AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Engineering

Lab Report

Experiment # 08

Title: Implementation of a weather forecast system using the ADC modules of an Arduino.

Date of Perform	6 April 2023	**Date of Submission**	9 April 2023
Course Title	MICROPROCESSOR AND EMBEDDED SYSTEMS		
Course Code	COE3104	**Section**	C
Semester	Spring 2022-23	**Degree Program**	BSc in CSE
Course Teacher	**RICHARD VICTOR BISWAS**		

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgment is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared, and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty Member for review by any internal/external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea, or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offense that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic and visual forms, including electronic data, and oral presentations. Plagiarism occurs when the origin of the source is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or copy my/our work.

* Student(s) must complete all details except the faculty use part.
** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group # 01

Sl No	Name	ID	Department
1	Zaid Amin **Rawfin**	20-42459-1	CSE
2	Abzana Sultan **Ira**	20-41973-1	CSE
3	MD. Sanjid Bin Karim **Sezan**	19-41702-3	CSE
4	Abdur Rahman **Swapnil**	18-38950-3	CSE

Faculty use only		
FACULTY COMMENTS	**Marks Obtained**	
	Total Marks	

Table of Contents

Experiment Title

Implementation of a weather forecast system using the ADC modules of an Arduino.

Abstract

This experiment aims to familiarize students with a microcontroller-based weather forecast system and measure environmental parameters such as temperature, pressure, and humidity. The BMP085/BMP180 or MPL115A absolute device is used to predict and measure barometric pressure to deduce weather patterns. The sensor should be kept in a relatively protected area from any strong airflows and kept at that static location during analysis. The pressure changes due to weather are slow, requiring a few hours to determine the sloping of the pressure change. Normalization takes local barometric pressure and shifts it to reflect sea level altitude, allowing meteorologists to see the weather pattern over a region. This experiment will help students gain an understanding of how weather prediction works and how weather patterns lead to high or low pressure.

Introduction

Weather forecasting is an essential aspect of our daily lives, and understanding how it works can help us make better decisions. With the advent of microcontrollers and sensors, it is now possible to develop low-cost weather stations that can measure environmental parameters such as temperature, pressure, and humidity. In this experiment, we aim to familiarize students with a microcontroller-based weather forecast system and how to measure these environmental parameters.

One of the critical aspects of weather prediction is measuring barometric pressure. Barometric pressure changes can directly correlate to changes in the weather. Low pressure is typically seen as the precursor to worsening weather, while high-pressure increases can be interpreted as improving or clearing weather. This experiment will use the BMP085/BMP180 or MPL115A absolute device to predict and measure barometric pressure to deduce weather patterns. The sensor should be kept in a relatively protected area from any strong airflows and kept at that static location during analysis

Theory and Methodology

Weather Prediction

The BMP085/BMP180 or MPL115A is a device that is capable of measuring and predicting barometric pressure for determining weather patterns. For an accurate analysis, the sensor should be kept in a fixed position for 2-3 hours as pressure changes related to weather are gradual. Any movement or strong airflow can affect the results. Furthermore, temperature changes can also impact the readings, especially during long-term measurements at different temperatures. However, BMP085/BMP180 has built-in temperature compensation and calibration, which makes it suitable for use over a broad temperature range of 0 to 85°C, without requiring auto-zeroing for temperature-induced changes in offset or span.

How Pressure Increases and Decreases with Weather

The BMP085/BMP180 or MPL115A devices are well-suited for predicting weather patterns due to their pressure range and resolution. Barometric pressure changes can directly correlate to changes in weather conditions. Low pressure is usually a precursor to worsening weather, while high-pressure increases indicate improving or clearing weather. This is because air with a higher concentration of water vapor is lighter than dry air, leading to variations in air pressure.

The formation of water-vapor clouds in an area during bad weather leads to falling pressure on a barometer, while high pressure signals the clearing of water vapor as the air dries. However, predicting weather patterns can be difficult in some regions such as cities at the base of mountainous regions or in areas like Hawaii, where high colder mountains meet low warm sea regions. Hurricanes and cyclones, despite having high winds of up to 150 mph, are actually low-pressure conditions surrounded by higher pressure. The rush of air from higher to lower-pressure regions create fast-moving winds, and the lower the pressure in the center, the greater the differential pressure between high and low areas, leading to a stronger cyclone or hurricane. A single sensor in a static location can be used to create a simple standalone weather station, although a network of sensors can provide more accurate results.

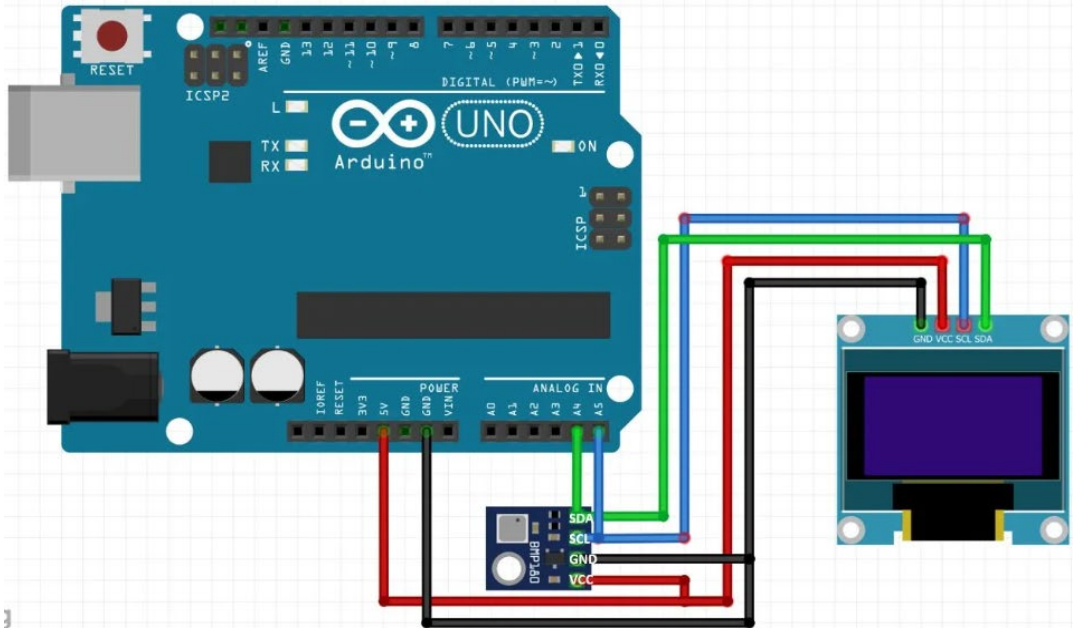
Local Weather Stations

When setting up a weather station, it is important to cross-reference the collected data with local forecasts. When looking up barometric pressure values, it is crucial to note that the weather reports are typically normalized to sea level altitude. This means that the local barometric pressure is adjusted to reflect the pressure at sea level, which allows meteorologists to map out weather patterns in a given area. Failing to normalize the data can lead to erroneous conclusions, as the altitude can greatly affect the reported pressure values. Airports are commonly used as reporting stations for barometric pressure, and some will only display normalized pressure values for the convenience of pilots who need to know the weather conditions for landing. Regardless of the location of the airport, normalization of the pressure values allows for standardized and accurate mapping of weather patterns.

Experimental Procedure

1. The microcontroller-based weather station was set up using the BMP085/BMP180 or MPL115A sensor.
2. The sensor was placed in a relatively protected area from any strong airflows and kept at that static location during analysis.
3. The environmental parameters such as temperature, pressure, and humidity were measured.
4. The barometric pressure readings were analyzed to deduce weather patterns.
5. The results were checked with a local forecast and the normalized barometric pressure values were compared to gain a better understanding of the weather pattern over a region.

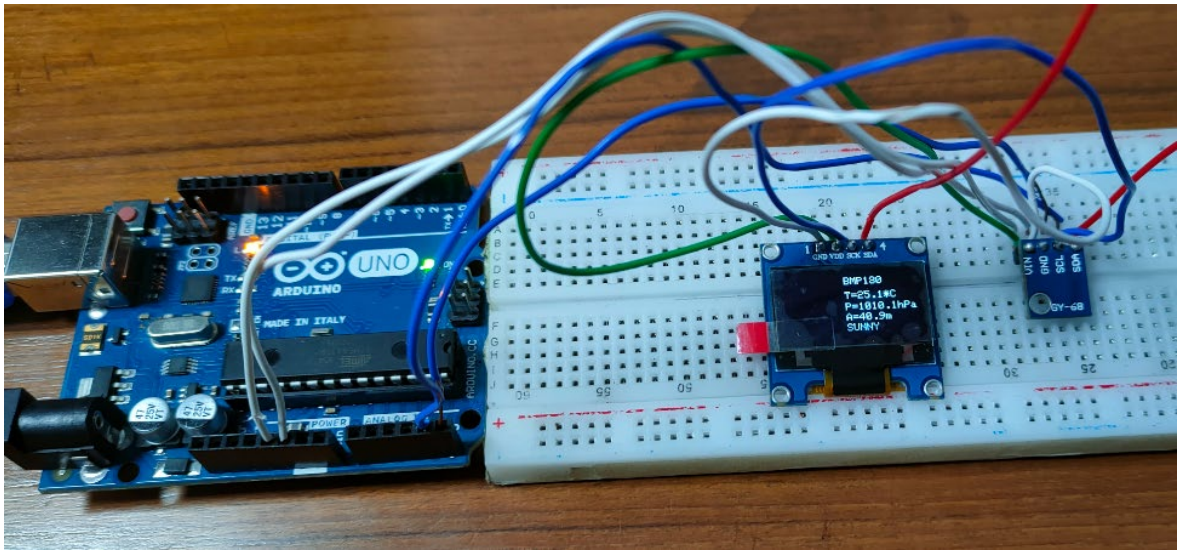
Circuit Diagram



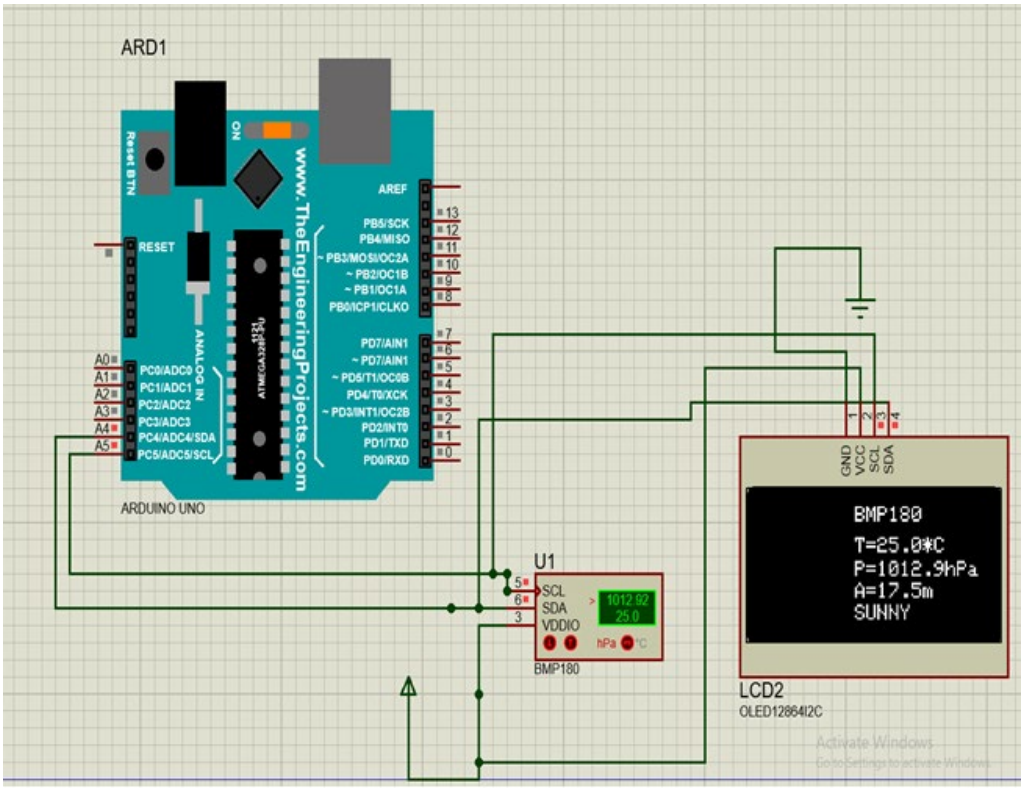
Apparatus

- Arduino Uno Board
- BMP085/BMP180 / MPL115A
- 96 inch OLED 128X64
- Breadboard and Jump Wires

Hardware Output Results



Simulation Output Results



Code/Program

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_BMP085.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT);
Adafruit_BMP085 bmp;

float simpleweatherdifference, currentpressure, predictedweather,
currentaltitude;

void setup() {
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  if (!bmp.begin()) {
    Serial.println("Could not find a valid BMP085 sensor, check wiring!");
    while (1) {}
  }
}

void loop() {
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
```

```
  display.setCursor(0, 5);
  display.print("BMP180");

  display.setCursor(60, 30);
  display.print("T=");
  display.print(bmp.readTemperature(), 1);
  display.println("*C");

  display.setCursor(60, 30);
  display.print("P=");
  display.print(bmp.readPressure()/100.0F, 1);
  display.println("hPa");

  display.setCursor(60, 30);
  display.print("A=");
  display.print(bmp.readAltitude(SEALEVELPRESSURE_HPA), 1);
  display.println("m");

  delay(6000);
  display.display();

  currentpressure = bmp.readPressure()/100.0;
  predictedweather = (101.3*exp(((float)(currentaltitude))/(-7900)));
  simpleweatherdifference = currentpressure - predictedweather;

  display.setCursor(0, 50);
  if (simpleweatherdifference > 0.25) {
    display.print("SUNNY");
  } else if (simpleweatherdifference <= 0.25) {
    display.print("SUNNY/CLOUDY");
  } else if (simpleweatherdifference < -0.25) {
    display.print("RAINY");
  }
  display.display();
  delay(2000);
}
```

Code Explanation

This is an Arduino sketch that uses a BMP180 sensor and an SSD1306 OLED display to measure temperature, pressure, and altitude and predict the weather based on the current altitude and pressure readings.

The sketch begins by including necessary libraries such as SPI, Wire, Adafruit_GFX, Adafruit_SSD1306, and Adafruit_BMP085. The SCREEN_WIDTH and SCREEN_HEIGHT macros are defined for the OLED display, and an instance of the Adafruit_SSD1306 and Adafruit_BMP085 classes are created.

In the setup() function, the OLED display is initialized and a check is performed to ensure that the BMP180 sensor is properly connected.

In the loop() function, the display is first cleared and text is printed on the OLED display to show the name of the sensor, current temperature, pressure, and altitude readings. A delay of 6000 milliseconds is added before the display is updated to allow time for the BMP180 sensor to take readings.

The current pressure reading is then assigned to the "currentpressure" variable, and a predicted weather value is calculated based on the current altitude reading and the constant "SEALEVELPRESSURE_HPA" which is set to 1013.25 hPa. This predicted weather value is assigned to the "predictedweather" variable.

A simple difference between the current pressure and predicted weather values is calculated and stored in the "simpleweatherdifference" variable. The code then checks this value to determine the predicted weather and prints it on the OLED display accordingly.

The display is then updated and a delay of 2000 milliseconds is added before the loop repeats.

Overall, the sketch uses the BMP180 sensor to measure atmospheric conditions, performs a simple calculation to predict weather based on the current readings, and displays the predicted weather on the OLED display. The sketch also demonstrates the use of various functions, variables, and libraries available in Arduino.

Discussion & Conclusion

Programming has become an essential skill in the modern world, with numerous applications in various fields, from education to healthcare, finance, and technology. It allows for the creation of sophisticated software systems and automates routine tasks, increasing efficiency and productivity. Moreover, learning to code offers an opportunity for personal growth, as it encourages problem-solving, logical thinking, and creativity. While it can be challenging to master, the benefits of programming are undeniable, and with the right resources and mindset, anyone can learn to code.

As the field of technology continues to evolve and shape our lives, programming will only become more critical. As such, individuals who can code are well-positioned to succeed in the workforce and contribute to society. By embracing the power of programming, we can drive innovation and solve complex problems, ultimately improving our world. Therefore, it is vital to encourage and support the development of programming skills in individuals of all ages, backgrounds, and interests, ensuring that we can all participate in the technological advancements of the future.

References

- https://www.arduino.cc
- AIUB Lab Manual