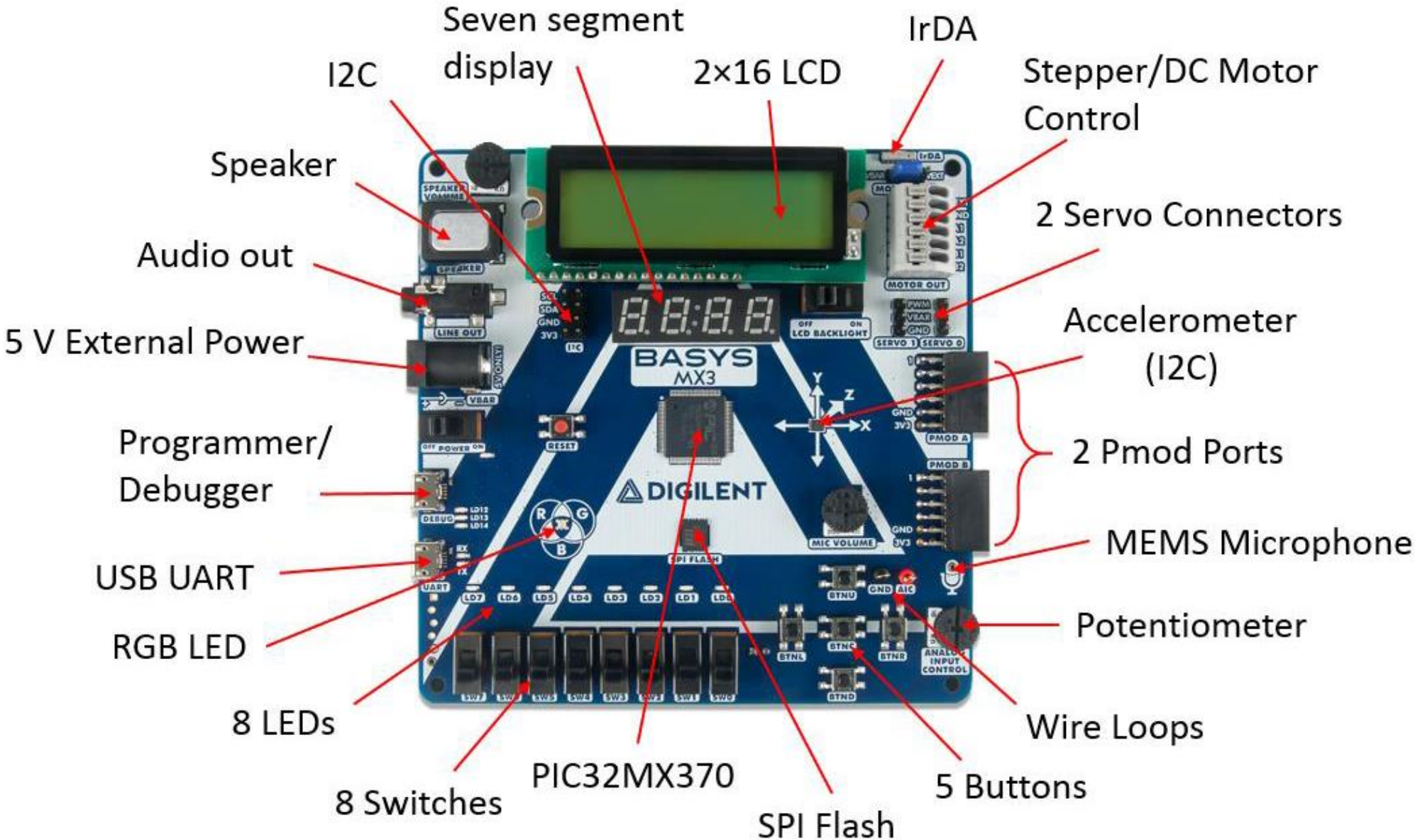
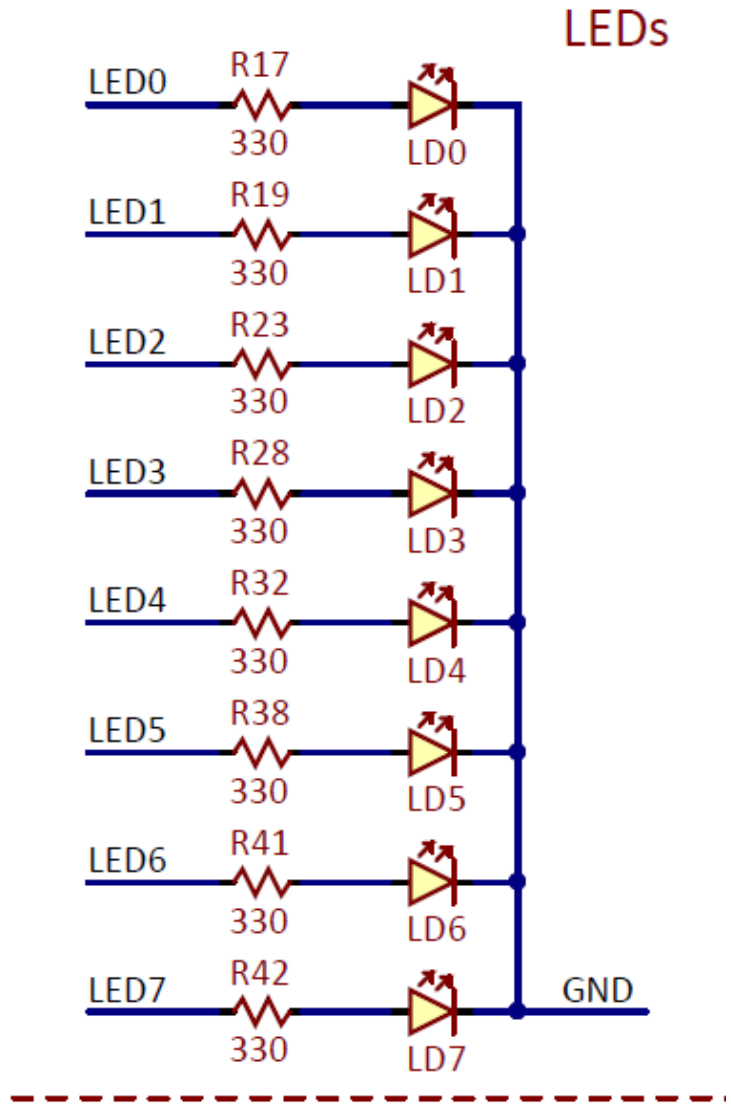

Seven segment display
IrDA
I2C
2×16 LCD
Stepper/DC Motor Control
Speaker
2 Servo Connectors
Audio out
Accelerometer (I2C)
5 V External Power
2 Pmod Ports
Programmer/ Debugger
MEMS Microphone
USB UART
Potentiometer
RGB LED
Wire Loops
8 LEDs
5 Buttons
PIC32MX370
8 Switches
SPI Flash
BASYS MX3
DIGILENT

הרצאה מספר 1

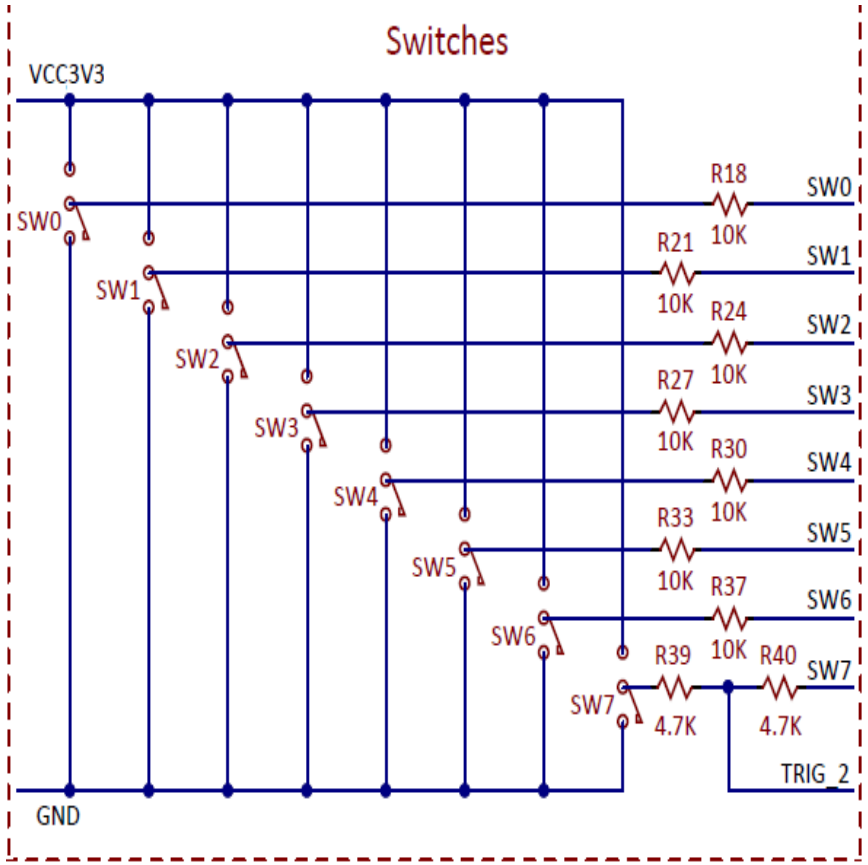
הפעלת הכרטיס למצב in/out

Description	PIC32 pin	Schematic name	Label
Led 0	TMS/CTED1/RA0	LED0	LD0
Led 1	TCK/CTED2/RA1	LED1	LD1
Led 2	SCL2/RA2	LED2	LD2
Led 3	SDA2/RA3	LED3	LD3
Led 4	TDI/CTED9/RA4	LED4	LD4
Led 5	TDO/RA5	LED5	LD5
Led 6	TRCLK/RA6	LED6	LD6
Led 7	TRD3/CTED8/RA7	LED7	LD7



```
#include <xc.h>
#pragma config JTAGEN = OFF
#pragma config FWDTEN = OFF
#pragma config FNOSC =FRCPLL
#pragma config FSOSCEN =     OFF
#pragma config POSCMOD =  EC
#pragma config OSCIOFNC =   ON
#pragma config FPBDIV =     DIV_1
#pragma config FPLLIDIV =     DIV_2
#pragma config FPLLMUL =     MUL_20
#pragma config FPLLODIV =    DIV_1

void main()
{ int j;
   TRISA &= 0xff00;
   PORTA =0x55;



}
```

```
#include <xc.h>
#pragma config JTAGEN = OFF
#pragma config FWDTEN = OFF
#pragma config FNOSC =        FRCPLL
#pragma config FSOSCEN =     OFF
#pragma config POSCMOD =  EC
#pragma config OSCIOFNC =   ON
#pragma config FPBDIV =     DIV_1
#pragma config FPLLIDIV =     DIV_2
#pragma config FPLLMUL =     MUL_20
#pragma config FPLLODIV =    DIV_1

void main()
{ int j;
   TRISA &= 0xff00;
    while(1)
    {
   PORTA ++;
   for(j=0;j<64000;j++);

    }
}
```



Switch 0	RPF3/RF3	SW0	SW0
Switch 1	RPF5/PMA8/RF5	SW1	SW1
Switch 2	RPF4/PMA9/RF4	SW2	SW2
Switch 3	RPD15/RD15	SW3	SW3
Switch 4	RPD14/RD14	SW4	SW4
Switch 5	AN11/PMA12/RB11	SW5	SW5
Switch 6	CVREFOUT/AN10/RPB10/CTED11PMA13/RB10	SW6	SW6
Switch 7	TRIG_2 AN9/RPB9/CTED4/RB9	SW7	SW7

```
#include <xc.h>
#pragma config JTAGEN = OFF
#pragma config FWDTEN = OFF
#pragma config FNOSC =  FRCPLL
#pragma config FSOSCEN =          OFF
#pragma config POSCMOD =          EC
#pragma config OSCIOFNC =         ON
#pragma config FPBDIV =     DIV_1
#pragma config FPLLIDIV =         DIV_2
#pragma config FPLLMUL =          MUL_20
#pragma config FPLLODIV =         DIV_1

void main()
{ int j;
   TRISA &= 0xff00;

TRISFbits.TRISF3 = 1; // RF3 (SW0) configured as input
TRISFbits.TRISF5 = 1; // RF5 (SW1) configured as input
TRISFbits.TRISF4 = 1; // RF4 (SW2) configured as input
TRISDbits.TRISD15 = 1; // RD15 (SW3) configured as input
TRISDbits.TRISD14 = 1; // RD14 (SW4) configured as input
TRISBbits.TRISB11 = 1; // RB11 (SW5) configured as input
ANSELBbits.ANSB11 = 0; // RB11 (SW5) disabled analog
TRISBbits.TRISB10 = 1; // RB10 (SW6) configured as input
ANSELBbits.ANSB10 = 0; // RB10 (SW6) disabled analog
TRISBbits.TRISB9 = 1; // RB9 (SW7) configured as input
ANSELBbits.ANSB9 = 0; // RB9 (SW7) disabled analog
```

```
 while(1)
    {
PORTAbits.RA0=PORTFbits.RF3 ; // RF3 (SW0) configured as input
PORTAbits.RA1=PORTFbits.RF5 ; // RF5 (SW1) configured as input
PORTAbits.RA2=PORTFbits.RF4; // RF4 (SW2) configured as input
PORTAbits.RA3=PORTDbits.RD15 ; // RD15 (SW3) configured as input
PORTAbits.RA4=PORTDbits.RD14; // RD14 (SW4) configured as input
PORTAbits.RA5=PORTBbits.RB11 ; // RB11 (SW5) configured as input
PORTAbits.RA6=PORTBbits.RB10 ; // RB10 (SW6) configured as input
PORTAbits.RA7=PORTBbits.RB9 ; // RB9 (SW7) configured as input

    }
}
```

```
#include <xc.h>
#pragma config JTAGEN = OFF
#pragma config FWDTEN = OFF
#pragma config FNOSC =          FRCPLL
#pragma config FSOSCEN =        OFF
#pragma config POSCMOD =        EC
#pragma config OSCIOFNC =       ON
#pragma config FPBDIV =     DIV_1
#pragma config FPLLIDIV =       DIV_2
#pragma config FPLLMUL =        MUL_20
#pragma config FPLLODIV =       DIV_1

void main()
{ int j;
   TRISA &= 0xff00;
   TRISFbits.TRISF3 = 1; // RF3 (SW0) configured as input
    while(1)
    {
     if(PORTFbits.RF3) // RF3 (SW0) configured as input
        PORTA++;
     else
         PORTA--;
     for(j=0;j<64000;j++);

    }
}
```