzy i programiści stosujący w swoich opracowaniach karty pamięciowe Flash oraz urządzenia peryferyjne zgodne ze standardami:

- MMC v4.2 (z magistralą 1, 4 i 8 bitów, bez zgodności z trybem SPI),
- SDMC v2.0 (z magistralą 1 i 4 bity),
- CE-ATA v1.1.

Maksymalna prędkość transmisji danych z karty MMC w trybie 8-bitowym wynosi 48 Mb/s, obsługę transferu danych ułatwiają przerwy generowane przez wybrane przez użytkownika flagi, a także możliwość transmisji danych (buforowanych w pamięci FIFO) przez kanał DMA.

Użyteczność interfejsu SDIO ma szansę szybko się powiększać, ponieważ coraz więk-



Rysunek 3. Sprzętowy interfejs warstwy MAC

sza liczba producentów oferuje zaawansowane urządzenia peryferyjne z nim zgodne, są to m.in. karty sieciowe WiFi oraz ZigBee (fotografia 1), kamery i aparaty fotograficzne (fotografia 2), itp.

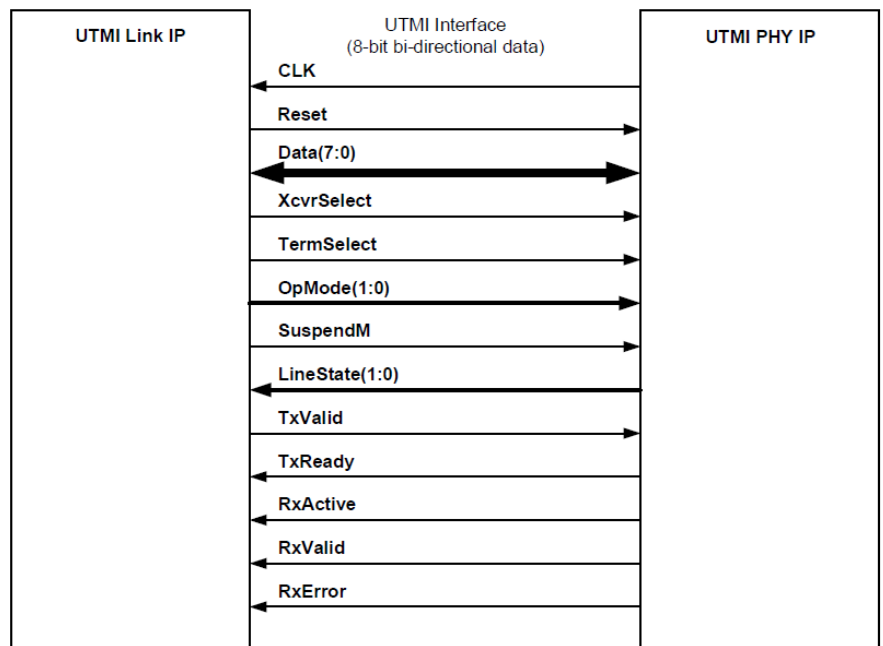
CAN: nie tylko dla aplikacji samochodowych

Interfejsy CAN (w nomenklaturze producenta: bxCAN od *Basic Extended CAN*) wbudowane w mikrokontrolery STM32F4 są zgodne z wersjami protokołów 2.0A oraz 2.0B, wyposażono je w bloki sprzętowe umożliwiające implementację rozwiązań bazujących na komunikacji TTC (*Time Triggered Communication*) z automatycznym znakowaniem czasowym przesyłanych wiadomości. Maksymalna prędkość transmisji wynosi 1 Mb/s, w torze nadawczym zaimplementowano 3 niezależne mailboxy z możliwością priorytetowania wysyłanych wiadomości. Interfejsy samodzielnie obsługują standardowe identyfikatory 11-bitowe oraz identyfikatory rozszerzone, o długości 29 bitów. Obsługę CAN ułatwiają 4 przypisane do interfejsu przerwania.

Ethernet MAC: bez sieci coraz trudniej

Niektóre typy mikrokontrolerów STM32F4 wyposażono w sprzętowy interfejs warstwy MAC (*Media Access Control* zgodnie z IEEE 802.3-2002 – rysunek 3) protokołu Ethernet, który we współpracy z zewnętrznym układem PHY (dołączanym za pomocą interfejsu MII lub RMII oraz – w celach konfiguracyjnych – SMI) obsługuje sieci o prędkości transferu danych 10 i 100 Mb/s. Integralną częścią interfejsu jest blok obsługi standardu dystrybucji i synchronizacji czasu w sieciach IEEE1588-2002 (PtP – *Precision Time Protocol*). Możliwości funkcjonalne MAC odpowiadają klasycznym wymogom standardu, jednostka obsługuje m.in. datagramy IPv4 i IPv6, komunikację *half-duplex* i *full-duplex*, kontrolę i generację CRC, wszystkie zalecane przez standard algorytmy filtrowania adresów, wybudzanie mikrokontrolera z trybu uśpienia poprzez sieć, a także elastycznie konfigurowaną długość ramki danych aż do 16 kB.

Podobnie jak i w przypadku innych bloków komunikacyjnych, także MAC komunikuje się z CPU za pomocą przerwania, a transfer danych może odbywać się za pośrednictwem DMA, bez konieczności długotrwałego angażowania CPU.



Rysunek 4. Uproszczony schemat blokowy interfejsu ULPI

USB 2.0: na dwa sposoby

Popularność interfejsu USB szybko rośnie, co zawdzięczamy m.in. jego dostępności w wielu współczesnych mikrokontrolerach, a także rosnącej łatwości jego obsługi, która wynika z dostępności coraz większej liczby bezpłatnych bibliotek zapewniających obsługę stosu USB.

Mikrokontrolery STM32F4 wyposażono w dwa interfejsy USB, z których jeden - *Full Speed* (umożliwia transfer danych z prędkością 1,5 Mb/s lub 12 Mb/s) – jest kompletny sprzętowo, mikrokontroler wyposażono bowiem w sprzętową warstwę fizyczną USB FS. Drugi wbudowany w mikrokontroler interfejs USB - *High Speed* (umożliwia transfer z prędkością do 480 Mb/s) – wymaga zastosowania zewnętrznego interfejsu warstwy fizycznej (na przykład popularnego układu USB3300 firmy SMSC), z którym komunikuje się za pomocą interfejsu ULPI (*UTMI Low Pin Interface*), którego uproszczony schemat blokowy pokazano na **rysunku 4**.

Obydwa interfejsy USB są przystosowane do pracy jako urządzenia *host* i *device*, mogą także pracować w trybie OTG (*On-the-Go*) z opcjonalną obsługą dodatkowej linii ID (identyfikującej rodzaj urządzenia dołączanego do interfejsu mikrokontrolera). Interesującą cechą interfejsów USB wbudowanych w mikrokontrolery STM32F4 jest możliwość monitorowania z zewnątrz chwili

rozpoczęcia transmisji ramki (SOF – *Start of Frame*), co bywa przydatne w aplikacjach audio, w których nie jest wykorzystywany izochroniczny transfer danych przez USB lub konieczna jest modyfikacja częstotliwości sygnału taktującego współpracujący z mikrokontrolerem podsystem audio.

Interfejsy USB są obsługiwane przez CPU z wykorzystaniem przerwania i DMA, wyposażono je także we własną pamięć RAM, służącą do buforowania przesyłanych danych.

Podsumowanie

W tym krótkim opisie przedstawiliśmy najważniejsze cechy i właściwości interfejsów komunikacyjnych wbudowanych w mikrokontrolery STM32F4. Są one niemal identyczne ze stosowanymi w bliźniaczych mikrokontrolerach z podrodziny STM32F2 (które to mikrokontrolery wyposażono w rdzenie Cortex-M3) i należą do nowszej generacji niż te, które są stosowane w mikrokontrolerach z podrodziny STM32F1 i STM32L. Ich aktualne implementacje krzemowe są niemal pozbawione niedociągnięć, a duża liczba przykładowych aplikacji udostępnianych przez producenta (także w pakiecie bibliotek CMSIS) powoduje, że rozpoczęcie samodzielnych działań z ich wykorzystaniem nie sprawi programistom większych trudności.

Andrzej Gawryluk

REKLAMA

<http://sklep.avt.pl>