Design of Weather Monitoring System Using Arduino and Web Devlopment

PREPARED BY

KEVALKUMAR MANDANKA
INFORMATION TECHNOLOGY
CHANDUBHAI S PATEL INSTITUTE
OF TECHNOLOGY
GUJARAT,INDIA
19it063@charusat.edu.in

AKSHAY MALAVIYA
INFORMATION TECHNOLOGY
CHANDUBHAI S PATEL INSTITUTE
OF TECHNOLOGY
GUJARAT,INDIA
19it061@charusat.edu.in

Under Guidance of :-
Prof. Ravi Patel

Chandubhai S Patel Institute
Of Technology
Department Of Information
And Technology

ABSTRACT:

KEYWORDS— DATABASE, DATA ACQUISITION, TEMPERATURE AND HUMIDITY SENSOR, ARDUINO SYSTEM AUTOMATION,WEB DEVELOPMENT.

I. INTRODUCTION

This article identifies the ability to associate control modules with database programming languages to achieve honest results. The weather station is considered to be a technical method that allows the use of specific equipment to measure weather parameters based on atmospheric conditions on land or at sea for a proposed location to achieve predicted weather conditions and study climate characteristics. The problem of weather forecasting began in the 19th century and officially started thereafter, and then under certain atmospheric conditions, the basis of measurement and recording based on real data at specific locations was changed [1]. The collected data can determine and confirm the recommended warranty for the selected location. The weather is mainly affected by two parameters, temperature and humidityThese parameters fluctuate especially in places affected by the solar radiation temperature and the vertical position of the sun radiating at a specific location based on the latitude of the tropical line. In fact, due to the high-level effects that may be degraded in the entire scheme under certain minor changes, the atmosphere is considered a chaotic and vital scheme. Therefore, this matter will cause trouble for climate predictions in the long future, and predictions may be made only in the following days. Therefore, weather station experts will constantly predict weather conditions based on theoretical research on climatic conditions. It must be mentioned that human struggles in the past followed some undesirable parameters and terminology, such as measuring pressure, humidity intensity research and its impact on temperature, climate conditions, and finally the conditions that affect the sky. There are almost no recommended parameters. A generation of microcomputers and microcontrollers based on prosperity and smart devices greatly simplified the terminology proposed, reducing it to a dynamic factor, and wisely responded to changes in atmospheric conditions. However, user input adaptation is considered to be the most important point that affects the preparation of a database model based on predictive output. This research paper introduces two weather parameters based on data packaging and collection used to generate suggested weather stations. These two weather parameters are identified by temperature (T) and humidity (H). In the end, the idea is considered perfect, especially for devices that control a specific house based on sensor readings.

II. RELATED WORKS

Most research papers and studies in this field have not prioritized data storage and data collection based on the output of the microcontroller sensor. In this article, the result data obtained by the system can be stored and drawn at the same time as the monitoring weather station system. In other words, you can demonstrate and view information through both direct and indirect techniques. The term directly means that the weather conditions can be displayed directly on the (16×2) LCD display without creating a database on the ground area. However, indirect technology means that as long as the sensor is measuring weather conditions, the weather conditions can be recorded and stored in the PC. The main challenge of this work is to show and confirm that a microcontroller with a high-energy sensor can be linked with the data acquisition system to create a database based on the weather station attributes. The proposed idea allows predicting opportunities based on the data realized by the microcontroller sensors, rather than directly monitoring the system. In order to reduce costs, the proposed system uses a sensor that is identified as a DHT sensor to provide temperature and humidity readings to build the heretical structure of the climate database. The question of the weather forecast depends entirely on the weather conditions on the last day in order to specify how much the weather may change in the future. The paper that proposed the idea of a renewable solar system and a hybrid power station discussed some factors that greatly affect weather conditions, such as the color of solar radiation due to environmental reactions and the reflection of changing temperature conditions and therefore changing humidity conditions proportionally. This article proposes a simple way to monitor/store data locally, that is, users can install the system in a specific location and start recording and monitoring data related to (day/night) automatic systems.

III. SYSTEM MODULE

The weather station system is a large, sensitive and reliable system. Therefore, the key reason why such

systems are so much needed is planning. For example, most airway companies and transportation systems need to plan timetables based on climatic conditions and possible changes. On the other hand, the design requirements for such large-scale systems are not cheap, especially for systems that regularly observe and monitor large cities. This article shows a simple way to evaluate the cost and performance of this important system. In addition, this work also created a data accumulation/recording center to initiate weather forecast tasks. The prosperity and superiority of Arduino microcontrollers greatly promoted this work. In this project, only one sensor was used to implement weather station attributes to create the proposed database accumulation system. DHT sensor, used to measure temperature (T) and humidity (H). The Arduino microcontroller uses these sensors to directly measure data through the 16×2 LCD display, or indirectly by storing the data from the Arduino I/O serial monitor to a specific location in the excel file. Then, analyze the recently implemented data, and draw it into a chart through the R language to display it as the database output result.

Internet of Things:

The major concept we had use in our project is Internet of Things(IOT).The internet of Things can be connecting various types of objects such as smart phone, laptops, tablets, etc. to the internet. This objects are able to collect and transfer data over a wireless network without human intervention.The IOT brings the power of the internet, data processing and analytics to the real world of physical objects. IOT helps us to improve our living Standard with smart technology. IOT is used in different fields such as healthcare, Transportation, etc .

MAJOR COMPONENT AND TOOLS:-

1. Arduino UNO Board ESP8266

Processor: L106 32-bit RISC microprocessor core based on the Tensilica Xtensa Diamond Standard 106Micro running at 80 MHz[5]

Memory:

- 32 KiB instruction RAM
- 32 KiB instruction cache RAM
- 80 KiB user-data RAM
- 16 KiB ETS system-data RAM

External QSPI flash: up to 16 MiB is supported (512 KiB to 4 MiB typically included)

IEEE 802.11 b/g/n Wi-Fi

- Integrated TR switch, balun, LNA, power amplifier and matching network
- WEP or WPA/WPA2 authentication, or open networks

16 GPIO pins

SPI

I^2C (software implementation)[6]

I^2S interfaces with DMA (sharing pins with GPIO)

UART on dedicated pins, plus a transmit-only UART can be enabled on GPIO2

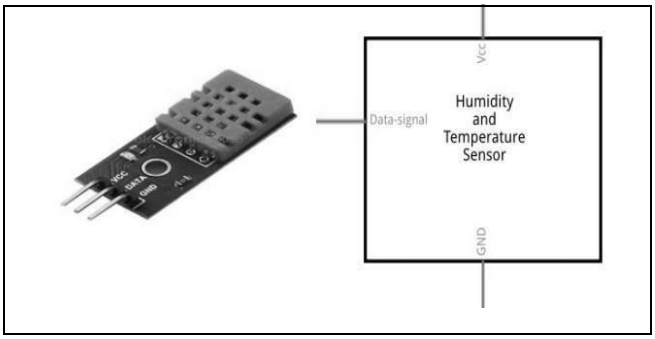
10-bit ADC (successive approximation ADC)

AURDINO UNO BOARD ESP8266

2. Temperature and Humidity Sensor (DHT)

The temperature/humidity sensor (DHT) shown in the figure periodically measures the values of (T) and (H) according to the update cycle, and adjusts the digital signal output. Due to the temperature/humidity sensing module based on proprietary digital signals and data, the sensor is considered to be reliable and stable. The DHT sensor is composed of a resistive element that reads humidity and a negative temperature coefficient

NTC element that reads temperature. The sensor uses 8-bit microcontrollers for processing, these microcontrollers have reliability, sensitivity, stability, high responsiveness, no interference, and the final cost is low. The temperature/humidity sensor (DHT) used is provided by 3 pins (VCC connected to 5V of Arduino, GND (connected to Arduino GND) and DATA (connected to digital pins of Arduino board)).



DHT SENSOR MODULE

3. LCD 16 × 2 display module

The proposed liquid crystal LCD display is considered to be the focus of this work, and it regularly displays several data simultaneously on 16 columns and 2 rows. The main function of the proposed LCD display is to display the information reported by the sensors used as several attributes, so as to clarify the system conditions regularly. According to the connection scheme shown in Figure 3 written by the Fritzing program, it is recommended to interface with the Arduino board.

LIQUID CRYSTAL LCD DISPLAY ALLOWS 16 PINS TO BE CONNECTED ALONGSIDE WITH ARDUINO MICROCONTROLLER AS EXPOSED IN FOLLOWING TABLE.

PIN NO.	FUNCTION	NAME
1	Ground 0 V	VSS
2	Supply Voltage	VDD
	5V	
3	Contrast Adjustment	VO
4	Register Select	RS
5	Read/write signal	R/W
6	Data read/write enable system	E
7 8 9 10	Low order data bus used for data transfer between microcontroller and LCD display.	D0-D3
11 12 13 14	High order data bus used for data transfer between microcontroller and LCD display.	D4-D7
15	Backlight 5V	LED+
16	Backlight 0V	LED-

In accordance with the diagram exposed in Fig, the pins of liquid crystal 16 × 2 LCD display are attached to Arduino UNO as follows:

- Vss pin of the LCD is connected to the GND.
- VDD pin is connected to the 5V.
- Vo is connected to a 10KΩ potentiometer to adjust the contrast of the display.
- Rs pin is connected to digital pin 7.
- R/W pin is connected to the GND.
- E pin is connected to pin 8.
- D0 – D3 are not connected.
- D4 is connected to pin 9, D5 to pin 10, D6 to pin 11, D7 to pin 12 of Arduino respectively.

4.Google Firebase:

Firebase

Firebase is a toolset to “build, improve, and grow your app”, and the tools it gives you cover a large portion of the services that developers would normally have to build themselves, but don’t really want to build, because they’d rather be focusing on

the app experience itself. This includes things like analytics, authentication, databases, configuration, file storage, push messaging, and the list goes on. The services are hosted in the cloud, and scale with little to no effort on the part of the developer.

5.Arduino IDE:

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards.

CURRENT STATUS OF CODE:-

```
#include <ESP8266WiFi.h>
#include <ESP8266WiFiAP.h>
#include <ESP8266WiFiGeneric.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266WiFiScan.h>
#include <ESP8266WiFiSTA.h>
#include <ESP8266WiFiType.h>
#include <WiFiClient.h>
#include <WiFiClientSecure.h>
#include <WiFiServer.h>
#include <WiFiUdp.h>

#include <Firebase.h>
#include <FirebaseArduino.h>
#include <FirebaseCloudMessaging.h>
#include <FirebaseError.h>
#include <FirebaseHttpClient.h>
#include <FirebaseObject.h>
```

```
#include <LiquidCrystal_I2C.h>
#include <dht.h>
dht DHT;
#define DHT11_PIN 7

#define FIREBASE_HOST "arduino-weather-station-c087c.firebaseio.com"
#define FRIEBASE_AUTH "mPpOUEyy3qsNKADINaodYTGygFFOI27140x6RdPE"

#define WIFI_SSID "gliot"
#define WIFI_PASSWORD "GLIOT1212"

LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
void setup() {

  //WIFI

  Serial.begin(9600);

  WiFi.begin(WIFI_SSID,WIFI_PASSWORD);
  Serial.print("Connecting");

  while(WiFi.status() != WL_CONNECTED)
  {
    Serial.print(",");
    delay(500);

  }

   Serial.println();
   Serial.print("CONNECTED");
   Serial.print(WiFi.localIP());

   //FIREBASE

   Firebase.begin(FIREBASE_HOST,FIREBASE_AUTH);
   delay(500);

  lcd.init();
  lcd.backlight();
  lcd.print("AURDINO WEATHER");
```

```
  lcd.init();
  lcd.backlight();
  lcd.print("AURDINO WEATHER");
  lcd.setCursor(0, 1);
  lcd.print("SYSTEM PROJECT");
  Serial.begin(9600);

}

void loop() {

  int chk = DHT.read11(DHT11_PIN);

  float h = DHT.humidity;
  float t = DHT.temperature;

   Firebase.setString("hum/humVal",String(h));
   delay(300);
    Firebase.setString("temp/tempVal",String(t));
   delay(300);
```

```
  Serial.print("Temperature = ");
  Serial.println(DHT.temperature);
  Serial.print("Humidity = ");
  Serial.println(DHT.humidity);
  delay(1000);

  lcd.setCursor(0, 0);
  lcd.print("Temperature:");
  lcd.print(t);
  lcd.print("C");

  delay(1000);
  lcd.setCursor(0, 1);
  lcd.print("Humidity:");
  lcd.print(h);
  lcd.print("%");

  //lcd.noCursor();
  delay(1000);
  // lcd.blink();

}
```

METHODOLOGY

The system main governor represented by Arduino UNO controls each component separately. As mentioned earlier, the system consists of an LCD display that directly displays the data provided by the sensorThe .DHT11 sensor measures (T) and (H) locally based on Arduino code. The following figure shows the entire connection scheme of the system, which is similar to the entire components of the proposed system. Each element in the system needs to provide 5V and GND from the Arduino boardTherefore, the data pin of the DHT sensor is connected to pin 5 of the Arduino, and the LCD display connection is shown above. The proposed system follows a specific work plan. The plan is proposed to make statistics on the control specifications prepared by the system designer. The controller specification brings trouble to the sensor, mainly the sensor responsible for realizing the attributes of the weather station. First, the DHT sensor starts to measure the temperature (T) in degrees Celsius (oC) and the humidity (H) in percentages (%), and transmit the data directly and indirectly to the display and I/O of the microcontroller, respectively. Monitor. Finally, the main challenge of this article is to store the suggested sensor output in a simple excel file by establishing a link between the Arduino serial monitor and the Arduino code-based data acquisition program to create our own R-based language The database system to solve, for any local specific area.

Reference:-

Conclusion:-