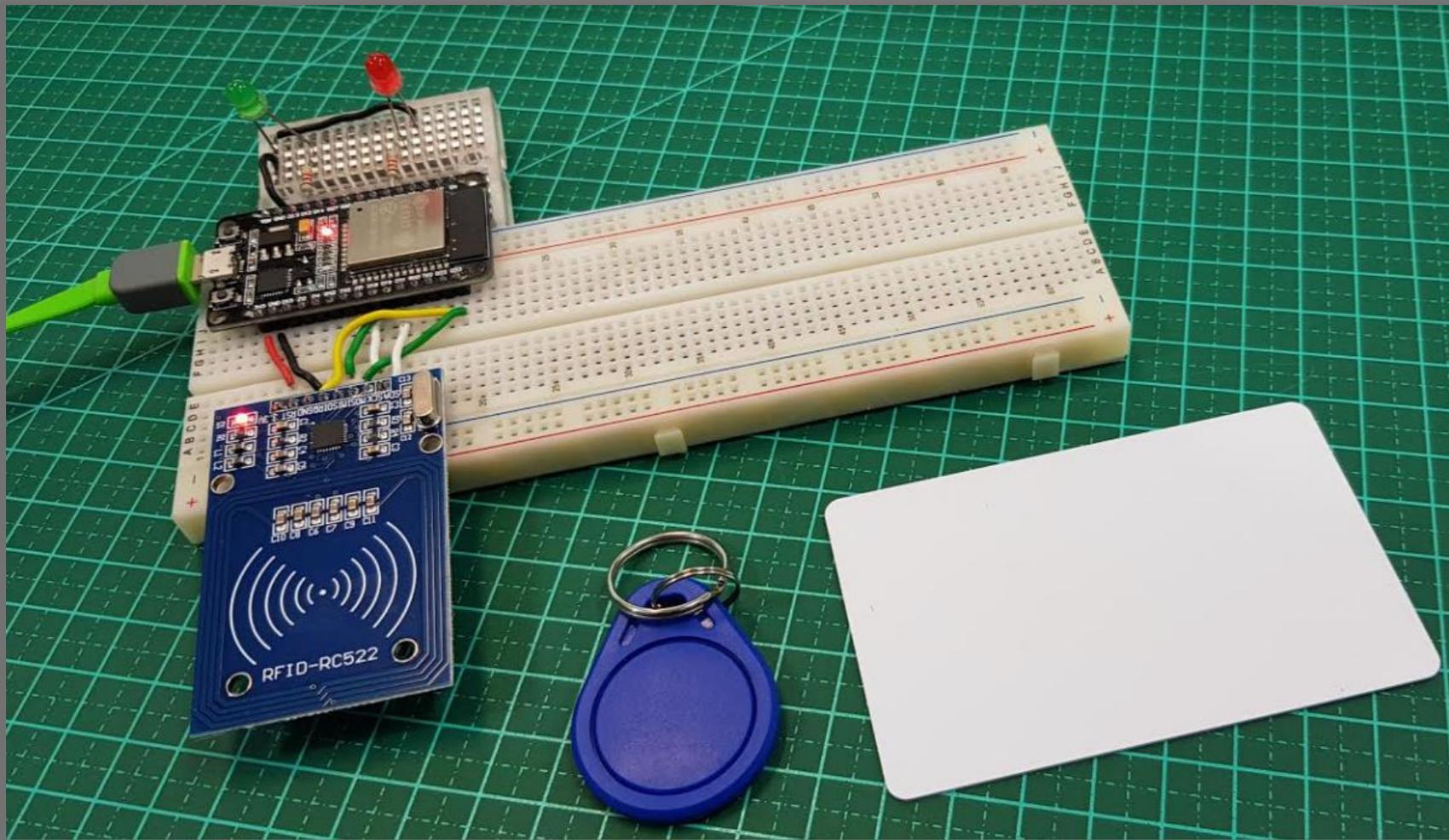


Esp32 com módulo RFID



Por Fernando Koyanagi



Objetivo

Nosso objetivo será criar um programa, no qual poderemos tanto fazer a leitura de um cartão (ou tag) RFID como gravar os dados nele.

Utilizaremos um WiFi NodeMCU-32S e um módulo RFID-RC522.



Introdução

Identificação por rádio frequência ou **RFID** é uma tecnologia com um método de identificação automática através de sinais de rádio. Podemos armazenar e recuperar dados remotamente através de dispositivos.

No nosso dia a dia encontramos essa tecnologia nos cartões de transporte público, por exemplo, bem como no setor de segurança, como controle de acesso, em bibliotecas para identificação de livros e outras operações, dentre outros.



Funcionamento

Um sistema RFID é composto basicamente de um transceptor com decodificador e uma antena e um transponder.



Memória

A memória EEPROM está organizada em 16 setores de 4 blocos. Cada bloco contém 16 bytes.

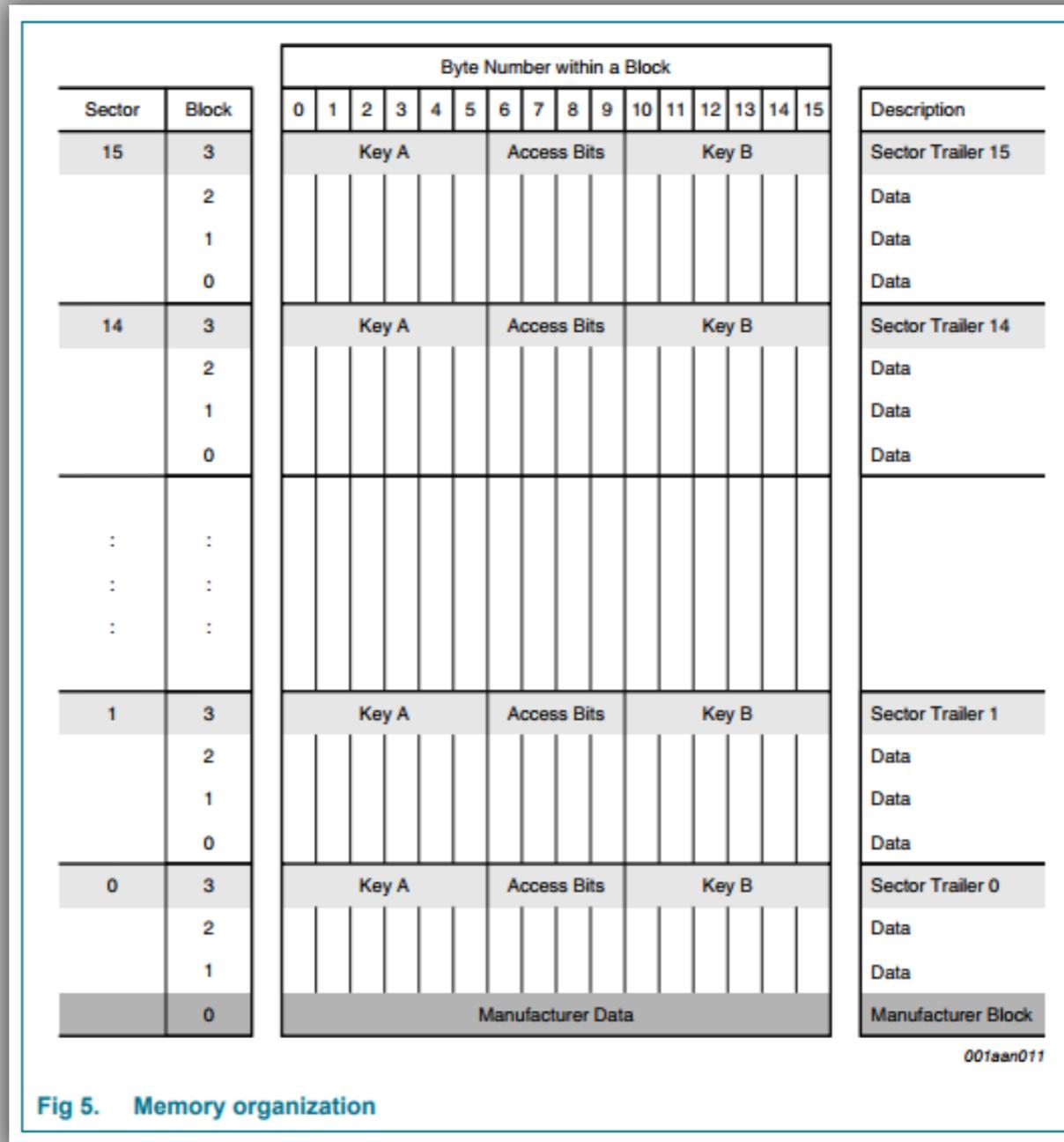


Fig 5. Memory organization





Em www.fernandok.com

- PRINCIPAL
- SOBRE FERNANDO K
- ARDUINO
- ESP8266
- ESP32
- LORAWAN
- MOTOR
- DISPLAY
- MATERIAIS
- DOWNLOAD

Receba o meu conteúdo GRATUITAMENTE

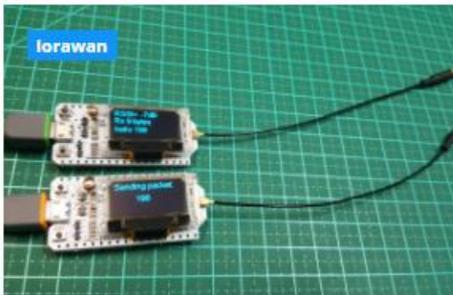
QUERO RECEBER GRÁTIS



Motor de Passo Nema 23 com Driver TB6600 e Arduino Due

by Fernando K Tecnologia - 2:44 PM
Hoje vamos voltar a falar de Motor de Passo. Vamos utilizar um Nema 23 que será controlado por um Driver TB6600 e um Arduino Due. É p...

Leia mais



ESP32 Longa Distância - LoRaWan

by Fernando K Tecnologia - 9:46 AM
Neste artigo vamos tratar da LoRaWAN, uma rede que vai longe gastando pouca energia. Mas, o quanto "longe"? Com o chip que uso no vídeo...

Leia mais



Motor de HD com Arduino

by Fernando K Tecnologia - 2:00 PM

QUAL ASSUNTO VOCÊ TEM

- Arduino
- ESP8266
- ESP32
- Motor
- Display
- Sensor

You may select multiple answers.
Votar Exibir resultados

Votos até o momento: 32
Dias restantes para votar: 49

FACEBOOK



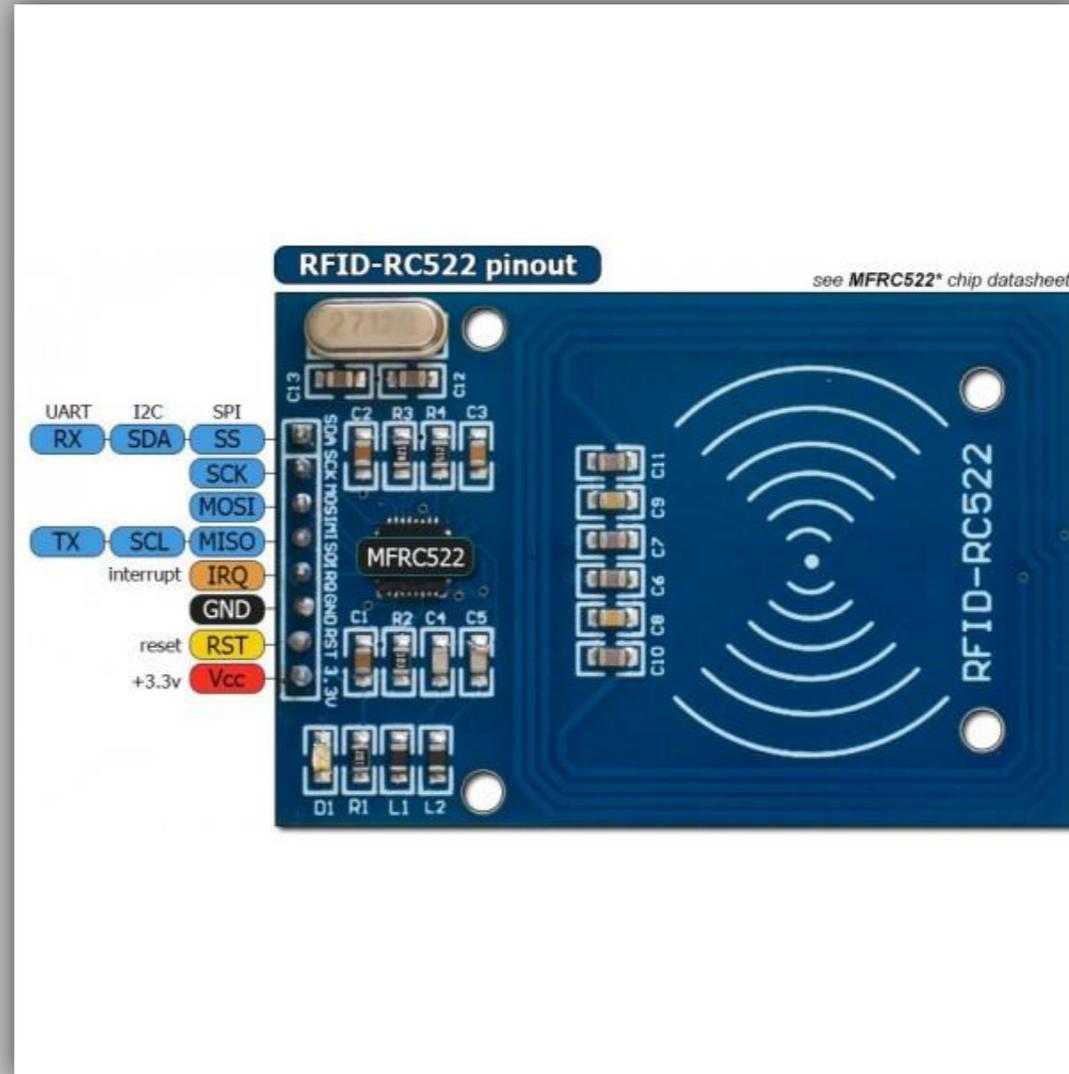
Seu e-mail



Inscreva-se



RFID-RC522



Montagem

ESP32 **RC522**

18 SCK

19 MISO

21 SDA

22 RST

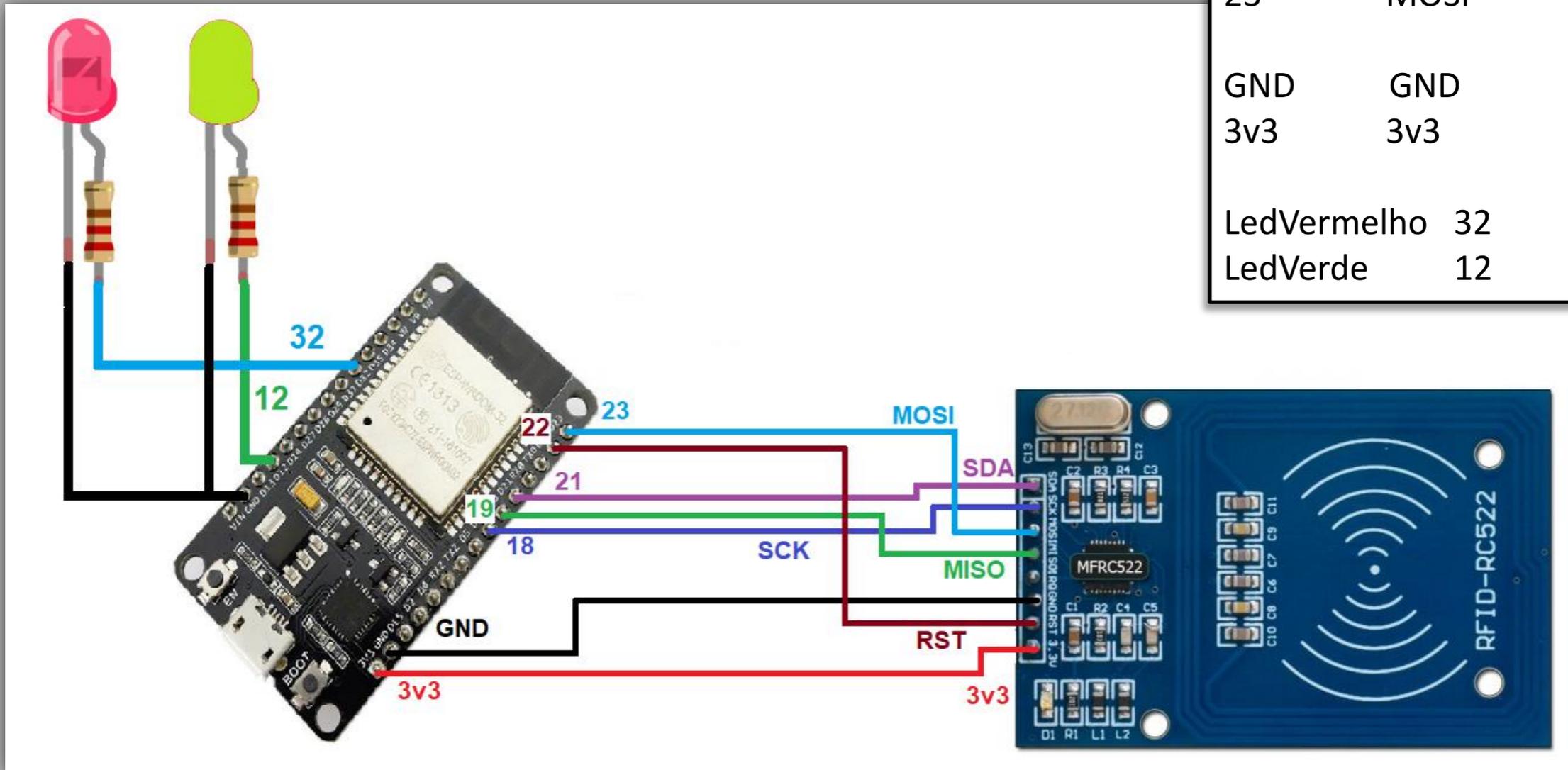
23 MOSI

GND GND

3v3 3v3

LedVermelho 32

LedVerde 12



Bibliotecas

Adicione a seguinte biblioteca “MFRC522”.

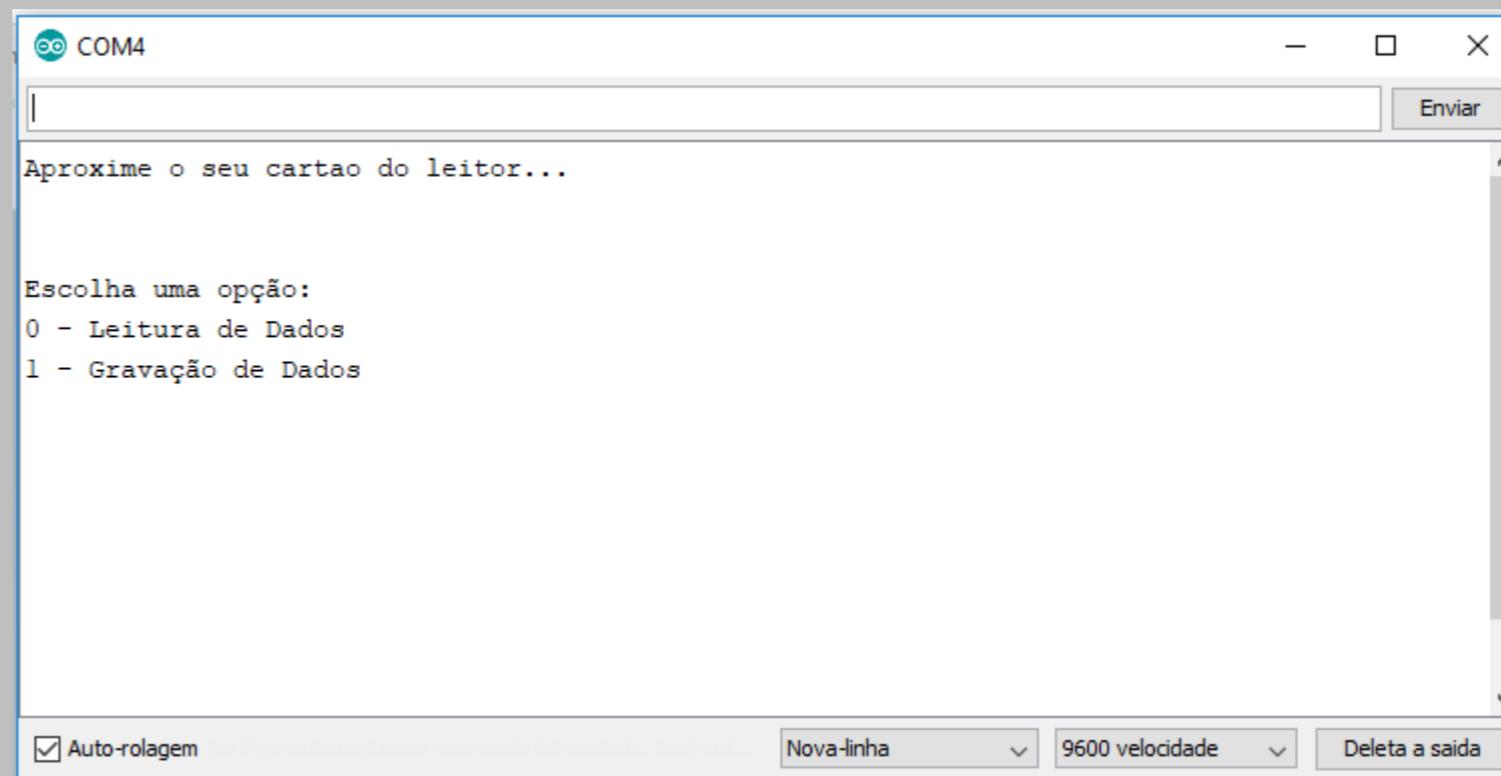
Basta acessar “Sketch >> Incluir Bibliotecas >> Gerenciar Bibliotecas...”



Vamos ao código

Nosso programa funcionará da seguinte maneira:

Após iniciar, o programa ficará esperando que um cartão ou tag seja identificado. Após isso, um menu surgirá para que o usuário escolha entre fazer uma leitura ou gravar algo. E em seguida a operação será realizada.



Setup

```
#include <MFRC522.h> //biblioteca responsável pela comunicação com o módulo RFID-RC522
#include <SPI.h> //biblioteca para comunicação do barramento SPI

#define SS_PIN          21
#define RST_PIN         22
#define SIZE_BUFFER     18 //tamanho do buffer (16 bits do bloco + 2 bits CRC)
#define MAX_SIZE_BLOCK  16 //tamanho dos dados do bloco
#define pinVerde        12
#define pinVermelho     32

//esse objeto 'chave' é utilizado para autenticação
MFRC522::MIFARE_Key key;
//código de status de retorno da autenticação
MFRC522::StatusCode status;
// Definições pino modulo RC522
MFRC522 mfrc522(SS_PIN, RST_PIN);

void setup() {
    // Inicia a serial
    Serial.begin(9600);
    SPI.begin(); // Init SPI bus
    pinMode(pinVerde, OUTPUT);
    pinMode(pinVermelho, OUTPUT);
    // Inicia MFRC522
    mfrc522.PCD_Init();
    // Mensagens iniciais no serial monitor
    Serial.println("Aproxime o seu cartao do leitor...");
    Serial.println();
}
```



Loop

```
void loop()
{
    // Aguarda a aproximacao do cartao
    if ( ! mfrc522.PICC_IsNewCardPresent() )
    {
        return;
    }
    // Seleciona um dos cartoes
    if ( ! mfrc522.PICC_ReadCardSerial() )
    {
        return;
    }
    //chama o menu e recupera a opção desejada
    int opcao = menu();
    if(opcao == 0)
        leituraDados();
    else if(opcao == 1)
        gravarDados();
    else {
        Serial.println(F("Opção Incorreta!"));
        return;
    }
    // instrui o PICC quando no estado ACTIVE a ir para um estado de "parada"
    mfrc522.PICC_HaltA();
    // "stop" a encriptação do PCD, deve ser chamado após a comunicação com
    //autenticação, caso contrário novas comunicações não poderão ser iniciadas
    mfrc522.PCD_StopCrypto1();
}
```



Leitura

```
//faz a leitura dos dados do cartão/tag
void leituraDados() {
    //Prepara a chave - todas as chaves estão configuradas para FFFFFFFFh (Padrão de fábrica).
    for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;
    byte buffer[SIZE_BUFFER] = {0}; //buffer para colocar os dados lidos
    byte bloco = 1; //bloco que faremos a operação
    byte tamanho = SIZE_BUFFER;
    //faz a autenticação do bloco que vamos operar
    status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, bloco, &key, &(mfrc522.uid));
    if (status != MFRC522::STATUS_OK) {
        Serial.print(F("Authentication failed: ")); //erro de autenticação
        Serial.println(mfrc522.GetStatusCodeName(status));
        digitalWrite(pinVermelho, HIGH); delay(1000); digitalWrite(pinVermelho, LOW);
        return;
    }
    status = mfrc522.MIFARE_Read(bloco, buffer, &tamanho); //faz a leitura dos dados do bloco
    if (status != MFRC522::STATUS_OK) {
        Serial.print(F("Reading failed: ")); //erro de leitura
        Serial.println(mfrc522.GetStatusCodeName(status));
        digitalWrite(pinVermelho, HIGH); delay(1000); digitalWrite(pinVermelho, LOW);
        return;
    }
    else {
        digitalWrite(pinVerde, HIGH); delay(1000); digitalWrite(pinVerde, LOW);
    }
    Serial.print(F("\nDados do bloco [")); Serial.print(bloco); Serial.print(F("]: "));
    //imprime os dados lidos
    for (uint8_t i = 0; i < MAX_SIZE_BLOCK; i++)
    {
        Serial.write(buffer[i]);
    }
    Serial.println(" ");
}
}
```



Gravação

```
//faz a gravação dos dados no cartão/tag
void gravarDados() {
    Serial.setTimeout(30000L) ; // aguarda 30 segundos para entrada de dados via Serial
    Serial.println(F("Insira os dados a serem gravados com o caractere '#' ao final\n[máximo de 16 caracteres:]"));
    //Prepara a chave - todas as chaves estão configuradas para FFFFFFFFh (Padrão de fábrica).
    for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;
    byte buffer[MAX_SIZE_BLOCK] = ""; //buffer para armazenamento dos dados que iremos gravar
    byte bloco = 1; //bloco que desejamos realizar a operação
    byte tamanhoDados; //tamanho dos dados que vamos operar (em bytes)
    //recupera no buffer os dados que o usuário inserir pela serial
    //serão todos os dados anteriores ao caractere '#'
    tamanhoDados = Serial.readBytesUntil('#', (char*)buffer, MAX_SIZE_BLOCK);
    //espaços que sobraem do buffer são preenchidos com espaço em branco
    for(byte i=tamanhoDados; i < MAX_SIZE_BLOCK; i++) {
        buffer[i] = ' ';
    }
    //Authenticate é um comando para autenticação para habilitar uma comunicação segura
    status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, bloco, &key, &(mfrc522.uid));
    if (status != MFRC522::STATUS_OK) {
        Serial.print(F("PCD_Authenticate() failed: ")); //erro de autenticação
        Serial.println(mfrc522.GetStatusCodeName(status));
        digitalWrite(pinVermelho, HIGH); delay(1000); digitalWrite(pinVermelho, LOW);
        return;
    }
    //Grava no bloco
    status = mfrc522.MIFARE_Write(bloco, buffer, MAX_SIZE_BLOCK);
    if (status != MFRC522::STATUS_OK) {
        Serial.print(F("MIFARE_Write() failed: ")); //erro de gravação
        Serial.println(mfrc522.GetStatusCodeName(status));
        digitalWrite(pinVermelho, HIGH); delay(1000); digitalWrite(pinVermelho, LOW);
        return;
    }
    else {
        Serial.println(F("MIFARE_Write() success: "));
        digitalWrite(pinVerde, HIGH); delay(1000); digitalWrite(pinVerde, LOW);
    }
}
```



Menu

```
//menu para escolha da operação
int menu()
{
    Serial.println(F("\nEscolha uma opção:"));
    Serial.println(F("0 - Leitura de Dados"));
    Serial.println(F("1 - Gravação de Dados\n"));

    //fica aguardando enquanto o usuário nao enviar algum dado
    while(!Serial.available()){};

    //recupera a opção escolhida
    int op = (int)Serial.read();
    //remove os proximos dados (como o 'enter ou \n' por exemplo) que vão por acidente
    while(Serial.available()) {
        if(Serial.read() == '\n') break;
        Serial.read();
    }
    return (op-48); //do valor lido, subtraímos o 48 que é o ZERO da tabela ascii
}
```





Em www.fernandok.com

- PRINCIPAL
- SOBRE FERNANDO K
- ARDUINO
- ESP8266
- ESP32
- LORAWAN
- MOTOR
- DISPLAY
- MATERIAIS
- DOWNLOAD

Receba o meu conteúdo GRATUITAMENTE

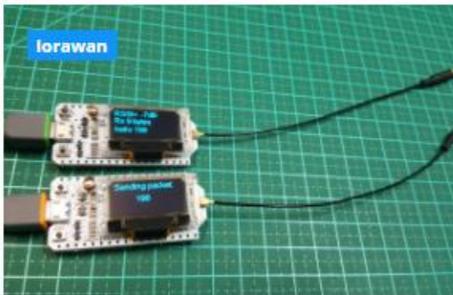
QUERO RECEBER GRÁTIS



Motor de Passo Nema 23 com Driver TB6600 e Arduino Due

by Fernando K Tecnologia - 2:44 PM
Hoje vamos voltar a falar de Motor de Passo. Vamos utilizar um Nema 23 que será controlado por um Driver TB6600 e um Arduino Due. É p...

Leia mais



ESP32 Longa Distância - LoRaWan

by Fernando K Tecnologia - 9:46 AM
Neste artigo vamos tratar da LoRaWAN, uma rede que vai longe gastando pouca energia. Mas, o quanto "longe"? Com o chip que uso no vídeo...

Leia mais



Motor de HD com Arduino

by Fernando K Tecnologia - 2:00 PM

QUAL ASSUNTO VOCÊ TEM

- Arduino
- ESP8266
- ESP32
- Motor
- Display
- Sensor

You may select multiple answers.
Votar Exibir resultados

Votos até o momento: 32
Dias restantes para votar: 49

FACEBOOK



Seu e-mail



Inscreva-se

