

## Tutorial GPIO

Programação de Periféricos – CC – FACIN – PUCRS

### Executando o Linux na Raspberry Pi:

Colocar o cartão na Raspberry Pi, conectá-la a rede usando um cabo ethernet e conectá-la a alimentação. A placa deverá ligar e em alguns segundos estará executando o Linux com um servidor SSH habilitado. Para obter o IP que foi designado a placa, executar o seguinte comando (lembrar de colocar o número correto do grupo):

```
nmap -sP 10.32.143.0/24
```

Procurar o IP atribuído baseado no MAC address da placa.

Conectar como root ao ip designado a placa (usar a senha gerada anteriormente):

```
ssh root@_IP_DA_PLACA_
```

Se tudo deu certo, você deve ter acesso ao Linux embarcado na Raspberry Pi!

[Nunca esqueça de antes de desconectar a placa da fonte de alimentação, desligar o sistema através do comando:](#)

```
halt
```

Caso isso não seja feito, o sistema de arquivos poderá ser corrompido e todo processo de geração do Linux deverá ser reelaborado.

### Manual de Acesso aos pinos de GPIO:

Abaixo serão listados comandos que deverão ser usados em uma próxima etapa para fazer acesso aos pinos de entrada e saída. Praticamente tudo no Linux é um “arquivo”. Isso inclui os pinos de GPIO. Para habilitar o acesso a algum pino de GPIO, identifique isso para o sistema escrevendo o número do pino no arquivo “/sys/class/gpio/export”. Por exemplo, para acessar o pino 18:

```
echo "18" > /sys/class/gpio/export
```

Esse comando habilita o pino 18 de GPIO. Os números de acesso aos pinos são definidos conforme a figura abaixo:

Pin 1	Pin 2	Pin 1	Pin 2	Pin 1	Pin 2
3.3 V	5 V	3.3 V	5 V		
GPIO 0	5 V	GPIO 2	5 V		
GPIO 1	GND	GPIO 3	GND		
GPIO 4	GPIO14	GPIO 4	GPIO14		
GND	GPIO15	GND	GPIO15		
GPIO17	GPIO18	GPIO17	GPIO18		
GPIO21	GND	GPIO27	GND		
GPIO22	GPIO23	GPIO22	GPIO23		
3.3 V	GPIO24	3.3 V	GPIO24		
GPIO10	GND	GPIO10	GND		
GPIO 9	GPIO25	GPIO 9	GPIO25		
GPIO11	GPIO 8	GPIO11	GPIO 8		
GND	GPIO 7	GND	GPIO 7		
Pin 25	Pin 26	Pin 25	Pin 26		Pin 25

Raspberry Pi (Rev. 1)      Raspberry Pi (Rev. 2)

Lembrar que estamos usando a Rev. 2 da Raspberry Pi.

Uma vez que o pino esteja habilitado, uma pasta será criada em `/sys/class/gpio` de acordo com o número do pino. No exemplo acima, será a pasta `gpio18`. Devemos então permitir escrita nos arquivos dessa pasta:

```
chmod -R 777 /sys/class/gpio/gpio18
```

O próximo passo é definir o pino como de entrada ou saída. Para tanto escrevemos essa informação no arquivo `direction` do diretório criado para o pino:

```
Para saída: echo "out" > /sys/class/gpio/gpio18/direction
```

```
Para entrada: echo "in" > /sys/class/gpio/gpio18/direction
```

O próximo passo é atribuir um valor lógico para pinos de saída ou ler valores de pinos de entrada. Para tanto, basta acessarmos o arquivo `value` criado na pasta do pino.

Para escrever um valor em pinos de saída:

```
Para escrever 0: echo "0" > /sys/class/gpio/gpio18/value
```

```
Para escrever 1: echo "1" > /sys/class/gpio/gpio18/value
```

Para ler um valor em pinos de entrada:

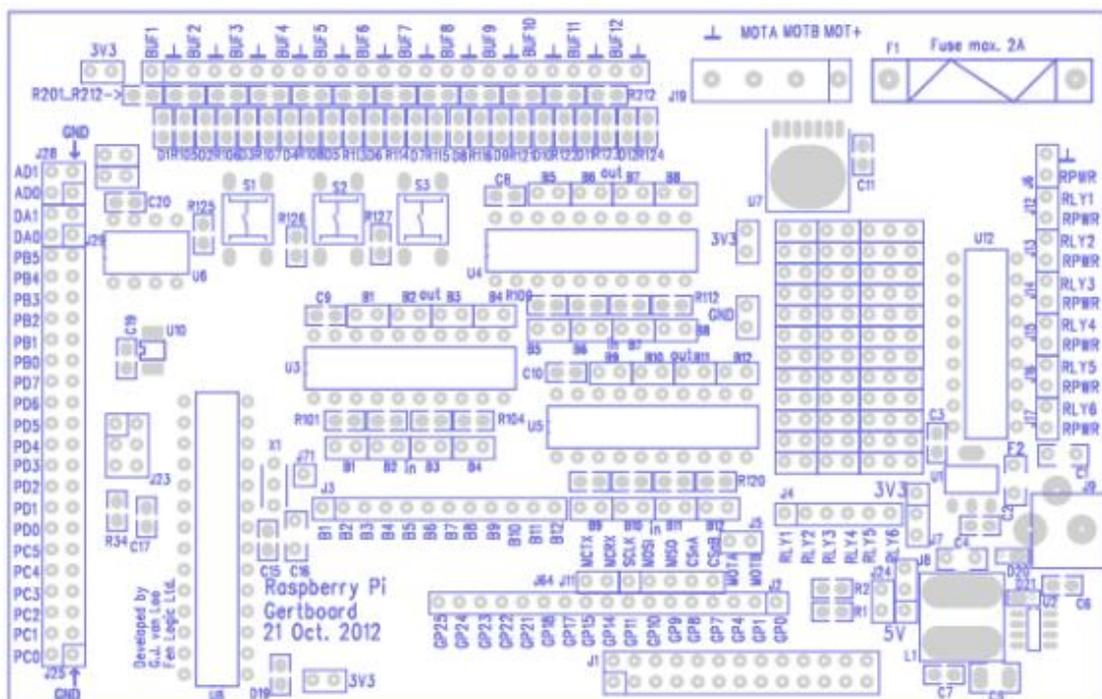
```
cat /sys/class/gpio/gpio18/value
```

Finalmente, para desconectar um pino, escreva o número do pino no arquivo `unexport` da pasta `/sys/class/gpio`

```
echo "18" > /sys/class/gpio/unexport
```

## Manual de acesso aos pinos da gertboard:

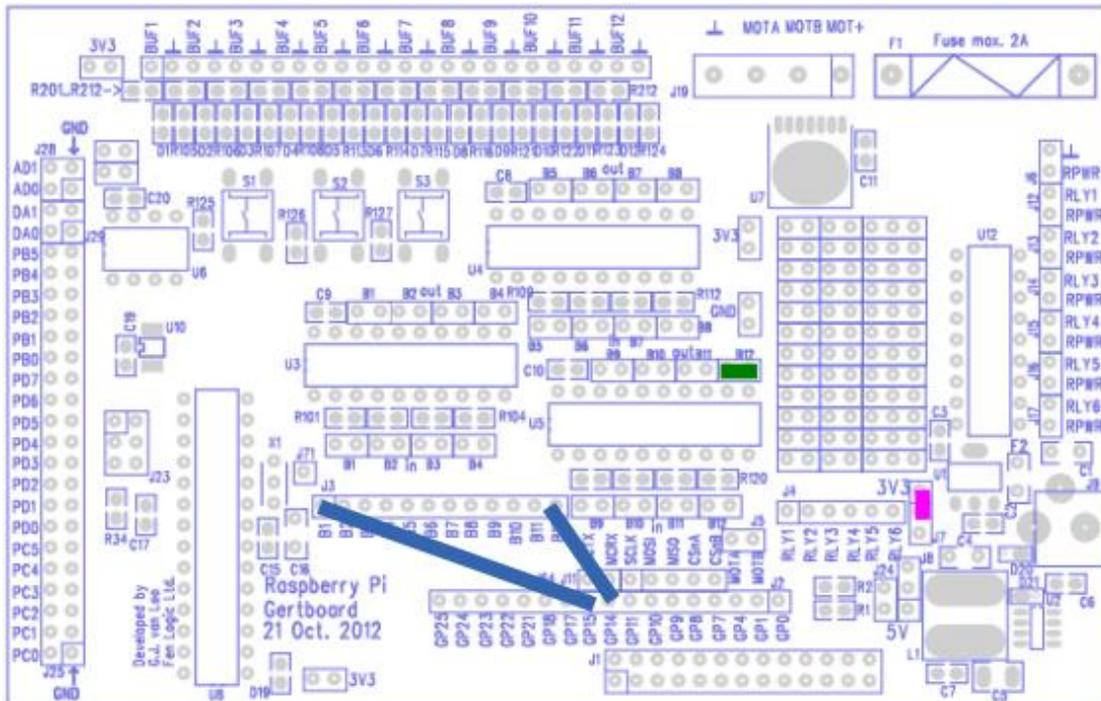
Todos os pinos de GPIO da Raspberry Pi estão acessíveis na Gertboard, conforme destacado na figura abaixo:



Entretanto, para utilizar as portas de IO da Gertboard (que proveem funcionalidades de buffers, leds e botões, entre outras), algumas configurações são necessárias. A Gertboard possui 12 portas básicas de IO.

Neste exemplo, o pino GP15 da Raspberry Pi está ligado ao pino 1 da Gertboard e o pino GP14 ao pino 12. Além disso, é necessário que seja definida a direção do pino da Raspberry Pi em

relação ao pino da Gertboard. Para tanto, devem ser usados jumpers para selecionar cada pino da Gertboard (B1, B2, B3, etc.) como entrada (input) ou saída (output). A figura abaixo mostra um exemplo dos pinos B1 e B12 sendo definidos como de saída. Para ligar a placa, conectar um jumper a alimentação de 3.3V, conforme a figura demonstra, e conectar a Gertboard a Raspberry Pi.



### Exemplos:

Executar os seguintes exemplos implementados em shell script (Disponíveis no Moodle):

#### botao.sh:

```
#!/bin/sh
echo "15" > /sys/class/gpio/export
echo "14" > /sys/class/gpio/export
chmod -R 777 /sys/class/gpio/gpio15
chmod -R 777 /sys/class/gpio/gpio14

echo "out" > /sys/class/gpio/gpio14/direction
while :
do
echo "out" > /sys/class/gpio/gpio15/direction
echo "1" > /sys/class/gpio/gpio15/value
echo "in" > /sys/class/gpio/gpio15/direction
TMP=`cat /sys/class/gpio/gpio15/value`
if [ "$TMP" == "0" ]; then
echo "0" > /sys/class/gpio/gpio14/value
else
echo "1" > /sys/class/gpio/gpio14/value
fi
done
```

Esse exemplo utiliza um led no GPIO 14 e um botão no GPIO 15. O procedimento é um laço

infinito que verifica o valor do botão e faz o seguinte: caso seja 0 apaga o led; caso seja 1 acende o led. Ou seja, se o botão estiver apertado o led será aceso. Se o botão não estiver apertado o led será apagado.

### **blink.sh:**

```
#!/bin/sh
echo "18" > /sys/class/gpio/export
echo "17" > /sys/class/gpio/export
echo "27" > /sys/class/gpio/export
chmod -R 777 /sys/class/gpio/gpio18
chmod -R 777 /sys/class/gpio/gpio17
chmod -R 777 /sys/class/gpio/gpio27
echo "out" > /sys/class/gpio/gpio18/direction
echo "out" > /sys/class/gpio/gpio17/direction
echo "out" > /sys/class/gpio/gpio27/direction
delay=100000
while :
do
for count in `seq 1 10`
do
echo "0" > /sys/class/gpio/gpio18/value
echo "0" > /sys/class/gpio/gpio17/value
echo "0" > /sys/class/gpio/gpio27/value
usleep $delay
echo "1" > /sys/class/gpio/gpio18/value
echo "1" > /sys/class/gpio/gpio17/value
echo "1" > /sys/class/gpio/gpio27/value
usleep $delay
done
delay=`expr $delay + 100000`
echo $delay
done
```

Esse exemplo utiliza três leds, nos GPIOs 17, 18 e 27. O procedimento é um laço infinito que pisca os leds com um período que começa em 100 ms e é incrementado em 100 ms a cada 10 repetições.

### **Tarefas:**

1. Entender os códigos fornecidos;
2. Implementar um procedimento que realize a funcionalidade de uma sinaleira. Para tanto serão usados três leds, mapeados nos GPIOs 17, 18 e 27 e um botão mapeado no GPIO 22. O led verde deverá ficar aceso por 10 segundos. Então o led amarelo deverá acender e o verde apagar. Esse estado deve manter-se por 2 segundos. Após isso, o led amarelo deve apagar e o led vermelho deve acender. Esse estado deve manter-se por 8 segundos. Então, o led vermelho deve ser apagado e o verde aceso novamente por 10 segundos e assim por diante. Caso o botão seja pressionado enquanto o led verde estiver aceso, o sistema deverá imediatamente desligar o led verde e ligar o amarelo por 2 segundos, depois desligar o amarelo e ligar o vermelho por 8 segundos. Após isso o led vermelho