

Přípravy na kroužek Mechatronika – KA6(A2d)

2. lekce – číselné soustavy, bitové posuny, avrdude, Atmel studio – blikání LED

Číselné soustavy používané v programování MCU

1. **Binární soustava** – zahrnuje pouze 2 hodnoty (0, 1).
2. **Desítková (decimální) soustava** – zahrnuje hodnoty (0, 1, 2, 3, 4, 5, 6, 7, 8, 9).
3. **Šestnáctková (hexadecimální) soustava** – zahrnuje stejné hodnoty jako soustava desítková, navíc přidává hodnoty (A, B, C, D, E, F), ty reprezentují čísla 10, 11, 12, 13, 14 a 15 v soustavě desítkové.

Binární soustava se používá především k vyjádření stavů (hodnoty bitů v jednotlivých registrech) atd. **Desítková** soustava se používá pro vyjádření hodnot, které chceme mít v běžném čitelném formátu a zároveň není potřeba číslo vyjadřovat v soustavě šestnáctkové.

Hexadecimální soustava se používá tam, kde je vyžadován úsporný zápis hodnot.

Jelikož při programování MCU pracujeme s osmibitovými hodnotami (byty), je rozsah vyjádřených hodnot v jednom bytu **0-255** a zápis vypadá takto:

V binární soustavě:

0b00000001

0b před vlastní číslovkou značí, že se jedná právě o binární soustavu.

V desítkové soustavě:

1

V šestnáctkové soustavě:

0x01

0x před vlastní číslovkou značí, že se jedná právě o hexadecimální soustavu.

Mezi jednotlivými soustavami je možné provádět převody. Většinou provádíme převody mezi desítkovou soustavou a některou ze soustav ostatních.

Převody z ostatních soustav na desítkovou provádím sečtení násobků mocnin čísla, které představuje základ dané soustavy (např. u binární soustavy sčítáme mocniny čísla 2).

Při převodu ze soustavy desítkové do statní soustavy naopak původní číslo postupně dělíme mocninami základního čísla soustavy a z výsledných podílů sestavíme konečnou hodnotu tak, že je zleva doprava poskládáme za sebe.

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Příklady hodnot (bin – dec – hex):

0b00100101 = 37 = 0x25

0b10101000 = 168 = A8

Bitové posuny

Při práci s binárními hodnotami můžeme provádět tzv. bitové posuny. Bitový posun můžeme provádět zleva doprava nebo zprava doleva. Bitový posun doleva značíme << a doprava >>.

Při převodu doleva bity, které "přetečou" přes 8. bit vypadnou ven, již se s nimi nepočítá. Obdobně je to i při bitovém posunu doprava.

Zápis:

1 << 5 – značí posunutí bitu 1 doleva o 5 míst, tedy:

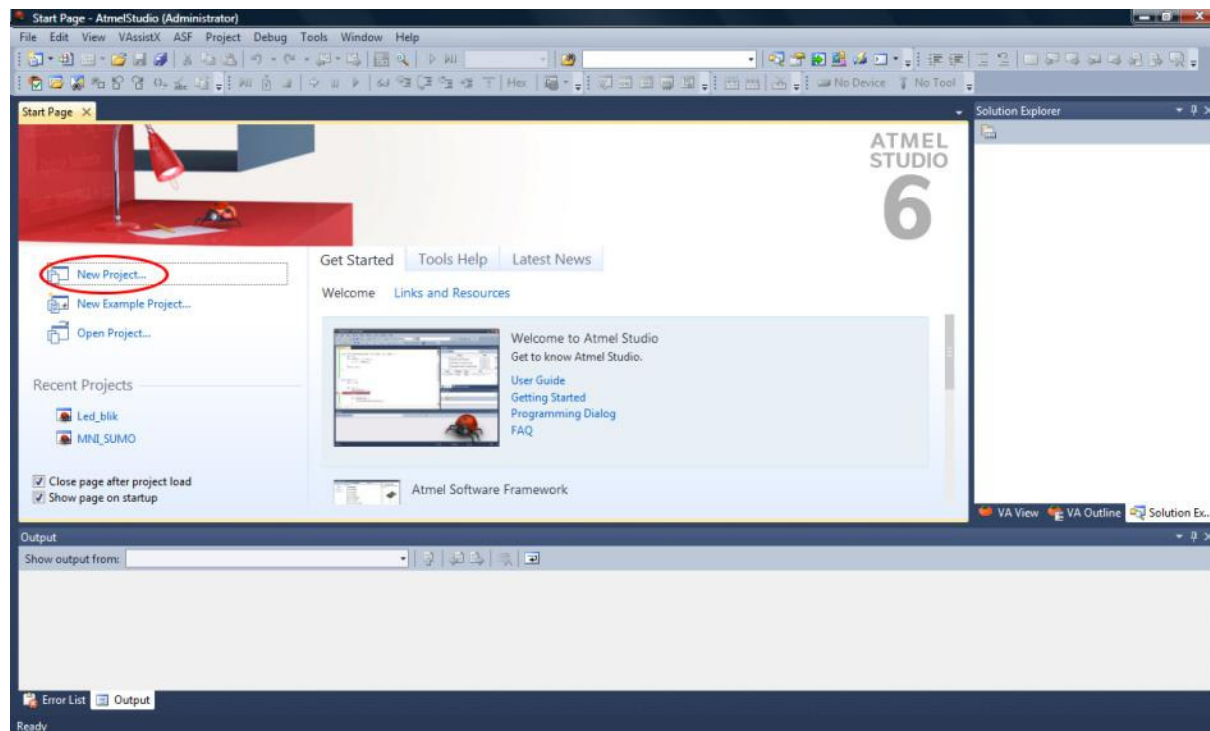
0b00000001 << 5 = 0b00100000

opačně:

0b10000000 >> 5 = 0b00000100

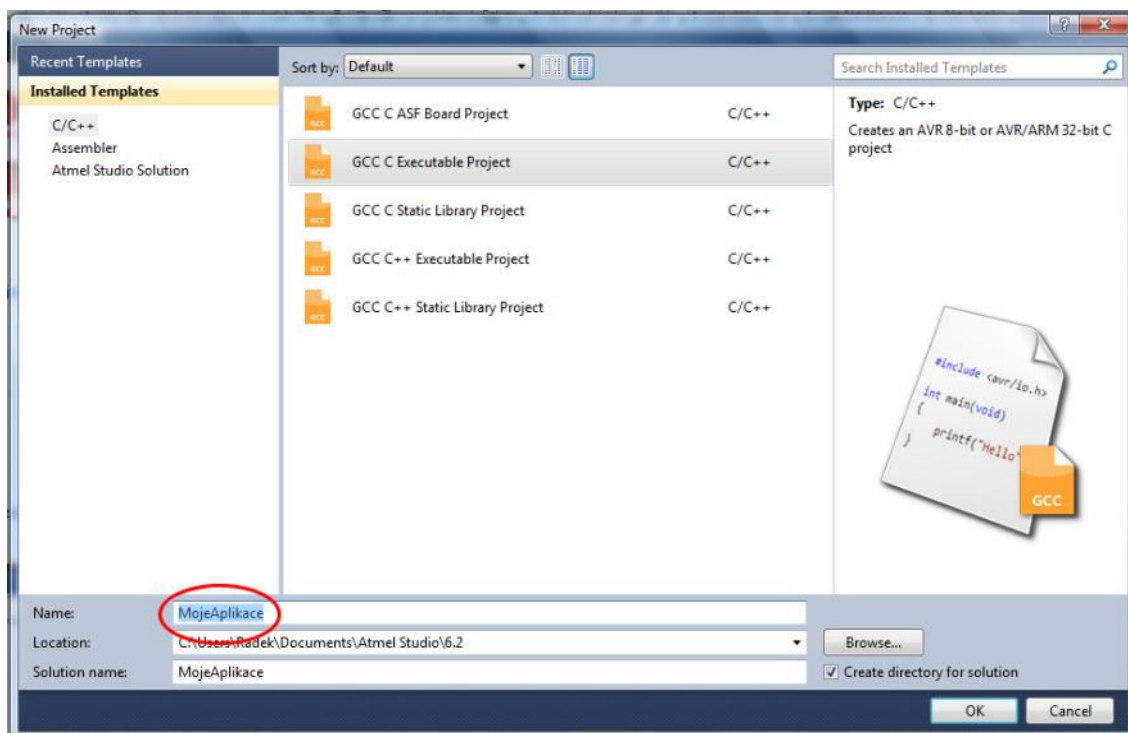
Dále: 0b01000001 << 4 = 0b00010000 a 0b01000001 >> 4 = 0b00000100.

Atmel studio – založení nového projektu

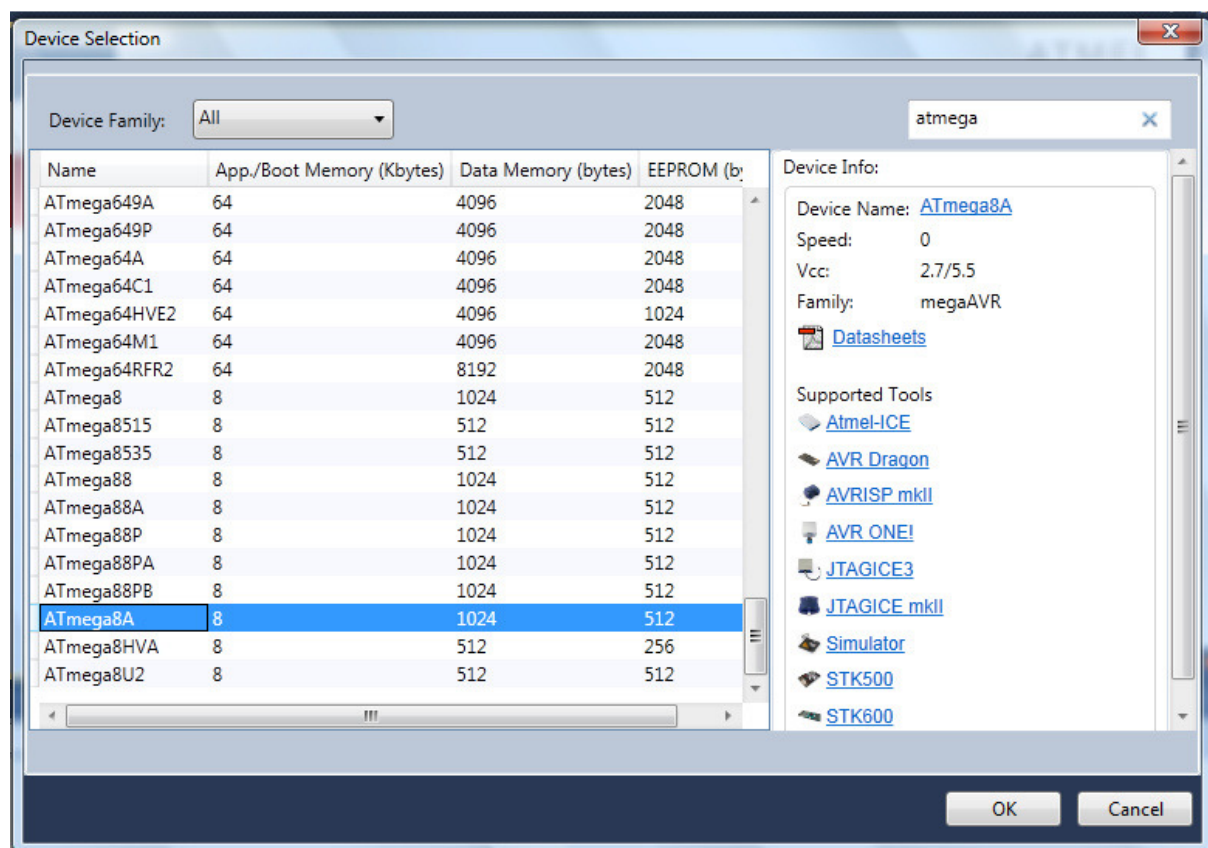


Obrázek 1

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

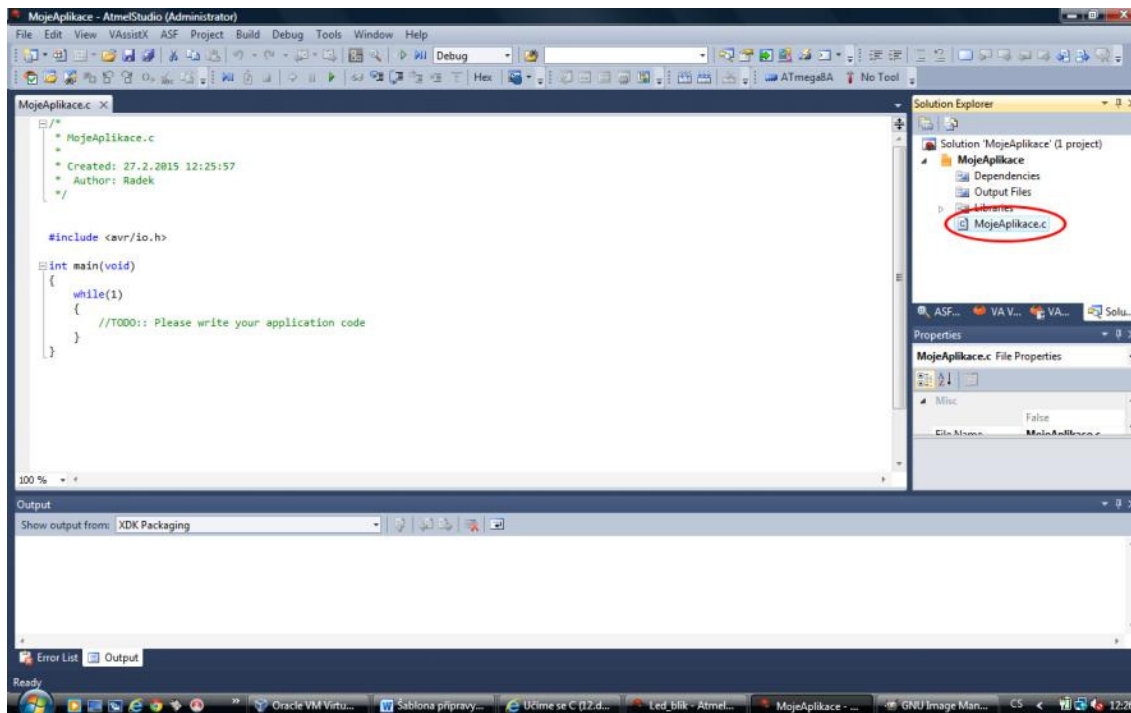


Obrázek 2



Obrázek 3

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ



Obrázek 4

Postup:

1. Spustíme Atmel studio, klikneme na New project (obr. 1).
2. Vybereme GCC C Executable Project a napíšeme název aplikace (obr. 2).
3. Vybereme procesor (obr. 3).
4. Můžeme pracovat s otevřeným projektem.

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

První program – blikání LED

Do souboru .c s naším projektem zapíšeme následující kód.

```
/*  
 * Led_blik.c  
 *  
 * Created: 4.2.2015 21:51:51  
 * Author: Radek  
 */  
  
#define F_CPU 8000000UL  
#include <avr/io.h>  
#include <util/delay.h>  
  
int main(void)  
{  
    DDRB=0b00000001;  
    PORTB=0b00000001;  
  
    while(1)  
    {  
        PORTB=0b00000001;  
        _delay_ms(1000);  
        PORTB=0b00000000;  
        _delay_ms(1000);  
    }  
    return 0;  
}
```

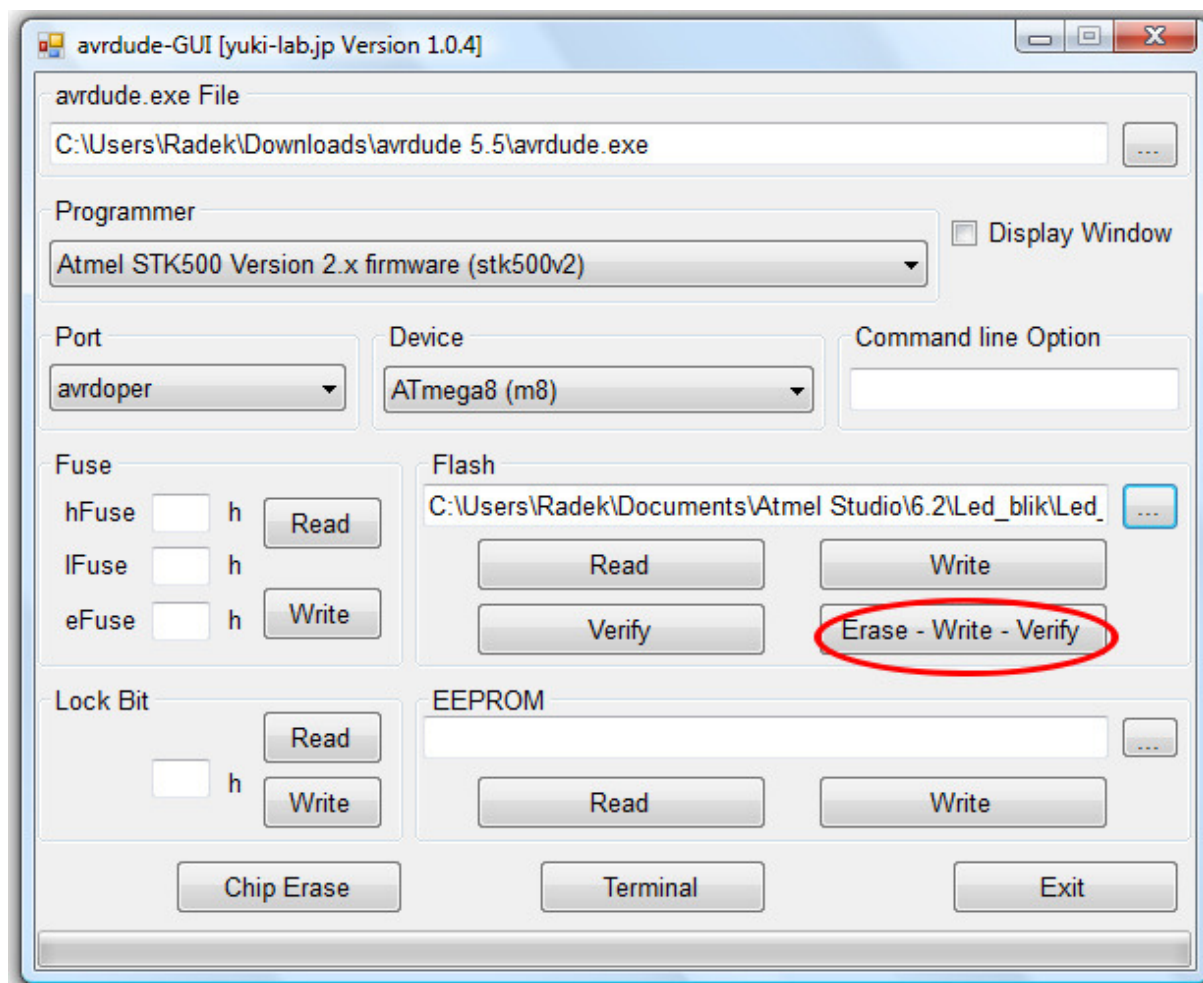
Pomocí define zadáme makro **F_CPU**, které udává funkci **_delay_ms** rychlost procesoru MCU. Pomocí include připojíme potřebné knihovny (**io.h** pro ovládání výstupů MCU a **delay.h** pro funkci **_delay_ms**).

Následně v hlavní funkci **main** nejprve nastavíme **první pin PORTB** na výstup pomocí **registru DDRB**. Zapneme výstup (napětí) na příslušném pinu **PORTB**.

V nekonečném cyklu **while** se pak nachází vlastní blikání kdy střídavě vypínáme a zapínáme příslušný výstup **PORTB**, o prodlevy se stará funkce **_delay_ms(1000)**, která udává prodlevu mezi stavy v milisekundách.

Nakonec v Atmel studiu stiskneme klávesu F7, čímž dojde k sestavení projektu do binárního souboru s příponou .hex. Tento soubor následně nahrajeme pomocí **avrdude** do MCU.

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ



Postup nahrání:

1. Spustíme grafickou nadstavbu avrdude.
2. Zbylé věci nastavíme dle předlohy v obrázku, do pole flash zadáme cestu k našemu programu se souborem .hex.
3. Připojíme programátor do PC a na desku s MCU.
4. Klikneme na Erase – Write – Verify, tím dojde k zapsání programu na desku MCU.

Po nahrání softwaru na desku MCU připojíme LED na výstup označený jako PORTB0 (dbáme na správnou polaritu), připojíme zdroj napájení a zapnutím můžeme program vyzkoušet.

Vypracoval: Radek Zvěřina. Použité materiály: Atmel Studio 6, avrdude.