# Weather Monitoring System

By Anubhav Mishra (RA1911029010023)
and Taha Baba (RA1911029010029)

# Description

Climatic change and environmental monitoring have received much attention recently. Man wants to stay updated about the latest weather conditions of any place like a college campus or any other particular building. Since the world is changing so fast so there should be the weather stations. Here in this presentation we present a weather station that is very helpful for any places. This weather station is based on IOT (internet of things). It is equipped with environmental sensors used for measurements at any particular place and report them in real time on cloud. To accomplish this we used Arduino Uno and different environmental sensors like DHT11, soil moisture sensor and rain drop sensor .The sensors constantly sense the weather parameters and keeps on transmitting it to the online web server over a wifi connection. The weather parameters are uploaded on the cloud and then provides the live reporting of weather information. It also focuses on the IOT application in the new generation of environmental information and provides a new paradigm for environmental monitoring in future.

# Hardware Requirements

The Hardware Requirement For Smart Weather Monitoring Systems Are:
1.Microcontroller ( E.G:ArduinoUno, Raspberry Pi)
2.DHT11 Digital Temperature/Humidity Sensor
3.Rain Drop Sensors
4.Wi-Fi Module (E.G: ESP8266)
5.LCD Display, LEDs
6.Resistors, Capacitors, Transistors And Diodes
7. Switches, Cables And Connectors
8.A Crystal Oscillator To Create An Electrical Signal With A Constant Frequency.
9.Breadboards

# STEPS TO RUN IT

Since IoT Based Weather Monitoring Systems Work Mostly On Sensors, Hence It Also Needs A Step-Down Transformer Which Will Bring Down The 230V AC Supply To 6V AC Supply.The Electronic Components Also Work Only On DC Supply Hence A Rectifier Will Convert The AC Supply Into DC Supply. Microcontroller Will Be Controlling Each And Every Sensor Connect To It. LCD Display, Rain Drop Sensor, Humidity Sensor And Temperature Sensor All Will Be Interfaced Will This Microcontroller And The Programming Will Be In C Language If Microcontroller Is An Arduino Or NodeMCU Or The Language Will Be Python In Case Of Microcontroller As Raspberry Pi.

In Case Of Temperature And Humidity Sensors, Whenever The Temperature Will Increase, Voltage Associated With The Temperature Sensor Will Also Increase And Whenever Humidity Will Increase Then The Current Associated With The Humidity Sensor Will Increase. This Increase In Voltage And Current Will Be Measure By Microcontroller.
This Data Will Be Send Via Wi-Fi Module And Can Be View On An IoT Webpage As Well. Hence This Is A Simple Working Of A Weather Reporting System Using IoT.
From This Blog We Can Conclude That The Weather Reporting System Using IoT Is Quite Cost Effective, Efficient And The Data Collected Can Also Be Post On Cloud As Well. Hence ,This System Is Useful In Continuous Monitoring Of The Environment.

# SIMULATION RESULTS

After sensing the data from different sensor devices, which are placed in particular area of interest. The sensed data will be automatically sent to the web server, when a proper connection is established with sever device. The web server page which will allow us to monitor and control the system. The web page gives the information about the temperature, humidity and the CO level variations in that particular region, where the embedded monitoring system is placed. The sensed data will be stored in cloud (Google Spread Sheets). The data stored in cloud can be used for the analysis of the parameter and continuous monitoring purpose. The temperature and humidity levels and CO levels in air at regular time intervals. All the above information will be stored in the cloud, so that we can provide trending of temperature and humidity levels and CO levels in a particular area at any point of time.

# Data Collection Devices

Meteorologists use all kinds of equipment to measure the weather. These include thermometers, radar systems, barometers, rain gauges, wind vanes, anemometers, transmissometers, and hygrometers. Let's go through them one at a time and talk about what they do. Thermometers are probably the most well-known weather data device. They measure the temperature by allowing a liquid inside the thermometer to expand as it gets hotter and contract as it gets cooler. There are also digital thermometers in wide use.Radar systems are used to create maps of rain and snow and measure the motion of rain clouds. This works by bouncing radio waves off the clouds and measuring how long it takes for them to return.Barometers are devices that measure the pressure in the atmosphere. Pressure is how thick the air is: how much air can be found in a particular volume. For example, the higher up a mountain you go, the less air there is. These devices can be quite simple and generally involve a liquid which gets pushed up or down an inverted tube depending on how strong the atmospheric pressure is. Rain gauges are also very simple, and measure how much rain falls into a container. Wind vanes are simple sails that get pushed by the wind to show the wind speed. These work alongside anemometers, which measure wind speed by catching the wind in a cup-shaped container causing a dial to turn.Transmissometers measure visibility by shining a laser through the air and detecting how much light is lost. For example, if there is thick fog the visibility will be particularly low.Last of all, hygrometers measure the humidity (how much water the air contains) at a particular location. They work in lots of different ways, but one way is by measuring how easily the air conducts electricity, since water conducts electricity better than air itself.

# SENSORS AND ACTUATORS

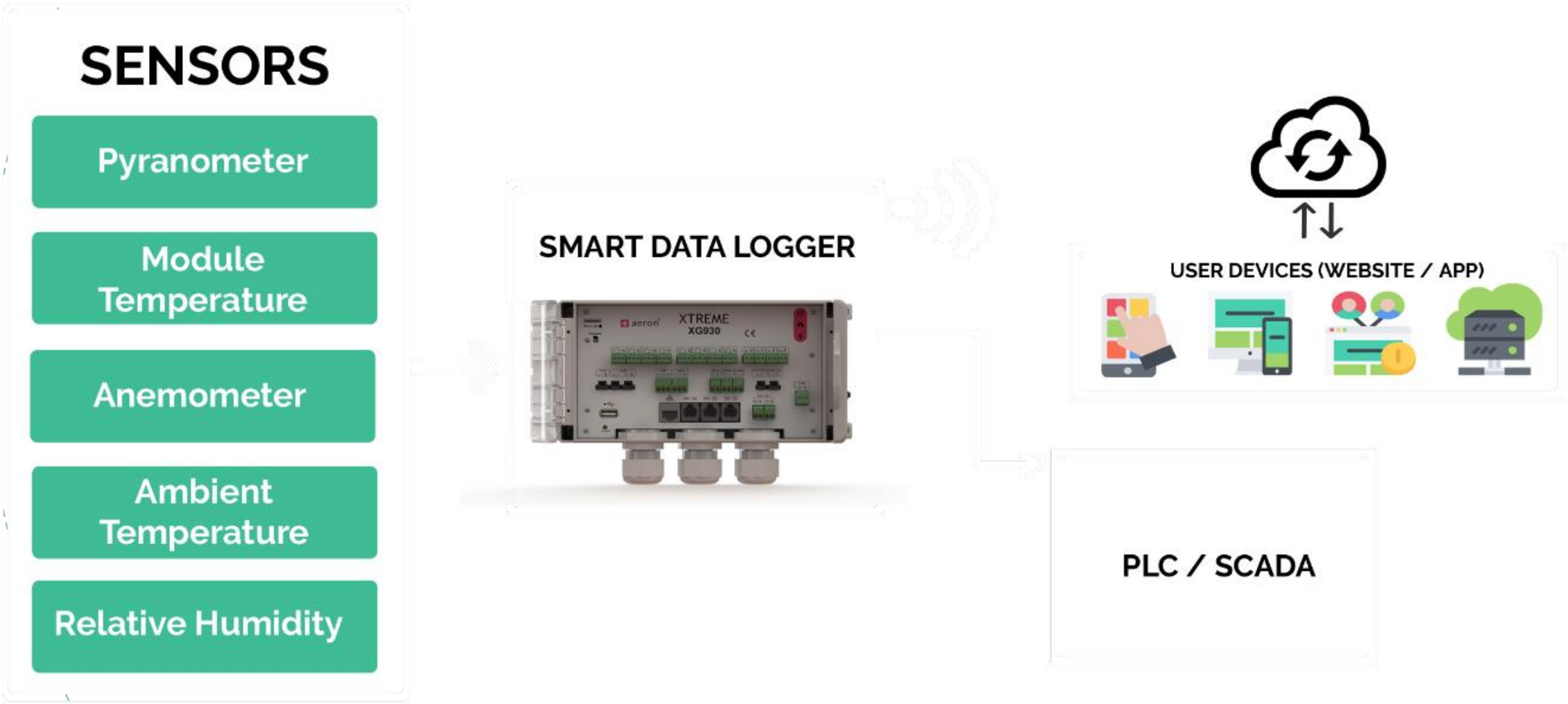
**Temperature & Humidity Sensor:** This DHT11 Temperature and Humidity Sensor features digital signal output .It is integrated with a high-performance 8-bit microcontroller. Its technology ensures the high reliability and excellent long-term stability. It has excellent quality, fast response, anti-interference ability and high performance. Each DHT11 sensors features extremely accurate calibration of humidity calibration chamber. The calibration coefficients stored in the OTP program memory, internal sensors detect signals in the process, we should call these calibration coefficients. The single-wire serial interface system is integrated to become quick and easy. Small size, low power, signal transmission distance up to 20 meters, enabling a variety of applications and even the most demanding ones. The product is 4-pin single row pin package. Convenient connection, special packages can be provided according to users need.

**Soil Moisture Sensor:** Soil moisture sensors measure the contents in soil. Since the direct gravimetric measurement of free soil moisture requires removing, drying, and weighting of a sample, soil moisture sensors measure the volumetric water content indirectly by using some other property of the soil, such as electrical resistance, dielectric constant, or interaction with neutrons, as a proxy for the moisture content. The relation between the measured property and soil moisture must be calibrated and may vary depending on environmental factors such as soil type, temperature, or electric conductivity. Reflected microwave radiation is affected by the soil moisture and is used for remote sensing in hydrology and agriculture. Portable probe instruments can be used by farmers or gardeners.

**Arduino Uno:** Arduino Uno is a microcontroller board developed by Arduino. Which is an open-source electronics platform mainly based on AVR microcontroller Atmega328. The current version of Arduino Uno comes with USB interface, 6 analog input pins, 14 I/O digital ports that are used to connect with external electronic circuits. Out of 14 I/O ports, 6 pins can be used for PWM output. It allows the designers to control and sense the external electronic devices in the real world. This board comes with all the features required to run the controller and can be directly connected to the computer through USB cable that is used to transfer the code to the controller using IDE software, mainly developed to program Arduino. Programming languages like C and C++ are used in IDE. Apart from USB, battery or AC to DC adopter can also be used to power the board. Arduino Uno are the most official versions that come with Atmega328 8-bit AVR Atmel microcontroller where RAM memory is 32KB.
**WiFi Module:** The Arduino Uno WiFi is an Arduino Uno with an integrated WiFi module. The board is based on the ATmega328P with an ESP8266 WiFi Module integrated. The ESP8266 WiFi Module is a self contained SoC with integrated TCP/IP protocol stack that can give access to your WiFi network (or the device can act as an access point). One useful feature of Uno WiFi is support for OTA (over-the-air) programming, either for transfer of Arduino sketches or WiFi firmware.

**Rain level sensor:** The rain sensor module is an easy tool for rain detection. It can be used as a switch when raindrop falls through the raining board and also for measuring rainfall intensity. The analog output is used in detection of drops in the amount of rainfall. Connected to 5V power supply, the LED will turn on when induction board has no rain drop, and DO output is high. When dropping a little amount water, DO output is low, the switch indicator will turn on. Brush off the water droplets, and where restored to the initial state, outputs high level. A rain sensor or rain switch is a switching device activated by rainfall.

**LCD Display:** A Liquid Crystal Display commonly abbreviated as LCD is basically a display unit built using Liquid Crystal technology. When we build real life/real world electronics based projects, we need a medium/device to display output values and messages. The most basic form of electronic display available is 7 Segment displays – which has its own limitations. The next best available option is Liquid Crystal Displays which comes in different size specifications. Out of all available LCD modules in market, the most commonly used one is 16×2 LCD Module which can display 32 ASCII characters in 2 lines. To establish a good communication between human world and machine world, display units play an important role. And so they are an important part of embedded systems. Display units – big or small, work on the same basic principle


SENSORS
Pyranometer
Module Temperature
Anemometer
Ambient Temperature
Relative Humidity
SMART DATA LOGGER
XTREME
XG930
USER DEVICES (WEBSITE / APP)
PLC / SCADA

## Barometric Pressure

The barometric sensor is used to measure the atmospheric pressure at the location of the solar and wind energy plan

## Wind Speed and Wind Direction

The wind data is very critical for the wind farms because the energy output is dependent on the wind speed. The wind at the solar project site affects the temperature of the solar modules which in turn affects the energy output. Therefore wind data is useful for solar as well as wind energy plants.

## Rain Gauge

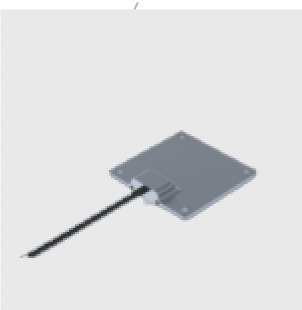
The rain gauge measures the rainfall at the project site. The rainfall data helps in better estimation of the annual energy generation. This data is also sometimes used for the insurance of the power plant.

## Air Temperature & Humidity

The ambient temperature and humidity affect the heating of the solar panel and thus affects the energy generation. In wind farms, the heating of the wind turbines is dependent on the ambient conditions. Hence this data is important for renewable energy plants.

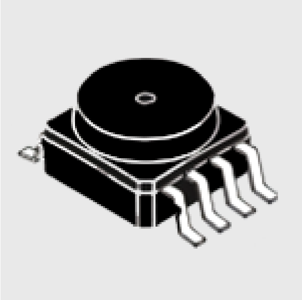
## Pyranometer

The pyranometer measures solar radiation available at the project site. The irradiance measured by the pyranometer i
accurate measurement.



## Module Temperature

The energy output from a solar panel depends on its temperature. Therefore, the measurement of module temperatu
temperature is measured at multiple locations in the plants, typically at every block level or more.

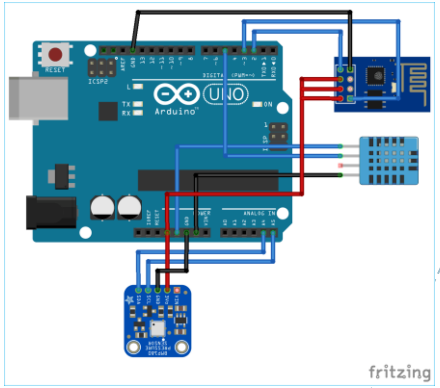


## Cloud Cover Sensor

The cloud cover sensor measures the cloud height and cloud base. The presence of the cloud at a solar project site adversely affects the energy generation. The cloud cover data helps in the better estimation of energy generation. There are different types of cloud cover sensors available for use in solar power plants.

## Data Logger Options

STORAGE MEMORY
8 GB flash memory (expandable up to 32 GB)
ANALOG INPUTS
16 single ended or 8 differential channels,
0-10V, 0-100mV, 0-20mA input range
DISPLAY
Graphic LCD display with 128x64 resolution
ADC RESOLUTION
24 bit ADC resolution for analog inputs, highest resolution in this class of products
CONNECTIVITY
Wireless connectivity over GSM/GPRS (2G) and wired connectivity over RS485 MODBUS, RS232 Serial, and USB as mass storage
TIME SYNCHRONIZATION
Time synchronization through GPS, internet, and RTC with backup battery
POWER
Built-in Li-ion batteries for independent operation. External power supply of 9-28 VDC
The complete circuit for **Arduino based IoT Weather Station** is shown below.



Code

```
#include <WiFi.h>
#include <DHT.h>
#include <Wire.h>
#include <SoftwareSerial.h>
#include <stdlib.h>
#include <SFE_BMP180.h>
SFE_BMP180 pressure;
#define DHTPIN 5
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
#define TEMPTYPE 0
#define ALTITUDE 160 // Altitude from Bussero (MI) Italy
#define ssid  "Enter Your WiFi Name Here "     // "WiFi Name"
#define pass "WiFi Password"       // "Password"
#define server = "api.thingspeak.com";
String apiKey ="Enter the API Key";
char buffer[10];
char t_buffer[10];
char h_buffer[10];
char P_buffer[10];
```

```
SoftwareSerial ser(2, 3); // RX, TX
void setup() {
  Wire.begin();
  pressure.begin();
  // enable debug serial

  Serial.begin(9600);
  Serial.println("AT");
  delay(5000);
  if(Serial.find("OK")){
    connectWiFi();

}
void loop()
{
  Trsmission(); // ESP8266
  delay(60000); // 60 seconds
    }


void Trsmission()
{
  int8_t h = dht.readHumidity();
  int16_t t = dht.readTemperature(TEMPTYPE);
  char status;
  double T,P,p0,a;
  status = pressure.startTemperature();
  if (status != 0)
```

```
 {
   delay(status);
   status = pressure.getTemperature(T);
   if (status != 0)
   {

     status = pressure.startPressure(3);
     if (status != 0)
     {
       // Wait for the measurement to complete:
       delay(status);

       status = pressure.getPressure(P,T);
       if (status != 0)
       {

         p0 = pressure.sealevel(P,ALTITUDE); // we're at 1655
meters (Boulder, CO)
         a = pressure.altitude(P,p0);
       }
       else Serial.println("error retrieving pressure
measurement\n");
     }
     else Serial.println("error starting pressure
measurement\n");
   }
   else Serial.println("error retrieving temperature
measurement\n");
 }
```

```
float temp = t;
  float humidity = h;
  float Pression = p0;

  String strTemp = dtostrf(temp, 4, 1, t_buffer);
  String strHumid = dtostrf(humidity, 4, 1, h_buffer);
  String strPres = dtostrf(Pression, 4, 2, P_buffer);

  Serial.print("Temperature: ");
  Serial.println(strTemp);
  Serial.print("Humidity: ");
  Serial.println(strHumid);
  Serial.print("Pression: ");
  Serial.println(strPres);

  String cmd = "AT+CIPSTART=\"TCP\",\"";
  cmd += "184.106.153.149"; // api.thingspeak.com
  cmd += "\",80";
  ser.println(cmd);
  if(ser.find("Error")){
    Serial.println("AT+CIPSTART error");
    return;
  }
```

```
if(ser.find("Error")){
  Serial.println("AT+CIPSTART error");
  return;
}

// prepare GET string
String getStr = "GET /update?api_key=";
getStr += apiKey;
getStr +="&field1=";
getStr += String(strTemp);
getStr +="&field2=";
getStr += String(strHumid);
getStr +="&field3=";
getStr += String(strPres);
getStr += "\r\n\r\n";

// send data length
 cmd = "AT+CIPSEND=";
cmd += String(getStr.length());
ser.println(cmd);
//ser.print(getStr);
if(ser.find(">")){
  ser.print(getStr);
}
```

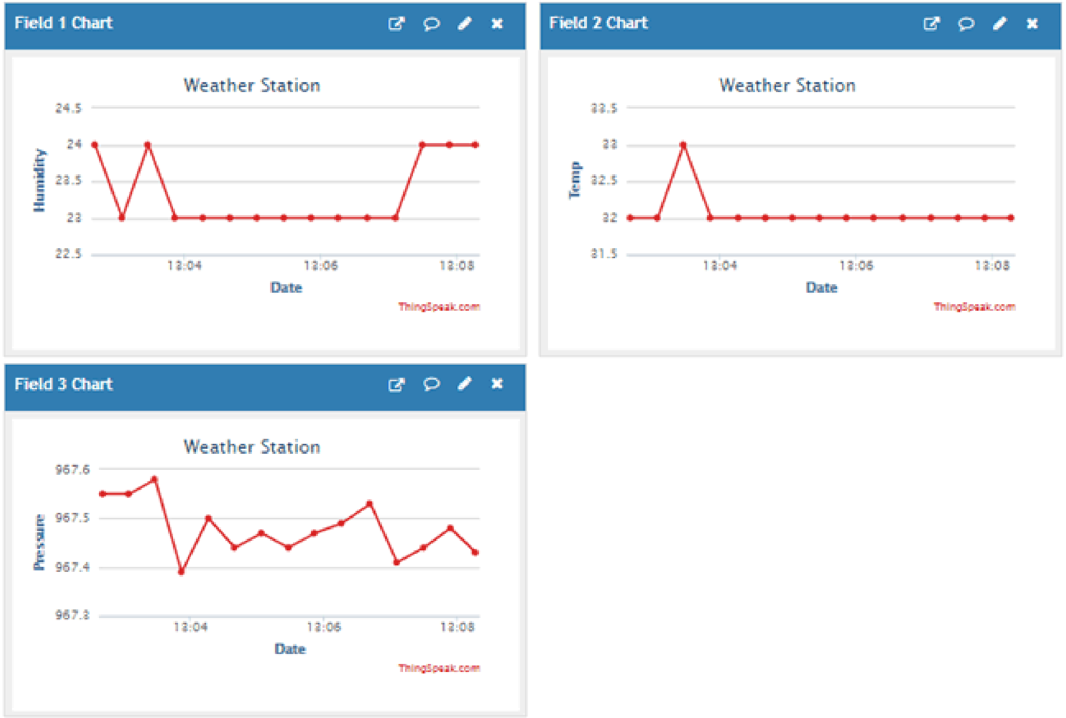
```
 else{
   ser.println("AT+CIPCLOSE");
   // alert user
   Serial.println("AT+CIPCLOSE");
   ser.println("AT+RST");
 }

 char buffer[10] = "";
}
```

### Running the IoT Arduino Weather Station

Now connect the Arduino with the laptop and choose the board and port correctly and then click the *Upload* button. After uploading the code, open the serial monitor. Make the baud rate of the serial monitor **9600**. You will see your Wi-Fi Id, password, and temperature, humidity, and pressure values on the serial monitor.

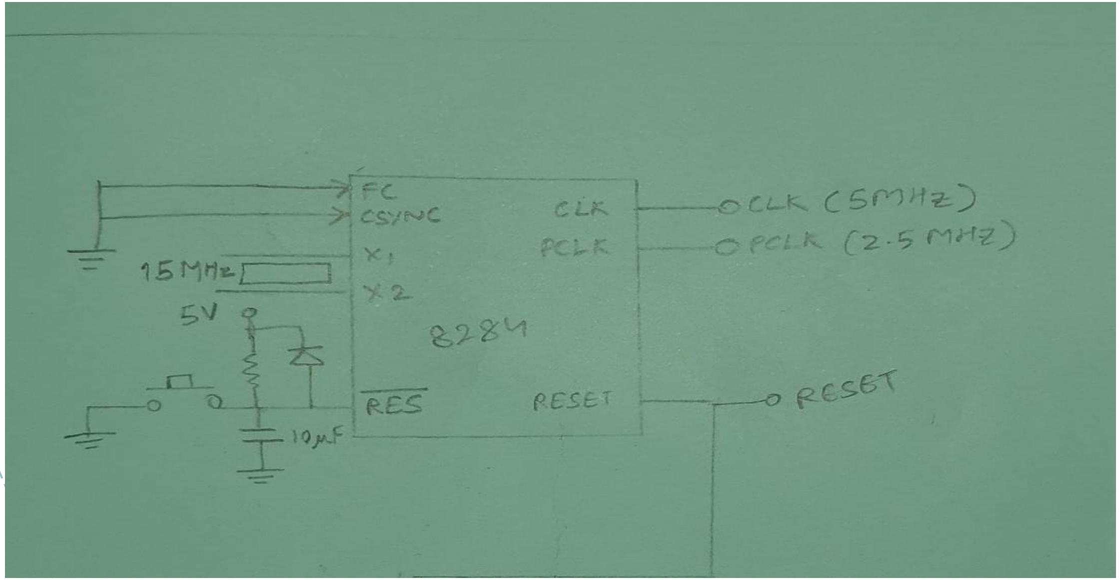
```
AT
AT+CWMODE=1
AT+CWJAP="CircuitLoop","circuitdigest101"
AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=56
AT+CIPCLOSE
AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=56
GET /update?key=9B6ILVOYMUSVOADA&field1=19.7&field2=50
```

Now navigate to the ThingSpeak channel and check your channel, you will see the temperature, humidity, and pressure values as shown in the below graphs.

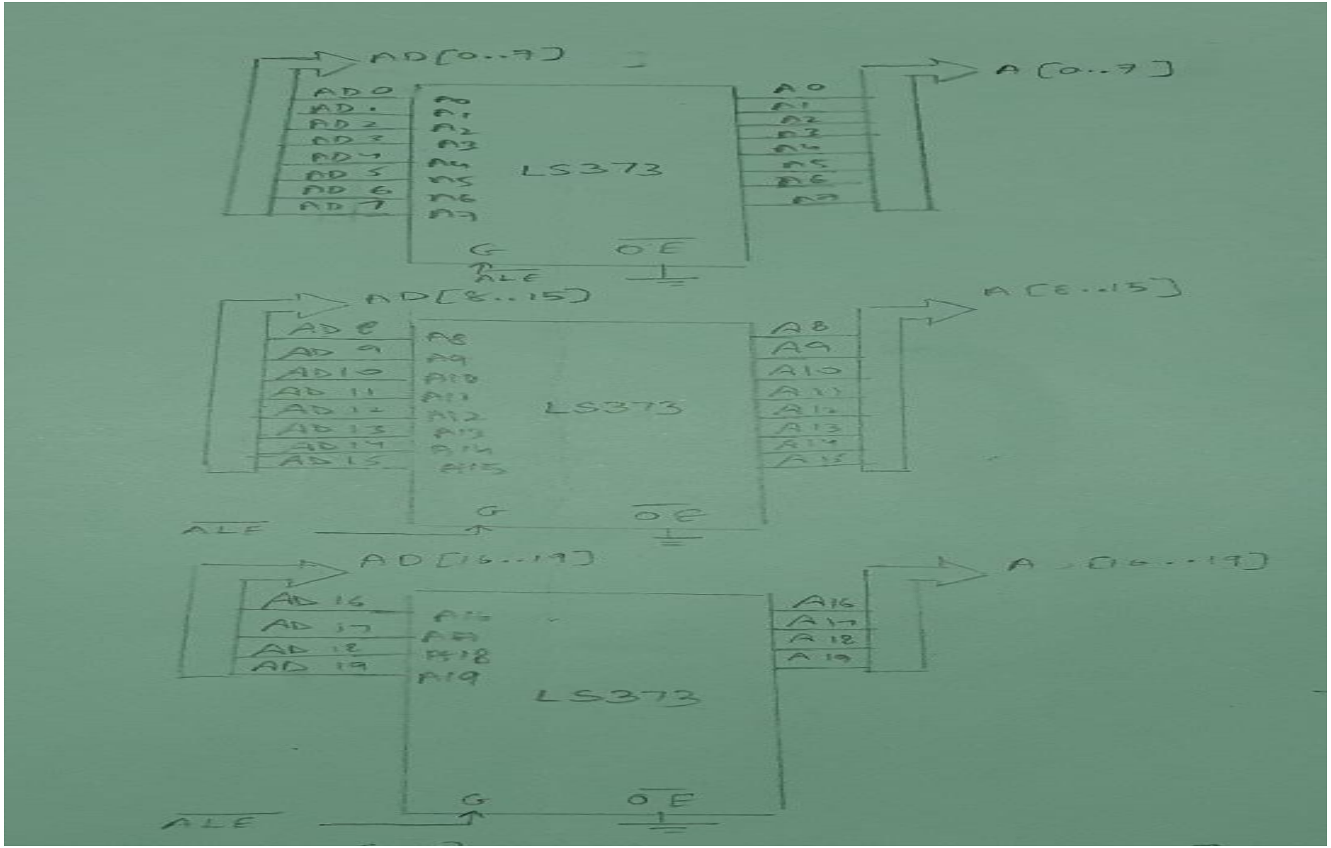


Circuit Diagrams:

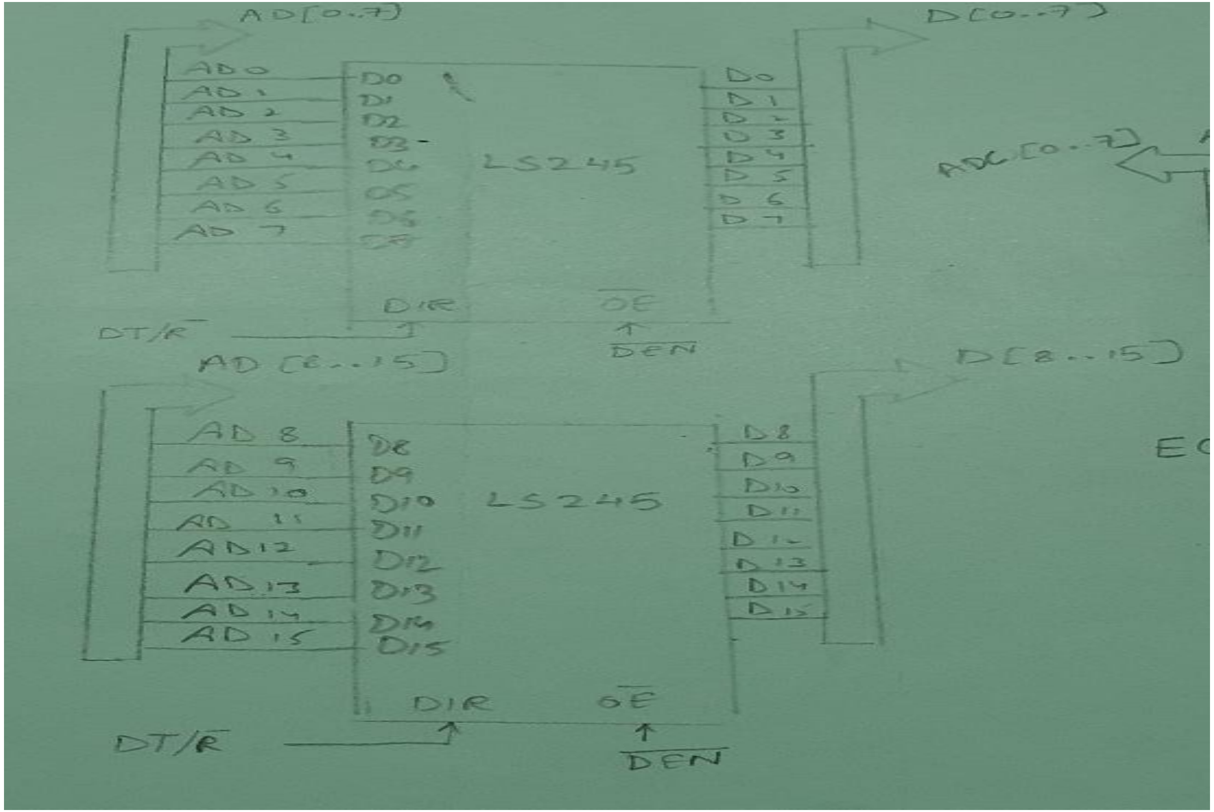
# Clock Generator ( 8284 ) :



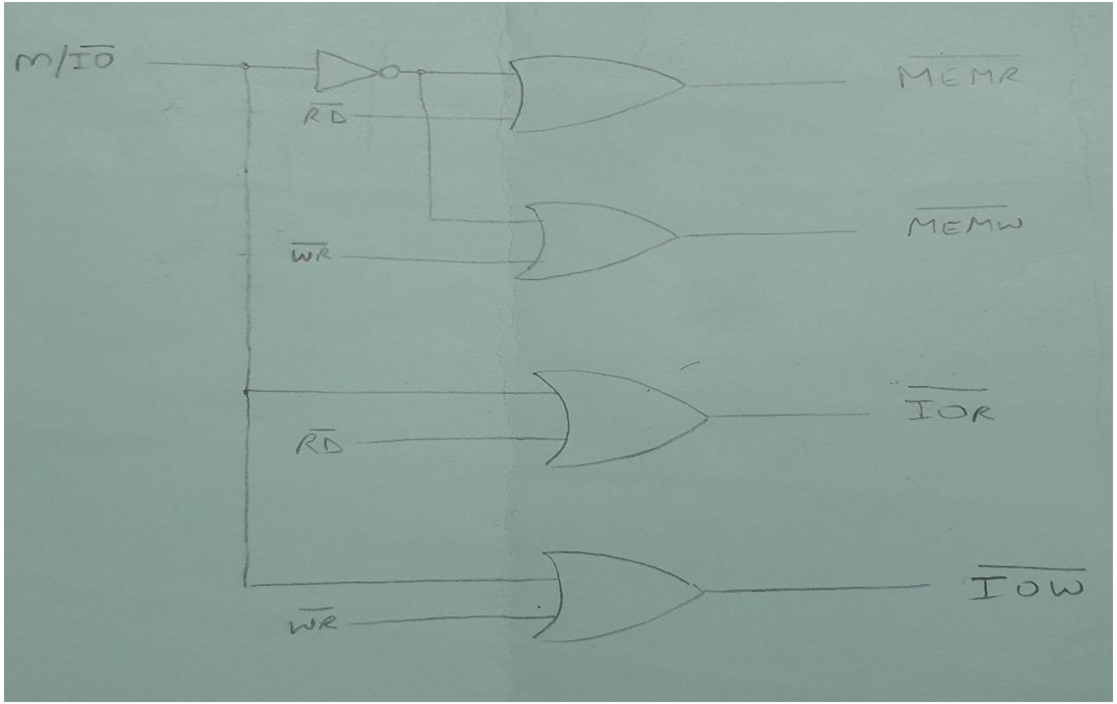
# System Bus (Address) :



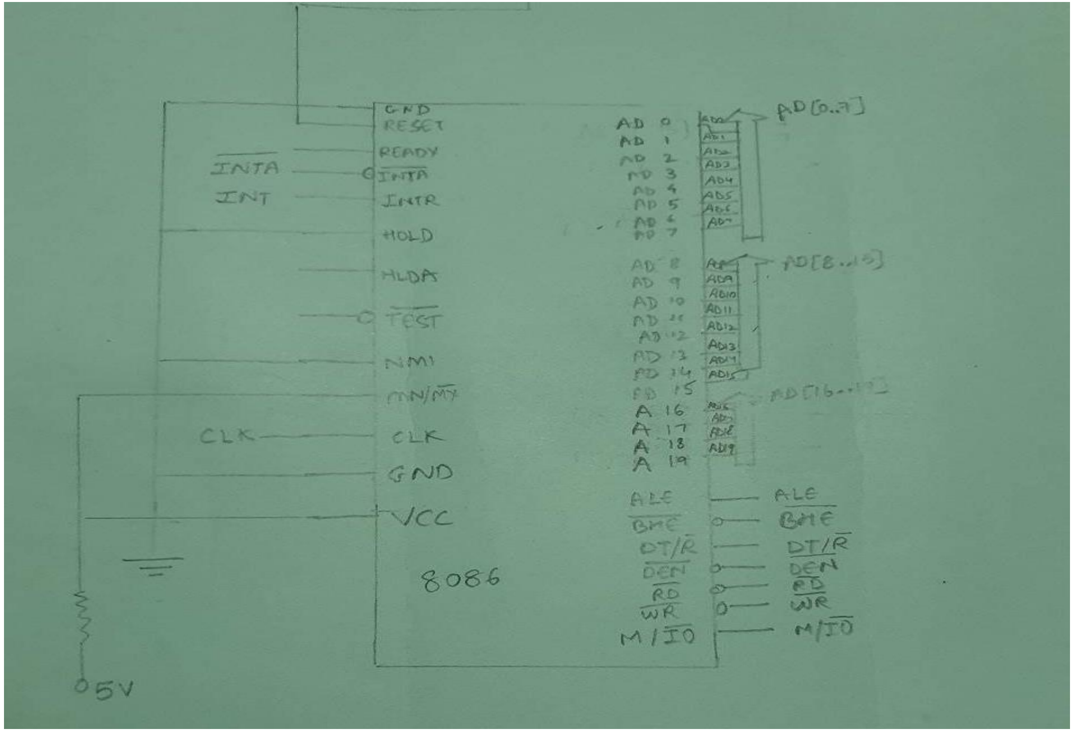
# System Bus Data



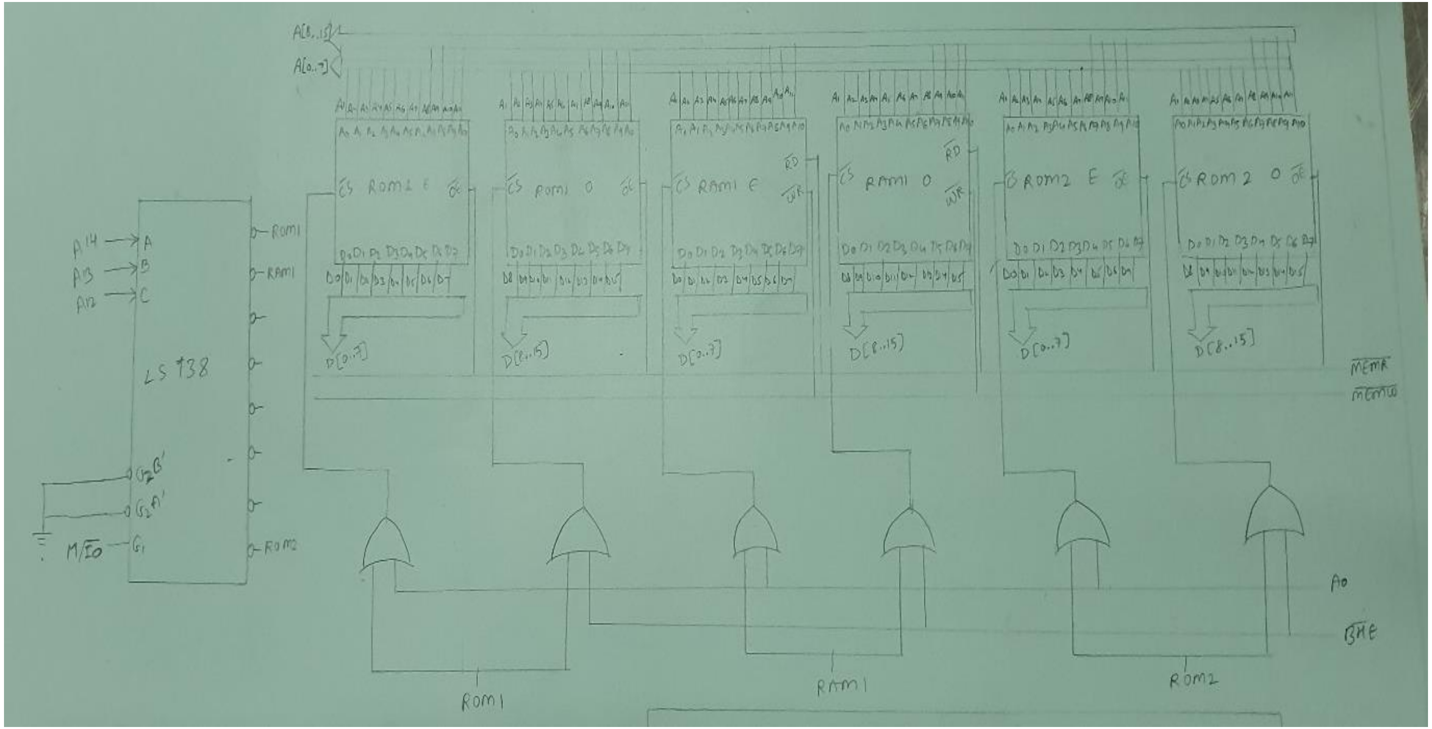
# System Bus(Control Signals):



# 8086:

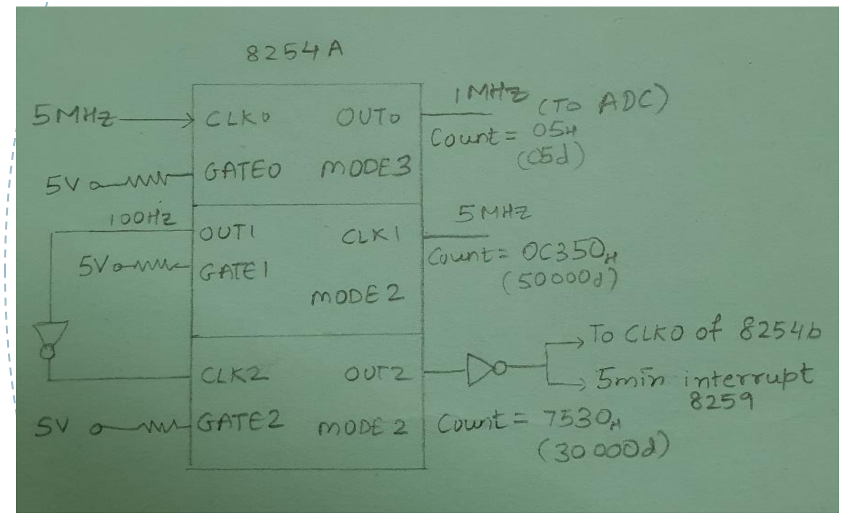

GND
RESET
READY
INTA
INTR
HOLD
HLDA
TEST
NMI
MN/MX
CLK
GND
VCC
8086
INTA
INT
CLK
5V
AD 0
AD 1
AD 2
AD 3
AD 4
AD 5
AD 6
AD 7
AD 8
AD 9
AD 10
AD 11
AD 12
AD 13
AD 14
AD 15
A 16
A 17
A 18
A 19
ALE
BHE
DT/R
DEN
RD
WR
M/IO
AD[0..7]
AD[8..15]
ALE
BHE
DT/R
DEN
RD
WR
M/IO

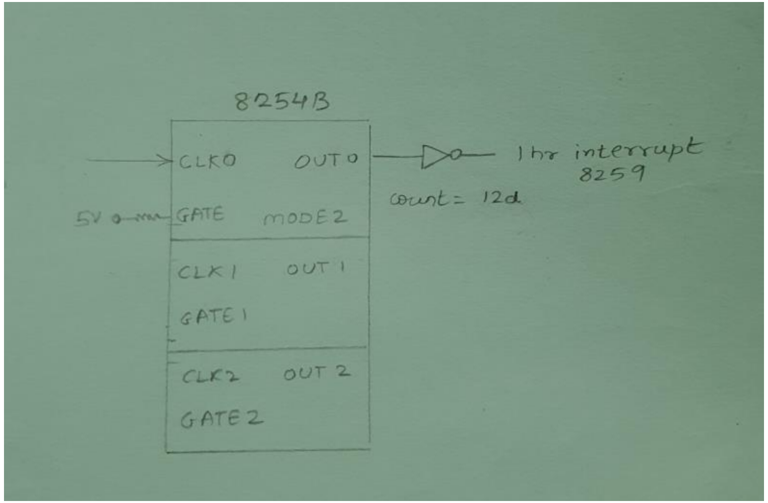
# Memory Interfacing:



# Timers:

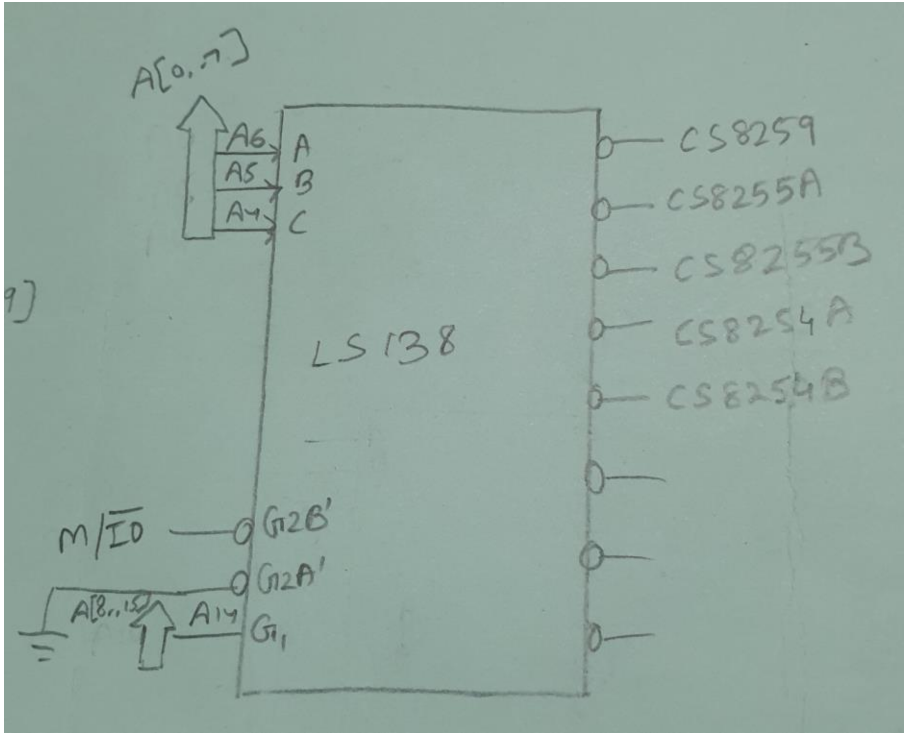
## •8254A



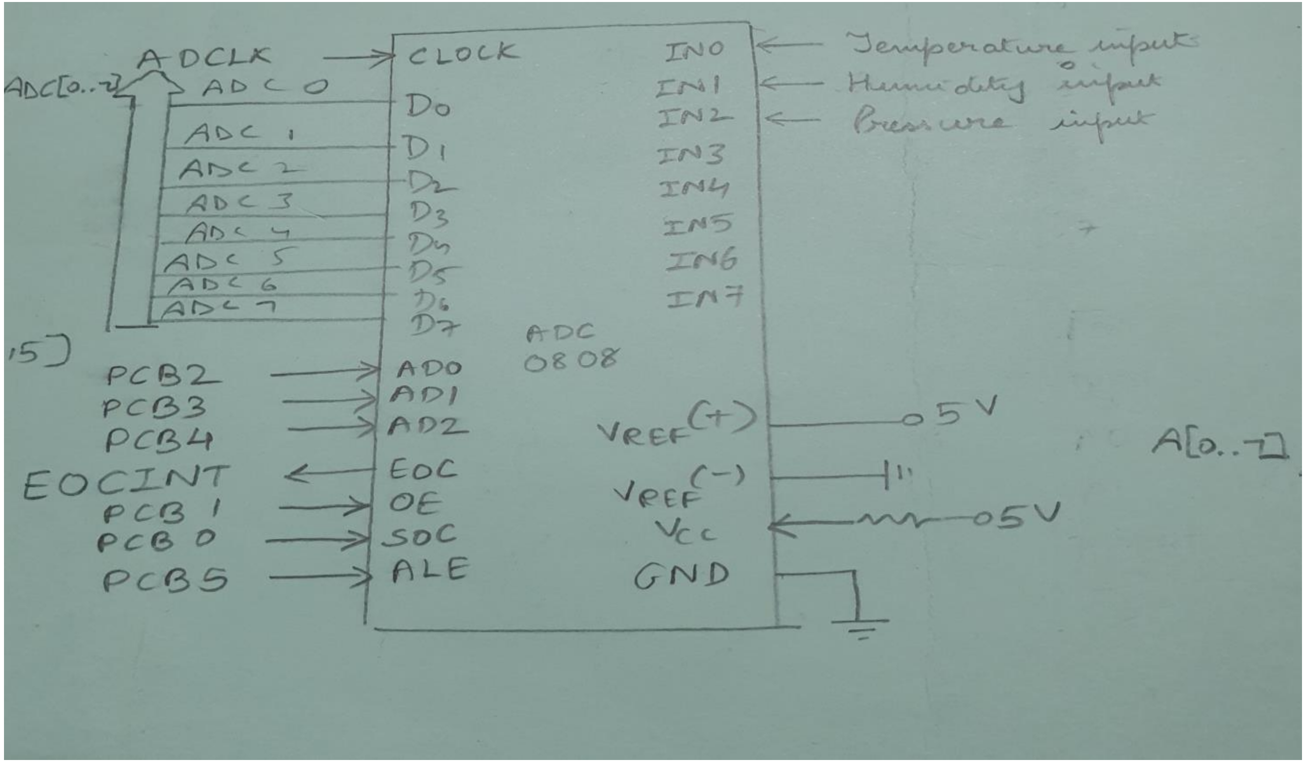

## •8254B

I/O Decoder :



A[0..7]
A6
A5
A4
A
B
C
LS138
CS8259
CS8255A
CS8255B
CS8254A
CS8254B
M/IO
G2B'
G2A'
G1
A[8..15]
A14

# ADC:



ADCLK
CLOCK
ADC[0..7]
ADC 0
ADC 1
ADC 2
ADC 3
ADC 4
ADC 5
ADC 6
ADC 7
D0
D1
D2
D3
D4
D5
D6
D7
IN0
IN1
IN2
IN3
IN4
IN5
IN6
IN7
Temperature input
Humidity input
Pressure input
ADC
0808
15]
PCB2
PCB3
PCB4
EOCINT
PCB 1
PCB 0
PCB5
AD0
AD1
AD2
EOC
OE
SOC
ALE
VREF(+)
VREF(-)
Vcc
GND
5V
5V
A[0..7]

# CONCLUSION

By keeping the embedded devices in the environment for monitoring enables self protection (i.e., smart environment) to the environment. To implement this need to deploy the sensor devices in the environment for collecting the data and analysis. By deploying sensor devices in the environment, we can bring the environment into real life i.e. it can interact with other objects through the network. Then the collected data and analysis results will be available to the end user through the Wi-Fi. The smart way to monitor environment and an efficient, low cost embedded system is presented with different models in this paper. In the proposed architecture functions of different modules were discussed. The temperature, humidity and CO value can be monitored with Internet of Things (IoT) concept experimentally tested for monitoring three parameters. It also sent the sensor parameters to the cloud (Google Spread Sheets). This data will be helpfulfor future analysis and it can be easily shared to other end users. This model can be further expanded to monitor the developing cities and industrial zones for weather monitoring. To protect the public health from pollution, this model provides an efficient and low cost solution for continuous monitoring of environment.

# FUTURE SCOPE

This will conclude that the real time data successfully helpful because of low agriculture crops and wrong prediction of weather. The future of this system is very wide. Internet of Things is just opening its arms, Same system can be applicable to the variety of applications like Data monitoring ,sending and controlling of data at remote location.

# THANK YOU

Anubhav Mishra (RA1911029010023)
Taha Baba (RA1911029010029)