INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 2)
Available online at: www.ijariit.com

Low cost cloud-based intelligent indoor climate control system

Chinmay Avinash Anaokar
canaokar@gmail.com
Shivajirao S Jondhale College of Engineering, Dombivli, Maharashtra

ABSTRACT

The main objective of this project is to develop a method using Internet of Things to control and manage the Household Air Conditioning System. The system will include control of all the functions of an air conditioning system which we use in a normal household. The system will introduce an Intelligent System Agent which will control the temperature and other settings of the air conditioner using various parameters obtained from the environment and the internet. This system will use the Amazon Web Services (AWS) Platform to control and process the data.

Keywords: *Intelligent AC control system, Intelligent air conditioner control system, Household air conditioning control system, Internet of things ac control, Cloud-Based climate control, Intelligent climate control system.*

1. INTRODUCTION

There are various Smart Home Automation Systems available on the market right now. Most of the systems are very expensive when it comes to implementation as the hardware is expensive and there is a lot of modification is required to implement that. The most common used system is a system where a device is connected to the air conditioner which controls the system according to the information fed to it. Such systems are not intelligent and cannot take decisions on their own. To add more features to such systems, we need to upgrade them to higher variants, prices of which are very high and go on increasing according to the feature set. The main aim of our system is to develop an end to end solution which is cost efficient, intelligent, cloud based and controls all the functions of the air conditioner natively without a need of an upgrade. Our idea is to develop a software which will run on the NodeMCU ESP8266 Board[5]. It will be constantly connected to the Amazon Web Services (AWS) Platform which will be controlling and processing the data and will provide an interface to the mobile application. The system will have offline sensors as well in case of an internet connection failure, it will still operate and control the device seamlessly. Various sensors will provide information to the system based on which, it will control the system. The system would not only operate the device based on the ambient temperature but also will take humidity in consideration. This will get better results as the humidity in the atmosphere plays a very important role in determining the actual temperature and the comfort level.

2. SYSTEM OVERVIEW

The controller used in the system will be the NodeMCU ESP8266 Module. Along with the base module, we will be using the DHT11 Sensors to read the real-time temperature data. There will be 2 DHT11 Sensors to provide redundancy in the readings. These sensors will send the Temperature and Humidity information to the ESP8266 Module, which then will process the information. The ESP8266 will be loaded with program which is written in C language which will process the data and send it to the AWS Servers. The AWS Servers will process the data and send the commands to the ESP8266 which will then process them and execute the operations. We will be using the serverless architecture provided by AWS known as AWS Lambda which will have a server-side code written in Node.js. The Lambda code will receive the data sent by the ESP8266 at regular intervals. It will also acquire data from the weather APIs like the Dark Sky Weather API (Application Programming Interface) and use the readings. Along with the temperature data, ESP8266 will also send the current air conditioner status and settings to Lambda which will then send it to the app server which acts as an interface to the Mobile Application so that the devices can be controlled from the Mobile Application.

The system also has a fail-safe code installed which enables the ESP8266 to operate independently in case there is a failure in the internet connectivity.

2.1 Hardware Implementation

The hardware components used in the proposed system need some discussion.

As we need to detect the temperature and humidity of a normal room, we will consider using the DHT11 Sensor which has a range of approximately 0 degree centigrade to 50 degrees centigrade. It is very small in size which will ultimately reduce the form factor of the whole device.

The NodeMCU ESP8266[5] is a system-on-a-chip (SoC) designed by Espressif Systems which is based on the 32-bit RISC CPU with the Tensilica Xtensa LX106 processor. It has features like inbuilt Wi-Fi (802.11b/g/n), GPIO (General Purpose Input/Output), analog-to-digital converter, UART (Universal Asynchronous Receiver/Transmitter), Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I2C) and pulse-width Modulation (PWM).

In this project, the ESP8266 will connect to the Internet while acting as a Wireless Client and connecting to the Wireless Router. When it connects to the router using the SSID (Service Set Identifier), it will automatically get the assigned IP (Internet Protocol) address with DHCP (Dynamic Host Control Protocol).

By default, the ESP8266 Board does not have any code. It must be configured to function according to our needs. That will be done using the Arduino IDE (Integrated Development Environment) running on a laptop through its integrated UART Interface using Micro USB Cable. The board is flashed using the C Code uploaded through the Arduino IDE.
All paragraphs further should have same styling. Some content related to your research work in running paragraphs.

The ESP8266 will is connected to the Air Conditioner's control system using the I^{2}C Inter-Integrated Circuit Interface which will send all the commands directly to the microcontroller of the Air Conditioner.

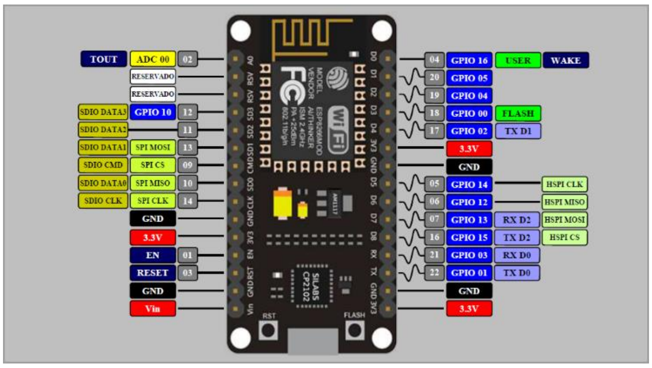

Fig-1 NodeMCU ESP8266

2.2 Software Implementation

- Amazon Web Services (AWS) Platform – The platform provides us serverless architecture to handle the data we send to the cloud. The main benefit of the serverless architecture is that the processing speeds are fast, and it is very cost efficient when compared to the conventional web-server based architecture. The traditional web-server architecture also requires a lot of configuration before we can start using it. The AWS Platform is completely ready-to-deploy and we can get started within a few minutes.
- AWS Lambda[7] – The AWS Lambda service provides us direct access to run functions/methods without deploying an independent server. We will write a AWS Lambda Function to get the data from the AWS IoT Service and then process it. In the function, weather APIs will also be used to get the weather forecast data of the given location. The location will be acquired using the IP address of the device. The function will then use the temperature and humidity data which are read from the DHT11 Sensor, and the forecast data which are obtained directly from the weather APIs.
- AWS IoT[8] – This service provides a native MQTT (Message Queueing Telemetry Transport) Service. It works on the TCP/IP (Transmission Control Protocol / Internet Protocol). MQTT uses the principal of Publish/Subscribe wherein the data is sent/received in channels where multiple parties can access the channels. The ESP8266 will send the sensor data and the air conditioner control data to the MQTT Channel which is subscribed. This data can be accessed by AWS Lambda, which will process it further.

- Arduino IDE[9] (Integrated Development Environment) – The Arduino IDE, in addition to programming the Arduino Boards, also can be used to configure and flash the ESP8266 boards. The libraries required for the ESP8266 and the DHT11 Sensors are attached at the time of compilation[6].
- Android Application – The Android application will retrieve the data from the AWS IoT MQTT Channels[10] and will display the current room information which will include temperature, humidity, current air conditioner settings, etc. The user will also be able to manually override the Intelligent Control System to change the AC Settings as per their liking.

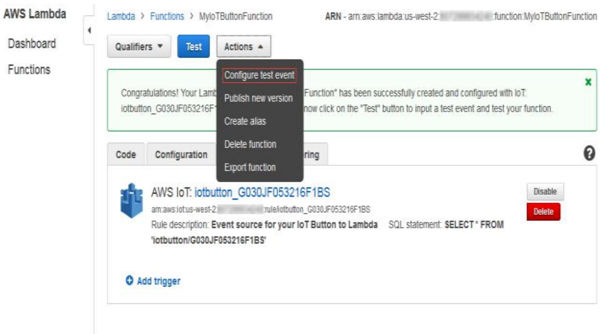

Fig-2 AWS Lambda Dashboard

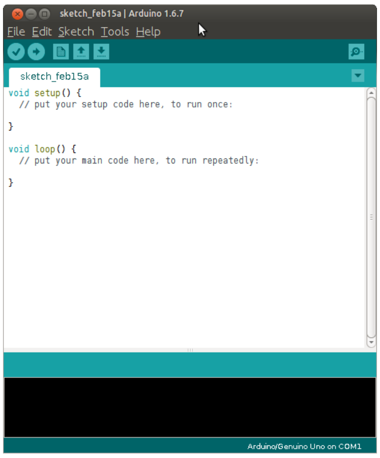

Fig-3 Arduino IDE

3. FLOW STEPS

- The DHT11 Sensors sense the temperature and humidity values from two different locations.
- ESP8266 Receives the data from the sensors through the GPIO (General Purpose Input/Output).
- The values are checked to be close to each other. If there is a lot of variation, then the data set are discarded, and new ones are obtained.
- The ESP8266 connects to the Wireless Router using the SSID and Password stored in the code.
- The ESP8266 establishes the connection to the AWS IoT Platform using the API Keys stored in the code using TCP/IP.
- The data published in the channels is then read by AWS Lambda Function.
- The AWS Lambda function then retrieves the forecast data from the weather APIs.
- All the data acquired are then processed together using a specialized algorithm in the AWS Lambda Function.
- According to the data obtained, the AWS Lambda function publishes the relevant commands to the AWS IoT MQTT Channel which is then read by the ESP8266.
- The ESP8266 then reads the data from the channel and controls the Air Conditioning unit accordingly.
- In case there is no internet connectivity, the ESP8266 executed the fail-safe function and controls the air conditioner using the offline sensor data.

4. RESULTS

As the prototype system is deployed at a setup, the temperature and humidity data is received by the AWS Servers and the Air Conditioning Unit is operated accordingly. The air conditioner temperature is controlled throughout the day as per the variations in the actual temperature and humidity outside.

5. CONCLUSION

A prototype Intelligent Indoor Climate Control system is developed with the help of ESP8266 and DHT11 Sensors which is deployed on the cloud. The prototype fared well in a 24-hour test run and the ESP8266 showed a steady performance. The system intelligently calculates the ideal temperature of the Air Conditioning unit and thus changes the settings of the Air Conditioner to get the desired temperature. The Intelligent system also helps reduce the electricity costs while maintaining comfort of the residents.

6. REFERENCES

[1] Michael G. Rodriguez, Luis E. Ortiz Uriarte, Yi Jia, Kazutomo Yoshii, Robert Ross, Peter H. Beckman “Wireless Sensor Network for Data-Center Environmental Monitoring,” published in 2011 Fifth International Conference on Sensing Technology (ICST)

[2] Designing the Internet of Things by Adrian McEwen and H. Cassimally

[3] The Internet of Things in the Cloud by H. Zhou

[4] Laurentius Kuncoro Probo Saputra, Yuan Lukito “Implementation of Air Conditioning Control System Using REST Protocol Based on NodeMCU ESP8266” published in 2017 International Conference on Smart Cities, Automation & Intelligent Computing Systems Yogyakarta, Indonesia.

[5] http://espressif.com ESP8266 Details

[6] http://playground.arduino.cc/Main/DHT11Lib DHT library

[7] https://docs.aws.amazon.com/lambda/latest/dg/welcome.html – Amazon Web Services Lambda

[8] https://aws.amazon.com/iot/ - AWS IoT

[9] https://www.arduino.cc Arduino software download

[10].https://en.wikipedia.org/wiki/MQTT-MQTT.

