

INFORME PRÀCTICA 6_2

CODI

```
#include <Arduino.h>
#include <MFRC522.h> //library responsible for communicating with the
module RFID-RC522
#include <SPI.h> //library responsible for communicating of SPI bus
#define SS_PIN    21
#define RST_PIN    22
#define SIZE_BUFFER    18
#define MAX_SIZE_BLOCK 16
#define greenPin    12
#define redPin      32
//used in authentication
MFRC522::MIFARE_Key key;
//authentication return status code
MFRC522::StatusCode status;
// Defined pins to module RC522
MFRC522 mfrc522(SS_PIN, RST_PIN);
int menu();
void readingData();
void writingData();
void setup()
{
  Serial.begin(115200);
  SPI.begin(); // Init SPI bus
  pinMode(greenPin, OUTPUT);
  pinMode(redPin, OUTPUT);

  // Init MFRC522
  mfrc522.PCD_Init();
  Serial.println("Approach your reader card...");
  Serial.println();
}

void loop()
{
  // Aguarda a aproximacao do cartao
  //waiting the card approach
  if ( ! mfrc522.PICC_IsNewCardPresent())
  {
    return;
  }
  // Select a card
  if ( ! mfrc522.PICC_ReadCardSerial())
  {
    return;
  }
}
```

```

        // Dump debug info about the card; PICC_HaltA() is automatically
called
        // mfrc522.PICC_DumpToSerial(&(mfrc522.uid));</p><p> //call menu
function and retrieve the desired option
        int op = menu();

        if(op == 0)
            readingData();
        else if(op == 1)
            writingData();
        else {
            Serial.println(F("Incorrect Option!"));
            return;
        }

        //instructs the PICC when in the ACTIVE state to go to a "STOP" state
mfrc522.PICC_HaltA();
        // "stop" the encryption of the PCD, it must be called after communication
with authentication, otherwise new communications can not be initiated
mfrc522.PCD_StopCrypto1();
    }

    //reads data from card/tag
    void readingData()
    {
        //prints the technical details of the card/tag
mfrc522.PICC_DumpDetailsToSerial(&(mfrc522.uid));

        //prepare the key - all keys are set to FFFFFFFFFFFFFh
        for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;

        //buffer for read data
        byte buffer[SIZE_BUFFER] = {0};

        //the block to operate
        byte block = 1;
        byte size = SIZE_BUFFER;</p><p> //authenticates the block to operate
        status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, block,
&key, &(mfrc522.uid)); //line 834 of MFRC522.cpp file
        if (status != MFRC522::STATUS_OK) {
            Serial.print(F("Authentication failed: "));
            Serial.println(mfrc522.GetStatusCodeName(status));
            digitalWrite(redPin, HIGH);
            delay(1000);
            digitalWrite(redPin, LOW);
            return;
        }

        //read data from block
        status = mfrc522.MIFARE_Read(block, buffer, &size);
        if (status != MFRC522::STATUS_OK) {
            Serial.print(F("Reading failed: "));
            Serial.println(mfrc522.GetStatusCodeName(status));
            digitalWrite(redPin, HIGH);

```

```

        delay(1000);
        digitalWrite(redPin, LOW);
        return;
    }
    else{
        digitalWrite(greenPin, HIGH);
        delay(1000);
        digitalWrite(greenPin, LOW);
    }

    Serial.print(F("\nData from block ["));
    Serial.print(block);Serial.print(F("]: "));

    //prints read data
    for (uint8_t i = 0; i < MAX_SIZE_BLOCK; i++)
    {
        Serial.write(buffer[i]);
    }
    Serial.println(" ");
}

void writingData(){
    //prints thecnical details from of the card/tag
    mfrc522.PICC_DumpDetailsToSerial(&(mfrc522.uid));

    // waits 30 seconds dor data entry via Serial
    Serial.setTimeout(30000L) ;
    Serial.println(F("Enter the data to be written with the '#' character at
the end \n[maximum of 16 characters]:"));

    //prepare the key - all keys are set to FFFFFFFFh
    for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;

    //buffer para armazenamento dos dados que iremos gravar
    //buffer for storing data to write
    byte buffer[MAX_SIZE_BLOCK] = "";
    byte block; //the block to operate
    byte dataSize; //size of data (bytes)

    //recover on buffer the data from Serial
    //all characters before chacactere '#'
    dataSize = Serial.readBytesUntil('#', (char*)buffer, MAX_SIZE_BLOCK);
    //void positions that are left in the buffer will be filled with
whitespace
    for(byte i=dataSize; i < MAX_SIZE_BLOCK; i++)
    {
        buffer[i] = ' ';
    }

    block = 1; //the block to operate
    String str = (char*)buffer; //transforms the buffer data in String
    Serial.println(str);

    //authenticates the block to operate

```

```

//Authenticate is a command to hability a secure communication
status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A,
                                   block, &key, &(mfrc522.uid));

if (status != MFRC522::STATUS_OK) {
    Serial.print(F("PCD_Authenticate() failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    digitalWrite(redPin, HIGH);
    delay(1000);
    digitalWrite(redPin, LOW);
    return;
}
//else Serial.println(F("PCD_Authenticate() success: "));

//Writes in the block
status = mfrc522.MIFARE_Write(block, buffer, MAX_SIZE_BLOCK);
if (status != MFRC522::STATUS_OK) {
    Serial.print(F("MIFARE_Write() failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    digitalWrite(redPin, HIGH);
    delay(1000);
    digitalWrite(redPin, LOW);
    return;
}
else{
    Serial.println(F("MIFARE_Write() success: "));
    digitalWrite(greenPin, HIGH);
    delay(1000);
    digitalWrite(greenPin, LOW);
}

}

int menu(){
    Serial.println(F("\nChoose an option:"));
    Serial.println(F("0 - Reading data"));
    Serial.println(F("1 - Writing data\n"));

    //waits while the user does not start data
    while(!Serial.available()){};

    //retrieves the chosen option
    int op = (int)Serial.read();

    //remove all characters after option (as \n per example)
    while(Serial.available()) {
        if(Serial.read() == '\n') break;
        Serial.read();
    }
    return (op-48);//subtract 48 from read value, 48 is the zero from ascii
}

```

table

FUNCIONAMENT

Quan executem el programa surt un comentari que demana que apropem la targeta al lector. Un cop l'apropem, s'ens obra un menú amb dos opcions: 0.Reading data 1.Writing data Si li donem a l'opció d'escriure, ens surt el CARD UID, que és el id de la targeta que hem aprofitat. Tot seguit, ens surt el card SAK i el PICC type

VIDEO DE L'EXECUCIÓ

https://drive.google.com/file/d/1X6pk_q64t4y2T8K3rqXwqe85t5aqKoYO/view?usp=sharing