

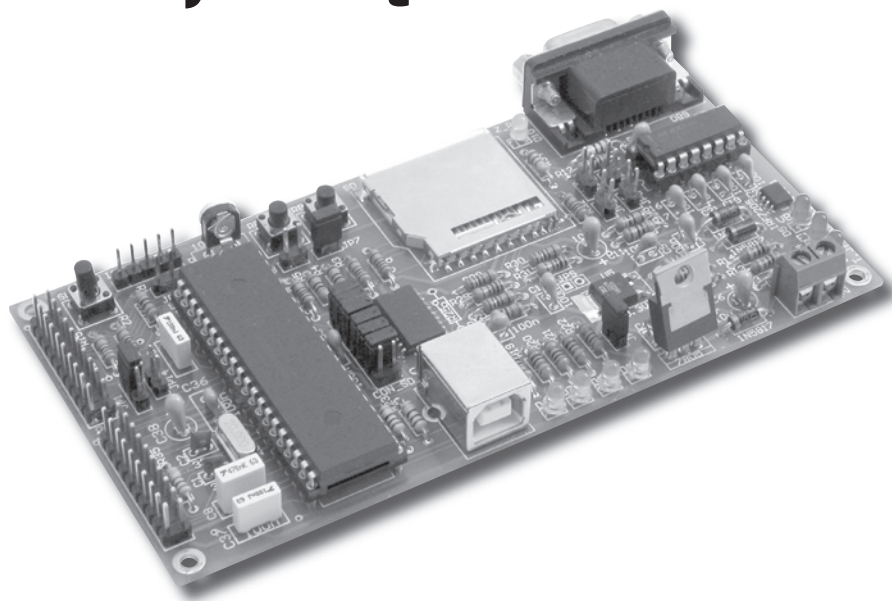
Zestaw uruchomieniowy USB z PIC18F4550, część 1

AVT-971

Interfejs USB jest obecnie najczęściej stosowanym portem do lokalnej komunikacji między komputerem, a urządzeniami zewnętrznymi. Pomimo dość skomplikowanego protokołu transmisji, coraz chętniej korzystają z niego również elektroniky-amatorzy, a to głównie za sprawą popularnych U-UART-ów, czyli konwerterów USB<->UART. Konstruktorzy, którzy chcą w pełni wykorzystać możliwości standardu USB mogą już sięgać po mikrokontrolery posiadające wbudowany interfejs USB.

Rekomendacje:

moduł został opracowany na podstawie materiałów firmy Microchip w celu dobrego poznania interfejsu USB zaimplementowanego w mikrokontrolerze PIC18F4550. Jego wykonanie, uruchomienie i przeprowadzenie prób polecamy doświadczonym elektronikom.



Kiedyś wpadł mi w ręce moduł peryferyjny od komputera wyprodukowanego bardzo dawno temu. Była to sporych rozmiarów płyta z wieloma układami TTL, tranzystorami i opornikami. Bardzo solidnie wykonana, umieszczona w odlewanej, aluminiowej ramie, z pięknymi złączami z połączonymi stykami. Po bliższych oględzinach okazało się, że jest to moduł podwójnego złącza szeregowego z konwerterem TTL/RS232. No tak, RS232 to modemy, myszka, programatory, komunikacja z modułami mikrokontrolerów, nawet drukarki, ale co to złącze robiło w komputerze z połowy lat 70. ubiegłego stulecia? RS232 formalnie został zdefiniowany w 1969 roku, ale powstał wcześniej i był szeroko stosowany w komputerach do komunikacji z mechanicznymi urządzeniami peryferyjnymi. Osobiście widziałem jak czytnik taśmy perforowanej z wydziurkowanym kodem programu wczytywał ten kod do pamięci operacyjnej komputera wielkości szafy na ubrania. Czytnik był dołączony do komputera przez RS232 i odczytywał dane z jakąś „zawrotną” prędkością. Innym znanym kiedyś urządzeniem był dalekopis firmy Teletype spełniający rolę monitora i klawiatury. Praca na nim polegała na wpisywaniu poleceń „on line” na czymś, co wyglądało jak stara elektryczna maszyna do pisania. Informacja zwrotna od komputera (na przykład wyniki obliczeń) była drukowana na papierze. Dalekopis był pod-

łączony do komputera złączem RS232 (przesyłano znaki ASCII), a transmisja miała prędkość 110 b/s.

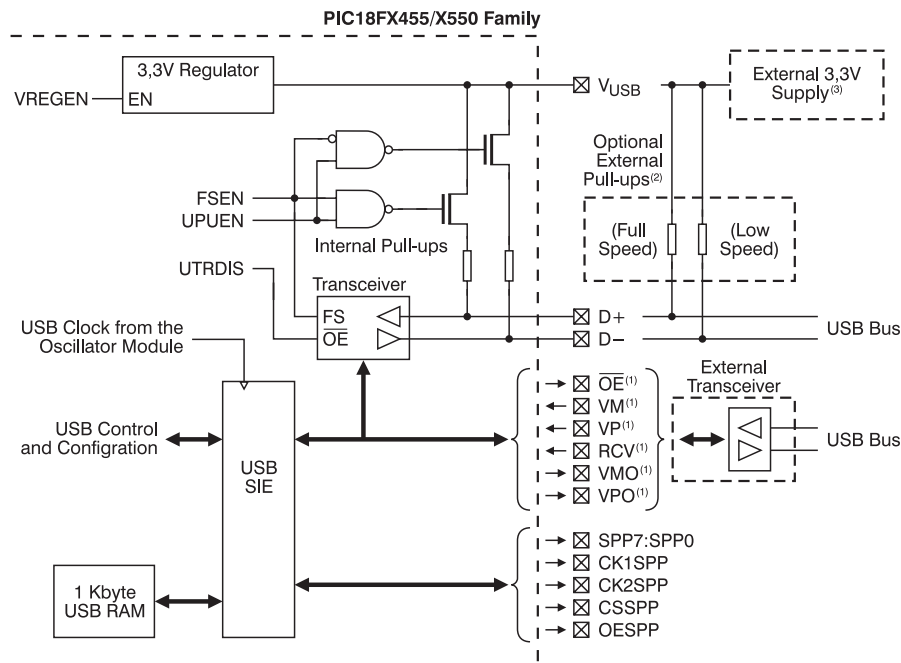
Mimo sędziwego wieku, standard RS232 miał się jeszcze do niedawna całkiem dobrze. Każdy komputer był wyposażony w 2 złącza COM1 i COM2 chętnie wykorzystywane przez konstruktorów do komunikacji z urządzeniami zewnętrznymi (na przykład z myszką), ale też z modułami mikrokontrolerów. Połączenie z tymi ostatnimi było o tyle łatwe, że większość mikrokontrolerów posiada wbudowany sprzętowy moduł transmisji szeregowej UART, a konwersję poziomów można wygodnie zrealizować używając kultowego układu MAX232.

Niewątpliwą zaletą standardu RS232 jest jego prostota, wadą natomiast ograniczenie prędkości transmisji i liczby kanałów. Wraz z rozwojem komputerów pojawiła się potrzeba przyłączenia nowych urządzeń wymagających dużej prędkości transmisji i dlatego zaczęto szukać innego sposobu wymiany informacji pomiędzy komputerem, a urządzeniami peryferyjnymi. W założeniu miało być to rozwiązanie umożliwiające znacznie szybsze przesyłanie danych, ale nie tylko. Przyjęto, że nowe rozwiązanie nazwane USB będzie również znacznie bardziej elastyczne i przyjazne dla zwykłego użytkownika.

Okazało się, że nowy standard jest tak atrakcyjny, że szybko się przyjął w komputerach klasy PC. Dzisiaj

PODSTAWOWE PARAMETRY

- Płytko o wymiarach 135x68 mm
- Zasilanie 9...16 VDC
- Dopuszczalny pobór prądu zasilającego z gniazda USB 100 mA
- Sygnalizacja stanu transmisji za pomocą diod LED
- Do wykorzystania na płycie:
 - Karta pamięciowa SD
 - Port RS232



Rys. 1. Schemat ogólny modułu USB

myszki, klawiatury, drukarki, pamięci Flash (popularne pen drive), dyski twarde dołączane są do komputera przez złącze USB. Popularne do niedawna interfejsy RS232 i równoległy Centronics często nie są już montowane w laptopach, a nawet w komputerach stacjonarnych. Nie ma się czemu dziwić, wystarczy mieć jedno gniazdo USB i dokupić tani hub, by bez problemu powielić możliwość podłączenia nowych urządzeń. Wbudowany mechanizm *plug-and-play* pozwala na dołączenie wielu typów urządzeń do komputera wyposażonego w najbardziej obecnie popularny system operacyjny Windows XP bez konieczności instalowania sterowników, albo instalowanie ich jest banalnie proste. Automatycznie są na przykład wykrywane bardzo popularne pamięci masowe Flash typu pendrive.

Elastyczność i wygoda USB powoduje, że wymiana informacji pomiędzy komputerem-hostem i urządzeniem końcowym USB jest zdecydowanie bardziej skomplikowana, niż to było w przypadku standardu RS232. To, co jest dobre dla przeciętnego użytkownika, który chce bez problemu korzystać z dołączanego urządzenia skomplikowało trochę życie konstruktorom sprzętu elektronicznego. Szybka transmisja i protokół wymiany oparty na rozwiązaniach sieciowych jest nie do zaakceptowania dla wielu popularnych 8-bitowych mikrokontrolerów. Bez specjalizowanego modułu sprzę-

towego nawet szybsze mikrokontrolery nie dają sobie z tym rady. Jak się należało spodziewać producenci będą się starali tę lukę jakoś zapełnić. Przykładem takich starań jest moduł USB wbudowany w mikrokontrolery PIC18F4550 firmy Microchip.

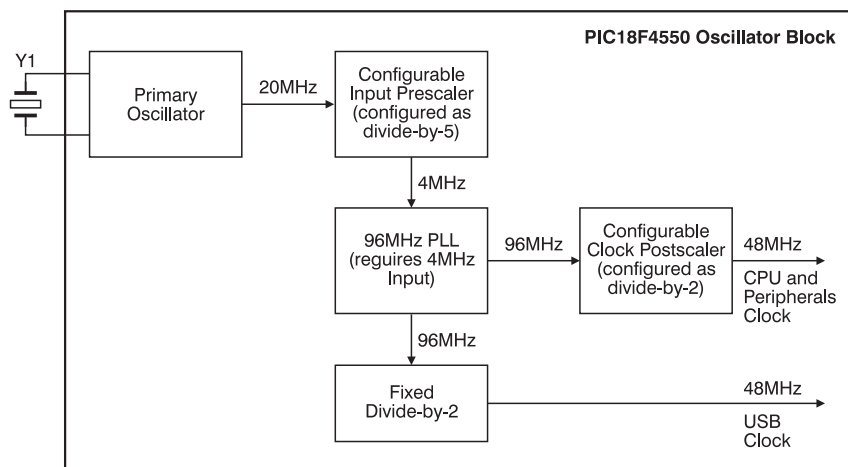
Moduł USB mikrokontrolera PIC18F4550

PIC18F4550 jest nowoczesnym, dobrze wyposażonym w moduły peryferyjne, 8-bitowym, dość szybkim mikrokontrolerem. Może być zasilany napięciem z zakresu 2,0...5,5 V. Przyjrzymy się bliżej wbudowanemu modułowi USB. Jest on zgodny ze standardem USB V2.0 i może pracować w trybach: *Low Speed* (1,5 Mb/s) i *Full Speed* (12 Mb/s).

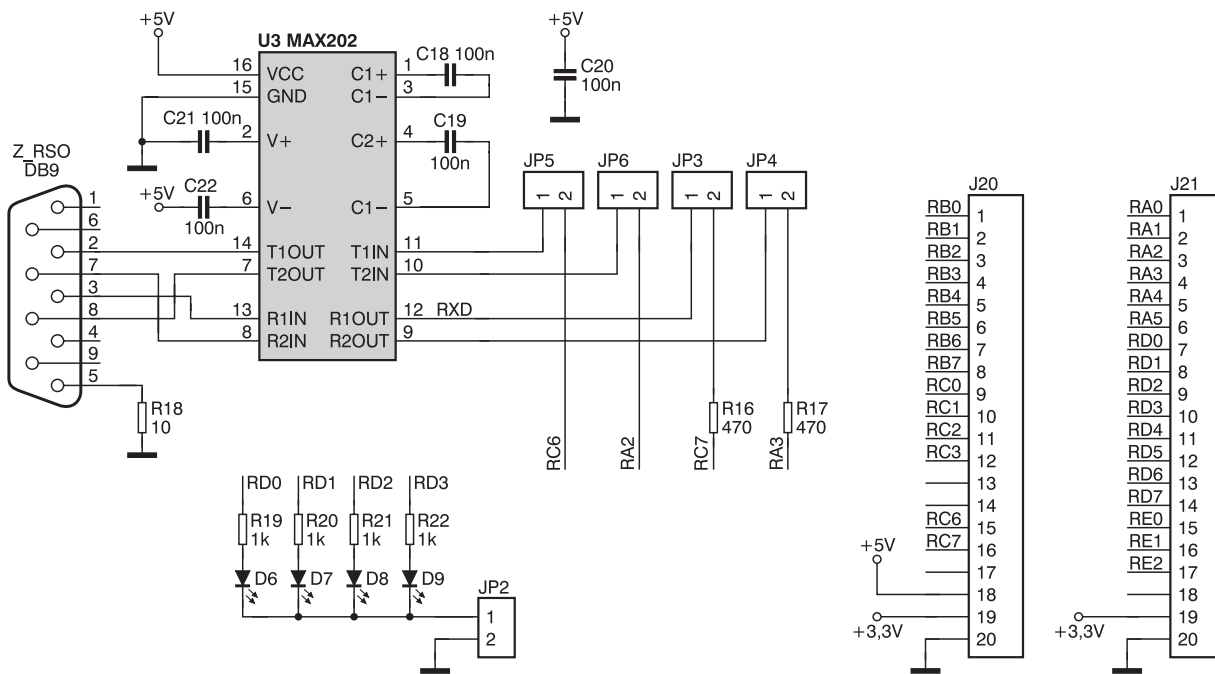
Symetryczny sygnał danych ze złącza USB jest podłączany bezpośrednio do wyprowadzeń D+ (RC5) i D- (RC4). Można również wykorzystać układ zewnętrznego transceiwer'a (rys. 1). Moduł ma wbudowaną własną, dwuportową pamięć RAM o pojemności 1 kB zamapowaną w bankach od 4 do 7 obszaru pamięci RAM GPR. Dostęp do tej pamięci ma moduł SIE USB i może do niej zapisywać i odczytywać dane bez udziału CPU. Jednocześnie CPU mikrokontrolera widzi ten sam obszar jako zwykły obszar GPR. Jeżeli moduł USB jest włączony, to bank RB4 nie powinien być wykorzystywany jako zbiór rejestrów ogólnego przeznaczenia.

W aplikacjach z wykorzystaniem USB, mikrokontroler jest zasilany napięciem +5 V, ale układy transceiwer'a muszą być zasilane napięciem +3,3 V dostarczanym z wewnętrznego stabilizatora lub przez linię VUSB. Źródło napięcia jest przełączane programowo. Konstrukcja modułu USB umożliwia też sprzętowe wsparcie przesyłania danych na zewnątrz poprzez interfejsy szybkiej komunikacji równoległej SPP (dwukierunkowa, równoległa, 8-bitowa szyna danych z sygnałami potwierdzenia).

Żeby uzyskać obie dostępne prędkości transmisji, mikrokontroler został wyposażony w rozbudowany system źródeł zegara taktującego, oparty na pętli PLL. W trybie *Low Speed* zegar taktujący modułem musi mieć częstotliwość 6 MHz, a w trybie *Full Speed* częstotliwość 48 MHz. Na rys. 2 pokazano system generowania sygnału zegarowego, gdy mikrokontroler jest taktowany oscylatorem z kwarcem o częstotliwości 20 MHz.



Rys. 2. System generowania zegara systemowego



Rys. 3. Schemat elektryczny płytki testowej cd.

kontroler PIC18F4550 moduł USB nie stał się tylko nic nieznaczącym gadżetem, w firmie Microchip przygotowano i udostępniono odpowiednie wsparcie w postaci modułu uruchomieniowego PICDEM FS USB i niezbędnego oprogramowania. Płytkę PICDEM FS USB można dostać u dystrybutora, ale ponieważ nie jest zbyt skomplikowana i co najważniejsze nie zawiera trudnych do zdobycia elementów, to można ją wykonać samodzielnie. Ponieważ byłem ciekawy jak Microchip poradził sobie z problemem, to postanowiłem zaprojektować na podstawie dokumentacji producenta własną wersję modułu PICDEM FS USB. Jedną z not aplikacyjnych pokazuje jak można wykorzystać połączenie płytki PICDEM FS USB z płytką PICTail, na której zostały umieszczone: złącze karty SD Flash i układy pomocnicze (bufor konwersji poziomów logicznych 5 V/3,3 V i stabilizator 3,3 V). Połączenie mikrokontrolera z wbudowanym modułem USB i z możliwością komunikacji z kartą pamięci SD wydało mi się na tyle atrakcyjne, że na mojej płytce zostały umieszczone połączone układy z obu płytek Microchipa: PICDEM FS USB i PICTail. Schemat ideowy płytki testowej pokazano na rys. 3.

Złącze USB może dostarczać napięcia zasilającego +5 V do urządzenia końcowego. Linie zasilającą można zawsze obciążyć prądem o natężeniu nie przekraczającym 100 mA (maksymalnie 500 mA). Układ zasilania płytki został tak pomyślany, żeby

można go było zasilac ze złącza USB lub z zewnętrznego źródła o napięciu 9...16 VDC. Kiedy do złącza ZAS zostanie podłączone źródło zewnętrzne, to zasilanie automatycznie przełączy się na to źródło. Dioda Schottky'ego 1N5817 (D1) zabezpiecza przed odwrotnym połączeniem zasilania. Stabilizowanego napięcia +5 V dostarcza scalony stabilizator U2 typu 7805 w standardowej obudowie TO-220. Napięcie wyjściowe zasila układ przez diodę Schottky'ego D2 i jednocześnie blokuje przewodzenie tranzystora MOSFET z kanałem typu P – IRF7205 (T1), odłączając przez to zasilanie układu z wyprowadzenia 1 złącza USB. Dioda LED – D4 sygnalizuje zasilanie zewnętrzne. Obecność napięcia +5 V na wyprowadzeniu złącza USB przy braku napięcia na złączu ZAS powoduje, że tranzystor T1 zaczyna przewodzić (polaryzacja bramki rezystorem R10) i układ jest zasilany napięciem z USB przez diodę D3. Jest to sygnalizowane świeceniem się diody LED – D5. Do sygnalizacji stanu układu zasilania używane są linie RA1 i RA2 mikrokontrolera U1. Linia RA1 kontroluje czy moduł jest zasilany ze złącza USB, a linia RA2 czy jest zasilany ze źródła zewnętrznego. Informacja ta jest wykorzystywana, kiedy host wyśle zapytanie o to, czy urządzenie jest zasilane ze złącza USB.

WYKAZ ELEMENTÓW

Rezystory

R18, R23, R24: 10 Ω
 R33, R34: 180 Ω
 R1, R4, R5, R7, R16, R17: 470 Ω
 R13, R14, R19...R22: 1 kΩ
 R11: 4,7 kΩ
 R2, R3, R6, R12, R15, R28...R32: 10 kΩ
 R10: 33 kΩ
 R9: 100 kΩ
 R35: 1 MΩ
 R8: potencjometr 10 kΩ

Kondensatory

C1, C2: 33 pF
 C3, C4, C7, C9, C10, C18...C22, C33, C35, C36, C37: 100 nF
 C8: 470 nF

C6, C38: 1 μF/16 V
 C5, C11: 10 μF/16 V
Półprzewodniki
 D1, D2, D3: 1N5817
 T1: IRF7205
 U2: 7805
 U5: SPX1117
 U4: 74LCX244
 U3: MAX202
 U1: PIC18F4550

Inne

Złącza IDC20 (goldpin) 2 szt.
 Złącze USB typu B
 Złącze karty SD/MMC
 Złącze DB9 do druku żeńskie
 Rezonator kwarcowy 20 MHz
 mikrosprzycisk

Układ zerowania mikrokontrolera jest dołączony do wyprowadzenia MCLR zwierając styki 1 i 2 zworki JP1. Po zwarceniu zworki JP14 dołączony jest kondensator C3 i po włączeniu zasilania wytwarzany jest impuls zerujący, którego czas trwania jest zależny od stałej określonej wartościami elementów R2 i C3. Kiedy układ RC nie jest wymagany, to zworka JP14 jest rozwarta, a konieczny stan wysoki na wejściu MCLR jest wymuszany rezystorami R1 i R2. Zworka ta musi być rozwarta w trakcie programowania mikrokontrolera w układzie poprzez złącze ICD.

Do linii portów RB4 i RB5 przez zworki JP7 i JP8 mogą być dołączone 2 mikrostyki S1 i S2 wykorzystywane przez firmowe aplikacje Microchipsa. Zworką JP9 można dołączyć napięcie regulowane przez potencjometr R8 do wejścia RA0 skonfigurowanego jako analogowe wejście przetwornika analogowo-cyfrowego. Diody LED – D1, D2, D3 i D4 są wykorzystywane przez firmowe Microchipsa do sygnalizacji stanu transmisji przez złącze USB. We

własnych aplikacjach można je wyłączyć rozwierając zworkę JP2.

Zwarcie zworki JP5, JP6, JP3 i JP4 dołącza linie wewnętrznego modułu USART mikrokontrolera do konwertera poziomów TTL/RS232 – układu U3 (MAX202). Linie interfejsu RS232 są podłączone do standardowych wyprowadzeń 9-pinowego złącza szufladowego Z_RSO.

Do połączenia modułu z hubem USB przeznaczone jest gniazdo USB typu B. Rezystory R23 i R24 stanowią dodatkowe zabezpieczenie przed zwarcieniem na linii danych. Dławik L1 ma za zadanie tłumić zakłócenia na linii zasilającej +5 V.

Do złącz J20 i J21 typu IDC20 doprowadzono wszystkie wyprowadzenia portów mikrokontrolera oraz linie zasilania +5 V i +3,3 V. Można to wykorzystać do połączenia układów zewnętrznych we własnych aplikacjach.

Wszystkie do tej pory opisane układy odpowiadają konstrukcji modułu PICDEM FS USB. Pomiędzy innymi tylko czujnik temperatury TC77. Pozostałe układy na płytce będą funkcjonalnym odpowiednikiem układów modułu

PICtail. Wymagania aplikacji USB powodują, że mikrokontroler musi być zasilany napięciem +5 V. Karty SD nie tolerują tak wysokiego napięcia. Standardowo większość kart wymaga napięcia zasilania +3,3 V i nie toleruje wyższych poziomów logicznych niż własne napięcie zasilania. Stabilizator LDO U5 dostarcza napięcia +3,3 V do zasilania karty i układu konwerterów poziomów 5 V/3,3 V układu U4 (74LCX244). Zworka JPS powinna być zwarta, gdy zastosujemy stabilizator o fabrycznie ustawionym napięciu wyjściowym +3,3 V. W przypadku stabilizatora o ustawianym napięciu wyjściowym, zworka JPS jest rozwarta, a napięcie wyjściowe jest ustalane rezystorami R36 i R37. Takie rozwiązanie może być przydatne, gdy karta jest zasilana napięciem innym niż 3,3 V. Sygnały sterujące pracą karty są dołączane do konwertera poziomów przez zwarcie styków złącza CON_SD. W aplikacjach nie używających karty, trzeba ją wyjąć ze złącza SD_CARD lub rozewrzeć styki CON_SD.

Tomasz Jabłoński, EP
tomasz.jablonski@ep.com.pl

www.sklep.avt.pl

HQPOWER™

oświetlenie efektowe
 oświetlenie dekoracyjne
 reflektory
 ruchome głowy
 lasery
 wytwornice dymu
 sterowniki DMX
 nagłośnienie

Zadzwoń lub napisz, otrzymasz gratis katalog

Detaliczna sprzedaż wysyłkowa.
 Dział Handlowy AVT 01-939 Warszawa, ul. Burleska 9,
 tel. 022 568 99 50, fax 022 568 99 55,
 lub w internecie: www.sklep.avt.pl e-mail: handlowy@avt.pl

AVT
 SOUND & LIGHT

Zestaw uruchomieniowy USB z PIC18F4550, część 2 AVT-971



Interfejs USB jest obecnie najczęściej stosowanym portem do lokalnej komunikacji między komputerem, a urządzeniami zewnętrznymi. Pomimo dość skomplikowanego protokołu transmisji, coraz chętniej korzystają z niego również elektronicy-amatorzy, a to głównie za sprawą popularnych U-UART-ów, czyli konwerterów USB<->UART. Konstruktorzy, którzy chcą w pełni wykorzystać możliwości standardu USB mogą już sięgać po mikrokontrolery posiadające wbudowany interfejs USB.

Rekomendacje:

moduł został opracowany na podstawie materiałów firmy Microchip w celu dobrego poznania interfejsu USB zaimplementowanego w mikrokontrolerze PIC18F4550. Jego wykonanie, uruchomienie i przeprowadzenie prób polecamy doświadczonym elektronikom.

PODSTAWOWE PARAMETRY

- Płytko o wymiarach 135x68 mm
- Zasilanie 9...16 VDC
- Dopuszczalny pobór prądu zasilającego z gniazda USB 100 mA
- Sygnalizacja stanu transmisji za pomocą diod LED
- Do wykorzystania na płytce:
 - Karta pamięciowa SD
 - Port RS232

Montaż i uruchomienie

Schemat montażowy płytki pokazano na **rys. 4**. Montaż nie jest skomplikowany, jedynym elementem SMD jest złącze karty SD i stabilizator U5. Mikrokontroler U1 proponuję umieścić w podstawce. W zmontowanym układzie najpierw powinien być sprawdzony układ zasilania bez włożonej karty SD. Kiedy jest pewność, że nie ma żadnych zwarcień można podłączyć zewnętrzne napięcie zasilania do złącza ZAS, zwracając uwagę na polaryzację. Jeżeli wszystko działa prawidłowo, to powinna się świecić dioda D4 (zasilanie zewnętrzne) i dioda D11 (obecność napięcia +3,3 V). Napięcie zmierzone na nóżce 11 mikrokontrolera ma wartość ok. 4,8 V (5 V-0,2 V spadku na diodzie D2), a na wyjściu stabilizatora U5 wartość +3,3 V. Odłączamy napięcie ze złącza ZAS i podłączamy moduł do portu USB komputera za pomocą kabla USB. Powinny się świecić diody D5 i D11. Po sprawdzeniu napięcia na nóżce 11 mikrokontrolera można włożyć zaprogramowany mikrokontroler do podstawki, lub zaprogramować go w układzie podłączając programator przez złącze ICD (układ musi wtedy być zasilany).

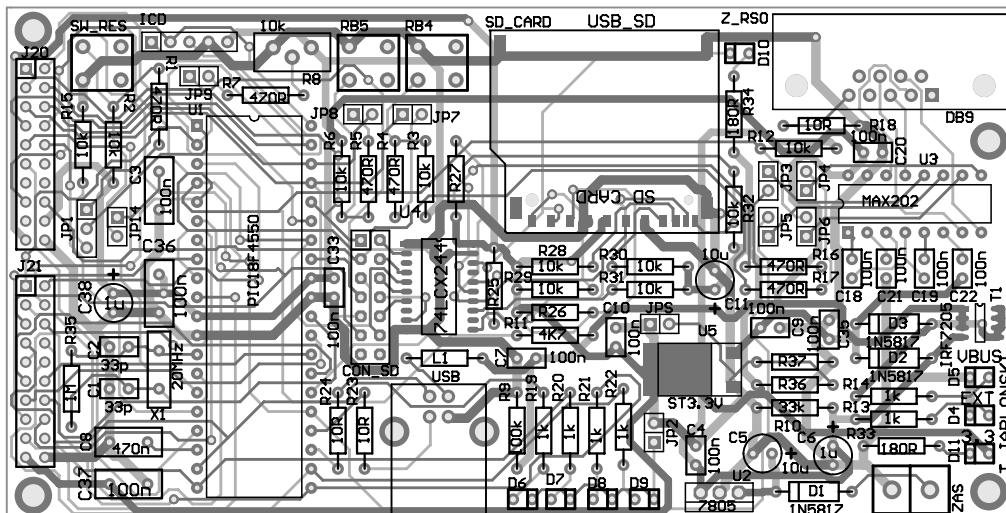
Programy testowe modułu PICDEM FS USB

Sprzedawany przez Microchip PICDEM FS USB jest fabrycznie zaprogramowany programem testowym umożli-

wiający sprawdzić działanie podstawowych aplikacji testujących działanie złącza USB. Na dołączonej w komplecie płytce CD zamieszczone są niezbędne pliki. W naszym przypadku dokumentację, pliki źródłowe i drivery musimy pobrać ze strony http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en021940&part=DM163025.

Na początek będzie potrzebna dokumentacja „PICDEM FS USB Demonstration Board User's Guide” i plik MCHPFSUSB_Setup.zip. Po jego rozpakowaniu i uruchomieniu wykonana zostaje instalacja polegająca na umieszczeniu w katalogu C:\MCHPFUSB wszystkich niezbędnych plików. W katalogu C:\MCHPFUSB\fw\factory_hex jest umieszczony plik *picdem-fsusb.hex* przeznaczony do zaprogramowania mikrokontrolera w fabrycznie dostarczonym module PICDEM FS USB. Tym plikiem trzeba zaprogramować mikrokontroler, żeby można było uruchomić testową aplikację przygotowaną przez producenta. Zaprogramowany moduł (bez karty SD) podłączamy standardowym kablem USB do portu komputera z zainstalowanym systemem operacyjnym Windows XP. Świecą się wtedy diody LED D5 (zasilanie z portu USB), D11 (napięcie +3,3 V), a diody D6 i D7 migają.

Prawidłowo działający moduł jest automatycznie wykrywany i system operacyjny rozpoczyna instalowanie dedykowanego sterownika. Po znalezieniu sprzętu trzeba podać ścieżkę dostępu driver'a: C:\MCHPUSB\PC\MCHPUSB Driver\Release - **rys. 5**. Kiedy driver jest zainstalowany i moduł jest połączony kablem z portem USB, można uruchomić testową aplikację



Rys. 4. Schemat montażowy płytki testowej

pdfsub.exe umieszczoną w katalogu C:\MCHPUSB\PC\PDSFUSB. Działanie programu jest podzielone na 2 tryby pracy: *Bootload Mode* i *Demo Mode*. Oba tryby pracy są w zasadzie oddzielnymi aplikacjami i zainstalowany driver po wykryciu każdej z nich działa inaczej. Okno trybu *Demo Mode* pokazane na rys. 6 otwiera się po kliknięciu na zakładkę *Demo Mode*. Można tu na bieżąco obserwować zmiany położenia suwaka potencjometru R8 i pomiaru temperatury z czujnika TC77, ale najpierw trzeba w okienku „Select PICDEM FS USB Board” wybrać *PICDEM FS USB* i kliknąć na belkę *Connect*. W naszym

module nie mamy czujnika TC77, ale położenie suwaka i wartość rezystancji jest wyświetlane prawidłowo w okienku „Potentiometer Display”. Można też testować zapalanie diod opisanych jako D3 i D4 w okienku „Toggle LEDs” – w module będą się zapalały i gasły odpowiednio diody D8 i D9. W trakcie pracy, diody D6 i D7 sygnalizują stany transmisji łącza USB. Pozytywny wynik testów wskazuje, że wszystkie elementy połączenia USB działają prawidłowo.

Możemy teraz sprawdzić działanie drugiej części programu testowego, czyli tryb pracy *bootloadera*. Żeby moduł wprowadzić w ten tryb, trzeba jednocześnie nacisnąć przycisk S1 (RB4) i przycisk zerowania SW_RES, a następnie puścić SW_RES przy przyciśniętym S1. Linia RB4 połączona z przyciskiem S1 jest jednocześnie połączona z wyjściem bufora U4 (linia Card detect karty SD), dlatego zworka łącząca linię RB4 z tym wyjściem w złączu CON_SD musi być rozwarta. System operacyjny wykryje nowe urządzenie i zainstaluje tę część sterownika, która odpowiada za jego obsługę. Połączenie nowej aplikacji z modulem następuje po wybraniu w oknie „Select PICDEM FS USB Board” opcji *PICDEM FS USB 0 (Boot)* – rys. 7. Stają się wtedy aktywne przyciski sterujące aplikacją:

- Load HEX File – ładuje wybrany plik z rozszerzeniem .HEX do bufora aplikacji *Bootload mode*. Musi to być plik wykonywanego programu i mieć format Intel HEX32. W dolnym oknie aplikacji wyświetlane są wykonywane czynności (tutaj ładowanie pliku) i zawartość bufora w formacie heksadecymalnym.

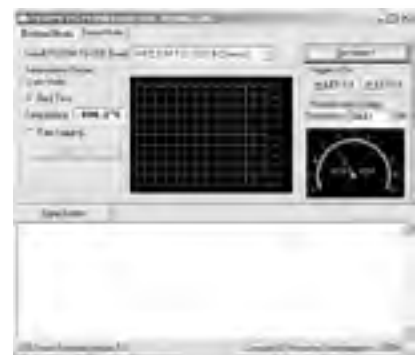


Rys. 5. Instalacja driver'a dla PICDEM FS USB

- Read Device – odczytuje zawartość pamięci mikrokontrolera, w tym również pamięci EEPROM i konfiguracyjnej. Kompletne, prawidłowe odczytanie jest potwierdzone komunikatem *MESSAGE – Read Completed*. Kiedy pojawiają się problemy, wysłany jest komunikat *WARNING – Failed to read*.
- Erase Device – kasuje ten fragment pamięci programu, w którym jest uruchamiany program (0x800...0x7FFF). Nie jest kasowana pamięć EEPROM, ID i konfiguracyjna.
- Exexute – wysyła komendę zerowania do programu

bootadadera – odpowiada ona zerowaniu mikrokontrolera. Jeżeli przycisk S1 nie jest zwarty, to zaczyna się wykonywać program użytkownika, ale kiedy S1 jest zwarty, to ponownie jest wprowadzany tryb *bootloadera*.

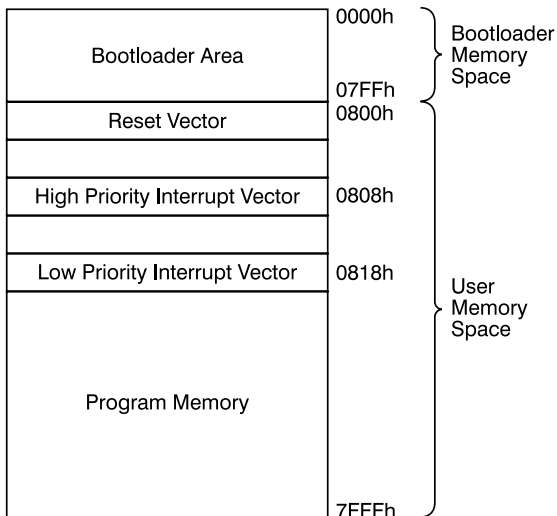
- Save to HEX File – zapisuje do pliku z rozszerzeniem .HEX zawartość bufora aplikacji *Bootload Mode*.
- Program Device – zapisuje zawartość bufora aplikacji *Bootload Mode* do pamięci mikrokontrolera. Programy pisane z myślą o uruchamianiu w trybie *bootloadera* mu-



Rys. 6. Tryb pracy *Demo Mode*



Rys. 7. Tryb pracy *Bootload Mode*



Rys. 8. Organizacja pamięci programu z zaimplementowanym bootloadere

szą podlegać pewnym ograniczeniom. Na rys. 8 pokazano mapę pamięci programu mikrokontrolera z zaimplementowanym bootloadere. Pierwsze 2 kB zajmuje program bootloadera zawierającego własne procedury obsługi USB i interpreter komend. Ten obszar ma włączoną protekcję zapisu. Ponieważ wektory zerowania i przerwań mikrokontrolera są umieszczone w zabezpieczonej przed zapisem przestrzeni pamięci programu, to konieczne stało się ich zamapowanie w obszarze programu użytkownika (adresy od 0x800 do 0x7FFF). Wektor zerowania jest umieszczony pod lokacją 0x800, a wektory przerwań pod lokacjami 0x808 (niższy priorytet) i 0x818 (wyższy priorytet). Gdy zgłaszane jest przerwanie o niskim priorytecie, to pod adresem 0x0008 umieszczony

jest rozkaz GOTO 0x0808 przekazujący sterowanie procedurą przerwania do nowego wektora. W krytycznych czasowo aplikacjach trzeba pamiętać, że czas od zgłoszenia przerwania do rozpoczęcia jego obsługi wydłuży się o czas trwania 2 cykli rozkazowych.

Przy takiej organizacji pamięci kod programu użytkownika musi się zaczynać od adresu 0x800. Skompilowany, relokowalny kod wynikowy kompilatora jest umieszczany w pamięci przez program linkera. W narzędziach firmy Microchip pracą linkera steruje specjalny plik tekstowy nazywany skryptem linkera. Przykład takiego skryptu przedstawiono na list. 1. Modyfikując ustawienia skryptu można zlecić linkerowi umieszczenie kodu wynikowego pod żadaną lokacją.

Obszar zdefiniowany jako *Boot* jest zarezerwowany na program bootloadera. Kolejne definicje obszarów to:

- Vectors – dla wektorów przerwań programu użytkownika
- Page – dla programu użytkownika
- Idlocs – dla obszaru ID użytkownika
- Config – dla obszaru bitów konfiguracyjnych
- Eedata – dla obszaru pamięci EEPROM

Oprócz modyfikacji skryptu linkera trzeba jeszcze dokonać zmiany w procedurze startowej c018i.c, określając nowy wektor zerowania mikrokontrolera,

a w procedurze obsługi przerwania muszą być wpisane zmodyfikowane wektory (list. 2).

Kompletna aplikacja gotowa do uruchomienia w trybie *bootload mode* została przygotowana przez Microchip i umieszczona w katalogu C:\MCHPFUSB\fw\hid\mouse. Jest to projekt napisany w języku C i przeznaczony do skompilowania kompilatorem MPALB C-18 w wersji V2.01 i wyższej. Po otwarciu pliku projektu w MPLAB IDE można przestudiować wersje źródłowe plików napisanych w C i skrypt linkera. Po skompilowaniu i linkowaniu, powstałym plikiem wynikowym można zaprogramować pamięć mikrokontrolera, wykonując polecenie Program Device przez aplikację bootloadera. Przykładowy program po uruchomieniu poleceniem Execute powoduje, że wskaźnik myszy na ekranie wykonuje ruchy obrotowe. Jeżeli chcemy przywołać mysz komputera do porządku, trzeba wykonać polecenia Erase Device.

Na początku zaprogramowaliśmy pamięć mikrokontrolera plikiem picdemfsusb.hex. Przeprowadzone próby pokazały, że plik jest przygotowany prawidłowo i wszystko działa poprawnie. Tylko, jaki jest z tego pożytek dla konstruktora chcącego wykorzystać moduł USB mikrokontrolera PIC18F4550 do własnych aplikacji? Otóż w katalogu C:\MCHPFUSB\fw\demo i C:\MCHPFUSB\fw\boot zostały umieszczone wszystkie wersje źródłowe programów obsługujących moduł USB napisanych w języku C i przeznaczonych do skompilowania kompilatorem MPALB C-18. Analiza zawartych tam procedur jest bardzo dobrym wstępem do napisania własnych aplikacji z wykorzystaniem modułu USB mikrokontrolera PIC18F4550. Dodatko-

List. 1 Przykład skryptu linkera MPLINK dla programu z bootloadere

```
//Sample linker command file for 18F4550 with Bootloader

LIBPATH .

FILES c0181.o
FILES c11b.lib
FILES p18f4550.lib

CODEPAGE NAME-boot START-0x0 END-0x7FF PROTECT
CODEPAGE NAME-vectors START-0x800 END-0x829 PROTECT
CODEPAGE NAME-page START-0x82A END-0x7FFF
CODEPAGE NAME-idlocs START-0x200000 END-0x200007 PROTECT
CODEPAGE NAME-config START-0x300000 END-0x30000D PROTECT
CODEPAGE NAME-devid START-0x3FFFFFFF END-0x3FFFFFFF PROTECT
CODEPAGE NAME-eedata START-0xF00000 END-0xF000FF PROTECT

ACCESSBANK NAME-accessram START-0x0 END-0x5F
DATABANK NAME-gpr0 START-0x60 END-0xFF
DATABANK NAME-gpr1 START-0x100 END-0x1FF
DATABANK NAME-gpr2 START-0x200 END-0x2FF
DATABANK NAME-gpr3 START-0x300 END-0x3FF
DATABANK NAME-usb4 START-0x400 END-0x4FF PROTECT
DATABANK NAME-usb5 START-0x500 END-0x5FF PROTECT
DATABANK NAME-usb6 START-0x600 END-0x6FF PROTECT
DATABANK NAME-usb7 START-0x700 END-0x7FF PROTECT
ACCESSBANK NAME-accessfr START-0xF60 END-0xFFFF PROTECT

SECTIONNAME-CONFIG ROM-config

SRACK SIZE-0x100 RAM-gpr3
```



Rys. 9. Wykrycie czytnika kart SD/MMC


```

List. 2. Modyfikacja wektorów przerwań
Extern void _startup (void)
// see c0191.c In your C18 compiler directory
#pragma code _RESET_INTERRUPT_VECTOR - 0x000800
Void _reset (void)
{
    _asm goto _startup _endasm
}
#pragma code

```

wą pomocą jest lektura dokumentu „PICDEM FS USB User's Guide”, a szczególnie jej czwarty rozdział „Using the Microchip USB Firmware Framework”. Trzeba jednak przyznać, że analizowany materiał nie jest łatwy i będzie wymagał od Czytelnika przynajmniej podstawowej wiedzy o strukturze transferu danych host – urządzenie USB.

Nota aplikacyjna AN1003

Oprócz oprogramowania przeznaczonego do testowania modułu PICDEM FS USB została również przygotowana przez Micro-

chip bardzo ciekawa nota aplikacyjna AN1003 „USB Mass Storage Device Using a PIC MCU”. Nota zawiera kompletną dokumentację i oprogramowanie przeznaczone również dla modułu PICDEM FS USB połączonego z modułem PICtail. Na płytce są umieszczone układy z obu modułów tak, by można było sprawdzić działanie oprogramowania zawartego w nocie aplikacyjnej.

Przed przystąpieniem do prób, sygnały sterujące kartą z portów mikrokontrolera muszą być połączone z liniami bufora 74LCX244 przez zwarcie

pinów złącza CON_SD. Ze strony http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1999&ty=&dt=§ion=&NextRow=&ssUserText=an1003&x=0&y=0 trzeba ściągnąć plik mchpmsd.zip. Po rozpakowaniu i uruchomieniu mchpmsd.exe w katalogu C:\MCHPMSD zostaną umieszczone: nota aplikacyjna w postaci dokumentu PDF i wszystkie pliki źródłowe. Mikrokontroler trzeba zaprogramować plikiem MassStorage.hex umieszczonym w katalogu C:\MCHPMSD\MSD_output. Plik wynikowy można wygenerować uruchamiając gotowy plik projektu MassStorage.mcp przeznaczony dla środowiska MPALB w wersji V7.20 i wyższej. Pliki muszą być kompilowane kompilatorem MPALB C-18. Płytkę z zaprogramowanym mikrokontrolerem staje się czytnikiem kart pamięci Flash MMC/SD. Po połączeniu kablem USB do komputera PC z zainstalowanym systemem operacyjnym Windows XP, czytnik jest automatycznie wykrywany i widziany jako dodatkowy dysk – (rys. 9). Można go formatować i zapisywać dane w formacie FAT16, FAT32 i NTFS. Nie ma potrzeby instalowania dodatkowego specjalnego drivera, bo system Windows XP używa standardowego drivera usbstor.sys.

Podobnie jak w przypadku poprzednio opisanych programów, można je przeanalizować, żeby ewentualnie wykorzystywać we własnych programach. Dodatkową, dużą pomocą będzie lektura noty aplikacyjnej AN1003 zamieszczonej w dokumencie DS01003A.

Podsumowanie

Płytkę testową i zaproponowane przez Microchip'a rozwiązania programowe na pewno ułatwią poznanie rozwiązań obsługujących standard USB V2.0. Jak już wspomniałem, materiał nie jest prosty do opanowania, ale producent układu wykonał olbrzymią pracę, żeby najpierw zachęcić do prób, a potem wykorzystywać w praktyce gotowe rozwiązania. Świadczy o tym udostępnienie wersji źródłowej z dość wyczerpującym opisem i dokumentacją modułów PICDEM FS USB, PICtail i noty aplikacyjnej AN1003. O tym, że firma będzie wspierać swoje rozwiązania USB świadczy fakt, że według nieoficjalnych informacji PIC18F4550 został wykorzystany w najnowszej konstrukcji znakomitego debugera/programatora ICD2 łączącego się z komputerem poprzez USB.

Tomasz Jabłoński, EP
tomasz.jablonski@ep.com.pl