RFID BASED AUTO-GATE SYSTEM

INTRODUCTION:

The proposed solution is aimed to automate a private parking.

To do so we will be using RFID Tags, those RFID tags will be placed on specific car's license plates and whenever the car will be close enough to the gate, the RFID reader will read from the card and either grant access or deny it.

When the car will be identified, the gate will open automatically and send the name of the client via the Blynk app, those data will then be collected and used for billing.

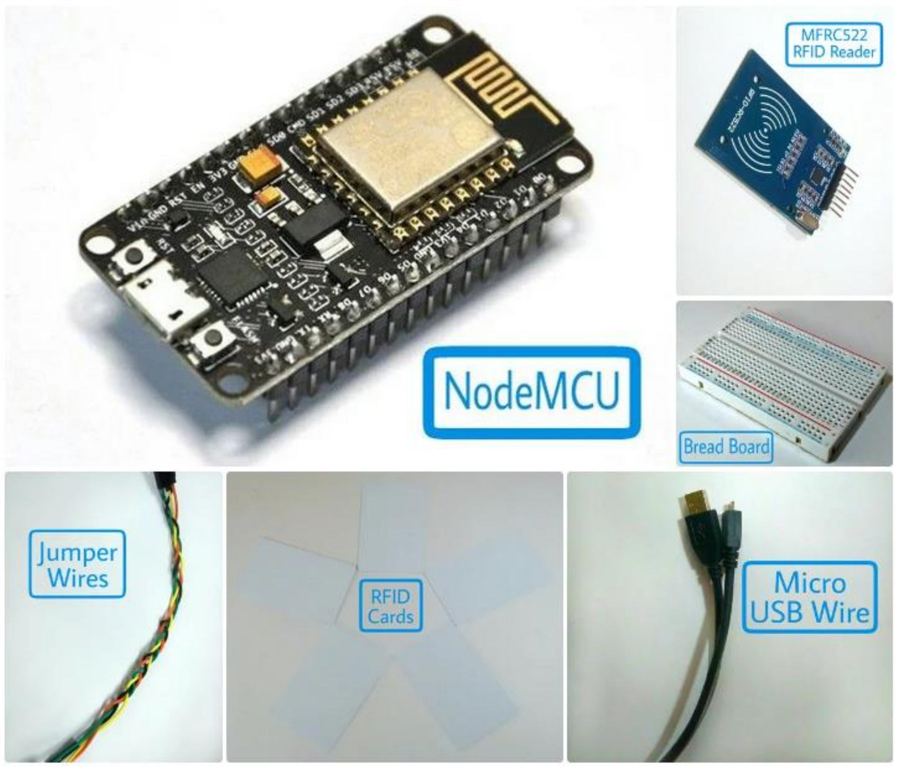
In this project, we will be using simple RFID cards which will act as the tags we intend on placing on license plates and we will also be using servo motor which will simulate the gate opening and closing mechanism by its rotation.

HARDWARE COMPONENTS:

1. NodeMCU
2. MFRC522 RFID Reader
3. RFID Tags
4. Servo Motor
5. LED
6. Bread Board
7. Jumper Wires
8. Micro USB Cable

SOFTWARE COMPONENTS:

1. Arduino IDE
2. Blynk app



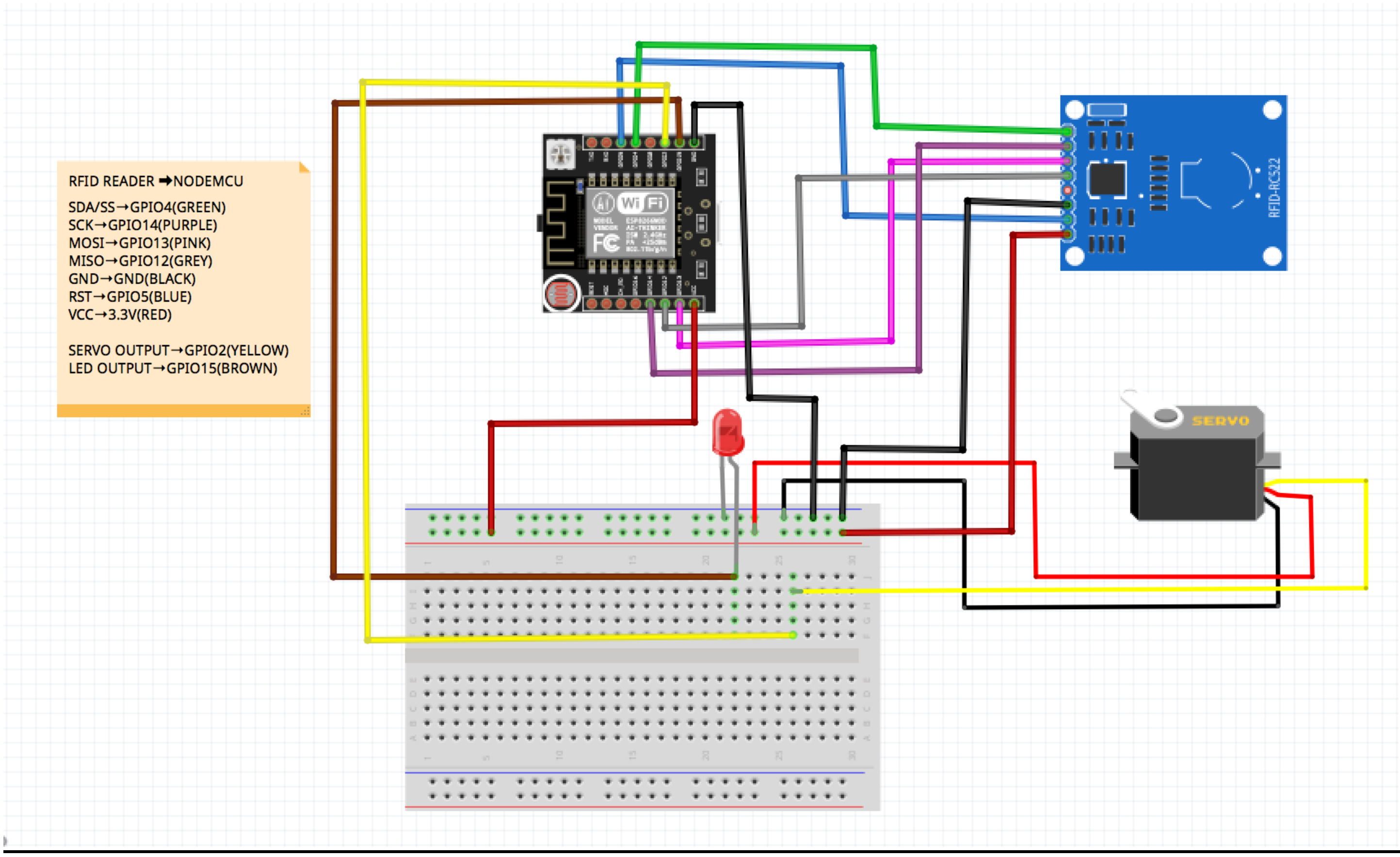
CIRCUIT CONNECTIONS:

NodeMCU	MFRC522
D4 (GPIO4)	SDA (SS)
D5 (GPIO14)	SCK
D7 (GPIO13)	MOSI
D6 (GPIO12)	MISO (SCL)
	IRQ
GND	GND
D3 (GPIO0)	RST (FLASH)
3V	3.3V

NodeMCU

GPIO2	SERVO MOTOR
GPIO15	LED

CIRCUIT DIAGRAM:



RFID READER ➡NODEMCU

SDA/SS→GPIO4(GREEN)
SCK→GPIO14(PURPLE)
MOSI→GPIO13(PINK)
MISO→GPIO12(GREY)
GND→GND(BLACK)
RST→GPIO5(BLUE)
VCC→3.3V(RED)

SERVO OUTPUT→GPIO2(YELLOW)
LED OUTPUT→GPIO15(BROWN)

GETTING STARTED WITH THE CODING:

1. Library download:

Before you download the library, you need Arduino IDE to get started.

- Download Arduino IDE
- Download the RFID library from GitHub
- Unzip the RFID library.
- Install the RFID library in your Arduino IDE.
- Restart your Arduino IDE
- Open Preferences window.
- Enter http://arduino.esp8266.com/stable/package_esp8266com_index.json into Additional Board Manager URLs field. You can add multiple URLs, separating them with commas.
- Open Boards Manager from Tools > Board menu and find esp8266 platform.
- Select the version you need from a drop-down box.
- Click install button.
- Don't forget to select your ESP8266 board from Tools > Board menu after installation.

2. Reading data from a RFID Tag

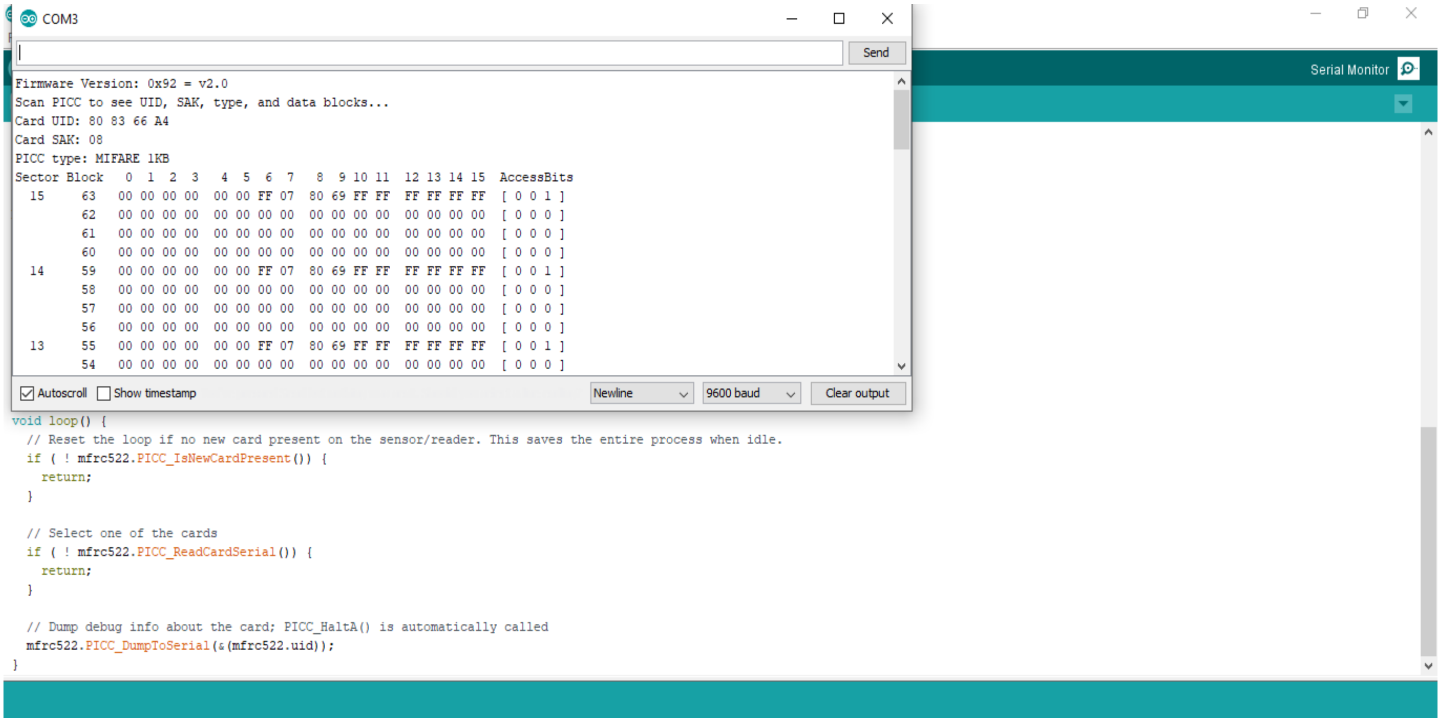
After having the circuit ready

- Go to File > Examples > MFRC522 > DumpInfo > Upload the code.

This code will be available in your Arduino IDE (after installing the RFID library).

- Then, open the serial monitor. You should see something like the image below.
- Write down your UID card (" Card UID : xx xx xx xx ") because you'll need it later.

The code in the dumpinfo file will allow you to see the unique identification number for each RFID Tag you will use. The need of writing down the UID is that for our project we want to provide access to some vehicles which means we need a way to identify them and because each tag has its own UID we will be able to specify in our codes the UID for which we want to grant access to.



```
Firmware Version: 0x92 = v2.0
Scan PICC to see UID, SAK, type, and data blocks...
Card UID: 80 83 66 A4
Card SAK: 08
PICC type: MIFARE 1KB
Sector Block   0  1  2  3   4  5  6  7   8  9 10 11  12 13 14 15  AccessBits
  15     63   00 00 00 00  00 00 FF 07  80 69 FF FF  FF FF FF FF  [ 0 0 1 ]
         62   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
         61   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
         60   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
  14     59   00 00 00 00  00 00 FF 07  80 69 FF FF  FF FF FF FF  [ 0 0 1 ]
         58   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
         57   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
         56   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
  13     55   00 00 00 00  00 00 FF 07  80 69 FF FF  FF FF FF FF  [ 0 0 1 ]
         54   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
```

☑ Autoscroll ☐ Show timestamp Newline 9600 baud Clear output

```
void loop() {
  // Reset the loop if no new card present on the sensor/reader. This saves the entire process when idle.
  if ( ! mfrc522.PICC_IsNewCardPresent()) {
    return;
  }

  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial()) {
    return;
  }

  // Dump debug info about the card; PICC_HaltA() is automatically called
  mfrc522.PICC_DumpToSerial(&(mfrc522.uid));
}
```

FINAL PROJECT CODE:

The code below grants or denies access to some of the tags.

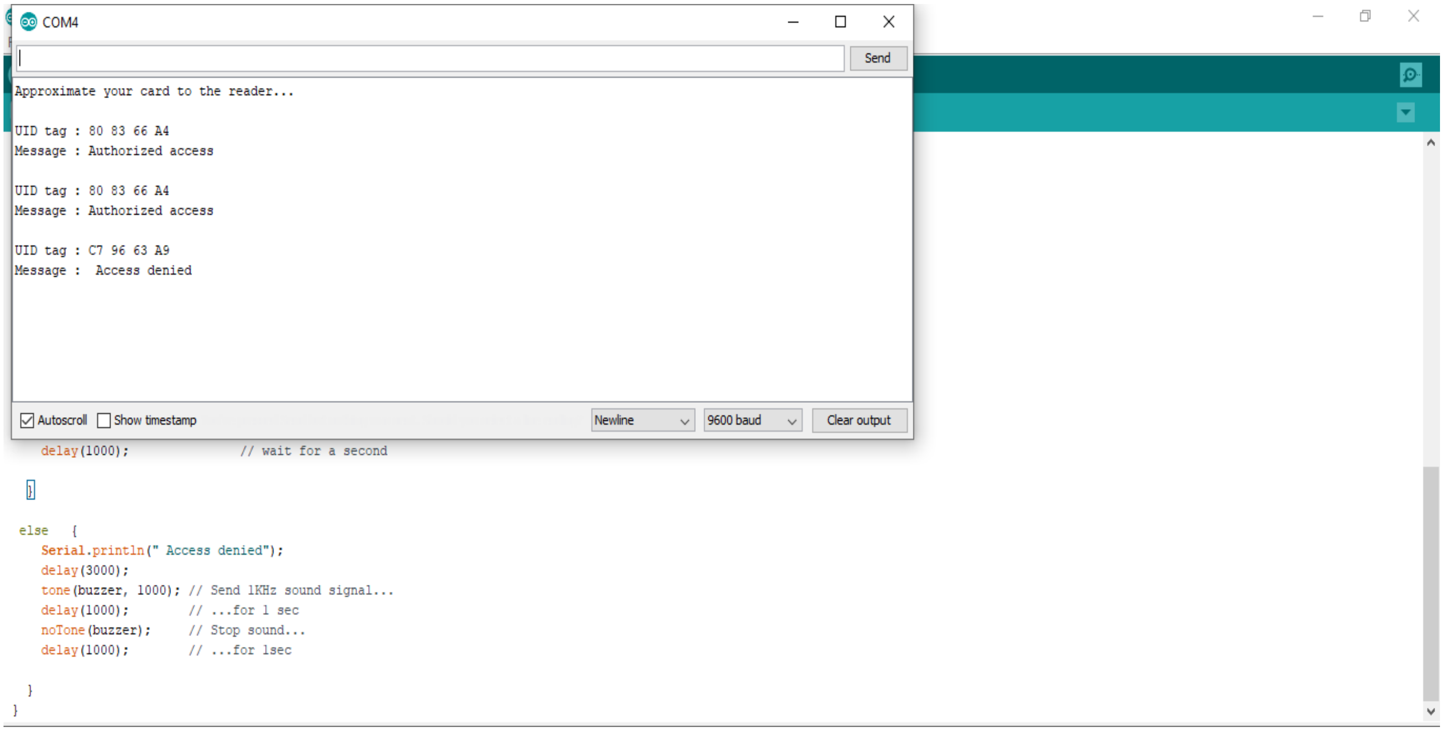
Tags with access will enable the servo motor to rotate while simulating a gate opening and closing mechanism, a message “Authorized access” will be viewed on the serial monitor of the Arduino IDE and the name of the client on which the card has been registered to will be displayed on the Blynk app linked by the code.

Tags without access will make the LED blink as an alert signal, a message “Access denied” will be viewed on the serial monitor of the Arduino IDE and “DENIED” will also be displayed on the Blynk app linked by the code.

```
#include <MFRC522.h>

#include <Servo.h>

#include <SPI.h>

#include <ESP8266WiFi.h>

Servo servo;

#include <BlynkSimpleEsp8266.h>
```

```
#define SS_PIN 4
#define RST_PIN 5


char auth[] = "a0a47eb6e3ce472e8371f8b72b14b820";
char ssid[] = "Sarai";
char pass[] = "123456789";

int led = 15;
MFRC522 mfrc522(SS_PIN, RST_PIN);   // Create MFRC522 instance.

void setup()
{

  Serial.begin(115200);
  servo.attach(2);
  servo.write(0);
  delay(2000);

  Blynk.begin(auth, ssid, pass);
  SPI.begin();      // Initiate  SPI bus
  mfrc522.PCD_Init();   // Initiate MFRC522
  Serial.println("Approximate your card to the reader...");
  Serial.println();

  pinMode(led, OUTPUT);
}

void loop()
```

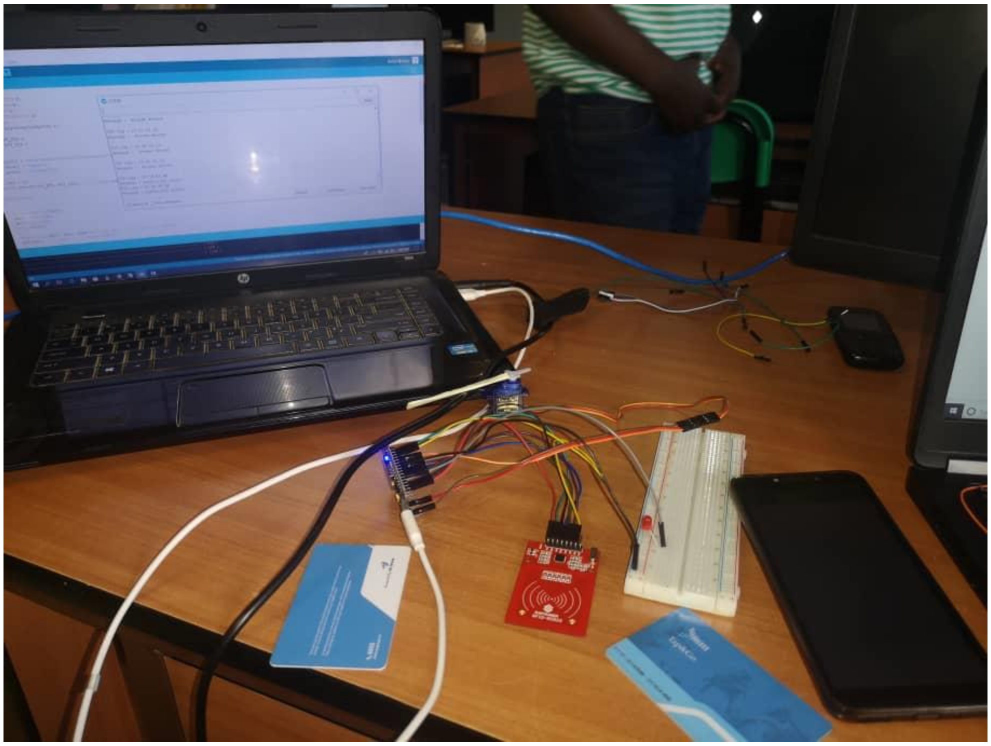
```
{
  Blynk.run();

if ( ! mfrc522.PICC_IsNewCardPresent())
  {
    return;
  }
  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial())
  {
    return;
  }
  //Show UID on serial monitor
  Serial.print("UID tag :");
  String content= "";
  byte letter;
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {
     Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
     Serial.print(mfrc522.uid.uidByte[i], HEX);
     content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
     content.concat(String(mfrc522.uid.uidByte[i], HEX));
  }
  Serial.println();
  Serial.print("Message : ");
  content.toUpperCase();
  if (content.substring(1) == "DC 5D A4 BB") //change here the UID of the card/cards that you want to
give access
  {
    Serial.println("Authorized access");
    Blynk.virtualWrite(V5,"BRICE");
    servo.write(90);
    delay(3000);
    servo.write(0);
  }
  else if (content.substring(1) == "EC C1 C6 35") //change here the UID of the card/cards that you want to give access
  {
    Serial.println("Authorized access");
    Blynk.virtualWrite(V5,"SARAH");
    servo.write(90);
    delay(3000);
    servo.write(0);
  }

 else  {
    Serial.println(" Access denied");
    Blynk.virtualWrite(V5,"DENIED");
    Serial.println();
    delay(3000);

    digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
    delay(500);               // wait for a second
    digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
    delay(500);               // wait for a second



  }


}
```

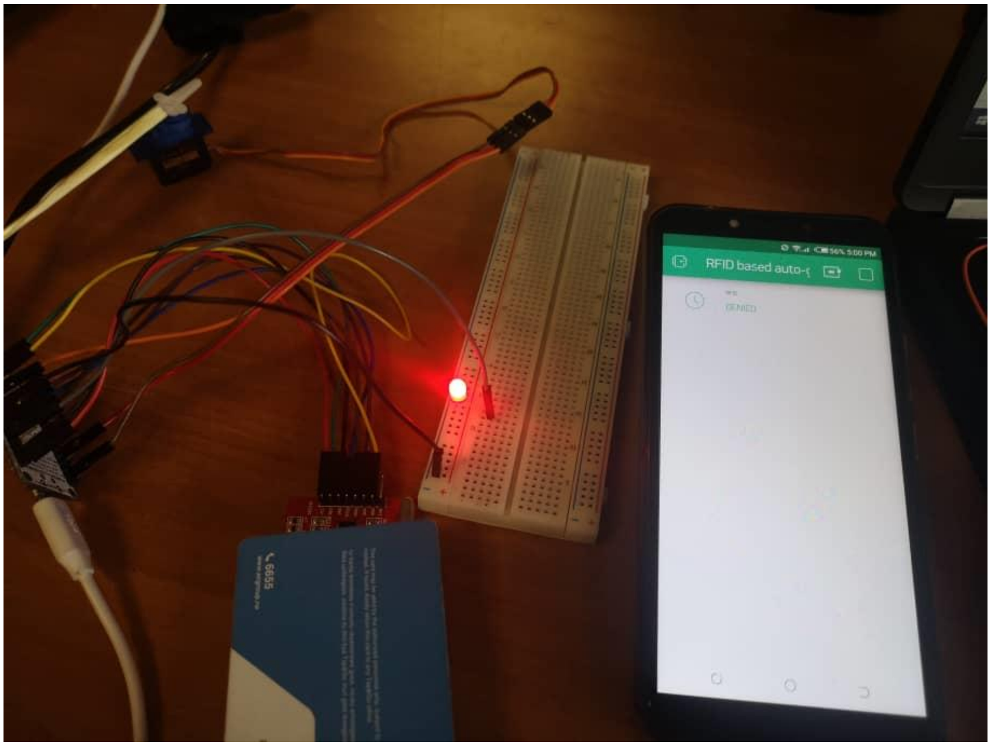


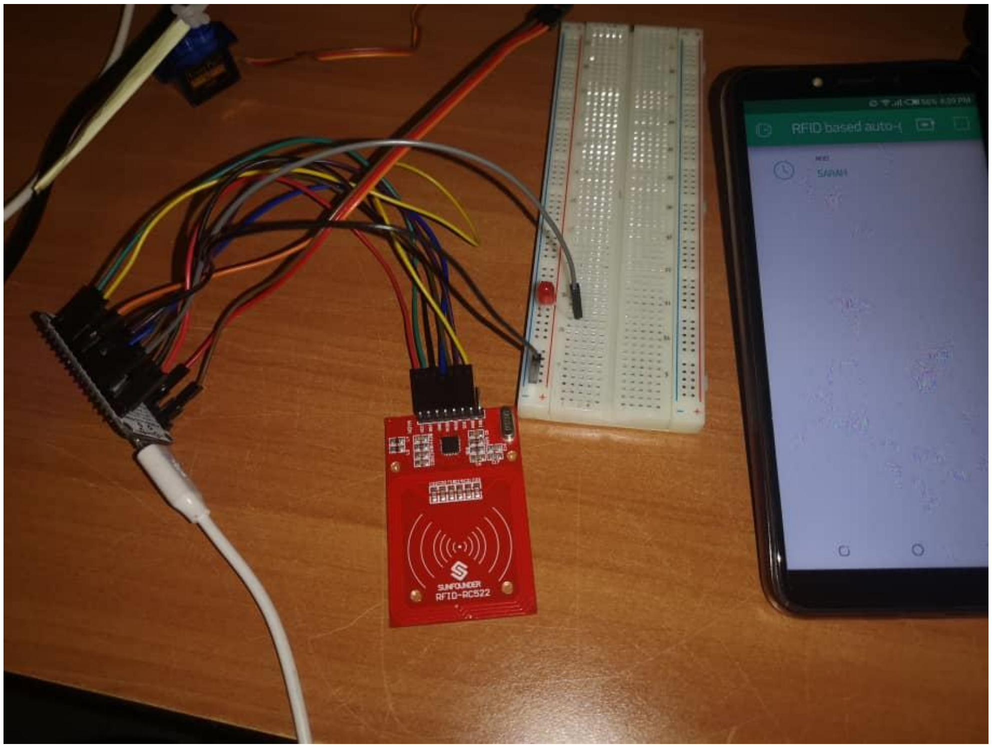
```
Approximate your card to the reader...

UID tag : 80 83 66 A4
Message : Authorized access

UID tag : 80 83 66 A4
Message : Authorized access

UID tag : C7 96 63 A9
Message :  Access denied
```

```
    delay(1000);              // wait for a second

  }

 else   {
    Serial.println(" Access denied");
    delay(3000);
    tone(buzzer, 1000); // Send 1KHz sound signal...
    delay(1000);        // ...for 1 sec
    noTone(buzzer);     // Stop sound...
    delay(1000);        // ...for 1sec

  }
}
```

PROJECT IMPLEMENTATION IMAGES:




RFID based auto-(
DENIED


RFID based auto-(
SARAH
SUNFOUNDER
RFID-RC522