Tutorial - Segurança residencial utilizando Arduino, sensores de presença PIR, finger print e shield ethernet

Bruno Tomé, Cláudio Menezes

[1]Departamento de Ciência da Computação
Instituto Federal de Minas Gerais,Campus Formiga (IFMG)
Formiga, MG – Brasil

ibrunotome@gmail.com, claudiomenezio@gmail.com

Resumo. *Neste tutorial é demonstrado como interligar o Arduino com os shield's PIR, ethernet e fingerprint. Bem como a interação do Arduino com os shield's e suas bibliotecas. Como projeto final temos a união destes componentes para um sistema de segurança residencial.*

1. Introdução

Neste tutorial será apresentado o passo a passo de como fazer as ligações entre o Arduino, *protoboard* e *shield's*. Em seguida mostraremos o código utilizado em cada *shield* e como resultado final temos um protótipo de sistema de segurança residencial que utiliza sensores PIR, sensor *fingerprint* e um *shield ethernet* para enviar as informações de presença para um servidor web, que envia *emails* ao dono da residência informando o local da movimentação.

1.1. Módulo Ethernet

O *shield ethernet* é simplesmente acoplado em cima do Arduino. Com o *software Arduino* aberto, vá em *File/Examples/Ethernet/WebClient* que serve como um cliente que envia informações ao servidor.

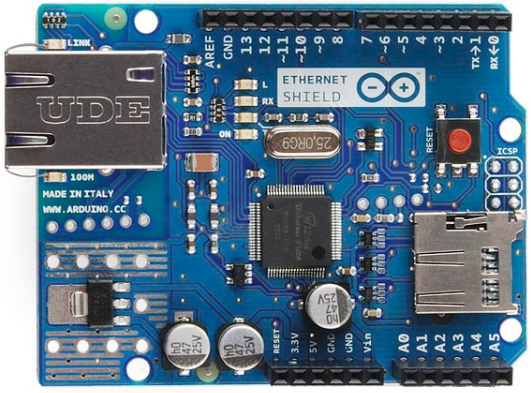

Figura 1. Decodificador para o display de 7 segmentos

2. Sensor de Presença - PIR

Na figura abaixo podemos ver como fazer a pinagem do sensor de presença PIR. São somente três saídas, uma saída para *GND, VCC* e um *OUTPUT* para passar as informações lidas:

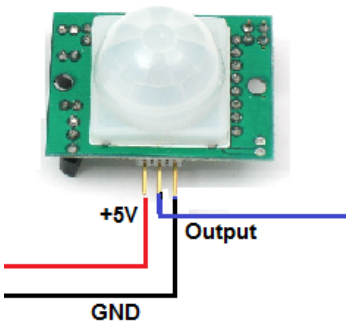

Figura 2. Descrição da pinagem sensor PIR

Em seguida é realizado as ligações das três saídas do sensor *PIR* no Arduino. A saída da esquerda é ligada a entrada de *5V* e a da direta é ligada ao *GND*, do arduino. O *OUTPUT* que é a saída do meio do sensor é conectada ao pino número 2 do Arduino, como mostrado na figura abaixo:

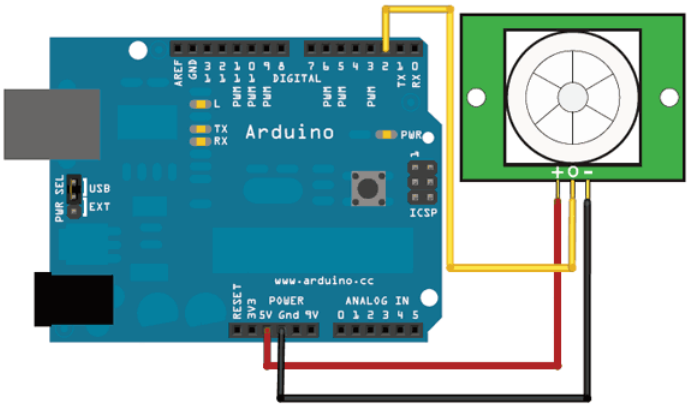

Figura 3. Sensor PIR ligado ao Arduino

2.1. Código sensor PIR

O código abaixo exibe no monitor serial a frase *Movimento detectado* quando o sensor é acionado:

```
int PIR_Tutorial = 2;

void setup() {
  // Initialize the serial
  Serial.begin(250000);
  // Set the pin for PIR_Tutorial
  pinMode(PIR_Tutorial, INPUT);
}

void loop() {
  // If the movement is detected, print to serial
  if (digitalRead(PIR_Tutorial) == HIGH) {
    Serial.println("Movimento detectado");
  }
}
```

Listing 1. Código fonte para o Sensor de presença PIR

3. Leitor Biométrico - Finger Print

A figura abaixo mostra a pinagem para o leitor biométrico, contendo quatro saídas, sendo duas para *GND, VCC* e outras duas, uma para *RX(Data IN)* e a outra *TX(Data OUT)* para passar as informações lidas no módulo:

Figura 4. Descrição da pinagem para o leitor biométrico

A próxima figura mostra como é realizada a ligação das saídas do Leitor Biométrico no Arduino. As saídas em vermelho e preto são ligadas no Arduino em *5V* e *GND* respectivamente. As duas saídas restantes são a *RX* em amarelo e em vermelho a *TX*, respectivamente ligadas nos pinos do Arduino *RX e TX*.

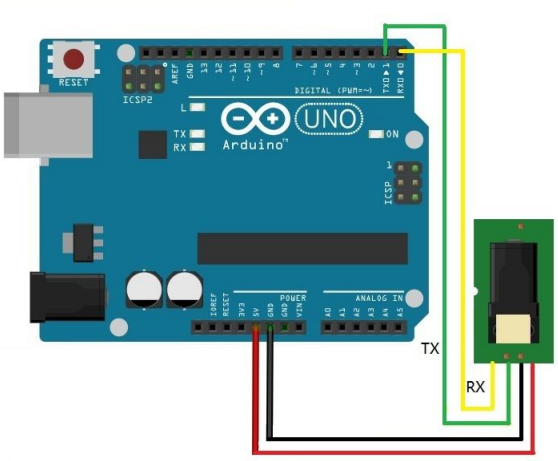

Figura 5. Leitor biométrico ligado ao Arduino

3.1. Código

Inicialmente para trabalhar com o leitor deve-se baixar a biblioteca *Adafruit library* disponível neste link: *https://github.com/adafruit/Adafruit-Fingerprint-Sensor-Library* e colocá-la na pasta de bibliotecas do Arduino.

Podemos entender como é feito o ato de gravar uma digital no módulo. Com o software *Arduino* aberto, vá em *File/Examples/Adafruit Fingerprint Sensor Library/enroll*

Um exemplo de código que mostra como é feito o *match* entre uma digital já cadastrada no módulo e uma recém inserida no módulo pode ser visto com o software *Arduino* aberto, vá em *File/Examples/Adafruit Fingerprint Sensor Library/fingerprint*.

4. Projeto Desenvolvido

O projeto desenvolvido é um protótipo de sistema de segurança residencial que utiliza sensores PIR, sensor *fingerprint* e um *shield ethernet* para enviar as informações

de presença para um servidor *web*, que envia *emails* ao dono da residência informando o local da movimentação.

O sistema requere a autenticação da digital para começar a funcionar e assim que ativado, os sensores PIR começam a captar os movimentos e enviar *POSTS* ao servidor contendo informações de quem está enviado, o que está acontecendo e em qual cômodo da residência está acontecendo. Para desativar o sistema, basta autenticar com a digital novamente.

4.1. Materiais Utilizados

- Um Arduino UNO
- Um shield Ethernet W5100
- Um Leitor Biométrico De Impressão Digital
- Quatro sensores de infra vermelho por temperatura (PIR)
- Protoboard

4.2. Código Arduino

Abaixo está listado o código utilizado no Arduino para realização do projeto.

```
#include <SPI.h>
#include <Ethernet.h>
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>

/*##################################
  # FINGERPRINT SETTINHGS
  ##################################*/
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
bool fingerPrintValid = true;

/*##################################
  # PIR SETTINHGS
  ##################################*/

int PIRSala = 4;
int PIRCozinha = 5;
int PIRDormitorio_casal = 6;
int PIRDormitorio_1 = 7;
int calibrationTime = 5;
bool enviouSala = false;
bool enviouCozinha = false;
bool enviouDormitorioCasal = false;
bool enviouDormitorio1 = false;

/*##################################
  # ETHERNET SETTINHGS
  ##################################*/

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
// IPAddress server(45, 55, 185, 215);
char server[] = "www.brunotome.com"; // Change to your server here
String data;

// Set the static IP address to use if the DHCP fails to assign
IPAddress ip(192, 168, 0, 177);

/**
   Send a post to my server with the data parameters
*/
void sendPost(String typeAlert, String location) {
  /* Initialize the Ethernet client library with the IP address and port of the server
     that you want to connect to (port 80 is default for HTTP): */
```

```
    EthernetClient client;
    // start the Ethernet connection:
    if (Ethernet.begin(mac) == 0) {
      Serial.println("Failed to configure Ethernet using DHCP");
      // try to congifure using IP address instead of DHCP:
      Ethernet.begin(mac, ip);
    }

    delay(1000);

    /*
        id = 1 // pk of the user Bruno Tome
        &typeAlert = arrombamento || movimentacao
        &location = sala || cozinha || dormitorio_1 || dormitorio_2 || dormitorio_casal
    */
    data = "id=1&typeAlert=" + typeAlert + "&location=" + location;

    if (client.connect(server, 80)) {
      client.println("POST /projects/embarcados/update.php HTTP/1.1"); // Change to your post address here
      client.println("Host: www.brunotome.com");
      client.println("Content-Type: application/x-www-form-urlencoded");
      client.print("Content-Length: ");
      client.println(data.length());
      client.println();
      client.print(data);
      Serial.println("POST enviado: " + data);
    } else {
      Serial.println("Falhou ao conectar");
    }
}

/**
    Check movements on any PIR sensor and call the sendPost method with
    the alertType and location
*/
void checkMovement() {
    if ((digitalRead(PIRSala) == HIGH) && enviouSala == false) {
      enviouSala = true;
      sendPost("movimentacao", "sala");
    }

    if ((digitalRead(PIRCozinha) == HIGH) && enviouCozinha == false) {
      enviouCozinha = true;
      sendPost("movimentacao", "cozinha");
    }

    if ((digitalRead(PIRDormitorio_casal) == HIGH) && enviouDormitorioCasal == false) {
      enviouDormitorioCasal = true;
      sendPost("movimentacao", "dormitorio_casal");
    }

    if ((digitalRead(PIRDormitorio_1) == HIGH) && enviouDormitorio1 == false) {
      enviouDormitorio1 = true;
      sendPost("movimentacao", "dormitorio_1");
    }
}

/**
    Check if inputed finger match with the id of the owner
*/
bool getFingerprintIDez() {
    uint8_t p = finger.getImage();
    if (p != FINGERPRINT_OK) return false;

    p = finger.image2Tz();
    if (p != FINGERPRINT_OK) return false;

    p = finger.fingerFastSearch();
    if (p != FINGERPRINT_OK) return false;
```

```
  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print(" with confidence of "); Serial.println(finger.confidence);
  return true;
}

void setup() {
  while (!Serial);
  // Open serial communications and wait for port to open:
  Serial.begin(250000);
  Serial.println(" Inicializando ... ");
  pinMode(PIRSala, INPUT);
  pinMode(PIRCozinha, INPUT);
  pinMode(PIRDormitorio_casal, INPUT);
  pinMode(PIRDormitorio_1, INPUT);
  digitalWrite(PIRSala, LOW);
  digitalWrite(PIRCozinha, LOW);
  digitalWrite(PIRDormitorio_casal, LOW);
  digitalWrite(PIRDormitorio_1, LOW);
  // give the sensor some time to calibrate
  Serial.print("Calibrando sensores PIR ");
  for (int i = 0; i < calibrationTime; i++) {
    Serial.print(".");
    delay(1000);
  }
  Serial.println(" pronto");
  // set the data rate for the sensor serial port, must be 57600
  finger.begin(57600);
  delay(1000);
}

void loop() {
  if (fingerPrintValid) {
    Serial.println("Aguardando uma digital valida para ativar o sistema");
    if (getFingerprintIDez()) {
      Serial.println("Sistema ativado");
      fingerPrintValid = false;
      enviouSala = false;
      enviouCozinha = false;
      enviouDormitorioCasal = false;
      enviouDormitorio1 = false;
    }
  } else {
    checkMovement();
    if (getFingerprintIDez()) {
      fingerPrintValid = true;
      Serial.println("Sistema desativado");
    }
  }

  // Do again after 1 second
  delay(1000);
}
```

Listing 2. Código fonte do projeto

4.3. Web

O protótipo contou com uma interface *web* para analisar a situação atual da residência e também com o envio de emails para o dono da mesma. Assim que o usuário realiza o *login* no sistema ele vê a atualização mais recente sobre sua residência e tem acesso a um menu listando todos os alertas feitos até o momento com suas respectivas datas.

A figura 6 abaixo mostra a interface web criada para o projeto.

Figura 6. Interface web para controle dos alertas recebidos

A figura 7 mostra um exemplo de email recebido pelo dono da residência:

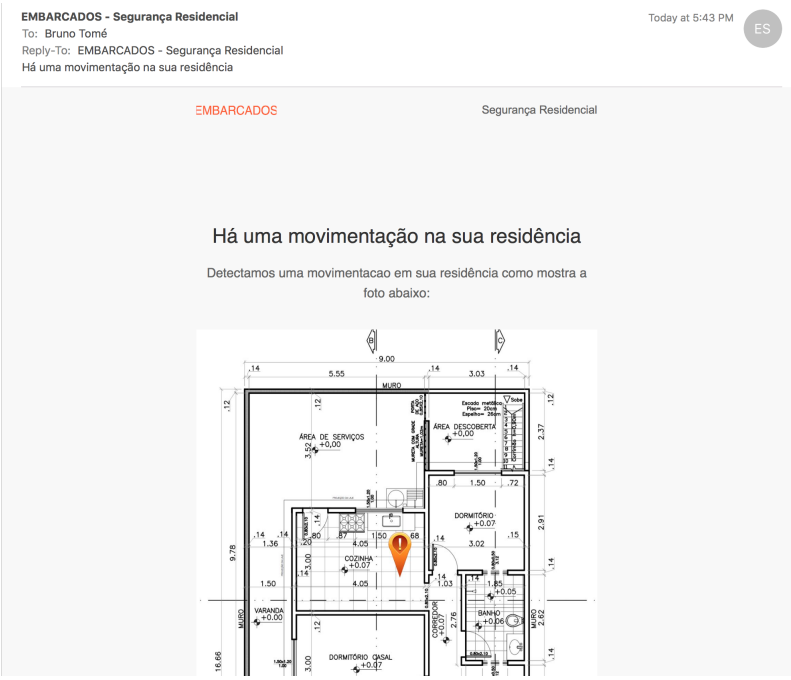

Figura 7. Exemplo de email recebido pelo dono da residência

4.3.1. Código PHP

Abaixo está listado o código PHP que recebe os posts enviados pelo Arduino, armazena tais informações e envia emails ao proprietário da residência quando necessário.

```php
<?php
    require_once('controllers/session.php');
    // Check if the POST have all the necessary attributes
    if (!empty($_POST['id']) && !empty($_POST['typeAlert']) && !empty($_POST['location'])) {
        // Get the owner of the POST
        $userAux = $user->find($_POST['id']);
        if (($userAux != NULL)) {
            $alert = new Alert();
            $alertUserAux = $alert->getLastAlertUser($userAux['id']);
            $alert->id_user = $userAux['id'];
            $alert->type_alert = $_POST['typeAlert'];
            $alert->location = $_POST['location'];
            $alert->date_update = $now;

            if (($alertUserAux == NULL) || (($alertUserAux['type_alert'] != $_POST['typeAlert']) || ($alertUserAux['location'] != $_POST['location'])) && ($_POST['typeAlert'] == 'normal')) { // Just update the alert to "normal" situation
                $alert->create();
            } else if (($alertUserAux == NULL) || (($alertUserAux['type_alert'] != $_POST['typeAlert']) || ($alertUserAux['location'] != $_POST['location'])) && ($_POST['typeAlert'] != 'normal')) { // If new situation is different from the last one, create a new record and send a email to the owner
                $alert->create();
                $email = new Email();
                $email->send($userAux['email'], $_POST['typeAlert'] . '_' . $_POST['location']);
            } else { // Just update the situation
                $alert->save($alertUserAux['id']);
            }
        }
    }
```

Listing 3. Código fonte do projeto

4.4. Resultado final

Abaixo na figura 8, o circuito do protótipo final do sistema de segurança residencial.

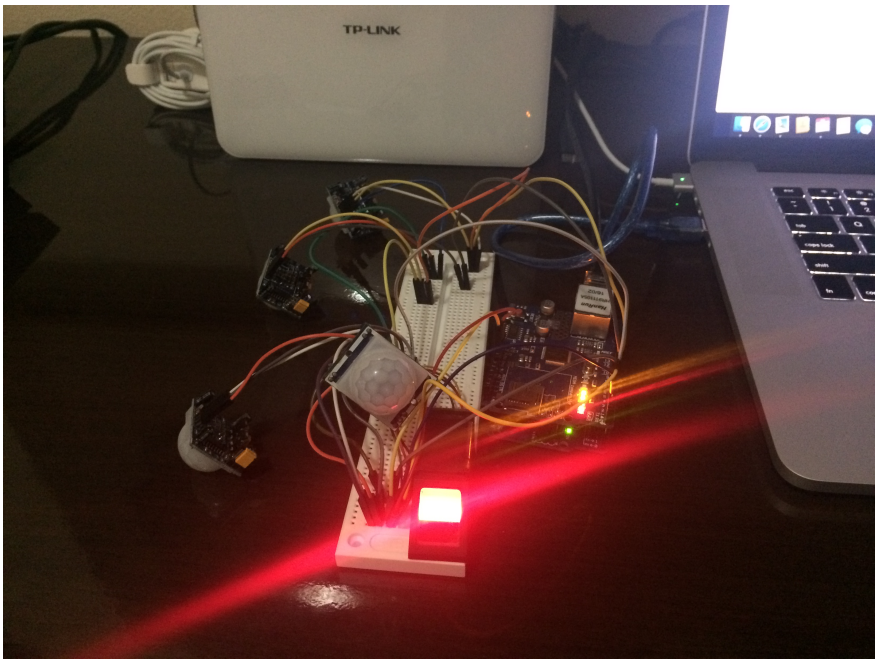

Figura 8. Exemplo de email recebido pelo dono da residência

5. Bibliografia

Getting Started with the Arduino Ethernet Shield. Disponível em: <https://www.arduino.cc/en/Guide/ArduinoEthernetShield>

Manual Adafruit Optical Fingerprint Sensor

Tutorial: como utilizar o Sensor PIR (Passive Infrared) com Arduino. Disponível em: <http://labdegaragem.com/profiles/blogs/tutorial-emissor-e-receptor-infra-vermelho-com-arduino>