

PROJETO DE INTERFACEAMENTO DO RASPBERRY PI E CI PCF8591

O projeto consiste no desenvolvimento de um interfaceamento entre um Raspberry Pi e um CI PCF8591 que possui quatro conversores A/D e um D/A. O projeto foi desenvolvido para permitir o controle de plantas externas via acesso remoto, monitorando dados e respondendo via PWM ou via Conversor A/D. Desta forma foram realizados estudos sobre diversas áreas: Raspberry Pi, verificando suas especificações e recursos disponíveis (PWM, I2C); I2C, para realizar a comunicação entre Raspberry e o conversor; Sockets, para realizar a comunicação remota.

Tutorial de Configuração

```
apt-get update           //para atualizar a data.
apt-get install mc       //Instalação do mcedit (editor de texto)
```

→ Para alterar configurações do profile pi:

```
cd /root/etc             //entra no diretório /root/etc
mcedit profile           //através do mcedit abre o arquivo profile para edição
```

No final do arquivo insira:

```
alias ll='ls -l --coloe=auto'      #insere o atalho do comando ll
alias la='ls -la --coloe=auto'     #insere o atalho do comando la
export EDITOR=mcedit              #configura mcedit como editor padrão
export HISTSIZE=1000              #configura o tamanho max. do history
export HISTFILESIZE=1000
```

→ Para alterar configurações do root:

```
cd /root                 //entra no diretório /root
mcedit .bashrc           //através do mcedit abre o arquivo .bashrc para edição
```

No final do arquivo insira:

```
alias ll='ls -l --coloe=auto'      #insere o atalho do comando ll
alias la='ls -la --coloe=auto'     #insere o atalho do comando la
export EDITOR=mcedit              #configura mcedit como editor padrão
```

```
export HISTSIZE=1000    #configura o tamanho max. do history
export HISTFILESIZE=1000
```

→ Para configurar tela cheia:

```
cd /boot                //entra no diretório /boot
mcedit config.txt       //abre o arquivo para edição

Descomente tirando o caracter '#' antes de: "disable_overscan=1"
```

→ Para retirar a necessidade de login quando iniciado o raspberry execute:

```
mcedit /etc/inittab
```

Dentro deste 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
```

→ Tanto para configurar o root e pi, quanto configurar tela cheia e login deve-se dar reboot para aplicar as mudanças

Configurações e Pacotes executados durante o projeto

→ Para configurar e testar i2C

Abra as duas páginas para auxiliar a conexão e configuração do I2C com o PCF: <http://blog.chrysome.net/search/label/Raspberry%20Pi> e o diagrama da GPIO do Raspberry: <https://projects.drogon.net/raspberry-pi/wiringpi/pins/>

Instale a biblioteca i2c-dev através do comando:

```
apt-get install i2c-tools
```

Para utilizar a biblioteca i2c-tools deve-se inserir no código: *#include <linux/i2c-dev.h>*

Através do comando:

```
mcedit /etc/modprobe.d/raspi-blacklist.conf
```

comente as duas linhas: *spi* e *i2c*

Agora com a placa do CI PCF8591 conectada no Raspberry (verifique a correta conexão avaliando o esquemático da PCF e o diagrama da GPIO) habilite e verifique o endereço do dispositivo através de:

```
gpio load i2c
i2cdetect 1
```

Para realizar testes via linha de comando, siga os passos descritos em: <http://blog.chrysome.net/2012/12/i2c-analog-to-digital-converter.html>.

Para alterar o baudrate do I2C, edite através do mcedit o seguinte arquivo de configuração (max. de 400kHz): */etc/modprobe.d/i2c.conf*

Altere o valor especificado na linha: *options i2c_bcm2708 baudrate=100000*

As configurações somente serão aplicadas quando o Raspberry for reiniciado.

→ Para enviar e receber dados via socket

Através do código exemplo de um servidor e cliente via socket disponível em: http://cs.smith.edu/dftwiki/index.php/Tutorial:_Client/Server_on_the_Raspberry_Pi

Elaborados em linguagem c++. Caso a versão do Linux não possua o g++ utilize o comando:

```
apt-get install g++
```

Utilizando o Cap. 12 do livro “Sockets Linux” de Maicon Melo Alves é possível alterar o código para múltiplos clientes.

→ Testando os códigos desenvolvidos:

O projeto final possui dois códigos fontes: *myServer2* e *myClient*.

O código *myServer2* é o responsável por iniciar o servidor e realizar leituras do CI, ou seja, deverá ser compilado e executado no Raspberry Pi com o CI conectado. Quando em execução, o programa aguardará a uma requisição de conexão de algum cliente.

O código *myClient* é o responsável pelo recebimento e tratamento dos dados quando conectado ao servidor, portanto, o cliente poderá ser executado na própria máquina do servidor ou em outras máquinas com Linux e conectados na rede. Antes de executar,

deve-se verificar o IP do servidor (utilize o comando ifconfig no Raspberry) e se necessário alterar no campo *serverIP[]* no código do cliente.

Após a compilação e execução, o cliente estabelecerá conexão com o servidor e iniciará o menu para iteração com o usuário.

Configurações Adicionais

→ Para instalar a biblioteca bcm2835:

```
wget www.lt38c.hturbo.com/bcm2835-1.33.tar.gz -o /tmp/bcm2835.tar.gz
```

```
tar -zxvf bcm2835-1.33.tar.gz
```

```
cd bcm2835-1.33
```

```
./configure
```

```
make
```

```
make check
```

```
make install
```

```
cp -R /tmp/bcm2835-1.33/examples/ /root/bcm2835
```

-- Para compilar os programas dentro da pasta examples utilize: gcc -o blink blink.c -lbcm2835

→ Para instalar o servidor apache, php e mysql: (não é usado neste projeto)

```
apt-get install apache2 php5 php5-mysql mysql-server
```

```
sh -c 'echo "<?php phpinfo(); ?>" > /var/www/phpteste.php'
```

Para testar se a instalação foi bem sucedida, entre no modo gráfico e usando o Midori acesse o phpteste.php

→ Para instalar a biblioteca wiringPi:

Usando do modo gráfico, acesse o link: <http://wiringpi.com/download-and-install/>

Execute os passos do Plano B, após baixado mova a pasta para o local desejado e descompacte com:

```
tar xzf wiringPi-f18c8f7.tar.gz
```

→ Para instalar a biblioteca ncurses-dev:

```
apt-get install ncurses-dev
```

→ Comando Básicos

raspi-config	//comando para entrar no menu de configuração do raspberry
	//por exemplo: expandir a memória, ou configurar o teclado
clear	//limpa a tela
reboot	//reinicia o sistema
startx	//inicia modo gráfico (apenas no usuário pi)
sudo su	//entra no superuser
exit	//sai do superuser
ifconfig	//para verificar as redes conectadas
halt	//desligar o raspberry