Ahmedabad University

ECE302 Embedded System Design

Lab Report 5

Section 1

Submitted to faculty: Anurag Lakhlani

Date of Submission: November 26, 2021

Student Details

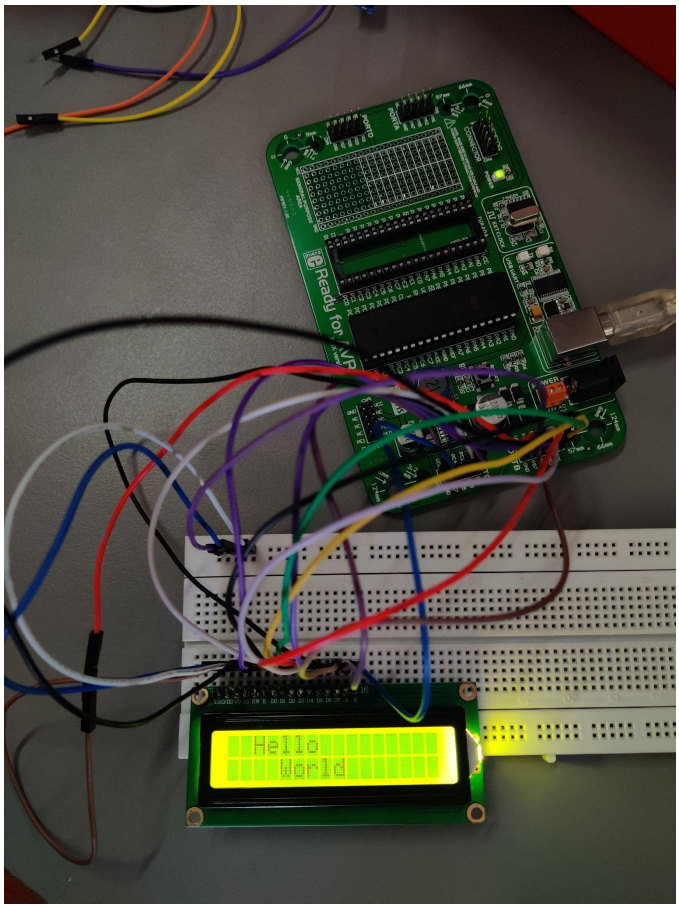
Roll No.	Name of the Student	Name of the Program
AU1940049	Sameep Vani	BTech CSE

2021-2022 (Monsoon Semester)

All circuit diagrams can be accessed here.

Experiment 1

Experiment Result



Code

```
#define F_CPU 8000000UL /* Define CPU Frequency e.g. here 8MHz */

#define LCD_Data_Dir DDRB       /* Define LCD data port direction */
#define LCD_Command_Dir DDRC    /* Define LCD command port direction
register */
#define LCD_Data_Port PORTB     /* Define LCD data port */
#define LCD_Command_Port PORTC /* Define LCD data port */
#define RS PORTC                /* Define Register Select
(data/command reg.)pin */
#define RW PORTC.b1             /* Define Read/Write signal pin */
#define EN PORTC.b2             /* Define Enable signal pin */

void LCD_Command(unsigned char cmnd)
{
    PORTB = cmnd;
```

```
    PORTC &= ~(0x01); /* RS=0 command reg. */
    PORTC &= ~(0x02); /* RW=0 Write operation */
    PORTC |= (0x04);  /* Enable pulse */
    delay_ms(100);
    PORTC &= ~(0x04);
    delay_ms(100);
}

void LCD_Char(unsigned char char_data) /* LCD data write function */
{
    LCD_Data_Port = char_data;
    LCD_Command_Port |= (0x01);  /* RS=1 Data reg. */
    LCD_Command_Port &= ~(0x02); /* RW=0 write operation */
    LCD_Command_Port |= (0x04);  /* Enable Pulse */
    delay_ms(100);
    LCD_Command_Port &= ~(0x04);
    delay_ms(100);
}

void LCD_Init1() /* LCD Initialize function */
{
    DDRC = 0xFF; /* Make LCD command port direction as o/p */
    DDRB = 0xFF; /* Make LCD data port direction as o/p */
    delay_ms(100);

    LCD_Command(0x38); /* Initialization of 16X2 LCD in 8bit mode */
    LCD_Command(0x0C); /* Display ON Cursor OFF */
    LCD_Command(0x06); /* Auto Increment cursor */
    LCD_Command(0x01); /* Clear display */
    LCD_Command(0x80); /* Cursor at home position */
}

void LCD_String(char *str) /* Send string to LCD function */
{
    int i;
    for (i = 0; str[i] != 0; i++) /* Send each char of string till
the NULL */
    {
        LCD_Char(str[i]);
    }
}

void LCD_String_xy(char row, char pos, char str) /* Send string to
LCD with xy position */
{
    if (row == 0 && pos < 16)
        LCD_Command((pos & 0x0F) | 0x80); /* Command of first row and
required position<16 */
    else if (row == 1 && pos < 16)
        LCD_Command((pos & 0x0F) | 0xC0); /* Command of first row and
required position<16 */
    LCD_String(str);                    /* Call LCD string function
*/
```

```
}

void LCD_Clear()
{
    LCD_Command(0x01); /* clear display */
    LCD_Command(0x80); /* cursor at home position */
}

int main()
{

    LCD_Init1();          /* Initialize LCD */
    LCD_Command(0x82);    /* Go to 2nd line*/
    LCD_String("Hello"); /* write string on 1st line of LCD*/
    LCD_Command(0xC4);    /* Go to 2nd line*/
    LCD_String("World"); /* Write string on 2nd line*/

    return 0;
}
```

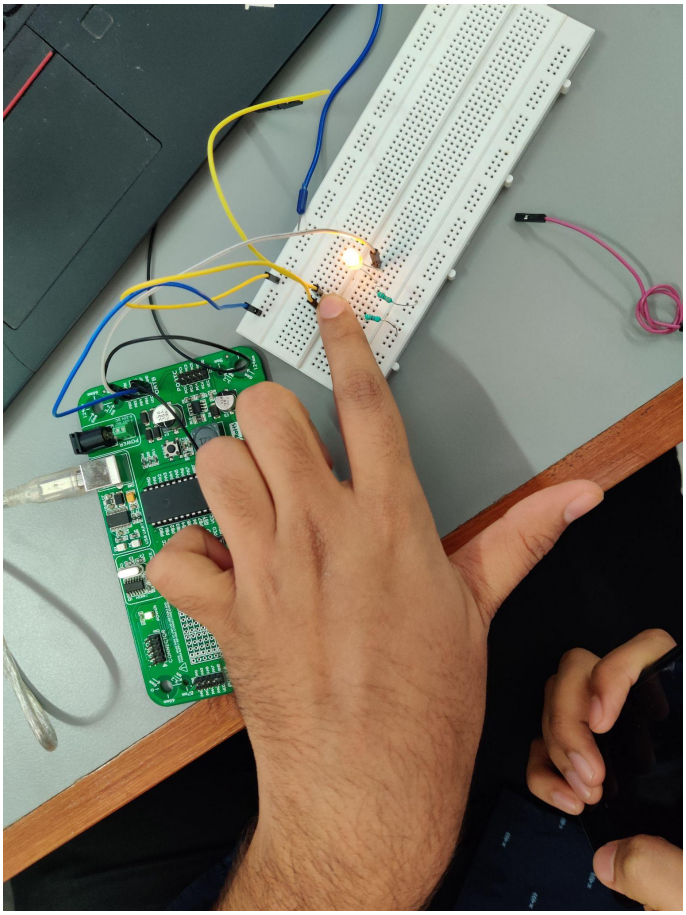
[Video]

Description

In this experiment, we had to print “Hello World” on an LCD. The code being long is difficult to understand. However, it is the same that was taught in class with the difference being only in the string to be printed and some function names. The circuit connections were made according to the circuit diagram shown in class in interfacing LCD with AVR class.

Experiment 2

Experiment Result



Code

```
void main()
{
     DDRB = 0x01;
     while (1)
     {
          if (PINB.b1 == 1)
          {
               PORTB = 0x01;
          }
          else
          {
               PORTB = 0x00;
          }
     }
}
```

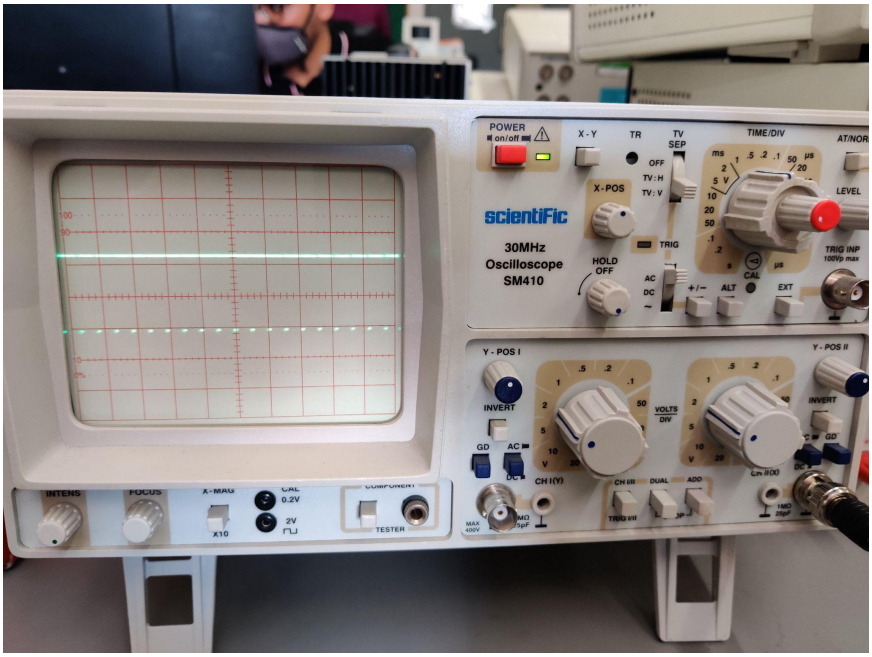
Video

Description

In this experiment, we had to actually use a DC motor to detect its speed. But due to the absence of a motor, we were asked to use an led. When the switch is pressed, the led will get turned on and on releasing the switch, the led is turned off. The connections are quite simple, PORTB.b1 is connected to that side of the switch where the ground is also connected. This takes the input. The led is connected to PORTB.b0 and through a resistor to the ground. This enables current to flow and led to turn on.

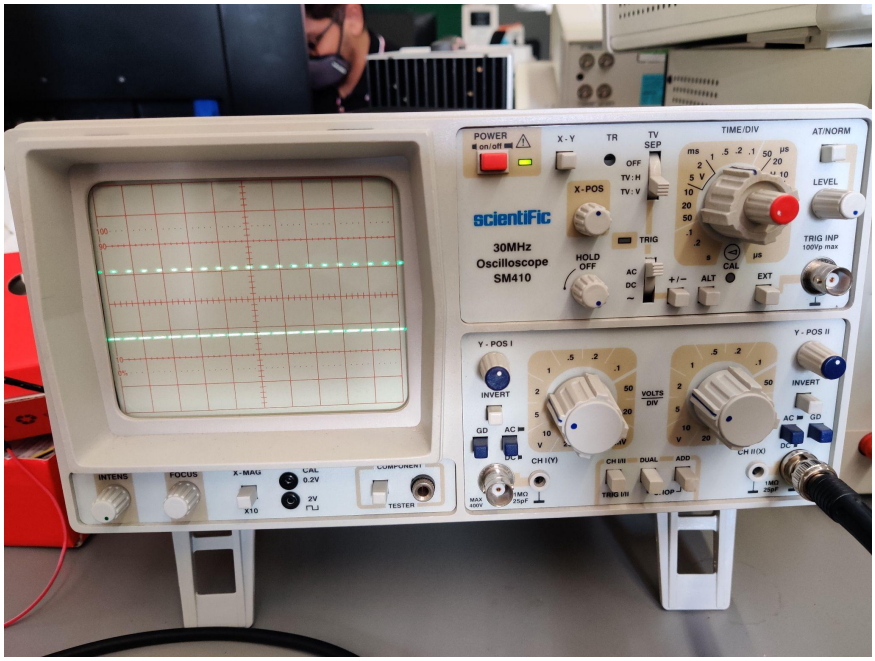
Experiment 3a

Task 1 Result



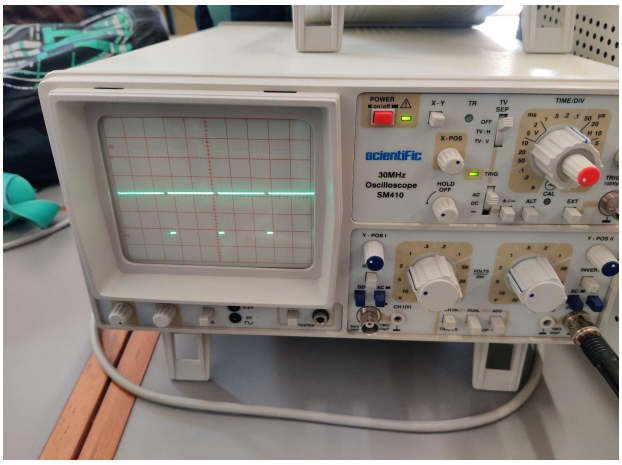
[Video]

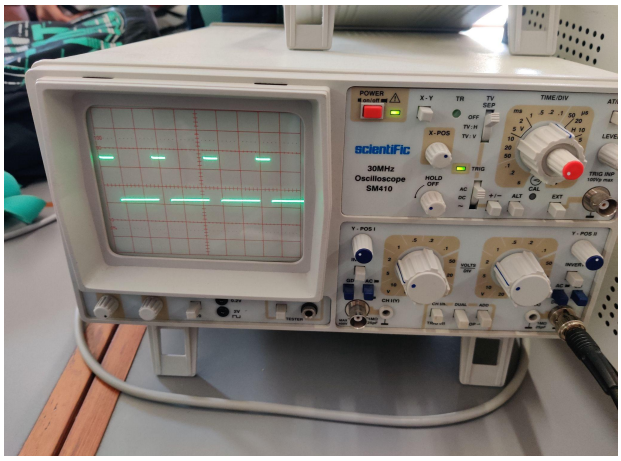
Task 2 Result



Video

Task 3 Result





Video

Code

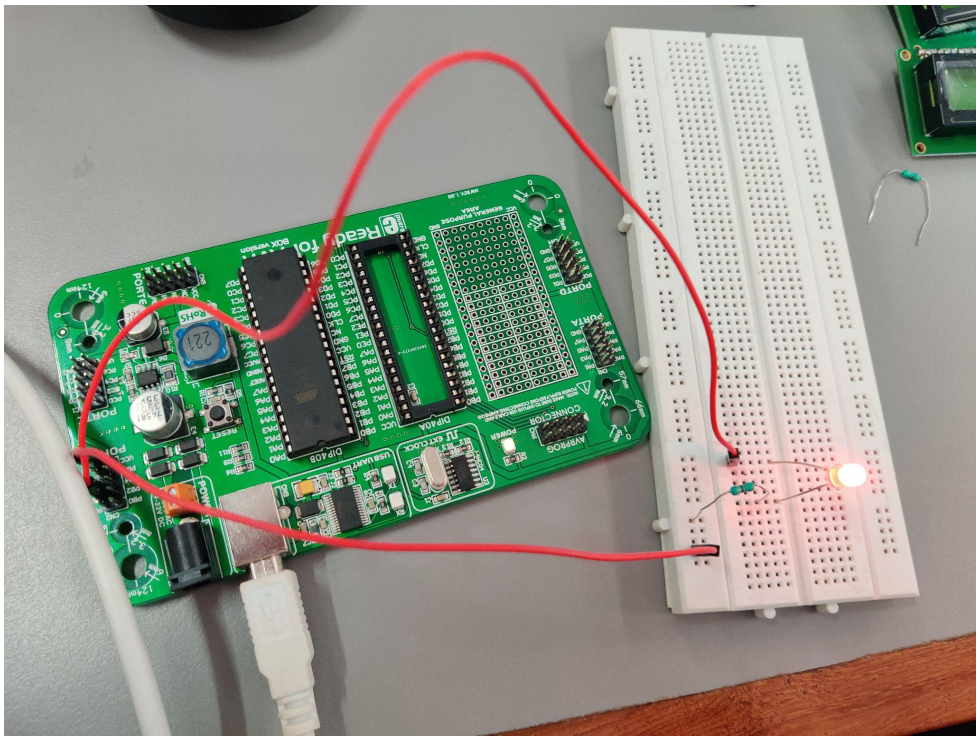
```
// Task 1
void main()
{
    DDRB = 0x08;
    OCR0 = 230;
    TCCR0 = 0x6A;
    while (1);
}

// Task 2
void main()
{
    DDRB = 0x08;
    OCR0 = 63;
    TCCR0 = 0x6A;
    while (1);
}

// Task 3
void main()
{
    int flag = 0;
    DDRB = 0x08;

    while (1)
    {
        TCCR0 = 0x6A;
        if (flag == 0)
        {
            OCR0 = 230;
            delay_ms(5000);
            flag = 1;
        }
        else
        {
            OCR0 = 63;
            delay_ms(5000);
            flag = 0;
        }
    }
}
```

Description

In this experiment, we had three tasks to perform. The first one is using a DC motor and providing it with a square wave of 90% duty cycle using timer0 and fast PWM. But due to the non-availability of motors, we used CRO to confirm it. Upon closer inspection of the result of task 1, it can be seen that 90% of the time the wave is “1” and 10% of the time the wave is “0”. Finally, the connections are simple with PB3 connected to CRO. Task 2 is similar to 1 with a duty cycle of 25% and task 3 is a combination of both. Note that for tasks 1 and 2, the CRO had some issues because of which the wave is moving and not staying stable but for task 3, we switched to another CRO and it is stable.

Experiment 3b

Experiment Result



Code

Same as that of Experiment 3a.

[Video Task 1]

[Video Task 2]

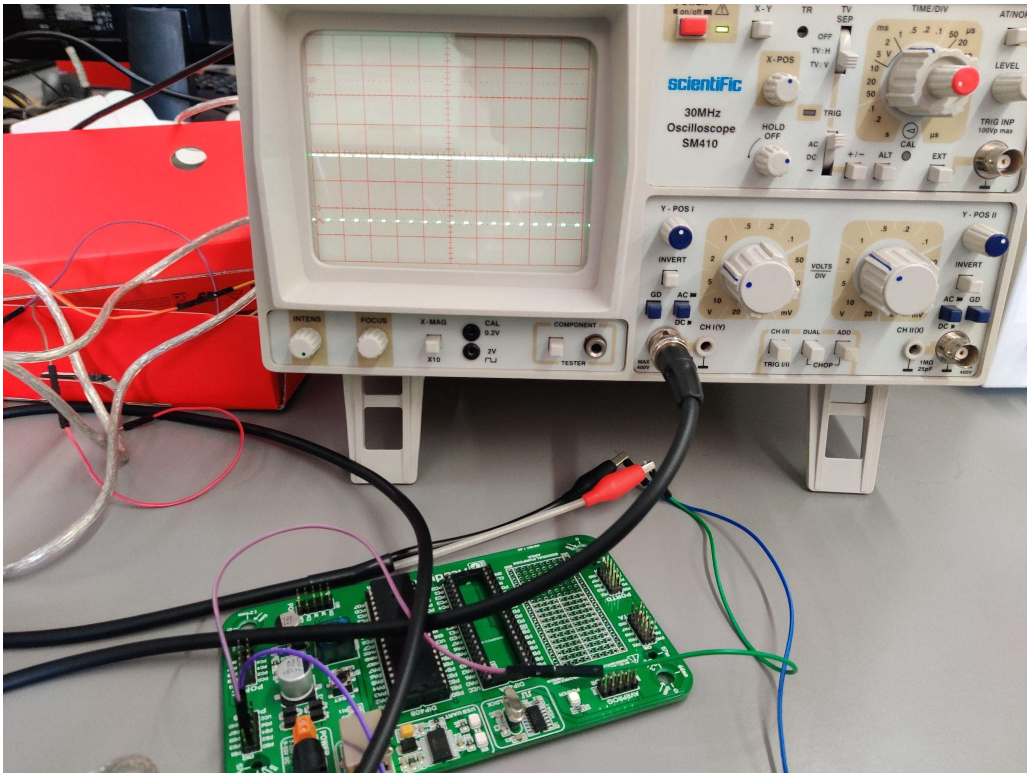
[Video Task 3]

Description

This experiment is similar to Experiment 3a. The difference is that instead of CRO, we need to use an LED. The difference here is in the intensity of the LED. It is evident in Task 3 where we can see the change in brightness of the LED every 5 seconds. The connections are again simple. Connect PB3 to LED and through a resistor which is connected to another terminal of LED, it is grounded to establish a path for the current to flow.

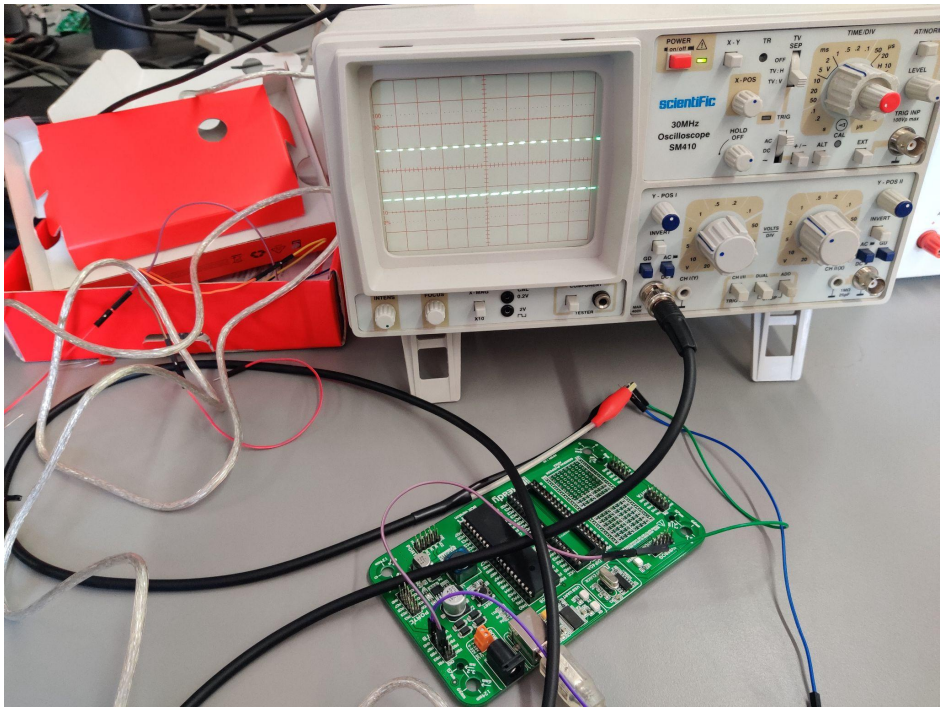
Experiment 4a

Task 1 Result



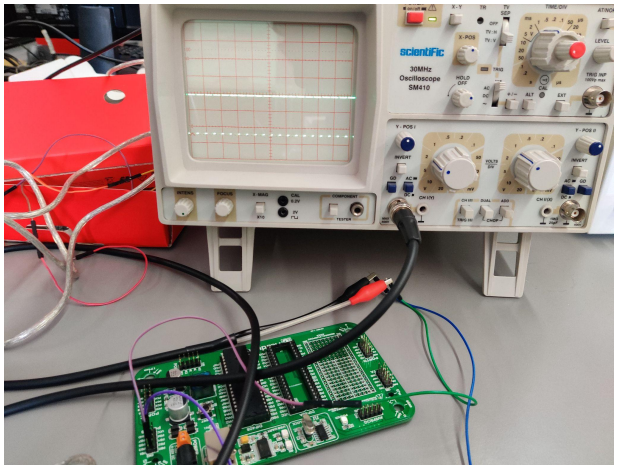
Video

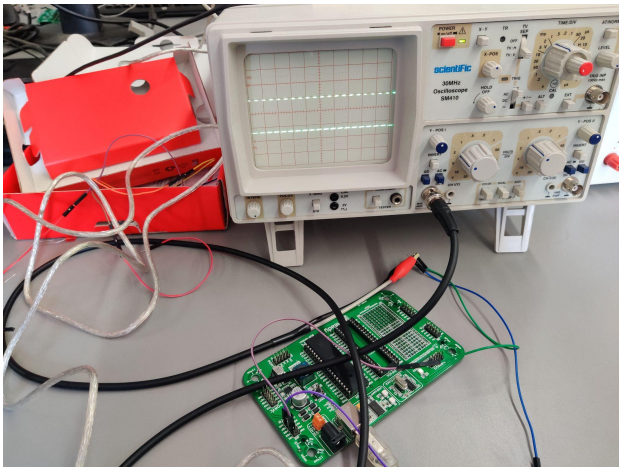
Task 2 Result



Video

Task 3 Result





Video

Code

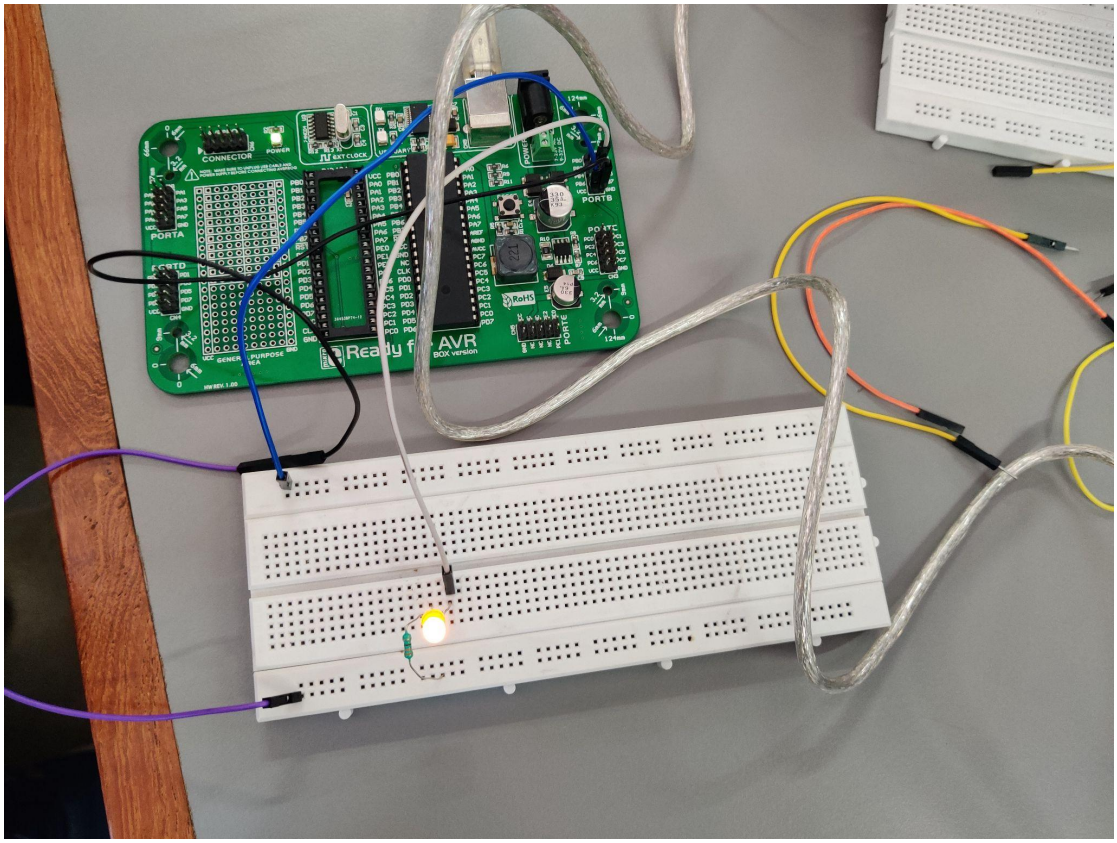
```
// Task 1
void main()
{
    DDRB = 0X08;
    OCR0 = 204;
    TCCR0 = 0X62;
    while (1);
}

// Task 2
void main()
{
    DDRB = 0X08;
    OCR0 = 90;
    TCCR0 = 0X62;
    while (1);
}

// Task 3
void main()
{
    int flag = 0;
    DDRB = 0X08;

    while (1)
    {
        TCCR0 = 0x62;
        if (flag == 0)
        {
            OCR0 = 204;
            delay_ms(5000);
            flag = 1;
        }
        else
        {
            OCR0 = 90;
            delay_ms(5000);
            flag = 0;
        }
    }
}
```

Description

In this experiment, firstly the connections are similar to that of Experiment 3a. We had to use phase correct PWM here. Thus, the value of TCCR0 is different here. Further, we had to generate a square wave of duty cycle 80% and 35% and thus, the value of OCR0 is different. As again evident from the images above, we can it in action. Also, note that the CRO had some issues and hence the wave is not stable.

Experiment 4b

Experiment Result



Code

Same as Experiment 4a.

[Video Task 1]

[Video Task 2]

[Video Task 3]

Description

This experiment was quite simple and connections were similar to experiment 3b. Here, also the difference was in intensity. The intensity with which the led glowed was again different. In task 3 it can be observed quite clearly.