CENTRALIZED SENSOR MONITORING SYSTEM

Akshay Pathak, Himanshu Tharad, Satyam Kumar, Pavan Kumar,S.Visnu Dharsini

Student,Computer Science and Engineering , SRM Institute Of Science And Technology, Tamil Nadu, India
Student,Computer Science and Engineering , SRM Institute Of Science And Technology, Tamil Nadu, India
Student,Computer Science and Engineering , SRM Institute Of Science And Technology, Tamil Nadu, India
Student,Computer Science and Engineering , SRM Institute Of Science And Technology, Tamil Nadu, India
Assistant Professor,Computer Science and Engineering, SRM Institute Of Science And Technology, Tamil Nadu, India

ABSTRACT

Smart cities is a major outbreak in implementation of different IT and computer technology. The focus is on providing environmental sustainability and efficiency, sustainable homes and buildings, efficient use of energy resources by implementing smart roads and buildings monitoring system, smart transportation, health care, global warming monitoring and security system. Various sensors are used to collect data like temperature, light, pressure and process it to form a meaningful interpretation. This technology is one of the major application of the smart city idea. The implementation of the technology ranges from homes and schools to large industries.The Proposed system enables to monitor different data sent from various sensors installed in a smart city.The data can be used to study the fluctuation and anticipate any possibilities of unnatural incident.

Keyword : - *Arduino Uno , Sensor, Monitoring System*

1. INTRODUCTION

The objective of this project was to use and Arduino to read a sensor and send the values to the internet, to be stored in a Web Server and displayed.It consists in an Arduino Uno with an Ethernet Shield and a DHT 11 temperature / moisture sensor, acting as a Web Client. It sends POST requests with the readings to a web server running a custom Database and PHP application.

The PHP app stores the values when new POST requests are received and also serves the pages that display the information. In Part 2, i will explain the use of D3.js to dynamically show the data stored in the Database.The Arduino it's configured to use a Dynamic IP Address, in order to solve any conflicting IP issues, and also to work easily with most home networks/routers.

2. SYSTEM REQUIREMENTS

Following is the system requirements -

2.1 Esp8266 Module

ESP8266 is a 3V WiFi module very popular for its Internet of Things applications. ESP 8266 maximum working Voltage is 3.6V

Fig -1: Esp8266 Module

2.2 Arduino Uno

The UNO is the best board to get started with electronics and coding. If this is your first experience tinkering with the platform, the UNO is the most robust board you can start playing with. The UNO is the most used and documented board of the whole Arduino family.

Fig -2: Arduino Uno

2.3 DHT11 Temperature Humidity Sensor Module

The DHT11 humidity and temperature sensor makes it really easy to add humidity and temperature data to your DIY electronics projects. It’s perfect for remote weather stations, home environmental control systems, and farm or garden monitoring systems.

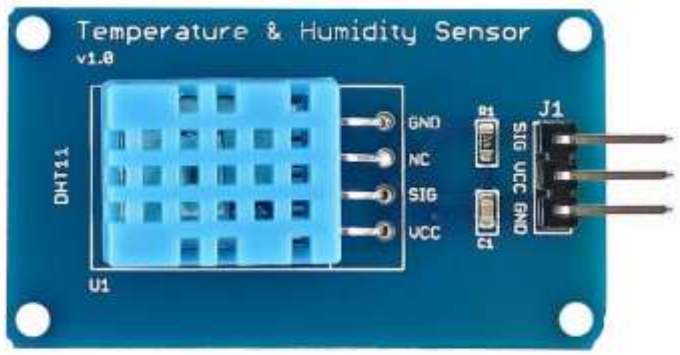

Fig -3: DHT11 Temperature Humidity Sensor Module

2.3 Softwares

You need access to a web server (can be from a free hosting company) with capability to run PHP applications and also to create databases.

3. Architecture

Introduction related your research work Introduction related your research work Introduction related your research work Introduction related your research work Introduction related your research work Introduction related your research work Introduction related your research work Introduction related your research work Introduction related your research work Introduction related your research work

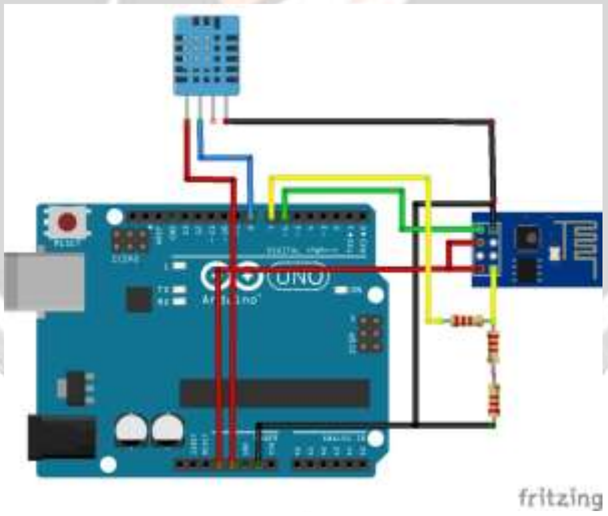

Fig -4: System Architecture

4. Power Consumption

First, you need to keep in mind that the Esp8266 runs on 3.3V , connecting it to 5V could work but it is not recommanded.According to its Datasheet,the amperage needed for the Esp8266 to work correctly varies according to its state (transmitting, receiving, deep sleep ,etc...), and reaches its highest at around **200 mA** while the Arduino's DC Current for 3.3V Pin **is Only 50mA,**it actually works but you won't get a perfect result.

So, there are plenty of other ways to power the module (using diodes, using Zener diode, voltage regulator). I believe that the best solution is to make an independant 5 to 3.3V power regulator circuit, you will need the following components :

LM1117 or LD1117 : A very common and cheap 3.3 V regulator that allows you to convert the 5V power used by the Arduino to the 3.3V needed by the Esp8266 module.

A 10 µF capacitor.

Then, when it comes to the TX and RX connections, you have to avoid connecting the TX of the Arduino directly to the RX of the Esp module since that could damage both of them, so two easy solutions you can use to solve this issue. The first one is to use a simple voltage divider with resistors (as in the diagram above) to drop down the 5V voltage from the TX Pin of the Arduino to 3.3 V or you can use a level shifter with a diode, they both do the job correctly and it's up to you to choose either one for your project

5. Working Explanation

First of all we need to connect our Wi-Fi module to Wi-Fi router for network connectivity. Then we will Configure the local server, Send the data to Web and finally Close the connection. This process and commands have been explained in below steps:

1. First we need to test the Wi-Fi module by sending *AT* command, it will revert back a response containing *OK*.

2. After this, we need to select mode using command AT+CWMODE=mode_id , we have used Mode id =3. Mode ids:

    1 =Stationmode(client)
    2 = AP mode (host)
    3 = AP + Station mode (Yes, ESP8266 has a dual mode!)

3. Now we need to disconnect our Wi-Fi module from the previously connected Wi-Fi network, by using the command AT+CWQAP, as ESP8266 is default auto connected with any previously available Wi-Fi network

4. After that, user can Reset the module with *AT+RST* command. This step is optional.

5. Now we need to connect ESP8266 to Wi-Fi router using given command

    AT+CWJAP=”wifi_username”,”wifi_password”

6. Now get IP Address by using given command:

    AT+CIFSR

    It will return an IP Address.

7. Now enable the multiplex mode by using *AT+CIPMUX=1* (1 for multiple connection and 0 for single connection)

8. Now configure ESP8266 as server by using *AT+CIPSERVER=1,port_no* (port may be 80). Now your Wi-Fi is ready. Here ‘1’ is used to create the server and ‘0’ to delete the server.

9. Now by using given command user can send data to local created server:

    AT+CIPSEND =id, length of data

    Id = ID no. of transmit connection

    Length = Max length of data is 2 kb

10. After sending ID and Length to the server, we need to send data like : Serial.println(“circuitdigest@gmail.com”);

11. After sending data we need close the connection by given command:

AT+CIPCLOSE=0

Now data has been transmitted to local server.

12. Now type IP Address in Address Bar in web browser and hit enter. Now user can see transmitted data on webpage.

6. CONCLUSIONS

This project is useful for the implementation of smart cities using sensors . In this project we explained a centralized sensor monitoring system using arduino uno as a hardware and PHP web servers to display the readings on a web portal. With the help of web application we can see the readings easily on a web page , which will be very user friendly.

7. REFERENCES

[1]. ISSU (https://issuu.com/wyliodrin/docs/building_a_smart_city_infrastructur)
[2]. Arduino (https://www.arduino.cc/en/Tutorial/HomePage)
[3]. Tutorials Point (https://www.tutorialspoint.com/arduino/index.htm)