

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**CAMPUS CAMPO MOURÃO**

**DISCIPLINA DE SISTEMAS EMBARCADOS – LT38C**

***WWW.LT38C.HTURBO.COM***

**ALUNOS: AMANDA SABAINE PINTO, BRUNO SCHUAVAB PENHA, MARIANA EMER DE QUEIROZ**

**SENSOR DE PRESENÇA HCSR 501 PIR**

**PROJETO FINAL DE DISCIPLINA**

**CAMPO MOURÃO, DEZEMBRO 2015**

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

Graduação em Engenharia Elétrônica

|  |
| --- |
| **TRABALHO DE CONCLUSÃO DE DISCIPLINA**apresentado a UTFPRpara obtenção da nota final referente à disciplina de**SISTEMAS EMBARCADOS**por **ALUNOS: AMANDA SABAINE PINTO, BRUNO SCHUAVAB PENHA, MARIANA EMER DE QUEIROZ** |
| **SENSOR DE PRESENÇA HCSR 501 PIR** |

|  |  |
| --- | --- |
| **PROF. MSC. PAULO DENIS GARCEZ DA LUZ** | **UTFPR** |

CAMPO MOURÃO, DEZEMBRO 2015.

**ALUNOS: AMANDA SABAINE PINTO, BRUNO SCHUAVAB PENHA, MARIANA EMER DE QUEIROZ**

**SENSOR DE PRESENÇA HCSR 501 PIR**

Trabalho de conclusão de disciplina apresentada ao Professor de Sistemas Embarcos no curso de Graduação em Engenharia Elétrônica da Universidade Tecnológica Federal do Paraná, como requisito parcial para a obtenção da aprovação na disciplina.

Orientador: Prof. Msc. Paulo Denis Garcez da Luz

CAMPO MOURÃO, DEZEMBRO 2015

**LISTA DE FIGURAS**

[Figura 1 - Layout Raspberry Pi 3](#_Toc437436337)

[Figura 2 - Tela de configuração Raspberry Pi 4](#_Toc437436338)

[Figura 3 – Configuração teclado 1 4](#_Toc437436339)

[Figura 4 – Configuração do teclado 2 4](#_Toc437436340)

[Figura 5 – Configuração do fuso horário 1 5](#_Toc437436341)

[Figura 6 – Configuração do fuso horário 2 5](#_Toc437436342)

[Figura 7 – Configuração do fuso horário 3 5](#_Toc437436343)

[Figura 8 – Encerramento das configurações 6](#_Toc437436344)

[Figura 9 - Expansão do micro-sd 6](#_Toc437436345)

[Figura 10 - Confirmação de configuração 6](#_Toc437436346)

[Figura 11 - Ligação elétrica do sensor 16](#_Toc437436347)

**LISTA DE ABREVIATURAS E SIGLAS**

GPIO - General Purpose Input/Output

MCedit - Editor de texto Linux

MySql - Sistema de gerencia de banco de dados

RISC - Reduced Instruction Set Computer

SQL - Structured Query Language

**SUMÁRIO**

[INTRODUÇÃO 1](#_Toc434406592)

[1.1 SISTEMAS EMBARCADOS 1](#_Toc434406593)

[1.2 MOTIVAÇÕES 1](#_Toc434406594)

[1.3 OBJETIVOS 2](#_Toc434406595)

[1.4 ESTRUTURA DO TRABALHO 2](#_Toc434406596)

[INSTALAÇÂO DO SISTEMA INICIAL 3](#_Toc434406597)

[2.1 INTRODUÇÃO 3](#_Toc434406598)

[2.2 DESCRIÇÃO SUSCINTA DAS INTALAÇÔES 3](#_Toc434406599)

[2.2.1 BCM 2835-1.45 8](#_Toc434406600)

[2.2.2 WIRING PI 9](#_Toc434406601)

[2.2.3 MYSQL 9](#_Toc434406601)

[2.4 CONCLUSÃO 9](#_Toc434406603)

[PROGRAMAS CONSTRUÍDOS OU ADAPTADOS 10](#_Toc434406604)

[3.1 INTRODUÇÃO 10](#_Toc434406605)

[3.2 PROGRAMA PRINCIPAL 10](#_Toc434406606)

[3.3 FLUXO DAS INFORMAÇÕES 13](#_Toc434406609)

[3.4 ESTRUTURA DO BANCO DE DADOS 13](#_Toc434406610)

[3.5 ESTRUTURA DOS DIRETÓRIOS DO PROJETO 14](#_Toc434406611)

[3.6 CONCLUSÃO 14](#_Toc434406612)

[DESCRIÇÃO DO HARDWARE ADICIONAL NA GPIO 15](#_Toc434406613)

[4.1 INTRODUÇÃO 15](#_Toc434406614)

[4.2 SENSOR DE PRESENÇA HCSR 501 PIR 15](#_Toc434406615)

[4.3 CONCLUSÃO 16](#_Toc434406616)

REFERÊNCIAS BIBLIOGRÁFICAS 18

APÊNDICE 1 – INSTALADOR WEB 19

**RESUMO**

O presente trabalho tem como objetivo demonstrar os passos do projeto desenvolvido na disciplina de Sistemas Embarcados. O projeto consiste em um software capaz de monitorar e armazenar as detecções de um sensor de presença (HSCR 501 PIR), através do sistema embarcado Raspberry Pi, onde as informações são registradas em um banco de dados (MySQL).

**CAPÍTULO 1**

# INTRODUÇÃO

## 1.1 SISTEMAS EMBARCADOS

O sistema embarcado, também chamado de sistema embutido, é um sistema microprocessado em que um computador está anexado ao sistema que ele controla. Um sistema embarcado pode realizar um conjunto de tarefas que foram predefinidas. O sistema é usado para tarefas específicas, e assim, através de engenharia é possível otimizar um determinado produto e diminuir o tamanho, bem como os recursos computacionais e o seu valor final.

Estes podem ser definidos como dispositivos que funcionam como computadores, que contam com memória, processador, interface de entrada e saída, porém, com o diferencial que desempenham uma tarefa específica. Por essa razão, usamos o micro-ondas como exemplo. No geral, os sistemas são usados em aplicações sem muitas exigências. Os [sistemas embarcados](https://www.oficinadanet.com.br/artigo/737/desafios_com_os_sistemas_de_informacao) costumam ser projetados a partir de uma aplicação. O controle remoto é outro bom exemplo, dentro dele há processador, memória, software, enfim, todos os itens encontrados em um computador convencional. [Sistemas](https://www.oficinadanet.com.br/artigo/738/tipos_de_sistemas_de_informacao_na_empresa) embarcados costumam ser desenvolvidos para uma tarefa específica. O sistema pode ainda ser executado com recursos computacionais bastante limitados, ou seja, com a ausência de teclado, tela e mesmo com pouca memória.

## 1.2 MOTIVAÇÕES

Com a automatização de sistemas e processos, se faz necessário o uso de sistemas de monitoramento e registro de eventos dos mesmos, para maior obtenção de informações e maior controle dos dados obtidos. Dado a crescente busca por tecnologias cada vez mais acessíveis e por concorrências mais acirradas, a utilização de sistemas embarcados tem crescido muito, devido a sua alta eficiência e custo relativamente baixo.

## 1.3 OBJETIVOS

O objetivo deste trabalho é desenvolver um software capaz de monitorar e registrar eventos de um sensor de presença (HCSR 501 PIR). Tendo como finalidade, monitorar o sensor, registrar qualquer ocorrência de eventos no mesmo, registrando data da ocorrência e momento em que houve ocorrência.

## 1.4 ESTRUTURA DO TRABALHO

O trabalho será dividido em quatro capítulos, sendo que o primeiro levará em consideração uma breve explicação sobre sistemas embarcados, juntamente com objetivos e motivações.

 O segundo capítulo abordará as instalações necessárias e os primeiros ajustes para realizar o trabalho, juntamente com as principais bibliotecas e as especificações e comandos das mesmas.

 Já no terceiro capítulo será feita uma descrição do programa principal adaptado, para que houvesse a realização do projeto e também sua funcionalidade.

 E o por fim, o quarto capítulo explicará sucintamente algumas funcionalidades como fazer a instalação na Raspberry, alémde algumas considerações finais para garantir a finalidade do trabalho.

**CAPÍTULO 2**

# INSTALAÇÃO DO SISTEMA INICIAL

## 2.1 INTRODUÇÃO

O sistema Linux que será utilizado é o Raspbian, como ele é um software livre, oferece mais de 35.000 pacotes de de software, os quais são pré-compilados, para serem facilmente instalados na Raspberry Pi.

## 2.2 DESCRIÇÃO SUSCINTA DAS INTALAÇÔES

Inicia-se a instalação no micro-SD com o sistema RASPIBIAN encontrado no site da Raspberry Foundation. A instalação se dá em alguns passos, como a Raspberry não possui um memória não volátil é necessário utilizar em conjunto um cartão de memória SSD para armazenar o sistema operacional e os diversos programas, atualizações e pacotes que serão utilizados na desenvoltura do projeto.

Durante o processo de execução do projeto é necessário fazer a instalação de pacotes que contém instruções e comandos que facilitam a utilização da entrada e saída da placa GPIO (pinos) e facilitam e aprimoram os programas.

Primeiramente é necessário fazer o download do programa Win32DiskImager, que é um programa que possui uma plataforma que faz o boot (instalação) do RASPIBIAN para o cartão de memória, que posteriormente será conectado na placa e será executável quando a placa é alimentada.

Após o sistema operacional instalado, já se pode ligar a placa Raspberry PI. A placa oferece a conexão de vários periféricos, como visto na figura 1.



Figura 1 - Layout Raspberry Pi

Após a instalação e a conexão dos periféricos necessários, ao inicializar pela primeira vez o sistema operacional aparecerá a tela de configurações da Raspberry PI, denominada Rasp-Config (Figura 2). Faz-se necessário então a realização das configurações da placa.



Figura 2 - Tela de configuração Raspberry Pi

O passo seguinte será a configuração do teclado, nesse caso, o teclado padronizado brasileiro.

.



Figura 3 – Configuração teclado 1



Figura 4 – Configuração do teclado 2

A próxima etapa é selecionar o fuso horário da raspberry, de modo que ela utiliza isso para atualizar sua data e hora, pois ela os atualiza via internet.



Figura 5 – Configuração do fuso horário 1



Figura 6 – Configuração do fuso horário 2



Figura 7 – Configuração do fuso horário 3



Figura 8 – Encerramento das configurações

Logo, pode-se iniciar a configuração da expansão da partição raiz do cartão SD, com o objetivo de tornar toda a memória utilizado, confirmando com a opção em “OK”.



Figura 9 - Expansão do micro-sd



Figura 10 - Confirmação de configuração

Prosseguindo, o SO define dois grandes usuários o ROOT, que possui permissão para fazer alterações ilimitadas e o PI, usuário com menor poder nas configurações básicas, comumente utiliza-se o comando sudo.

Todo o desenvolvimento do projeto é feito em ambiente, similar ao terminal, ou seja, sem um ambiente gráfico, somente texto. Por isso é necessário um editor de texto compatível e de fácil manipulação, o programa que será utilizado é o MCedit. Alguns comandos são essenciais, para o melhor funcionamento da Raspberry, que são responsáveis por atualizar o software, instalar programas, alterar processos e afins.

* Atualização do software e instalação do editor de texto Mcedit:

apt-get update - Realiza a atualização do sistema;

apt-get install mc - Inicializa a instalação do editor de texto mcedit;

apt-get upgrade - Atualiza o Sistema da Raspberry PI.

* Configuração dos atalhos do profile Pi:

cd /root/etc - entrada no diretório

mcedit profile - abertura através do mcedit do arquivo profile

Após os comandos, no final do arquivo insere-se:

alias ll='ls -l --color=auto'

alias la='ls -la --color=auto'

export EDITOR=mcedit

export HISTSIZE=1000

export HISTFILESIZE=1000

* Alteração das configurações do root:

cd /root - entrada no diretório

mcedit .bashrc - abertura através do mcedit do arquivo .bashrc

Após os comandos, no final do arquivo insere-se:

alias ll='ls -l --color=auto'

alias la='ls -la --color=auto'

export EDITOR=mcedit

export HISTSIZE=1000

export HISTFILESIZE=1000

* Configuração da tela cheia:

cd /boot - entrada no diretório

mcedit config.txt - abertura através do mcedit do arquivo config.txt

Após os comandos, retira-se o caracter '#' antes de: "disable\_overscan=1"

* Desabilitação da necessidade do login:

cd /root/etc - entrada no diretório

mcedit inittab - abertura do arquivo

Após os comandos, abaixo de #1:2345:respawn:/sbin... insere-se: #1:2345:respawn:/bin/login -f pi tty1 </dev/tty1 >/dev/tty1 2>&12.2.1 BCM 2835-1.45

Tanto para configurar o root e pi, quanto configurar tela cheia e login deve-se dar reboot para aplicar as mudanças

A seguir serão instalados os seguintes pacotes e serviços adicionais, que auxiliarão em funções, que facilitam o entendimento e desenvolvimento do projeto. No projeto em questão foram utilizados os pacotes BCM 2835-versão 1.45, Wiring pi e MySQL.

### 2.2.1 BCM 2835-1.45

É uma biblioteca em C para a Raspberry, tal qual possui comandos para controlar entradas e saídas, facilitando o acesso dos pinos. Ela fornece funções para a leitura das entradas digitais e define saídas digitais. Para efetuar a instalação é necessário seguir os passos abaixo:

wget www.lt38c.hturbo.com/bcm2835-1.45.tar.gz -O /tmp/bcm2835.tar.gz

cd/tmp

tar xvfz bcm2835-1.45.tar.gz

cd bcm2835-1.45

./configure

make

sudo make check

sudo make install

### 2.2.2 WIRING PI

Semelhante à biblioteca BCM2835, tem a função de auxiliar e facilitar os comandos das entradas e saída de sinais da placa. Comandos de instalação:

sudo apt-get install git-core

wget www.lt38c.hturbo.com/wiringPi-5edd177.tar.gz

tar xvfz wiringpi-5edd177.tar.gz

cd wiringpi-5edd177

./build

gpio – v

gpio readall

2.2.3 MYSQL

O MYSQL gerencia o banco de dados para servidores em plataforma SQL*,* é de fácil aplicação e desenvolvimento. Possui uso gratuito tanto para fins comerciais, quanto particulares. Sua instalação não difere dos pacotes anteriormente instalados. Segue abaixo os comandos necessários para concluir a instalação da biblioteca:

sudo apt-get install libmysql++3

sudo apt-get install libmysql+-dev

## 2.3 CONCLUSÃO

Neste capítulo foi apresentada de forma sucinta a estrutura do projeto, mostrando alterações para que o sistema entre sem pedir senha, posssua tela dimensionada para uso de 100% da área. Tenha a GPIO liberada para acesso via programação C, e também o servidor Mysql funcionando corretamente. Assim, após as instalações e configurações iniciais a Raspberry Piestará pronta para ser utilizada e descarregar o programa principal e suas pertinências. Juntamente com a conexão com o sensor de presença.

**CAPÍTULO 3**

# PROGRAMAS CONSTRUÍDOS OU ADAPTADOS

## 3.1 INTRODUÇÃO

O desenvolvimento do programa, foi baseado na linguagem de programação C, e utilizou-se de recursos do Mysql para a criação dos bancos de dados. Muitos dos comando utilizados, são ferramentas já dispostas nas bibliotecas instaladas.

## 3.2 PROGRAMA PRINCIPAL

#include<stdio.h>

#include<stdlib.h>

#include<bcm2835.h>

#include<time.h>

#include<mysql/mysql.h>

#include<string.h>

#define SENSOR RPI\_GPIO\_P1\_07

#define LED RPI\_GPIO\_P1\_11

#define ACTIVE\_VALUE 1 int gravar\_banco();

int sensoriamento();

int conectarBanco();

void convertData(char \*string);

void convertHora(char \*string);

MYSQL conexao;

char data[15];

char hora[15];

void main() {

int x=conectarBanco("localhost","root","root","projeto");

if(x=1)

sensoriamento();

}

int sensoriamento()

{

if(!bcm2835\_init())

{

printf("por favor executar como super usuario\n");

return EXIT\_FAILURE;

}

bcm2835\_gpio\_fsel(SENSOR, BCM2835\_GPIO\_FSEL\_INPT);

bcm2835\_gpio\_fsel(LED,BCM2835\_GPIO\_FSEL\_OUTP);

bcm2835\_gpio\_set\_pud(SENSOR, BCM2835\_GPIO\_PUD\_UP);

uint8\_t state = ACTIVE\_VALUE;

while(1)

{

state=bcm2835\_gpio\_lev(SENSOR);

if(state!=ACTIVE\_VALUE)

{

printf("Nada detectado\n");

sleep(2);

}

else

{

gravarBanco();

printf("Algo foi detectado\nGravando no banco\n");

sleep(2);

}

}

}

int gravarBanco()

{

time\_t timer = time(NULL);

char \*dataHora;

dataHora = ctime(&timer);

convertData(dataHora);

convertHora(dataHora);

char comando[150]="insert into sensor(data, hora, sensor) values('"; strcat(comando,data); strcat(comando,"','");

strcat(comando,hora);

strcat(comando,"','BCM2835\_PIR');");

mysql\_query(&conexao,comando);

}

int conectarBanco(char endereco[], char usuario[], char senha[], char banco[])

{

mysql\_init(&conexao);

if(mysql\_real\_connect(&conexao,endereco,usuario,senha,banco,0,NULL,0))

{

printf("banco conectado!!!\n");

return 1;

}

else

{

printf("Ocorreu um erro ao conectar banco!!!\n");

return 0;

 }

}

void convertData(char \*string)

{

data[0]=string[8];

data[1]=string[9];

data[2]=string[7];

data[3]=string[4];

data[4]=string[5];

data[5]=string[6];

data[6]=string[7];

data[7]=string[20];

data[8]=string[21];

data[9]=string[22];

data[10]=string[23];

}

void convertHora(char \*string)

{

hora[0]=string[11];

hora[1]=string[12];

hora[2]=string[13];

hora[3]=string[14];

hora[4]=string[15];

hora[5]=string[16];

hora[6]=string[17];

hora[7]=string[18];

}

## 3.3 FLUXO DAS INFORMAÇÕES

O programa recebe cada detecção de movimento do sensor, indicando quando há ou não há essa detecção, com impressão em tela. Conforme há o registro, armazena-se no banco de dados, em uma tabela, a data e a hora do mesmo, tendo uma precisão de diferença de segundos entre os dados.

## 3.4 ESTRUTURA DO BANCO DE DADOS

A linguagem SQLé uma das linguagens mais utilizadas para consulta, inclusão, exclusão e alteração de dados em bases de dados relacionais e objeto-relacionais. Dentre os grandes gerenciadores de banco de dados que se utilizam desta linguagem pode-se citar o Oracle, o Microsoft SQL Server, o Sybase, o Postgree SQL, o MySql entre outros. A linguagem SQL é um conjunto de comandos de manipulação de banco de dados utilizado para criar e manter a estrutura desse banco de dados, além de incluir, excluir, modificar e pesquisar informações nas tabelas dele (OLIVEIRA, 2002).

A linguagem SQL não é uma linguagem de programação autônoma, podendo ser chamada de “sublinguagem”. Quando se escrevem aplicações para banco de dados, é necessário utilizar uma linguagem de programação tradicional (C, Java, Pascal, Cobol etc.) e embutir comandos SQL para manipular os dados (OLIVEIRA, 2002).

## 3.5 ESTRUTURA DOS DIRETÓRIOS DO PROJETO

A estrutura dos diretórios do projeto é a seguinte:

/lt38c

/lt38c/2015\_2s\_sensor\_presença

/lt38c/2015\_2s\_ sensor\_presença /apt

/lt38c/2015\_2s\_ sensor\_presença /arquivos.sys : arquivos modificados;

/lt38c/2015\_2s\_ sensor\_presença /exe : arquivos execut

/lt38c/2015\_2s\_ sensor\_presença /fontes.c : arquivos em linguagem C;

/lt38c/2015\_2s\_ sensor\_presença /history : histórico;

/tl38c/2015\_2s\_ sensor\_presença /libs : bibliotecas;

/lt38c/2015\_2s\_ sensor\_presença /maff-mht : tutoriais;

/lt38c/2015\_2s\_ sensor\_presença /mysql.sql : cópia banco de dados;

/lt38c/2015\_2s\_ sensor\_presença /site : sites utilizados;

/lt38c/2015\_2s\_ sensor\_presença /vídeos : vídeos utilizados.

## 3.6 CONCLUSÃO

O programa principal, foi uma adaptação dependente das especificações solicitadas, de modo que o mesmo recebe informações do sensor, gerencia o banco de dados, e cria a tabela com as detecções, juntamente com a atualização do banco de dados. Boa parte do programa utiliza-se de comandos existentes nas bibliotecas anteriormente instaladas.

**CAPÍTULO 4**

# DESCRIÇÃO DO HARDWARE ADICIONAL NA GPIO

## 4.1 INTRODUÇÃO

Para este projeto foi necessário acoplar o sensor de presença HCSR 501 PIR na GPIO. O sensor é um módulo automático de controle que utiliza um sensor piroelétrico capaz de detectar movimentos baseados na variação da luz infravermelha emitida pela radiação do corpo humano.

## 4.2 SENSOR DE PRESENÇA HCSR 501 PIR

Quando em uso, o sensor é capaz de detectar movimento em uma área de até 7 metros trabalhando com um ângulo de até 100°, é claro que como age por sistema infravermelho capta somente movimento de objetos que exalam calor como, por exemplo, o corpo humano.

No momento em que o sensor detecta algum movimento em sua área de alcance, o mesmo emite uma um sinal de 3.3V através de seu pino OUT. O sensor possui outra peculiaridade que é o longo atraso que o mesmo pode demonstrar logo após de realizar a detecção. Isto ocorre pois em sua parte traseira o sensor PIR possui um potenciômetro que permite uma configuração mínima de aproximadamente 3 segundos.

Este sensor PIR possui ainda a capacidade de ajuste para o tempo de retardo (de até 200 segundos), isto é, o tempo ao qual o sensor continuará a enviar sinal do pino OUT através do VR1 e ainda o ajuste da sensibilidade através do VR2. Além disso, possui em sua parte frontal uma proteção removível que garante ao seu sensor uma maior proteção sem interferir em sua verificação.

Características**:**

* Sensor PIR;
* Realiza detecção através de sistema infravermelho;
* Detecta movimento em uma área de até 7 metros;
* Raio de alcance de aproximadamente 100°;
* Ajuste do tempo de retardo e sensibilidade;
* Tempo de retardo máximo de 200 segundos;
* Possui proteção para o ponto sensorial infravermelho.

Especificações:

* Modelo: HCSR501;
* Chip: BISS0001;
* Alimentação: 5V - 20VDC;
* Saída TTL/OUT: 3.3V;
* Tempo de retardo ajustável: 2s - 200s;
* Distância detectável (ajustável): 3m - 7m;
* Ângulo de detecção: <100°;
* Dimensões (CxLxA): 32x24x24mm;
* Peso: 5g.

A ligação elétrica do sensor é bem simples e está exemplificada na figura 11. Têm-se os pinos VCC, GND e OUT. Eles são ligados ao 5V, GND e GPIO dos Raspberry Pi. A conexão é a mesma em qualquer modelo da placa e o GPIO pode ser trocado por outro, caso seja necessário. Como a saída do pino OUT desse modelo de sensor é 3.3V, será ligado diretamente ao Raspberry Pi.



Figura 11 - Ligação elétrica do sensor

4.3 CONCLUSÃO

A estrutura de hardwre proposta é uma das muitas possibilidades existentes para conseguir o monitoramento de dados em tempo real. Após a instalação do programa e sua execução, basta apenas instalar e conectar o sensor de presença. E observar os valores e as informações contidas no banco de dados.

REFERÊNCIAS BIBLIOGRÁFICAS

 [1] ARLEI. **Comandos Básicos**. Disponível em: <http://wiki.ubuntu-br.org/ComandosBasicos>.

[2] **Como conectar o sensor PIR HC-SR501 ao Raspberry Pi.** Blog Every Pi.

Disponivel em: <<http://blog.everpi.net/2015/10/raspberry-pi-ligar-pir-sensor-movimento-hc-sr501.html>>.

 [3] LUCCA, R. R.. **Usando MySQL na linguagem C.** Disponível em: <http://www.vivaolinux.com.br/artigo/Usando-MySQL-na-linguagem-C?pagina=1>.

[4] OLIVEIRA, André Scheneider; ANDRADE, Fernando S. **Sistemas Embarcados, hardware e firmware na prática.** 1ªEdição. Editora Érica. 2006.

[5] **Primeiros passos para configuração da raspberryPI**. Disponível em: < http://rpi.nersd.ifsc.edu.br/?p=18>.

[6] UPTON, Eben. **Raspberry Pi- Manual do usuário.** Editora Novatec. São Paulo. 2013.

**APÊNDICE 1**

**INSTALADOR WEB**

#
apt-get update
apt-get install mc cmake libmysql++-dev libmysql++3 -y

#Instalar MYSQL sem senha
export DEBIAN\_FRONTEND=noninteractive
apt-get -q -y install mysql-server

#configurar SENHA do MYSQL
mysqladmin -u root password root

#bcm2835
wget [www.lt38c.hturbo.com/bcm2835-1.45.tar.gz](http://www.lt38c.hturbo.com/bcm2835-1.45.tar.gz) -O /tmp/bcm2835.tar.gz
cd /tmp
tar -zxvf bcm2835.tar.gz
cd bcm2835-1.45
./configure
make #compilar
make check #checar
make install #instalar

#wiringPi
cd /tmp
wget [www.lt38c.hturbo.com/wiringPi-5edd177.tar.gz](http://www.lt38c.hturbo.com/wiringPi-5edd177.tar.gz) -O /tmp/wiringPi.tar.gz
tar -zxvf wiringPi.tar.gz
cd wiringPi-5edd177
./build #compila,checa e instala

#Criacao de pastas e downloads
mkdir /lt38c
mkdir /lt38c/2015\_2s\_sensor\_presenca
mkdir /lt38c/2015\_2s\_sensor\_presenca/arquivos.sys
mkdir /lt38c/2015\_2s\_sensor\_presenca/fontes.c
mkdir /lt38c/2015\_2s\_sensor\_presenca/maff-mht
mkdir /lt38c/2015\_2s\_sensor\_presenca/mysql.sql
mkdir /lt38c/2015\_2s\_sensor\_presenca/site
mkdir /lt38c/2015\_2s\_sensor\_presenca/history
mkdir /lt38c/2015\_2s\_sensor\_presenca/exe
mkdir /lt38c/2015\_2s\_sensor\_presenca/videos
mkdir /lt38c/2015\_2s\_sensor\_presenca/apt

#bkp dos arquivos.sys
wget [www.lt38c.hturbo.com/Projetos/2015\_2s\_sensor\_presenca/arquivos.sys/config.txt](http://www.lt38c.hturbo.com/Projetos/2015_2s_sensor_presenca/arquivos.sys/config.txt) -O /tmp/config.txt
chmod 755 /tmp/config.txt
chown root:root /tmp/config.txt
cp /tmp/config.txt /lt38c/2015\_2s\_sensor\_presenca/arquivos.sys
cp /lt38c/2015\_2s\_sensor\_presenca/arquivos.sys/config.txt /boot

wget [www.lt38c.hturbo.com/Projetos/2015\_2s\_sensor\_presenca/arquivos.sys/.bashrc](http://www.lt38c.hturbo.com/Projetos/2015_2s_sensor_presenca/arquivos.sys/.bashrc) -O /tmp/.bashrc
chmod 644 /tmp/.bashrc
chown root:root /tmp/.bashrc
cp /tmp/.bashrc /lt38c/2015\_2s\_sensor\_presenca/arquivos.sys
cp /lt38c/2015\_2s\_sensor\_presenca/arquivos.sys/.bashrc /root

wget [www.lt38c.hturbo.com/Projetos/2015\_2s\_sensor\_presenca/arquivos.sys/my.cnf](http://www.lt38c.hturbo.com/Projetos/2015_2s_sensor_presenca/arquivos.sys/my.cnf) -O /tmp/my.cnf
chmod 644 /tmp/my.cnf
chown root:root /tmp/.bashrc
cp /tmp/my.cnf /lt38c/2015\_2s\_sensor\_presenca/arquivos.sys
cp /lt38c/2015\_2s\_sensor\_presenca/arquivos.sys/my.cnf /etc/mysql

wget [www.lt38c.hturbo.com/Projetos/2015\_2s\_sensor\_presenca/arquivos.sys/profile](http://www.lt38c.hturbo.com/Projetos/2015_2s_sensor_presenca/arquivos.sys/profile) -O /tmp/profile
chmod 644 /tmp/profile
chown root:root /tmp/profile
cp /tmp/profile /lt38c/2015\_2s\_sensor\_presenca/arquivos.sys
cp /lt38c/2015\_2s\_sensor\_presenca/arquivos.sys/profile /etc

wget [www.lt38c.hturbo.com/Projetos/2015\_2s\_sensor\_presenca/arquivos.sys/rc.local](http://www.lt38c.hturbo.com/Projetos/2015_2s_sensor_presenca/arquivos.sys/rc.local) -O /tmp/rc.local
chmod 755 /tmp/rc.local
chown root:root /tmp/rc.local
cp /tmp/rc.local /lt38c/2015\_2s\_sensor\_presenca/arquivos.sys
cp /lt38c/2015\_2s\_sensor\_presenca/arquivos.sys/rc.local /etc/init.d

wget [www.lt38c.hturbo.com/Projetos/2015\_2s\_sensor\_presenca/arquivos.sys/locale](http://www.lt38c.hturbo.com/Projetos/2015_2s_sensor_presenca/arquivos.sys/locale) -O /tmp/locale
chmod 644 /tmp/locale
chown root:root /tmp/locale
cp /tmp/locale /lt38c/2015\_2s\_sensor\_presenca/arquivos.sys
cp /lt38c/2015\_2s\_sensor\_presenca/arquivos.sys/locale /etc/default

wget [www.lt38c.hturbo.com/Projetos/2015\_2s\_sensor\_presenca/arquivos.sys/keyboard](http://www.lt38c.hturbo.com/Projetos/2015_2s_sensor_presenca/arquivos.sys/keyboard) -O /tmp/keyboard
chmod 644 /tmp/keyboard
chown root:root /tmp/keyboard
cp /tmp/keyboard /lt38c/2015\_2s\_sensor\_presenca/arquivos.sys
cp /lt38c/2015\_2s\_sensor\_presenca/arquivos.sys/keyboard /etc/default

wget [www.lt38c.hturbo.com/Projetos/2015\_2s\_sensor\_presenca/arquivos.sys/timezone](http://www.lt38c.hturbo.com/Projetos/2015_2s_sensor_presenca/arquivos.sys/timezone) -O /tmp/timezone
chmod 644 /tmp/timezone
chown root:root /tmp/timezone
cp /tmp/timezone /lt38c/2015\_2s\_sensor\_presenca/arquivos.sys
cp /lt38c/2015\_2s\_sensor\_presenca/arquivos.sys/timezone /etc

wget [www.lt38c.hturbo.com/Projetos/2015\_2s\_sensor\_presenca/arquivos.sys/inittab](http://www.lt38c.hturbo.com/Projetos/2015_2s_sensor_presenca/arquivos.sys/inittab) -O /tmp/inittab
chmod 644 /tmp/inittab
chown root:root /tmp/initab
cp /tmp/inittab /lt38c/2015\_2s\_sensor\_presenca/arquivos.sys
cp /lt38c/2015\_2s\_sensor\_presenca/arquivos.sys/inittab /etc
#fim do bkp dos arquivos.sys

wget [www.lt38c.hturbo.com/Projetos/2015\_2s\_sensor\_presenca/history/history](http://www.lt38c.hturbo.com/Projetos/2015_2s_sensor_presenca/history/history) -O /tmp/history
cp /tmp/history /lt38c/2015\_2s\_sensor\_presenca/history

#bkp do FONTES.C
wget [www.lt38c.hturbo.com/Projetos/2015\_2s\_sensor\_presenca/fontes.c/codigo1.c](http://www.lt38c.hturbo.com/Projetos/2015_2s_sensor_presenca/fontes.c/codigo1.c) -O /tmp/codigo1.c
cp /tmp/codigo1.c /lt38c/2015\_2s\_sensor\_presenca/fontes.c
#fim do bkp da FONTE.C

#bkp do video
wget [www.lt38c.hturbo.com/Projetos/2015\_2s\_sensor\_presenca/videos/mysql.mp4](http://www.lt38c.hturbo.com/Projetos/2015_2s_sensor_presenca/videos/mysql.mp4) -O /tmp/mysql.mp4
cp /tmp/mysql.mp4 /lt38c/2015\_2s\_sensor\_presenca/videos/mysql.mp4
#fim do kkp do video

#bkp dos maff
wget [www.lt38c.hturbo.com/Projetos/2015\_2s\_sensor\_presenca/maff-mht/mysql.pdf](http://www.lt38c.hturbo.com/Projetos/2015_2s_sensor_presenca/maff-mht/mysql.pdf) -O /tmp/mysql.pdf
cp /tmp/mysql.pdf /lt38c/2015\_2s\_sensor\_presenca/maff-mht/mysql.pdf
#fim do bkp maff

#bkp do BANCO DE DADOS
wget [www.lt38c.hturbo.com/Projetos/2015\_2s\_sensor\_presenca/mysql.sql/projeto.sql](http://www.lt38c.hturbo.com/Projetos/2015_2s_sensor_presenca/mysql.sql/projeto.sql) -O /tmp/projeto.sql
cp /tmp/projeto.sql /lt38c/2015\_2s\_sensor\_presenca/mysql.sql

#Restaurar BANCO DE DADOS
cd /lt38c/2015\_2s\_sensor\_presenca/mysql.sql
mysql -u root -proot -Bse "create database projeto"
cat projeto.sql | mysql -u root -proot projeto

#Configurar DATA
dpkg-reconfigure -f noninteractive tzdata

#Compilação do programa
gcc -o /lt38c/2015\_2s\_sensor\_presenca/fontes.c/codigo1.c -lmysqlclient -o /lt38c/2015\_2s\_sensor\_presenca/exe/exe -l bcm2835