AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

FACULTY OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE 4103: Microprocessor and Embedded Systems Laboratory

Fall 2023-2024

Section: L, Group:8

OEL REPORT:

Implementation of Weather Forecasting System Using BMP180 Sensor, I2C Converter, 16×2 LCD Display with Arduino Code of an Arduino.

Supervised By

PROTIK PARVEZ SHEIKH

Group members:

Name	ID	Department
Nabiha Tahsin	21-45685-3	CSE
Md. Abu Talha	21-45688-3	CSE
Tanvir Arafat Sahil	21-45692-3	CSE
Mahidur Rahman Lincon	21-45700-3	CSE
Faisal Ahmed	21-45783-3	CSE

Date of Submission: 14.12.23

Title: Implementation of weather forecasting system using BMP180 sensor, I2C Converter, 16×2 LCD Display with Arduino Code of an Arduino.

Introduction: The goal of this project is to learn how to interface a BMP180 sensor with an Arduino to measure pressure and temperature. The BMP180 sensor here is a barometric sensor capable of measuring pressure, temperature. The hardware components are implemented in the breadboard with Arduino and simulation is done on proteus to get satisfactory result.

Apparatus:

- Arduino UNO Board
- BMP180 Sensor
- 16x2 LCD Display
- I2C converter
- Breadboard and Jump Wires

Circuit Diagram:

The circuit diagram was connected as follows, the program was uploaded into the board, and then the program was run. The weather parameters were observed on the LCD display.

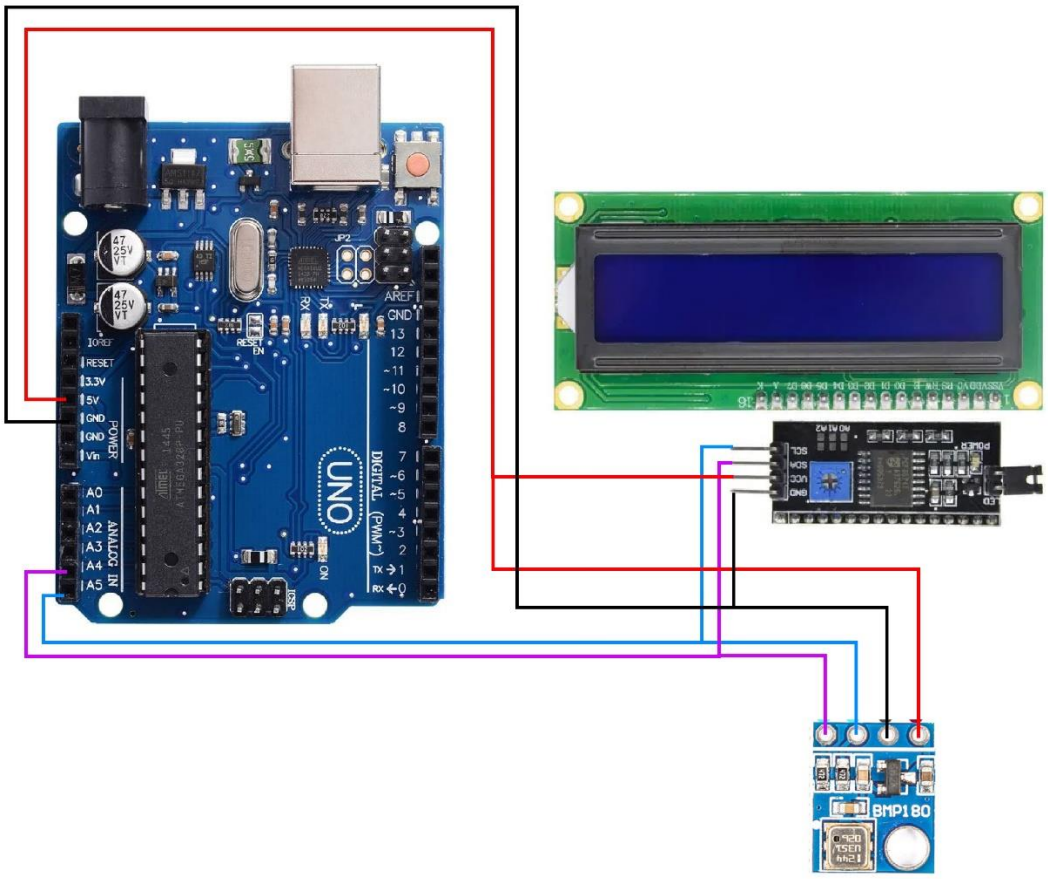


Fig. Arduino Uno with BMP180 and 16×2 LCD display

Necessary Code:

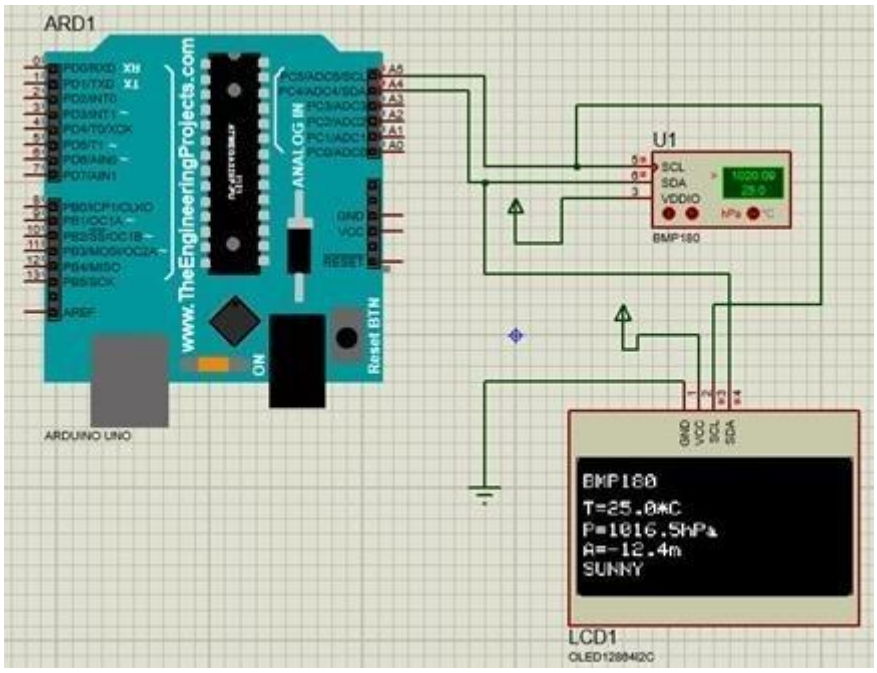
```
#include <Servo.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

Servo myservo;
#define ir_enter 2
#define ir_back  4
#define ir_car1 5
#define ir_car2 6
#define ir_car3 7
#define ir_car4 8

 int S1 = 0, S2 = 0, S3 = 0, S4 = 0;
int flag1 = 0, flag2 = 0;
int slot = 4;

 void setup() {
  Serial.begin(9600);

  pinMode(ir_car1, INPUT);
  pinMode(ir_car2, INPUT);
  pinMode(ir_car3, INPUT);
  pinMode(ir_car4, INPUT);

  pinMode(ir_enter, INPUT);
  pinMode(ir_back, INPUT);

  myservo.attach(3);
  myservo.write(90);

  lcd.init();
  lcd.backlight();
  lcd.setCursor (0, 0);
  lcd.print(" Hi Welcome to ");
  lcd.setCursor (0, 1);
  lcd.print("  Car Parking");
  delay (5000);
  lcd.clear();
  lcd.setCursor (0, 0);
  lcd.print("    We Are   ");
  lcd.setCursor (0, 1);
  lcd.print("   Group 8  ");
  lcd.setCursor (0, 2);
  lcd.print("       System     ");
  delay (5000);
  lcd.clear();

  Read_Sensor();
```

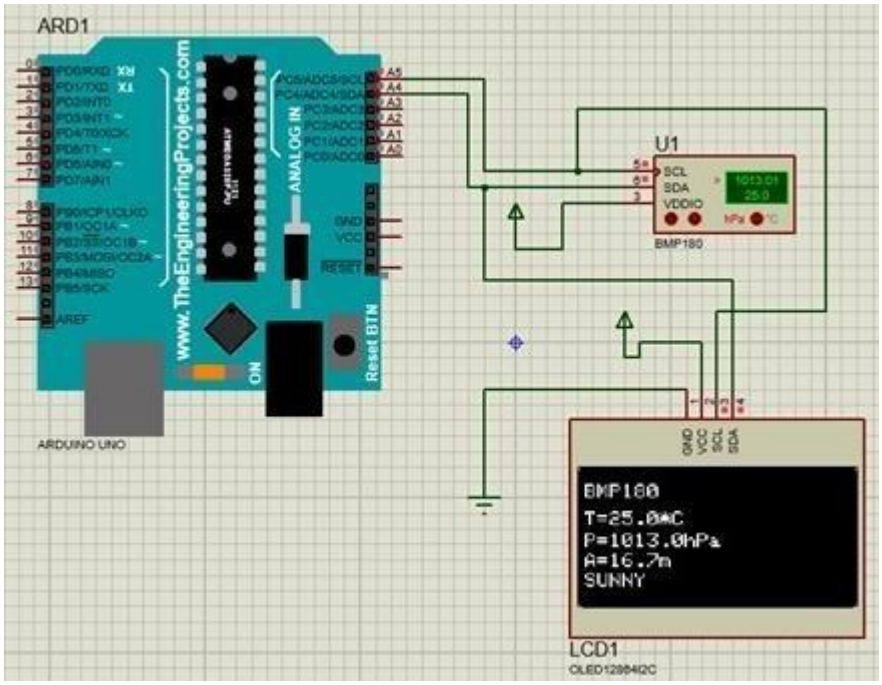
```
  int total = S1 + S2 + S3 + S4;
  slot = slot - total;
}

void loop() {

  Read_Sensor();

  lcd.setCursor (0, 0);
  lcd.print("  Available- ");
  lcd.print(slot);
  lcd.print("    ");

  lcd.setCursor (0, 1);
  if (S1 == 1) {
    lcd.print("F1:Full ");
  }
  else {
    lcd.print("F1:Emty");
  }

  lcd.setCursor (9, 1);
  if (S2 == 1) {
    lcd.print("F2:Full ");
  }
  else {
    lcd.print("F2:Emty");
  }

  lcd.setCursor (0, 2);
  if (S3 == 1) {
    lcd.print("S3:Full ");
  }
  else {
    lcd.print("S3:Empty");
  }

  lcd.setCursor (11, 2);
  if (S4 == 1) {
    lcd.print("S4:Full ");
  }
  else {
    lcd.print("S4:Empty");
  }

  if (digitalRead (ir_enter) == 0 && flag1 == 0) {
    if (slot > 0) {
      flag1 = 1;
      if (flag2 == 0) {
        myservo.write(180);
```

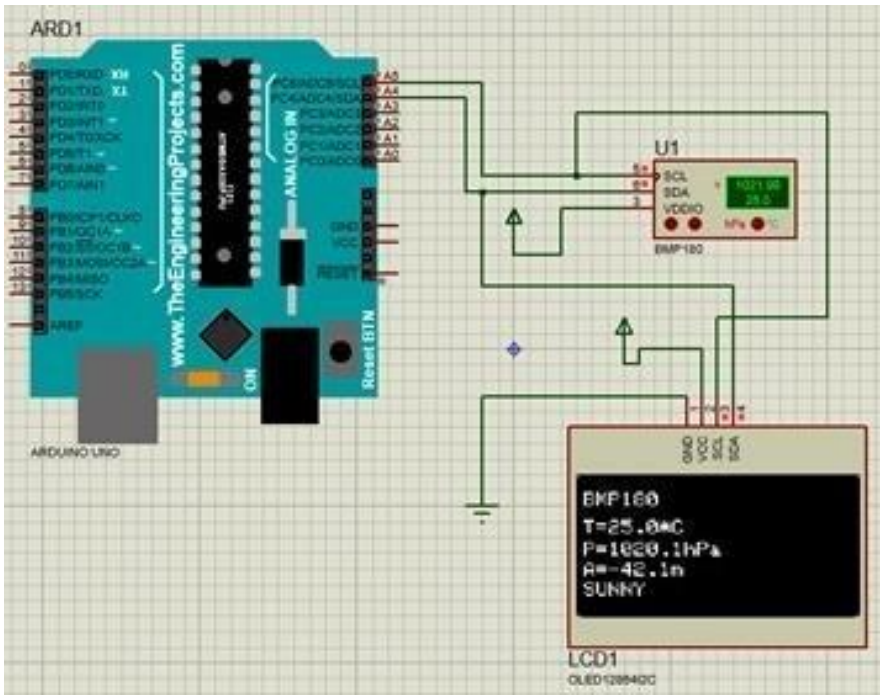
```
      slot = slot - 1;
     }
    } else {
     lcd.setCursor (0, 0);
     lcd.print(" Parking Full ");
     delay(1500);
    }
  }

  if (digitalRead (ir_back) == 0 && flag2 == 0) {
    flag2 = 1;
    if (flag1 == 0) {
     myservo.write(180);
     slot = slot + 1;
    }
  }

  if (flag1 == 1 && flag2 == 1) {
    delay (1000);
    myservo.write(90);
    flag1 = 0, flag2 = 0;
  }

  delay(1);
}

void Read_Sensor() {
  S1 = 0, S2 = 0, S3 = 0, S4 = 0;

  if (digitalRead(ir_car1) == 0) {
    S1 = 1;
  }
  if (digitalRead(ir_car2) == 0) {
    S2 = 1;
  }
  if (digitalRead(ir_car3) == 0) {
    S3 = 1;
  }
  if (digitalRead(ir_car4) == 0) {
    S4 = 1;
  }

}
```

Simulation:

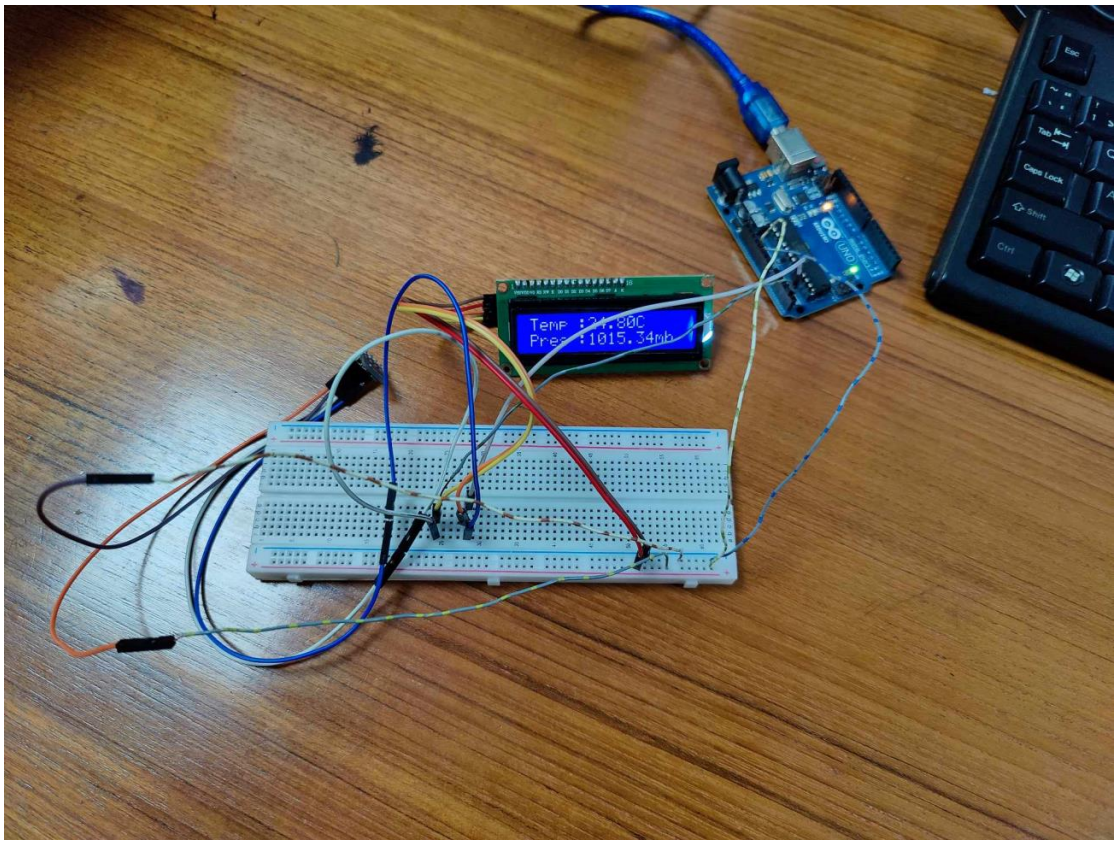
Here is the simulation implementation of a weather forecast system using the ADC modules of an Arduino and the necessary explanation of the implementation:

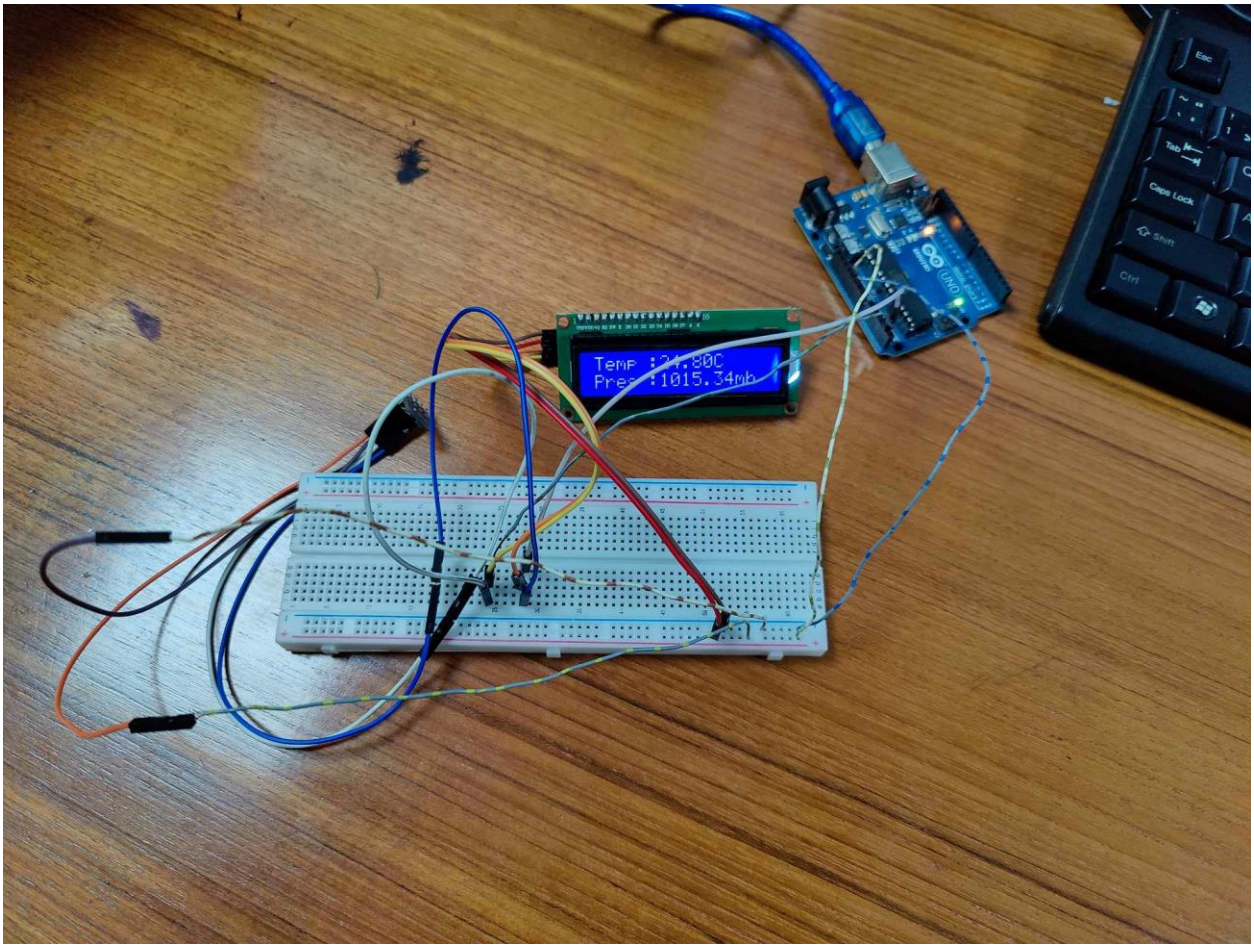






Hardware Result:





Discussion & Conclusion:

In this experiment, we have used the ADC modules of an Arduino microcontroller to create a weather forecast system. The system was developed to use the BMP180 sensors to measure various meteorological factors such as temperature, and air pressure and then convert them to digital information using I2C converter. These digital signals were then analyzed and shown as a weather forecast on an LCD panel. The experiment's results confirmed the system's ability to provide accurate weather forecasts.
This approach, however, has certain limitations. The procedure is dependent on the precision of the sensors used to measure meteorological factors. Furthermore, the algorithm only provides a general weather forecast and cannot predict extreme weather events like hurricanes, tornadoes, or thunderstorms.

In this lab, we learned about Arduino Uno board configuration which we used and got successful results using Arduino IDE. We also tested the logic of this lab by simulating accordingly. So, the objectives of this lab were achieved.

Reference(s):

[1] Arduino IDE, https://www.arduino.cc/en/Main/Software accessed on May 3, 2019.
[2] Arduino and Proteus Library, https://etechnophiles.com/add-simulate-ultrasonic-sensorproteus-2018-edition/ accessed on May 3, 2019.