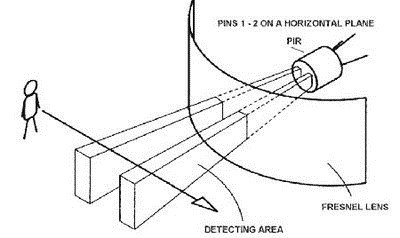
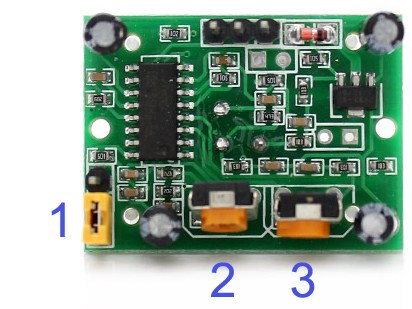
**Modo de ligação do sensor de presença que será utilizado**

O **Sensor de Movimento PIR HC-SR501** funciona detectando o calor emitido em forma de radiação infravermelha por pessoas, animais ou objetos. Está formado por um sensor piroelétrico dividido em duas zonas sensíveis; a radiação infravermelha e uma lente especial chamada “lente fresnel” que multiplica a zona de detecção, é essa tampa branca em forma de cúpula a cima do sensor piroelétrico.[1]



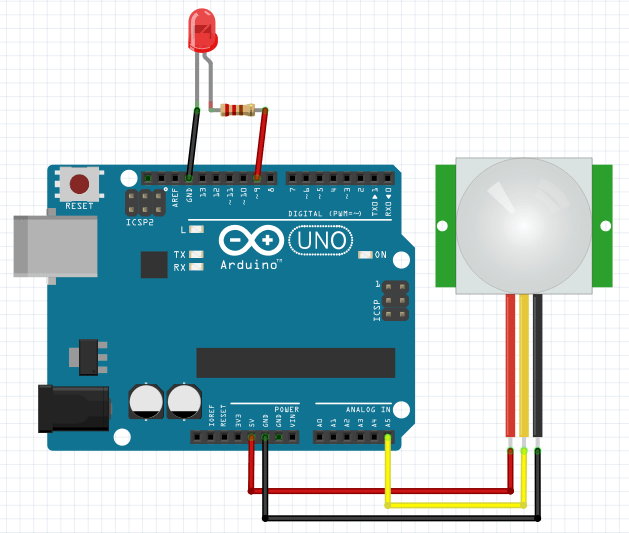
Contudo, o sensor deve ser ajustado de forma que pode-se alterar p alcance da distância medida, tempo de rearmado, comportamento, detecção repetitiva e outras características.



**#1: Modo de trabalho:** Através deste jumper, seleciona-se o modo de detecção; modo contínuo, ou seja, se o sensor detectar movimento de maneira continuada vai manter um sinal continuo, ou em modo repetição. Para o presente projeto será utilizado o modo de detecção continuo.

**#2: Sensibilidade do sensor:** Com este potenciómetro aumenta ou diminui a sensibilidade do sensor piroelétrico para ajustar a distância e a quantidade de movimento necessário para ativar, por exemplo, para distinguir entre pessoas e animais.

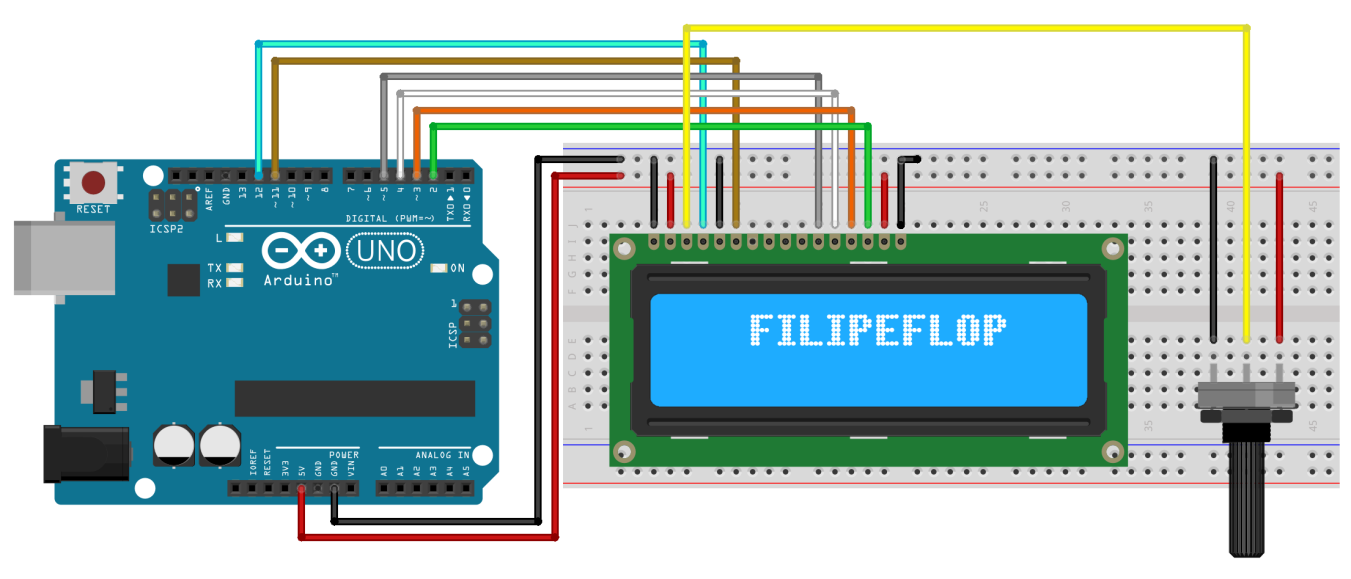
**#3: Temporizador:** Como foi escolhido o modo continuo não é necessário ajustar esse parâmetro, uma vez que não se faz necessário.



Portanto a ligação feita visa detectar uma pessoa passando e acionar o led vermelho que futuramente será substituído por um Buzzer, assim emitindo o sinal de que alguém passou.

**Display LCD 16×2**

**Esse display LCD tem 16 colunas e 2 linhas, com backlight (luz de fundo) azul e letras na cor branca. Para conexão, são 16 pinos, dos quais usamos 12 para uma conexão básica, já incluindo as conexões de alimentação (pinos 1 e 2), backlight (pinos 15 e 16) e contraste (pino 3).[3]**



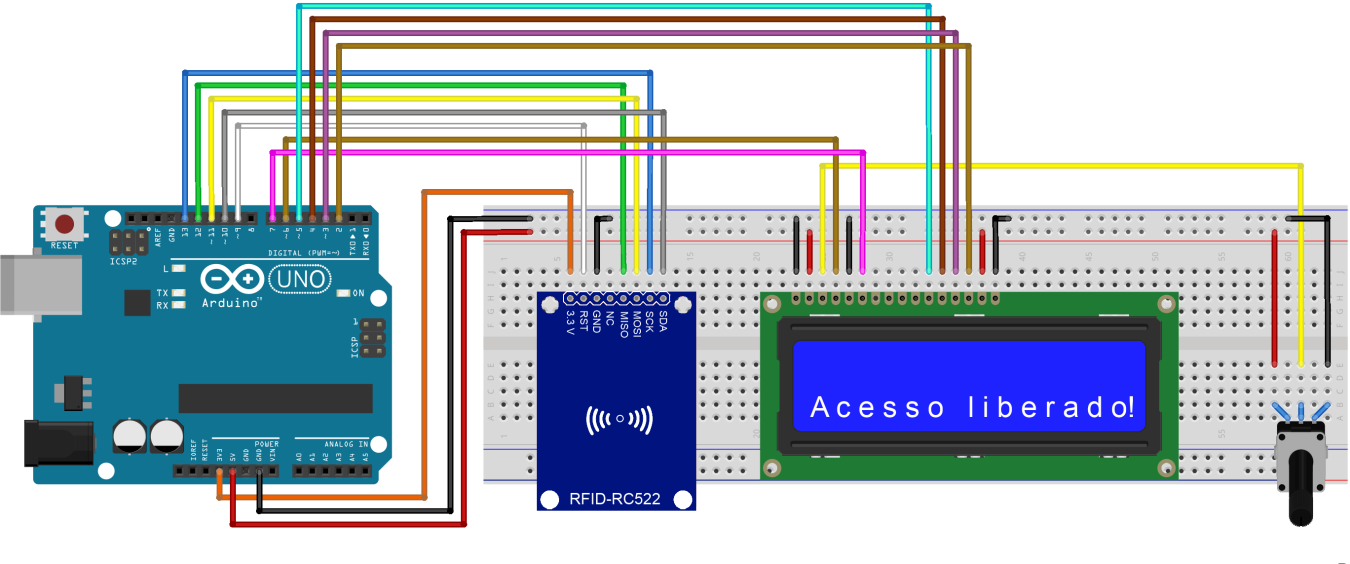
Será utilizado para instruir e ao mesmo tempo indicar o acesso da passagem ou a não leitura correta do RFID

**Montagem do sensor RFID**

**RFID** (Radio Frequency Identification, ou Identificação por Radiofrequência) costumam ser utilizados para controle de acesso e identificação de pessoas e equipamentos, seja por meio de crachás ou etiquetas aplicadas à produtos. No nosso dia-a-dia, podemos encontrar a **tecnologia RFID** nos pedágios (no popular “Sem Parar”), ou em cartões tipo [Bilhete Único](https://pt.wikipedia.org/wiki/Bilhete_%C3%9Anico), utilizados em várias cidades brasileiras para acesso ao transporte coletivo.[2]



Cada etiqueta/tag do leitor RFID tem a sua própria identificação (**UID**), e é com essa identificação que vamos montar um controle de acesso que irá ler o UID do cartão e exibir as informações de acesso num display LCD 16×2. Com pequenas alterações no programa é possível acionar as outras portas do Arduino e ligar motores, sensores, luzes e outros dispositivos. [2]

 Para a montagem do projeto e visualização da comunicação entre os componentes. Foi necessário montar o sensor de presença fazendo a comunicação com o RFID, ou seja, se alguma pessoa passou sem ter se identificado no RFID, será acionado um buzzer que fará um sinal sonoro alertando todas as pessoas que alguém esta tentando passar sem ter o real acesso ao ambiente.

**Referências**

[1] - <https://www.arduinoportugal.pt/uso-do-sensor-movimiento-hc-sr501-arduino/>

[2] - <https://www.filipeflop.com/blog/controle-acesso-leitor-rfid-arduino/>

[3] - https://www.filipeflop.com/blog/controlando-um-lcd-16x2-com-arduino/

https://www.filipeflop.com/blog/acendendo-uma-lampada-com-sensor-de-presenca/

**Códigos**

**Sensor de presença par acionar um indicador**.

//Declaramos os pinos que vamos a usar

int analogPin = A5;   //Analogico A5 para o sensor

int ledPin = 9;       //Digital 9 para o LED

//Declaramos uma variável para armazenar o valor que nos devolve o sensor

int val = 0;

void setup()

{

  Serial.begin(9600);         //Iniciamos a comunicação serial

  pinMode(ledPin, OUTPUT);    //Configuramos o pino digital 9 como saída

  pinMode(analogPin, INPUT);  //Configuramos o pino analogico A5 como entrada

}

void loop()

{

  val = analogRead(analogPin);    //Lemos o valor analogico que nos devolve o sensor

  if(val > 100)                   //Si o valor é maior que 100

  {

    Serial.println(val);          //Sacamos o valor do sensor por serial

    digitalWrite(ledPin,HIGH);    //Acendemos o LED

    delay(3000);                  //Fazemos uma pausa de 3 segundos

  }

  else                            //Se o valor é menor que 100

  {

     Serial.println(val);         //Obtemos o valor do sensor por serial

     digitalWrite(ledPin,LOW);    //Apagamos o LED

  }

}

**Sensor RFID**

/Programa : RFID - Controle de Acesso leitor RFID

//Autor : FILIPEFLOP

#include <SPI.h>

#include <MFRC522.h>

#include <LiquidCrystal.h>

#define SS\_PIN 10

#define RST\_PIN 9

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

LiquidCrystal lcd(6, 7, 5, 4, 3, 2);

char st[20];

void setup()

{

  Serial.begin(9600); // Inicia a serial

  SPI.begin(); // Inicia  SPI bus

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

  Serial.println("Aproxime o seu cartao do leitor...");

  Serial.println();

  //Define o número de colunas e linhas do LCD:

  lcd.begin(16, 2);

  mensageminicial();

}

void loop()

{

  // Look for new cards

  if ( ! mfrc522.PICC\_IsNewCardPresent())

  {

    return;

  }

  // Select one of the cards

  if ( ! mfrc522.PICC\_ReadCardSerial())

  {

    return;

  }

  //Mostra UID na serial

  Serial.print("UID da tag :");

  String conteudo= "";

  byte letra;

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

  {

     Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");

     Serial.print(mfrc522.uid.uidByte[i], HEX);

     conteudo.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));

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

  }

  Serial.println();

  Serial.print("Mensagem : ");

  conteudo.toUpperCase();

  if (conteudo.substring(1) == "ED 78 03 CA") //UID 1 - Chaveiro

  {

    Serial.println("Ola FILIPEFLOP !");

    Serial.println();

    lcd.clear();

    lcd.setCursor(0,0);

    lcd.print("Ola FILIPEFLOP !");

    lcd.setCursor(0,1);

    lcd.print("Acesso liberado!");

    delay(3000);

    mensageminicial();

  }

  if (conteudo.substring(1) == "BD 9B 06 7D") //UID 2 - Cartao

  {

    Serial.println("Ola Cartao !");

    Serial.println();

    lcd.clear();

    lcd.setCursor(0,0);

    lcd.print("Ola Cartao !");

    lcd.setCursor(0,1);

    lcd.print("Acesso Negado !");

    delay(3000);

    mensageminicial();

  }

}

void mensageminicial()

{

  lcd.clear();

  lcd.print(" Aproxime o seu");

  lcd.setCursor(0,1);

  lcd.print("cartao do leitor");

}

**Display LCD**

|  |  |
| --- | --- |
|  | //Programa: Teste de Display LCD 16 x 2  //Autor: FILIPEFLOP    //Carrega a biblioteca LiquidCrystal  #include <LiquidCrystal.h>    //Define os pinos que serão utilizados para ligação ao display  LiquidCrystal lcd(12, 11, 5, 4, 3, 2);    void setup()  {    //Define o número de colunas e linhas do LCD    lcd.begin(16, 2);  }    void loop()  {    //Limpa a tela    lcd.clear();    //Posiciona o cursor na coluna 3, linha 0;    lcd.setCursor(3, 0);    //Envia o texto entre aspas para o LCD    lcd.print("FILIPEFLOP");    lcd.setCursor(3, 1);    lcd.print(" LCD 16x2");    delay(5000);      //Rolagem para a esquerda    for (int posicao = 0; posicao < 3; posicao++)    {      lcd.scrollDisplayLeft();      delay(300);    }      //Rolagem para a direita    for (int posicao = 0; posicao < 6; posicao++)    {      lcd.scrollDisplayRight();      delay(300);    }  } |