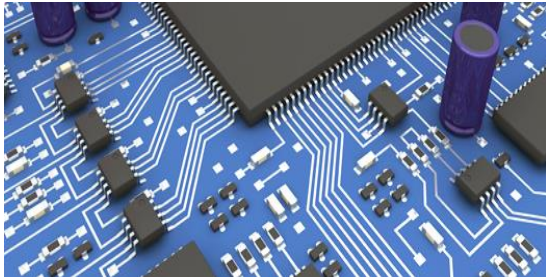


Aplikace Embedded systémů v Mechatronice



Michal Bastl
A2/713a

Aplikace Embedded systémů v Mechatronice

Obsah přednášky:

- Opakování
- Funkce v C
- Tvorba knihoven
- Konfigurační bity `#pragma`
- Makra v C `#define`
- Debugging v MPLAB
- Hardware poznámky



Opakování

K čemu je dobrý koncept bitových masek?

Jaký používáme kompilátor?

Jaké je uplatnění bitového posunu?

Co víme o MCU, které používáme?

Co je hlavní rozdíl v jazyce C oproti MATLABU?

Co je to ASCII?

Jak je v C uchován řetězec znaků?

Komu se podařilo „rozjet“ doma?



podminky if..else

```
//příklad podmínky  
if...else
```

```
int a = 10;  
  
if (a == 10){  
    //function1  
}  
else{  
    //function2  
}
```

```
//příklad podmínky  
if...else
```

```
int a = 10;  
  
if (a == 10){  
    //function1  
}  
else if(a < 10){  
    //function2  
}  
else{  
    //function3  
}
```

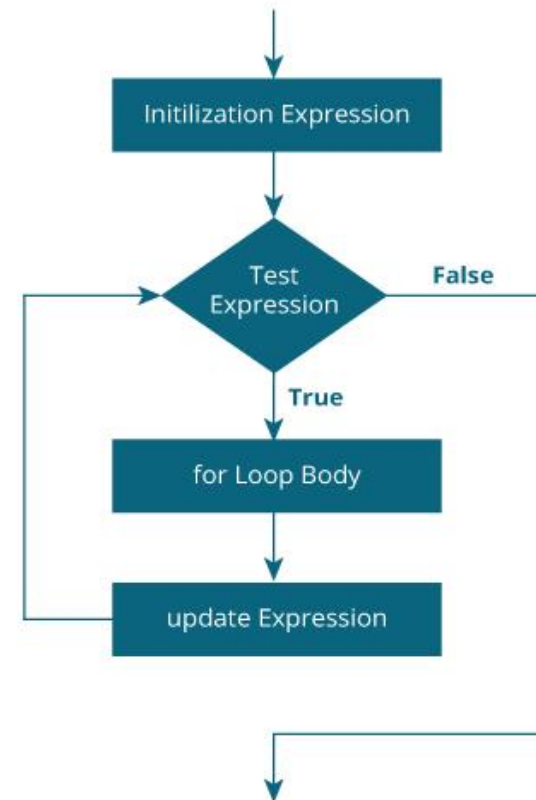
for smyčka

//příklad for smyčky

```
int i;  
  
for (i=0; i < 10; i++){  
    printf( "Ahoj svete" );  
}
```

Kolik vytisknutí na terminál se provede??

For smyčka se používá tehdy, kdy znám dopředu počet cyklů, které chci provést.



while smyčka

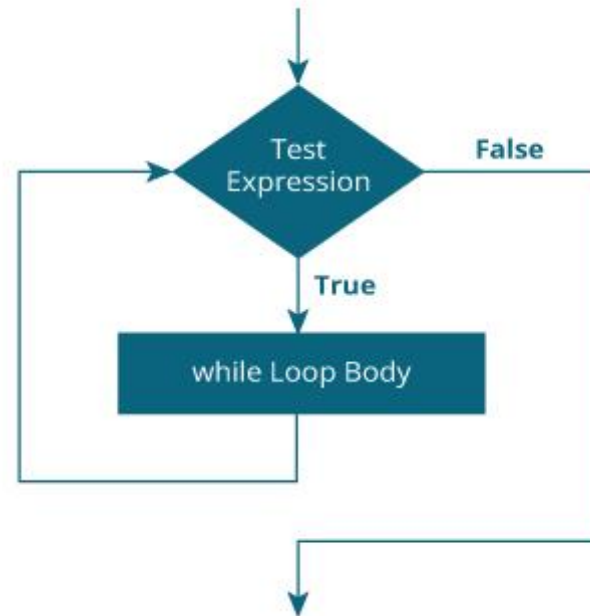
//příklad while
smyčky

```
int a = 10;
```

```
while(a <= 100){  
    a = a * 10;  
}
```

Kolik bude v proměnné a??

While smyčka funguje jinak, jednoduše opakuje blok programu v jeho těle dokud platí podmínka



Funkce v C

- Funkce nemusí vracet hodnotu slovo void, jinak může vracet datové typy např. int, char apod... klíčové slovo return. `return 0;`
- Funkce může přijímat parametry (int a, char, b...) nebo žádné nemá (void)
- Překladač před použitím musí funkci „znát“, případně ví, že funkce někde existuje. Je definovaná .
- Používáme prototyp funkce před prvním použitím. Deklarujeme ji.

```
void putchar(char c);    // prototype pozor na ;
```

```
//definice funkce
```

```
void putchar(char c){  
    bufferToSend = c;  
}
```

Ukázka!

Funkce v C

Deklarace vs. Definice

```
//deklarace (prototyp)
int soucet(int a, int b);
//main
int main(){
    int c;
    c = soucet(10, 5);
    return 0;
}
//definice
int soucet(int a, int b){
    return a + b;
}
```

```
//definice
int soucet(int a, int b){
    return a + b;
}
//main
int main(){
    int c;
    c = soucet(10, 5);
    return 0;
}
```


Vytvoření knihovny

K vytvoření knihovny potřebuji tzv. hlavičkový soubor a skript, kde mám své funkce případně datové typy atd...

- Vytvoříme knihovnu a zavedeme funkce pro součet a odečet dvou celočíselných proměnných.
- `#ifndef` zabraňuje vícenásobnému vložení téhož kódu. Prostředí Vám doplní do .h souboru automaticky

Soubor MyMath.h

```
#ifndef MYMATH_H
#define MYMATH_H

int soucet(int a, int b);
int odecet(int a, int b);

#endif
```

Soubor MyMath.c

```
#include "MyMath.h"

int soucet(int a, int b){
    return a + b;
}

int odecet(int a, int b){
    return a - b;
}
```

Konfigurační bity/pojistky

- Používáme direktivu preprocesoru `#pragma`
- Často obsahuje první nastavení procesoru, jako zdroj hodinového signálu, watchdog timer apod.

Ukázka!

```
#pragma config WDTEN = OFF  
#pragma config FOSC = INTIO7  
#pragma config MCLRE = EXTMCLR  
#pragma config FCMEN = ON
```

```
#pragma config FOSC = XT      // Oscillator Selection bits (XT oscillator)  
#pragma config PLLCFG = OFF   // 4X PLL Enable (Oscillator used directly)  
#pragma config PRICKEN = ON   // Primary clock enable bit (Primary clock is always enabled)  
#pragma config FCMEN = OFF    // Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock Monitor disabled)  
#pragma config IESO = OFF     // Internal/External Oscillator Switchover bit (Oscillator Switchover mode disabled)
```

Debugging

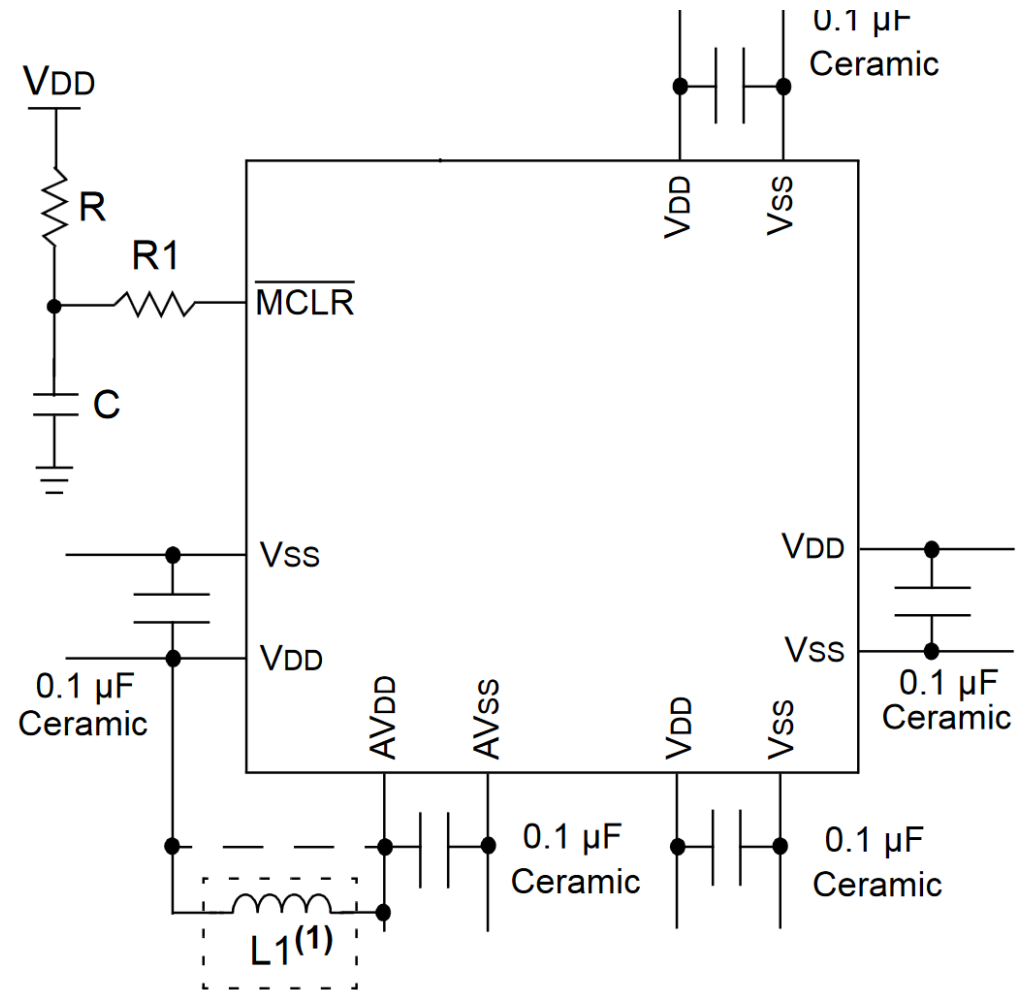
- Slouží k ladění programu
- V prostředí MPLAB lze s pomocí PICkit3
- krokování programu
- Breakpointy
- Pozorování proměnných a registrů v procesoru

Ukázka!

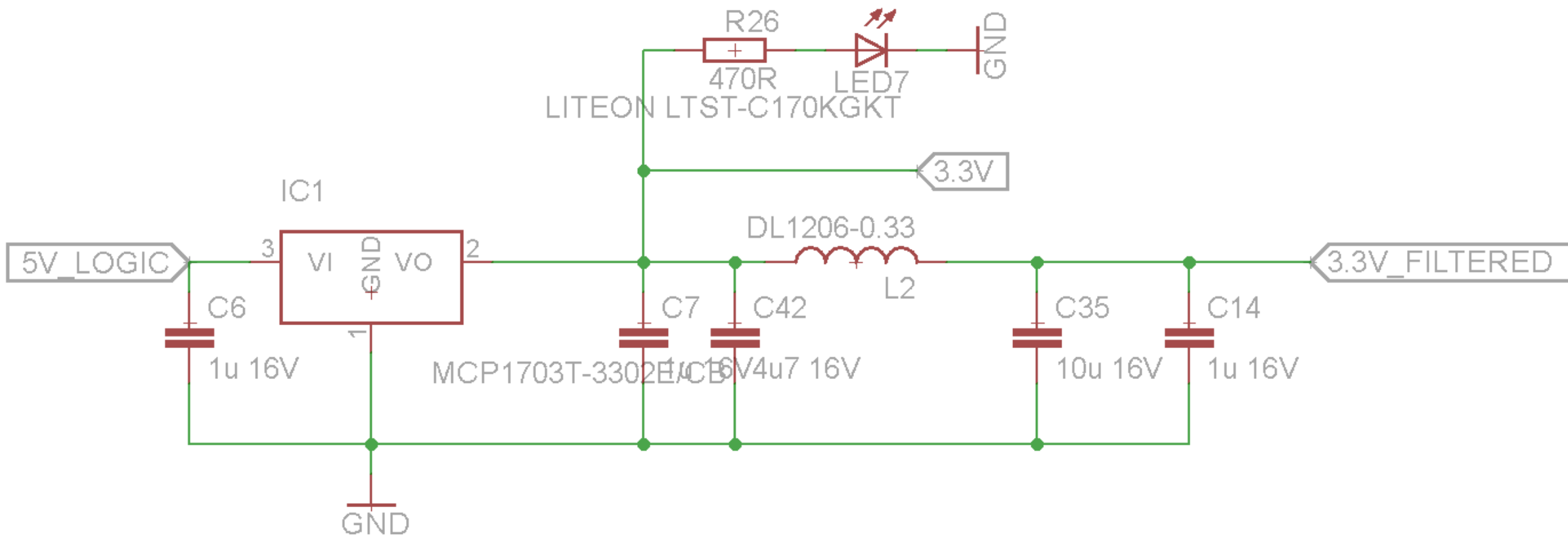


Zapojení PIC18

- Stabilizovaný zdroj napětí (nejlépe LDO 1%)
- Blokovací kondenzátory napájecích pinů
- Přesné napětí pro AVdd
- Koncept pull-up a pull-down rezistorů
- Omezení proudu = ochrana I/O pinů

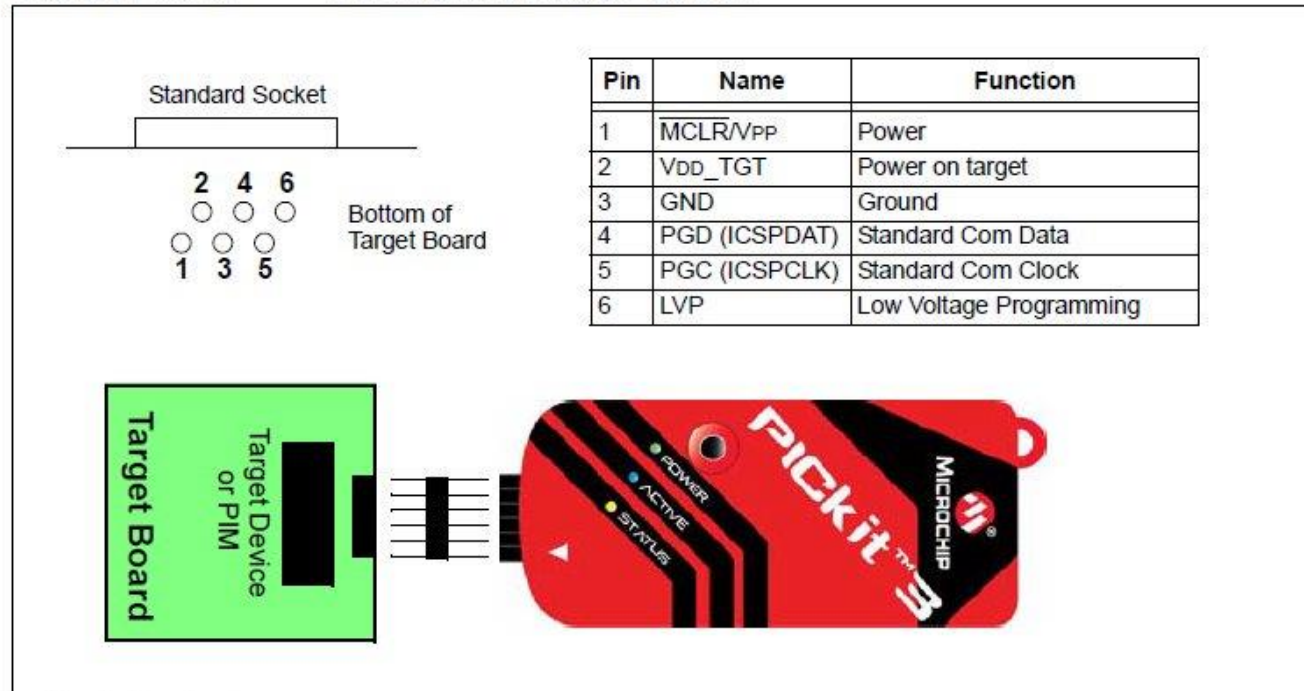


Zapojení PIC18

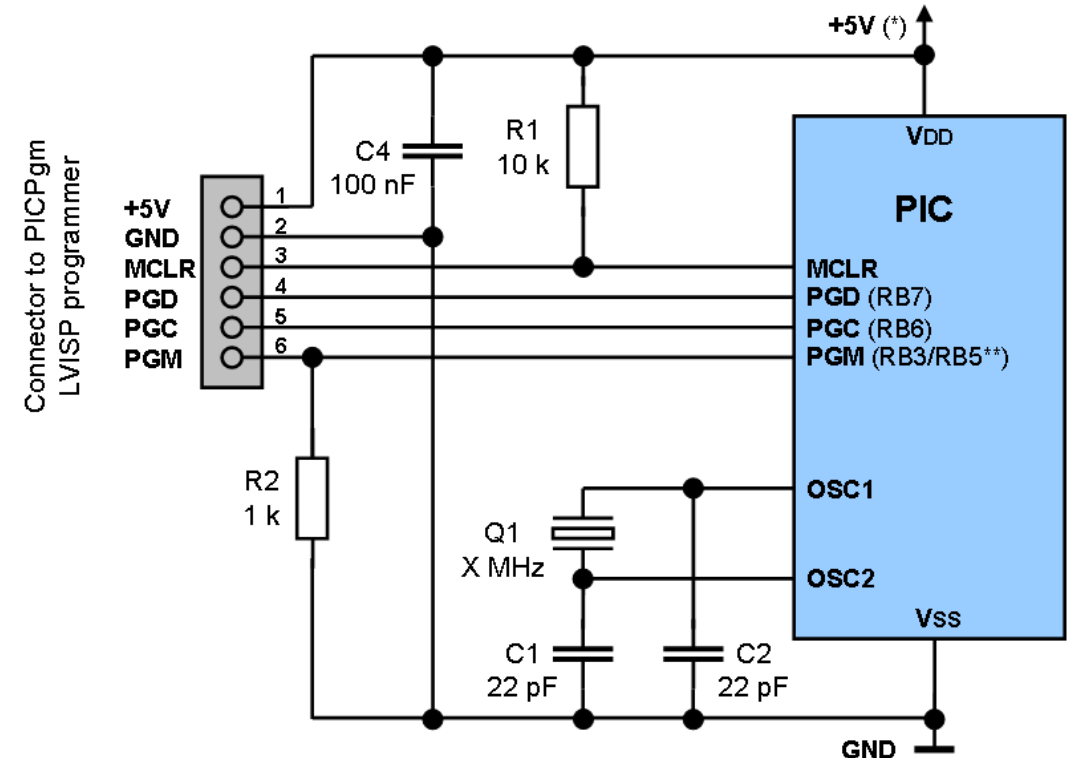


Zapojení PIC18

FIGURE 10-1: 6-PIN STANDARD PINOUT



Microchip Technology Inc.



* Connect to +5V power supply

** PGM pin is RB3 or RB5. This depends on the PIC you are using. Check PIC datasheet for details.

picpgm_lvisp_connect2target.ppt, © Christian Stadler, v1.0, 23.08.2010