Internet of Things Project

RFID-enabled Remote Door Lock System

By:

TANUMON ROY

B. Tech (3/4) – ECE

Section - B

Roll No. - 174259

Submitted to:

Sir K. Ravi Kishore

Associate Professor

ECE Department

NIT Warangal

RFID-Enabled remote door lock system

Tanumon roy [B. Tech (3/4) - ece - 174259]

# introduction:

This project deals with making a RFID-enabled door lock system. As a overview, the RFID sensor on the door reads the UID of the RFID tag given to guest. The host gets an email notification on his/her smartphone which contains the guest’s RFID tag UID as well as the details of the guest. The user have to just open the Blynk App and then choose whether to allow the guest in or not. This activity can be executed remotely from anywhere around the globe.

# Features & Applications:

* This system can be used as a part of Home Automation.
* The system can also be used as a part of Home Security.
* This system is ideal for the office room of a single individual. The user can track and control who comes in the office room.
* The use of this system is very simple from the user’s perspective. The user has to just have to read the “Knock Knock” email notification and then open Blynk App in his/her smartphone and decide whether to open door or not.
* The system uses AWS as a cloud platform to track and notify who comes in the office and when.
* This activity can be executed remotely from anywhere around the globe.

# LIST OF COMPONENTS:

* RC522 RFID reader (Sensor)
* ESP8266 (WiFi Module & controller)
* AWS IoT Core (Cloud Platform)
* Amazon SNS (Notification Service)
* Blynk App (Application Layer)

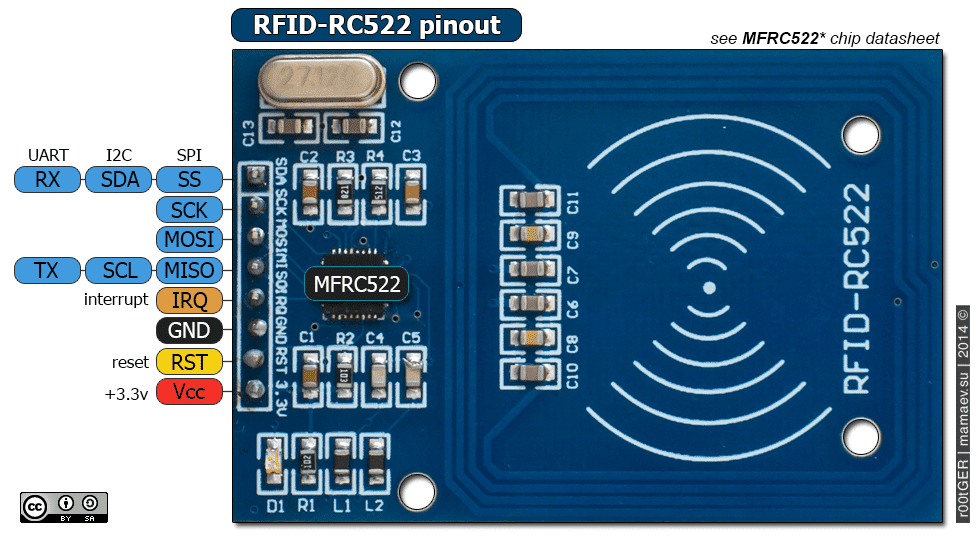
# explanation OF COMPONENTS:

## RFID-RC522:

* RFID is an acronym for “radio-frequency identification” and refers to a technology whereby digital data encoded in RFID tags or smart labels (defined below) are captured by a reader via radio waves.
* RFID tag consists of an integrated circuit and an antenna. The tag is also composed of a protective material that holds the pieces together and shields them from various environmental conditions.

RFID-RC522 reader and tags





RC522 pinout diagram

## ESP8266:

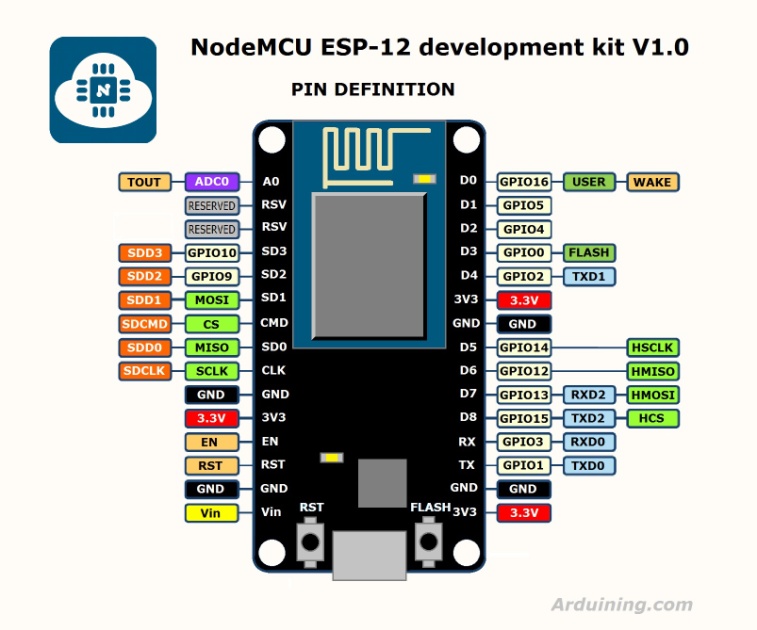
* ESP8266 is a WiFi Module produced by manufacturer Espressif Systems in Shanghai, China.
* It is a self contained SOC with integrated TCP/IP protocol stack
* It can give any microcontroller access to the WiFi network

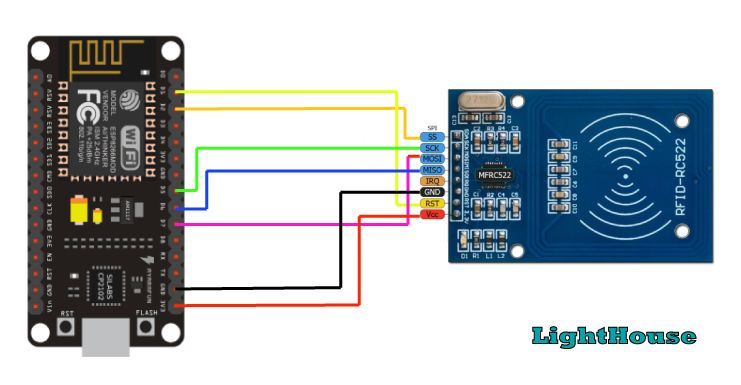
Features:

* 802.11 b/g/n
* Wi-Fi Direct (P2P), soft-AP
* Integrated TCP/IP protocol stack
* Integrated TR switch, balun, LNA, power amplifier and matching network
* Integrated PLLs, regulators, DCXO and power management units
* +19.5dBm output power in 802.11b mode
* Power down leakage current of <10uA
* 1MB Flash Memory
* Integrated low power 32-bit CPU could be used as application processor
* SDIO 1.1 / 2.0, SPI, UART
* STBC, 1×1 MIMO, 2×1 MIMO
* A-MPDU & A-MSDU aggregation & 0.4ms guard interval
* Wake up and transmit packets in < 2ms
* Standby power consumption of < 1.0mW (DTIM3)

APPLICATIONS:

* ESP8266 module comes pre-programmed with an AT command set firmware, meaning, we can simply hook this up to our Arduino device and get about as much WiFi-ability as a WiFi Shield offers
* This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime
* Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area.

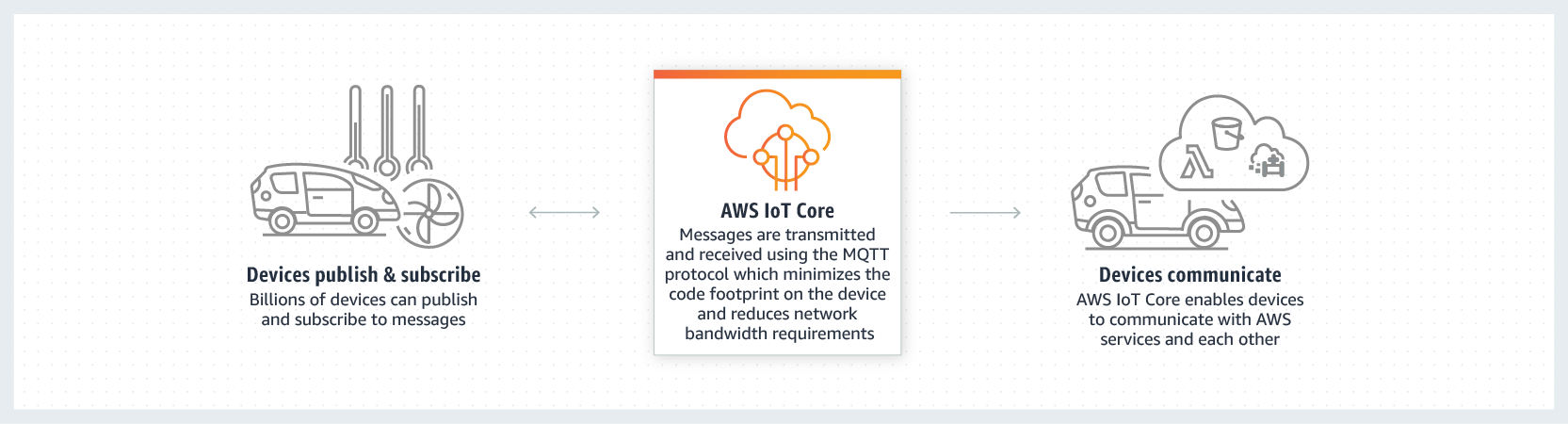




Connecting RC522 to ESP8266

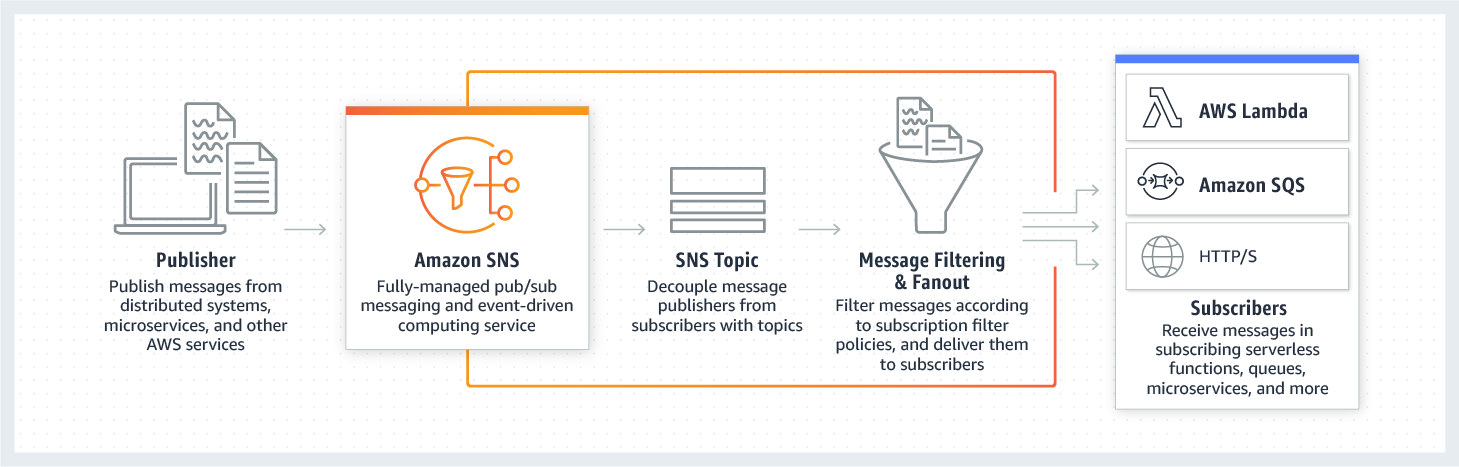
## AWS IOT CORE:

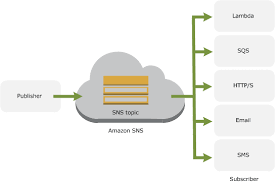
* AWS IoT Core is a managed cloud service that lets connected devices easily and securely interact with cloud applications and other devices.
* AWS IoT Core can support billions of devices and trillions of messages, and can process and route those messages to AWS endpoints and to other devices reliably and securely.
* With AWS IoT Core, applications can keep track of and communicate with all other related devices, all the time, even when they aren’t connected.
* Connected devices, such as sensors, actuators, embedded devices, smart appliances, and wearable devices, connect to AWS IoT Core over HTTPS, WebSockets, or secure MQTT. Included in AWS IoT Core is a Device Gateway that allows secure, low-latency, low-overhead, bi-directional communication between connected devices and your cloud and mobile applications.



## AMAZON SNS:

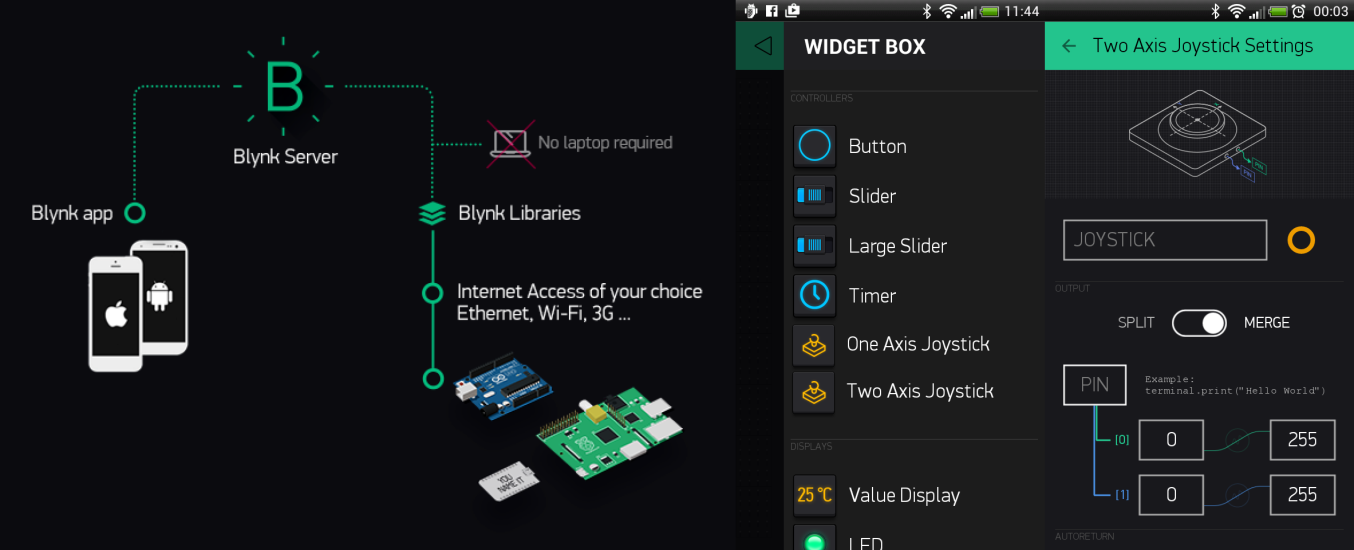
* Amazon Simple Notification Service (SNS) is a highly available, durable, secure, fully managed pub/sub messaging service that enables you to decouple microservices, distributed systems, and serverless applications. Amazon SNS provides topics for high-throughput, push-based, many-to-many messaging.





## BLYNK APP:

* Blynk is a Platform with IOS and Android apps to control Arduino, Raspberry Pi, ESP8266 and the likes over the Internet. It’s a digital dashboard where you can build a graphic interface for our project by simply dragging and dropping widgets.

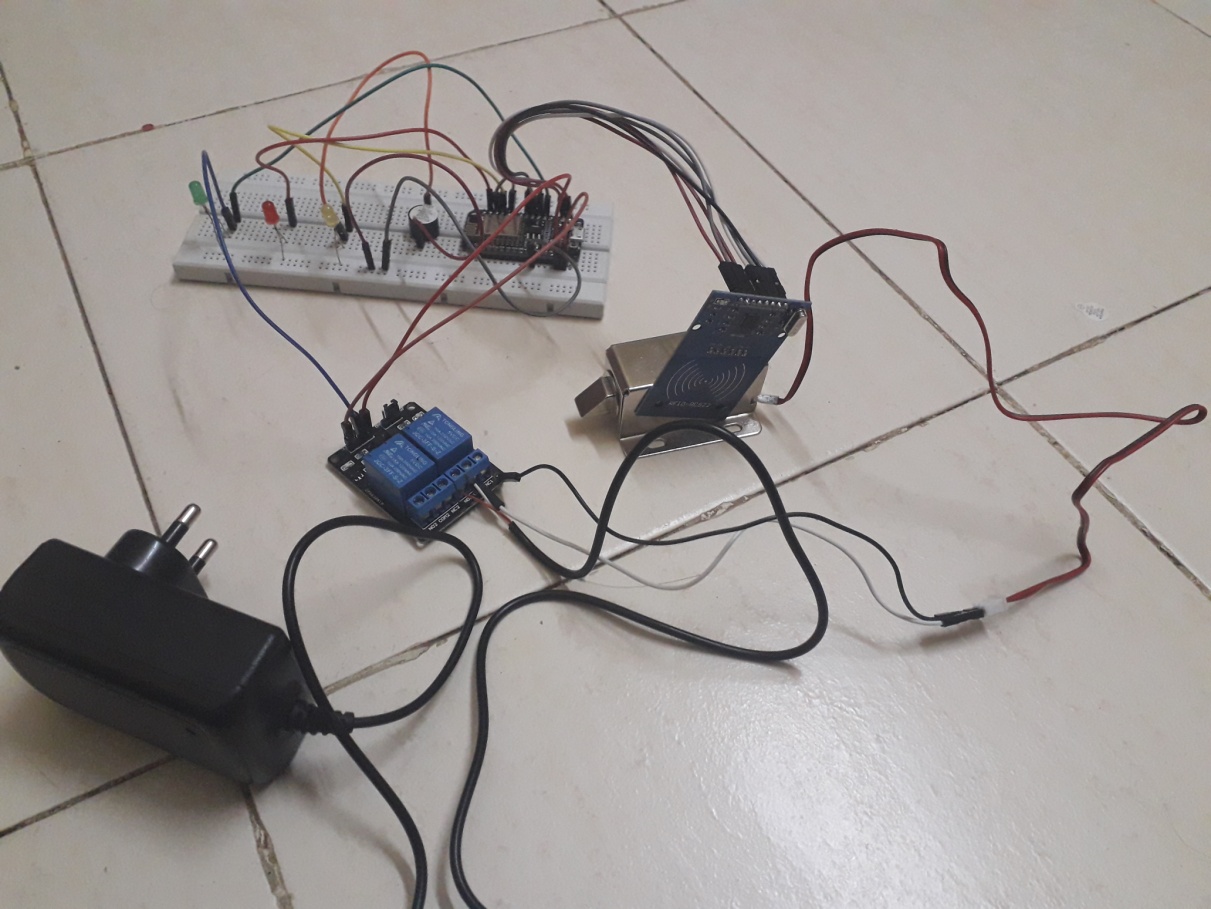


# DESIGN FLOW:

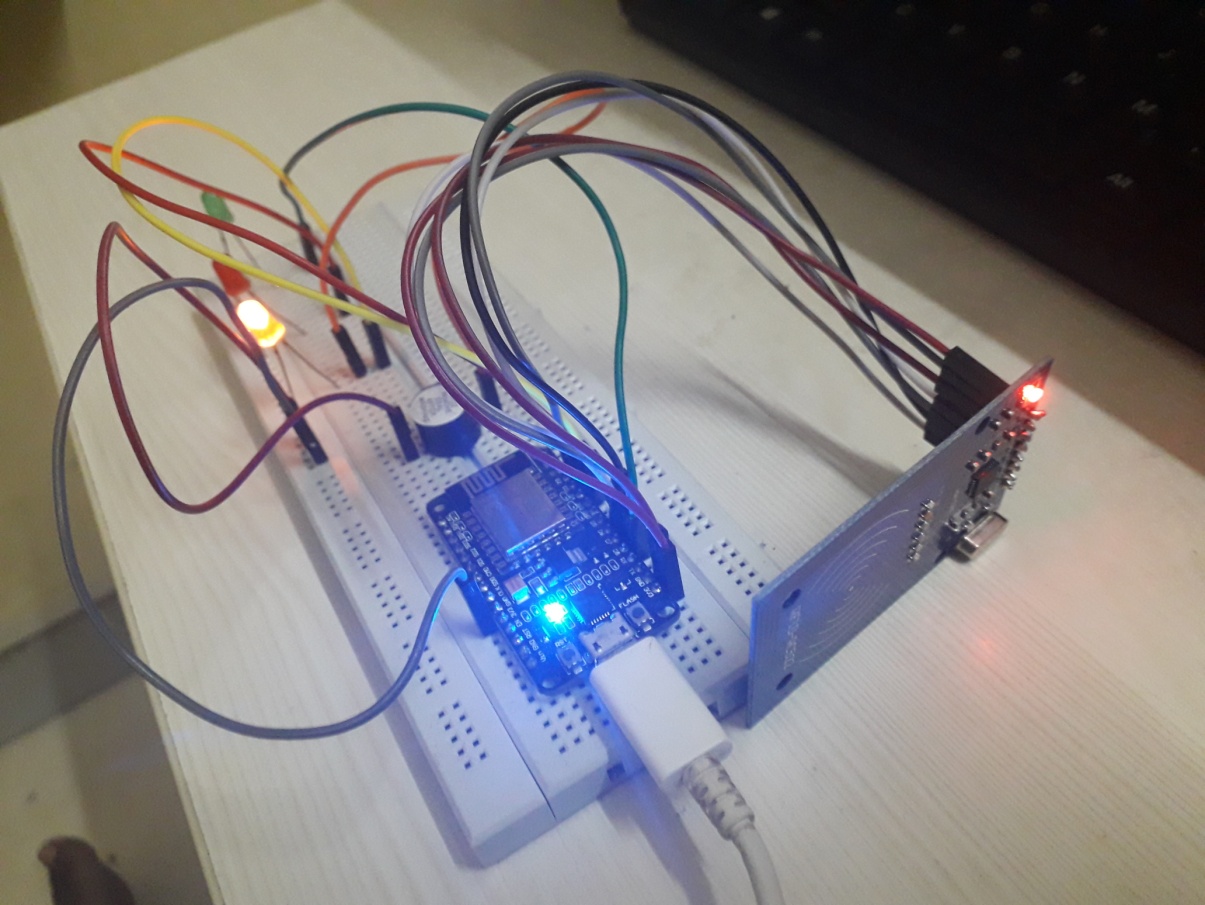
# POSSIBLE MODIFICATIONS & IMPROVEMENTS:

* A master-tag can be formed for the host which will give instant access without any user interaction from the host required.
* More details about the guests can be included in the Arduino code other than the tag UIDs.
* Similar design can be made for any person going out of the door, opening the door as soon as the same RFID tag is shown. The exit time may also be uploaded on the cloud platform and analysed.
* The application layer can be made by using some artificial “chatbot” as in Telegram. For doing this, the payload message has to be parsed into .py from .json using AWS Lambda functions along with Dynamo DB. Then a “chatbot” and channel has to be created in Telegram. The HTTP API link to this “chatbot” is used as the endpoint for Amazon SNS.
* The opening of the door can also be made voice-controlled by introducing Google Assistant. For that we have to use IFTTT and set the trigger as a specific command said by the user in Google Assistant.
* This project can also be done using Adafruit.io as the MQTT service, IFTTT and Telegram*. (But, it was seen that it takes much more time for the user to receive the notification. Hence this design is not preferred)*

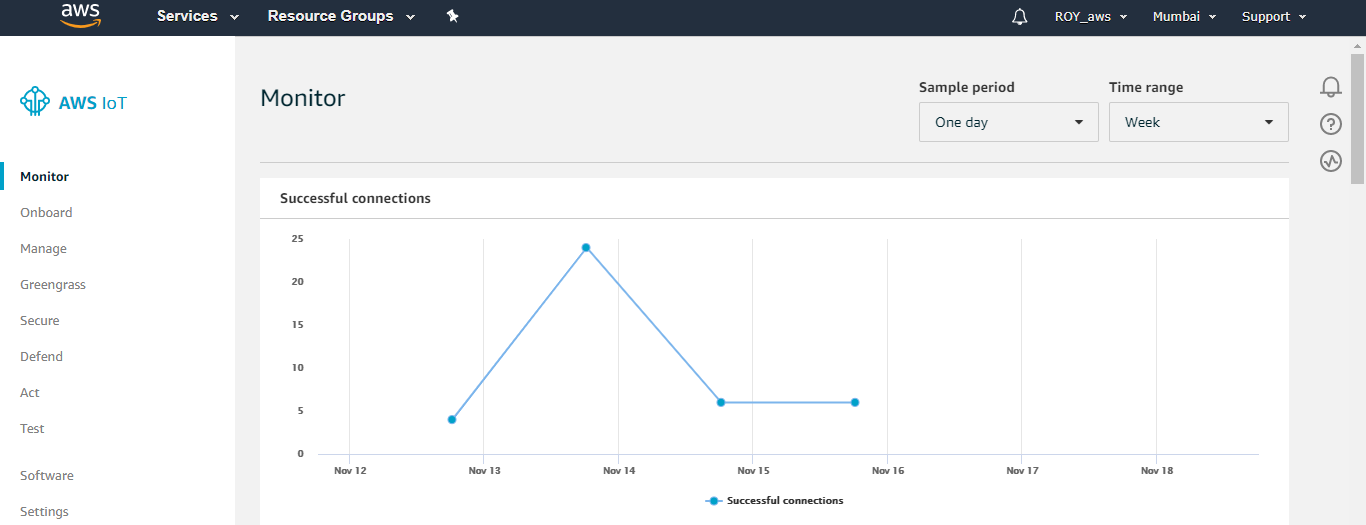
# RELATED SNAPSHOTS:

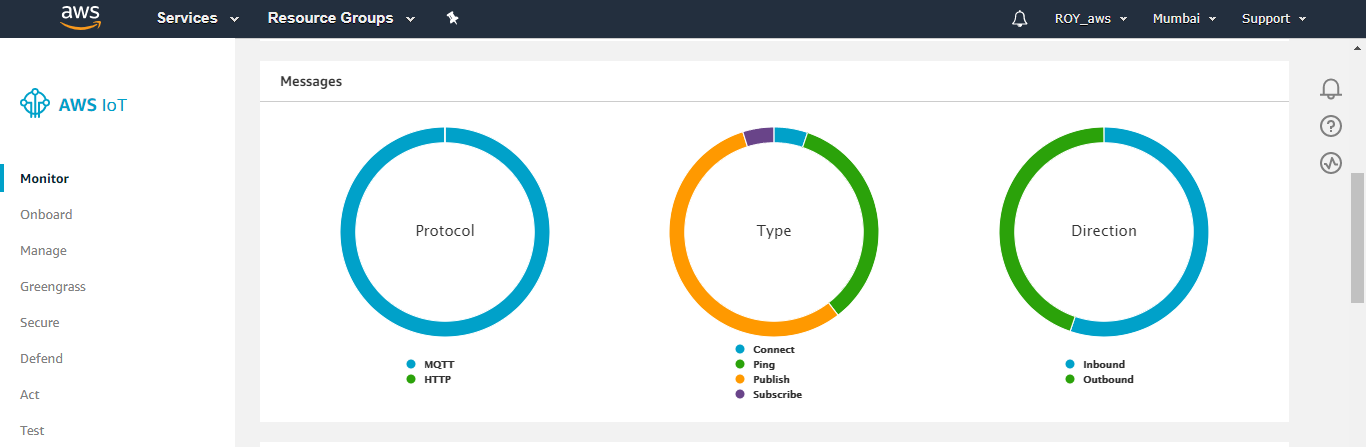


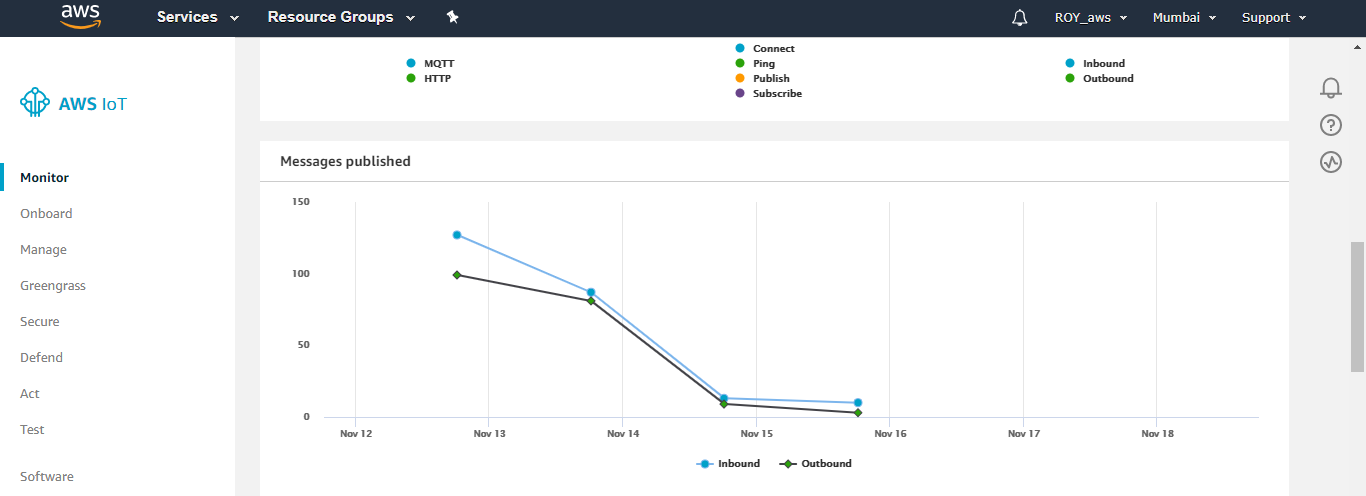
Overall Circuit

**

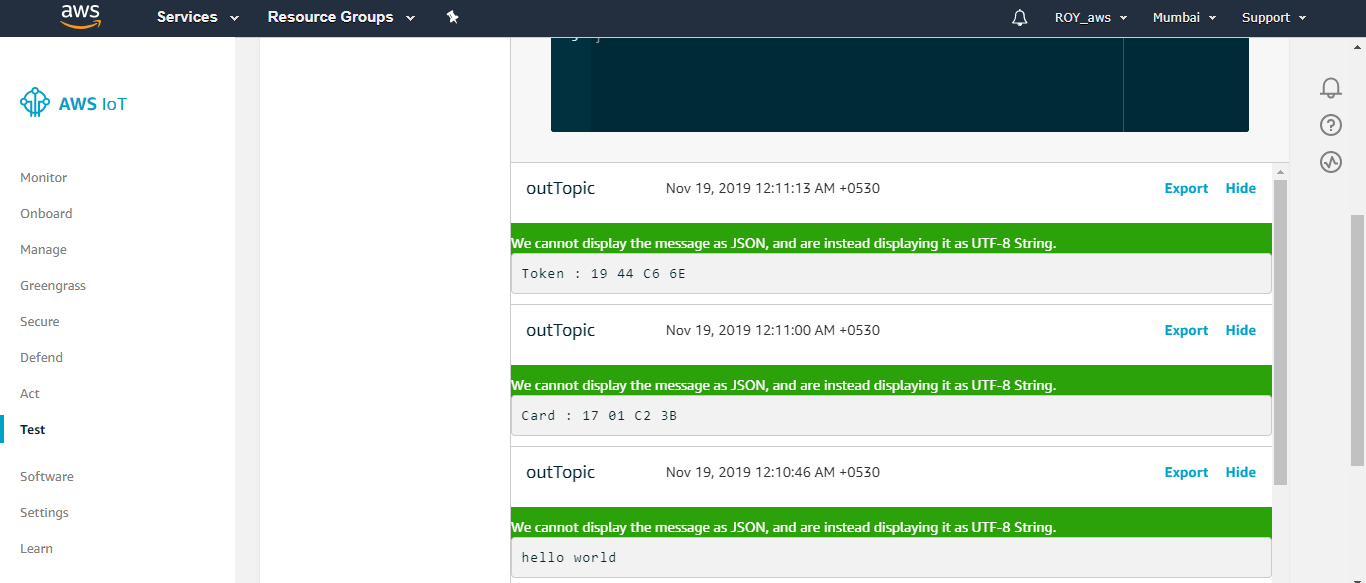
Circuit after RFID tag is shown

**

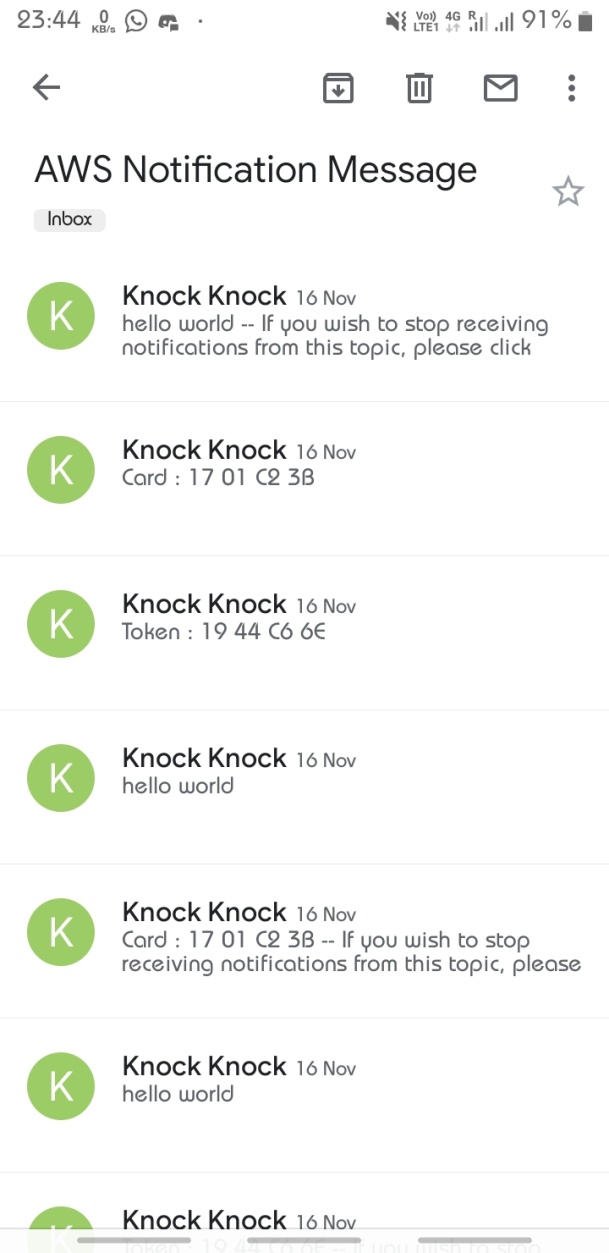
**

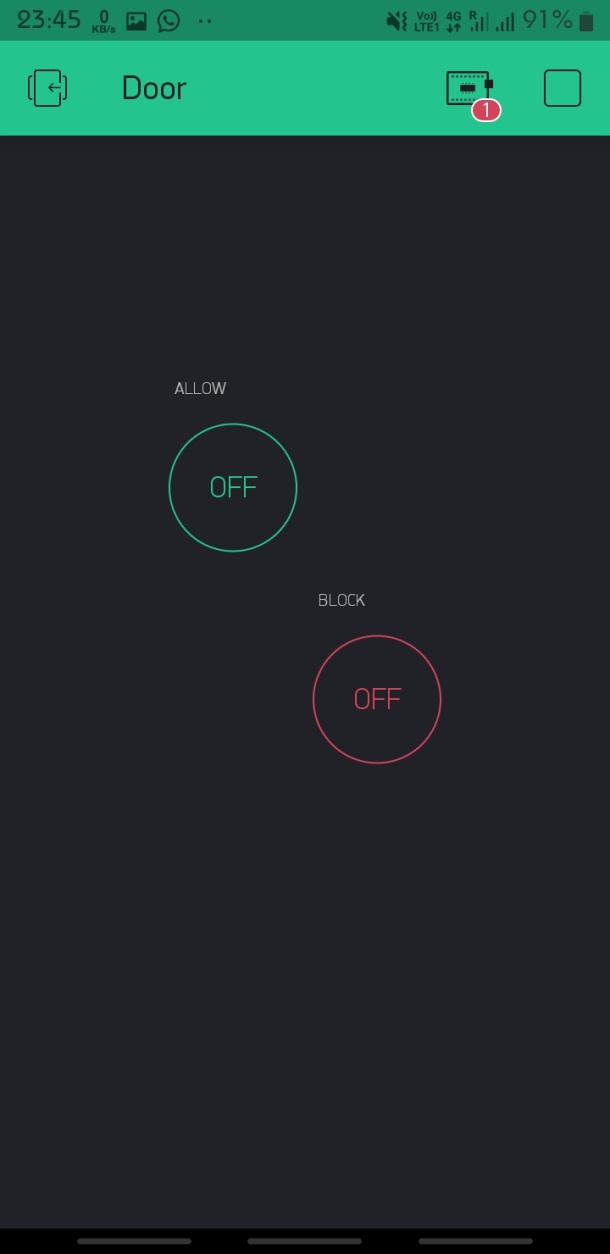
**

AWS Messages’ Statistics

**

Messages sent to AWS MQTT Topic “outTopic”

**Email Notifications

**Blynk App Dashboard

# REFERENCE URL:

* https://www.youtube.com/watch?v=sRgosRd32eA
* https://aws.amazon.com/iot-core/
* https://aws.amazon.com/sns/
* https://docs.blynk.cc/
* https://github.com/esp8266/arduino-esp8266fs-plugin
* http://bit.ly/Electronics\_Innovation