

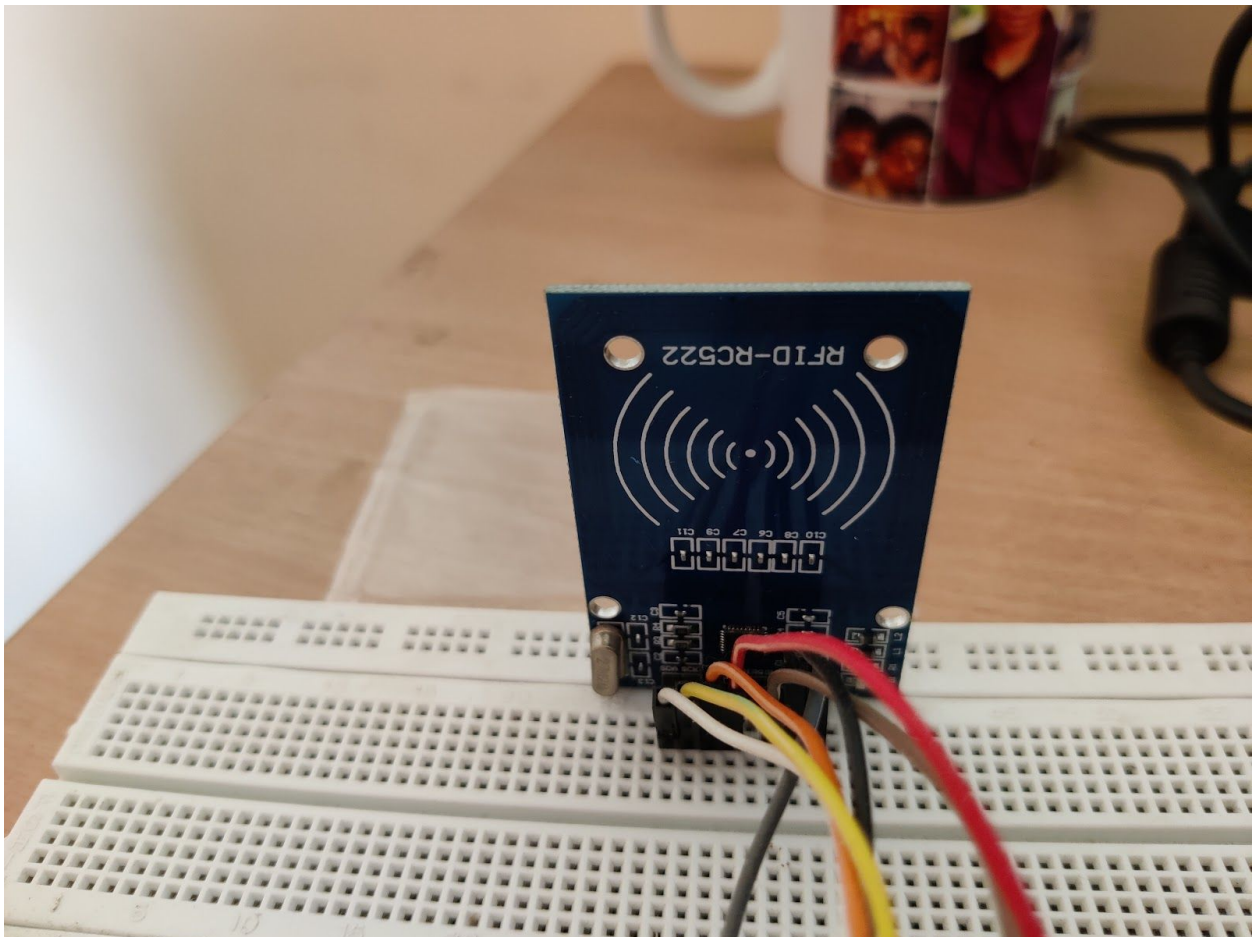
RFID ATTENDANCE SYSTEM

AIM:

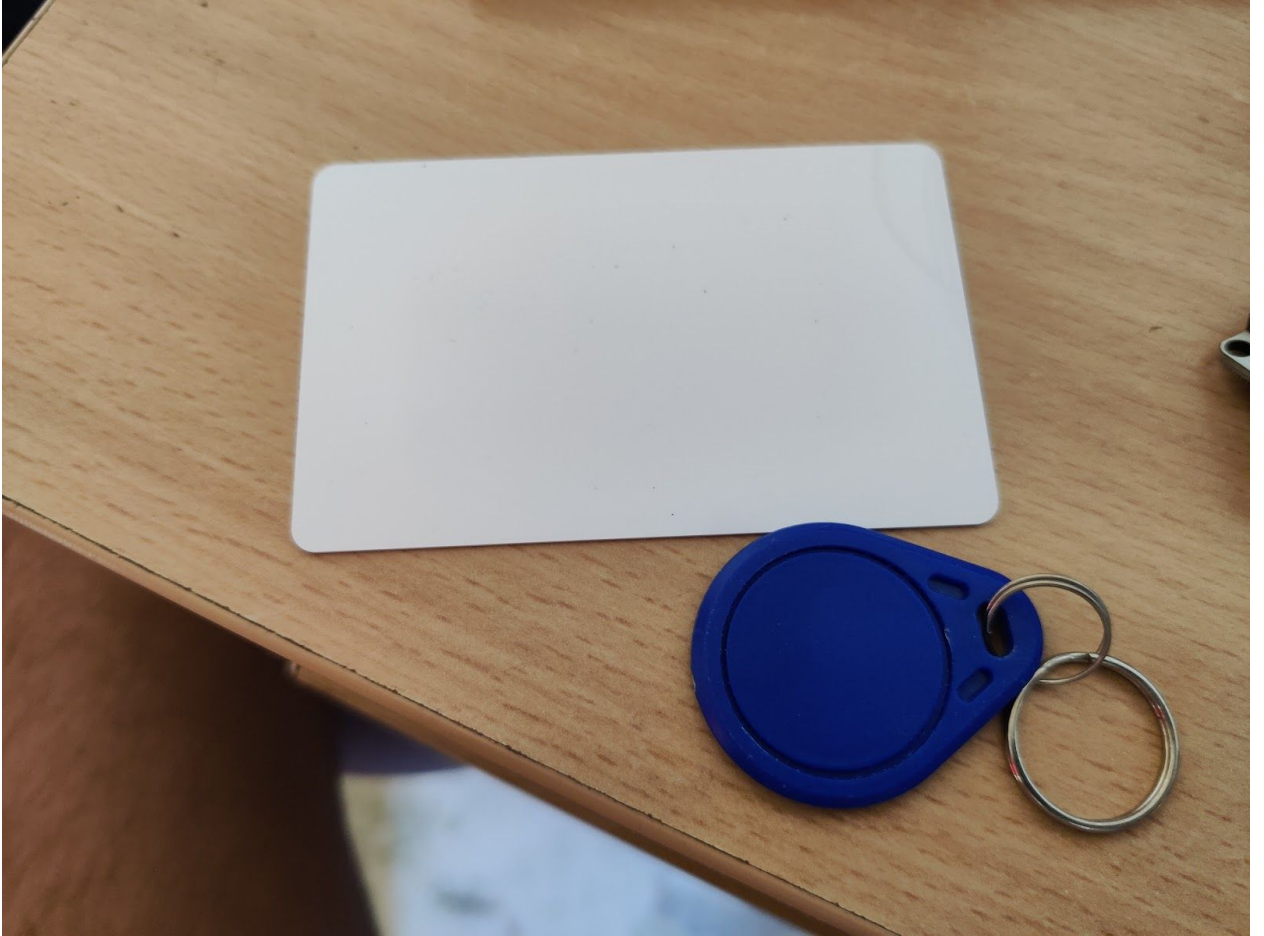
To create an attendance system which uses Radio Frequency Identification sensor(RFID) to notify if a person is present or absent.

Components Used:

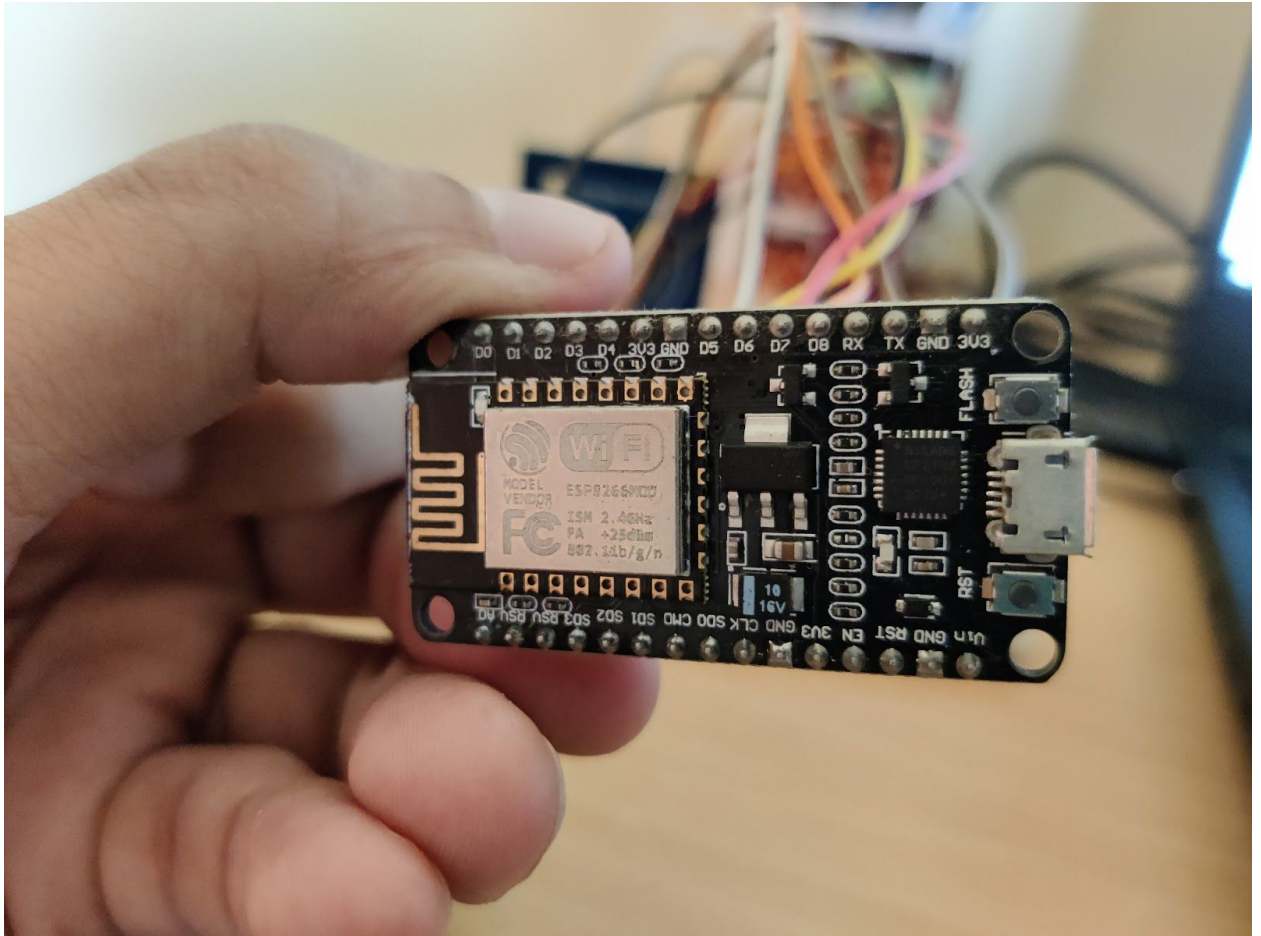
- RFID Sensor



- RFID card or tag



- NodeMCU esp8266

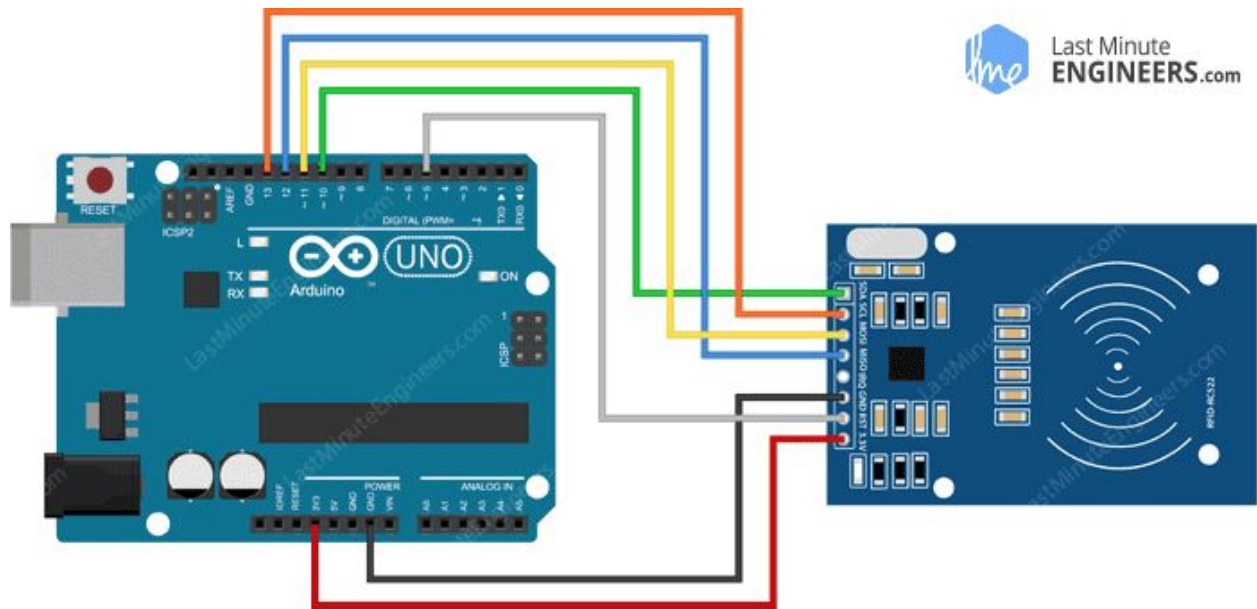


- Breadboard
- Jumper wires

Software Used:

- Arduino Ide (for coding)
- Adafruit.io (Cloud platform to store attendance)

Connections:



RFID Sensor has the following pins:

- **MISO/SCL/Tx:-** It acts as Master-In-Slave-Out when SPI interface is enabled.
- **MOSI (Master Out Slave In):-** It is SPI input to the RC522 module.
- **SCK (Serial Clock):-** It accepts clock pulses provided by the SPI bus Master i.e.NodeMCU.
- **SS/SDA/Rx:-** It acts as Signal input when SPI interface is enabled.

Therefore, the connections to the NodeMCU are as follows:

- RST pin is connected to D3 pin of NodeMCU.
- 3.3 V pin is connected to 3 V pin of NodeMCU.

Project:

STEP1: Obtaining the UID of the RFID card or tag.

- Every RFID card or tag has a specific UID.
- This UID will be detected by our RFID sensor and the output stored for specific UID will be displayed.
- So, in order to find the UID , we use 2 libraries:
 - ❖ <SPI.h>
 - ❖ <MFRC522.h>

SPI:

- Serial Peripheral Interface(SPI) is a synchronous serial data protocol.
- It is basically included because using this header, our NodeMCU will be able to communicate with RFID Sensor quickly through short distances.
- MISO is for sending data to the master.
- MOSI is used for sending data to the peripherals(sensor).
- SCK is the clock pulses which synchronize data transmission generated by the master.
- SS is used by the master to enable and disable devices. When the sensor's SS pin is Low, it communicates with the master. If the pin is High, it ignores the master.

MFRC522:

- This is basically the library which helps in every functioning of the RFID sensor. It is a library specific for RFID sensors.
- Used to read/write RFID tags or cards by the RFID sensor.

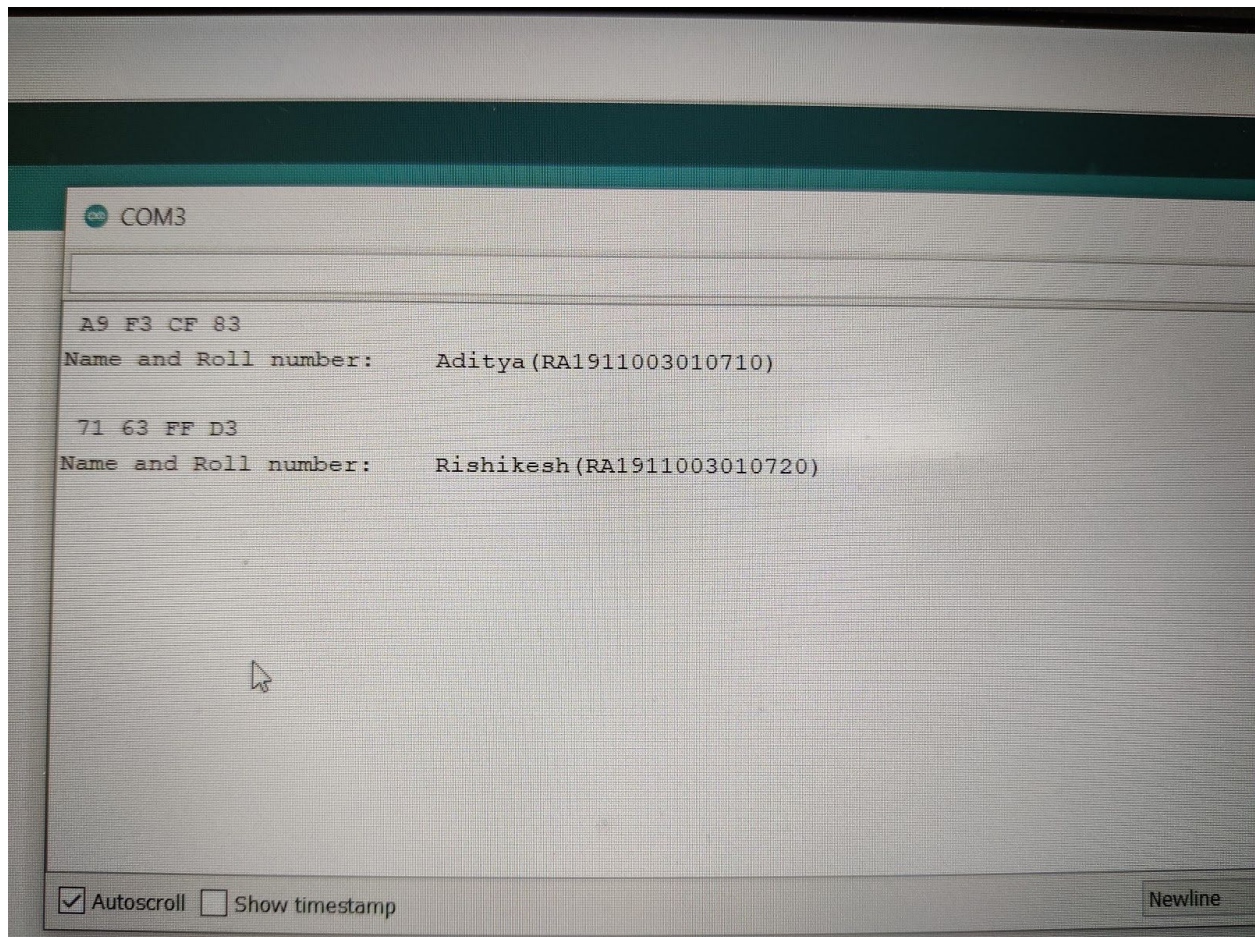
STEP1(Continuation):

- We define the RST and SS pin.
- Then we create a MFRC522 instance using these 2 pins.
- We initialize the SPI bus for communication and MFRC522 for the same.
- We initialize the Serial monitor to visualize the UID.
- So, once we show the RFID card or tag , we get a 64 x 64 array of code.
- We then have to convert this code into an 8 digit UID tag. This UID tag will be used further to make the Attendance system.

For step 1 demo, click on this [link](#).

STEP2:Checking if we are able to print specific output for each card or tag

- In the same manner as the last step, we initialize and define the necessary variables.
- We follow the same code, but this time after the code of our card or tag is converted to UID, we use that specific UID of each card to print a specific Name and Registration number on the serial monitor.
- In this manner, for each card or tag , we will be able to output the name and registration number of the card or tag holder.

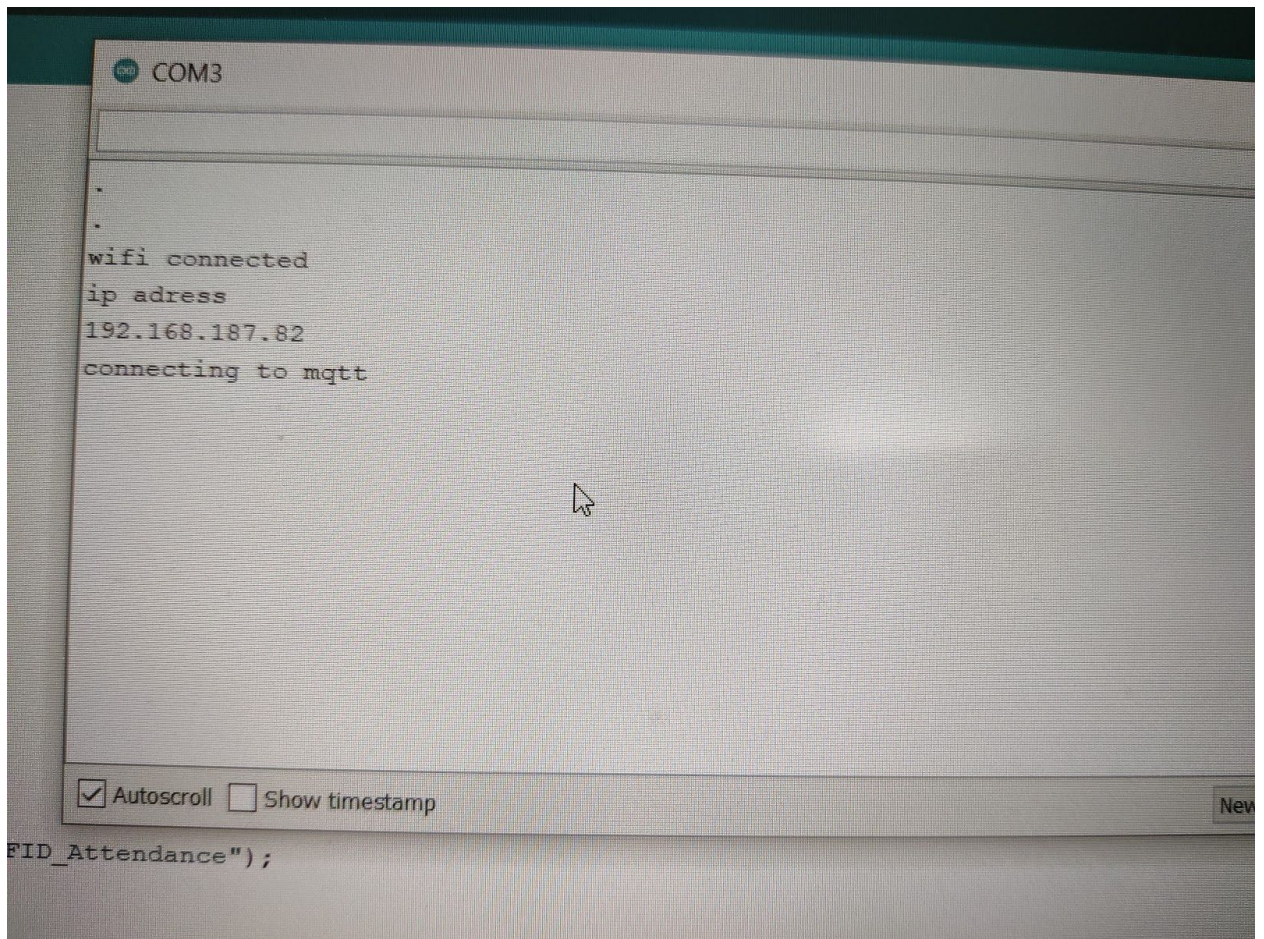


STEP3:Building the final Attendance system

- Now , we must improvise on our previous code. Here, we have to connect our NodeMCU to Adafruit.io cloud platform to store the attendance.
- For this purpose, we use 2 more libraries:
 - ESP8266Wifi
 - Adafruit_MQTT_Client

ESP8266Wifi:

- Our NodeMCU has a wifi module inbuilt in it. Therefore, we can connect our NodeMCU to our local network, i.e. , Wifi and access the internet using our NodeMCU module.



Adafruit_MQTT_Client:

- Once , our NodeMCU is connected to the internet, we must also connect it to the cloud platform so that the output provided, basically the attendance of each RFID card or tag could be uploaded to the cloud platform which faculties or boss can use to keep a check on the attendance of every individual.
- The cloud platform we are using is Adafruit.io which uses MQTT protocol for communication.
- MQTT is a simple publish-subscribe network protocol that transfers messages between devices.

STEP3:(Continuation)

- We have to first go to [Adafruit website](#), create an account if we don't have one. We have to note the API key of our Adafruit account which we'll further use in the code .
- We then have to create a dashboard and then inside the dashboard create a field where we would be uploading the recorded attendance.
- It's better if we use a Stream or Text block to show attendance as we have to display name and registration number.
- Now, coming back to our code, we must define our server and port (1883 default) and also define our Adafruit username and API key.
- Now, we could publish our Adafruit field .
- Now, once our NodeMCU is connected to the internet and our MQTT Client is established, the output which we receive when our card or tag is placed in front of our RFID sensor , could now be published onto our Adafruit field and we could keep a track on the attendance of individuals.

[Profile](#)[Feeds](#)[Dashboards](#)[Triggers](#)

rishusiva > Dashboards > RFID_Attendance

Attendance:

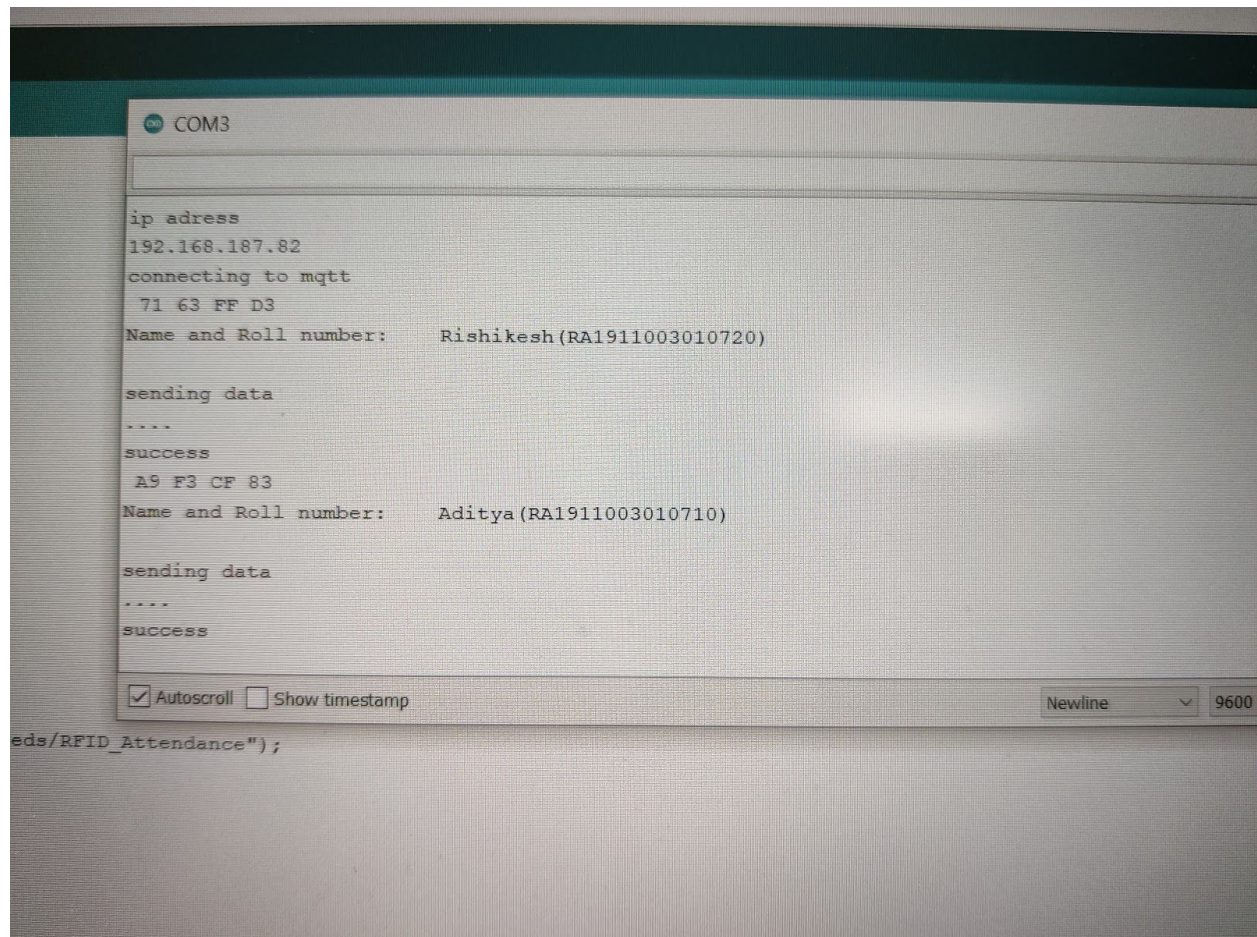
2021/02/25 12:48PM RFID_Attendance
Rishikesh(RA1911003010720) is
present!
2021/02/25 12:49PM RFID_Attendance
Aditya(RA1911003010710) is present!

Attendance:

Aditya(RA1911003010710)
is present!

[Get Help](#)
[Quick Guides](#)

[Learn](#)
[IO Plus](#)



For viewing the final project, click on this [link](#).

APPLICATIONS:

- Could be used for taking attendance during both offline and online classes.
- In case of online classes, every student could publish their attendance on the Feed created by the faculty and the faculty could instantly take a note of the absentees. This is better than calling out each student's name and noting it down manually.

- In case of offline classes, RFID sensors could be put up in the entrance and students could just punch their card on the sensor while coming in. This is an easy way to record attendance.
- Not only in institutions and schools, this could be used in factories and companies too.