

# 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

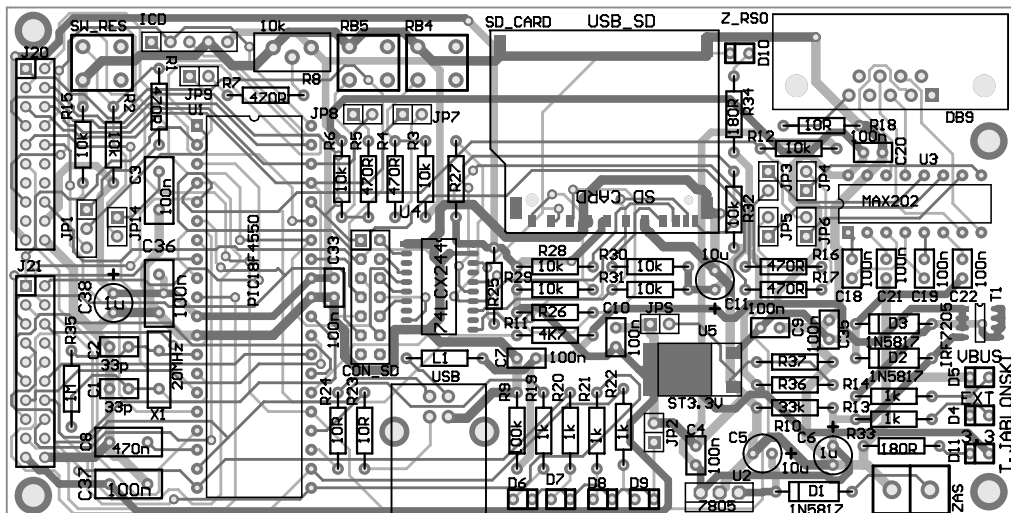
Schemat montażowy płytki pokazano na rys. 4. Montaż nie jest skomplikowany, jedynym elementem SMD jest złącze karty SD i stabilizator U5. Mikrokontroler U1 proponuję umieścić w podstawce. W zmontowanym układzie najpierw powinien być sprawdzony układ zasilania bez włożonej karty SD. Kiedy jest pewność, 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 umożliwiającym sprawdzenie 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](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

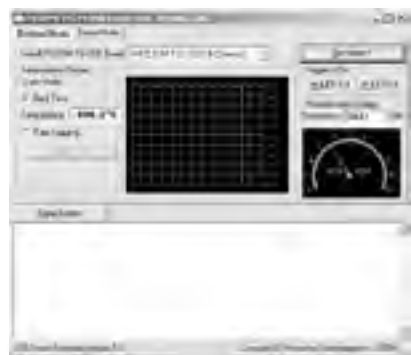
pdfsub.exe umieszczoną w katalogu C:\MCHPUSB\PC\PDSFUSB. Działanie programu jest podzielone na 2 tryby pracy: *Bootload Mode* i *Demo Mode*. Oba tryby pracy są w zasadzie oddzielnymi aplikacjami i zainstalowany driver po wykryciu każdej z nich działa inaczej. Okno trybu *Demo Mode* pokazane na rys. 6 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.

- 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-



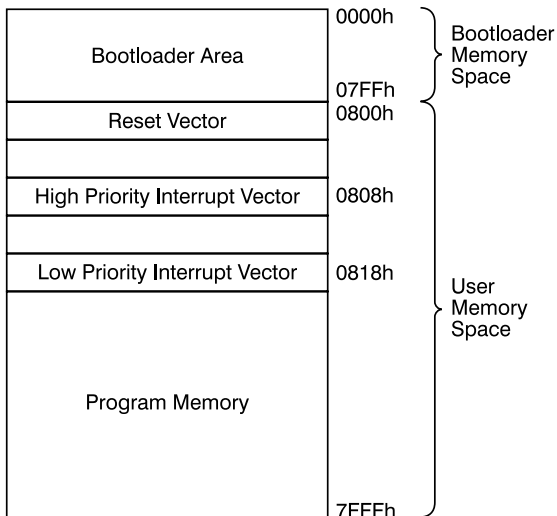
Rys. 6. Tryb pracy Demo Mode



Rys. 7. Tryb pracy Bootload Mode



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



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-0x3FFFFF END-0x3FFFFF 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 przerw
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 zwar-

pinów złącza CON\_SD. Ze strony [http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&nodeId=1999&ty=&dt=&section=&NextRow=&ssUserText=an1003&x=0&y=0](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1999&ty=&dt=&section=&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 wspominałem, 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 debuggera/programatora ICD2 łączącego się z komputerem poprzez USB.

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



**PDW MARTHEL**  
 WIĘCEJ NIŻ PROFESJONALNA  
 DYSTRYBUCJA

PDW MARTHEL  
 ul. Sosnowa 24-5  
 Bielany Wrocławskie  
 55-040 Kobierzyce  
 tel. +48 71 3110711, 12  
 fax +48 71 3110713

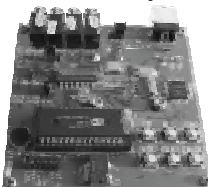
### Układy dźwiękowe serii ISD17xx firmy Winbond

Oferujemy układy scalone serii ISD17xx do zapisu i odtwarzania dźwięku, wykonane w unikalnej technologii nieulotnego zapisu wielopoziomowego (Multilevel Storage Technology). Trwałość zapisu 100 lat, 100 tys. cykli zapisu. Ze względu na niewielki pobór mocy i niski koszt stanowią idealne rozwiązanie dla urządzeń do powiadomień akustycznych, szczególnie dla urządzeń zasilanych bateryjnie.

- szeroki zakres napięcia zasilania 2,4...5,5 V
- programowana częstotliwość próbkowania 4...12 kHz
- zmienne pasmo zapisu 1,7... 5,1 kHz
- dwa tryby pracy:
  - autonomiczny ( interfejs przyciskowy)
  - z interfejsem szeregowym SPI
- możliwość wykonania wielu niezależnych nagrań w trybie sekwencyjnym lub adresowym oraz 4-ch niezależnych krótkich efektów dźwiękowych
- zapis z mikrofonu lub przez wejście audio
- wyjście audio do zewnętrznego wzmacniacza
- bezpośrednie wyjście głośnikowe PWM
- funkcja kasowania nagrań
- automatyczny tryb czuwania
- standardowy i przemysłowy zakres temperatury pracy
- obudowa bezolowiowa, zgodna z RoHS



- programowany czas zapisu:
  - ISD1730 - 20...60 s
  - ISD1740 - 26...80 s
  - ISD1750 - 33...100 s
  - ISD1760 - 40...120 s
  - ISD1790 - 60...180 s
  - ISD17120 - 80...240 s
  - ISD17150 - 100...300 s
  - ISD17180 - 120...360 s
  - ISD17210 - 140...420 s
  - ISD17240 - 160...480 s



Posiadamy również w ofercie zestaw uruchomieniowy ISD-ES17XX\_USB\_PB współpracujący z PC przez interfejs USB, oraz moduł demonstracyjny ISD-COB 1700

Ponadto oferujemy płytkę drukowaną, opracowaną w naszej firmie, uniwersalnego modułu do testowania układów serii ISD17xx, którą dodajemy bezpłatnie do zakupionych u nas detalicznych ilości tych układów.



**EBS** Renomowany producent drukarek INK-JET oferuje wysokiej klasy

**Aktywny detektor podczuwieni do zastosowań w układach automatyki i zabezpieczeń**

małe wymiary budowy (M18x1)  
 duża odporność na zakłócenia  
 wbudowany wskaźnik zadziałania  
 wyjścia odporne na zwarcie  
 wykonania PNP, NPN

**EBS Ink-Jet Systems Poland Sp. z o.o.**  
 ul. Główna 10, 50-012 Wrocław  
 tel. 071 37 04 11, fax 071 3 373 32 88