

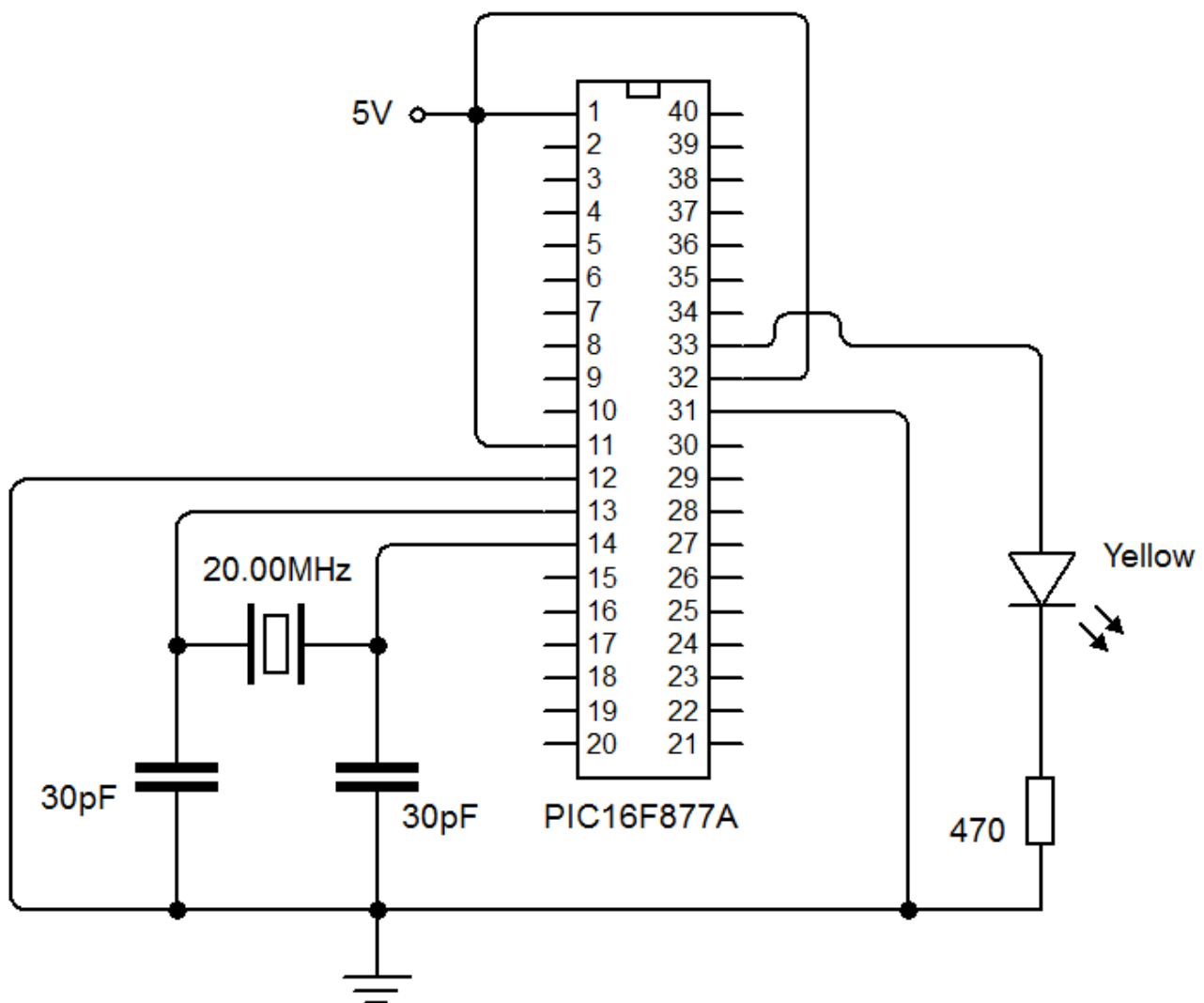
Experiment no : 01

Experiment name : Develop a program for LED blinking system.

Apparatus:

- Microcontroller PIC16F877A - [1 piece]
- Computer [Laptop or Desktop]
- PIC Programmer Kit - [1 piece]
- LED - [1 piece]
- Resistor 470 Ω - [1 piece]
- Breadboard - [1 piece]
- Connecting Wires

Circuit Diagram :



Program :

```
/* Name      : main.c
 * Purpose    : Main file for blinking an LED with PIC16F877A.
 * User       : N@IM_0CTAL
 * Github     : https://github.com/naimoctal
 * Software   : MPLAB IDE v8.76
 */

#include<htc.h>

// Configuration word for PIC16F877A
__CONFIG( FOSC_HS & WDTE_OFF & PWRTE_ON & CP_OFF & BOREN_ON
          & LVP_OFF & CPD_OFF & WRT_OFF & DEBUG_OFF);

// Define LED pin
#define LED RB0

// Define CPU Frequency
// This must be defined, if __delay_ms() or
// __delay_us() functions are used in the code
#define _XTAL_FREQ 20000000

void main(void)
{
    TRISB0 = 0;           // Make RB0 pin output
    LED = 0;              // Make RB0 low

    while(1)
    {
        __delay_ms(500);  // Half sec delay
        LED = 0;          // LED off
        __delay_ms(500);  // Half sec delay
        LED = 1;          // LED on
    }
}
```

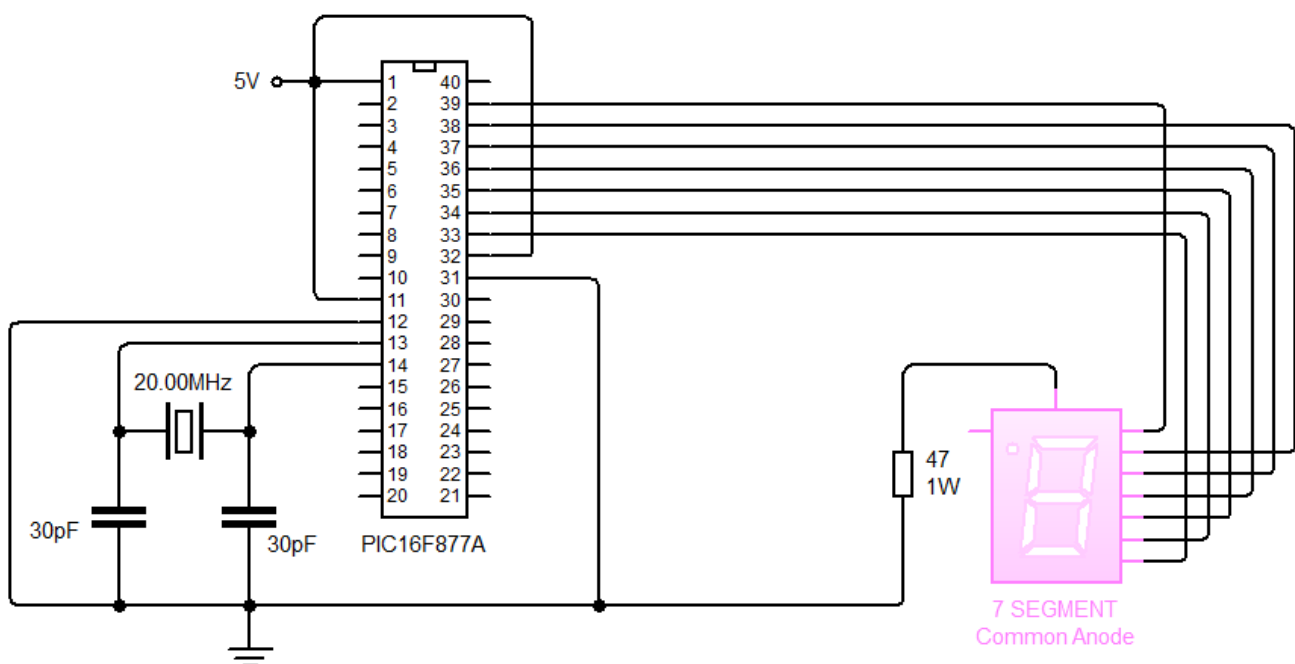
Experiment no : 02

Experiment name : Develop a program for displaying decimal digit (0-9) or hexadecimal digit on 7 - Segment display by using any MCU.

Apparatus:

- Microcontroller PIC16F877A - [1 piece]
- Computer [Laptop or Desktop]
- PIC Programmer Kit - [1 piece]
- 7 Segment display (Common Cathode) - [1 piece]
- Resistor 47Ω 1 watt - [1 piece]
- Breadboard - [1 piece]
- Connecting Wires

Circuit Diagram :



Program :

```
/* Name      : main.c
 * Purpose   : Interfacing 7 segment Display with PIC Microcontroller
 * User      : N@IM_0CTAL
 * Github    : https://github.com/naimoctal
 * Software  : MPLAB IDE v8.76
 */

#include<htc.h>

// Pin description
/* PORTB is used for 7 segment
RB0 is attached with 'A' segment
RB1 is attached with 'B' segment
RB2 is attached with 'C' segment
RB3 is attached with 'D' segment
RB4 is attached with 'E' segment
RB5 is attached with 'F' segment
RB6 is attached with 'G' segment
*/

// Configuration word for PIC16F877A
__CONFIG( FOSC_HS & WDTE_OFF & PWRTE_ON & CP_OFF & BOREN_ON
          & LVP_OFF & CPD_OFF & WRT_OFF & DEBUG_OFF);

// Define CPU Frequency
// This must be defined, if __delay_ms() or
// __delay_us() functions are used in the code
#define _XTAL_FREQ 20000000

// Display on seven segment function
void DisplayOn7Segment(char ch) // ch can have a value from '0' to 'F' only
{
    switch(ch)
    {
        case '0':    PORTB = 0x3F; break;
        case '1':    PORTB = 0x06; break;
        case '2':    PORTB = 0x5B; break;
        case '3':    PORTB = 0x4F; break;
        case '4':    PORTB = 0x66; break;
        case '5':    PORTB = 0x6D; break;
        case '6':    PORTB = 0x7D; break;
        case '7':    PORTB = 0x07; break;
```

```

    case '8':    PORTB = 0x7F; break;
    case '9':    PORTB = 0x6F; break;
    case 'A':    PORTB = 0x77; break;
    case 'B':    PORTB = 0x7C; break;
    case 'C':    PORTB = 0x39; break;
    case 'D':    PORTB = 0x5E; break;
    case 'E':    PORTB = 0x79; break;
    case 'F':    PORTB = 0x71; break;

```

```

    default: PORTB = 0x3F; break;

```

```

}

```

```

}

```

```

// Main function

```

```

void main()

```

```

{

```

```

    char ch = '0';        // Character to be displayed on 7seg

```

```

    TRISB = 0x00;          // Make PORTB pins output

```

```

    PORTB = 0x00;          // Make PORTB pins low

```

```

    while(1)

```

```

    {

```

```

        DisplayOn7Segment(ch);    // Display ch on 7seg

```

```

        __delay_ms(1000);    // One sec delay

```

```

        switch(ch)              // Update ch with new value to be displayed

```

```

        {

```

```

            case '0':    ch = '1'; break;

```

```

            case '1':    ch = '2'; break;

```

```

            case '2':    ch = '3'; break;

```

```

            case '3':    ch = '4'; break;

```

```

            case '4':    ch = '5'; break;

```

```

            case '5':    ch = '6'; break;

```

```

            case '6':    ch = '7'; break;

```

```

            case '7':    ch = '8'; break;

```

```

            case '8':    ch = '9'; break;

```

```

            case '9':    ch = 'A'; break;

```

```

            case 'A':    ch = 'B'; break;

```

```

            case 'B':    ch = 'C'; break;

```

```

            case 'C':    ch = 'D'; break;

```

```

            case 'D':    ch = 'E'; break;

```

```

            case 'E':    ch = 'F'; break;

```

```

            case 'F':    ch = '0'; break;

```

```

        default: ch = '0'; break;
    }
}
}

```

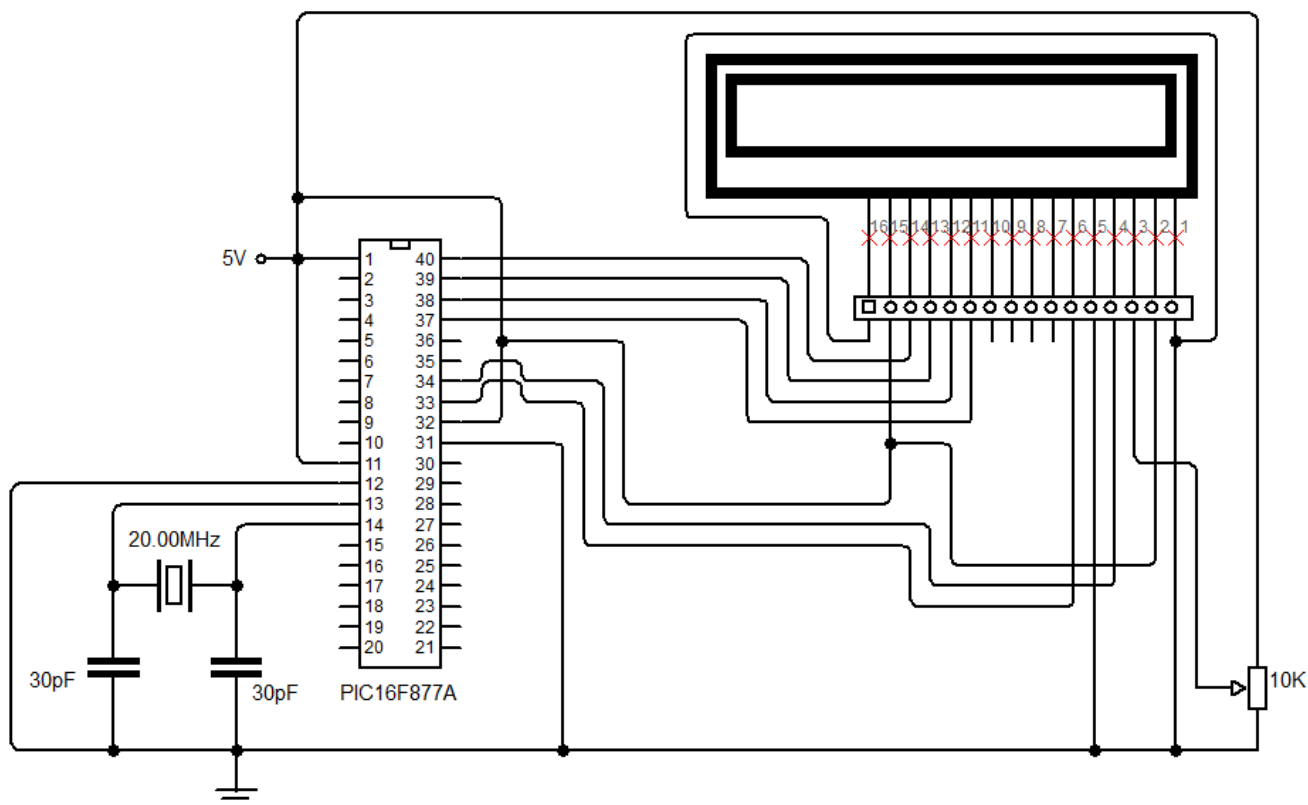
Experiment no : 03

Experiment name : Develop a program for interfacing LCD by using any MCU.

Apparatus:

- Microcontroller PIC16F877A - [1 piece]
- Computer [Laptop or Desktop]
- PIC Programmer Kit - [1 piece]
- 16x2 LCD - [1 piece]
- Potentiometer 10k Ω - [1 piece]
- Breadboard - [1 piece]
- Connecting Wires

Circuit Diagram :



Program:

```
/* Name      : main.c
 * Purpose   : Main file for using LCD with PIC16F877A in 4bit mode
 * User      : N@IM_0CTAL
 * Github    : https://github.com/naimoactal
 * Software  : MPLAB IDE v8.76
 */

#include "Includes.h"

// Configuration word for PIC16F877A
__CONFIG( FOSC_HS & WDTE_OFF & PWRTE_ON & CP_OFF & BOREN_ON
          & LVP_OFF & CPD_OFF & WRT_OFF & DEBUG_OFF);

// Main Function
void main(void)
{
    InitLCD();                                // Initialize LCD in 4bit mode

    ClearLCDScreen();                         // Clear LCD screen
    WriteStringToLCD("Hello World!"); // Write Hello World on LCD

    while(1)
    {
    }
}
```

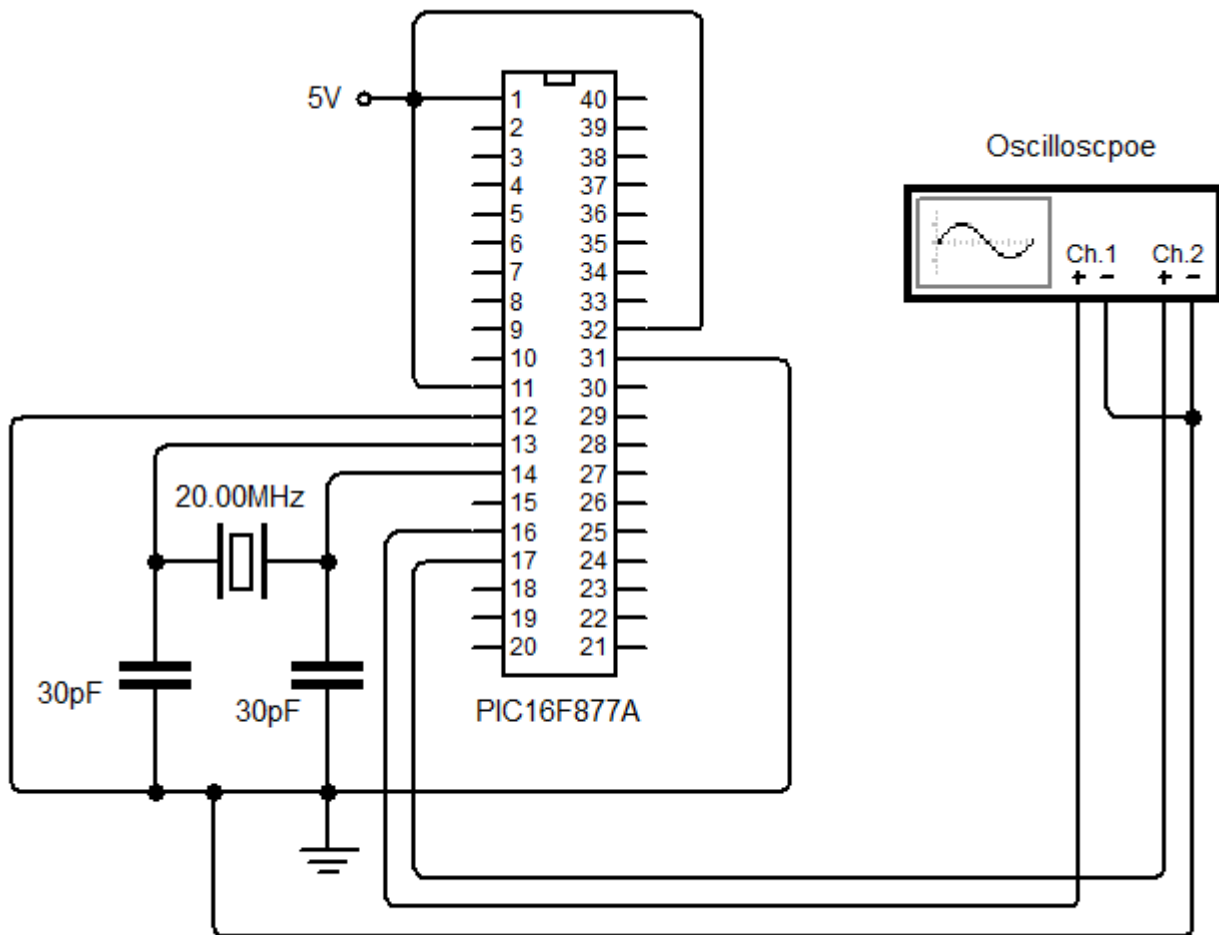
Experiment no : 04

Experiment name : Perform the construction of a MCU based timing pulse generation system by using build-in timer .

Apparatus:

- Microcontroller PIC16F877A - [1 piece]
- Computer [Laptop or Desktop]
- PIC Programmer Kit - [1 piece]
- Oscilloscope - [1 piece]
- Probe - [2 pieces]
- Breadboard - [1 piece]
- Connecting Wires

Circuit Diagram:



Program :

```
/* Name      : main.c
 * Purpose   : Main file for generating PWM using PIC16F877A.
 * User      : N@IM_0CTAL
 * Github    : https://github.com/naimoctal
 * Software  : MPLAB IDE v8.76
 */
```

```
#include "Includes.h"
```

```
// Configuration word for PIC16F877A
```

```
__CONFIG( FOSC_HS & WDTE_OFF & PWRTE_ON & CP_OFF & BOREN_ON
          & LVP_OFF & CPD_OFF & WRT_OFF & DEBUG_OFF);
```

```
// Define CPU Frequency
```



```

// This must be defined, if __delay_ms() or
// __delay_us() functions are used in the code
#define _XTAL_FREQ 20000000

// Main function
void main(void)
{
    InitPWM();

    SetPWMDutyCycle(1,127); //50% duty cycle on CCP1
    SetPWMDutyCycle(2,191); //75% duty cycle on CCP2

    while(1)
    {

    }
}

```

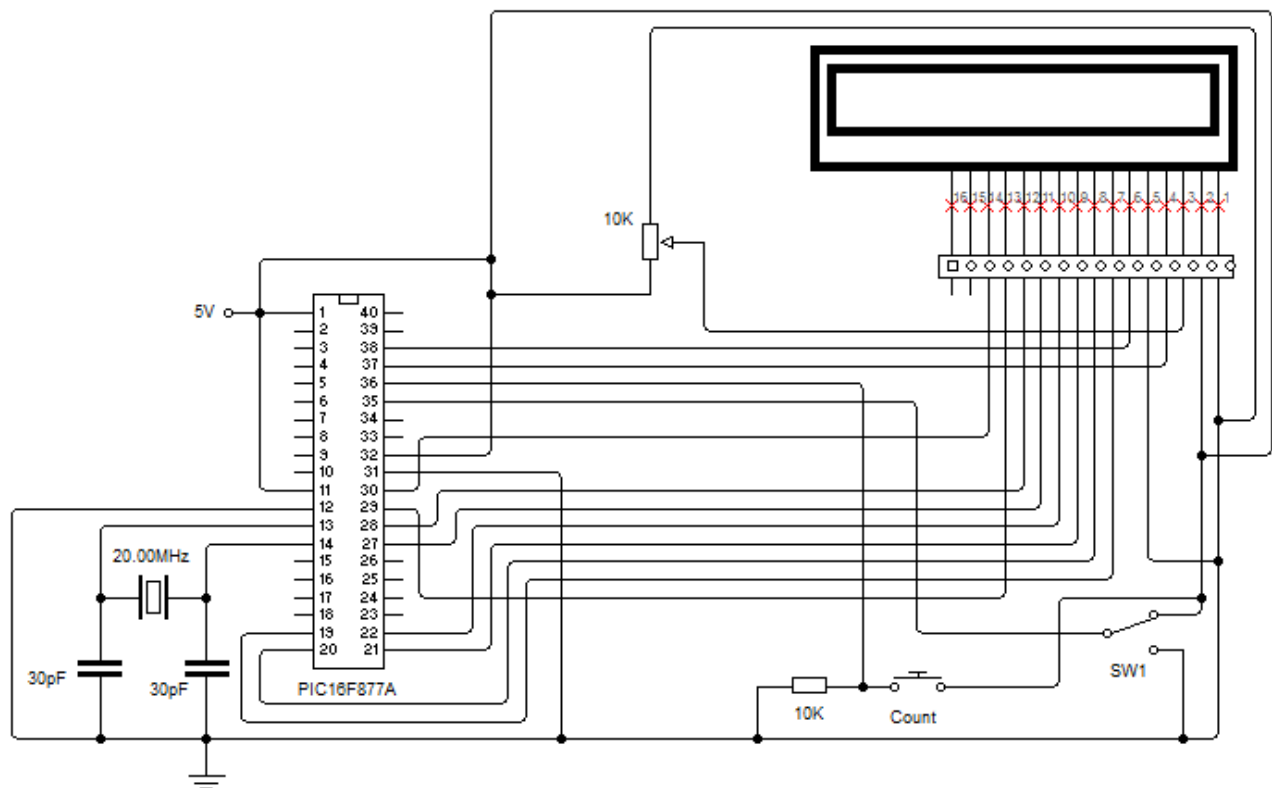
Experiment no : 05

Experiment name : Perform the construction of a MCU based system to count pulses .

Apparatus:

- Microcontroller PIC16F877A - [1 piece]
- Computer [Laptop or Desktop]
- PIC Programmer Kit - [1 piece]
- SPDT switch - [1 piece]
- Push switch - [1 piece]
- Potentiometer 10k Ω - [1 piece]
- Resistor 10k Ω - [1 piece]
- Breadboard - [1 piece]
- Connecting Wires

Circuit Diagram:



Program :

```
/* Name      : main.c
 * Purpose    : Main file for making Up/Down counter using LCD with PIC16F877 in 8bit mode.
 * User       : N@IM_0CTAL
 * Github     : https://github.com/naimoactal
 * Software   : MPLAB IDE v8.76
 */
```

```
#include "Includes.h"
```

```
// Configuration word for PIC16F877
```

```
__CONFIG( FOSC_HS & WDTE_OFF & PWRTE_ON & CP_OFF & BOREN_ON
          & LVP_OFF & CPD_OFF & WRT_ON & DEBUG_OFF);
```

```
// Define Pins
```

```
#define Up_Or_Down          RB2
```

```
#define Count_In            RB3
```

```

#define Up                1
#define Down              0

// Main Function
void main(void)
{
    unsigned int Count = 0;

    InitLCD();                // Initialize LCD in 8bit mode

    TRISB2 = 1;               // Make PORTB2 pin input -> Up/Down Input
    TRISB3 = 1;               // Make PORTB3 pin input -> Count input

    ClearLCDScreen();         // Clear LCD screen
    WriteStringToLCD("Starting..."); // Write Message on LCD

    __delay_ms(500);          // Some delay

    while(1)
    {
        if(Up_Or_Down == Up) // If Up is selected
        {
            if(Count_In)      // If Count button is pressed
            {
                while(Count_In); // Wait for de-assertion of the button
                __delay_ms(20);   // Small delay

                Count++;          // Increment Count value

                if(Count==100)    // If Count reaches 100
                    Count = 0;    // Reset Count
            }
        }
        else if(Up_Or_Down == Down) // If Down is selected
        {
            if(Count_In)          // If Count button is pressed
            {
                while(Count_In);   // Wait for de-assertion of the button
                __delay_ms(20);     // Small delay

                if(Count==0)        // If Count becomes zero
                    Count = 99;     // Reset Count to 99
                else
                    Count--;         // Decrement Count
            }
        }
    }
}

```

```

        ClearLCDScreen();                // Clear LCD screen

        // Display Count on LCD
        if(Count<10)
            WriteDataToLCD(Count+0x30);
        else
        {
            WriteDataToLCD((unsigned char)(Count/10)+0x30);
            WriteDataToLCD((unsigned char)(Count%10)+0x30);
        }

        __delay_ms(200);                // Some delay
    }
}

```

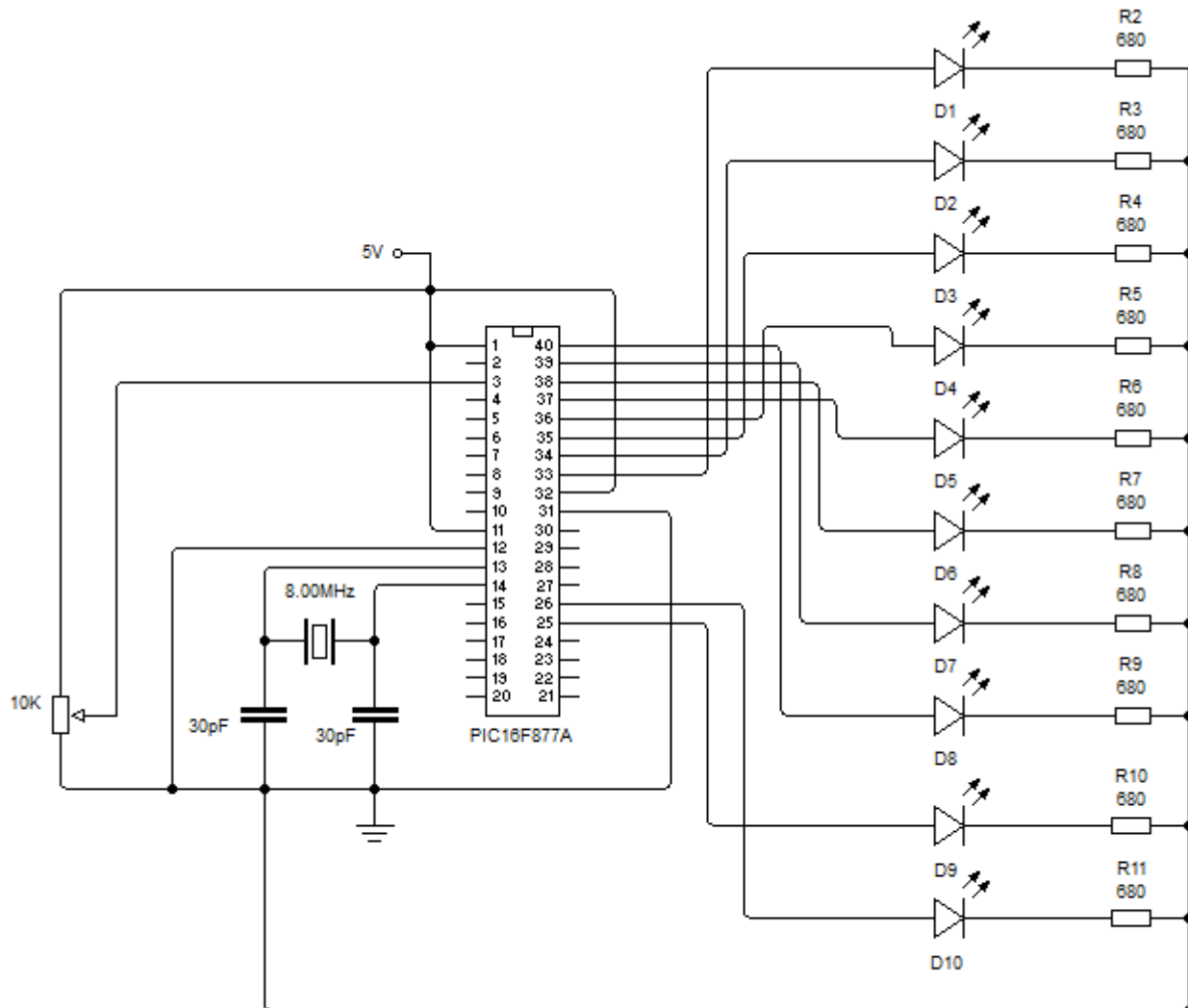
Experiment no : 06

Experiment name : Perform the construction of a MCU based ADC interface system .

Apparatus:

- Microcontroller PIC16F877A - [1 piece]
- Computer [Laptop or Desktop]
- PIC Programmer Kit - [1 piece]
- LEDs - [10 pieces]
- Potentiometer 10k Ω - [1 piece]
- Resistor 680 Ω - [10 pieces]
- Breadboard - [1 piece]
- Connecting Wires

Circuit Diagram:



Program :

```
/* Name      : main.c
 * Purpose    : Main file for ADC interface system using PIC16F877A.
 * User       : N@IM_0CTAL
 * Github     : https://github.com/naimoactal
 * Software   : mikroC Pro PIC
 */
```

```
unsigned int adc;
void main()
{
    ADCON1 = 0x80;
    TRISA = 0xFF; // PORTA is input
    TRISC = 0x3F; // Pins RC7, RC6 are outputs
```

```

TRISB = 0; // PORTB is output
do
{
adc = ADC_Read(1); // Get 10-bit results of AD conversion
//of channel 1
PORTB = adc; // Send lower 8 bits to PORTB
PORTC = adc >> 2; // Send 2 most significant //bits to RC7, RC6
} while(1);
}

```

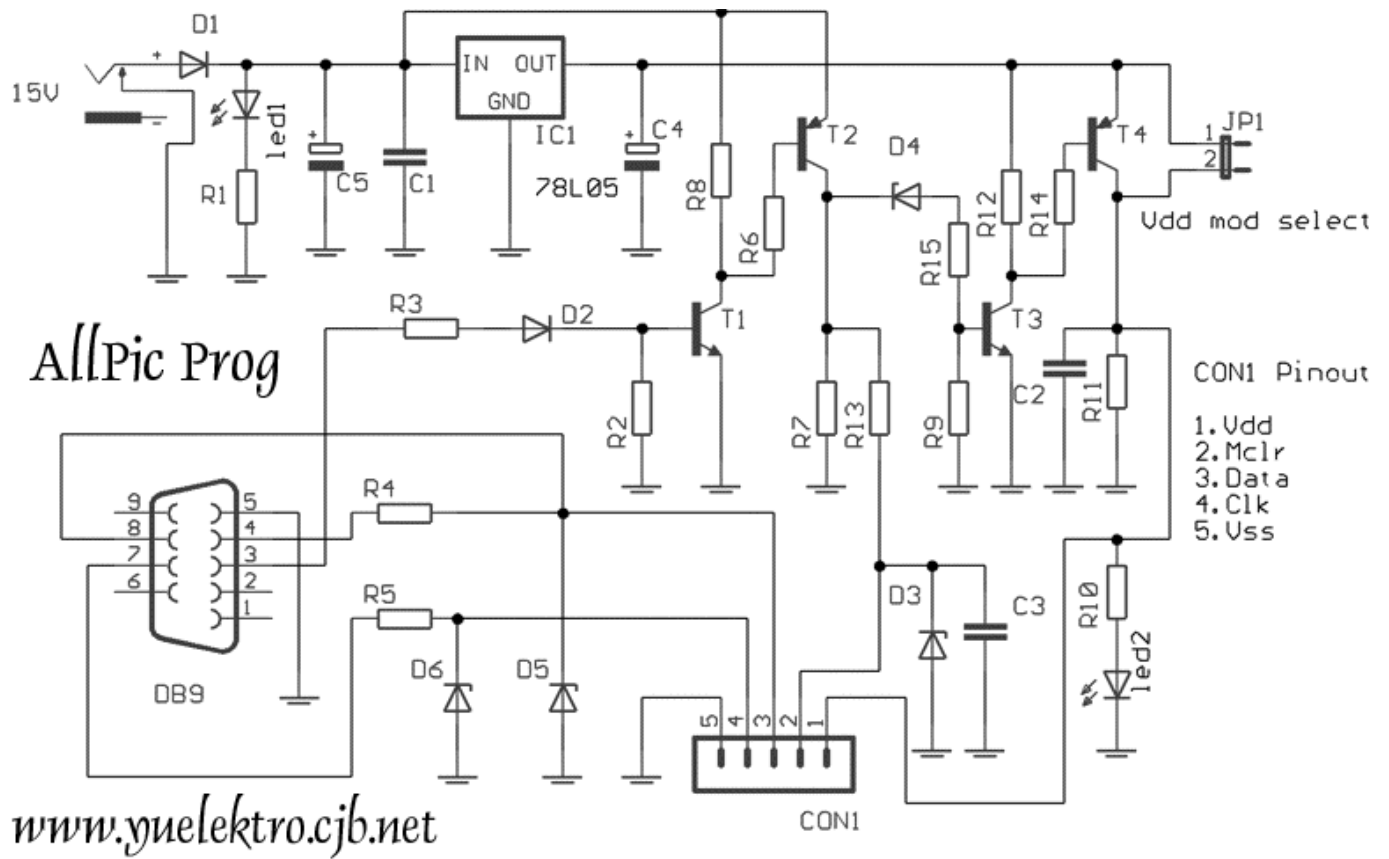
Experiment no : 07

Experiment name : Make a flashing tool to flash a program from computer to MCU.

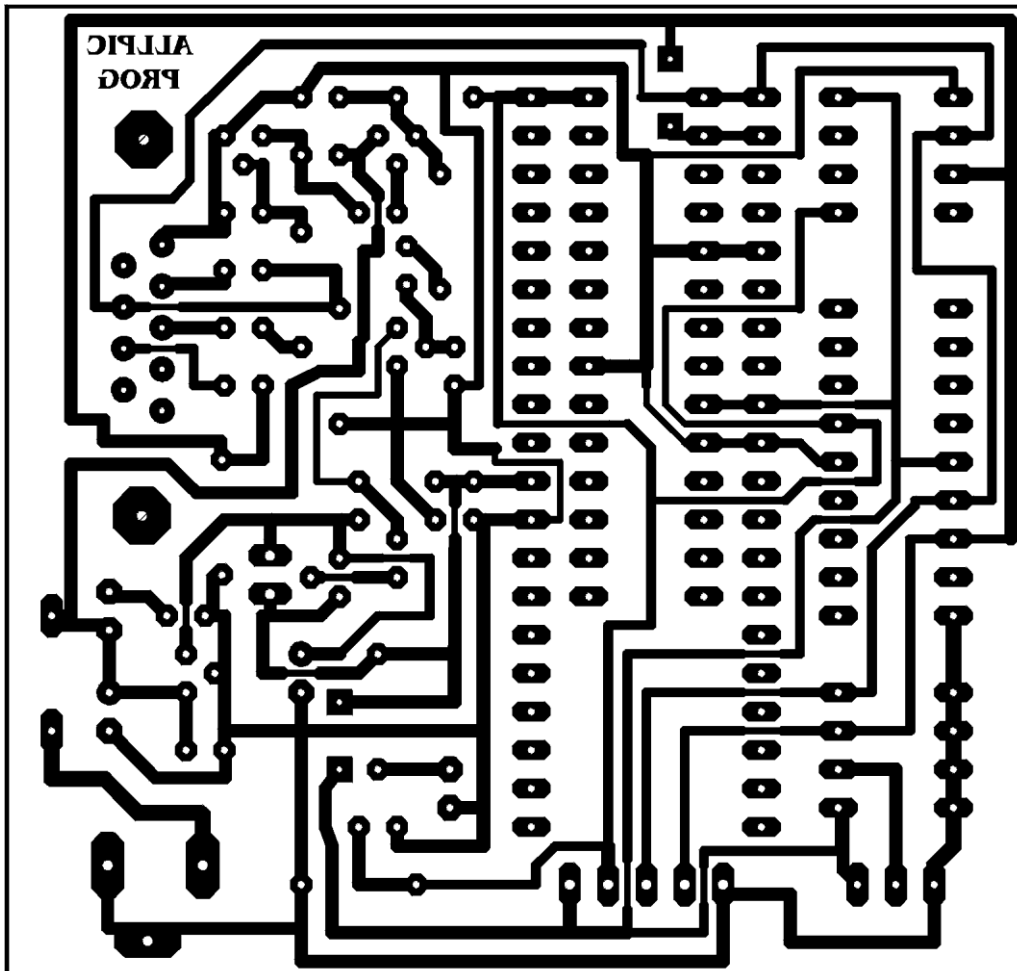
Components:

- Voltage Regulator IC - [Ic1 - 78L05]
- Transistors NPN - [T1,T3 - bc547]
- Transistors PNP - [T2,T4 - bc557]
- Diodes - [D1,D2 - 1n4148]
- Zener diode - [D3 - 13v]
- Zener diode - [D4 - 12v]
- Zener diodes - [D5,D6 - 5.1v]
- Resistor - [R1 - 1.2kΩ]
- Resistor - [R10 - 270Ω]
- Resistor - [R15 - 1kΩ]
- Resistors - [R7,R8 - 10kΩ]
- Resistors - [R4,R5,R13,R14 - 1.8kΩ]
- Resistors - [R2,R3,R6,R9,R11,R12 - 5.6kΩ]
- Capacitors - [c1,c2,c3 - 100nF]
- Capacitor - [c4-10uF/16v]
- Capacitor - [c5-100uF/25v]
- LED - [Any color - 5mm led (2 pieces)]
- 2x8pin,18pin,28pin i 40pin IC bases (per 1 piece)
- Jumper [jp1 - 1x2]
- Jumper [jp2 - 1x3]
- Connector - [con1- 1x5]
- DC power Jack - [1 piece]
- DB9 Rs232 Connector Base - [1 piece]

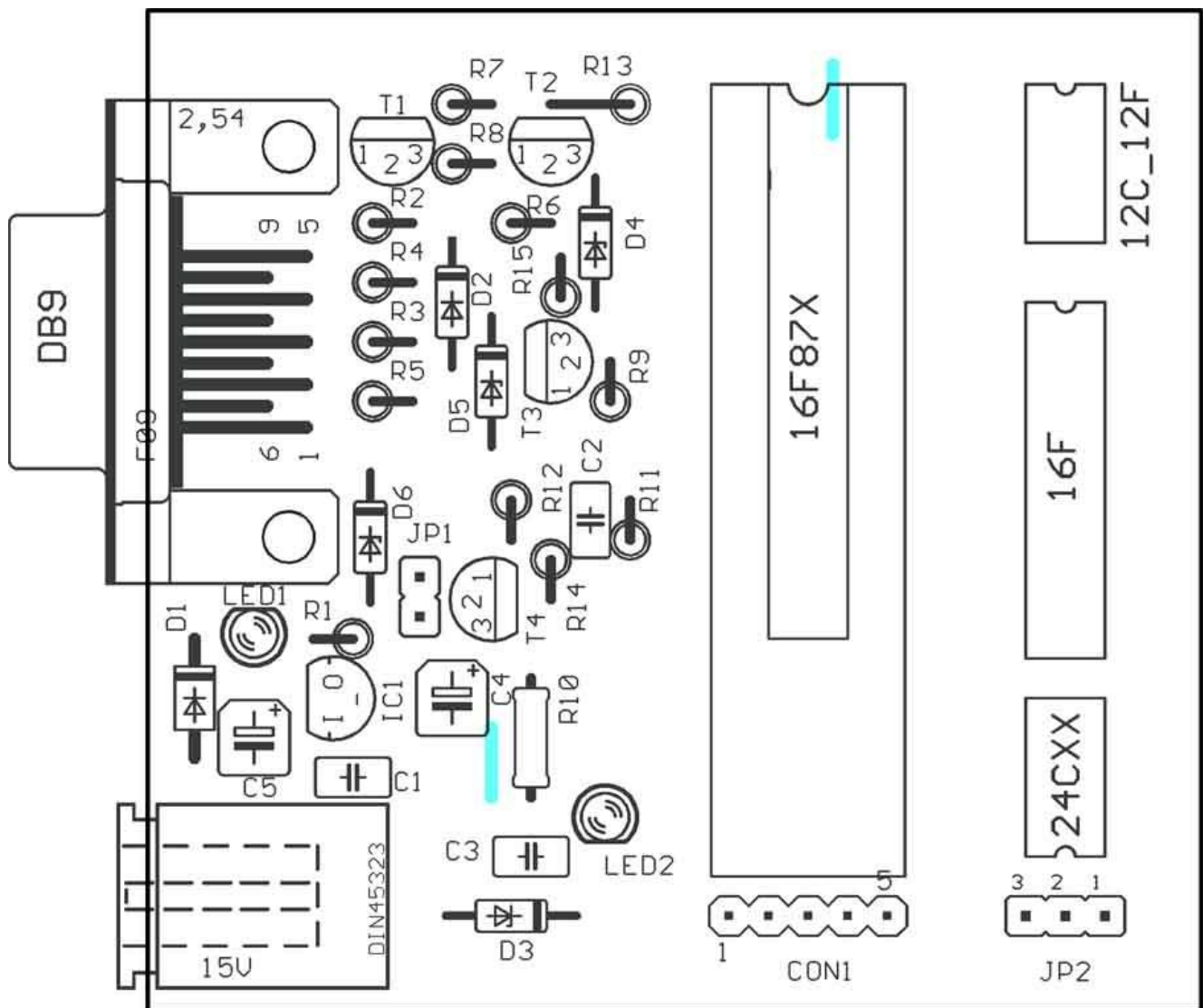
Circuit Diagram :



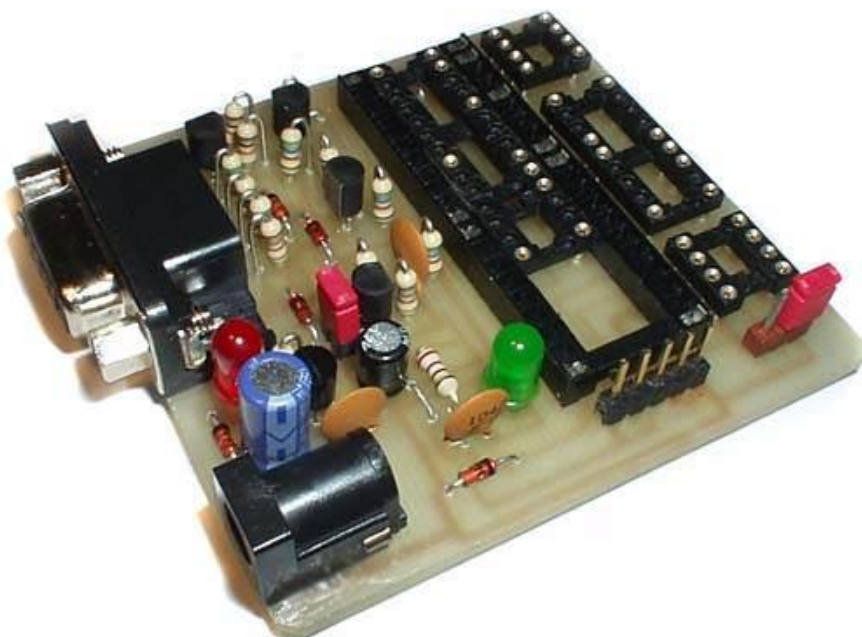
PCB Layout :



Placement :



Complete Picture :



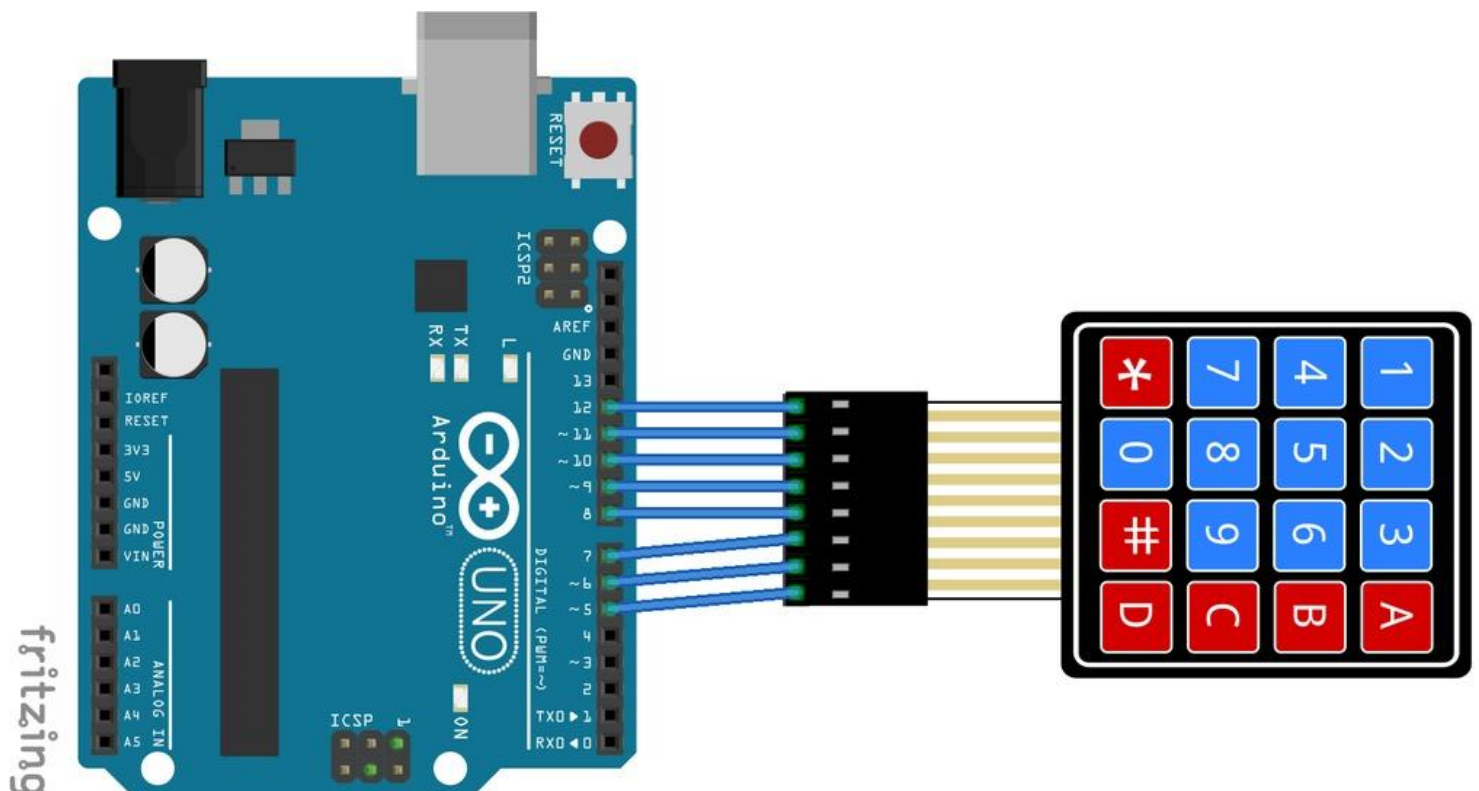
Experiment no : 08

Experiment name : Develop a program for Interfacing Keyboard using MCU/Arduino .

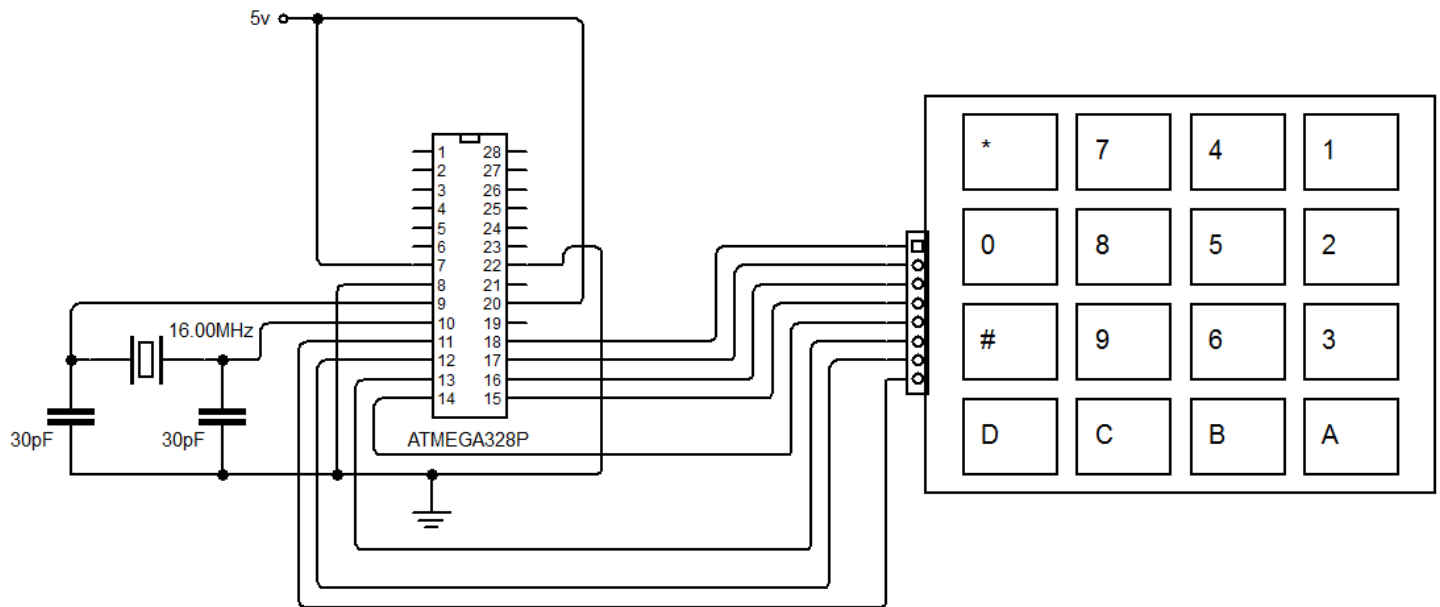
Apparatus:

- Microcontroller Arduino UNO R3 - [1 piece]
- Computer [Laptop or Desktop]
- Breadboard - [1 piece]
- 4x4 Matrix Keypad - [1 piece]
- Connecting Wires

Connection Diagram:



Circuit Diagram:



Program:

```
/* Name      : keypad.ino
 * Purpose   : Interfacing Keyboard using MCU/Arduino.
 * User      : N@IM_0CTAL
 * Github    : https://github.com/naimoactal
 * Software  : Arduino IDE
 */
```

```
#define PIN_R0 12
```

```
#define PIN_R1 11
```

```
#define PIN_R2 10
```

```
#define PIN_R3 9
```

```
#define PIN_C0 8
```

```
#define PIN_C1 7
```

```
#define PIN_C2 6
```

```
#define PIN_C3 5
```

```
const char keys[4][4] = {
```

```
{ '1', '2', '3', 'A' },
```

```
{ '4', '5', '6', 'B' },
```

```
{ '7', '8', '9', 'C' },
```

```
{ '*', '0', '#', 'D' }
```

```

};

// setup for reading key on a specify row r -- set PIN_Rr to 0, all other PIN_Rx to 1
void setupForReadingRow(int r) {
    digitalWrite(PIN_R0, r != 0);
    digitalWrite(PIN_R1, r != 1);
    digitalWrite(PIN_R2, r != 2);
    digitalWrite(PIN_R3, r != 3);
}

// read which column with key press; -1 if none
int readColumnOfKeyPress() {
    int c0 = digitalRead(PIN_C0);
    int c1 = digitalRead(PIN_C1);
    int c2 = digitalRead(PIN_C2);
    int c3 = digitalRead(PIN_C3);
    int c = -1;
    if (c0 == 0) {
        c = 0;
    } else if (c1 == 0) {
        c = 1;
    } else if (c2 == 0) {
        c = 2;
    } else if (c3 == 0) {
        c = 3;
    }
    return c;
}

void setup() {
    pinMode(PIN_R0, OUTPUT);
    pinMode(PIN_R1, OUTPUT);
    pinMode(PIN_R2, OUTPUT);
    pinMode(PIN_R3, OUTPUT);
    pinMode(PIN_C0, INPUT_PULLUP);

```

```

pinMode(PIN_C1, INPUT_PULLUP);
pinMode(PIN_C2, INPUT_PULLUP);
pinMode(PIN_C3, INPUT_PULLUP);
Serial.begin(115200);
}

void promptForKeyPressOnRow(int r) {
    Serial.println(String("press a key on row ") + String(r) + String(":"));
    setupForReadingRow(r);
    while (true) {
        int c = readColumnOfKeyPress();
        if (c != -1) {
            char key = keys[r][c];
            Serial.println(String("- you pressed [") + String(key) + String("]"));
            break;
        }
    }
}

void loop() {
    promptForKeyPressOnRow(0);
    promptForKeyPressOnRow(1);
    promptForKeyPressOnRow(2);
    promptForKeyPressOnRow(3);
}

```

Experiment no : 09

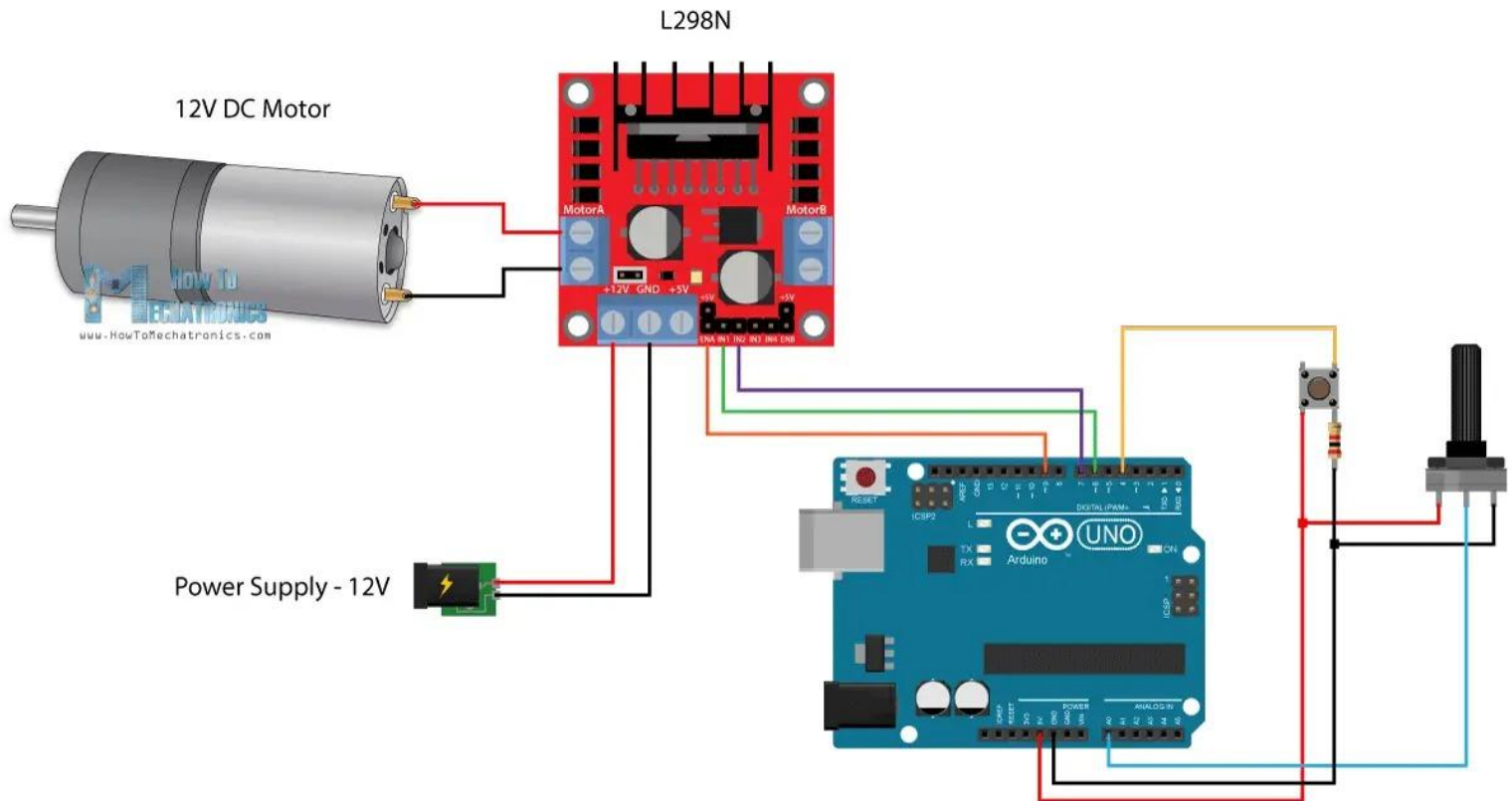
Experiment name : Develop a program for Interfacing DC motor by using MCU/Arduino .

Apparatus:

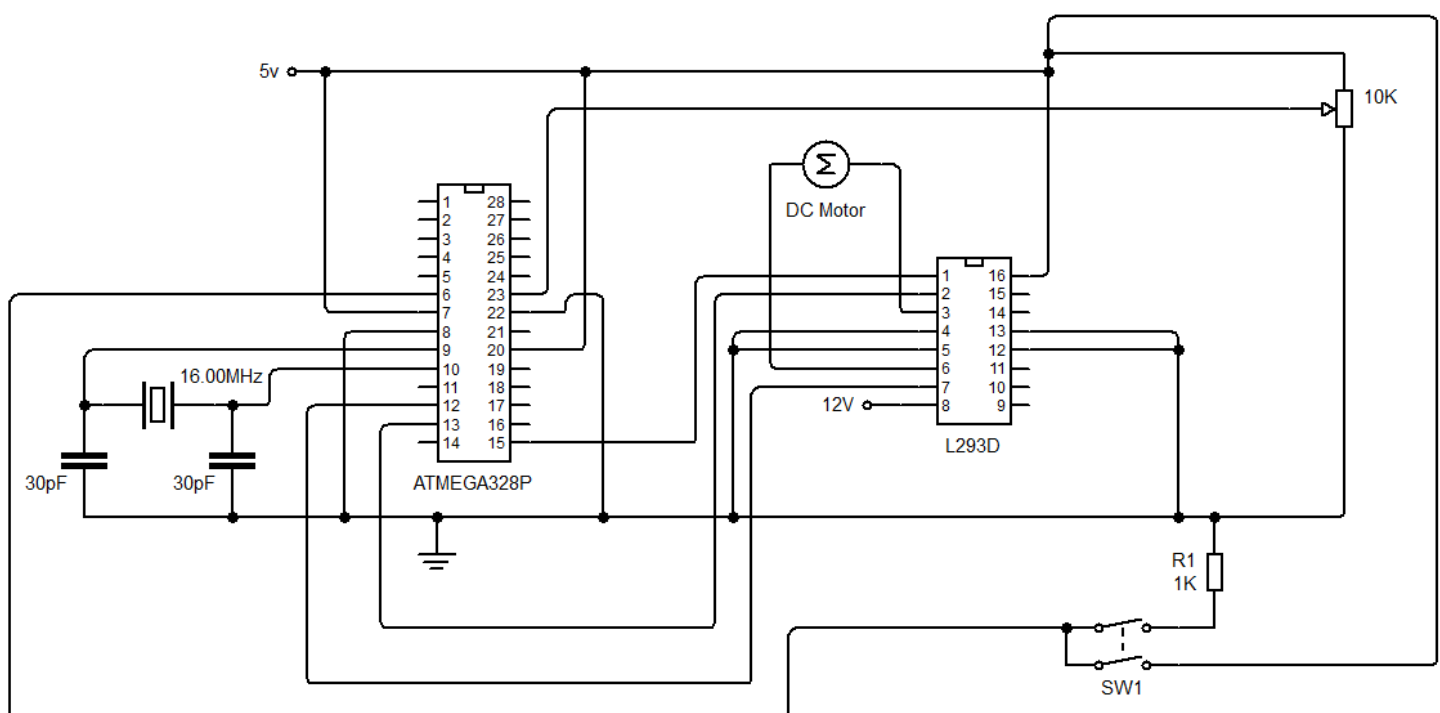
- Microcontroller Arduino UNO R3 - [1 piece]
- Computer [Laptop or Desktop]
- L298N/ L293D Driver - [1 piece]
- 12V High Torque DC Motor - [1 pieces]
- DC Motor w/ Plastic Tire Wheel - [1 pieces]

- Breadboard - [1 piece]
- Potentiometer 10k Ω - [1 piece]
- Push switch - [1 piece]
- Resistor 1k Ω - [1 piece]
- Connecting Wires

Connection Diagram:



Circuit Diagram:



Program:

```
/* Name      : DC_motor.ino
 * Purpose   : Interfacing DC motor by using MCU/Arduino.
 * User      : N@IM_0CTAL
 * Github    : https://github.com/naimoctal
 * Software  : Arduino IDE
 */

#define enA 9
#define in1 6
#define in2 7
#define button 4

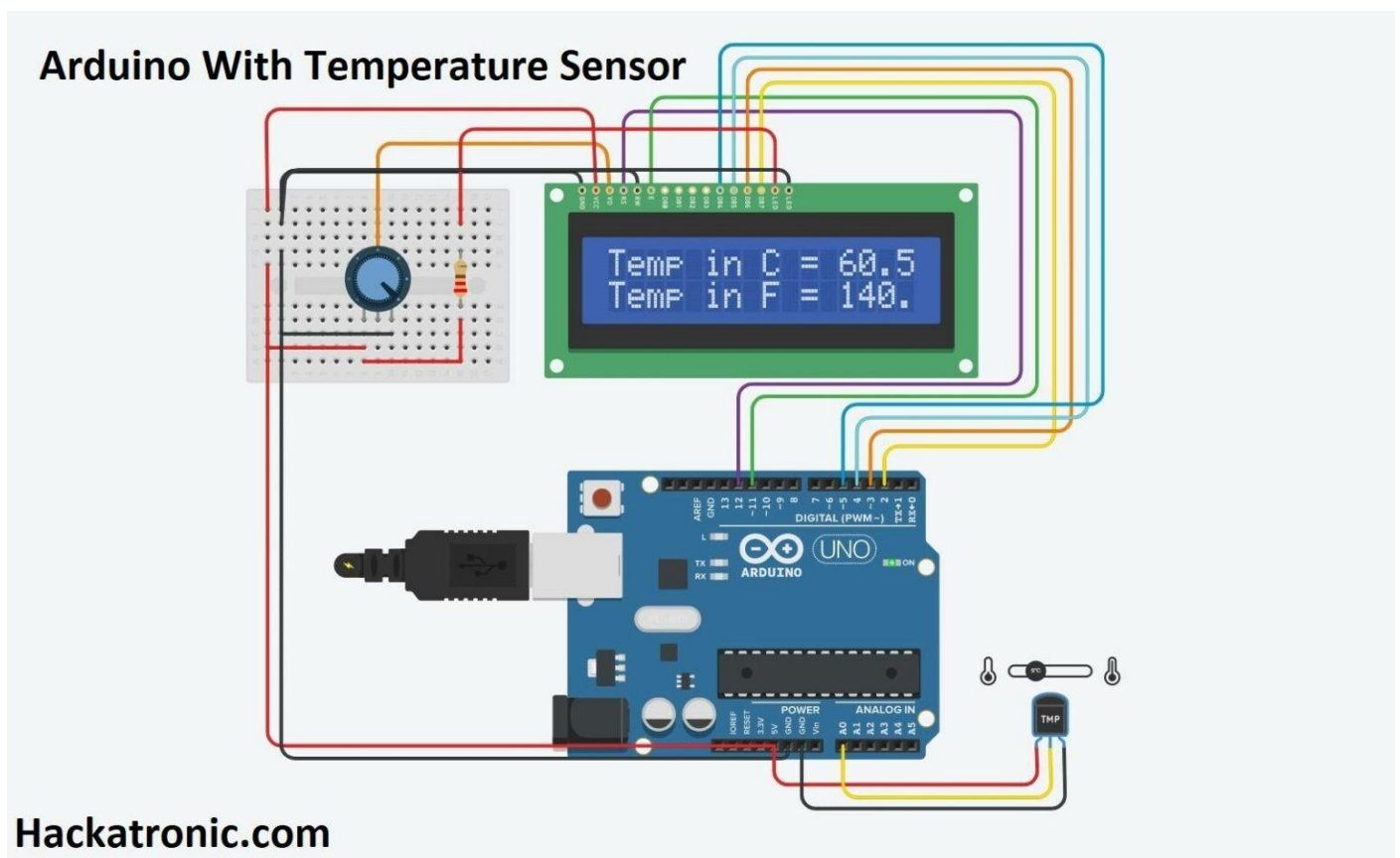
int rotDirection = 0;
int pressed = false;

void setup() {
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(button, INPUT);
  // Set initial rotation direction
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
}

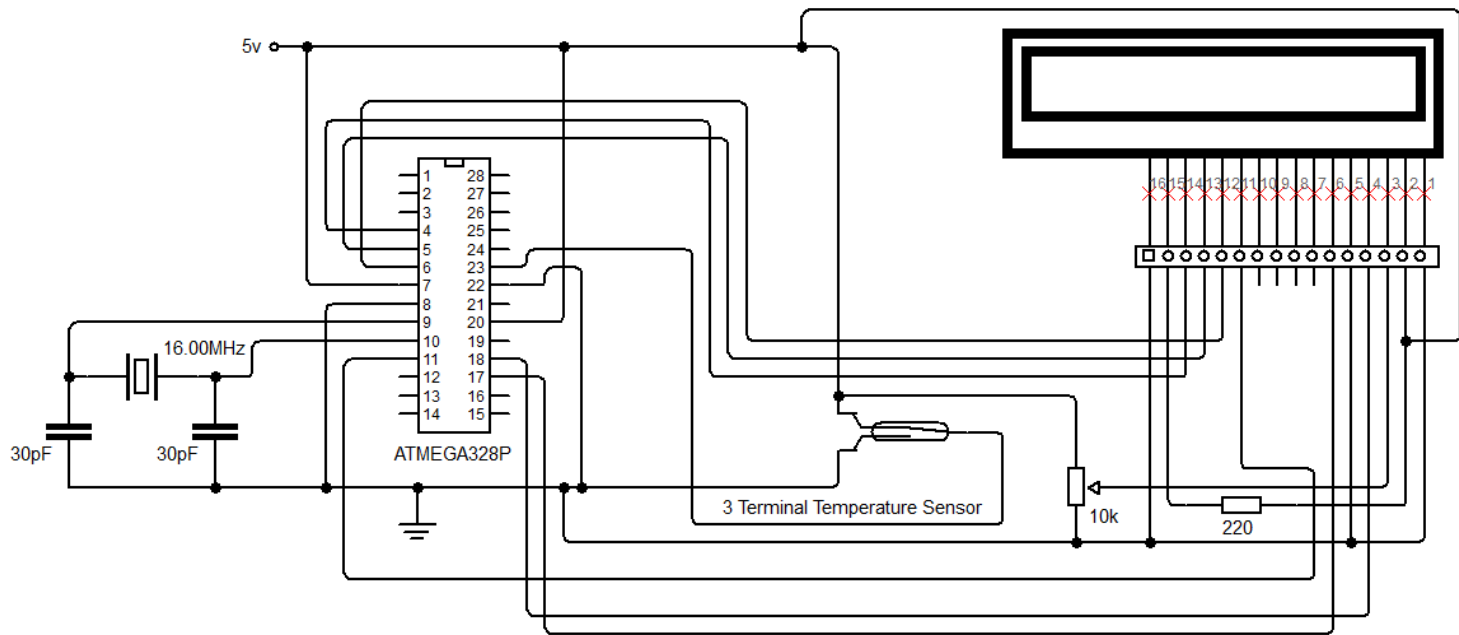
void loop() {
  int potValue = analogRead(A0); // Read potentiometer value
  int pwmOutput = map(potValue, 0, 1023, 0, 255); // Map the potentiometer value from 0 to 255
  analogWrite(enA, pwmOutput); // Send PWM signal to L298N Enable pin

  // Read button - Debounce
  if (digitalRead(button) == true) {
    pressed = !pressed;
  }
  while (digitalRead(button) == true);
  delay(20);

  // If button is pressed - change rotation direction
  if (pressed == true & rotDirection == 0) {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    rotDirection = 1;
    delay(20);
  }
  // If button is pressed - change rotation direction
  if (pressed == false & rotDirection == 1) {
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    rotDirection = 0;
    delay(20);
  }
}
```



Circuit Diagram:



Program:

```
/* Name      : temperature.ino
 * Purpose    : Interfacing temperature sensor by using Arduino kit
 * User       : N@IM_0CTAL
 * Github     : https://github.com/naimoactal
 * Software   : Arduino IDE
 */
```

```
#include <LiquidCrystal.h>
```

```
float temp;
```

```
int sensor = A0;
```

```
float tempc;
```

```
float tempf;
```

```
LiquidCrystal lcd (12, 11, 5, 4, 3, 2);
```

```
//The instance of the LiquidCrystal lcd shows LCD pins connected to the Arduino digital outputs.
```

```
void setup () {
```

```
// set the number of columns and the number of lines of lcd
```

```
lcd.begin (16, 2);
```

```
}
```

```
void loop () {
```

```
temp=analogRead(sensor);
```

```
//Read analog voltage from sensor and store it in a temporary float variable.
```

```
tempc=(temp*4.88)/10;
```

```
//This line converts float value into °C.
```

```
tempf=(tempc*1.8)+32;
```



```
//This line converts °C into Fahrenheit.
```

```
lcd.setCursor(0,0);  
lcd.print("Temp in C = ");  
lcd.println(tempc);
```

```
//These two lines print Temperature value in °C.
```

```
lcd.setCursor(0,1);  
lcd.print("Temp in F = ");  
lcd.println(tempf);
```

```
//Above two lines print value in Fahrenheit  
}
```

Experiment no : 11

Experiment name : Develop a program for driving electromagnetic relay by using Arduino kit with any wireless.

Apparatus:

- Microcontroller Arduino UNO R3 - [1 piece]
- Computer [Laptop or Desktop]
- Breadboard - [1 piece]
- Bluetooth Module HC05 - [1 piece]
- Relay Module 4 Channel - [1 piece]
- Connecting Wires

Programm :

```
/* Name      : wireless_relay.ino  
 * Purpose   : Driving electromagnetic relay by using Arduino kit with any wireless  
 * User      : N@IM_OCTAL  
 * Github    : https://github.com/naimoctal  
 * Software  : Arduino IDE  
 */
```

```
long int ac=2;  
long int bulb=3;  
long int heater=4;  
long int fan=5;  
char x;  
void setup()  
{  
  pinMode(ac,OUTPUT);  
  pinMode(bulb,OUTPUT);  
  pinMode(heater,OUTPUT);  
  pinMode(fan,OUTPUT);  
  digitalWrite(ac,LOW);  
  digitalWrite(bulb,LOW);
```

```

digitalWrite(heater,LOW);
digitalWrite(fan,LOW);
Serial.begin(9600);
}
void loop()
{
  if(Serial.available()>0)
  {
    x=Serial.read();
  }
  if(x=='a')
    digitalWrite(ac,HIGH);
  if (x=='b')
    digitalWrite(ac,LOW);
  if (x=='c')
    digitalWrite(bulb,HIGH);
  if(x=='d')
    digitalWrite(bulb,LOW);
  if(x=='e')
    digitalWrite(heater,HIGH);
  if(x=='f')
    digitalWrite(heater,LOW);
  if(x=='g')
    digitalWrite(fan,HIGH);
  if(x=='h')
    digitalWrite(fan,LOW);
}

```

Connection Diagram:

