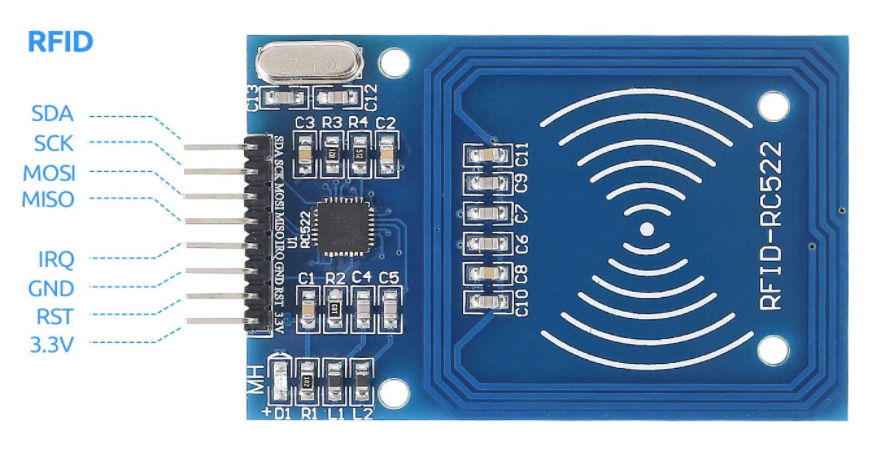
**INTRODUCTION**

The traditional pen-paper based attendance is very time consuming, insecure and causes human errors as well as can cause misconduct. Hence to overcome this problem, a real time **iot enabled rfid based attendance system** has been developed in order to represent the data in the form of unique rfid cards and tags which are easily scannable on the rfid scanner. The system implemented consists of hardware as well as software with the implementation of **MQTT protocol** which is a lightweight, small in size, public subscribe network protocol that transports messages between devices using **ESP8266 Nodemcu** and **Adafruit io** cloud platform. The main objective of this system is to make efficient computerized based attendance system, details of which can be accessed any time from any place using the cloud platform.

**COMPONENTS USED –**

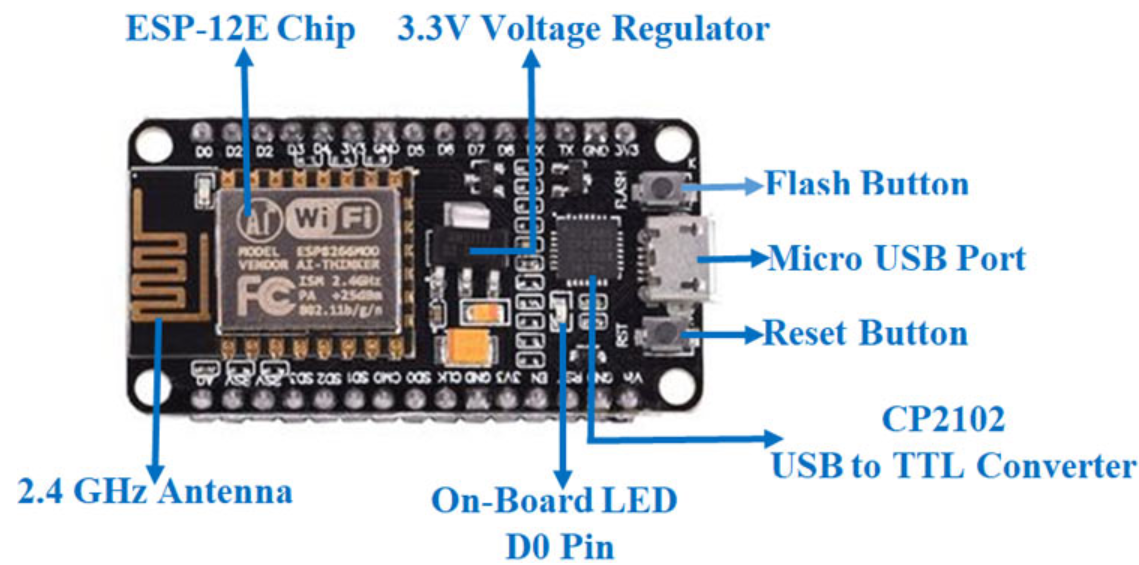
**1.MFRC522 RFID SCANNER AND CARDS**

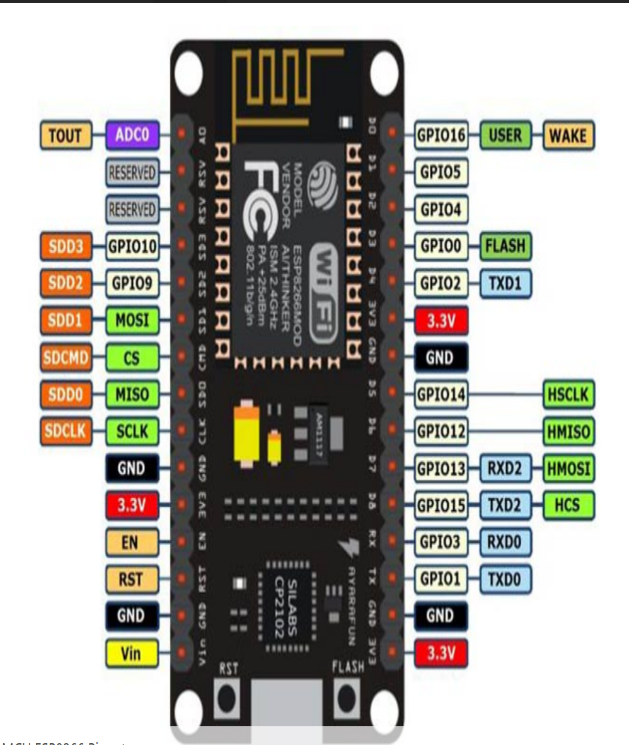
A Reader consists of a Radio Frequency module and an antenna which generates high frequency electromagnetic field. On the other hand, the tag is usually a passive device, meaning it doesn’t contain a battery instead it contains a microchip that stores and processes information, and an antenna to receive and transmit a signal. To read the information encoded on a tag, it is placed in close proximity to the Reader (does not need to be within direct line-of-sight of the reader). A Reader generates an electromagnetic field which causes electrons to move through the tag’s antenna and subsequently power the chip. The powered chip inside the tag then responds by sending its stored information back to the reader in the form of another radio signal. This is called backscatter. The backscatter, or change in the electromagnetic/RF wave, is detected and interpreted by the reader which then sends the data out to a computer or microcontroller.

** **

**2.ESP8266 NODEMCU**

NodeMCU is an open-source Lua based firmware and **development board** specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The **NodeMCU ESP8266 development board** comes with the ESP-12E module containing the ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects. NodeMCU can be powered using a Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.



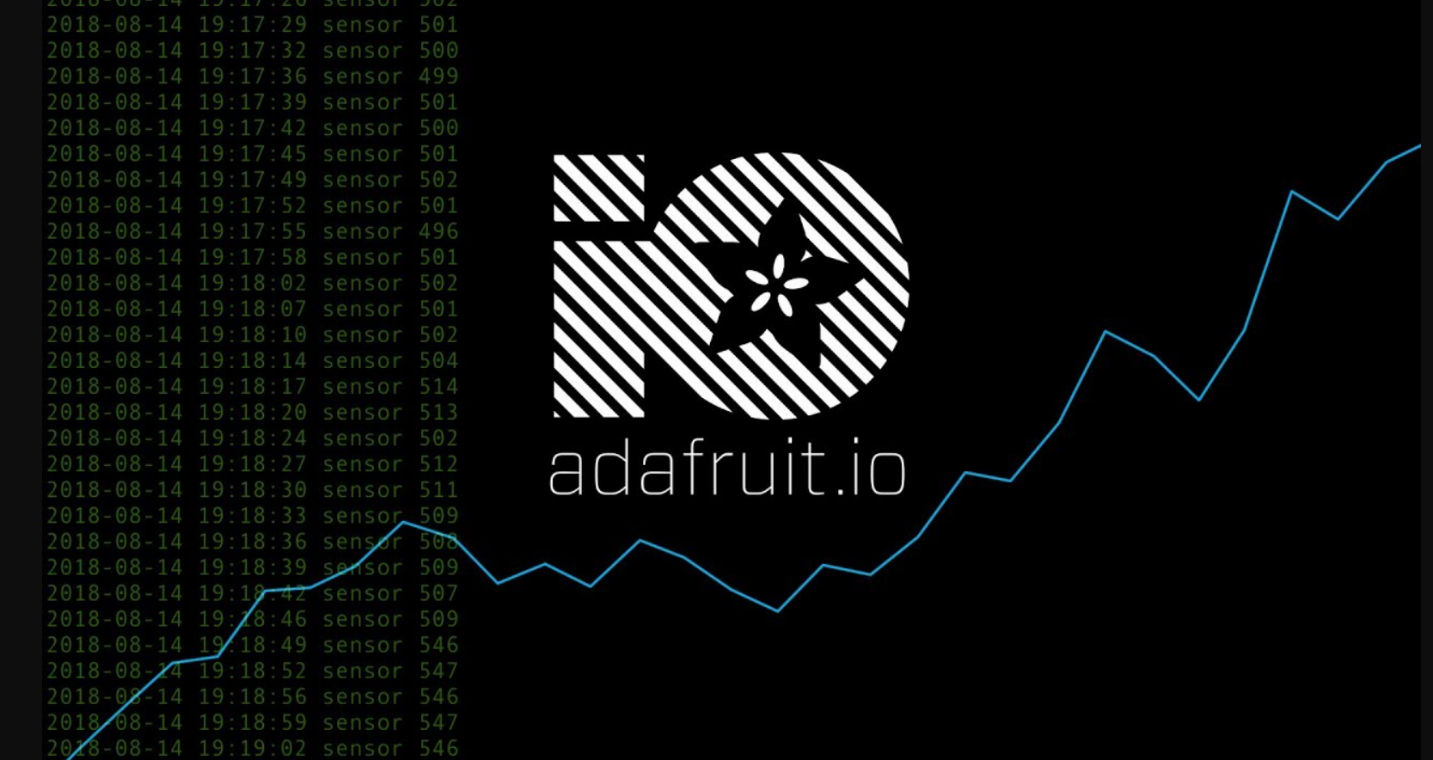


**3.ADAFRUIT IO PLATFORM**

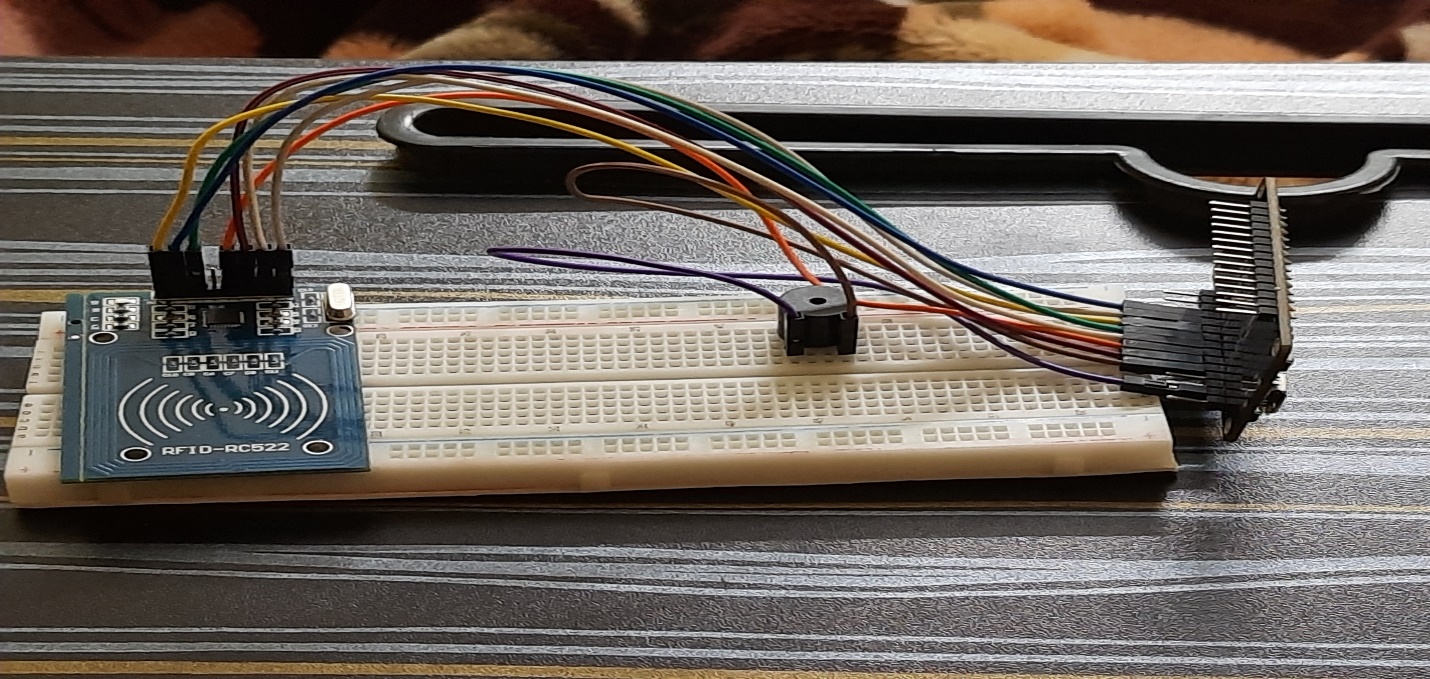
**Adafruit.io** is a cloud service - that just means they run it for us and we don't have to manage it. We can connect to it over the Internet. It's meant primarily for storing and then retrieving data but it can do a lot more than just that.

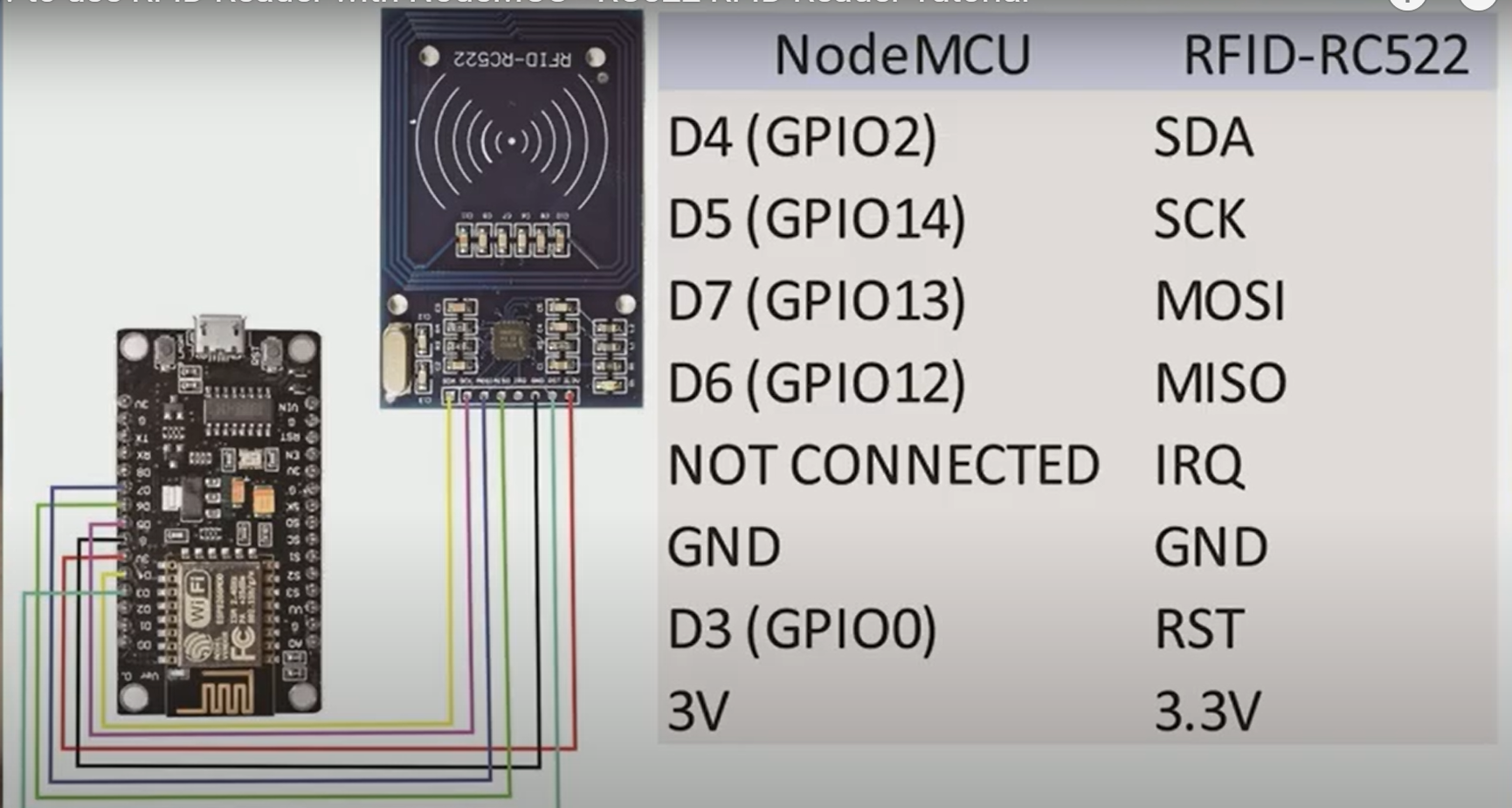
The main functions of this cloud platform are

* Display our data in real-time, online
* Make our project internet-connected: Control motors, read sensor data, and more!
* Connect projects to web services like Twitter, RSS feeds, weather services, etc.
* Connect our project to other internet-enabled devices
* The best part? All of the above is do-able for **free**with Adafruit IO.



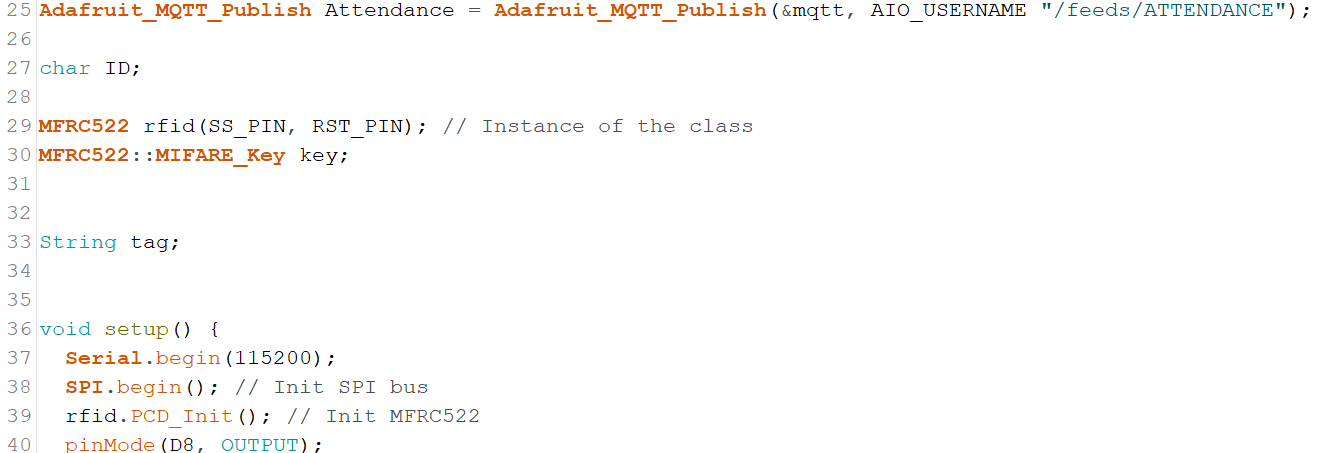
**CONNECTION DIAGRAM AND PIN CONFIGURATION**

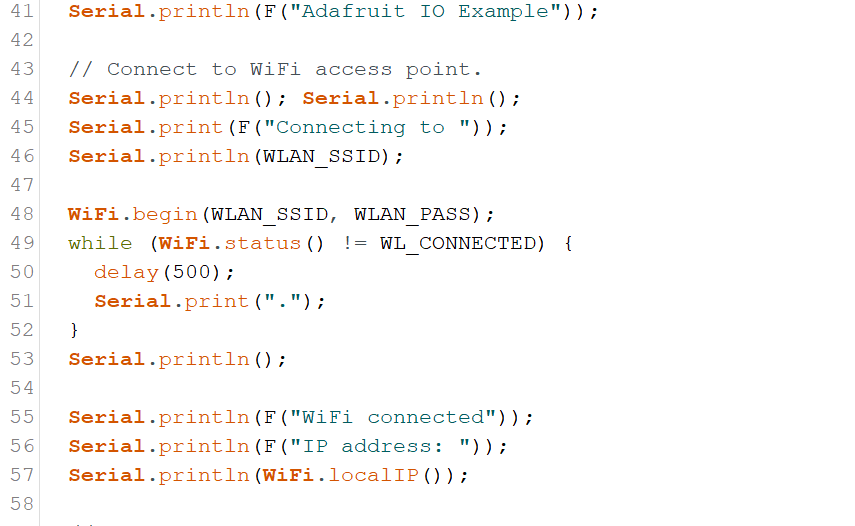
****

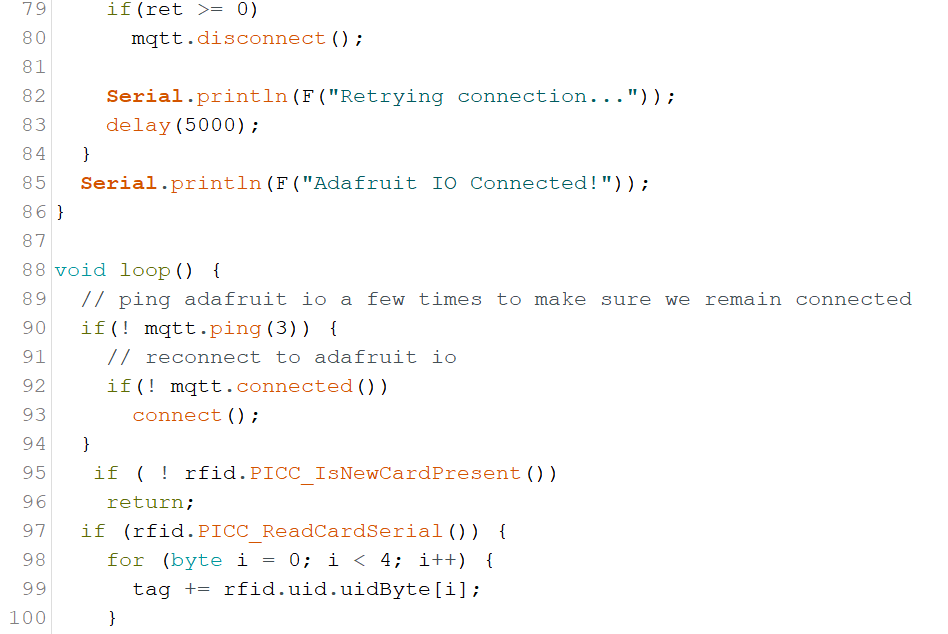
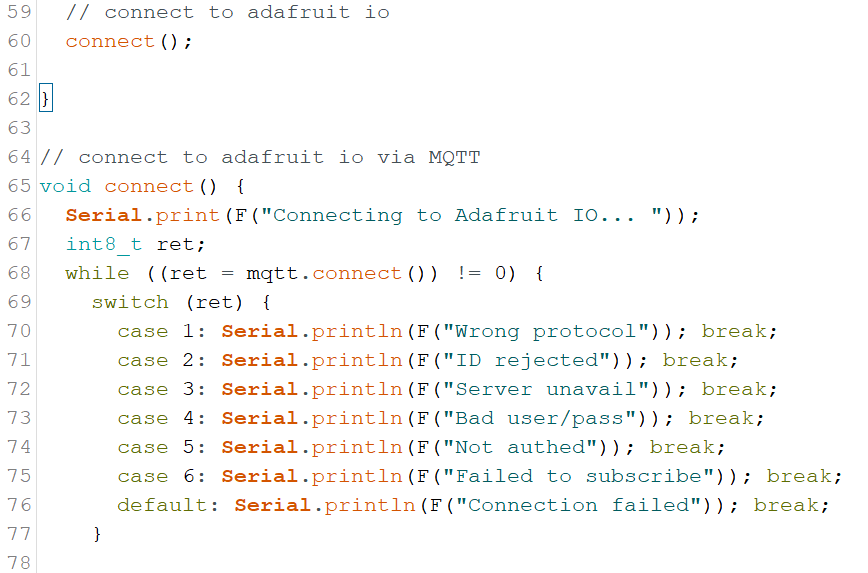
****

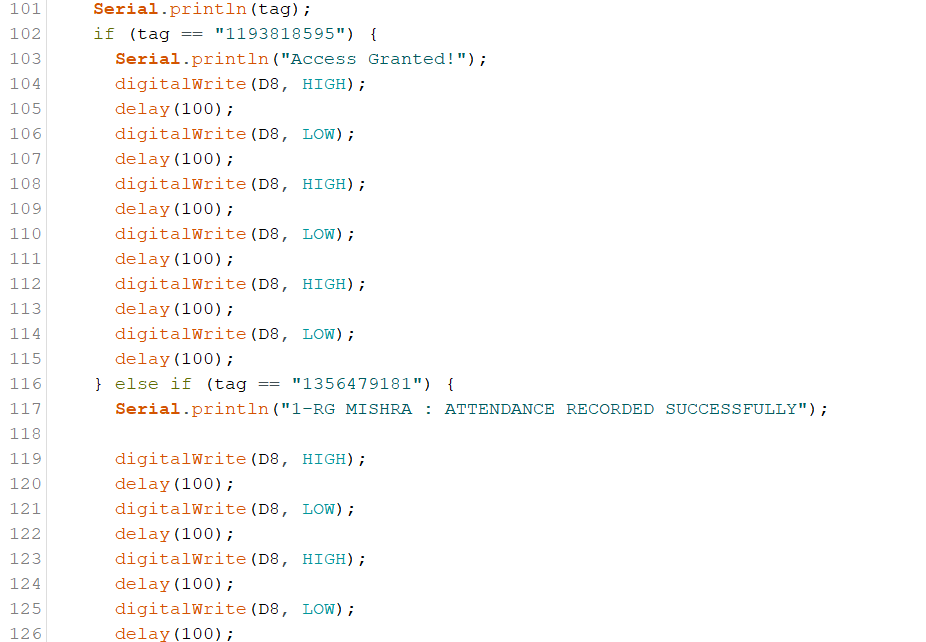
**CODE IMPLEMENTATION**

****





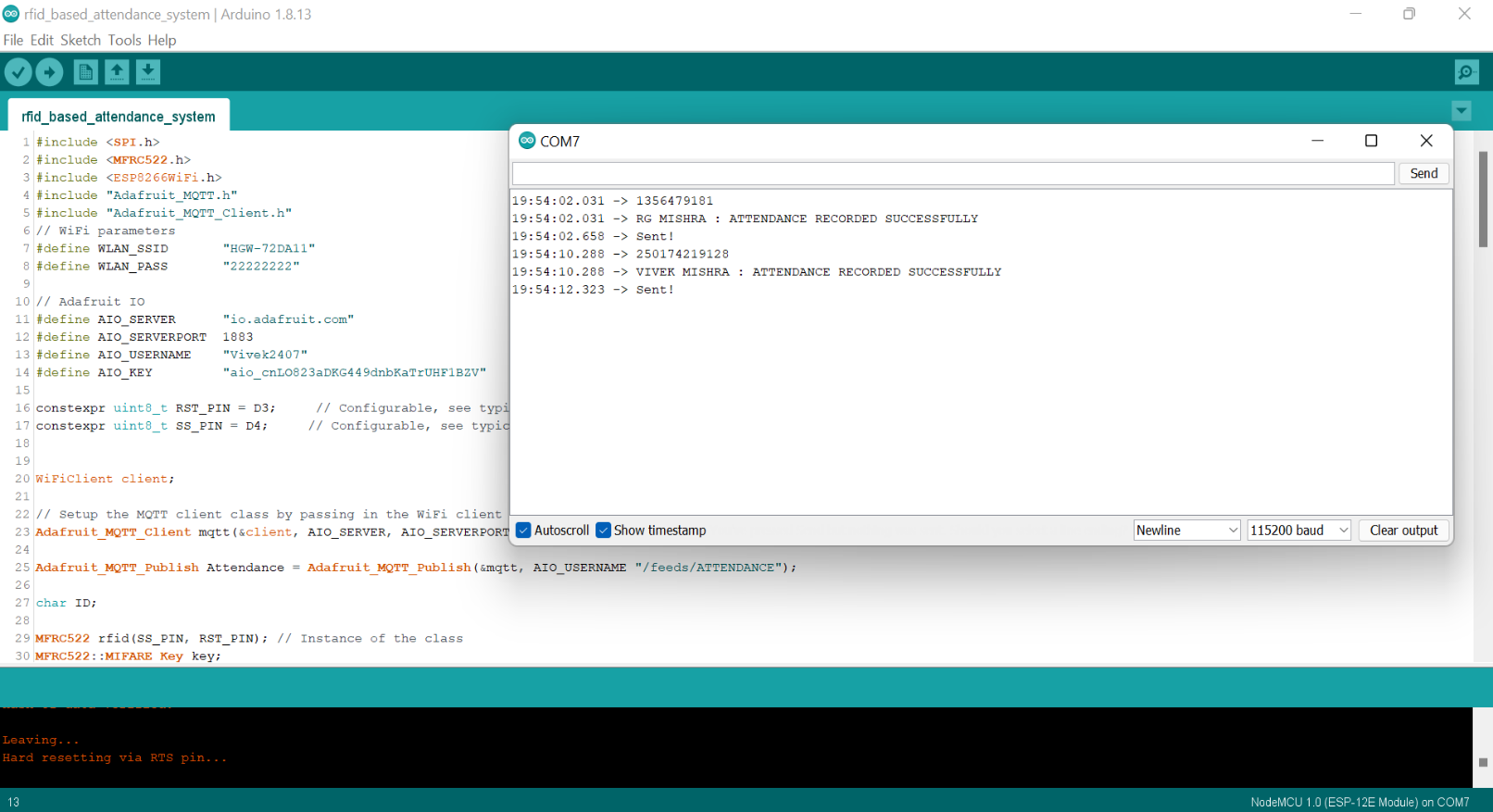
****

****

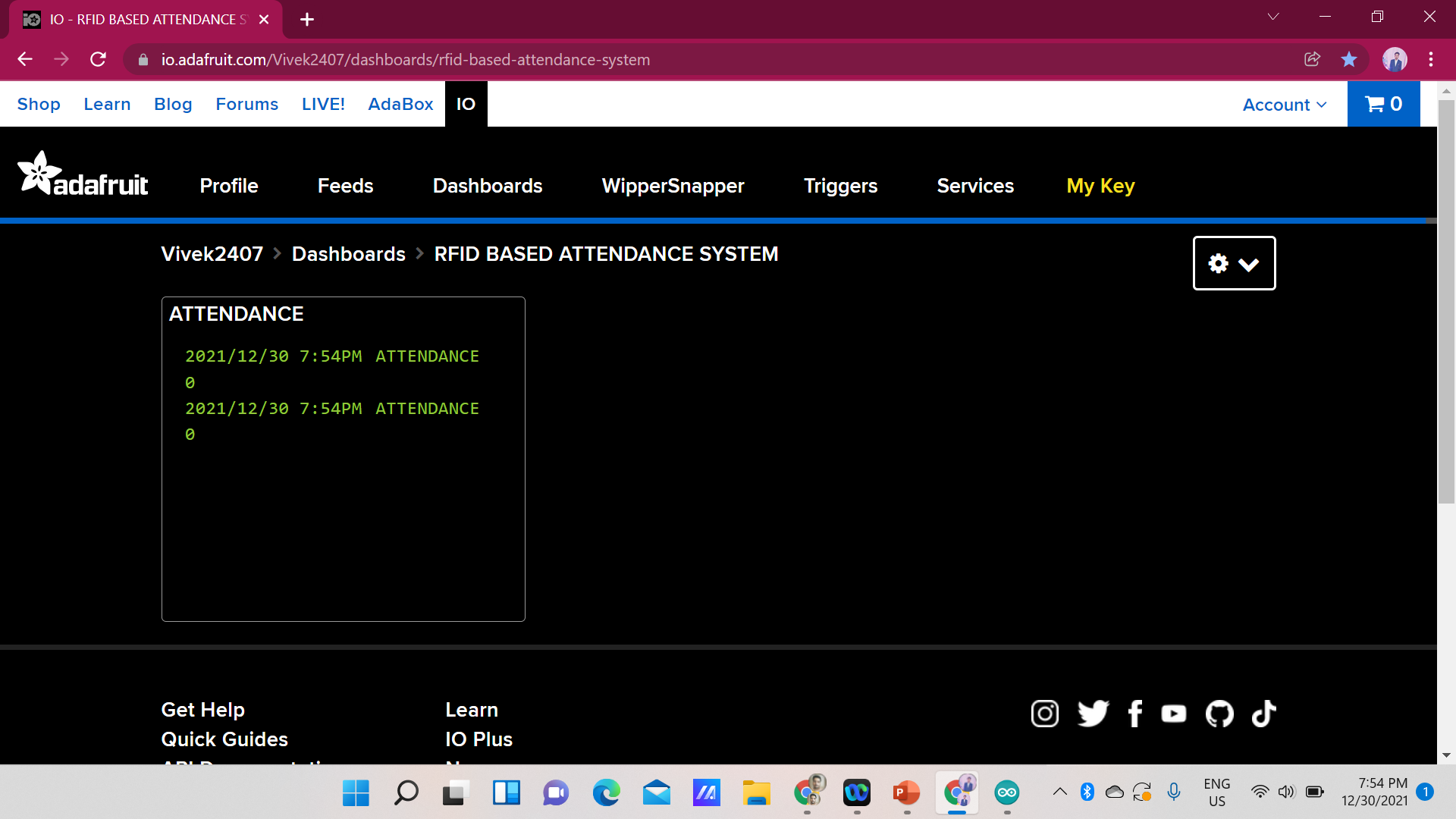
****

****

**RESULTS -**

****

**PUBLISHING TO ADAFRUIT**

****

**CONCLUSION**

The main aim is to get an **effective and accurate** as well as time saving iot enabled attendance system that can record the attendance in real time and at the same time store the record in cloud platform. The proposed system provides more accurate identification as well as **user friendly**, **easy to use** and **cost efficient**.