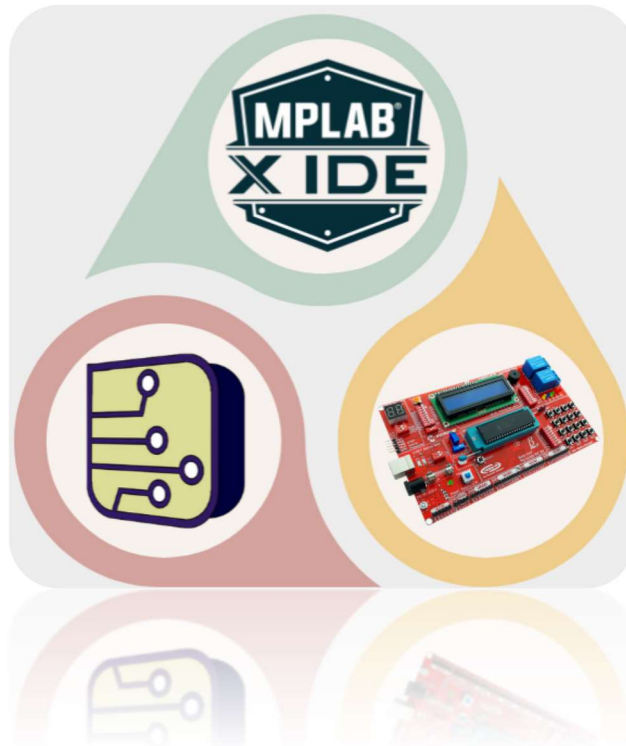
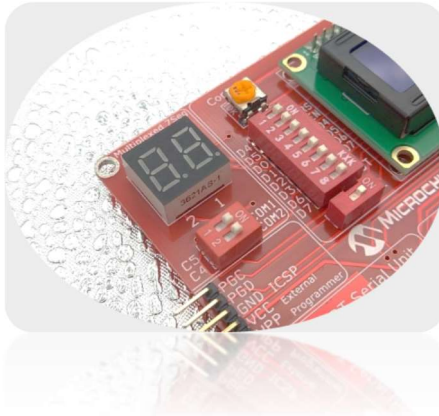


Basic I/O Operations with PIC and Interrupt (Seven Segments Application)

Practical Lab 2



Objective:

In this Lab we are going to learn how to use the GPIO peripherals in the PIC microcontroller and the Interrupts, and integrate this knowledge in a practical example.

Requirements:

➤ **Software:**

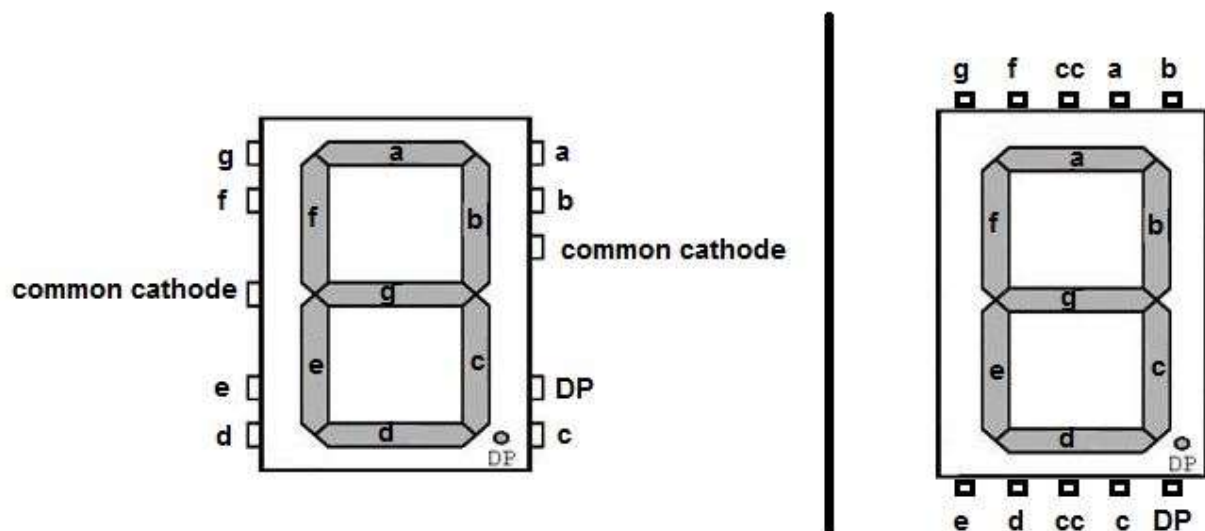
1. MPLAB
2. Simulide
3. PICSimLab

7-Segment

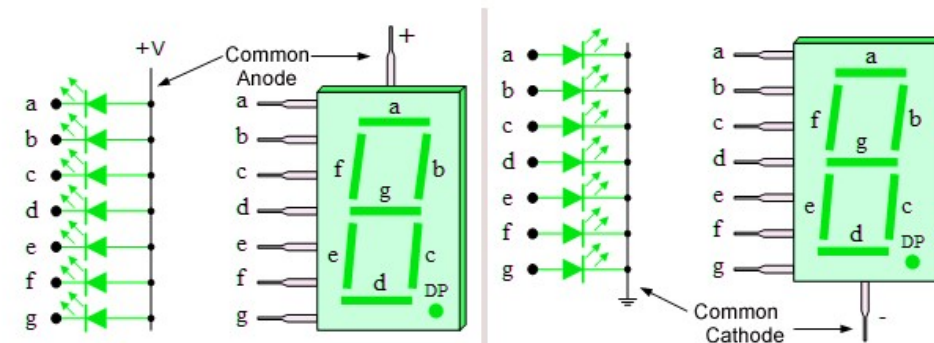
A 7-segment display is commonly used in electronic display devices for decimal numbers from 0 to 9

The use of light-emitting diodes (LEDs) in seven-segment displays made it more popular

7-Segment Display Pinout



7-Segment Display Types



Common Anode and Common Cathode 7 Segment LED Display

1. Common Anode 7 Segment Display:

All the Negative terminals (Anode) of all the 8 LEDs are connected. All the positive terminals are left alone.

2. Common Cathode 7 Segment Display:

All the positive terminals (Cathode) of all the 8 LEDs are connected together. All the negative terminals are left alone.

7-Segment Display Codes

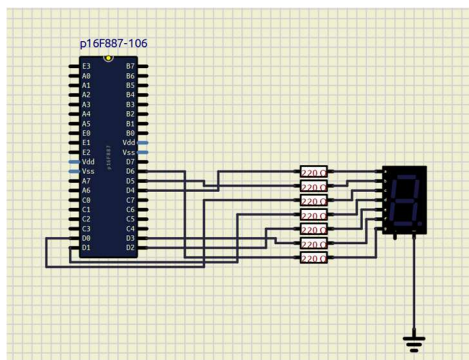
The below table shows the 0-9 codes for the seven-segment LED display.

| Number | g f e d c b a | Hex code |
|--------|---------------|----------|
| 0 | 1000000 | C0 |
| 1 | 1111001 | F9 |
| 2 | 0100100 | A4 |
| 3 | 0110000 | B0 |
| 4 | 0011001 | 99 |
| 5 | 0010010 | 92 |
| 6 | 0000010 | 82 |
| 7 | 1111000 | F8 |
| 8 | 0000000 | 80 |
| 9 | 0010000 | 90 |

Experiment Steps:

1. Prepare the Circuit in the SimulIDE

a. Add the PIC16F887 and the 7 segment.



A: D4

B: D5

C: D0

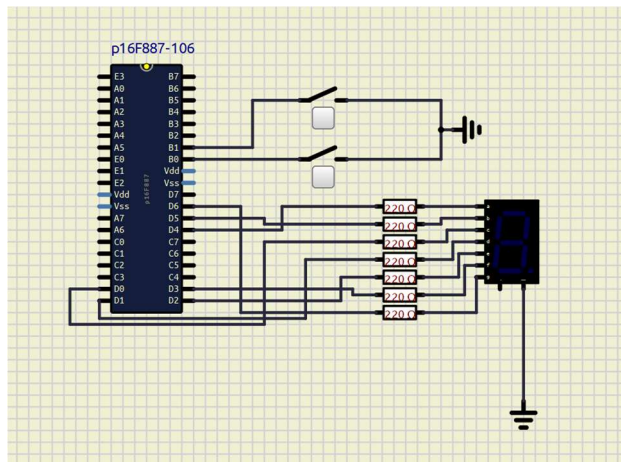
D: D1

E: D2

F: D3

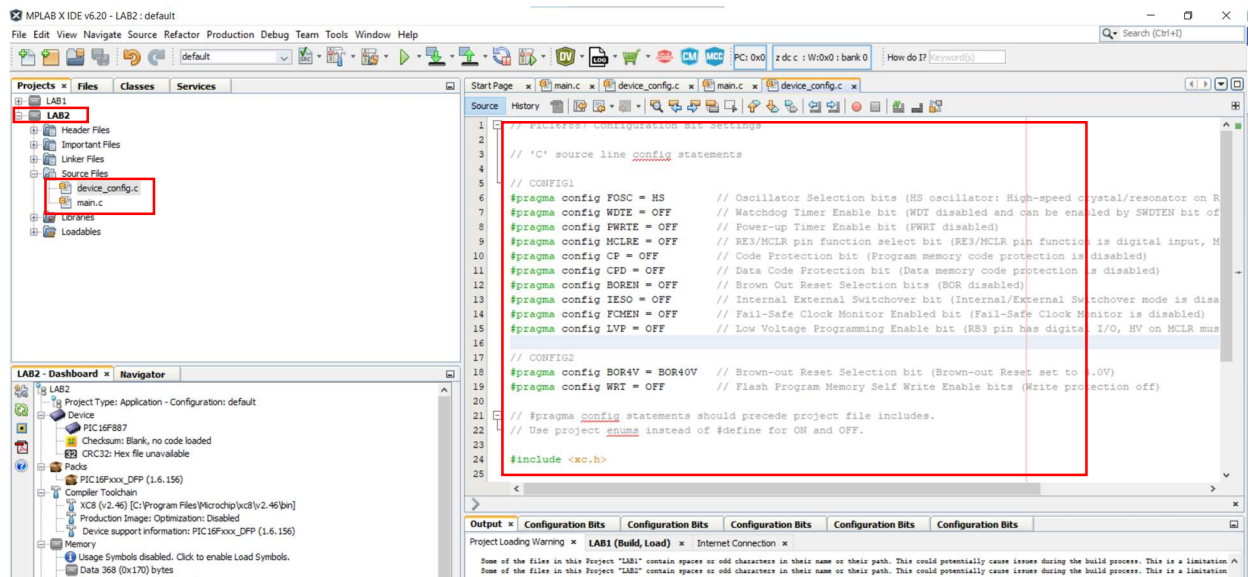
G: D6

b. add two switches on Port B pin0 and pin1



2. Open MPLAB IDE:

a. Open MPLAP, create a project, create main.c and device_config.c and copy the configurations from LAB1 to the device_config.c of LAB2



b. in main.c, we'll start with making all pins of the GPIO digital pins.

```
#include "xc.h"

int main()
{
    //[[1]]Configure all I/O to be digital
    ANSEL = 0;
    ANSELH = 0;

    return 0;
}
```

c. configure port D to be output.

```
#include "xc.h"
int main()
{
    //[1]Configure all I/O to be digital
    ANSEL = 0 ;
    ANSELH = 0;
    //[2]Configure Port D to be output
    TRISD = 0b00000000;
    return 0;
}
```

d. Configure Port B pin0 & pin1 to be input

```
#include "xc.h"
int main()
{
    //[1]Configure all I/O to be digital
    ANSEL = 0 ;
    ANSELH = 0;
    //[2]Configure Port D to be output
    TRISD = 0b00000000;
    //[3]Configure Port B pin0 & pin1 to be input
    TRISB = 0b00000011;

    return 0;
}
```

e. Enable the Global interrupt and allow the interrupt on data change to PORTB Pin0&Pin1.

```
#include "xc.h"
int main()
{
    //[1]Configure all I/O to be digital
    ANSEL = 0 ;
    ANSELH = 0;
    //[2]Configure Port D to be output
    TRISD = 0b00000000;
    //[3]Configure Port B pin0 & pin1 to be input
    TRISB = 0b00000011;
    //[4] Enable Global interrupt
    INTCON |= 1<<7; // set bit 7 in INTCON_REG "GIE"
    //[5] Enable interrupts for portB
    INTCON |= 1<<3; // set bit 3 in INTCON_REG "RBIE"
    //[6] Enable interrupts on datachange for PortB Pin0&Pin1
    IOCBO = 1;      // interrupt on PORTB pin0 change is enabled
    IOCB1 = 1;      // interrupt on PORTB pin1 change is enabled

    return 0;
}
```


f. Write a while loop that counts from 1 to 3 on the 7seg.

```

1  /***** Application *****/
.  *****/
while(1)
{
    PORTD = 0b00100001;    // 7seg display no.1
    for(unsigned int i=0 ; i<65000;i++);    // small delay
    PORTD = 0b01110110;    // 7seg display no.2
    for(unsigned int i=0 ; i<65000;i++);    // small delay
    PORTD = 0b01110011;    // 7seg display no.3
    for(unsigned int i=0 ; i<65000;i++);    // small delay
}

return 0;
}

```

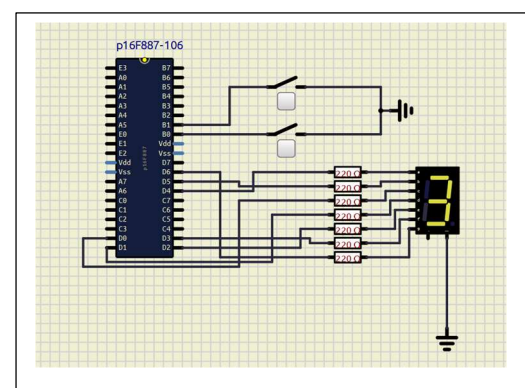
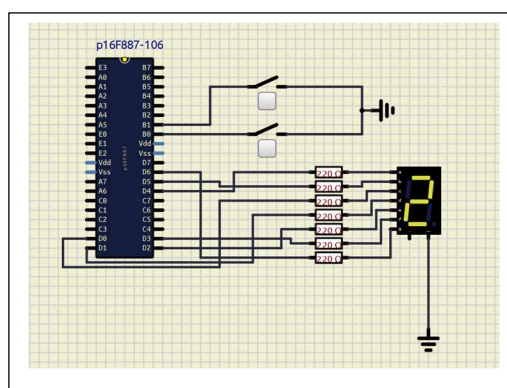
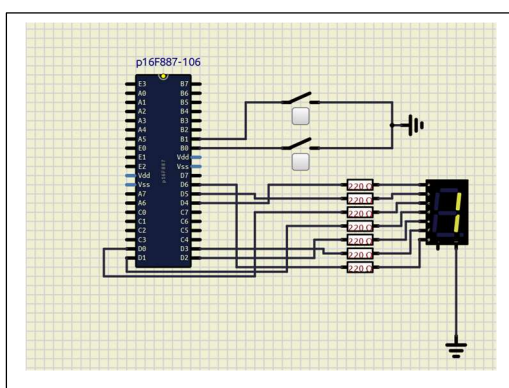
g. Write the ISR function to turn on and off the 7 Segments using the external switches.

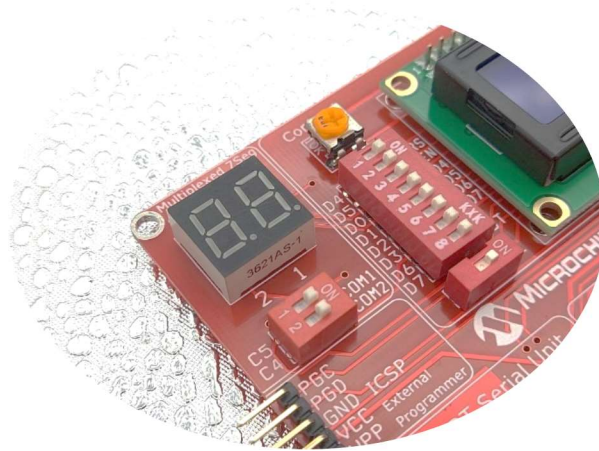
```

- void __interrupt() ISR(void) {
    if (0 == ((PORTB >> 0) & 1)) {
        TRISD = 0xFF;
    }
    if (0 == ((PORTB >> 1) & 1)) {
        TRISD = 0x00;
    }
    INTCON &= ~ 1 << 0;
}

```

3. Test the output on the simulIDE:

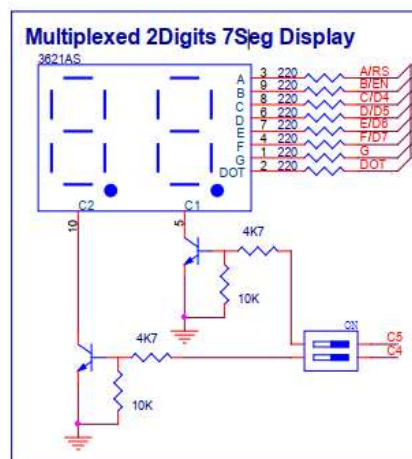




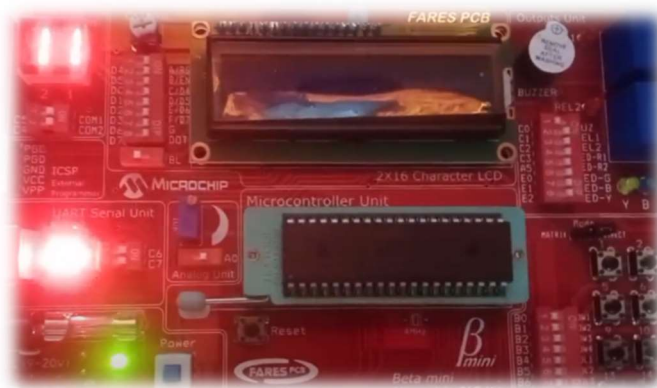
Multiplexed Two Digits 7seg Display Unit

Two multiplexed 7seg digits are added to *Betamini* kit.

| Segment | MCU pins |
|---------|----------|
| A | D4 |
| B | D5 |
| C | D0 |
| D | D1 |
| E | D2 |
| F | D3 |
| G | D6 |
| DOT | D7 |
| Common | MCU pins |
| Com 1 | C5 |
| Com 2 | C4 |

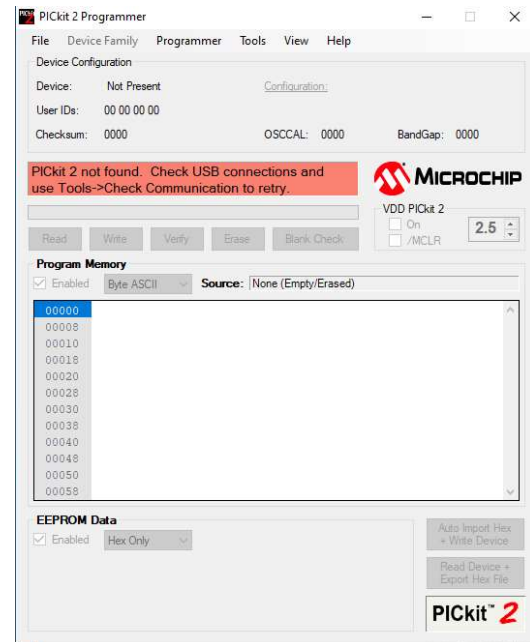


Tring code in Kit



Burning Code in Kit

You Have a PiC Kit 2 programmer



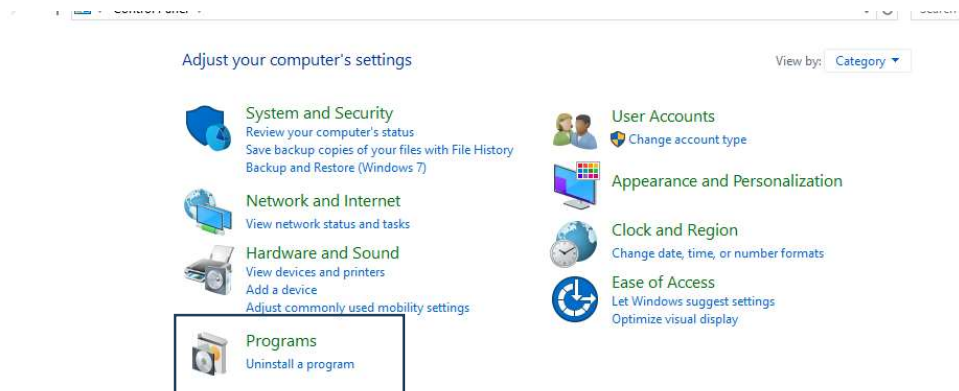
To download

https://fares-pcb.com/product/pic-development-kit-betamini/?attachment_id=23197&download_file=inv661k497vmy

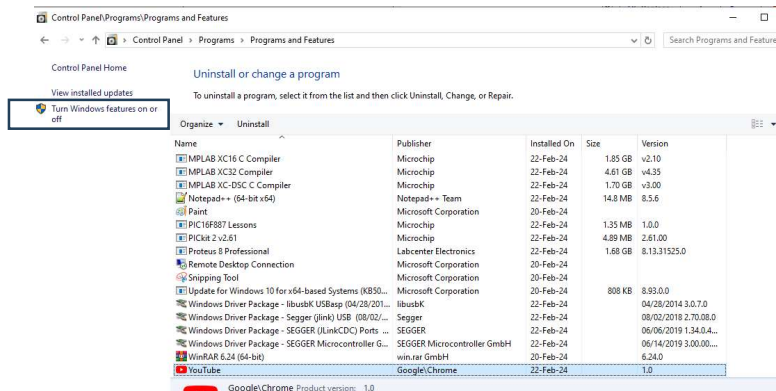
Some time It Require .net frame work v2.0

“ If You have a problem with install You can try this “

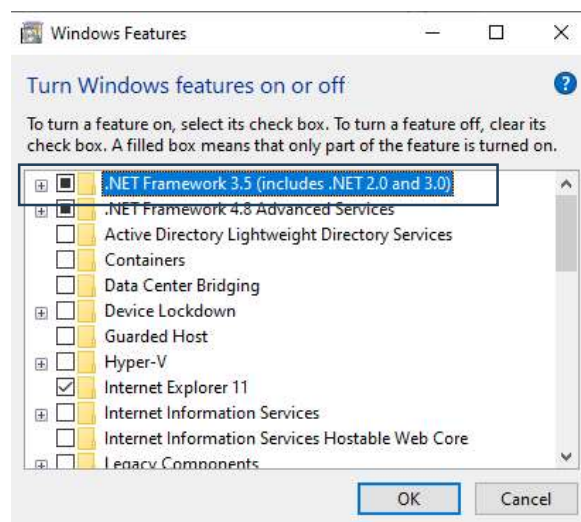
Open **control panel** >



Select **Uninstall programs**



Then check on



Wait the windows to download the framework then install the programmer software

Lab report:

Submit a PDF file with Code, snapshots and “Small Video for the practical work” of the work you did and upload the project file.