

# Vážení zákazníci,

dovolujeme si Vás upozornit, že na tuto ukázkou knihy se vztahují autorská práva, tzv. copyright.

To znamená, že ukáзка má sloužit výhradně pro osobní potřebu potenciálního kupujícího (aby čtenář viděl, jakým způsobem je titul zpracován a mohl se také podle tohoto, jako jednoho z parametrů, rozhodnout, zda titul koupí či ne).

Z toho vyplývá, že není dovoleno tuto ukázkou jakýmkoliv způsobem dále šířit, veřejně či neveřejně např. umístováním na datová média, na jiné internetové stránky (ani prostřednictvím odkazů) apod.

*redakce nakladatelství BEN – technická literatura*  
[redakce@ben.cz](mailto:redakce@ben.cz)



Můj startkit ještě obsahuje stabilizátor 7805. Převodník MAX232 IC1 je používán při ISP programování. K P0.24 až P0.31 se přes propojky JP5 dá připojit 8 diod LED, k P0.16 až P0.23 se přes JP6 dají připojit odpory přivádějící na tyto piny log.1 a dále 8 tlačítek po jejichž stisknutí se na odpovídající pin přivede log.0 (na obr. 3.13 nejsou tato tlačítka zakreslena – obrázek by již nebyl tak přehledný). Druhý MAX232 spolu s X2 je určen pro případnou komunikaci uživatelským programem s dalším sériovým portem počítače PC.

### 3.3 Ukázka konkrétní práce s LPC2106

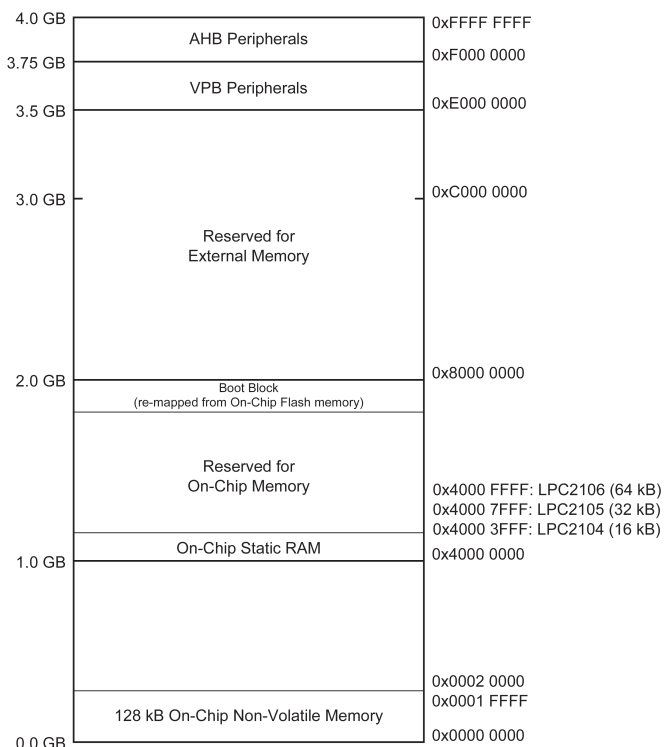
Při práci s 8bitovými jednočipy bývá prvním programem, který si odzkoušíme, program, který např. střídavě rozsvěcuje a zhasíná liché a sudé diody LED připojené k paralelnímu portu o 8 pinech (jde o 8bitový počítač). Pokusíme se napsat obdobný program, který bude předpokládat 8 LED diod připojených k P0.24 až P0.31 (obr. 3.13). Mj. musíme pamatovat na to, že délka slova je nyní 32 bitů. Dále si musíme zjistit, jak k „paralelnímu portu“ budeme přistupovat. Zatím jsme se zabývali jen jádrem ARM. U LPC210x se k perifériím přistupuje přes registry. Přístup k periférii na obr. 3.1 označené jako 32 General Purpose I/O (vlastně paralelní port) je v uživatelské příručce k LP2104/2105/2106 popsán v kap. 8 GPIO.

Než se pustíme do programování, musíme si zjistit další podrobnosti k LPC2106. Nejprve zjistíme, jak vypadá jeho paměťový prostor obr. 3.14.

Je vidět, že k perifériím (viz obr. 3.1) se přistupuje pomocí instrukcí přístupu k paměti. Jenom místo paměti SRAM jsou na odpovídajících adresách umístěny registry periférií. Podrobnosti najdeme v již zmiňované uživatelské příručce. Konkrétně pro GPIO je použito:

Adresa	Jméno	Popis	Přístup
0xE0028000	IOPIN	Slouží ke čtení aktuálního stavu GPIO pinů	Jen čtení
0xE0028004	IOSET	GPIO 0 výstupní 32bitový registr pro nastavení hodnot 1 na odpovídajících pinech periferie. Jednička zapsaná do tohoto registru se projeví jako 1 na odpovídajícím výstupu, nula zapsaná do tohoto registru se nijak neprojeví.	Čtení/nastavení
0xE0028008	IODIR	GPIO 0 řídicí registr pro určení směru I/O pro každý z bitů GPIO periferie. Tj. určuje, které bity budou vstupní a které výstupní.	Čtení/zápis
0xE002800C	IOCLR	GPIO 0 výstupní 32bitový registr pro nastavení hodnot 0 na odpovídajících pinech periferie. Jednička zapsaná do tohoto registru se projeví jako 0 na odpovídajícím výstupu, nula zapsaná do tohoto registru se nijak neprojeví.	Pouze nulování

**POZNÁMKA:** Všimněme si, že pro zápis 0 a pro zápis 1 máme dva různé registry. U 8bitových jednočipů většinou byl k dispozici společný 8bitový registr; do něhož jsme zapisovali slovo, jehož obsah se pak projevil na pinech paralelního portu.



**Obr. 3.14**

Nyní je již zřejmé, že pro naše zapojení z *obr. 3.13* budeme muset do IODIR zapsat hodnotu 0xFF000000, aby piny, k nimž mám připojeny LED diody byly výstupními. Aby LED svítily, musí být na příslušném pinu 1, aby nesvítily, musí na pinu být 0.

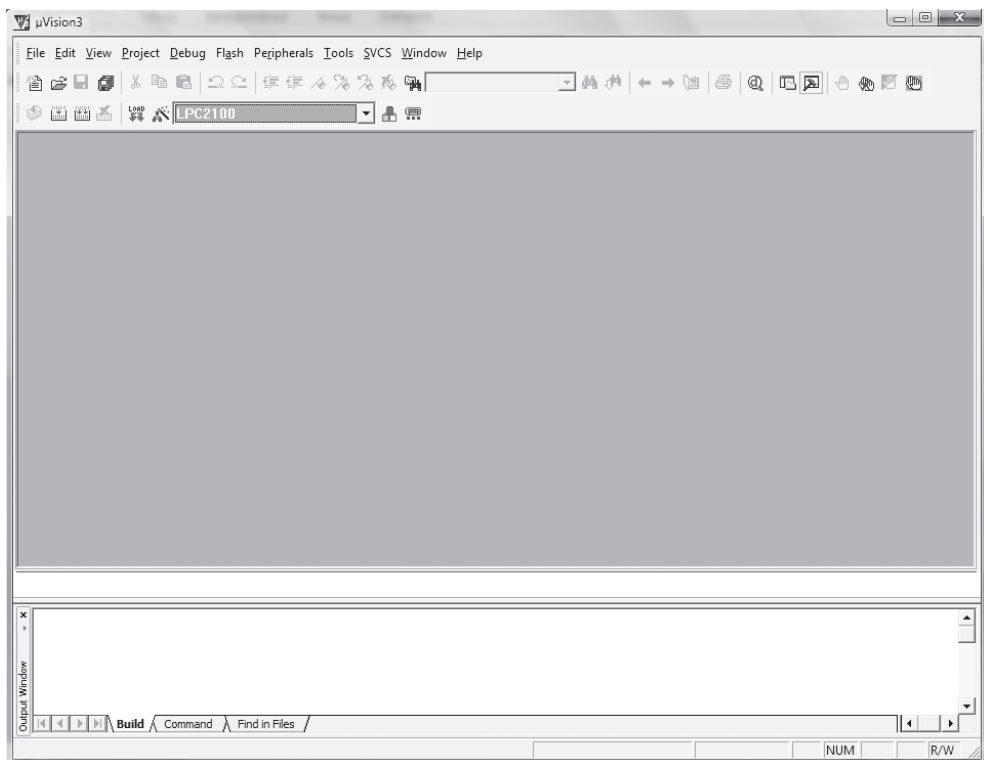
Aby svítily liché LED diody musíme do IOSET zapsat 0x55000000 a do IOCLR 0xAA000000.

Pro rozsvícení sudých LED diod do IOSET zapíšeme 0xAA000000 a do IOCLR 0x55000000.

Pro vytvoření blikáče pak umístíme výše zmíněné zápisy do IOSET a IOCLR do nekonečné smyčky. Dále musíme použít nějaký podprogram pro vytvoření časové prodle-  
vy a volání tohoto podprogramu umístíme v nekonečné smyčce vždy za dvojici zápisů do IOSET a IOCLR. Parametry podprogramu časového zpoždění pak určují rychlost blikání LED diod.

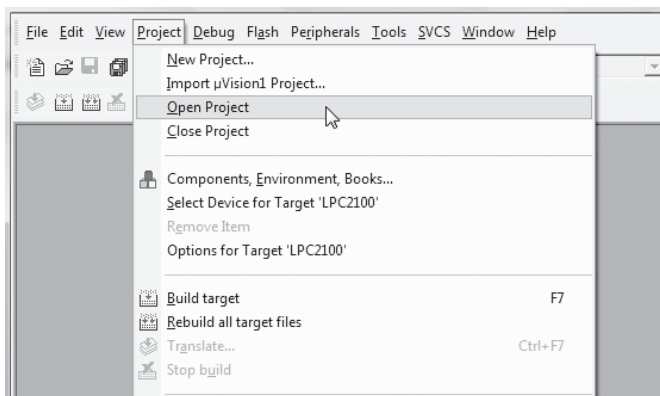
Nyní si vše prakticky předvedeme. Nejjednodušší bude asi napsat program v jazyce C. Pro překlad z jazyka C do kódu ARM existuje řada překladačů jazyka C, spolu s knihovnamy a popř. i s vývojovým prostředím IDE. Pro předvedení použijeme uVision3 firmy Keil Software [2]. Tato firma je podporovaná firmou ARM a má proto jistě nejlepší informace pro tvorbu vývojového software. Po registraci na jejich stránkách můžeme zdarma používat evaluation verzi jejich vývojových nástrojů. Nainstalujeme je do PC s OS Win-

dows. Pokud použijeme přednastavené volby, nainstaluje se do C:\Keil. Součástí instalace je i podsložka Examples obsahující několik ukázkových projektů. Mezi nimi je i projekt **blinky** používaný jako ukázkový program u řady startkitů i vývojových programů. Jeho výsledkem je „běžící“ LED dioda. Práci si usnadníme tím, že ho po úpravě použijeme. Pro náš první project jsme vytvořili podsložku **program01** (kdekoli na disku, umístili jsme tuto složku do složky data na disku D. Do této složky zkopírujeme obsah složky **blinky** z složky **examples** z C:\Keil\ARM\Examples. Poté spustíme vývojové prostředí *obr. 3.15*.



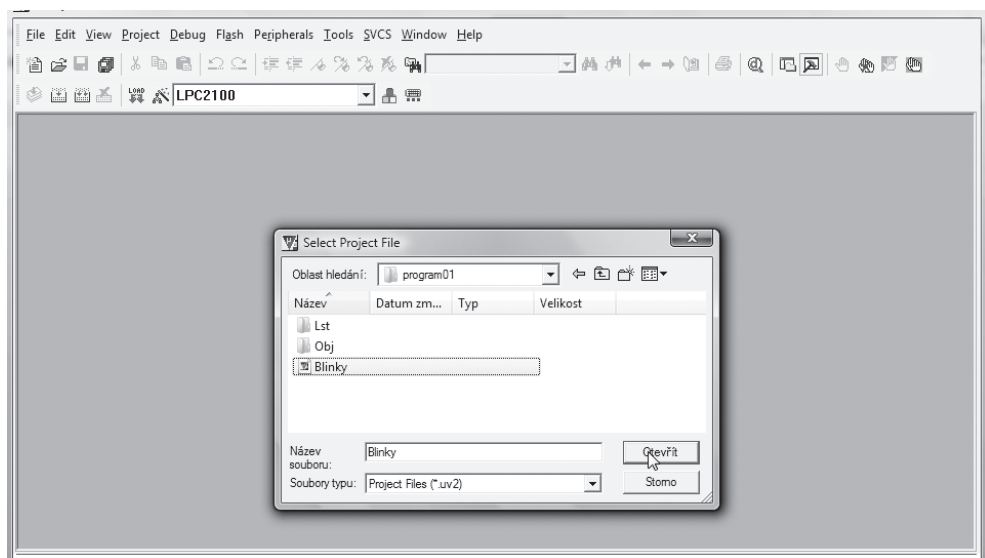
**Obr. 3.15**

V menu vybereme **Project** → **Open Project** *obr. 3.16*. Použijeme totiž již hotový project **blinky** zkopírovaný do nové složky. Na *obr. 3.15* je také vidět, jak budeme postupovat, budeme-li vytvářet skutečně nový project – vybereme položku **New project**. Tento výběr bude později asi častější.



**Obr. 3.16**

Po výběru **Project** → **Open Project** se objeví okno **Select Program File** s jehož pomocí ve windows běžným způsobem nastavíme umístění našeho upravovaného projektu **blinky**. Dále označíme soubor **blinky.uv2** a klikneme na tlačítko **Otevřít** obr. 3.17.



**Obr. 3.17**

Nyní již můžeme pracovat se zdrojovým kódem v jazyce C, obr. 3.17.

```

D:\data\arm7\moje_kniha\program01\Blinky.c
01  /* ***** */
02  /* This file is part of the uVision/ARM development tools */
03  /* Copyright: KEIL ELEKTRONIK GmbH 2002-2004 */
04  /* ***** */
05  /* */
06  /* BLINKY.C: LED Flasher */
07  /* */
08  /* ***** */
09
10  #include <LPC21xx.H> /* LPC21xx definitions */
11  #include "Timer.h"
12
13  extern long volatile timeval;
14
15
16  void wait (void) { /* wait function */
17      unsigned long i;
18
19      i = timeval;
20      while ((i + 10) != timeval); /* wait 100ms */
21  }
22
23
24  int main (void) {

```

**Obr. 3.18**

Především nastavíme piny GPIO, ke kterým máme připojeny diody LED a dále zcela změňme obsah těla nekonečné smyčky while (1) { }. Výsledkem je kód:

```

#include <LPC21xx.H> /* LPC21xx definitions */
#include „Timer.h“
extern long volatile timeval;
void wait (void) { /* wait function */
    unsigned long i;
    i = timeval;
    while ((i + 10) != timeval); /* wait 100ms */
}
int main (void) {

    IODIR1 = 0xFF000000; /* P1.16..23 definovány jako výstupy */

    init_timer();

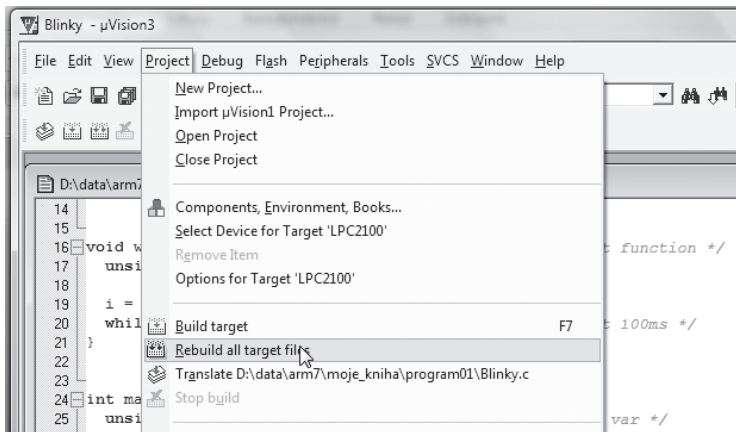
    while (1) { /* Loop forever */

        IOSET1 = 0x55000000;
        IOCLR1 = 0xAA000000;
        wait (); /* call wait function */
        IOSET1 = 0xAA000000;
        IOCLR1 = 0x55000000;
        wait ();

    }
}

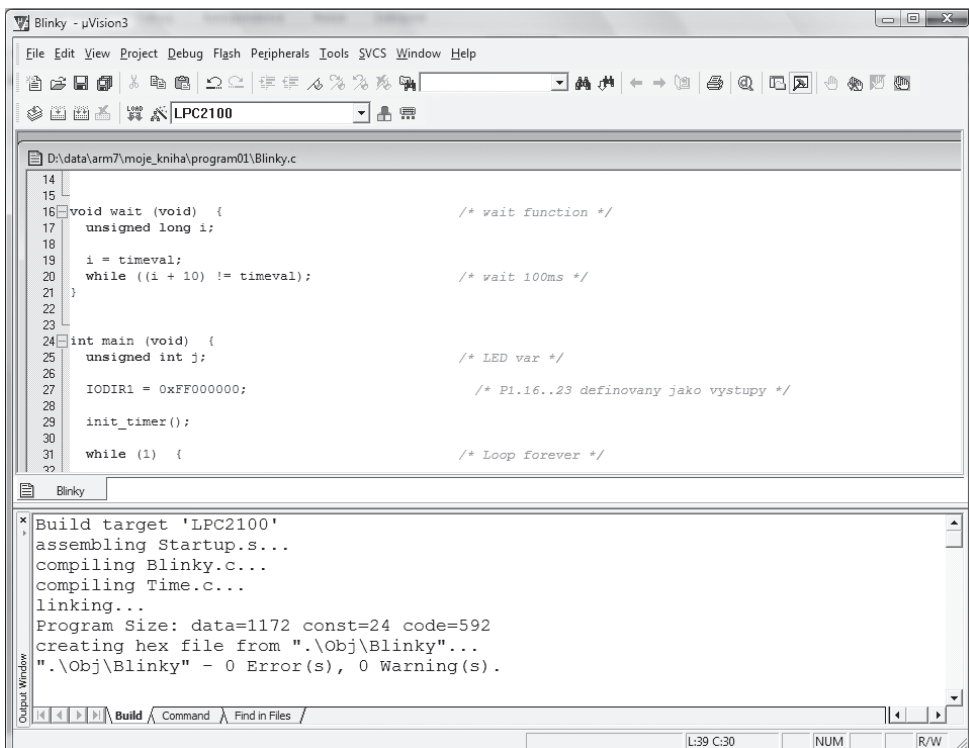
```

Zbývá kód přeložit. V menu vybereme **Project** → **Rebuild all target files** obr. 3.19.



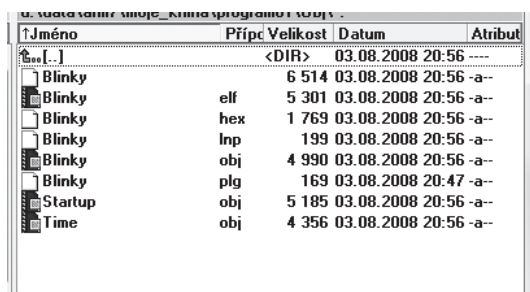
Obr. 3.19

V dolním okně **Build** uvidíme informaci o průběhu kompilace a linkování. V případě chyb uděláme opravu. Bezchybný výsledek je na obr. 3.20.



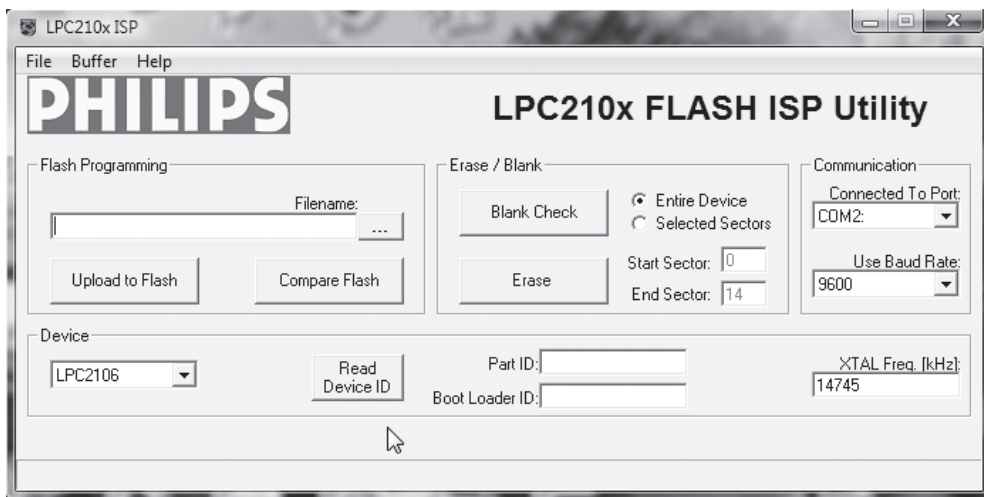
Obr. 3.20

V podsložce **obj** se objevil nový soubor blinky.hex. Je to požadovaný výsledek naší činnosti, obr. 3.21.



Obr. 3.21

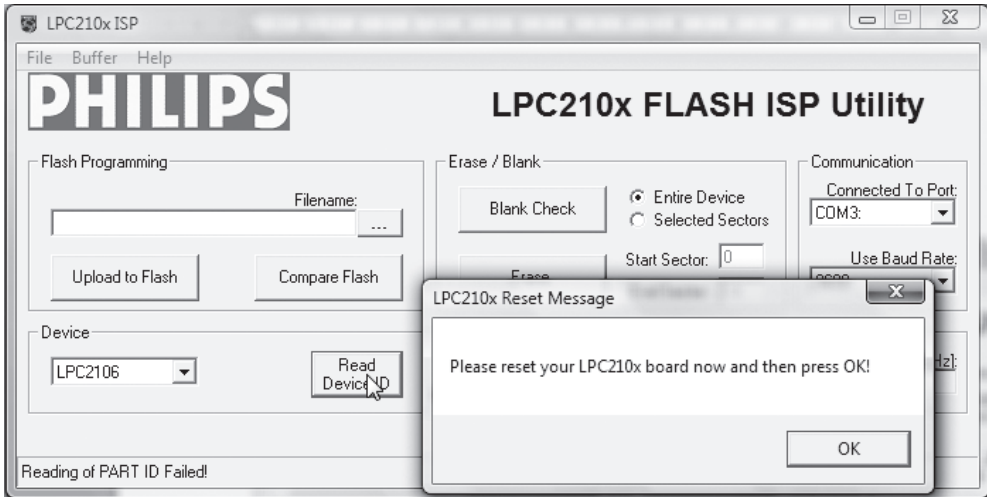
Jeho obsah umístíte do programové paměti Flast LPC2106 pomocí „LPC210x FLASH ISP Utility“, kterou proto spustíme, obr. 3.22.



Obr. 3.22

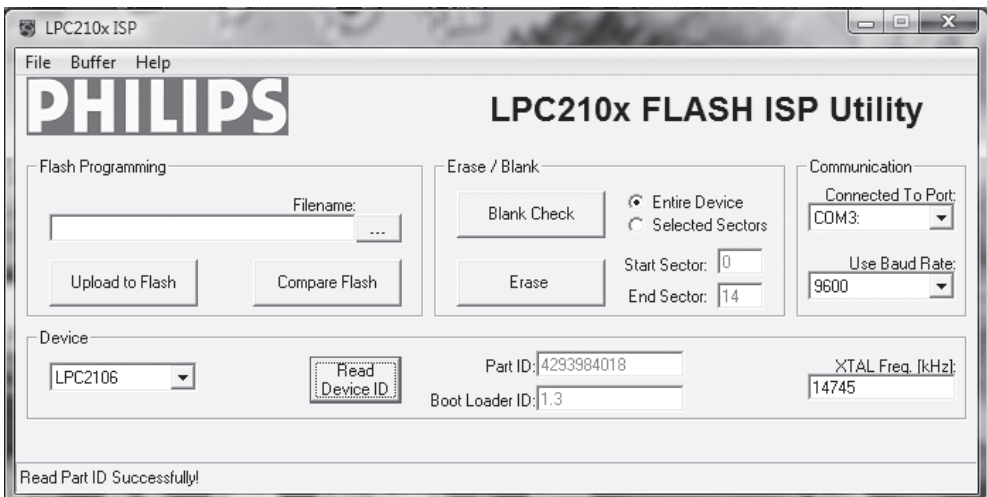
Konektor X1 startkitu z obr. 3.13 spojíme kabelem se sériovým portem PC. Na startkitu pomocí propojky připojíme odpor R1 k P0.14 a tím na tento pin přivedeme log.0. Dále v listboxu **Connected To Port** nastavíme pojmenování sériového portu PC, ke kterému máme připojen kabel a poté klikneme na tlačítko **Read Device ID**. Objeví se okno **LPC210x Reset Message** obr. 3.23. V našem případě jsme jako sériový port nastavili COM3, neboť na mém PC mám pouze USB a proto jsem použil převodník USB/sériový port. Po nainstalování tohoto převodníku se objeví virtuální sériový port COM3.





**Obr. 3.23**

Stiskneme tlačítko **RESET** na startkitu a poté tlačítko **OK** okna **LPC210x Reset Message**. Úspěšná komunikace utility s LPC2106 se projeví oknem obr. 3.24.



**Obr. 3.24**

Kliknutím na tlačítko se třemi tečkami napravo od okénka **Filename** se objeví okno **Otevřít**, pomocí kterého nastavíme cestu k souboru blinky.hex obr. 3.25.