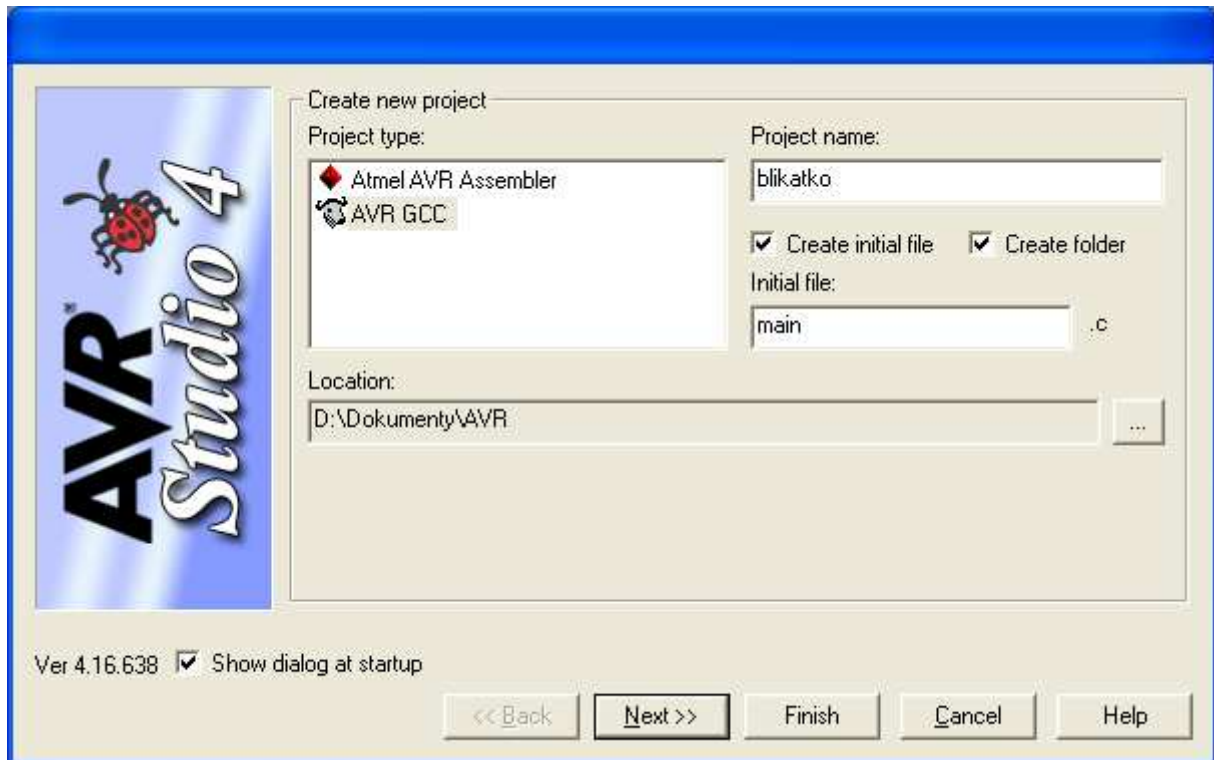
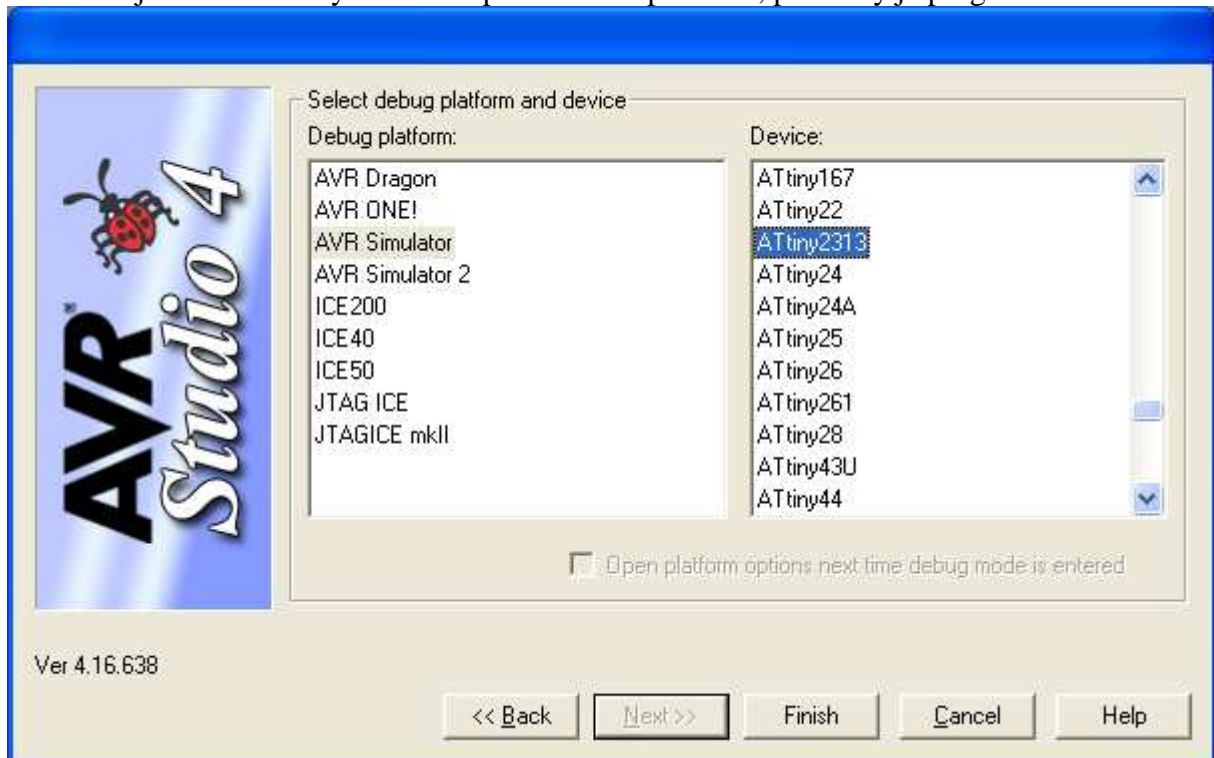


Nový projekt:



Je vhodné, aby se hlavní soubor jmenoval „main.c“. Každý program v C totiž musí obsahovat minimálně 1 funkci nazvanou „main()“ v níž program VŽDY začíná. Když se inicializační soubor jmenuje stejně, usnadňuje to orientaci.

V následujícím okně se vybírá ladící prostředek a procesor, pro který je program tvořen.



To lze kdykoli později změnit v „Debug->Select platform and device“.

V případě využívání časově závislých knihoven (např. při použití funkce Delay()) je potřeba v „Project->Configuration options“ nastavit cílovou frekvenci procesoru. Zde se také nastavují optimalizace překladače.

Nyní je před Vámi otevřené prázdné okno s inicializačním souborem, v našem případě je to „main.c“. Jak je uvedeno výše, každý program v C začíná funkcí „main()“. Do souboru tedy vložíme následující kód:

```
#include <avr/io.h>
// #include <avr/interrupt.h> // odkomentovat při používání přerušení

int main (void)
{
    // vstupní místo do programu
    // zde nebo ve funkci zde volané provést inicializaci mcu

    for(;;)
    {
        // nekonečná smyčka, po inicializaci program skončí zde
    }
    return 0;
}
```

Obsluhu přerušení nebo alespoň její spuštění obstarává v GCC funkce „Interrupt Service Request“ jejíž syntaxe je následující:

```
ISR(požadovaný_vektor_přerušení)
{
    // zde vložený kód se vykoná po vyvolání přerušení
}
```

Pro správnou funkci je třeba na začátek kódu vložit hlavičkový soubor „avr/interrupt.h“:
`#include <avr/interrupt.h>`

Přesný název vektoru přerušení lze najít např. v hlavičkovém souboru daného procesoru. V levém okně rozbalit „External Dependencies“. Použitý procesor je Tiny2313. V tomto případě nás tedy zajímá soubor „ioint2313.h“. V něm lze nalézt následující řádky:

```
/* Timer/Counter1 Overflow */
#define TIMER1_OVF_vect          _VECTOR(5)
```

Právě „TIMER1_OVF_vect“ je název vektoru přerušení pro Timer1 overflow.

Nyní se pokusíme vytvořit blikátko, $f_{\text{CPU}} = 1\text{MHz}$, $f_{\text{BLIK}} \sim 1\text{Hz}$, využití přerušení, timer 1. Program je řízen přerušením, proto nekonečná smyčka v „main“ nic neobsahuje.

```
#include <avr/io.h>
#include <avr/interrupt.h>

void timer1Start(void)
{
    // nastavení a spuštění timeru1
    // delička 8 ->  $1e6/(8*65536) = 1,9$  tj. téměř 2 přetečení za vteřinu
    // to odpovídá frekvenci ~1Hz

    TCCR1B |= (1<<CS11);    // nastaví CS11 do "1", ostatní zůstává na původní
hodnote
                        // clkI/O /8 (From prescaler)
    TIMSK |= (1<<TOIE1);    // Timer/Counter1, Overflow Interrupt Enable
}

int main (void)
{
    // vstupní místo do programu
    // zde nebo ve funkci zde volané provést inicializaci mcu

    DDRB |= (1<<PB0); // pin diody = výstup
    timer1Start();
    sei();           // global. povolení přerušování

    for(;;)
    {
        // nekonečná smyčka, po inicializaci program skončí zde
    }
    return 0;
}

ISR(TIMER1_OVF_vect)
{
    // obsluha přerušení, zde Timer1 overflow
    // dioda připojena na PB0

    if(PINB & (1<<PB0))
    {
        // na pinu diody je "1"
        PORTB &= ~(1<<PB0);    // nastaví PB0 na "0"
    }
}
```

```
else
{
    // na pinu diody je "0"
    PORTB |= 1<<PB0; // nastaví PB0 na "1"
}
}
```

Kompilace a debugging je shodný s použitím asm.