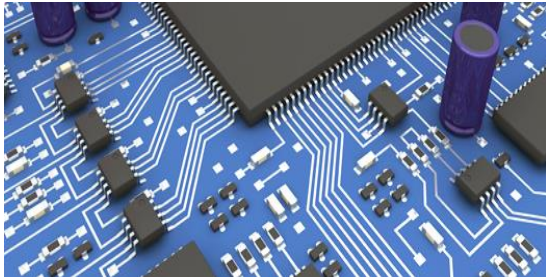


Aplikace Embedded systémů v Mechatronice



Michal Bastl
A2/713a

Aplikace Embedded systémů v Mechatronice

Obsah přednášky:

- Organizace předmětu
- EduKit
- Software
- číselné soustavy
- Datové typy
- ASCII
- Struktura C programu
- Příklad kódu



Aplikace Embedded systémů v Mechatronice

Přednáška: nepovinná, ale předchází cvičení (Pojmy vysvětlené na přednášce budeme považovat za probrané)

Cvičení: 3-4 x základy jazyka C; 8 x MCU PIC18

Hodnocení předmětu 2019

- | | |
|-------------------------------------|-----|
| • 3 domácí úlohy v průběhu semestru | 15b |
| • Závěrečný písemný test | 50b |
| • Semestrální projekt | 35b |

[Blok 1](#) – první program, vstup/výstup na terminál, proměnné, řídicí struktury

[Blok 2](#) – funkce

[Blok 3](#) – ukazatele, pole, řetězce

[Blok 4](#) – struktury

[Blok 5](#) – PIC18, I/O

[Blok 6](#) – PIC18, čítače, přerušení

[Blok 7](#) – PIC18, UART + STDIO

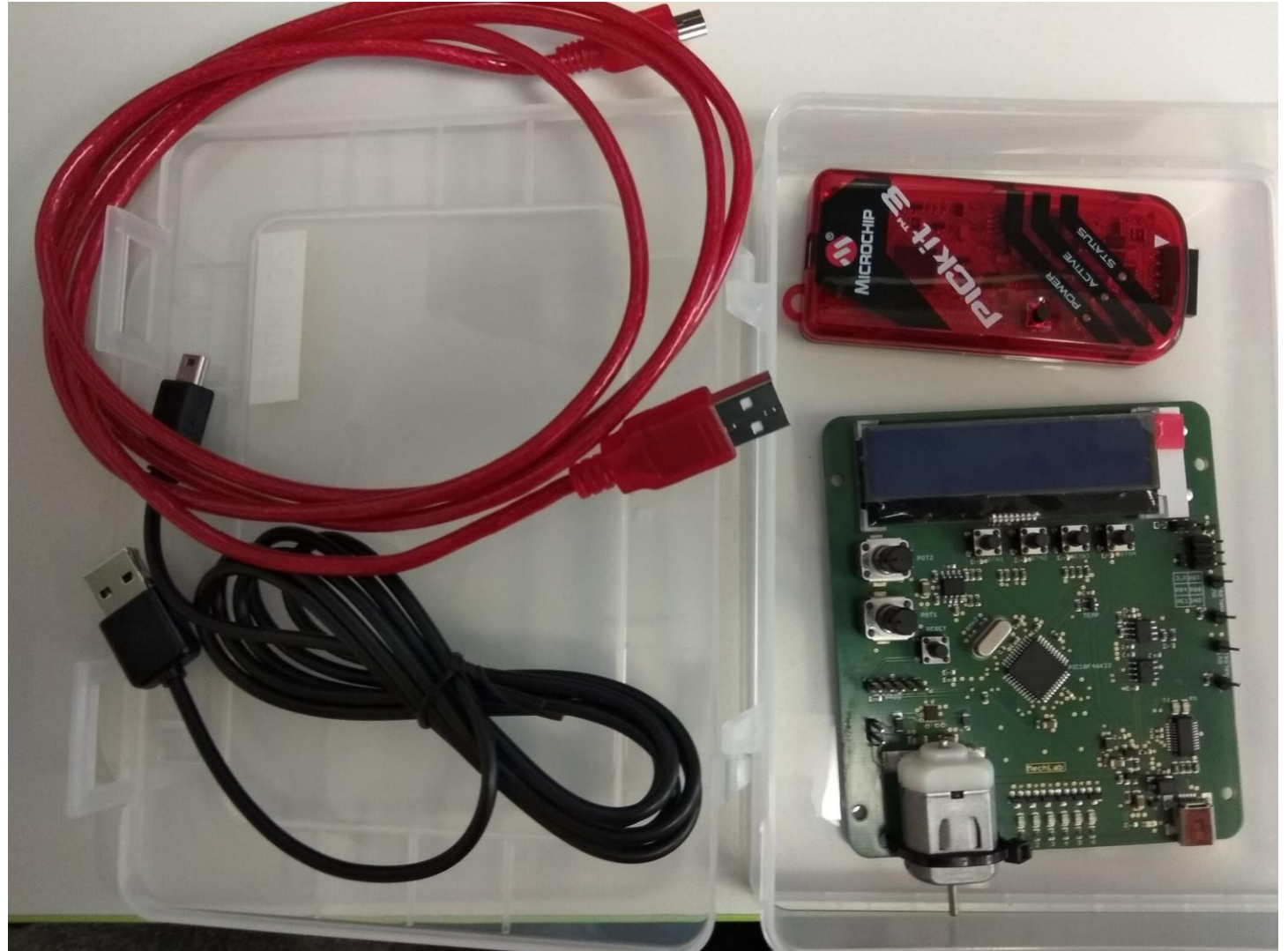
[Blok 8](#) – PIC18, ADC, DAC

[Blok 9](#) – PIC18, PWM

Aplikace Embedded systémů v Mechatronice

Součásti EduKitu:

- Krabice
- Programátor PICKIT3
- 2x USB kabel
- Edukit

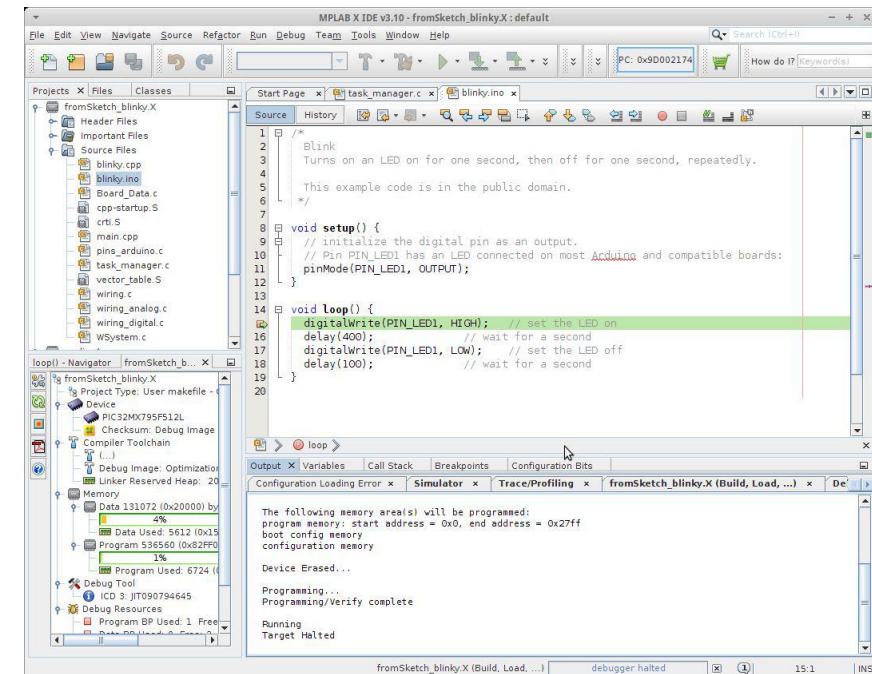
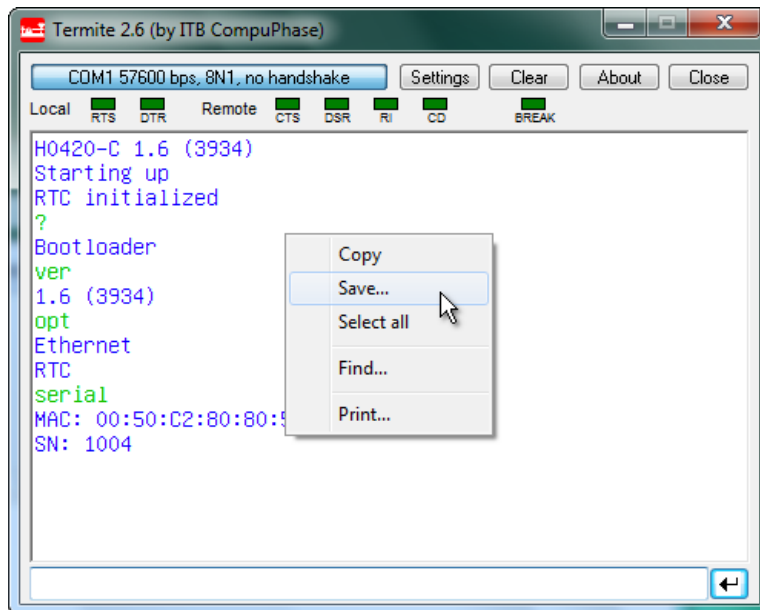


Ukázka potřebných nástrojů

Software:

- Microchip MPLAB
- Termit

vývojové prostředí
sériový terminál



Hlavní rozdíly mezi MATLAB a C:

C

- Kompilovaný jazyk
- Datové typy musí určit uživatel
- Přímý přístup do paměti
- málo klíčových slov
- Poměrně blízko hardwaru
- Oproti ASM, nebo .HEX je vzniklý kód čitelný (pokud byl programátor slušný : -)
- Je možné natropit chyby, které kompilátor nemůže ověřit a špatně se potom hledají
- Indexuje od 0

MATLAB

- Interpretovaný jazyk
- Datové typy určí interpreter
- Uživatel nemá přímý přístup do paměti
- Mnoho funkcí a klíčových slov
- Propast mezi hardwarem a programátorem je větší
- Je dobře čitelný, vývoj je rychlejší
- Interpreter pracuje s pamětí sám. Nelze udělat zásadní chyby a jejich hledání je jednoduché
- Indexuje od 1

Číselné soustavy

Dekadická:	1052		$10^3 10^2 10^1 10^0$
Binární:	1010	zápis v C 0b1010	$2^3 2^2 2^1 2^0$
Hexadecimální:	A7	zápis v C 0xA7	$16^1 16^0$

Uvnitř MCU se používá pouze binární. Hexadecimální soustava se používá ke zkrácení zápisu. Jeden znak Hexadecimální soustavy odpovídá 4 bitům.

Překladači je samozřejmě jedno jaký zápis použijete, ale například nastavit 32 bitový registr binárním zápisem je poněkud nepřehledné.

Například hodnota 200 v DEC je v BIN 0b11001000
v HEX 0xC8

Datové typy v jazyce C

TABLE 5-1: INTEGER DATA TYPES

Type	Size (bits)	Arithmetic Type
bit	1	Unsigned integer
signed char	8	Signed integer
<u>unsigned char</u>	8	Unsigned integer
<u>signed short</u>	16	Signed integer
unsigned short	16	Unsigned integer
<u>signed int</u>	16	Signed integer
unsigned int	16	Unsigned integer
<u>signed short long</u>	24	Signed integer
unsigned short long	24	Unsigned integer
<u>signed long</u>	32	Signed integer
unsigned long	32	Unsigned integer
signed long long	32	Signed integer
unsigned long long	32	Unsigned integer

TABLE 5-5: FLOATING-POINT DATA TYPES

Type	Size (bits)	Arithmetic Type
float	24 or 32	Real
double	24 or 32	Real
long double	same as double	Real

Bitové operátory

Využití:

Rychlé násobení a dělení >> bitový posuv:

$A \ll n$ (násobení 2^n) ; $A \gg n$ (dělení 2^n)

Masky:

AND, OR a případně XOR

&		^
10011110	10011110	10011110
<u>00001111</u>	<u>00000001</u>	<u>00001111</u>
00001110	10011111	10010001

Symbol	Operator
&	bitwise AND
	bitwise inclusive OR
^	bitwise XOR (eXclusive OR)
<<	left shift
>>	right shift
~	bitwise NOT (one's complement) (unary)

ASCII tabulka

- Zápis v jazyce C pomocí jednoduchých uvozovek '0'
- Používáme datový typ char 0..255
- V původním rozsahu má 128 znaků (rozšířená pak 255)
- Pro více znaků se používá pole char[x]
- Pro uložení řetězce znaků „ahoj“
- char pozdrav[5] = "ahoj";
- char pozdrav[5] = {'a', 'h', 'o', 'j', '\0'}; v c je třeba o bit více pro znak \0 - null character

Příklad:

- Chci vypisovat číslce 0..9
- ```
char c = 2;
printf("cislice je %c", c + '0');
```

| Dec | Hx | Oct | Char                        | Dec | Hx | Oct | Html  | Chr   | Dec | Hx | Oct | Html  | Chr | Dec | Hx | Oct | Html   | Chr |
|-----|----|-----|-----------------------------|-----|----|-----|-------|-------|-----|----|-----|-------|-----|-----|----|-----|--------|-----|
| 0   | 0  | 000 | NUL (null)                  | 32  | 20 | 040 | &#32; | Space | 64  | 40 | 100 | &#64; | @   | 96  | 60 | 140 | &#96;  | `   |
| 1   | 1  | 001 | SOH (start of heading)      | 33  | 21 | 041 | &#33; | !     | 65  | 41 | 101 | &#65; | A   | 97  | 61 | 141 | &#97;  | a   |
| 2   | 2  | 002 | STX (start of text)         | 34  | 22 | 042 | &#34; | "     | 66  | 42 | 102 | &#66; | B   | 98  | 62 | 142 | &#98;  | b   |
| 3   | 3  | 003 | ETX (end of text)           | 35  | 23 | 043 | &#35; | #     | 67  | 43 | 103 | &#67; | C   | 99  | 63 | 143 | &#99;  | c   |
| 4   | 4  | 004 | EOT (end of transmission)   | 36  | 24 | 044 | &#36; | \$    | 68  | 44 | 104 | &#68; | D   | 100 | 64 | 144 | &#100; | d   |
| 5   | 5  | 005 | ENQ (enquiry)               | 37  | 25 | 045 | &#37; | %     | 69  | 45 | 105 | &#69; | E   | 101 | 65 | 145 | &#101; | e   |
| 6   | 6  | 006 | ACK (acknowledge)           | 38  | 26 | 046 | &#38; | &     | 70  | 46 | 106 | &#70; | F   | 102 | 66 | 146 | &#102; | f   |
| 7   | 7  | 007 | BEL (bell)                  | 39  | 27 | 047 | &#39; | '     | 71  | 47 | 107 | &#71; | G   | 103 | 67 | 147 | &#103; | g   |
| 8   | 8  | 010 | BS (backspace)              | 40  | 28 | 050 | &#40; | (     | 72  | 48 | 110 | &#72; | H   | 104 | 68 | 150 | &#104; | h   |
| 9   | 9  | 011 | TAB (horizontal tab)        | 41  | 29 | 051 | &#41; | )     | 73  | 49 | 111 | &#73; | I   | 105 | 69 | 151 | &#105; | i   |
| 10  | A  | 012 | LF (NL line feed, new line) | 42  | 2A | 052 | &#42; | *     | 74  | 4A | 112 | &#74; | J   | 106 | 6A | 152 | &#106; | j   |
| 11  | B  | 013 | VT (vertical tab)           | 43  | 2B | 053 | &#43; | +     | 75  | 4B | 113 | &#75; | K   | 107 | 6B | 153 | &#107; | k   |
| 12  | C  | 014 | FF (NP form feed, new page) | 44  | 2C | 054 | &#44; | ,     | 76  | 4C | 114 | &#76; | L   | 108 | 6C | 154 | &#108; | l   |
| 13  | D  | 015 | CR (carriage return)        | 45  | 2D | 055 | &#45; | -     | 77  | 4D | 115 | &#77; | M   | 109 | 6D | 155 | &#109; | m   |
| 14  | E  | 016 | SO (shift out)              | 46  | 2E | 056 | &#46; | .     | 78  | 4E | 116 | &#78; | N   | 110 | 6E | 156 | &#110; | n   |
| 15  | F  | 017 | SI (shift in)               | 47  | 2F | 057 | &#47; | /     | 79  | 4F | 117 | &#79; | O   | 111 | 6F | 157 | &#111; | o   |
| 16  | 10 | 020 | DLE (data link escape)      | 48  | 30 | 060 | &#48; | 0     | 80  | 50 | 120 | &#80; | P   | 112 | 70 | 160 | &#112; | p   |
| 17  | 11 | 021 | DC1 (device control 1)      | 49  | 31 | 061 | &#49; | 1     | 81  | 51 | 121 | &#81; | Q   | 113 | 71 | 161 | &#113; | q   |
| 18  | 12 | 022 | DC2 (device control 2)      | 50  | 32 | 062 | &#50; | 2     | 82  | 52 | 122 | &#82; | R   | 114 | 72 | 162 | &#114; | r   |
| 19  | 13 | 023 | DC3 (device control 3)      | 51  | 33 | 063 | &#51; | 3     | 83  | 53 | 123 | &#83; | S   | 115 | 73 | 163 | &#115; | s   |
| 20  | 14 | 024 | DC4 (device control 4)      | 52  | 34 | 064 | &#52; | 4     | 84  | 54 | 124 | &#84; | T   | 116 | 74 | 164 | &#116; | t   |
| 21  | 15 | 025 | NAK (negative acknowledge)  | 53  | 35 | 065 | &#53; | 5     | 85  | 55 | 125 | &#85; | U   | 117 | 75 | 165 | &#117; | u   |
| 22  | 16 | 026 | SYN (synchronous idle)      | 54  | 36 | 066 | &#54; | 6     | 86  | 56 | 126 | &#86; | V   | 118 | 76 | 166 | &#118; | v   |
| 23  | 17 | 027 | ETB (end of trans. block)   | 55  | 37 | 067 | &#55; | 7     | 87  | 57 | 127 | &#87; | W   | 119 | 77 | 167 | &#119; | w   |
| 24  | 18 | 030 | CAN (cancel)                | 56  | 38 | 070 | &#56; | 8     | 88  | 58 | 130 | &#88; | X   | 120 | 78 | 170 | &#120; | x   |
| 25  | 19 | 031 | EM (end of medium)          | 57  | 39 | 071 | &#57; | 9     | 89  | 59 | 131 | &#89; | Y   | 121 | 79 | 171 | &#121; | y   |
| 26  | 1A | 032 | SUB (substitute)            | 58  | 3A | 072 | &#58; | :     | 90  | 5A | 132 | &#90; | Z   | 122 | 7A | 172 | &#122; | z   |
| 27  | 1B | 033 | ESC (escape)                | 59  | 3B | 073 | &#59; | ;     | 91  | 5B | 133 | &#91; | [   | 123 | 7B | 173 | &#123; | {   |
| 28  | 1C | 034 | FS (file separator)         | 60  | 3C | 074 | &#60; | <     | 92  | 5C | 134 | &#92; | \   | 124 | 7C | 174 | &#124; |     |
| 29  | 1D | 035 | GS (group separator)        | 61  | 3D | 075 | &#61; | =     | 93  | 5D | 135 | &#93; | ]   | 125 | 7D | 175 | &#125; | }   |
| 30  | 1E | 036 | RS (record separator)       | 62  | 3E | 076 | &#62; | >     | 94  | 5E | 136 | &#94; | ^   | 126 | 7E | 176 | &#126; | ~   |
| 31  | 1F | 037 | US (unit separator)         | 63  | 3F | 077 | &#63; | ?     | 95  | 5F | 137 | &#95; | _   | 127 | 7F | 177 | &#127; | DEL |

# Příklady

Načtení standardních knihoven

Načtení knihovny překladače XC8

Načtení vlastních knihoven REV

Direktivy nastavující pojistky tzv. configuration bits

Funkce main, která má zásadní postavení. Může být jen jedna a program zde „začíná“

Proměnné, které v programu použijeme

REV\_init() inicializuje potřebná nastavení procesoru

Nekonečný cyklus ekvivalent zápisu while(1)

Inkrementování count = count + 1

Funkce ovládající LED diody kitu

Odeslání zprávy na sériový port

Počkej 100ms

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <stdint.h>
4 #include <xc.h>
5
6 #include "rev-basic.h"
7 #include "simdelay.h"
8
9 #pragma config WDTCN = OFF
10 #pragma config FOSC = INTIO7
11 #pragma config MCLRE = EXTMCLR
12 #pragma config FCEN = ON
13
14 void main(void) {
15 char count = 0;
16 int i;
17 char a = 'x';
18
19 REV_init();
20
21 for(;;) {
22 count++;
23
24 REV_led(1, count & 1);
25 REV_led(2, count & 2);
26 REV_led(3, count & 4);
27
28 printf("count: %d (0x%x) (%c)\n", count, count, a);
29
30 DelayMs(100);
31 }
32 }
```