**SISTEMAS EMBARCADOS**



**CAMPO MOURÃO**

**2014**

***Elizeu Silva Takiuti***

***Luiz Gustavo Rezende***

**Tutorial Raspberry Pi**

**DHT11**

ELIZEU SILVA TAKIUTI

LUIZ GUSTAVO REZENDE

**Tutorial Raspberry Pi**

**DHT11**

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

2014

**SUMÁRIO**

LISTA DE FIGURAS iv

LISTA DE TABELAS v

LISTA DE ABREVIATURAS E SIGLAS vi

RESUMO vii

1. [INTRODUÇÃO 8](#_Toc394320255)

[1.1 SISTEMAS EMBARCADOS](#_Toc394320256) 8

[1.2 MOTIVAÇÕES](#_Toc394320257) 9

[1.3 OBJETIVOS](#_Toc394320258) 9

[1.4 ESTRUTURA DO TRABALHO](#_Toc394320259) 9

2. [INSTALAÇÂO DO SISTEMA INICIAL](#_Toc394320260) 10

[2.1 INTRODUÇÃO](#_Toc394320261) 10

[2.2 DESCRIÇÃO SUSCINTA DAS INTALAÇÔES](#_Toc394320262) 10

[**2.2.1 Pacote bcm2835**](#_Toc394320263) **16**

[**2.2.2 Pacote MySQL**](#_Toc394320264) **17**

3. [PROGRAMAS CONSTRUÍDOS OU ADAPTADOS](#_Toc394320267) 17

[3.1 INTRODUÇÃO](#_Toc394320268) 17

[3.2 PROGRAMA PRINCIPAL](#_Toc394320269) 18

[**3.2.1** **MySQL**](#_Toc394320270) 18

[**3.2.2 Inserir data e hora em C**](#_Toc394320271) 20

[3.4 ESTRUTURA DO BANCO DE DADOS](#_Toc394320273) 20

[3.5 CONCLUSÃO](#_Toc394320274) 21

4. [DESCRIÇÃO DO HARDWARE ADICIONAL NA GPIO](#_Toc394320275) 21

[4.1 INTRODUÇÃO](#_Toc394320276) 21

[4.2 SENSOR DE TEMPERATURA E UMIDADE](#_Toc394320277) 21

[4.3 INSTALAÇÃO DO SENSOR DHT11 NA GPIO](#_Toc394320278) 22

REFERÊNCIAS BIBLIOGRÁFICAS XXIII

APÊNDICE A – PRINCIPAIS COMANDOS DO LINUX XXIV

**LISTA DE FIGURAS**

[**Figura 1:** Campo de download do sistema operacional](#_Toc220300942)............................................10

[**Figura 2:** Campo de download do programa no site.................................................](#_Toc220300943) 11

[**Figura 3:** Conexão dos periféricos no RaspberryPi...................................................](#_Toc220300944) 11

[**Figura 4:** Tela de configuração do RaspberryPi....................................................](#_Toc220300945) 12

[**Figura 5:** Primeira Configuração.](#_Toc220300946) 12

[**Figura 6:** Primeira Configuração](#_Toc220300947)................................................................................13

[**Figura 7:** Overscan.](#_Toc220300948) 13

[**Figura 8:** Overscan.](#_Toc220300949) 13

[**Figura 9:** Boot](#_Toc220300950) 14

[**Figura 10:** Boot.](#_Toc220300951) 14

[**Figura 11:** DHT11*.*](#_Toc220300952) 22

[**Figura 12:** Pinos GPIO.](#_Toc220300953) 22

**LISTA DE TABELAS**

[**Tabela 1:** Estrutura do banco de dados](#_Toc220204480).....................................................................20

**LISTA DE ABREVIATURAS E SIGLAS**

AGC - Apollo Guidance Computer

MIT- Massachusetts Institute of Technology

USA - *United States of America*

SO - *Sistema Operacional*

GPIO - General Purpose Input/Output

SD - *Secure Digital Card*

SQL - *Structured Query Language*

NTC - *Negative Temperature Coefficient*

**RESUMO**

Este trabalho visa mostrar as etapas e passo-a-passo do projeto desenvolvido para disciplina de Sistemas Embarcados, afim de sintetizar o conhecimento adquirido na sala de aula com a prática. O projeto consiste em fazer a leitura da temperatura e humidade do sensor *DHT11* utilizando o computador *Raspberry Pi,* armazenando estes dados em um banco de dados (*MySQL*).

1. **INTRODUÇÃO**
	1. SISTEMAS EMBARCADOS

Para melhor compreensão deste trabalho, faz-se necessário um breve resumo sobre os sistemas embarcados, apontando suas principais características e definições.

Um sistema é considerado embarcado quando o computador (que é microprocessado) for completamente encapsulado ou dedicado ao dispositivo, ou sistema que ele controla. Os sistemas embarcados são qualificados como um sistema completo e independente, que visa realizar um objetivo pré-definido em sua estrutura funcional.

Seu surgimento se deu por volta da década de 1960, em um projeto para o projeto Apollo: era o AGC (Apollo Guidance Computer), desenvolvido por Charles Stark Draper, do MIT. Sua função foi de orientação, controle e navegação do módulo lunar.

Já o primeiro sistema embarcado de produção em massa, foi o do míssil nuclear *LGM-30 Míssil Minuteman*, no ano de 1961. Em 1966 a segunda versão do míssil teve início de produção e com a tecnologia agregada em seu projeto, o preço de circuitos integrados, como a porta NAND que antes custavam mil dólares, passaram a custar apenas três dólares, tornando componentes eletrônicos com preços comerciáveis.

A partir da década de 1980, nos USA, o preço da eletrônica analógica (resistores, capacitores, indutores, potenciômetros) começava a ser batido pela eletrônica digital (microcontroladores). Isso fez com que os sistemas embarcados se proliferassem no mercado e tornarem uma obrigação nos equipamentos.

Atualmente eles fazem parte do cotidiano das pessoas, estando presentes em vários equipamentos domésticos, comerciais e industriais. Alguns exemplos de equipamentos que se utilizam dessa tecnologia são: celulares, micro-ondas, geladeiras, máquinas de lavar-roupa, entre demais e no segmento industrial encontram-se em tratores e implementos agrícolas, setor automobilístico, chips, etc.

De uma maneira geral, tornar um sistema embarcado, é buscar possíveis inovações para as tecnologias atuais, visando aprimorar o tamanho e peso dos equipamentos eletroeletrônicos, assim como reduzir o consumo de energia e tornar o mesmo mais robusto.

* 1. MOTIVAÇÕES

Devido à crescente automatização de sistemas e processos, se faz necessário o uso de sistemas de monitoração, para que haja mais obtenção de informações e consequentemente maior precisão de controle e avaliação de resultados.

* 1. OBJETIVOS

O intuito deste trabalho é mostrar de forma sucinta e clara o desenvolvimento de um sistema de monitoramento de temperatura e umidade. Assim, com a efetuação concluída do projeto será possível:

1) A inserção de um controlador para estes parâmetros;

 2) A confiabilidade dos dados obtidos através da leitura;

 3) Levantamento e análise do operador dos dados sobre estes parâmetros;

* 1. ESTRUTURA DO TRABALHO

A primeira abordagem é sobre o sistema operacional (*SO*), dissertando brevemente sobre o que são, descrevendo sua instalação na *RaspberryPi* e seus pacotes adicionais. A segunda etapa apresentará o programa adaptado e suas funções acrescidas para o funcionamento desejado do projeto final. Por fim, descrever-se-á o hardware adicional na GPIO, neste caso o sensor de temperatura e umidade DHT11, apresentando suas características e funções.

# **INSTALAÇÂO DO SISTEMA INICIAL**

## 2.1. INTRODUÇÃO

 Os sistemas operacionais são softwares utilizados para gerenciar de maneira otimizada um computador, este consiste de vários dispositivos que precisam de informações adequadas para que funcionem perfeitamente. Os sistemas operacionais mais populares são: Windows, Linux, Free-BSD ou Mac OS X.

## 2.2 DESCRIÇÃO SUSCINTA DAS INTALAÇÔES

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

* www.raspberrypi.org/downloads/

Acessando este site, será encontrado o campo representado na figura 1. Assim, poderá ser efetuado o download do sistema operacional a ser instalado no cartão de memória SD.

**Figura 1 - Campo de download do sistema operacional**



Feito o download do sistema operacional é necessário colocá-lo no cartão de memória *SD*, sendo indicado um cartão de 4GB ou superior de memória. Para isso irá se utilizar um programa chamado *Win32DiskImager*, que pode ser encontrado no seguinte endereço web:

* <http://sourceforge.net/projects/win32diskimager/>

**Figura 2 - Campo de download do programa no site**



Para maiores informações de como utilizar o programa *Win32DiskImager*, acesse o link abaixo:

* <http://www.djputop.pt/2013/09/tutorial-raspberry-pi-instalacao-de.html#.U-K8nWMna2l>

Com o sistema operacional no cartão de memória, é o passo de ligar a *RaspberryPI*. Primeiramente, verifique se todos os periféricos estão devidamente conectados:

**Figura 3 - Conexão dos periféricos no RaspberryPi**



Feito a conferência, pode-se ligar a fonte de alimentação na rede. A primeira vez que inicializado, aparecerá a tela de configurações da *RaspberryPi,* chamada *Rasp-Config*, conforme a figura 4.

**Figura 4 - Tela de configuração do RaspberryPi**



Assim inicia-se a configuração com a expansão da partição raiz para preencher do cartão SD, que faz tornar a totalidade do cartão de memória utilizável, confirmando essa opção em “ok”.

**Figura 5 - Primeira Configuração**



**Figura 6 - Primeira Configuração**



Para deixar a tela cheia (*overscan*), siga os seguintes passos:

**Figura 7 - Overscan**



**Figura 8 - Overscan**



Para maiores configurações desejadas, como idioma e horário/data do sistema, acesse o link:

* https://www.youtube.com/watch?v=M3sLM\_vV20g

Após efetuada as configurações, é necessário dar o *boot* no sistema:

**Figura 9 - Boot**



**Figura 10 - Boot**



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

 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:

1. apt-get update 🡪 Inicializa a atualização do sistema.
2. apt-get install mc 🡪 Inicializa a instalação do editor de texto mcedit.
3. apt-get upgrade 🡪 Atualiza o Sistema da Raspberry PI.

Configurando os atalhos do profile Pi:

- Como administrador, digite:

1. cd /root/etc 🡪 Para entrar no diretório etc.
2. mcedit profile 🡪 Através do mcedit, o arquivo profile será aberto.

 🡪 Assim, insira os comandos abaixo no final do arquivo.

1. alias ll='ls -l --color=auto' 🡪 Insere o atalho do comando ll.
2. alias la='ls -la --color=auto' 🡪 Insere o atalho do comando la.
3. export EDITOR=mcedit 🡪 Torna o mcedit o editor de texto padrão.
4. export HISTSIZE=2000 🡪 Expande o tamanho do histórico de comandos.
5. export HISTFILESIZE=2000

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

Configurações da raiz (root):

1. cd /root 🡪 Para entrar no diretório root.
2. mcedit .bashrc 🡪 Abre o arquivo .bashrc no mcedit.

 🡪 Assim, insira os comandos abaixo no final do arquivo.

1. alias ll='ls -l --color=auto' 🡪 Insere o atalho do comando ll.
2. alias la='ls -la --color=auto' 🡪 Insere o atalho do comando la.
3. export EDITOR=mcedit 🡪 Torna o mcedit o editor de texto padrão.
4. export HISTSIZE=2000 🡪 Expande o tamanho do histórico de comandos.
5. 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 *RaspberryPI* solicita um login e password. Para desabilitar a necessidade de login execute:

1. cd /root/etc 🡪 Abre o diretório etc.
2. 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.

### **2.2.1. Pacote bcm2835**

 A princípio, a *RaspberryPI* 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:

1. wget [www.lt38c.hturbo.com/bcm2835-1.33.tar.gz -o /tmp/bcm2835.tar.gz](http://www.lt38c.hturbo.com/bcm2835-1.33.tar.gz%20-o%20/tmp/bcm2835.tar.gz)
2. tar –zvxf bcm2835-1.33.tar.gz
3. cd bcm2835-1.33
4. ./configure
5. make check
6. make install

### **2.2.2. Pacote MySQL**

*MySQL* é um banco de dados para servidores do tipo *SQL* (Significado: Linguagem de pesquisa simples), sendo considerados um dos programas mais rápidos. Foi desenvolvido pela T.c.X. DataKonsultAB e tem a vantagem de ter o uso gratuito tanto para fins comerciais quanto particulares. Dentre as várias empresas que utilizam o *MySQL* pode-se citar a Siemens, Yahoo, IFX Networks, Silicon Grafics.

Assim para instalar o *MySQL* utiliza-se o seguinte comando:

1. apt-get install mysql-server

Em seguida, digite o comando abaixo para garantir o acesso do *root* ao *MySQL*.

1. mysql -u root –p

Agora é possível acessar o *MySQL*.

1. 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

3. [PROGRAMAS CONSTRUÍDOS OU ADAPTADOS](#_Toc394320267)

## 3.1. INTRODUÇÃO

No projeto em questão serão utilizados os seguintes recursos de programação: linguagem C e *MySQL.*

## 3.2. PROGRAMA PRINCIPAL

O programa principal é a adaptação do código encontrado no seguinte endereço web:

* https://github.com/adafruit/Adafruit-Raspberry-Pi-Python-Code

Foram feitas alterações no programa:

* A inserção de um menu;
* A inserção de uma função para conectar com o banco de dados;
* Chamada interrupta da função *readDHT*;
* Comandos para armazenamento dos dados da temperatura e umidade na tabela criada no *MySQL*;

Para compilar digite:

gcc Adafruit\_DHT.c -l bcm2835 -std=gnu99 -o Adafruit\_DHT -lmysqlclient

Para executar digite:

sudo ./Adafruit\_DHT 11 4

### **3.2.1. MySQL**

Para criação de usuário, banco de dados, tabelas e manipulação do *MySQL* como um todo utilizou-se o seguinte tutorial (com alterações):

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

Para entrar no MySQL utiliza-se o seguinte comando:

* mysql –user=root –p
* Enter password: <senha aqui>

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

* create database *nomedobancodedados*;
* grant all on *nomedobancodedados*.\* to guest identified by “*senhadobancodedados*”;
* flush privileges;
* 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:

* 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:

* connect *nomedobancodedados*;

Criando a tabela:

* create table *nomedatabela* (*nomedacoluna* VARCHAR(40) NOT NULL, *nomedacoluna2* VARCHAR(40) NOT NULL, data\_hora 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:

* select \* from *nomedatabela*;

### **3.2.2. Inserir data e hora em C**

Foi utilizado a biblioteca *time.h*, inserindo a seguinte linha de comando:

* *const time\_t timer = time(NULL);*

Esta linha de comando retorna o dia, mês, ano e hora do sistema.

##

## 3.4. ESTRUTURA DO BANCO DE DADOS

O banco de dados consiste de uma tabela onde são armazenados os parâmetros aferidos, que são: Temperatura, Umidade, Hora e Data. Como Pode ser observado na tabela abaixo:

**Tabela 1 – Estrutura do banco de Dados**

|  |  |  |
| --- | --- | --- |
| Temperatura | Umidade | Hora e Data |
| Dados colhidos 0 | Dados colhidos 0 | Dados colhidos 0 |
| Dados colhidos 1 | Dados colhidos 1 | Dados colhidos 1 |
| Dados colhidos 2 | Dados colhidos 2 | Dados colhidos 2 |
| Dados colhidos 3 | Dados colhidos 3 | Dados colhidos 3 |
| ... | ... | ... |
| ... | ... | ... |
| Dados colhidos n | Dados colhidos n | Dados colhidos n |

Com estes dados é possível realizar consultas básicas e monitorar em tempo real a temperatura e umidade, para aplicações conforme desejada. Um exemplo, seria o monitoramento de temperatura e umidade aplicada em uma estufa, no qual é preciso manter um certo clima dentro da mesma, para que as plantas se proliferem adequadamente.

## 3.5. 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.

# **4.** **DESCRIÇÃO DO HARDWARE ADICIONAL NA GPIO**

## 4.1. INTRODUÇÃO

Para este projeto foi necessário acoplar o sensor de temperatura e umidade DHT11 na GPIO.

## 4.2. SENSOR DE TEMPERATURA E UMIDADE

O sensor utilizado neste projeto foi o DHT11 que mensura temperatura e umidade. Possui um sensor resistivo para a umidade e o sensor NTC para a temperatura. Suas

1. Precisão de ±5% para umidade relativa;
2. Precisão de 1ºC para a temperatura;
3. Range da umidade: 20 – 90%RH;
4. Range da temperatura: 0 – 50ºC;
5. Tempo de resposta mínimo: 6 segundos;

O usado no projeto já veio acoplado em uma placa de circuito impresso, no qual já contém a configuração necessária para seu funcionando, fornecendo três pinos de saídas para a conexão. São eles: o pino Vcc, GND e o de dados.

**Figura 11 - DHT11**



## 4.3. INSTALAÇÃO DO SENSOR DHT11 NA GPIO

 O primeiro passo para acoplar o DHT11 na *RaspberryPI* é verificar se a mesma não encontra-se alimentada, ou seja, ligada. A ligação deve ser feita com o computador desligado.

 Em seguida, o sensor disponibiliza três fios coloridos, um **preto**, um amarelo e um vermelho. Para ligar na GPIO, é preciso habilitar a GPIO 4 conforme instruímos no tutorial acima. O fio **preto** é ligado no GND nos pinos: 6, 9, 14, 20 e 25, o fio amarelo vai na GPI 4 que encontra-se no pino 7, já o fio vermelho é conectado na alimentação 5V, que situa-se no pino 2 ou 4, conforme ilustrado na figura 12.

**Figura 122 - Pinos GPIO.**

**REFERÊNCIAS BIBLIOGRÁFICAS**

ADA, L.. DHT Humidity Sensing on Raspberry Pi with GDocs Logging. **Adafruit Learning Systems.** Disponível em: <https://www.google.com.br/search?q=dht-humidity-sensing-on-raspberry-pi-with-gdocs-logging&ie=utf-8&oe=utf-8&aq=t&rls=org.mozilla:pt-BR:official&client=firefox-a&channel=sb&gfe\_rd=cr&ei=nWblU7SLKamX8QfX9YHYAQ#>. Acesso em: 08 ago. 2014.

ARLEI. Comandos Básicos. Disponível em: <http://wiki.ubuntu-br.org/ComandosBasicos>. Acesso em: 08 ago. 2014.

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.

**APÊNDICE A – PRINCIPAIS COMANDOS DO LINUX**

|  |  |
| --- | --- |
| **Comando** | **Descrição** |
| **sudo** | Permite um usuário executar comandos com as permissões de um “super-usuário” (root). |
| **sudo halt** | 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. |