

Projeto Final Embarcados

Sistema de Monitoramento Inteligente

Derick Horrana

Matrícula: 10/0009891

Universidade de Brasília, Campus Gama.

E-mail: derickhorrana12@gmail.com

Brasília, Brasil

Eduardo Henrique

Matrícula: 11/0148011

Universidade de Brasília, Campus Gama.

E-mail: eduardoons@gmail.com

Brasília, Brasil

Abstract— This work aims to implement and develop an intelligent residential monitoring system, using Raspberry Pi. For example, when motion is detected, the camera automatically initiate recording and the Raspberry Pi device alerts the owner of the possible intrusion having a smart phone.

Keywords— Raspberry Pi; monitoring; system; email

I. INTRODUÇÃO

Este trabalho tem como objetivo apresentar um sistema de monitoramento residencial que busca minimizar o tempo entre invasão/roubo e o conhecimento do usuário/proprietário. Para o desenvolvimento do projeto, serão empregados uma Raspberry Pi modelo B, o modulo oficial da câmera Raspberry Pi, um sensor de presença PIR e um adaptador Wireless. Os componentes serão configurados e testados de modo que, ao ser percebida movimentação no ambiente monitorado, o sistema seja capaz de avisar o usuário com o envio de e-mail com imagem em anexo mostrando o local onde o movimento foi percebido pelo sensor PIR.

II. JUSTIFICATIVA

A ausência de um sistema de segurança, nas residências em geral, pode acarretar um tempo de espera muito grande até que o crime seja percebido pela vítima, trazendo ainda mais prejuízos e atrasando a intervenção das autoridades competentes. Dessa forma, a implementação de um sistema de segurança que proporcione os benefícios de uma resposta rápida para a vítima, pode ajudar no reconhecimento do criminoso e servir como material para a ação policial, oferecendo mecanismo adicional na proteção do patrimônio. Se a pessoa estiver em casa ou fora da cidade, a ideia por trás de um sistema de segurança é que ele impede intrusões ao notificar potenciais criminosos e que um sistema de alarme está em uso alertando o usuário ou a empresa de segurança quando o sistema de segurança é violado.

A. Objetivos

Desenvolver um hardware, composto por plataforma Raspberry Pi, com sensor de presença, e uma mini câmera utilizando a tecnologia de infravermelho em conjunto com um adaptador Wireless, para se conectar à rede e transmitir a informação e, que é o e-mail de notificação de movimentação.

Os dados do streaming multimídia serão transmitidos pelo Raspberry Pi e estarão acessíveis ao proprietário, permitindo assim a visualização do local monitorado.

O sistema será capaz de:

- Transmitir um streaming de vídeo de boa qualidade, ou seja, o usuário deverá ser capaz de identificar as imagens com clareza e precisão;
- Notificar o usuário, por meio de um e-mail, caso haja alguma movimentação no ambiente monitorado;

III. HARDWARE E SOFTWARE

A. HARDWARE

A primeira coisa que é utilizada neste projeto é um Raspberry Pi 3 B, pois é a versão mais recente disponível até a data do início do projeto. Ela também tem recursos bem agradáveis como [1]:

- Uma CPU ARMv8 qual-core de 64 bits de 1.2 GHz
- LAN sem fio 802.11n
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)
- 1GB de RAM
- 4 portas USB
- 40 pinos GPIO
- Porta HDMI completa
- Conexão de Rede
- Tomada de áudio combinada de 3,5 mm e vídeo composto
- Interface da câmera (CSI)
- Interface de exibição (DSI)
- Ranhura do cartão Micro SD (agora empurrar-puxar em vez de push-push)

-
- GPIO Pinout Diagram**
- | Pin | Function | Pin | Function | Pin | Function | Pin | Function | Pin | Function |
|-----|----------|-----|----------|-----|----------|-----|----------|-----|----------|
| 1 | 5V | 11 | Ground | 21 | Ground | 31 | Ground | | |
| 2 | 5V | 12 | GPIO15 | 22 | GPIO23 | 32 | GPIO33 | | |
| 3 | 5V | 13 | GPIO16 | 23 | GPIO24 | 33 | GPIO34 | | |
| 4 | 5V | 14 | GPIO17 | 24 | GPIO25 | 34 | GPIO35 | | |
| 5 | 5V | 15 | GPIO18 | 25 | GPIO26 | 35 | GPIO36 | | |
| 6 | 5V | 16 | GPIO19 | 26 | GPIO27 | 36 | GPIO37 | | |
| 7 | 5V | 17 | GPIO20 | 27 | GPIO28 | 37 | GPIO38 | | |
| 8 | 5V | 18 | GPIO21 | 28 | GPIO29 | 38 | GPIO39 | | |
| 9 | 5V | 19 | GPIO22 | 29 | GPIO30 | 39 | GPIO40 | | |
| 10 | 5V | 20 | GPIO23 | 30 | GPIO31 | 40 | Ground | | |
| 11 | Ground | 21 | Ground | 31 | Ground | | | | |
| 12 | GPIO15 | 22 | GPIO23 | 32 | GPIO33 | | | | |
| 13 | GPIO16 | 23 | GPIO24 | 33 | GPIO34 | | | | |
| 14 | GPIO17 | 24 | GPIO25 | 34 | GPIO35 | | | | |
| 15 | GPIO18 | 25 | GPIO26 | 35 | GPIO36 | | | | |
| 16 | GPIO19 | 26 | GPIO27 | 36 | GPIO37 | | | | |
| 17 | GPIO20 | 27 | GPIO28 | 37 | GPIO38 | | | | |
| 18 | GPIO21 | 28 | GPIO29 | 38 | GPIO39 | | | | |
| 19 | GPIO22 | 29 | GPIO30 | 39 | GPIO40 | | | | |
| 20 | GPIO23 | 30 | GPIO31 | 40 | Ground | | | | |
- 4 Squarely Placed Mounting Holes
- 40 GPIO Headers
- SMSC LAN9514 USB Ethernet Controller
- Run Header Used to Reset the PI
- Broadcom BCM2835
- MicroSD Card Slot (Underneath)
- DSI Display Connector
- Switching Regulator for Less Power Consumption
- 5V Micro USB Power
- HDMI Out Port
- CSI Camera
- 3.5mm Audio and Composite Output Jack
- 2x USB-A Ports to PC
- Ethernet Out Port

Outro componente que é utilizado é o módulo oficial da câmera *Raspberry Pi* para capturar imagens. O módulo de câmera *Raspberry Pi* v2 tem um sensor Sony *IMX219* de 8 *megapixels*. O módulo da câmera pode ser usado para capturar vídeo de alta definição, bem como fotografias. Não é muito difícil de ser utilizado e tem muito para oferecer aos usuários.

Esse módulo da câmera é bastante utilizado em aplicações de segurança doméstica, e em armadilhas de câmera de vida selvagem.



Também é usado um sensor de presença PIR. O Sensor de Movimento PIR DYP-ME003 consegue detectar o movimento de objetos que estejam em uma área de até 7 metros. Caso algo ou alguém se movimentar nesta área o pino de alarme é ativo. É possível ajustar a duração do tempo de espera para estabilização do PIR através do potenciômetro amarelo em baixo do sensor bem como sua sensibilidade. A estabilização pode variar entre 5-200 seg [3].



Uma vez que o *Raspberry Pi* será acessado remotamente, é necessário um dongle *USB WiFi* simples.



Figura 4- USB WiFi.

Serão utilizados também alguns fios jumper. No final, serão feitas as conexões entre a Raspberry Pi, a câmera, o sensor e o USB Wifi.

É importante e necessário já ter uma distribuição Linux instalada na Raspberry Pi. Isso é para garantir que se tenha uma Raspberry Pi completamente funcional. Neste projeto será utilizada a Raspbian.

Feito isso, conecta-se o Raspberry Pi à sua rede Wifi local e instala-se um driver para o chip BCM2835 para ler os dados do sensor PIR.

É importante baixar e instalar esses drivers seguindo as instruções nesta página [4].

Será necessário instalar drivers para o chip BCM2835 [5].

Para acessar o Raspberry Pi em sua rede Wifi local via Raspberry Pi será necessário instalar alguns pacotes.

IV.DESCRICÃO FUNCIONAL

- A. A câmera USB: captura a imagem e a envia para a porta USB da placa Raspberry Pi. O modelo ainda não foi decidido.
- B. 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.

- C. Sensor de presença: 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.
- D. Adaptador Wireless: O adaptador wifi utiliza chipset Realtek 5370 e trabalha na frequência de 2.4 GHz, podendo se conectar à redes wireless padrões 802.11b, g e n, alcançando taxas de transmissão de 150 Mbps. Os padrões de segurança aceitos são o WEP, WPA, WPA2, WPA-PSK e WPA2-PSK (TKIP e AES).

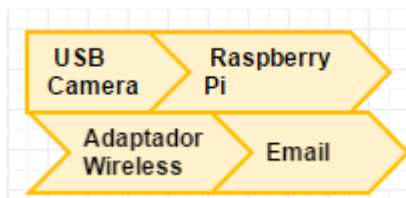


Figura 5-Eschema de Funcionamento.

V.ETAPAS DO DESENVOLVIMENTO

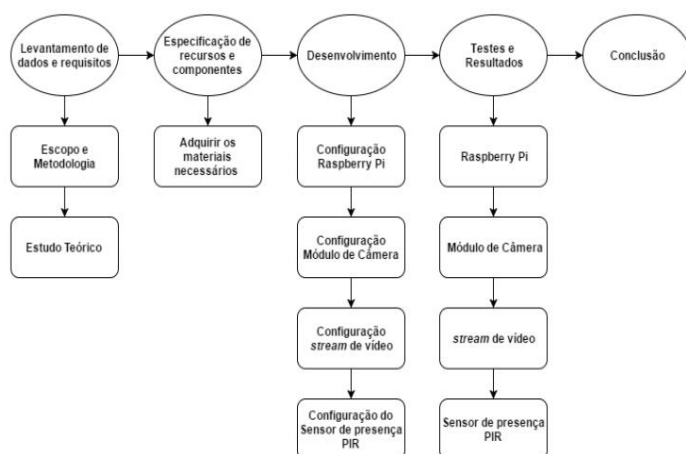


Figura 6-Diagrama de Construção.



Figura 7-Circuito Completo.

- A. A Figura 6 apresenta os ciclos do desenvolvimento do projeto. A primeira etapa é aquisição de dados e requisitos, determinando o escopo e a estrutura além da base teórica a serem utilizados ao longo do desenvolvimento do projeto de sistema de monitoramento residencial.
- B. A segunda parte é descrever recursos e componentes. Aborda as ferramentas e materiais adquiridos para a elaboração do sistema.
- C. Já a terceira parte é subdividida em quatro fases, sendo elas referentes à disposição dos materiais previamente levantados..
- D. A quarta parte é a mais importante:aborda o teste de funcionalidade do sistema como um todo e de cada um de seus módulos. É fundamental a utilização da ordem lógica de configuração e testes dos elementos expostos nas etapas 3 e 4, uma vez que são dependentes um do outro em sua codificação. Por fim, a quinta etapa, conclusão, remete ao bom funcionamento do projeto de monitoramento residencial.

VI.SIMULAÇÕES E RESULTADOS

A. CÓDIGO PARA DETECTAR MOVIMENTO(SENSOR PIR)

```

#include <wiringPi.h>
#include <stdio.h>
#include <sys/types.h>
#include <signal.h>
#include <pthread.h>

void move_now(void){

fprintf(stderr,"acionar agora\n");

}

int main(){

wiringPiSetup();

pinMode(0,INPUT);
wiringPiISR(0,INT_EDGE_RISING, move_now);

pthread_exit(NULL);
return 0;

}
  
```

B. Código do Sensor de Umidade

```

#include <wiringPi.h>

#include <stdio.h>
#include <stdlib.h>
  
```

```

#include <stdint.h>
#define MAXTIMINGS 85
#define DHTPIN 7
int dht11_dat[5] = { 0, 0, 0, 0, 0 };

void read_dht11_dat()
{
    uint8_t laststate = HIGH;
    uint8_t counter = 0;
    uint8_t j = 0, i;
    float f; /* fahrenheit */

    dht11_dat[0] = dht11_dat[1] = dht11_dat[2] =
dht11_dat[3] = dht11_dat[4] = 0;

    /* pull pin down for 18 milliseconds */
    pinMode( DHTPIN, OUTPUT );
    digitalWrite( DHTPIN, LOW );
    delay( 18 );
    /* then pull it up for 40 microseconds */
    digitalWrite( DHTPIN, HIGH );
    delayMicroseconds( 40 );
    /* prepare to read the pin */
    pinMode( DHTPIN, INPUT );

    /* detect change and read data */
    for ( i = 0; i < MAXTIMINGS; i++ )
    {
        counter = 0;
        while ( digitalRead( DHTPIN ) == laststate )
        {
            counter++;
            delayMicroseconds( 1 );
            if ( counter == 255 )
            {
                break;
            }
        }
        laststate = digitalRead( DHTPIN );

        if ( counter == 255 )
            break;

        /* ignore first 3 transitions */
        if ( ( i >= 4 ) && ( i % 2 == 0 ) )
        {
            /* shove each bit into the storage
bytes */
            dht11_dat[j / 8] <= 1;
            if ( counter > 16 )
                dht11_dat[j / 8] |= 1;
            j++;
        }
    }
}
/*

```

```

        * check we read 40 bits (8bit x 5 ) + verify checksum
in the last byte
        * print it out if data is good
        */
        if ( ( j >= 40 ) &&
            (dht11_dat[4] == ( (dht11_dat[0] + dht11_dat[1] +
dht11_dat[2] + dht11_dat[3]) & 0xFF) ) )
        {
            f = dht11_dat[2] * 9. / 5. + 32;
            printf( "Humidity = %d.%d %%\n",
Temperature = %d.%d *C (%.1f *F)\n",
                dht11_dat[0], dht11_dat[1],
dht11_dat[2], dht11_dat[3], f );
        }else {
            printf( "Data not good, skip\n" );
        }
    }

int main( void )
{
    printf( "Raspberry Pi wiringPi DHT11 Temperature
test program\n" );

    if ( wiringPiSetup() == -1 )
        exit( 1 );

    while ( 1 )
    {
        read_dht11_dat();
        delay( 1000 ); /* wait 1sec to refresh */
    }

    return(0);
}

```

C. Código da Câmera

```

Em Python:
from picamera import PiCamera
from time import sleep
camera = PiCamera()
camera.start_preview()

for i in range(3):
    sleep(1)

camera.capture('/home/pi/projeto_final_eduardo/python/fotos/i
mage%s.jpg' % i)

camera.stop_preview()

```

D. Código principal e final(como capturar uma foto do movimento detectado)

```

#include <wiringPi.h>
#include <stdio.h>
#include <sys/types.h>

```

```

#include <signal.h>
#include <pthread.h>
#include <stdlib.h>
#include <stdint.h>
#include <python2.7/Python.h>

#define MAXTIMINGS      85
#define DHT_PIN         7
int dht11_dat[5] = { 0, 0, 0, 0, 0 };

//copilar gcc codigo_pi_principal.c -o codigo_pi_principal -
g -Wall -lpython2.7.9 -lm -L/usr/lib/python2.7.9/config
//./codigo_main_principal

void sensor_dht11_dat();

//FUNÇÃO PARA REALIZAR A LEITURA DA PORTA
do sensor de presença PIR
void sensor_PIR(void)
{
    //      pinMode(0,INPUT);
    //      int i=0;

    //Leitura do pino 17, segundo o mapa de pinos
da biblioteca wiringpi
    //      i=digitalRead (0); //porta 17 do RASPEBERRY
PI

    //      if (i == 1)
    //      {
    //          //Sensor com movimento
    //          printf("Presença de Movimento
detectado\n");

    //          //Acionar a camera;

    //          FILE *python;

    //          Py_Initialize();

    //          python = fopen("codigo_camera.py",
"r");

    //          PyRun_SimpleFile(python,
"codigo_camera.py");

    //          Py_Finalize();

    //          //Leitura do sensor de temperatura e
umidade

    //          printf( "Dados do local\n" );
    //          sensor_dht11_dat();
    //          delay( 1000 ); /* Aguardo 1 segundo
*/
}

//          }
}

void sensor_dht11_dat()
{
    uint8_t ultimo_estado      = HIGH;
    uint8_t contador           = 0;
    uint8_t j                  = 0, i;
    float      f; /* fahrenheit */

    dht11_dat[0] = dht11_dat[1] = dht11_dat[2] =
dht11_dat[3] = dht11_dat[4] = 0;

    /* Pino de saida depois esperar 18 milliseconds
*/

    pinMode( DHT_PIN, OUTPUT );
    digitalWrite( DHT_PIN, LOW );
    delay( 18 );
    /* pino alto depois espera 40 microseconds */
    digitalWrite( DHT_PIN, HIGH );
    delayMicroseconds( 40 );
    /* Prepara o pino para entrada */
    pinMode( DHT_PIN, INPUT );

    /* Realizar a leitura */
    for ( i = 0; i < MAXTIMINGS; i++ )
    {
        contador = 0;
        while ( digitalWrite( DHT_PIN ) ==
ultimo_estado )
        {
            contador++;
            delayMicroseconds( 1 );
            if ( contador == 255 )
            {
                break;
            }
        }
        ultimo_estado = digitalWrite(
DHT_PIN );
    }
}

```

```

        if ( contador == 255 )
            break;

        if ( (i >= 4) && (i % 2 == 0) )
        {
            dht11_dat[j / 8] <<= 1;
            if ( contador > 16 )
                dht11_dat[j / 8]
|= 1;

            j++;
        }

    }

    /*
    * check we read 40 bits (8bit x 5) + verify
checksum in the last byte
    * print it out if data is good
    */
    if ( (j >= 40) &&
        (dht11_dat[4] == ( (dht11_dat[0] +
dht11_dat[1] + dht11_dat[2] + dht11_dat[3]) & 0xFF) ) )
    {
        f = dht11_dat[2] * 9. / 5. + 32;
        printf( "Umidade = %d.%d %%\n",
Temperatura = %d.%d *C (%.1f *F)\n",
                dht11_dat[0],
dht11_dat[1], dht11_dat[2], dht11_dat[3], f );
    }else {
        printf( "Sem bons dados\n" );
    }
}

```

```

int main(){

    printf( "Projeto de Monitoramento residencial -
Projeto Embarcados 2017" );

    if ( wiringPiSetup() == -1 )
        exit( 1 );

    // while ( 1 )
    // {

```

```

//          leitura do sensor PIR
//          sensor_PIR();
//
//          }

//Realiza a leitura quando o sensor PIR É
ACIONADO NA BORDA
wiringPiISR(0,INT_EDGE_RISING,
sensor_PIR);

//pthread_exit(NULL);
return 0;

}

```

CONCLUSÃO

Nesse trabalho foi possível conceber um sistema de vigilância inteligente capaz de capturar uma imagem ou gravar um vídeo, e transmiti-los para um telefone inteligente via e-mail. É vantajoso, pois oferece confiabilidade, privacidade e segurança, tanto partindo do ponto de envio como do ponto de chegada.

Do ponto de vista de quem recebe a informação, a vantagem é que as ações iniciais podem ser tomadas em curto espaço de tempo no caso de condições de emergência como: pessoa idosa ou doente, áreas militares, casas inteligentes, escritórios, indústrias, etc.

As perspectivas para trabalho futuro é localizar o número de pessoas que estão exatamente nas áreas mencionadas anteriormente e suas posições, para que, informações precisas possam ser obtidas para o destinatário.

REFERENCES.

- [1] Disponível em: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. Último acesso em Maio de 2017.
- [2] Disponível em: <https://www.raspberrypi.org/products/camera-module-v2/>. Último acesso em Maio de 2017.
- [3] Disponível em: <http://www.filipeflop.com/pd-6b901-sensor-de-movimento-presenca-pir>. Último acesso em Maio de 2017.
- [4] Disponível em: <http://blog.everpi.net/2014/07/raspberry-pi-sensor-temperatura-humidade-dht11.html>. Último acesso em Maio de 2017.
- [5] Disponível em: <http://blog.everpi.net/2015/10/raspberry-pi-ligar-pir-sensor-movimento-hc-sr501.html/>. Último acesso em Maio de 2017.
- [6] .