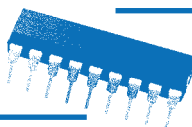


Mikrokontroléry PIC (19)



První projekt s mikrokontrolérem PIC16F88 – blikající LED

V minulém dílu jsme se seznámili s detaily o vstupech a výstupech mikrokontrolérů PIC a s pěti speciálními funkčními registry, pomocí kterých lze zapisovat nebo číst úroveň na vstupně-výstupních portech a nastavovat jejich chování. V tomto dílu tyto informace využijeme v praxi pro návrh našeho prvního projektu s mikrokontrolérem PIC. Na obr. 21 je uvedeno velice jednoduché zapojení mikrokontroléru PIC16F88, který řídí jednu LED s předřadným rezistorem 1 kΩ zapojenou na vývod RB4. Ve schématu je rovněž zobrazeno připojení programovacích signálů PGD, PGC, PGM a VPP pro případ, že programujete mikrokontrolér přímo v aplikaci. Využíváte-li programátor s objímkou, tyto informace můžete ignorovat. Rezistor připojený mezi pin PGM a zem není potřeba zapojovat v případě, že jsme již při předchozím programování vypnuli mód nízkonapětového programování vynulováním bitu LVP v konfiguračním slově mikrokontroléru. Vlastníte-li programátor PRESTO nebo obdobný USB programátor, a nemáte-li po ruce vhodný napájecí zdroj, můžete celou aplikaci napájet přímo z programátoru. V případě programátoru PRESTO stačí jednoduše v okně nastavení obslužného programu UP vybrat *Napájení v klidu: Interní 5 V*.

Zdroj hodinového signálu

Mikrokontrolér PIC16F88 je vybaven vnitřním RC oscilátorem s maximálním kmitočtem 8 MHz, díky němuž odpadá v aplikacích nenáročných na velkou přesnost a stabilitu kmitočtu hodinového signálu nutnost zapojovat jakékoliv externí součástky. Mikrokontrolér podporuje celkem osm módů oscilátoru, se kterými se seznámíme v příštích dílech. Pro naši první jednoduchou aplikaci využijeme mód INTIO2, který využívá interní oscilátor. Mód oscilátoru se nastavuje třemi konfiguračními bity FOSC2:FOSC0. Ve výpisu programu v tab. 6 je již tento mód předdefinován pomocí direktivy `__CONFIG` jako `__INTRC_IO`. V programu UP je tento mód označen obdobně jako `INTRC-I/O`. K nastavení kmitočtu interního oscilátoru slouží re-

gistr OSCCON. My v našem prvním projektu využijeme nejnižší implicitně nastavený kmitočet 31,25 kHz, čímž odpadá nutnost měnit hodnoty bitů v registru OSCCON.

Program v jazyce assembleru

Ve vývojovém prostředí MPLAB nejprve vytvoříte nový projekt (menu *Project > New...*) a následně vytvoříte nový `.asm` soubor, který bude obsahovat zdrojový kód. Soubor se zdrojovým kódem je třeba přidat do projektu tak, aby byl přitomen ve složce Source Files v okně projektu, které zobrazíme pomocí menu *View > Project*. Nejjednodušší je vytvořit nový `.asm` soubor pomocí příkazu *Add New File to Project...* v menu *File*. V tomto případě však specifikujte v poli *File name* nejen jméno souboru, ale i jeho typ (`.asm`).

Úkolem programu je periodicky měnit úroveň na vývodu RB4. Protože je LED zapojena proti zemi, bude svítit při úrovni H, zatímco při úrovni L zůstane zhasnutá. Program bude sestávat z následujících kroků:

1. Čekací smyčka.
2. Nastavení úrovně H na pinu RB4.
3. Čekací smyčka.
4. Nastavení úrovně L na pinu RB4.
5. Skok na první čekací smyčku.

Program začíná direktivou `LIST`, pomocí které nastavíme typ procesoru. Následuje direktiva `#INCLUDE`, která umožňuje zahrnout do programu externí kód. Soubor `P16F88.inc` je součástí instalace softwarového prostředí MPLAB a nachází se v adresáři `Microchip/MPASM Suite`. Soubor obsa-

huje definice názvů registrů a některých bitů. Adresy registrů `PORTA`, `PORTB`, `TRISA`, `TRISB` a `ANSEL`, které používáme v našem programu, jsou definovány právě v tomto `.inc` souboru. Soubor si můžete prohlédnout v textovém editoru, nic v něm však neměňte, protože byste tím mohli způsobit nefunkčnost vyvíjených programů. Direktiva `__CONFIG` slouží k nastavení konfiguračních bitů mikrokontroléru (popis konfiguračních slov mikrokontroléru PIC16F88 viz díl 8., PE 8/2008).

V programu dále definujeme pomocí direktivy `EQU` proměnnou `d1` na adrese `7Fh`, kterou použijeme v čekací smyčce, a dále symbol `RB4`. Program bude začínat na adrese `0000h`, což specifikujeme pomocí direktivy `ORG`. Na začátku programu nejprve inicializujeme I/O porty. Na všech pinech nastavíme úroveň L vynulováním registrů `PORTA` a `PORTB` pomocí instrukce `CLRF`, dále nastavíme piny `AN<6:0>` jako digitální I/O vynulováním registru `ANSEL` a na závěr nastavíme všechny I/O piny jako výstupy. Nastavením nepoužitých vývodů jako výstupy zabráníme pronikání rušivých signálů do mikrokontroléru. Registry `PORTA` a `PORTB` se spolu s proměnnou `d1` nacházejí v bance 0, zatímco registry `TRISA`, `TRISB` a `ANSEL` se nacházejí v bance 1. Příslušnou banku datové paměti vybereme direktivou `BANKSEL`. Hlavní část programu, která periodicky mění úroveň na výstupu RB4, se vykonává ve smyčce, která začíná návěštím `LOOP`. K nastavení nebo vynulování jediného bitu v určitém registru slouží instrukce `BSF` (Bit Set f – nastavení bitu v registru) a `BCF` (Bit Clear f – vynulování bitu v registru). Instrukce `CALL` volá podprogram, v našem případě jednoduchou čekací smyčku `DELAY`. Instrukce `MOVLW` slouží k přesunu konstanty do registru `W` a instrukce `MOVWF` zkopíruje

Tab. 6.
Výpis programu v assembleru pro blikající LED.
Program nemusíte opísat, můžete si jej stáhnout na www.aradio.cz na stránce „Programy k mikroprocesorovým zařízením a PC“

; Projekt: Blikající LED

```
LIST          p=16F88                ; nastavení typu procesoru
#include      <P16F88.INC>           ; definice proměnných pro tento procesor

; NASTAVENÍ KONFIGURAČNÍHO SLOVA
; (direktiva __CONFIG musí být na jednom řádku)

__CONFIG __CONFIG1, _CP_OFF & _CCP1_RB0 & _DEBUG_OFF & _WRT_PROTECT_OFF
& _CPD_OFF & _LVP_OFF & _BODEN_OFF & _MCLR_ON & _PWRTE_ON & _WDI_OFF & _INTRC_IO
__CONFIG __CONFIG2, _IESO_OFF & _FCMEN_OFF

; DEFINICE PROMĚNNÝCH

d1            EQU            0x7F
RB4           EQU            0x04

; VLASTNÍ PROGRAM

RESET        ORG            0x0000    ; začátek programu na adrese 0000h

; Inicializace portů
BANKSEL      PORTA                ; výběr banky, ve které se nachází registr PORTA
CLRF         PORTA                ; vynulování PORTA
BANKSEL      PORTB                ; výběr banky, ve které se nachází registr PORTB
CLRF         PORTB                ; vynulování PORTB

BANKSEL      ANSEL                ; výběr banky, ve které se nachází registr ANSEL
CLRF         ANSEL                ; piny AN<6:0> jako digitální I/O

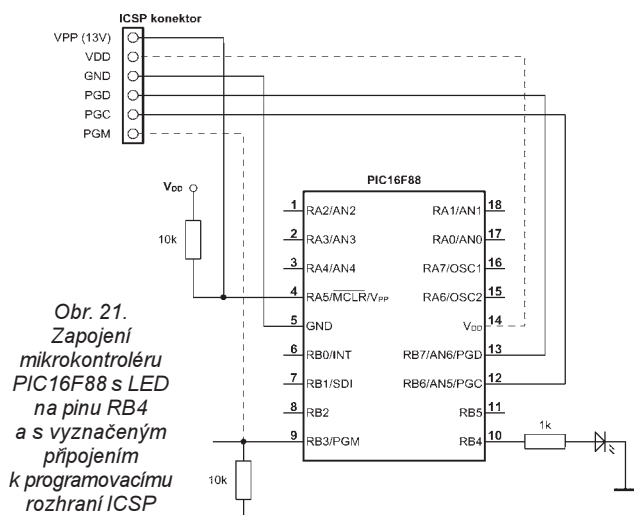
CLRF         TRISA                ; piny RA<7:0> jako výstupy
CLRF         TRISB                ; piny RB<7:0> jako výstupy

; Ovládání LED
BANKSEL      PORTB                ; výběr banky 0
LOOP:        CALL    DELAY         ; volání podprogramu DELAY (čekací smyčka)
            BSF      PORTB, RB4    ; RB4 = H
            CALL    DELAY         ; volání podprogramu DELAY (čekací smyčka)
            BCF      PORTB, RB4    ; RB4 = L
            GOTO     LOOP          ; skok na LOOP

; Podprogram DELAY s čekací smyčkou
DELAY:       MOVLW    .255          ; přesuň konstantu 255 do pracovního registru W
            MOVWF    d1            ; přesuň obsah registru W do proměnné d1

DELAY_0:     d1, f                ; odečti jedničku od hodnoty registru d1
            GOTO     DELAY_0       ; je-li d1 > 0, provede se skok na DELAY_0
            RETURN                ; je-li d1 = 0, provede se návrat z podprogramu

END
```



Obr. 21.
Zapojení mikrokontroléru PIC16F88 s LED na pinu RB4 a s vyznačeným připojením k programovacímu rozhraní ICSP