

Ako som si PIC-ko bootloaderom naflešoval

Ako už bolo spomenuté v článku "[Bootloader, čo, ako, prečo](#)", bootloader je malý program, ktorý prevezme kontrolu nad MCU po resete, je schopný komunikovať cez vhodné rozhranie a meniť obsah nonvolatile programovej pamäte (väčšinou FLASH), pričom – z pochopiteľných dôvodov – bootloader samotný zostane neporušený. Väčšinou sa jedná o programovú FLASH pamäť a sériovú RS232 linku. Tým sa zjednodušuje preprogramovanie tejto pamäte, na ktoré nie je treba nijaké špeciálne nástroje, iba RS232 rozhranie, ktoré v MCU aplikáciách býva tak ako tak obsiahnuté.

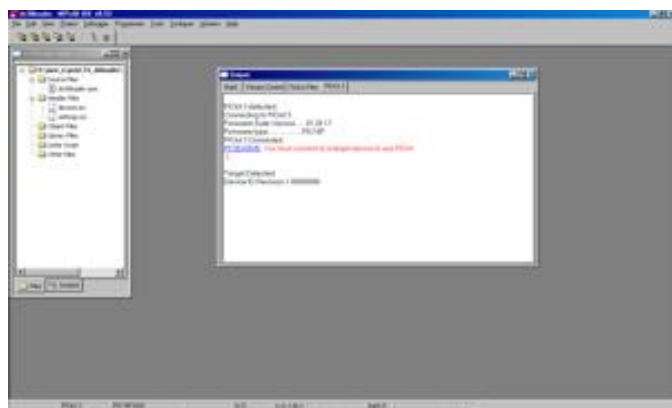
Aj keď bootloader nepoužívam, rozhodol som sa jeden skúsiť. Pretože podstatnú väčšinu svojich konštrukcií som založil na PIC mikrokontroléroch, hľadal som niečo pre túto platformu. Ako najlepšia alternatíva sa mi zdal bootloader pre prakticky všetky PIC 8 a 16-bitové mikrokontroléry na. Autor dodáva zdrojové kódy pre všetky hw platformy a aj SW v PC a je k nemu celkom pekná diskusia na Microchip fórach [tu](#), [tu](#) a [tu](#).

Ako na to

V zásade je to jednoduché. Človek potrebuje nejaký programátor na napálenie bootloadera do MCU, nejaký HW na ktorom to bude pracovať a sériový port. Pretože sériové porty sú „nemoderné“, s týmto môže byť problém, ale funguje to aj s USB – RS232 prevodníkmi.

Tak som si teda stiahol inštalračný balíček z [6] a po inštalácii som našiel adresár s firmware, zvolil som si firmware pre PIC18F, pretože môj hw je založený na PIC18F2420. Práca s bootloaderom pre iné platformy bude asi veľmi podobná.

Projekt aj so zdrojovými súbormi som skopíroval do nového adresára a otvoril ho v MPLAB-e a zvolil programátor (v mojom prípade PicKit3). Tento krok vyzerá asi takto:



Nastavenia bootloadera sú v súbore settings.inc. Treba zvoliť

1. typ MCU (defaultne je tam PIC18F2550),
2. frekvenciu oscilátora (defaultne je tam 48MHz),
3. baudovú rýchlosť (defaultne 115,2kbaud) a
4. nastaviť konfiguračné bity.

Takže postupne:

1, je to pomerne jednoduché, pre typ 18F2420 sa jednoducho prepíše

1.
;
- 2.

; Device

3.

;

4.

LIST P=18F2550; ;xxx

na

1.

;

2.

; Device

3.

;

4.

LIST P=18F2420; ;xxx

Bod 2 a 3 je opäť jednoduché vykonať. V mojom prípade mám na doske externý kryštál 7,3728MHz a teda som si mohol zvoliť pomerne vysokú prenosovú rýchlosť, 115,2kbud, tak som zmenil

1.

#define FOSC 48000000 ;xxx

2.

#define BAUDRATE 115200 ;xxx baudrate

na

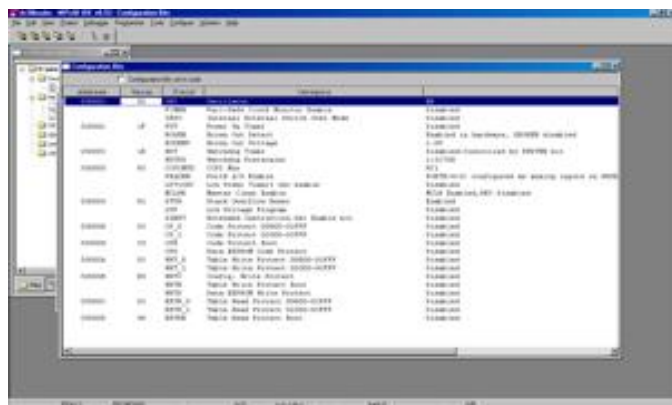
1.

#define FOSC 7372800 ;xxx

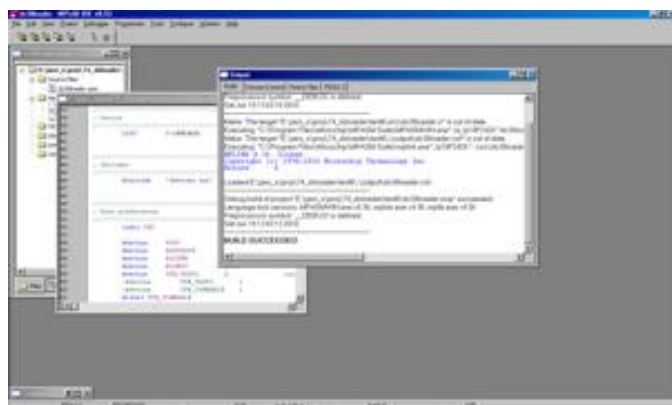
2.

```
#define BAUDRATE 115200 ;xxx baudrate
```

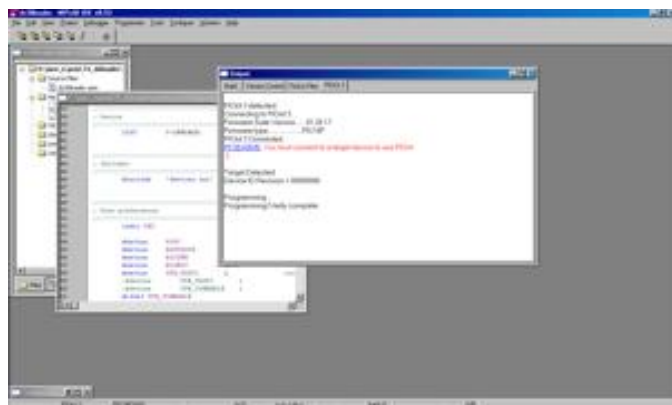
Bod 4 vyžaduje trochu viac akcie. Z defaultných nastavení v MPLAB-e som zmenil toto - nastavil oscilátor na HS, vypol watchdog a LVP:



Tým bol bootloader pripravený na Build, tak som stlačil F10 a MPLAB zahlásil, že je všetko OK:

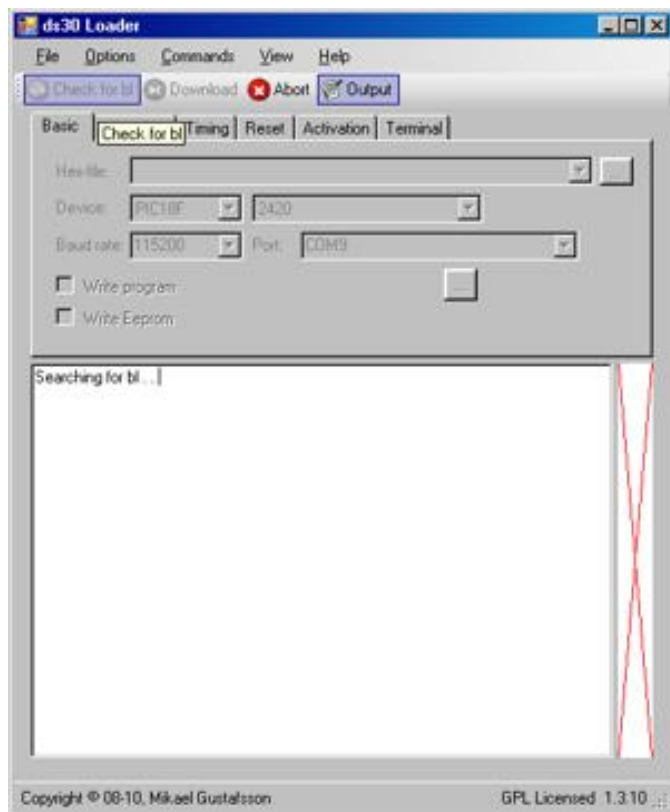


A naprogramoval som ho do cieľového MCU:

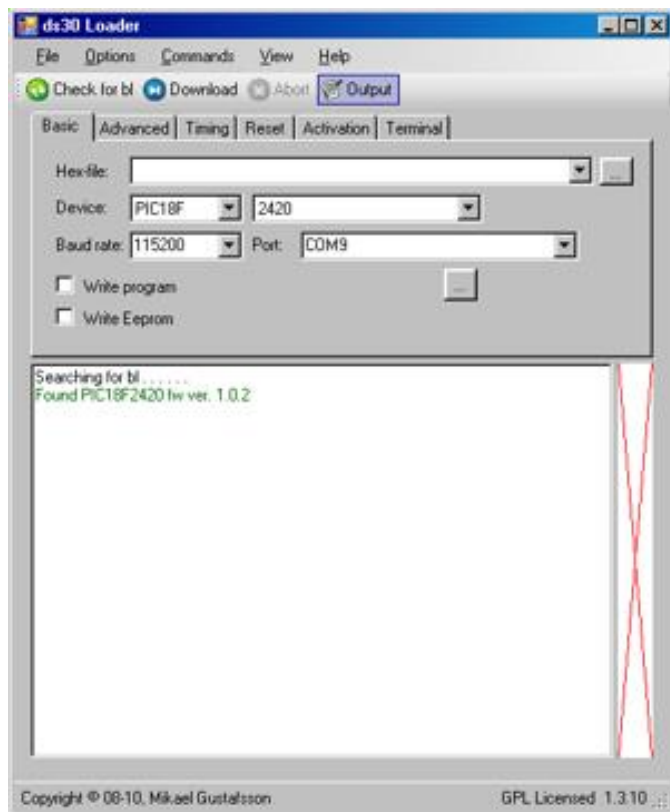


Tým je MCU pripravený - za predpokladu, že je všetko korektne nastavené - na prácu s bootloaderom.

Spustil som GUI bootladeru, v položke „View“ som zvolil „Advanced mode“ a nastavil všetko potrebné a stlačil „Search for bl“, čím sa program snažil naviazať komunikáciu s bootloaderom:



To sa mu nepodarilo až kým som cieľový HW nespustil z reset-u, MCU zbehol na bootloader a začal komunikovať s PC programom:



No, takže teraz už len nájsť niečo, čo cez ten bootloader naliať do MCU, vytvoril som si jednoduchú aplikáciu na blikanie LED-kami na porte C:

```
1.
#include

2.
#include

3.

4.
void main (void)

5.
{

6.
    TRISC = 0;

7.
    LATC = 0;

8.
    while (1)

9.
    {

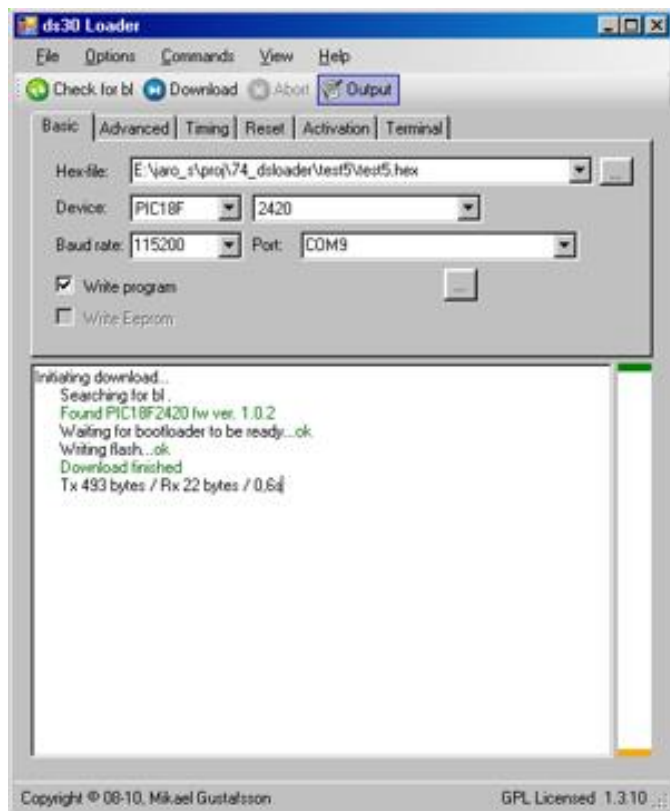
10.
        LATC = ~ LATC;

11.
        Delay10KTCYx(100);

12.
    }

13.
}
```

Po kompilácii kompilátorom MCC18 som GUI bootloadera nasmeroval na vygenerovaný súbor a stlačil „Download“:



Krátko po tom mi dve LED diódy na porte C začali blikať, čo znamená, že misia bola splnená.

Čo ďalej

Tento konkrétny bootloader toho ponúka aj viac. Okrem toho, že má GUI, má aj verziu spustiteľnú z príkazového riadku, umožňuje aj možnosť automatického resetu pred pokusom o programovanie, aby človek nemusel ručne cvičiť resetovacím tlačidlom. V takom prípade je reset zvisaný so stavom linky DTR alebo RTS. GUI ponúka aj jednoduchý terminál:



Takže odchyťavanie ladiacich výpisov a programovanie sa dá robiť v tom istom okne.

Záver

No, a to je zhruba tak všetko. Bootloader je v zásade jednoduchá vec a to, čo ponúka je len programovanie MCU bez programovacieho rozhrania, čo sa hodí napríklad „v teréne“, hoci možnosti, ktoré ponúka programovací mechanizmus na to určený (v prípade PIC je to ICSP) nemôže prekonať.

Odkazy

- [1] [Bootloader - čo, ako, prečo](#)
- [2] [ds30loader](#)
- [3] [microchip fórum - 1](#)
- [4] [microchip fórum - 2](#)
- [5] [microchip fórum - 3](#)
- [6] [ds30_Loader_100529.exe](#)