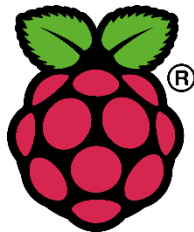


# SISTEMAS EMBARCADOS

A faint, semi-transparent image of a Raspberry Pi circuit board is visible in the background, showing various components like the processor, memory, and connectors.

## Tutorial Raspberry Pi Monitoramento de Teclado Matricial 4x4

*Evandro Pozzatti*

*Heitor José Gonçalves*

*Wendell C. Ferruccio dos Santos*

CAMPO MOURÃO  
2015

EVANDRO POZZATTI  
HEITOR JOSÉ GONÇALVES  
WENDELL C. FERRUCCIO DOS SANTOS

**Tutorial Raspberry Pi**  
**MONITORAMENTO DE TECLADO MATRICIAL 4X4**

Projeto apresentado à disciplina de Sistemas Embarcados, do Curso Bacharelado em Engenharia Eletrônica do Departamento Acadêmico de Engenharia Eletrônica – DAELN - da Universidade Tecnológica do Paraná – UTFPR, como requisito parcial para obtenção da aprovação da disciplina de Sistemas Embarcados.

Professor: Paulo Dênis Garcez da Luz

CAMPO MOURÃO  
2015

## Sumário

<b>1</b>	<b>RESUMO .....</b>	<b>6</b>
<b>2</b>	<b>INTRODUÇÃO .....</b>	<b>7</b>
2.1	SISTEMAS EMBARCADOS.....	7
2.2	MOTIVAÇÕES.....	7
2.3	OBJETIVOS.....	7
2.4	ESTRUTURA DO TRABALHO .....	8
<b>3</b>	<b>INSTALAÇÃO DO SISTEMA OPERACIONAL.....</b>	<b>8</b>
3.1	INTRODUÇÃO.....	8
3.2	DESCRIÇÃO SUSCINTA DAS INTALAÇÕES .....	8
3.3	PACOTE BCM2835.....	16
3.4	PACOTE MYSQL .....	17
<b>4</b>	<b>PROGRAMAS CONSTRUÍDOS OU ADAPTADOS.....</b>	<b>18</b>
4.1	INTRODUÇÃO.....	18
4.2	PROGRAMA DESENVOLVIDO .....	18
4.2.1	MONITORAMENTO E REGISTRO DE EVENTOS DO TECLADO MATRICIAL .....	18
4.3	MYSQL.....	21
4.4	ASPECTOS OPERACIONAIS .....	22
4.4.1	BIBLIOTECA TIME.H.....	22
4.5	ESTRUTURA DA TABELA DO BANCO DE DADOS .....	23
4.6	CONCLUSÃO .....	23
<b>5</b>	<b>DESCRIÇÃO DO HARDWARE ADICIONAL NA GPIO.....</b>	<b>24</b>
5.1	INTRODUÇÃO.....	24
5.2	TECLADO MATRICIAL 4x4 .....	24
5.3	INSTALAÇÃO DO TECLADO MATRICIAL NOS PINOS DE ENTRADA E SAÍDA DO PROJETO .....	25
<b>6</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>27</b>
	<b>APÊNDICE .....</b>	<b>28</b>

## Lista de Figuras

Figura 1 - Interface de download Raspberry PI .....	9
Figura 2 - Campo de Download Disk Imager.....	9
Figura 3 - Layout Raspberry Pi.....	10
Figura 4- Tela de configuração Raspberry Pi.....	10
Figura 5 - Primeira Configuração .....	11
Figura 6 - Confirmação de configuração. ....	11
Figura 7 - Overscan.....	12
Figura 8 - Overscan.....	12
Figura 9 - Boot.....	13
Figura 10- Boot.....	13
Figura 11 - Layout da tabela do Banco de Dados. ....	23
Figura 12 Teclado Matricial. ....	24
Figura 13 - Identificação dos Pinos da RaspberriPi.....	25
Figura 15 - Foto dos pinos gpio da RaspberriPi.....	25
Figura 14 - Identificação do Pinos gpio da Raspberry Pi.....	25
Figura 16 - esquemático de ligação teclado matricial, abreviações em 'L' são linha, em 'C' são colunas. ....	26
Figura 17- Teclado Matricial 4x4 .....	26

## Lista de Tabela

Tabela 1- Ligações Correspondentes entre pinos de gpio e teclado matricial. ....	26
--	----

## LISTA DE ABREVIATURAS E SIGLAS

SO	- <i>Sistema Operacional</i>
GPIO	- General Purpose Input/Output
SD	- <i>Secure Digital Card</i>
SQL	- <i>Structured Query Language</i>

# 1 RESUMO

O trabalho apresentado tem como objetivo demonstrar o passo a passo do projeto desenvolvido na disciplina de Sistemas Embarcados. O projeto consiste em um software capaz de monitorar e registrar as ações das teclas de um teclado matricial 4x4 através do sistema embarcado *Raspberry Pi*, onde o registro é armazenado em um banco de dados (MySQL)

## 2 INTRODUÇÃO

### 2.1 SISTEMAS EMBARCADOS

É considerado um sistema embarcado quando um computador (que é um dispositivo microprocessado) é completamente integrado ou dedicado somente a um dispositivo, ou sistema que ele controla. Este sistema é considerado independente e capaz de realizar um determinado objetivo em sua estrutura funcional.

Seu surgimento se deu por volta da década de 60 através da corrida espacial de americanos e russos, tendo notável no projeto Apollo, que foi desenvolvido por Charles Stark Draper. Sua função era orientar o módulo lunar. Porém o primeiro sistema embarcado que alcançou a produção em massa quando teve seu uso destinado ao míssil nuclear LGM-30 *Míssil Minuteman no ano de 1960, é interessante lembrar que no ano de 1966 quando foi desenvolvida a segunda versão deste míssil, através da tecnologia agregada a este foi possível tornar alguns componentes eletrônicos comerciáveis, como a porta lógica NAND que passou de 1000 dólares para 3 dólares. Já em 1980 com a eletrônica digital tomando o espaço da eletrônica a analógica, sistemas microcontrolados e sistemas embarcados puderam enfim se consolidar no mercado civil.*

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

### 2.2 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.

### 2.3 OBJETIVOS

O objetivo deste trabalho é desenvolver um software capaz de monitorar e registrar eventos em teclado matricial 4x4. Este software tem como finalidade realizar as seguintes ações:

- 1) Monitorar teclas de um teclado matricial 4x4.

2) Registrar qualquer ocorrência de eventos nas teclas, registrando data da ocorrência e momento em que houve ocorrência, além do tipo de ocorrência(ON/OFF) em um banco de dados;

3) Confiabilidade dos dados obtidos e armazenados;

## 2.4 ESTRUTURA DO TRABALHO

Em primeiro momento irá se discorrer sobre os sistema operacional (S0) utilizado, apresentando suas características, funcionalidades, instalação na placa Raspberry PI com os pacotes adicionais. No segundo momento irá se apresentar os programas nos quais o projeto se baseou para o desenvolvimento, e suas funções acrescidas, seguido da descrição do hardware adicional, o teclado matricial 4x4.

# 3 INSTALAÇÃO DO SISTEMA OPERACIONAL

## 3.1 INTRODUÇÃO

Sistemas operacionais são softwares que possuem como objetivo 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.

## 3.2 DESCRIÇÃO SUSCINTA DAS INTALAÇÕES

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

- [www.raspberrypi.org/downloads/](http://www.raspberrypi.org/downloads/)

Para a realização deste trabalho utilizou-se o sistema operacional *Linux*, versão *Debian* para o sistema embarcado *Raspberry PI* (RASPBIAN), disponível para download no seguinte endereço web:

- [www.raspberrypi.org/downloads/](http://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 no cartão de memória SD, é 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/](http://sourceforge.net/projects/win32diskimager/)



Figura 2 - Campo de Download Disk Imager

Para saber como utilizar o programa, pode-se acessar o link abaixo:

- [www.djputop.pt/2013/09/tutorial-raspberry-pi-instalacao-de.html#.UK8nWMna2I](http://www.djputop.pt/2013/09/tutorial-raspberry-pi-instalacao-de.html#.UK8nWMna2I)

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 3.

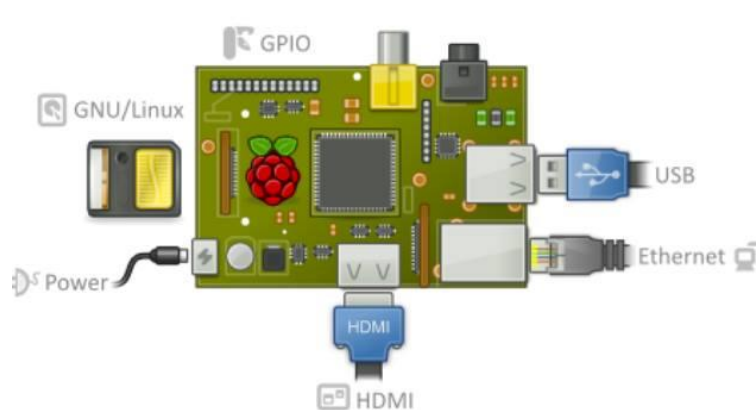


Figura 3 - Layout Raspberry Pi

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 como visto na figura 4.



Figura 4- Tela de configuração Raspberry Pi

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”.

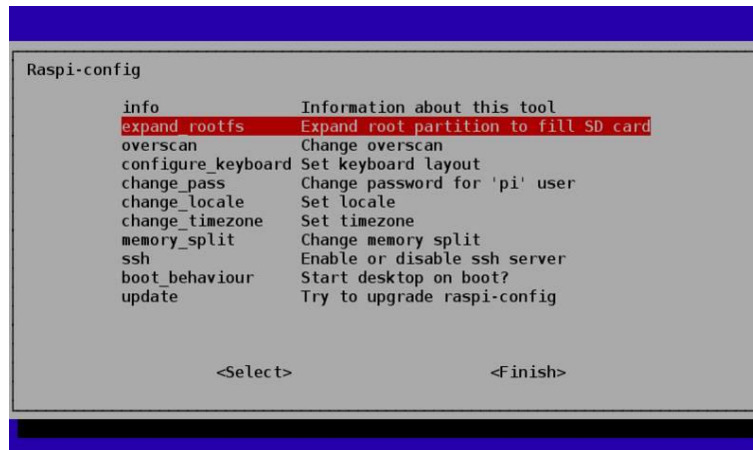


Figura 5 - Primeira Configuração

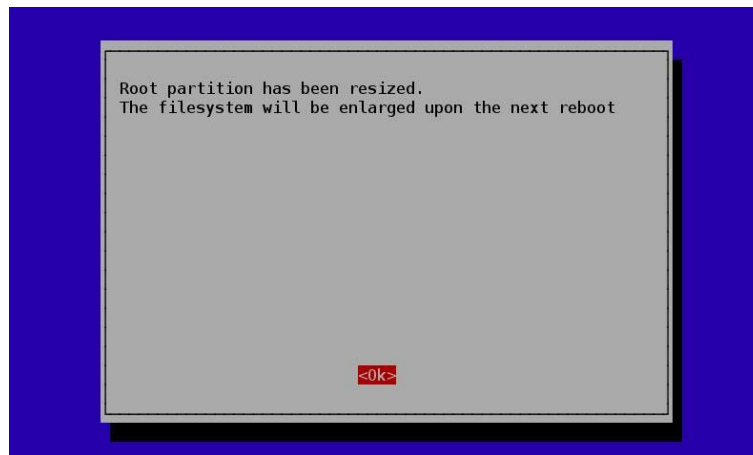


Figura 6 - Confirmação de configuração.

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:

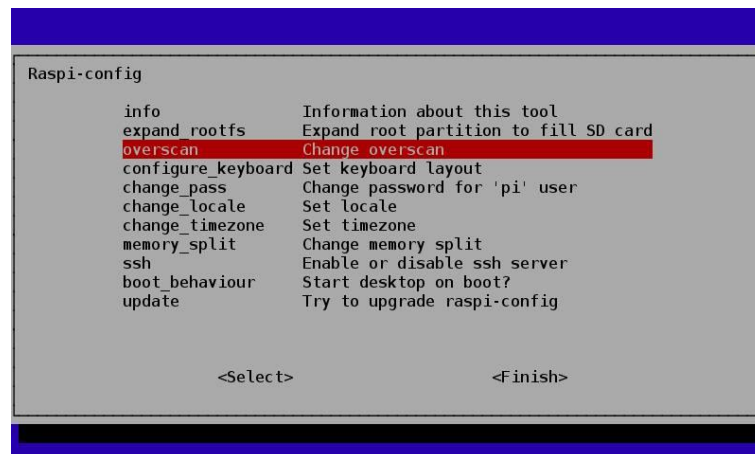


Figura 7 - Overscan.

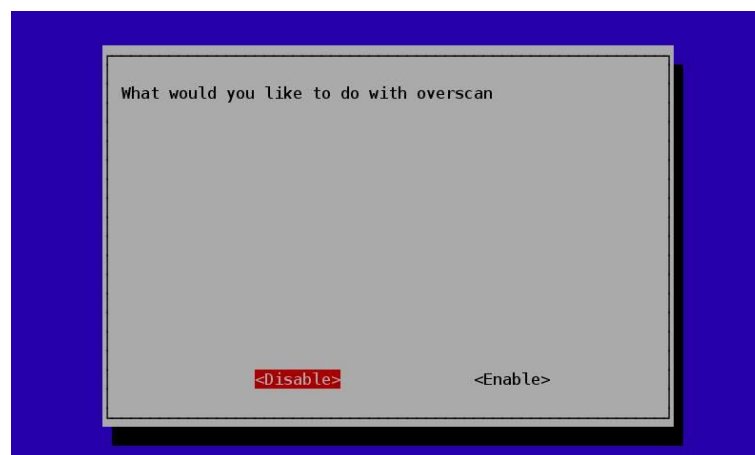


Figura 8 - Overscan.

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](http://www.youtube.com/watch?v=M3sLM_vV20g)

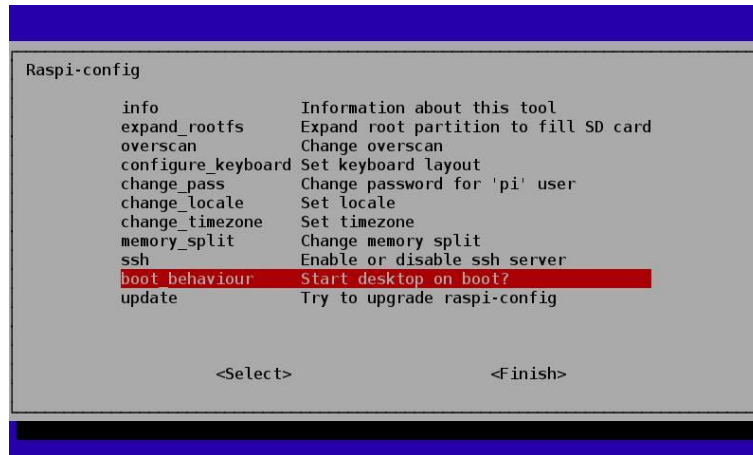


Figura 9 - Boot

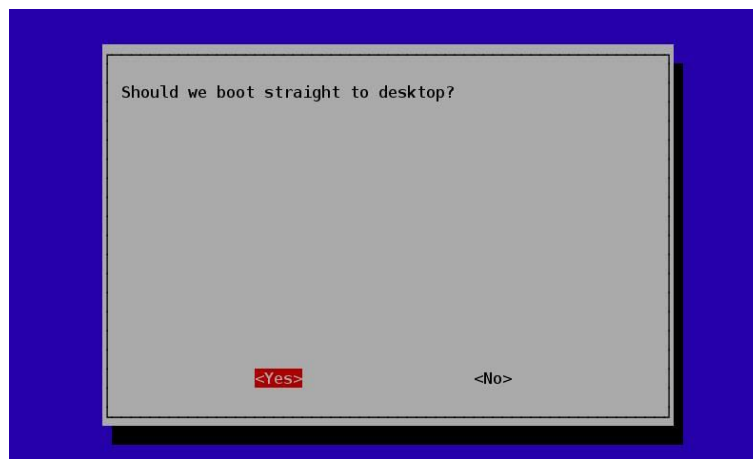


Figura 10- Boot

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 su** 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.

Configurando os atalhos do Profile Pi:

Configurando os atalhos do profile Pi:

- Como administrador, digite:

- I. `cd /root/etc` → Para entrar no diretório etc.
- II. `mcedit profile` → Através do mcedit, o arquivo profile será aberto.  
→ Assim, insira os comandos abaixo no final do arquivo.
- III. `alias ll='ls -l --color=auto'` → Insere o atalho do comando ll.
- IV. `alias la='ls -la --color=auto'` → Insere o atalho do comando la.
- V. `export EDITOR=mcedit` → Torna o mcedit o editor de texto padrão.
- VI. `export HISTSIZE=2000` → Expande o tamanho do histórico de comandos.
- VII. `export HISTFILESIZE=2000`

- Os comandos ll e la, são utilizados para listar os arquivos do diretório atual.

Configurações da raiz (root):

- I. `cd /root` → Para entrar no diretório root.
- II. `mcedit .bashrc` → Abre o arquivo .bashrc no mcedit.  
→ Assim, insira os comandos abaixo no final do arquivo.
- III. `alias ll='ls -l --color=auto'` → Insere o atalho do comando ll.
- IV. `alias la='ls -la --color=auto'` → Insere o atalho do comando la.
- V. `export EDITOR=mcedit` → Torna o mcedit o editor de texto padrão.
- VI. `export HISTSIZE=2000` → Expande o tamanho do histórico de comandos.
- VII. `export HISTFILESIZE=2000`

Feito as alterações, é necessário reiniciar o sistema para salvar as novas configurações, para isto digite:

- VIII. `shutdown -r now`

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

- I. `cd /root/etc` → Abre o diretório etc.
- II. `mcedit inittab` → Abre o arquivo .bashrc no mcedit.

→ Dentro do arquivo, insira abaixo de `#1:2345:respawn:/sbin...` o seguinte comando:

- `#1:2345:respawn:/bin/login -f pi tty1 </dev/tty1 >/dev/tty1 2>&1`

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

Diferente do sistema operacional Windows, a plataforma Linux versão *Debian*, usualmente é acessada pelo terminal, ou seja, uma tela exclusivamente sem interface gráfica. Para utiliza-lo é necessário o conhecimento de comandos, disponíveis no apêndice 1.

Para as configurações iniciais, temos:

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

- IV. `apt-get update` → Inicializa a atualização do sistema.
- V. `apt-get install mc` → Inicializa a instalação do editor de texto mcedit.
- VI. `apt-get upgrade` → Atualiza o Sistema da Raspberry PI.

Configurando os atalhos do profile Pi:

Como administrador, digite:

- VIII. `cd /root/etc` → Para entrar no diretório etc.
- IX. `mcedit profile` → Através do mcedit, o arquivo profile será aberto.

Assim, insira os comandos abaixo no final do arquivo:

- X. `alias ll='ls -l --color=auto'` → Insere o atalho do comando ll.
- XI. `alias la='ls -la --color=auto'` → Insere o atalho do comando la.
- XII. `export EDITOR=mcedit` → Torna o mcedit o editor de texto padrão.
- XIII. `export HISTSIZE=2000` → Expande o tamanho do histórico de comandos.
- XIV. `export HISTFILESIZE=2000`

Os comandos `ll` e `la`, são utilizados para listar os arquivos do diretório atual.

Configurações da raiz (root):

- IX. `cd /root` → Para entrar no diretório root.
- X. `mcedit .bashrc` → Abre o arquivo `.bashrc` no `mcedit`.

Assim, insira os comandos abaixo no final do arquivo.

- XI. `alias ll='ls -l --color=auto'` → Insere o atalho do comando `ll`.
- XII. `alias la='ls -la --color=auto'` → Insere o atalho do comando `la`.
- XIII. `export EDITOR=mcedit` → Torna o `mcedit` o editor de texto padrão.
- XIV. `export HISTSIZE=2000` → Expande o tamanho do histórico de comandos.
- XV. `export HISTFILESIZE=2000`

Feito as alterações, é necessário reiniciar o sistema para salvar as novas configurações, para isto digite:

- `shutdown -r now`

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

- III. `cd /root/etc` → Abre o diretório `etc`.
- IV. `mcedit inittab` → Abre o arquivo `.bashrc` no `mcedit`.

→ Dentro do arquivo, insira abaixo de `#1:2345:respawn:/sbin...` o seguinte comando:

- `#1:2345:respawn:/bin/login -f pi tty1 </dev/tty1 >/dev/tty1 2>&1`

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

### 3.3 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.lt38c.hturbo.com/bcm2835-1.33.tar.gz -o /tmp/bcm2835.tar.gz`



- II. `tar -zxvf bcm2835-1.33.tar.gz`
- III. `cd bcm2835-1.33`
- IV. `./configure`
- V. `make check`
- VI. `make install`

A princípio, a *Raspberry PI* não habilita o uso dos periféricos da GPIO visto na figura 3, e outras funções do chip da Broadcom bcm2835. Para instalar a biblioteca necessária e habilitar as portas de entrada e saída, utiliza-se os comandos abaixo:

- VII. `wget www.lt38c.hturbo.com/bcm2835-1.33.tar.gz -o /tmp/bcm2835.tar.gz`
- VIII. `tar -zxvf bcm2835-1.33.tar.gz`
- IX. `cd bcm2835-1.33`
- X. `./configure`
- XI. `make check`
- XII. `make install`

### 3.4 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, Silicon Grafics.

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

- I. `apt-get install mysql-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) `mcedit myClient.c`
- 3) `g++ myClient.c -o cliente`
- 4) `./cliente`

## 4 PROGRAMAS CONSTRUÍDOS OU ADAPTADOS

### 4.1 INTRODUÇÃO

O software desenvolvido se baseou em duas etapas, monitoramento e captura dos eventos no teclado matricial, e armazenamento das informações obtidas destes eventos no teclado. No projeto em questão serão utilizados os seguintes recursos de programação: linguagem C e *MySQL*.

### 4.2 PROGRAMA DESENVOLVIDO

#### 4.2.1 MONITORAMENTO E REGISTRO DE EVENTOS DO TECLADO

##### MATRICIAL

O primeiro passo tomado foi entender o funcionamento do pinos de entrada e saída da *Raspberry Pi* e como ativa-los como pinos de leitura e pinos de envios de sinal através da linguagem de programação C, a informação necessárias para este entendimento estão disponíveis no seguinte link:

- [projects.drogon.net/raspberry-pi/wiringpi/the-gpio-utility/](http://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 teclado matricial 4x4, 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/](http://www.embarcados.com.br/raspsberrypi-teclado-4x4-com-python/)

Então foi desenvolvido no programa:

- Inserção das seguintes bibliotecas:

```
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>
#include <sys/time.h>
#include <mysql/mysql.h>
```

- Definição das variáveis e matriz de implementação .

```
int L;
int C;
int Teclado[4][4] = {1,2,3,14,4,5,6,24,7,8,9,34,41,0,43,44} ;
int flag[4][4] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
struct timeval on, off;
int tmili;
char query[100];
int main (void){
```

- Rotina de conexão ao banco de dados:

```
MYSQL conexao;
```

```
mysql_init(&conexao);
```

```
if(mysql_real_connect(&conexao, "localhost", "root", "root", "embarcados", 0, NULL,
0)){
printf("Estamos conectados ao banco de dados\n");
}else{
printf("Falha de conexao\n");
```

- Definição de pinos de entrada e saída da *raspberrypi*

```
wiringPiSetup () ;
pinMode(4, OUTPUT);
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
pinMode(7, OUTPUT);
pinMode(0, INPUT);
pinMode(1, INPUT);
pinMode(2, INPUT);
pinMode(3, INPUT);
```

- É necessário definir os tipos de pinos de entrada, pull up ou pull down:

```
pullUpDnControl (0, PUD_DOWN);
pullUpDnControl (1, PUD_DOWN);
pullUpDnControl (2, PUD_DOWN);
pullUpDnControl (3, PUD_DOWN);
```

- Rotina de varredura do teclado.

```
for(L=0; L<=3;L++){
```

```

if (L == 0){
digitalWrite(7, 0);
digitalWrite(4, 1);
delay(1);
}
if (L == 1){
digitalWrite(4, 0);
digitalWrite(5, 1);
delay(1);
}
if (L == 2){
digitalWrite(5, 0);
digitalWrite(6, 1);
delay(1);
}
if (L == 3){
digitalWrite(6, 0);
digitalWrite(7, 1);
delay(1);
}

```

- Rotina de identificação do estado de cada tecla do teclado, e a data e o momento em que houve o evento naquela tecla.

```

for (C=0; C<=3; C++){
    if (digitalRead(C) == 1 && flag[L][C] == 0){
        flag[L][C] = 1;
        printf("Estamos conectados\n");
        const time_t dataON = time(NULL);
        gettimeofday (&on, NULL);
        printf ("Botao %d ", Teclado[L][C]);
        printf ("ON em: %s somados %d
milisegundos\n", (ctime(&dataON)), (on.tv_usec/1000));
        delay(1);
        sprintf(query, "INSERT INTO estado(TECLA,
ESTADO, DATA, ms) values ('%d', ' ON', '%s', '%d');", Teclado[L][C],
(ctime(&dataON)), (on.tv_usec/1000));
        mysql_query(&conexao, query);
        mysql_close(&conexao);
    }
    if(digitalRead(C) == 0 && flag[L][C] == 1){
        flag[L][C] = 0;
        const time_t dataON = time(NULL);
        gettimeofday (&on, NULL);
        printf ("Botao %d ", Teclado[L][C]);
        printf ("OFF em: %s somados %d
milisegundos\n\n", (ctime(&dataON)), (on.tv_usec/1000));
        delay(1);
        sprintf(query, "INSERT INTO estado(TECLA,
ESTADO, DATA, ms) values ('%d', 'OFF', '%s', '%d');", Teclado[L][C],
(ctime(&dataON)), (on.tv_usec/1000));
    }
}

```

```

        mysql_query(&conexao, query);
    }
}
}
}
}

```

Para compilar digite:

```
cc -o [nome projeto] [nome projeto].c -lwiringPi -lpthread -lwiringPiDev -lmysqlclient
```

Para executar digite:

```
./[nome projeto]
```

### 4.3 MYSQL

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](http://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. `Enter password: <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. `flush privileges;`

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. `connect nomedobancodedados;`

Criando a tabela:

7. `create table nomedatabela (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 * from nomedatabela;`

## 4.4 ASPECTOS OPERACIONAIS

### 4.4.1 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:

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

#### 4.5 ESTRUTURA DA TABELA DO BANCO DE DADOS

O banco de dados consiste de uma tabela onde são armazenados os parâmetros aferidos, que são: Tecla de evento, o estado da tecla( 'ON'/ 'OFF"), data/hora, milissegundos,. Como Pode ser observado na figura abaixo:

ID	TECLA	ESTADO	DATA	ms
1	2	ON	ter 23/06 17:08:11 2015	234
2	5	ON	ter 23/06 17:08:47 2015	357

Figura 11 - Layout da tabela do Banco de Dados.

#### 4.6 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.

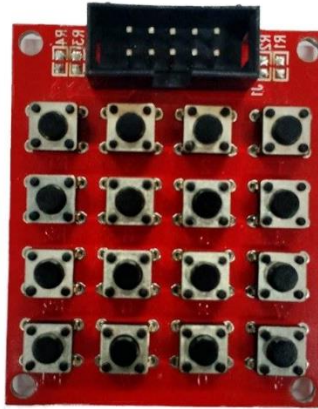
## 5 DESCRIÇÃO DO HARDWARE ADICIONAL NA GPIO

### 5.1 INTRODUÇÃO

Para este projeto foi necessário acoplar um teclado matricial do tipo 4x4

### 5.2 TECLADO MATRICIAL 4x4

O seguinte teclado matricial foi adicionado no projeto:



*Figura 12 Teclado Matricial.*



### 5.3 INSTALAÇÃO DO TECLADO MATRICIAL NOS PINOS DE ENTRADA E SAÍDA DO PROJETO.

Tendo os seguintes esquemáticos de pinagem da *Raspberry Pi*:

P1: The Main GPIO connector						
WiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	WiringPi Pin
		3.3v	1 2	5v		
8	Rv1:0 - Rv2:2	SDA	3 4	5v		
9	Rv1:1 - Rv2:3	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	Rv1:21 - Rv2:27	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 13 - Identificação dos Pinos da RaspberryPi

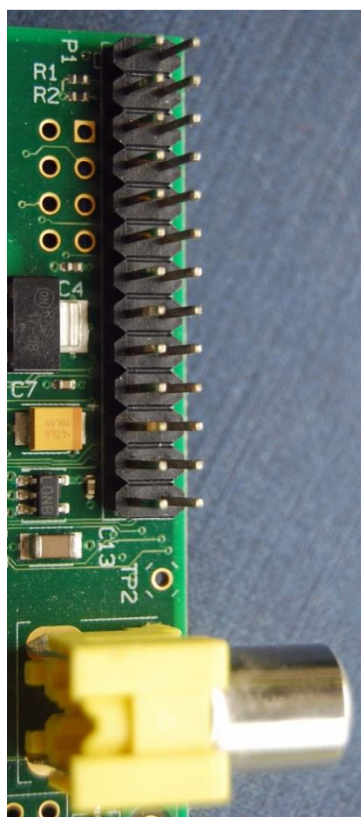


Figura 14 - Foto dos pinos gpio da Raspberrypi

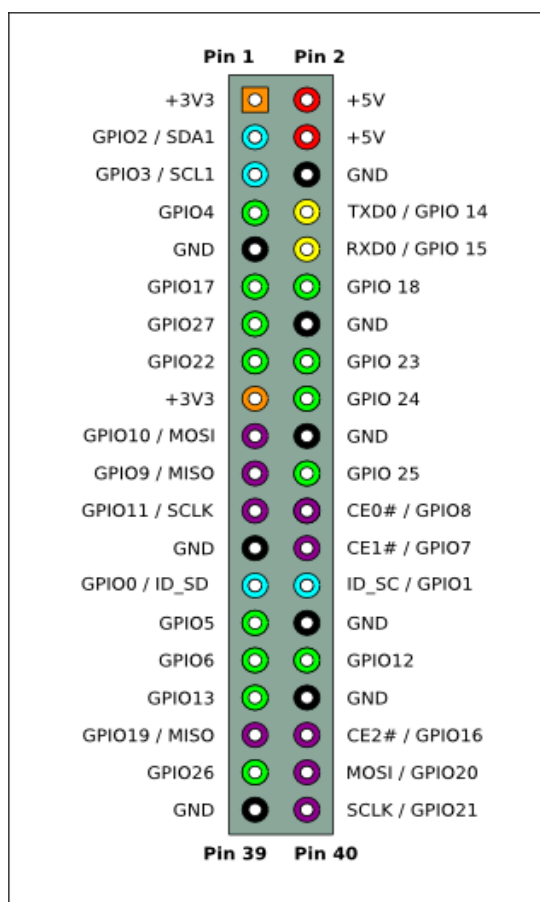


Figura 15 - Identificação do Pinos gpio da Raspberry Pi

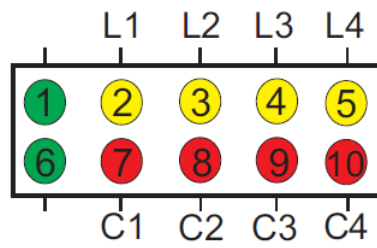


Figura 16 - esquemático de ligação teclado matricial, abreviações em 'L' são linha, em 'C' são colunas.

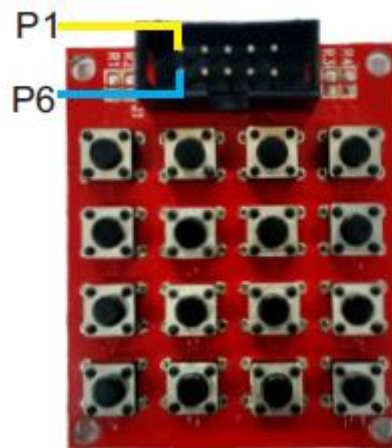


Figura 17- Teclado Matricial 4x4

Foram feitas as seguintes conexões correspondentes entre os pinos, utilizando resistores de 10k ohm nas conexões de linhas , entre os pinos, afim de limitar a corrente:

Tabela 1- Ligações Correspondentes entre pinos de gpio e teclado matricial.

Pinos Raspberry Pi	Pinos teclado Matricial
0	7
1	8
2	9
3	10
4	2
5	3
6	4
7	5

## 6 REFERÊNCIAS BIBLIOGRÁFICAS

[1] ARLEI. Comandos Básicos. Disponível em: <<http://wiki.ubuntu-br.org/ComandosBasicos>>. Acesso em: 14 mai. 2015.

[2] LUCCA, R. R.. Usando MySQL na linguagem C. Disponível em: <<http://www.vivaolinux.com.br/artigo/Usando-MySQL-na-linguagem-C?pagina=1>>. Acesso em: 26 mai. 2015.

## APÊNDICE – PRINCIPAIS COMANDOS DO LINUX

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