

Controlando GPI no Console

Transcrição

Neste capítulo vamos dar continuidade na interação com o GPIO. Com o circuito ainda montado, e com a conexão remota ao Raspberry Pi, executamos o seguinte comando:

```
gpio
```

O comando `gpio` é a base de tudo, com esse comando, seguido dos parâmetros, iremos ter acesso direto à nossa interface. Por exemplo:

```
gpio readall
```

```
pi@r01:~$ gpio readall
```

BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM
		3.3v			1	2		5v		
2	8	SDA.1	IN	1	3	4		5V		
3	9	SCL.1	IN	1	5	6		0v		
4	7	GPIO.7	OUT	1	7	8	1	ALTO	15	14
		0v			9	10	1	ALTO	16	15
17	0	GPIO.0	IN	0	11	12	0	IN	1	18
27	2	GPIO.2	IN	0	13	14		0v		
22	3	GPIO.3	IN	0	15	16	0	IN	4	23
		3.3v			17	18	0	IN	5	24
10	12	MOSI	IN	0	19	20		0v		
9	13	MISO	IN	0	21	22	0	IN	6	25
11	14	SCLK	IN	0	23	24	1	IN	10	8
		0v			25	26	1	IN	11	7

Esse comando mostra o número do GPIO, do pino, se a porta está em *IN/OUT*, se está ligada ou desligada (1 ou 0, respectivamente). É uma forma de visualizar a nossa interface.

Definindo o estado de uma porta

No capítulo passado, nós já vimos como se define o estado de uma porta (entrada ou saída). Por exemplo, para definir o estado do pino 7 como **saída**, executamos:

```
gpio mode 7 out
```

Para ligar o LED, precisamos colocar a porta (o pino) em alta (ligá-lo). Na linha de comando, em alta é representado pelo número **1**:

```
gpio write 7 1
```

E para desligar o LED, colocamos a porta em baixa. Na linha de comando, em baixa é representado pelo número **0**:

```
gpio write 7 0
```

Podemos fazer isso com qualquer porta, não precisa necessariamente estar com o LED. Para saber o estado de uma porta, usamos o comando:

```
gpio read 7
```

E o retorno será **0** para **porta em baixa** e **1** para **porta em alta**.

Com isso conseguimos interagir com o GPIO sem a necessidade de uma linguagem de programação no meio, diretamente da linha de comando.