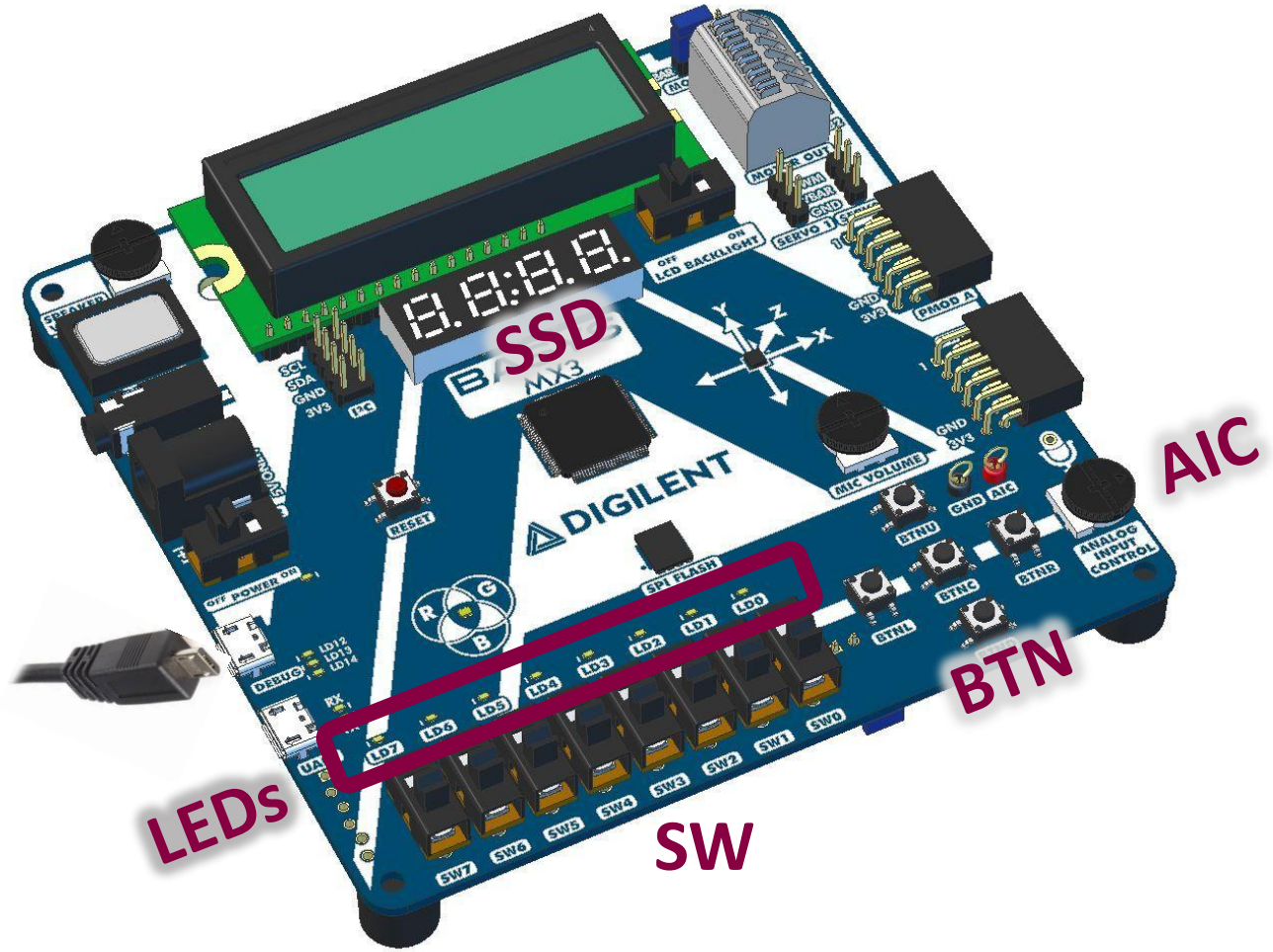
TALLINNA
TEHNIKAÜLIKOOL

Microcontroller „Basys MX3“ and TASKs

Hardi Selg

22.11.2018

What we will learn to program



Simple test program

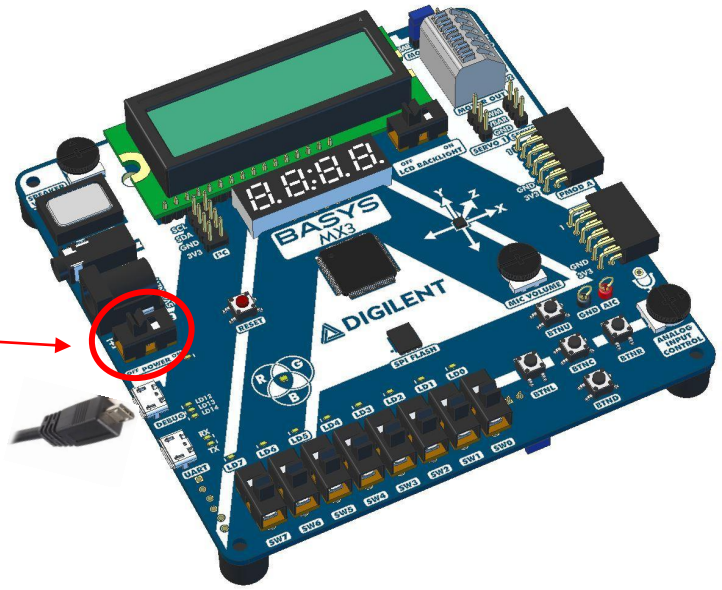
Source | History

```
22
23  //Libraries
24  #include <xc.h>
25  #include <sys/attribs.h>
26  #include "config.h"
27
28  #define DELAY_IN_MSEC_50     50
29  #define DELAY_IN_MSEC_100   100
30  #define DELAY_IN_MSEC_500   500
31
32  //Main program
33  int main(void) {
34      //Has to be the first function call after main()
35      init(); //Includes PIC16F690 basic configuration
36      //Loop forever
37      while(1)
38      {
39          //Write your code here
40          if(BTND == 1){
41              LED0 = 1;
42          }
43          else{
44              LED0 = 0;
45          }
46          DelayForAproxmSeconds(DELAY_IN_MSEC_100);
47
48      }
49      return 0;
50  }
```

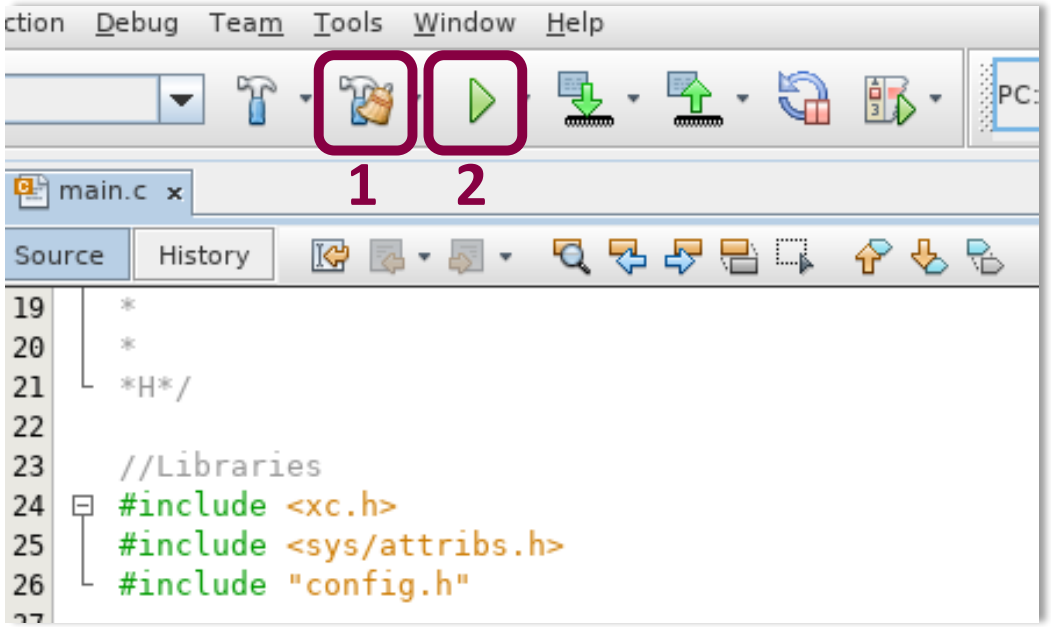
- When you push button „**BTND**“ down, LED “**LD0**” should light up.
- When you release it, it will dim out.

!NB! – The while(1) loop should always have at least **one** delay.

Running your code



Make sure that the power switch is **ON**



Running you code consists of 2 stages:

1. 'Clean and Build' Project
2. Run Project

Using 7-segment indicators


```
WriteDigits(SEGM3, 15, DOT_ON);
```

```
WriteDigits(SEGM0, 9, DOT_OFF);
```

OR

SEGM3 SEGM0

```
SSD_WriteDigits(15, 0, 0, 9, DOT_ON, 0, 0, 0);
```

To assign values to indicator, call the function **`WriteDigits`** with 3 parameters:

1. Segment name (SEGM3)
2. Numeric value to be displayed (10)
3. Choose whether the „DOT" is **ON** or **OFF**

OR

Use **`SSD_WriteDigits`** to assign values to Segment indicators at the same time

- Numeric values can be in:

  binary: 0b1010

  decimal: 11

  hex: 0x0C

Additional values for indicators

Symbol	Symbol value
NULL (all segments are off)	16
- (minus)	17
FULL (all segments are on)	18
H	19
L	20
P	21
I	22
U	23
N	24

Example program

Source | History

```
31
32  //Main program
33  int main(void) {
34      //Has to be the first function call after main()
35      init(); //Includes PIC16F690 basic configuration
36      int value;
37      //Loop forever
38      while(1)
39      {
40          //Write your code here
41          value = SW0 + SW1 + SW2 + SW3 + SW4 + SW5 + BTNL;
42          LED_SetGroupValue(value);
43
44          WriteDigits(SEGM0,  0b1100, SW6);
45          WriteDigits(SEGM1,  0x0B,   SW7);
46          WriteDigits(SEGM2,  10,     BTNL);
47          WriteDigits(SEGM3,  0,      DOT_ON);
48
49          DelayForAproxmSeconds(DELAY_IN_MSEC_100);
50
51      }
52      return 0;
53  }
```

Explanation:

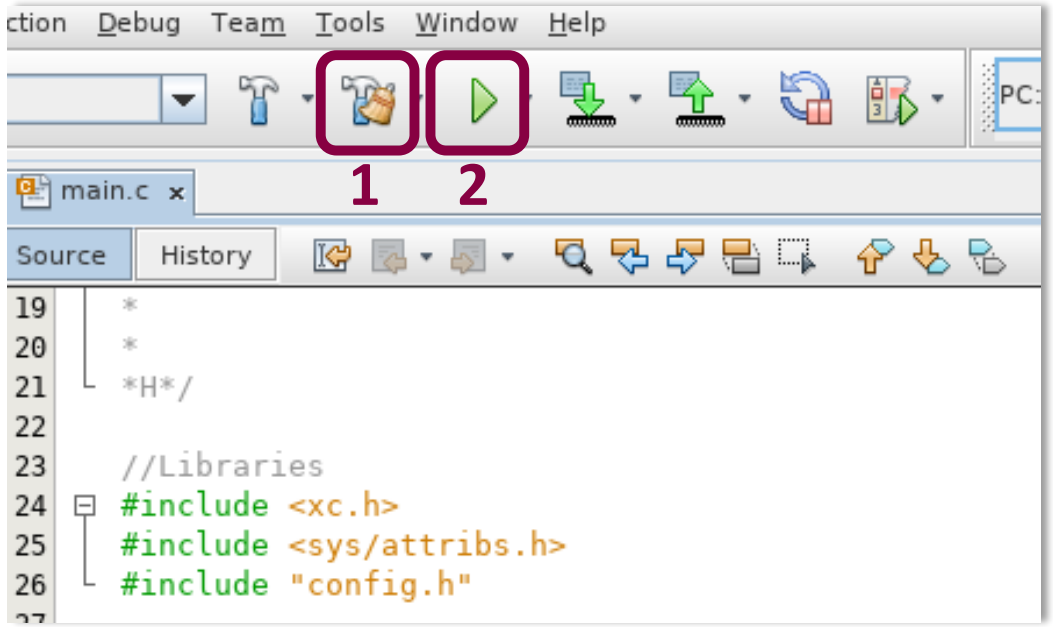
- function **LED_SetGroupValue** displays any given (parameter) numeric value as a binary number on LEDs

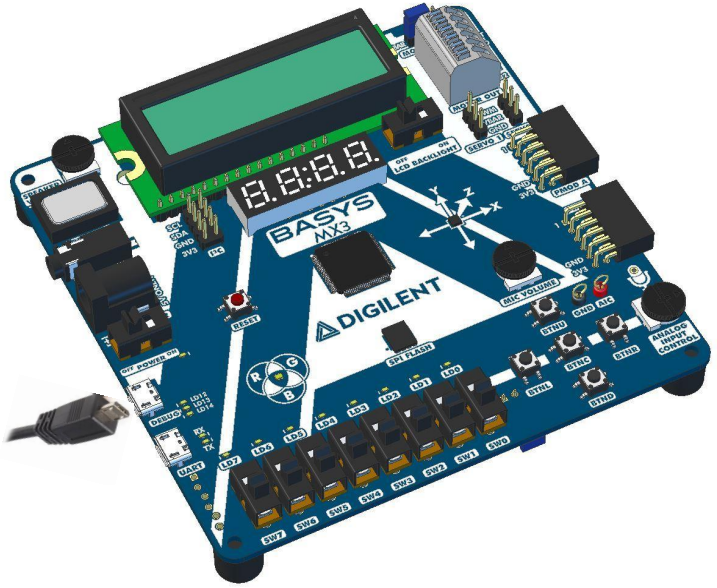
```
LED_SetGroupValue(5);
```

○ ○ ○ ○ ○ ● ○ ●

- Indicator '**DOT**'s can also be directly tied to either **switches**, **buttons**, or given a **constant** value
- Each iteration is delayed by 100ms

Running your code





Running you code consists of 2 stages:

1. 'Clean and Build' Project
2. Run Project

Lab task 1 – Controlling LED’s

- Write a program, that assigns a simple logic element to 6 LED’s witch has 2 inputs (switches).
  The same inputs can be used for all of the logic elements
- List of logic elements to be implemented:

AND, NAND, OR, NOR, XOR, XNOR

- Example: **SW0** and **SW1** are assigned to inputs of an 2 input AND gate. So if both inputs are ON LED **LD0** will light up.

A	B	AND	NAND	OR	NOR	XOR	XNOR
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1

Lab task 2 – Running lights

Use of library **`math.h`** is **forbidden**

- Write a program that lights up only one LED at given time.
  On the next iteration LED next to the previous LED will light up and so on.
  Effect should be that if the lit up LED reaches the edge, it will start coming back and so on.
- Example: It is the same as the front red light of KITT from Knight Rider.

Lab task 3 – Modified example (page 7)

Change the example code so that:

- A BUTTON push will add up switched on **SW**-switches
- First indicator (SEGM0) would show the sum of the previously found value
- Second segment (SEGM1) is **off** at all times
- Segments SEGM3 and SEGM2 must show how many times switches have been added up
  - If the number reaches 99, it must restart counting from 0
- LEDs must show the residue value when dividing the **SW** value by 4
- **ALL** of the **outputs** must only be refreshed when the **BUTTON** is pushed

Lab task 4 – Egg timer

- Write a program that starts to count down from given binary value until it reaches zero. Current values must be displayed on the 7-segment indicators. Timers starting value must be given using the **SW** switches. When timer reaches ZERO the LED's must start blinking on and off. Timers starts only when **BUTTON** is pushed.
- Example: If **SW**s have value of "00001111" and the **BUTTON** is pushed, the indicators will show value of "0015" and will start counting down until it reaches zero. Then LED's will start to blink. LED blinking repetition is not defined.
- The values must decrease once per second
- Values that can be entered by the user must be in the range of 0 to 255 ("00000000" to "11111111").

TALLINNA
TEHNIKAÜLIKOOL