

Projeto Final - Sistemas Embarcados

Sistema de Monitoramento de Segurança por Presença

Pedro Willian Santos Ribeiro Calile, 15/0144989
Programa de Graduação em Engenharia Eletrônica, Faculdade
do Gama
Universidade de Brasília
Gama, DF, Brasil
calile-@hotmail.com

Rafael Alves Magalhães, 12/0020718
Programa de Graduação em Engenharia Eletrônica, Faculdade
do Gama
Universidade de Brasília
Gama, DF, Brasil
magalhaesrafael07@gmail.com

Resumo — Este trabalho visa implementar e desenvolver um sistema de monitoramento autônomo por presença e captura de imagem, usando o Raspberry Pi. Quando um movimento é detectado, a câmera inicia automaticamente a captura de imagem, então o dispositivo Raspberry Pi alerta o usuário via notificação no smartphone da presença de algo movimentando-se e recebe a imagem do ocorrido.

Palavras-chave — Raspberry Pi; monitoramento; sistema; smartphone; crianças;

I. INTRODUÇÃO

Este trabalho visa implementar um sistema de monitoramento de ambientes residenciais e/ou comerciais que controlado por uma Raspberry Pi envia um sinal de alerta para um email cadastrado. Esse alerta será enviado quando for detectado um movimento pelo sensor de presença PIR que irá acionar uma câmera que então irá capturar duas imagens que serão enviadas para um e-mail cadastrado.

II. JUSTIFICATIVA

A ausência de um sistema de segurança autônomo e inteligente de monitoramento nas residências e comércios pode acarretar um tempo de espera muito grande até que o crime seja percebido pelas vítimas, retardando o socorro policial necessário. Dessa forma, a implementação de um sistema de monitoramento que proporcione os benefícios de uma notificação rápida para os usuários, pode ajudar no reconhecimento do criminoso e servir, por exemplo, como material de prova do ocorrido.

Neste ponto de controle será escrito o código principal para leitura do sensor PIR, refinamento do protótipo, acrescentando recursos básicos de sistema (múltiplos processos e threads, pipes, sinais, semáforos, MUTEX etc.)

III. OBJETIVOS

Será desenvolvido um hardware composto por plataforma Raspberry Pi, sensor de presença e uma Raspberry Pi Camera que em conexão a internet possa enviar um email de notificação de movimentação juntamente com um arquivo de imagem. O sistema será capaz de transmitir duas imagens de boa qualidade, ou seja, o usuário deverá ser capaz de identificar as imagens com clareza.

IV. REQUISITOS

Buscando alcançar um sistema que proporcione transmissão de informações confiáveis, serão necessários no projeto, a priori, os seguintes requisitos:

- Raspberry Pi - A Raspberry Pi é um computador do tamanho de um cartão de crédito, capaz de desempenhar várias funcionalidades, como sistemas de vigilância, aplicações militares, etc;
- Sensor de presença PIR - Os dispositivos PIR dispõem de dois apetrechos sensíveis à luz infravermelha que são orientados para o ambiente. No mercado, esses sensores possuem variável sensibilidade. O efeito ocelar das lentes utilizadas no dispositivo, estabelecem diferentes características de alcance, raio e padrão de detecção.

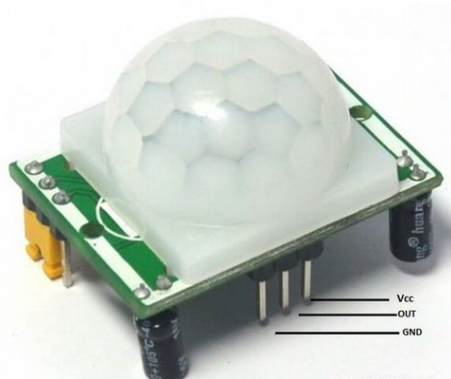


Figura 1 - Sensor de Movimento Presença PIR.

- Raspberry Pi Camera - câmera de 5MP de resolução de imagem ou gravação em 1080p que utiliza um plug de 15 pinos que se conecta diretamente na entrada CSI (Camera Serial Interface) da Raspberry.

V. BENEFÍCIOS

Desenvolver de forma caseira e de baixo custo um sistema de monitoramento residencial capaz de manter o usuário informado sobre a situação de sua residência de forma remota, tendo acesso a conexão de internet.

VI. IMPLEMENTAÇÃO

Como mostrado na Figura 2, sistema é iniciado com a detecção de atividade pelo sensor de presença que, por sua vez, aciona a captura de imagem pela câmera que já está posicionada no local desejado. Esta imagem é recebida pelo Raspberry Pi que enviará esse arquivo para um email previamente cadastrado, onde o usuário é notificado por dispositivos de uso pessoal com acesso às notificações de emails. O email recebido, além de conter a imagem do evento ocorrido, também contará com o dia e o horário em que o evento ocorreu.

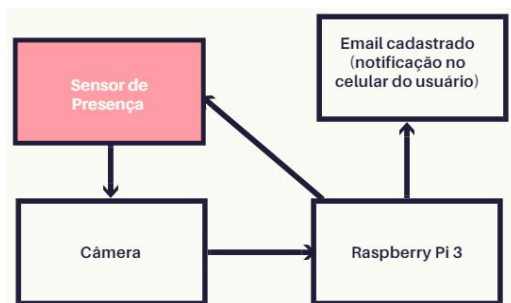


Figura 2 - Diagrama de funcionamento do sistema.

Os sensores PIR captam luz infravermelha e utilizam a variação dessa radiação no ambiente para identificar quando há movimentos. São construídos com materiais cristais

piroelétricos e contém inclusive uma lente de plástico. O funcionamento do sensor é ilustrado na Figura 3

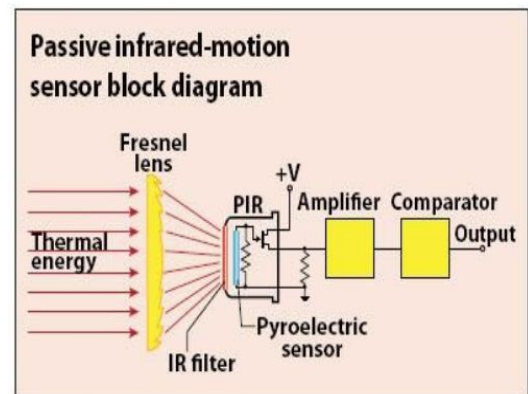


Figura 3 - Funcionamento do sensor de presença PIR

A ligação elétrica do sensor PIR é composta dos pinos VCC, GND e OUT. Ele são ligados ao 5V, GND e GPIO17 da Raspberry Pi, como o modelo da Raspberry é de 3,3V conecta diretamente ao Raspberry Pi.

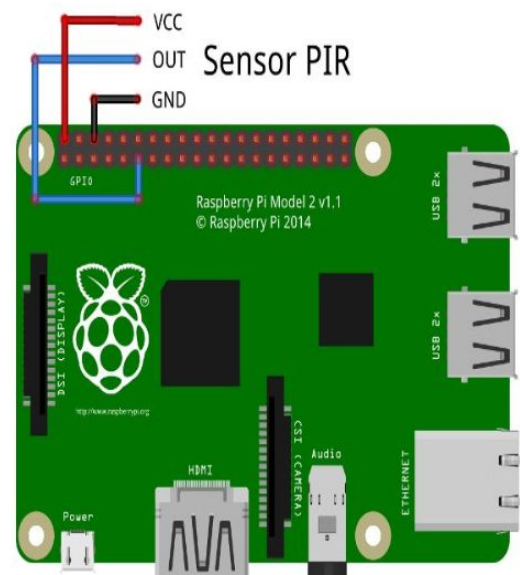


Figura 4 - Esquema de ligação dos terminais do Raspberry PI e o sensor PIR.

A seguir serão descritos os protótipos usados para a construção do sistema final utilizando linguagem python..

Para a captura da imagem foi utilizado o código presente na Figura 2, onde nas linhas 1 e 2 são importadas as bibliotecas utilizadas para utilizar a Raspberry Pi Camera e a função sleep, respectivamente. A função *camera.rotation* rotaciona a imagem da câmera em 180 graus devido a posição da mesma. A imagem é capturada 5 segundos após

o início da tela de *preview* ser iniciada e é salva em uma pasta no formato .jpg.

```
1 from picamera import PiCamera
2 from time import sleep
3
4 camera = PiCamera()
5
6 camera.rotation = 180
7
8 camera.start_preview()
9 sleep(5)
10 camera.capture('/home/pi/Monitoramento/imagem.jpg')
11 camera.stop_preview()
```

Figura 2 - Código para captura de imagem.

As figura 3 e 4 mostram o código utilizado para o envio da imagem por email. Primeiramente é importado bibliotecas como *MIMEMultipart*, *MIMEImage* e *MIMEText* que servem para adicionar anexos em um email que será enviado, já a biblioteca *smtplib* utiliza gratuitamente servidores portáteis SMTP também utilizados pelo Google Gmail e Apps. Em seguida é feito o cadastro dos emails de remetente e destinatário. Na Figura 4 a imagem é anexada ao email e é enviada. Por última uma mensagem de confirmação de email é mostrada na tela do terminal.

```
1 from email.mime.multipart import MIMEMultipart
2 from email.mime.image import MIMEImage
3 from email.mime.text import MIMEText
4 import smtplib
5
6 msg = MIMEMultipart()
7
8 senha = "
9 msg['Remetente'] = "projetodemonitoramento@gmail.com"
10 msg['Destino'] = "magalhaesrafael07@gmail.com"
11 msg['Assunto'] = "Sistema de Monitoramento"
```

Figura 3 - Envio de email, parte 1.

```
13 msg.attach(MIMEImage(file('/home/pi/Monitoramento/imagem.jpg').read()))
14
15 server = smtplib.SMTP('smtp.gmail.com: 587')
16 server.starttls()
17 server.login(msg['Remetente'], senha)
18 server.sendmail(msg['Remetente'], msg['Destino'], msg.as_string())
19 server.quit()
20
21 print "Email enviado com sucesso para %s:" % (msg['Destino'])
```

Figura 4 - Envio do email, parte 2.

Nas Figuras 5 e 6 são implementados um teste de controle dos pinos GPIO da Raspberry que acende um LED (GPIO04) quando há detecção de movimento pelo sensor PIR, que envia um sinal de alto nível para o pino GPIO17 (modo BCM).

```
1 import RPi.GPIO as GPIO
2 import time
3 GPIO.setwarnings(False)
4 GPIO.setmode(GPIO.BCM)
5 pir_pin = 17
6 GPIO.setup(pir_pin, GPIO.IN)
7 GPIO.setup(4, GPIO.OUT)
8
```

Figura 5 - Biblioteca e definição de GPIO.

```
9
10 try:
11     while True:
12         i = GPIO.input(pir_pin)
13
14         if i == 0:
15             print ("Sem movimento")
16             GPIO.output(4, 0)
17             time.sleep(0.1)
18         elif i == 1:
19             print ("Movimento detectado")
20             GPIO.output(4, 1)
21             time.sleep(0.1)
22
23 except KeyboardInterrupt:
24     print ("quit")
25     GPIO.cleanup()
26
```

Figura 6 - Teste de estado do sensor PIR com LED.

VII. RESULTADOS E DISCUSSÕES

Com o sistema montado e rodando o script presente no anexo, foi observado que o sinal chegando do sensor PIR era sempre 1, que indica que o sensor está percebendo movimento, mesmo quando fechado dentro de uma caixa imóvel. Para resolver o problema a sensibilidade do sensor PIR foi ajustada para o nível mais baixo assim como o tempo de resposta. Estes ajustes foram feitos por meio dos potenciômetros presentes no próprio sensor. Por fim, os tempos de delay foram ajustados no script (utilizando a função `time.sleep()`) para obter melhores resultados de detecção, captura das imagens e envio do e-mail de alerta no menor intervalo de tempo possível.

VIII. CONCLUSÃO

Foi possível concluir com o projeto de Monitoramento de Segurança por Presença que o sistema de informação sobre acontecimentos externos com captação de recursos visuais e notificação próxima do instante do ocorrido, conforta e assegura o usuário sobre seu patrimônio e segurança de sua família. Hoje em dia, as ferramentas de segurança permitem-nos ter um conforto e tranquilidade, bem como abastecer de informações e provas sobre

situações onde é necessário ter um acionamento das autoridades responsáveis com rapidez.

Portanto foi possível concluir com o projeto que o sistema estando bem calibrado conforme os requisitos do usuário, leva comodidade e praticidade ao cliente. Os possíveis ajustes que podem ser implementados, ajudam na especificação para utilidades mais específicas. Entretanto, o projeto de Sistema de Monitoramento de Segurança por Presença atinge com sucesso seu objetivo principal de captar presença e notificar com imagem do ocorrido ao usuário.

VII. REFERÊNCIAS

RASPBERRY FOUNDATION. RASPBERRY PI 2 MODEL B. Raspberry PI, 2015.

Sharma, Rupam Kumar, et al. "Android interface based GSM home security system." Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on. IEEE, 2014.

<https://www.filipeflop.com/blog/automacao-residencial-com-raspberry-pi-a-larme/> (acesso em 04/09/2018).

<https://uk.pi-supply.com/products/raspberry-pi-camera-board-v1-3-5mp-1080p> (acesso em 05/08/2018).

https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Basico_sobre_HTTP/MIME_types (acesso em 16/10/2018).

<https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/> (acesso em 16/10/2018).

<https://www.digitalocean.com/community/tutorials/how-to-use-google-s-smtplib-server> (acesso em 17/10/2018).

<https://code.tutsplus.com/pt/tutorials/sending-emails-in-python-with-smtp-cms-29975> (acesso em 17/10/2018).

<https://raspberrypi.stackexchange.com/questions/12966/what-is-the-difference-between-board-and-bcm-for-gpio-pin-numbering> (acesso em 14/11/2018).

<https://www.raspberrypiportugal.pt/sensor-movimento-infravermelho-hc-sr501/> (acesso em 22/11/2018)

ANEXO

```
#Carrega as bibliotecas para envio de e-mail
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email.mime.image import MIMEImage
from email import encoders

# Carrega as bibliotecas da camera, GPIO e tempo
from picamera import PiCamera
import RPi.GPIO as GPIO
import time
from time import gmtime, strftime

#Define a GPIO do sensor PIR
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(11, GPIO.IN)

#Prepara a camera
camera = PiCamera()
camera.rotation = 180

#Definicao do email
i=0
estado = 0

while True:
    i=0
    estado = GPIO.input(11) #valor binário é registrado pelo sensor e
    guardado na variavel estado

    if estado == 0:
        time.sleep(0.1)

    elif estado == 1:
        #Captura das fotos
        print ("Movimento detectado");
        for i in range(2):
            sleep(2)
            camera.capture('/home/pi/Monitoramento/fotos/imagem%s.jpg' % i)
            print ("A próxima leitura será feita em 20 segundos\n");
            time.sleep(20)

        #Envia o email
        fromaddr = "projetodemonitoramento@gmail.com"
        toaddr = "magalhaesrafael07@gmail.com"

        msg = MIMEMultipart()

        msg['From'] = fromaddr
        msg['To'] = toaddr
        msg['Subject'] = "Registro de movimento"

        diaehora = strftime ("%d-%m-%Y %H:%M:%S", gmtime())
```

```
mensagem = "Movimento detectado em:\n" + diaehora

msg.attach(MIMEText(mensagem, 'plain'))

filename = "imagem0.jpg"
attachment = open("/home/pi/Monitoramento/fotos/imagem0.jpg",
"rb")

part = MIMEBase('application', 'octet-stream')
part.set_payload((attachment).read())
encoders.encode_base64(part)
part.add_header('Content-Disposition', "attachment; filename= %s" %
filename)
msg.attach(part)

filename = "imagem1.jpg"
attachment = open("/home/pi/Monitoramento/fotos/imagem1.jpg",
"rb")

part = MIMEBase('application', 'octet-stream')
part.set_payload((attachment).read())
encoders.encode_base64(part)
part.add_header('Content-Disposition', "attachment; filename= %s" %
filename)
msg.attach(part)

server = smtplib.SMTP('smtp.gmail.com', 587)
server.starttls()
server.login(fromaddr, "gama2018")
text = msg.as_string()
server.sendmail(fromaddr, toaddr, text)
server.quit()
```