SDT using MQTT SSL Protocol

P Durga Prasad[1], K Hanurathan[2], P Pratyusha[3]
Dadi Institute of Engineering & Technology

ABSTRACT

A new technological era known as the Internet of Things (IoT) is about to begin. Whatever you want to call it—machine to machine, machine to infrastructure, machine to environment, Internet of Everything, smart IoT, smart systems—it's occurring, and it has enormous promise. The Internet of Things (IoT) is built on information that is intelligently provided by embedded functionalities and is envisioned as a network of millions of intelligent individuals connected to "everything". Intelligent systems that interact and communicate with other systems, such as items, surroundings, and architectural designs, make up the Internet of Things. This has led to the creation of numerous documents that have been transformed into helpful features that can "level and control" things, improve our quality of life, make it safer.

Keywords : Micro Controller, Node Mcu, cloud technology, DHT sensor, Arduino IDE.

INTRODUCTION

By providing socially acceptable and timely assistance to perceive, anticipate, and respond to behaviours in the home, smart homes significantly improve the lives of families. Smart houses also need to connect with other smart buildings, equipment, and their underlying protocols. MQTT is a crucial protocol for the communication between intelligent devices and its features. This is a component of the technology used in smart devices to enable user and device-to-device interaction. The motivation behind this research is the fact that the MQTT protocol is one of the standardised protocols used in the Internet of Things for communication in an end-to-end network scenario where security considerations are vital. It was also evident that many IoT developers adopted MQTT due to its advantages, including its capacity to operate on a minimal amount of memory and CPU resources as well as its modest bandwidth requirements. Unfortunately, the primary reasons why the MQTT protocols are not widely employed in networks are still security and data integrity. Since MQTT communications are not by default encrypted, turning on SSL is strongly advised when using a public network. The main objective of this project is to incorporate the advantages of MQTT protocol into IoT communication by providing a unique key. This project has vast number of applicationsin all the fields where IoT is present that may be in medical fields, home automations, smartcities etc.

NODE MCU

"NodeMCU" is a combination of the words "node" and "MCU" (microcontroller unit). In actuality, the firmware rather than the related development kit is what is meant by "NodeMCU" in this context. The ESP8266 Espressif Non-OS SDK produced the firmware, which is based on the eLua project. It makes use of several open-source initiatives, including SPIFFS and lua-cjson. Users must select the model that is relevant to their business and design the firmware that meets their demands due to restrictions. Support for the 32-bit ESP32 is also available. Embrace the message The breadboard may be easily prototyped thanks to the DIP format selection. The ESP-12 module for the ESP8266, a popular Wi-Fi SoC from Tensilica called the LX106 core, is the design's foundation.

WORKING PROCESS

The ESP8266 Wi-Fi Chip is the same in all modules. The availability of GPIO Pins is the primary distinction. For instance, whereas the ESP-12E Module contains 17 GPIO Pins, an ADC Pin, SPI Pins, and other pins, the ESP-01 Module only has 2 GPIO Pins. The ESP-12E is the most well-liked of all of these modules. The NodeMCU team created the NodeMCU Devkit, also known as the NodeMCU Board, using this module as its primary board.The NodeMCU board has 30 Pins (15 on each side), an ESP-12E Module with PCB Antenna, a Silicon Labs CP2102 USB to UART Bridge Controller, two push buttons (one for RESET and the other for Flash), a micro-USB Connector for Power, and two push buttons.

PROPOSED METHOD

In this article, we'll combine the Nodemcu ESP8266 with the Arduino IoT Cloud. With the Nodemcu ESP8266, we'll use the DHT11 temperature and humidity module. Temperature and humidity will be measured, and the results will be sent to the Arduino IoT Cloud where they will be presented on gauges, charts, and message widgets. Both your PC or laptop as well as a dashboard on your smartphone can be used to keep track of these values. The dashboard that you can view on your phone is generated automatically, as I have previously covered in my courses. In any case, if you want to, you may use this project to track the temperature and humidity readings from anywhere in the world. In this we have connected DHT sensor to measure the temperature and humidity in its surrounding.

RESULTS

The results will show the how it is denying when other devices trying to control the parameters and this is the advantage of this proposed method. It will provide security to our information especially when we are relying on IoT.

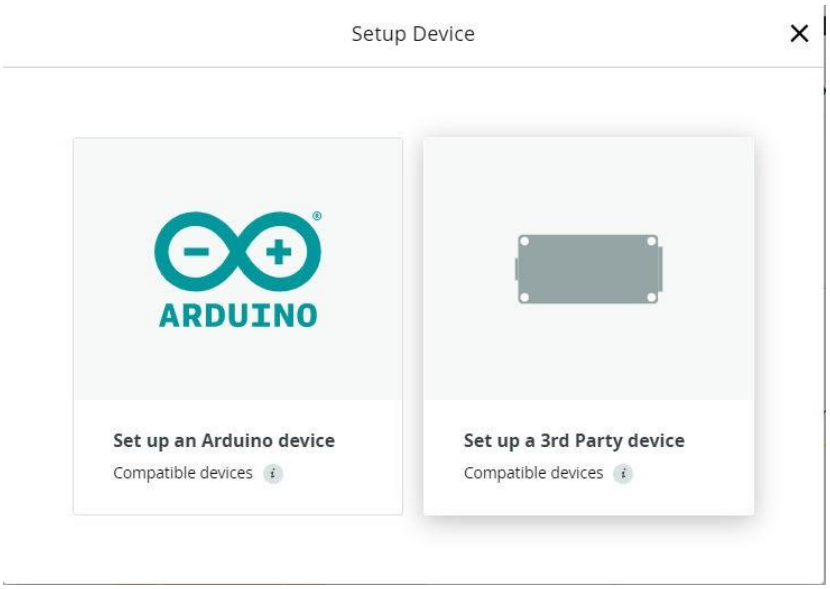
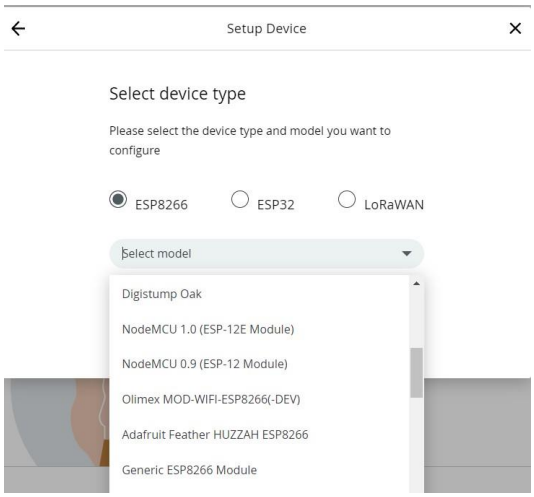


Fig1: Setting up of Arduino



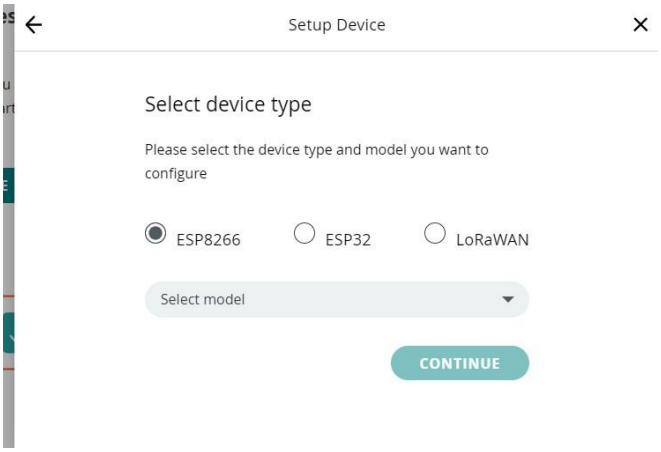


Fig2: Connections and Setup

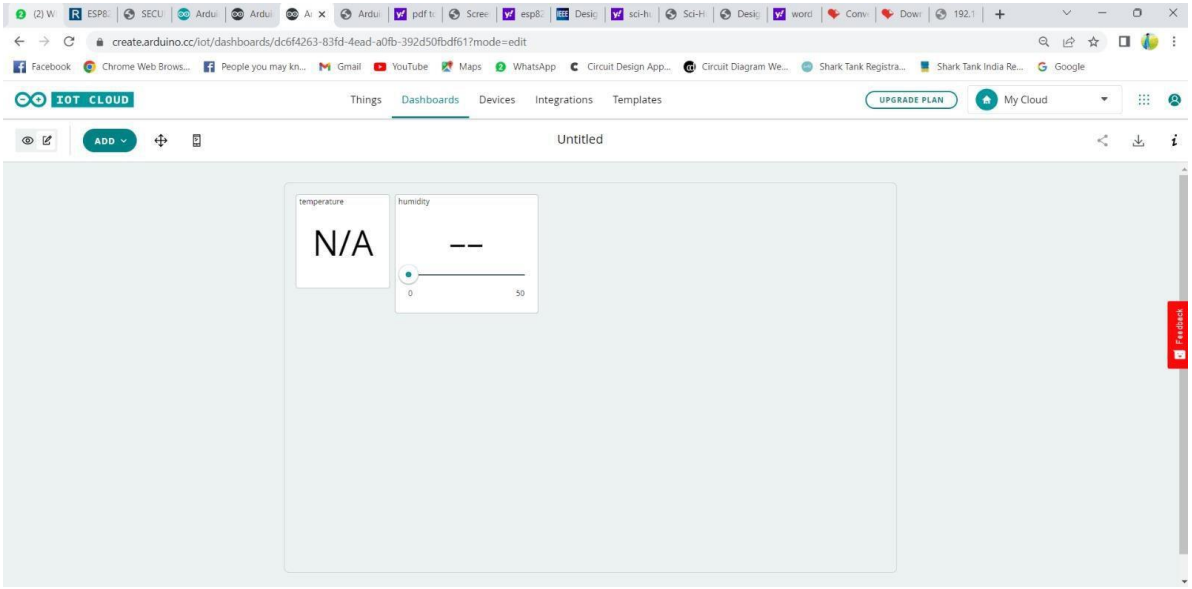


Fig 3: Showing error when we are trying with other device

CONCLUSION

Since MQTT communications are not by default encrypted, turning on SSL is strongly advised when using a public network. The connection will be set up to deny data from any board whose SSL certificate is not confirmed on the SSL database by using this.Therefore, access to the data can only be granted once creating a special key and authentication has been completed.

REFERENCES

[1] S. Andy, B. Rahardjo, and B. Hanindhito. Attack scenarios and security analysisof mqtt communication protocol in iot system. In 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), pages 1–6, 2017.

[2] T. Anagnostopoulos, A. Zaslavsky, and A. Medvedev. Robust waste collection exploiting cost efficiency of iot potentiality in smart cities. In 2015 InternationalConference on Recent Advances in Internet of Things (RIoT), pages 1–6, 2015

[3] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, MichalisKallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. Understanding the mirai botnet. In 26th USENIX Security Symposium (USENIXSecurity 17), pages 1093–1110, Vancouver, BC, August 2017. USENIX Association

[4] A. Bhawiyuga, M. Data, and A. Warda. Architectural design of token basedauthentication of mqtt protocol in constrained iot device. In 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA), pages 1–4, 2017.

[5] Leor Brenman. API Builder and MQTT for IoT. https://devblog.axway.com/apis/api-builder-and-mqtt-for-iot-part-1/2018.