JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

"Implementation Of Real Time Temperature, Humidity, Atmospheric Pressure Measurement Using Arduino"

Manikant Raddi , Dr.Sudha M S, Moulya Reddy N R, Likitha G, Likith P , Naveen Kumar
Student, Assistant Professor , Student, Student, Student, Student
Electronics And Telecommunication Engineering
Cambridge Institute Of Technology -Bengalore

Abstract :Arduino, an open-source platform, facilitates the swift development of electronics projects by offering both a physical Programmable Circuit Board (PCB) and software, known as the Integrated Development Environment (IDE), compatible with all major operating systems. In Mauritius, where the island faces a growing number of meteorological hazards, climate monitoring has become imperative. Given the significant reliance on tourism, a key sector of the Mauritian economy, the community's livelihoods are directly impacted by climate events. Thus, raising awareness about climate change through hands-on educational approaches is crucial, particularly in schools.

This paper illustrates how employing a low-cost Arduino-based mobile weather station can empower learners to track climate changes in their immediate surroundings and contribute to decision-making processes. Additionally, the study explores various climate metrics, including indoor/outdoor temperatures, humidity, atmospheric pressure, altitude, dew point temperature, and heat index, to gain insights into the island's climate dynamics.

I. Introduction

As our world continues to evolve with increasing interconnectivity and technological advancements, the demand for accessible and informative weather monitoring systems is on the rise. The Arduino Weather Station project addresses this demand by harnessing the versatility and simplicity of Arduino microcontrollers to develop a practical and cost-effective solution for measuring and displaying crucial environmental parameters, namely temperature, humidity, and atmospheric pressure.

In recent times, the Arduino platform has surged in popularity due to its user-friendly development environment, making it an excellent choice for both novices and seasoned developers. This project aims to leverage the capabilities of Arduino to create a weather station that not only delivers accurate real-time data but also serves as an educational tool for understanding sensor integration, data processing, and user interface design fundamentals.

At the core of the Arduino Weather Station project lies the careful selection of appropriate sensors. Utilizing the DS18B20 sensor for temperature measurements, the DHT22 sensor for humidity readings, and the BMP280 sensor for atmospheric pressure monitoring, known for their precision and compatibility with Arduino, form the backbone for capturing essential weather data.

The project's main objectives are twofold: to precisely measure environmental parameters and present the data in a clear and accessible manner. The Arduino microcontroller reads sensor data, processes it using embedded algorithms, and showcases the results through a user-friendly interface. This interface may incorporate a Liquid Crystal Display (LCD) for on-site monitoring, enabling users to effortlessly access up-to-date weather information. To augment the functionality of the Arduino Weather Station, additional features such as wireless communication and data logging have been contemplated. Integration of wireless protocols like Wi-Fi or Bluetooth facilitates remote access to weather data. Moreover, the capability to log historical data offers valuable insights into weather patterns over time.

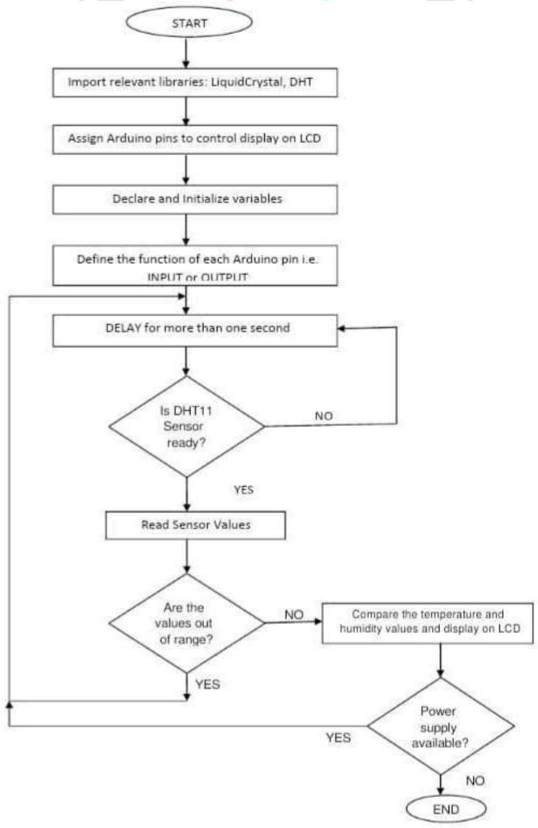
II. Literature Review

Setiyono, Budi, and Rafdito Harisuryo et.al [1]. They have designed an autonomous small cube satellite which provides the weather information without using any internet network. The limitations of this system are that it may not communicate to a long distance without powerful transceivers section, there may be problem in recording data at higher altitude with the help of gas balloon. The components have no protection from rain so they may get damaged even due to long time use.
Barik, L. et.al [2]. They have presented an automated system for weather monitoring which uses different sensors like DHT11, Light dependent resistor and Rain sensor.
Karra, A., Kondi, B., & Jayaraman, R et.al [3]. The authors have proposed a system which senses the temperature and humidity of a particular room. The system cannot be operated from anywhere and the data collected is not accessible.
Suryana, Anang, [4]. The authors have described a system with Arduino which functions using a Wi-Fi shield and different sensors like DHT11, BMP 185, Rain sensor, soil moisture sensor, etc. They used Thinkspeak in order to use MATLAB to get knowledge from the information obtained from the readings on the server.
Krishnamurthi, K., Thapa, S., Kothari, S. and Prakash, A et.al [5]. The R language is used to evaluate results and reveal outputs. They have setup a control unit which can operate other appliances like a.c, heater, fans etc

III.RESEARCH METHODOLOGY

Creating an Arduino Weather Station involves a systematic methodology that encompasses various stages, from defining the project objectives to the final deployment. Here's a step-by-step methodology for building an Arduino Weather Station to measure and display temperature, humidity, and atmospheric pressure:

Flow Chat For Program Design


START
Import relevant libraries: LiquidCrystal, DHT
Assign Arduino pins to control display on LCD
Declare and Initialize variables
Define the function of each Arduino pin i.e. INPUT or OUTPUT
DELAY for more than one second
Is DHT11 Sensor ready?
NO
YES
Read Sensor Values
Are the values out of range?
NO
Compare the temperature and humidity values and display on LCD
YES
Power supply available?
YES
NO
END

PROGRAM:

```
#include LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars and 2 line display byte degree_symbol[8] = { 0b00111, 0b00101, 0b00111, 0b00000, 0b00000, 0b00000, 0b00000, 0b00000 };
int gate=11;
volatile unsigned long duration=0;
unsigned char i[5];
 unsigned int j[40];
unsigned char value=0;
unsigned answer=0;
int z=0;
 int b=1;
void setup()
{
 lcd.init(); // initialize the lcd
lcd.init();
lcd.backlight();
lcd.print("Temp = ");
lcd.setCursor(0,1);
lcd.print("Humidity = ");
 lcd.createChar(1, degree_symbol);
lcd.setCursor(9,0);
lcd.write(1);
lcd.print("C");
lcd.setCursor(13,1);
lcd.print("%");
}
void loop()
delay(1000);
while(1)
{
delay(1000);
pinMode(gate,OUTPUT);
digitalWrite(gate,LOW);
delay(20);
digitalWrite(gate,HIGH);
pinMode(gate,INPUT_PULLUP);//by default it will become high due to internal pull up //
delayMicroseconds(40);
duration=pulseIn(gate, LOW);
if(duration <= 84 && duration >= 72)
 {
while(1)
{
 duration=pulseIn(gate, HIGH);
if(duration <= 26 && duration >= 20)
{
value=0;
}
else if(duration <= 74 && duration >= 65)
{
 value=1;
}
 else if(z==40)
{
 break;
}
i[z/8]|=value<<(7- (z%8));
 j[z]=value;
 z++;
 }
 }
answer=i[0]+i[1]+i[2]+i[3];
lcd.setCursor(11,1);
lcd.print(i[0]);
}
if(answer==i[4] && answer!=0)
{
lcd.setCursor(7,0);
lcd.print(i[2]);
z=0;
i[0]=i[1]=i[2]=i[3]=i[4]=0;
}
}
```

IV.Results And Discussion

The Arduino IDE was used in developing the sketches that were uploaded as firmware into the microcontroller. Thereafter, the system could work without the user's intervention. Libraries are required for a robust firmware development using Arduino. In this case, we used the “Liquid Crystal” and “dht” libraries. Next we set the Arduino pins and attached them to the LCD for display. Arduino pins 9, 10, 4, 5, 6, 7 were attached to the RS, E, D4, D5, D6, D7 pins respectively on the LCD. The pin mode of Arduino pin 12 was set as INPUT. This is the pin that reads the numeric values from the signal pin of the DHT11 sensor. At least a second delay is required to get reliable readings from the DHT11 sensor. However, we used three seconds delay to ensure that the previous values have been displayed. It is also important to confirm that the temperate and humidity readings are within the acceptable range for the sensor. In this work the humidity range was between 20 - 95 relative humidity, while the temperature ranged between 0-50°c. Once the read values are within range, it is displayed on the LCD screen

The next step of the study is to compare these competing models to evaluate that which one of these models is more supported by data.This study follows the methods used by Chen (1983), the Davidson and Mackinnon equation (1981) and the posterior odds ratio (Zellner, 1979) for comparison of these Models.

V.FUTURE SCOPE

Arduino Weather Station project can be broad and exciting, with potential enhancements and adaptations to meet evolving technological and user needs. Here are some future directions and features that could be explored. Integration of Additional Sensors, Energy Harvesting for Power Efficiency, Modular Design for Expandability, Education and Outreach Programs, Integration with Smart Agriculture Practices. As technology advances and user needs evolve, the Arduino Weather Station project can continueto evolve, offering new features, enhanced capabilities, and increased accessibility for users in various domains. The future scope is dynamic, with opportunities for innovation and collaboration in the field of weather monitoring and environmental sensing.

VI.CONCLUSION

We can determine if the weather is HOT, NORMAL, or COLD based on the air temperature and humidity read by a DHT11 sensor. All the components used in this project were cased with plasticfoam - which would have otherwise being discarded as waste. The circuit diagram and the component connections used for the design. We also presented the flowchart used for the firmware design which was uploaded into the Arduino-based microcontroller.

This weather monitoring system will provide farmers, pharmacists, event planners and others with accurate information to guide them to take appropriate action. The system is not currently designedto control other devices. However, future studies can extend the system to control several task based on weather conditions

VII.Acknowledgment

References

[1] . Setiyono, Budi, and Rafdito Harisuryo. "Measurement system of temperature, humidity and airpressure over 433 MHz radio frequency: an application on quadrotor." 2015 2nd international conference on information technology, computer, and electrical engineering (ICITACEE).IEEE, 2015.

[2] Barik, L. (2019) “IoT based temperature and humidity controlling using Arduino and RaspberryPi.” International Journal of Advanced Computer Science and Applications, 10(9).

[3] Karra, A., Kondi, B., & Jayaraman, R. (2020, July). “Implementation of wireless communication to transfer temperature and humidity monitoring data using Arduino Uno.” In 2020 international conference on communication and signal processing (ICCSP) (pp. 1101-1105).

[4] Suryana, Anang, et al. "Implementation of Weather Station for The Weather Reality In A Room." 2020 6th International Conference on Computing Engineering and Design (ICCED).

[5] Suryana, Anang, et al. "Implementation of Weather Station for The Weather Reality In A Room." 2020 6th International Conference on Computing Engineering and Design (ICCED).