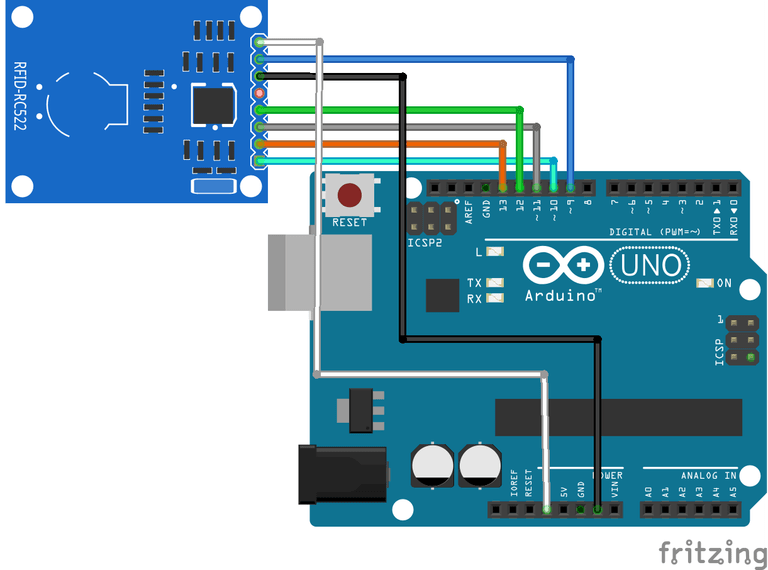
***UFID finder :-***



#include <SPI.h>

#include <MFRC522.h>

#define RST\_PIN 9 // Configurable, see typical pin layout above

#define SS\_PIN 10 // Configurable, see typical pin layout above

MFRC522 mfrc522(SS\_PIN, RST\_PIN); // Create MFRC522 instance

void setup() {

Serial.begin(9600); // Initialize serial communications with the PC

while (!Serial); // Do nothing if no serial port is opened (added for Arduinos based on ATMEGA32U4)

SPI.begin(); // Init SPI bus

mfrc522.PCD\_Init(); // Init MFRC522

delay(4); // Optional delay. Some board do need more time after init to be ready, see Readme

mfrc522.PCD\_DumpVersionToSerial(); // Show details of PCD - MFRC522 Card Reader details

Serial.println(F("Scan PICC to see UID, SAK, type, and data blocks..."));

}

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;

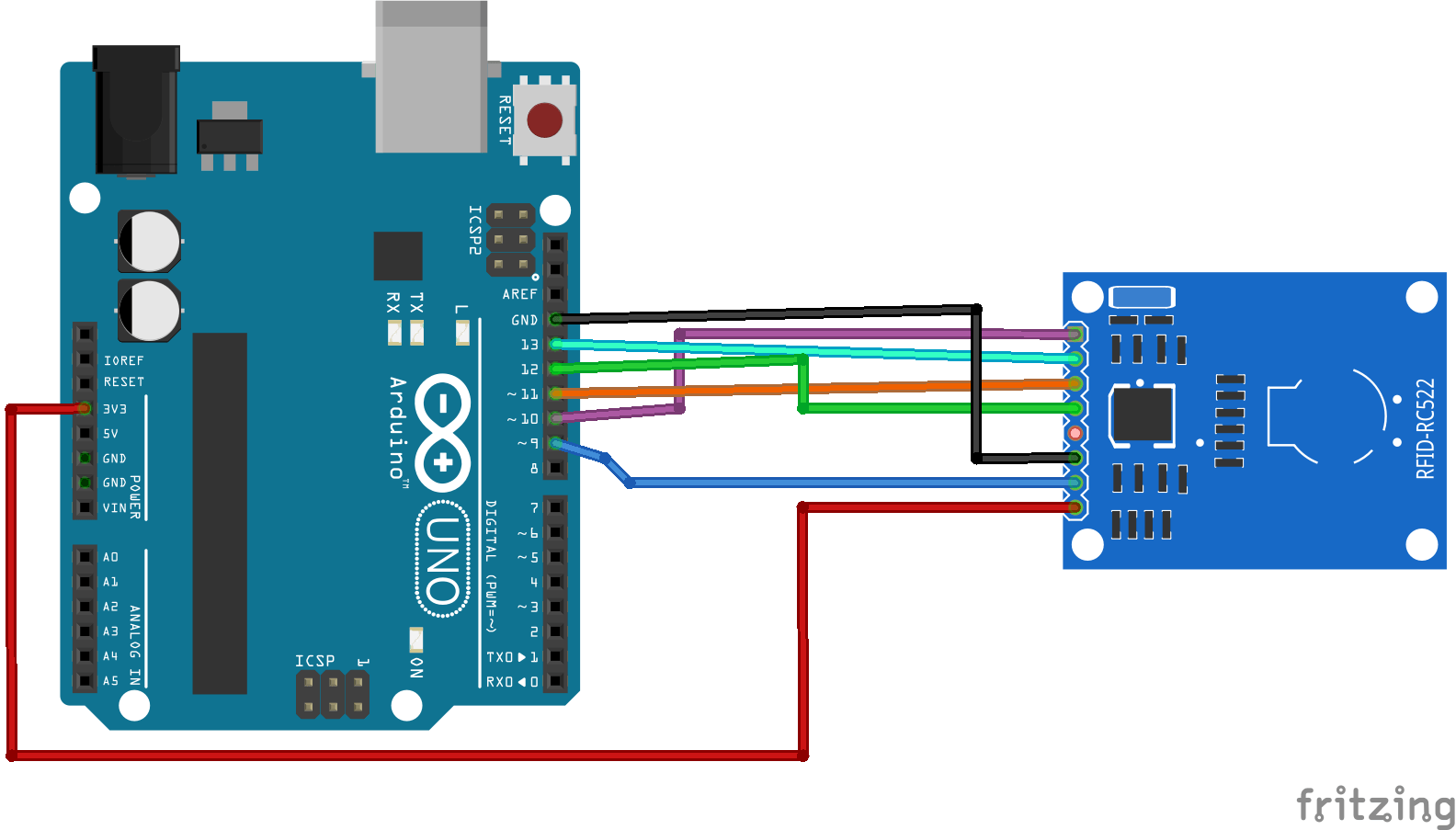
}

// Dump debug info about the card; PICC\_HaltA() is automatically called

mfrc522.PICC\_DumpToSerial(&(mfrc522.uid));

}

***Checking UFID Access :-***



/\*

\*

\* All the resources for this project: https://www.hackster.io/Aritro

\* Modified by Aritro Mukherjee

\*

\*

\*/

#include <SPI.h>

#include <MFRC522.h>

#define SS\_PIN 10

#define RST\_PIN 9

MFRC522 mfrc522(SS\_PIN, RST\_PIN); // Create MFRC522 instance.

void setup()

{

Serial.begin(9600); // Initiate a serial communication

SPI.begin(); // Initiate SPI bus

mfrc522.PCD\_Init(); // Initiate MFRC522

Serial.println("Approximate your card to the reader...");

Serial.println();

}

void loop()

{

// Look for new cards

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) == "BD 31 15 2B") //change here the UID of the card/cards that you want to give access

{

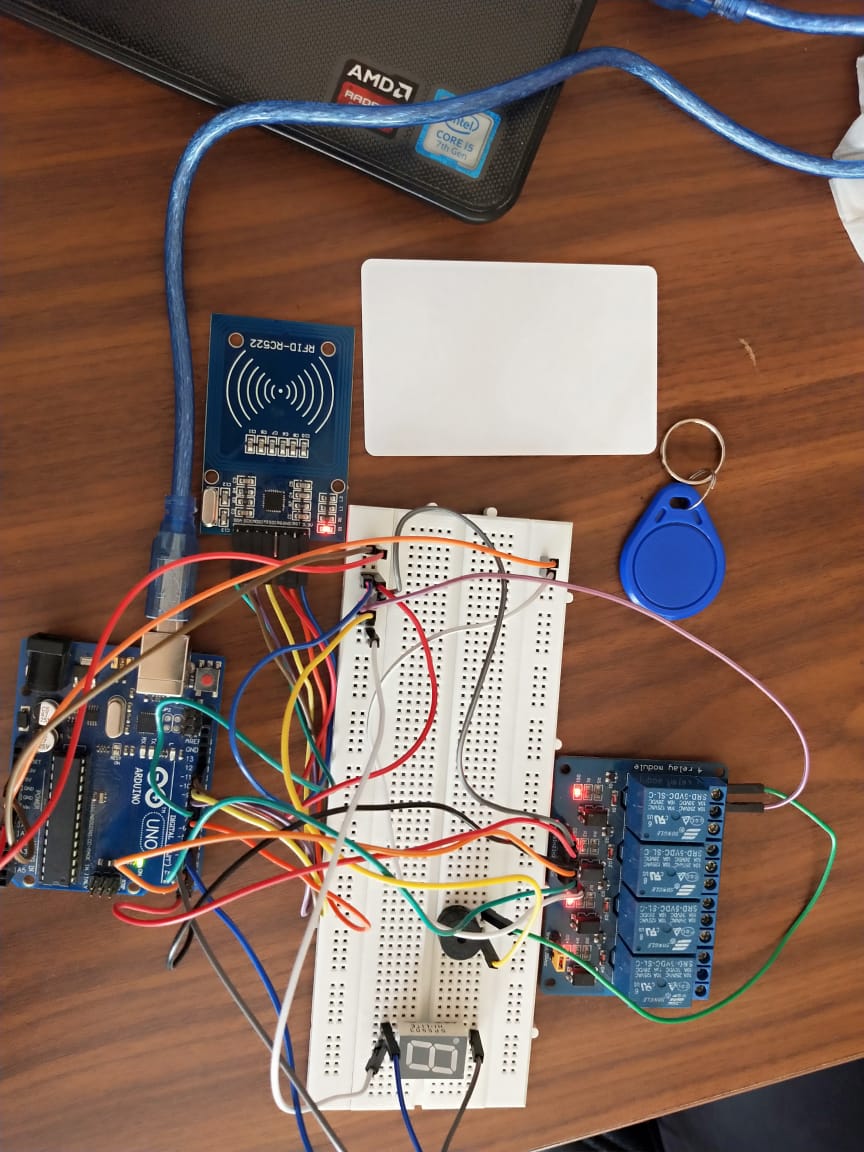
Serial.println("Authorized access");

Serial.println();

delay(3000);

}

Final Programme :-

#include <SPI.h>

#include <MFRC522.h>

#define SS\_PIN 10

#define RST\_PIN 9

MFRC522 mfrc522(SS\_PIN, RST\_PIN); // Create MFRC522 instance.

void setup()

{

Serial.begin(9600); // Initiate a serial communication

SPI.begin(); // Initiate SPI bus

mfrc522.PCD\_Init(); // Initiate MFRC522

Serial.println("Approximate your card to the reader...");

Serial.println();

pinMode(2, OUTPUT);//Light2 pin

pinMode(3, OUTPUT);//Light3 pin

pinMode(4, OUTPUT);//AC pin

pinMode(5, OUTPUT);//

}

void loop()

{

// Look for new cards

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) == "C0 2A 5A 56") //change here the UID of the card/cards that you want to give access

{

Serial.println("Authorized access");

digitalWrite(2,HIGH);

digitalWrite(3,HIGH);

digitalWrite(4,HIGH);

digitalWrite(5,HIGH);

digitalWrite(6,HIGH);

digitalWrite(7,HIGH);

Serial.println();

delay(1000);

if ( !mfrc522.PICC\_ReadCardSerial())

{

digitalWrite(2,LOW);

digitalWrite(3,LOW);

digitalWrite(4,LOW);

digitalWrite(5,LOW);

digitalWrite(6,LOW);

digitalWrite(7,LOW);

}

}

else

{

digitalWrite(2,LOW);

digitalWrite(3,LOW);

digitalWrite(4,LOW);

digitalWrite(5,LOW);

digitalWrite(6,LOW);

digitalWrite(7,LOW);

}

}