



**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CAMPO MOURÃO**

GERÊNCIA DE PESQUISA E GRADUAÇÃO

DISCIPLINA DE SISTEMAS EMBARCADOS – LT38C
WWW.LT38C.HTURBO.COM

FRANCINE MARTINES
FERNANDO HENRIQUE ORTEGA
VINICIUS JOSÉ ALMEIDA DE SOUZA

**MONITORAMENTO DE SENSOR INFRAVERMELHO DE CURTO
ALCANÇE**

PROJETO FINAL DE DISCIPLINA

CAMPUS MOURÃO, AGOSTO 2014

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
Graduação em Engenharia Eletrônica

TRABALHO DE CONCLUSÃO DE DISCIPLINA
apresentado a UTFPR
para obtenção da nota final

SISTEMAS EMBARCADOS

por

FRANCINE MARTINES
FERNANDO HENRIQUE ORTEGA
VINICIUS JOSÉ ALMEIDA DE SOUZA

**MONITORAMENTO DE SENSOR INFRAVERMELHO DE
CURTO ALCANÇE**

Banca Examinadora:

Presidente:

PROF. MSC. PAULO DENIS GARCEZ DA LUZ

UTFPR

Examinadores:

PROF. DR. XXXX

UTFPR

PROF. DR. YYYY

UTFPR

PROF. DR. ZZZZ

UTFPR

CAMPO MOURÃO, AGOSTO 2014.

FRANCINE MARTINES
FERNANDO HENRIQUE ORTEGA
VINICIUS JOSÉ ALMEIDA DE SOUZA

MONITORAMENTO DE SENSOR INFRAVERMELHO DE CURTO ALCANÇE

Trabalho de conclusão de disciplina apresentada ao Professor de Sistemas Embarcos no curso de Graduação em Engenharia Eletrônica 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, AGOSTO 2014

AGRADECIMENTOS

A realização deste trabalho exigiu mais que nossa vontade e empenho pessoal, sendo de fundamental importância o envolvimento de pessoas e entidades, a quem devemos nossos sinceros agradecimentos.

À Deus em primeiro lugar.

Aos familiares pela confiança e apoio em todos os momentos.

Ao professor Paulo pelos ensinamentos.

À UTFPR por disponibilizar sua estrutura e professores para a capacitação profissional de pesquisa científica.

E por fim, mas não menos importante aos amigos e colegas que dentro e fora da Universidade fizeram dessa caminhada um momento especial, tornando-se assim uma segunda família.

ALUNOS: FRANCINE MARTINES, FERNANDO HENRIQUE ORTEGA, VINICIUS JOSÉ

ALMEIDA DE SOUZA

CAMPO MOURÃO, DEZEMBRO 2015

SUMÁRIO

INTRODUÇÃO	1
1.1 SISTEMAS EMBARCADOS	1
1.2 MOTIVAÇÕES	2
1.3 OBJETIVOS.....	2
1.4 ESTRUTURA DO TRABALHO	2
INSTALAÇÃO DO SISTEMA INICIAL	3
2.1 INTRODUÇÃO.....	3
2.2 DESCRIÇÃO SUSCINTA DAS INSTALAÇÕES	3
2.2.1 PACOTE BCM2835.....	7
2.2.2 PACOTE WIRING.....	8
2.2.3 PACOTE MYSQL.....	8
2.3 AJUSTES FINAIS DE CONFIGURAÇÃO.....	9
2.3.1 CONFIGURAÇÃO DA BIOS.....	9
2.3.2 EDITAR ARQUIVOS DE USUÁRIOS.....	9
2.3.3 TIRANDO USUARIO E SENHA.....	10
2.4 CONCLUSÃO.....	10
PROGRAMAS CONSTRUÍDOS OU ADAPTADOS	11
3.1 INTRODUÇÃO.....	11
3.2 PROGRAMA DESENVOLVIDO.....	11
3.3 ESTRUTURA DO BANCO DE DADOS	15
3.4 BIBLIOTECA TIME.H.....	16
3.5 ESTRUTURA DOS DIRETÓRIOS DO PROJETO	17
3.6 CONCLUSÃO.....	18
DESCRIÇÃO DO HARDWARE ADICIONAL NA GPIO	19
4.1 INTRODUÇÃO.....	19
4.2 SENSOR INFRAVERMELHO DE CURTO ALCANCE	19
4.3 CONEXÃO DO SENSOR NA GPIO DA RASPBERRY PI.....	20
4.4 CONCLUSÃO.....	20

LISTA DE FIGURAS

Figura 1: Interface de download Raspberry Pi.....	3
Figura 2: Programa para gravar a micro SD no Windows	4
Figura 3: Layout Raspberry Pi	4
Figura 4: Tela de configuração Raspberry Pi.....	4
Figura5: Primeira configuração.....	5
Figura 6: Confirmação de configuração.....	5
Figura7: Overscan.....	6
Figura 8: Overscan.....	6
Figura9: Boot	6
Figura 10: Boot	7
Figura 11: Sensor Infravermelho KY-033.....	19
Figura 12: Tabela de conexão do sensor IR com a placa Raspberry PI	20

LISTA DE ABREVIATURAS E SIGLAS

BIOS	- <i>Basic Input/Output System</i>
GPIO	- <i>General Purpose Input/Output</i>
IR	- <i>Infrared</i>
I2C	- <i>Inter-IntegratedCircuit</i>
SD	- <i>Secure Digital</i>
SO	- <i>Sistema Operacional</i>
SPI	- <i>Serial Peripheral Interface</i>
SQL	- <i>Structured Query Language</i>

RESUMO

O presente trabalho apresenta o desenvolvimento de um sistema capaz de monitorar e registrar presença em um sensor infravermelho de curto alcance através do sistema embarcado *RaspberryPi*, sendo o registro armazenado em um banco de dados (MySQL).

CAPÍTULO 1

INTRODUÇÃO

1.1 SISTEMAS EMBARCADOS

As maiorias das funções dos sistemas eletrônicos atuais, em geral, envolvem algum tipo de computação e controle e são realizadas por componentes digitais. Atualmente é uma tendência que sinais analógicos sejam processados como sinais digitais, de forma que componentes para processamento digital são dominantes nos sistemas eletrônicos. Segundo De Micheli existem três classes básicas de sistemas digitais: emulação e sistemas de prototipação, sistemas de computação de propósito geral e sistemas embarcados (embedded systems).

A alma de um sistema embarcado é o seu microcontrolador, que é o processador, a memória, e as interfaces de entrada e saída todas dentro de um mesmo chip, tornando muito mais simples a confecção de um sistema. Poderíamos dizer que praticamente quase todos os equipamentos a eletricidade já possuem (ou possuirão em breve) algum sistema de computação embutido no mesmo. O mercado de sistemas embarcados apresenta-se como um nicho extremamente atrativo, porém bastante crítico com relação a alguns aspectos de projeto tais como: custo e tempo de desenvolvimento, bem como o desempenho do produto final. Sistemas embarcados possuem algumas características que são comuns:

1. Funcionalidade única: usualmente um sistema embarcado executa somente um programa repetidamente;

2. Restrições de projeto mais rígidas: todos os sistemas de computação possuem em geral alguma restrição de projeto a ser satisfeita, no entanto, nos sistemas embarcados, estas restrições são normalmente mais rígidas, por exemplo, o custo de um sistema não pode ser muito alto para não onerar o custo do equipamento, o tempo de resposta deve permitir em várias aplicações processamento em tempo real e devem dissipar pouca potência para permitir uma maior duração da bateria ou não necessitar de um sistema de refrigeração;

3. Sistemas reativos de tempo real: muitos sistemas embarcados devem reagir a mudanças no ambiente e devem fornecer resultados em tempo real. Por exemplo, um piloto automático, ele deve computar a aceleração e desaceleração repetidamente num intervalo de tempo. Caso haja um retardo o controle do carro pode ser perdido.

De maneira geral tornar um sistema embarcado é torná-lo mais flexível, mais funcional devido a sua alta gama de aplicações e inovações.

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.

1.3 OBJETIVOS

O objetivo deste trabalho é desenvolver um software capaz de monitorar e registrar eventos de presença em um sensor infravermelho de curto alcance. Este software tem como finalidade realizar as seguintes ações:

- 1) Monitorar o sensor IR;
- 2) Registrar qualquer ocorrência de eventos de presença, registrando data da ocorrência e momento em que houve ocorrência;
- 3) Confiabilidade dos dados obtidos e armazenados.

1.4 ESTRUTURA DOTRABALHO

No primeiro momento apresentou-se sobre o que é e quais as características dos sistemas embarcados, quais as motivações e objetivos do presente trabalho.

No segundo capítulo ocorrerá a apresentação das instalações iniciais, apresentando o sistema operacional (SO) utilizado, bem como, suas características, funcionalidades e instalação na placa *Raspberry PI* com os pacotes adicionais.

O terceiro capítulo contém os programas nos quais o projeto se baseou para o desenvolvimento, e suas funções acrescentadas.

Por fim, o quarto capítulo irá apresentar a descrição do hardware adicional.

CAPÍTULO 2

INSTALAÇÃO DO SISTEMA INICIAL

2.1 INTRODUÇÃO

Sistemas operacionais são softwares que possuem como objetivo gerenciar e otimizar o funcionamento de um computador, dispositivo que é constituído por vários outros dispositivos que necessitam de informações específicas para o correto funcionamento. Os mais populares sistemas operacionais são: Windows, Linux e Mac OS X.

2.2 DESCRIÇÃO SUSCINTA DAS INSTALAÇÕES

Neste trabalho foi utilizado o sistema operacional Linux, na versão Debian para os sistema embarcado *RaspberryPI* (RASPBIAN), disponível para download no seguinte endereço web:

- www.raspberrypi.org/downloads/

Ao se acessar o site pode-se realizar o download na seguinte botão apresentado na figura 1, que será instalado na memória SD.



Figura 1: Interface de download Raspberry PI

Quando realizado o download do sistema operacional é necessário inseri-lo em um cartão de memória SD, sendo viável utilizar um cartão de memória de no mínimo 4 GB de memória ou superior. Para realizar este tipo de tarefa pode ser utilizado um programa chamado Win32DiskImager, encontrado do seguinte endereço de Web:

- sourceforge.net/projects/win32diskimager/



Figura 2: Programa para gravar a microSD no Windows

Após o sistema operacional instalado, já se pode ligar a placa *RaspberryPI*. A placa oferece a conexão de vários periféricos, como visto na figura 3.

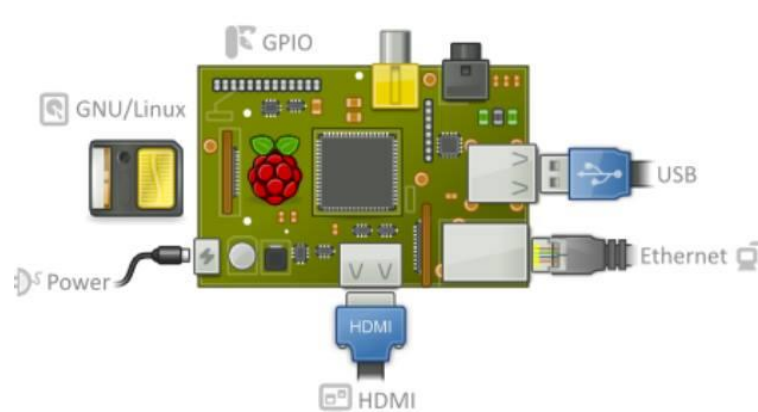


Figura 3: Layout RaspberryPi

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 como visto na figura 4.

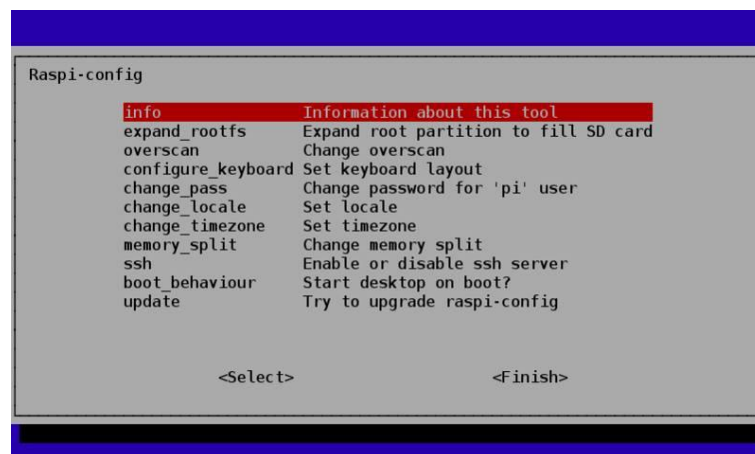


Figura 4: Tela de configuração RaspberryPi

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

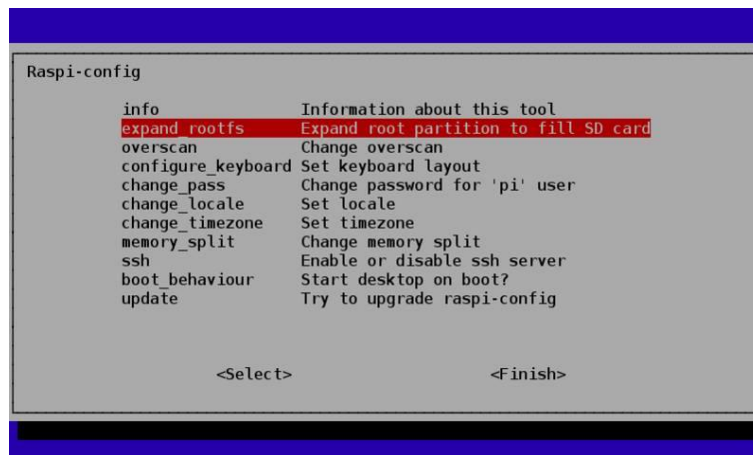


Figura5: Primeira configuração

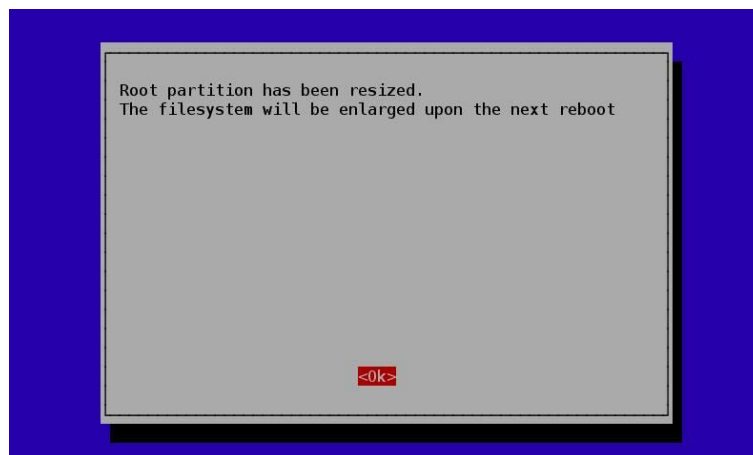


Figura 6: Confirmação de configuração.

Em primeiro momento o software não irá utilizar toda a tela do monitor, para trabalhar com a tela cheia (em overscan) siga os seguintes passos:

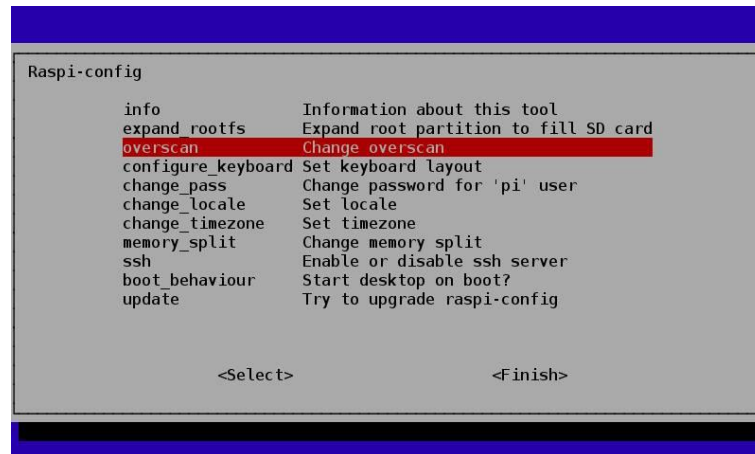


Figura7: Overscan.

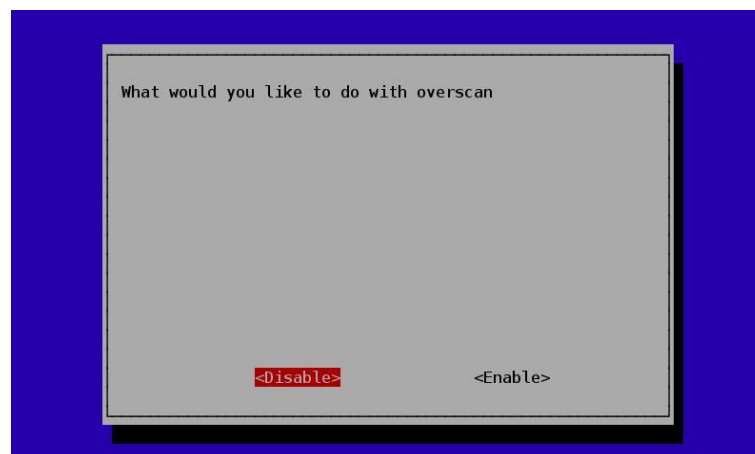


Figura 8: Overscan.

Existem diversos tipos de configurações desejadas, para maiores informações sobre a configuração do sistema, como hora/data do sistema, idioma, acesse o seguinte link:

- www.youtube.com/watch?v=M3sLM_vV20g

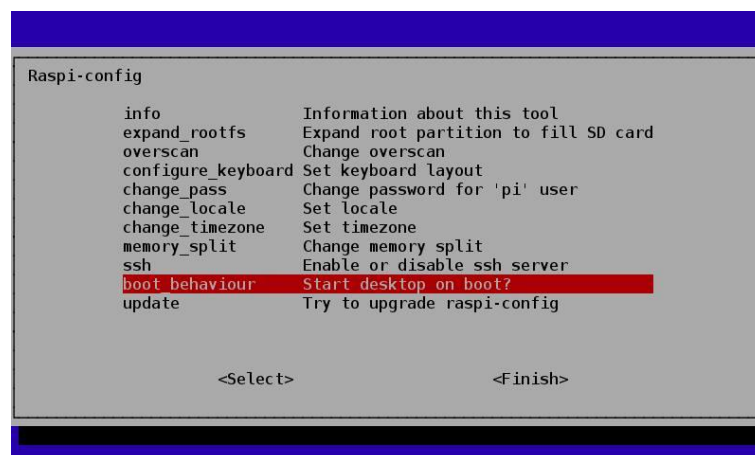


Figura9: Boot

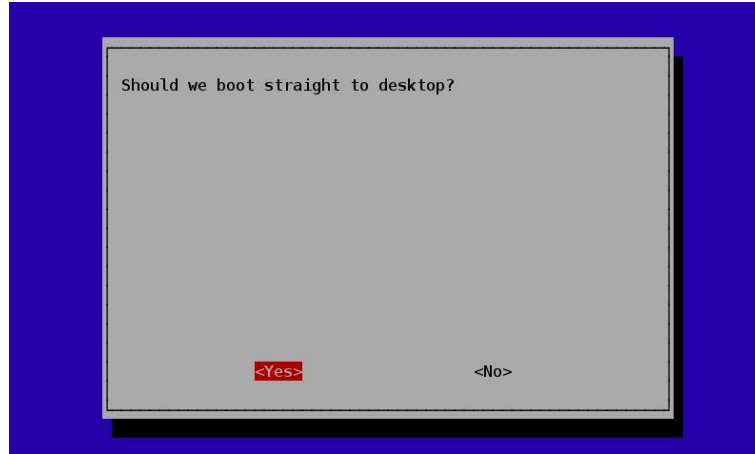


Figura 10: Boot

Visto todos estes passos, e realizados, o sistema operacional está instalado e pronto para o uso.

A plataforma Linux apresenta algumas particularidades quando comparada com outras plataformas. Esta plataforma é conduzida através de um terminal de comandos, sem interface gráfica. Para utilizar este terminal de comando é necessário o conhecimento dos comandos principais, disponíveis no apêndice 1.

Inicialmente, para a atualização do sistema realiza-se os seguintes comandos:

- Digitar: *sudo* no terminal para ter acesso como administrador, e então:

- I. *apt-get update* → Inicializa a atualização do sistema.
- II. *apt-get install mc* → Inicializa a instalação do editor de texto mcedit.
- III. *apt-get upgrade* → Atualiza o Sistema da *Raspberry PI*.

Enfim, pode-se considerar o sistema operacional instalado e disponível para o uso.

Iniciamos a instalação do micro-sd. A seguir serão instalados os seguintes pacotes e serviços adicionais: Pacote BCM2835 e Pacote Wiring.

2.2.1 PACOTE BCM2835

Para a utilização dos periféricos da GPIO *RaspberryPi* visto na figura 3, e outras funções do chip da Broadcom bcm2835, é necessário a instalação de biblioteca para a habilitar as portas de entrada e saída. Para instalar a biblioteca é necessário os seguintes comandos:

- I. *wget* www.lit38c.hturbo.com/bcm2835-1.45.tar.gz -o /tmp/bcm2835.tar.gz

- II. `tar -zxvf bcm2835.tar.gz`
- III. `cd bcm2835-1.45`
- IV. `./configure`
- V. `make`
- VI. `sudomakecheck`
- VII. `sudomakeinstall`

Tela verde significa que o pacote foi instalado com sucesso.

2.2.2 PACOTE WIRING

Para a utilização dos periféricos da GPIO *RaspberryPi* através do pacote Wiring é necessário a instalação da biblioteca através dos seguintes comandos:

- I. `cd /tmp`
- II. `sudo apt-get install git-core`
- III. `wget www.lt38c.hturbo.com/wiringPi-5edd177.tar.gz`
- IV. `tar xvfz wiringPi-5edd177`
- V. `./build`
- VI. `Gpio -v`
- VII. `Gpioreadall`

2.2.3 PACOTE MYSQL

MySQL é um banco de dados para servidores do tipo *LINGUAGEM DE PESQUISA SIMPLES (SQL)*, este tipo de software é muito utilizado devido a sua eficácia. Sendo desenvolvido pela *T.c.X.DataKonsultAB*, o software apresenta a vantagem de ser gratuito, tanto para o uso comercial, quanto para o uso particular. Prova que se seu uso pode trazer grandes benefícios temos empresas renomadas que fazem sua utilização, como Yahoo, IFX Networks, SiliconGrafics.

Para a instalação do pacote MySQL na *RaspberryPI* temos que executar os seguintes comandos:

- I. `apt-getinstallmysql-server`

O próximo comando é necessário para que se possa garantir o acesso do “root” ao *MySQL*:

- II. `mysql -u root -p`

Aparecerá a seguinte mensagem, confirmando o acesso do root:

III. GRANT ALL ON *.* TO 'root'@'localhost' IDENTIFIED BY 'raspberry';

Para a ativação na linguagem C, é necessário compilar e executar o seguinte arquivo:

- 1) `sudo apt-get install libmysql-dev`
- 2) `mceditmyClient.c`
- 3) `g++ myClient.c -o cliente`
- 4) `./cliente`

2.3 AJUSTES FINAIS DE CONFIGURAÇÃO

2.3.1 CONFIGURAÇÃO DA BIOS

Para a configuração da BIOS da GPIO *RaspberryPi* utilizou-se os seguintes comandos:

- I. `cd /`
- II. `ls -l`
- III. `cd boot`
- IV. `ls -l`

Entra no documento *config.txt* apague o comentário da sexta linha de cima para baixo.

Dê F2 para salvar e F10 para sair e dar reboot.

- I. `sudo su`
- II. `cd /boot`
- III. `ls -l`

Entre no documento *start.lmx*

- I. `cmd line.txt`
- II. `config.txt`

2.3.2 EDITAR ARQUIVOS DE USUÁRIOS

Como administrador, digite:

- I. `cd /root`
- II. `ls -la`
- III. `.bashrc`
- IV. `mcedit . bashrc`

- V. `aliasll='ls -l --color=auto'` → Insere o atalho do comando `ll`.
- VI. `aliasla='ls -la --color=auto'` → Insere o atalho do comando `la`.
- VII. `export EDITOR=mcedit` → Torna o `mcedit` o editor de texto padrão.
- VIII. `export HISTSIZE=1000` → Expande o tamanho do histórico de comandos.
- IX. `export HISTFILESIZE=1000`

Dê F2 para salvar e F10 para sair e dar reboot. Logar como usuário PI

- I. `cd /etc`
- II. `mcedit profile`

Copiar novamente o trecho de V a IX, dar F2 para salvar e F10 para sair e dar reboot com `shutdown -r now`.

2.3.3 TIRANDO USUARIO E SENHA

Sempre que inicializado ou reinicializado, a *RaspberryPI* solicita um login e password. Para desabilitar a necessidade de login execute:

- I. `Sudomcedit /etc /inittab`

Dentro do arquivo, comentar a linha:

`l:2345:respawn:/sbin/getty 38400 tty1 </dev/tty1 >/dev/tty1 2>&l`

Insira abaixo o seguinte comando:

`l:2345:respawn:/bin/login -f pi tty1 </dev/tty1 >/dev/tty1 2>&l`

E em seguida, reinicie o Sistema para salvar as alterações.

2.4 CONCLUSÃO

Neste capítulo foi apresentada de forma sucinta a estrutura da instalação do sistema inicial, obtendo os seguintes resultados: o sistema entra sem pedir senha, tem a tela redimensionada para uso de 100% da área, tem a GPIO liberada para acesso via programação C, tem o I2C configurado, tem o SPI configurado, tem os periféricos habilitados e tem um servidor MySQL rodando.

CAPÍTULO 3

PROGRAMAS CONSTRUÍDOS OU ADAPTADOS

3.1 INTRODUÇÃO

O software desenvolvido se baseou em duas etapas, sendo estas o monitoramento e captura dos eventos de presença frente ao sensor IR, e o armazenamento das informações obtidas destes eventos no sensor. No projeto em questão foram utilizados os seguintes recursos de programação: linguagem C e *MySQL*.

3.2 PROGRAMADESENVOLVIDO

O primeiro passo tomado foi entender o funcionamento dos pinos de entrada e saída da *RaspberryPi* e como ativá-los como pinos de leitura e de envios de sinal através da linguagem de programação C. As informações necessárias para este entendimento estão disponíveis no seguinte link:

- projects.drogon.net/raspberry-pi/wiringpi/the-gpio-utility/

Após o entendimento do funcionamento dos pinos de entrada e saída da *RaspberryPi*, foi analisado o funcionamento e formas de operação do sensor IR, e a partir das informações coletadas nos seguinte link a seguir, foi desenvolvido o código em linguagem C para a varredura e leitura do teclado matricial:

- www.embarcados.com.br/raspsberrypi-teclado-4x4-com-python/

Então foi desenvolvido no programa:

- Inserção das seguintes bibliotecas:

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

```
# include <unistd.h>
```

```
# include <mysql/mysql.h>
```

```
# include <sys/time.h>
```

```
# include <wiringPi.h>
```

- Definição das variáveis.

```
//Variáveis para DATA e HORA
```

```
time_t timer;
```

```
struct tm y2k = {0};
```

```

double seconds;
double min, hora, dia;
intmesAtual, dias, diaAtual;
inthoraInt, minInt, diaInt;
inthoraAtual, minAtual, segundoAtual;
int i = 0;
intano = 2000;
intdiasMes[12] = {31,28,31,30,31,30,31,31,30,31,30,31};

y2k.tm_hour = 0; y2k.tm_min = 0; y2k.tm_sec = 0;
y2k.tm_year = 100; y2k.tm_mon = 0; y2k.tm_mday = 1;
//Final das variáveis para DATA e HORA

//Variáveis para BANCO DE DADOS
MYSQL conexao;
MYSQL_RES *resp;
MYSQL_ROW linhas;
MYSQL_FIELD *campos;

int res;
char query[120];
charqueryConsult[] = "SELECT * FROM tabela";
int sensor, mes, seg;
int count;
//Final das variáveis para BANCO DE DADOS

//Variáveis para LER o SENSOR
wiringPiSetup();
pinMode(7,INPUT);
//Final das variáveis para LER o SENSOR

```

- Rotina de conexão ao banco de dados:

```

mysql_init(&conexao);

    if(mysql_real_connect(&conexao,"localhost","guest","guest","sensorIR",0,NULL,0)
){ //conecta e verifica conexao

printf("Conectado com sucesso!\n");

}

```

- Rotina de leitura do sensor:

```
sensor = digitalRead(7);
```

- Rotina de identificação do estado do sensor, data e o momento em que houve o evento no sensor:

```
//Código que calcula DATA E HORA
```

```

y2k.tm_hour = 0; y2k.tm_min = 0; y2k.tm_sec = 0;
y2k.tm_year = 100; y2k.tm_mon = 0; y2k.tm_mday = 1;

time(&timer); /* get current time; same as: timer = time(NULL) */

ano = 2000;

seconds = difftime(timer,mktime(&y2k));

//printf ("\n\n%.f seconds since January 1, 2000 in the current timezone\n\n", seconds);

min = seconds/60;
hora = min/60;
dia = hora/24;
dias = dia;

while(dias >= 365){
    if ( ( ano % 4 == 0 && ano % 100 != 0 ) || ano % 400 == 0 ){
        ano++;
        dias = dias - 366;
    }
    else{
        ano++;
        dias = dias - 365;
    }
}

if ( ( ano % 4 == 0 && ano % 100 != 0 ) || ano % 400 == 0 )
    diasMes[1] = 29;
else
    diasMes[1] = 28;

diaAtual = dias;
mesAtual = 0;

while(diaAtual > 31){
    diaAtual = diaAtual - diasMes[i];
    mesAtual++;
    i++;
}

i = 0;

horaInt = min/60;
minInt = seconds/60;
diaInt = hora/24;

horaAtual = hora - (diaInt * 24);
minAtual = min - (horaInt * 60);

```

```
segundoAtual = seconds - (minInt * 60);
```

```
//Fim do código que calcula DATA E HORA
```

- Rotina de consulta:

```
//Consulta a tabela
```

```
if(mysql_query(&conexao,queryConsult))
    printf("Erro:%s\n", mysql_error(&conexao));

else{

    resp = mysql_store_result(&conexao);

    if(resp){
        campos = mysql_fetch_fields(resp);
        for(count=0;count<mysql_num_fields(resp);count++){
            printf("%s",campos[count].name);
            if(mysql_num_fields(resp)>1)
                printf("\t");
        }

        printf("\n");

        while((linhas=mysql_fetch_row(resp)) != NULL){

            for(count=0;count<mysql_num_fields(resp);count++)
                printf("%s\t",linhas[count]);
            printf("\n");

        }

        mysql_free_result(resp);

        while(sensor == 0)
            sensor = digitalRead(7);
        while(sensor == 1)
            sensor = digitalRead(7);
            sleep(1);

        }
    }
    else
        sensor = digitalRead(7);
    }

    mysql_close(&conexao);
```

```

}
else{

    printf("Falha de conexao!\n");
    printf("Erro %d : %s\n",mysql_errno(&conexao),mysql_error(&conexao));

}

}

```

3.3 ESTRUTURA DO BANCO DE DADOS

A linguagem SQL é 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 *PostgreSQL*, 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).

Para o armazenamento das informações obtidas do teclado foi desenvolvido um banco de dados do tipo MySQL. O link a seguir disponibiliza informações básicas, porém de suma importância ao se entrar em contato com este tipo de banco de dados em primeiro momento.

- www.vivaolinux.com.br/artigo/Usando-MySQL-na-linguagem-C?pagina=1

Visto as informações contidas no link é possível criar um banco de dados. Foram listados os comando utilizados para a manipulação do banco de dados.

Para entrar no MySQL utiliza-se o seguintes comandos:

1. *mysql -user=root -p*
2. *Enterpassword: <senha aqui>*

Para criar um banco de dados utiliza-se o seguinte comando:

```

CREATE TABLE `estado` (
  `ID` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `TECLA` int(2) DEFAULT NULL, `ESTADO` char(3) DEFAULT NULL,

```



```

`DATA` varchar(24) DEFAULT NULL,
`ms` int(3) DEFAULT NULL,
PRIMARY KEY (`ID`)
) ENGINE=InnoDB AUTO_INCREMENT=1569 DEFAULT
CHARSET=latin1;

```

Aparecerá a seguinte mensagem na tela confirmando a operação:

- grant all on *nomedobancodedados.** to guest identified by “*senhadobancodedados*”;

Em seguida digite o commando para garantir total privilegio do usuário root no banco de dados:

3. *flushprivileges;*
4. *exit*

Obs.: É importante que se salve o nome do banco de dados e a senha do mesmo.

O comando para sair do MySQL é:

5. *exit*

Agora, tendo o seu banco de dados, é possível criar tabelas nele. Para isso, é necessário logar no seu banco de dados, e conectá-lo através de:

6. *connectnomedobancodedados;*

Criando tabela:

7. *createtablenomedatabela (nomedacoluna VARCHAR(40) NOT NULL, nomedacoluna2 VARCHAR(40) NOT NULL, nomedacoluna VARCHAR(60) NOT NULL) ENGINE=CSV;*

**VARCHAR é o tipo de variável (indicado para economia de espaço), NOT NULL que não poderá ser vazio.*

Para mostrar a tabela:

8. *select* fromnomedatabela;*

3.4 BIBLIOTECA TIME.H

Para que o código fosse desenvolvido de maneira mais viável foi utilizado a biblioteca *time.h*, que oferece ao usuário personalizar o software com o intuito de extrair a data e o momento em que ocorreu o evento solicitado pelo programador, que é identificada pela seguinte rotina no código:

```

consttime_t [nome da variável] = time(NULL);
gettimeofday (&on, NULL);
ctime(&[nome da variável]);

```

3.5 ESTRUTURA DOS DIRETÓRIOS DO PROJETO

O projeto do Sensor IR foi estruturado e organizado em diversos diretórios, visando documentar de forma organizada e objetiva todas as informações necessárias para a execução do mesmo. O diretório “arquivos.sys” contém os arquivos de instalação e personalização do sistema para a placa *RaspberryPi*. O diretório “apt” contém os arquivos de aplicativos utilizados. O diretório “exe” contém o arquivo executável do projeto. O diretório “fontes.c” contém os códigos do programa projetado. O diretório “history” contém todos os comandos utilizados no decorrer do trabalho. O diretório “libs” contém os arquivos das bibliotecas instaladas. O diretório “maff-mht” contém os arquivos em formato maff-mht de todos os sites onde houve consulta. A pasta “mysql.sql” contém os arquivos referentes ao uso de mysql utilizados no desenvolvimento do banco de dados. O diretório “site” contém os sites que serviram de base na pesquisa deste trabalho. O diretório “vídeos” contém os arquivos vídeos e tutoriais consultados.

```

/lt38c
/lt38c/2015_2s_SensorIR
/lt38c/2015_2s_SensorIR/apt software e pacotes
/lt38c/2015_2s_SensorIR/arquivos.sys
/lt38c/2015_2s_SensorIR/exe
/lt38c/2015_2s_SensorIR/fontes.c
/lt38c/2015_2s_SensorIR/history
/lt38c/2015_2s_SensorIR/libs
/lt38c/2015_2s_SensorIR/maff-mht
/lt38c/2015_2s_SensorIR/mysql.sql
/lt38c/2015_2s_SensorIR/site
/lt38c/2015_2s_SensorIR/videos

```

3.6 CONCLUSÃO

A estrutura proposta é uma das muitas possibilidades existentes para conseguir o monitoramento de dados em tempo real, tendo sido desenvolvida com base em decisões e análises feitas de forma a se obter um sistema de monitoramento em tempo real.

CAPÍTULO 4

DESCRIÇÃO DO HARDWARE ADICIONAL NA GPIO

4.1 INTRODUÇÃO

Para este projeto foi necessário acoplar na GPIO um sensor infravermelho de curto alcance KY-033.

4.2 SENSOR INFRAVERMELHO DE CURTO ALCANCE

O sensor utilizado tem como características principais:

- 1) Tensão de funcionamento: 2.5V - 12V (não podendo ultrapassar 12V).
- 2) Corrente de trabalho: 18mA - 20 mA em 5V.
- 3) Saída de sinal de nível lógico: nível lógico baixo quando há detecção de objetos/ nível lógico alto quando não há detecção de objetos.

Existem sensores de infravermelho ativos e passivos. Um sensor de infravermelho ativo é composto por um emissor de luz infravermelha e um receptor que reage a essa luz.

Pode-se dizer que o sensor de infravermelho, de fato, é o receptor, um componente de três pinos que mede a frequência da luz.

O sensor dispõe de três terminais elétricos sendo estes Vcc, GND e Sinal (ou Vout, como na imagem).



Figura 11: Sensor Infravermelho KY-033

O terminal de sinal (Vout) nos fornece uma corrente elétrica de duração de meio período da luz emitida, o que é um tempo muito pequeno, o que dificulta um pouco sua utilização.

4.3 CONEXÃO DO SENSOR NA GPIO DA RASPBERRY PI

- Conectar o Vcc no pino indicado pelo Header 1;
- Conectar o GND no pino indicado pelo Header 6;
- Conectar o pino de sinal no pino indicado pelo Header 7;

wiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	wiringPi Pin
—	—	3.3v	1 2	5v	—	—
8	R1:0/R2:2	SDA	3 4	5v	—	—
9	R1:1/R2:3	SCL	5 6	0v	—	—
7	4	GPIO7	7 8	TxD	14	15
—	—	0v	9 10	RxD	15	16
0	17	GPIO0	11 12	GPIO1	18	1
2	R1:21/R2:27	GPIO2	13 14	0v	—	—
3	22	GPIO3	15 16	GPIO4	23	4
—	—	3.3v	17 18	GPIO5	24	5
12	10	MOSI	19 20	0v	—	—
13	9	MISO	21 22	GPIO6	25	6
14	11	SCLK	23 24	CE0	8	10
—	—	0v	25 26	CE1	7	11
wiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	wiringPi Pin

Figura 12: Tabela de conexão do sensor IR com a placa Raspberry PI

4.4 CONCLUSÃO

A estrutura de hardware proposta é uma das muitas possibilidades existentes para conseguir o monitoramento de dados em tempo real sobre a existência ou não de obstáculo em curto alcance. Com o projeto obtivemos em tempo real uma tabela indicando quais dias,

meses, anos, horas, minutos e segundos foram registrados acontecimentos de existência de obstáculo em curto alcance através do sensor infravermelho.

REFERÊNCIAS BIBLIOGRÁFICAS

AGILENT TECHNOLOGIES, *Practical Temperature Measurements*. Manual Técnico. Application Note 290, 2000.

ALBUQUERQUE, L. R. *Segurança em Redes ad Hoc*. Programa de Engenharia de Sistema e Computação – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2003.

ANDRADE, M. B.; COLLI, R. *Redes ad hoc*. In: *Comunicações Digitais e Tópicos Relacionados*, vol. 1, Departamento de Engenharia – Instituto de Educação Superior de Brasília, agosto de 2003.

BENEDICT, R.P. *Fundamentals of Temperature, Pressure and Flow Measurements*. Third Edition. New York: John Wiley & Sons, Inc., 1984.

SOARES, W. *AJAX: Asynchronous JavaScript and XML Guia Prático*. São Paulo: Erica, 2006.

TANENBAUM, A. S. **Redes de Computadores**. Rio de Janeiro: Campus, 1997.

WANG, J.Y.; FELTON, C.M.M. *Instruments for physical environmental measurements*. 2nd ed. Dubuque: Kendall/Hunt Publishing Company, 1983.

WEBSTER, J.G. *Medical Instrumentation – Application and Design*. Third Edition. New York John Wiley: & Sons, Inc., cap.2, p.61-85, 1998.

YI, S.; NALDURG, P.; KRAVETS, R. *Security-Aware Ad-Hoc Routing for Wireless Networks*. Agosto, 2001.

APÊNDICE 1

PRINCIPAIS COMANDOS DO LINUX

Comando	Descrição
sudo	Permite um usuário executar comandos com as permissões de um “super-usuário” (root).
sudohalt	Encerra o sistema.
sudo reboot	Renicializa o sistema.
sudo shutdown -h now	Encerra o sistema imediatamente.
“comando”-help	Mostra o arquivo de ajuda do comando digitado.
ls	Lista arquivos e diretórios da pasta que se está.
clear	Limpa a tela.
cd “nomedodiretório”	Entra no diretório requerido.
cd	Entra direto no diretório raiz do usuário.
pwd	Mostra o diretório em que está.
Ifconfig	Para as configurações das interfaces da rede.
startx	Inicializa a interface gráfica.
mkdir	Cria um diretório.
rmdir	Deleta um diretório.
rm	Deleta um arquivo.
find -name “nomedoarquivo”	Pesquisa o arquivo desejado.