LAB NO. 3

MULTIPLEXED 7-SEGMENT DISPLAYS WITH PIC

OBJECTIVES: -

- To understand what is multiplexing and how it works.
- To understand how to use four 7-segment displays with a single port in PIC.
- To simulate the circuit on proteus.

APPARATUS: -

- Laptop
- MPLAB X
- Proteus Circuit Simulator

CLASS TASK: -

- Write a code PIC to display “0123” on four seven segment displays using a single port.

THEORY: -

MULTIPLEXING: -

In literature it is technique by which different types of signals can be combined to one signal without interference. In Digital Electronics it means that different data can be passed to the single line without distortion, and we are discussing the same.

In this case multiplexing is simply that the LEDs are turned on and off at such a fast rate that one will assume that it is always on or off. For example to display “0123” on the LEDs we have to send “0” to the port and turn the first LED on while other being off, then after a very short delay (here very short means some micro seconds) then “1” will be sent to the port and all the displays being off except display number 2 and so on for “2” and “3”.

Thus, displays will turn off and on at a very fast rate and human eye can’t detect the delay and it is assumed that the LEDs are always on.

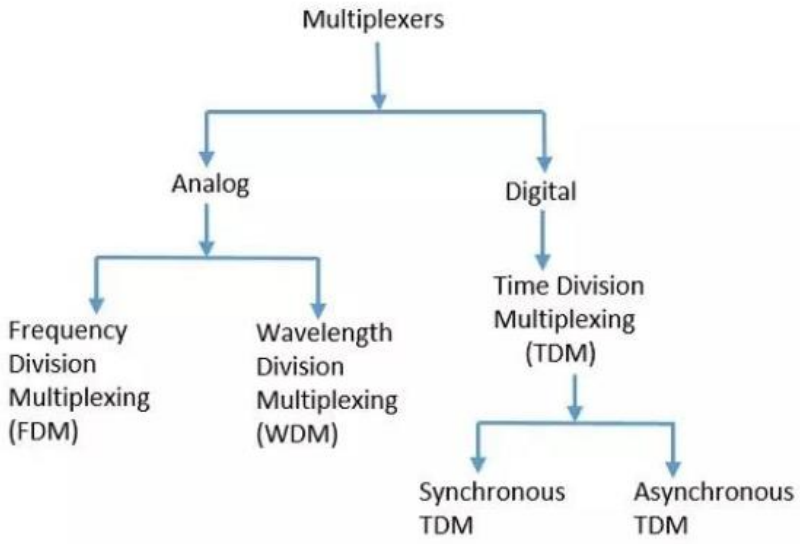

Figure 3.1: Types of Multiplexing

BJTs (Bipolar Junction Transistors): -

The transistor is a three-layer semiconductor device consisting of either two n-type and one p-type layers of material or two p-type and one n-type layers of material. The first one is called NPN transistor, and the second is known as PNP transistor.

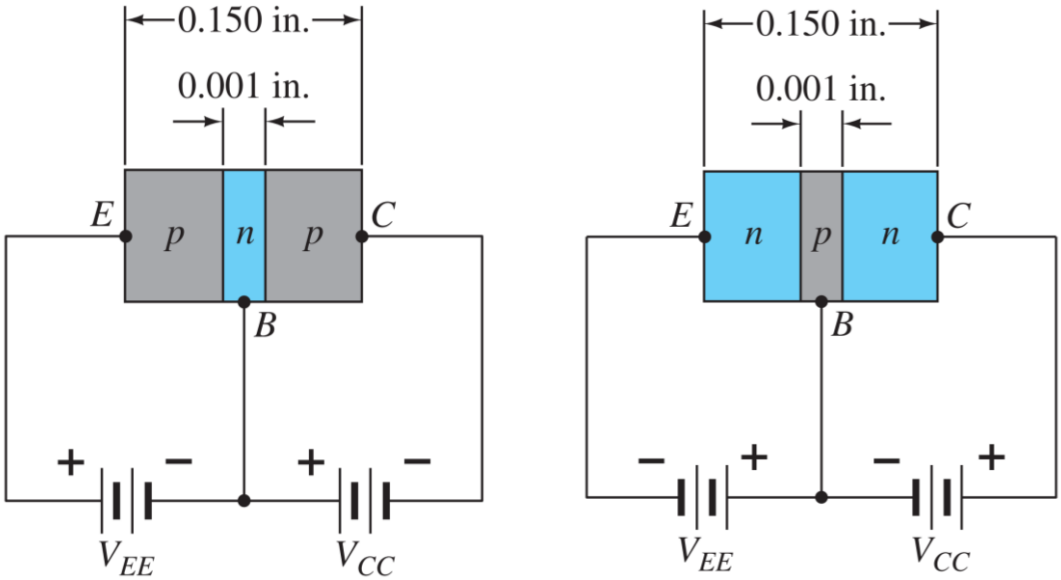

Figure 3.2: BJT (both PNP & NPN) Common Base DC Biasing

It is a three pined device which is similar to two back to back connected P-N junction diodes, the tree pins are named as follows: -

- Emitter (E)
- Base (B)
- Collector (C)

These are current controlled devices and turned on if the base current is applied can be used as switch for this purpose to turn on or off the LEDs. There are some common formulas can be sued to determine the biasing resistors for this purpose.

$$I_E = I_C + I_B$$

But I_C is so small that can be neglected and thus,

$$I_E \cong I_C$$

$$I_C \propto I_B$$

$$I_C = \beta\, I_B$$

Where β is called h_{FE} or current gain which is a constant value for every transistor.

PROGRAM: -

CODE: -

```
#include "P18F452.inc"
list P=18F452, F=INHX32, MM=OFF

    CONFIG OSC=XT
```

```
    CONFIG WDT=OFF
    #define LEDs PORTB

    CBLOCK 0X00
    BCD1, BCD2, BCD3, BCD4, BCD_COUNT, TEMP_0, TEMP_1
    ENDC

    ORG 0X00
    GOTO MAIN
    ORG 0X200

MAIN:
    MOVLB 0X01
    CLRF TRISB
    CLRF TRISD
    CLRF BCD_COUNT, 1
    MOVLW 0X3F
    MOVWF BCD1, 1
    MOVLW 0X06
    MOVWF BCD2, 1
    MOVLW 0X5B
    MOVWF BCD3, 1
    MOVLW 0X4F
    MOVWF BCD4, 1
    CLRF TEMP_0, 1
    CLRF LEDs

ENDLESS_LOOP
    CALL DISPLAY
    CALL DELAY
    BRA ENDLESS_LOOP

DISPLAY
    MOVF BCD_COUNT, W, 1
    ANDLW 0X03
    MULLW 0X02
    MOVF PRODL, WREG
    ADDWF PCL, F
    BRA DIGIT_1
    BRA DIGIT_2
    BRA DIGIT_3
    BRA DIGIT_4
    RETURN

DIGIT_1
    CLRF PORTB
    CLRF PORTD
    MOVF BCD1, W, 1
```

```
    MOVWF PORTB
    BSF PORTD, 0
    INCF BCD_COUNT, F, 1
    RETURN

DIGIT_2
    CLRF PORTB
    CLRF PORTD
    MOVF BCD2, W, 1
    MOVWF PORTB
    BSF PORTD, 1
    INCF BCD_COUNT, F, 1
    RETURN

DIGIT_3
    CLRF PORTB
    CLRF PORTD
    MOVF BCD3, W, 1
    MOVWF PORTB
    BSF PORTD, 2
    INCF BCD_COUNT, F, 1
    RETURN

DIGIT_4
    CLRF PORTB
    CLRF PORTD
    MOVF BCD4, W, 1
    MOVWF PORTB
    BSF PORTD, 3
    INCF BCD_COUNT, F, 1
    RETURN

DELAY
        MOVLW D'255'
        MOVWF TEMP_1

    L1  DECFSZ TEMP_1, F
        BRA L1
    RETURN
END
```

EXPLANATION: -

In the code the separate GPRs are defined and assigned values of hex codes of 7-segment of 0, 1, 2 and 3 respectively named BCD_1, BCD_2, BCD_3 and BCD_4. Then using concept of program counter a variable is incremented and reset to zero if it becomes 0x03. This variable is used to call the sub-routines for digit number 1, 2, 3 or 4 respectively.

When a DIGIT sub routine is called the hex code of the respective BCD is sent to the PORTB and PORTD's respective bit is turned HIGH to turn that specific LED on and all other being off. Thus, a multiplexed display is created with a short delay.

In the DIGIT sub-routine, the incrementation variable is incremented and program returns to the Endless_Loop and again other digit is called and other LED on held on while other being off.

CIRCUIT CALCULATIONS: -

In this experiment common collector configuration of the transistor is bein used in this configuration emmitter is grounded and base current is controlled by the base resistor and base voltages.

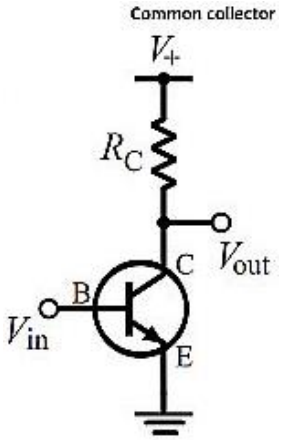

Figure 3.3: Common Collector Configuration of NPN BJT

Now the typical LEDs require about 15mA (maximum) at 2V to work so there are 7-segments in a display thus the current becomes 15 x 7 = 105mA for a display. Thus, the collector current of the transistor should be equal to 105mA or 110mA for safety purpose.

$$I_C = 110mA \quad ; \quad V_+ = 5V$$

The transistor used in this experiment is 2N2222 have an h_{FE} of it is 100. Thus,

Emitter Cut-off Current	I_{EBO}	-	-	100	nA	V_{EB} = 3V, I_C = 0
DC Current Gain	$h_{FE(1)}$	100	-	300		V_{CE} = 10V, I_C = 150mA
	[illegible]	[illegible]	[illegible]	[illegible]		[illegible]
	$h_{FE(3)}$*	42	-	-		V_{CE} = 10V, I_C = 500mA
Collector-Emitter Saturation Voltage	$V_{CE(sat)(1)}$*	-	-	0.6	V	I_C = 500mA, I_B = 50mA
	$V_{CE(sat)(2)}$*	-	-	0.3	V	I_C = 150mA, I_B = 15mA
Base-Emitter Saturation Voltage	$V_{BE(sat)}$*	-	-	1.2	V	I_C = 500mA, I_B = 50mA

Figure 3.2: BJT (both PNP & NPN) Common Base DC Biasing

$$I_B = 110m / 100 = 1.1mA$$

And for every silicon transistor VBE = 0.7V. Thus,

$$R_B = 0.7 / 1.1m = 0.636\ K\Omega$$

Now V_+ = 5V and we want a V_C of 2V thus,

$$R_C = (5 - 2) / 110m = 0.0272\ K\Omega$$

CIRCUIT: -

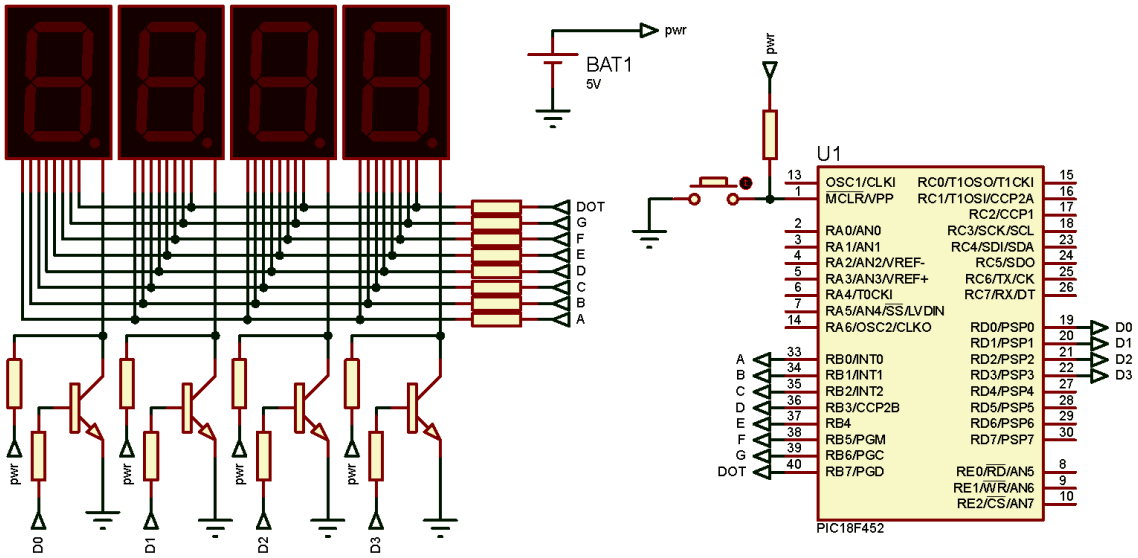

Figure 3.4: Multiplexed 7-Segment Displays in Proteus

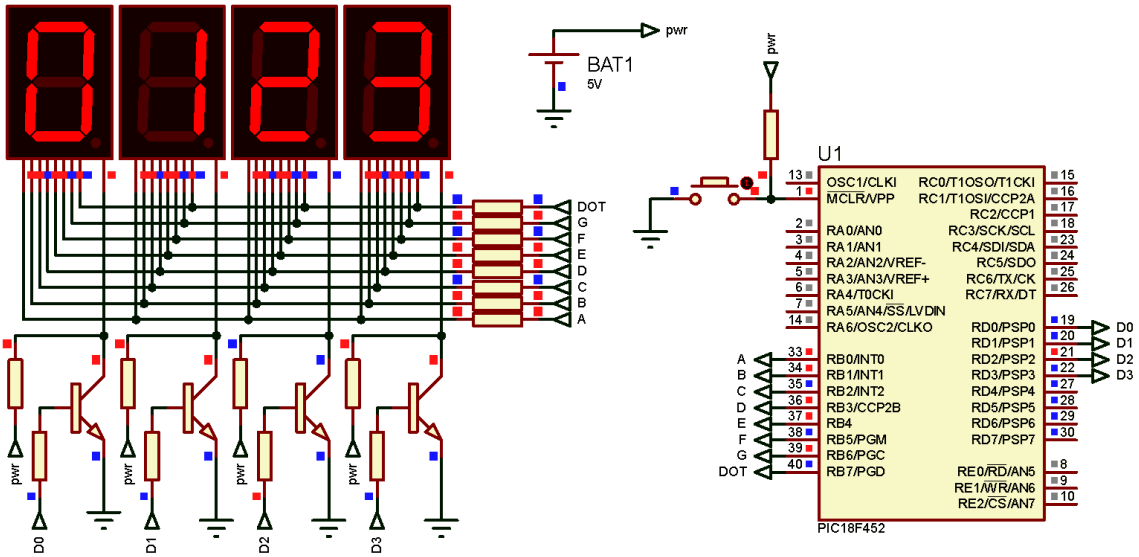

Figure 3.5: Working 7-segment Display Displaying 0123

HOME TASK: -

- A program in C for PIC microcontroller to display count from 0 to 255 on multiplexed 7-segment LEDs.

PROGRAM: -

CODE: -

```
#include <pic18f452.h>
#include <xc.h>

void num2dig(int);
int sevenSeg(int);
```

```
void display();
void delay();

int dig[3];
unsigned int code[3];

int main()
{
    unsigned char mainCount = 0;
    TRISB = TRISD = 0;
    PORTB = PORTD = 0;
    while (1)
    {
        if (mainCount > 255)
            mainCount = 0;
        num2dig(mainCount);
        mainCount++;
        for (int i = 0; i < 3; i++)
            code[i] = sevenSeg(dig[i]);
        display();
    }
    return 0;
}

void num2dig(int num)
{
    int k = 0;
    while (num > 0)
    {
        dig[k] = num % 10;
        num /= 10;
        k++;
    }
}

int sevenSeg(int num)
{
    switch (num)
    {
    case 0:
        return 0x3F;
        break;
    case 1:
        return 0x06;
        break;
    case 2:
        return 0x5B;
        break;
```

```
    case 3:
        return 0x4F;
        break;
    case 4:
        return 0x66;
        break;
    case 5:
        return 0x6D;
        break;
    case 6:
        return 0x7D;
        break;
    case 7:
        return 0x07;
        break;
    case 8:
        return 0x7F;
        break;
    case 9:
        return 0x6F;
        break;
    }
    return 0;
}

void display()
{
    int count = 0;
    while (count <= 100)
    {
        PORTB = code[0];
        PORTDbits.RD0 = 1;
        delay();
        PORTB = code[1];
        PORTDbits.RD1 = 1;
        delay();
        PORTB = code[2];
        PORTDbits.RD2 = 1;
        delay();
        count++;
    }
}

void delay()
{
    for (int j = 0; j <= 12; j++)
    {
    }
```

```
    PORTB = PORTD = 0;
}
```

EXPLANATION: -

In this program the concept of multiplexing is used as did in previous program, but in that program a fixed number that is 0123 was displayed in the display but in this program related to a counter.

Here a variable is used to increment the count and that count is displayed in the multiplexed seven segment display. Now here is the deal, the count is a single number that is let say 199 we have displayed “1” on a separate LED “9” on separate and other “9” on another display. For that purpose, these number have to be separate into three digits.

Now let’s discuss the program, first of all a carriable is initialized with zero and that is the variable which is going to display on the LEDs, to convert that number into separate three digits num2digit function is used which will separate the numbers into three digits.

Then these digits are passed to sevenSeg function which will convert the digits in to hexa-decimal codes for that specific digit. All the three digits are passed and converted to the hex codes. Then these digits are separately passed to the function display which will create a multiplexing effect and will display the number. The delay function will create a very short delay between turning LEDs on and off.

CIRCUIT: -

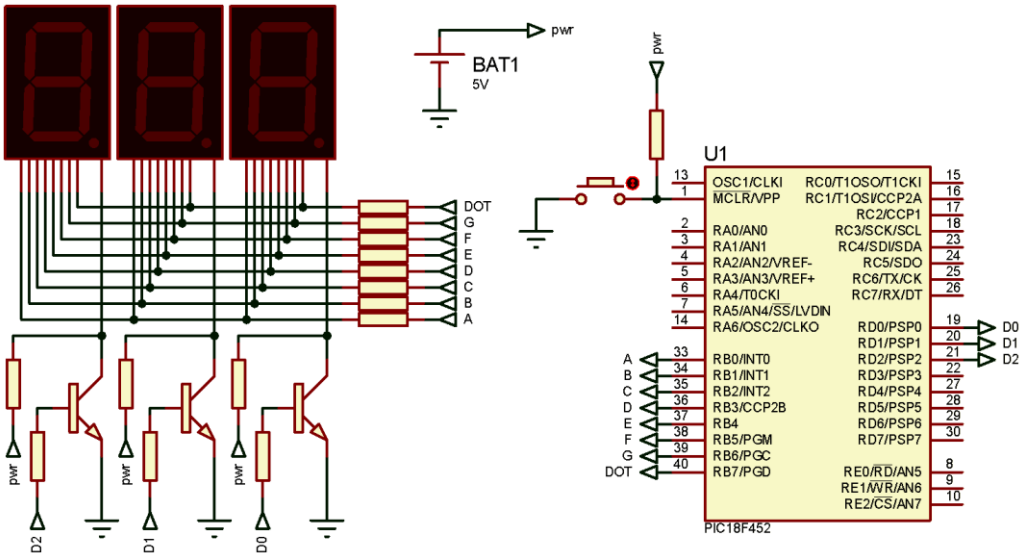

Figure 3.6: Three Multiplexed 7-Segment Displays

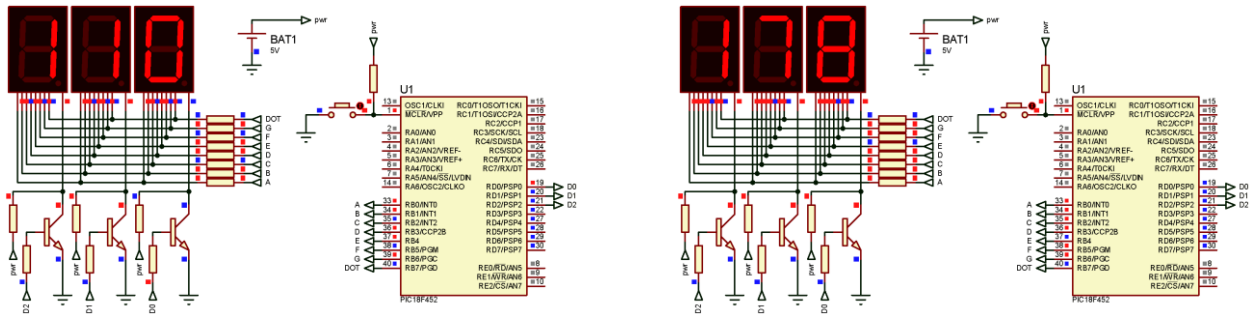

Figure 3.7: PIC Counter Showing Random Numbers

CONCLUSION: -

- In this lab I have learnt what is multiplexing and how to work with multiplexed displayed on a single port in PIC assembly and C.
- The idea of multiplexing can also be implemented on other electronics devices which is same as that of this experiment.
- In this lab I have learnt how to work with transistors and how to calculated the values of the biasing resistors I DC biasing of the BJT.
- In this lab some algorithmic techniques are also studies which includes the separating of the multi-digited into single digits.
- C programming was also introduced in the lab.