Sistema de segurança

Fernando Lopes, Nil Silva

Instituto Politécnico Guarda

Abstract

Criação de um Sistema de Segurança. Este projeto baseia-se num sensor de movimento que, através da plataforma Arduíno, envia os dados para um programa em Python, este processa os dados e coloca-os na plataforma ThingSpeak. Se um botão não for pressionado em menos de 30 segundos, o python vai enviar um email de aviso.

Palavras-chave: Arduíno; Sensor de movimento; Python;

**Introdução**

Criar um Sistema de Segurança que capte movimento e envie dados para a Internet através dos conhecimentos adquiridos na disciplina Arquitetura de Computadores. Existem várias soluções bastante superiores como o FrontPoint, ADT, Vivint, etc, mas este projeto será muito simplificado devido a fatores como o tempo, conhecimento técnico e dinheiro. Esperasse desenvolver um sistema de segurança eficaz e simples.

O projeto esta disponível através do seguinte [link](https://github.com/NilSilva/SistemaDeSeguranca).

**Métodos**

Utilizando um microcontrolador Arduino, um sensor de movimento de infravermelhos e um botão, criar um circuito (Figura 3) para atingir este objetivo. Em código Arduino Language controlamos o arduino utilizando uma máquina de estados (Figura 2), que envie para um programa em python os dados capturados dependendo do estado em que se encontra. No programa python, processar e enviar os dados para o ThingSpeak e se passados 30 segundos (depois de capturar movimento) o botão não tiver sido pressionado, enviar um email. Depois o sensor espera um minuto antes de poder voltar a captar movimento, como pode ser visto no fluxograma (Figura 1).

**Algoritmo:**

1-Inicializar o sensor

2-Esperar 60 segundos

3-Ler dados do sensor

4- Se detetar movimento esperar 30 segundos ou que alguém carregue no botão.

5-Se ninguém carregar no botão nos 30 segundos, enviar dados para o ThingSpeak e enviar um email.

6-Se alguém carregar no botão envia dados para o ThingSpeak.

7-Esperar 60 segundos para poder capturar novamente movimento.

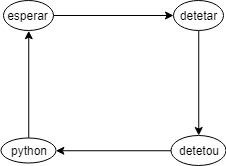
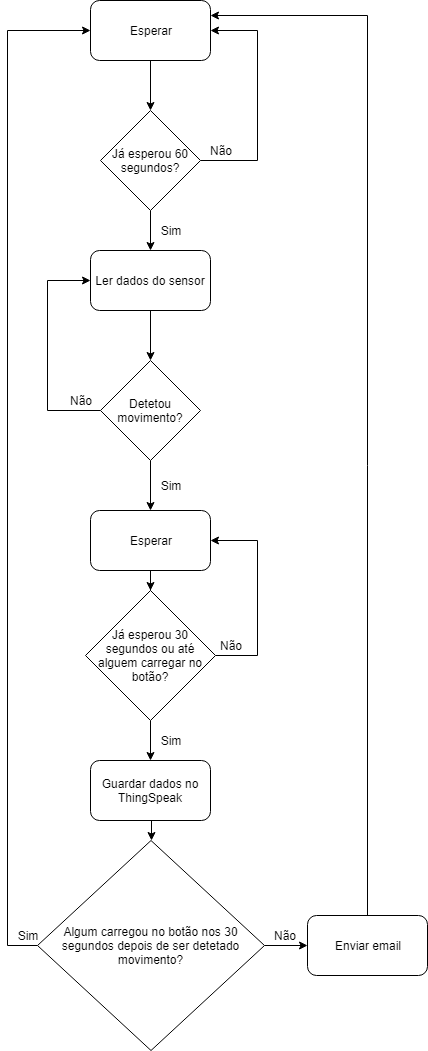


Figura 1. Fluxograma

Figura 2. Diagrama da máquina de estados

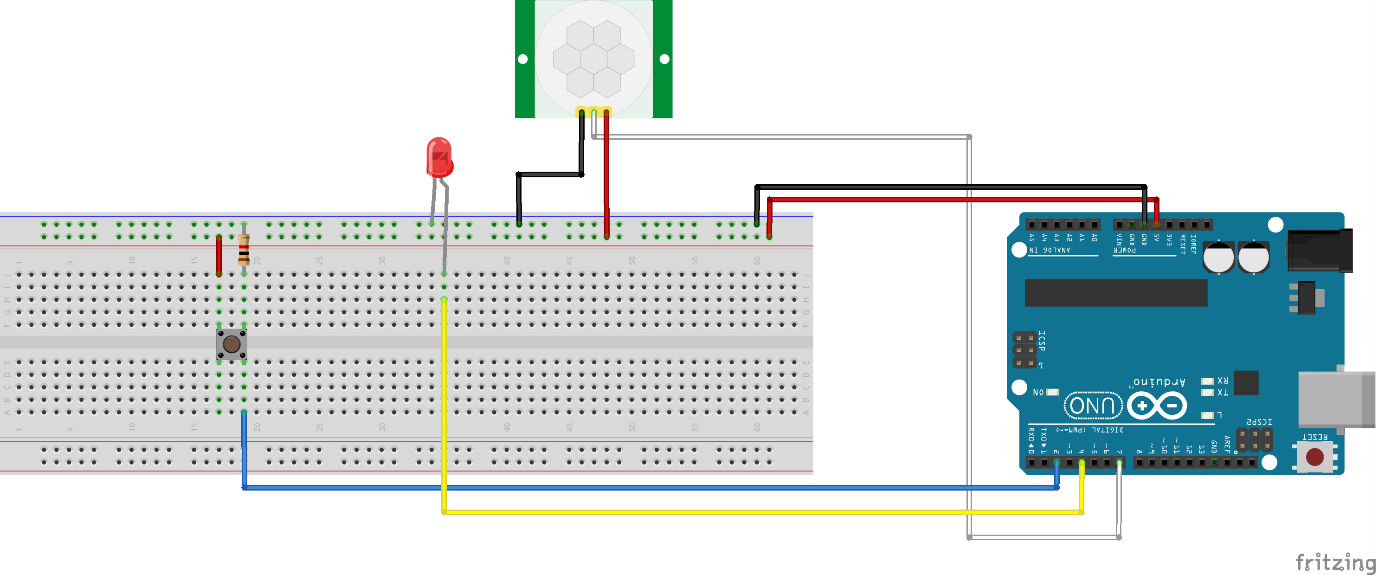
****

Figura 3.Circuito

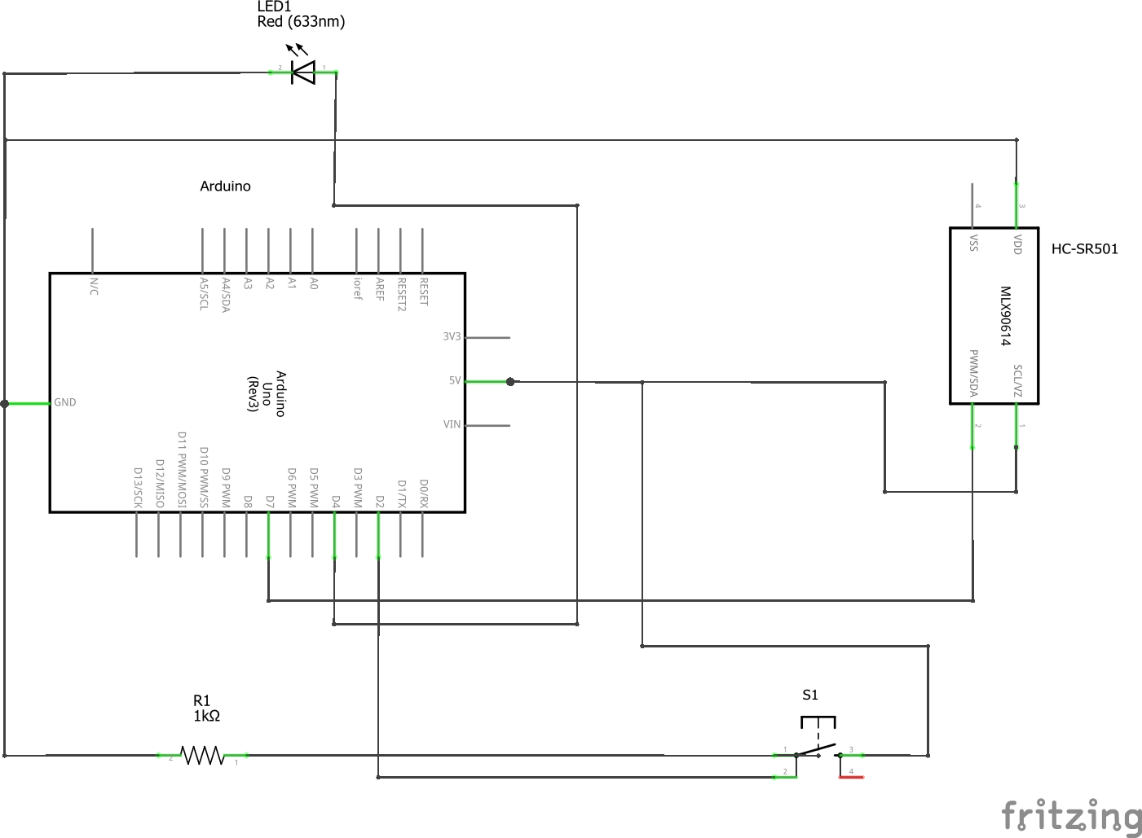
****

Figura 4.Esquemático

**Resultados**

Para testarmos o projeto foram feitos 30 testes (15 a carregar no botão e 15 sem carregar no botão) ao longo de vários dias e em duas montagens diferentes. O projeto funcionou devidamente em todos os testes nas duas montagens.

**Conclusão**

Os resultados foram satisfatórios, pois foi alcançado o principal objetivo que era criar um sistema de segurança eficaz. O Sistema tem algumas limitações, é necessário esperar 1 minuto para o sensor iniciar a captura de movimento, o alcance é de apenas 3 a 7 metros, deteta movimentos que não devia como por exemplo cortinas e é um Sistema de Segurança pouco robusto. Devido a recursos limitados, como tempo e dinheiro, não foi possível tornar o Sistema mais complexo. Algumas das ideias para melhorar o Sistema eram adicionar uma camera para captar movimento, adicionar um comando para ligar desligar o sensor, adicionar um código para desativar o sensor ao invés de um botão.

References

Informação sobre o sensor utilizado:

<http://henrysbench.capnfatz.com/henrys-bench/arduino-sensors-and-input/arduino-hc-sr501-motion-sensor-tutorial/>

**Anexos**

**Código ARDUINO LANGUAGE**

//--Constantes--

//definição dos pins

const int LED\_PIN = 4; // LED pin

const int PIR\_PIN = 7; // HC-S501 pin

const int BUTTON\_PIN = 2; //button pin

//definição dos estados

const int ESPERAR = 0;// espera durante 60 segundos

const int DETETAR = 1;// lê dados do sensor

const int DETETOU = 2;// detetou movimento

const int PYTHON = 3;// envia dados para o python

//--Variaveis--

long time = 0;

int pirValue;

int button = 1;

int state = ESPERAR;

void setup() {

pinMode(LED\_PIN, OUTPUT);

pinMode(PIR\_PIN, INPUT);

attachInterrupt(digitalPinToInterrupt(BUTTON\_PIN), butt, RISING);

Serial.begin(9600);

}

void loop() {

switch (state) {

case ESPERAR:

time = millis();

digitalWrite(LED\_PIN, LOW);

while (millis() - time < 60000);

digitalWrite(LED\_PIN, HIGH);

state = DETETAR;

break;

case DETETAR:

pirValue = digitalRead(PIR\_PIN);

if (pirValue == 1) {

state = DETETOU;

} else {

state = DETETAR;

}

break;

case DETETOU:

time = millis();

digitalWrite(LED\_PIN, LOW);

while (millis() - time < 30000);

state = PYTHON;

break;

case PYTHON:

Serial.println(button);

state = ESPERAR;

button = 1;

break;

}

}

void butt() {

if (state == DETETOU) {

time = 0;

button = 0;

}

}

**Código Python**

import serial

import smtplib

from email.message import EmailMessage

import urllib.request

import time

def enviaMail():

from\_mail = 'APCS123456789'

palavra\_chave = 'olaola123456789'

to\_mail = 'nil\_silva\_12@outlook.com'

subject = 'ALERTA'

content = 'Movimento detetado.'

msg = EmailMessage()

msg['from'] = from\_mail

msg['to'] = to\_mail

msg['subject'] = subject

msg.set\_content(content)

mail = smtplib.SMTP('smtp.gmail.com', 587)

mail.ehlo()

mail.starttls()

try:

mail.login(from\_mail, palavra\_chave)

print('login successful')

except:

print('login failed')

try:

mail.sendmail(from\_mail, to\_mail, msg.as\_string())

print('email sent')

except:

print('email not sent')

ser = serial.Serial(port='COM13',baudrate=9600)

pirValue = -1

url = 'http://api.thingspeak.com/update?api\_key=18UCXEIIBBDNJDQ5&field1='

print(time.asctime(time.localtime(time.time())))

while(True):

v1 = str(ser.readline())

v2 = v1.split('\'')

v3 = v2[1].split('\\')

pirValue = int(v3[0])

print(pirValue)

print(time.asctime(time.localtime(time.time())))

if pirValue == 0:

f = urllib.request.urlopen(url + str(pirValue))

f.read()

f.close()

elif pirValue ==1:

f = urllib.request.urlopen(url + str(pirValue))

f.read()

f.close()

enviaMail()