# RFID UNIVERSAL KEY

A single Key solution using RFID via ESP32

Project by:

P NAGA LAKSHMI NARAYANA SARMA

Karunya Institute of technology and sciences.

### INTRODUCTION

RFID UNIVERSAL KEY relies on the concept of simplyfing the daily use of different types of KEYS and IDcards by replacing it with RFID cards. In this process we can reduce the usage of different KEYS such as House keys, vehicle keys, office keys etc.

Similarly we can replace our all conventional IDs such as Aadhar card, Driving Licence, Office/College IDs etc with RFID in turn to solves the problem of storage and increases the convinence of accessibility.

## OBJECTIVE

Aim of this project is to come up with a prototype of a RFID system interfaced via ESP32 to make 3 Basic stages of the project.

Stage1:RFID Access System

Stage2:RFID Personalized switch

Stage3: RFID Universal Identification

**How Does RFID Work?**

Radio Frequency Identification (RFID) is the wireless non-contact use of radio frequency waves to transfer data. Tagging items with [RFID tags](https://www.atlasrfidstore.com/rfid-tags/?utm_source=RFID-Beginners-Guide&utm_medium=eBook&utm_campaign=Content&utm_content=rfid-tags) allows users to automatically and [uniquely identify](https://blog.atlasrfidstore.com/what-is-rfid?utm_source=RFID-Beginners-Guide&utm_medium=eBook&utm_campaign=Content&utm_content=what-is-rfid) and track inventory and assets. RFID takes auto-ID technology to the next level by allowing tags to be read without line of sight and, depending on the type of RFID, having a read range between a few centimeters to over 20+ meters.

RFID has come a long way from its first application of identifying airplanes as friend or foe in World War II. Not only does the technology continue to improve year over year, but the cost of implementing and using an RFID system continues to decrease, making RFID more cost-effective and efficient.

# COMPONENTS

1. ESP32
2. BASIC SHIELD
3. RFID CARDS
4. RFID SENSOR
5. SERVO MOTOR
6. JUMPER WIRES

# BLOCK DIAGRAM

# BLOCK

STAGE 1: ACCESS SYSTEM

In the ACCESS SYSTEM the RFID sensor is interfaced with ESP32 using

SPI communication protocol. Here servo motor and LED is connected with ESP32 and is triggered with RFID card.

Selected RFID Card address is stored in the ESP32 which gets triggered when that particular card id showed to the RFID sensor

The RFID sensor scans the address and compares with the stored address.

When matched the LED turns green showing access GRANTED and then the servo motor works indicating opening of the doors.

Thus this system gives access to the person with Authorised RFID.

ACCESS GRANTED CODE:

#include <SPI.h>

#include <MFRC522.h>

#include <Servo.h>

constexpr uint8\_t RST\_PIN = 5; // Configurable, see typical pin layout above

constexpr uint8\_t SS\_PIN = 4; // Configurable, see typical pin layout above

static const int servoPin = 21; //servo motor output

Servo servo1;

MFRC522 rfid(SS\_PIN, RST\_PIN); // Instance of the class

MFRC522::MIFARE\_Key key;

// Init array that will store new NUID

byte nuidPICC[4];

int arr\_b[] = {219,218,138,53};

int arr\_a[] = {203,57,99,54};

void setup() {

pinMode(2,OUTPUT); //green bulb

pinMode(14,OUTPUT); //red bulb when it is denined

pinMode(22,OUTPUT); //red bulb standard

Serial.begin(9600);

servo1.attach(servoPin);

SPI.begin(); // Init SPI bus

rfid.PCD\_Init(); // Init MFRC522

for (byte i = 0; i < 6; i++) {

key.keyByte[i] = 0xFF;

}

Serial.println(F("This code scan the MIFARE Classsic NUID."));

Serial.print(F("Using the following key:"));

printHex(key.keyByte, MFRC522::MF\_KEY\_SIZE);

}

void loop() {

digitalWrite(22,HIGH);

// Look for new cards

if ( ! rfid.PICC\_IsNewCardPresent())

return;

// Verify if the NUID has been readed

if ( ! rfid.PICC\_ReadCardSerial())

return;

Serial.print(F("PICC type: "));

MFRC522::PICC\_Type piccType = rfid.PICC\_GetType(rfid.uid.sak);

Serial.println(rfid.PICC\_GetTypeName(piccType));

// Check is the PICC of Classic MIFARE type

if (piccType != MFRC522::PICC\_TYPE\_MIFARE\_MINI &&

piccType != MFRC522::PICC\_TYPE\_MIFARE\_1K &&

piccType != MFRC522::PICC\_TYPE\_MIFARE\_4K) {

Serial.println(F("Your tag is not of type MIFARE Classic."));

return;

}

if (rfid.uid.uidByte[0] == arr\_a[0] &&

rfid.uid.uidByte[1] == arr\_a[1] &&

rfid.uid.uidByte[2] ==arr\_a[2] &&

rfid.uid.uidByte[3] == arr\_a[3] )

{

Serial.println(F("GRANTED."));

digitalWrite(22,LOW);

delay(2);

digitalWrite(2,HIGH);

for(int posDegrees = 0; posDegrees <= 180; posDegrees++) {

servo1.write(posDegrees);

Serial.println(posDegrees);

delay(1);

}

delay(3000);

digitalWrite(2,LOW);

delay(2);

digitalWrite(22,HIGH);

for(int posDegrees = 180; posDegrees >= 0; posDegrees--) {

servo1.write(posDegrees);

Serial.println(posDegrees);

delay(1);

}

}

else if (rfid.uid.uidByte[0] == arr\_b[0] &&

rfid.uid.uidByte[1] == arr\_b[1] &&

rfid.uid.uidByte[2] ==arr\_b[2] &&

rfid.uid.uidByte[3] == arr\_b[3] )

{

Serial.println(F("GRANTED."));

digitalWrite(2,HIGH);

delay(2);

digitalWrite(22,LOW);

for(int posDegrees = 0; posDegrees <= 180; posDegrees++) {

servo1.write(posDegrees);

Serial.println(posDegrees);

delay(1);

}

delay(3000);

digitalWrite(2,LOW);

delay(2);

digitalWrite(22,HIGH);

for(int posDegrees = 180; posDegrees >= 0; posDegrees--) {

servo1.write(posDegrees);

Serial.println(posDegrees);

delay(1);

}

}

else

{Serial.println(F("DENINED"));

digitalWrite(14,HIGH);

delay(2);

digitalWrite(22,HIGH);

delay(3000);

digitalWrite(14,LOW);

delay(2);

digitalWrite(22,HIGH);

}

// Halt PICC

rfid.PICC\_HaltA();

// Stop encryption on PCD

rfid.PCD\_StopCrypto1();

}

/\*\*

\* Helper routine to dump a byte array as hex values to Serial.

\*/

void printHex(byte \*buffer, byte bufferSize) {

for (byte i = 0; i < bufferSize; i++) {

Serial.print(buffer[i] < 0x10 ? " 0" : " ");

Serial.print(buffer[i], HEX);

}

}

/\*\*

\* Helper routine to dump a byte array as dec values to Serial.

\*/

void printDec(byte \*buffer, byte bufferSize) {

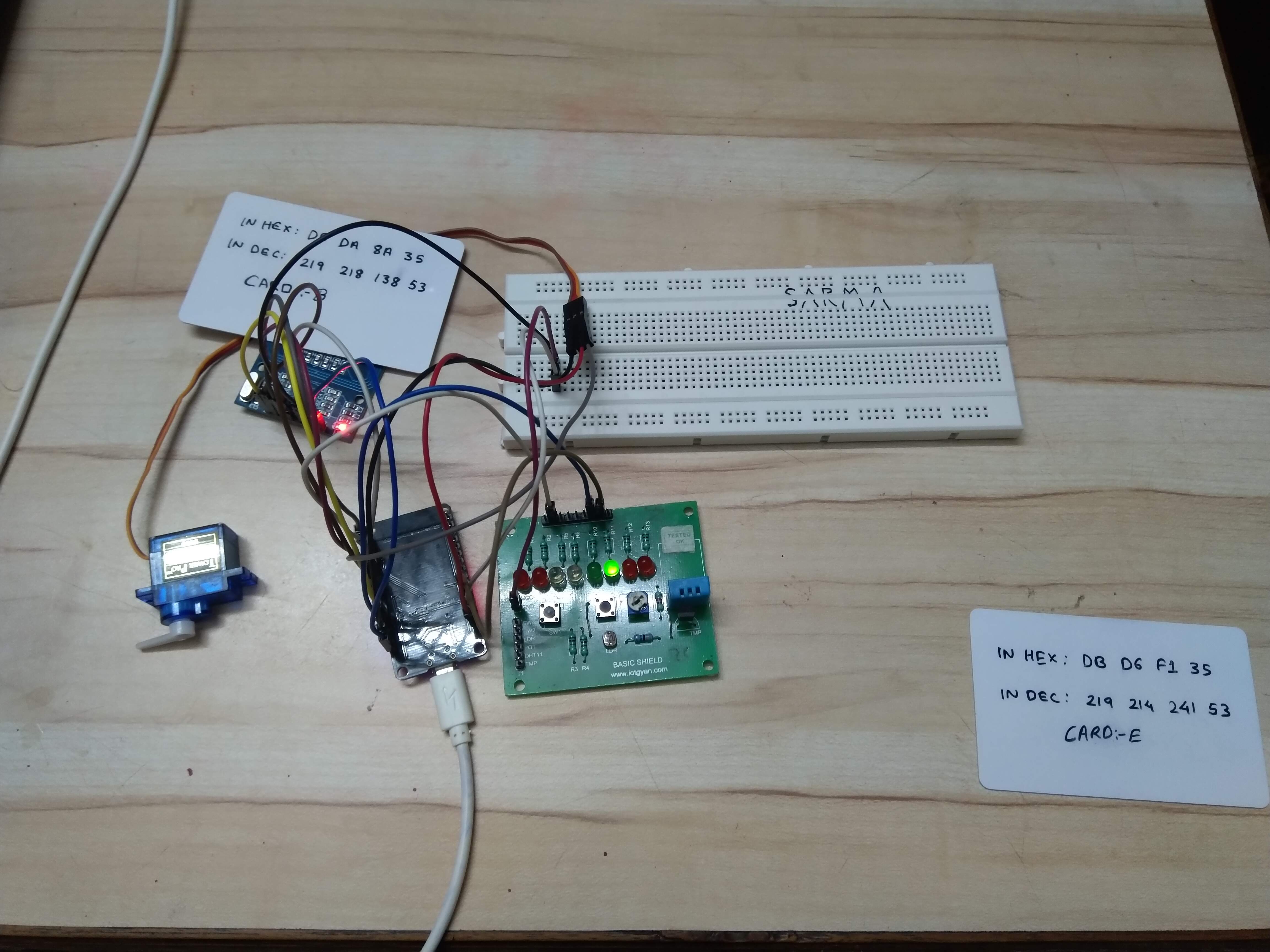
for (byte i = 0; i < bufferSize; i++) {

Serial.print(buffer[i] < 0x10 ? " 0" : " ");

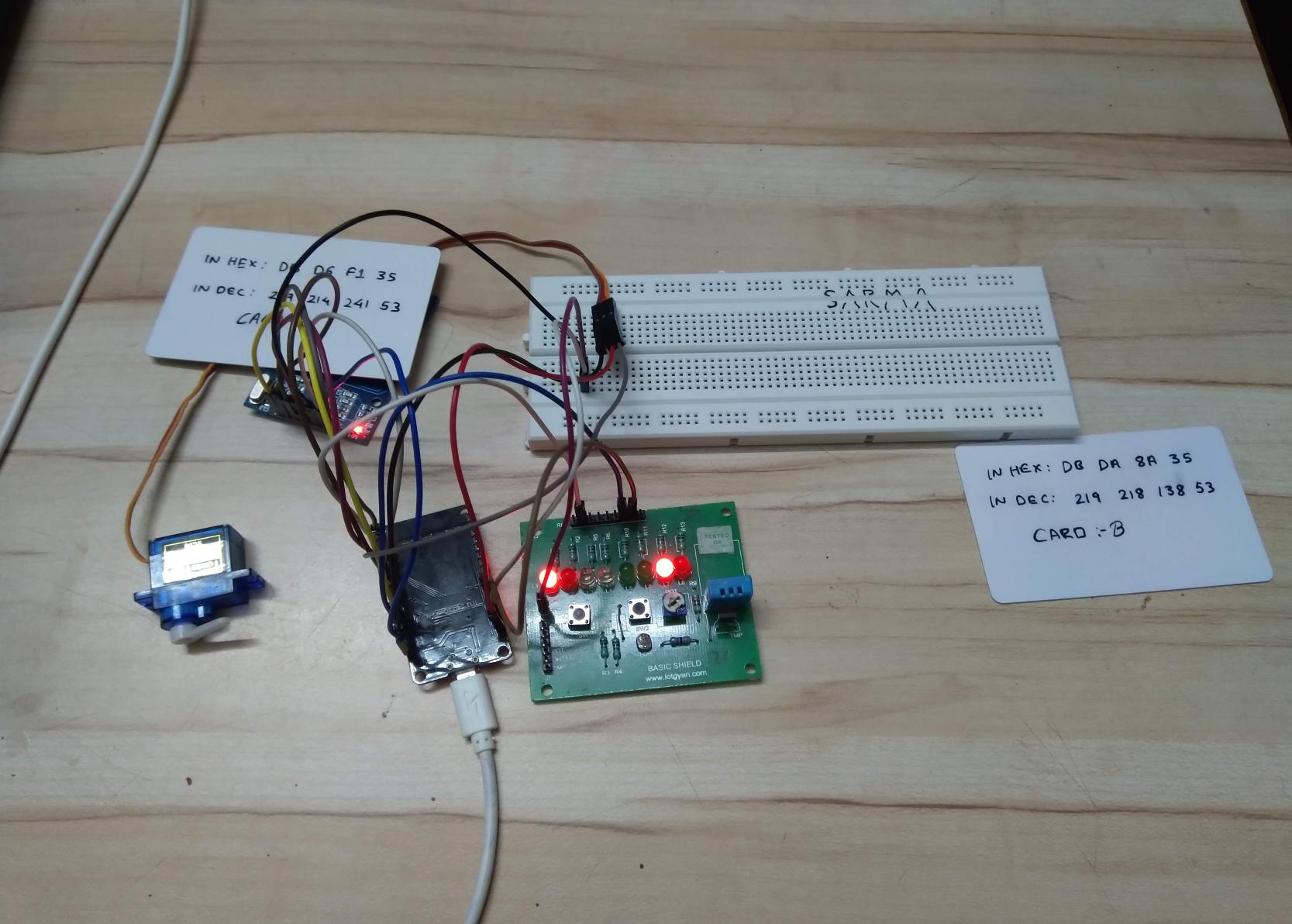
Serial.print(buffer[i], DEC);

}

}



ACCESS DENINED LAYOUT:



STAGE 2:PERSONAL SWITCH

Similar to access system this module scanns for the Authorized RFID

But when GEANTED the work done can only be controlled with the same RFID

Another Authorized RFID cannot control the work until the GRANTED one stops the work.

STAGE 2 CODE:

#include <SPI.h>

#include <MFRC522.h>

constexpr uint8\_t RST\_PIN = 5; // Configurable, see typical pin layout above

constexpr uint8\_t SS\_PIN = 4; // Configurable, see typical pin layout above

int a=0; //counter for a

int b=0; //counter for c

MFRC522 rfid(SS\_PIN, RST\_PIN); // Instance of the class

MFRC522::MIFARE\_Key key;

// Init array that will store new NUID

byte nuidPICC[4];

int arr\_c[] = {203,185,242,54}; //data of the pre saved card

int arr\_a[] = {203,57,99,54};

void setup() {

Serial.begin(9600);

pinMode(12,OUTPUT); //leb bulb

SPI.begin(); // Init SPI bus

rfid.PCD\_Init(); // Init MFRC522

for (byte i = 0; i < 6; i++) {

key.keyByte[i] = 0xFF;

}

Serial.println(F("This code scan the MIFARE Classsic NUID."));

Serial.print(F("Using the following key:"));

printHex(key.keyByte, MFRC522::MF\_KEY\_SIZE);

}

void loop() {

// Look for new cards

if ( ! rfid.PICC\_IsNewCardPresent())

return;

// Verify if the NUID has been readed

if ( ! rfid.PICC\_ReadCardSerial())

return;

Serial.print(F("PICC type: "));

MFRC522::PICC\_Type piccType = rfid.PICC\_GetType(rfid.uid.sak);

Serial.println(rfid.PICC\_GetTypeName(piccType));

// Check is the PICC of Classic MIFARE type

if (piccType != MFRC522::PICC\_TYPE\_MIFARE\_MINI &&

piccType != MFRC522::PICC\_TYPE\_MIFARE\_1K &&

piccType != MFRC522::PICC\_TYPE\_MIFARE\_4K) {

Serial.println(F("Your tag is not of type MIFARE Classic."));

return;

}

if (rfid.uid.uidByte[0] == arr\_a[0] &&

rfid.uid.uidByte[1] == arr\_a[1] &&

rfid.uid.uidByte[2] ==arr\_a[2] &&

rfid.uid.uidByte[3] == arr\_a[3] )

{

Serial.println(F("GRANTED."));

if(a%2==0)

{

digitalWrite(12,HIGH);

}

else{

digitalWrite(12,LOW);}

a++;

}

else if (rfid.uid.uidByte[0] == arr\_c[0] &&

rfid.uid.uidByte[1] == arr\_c[1] &&

rfid.uid.uidByte[2] ==arr\_c[2] &&

rfid.uid.uidByte[3] == arr\_c[3] )

{

Serial.println(F("GRANTED."));

if(b%2==0)

{

digitalWrite(12,HIGH);

}

else{

digitalWrite(12,LOW);}

b++;

}

else Serial.println(F("DENINED"));

// Halt PICC

rfid.PICC\_HaltA();

// Stop encryption on PCD

rfid.PCD\_StopCrypto1();

}

/\*\*

\* Helper routine to dump a byte array as hex values to Serial.

\*/

void printHex(byte \*buffer, byte bufferSize) {

for (byte i = 0; i < bufferSize; i++) {

Serial.print(buffer[i] < 0x10 ? " 0" : " ");

Serial.print(buffer[i], HEX);

}

}

/\*\*

\* Helper routine to dump a byte array as dec values to Serial.

\*/

void printDec(byte \*buffer, byte bufferSize) {

for (byte i = 0; i < bufferSize; i++) {

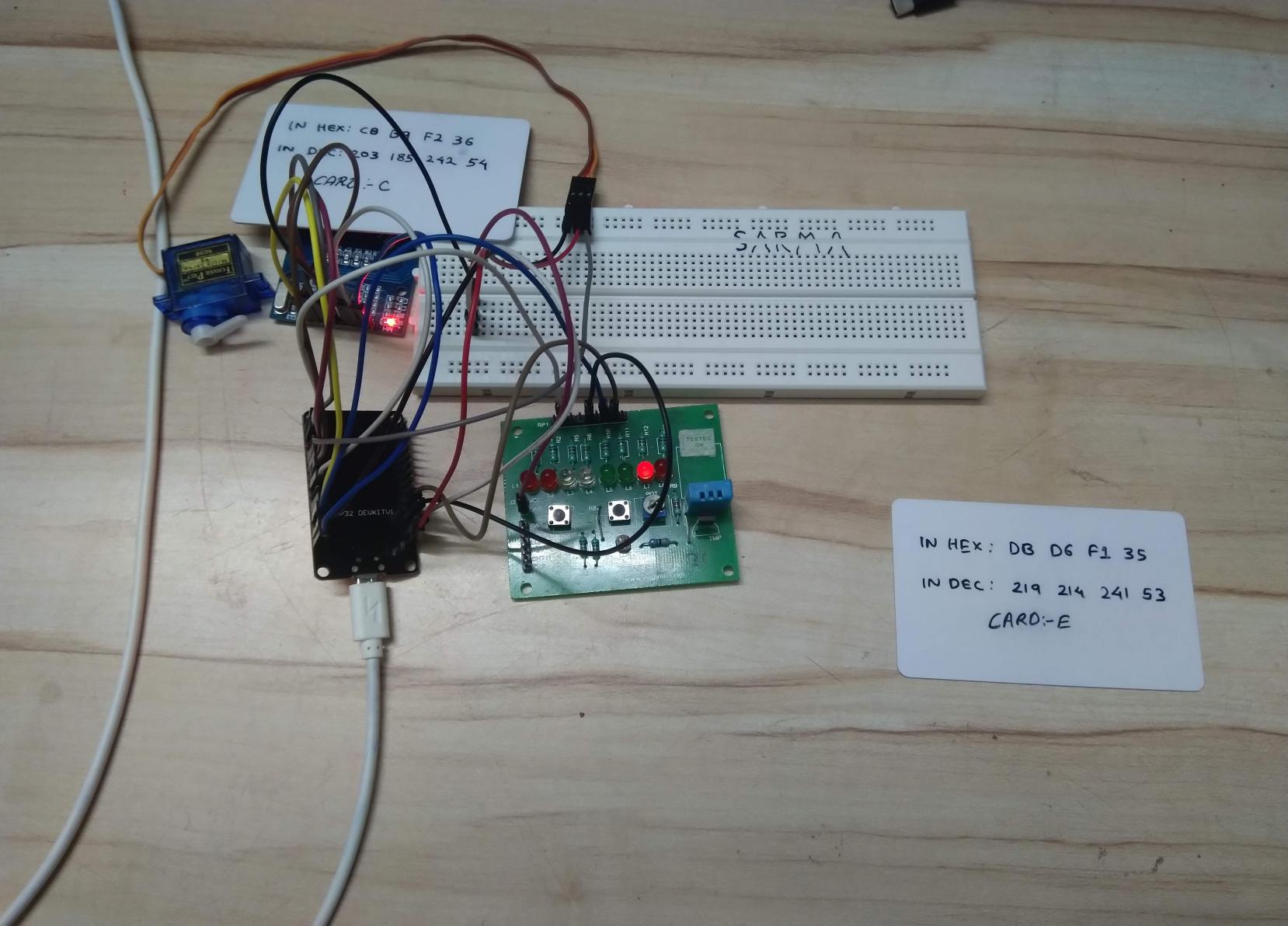
Serial.print(buffer[i] < 0x10 ? " 0" : " ");

Serial.print(buffer[i], DEC);

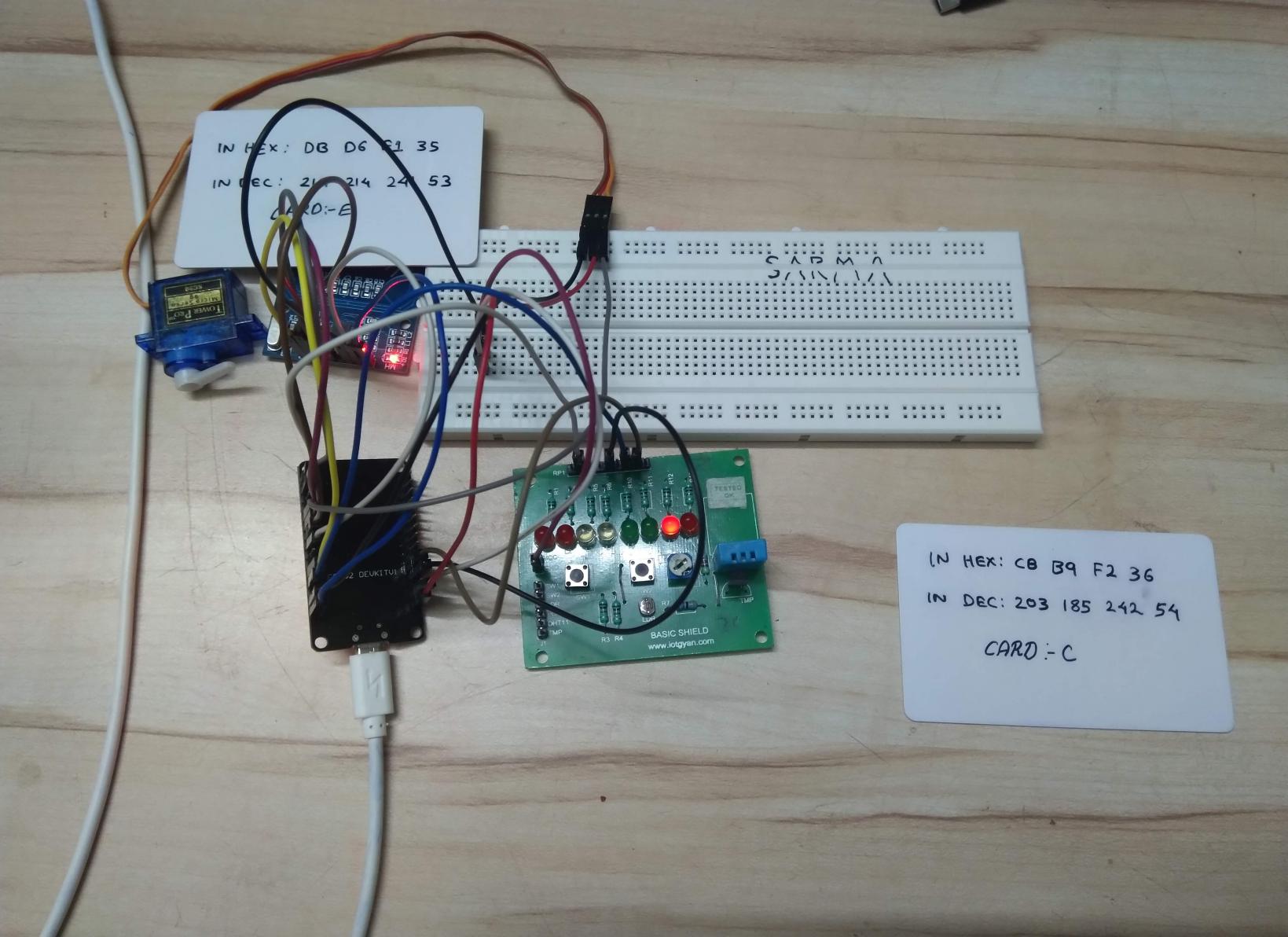
}

}

RFID GTANTED card starts LED:



RFID 2nd CARD cannot hinder the process:



# STAGE 3: UNIVERSAL IDENTIFICATION

This module scans the RFID and shows the data synced to it.

The RFID scans the card and via ESP32 we can access the details online in a webpage

The details can be as College ID, Aadhar card, Driving license etc.

STAGE 3 CODE:

#include <deprecated.h>

#include <MFRC522.h>

#include <MFRC522Extended.h>

#include <require\_cpp11.h>

#include <ETH.h>

#include <WiFi.h>

#include <WiFiAP.h>

#include <WiFiClient.h>

#include <WiFiGeneric.h>

#include <WiFiMulti.h>

#include <WiFiScan.h>

#include <WiFiServer.h>

#include <WiFiSTA.h>

#include <WiFiType.h>

#include <WiFiUdp.h>

#include <SPI.h>

const char\* ssid = "Sarma";

const char\* password = "vector103";

WiFiClient client;

WiFiServer server(80);

constexpr uint8\_t RST\_PIN = 5; // Configurable, see typical pin layout above

constexpr uint8\_t SS\_PIN = 4; // Configurable, see typical pin layout above

MFRC522 rfid(SS\_PIN, RST\_PIN); // Instance of the class

MFRC522::MIFARE\_Key key;

// Init array that will store new NUID

byte nuidPICC[4];

int arr\_c[] = {203,185,242,54}; //data of the pre saved card

int arr\_b[] = {219,218,138,53};

int arr\_a[] = {203,57,99,54};

int arr\_d[] = {58,252,13,63};

void setup() {

Serial.begin(115200);

Serial.println();

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi is connected");

server.begin();

Serial.println("Server started");

Serial.println(WiFi.localIP());

SPI.begin(); // Init SPI bus

rfid.PCD\_Init(); // Init MFRC522

for (byte i = 0; i < 6; i++) {

key.keyByte[i] = 0xFF;

}

Serial.println(F("This code scan the MIFARE Classsic NUID."));

// Serial.print(F("Using the following key:"));

// printHex(key.keyByte, MFRC522::MF\_KEY\_SIZE);

}

void loop() {

// Look for new cards

if ( ! rfid.PICC\_IsNewCardPresent())

return;

// Verify if the NUID has been readed

if ( ! rfid.PICC\_ReadCardSerial())

return;

Serial.print("PICC type: ");

MFRC522::PICC\_Type piccType = rfid.PICC\_GetType(rfid.uid.sak);

Serial.println(rfid.PICC\_GetTypeName(piccType));

// Check is the PICC of Classic MIFARE type

if (piccType != MFRC522::PICC\_TYPE\_MIFARE\_MINI &&

piccType != MFRC522::PICC\_TYPE\_MIFARE\_1K &&

piccType != MFRC522::PICC\_TYPE\_MIFARE\_4K) {

Serial.println(F("Your tag is not of type MIFARE Classic."));

return;

}

if (rfid.uid.uidByte[0] == arr\_a[0] &&

rfid.uid.uidByte[1] == arr\_a[1] &&

rfid.uid.uidByte[2] ==arr\_a[2] &&

rfid.uid.uidByte[3] == arr\_a[3] )

{

Serial.println(F("GRANTED."));

WiFiClient client = server.available();

client.println("HTTP/1.1 200 OK");

client.println("Content-Type: text/html");

client.println("Connection: close");

client.println("Refresh: 10");

client.println();

client.println("<!DOCTYPE HTML>");

client.println("<html>");

client.print("<p style='text-align: center;'><span style='font-size: x-large;'><strong>UNIVERSAL RFID</strong></span></p>");

client.print("<p style='text-align: side;'><span style='color: #0000ff;'><strong style='font-size: large;'>Name:<pre>P NAGA LAKSHMI NARAYANA SARMA ");

client.print("<p style='text-align: side;'><span style='color: #0000ff;'><strong style='font-size: large;'>PH.no:<pre>7530027881 ");

client.print("<p style='text-align: side;'><span style='color: #0000ff;'><strong style='font-size: large;'>D.O.B:<pre>18/12/1997 ");

client.print("<p style='text-align: side;'><span style='color: #0000ff;'><strong style='font-size: large;'>SEX:<pre>MALE ");

client.print("<p style='text-align: side;'><span style='color: #0000ff;'><strong style='font-size: large;'>FATHER NAME:<pre> S JAGANNADHA RAO ");

client.print("<p style='text-align: side;'><span style='color: #0000ff;'><strong style='font-size: large;'>ADDRESS:FLAT NO:<pre>435 KALPKRITI PARISAR,RISALI, BHILAI,CHHATTISGARH. ");

client.print("<p style='text-align: side;'><span style='color: #0000ff;'><strong style='font-size: large;'>OCCUPATION:<pre>STUDENT ");

client.print("<p style='text-align: side;'><span style='color: #0000ff;'><strong style='font-size: large;'>INSTTUTION:<pre> KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES.");

client.print("<p style='text-align: side;'><span style='color: #0000ff;'><strong style='font-size: large;'>INSTITUTION ID:<pre> UR16EC077");

client.print("<p style='text-align: side;'><span style='color: #0000ff;'><strong style='font-size: large;'>AADHAR CARD NO:<pre>7944 0195 5822 ");

client.print("<p style='text-align: side;'><span style='color: #0000ff;'><strong style='font-size: large;'>DRIVING LICENCE NUMBER:<pre>CG07 20160010527 ");

client.print("<p style='text-align: side;'><span style='color: #0000ff;'><strong style='font-size: large;'>VEHICLE CLASS:<pre>LMV , MCWG ");

client.print("<p style='text-align: side;'><span style='color: #0000ff;'><strong style='font-size: large;'>DATE OF ISSUE:<pre> 21/7/2016");

client.print("<p style='text-align: side;'><span style='color: #0000ff;'><strong style='font-size: large;'>DRIVING LICENCE VALIDITY:<pre>20/7/2036");

client.print("</p>");

client.println("</html>");

delay(50);

}

else if (rfid.uid.uidByte[0] == arr\_b[0] &&

rfid.uid.uidByte[1] == arr\_b[1] &&

rfid.uid.uidByte[2] ==arr\_b[2] &&

rfid.uid.uidByte[3] == arr\_b[3] )

{

// Serial.println(F("GRANTED."));

WiFiClient client = server.available();

client.println("HTTP/1.1 200 OK");

client.println("Content-Type: text/html");

client.println("Connection: close");

client.println("Refresh: 10");

client.println();

client.println("<!DOCTYPE HTML>");

client.println("<html>");

client.print("<p style='text-align: center;'><span style='font-size: x-large;'><strong>MOBILE INFORMATION LOG</strong></span></p>");

client.print("<p style='text-align: center;'><span style='color: #0000ff;'><strong style='font-size: large;'>User name: Gausic ");

client.print("<p style='text-align: center;'><span style='color: #0000ff;'><strong style='font-size: large;'>Company : LENOVO");

client.print("<p style='text-align: center;'><span style='color: #0000ff;'><strong style='font-size: large;'>Mobile Number: 8220757658 ");

client.print("<p style='text-align: center;'><span style='color: #0000ff;'><strong style='font-size: large;'>Model name: VIBE K5 note ");

client.print("<p style='text-align: center;'><span style='color: #0000ff;'><strong style='font-size: large;'>Price: 12000 Rs ");

client.print("</p>");

client.println("</html>");

delay(50);

}

else if (rfid.uid.uidByte[0] == arr\_c[0] &&

rfid.uid.uidByte[1] == arr\_c[1] &&

rfid.uid.uidByte[2] ==arr\_c[2] &&

rfid.uid.uidByte[3] == arr\_c[3] )

{

// Serial.println(F("GRANTED."));

WiFiClient client = server.available();

client.println("HTTP/1.1 200 OK");

client.println("Content-Type: text/html");

client.println("Connection: close");

client.println("Refresh: 10");

client.println();

client.println("<!DOCTYPE HTML>");

client.println("<html>");

client.print("<p style='text-align: center;'><span style='font-size: x-large;'><strong>MOBILE INFORMATION LOG</strong></span></p>");

client.print("<p style='text-align: center;'><span style='color: #0000ff;'><strong style='font-size: large;'>User name: Esther ");

client.print("<p style='text-align: center;'><span style='color: #0000ff;'><strong style='font-size: large;'>Company : NOKIA ");

client.print("<p style='text-align: center;'><span style='color: #0000ff;'><strong style='font-size: large;'>Mobile Number: 9488601946 ");

client.print("<p style='text-align: center;'><span style='color: #0000ff;'><strong style='font-size: large;'>Model name: 3 ");

client.print("<p style='text-align: center;'><span style='color: #0000ff;'><strong style='font-size: large;'>Price: 9500 Rs ");

client.print("</p>");

client.println("</html>");

delay(50);

}

else if (rfid.uid.uidByte[0] == arr\_d[0] &&

rfid.uid.uidByte[1] == arr\_d[1] &&

rfid.uid.uidByte[2] == arr\_d[2] &&

rfid.uid.uidByte[3] == arr\_d[3] )

{

Serial.println(F("GRANTED."));

WiFiClient client = server.available();

client.println("HTTP/1.1 200 OK");

client.println("Content-Type: text/html");

client.println("Connection: close");

client.println("Refresh: 10");

client.println();

client.println("<!DOCTYPE HTML>");

client.println("<html>");

client.print("<p style='text-align: center;'><span style='font-size: x-large;'><strong>MOBILE INFORMATION LOG</strong></span></p>");

client.print("<p style='text-align: center;'><span style='color: #0000ff;'><strong style='font-size: large;'>User name: Swathi ");

client.print("<p style='text-align: center;'><span style='color: #0000ff;'><strong style='font-size: large;'>Company : HONOR");

client.print("<p style='text-align: center;'><span style='color: #0000ff;'><strong style='font-size: large;'>Mobile Number: 8085746851 ");

client.print("<p style='text-align: center;'><span style='color: #0000ff;'><strong style='font-size: large;'>Model name: 9lite ");

client.print("<p style='text-align: center;'><span style='color: #0000ff;'><strong style='font-size: large;'>Price: 11500 Rs ");

client.print("</p>");

client.println("</html>");

delay(50);

}

else Serial.println(F("DENINED"));

// Halt PICC

rfid.PICC\_HaltA();

// Stop encryption on PCD

rfid.PCD\_StopCrypto1();

}

