

WEATHER MONITORING SYSTEM

¹Purak sethi, ²Ravi Antil

¹Student, ²Assistant Professor

Department of Electronics and Communication Engineering
Mahavir Swami Institute of Technology, Sonipat, India

Abstract: The industry for Weather Monitoring System is growing. According to studies Weather Monitoring System product are increasing rapidly. Artificial intelligence and IoT are the future. It can be used in as well, for older people to work or for handicap. So they will not depend on other. In this project a DH22 module was used to measure temperature. The project consisted of jumper wire, Arduino Uno and dh22 sensor.

I. INTRODUCTION

Numerous things influence the climate. What's more, climate influences generally living just as nonliving things. At the Weather station investigation of various natural boundaries utilizing a few instruments and gear has been finished. So to meet the objective of climate observing we have planned IoT-based constant, minimal effort, versatile and high velocity climate stations utilizing Arduino Uno. At our climate station, we are estimating some ecological boundaries like temperature, moistness, light power, water level, pressing factor, and height. GSM module, ZigBee module, Ethernet module alongside ADC, and microcontroller are utilized by many climate observing frameworks for ecological boundaries checking. ARM-based Arduino Uno board can deal with numerous activities and a similar one is utilized in this framework. In this framework, there is no requirement for any outside module, microcontroller, and ADC. The proposed framework utilizes an IDLE word processor where projects can be written in C. Yield information can be seen on Firebase. IoT implies the Internet of Things. It gives between systems administration of actual gadgets, structures, vehicles, and different segments like sensors and actuators. By giving organization network to frameworks inserted with hardware, programming, sensors, and actuators

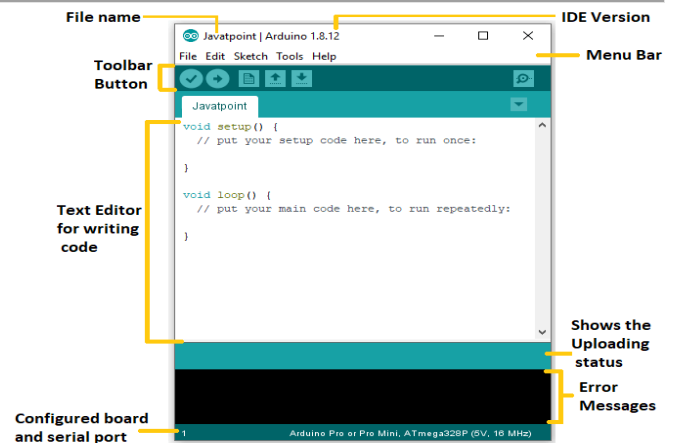
II. Literature Survey

1. In their paper, first there in introduction to devices
2. Process is to measure the temperature using DHT11 sensors
3. Code of dht22
4. Conclusion and references

III. CONSTRUCTION

1. Software

The code for the microcontroller was created in Arduino IDE



Arduino IDE source:Javatpoint

2. Hardware

a) Arduino UNO

Arduino UNO is a low-cost, flexible, and easy-to-use programmable open-source microcontroller board that can be integrated into a variety of electronic projects. This board can be interfaced with other Arduino boards, Arduino shields, and Raspberry Pi boards and can control relays, LEDs, servos, and motors as an output. This board contains a USB interface i.e. USB cable is used to connect the board with the computer and Arduino IDE (Integrated Development Environment) software is used to program the board.

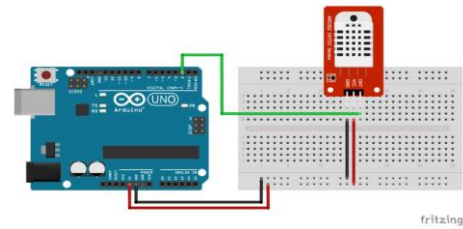
The unit comes with 32KB flash memory that is used to store the number of instructions while the SRAM is 2KB and EEPROM is 1KB.

The operating voltage of the unit is 5V which projects the microcontroller on the board and its associated circuitry operates at 5V while the input voltage ranges between 6V to 20V and the recommended input voltage ranges from 7V to 12V.



b) DH22 sensor

DHT22 is a highly accurate humidity and temperature sensor. This sensor measures relative humidity values. It uses the capacitive sensor element to measure Humidity. For measuring temperature it uses NTC thermistor .



c) Jumper Wire

A jump wire (also known as jumper, jumper wire, jumper cable, DuPont wire or cable) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering



IV. Code

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

#define DHTPIN      2      // Pin which is connected to
the DHT sensor.

// Uncomment the type of sensor in use:
// #define DHTTYPE    DHT11 // DHT 11
#define DHTTYPE    DHT22 // DHT 22 (AM2302)
// #define DHTTYPE    DHT21 // DHT 21 (AM2301)

// See guide for details on sensor wiring and usage:
// https://learn.adafruit.com/dht/overview

DHT_Unified dht(DHTPIN, DHTTYPE);

uint32_t delayMS;

void setup() {
  Serial.begin(9600);
  // Initialize device.
  dht.begin();
```

```
Serial.println("DHTxx Unified Sensor Example");
// Print temperature sensor details.
sensor_t sensor;
dht.temperature().getSensor(&sensor);
Serial.println("-----");
Serial.println("Temperature");
Serial.print ("Sensor:   "); Serial.println(sensor.name);
Serial.print ("Driver Ver: ");
Serial.println(sensor.version);
Serial.print ("Unique ID: ");
Serial.println(sensor.sensor_id);
Serial.print ("Max Value: ");
Serial.print(sensor.max_value); Serial.println(" *C");
Serial.print ("Min Value: ");
Serial.print(sensor.min_value); Serial.println(" *C");
Serial.print ("Resolution: ");
Serial.print(sensor.resolution); Serial.println(" *C");
Serial.println("-----");
// Print humidity sensor details.
dht.humidity().getSensor(&sensor);
Serial.println("-----");
Serial.println("Humidity");
Serial.print ("Sensor:   "); Serial.println(sensor.name);
Serial.print ("Driver Ver: ");
Serial.println(sensor.version);
Serial.print ("Unique ID: ");
Serial.println(sensor.sensor_id);
Serial.print ("Max Value: ");
Serial.print(sensor.max_value); Serial.println("%");
Serial.print ("Min Value: ");
Serial.print(sensor.min_value); Serial.println("%");
Serial.print ("Resolution: ");
Serial.print(sensor.resolution); Serial.println("%");
Serial.println("-----");
// Set delay between sensor readings based on sensor
details.
delayMS = sensor.min_delay / 1000;
}

void loop() {
  // Delay between measurements.
  delay(delayMS);
  // Get temperature event and print its value.
  sensors_event_t event;
  dht.temperature().getEvent(&event);
  if (isnan(event.temperature)) {
    Serial.println("Error reading temperature!");
  }
  else {
    Serial.print("Temperature: ");
```

```
Serial.print(event.temperature);  
Serial.println(" *C");  
}  
// Get humidity event and print its value.  
dht.humidity().getEvent(&event);  
if (isnan(event.relative_humidity)) {  
    Serial.println("Error reading humidity!");  
}  
else {  
    Serial.print("Humidity: ");  
    Serial.print(event.relative_humidity);  
    Serial.println("%");  
}  
}
```

V. Applications

- It does not required any human attention.
- The system deals with monitoring and controlling the environmental conditions like temp, relative humd.
- This system is fully automatic system
- Accuracy is high .
- It can be helpful in places where its quite difficult to stay for human beings longer.

VI. RESULT

We can measure temperature by dht22 sensor.

VII. CONCLUSION

In our project, we explored IOT and made weather monitoring system using Arduino Uno.

We gained in-depth knowledge on hardware components and its architecture

VIII. ACKNOWLEDGMENT

We would like to thank all those who have directly or indirectly encouraged us to take up this project. Specially the authors of different research papers referred during writing this paper.

REFERENCES

1. Arko Djajadi, Michael Wijanarko, "Ambient Environmental Quality Monitoring Using IoT Sensor Network", Internetworking Indonesia Journal (IJ) - Vol.8/No.1 (2016)
2. Tamilarasi B, Saravanakumar P, "Smart Sensor Interface for Environmental Monitoring in IoT", International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) - Volume 5, Issue 2, February 2016
3. Arduino, 'PWM' April 2020. [Online]. Available: <https://www.arduino.cc/en/tutorial/PWM>[Accessed: 28- Apr -2020]
4. Wikipedia
5. <https://pimylifeup.com/raspberry-pi-humidity-sensor-dht22/>
6. Ms. Padwal S. C., Prof. Manoj Kumar, "Application of WSN for Environment Monitoring in IoT Applications", International Conference On Emerging Trends in Engineering and Management <https://github.com/ofalvai/raspi-weather>