**SISTEMAS EMBARCADOS**



**CAMPO MOURÃO**

**2014**

***Boris Hugo Westphal***

***Wellington Maidana***

**Tutorial Raspberry Pi**

**ITEAD NFC PN532**

BORIS HUGO WESTPHAL

WELLINGTON MAIDANA

**Tutorial Raspberry Pi**

**Itead NFC PN532**

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. [DESCRIÇÃO DO HARDWARE ADICIONAL NA GPIO](#_Toc394320275) 18

[3.1 INTRODUÇÃO](#_Toc394320276) 18

[3.2 MÓDULO ITEAD PN532 NFC](#_Toc394320277) 18

 3.2.1 Biblioteca ITEAD PN352 NFC.................................................................19

3.2.2. Biblioteca LIBNFC...................................................................................20

[3.3 INSTALAÇÃO DO SENSOR DHT11 NA GPIO](#_Toc394320278) 23

4. [PROGRAMAS CONSTRUÍDOS OU ADAPTADOS](#_Toc394320267) 23

[4.1 INTRODUÇÃO](#_Toc394320268) 23

[4.2 PROGRAMA PRINCIPAL](#_Toc394320269) 23

[4.2.1 MySQL](#_Toc394320270) 24

[4.2.2 Inserir data e hora em C](#_Toc394320271) 24

[4.4 ESTRUTURA DO BANCO DE DADOS](#_Toc394320273) 24

[4.5 CONCLUSÃO](#_Toc394320274) 25

REFERÊNCIAS BIBLIOGRÁFICAS XXVI

APÊNDICE A – PRINCIPAIS COMANDOS DO LINUX XXVII

APÊNDICE B – CODIGO LEITURA DO CARTÃO................................................XXIII

APÊNDICE C – CODIGO GRAVAÇÃO BANCO DE DADOS XXIV

**LISTA DE FIGURAS**

[**Figura 1:** Campo de download do sistema operacional... **Erro! Indicador não definido.**](#_Toc220300942)0

[**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. **Erro! Indicador não definido.**](#_Toc220300947)3

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

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

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

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

[**Figura 11:** Ativação módulo SPI.](#_Toc220300951) 21

[**Figura 12:** Dispositivos SPI.](#_Toc220300951) 22

[**Figura 13:** Comandos de testes.](#_Toc220300951) 22

[**Figura 14:** Módulo ITEAD NFC PN532 acoplado na GPIO](#_Toc220300952) 23

**LISTA DE TABELAS**

[**Tabela 1:** Estrutura do banco de dados 24](#_Toc220204480)

**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*

**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 do cartão de um sistema RFID 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

Conforme o crescimento na automatização de sistemas e processos, torna-se 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 clara o desenvolvimento de um sistema de monitoramento da leitura de um RFID. 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

Primeiramente será abordado sobre o sistema operacional (*SO*), apresentando suas principais características, além de descrever 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, será apresentado o hardware adicional na GPIO, o Itead NFC PN532, apresentando suas características e funções.

# **INSTALAÇÂO DO SISTEMA INICIAL**

## 2.1. INTRODUÇÃO

 Os sistemas operacionais são softwares que gerenciam computadores de maneira otimizada, constituído por vários dispositivos que precisam de informações adequadas para seu devido funcionamento. Os sistemas operacionais mais populares são: Windows, Linux e Mac OS X.

## 2.2 DESCRIÇÃO SUSCINTA DAS INTALAÇÔES

Para o desenvolvimento 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/

Ao acessar este site, será encontrado o campo representado na figura 1, onde 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**

Por fim, 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 por 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 para criar atalhos para estes comandos.

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 atalhos 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 🡪 Reiniciar o sistema.

Sempre que inicializado ou reinicializado, a *RaspberryPI* solicita um login e password. Para excluir 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 usa 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, tem a vantagem de ter o uso gratuito tanto para fins comerciais quanto particulares.

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. GRANT ALL ON \*.\* TO root'%' IDENTIFIED BY 'raspberry';

Agora é possível acessar o *MySQL*.

1. mysql -u root –p

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

1. g++ myClient.c -o cliente
2. ./cliente

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

## 3.1. INTRODUÇÃO

Para este projeto foi necessário acoplar a placa ITEAD com leitor RFID PN532 na GPIO. Foi utilizado módulo ITEAD PN532 NFC está equipado com pinos de linha dupla, que pode ser conectado diretamente ao Raspberry Pi via cabo de conexão.

A PN532 é o chip NFC mais popular, e é o que está incorporado em praticamente qualquer telefone ou dispositivo que faz NFC. Ele pode muito bem fazer tudo isso, como ler e escrever para etiquetas e cartões, comunicação com telefones (dizer para processamento de pagamentos), e "atuar" como uma tag NFC. Se você quiser fazer qualquer tipo de trabalho NFC embutido, este é o chip que você vai querer usar.

NFC (Near Field Communications) é um modo para dois dispositivos muito próximos uns dos outros para se comunicar. Como uma espécie de curto Bluetooth intervalo que não requer autenticação. É uma extensão de RFID, então qualquer coisa que você pode fazer com RFID que pode fazer com NFC. Você pode fazer mais coisas com NFC, bem como, comunicação bi-direcional com celulares, dentre outras. Realizamos algumas tags diferentes que funcionam muito bem com este chip. O PN532 também é muito flexível, você pode usar 3.3V TTL UART em qualquer taxa de transmissão, I2C ou SPI se comunicar com ele. Este chip também é fortemente apoiado por libnfc, basta ligar um cabo FTDI e usar o dispositivo de porta serial FTDI para se comunicar, isso permite que você faça comunicação usando qualquer computador Linux / Mac / Windows.

## 3.2. MÓDULO ITEAD PN532 NFC

Em primeiro lugar, antes de instalar a biblioteca ITEAD\_PN532\_NFC, precisamos modificar algumas das configurações na raspberry para fazer módulo SPI ativar automaticamente quando se ligado. Digite a pasta de configuração:

1. cd /etc/modprobe.d/

Abra o arquivo de configuração como superusuário com o comando:

1. sudo nano Raspi-blacklist.conf

Comente a linha “Blachlist gole-bcm2708” conforme esta abaixo, para que o módulo SPI esteja ativado ao ligar.

1. # Blachlist gole-bcm2708

 3.2.1. Biblioteca ITEAD PN352 NFC

Após realizar todas modificações anteriores, é necessário realizar o download da biblioteca ITEAD\_PN532\_NFC, disponibilizada pelo link:

* http://blog.iteadstudio.com/to-drive-itead-pn532-nfc-module-with-raspberry-pi/

Então, você deve descompactar a biblioteca ITEAD\_PN532\_NFC no desktop, por meio da imagem gráfica do sistema. Feita a descompactação no diretorio desktop, deve-se entrar na pasta ITEAD\_PN532\_NFC com o comando:

1. cd / home / pi / Desktop / ITEAD\_PN532\_NFC

Dentro deste diretório é necessário sua instalação através do comando:

1. make install

Assim, o banco de dados foi compilado e instalado. Enquanto o arquivo de projeto conter nfc.h, as funções na biblioteca podem ser chamadas. Já durante a compilação, a compilação é feita acrescentando “-I NFC” na linha de comando.

3.2.2. Biblioteca LIBNFC

O módulo Itead PN532 é conectado via barramento SPI com a raspberry pi, portanto, o modo de funcionamento do módulo PN532 deve ser definido para o modo SPI, como mostrado abaixo:

 SET0-->L

 SET1-->H

Agora instale o pacote de software necessário:

1. sudo apt-get update
2. sudo apt-get install libusb-dev libpcsclite-dev

 Faça o download e posteriormente descompacte o pacote de código fonte libnfc, realizando os seguintes comandos:

1. cd ~
2. wget http://dl.bintray.com/nfc-tools/sources/libnfc-1.7.1.tar.bz2
3. tar -xf libnfc-1.7.1.tar.bz2

 Então instale e compile usando:

1. cd libnfc-1.7.1
2. ./configure --prefix=/usr --sysconfdir=/etc
3. make
4. sudo make install

 Agora é necessário a modificação do arquivo de configuração:

1. cd /etc
2. sudo mkdir nfc
3. sudo nano /etc/nfc/libnfc.conf

 Copie o seguinte conteúdo ao arquivo /etc/nfc/libnfc.conf:

# Allow device auto-detection (default: true)

# Note: if this auto-detection is disabled, user has to set manually a device

# configuration using file or environment variable

allow\_autoscan = true

# Allow intrusive auto-detection (default: false)

# Warning: intrusive auto-detection can seriously disturb other devices

# This option is not recommended, user should prefer to add manually his device.

allow\_intrusive\_scan = false

# Set log level (default: error)

# Valid log levels are (in order of verbosity): 0 (none), 1 (error), 2 (info), 3 (debug)

# Note: if you compiled with --enable-debug option, the default log level is "debug"

log\_level = 1

# Manually set default device (no default)

# To set a default device, you must set both name and connstring for your device

# Note: if autoscan is enabled, default device will be the first device available in device list.

device.name = "Itead\_PN532\_SPI"

device.connstring = "pn532\_spi:/dev/spidev0.0:500000"

 Agora é necessário realizar a ativação do módulo SPI, fazendo as seguintes modificações:

1. sudo nano /etc/modprobe.d/raspi-blacklist.conf

 Ao entrar no arquivo “rapi-blacklist.conf”, adicione # na linha em que esta escrito “blacklist spi-bcm2708”, ficando conforme a figura abaixo:



**Figura 11 – Ativação do módulo SPI.**

Agora reinicie o sistema realizando o comando “reboot”. Você pode ver dois dispositivos SPI após a reinicialização conforme a figura abaixo:



**Figura 12 – Dispositivos SPI.**

 Finalizando a instalação, é necessário realizar um comando de teste que verifique se a instalação foi devidamente instalada. Deve ser feito certos comandos e posteriormente passar o cartão para conferencia conforme a figura abaixo:



**Figura 13 – Comandos de testes.**

 Caso ocorra algum problema, siga o tutorial presente no link abaixo:

* <http://blog.iteadstudio.com/raspberry-pi-drives-itead-pn532-nfc-module-with-libnfc/>

## 3.3. INSTALAÇÃO DA PLACA ITEAD\_PN532\_NFC NA GPIO

 O primeiro passo para acoplar o ITEAD\_PN532\_NFC na *RaspberryPI* é verificar se a mesma não encontra-se alimentada, ou seja, ligada. A ligação deve ser feita com o computador desligado.

 Como o módulo Itead PN532 é personalizado para Raspberry Pi, ele pode ser conectado diretamente ao Raspberry Pi através do cabo adaptador como mostrado na imagem abaixo



**Figura 14 – Módulo ITEAD NFC PN532 acoplado na GPIO.**

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

## 4.1. INTRODUÇÃO

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

## 4.2. PROGRAMA PRINCIPAL

O programa principal é a adaptação do código readID disponível pela biblioteca ITEAD\_PM532\_NFC instalada anteriormente. Foi encontrado um problema muito raro ao se trabalhar com o Linux, ouve um erro de compilação em relação as bibliotecas, ou seja, a biblioteca NFC não funciona quando trabalhado com a biblioteca MySQL. Então houve a necessidade de compilá-las separadamente, chamando uma na outra quando preciso. O apêndice B mostra as modificações feitas no arquivo “readID.c”, já o apêndice C mostra um novo arquivo pra armazenamento de leitura do cartão na tabela criada no MySQL.

### 4.2.1. MySQL

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

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

### 4.2.2. Inserir data e hora em C

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

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

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

##

## 4.4. ESTRUTURA DO BANCO DE DADOS

O banco de dados consiste de uma tabela onde são armazenados os parâmetros necessários de acordo com o usuário, neste caso são: Número do Cartão, Data, Hora. Como Pode ser observado na tabela abaixo:

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

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Número do Cartão | Data | Hora |
| Nº de leitura 0 | Nº Cartão 0 | Data 0 | Time 0 |
| Nº de leitura 1 | Nº Cartão 1 | Data 1 | Time 1 |
| Nº de leitura 2 | Nº Cartão 2 | Data 2 | Time 2 |
| Nº de leitura 3 | Nº Cartão 3 | Data 3 | Time 3 |
| ... | ... | ... | ... |
| ... | ... | ... | ... |
| Nº de leitura n | Nº Cartão n | Data n | Time 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.

## 4.5. 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.

**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. |

**APÊNDICE B – CODIGO LEITURA DO CARTÃO**

#include "nfc.h"

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define NFC\_DEMO\_DEBUG 1

void Leitura (void);

void Leitura(void) { // Realiza a Leitura do cartão

int sair;

char data [20];

char hora[20];

char salva[200];

char buffer[20];

uint32\_t versiondata;

uint32\_t id=0;

FILE \*fp;

#ifdef NFC\_DEMO\_DEBUG

printf("Iniciando Leitura!\n");

#endif

begin();

versiondata = getFirmwareVersion();

if (! versiondata)

{

#ifdef NFC\_DEMO\_DEBUG

printf("Didn't find PN53x board\n");

#endif

while (1); // halt

}

#ifdef NFC\_DEMO\_DEBUG

// Got ok data, print it out!

printf("Found chip PN5");

printf("%x\n",((versiondata>>24) & 0xFF));

printf("Firmware ver. ");

printf("%d",((versiondata>>16) & 0xFF));

printf(".");

printf("%d\n",((versiondata>>8) & 0xFF));

printf("Supports ");

printf("%x\n",(versiondata & 0xFF));

#endif

// configure board to read RFID tags and cards

SAMConfig();

while (id==0)

 {

// look for MiFare type cards

id = readPassiveTargetID(PN532\_MIFARE\_ISO14443A);

if (id != 0)

 {

 #ifdef NFC\_DEMO\_DEBUG

 printf("Read card #%d\n",id);

// Capturando data e hora do sistema atraves da biblioteca time.h

 time\_t t= time(NULL);

 struct tm tm = \*localtime(&t);

 sprintf(data,"%d-%d-%d", tm.tm\_year + 1900, tm.tm\_mon+1, tm.tm\_mday);

 sprintf(hora,"%d:%d:%d", tm.tm\_hour, tm.tm\_min, tm.tm\_sec);

 //printf("data : %s", data);

 sprintf(salva,"%d %s %s",id,data,hora);

 printf("Dados na String: %s",salva);

 fp=fopen("temp.log","wb");

 if (fp!=NULL)

{

 fprintf(fp,"%s",salva);

}

 else

{

 printf("Erro ao abrir o arquivo temp.log\n");

}

 fclose(fp);

 system("./gravacao");

 #endif

 }

 }

}

void main(void){

int operacao;

do{

 printf("\n Operacao:\n "); // Menu de operações

 printf("\n 1-Leitor de Cartao:");

 // printf("\n 2-Mostrar leituras:");

 printf("\n 0 - sair:");

 printf("\n Digite a operacao: ");

 scanf("%d",&operacao);

if(operacao == 1){

// Chama a função que realiza a leitura do cartão

Leitura();

}

}while(operacao!=0);

}

APÊNDICE C – CODIGO GRAVAÇÃO BANCO DE DADO

#include <stdio.h>

#include <stdlib.h>

#include <mysql/mysql.h>

MYSQL conexao;

void main(void)

{

FILE \*fp;

char buffer[300],temp[3][100];

mysql\_init(&conexao);

 if (mysql\_real\_connect(&conexao,"localhost","root","rasp","rfid", 0, NULL, 0))

 {

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

 fp=fopen("temp.log","rb");

 if (fp!=NULL)

 {

 fscanf(fp,"%s %s %s",temp[0],temp[1],temp[2]);

 printf("%s %s %s",temp[0],temp[2],temp[2]);

 sprintf(buffer,"INSERT INTO tabela(id,ncartao,data,hora) values('','%s','%s','%s');",temp[0],temp[1],temp[2]);

 }

 else

 {

 printf("Erro ao abrir o arquivo temp.log\n");

 }

 fclose(fp);

 mysql\_query(&conexao,buffer);

 mysql\_close(&conexao);

 }

 else

 {

 printf("\n Erro conexao!! \n");

 }

}