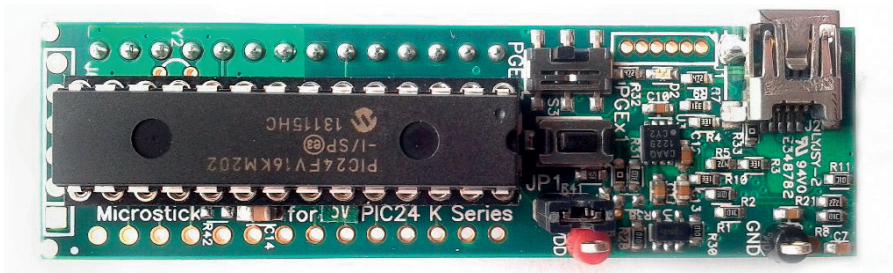


Nowy zestaw startowy Microstick dla 5-woltowych mikrokontrolerów PIC24



Urządzenia elektroniczne, w których pracują mikrokontrolery są dzisiaj bardzo powszechne. Jednak w odróżnieniu od konstruktora układów analogowych, impulsowych czy cyfrowych projektowanych „na piechotę” projektant używający mikrokontrolera musi być wyposażony w specyficzne narzędzia. Po pierwsze, musi dysponować środowiskiem programistycznym z edytorem tekstowym, kompilatorem języka programowania itd. Napisany program trzeba jakoś przesłać do mikrokontrolera, to jest potrzebny co najmniej programator wbudowanej pamięci Flash.

Na początku rozwoju techniki mikroprocesorowej pod pewnymi względami było łatwiej, ponieważ program był zapisywany do zewnętrznej pamięci EPROM, niezależnie od rodzaju mikrokontrolera. Wraz z pojawieniem się wbudowanych pamięci Flash sytuacja uległa skomplikowaniu, ponieważ każdy z producentów zastosował własny interfejs i algorytm programowania. Konstruktor decydujący się na jakąś rodzinę mikrokontrolerów zazwyczaj musiał kupić dość drogi programator i jeżeli dołożymy do tego jeszcze droższy kompilator, to kończyło to się przymusowym „programem lojalnościowym”. Początkowo producentom mikrokontrolerów było to na rękę, ale szybko się okazało, że ten, kto zaoferuje tanie lub łatwe do wykonania programatory, zyskuje dużą

popularność wśród amatorów i entuzjastów mikrokontrolerów, a w końcowym efekcie – wśród zawodowców. Wydaje się, że splot dwóch okoliczności: taniego i bardzo łatwego do wykonania programatora i pojawienie się programu Bascom było bezpośrednią przyczyną bardzo dużej popularności mikrokontrolerów AVR w Polsce.

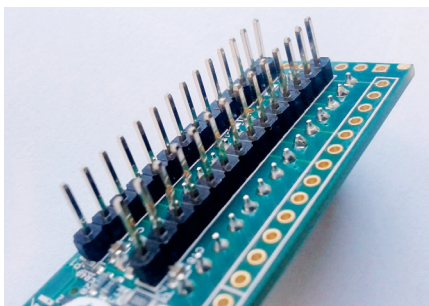
Firma Microchip, wielki konkurent Atmela, była i jest znana z narzędzi o bardzo dobrej jakości, w tym programatorów. Jednak w przeszłości ta jakość było związana z dość wysoką ceną. Od pionierskich czasów początków mikrokontrolerów wiele się zmieniło. Producenci mikrokontrolerów zrozumieli, że jeżeli dadzą konstruktorom tanie programatory, dostęp do testowych, ograniczonych wersji kompilatorów i gotowe tanie

moduły ewaluacyjne, to sprzedadzą więcej mikrokontrolerów. Przykładem takiej polityki marketingowej jest prezentowany tutaj minimoduł *Microstick for PIC24 5V K-series* oferowany przez Microchipa.

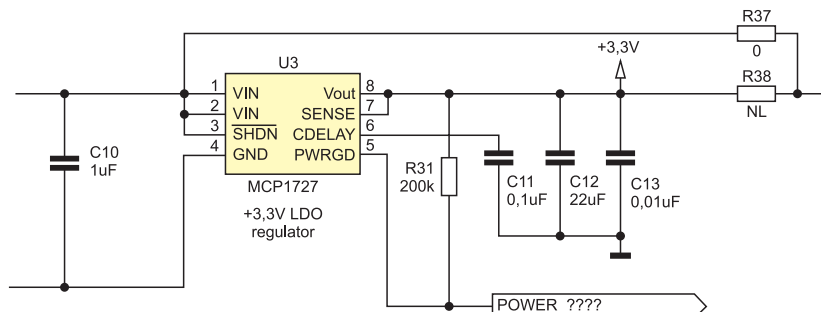
Idea jest bardzo prosta: dajemy płytkę podzieloną funkcyjnie na 2 części. W jednej jest podstawa, w której jest umieszczony mikrokontroler w obudowie do montażu przewlekane DIP 28, a w drugiej układ kompletnego programatora, który może pracować również jako debugger.

Wszystkie wyprowadzenia mikrokontrolera zostały wyprowadzone do dwu rzędów punktów lutowniczych (padów) rozmieszczonych w rastrze 2,54 mm, do których przyłutowano listwy goldpinów (fotografia 1). W ten sposób otrzymujemy przedłużenie wyprowadzeń mikrokontrolera w postaci goldpinów rozmieszczonych zgodnie ze standardem PDIP28. Microstick można dołączyć do układu docelowego zawierającego pozostałe układy np. wyświetlacz, klawiaturę, przekaźniki, obwody analogowe itp. przez wlutowanie w płytkę lub przez połączenie za pomocą popularnych gniazd z rastrem 2,54 mm, przystosowanych łącznie z listwami goldpinów. Z praktycznego punktu widzenia to bardzo wygodne rozwiązanie. Pozwala na użycie w płytce docelowej modułu Microstick zamiast mikrokontrolera, i po uruchomieniu urządzenia, na zamontowanie w jego miejsce mikrokontrolera w obudowie PDIP28. Jeżeli potrzebujemy typowego zestawu DIP28 (takiego, jak w starych pamięciach EPROM), to można sobie wlutować dodatkową listwę goldpinów w wolny rząd padów oznaczony P3.

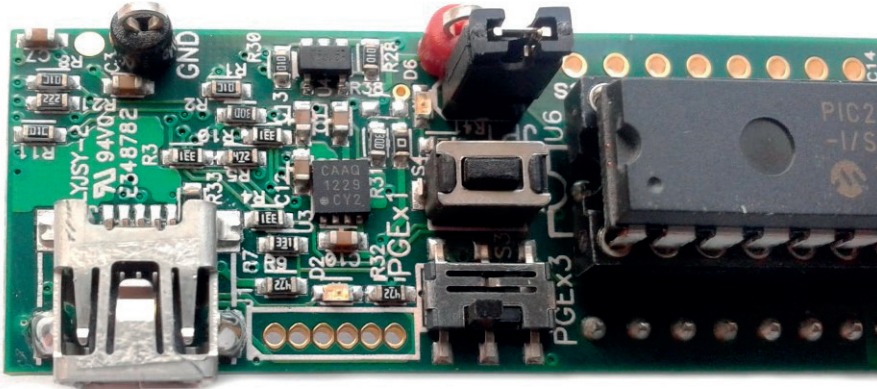
Programator/debugger umieszczony w drugiej części płytki jest zasilany, iden-



Fotografia 1. Listwy goldpinów



Rysunek 2. Układ zasilania mikrokontrolera



Fotografia 3. Układ debugera – widok od góry

tycznie jak i programowany mikrokontroler, ze złącza mini USB. Testowany przez mnie mikrokontroler był wykonany w wersji zasilanej napięciem do +5 V i być może dlatego jest bezpośrednio zasilany z napięcia +5 V dostępnego na złączu USB. Moduł ma też wbudowany stabilizator napięcia +3,3 V po to, aby można było zasilac i testować mikrokontrolery zasilane napięciem maksymalnym +3,3 V. Układ zasilania trzeba w tym przypadku „przekonfigurować” poprzez wylutowanie rezystora R37 (0 Ω) i wylutowanie go na pozycję R38 (rysunek 2).

Na rysunku 3 pokazano fragment płytki z częścią obwodów programatora/debugera. Jego zasadniczym elementem jest przylutowany do spodu mikrokontroler PIC24FJ256GB106. Mikrokontroler programatora/debugera łączy się z aplikacją sterującą MPALB IDE poprzez port USB z jednej strony i z programowanym mikrokontrolerem poprzez interfejs programujący ICSP z drugiej strony.

Interfejs ICSP Microchipa składa się z dwóch linii interfejsu szeregowego PGED i PGEC i może pracować w dwóch trybach: z programowaniem wysokonapięciowym i niskonapięciowym. Przy programowaniu wysokonapięciowym interfejs ICSP jest aktywowany przez podanie napięcia na wejście zerowania MCLR o wartości ok. +8...9 V (nowe konstrukcje) lub +12...13 V (stare konstrukcje mikrokontrolerów). Jest to bardzo rozpowszechniona metoda, ale wygenerowanie napięcia wyższego niż napięcie zasilające mikrokontroler jest zawsze kłopotliwe, bo wymaga wykonania przetwornicy napięcia, przez co programator jest bardziej rozbudowany, trudniejszy do zbudowania i droższy.

Od jakiegoś czasu Microchip równolegle stosuje programowanie niskonapięciowe wymagające aktywnego wejścia MCLR. Programator w Microsticku wspiera tylko programowanie niskonapięciowe. Dlatego zaleca się, by bity konfiguracyjne w trakcie używania programatora Microsticka konfigurowały linię MCLR/RA5 jako wejście MCLR, a nie jako linię GPIO (bit MCLRE w rejestrze konfiguracyjnym FPOR). W przeciwnym wy-

padku programowanie będzie możliwe tylko przez zewnętrzne programatory typu PICkit3 lub MPLAB ICD3 w trybie wysokonapięciowym. Pozwala ona na zaprogramowanie mikrokontrolera niezależnie od konfiguracji linii MCLR/RA5.

Ponieważ mikrokontroler PIC24FV15KM202 ma dwa porty ICSP, to zastosowano przełącznik mechaniczny pozwalający na przełączanie sygnałów PGED i PGEC z programatora na wejścia PGED0 i PGEC0 lub PGED1 i PGEC1. To, który port ICSP mikrokontrolera jest aktualnie aktywny określają bity FICD1:FICD0 w rejestrze konfiguracyjnym FICD.

Wyposażenie Microsticka w układy peryferyjne ogranicza się do jednej diody LED podłączanej do linii RA0 za pomocą typowego jumpiera JP1. Poza tym jest zamontowany układ RC wytwarzający sygnał zerujący po włączeniu zasilania. Przewidziano również miejsce na wlotowanie oscylatora kwarcowego o częstotliwości 32,768 kHz. Jest on połączony z liniami RB4/SOSCI i RA4/SOSCO.

Ze strony producenta można ściągnąć projekt przeznaczony dla środowiska MPLAB X z programem demonstracyjnym.

Listing 1. Ustawienie bitów konfiguracyjnych programu testowego

```

_FBS
(
  BWRP_OFF & // Boot Segment Write Protect (Disabled)
  BSS_OFF // Boot segment Protect (No boot flash segment)
)

_FGS
(
  GWRP_OFF & // General Segment Flash Write Protect (General segment may be
  written)
  GCP_OFF // General Segment Code Protect (No Protection)
)

_FOSCSSEL
(
  FNOSC_FRCDIV & // Oscillator Select (8MHz FRC with Postscaler (FRCDIV))
  SOSCSRC_ANA & // SOSC Source Type (Analog Mode for use with crystal)
  LPRCSEL_HP & // LPRC Power and Accuracy (High Power/High Accuracy)
  IESO_ON // Internal External Switch Over bit (Internal External
  Switchover mode enabled (Two-speed Start-up enabled))
)

_FOSC
(
  POSCMOD_NONE & // Primary Oscillator Mode (Primary oscillator disabled)
  OSCIOFNC_IO & // CLKO Enable Configuration bit (CLKO output signal enabled)
  POSCFREQ_MS & // Primary Oscillator Frequency Range Configuration bits
  (Primary oscillator/external clock frequency between 100kHz to 8MHz)
  SOSCSSEL_SOSCHP & // SOSC Power Selection Configuration bits (Secondary
  Oscillator configured for high-power operation)
  FCKSM_CSECMC // Clock Switching and Monitor Selection (Clock Switching and
  Fail-safe Clock Monitor Enabled)
)

_FWDT
(
  WDTPS_PS32768 & // Watchdog Timer Postscale Select bits (1:32768)
  FWPSA_PR128 & // WDT Prescaler bit (WDT prescaler ratio of 1:128)
  FWDTEN_OFF & // Watchdog Timer Enable bits (WDT disabled in hardware;
  SWDTEN bit disabled)
  WINDIS_OFF // Windowed Watchdog Timer Disable bit (Standard WDT selected
  (windowed WDT disabled))
)

// Warning:
// Always enable MCLRE_ON config bit setting so that the MCLR pin function will
// work for low-voltage In-Circuit Serial Programming (ICSP). The Microstick
// programming circuitry only supports low-voltage ICSP. If you disable MCLR
pin
// functionality, a high-voltage ICSP tool will be required to re-program the
// part in the future.
_FPOR
(
  BOREN_BOR3 & // Brown-out Reset Enable bits (Enabled in hardware; SBOREN
  bit disabled)
  PWRTEN_ON & // Power-up Timer Enable (PWRT enabled)
  I2C1SEL_PRI & // Alternate I2C1 Pin Mapping bit (Default SCL1/SDA1 Pins for
  I2C1)
  BORV_V18 & // Brown-out Reset Voltage bits (Brown-out Reset at 1.8V)
  MCLRE_ON // MCLR Pin Enable bit (RA5 input disabled; MCLR enabled)
)

_FICD
(
  ICS_PGx3 // ICD Pin Placement Select (EMUC/EMUD share PGC3/PGD3)
)

```

Program to może zbyt duże słowo, bo jest to prosta procedura powodująca miganie diodą LED zamontowaną na module. Testy działania tego programu rozpoczynamy od dołączenia modułu do portu USB, uruchomienia MPLAB X i otwarcia projektu testowego. Aby MPLAB X mógł łączyć się przez USB z programatorem/debugerem z Microsticka, trzeba wybrać w oknie *Hardware Tools* właściwości projektu (*File -> Project properties -> Hardware Tool -> Microchip Starter kits -> PKOB*). Jak już wspominałem, program jest trywialny, ale na początek warto przeanalizować ustawienie bitów konfiguracyjnych, łącznie z komentarzami. Może to nas uchronić przed przykrymi niespodziankami w czasie użytkowania mikrokontrolera tego typu. Odpowiedni fragment programu pokazano na **listingu 1**. Najważniejsze z nich to bity konfiguracyjne taktowanie mikrokontrolera i konfiguracyjne zerowanie mikrokontrolera, czyli wejście MCLR i timer PWRT.

Rodzina mikrokontrolerów PIC24Fxxxx może być taktowana z 4 źródeł:

- Głównego oscylatora (*Primary Oscillator*) stabilizowanego kwarem dołączonym do wyprowadzeń OSC1 i OSCO.
- Pomocniczego oscylatora (*Secondary Oscillator*) stabilizowanego kwarem dołączonym do wyprowadzeń SOSCI i SOSCO.
- Szybkim wewnętrznym oscylatorem RC (FRC) o dwóch częstotliwościach: 8 MHz i 500 kHz (*Low Power FRC*).
- Wewnętrznym oscylatorem RC o małym poborze mocy LPRC.

W naszym wypadku jest wybrany oscylator FRC o częstotliwości 8 MHz (*High Power i High Accuracy*). Oscylator główny jest wyłączony, a pomocniczy ustawiony w tryb low Power.

Konfiguracja zerowania mikrokontrolera musi zawierać definicję przypisania wyprowadzenia MCLR/RA4 do wejścia układu zerowania: MCLR odblokowany RA4 zablokowany. Jak już wspominałem, uaktywnienie RA4 spowoduje zablokowanie możliwości programowania mikrokontrolera przez programator wbudowany w Microsticka. Układ BOR jest ustawiany na zerowanie mikrokontrolera przy spadku napięcia zasilającego poniżej 1,8 V, a watchdog standardowo jest wyłączony. W czasie testów działania początkowo wykorzystałem program pokazany na **listingu 2**, dostarczany przez producenta.

Na **listingu 3** zamieszczono procedurę inicjowania Timera 1. Jest ona o tyle ważna, że przerwania są obsługiwane przez procedurę T1Interrupt pokazaną na **listingu 4**. Opóźnienie pomiędzy cykliczną zmianą stanu linii portu RA0 jest realizowane programowo w pętli. Po kliknięciu na ikonę *Make and Program Device* program jest kompilowany i jeśli nie ma błędów, to plik wynikowy służący do zapisania w pamięci mikro-

Listing 2. Fragment firmowego programu demonstracyjnego

```
while(1)
{
    for(i = 0; i < 65535; i++) //programowe opóźnienie
    {
        Nop();
        Nop();
        Nop();
        Nop();
    }

    LATAbits.LATA0 ^= 1; //cykliczna zmiana stanu linii portu RA0
}
```

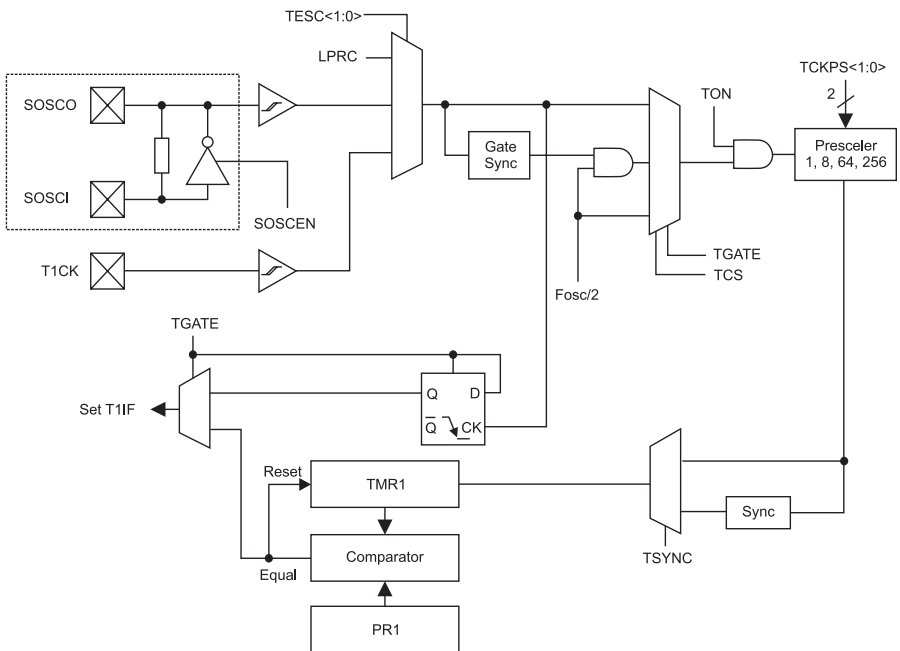
Listing 3. Inicjalizacja licznika Timer1

```
void InitTimer1(void)
{
    T1CONbits.TCS=0; //internal clock =FCS/2=4MHz
    T1CONbits.TCKPS=1; //4MHz/8=250kHz
    T1CONbits.TON=1; //Timer1 ON
    PR1=0x1f4; //PR1=500
    IEC0bits.T1IE=1; //Timer1 Interrupt enable
}
```

Listing 4. Procedura obsługi przerwania od wyzerowani aTimer1

```
#define DELAY_MS 200
int msec;

void __attribute__((__interrupt__, __auto_psv__)) _T1Interrupt( void ){
    IFS0bits.T1IF = 0; //zerowanie flagi przerwania
    --msec;
    if(msec<=0){
        LATAbits.LATA0 ^= 1;
        msec = DELAY_MS;
    }
}
```



Rysunek 4. Schemat blokowy modułu Timer1

kontrolera jest automatycznie przesyłany do pamięci Flash.

Debugowanie uruchamia się z menu *Debug -> Debug Projects*. MPLAB X automatycznie kompiluje program z opcją *Debug*.

Następnie modyfikowałem ten krótki programik tak, aby opóźnienie było odliczane przez zliczanie przerwań od zerowania licznika Timer1. Timer1 jest 16-bitowym licznikiem/timerem mającym możliwość zliczania impulsów z oscylatora pomocniczego (*Secondary Oscillator*), z wejścia T1CK lub impulsów o częstotliwości oscylatora taktującego rdzeń podzielonej przez 2. Schemat

blokowy Timera 1 pokazano na **rysunku 4**. Licznik zlicza do przodu i po osiągnięciu wartości równej wartości zapisanej do rejestru PR1 jest zerowany i jednocześnie jest ustawiana flaga przerwania T1IF.

Rdzeń mikrokontrolera jest taktowany oscylatorem RC o częstotliwości 8 MHz, zatem na wejściu modułu licznika mamy impulsy o częstotliwości 8 MHz/2=4 MHz. Po podzieleniu przez 8 w prescalerze, na wejściu rejestru licznika mamy przebieg o częstotliwości 500 kHz. Żeby licznik zgłaszał przerwanie co 1 ms, czyli z częstotliwością 1 kHz, do PR1 trzeba wpisać wartość 500

(500 kHz/500=1 kHz). Na list. 4 zamieszczono procedurę inicjalizacji licznika, tak aby zliczał impulsy o częstotliwości $F_{osc}/2/16$ i zgłaszał przerwanie po każdym wyzerowaniu.

W każdym przerwaniu jest dekrementowana zmienna *msek*. Kiedy *msek* się wyzeruje, to automatycznie jest ładowana wartością zdefiniowaną w DELAY_MS i zmienia się stan linii RA0 na przeciwny. W czasie testów stan linii RA0 zmieniał się na przeciwny dokładnie co 200 ms.

Podsumowanie

Zestaw Microstick for PIC24 5V K-series to dobrze pomyślany i wykonany minimoduł startowy. Jego główne zalety, to:

- małe wymiary,
- wbudowany programator/debuger,
- możliwość pracy z mikrokontrolerami PIC24 w obudowie PDIP28 zasilanymi napięciem +3,3 V i +5 V,
- bezproblemowa współpraca z pakietem MPLAB X.

Przeprowadzone testy pokazały, że od momentu wyjęcia z pudełka do napisania pierwszego działającego programu nie upływa zbyt wiele czasu. Wszystko działa od razu, a do dyspozycji mikrokontroler, programator i przykładową konfigurację bitów ustalających sposób funkcjonowania mikrokontrolera. Do tego bezpłatny pakiet MPLAB X i ewaluacyjną wersję kompilatora XC-16 z ograniczeniem optymalizacji i bez ograniczenia wielkości kodu wynikowego!

Jednak są i wady. Po pierwsze, wbudowanego programatora/debugera nie można użyć wprost jako programatora układów wlotowanych na płytce. Producent nie przewidział dodatkowego złącza z wyprowadzonymi sygnałami portu ICSP. Może nie jest to zbyt duża uciążliwość, bo zawsze można wyjąć układ z podstawki Microsticka i w jego miejsce włożyć drugą podstawkę z przylutowanymi przewodami. Również cena nie nastraja zbyt optymistycznie. Moduł można kupić w *Microchip Direct* w cenie ok. 30\$. Znając nasze realia, cena zakupu w Polsce nie będzie niższa niż 120...140 złotych. W porównaniu z konkurencyjnymi modułami, na przykład dla mikrokontrolerów STM32 z wbudowanym programatorem/debugerem STLink-2, jest to 2 razy drożej.

Jednak trzeba sobie zdawać sprawę, że taka cena dla zastosowania komercyjnego nie jest wysoka. Jeżeli potrzebujemy takiego mikrokontrolera do projektu, to zakup Microsticka spowoduje, że czas potrzebny do przygotowania prototypu można skrócić tak bardzo, jak to tylko możliwe. A to już olbrzymia oszczędność, bo czas pracy programisty jest zwykle bardzo drogi.

Tomasz Jabłoński, EP

REKLAMA

Regulowany zasilacz uniwersalny 1,5...32 V/3 A

AVT 1731



Zasilacz to aplikacja popularnego układu LM338, w obudowie którego umieszczono praktycznie wszystkie elementy regulatora napięcia wysokiej klasy.

www.sklep.avt.pl

ZAJRZYJ NA TE STRONY

