Introduction to C programming

Basic examples for the chipkit mx4 board

Dimitri de Smet

UCL

22 septembre 2017

Reminders

- Computer architecture
- Code/compile/program
- Your computer board

Today's plan

- C language examples
- Tools installation and exercices

Softwares

- MPLAB X IDE
- XC32 compiler

Download and intall the last version under the “Download Archive” tab.

Example 0

```
#include "main.h"
#include <stdio.h>

#include "delay.h"
#include "PmodCLP.h"
#include "analogInputs.h"

void main (void)
{
    ex16(); // Change it from ex01 to ex17
}
```

Lessons

1. Comments : *//*
2. Main function : *void main(void)*
3. An instruction line ends with ‘ ;’
4. Functions and levels of abstraction
    - Hierarchy

Example 1

```
void ex01(void)
{
    initIO();
    setLeds(0xa);
}
```

Lessons

1. Digital outputs
2. Arguments of a function between '( )'
3. Hexa notation
4. Levels of abstraction
    - Modularity
5. Function initIO()

Example 2

```
void ex02(void)
{
    initIO();
    if(getButton1())
        setLeds( 0b0011 );
    else
        setLeds( 0b1100 );
}
```

Lessons

1. Digital intputs
2. *if(. . .) . . . else*
3. Binary notation
4. The *main* function is executed once

Example 3

```
void ex03(void)
{
    initIO();
    while(1){
        if(getButton1())
            setLeds( 0b0011 );
        else
            setLeds( getButton2() );
    }
}
```

Lessons

1. while(1) = infinite loop
2. *setLeds(getButton2());*

Example 4

```
void ex04(void)
{
    char evenBits = 0x0a;
    char oddBits = 0x05;

    initIO();
    while(1){
        if( getButton1() && getButton2() )
            setLeds(evenBits);
        else
            setLeds(oddBits);
    }
}
```

Lessons

1. Variable declaration (here type *char*)
2. Logic operators

Opérations logiques		
&&	AND	`res = op1 && op2;`
||	OR	`res = op1 || op2;`
!	NOT	`res = !op;`

Opérations conditionnelles		
==	Est égal à	`res = op1 == op2;`
>=	Plus grand ou égal à	`res = op1 >= op2;`
<=	Plus petit ou égal à	`res = op1 <= op2;`
!=	Pas égal à	`res = op1 != op2;`
>	Plus grand que (strict.)	`res = op1 > op2;`
<	Plus petit que (strict.)	`res = op1 < op2;`

- ? If($(x > 4)$ && $(x < 10)$)

Example 5

```
void ex05(void){
    char leds = 0xa;

    initIO();
    while(1){
        leds = leds^0x3;
        setLeds(leds);
    }
}
```

Lessons

1. Typical structure of our programs
2. CPU frequency : 10 MHz
3. Bitwise operators

Opérations bit-à-bit		
&	Bitwise AND	`res = op1 & op2;`
|	Bitwise OR	`res = op1 | op2;`
~	Bitwise NOT	`res = ~op;`
^	Bitwise XOR	`res = op1 ^ op2;`
>>	Logical right shift	`res = op1 >> int1;`
<<	Logical left shift	`res = op1 << int1;`

- ??
  - leds = 0xaa & 0xf0
  - leds = 0xaa && 0xf0

Example 6

```
void ex06(void){
    char leds = 0b1010;
    long cpt;

    initIO();

    while(1){
            leds = leds^0b0011;
            setLeds(leds);
            for(cpt = 0; cpt<200000; cpt++)
            {
            }
    }
}
```

Lessons

1. Execution time

Example 7 (1/2)

```
void ex07(void){
    char leds = 0;
    initIO();
    while(1){
        leds = leds^0xf;
        setLeds(leds);
        loseSomeTime(200000);
    }
```

Example 7 (2/2)

```
void loseSomeTime(long cptMax)
{
        long cpt;
        for(cpt = 0; cpt<cptMax; cpt++){   }
}
```

Lessons

1. Modularity

Example 8

```
void ex08(void){
    char leds = 0;

    initIO();
    initDelay();
    leds = 0;
    while(1){
        leds = leds^0xf;
        setLeds(leds);
        DelayMs(300);
    }
}
```

Lessons

1. New functions provided by *delay.h*
   - *initDelay()*
   - *DelayMs(int tms)*
   - *DelayUs(short tus)*

Example 9

```
void ex09(void){
    unsigned short adc;
    initIO();
    initAnalogInputs(0x01);

    while(1){
        adc = readADC(0);
        setLeds(adc/64);
    }
}
```

Lessons

1. Analog inputs
2. 10-bits conversion
3. 4-bits display

Example 10

```
void ex10(void)
{
    char strA[16] = "␣␣␣LLSMF2018␣␣␣␣";
    char strB[16] = "␣␣␣␣Demo␣code␣␣␣";

    initIO();
    initLCD();
    writeLine(strA, 0);
    writeLine(strB, 1);
}
```

Lessons

1. 16 x 2 characters screen
2. Table of variable (here table of char)
3. functions provided by *PmodCLP.h*
    - *void initLCD();*
    - *void writeLine(char * string, char line);*
    - *void clearScreen();*
    - *void shiftScreen(unsigned char right);*

Example 11

```
void ex11(void){
    unsigned short adc;
    char txt[16];
    initIO();
    initAnalogInputs(0x01);
    initLCD();
    while(1){
        adc = readADC(0);
        setLeds(adc/64);
        sprintf(txt,"ADC : %4d         ", adc);
        writeLine(txt,0);
        sprintf(txt,"  U = %1.1f Volts ", (float)
        writeLine(txt,1);
    }
}
```

Lessons

1. function *sprintf()*
   - doc ==>**here**<==

Example 12

```
void ex12(void){
    char leds = 0;
    short adc;
    initIO();
    initDelay();
    initAnalogInputs(1);
    while(1){
            DelayMs( readADC(0) );
            leds ^= 0xf;
            setLeds(leds);
    }
}
```

Lessons

1. Frequency command

Example 13

```
void ex13(void){
    initIO(); initDelay();
    while(1){
        if(getButton2()){
            setLeds(0xff);
            DelayUs( 40 );
            setLeds(0x00);
            DelayUs( 10 );  }
        else{
            setLeds(0xff);
            DelayUs( 10 );
            setLeds(0x00);
            DelayUs( 40 );  }
    }
}
```

Lessons

1. Duty cycle command

Example 14

```
void ex14(void){
    initIO();
    initDelay();
    initAnalogInputs(1);

    while(1){
            setLeds(0xf);
            DelayUs( (long)(readADC(0)) );
            setLeds(0x0);
            DelayUs( 1023-(long)(readADC(0)) );
    }
}
```

Lessons

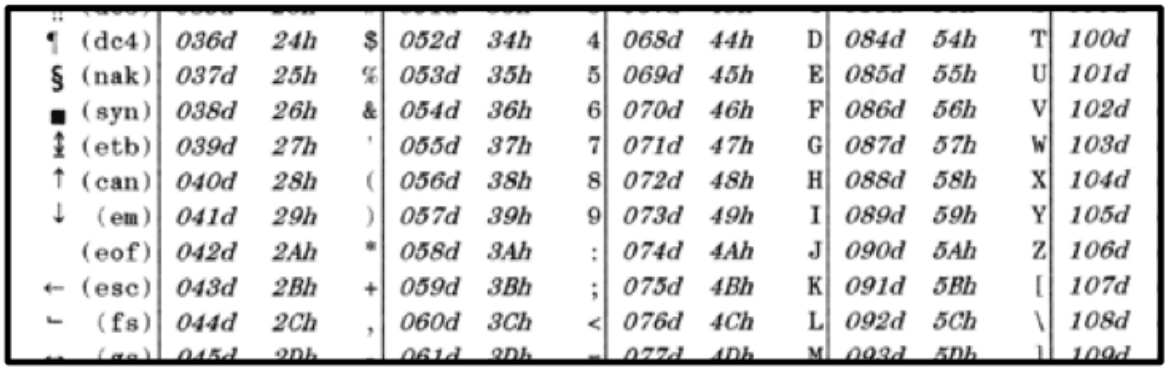
1. “Analog” output
2. Dimmer

Example 15

```
void ex15(void){
    char leds;
    initIO();
    while(1){
        leds = 'G';
        setLeds(leds+1);
    }
}
```

Lessons

1. ascii code



¶	(dc4)	036d	24h	$	052d	34h	4	068d	44h	D	084d	54h	T	100d
§	(nak)	037d	25h	%	053d	35h	5	069d	45h	E	085d	55h	U	101d
■	(syn)	038d	26h	&	054d	36h	6	070d	46h	F	086d	56h	V	102d
↨	(etb)	039d	27h	'	055d	37h	7	071d	47h	G	087d	57h	W	103d
↑	(can)	040d	28h	(	056d	38h	8	072d	48h	H	088d	58h	X	104d
↓	(em)	041d	29h	)	057d	39h	9	073d	49h	I	089d	59h	Y	105d
	(eof)	042d	2Ah	*	058d	3Ah	:	074d	4Ah	J	090d	5Ah	Z	106d
←	(esc)	043d	2Bh	+	059d	3Bh	;	075d	4Bh	K	091d	5Bh	[	107d
∟	(fs)	044d	2Ch	,	060d	3Ch	<	076d	4Ch	L	092d	5Ch	\	108d
	(gs)	045d	2Dh	-	061d	3Dh	=	077d	4Dh	M	093d	5Dh	]	109d

Example 16 (1/2)

```
void ex16(void){
    signed char i;
    unsigned char table[5]={0b0000, \
            0b1000, 0b1100, 0b1110, \
            0b1111};

    initIO();
    initDelay();
    initAnalogInputs(1);
```

Example 16 (2/2)

```
        setLeds(0);
        while(1){
            for (i=0; i<5; i++){
                DelayMs(readADC(0)+100 );
                setLeds(table[i]);
            }
        }
}
```

Lessons

1. Table
   - declaration and use

Example 17 (1/3)

```
void ex17(void){
    long a =0 , b = 0, c = 0, d = 0, op = 0;
    char selector=0;
    char formula[16], results[16];

    initIO();
    initDelay();
    initAnalogInputs(1);
    initLCD();

    while(1){
```

Example 17 (2/3)

```
setLeds(selector);
if(getButton2()){
    selector++;
    if(selector>2)
        selector = 0;
    DelayMs(500);
}
switch(selector){
    case 0 :
        a = (long) readADC(0); break;
    case 1 :
        b = (long) readADC(0); break;
    case 2 :
        c = (long) readADC(0); break;
}
```

Example 17 (3/3)

```
            if( b < 300){
                op = '+';
                d = a+c;}
            else if(b < 600){
                op = '-';
                d = a-c;}
            else{
                op = '*';
                d = a*c;}
            sprintf(formula,"␣%4d␣␣%c␣␣%4d␣␣",a,op,c);
            sprintf(results,"␣=␣%7d", d);
            writeLine(formula,0);
            writeLine(results,1);
        }
    }
```

Lessons

- A calculator
- Type casting