# American International University- Bangladesh

## Department of Electrical and Electronic Engineering

EEE4103: Microprocessor and Embedded Systems Laboratory

**Title:** Building a weather forecast system using pressure and heat index measurement.

**Introduction:**

The objective of this experiment is to get familiarized with Micro-controller based weather forecast system and heat index measurement.

**Theory and Methodology:**

### Weather Prediction

The MPL115A is an absolute device that can be used to predict and measure the barometric pressure to deduce weather patterns. Weather prediction requires a static location for the sensor and 2-3 hours to analyze a full weather pattern. Typically the pressure changes due to weather are slow, requiring a few hours to determine the sloping of the pressure change. Vertical movement or a significant airflow can interfere with results due to only weather patterns in barometric pressure. The sensor should be kept in a relatively protected area from any strong air flows, and kept at that static location during analysis. Temperature effects can change the results of a normal pressure sensor especially if the measurement is done over several hours in varying temperature. Due to the nature of the calibration and temperature compensation, MPL115A meets these requirements, compensating for temperature swings over a large 0 to 85°C operating range. It will not require auto-zeroing for shifts in offset or span over temperature.

### How Pressure Increases and Decreases with Weather

For weather pattern prediction, the MPL115A is a well suited device with its pressure range and resolution. Barometric pressure changes can directly correlate to changes in the weather. Low pressure is typically seen as the precursor to worsening weather. High pressure increases can be interpreted as improving or clear weather. The typical reasoning can be seen in a comparison of molecular weights. If air is approximately 21% $O_2(g)$, 78% $N_2(g)$, $O_2(g)$ has a molecular mass of 32, $N_2(g)$ has a mass of 28. $H_2O$ (g) has a molecular mass of 18. So if there is a large amount of water vapor present in air, this air is going to be lighter than just regular dry air by itself. It's an interesting fact that explains how weather patterns lead to high or low pressure.

If bad weather originates in an area in the formation of water-vapor clouds, this is falling pressure on a barometer. The vapor will reduce the barometric pressure as the $H_2O$ reduces the mass above that point on the earth. High pressure will signal the clearing of the water vapor as the air dries.

Another quandary is how weather during severe hurricanes/cyclones with high 150 mph winds be defined as low pressure? This is due to the fact that hurricanes are low pressure conditions surrounded by higher pressure. The rush of air from higher to low pressure creates the fast moving winds. The lower the pressure in the center, the greater differential pressure between high and low areas. This leads to a stronger cyclone or hurricane.

Some areas are harder to predict weather patterns. Cities located at the base of mountainous regions where condensation and fog are a daily occurrence is an example. An area like Hawaii where high colder mountains meet low warm sea regions can have harder to predict results. A network of sensors can give a more exact trend, but for a single sensor in a static location, there are a few ways to have a simple standalone weather station.

## Local Weather Stations

When implementing a weather station, it is best to will check results with a local forecast. When researching local weather pressures, such as barometric pressure at the closest airport, remember that the weather is normalized for altitude. Normalization takes local barometric pressure and shifts it to reflect sea level altitude. Sea Level is 101.3 kPa, and by normalizing various points on a map, a meteorologist can see the weather pattern over a region. Without the normalization, the effect of altitude on the pressure reported by collection points will lead to useless data. A mountain data point will have pressure affected by altitude and as it leads to the valleys, the pressure point there will be higher, telling nothing about the weather without the normalization.

Airports are typical reporting stations to check barometric pressure. Some display only normalized pressure during a web search. This is such that a pilot landing at any airport can deduce the weather conditions by knowing the barometric pressure. If the airport is located at the beach, or on a mountainous region, normalization of this value removes the barometric variation due to the altitude. It standardizes pressure so that weather patterns can be mapped.

### Example:

An airport located at 600 m elevation would have pressure of 93.97 kPa according to our pressure to altitude equation. If the weather was sunny and mostly clear, it would most probably have a published pressure of 101.3 kPa for weather conditions. It may not be extremely clear skies as this would be a high pressure weather system. It would be a stable pressure with neither extreme in low or high pressure.

Remember to discern this information when trying to see if the MPL115A value matches the local weather barometric pressure. Sometimes a disparity in the value occurs due to normalization.

### Algorithms for weather Simple Approach

How is weather predicted using the barometric sensor? There is a simple approach looking at increasing or decreasing pressure. Simply an increase over time is a trend that approaches "sunny" or "clear" days. Dropping pressure can signal a worsening "cloudy" or "rainy" day. This can be seen typically as a rising or falling bar on many simple solution weather stations. It can be interpreted as an

increase/decrease gradient for the user to interpret, but the time interval is not used extensively to reach weather predictions. The user can look at the results for a 12 hour time frame to predict the weather trend.

This table is typically used:

| Analysis | Output |
|---|---|
| dP > +0.25 kPa | Sun Symbol |
| -0.25 kPa < dP < 0.25 kPa | Sun/Cloud Symbol |
| dP < -0.25 kPa | Rain Symbol |

Another approach that is more direct and quicker in calculating the weather in the simple approach is to know the current altitude. This cuts the need to wait and see a "trend".

By using the equation below:

$$ph = p0 \cdot e^{\frac{-h}{7990m}}$$

Where p0 = 101.3 kPa, and h is the current altitude, the pressure for the local barometric can be calculated. This is the pressure for good sunny weather at current altitude location.

By using the pressure equation and knowing the normalized good weather pressure for the current location (best for a static weather station), the weather can be deduced by the difference. As in the table for the weather symbols, the ideal pressure is compared to the value from the MPL115A and the appropriate symbol of Sun/Cloud/Rain is selected.

Below simple C code from the DEMOAPEXSENSOR demo kit calculates which weather symbol to display on the LCD screen.

• CurrentAltitude - (m) Altitude in meters that is entered into the system by the user for that current static location.

• Pweather - (kPa) Pressure at the current altitude. It is calculated using the Height (m) to Pressure (kPa) exponential equation, inputting CurrentAltitude in meters. This is the ideal pressure for the current location on a stable relatively sunny day.

• decPcomp - (kPa) Value of compensated pressure from MPL115A.

**Simple Weather Station Code**

```
/////////////////////
   //SIMPLE WEATHER SECTION
   /////////////////////

   Pweather = (101.3 * exp(((float)(CurrentAltitude))/(-7900)));

   Simpleweatherdiff = decPcomp-Pweather;
   if (Simpleweatherdiff >0.25)
     Simpleweatherstatus = 0; //Sun Symbol
   if ((Simpleweatherdiff <=0.25) || (Simpleweatherdiff >=(-0.25)))
     Simpleweatherstatus = 1; //Sun/Cloud Symbol
   if (Simpleweatherdiff <(-0.25))
   Simpleweatherstatus = 2; //Rain Symbol
```

Lets look at some data:

| decPcomp (kPa) (MPL115A Pressure) | PWeather (kPa) (Ideal weather) | Simpleweatherdiff (kPa) | Weather Type |
|---|---|---|---|
| 96.6 | 96.85 | -0.25 | Sun/Cloud |
| 96.4 | 96.85 | -0.45 | Rain |
| 97.4 | 96.85 | 0.55 | Sun |
| 96.92 | 96.85 | 0.07 | Sun/Cloud |

For example: Altitude for Tempe, AZ = 359 m, thus PWeather = 101.3 x $e^{-359/7990m}$ = 96.85 kPa.

Observing Pressure over time will yield similar results over a 12 hour period. In this case, the changes in the pressure that take place over extended time will be enough for the simple method to figure out the weather patterns. This negates the need for user input of the approximate location/altitude. The weather algorithm will be more accurate at the end of the 12 hour interval as the trend is visible versus at initialization.

Knowing the altitude can also be useful for a dynamically changing location to predict simple weather. Take for example a GPS unit: a GPS unit can give an approximate altitude measurement. Measuring the difference from the MPL115A pressure sensor and calculated pressure from the GPS altitude, gives a close approximation of weather patterns quickly at that dynamically changing point. Weather approximation can be deduced in the symbol style as above.

**Advanced Version of Weather Station**

A more complex approach is to measure the P/t and see how the gradient is changing over time. As in the simple approach, this does need to be kept in a static location during measurement. Essentially as time progresses, the weather can be broken into more exact categories than the simple approach of basic symbols.

This can also use less time than waiting for a full 12 hours to see the pattern of pressure change. In Table 3, the ranges of pressure change over time leading to the definition of the

weather patterns is shown. It is a change in the pressure per one hour. 2-3 hours are needed to deduce how the pressure is migrating.

**Table 3. Advanced Weather Determination**

| Analysis | Output |
|---|---|
| dP/dt > 0.25 kPa/h | Quickly rising High Pressure System, not stable |
| 0.05 kPa/h < dP/dt < 0.25 kPa/h | Slowly rising High Pressure System, stable good |
| -0.05 kPa/h < dP/dt < 0.05 kPa/h | Stable weather condition |
| -0.25 kPa/h < dP/dt < -0.05 kPa/h | Slowly falling Low Pressure System, stable rainy |
| dP/dt < -0.25 kPa/h | Quickly falling Low Pressure, Thunderstorm, not |

In the provided flowchart, the pressure is sampled every minute for 3 hours/180 minutes into a data array. The first 5 minutes are averaged, followed by 5 minutes near the first ½ hour point. Consecutive ½ hour marks have 5 minute averaged data-points stored. This leads to 7 averaged results over the 180 minutes depicting the pressure every ½ hour. Once the data- points are collected, the patterns are deduced. A flowchart provides the method used in deducing the weather pattern. The initial starting point is the reference from which every ½ hour data point is compared to. As the pressure falls, the value is compared and divided so that the change in pressure per 1 hour is compared every half an hour.
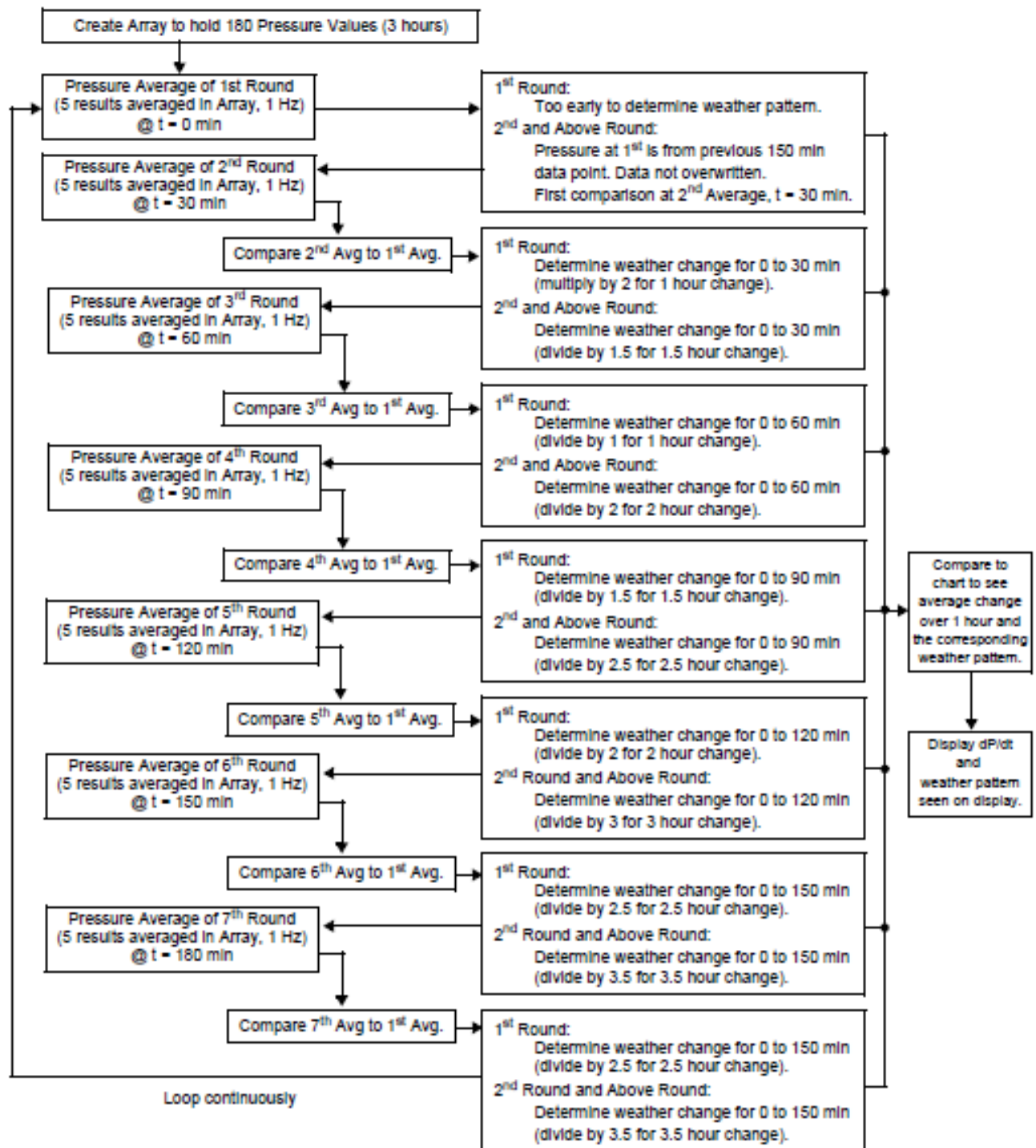
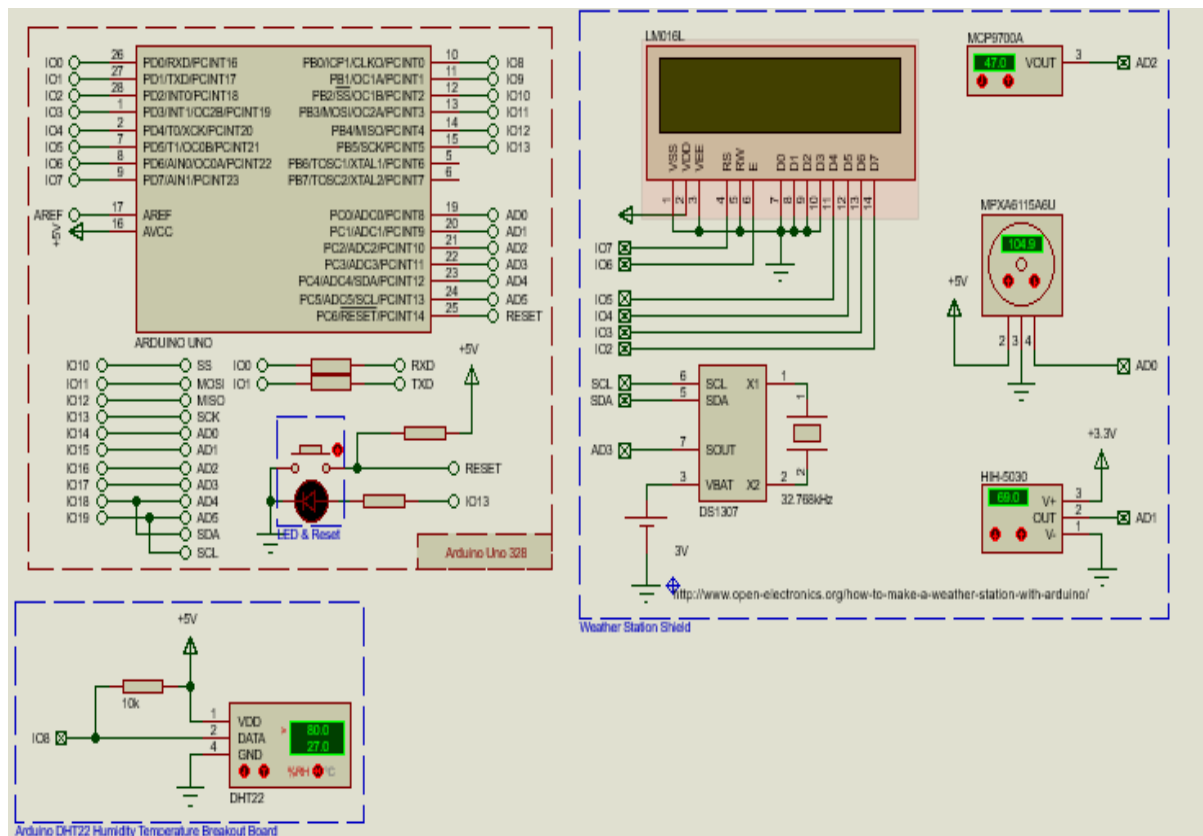**Figure 1. Advanced Weather Flowchart**
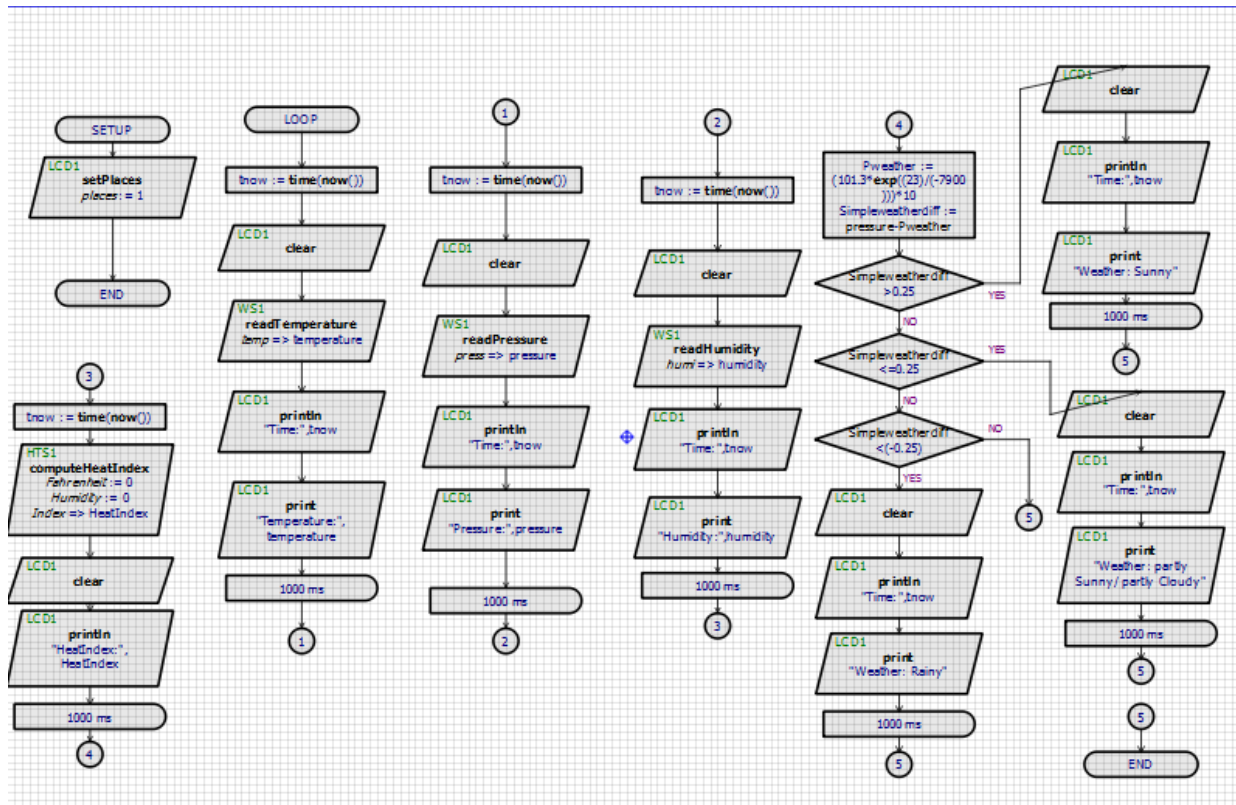
## Components List

- Arduino Board
- DHT22
- MPXA6115A6U / MPL115A
- MCP9700A

- HIH-5030
- LM016L
- Breadboard and Jump Wires

**Experimental Procedure:**

Start by creating a new Visual Designer Project and then Add the weather station Shield and DHT22 sensor.

## Flowchart for Proteus Visual Designer:



As always, you can program the physical board at any time via the upload button.

## Simulation and Measurement:

- Simulate the above system in proteus.
- Modify the design to display the average humidity measured from DHT22 and HIH-5030 sensor.

## Questions for report writing:

1) Include all codes and scripts into lab report following the writing template mentioned in appendix A of Laboratory Sheet Experiment 1.