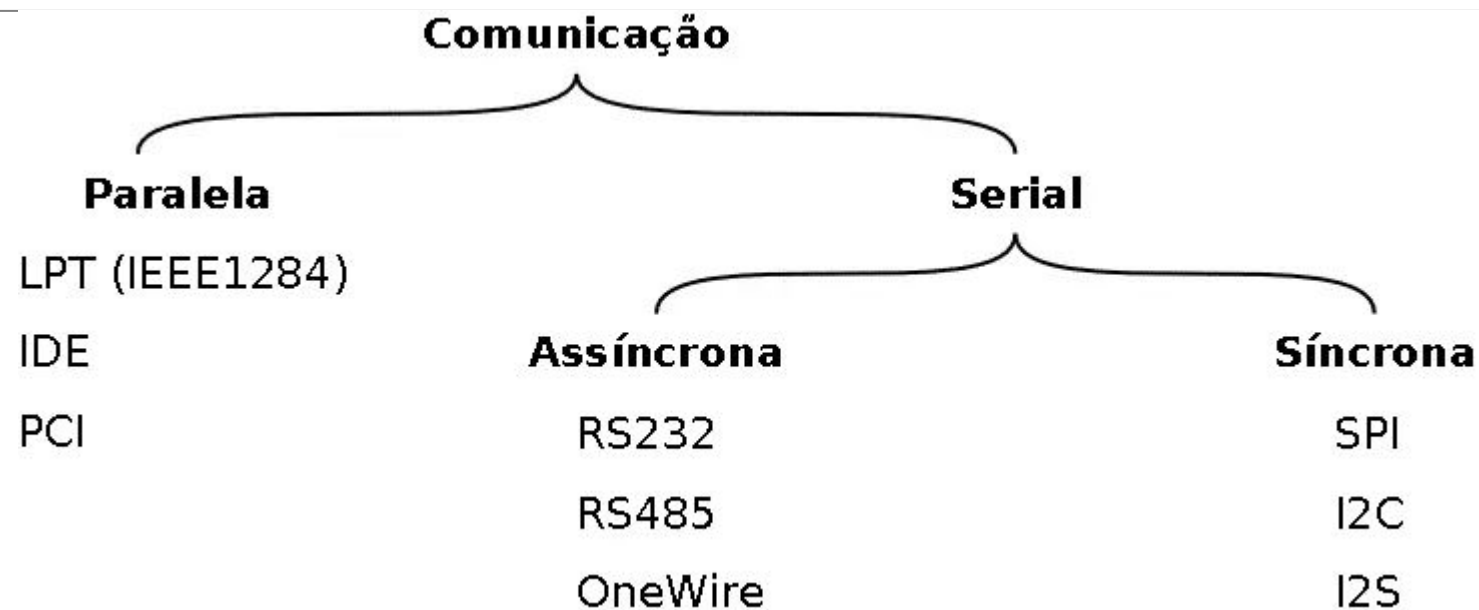


# Prática 01

---

Prof: Leonardo Augusto

# Comunicação SPI (Serial Peripheral Interface)



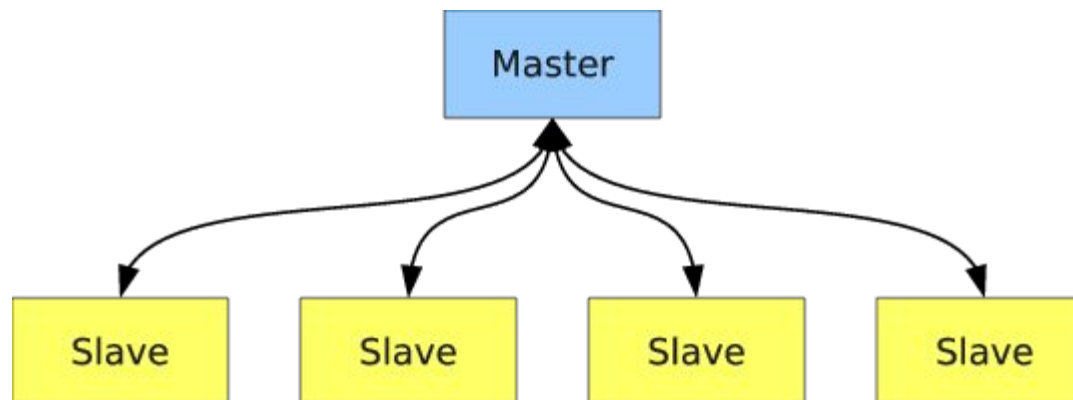
# Comunicação SPI

---

Tecnologia	Barramento de comunicação	Taxa máxima	Fluxo de dados
UART (RS232)	2 (sem controle de fluxo)	115.200 bps	Half ou Full Duplex
SPI	3 + nº de Slaves	2 Mbps	Full Duplex
I2C	2 (até 127 dispositivos)	400 Kbps	Half Duplex

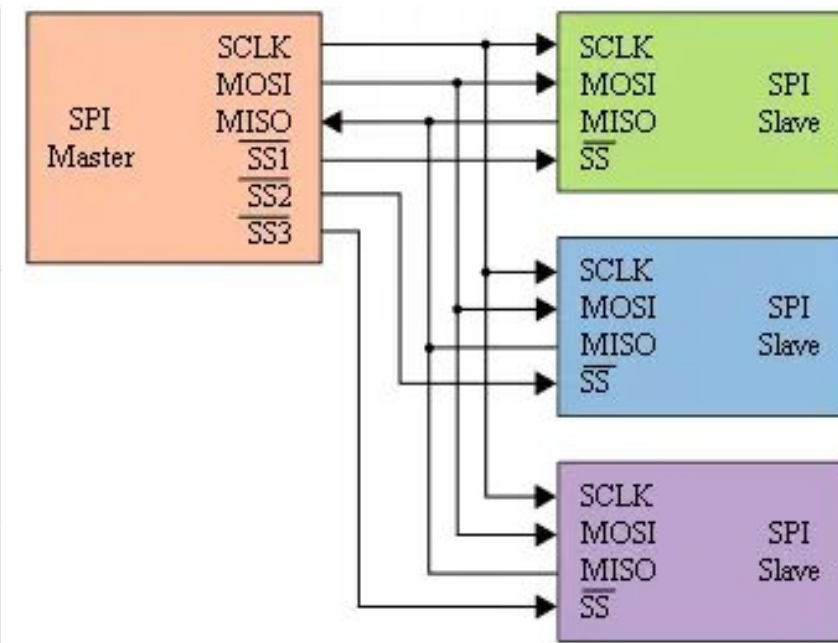
# Comunicação SPI

Na comunicação serial síncrona definimos também o conceito de Mestre-Escravo. Normalmente o gerador do sinal de sincronismo é definido como o Mestre (Master) da comunicação. Para os dispositivos que utilizam do sinal de sincronismo gerado, damos a definição de Escravo (Slave).



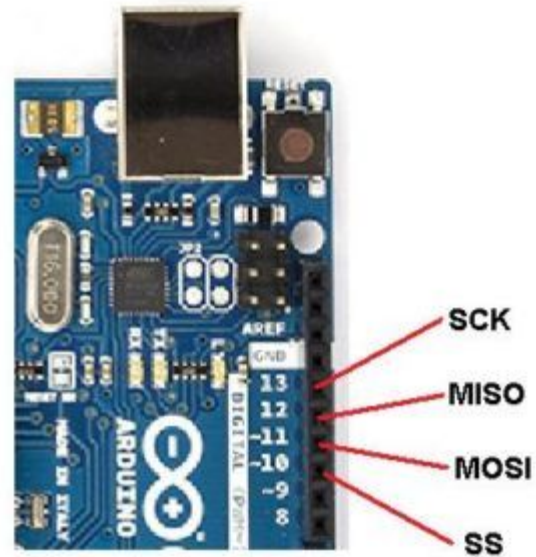
# Comunicação SPI

Pino	Nome Padrão	Significado	Nomes Alternativos
Do Master para o Slave	MOSI	Master Output Slave Input	SDO, DO, SO
Do Slave para o Master	MISO	Master Input Slave Output	SDI, DI, SI
Clock	SCLK	Serial Clock	SCK, CLK
Seleção de Slave	SS	Slave Select	CS, nSS, nCS



# Comunicação SPI

---



# Comunicação SPI

---

```
#include <SPI.H>
#define SS_SLAVE      3

void setup( void )
{
    pinMode ( SS_SLAVE , OUTPUT ) ;
    SPI.begin() ; // Inicializa a SPI do periférico.
}

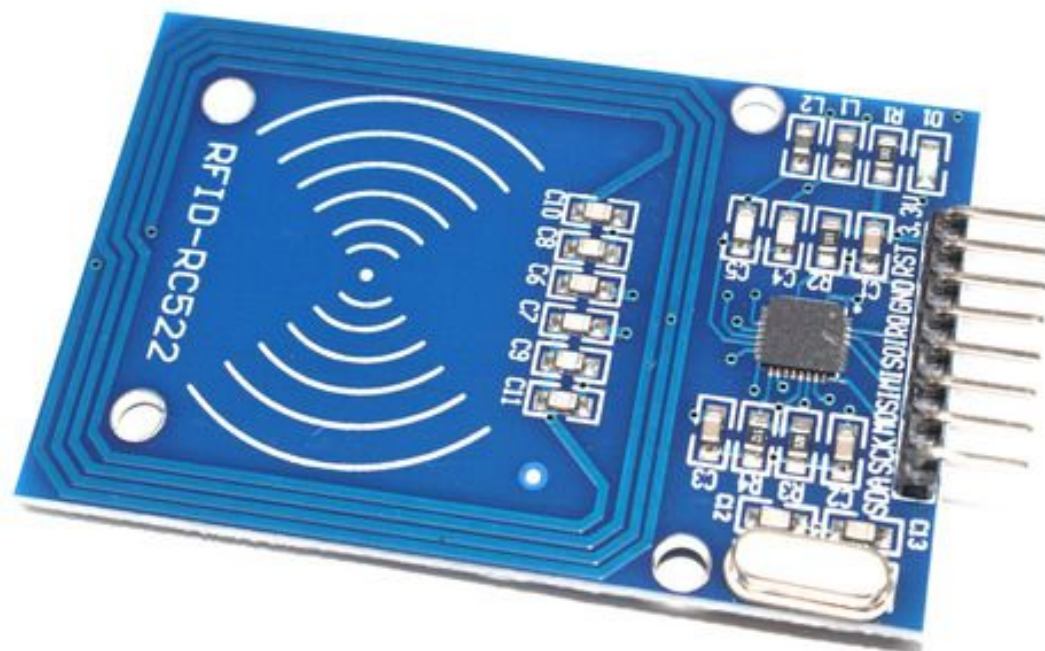
void loop( void )
{
    unsigned char retorno[ 3 ] ; // Retorno para cada byte.

    digitalWrite ( SS_SLAVE , LOW ) ; // Habilita o SS.
    retorno[ 0 ] = SPI.transfer( 0x00 ) ; // Envia 00h e recebe o retorno.
    retorno[ 1 ] = SPI.transfer( 0x3C ) ; // Envia 3Ch e recebe o retorno.
    retorno[ 2 ] = SPI.transfer( 0x5A ) ; // Envia 5Ah e recebe o retorno.
    digitalWrite ( SS_SLAVE , HIGH ) ; // Desabilita o SS.

    delay( 1000 ) ;
}
```

# Módulo RFID MFRC522

---





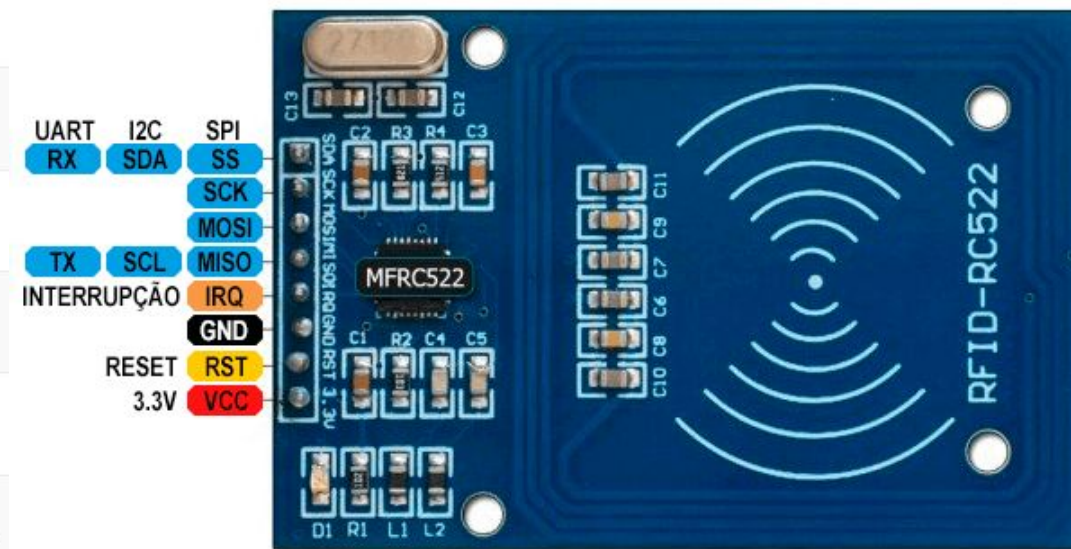
# MFRC522 - Especificações

---

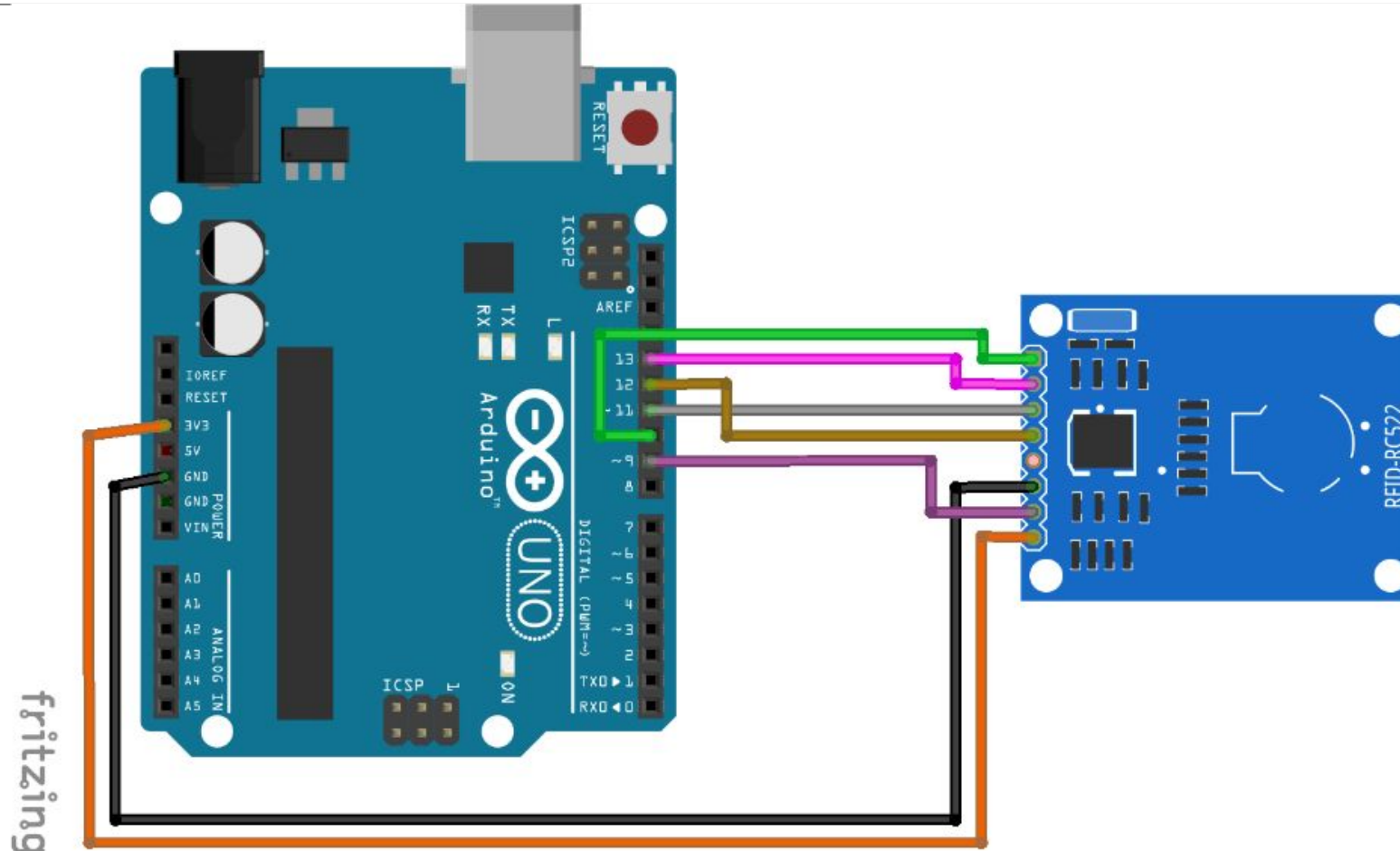
Tensão de Operação	2,5V à 3,3V
Corrente em Operação	13mA à 26mA
Corrente em Standby	10mA à 12mA
Frequência de Operação	13,56 MHz
Frequência Máxima do SPI	10 MHz
Frequência Máxima do I2C	3,4 MHz
Distância Mínima de Operação	50mm

# Pinout

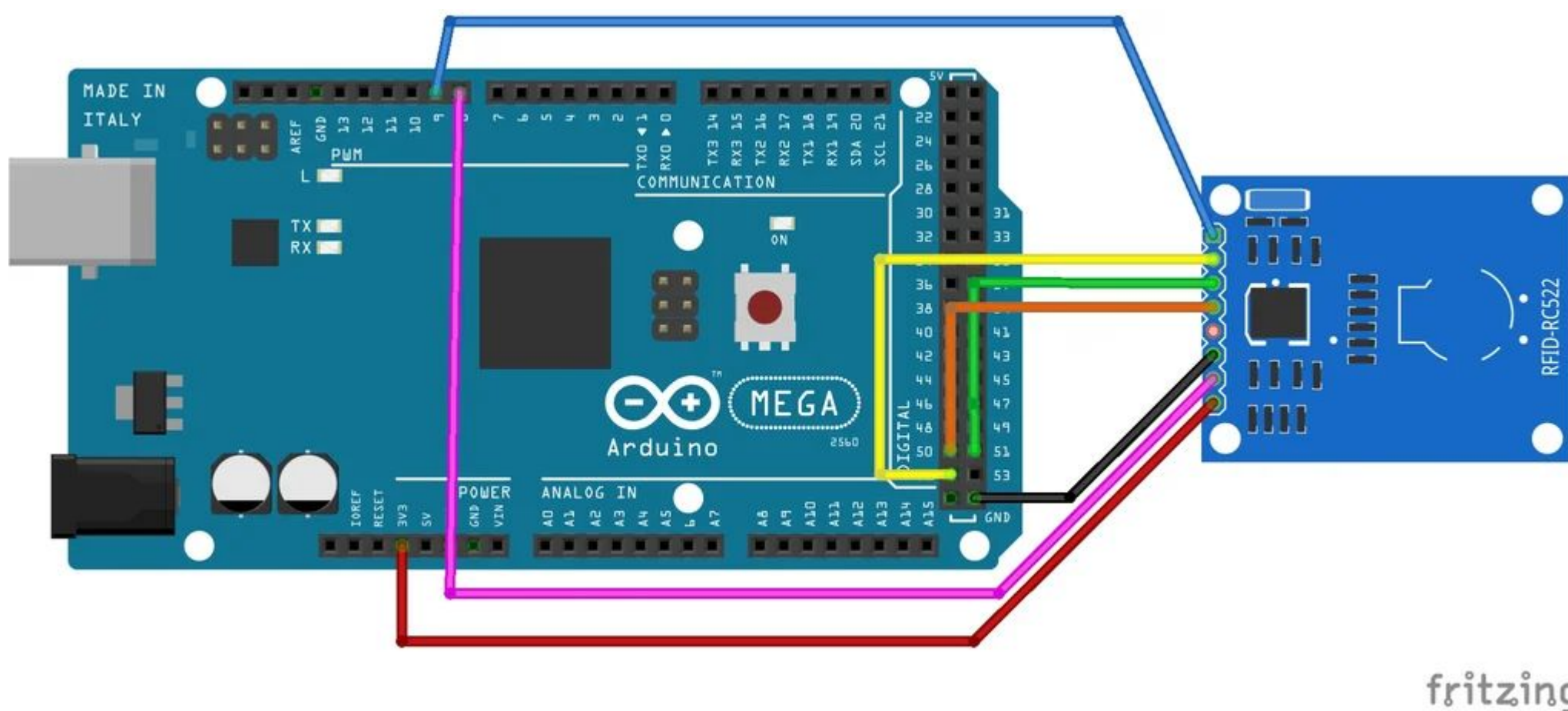
PINO	DESCRIÇÃO
SS ou CS	(Chip Select) seleciona o módulo ao qual deseja realizar uma comunicação
SCK	Origem do clock de sincronia entre os dispositivos
MOSI	Master Output Slave Input, é a saída de dados do Arduino
MISO	Master Input Slave Output, é a entrada de dados vindo do sensor para o Arduino.
IRQ	Gera um pulso de interrupção para o Arduino quando o sistema detecta atividade.
GND	Ground da alimentação
RST	Reinicia o módulo quando inserido um pulso neste pino
VCC	3.3V da alimentação



# Montagem com o Arduino Uno

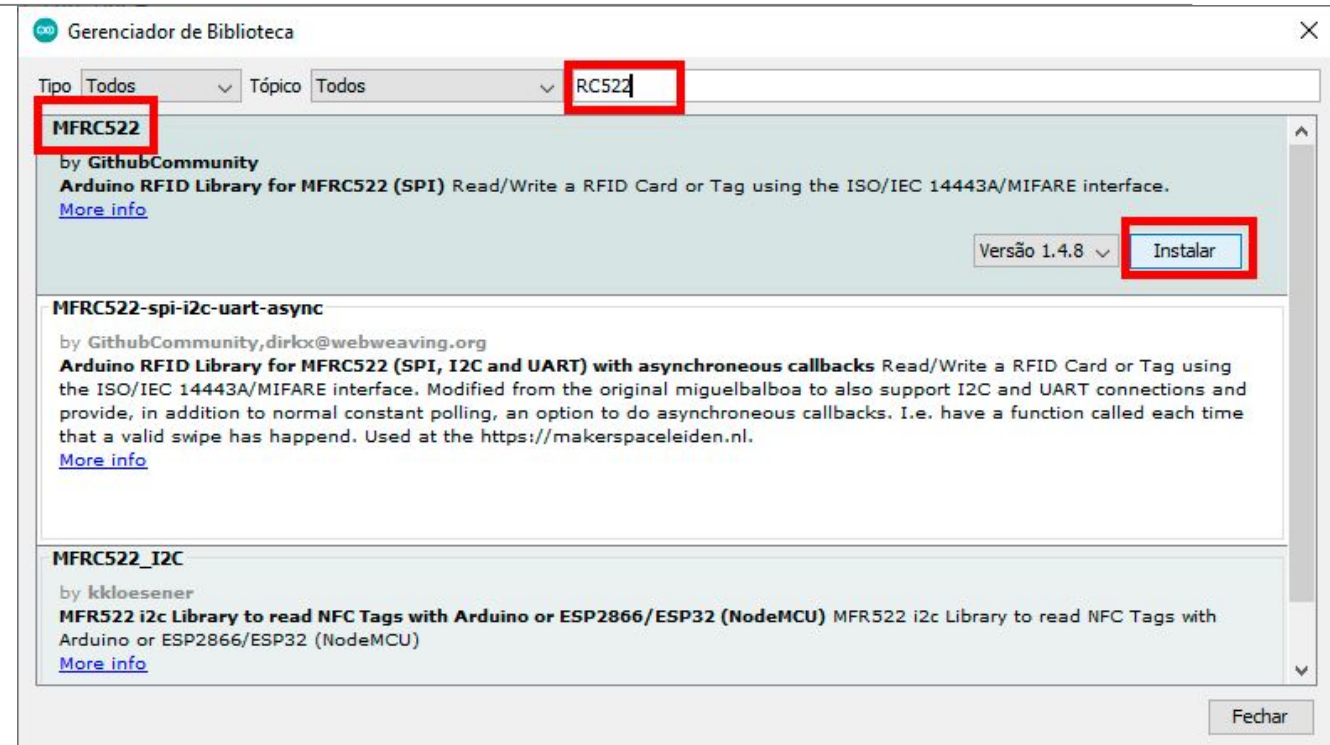
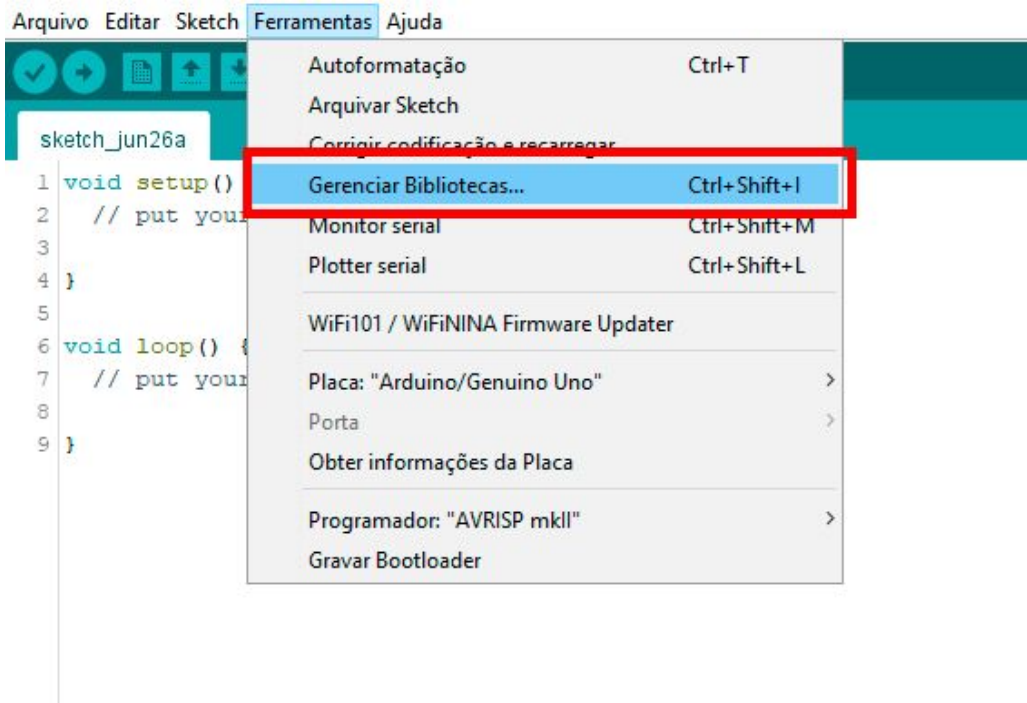


# Montagem com o Arduino Mega 2560





# Instalando lib



# Exemplo para Arduino Uno

```
//Inclui as bibliotecas do SPI e do RC522
#include <SPI.h>
#include <MFRC522.h>

//Definição dos pinos de conexão do projeto
#define PINO_SS 10
#define PINO_RST 9

//Cria o item para configurar o módulo RC522
MFRC522 mfrc522(PINO_SS, PINO_RST);

void setup() {
  Serial.begin(9600); // Inicializa a serial

  SPI.begin();// Inicializa a comunicação SPI

  mfrc522.PCD_Init(); // Inicializa o módulo MFRC522

  Serial.println("Sistema Inicializado: Aproxime o token");
}

void loop() {
  if (!mfrc522.PICC_IsNewCardPresent()) return;// Aguarda a
  aproximação do token

  if (!mfrc522.PICC_ReadCardSerial()) return; // Seleciona qual token
  vai ser utilizado

  Serial.print("UID da tag:"); // Mostra UID do token na serial

  String conteudo= ""; //Cria uma variável vazia, do tipo string

  byte letra;

  for (byte i = 0; i < mfrc522.uid.size; i++) {

    if(mfrc522.uid.uidByte[i] < 0x10){

      Serial.print(" 0");

    }

    else{

      Serial.print(" ");

    }

    Serial.print(mfrc522.uid.uidByte[i], HEX);// Printa a mensagem
    convertida em hexadecimal

    if(mfrc522.uid.uidByte[i] < 0x10){

      conteudo.concat(String(" 0"));

    }

    else{

      conteudo.concat(String(" "));

    }

    conteudo.concat(String(mfrc522.uid.uidByte[i], HEX));

  }

  Serial.println();

  conteudo.toUpperCase();//Coloca todas as letras da string em
  maiúscula

  if (conteudo.substring(1) == "60 85 73 A5") {

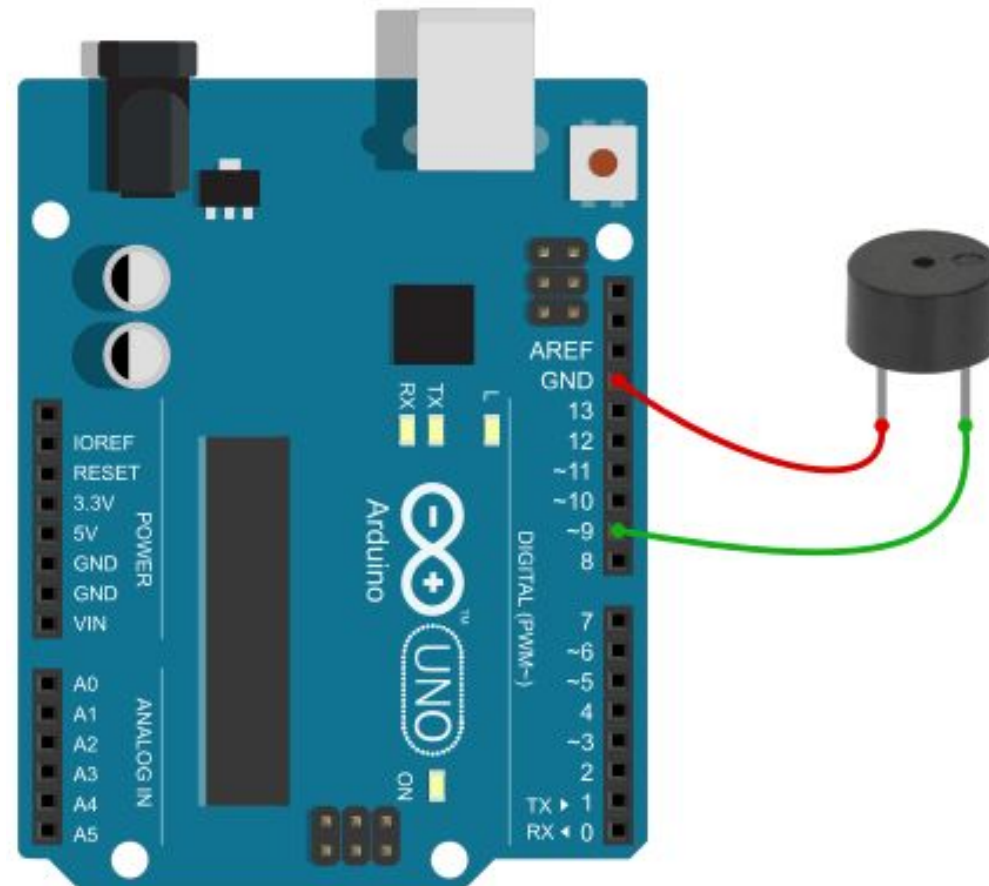
    Serial.println("Acesso Permitido!");

  }

  delay(1000);

}
```

# Buzzer



# Experimento

---

- ☐ Simule um controle de acesso com o módulo RFID. Utilize 2 leds e 1 buzzer. O led vermelho inicialmente ficará aceso. Ao inserir uma tag rfid previamente cadastrada, apague o led vermelho e acenda um led verde por 10 segundos e mostre na tela a mensagem “Acesso liberado”.
- ☐ Caso a tag não esteja cadastrada, mostre na tela a mensagem “cartão invalido” e pisque o led vermelho algumas 3 vezes.
- ☐ Em caso de inserir por 5 vezes uma tag que não esteja cadastrada, imprima a mensagem “SISTEMA BLOQUEADO”, o led vermelho ficará piscando e o buzzer acionado por 30 segundos.

**Desenvolva  
seu código!**