Sistema de segurança doméstica

Um sistema baseado em raspberry PI capaz de monitorar a distância residências, comunicando se aos usuários através do serviço de e-mail.

Paulo Henrique Bernardo Melo
Universidade de Brasília - UnB
Brasília-DF, Brasil
paulohb.melo@gmail.com

Pedro Raguzzoni
Universidade de Brasília - UnB
Brasília-DF, Brasil
pedro_raguzzoni@hotmail.com

***Resumo* — Esse documento expõe uma proposta detalhada de projeto de um sistema de vigilância doméstico que permite detectar a presença de invasores, podendo ser monitorado de qualquer lugar através do serviço de e-mail. O sistema consiste em um sensor de movimento (PIR) que detecta a presença de um potencial invasor e fotografa sua imagem, através de uma câmera digital, e rapidamente envia a imagem captada para o e-mail do usuário para alertá-lo da potencial invasão.**

Palavras-chaves - Sistema de segurança doméstico, Raspbarry Pi, sensor de movimento, câmera fotográfica e e-mail.

1. INTRODUÇÃO

O número de crimes contra o patrimônio, que incluem roubos e furtos, em residências no Distrito Federal saltaram de 1,7 mil para 2,1 mil para cada grupo de 100 mil habitantes entre 2015 e 2016. Esses dados foram anunciados pela secretária de Segurança Pública Márcia de Alencar, na apresentação do balanço da criminalidade em 2016 [1].

O crescente número de vítimas de crimes contra o patrimônio e a dificuldade de recuperação dos bens perdidos devido à demora na efetivação de procedimento de ocorrência policial, proporciona um grande desconforto e uma crescente preocupação aos moradores de áreas afetadas por este tipo de crime

A ausência de um sistema de segurança residencial pode acarretar um tempo de espera muito grande até que o crime seja percebido pela vítima, dificultando ainda mais a ação das autoridades competentes, devido à demora na efetivação de procedimento de ocorrência policial [1].

Com isso, a implementação de um sistema de segurança que ofereça os benefícios de uma resposta rápida para a vítima poderia ajudar na rápida identificação do criminoso ou mesmo de apoio à ação policial, oferecendo mecanismo adicional na proteção do patrimônio residencial.

Com isso em mente, a solução proposta de se desenvolver um sistema de monitoramento irá reduzir a insegurança proporcionada pelo aumento da criminalidade, uma vez que o tempo entre a invasão domiciliar e a ciência do morador sobre o problema será reduzido, bem auxiliará as autoridades na identificação do infrator [2].

2. OBJETIVOS

Desenvolver um sistema de monitoramento, para ser implementado em residências, a fim de auxiliar o cliente a monitorar sua residência e diminuir o tempo de resposta a uma eventual invasão do ambiente interno, bem como auxiliar as autoridades na identificação do infrator.

2.1. OBJETIVOS ESPECÍFICOS

- Diminuir o tempo de ciência do usuário em caso de uma violação no ambiente residencial monitorado.
- Proceder à operacionalização de uma placa Raspberry Pi com o Sensor de Presença (PIR) e um câmera fotográfica (PI camera).
- Desenvolver um código em C, para o módulo de monitoramento, capaz de detectar quando o invasor entra no alcance do sensor de presença (PIR). A raspberry PI enviará um comando a PI camera que tirará uma fotografia e irá salvá-la
- Promover a comunicação entre o protótipo e o usuário via e-mail. A Raspberry Pi criara um e-mail e enviará para o endereço de e-mail salvo com a foto recente tirada do suspeito.

3. REQUISITOS

3.1. Requisitos funcionais:

- O sistema deverá encaminhar em poucos segundos, a fotografia para o e-mail do cliente.
- O sistema deverá ter uma câmera com resolução de no mínimo 5MP.
- O sistema deverá ser apto para tirar fotos em ambientes com pouca luz.
- O sistema contará com a presença de um sensor de presença PIR.
- O sistema deverá ser capaz de emitir um alarme sonoro quando disparado para constranger e intimidar o provável invasor.
- O sistema deverá ser ativado por um botão e desativado por uma senha, inserida pelo usuário através de um teclado numérico.

3.2. Requisitos não-funcionais:

- O sistema deverá estar conectado à uma rede de Internet.
- Raspberry Pi 3 com módulo wi-fi e sistema operacional Raspbian.
- Alimentação para a Raspberry Pi 3.

4. DESENVOLVIMENTO

O desenvolvimento deste trabalho descreve detalhadamente o processo de desenvolvimento do sistema de segurança utilizando a RaspberryPi 3.

O diagrama geral de blocos do projeto está descrito na figura 1. O mesmo foi feito levando em consideração os requisitos funcionais citados no item 3 do presente relatório. Os principais componentes do diagrama abaixo são tratados a seguir:

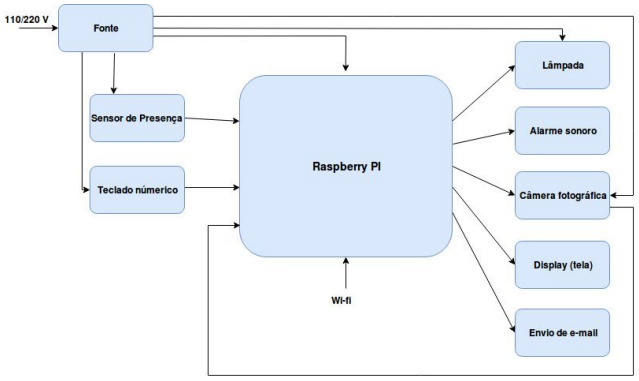

Figura 1 - Diagrama de blocos do projeto

O projeto é dotado de componentes de hardware e software que se interagem entre si. O hardware do projeto é responsável pela aquisição/exibição de dados do sistema, enquanto o software é responsável pelo gerenciamento dos dados de entrada e saída.

A operação do sistema foi pensada da seguinte forma:

1) O usuário aciona o sistema de segurança através de um botão ao sair de sua residência;
2) O sistema então passa a esperar a detecção de algum movimento no alcance de um sensor de presença utilizado.
3) Quando o sensor detectar algum movimento o sistema enviará um comando para que uma lâmpada se acenda iluminando o ambiente.
4) Então, o sistema enviará um outro comando para que uma câmera, estrategicamente colocada, tire uma foto do local e a envie para o usuário por e-mail, o alertando que sua residência foi invadida.
5) Em paralelo ao comando de tirar a foto, o sistema acionará um alarme sonoro para intimidar e constranger o possível invasor.
6) O sistema será desligado apenas quando o usuário digitar em um teclado número uma senha correta.

4.1. DESCRIÇÃO DO HARDWARE

Aqui será abordada a parte de desenvolvimento de hardware, explicando com o máximo de detalhes o funcionamento e conexões dos sensores, câmera e GPIO da RasperryPi 3.

Para a realização do projeto foram utilizados os seguintes componentes de hardware:

- 1 Raspberry pi 3 modelo b;
- 1 fonte de energia para a Raspberry Pi;
- 1 teclado número;
- 1 display (tela);
- 1 módulo câmera Rasberry Pi;
- 1 Sensor de movimento;
- 1 Buzzer (alarme sonoro);
- 1 lâmpada com bocal;
- 1 Relé comum de 5V;
- 1 Push Botom;
- 1 Transistor;
- 1 Diodo;
- 2 Resistores;
- Conectores elétricos;
- 1 Matriz de contatos (protoboard);

Abaixo descrevemos de maneira sucinta a utilidade de alguns dos componentes utilizados.

Sensor de movimento

Neste Projeto utilizamos o sensor infravermelho de movimento PIR (HC-SR501). O PIR HC-SR501 é um sensor de presença construído basicamente por um sensor piroelétrico, que é capaz de detectar níveis de radiação infravermelha. Ele é dotado de alta sensibilidade e confiabilidade e pode ser usado com placas como a Raspberry PI, bem como diretamente sem o auxílio de micro controladores. Sua tensão de alimentação pode estar entre 4.5V e 20V. Sua saída digital opera em 3.3V, podendo ser ligado tanto a placas de 5V como de 3.3V.

Além dos três pinos (GND, VCC e OUT) temos dois potenciômetros de ajuste, um para sensibilidade e outro para temporização. A figura 1 mostra os pinos do sensor de presença PIR.

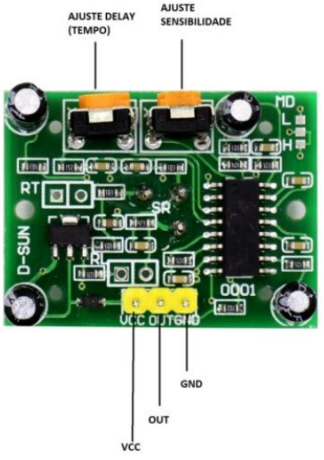

Figura 2 - Sensor de movimento PIR HC-SR501

A saída do sensor PIR vai para nível lógico alto (3.3V) após a detecção de algum movimento, e permanece em nível lógico alto por aproximadamente 3s. O tempo que

este permanece em nível lógico alto pode ser alterado utilizando o ajuste de *delay* mostrado na figura 1.

Câmera PI

O módulo Câmera Raspberry Pi utilizado neste Projeto utiliza os mesmo sensor e hardware utilizado nos smartphones, é leve (apenas 3g) e compacto (25 x 20 x 9 mm) e gera fotos com resolução de até 2592 x 1944 pixels. Uma vantagem deste módulo é que ele utiliza o próprio conector presente no Raspberry Pi, não ocupando nenhuma porta USB da placa.

Figura 3 - Módulo câmera Raspberry PI

Relé comum de 5V

O projeto fez uso de um relé comum de 5V. Os relés funcioam como interruptores , mas que são acionados por uma baixa tensão. Os relés comuns possuem um contato interno e uma bombina. Quando há corrente passando pela bobina, um campo magnético é induzido, atraindo um pino interno e fechando o contato.

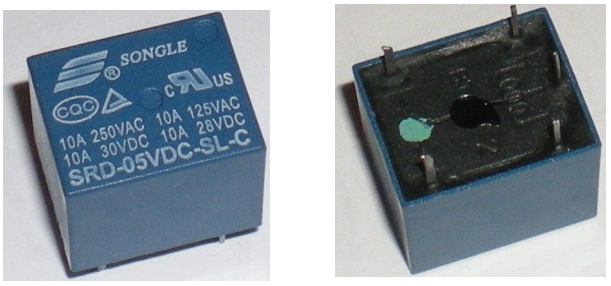

Figura 4 *– Relé comum*

Buzzer

Para emitir o alarme sonoro para a intimidação do invasor o projeto fez uso de um buzzer. Este componente é utilizado para a geração de ruídos sonoros a partir da excitação elétrica de componentes eletromecânicos ou piezoelétricos.

Figura 5 – Buzzer

4.2. DESENVOLVIMENTO DE HARDWARE

Nesta etapa da construção do Sistema de **Segurança Doméstico Baseado em Raspberry Pi 3**, conectou-se o módulo da Câmera PI ao slot da Raspberry PI, e o Sensor de Movimento (PIR) aos pinos GPIO da placa, sendo que o sensor PIR foi alimentado com 5V utilizando os pinos de tensão de 5V das Raspberry e o GND da Raspberry também. Já o pinos de saída do sensor foi conectado ao pino 7 das Raspberry.

Ademais foram realizadas as ligações para tratar os requisitos funcionais de deixar o sistema apto para tirar fotos em ambientes com pouca luz (sistema de iluminação) e capaz de emitir um sinal de alerta para constranger e intimidar o invasor.

Para a realização do sistema de iluminação foram utilizados os componentes: relé, transistor, diodo e de uma lâmpada, que se acenderá automaticamente antes do sistema tira a foto, com a seguinte configuração na protoboard.

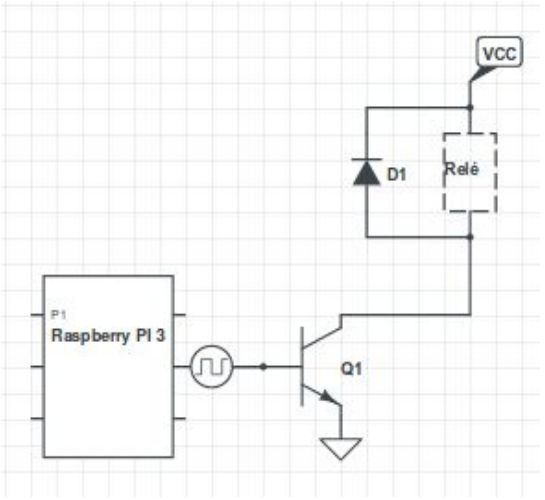

Figura 6 – Esquemático para a ligação do sistema de iluminação

Para a realização do sistema de alarme foi contecto o buzzer a uma pino GPIO da raspberry Pi que é settado em nível lógico auto quando o sensor de presença PIR é disparado.

Configuração da câmera PI

O primeiro passo para este projeto foi a configuração da placa Raspberry Pi 3, com a instalação do Sistema Operacional Raspbian.

Após procedemos com a instalação dos arquivos da biblioteca para a utilização da do modulo Pi Câmera que utilizamos neste Projeto. Seguindo os comandos abaixo:

$ sudo apt-get install python-picamera
$ sudo apt-get installphython3-picamera

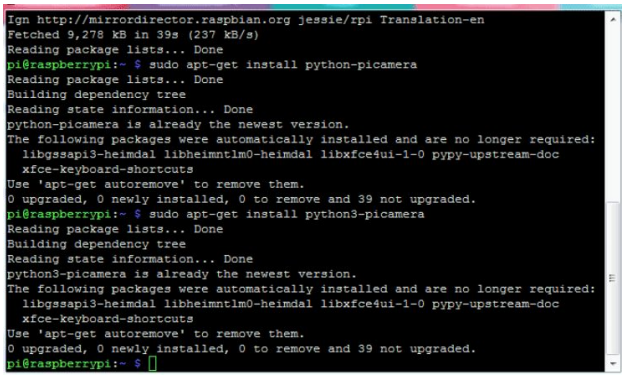

Figura 7 – resultado da instalação das bibliotecas para a Pi câmera

O resultado da instalação deverá ser algo semelhante ao apresentado na figura 6 mostra a cima.

Prosseguimos habilitando a Raspberry PI camera usando a ferramenta de configuração do software da Raspberry PI.

$ sudo raspi-config

Assim, selecionamos Enable Camera para habilitá-la. Como mostrado na figura 4.

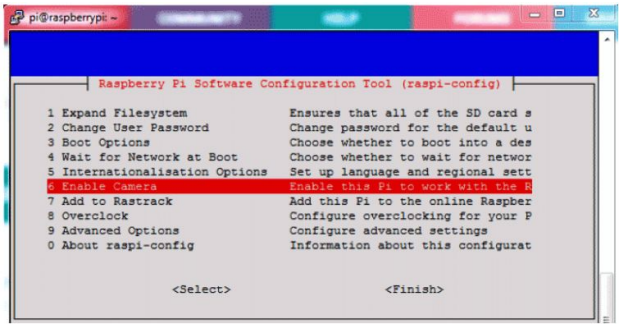

Figura 8 – Selecionar opção de enable da câmera

O passo seguinte após a configuração da Pi Camera, é a instalação do software para o envio de e-mails. Neste Projeto utilizou-se o protocolo SSMTP o qual é boa solução para o envio usando comandos de linha para o shell que pode ser facilmente utilizado em linguagem C utilizando a chamada system(). Para a utilização do protocolo de e-mail SSMPT é necessário executar as linhas de comando abaixo para que instale o protocolo SSMTP na RaspberryPi.

$ sudo apt-get install ssmtp
$ sudo apt-get install mailutils

Após a instalação desses programas é necessário configurar o arquivo "ssmtp.conf". Este arquivo pode ser acessado utilizando o seguinte comando no shell:

$ sudo nano /etc/ssmtp/ssmtp.conf

Após o acessar o arquivo é necessário reescrevê-lo de modo a ficar com o seguinte texto:

```
root=EndereçoDeEmail

mailhub=smtp.gmail.com:587

hostname=raspberrypi

AuthUser= EndereçoDeEmail

AuthPass=SenhaEmail

FromLineOverride=YES

UseSTARTTLS=YES

UseTLS=YES
```

Para enviar um email de teste para verificar o devido funcionamento do protocolo SSMTP utilizou-se a seguinte linha de comando, que enviar uma mensagem de texto para um destinatário com algum assunto:

$ echo "Corpo do texto" | mail -s "Assunto do Texto" email_destinatario@gmail.com

4.3. DESCRIÇÃO DE SOFTWARE

A implementação de software deste projeto consiste em chamadas system() para tirar a foto com a Câmera PI e outra chamada system() para enviar o e-mail para o destinatário. O controle das chamada system() é feito através do pino 7, em que está conectado a saída do sensor PIR. As chamadas para tirar a foto e enviar para o email só são feitas quando o pino 7 que estará configurado como pino de entrada ler uma tensão de nível lógico alto. Após a leitura o sistema irá capturar uma foto e enviá-la ao email cadastrado imediatamente. O tempo de envio de email dependerá da velocidade da internet em que a Raspberry encontra-se conectada.
O código final está em anexo com esse arquivo.

Além disso para inicializar o sistema foi inserido um botão ao pino 9 que ao ser pressionado liga o sistema através da chamada de um novo processo, ou seja, ao pressionar o botão o processo que será chamado executará o algoritmo descrito no parágrafo anterior. Para desligar o sistema é necessário que o usuário digite corretamente a senha que está cadastrada, ao digitar corretamente a senha é enviado um sinal SIGINT ao processo responsável por fazer a leitura do sensor, tirar a foto e enviar o email. Este sinal irá matar esse processo fazendo com que o sistema volte a ficar adormecido, sendo necessário pressionar novamente o botão para ativar o sistema novamente.

A figura abaixo mostra um diagrama que explica o funcionamento do software. Nesse diagrama o bloco *Start* é o processo principal que está em *loop* infinito apenas esperando o sinal SIGUSR vindo do botão conectado ao pino 11. Após o processo *Start* receber este sinal ele cria outros dois processos, o *Exec* e o Senha, e aguardo que os processos filhos terminem para que ele volte para o *loop* infinito novamente.

O processo *Exec* executa a função descrita no primeiro parágrafo deste tópico, em que é responsável por ler o sensor PIR, tirar a foto e enviar o email. Esse processo encontra-se em *loop* infinito também, e só será finalizado quando o processo Senha enviar um sinal SIGINT para matá-lo.

O processo Senha está aguardando que o usuário digite a senha cadastrada corretamente para enviar um sinal SIGINT para o processo *Exec*. Quando a senha válida for digitada o sinal SIGINT é enviado ao processo *Exec* matando-o. Caso seja digitada uma senha inválida o processo Senha requisita que o usuário insira uma nova senha e não é enviado nenhum sinal ao processo *Exec.*

Após a finalização do processo *Exec* o processo pai *Start* volta a ser executado e fica aguardando novamente o sinal do botão para executar novamente o que foi descrito acima.

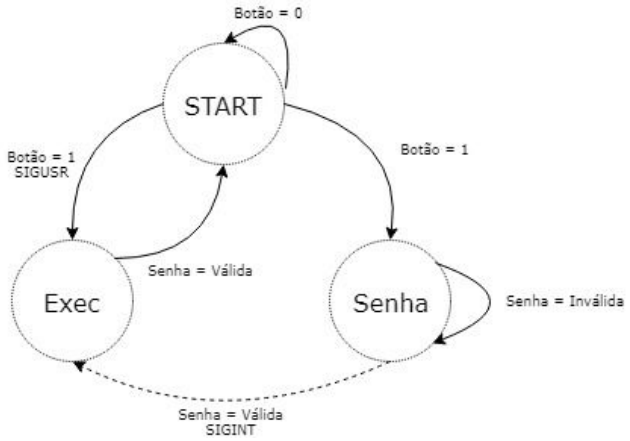

Figura 9 – Diagrama de estados do Software.

5. RESULTADOS

O Sistema de Segurança Doméstico Baseado em Raspberry Pi 3 proposto neste projeto é capaz de detectar a presença de um possível invasor através do Sensor de Presença, bem como é capaz de enviar ao usuário do sistema um e-mail avisando que um possível invasor está onde o sistema de vigilância foi instalado, anexando a este e-mail uma mensagem e uma fotografia.

O sistema ainda é capaz de tirar a fotografia em ambiente de pouca luz porque conta com um sistema de iluminação e além disso emite um alarme sonoro para constranger e intimidar o possível invasor, quando o sensor de presença é acionado de forma automatizada.

Para o acionar e desligar o sistema, o projeto conta com um botão e um teclado para a digitação da senha para sua desativação.

6. CONCLUSÃO

Como é possível observar foram atingidos os principais objetivos deste projeto, contudo o sistema ainda não está apto para ser comercializado por está em uma fase de protótipo.

Esta fase é caracterizada pelo sistema carecer ainda da implementação do teclado, da tela, e da interface que permita ao usuário controlar o sistema sem abrir o terminal da Raspberry com o uso de um teclado e um monitor convencional.

Desta forma, o projeto de segurança doméstica proposto necessita de aprimoramento para ser comercial.

7. Referências

[1] GUSTAVO AGUIAR, G1 GLOBO. Site de notícias do globo, 13 de Janeiro de 2017. ISSN <https://g1.globo.com/distrito-federal/noticia/roubos-cresc em-no-df-secretaria-de-seguranca-diz-que-cidade-e-segura. ghtml>. Acesso em 03 de Abril de 2018.

[2] COUTINHO, M. P. Sistema de Monitoramento Residencial. 2016. Dissertação (Graduação em Engenharia da Computação) – Centro Universtitário de Brasília – UNICEUB. Brasília.

ANEXO

Abaixo está descrito a integralidade do código do projeto:

```
#include<stdio.h>
#include<stdlib.h>
#include<wiringPi.h>
#include<unistd.h>
#include<string.h>
#include<signal.h>
#include <sys/types.h>
#include <sys/wait.h>
void func_liga();
void func_senha();
void func_desliga();
void func_btn_liga();

int main(int argc,char *argv[])
{
   int pid_pai, pid_btn_liga;

   wiringPiSetup(); // iniciando biblioteca wiringPi
   pinMode(7, INPUT); // saída do PIR, pino físico 7
   pinMode(0, INPUT); // botão de desligar, pino físico 11
   pullUpDnControl (0, PUD_UP);
         pinMode(2, OUTPUT); // pino para acender a lampada, pino físico 13
         pinMode(12, OUTPUT); // pino para acionar o alarme, pino físico 19
   digitalWrite(12, LOW);

   pid_pai = getpid(); // determinando o PID do processo pai
   pid_btn_liga = fork(); //função que lerá o botão de ligar o sistema

   if(pid_btn_liga == 0) // processo que lerá o botão
   {
         func_btn_liga(pid_pai);
   }


   signal(SIGUSR2, func_liga);

   sleep(1);

   while(1)
   {
         sleep(10);
   }


   return 0;
}

void func_btn_liga(int pid)
{
   printf("Pressione o botão para iniciar o sistema!\n");

   while(1)
   {
         if(digitalRead(0) == LOW)
         {
               kill(pid, SIGUSR2);
               sleep(2);
         }
   }
}

void func_liga()
{
   int pid_liga, pid_senha;

   pid_liga = fork();

   if (pid_liga == 0)
   {
         pid_liga = getpid();
         printf("Sistema iniciado! Digite a senha cadastrada para desativa-lo\n");

         pid_senha = fork(); // criando um processo filho para tratar senha

         if(pid_senha == 0) // processo para tratar senha
         {
               func_senha(pid_liga);
         }


         while(1)
         {
               if(digitalRead(7) == HIGH) // leitura do sensor PIR
               {
               printf("acendendo a lampada\n");
               digitalWrite(2,HIGH);
               //digitalWrite(12, HIGH);
               printf("tirando foto:\n");
               system("raspistill -o im1.jpg");
//tirar a foto
               digitalWrite(2,LOW);
               digitalWrite(12,LOW);
               usleep(1);
               printf("enviando email:\n ");
               system("mpack -s " \"Imagem camera rasp\" /home/pi/Desktop/embarcados/im1.jpg raguzzoni.pedro@gmail.com");
```

```
                printf("email enviado\n");
                sleep(3);
                }
        }

    }
    wait(&pid_liga);


}

void func_senha(int pid)
{

while(1){
   int i, len, len2;
   char *chave = "1234";
   char senha[100];

   scanf("%s", senha);

   len = strlen(chave);
   len2 = strlen(senha);

   if(len2 == len)
   {
        for(i = 0; i < len; i++)
        {
                if(chave[i] == senha[i])
                {

                        if(i == (len-1))
                        {
                         printf("Senha correta!\n");
                         printf("Sistema desativado!
Press\n");
                         kill(pid, SIGINT);
                         exit(1);
                        }
                }else
                {
                printf("Senha invalida!\n");
                        break;
                }

        }

   }else printf("Senha invalida!\n");

   }
}
```