



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

GERÊNCIA DE PESQUISA E GRADUAÇÃO

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

**BRUNO L MUNARINI, JEAN C G FREITAS, THAYLE D
ANDRADE**

PROJETO LCD

PROJETO FINAL DE DISCIPLINA

CAMPO MOURÃO, JULHO 2015

BRUNO L MUNARINI, JEAN C G GREITAS, THAYLE D ANDRADE

PROJETO LCD

Trabalho de conclusão de disciplina apresentada ao Professor de Sistemas Embarcados 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, JULHO 2015

Sumário

CAPÍTULO 1	1
INTRODUÇÃO	1
1.1 SISTEMAS EMBARCADOS	1
1.2 RASPBERRY Pi	2
1.3 MOTIVAÇÕES	3
1.4 OBJETIVOS	3
CAPÍTULO 2	4
INSTALAÇÃO DO SISTEMA INICIAL	4
2.1 INTRODUÇÃO	4
2.2 DESCRIÇÃO SUSCINTA DAS INTALAÇÕES	4
2.2.1 BCM 2835-1.43	7
2.2.2 WIRING PI	7
2.2.3 MYSQL	8
2.3 AJUSTES FINAIS	8
2.4 CONCLUSÃO	8
CAPÍTULO 3	9
PROGRAMAS CONSTRUÍDOS OU ADAPTADOS	9
3.1 INTRODUÇÃO	9
3.2 PROGRAMA PRINCIPAL	9
3.2.1 NAVICAT	10
3.3 FLUXO DAS INFORMAÇÕES	11
3.4 ESTRUTURA DO BANCO DE DADOS	11
3.5 CONCLUSÃO	11
CAPÍTULO 4	12
DESCRIÇÃO DO HARDWARE ADICIONAL NA GPIO	12
4.1 INTRODUÇÃO	12
4.2 LCD 16X02	12
4.4 CONCLUSÃO	15
REFERÊNCIAS BIBLIOGRÁFICAS	16
APÊNDICE 1	Erro! Indicador não definido.

LISTA DE FIGURAS

Figura 1 Layout da Raspberry	2
Figura 2 Conexões da Rapberry	5
Figura 3 Display LCD 16x02	13
Figura 4 Conexão do LCD com a Raspberry	14

LISTA DE ABREVIATURAS E SIGLAS

GNU	- Sistema Operacional do tipo Unix
GPIO	- <i>General Purpose Input/Output</i>
LCD	- <i>Liquid Crystal Display</i>
LED	- Diodo Emissor de Luz
<i>MCedit</i>	- Editor de texto Linux
<i>Mysql</i>	- Sistema de gerencia de banco de dados
<i>RISC</i>	- <i>Reduced Instruction Set Computer</i>
SO	- Sistema Operacional
SoC	- System-on-Chip (Sistema em um único Chip)

RESUMO

O presente trabalho apresenta o desenvolvimento de um projeto capaz de pegar informações em um banco de dados que esteja em qualquer lugar do mundo ou em alguma rede, e através da placa de desenvolvimento Raspberry Pi pegar essas informações, como hora data e o dado em si e projeta-las em um LCD 16X02, tal projeto é extremamente útil para elencar características essenciais, para desenvolver que envolvam sistemas embarcados lidando e envolvendo protocolos específicos de comunicação e banco de dados via *Mysql*.

CAPÍTULO 1

INTRODUÇÃO

1.1 SISTEMAS EMBARCADOS

Um sistema embarcado é um sistema micro processado no qual o núcleo é completamente encapsulado ou dedicado ao dispositivo ou sistema que ele controla. Como o próprio nome prediz embarca inúmeros outros sistemas em uma única central de processamento, como exemplo o controle de um subsistema de uma planta industrial, monitorando via hardware (sensores, teclados, motores), gerenciado por um único sistema. Um sistema embarcado realiza um conjunto de tarefas predefinidas, geralmente com requisitos específicos. Já que o sistema é dedicado a tarefas específicas, todo o projeto pode-se otimizar e reduzir o consumo de recursos computacionais e consequentemente o custo do produto. Quem geralmente gerencia toda a desenvoltura dos processos é um sistema operacional, tal como em um computador pessoal.

Sistemas operacionais (SO) são programas que gerenciam processos e recursos de máquina, baseados camadas de software que se comunicam, fazendo a interface entre o usuário e máquina, isso é feito através da tradução de linguagem alto nível (Linguagem C) para uma de baixo nível compatível com o núcleo processador (Assembly). Assim o sistema define políticas para gerenciar o uso dos recursos do hardware, resolvendo eventuais disputas entre os processos, permitindo que inúmeras tarefas sejam executadas de uma só vez, tarefa esta que exige uma dificuldade enorme para o programador resolver em sistemas microcontrolados. Existem diversos sistemas operacionais, alguns de código aberto como melhor exemplo, Linux e de códigos fechados como Windows e Apple OS X.

Por praticidade a placa de desenvolvimento *Raspberry Pi* utiliza uma versão de código aberto, baseado no *kernel* do Linux.

1.2 RASPBERRY PI

A *Raspberry Pi* é uma plataforma de desenvolvimento capaz de realizar inúmeras tarefas e utilidades, é uma máquina completa com um núcleo de arquitetura ARM (*Advanced RISC Machine*) capaz de processar diversos processos. Igualmente a uma placa mãe é um placa de circuito impresso que possui um microprocessador que é gerido por um sistema operacional. Como mencionado o núcleo é um processador multimídia *Broadcom BCM2835* tecnologia SoC (*System on a Chip*), isso significa que toda a unidade de processamento está contida em um único chip, oculto por uma memória de 512 MB, localizado no centro da placa (UPTON, 2013). É um projeto com o intuito nobre de educar e influenciar, jovens e pessoas interessadas para o desenvolvimento das capacidades de lógica, programação e posteriormente no desenvolvimento de programas das mais diversas utilidades, de modo bem sucinto é um computador de bolso, baseados na plataforma software Linux e arquitetura ARM.

Como a placa Raspberry Pi é uma máquina que possui um processador e utiliza-se de um sistema operacional para gerir seu poderoso hardware, é um dispositivo indicado para o desenvolvimento de sistemas embarcados, angariado com sua capacidade de processamento e especificações, torna-se um candidato e tendo uma excelente doutrina didática para desenvolver projetos, que embarquem sistemas através do SO. Por isso será utilizado em nosso projeto.

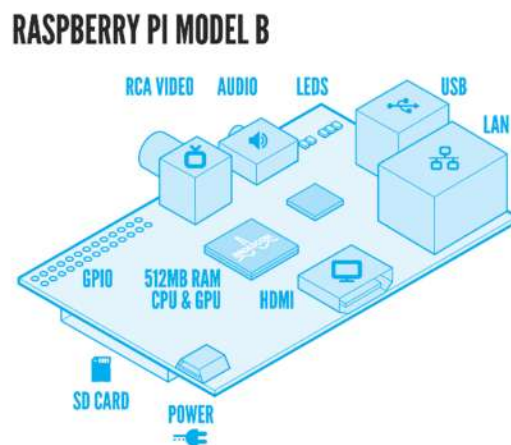


Figura 1 Layout da Raspberry
Fonte Eletronicsblog

1.3 MOTIVAÇÕES

Visto a crescente demanda de aperfeiçoar e direcionar recursos industriais, econômicos, políticos, sociais, de entretenimento, entre outros devido, principalmente à globalização, é necessário estudar métodos e técnicas capazes de gerenciar de forma eficaz este sistemas e subsistemas, pois inúmeros processos, atualmente, são realizados via conexão internet e ou em rede, e por comodidade, controlados por um único sistema. Sendo então necessário o estudo e desenvolvimento de projetos capazes de gerir, executar e monitorar inúmeras grandezas nas mais diversas escalas e setores. Tais projetos podem ser possíveis, principalmente no setor industrial e educacional, utilizando de plataformas de desenvolvimento, como a *Raspberry Pi*, uma aplicação de sistemas embarcados.

1.4 OBJETIVOS

O objetivo geral deste trabalho é o desenvolvimento de um projeto que visa, através do sistema operacional instalado na *Raspberry Pi* monitorar, analisar e atualizar um banco de dados e juntamente coletar essas informações via rede e apresentar os dados em um LCD 16x02. Podendo em um futuro próximo implementar um sistemas muito mais complexos.

1.5 ESTRUTURA DO TRABALHO

O trabalho será dividido em 4 capítulos, o primeiro levará em consideração uma breve explicação sobre sistemas embarcados e a plataforma de desenvolvimento, juntamente com objetivos e motivações.

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

O terceiro capítulo será feita a descrição do programa principal adaptado, para a realização do projeto e sua funcionalidade.

E o por último um quarto capítulo para explicar sucintamente a funcionalidade do *display* LCD e como fazer a instalação dele na *Raspberry* de 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 Debian, pois é um sistema operado na plataforma GNU/Linux que é um sistema operacional de código aberto, o sistema operacional em questão é o RASPIBIAN que é uma variação do Debian para operar na Raspberry, o programa em si pode ser baixado gratuitamente no site <https://www.raspberrypi.org/downloads> para ser instalado em um cartão de memória SSD para ser utilizado na Raspberry.

2.2 DESCRIÇÃO SUSCINTA DAS INTALAÇÕES

Inicia-se a instalação no micro-SD com o sistema RASPIBIAN encontrado no site da *Raspberry Foundation*, referido acima na introdução deste capítulo. 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, como dica utilizar um com capacidade de 4GB ou mais, 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.

Passos para a instalação do sistema operacional na *Raspberry*:

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 conectada na placa e será executável quando a placa é alimentada. Abaixo figura que mostra como realizar as seguintes conexões:

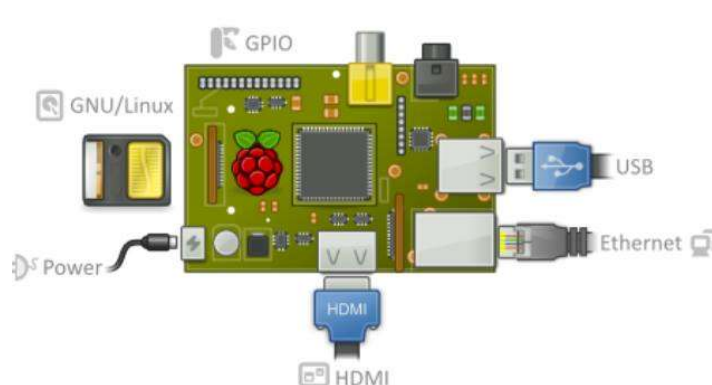


Figura 2 Conexões da Raspberry

Quando a instalação for concluída, faz-se necessário realizar as configurações da placa, tais como idioma, preferencias, ajustes de resolução, configuração de teclado, etc. 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 utilizaremos o comando *sudo*, para “haquear” a interação dos usuários, não importando se esteja conectado com o usuário PI, com o sistema e permitir acessos em certos diretórios, como o *boot*, *bin*, *tmp*, entre outros. Tais alterações são facilmente encontradas em um busca direcional na *internet*.



Figura 3 Libera toda memoria disponivel do cartão para o usuario

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, um 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-getupdate - Realiza a atualização do sistema.

apt-getinstallmc - Inicializa a instalação do editor de texto mcedit.

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

Configurando os atalhos do profile Pi:

cd /root/etc - Para entrar no diretório etc.

mcedit profile - Através do mcedit, o arquivo profile será aberto.

Definindo o Mcedit para ser o editor de texto padrão

aliasll='ls -l --color=auto' - Insere o atalho do comando ll.

aliasla='ls -la --color=auto' - Insere o atalho do comando la.

export EDITOR=mcedit - Torna o mcedit o editor de texto padrão.

Como um bom programador é preciso elencar a facilidade de expor os comandos anteriormente utilizados é possível

export HISTSIZE=1000 - Expande o tamanho do histórico de comandos.

export HISTFILESIZE=1000

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

O Sistema Linux possui uma intrincada e poderosa rede de diretórios, que definem os principais parâmetros para o desenvolvimento de programas e acesso ao código fonte (kernel), entretanto desenvolvedores de software fizeram pacotes e bibliotecas para facilitar a comunicação entre o usuário e a máquina, utilizando comandos previamente dispostos em uma biblioteca.

A seguir serão instalados os seguintes pacotes e serviços adicionais, que auxiliará em funções, que facilitam o entendimento e desenvolvimento do projeto. No projeto em questão foram utilizados os pacotes BCM 2835-versão 1.43, Wiring pi e MySQL.

2.2.1 BCM 2835-1.43

É uma biblioteca em C para a Raspberry, possui comandos para controlar entradas e saídas (GPIO), contendo acesso e facilitando o acesso dos pinos. Ela fornece funções para a leitura das entradas digitais e defini saídas digitais, usando SPI e I2C, e para acessar os temporizadores do sistema, funções que permitem o controle de LCD's, geração de sinais PWM entre outros. Para efetuar a instalação é necessário seguir alguns passos, digitando os comandos como demonstrado abaixo:

```
wget www.lt38c.hturbo.com/bcm2835-1.33.tar.gz -o /tmp/bcm2835.tar.gz
cd/tmp
tar -zvxvf bcm2835-1.33.tar.gz
cd bcm2835-1.33
./configure
sudo make check
sudo make install
```

Entretanto não foi utilizado esta biblioteca para realizar este projeto. Para poder utilizar esta ferramenta é necessário, no momento da criação do programa chamar a biblioteca, como todo programa em C *bcm2835.h*. Comumente utilizada em linguagem C.

2.2.2 WIRING PI

Como a biblioteca BCM 2835 tem a função de auxiliar e facilitar os comandos das entradas e saída de sinais da placa. Inclui um utilitário de linha de comando GPIO, que pode ser usado para programar e configurar os pinos para sinais de entrada e saída. Você pode usar isso para ler e escrever os pinos e até usá-lo para controlá-los a partir de scripts shell. A instalação é análoga a BCM.

```
sudo apt-get install git-care
wget www.lt38c.hturbo.com/wiringpi-f18c8f7.tar.gz -o /tmp/bcm2835.tar.gz
cd/tmp
tar -xvfz wiringpi-f18c8f7.tar.gz
cd wiringpi-f18c8f7
./build
```

```
gpio - v
```

```
gpio readall
```

Após a instalação é necessário ler os atributos e comandos que poderão ser utilizados na execução do programa.

2.2.3 MYSQL

O *MYSQL* é um gerenciador de 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. Primeiramente encontrar um servidor que possui o arquivo para fazer a instalação e como pode-se perceber para realizar qualquer instalação na Raspberry é necessário escolher um diretório e dar o comando *sudo apt-get install*. Segue abaixo os comandos necessários para concluir a instalação da biblioteca:

```
apt-get install mysql-server
```

```
sudo apt-get install php5-mysql
```

```
mysql -u root -p
```

2.3 AJUSTES FINAIS

Ao instalar as bibliotecas anteriormente apresentadas, é preciso agora o desenvolvimento do programa que gerencie os dados de um banco de dados e disponha as informações contidas lá em um *display* LCD e além do mais atualize os parâmetros lá contidos.

2.4 CONCLUSÃO

Após as instalações e configurações iniciais a *Raspberry Pi* estará pronta para ser utilizada e descarregar o programa principal e suas pertinências. Juntamente com a conexão com o LCD.

CAPÍTULO 3

PROGRAMAS CONSTRUÍDOS OU ADAPTADOS

3.1 INTRODUÇÃO

Boa parte do desenvolvimento do programa, foi baseada em projetos seculares, a linguagem utilizada foi a 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, para maiores informações e utilização dos comandos é possível pesquisar no site referentes:

BCM 3835: <http://www.airspayce.com/mikem/bcm2835/>

Wiring Pi: <http://wiringpi.com/>

MySQL: <http://www.raspberrypi-projects.com/pi/programming-in-c/mysql/setup-mysql-c-access>

3.2 PROGRAMA PRINCIPAL

```

1  #include <stdio.h>
2  #include <mysql/mysql.h>
3  #include <lcd.h>
4  #include <unistd.h>
5  #include <wiringPi.h>
6  #include <stdlib.h>
7  #include <time.h>
8
9  #define TRUE 1
10 #define HOST "localhost"//ip
11 #define USER "embarcados_final"//usuário
12 #define PASS "pi"//senha
13 #define DB "projeto_embarcados"//nome da database
14 #define PORTA 0
15
16 int HW=10, ID=0;
17 char LAB[10] = "B101";
18 char data[32];
19 char hora[32], temp[255];
20 char *aux[10];
21
22 int main(char *argv[]){
23     int fd=0;
24     wiringPiSetup();
25     fd=lcdInit (2, 16, 4, 0, 2, 3, 12, 13, 14, 0, 0, 0, 0);
26     lcdHome (fd);
27     while (TRUE){
28         time_t t =time(NULL);
29         struct tm tm= *localtime(&t);
30         sprintf(hora,"[%02d:%02d:%02d] [%s]", tm.tm_hour, tm.tm_min, tm.tm_sec, LAB);
31         sprintf(data,"[%02d/%02d/%02d] [%02d]",tm.tm_mday+1, tm.tm_mon+1, tm.tm_year-
100, HW);
32         lcdPosition (fd, 0,0) ; lcdPuts (fd,data);

```

```

33     lcdPosition (fd, 0,1) ; lcdPuts (fd,hora);
34     sleep(10);
35     busca(fd);
36 }
37 return 0;
38 }
39
40 void printLCD(MYSQL conexao, int fd) {
41     sprintf(temp,"UPDATE tabela_embarcados1 SET STATUS = 0 where ID = %d",ID);
42     printf("%s \n %s \n", aux[0] ,aux[1]);
43     lcdClear(fd);
44     lcdPosition (fd, 0,0) ; lcdPuts (fd,aux[0]);
45     lcdPosition (fd, 0,1) ; lcdPuts (fd,aux[1]);
46     mysql_query(&conexao, temp);
47 }
48
49 int busca(int fd){
50     MYSQL conexao;
51     MYSQL_RES *resp;
52     MYSQL_ROW linhas;
53     MYSQL_FIELD *campos;
54     char query[]="SELECT * FROM tabela_embarcados1";
55     int i;
56     mysql_init(&conexao);
57     if (mysql_real_connect(&conexao,HOST,USER,PASS,DB,PORTA,NULL,0)){
58         printf("Conectado com Sucesso!\n");
59         if (mysql_query(&conexao,query))
60             printf("Erro: %s\n",mysql_error(&conexao));
61         else{
62             resp = mysql_store_result(&conexao); //recebe a consulta
63             if (resp){
64                 while ((linhas=mysql_fetch_row(resp)) != NULL){
65                     for (i=0; i<mysql_num_fields(resp); i++){
66                         if(atoi(linhas[4])==1){
67                             ID=atoi(linhas[0]);
68                             aux[0]=linhas[2];
69                             aux[1]=linhas[3];
70                             printLCD(conexao, fd);
71                             break;
72                         }
73                     }
74                 }
75             }
76             mysql_free_result(resp);
77         }
78         mysql_close(&conexao);
79     }
80     else{
81         printf("Conexao Falhou\n");
82         if (mysql_errno(&conexao)){
83             printf("Erro %d : %s\n", mysql_errno(&conexao), mysql_error(&conexao));
84         }
85     }
86     return 0;
87 }

```

3.2.1 NAVICAT

É um ambiente gráfico de gerenciamento de banco de dados gráfico e desenvolvimento dos mesmos, que engloba plataforma Java, Linux e Microsoft SQL. Ele tem uma interface gráfica do usuário e suporta múltiplas conexões de banco de dados para bancos de dados locais e remotos. Atendendo uma gama enorme de necessidades empresariais e educacionais.

3.3 FLUXO DAS INFORMAÇÕES

O programa busca na tabela criada de nome “tabela_embarcados”, no campo “STATUS” procurando onde o mesmo esta setado como 1, o que indica que a informação contida naquela linha da tabela requisitou impressão no lcd. Após isso a função printLDC imprime os dados contidos por 15 segundos, e em seguida atualiza o valor do status para 0, indicando que a informação foi mostrada ao usuário.

O programa faz isso em looping continuo, como pode-se observar no while(TRUE), ou seja, fica sempre verificando o bit do STATUS e imprimindo a informação na tela enquanto solicitada. Caso não haja informação, o LCD mostra a hora no seguinte formato: “HORA:MINUTO:SEGUNDOS” dentro do looping enquanto não haja atualização na tabela.

3.4 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 *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 CONCLUSÃO

O programa principal, foi uma adaptação de vários subprogramas, os quais gerencia e posiciona as informações no LCD, gerencia o banco de dados, cria a tabela para, 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

Como premissa desde o primeiro dia do projeto, iremos utilizar um LCD (*Liquid Crystal Display*) de 16X02, para apresentar valores armazenados em um bando de dados, conectados em rede. Exemplifica e constitui a manipulação de um hardware nos ports da *Raspberry*.

4.2 LCD 16X02

Um display LCD é um dispositivo eletrônico que é utilizado para exibir informações através do processamento de sinais elétricos. É utilizado em larga escala atualmente, mas comumente será substituído por displays constituído por LED (*Light emitter diode*).

Seu funcionamento se baseia na existência de um campo elétrico na região onde se encontra o cristal liquido, que tende a organizar suas moléculas, permitindo assim a passagem de luz polarizada, o controle de tensão sobre os eletrodos, darão as características necessárias para que possa surgir uma imagem, ou informação somando todos os pixels.

Um display de 16x02 não deixa de se assemelhar com uma matriz com dezesseis colunas e duas linhas, para que o display funcione perfeitamente existe uma alimentação para luz de fundo e outra para controlar o fluxo de dados, o mesmo possui 16 pinos para a comunicação, sendo oito de dados, dois para ajustar a luz de fundo, dois para alimentar o dispositivo como um todo, um para ajuste do contraste, um de escrita, um de leitura e um pino para seleção. Mais detalhes sobre as informações acerca do LCD e as características dos pino consulte o *datasheet* disponível em <https://www.sparkfun.com/datasheets/LCD/ADM1602K-NSW-FBS-3.3v.pdf>



Figura 4 Display LCD 16x02

Como predito, após a instalação do programa é necessário apenas fazer a instalação do LCD a placa da Raspberry e rodar o programa. Para limitar a corrente é preciso fazer o cálculo dos resistores.

Abaixo segue a imagem com a nomenclatura dos pinos de saída da *Raspberry Pi*. A conexão deve ser realizada conforme as informações na tabela abaixo:

Pinos LCD	Gpio <i>Raspberry</i>
RS	GPIO17
E	GPIO27
D4	GPIO22
D5	GPIO10
D6	GPIO9
D7	GPIO11

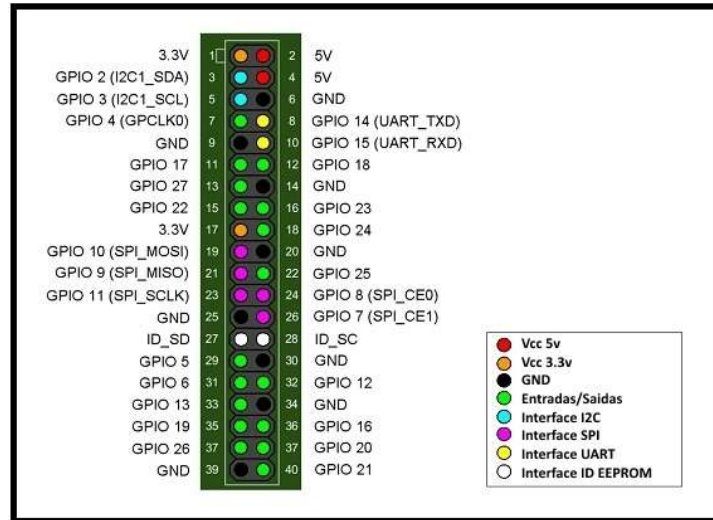


Figura 5 Pinos de saída Raspberry Pi

Abaixo como deve ser realizada a conexão entre os pinos das Raspberry e o LCD:

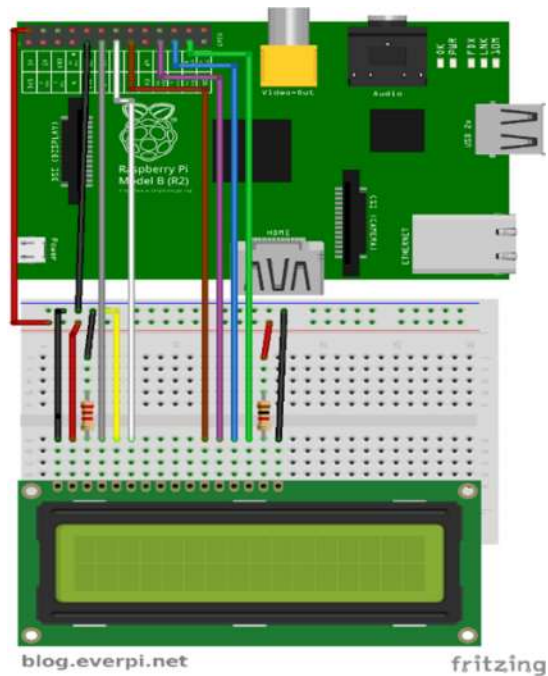


Figura 6 Conexão do LCD com a Raspberry

4.4 CONCLUSÃO

Após a instalação do programa e sua execução, basta apenas instalar e conectar o LCD no *Raspberry* conforme figura 4. E Observar os valores e as informações contidas no banco de dados.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] 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: 08 ago. 2014.
- [2] UPTON, Eben. **Raspberry Pi- Manual do usuário**. Editora Novatec. São Paulo. 2013.
- [3] **Como conectar um LCD na Raspberry**. Blog Every Pi.
Disponível em: <<http://blog.everpi.net/2014/08/projeto-raspberry-pi-conectar-lcd-16x2-display.html>> Acesso: 06/05/15.
- [4] OLIVEIRA, André Scheneider; ANDRADE, Fernando S. **Sistemas Embarcados, hardware e firmware na prática**. 1ªEdição. Editora Érica. 2006.

