GPIO Control from Command Line

Jan 2017

When creating projects using the GPIO pins on a Raspberry Pi, you can control all of the GPIO functions from Python. However, after attaching some hardware to the Raspberry Pi, it can be nice to toggle the GPIO pins to check functionality (especially handy if you are controlling something visual like an LED or a motor).

Install WiringPi

WiringPi contains the command line utility gpio that we can use to control our GPIO pins. To install WiringPi, use the following command.

```
sudo apt-get install wiringpi
```

Once installed, you will have access to the gpio utility.

As a quick test, from the command line type:

```
gpio -v
```

This will display the version of WiringPi you have installed.

```
gpio version: 2.32
```

Copyright (c) 2012-2015 Gordon Henderson

This is free software with ABSOLUTELY NO WARRANTY.

For details type: gpio -warranty

Unable to determine hardware version. I see: Hardware : BCM2835

expecting BCM2708 or BCM2709.

If this is a genuine Raspberry Pi then please report this to projects@drogon.net. If this is not a Raspberry Pi then you are on your own as wiringPi is designed to support the Raspberry Pi ONLY.

Another test, and something you can use a lot to check the status of the gpio is the </color red>readall</color> switch.

```
gpio readall
```

This will give an output similar to below (This output is from a Pi Zero v1.3)

BCM wPi Name Mode V Physical V Mode Name wPi BCM	+	+	+		+	-+	+-Pi Zero+	+ -		+		+
	E	BCM	wPi	Name	Mode	V	Physical V	Mode	Name	wPi	BCM	1
2 8 SDA.1 IN 1 3 4 5V	+	+	+		+	-+	++	+		+		+
3 9 SCL.1 IN 1 5 6 0v	1			3.3v	I		1 2		5v			1
4 7 GPIO. 7 IN 1 7 8 1 ALTO TXD 15 14	1	2	8	SDA.1	IN	1	3 4		5V			1
	ı	3	9	SCL.1	IN	1	5 6		0v			1
17 0 GPIO. 0 IN 0 11 12 0 OUT GPIO. 1 1 18 27 2 GPIO. 2 IN 0 13 14		4	7	GPIO. 7	IN	1	7 8 1	ALT0	TxD	15	14	
27 2 GPIO. 2 IN 0 13 14 0v				0v	l	1	9 10 1	ALT0	RxD	16	15	
22 3 GPIO. 3 IN 0 15 16 0 OUT GPIO. 4 4 23 		17	0	GPIO. 0	IN	0	11 12 0	0UT	GPIO. 1	1	18	
3.3v 17 18 1 OUT GPIO. 5 5 24		27	2	GPIO. 2	IN	0	13 14		0v			
		22	3	GPIO. 3	IN	0	15 16 0	0UT	GPIO. 4	4	23	1
10 12 MOST TN 0 10 10				3.3v			17 18 1	0UT	GPIO. 5	5	24	
10 12 11031 111 0 13 20 00		10	12	MOSI	IN	0	19 20		0v			

Last update: 2023/03/09 22:35 gpio control from command line http://cameraangle.co.uk/doku.php?id=gpio control from command line

		9	13	MISO	IN	0	21	Ш	22	0	OUT		GPIO. 6	6	1	25	
- 1		11	14	SCLK	IN	0	23	-11	24	1	IN		CE0	10	-	8	1
		- 1		0v			25	\Box	26	1	IN		CE1	11		7	
		0	30	SDA.0	IN	1	27	\Box	28	1	IN		SCL.0	31		1	
		5	21	GPI0.21	0UT	1	29	-11	30		1		0v				1
		6	22	GPI0.22	0UT	0	31	-11	32	0	IN		GPI0.26	26		12	1
		13	23	GPI0.23	0UT	1	33	-11	34		1		0v				1
- 1		19	24	GPI0.24	0UT	0	35	Ш	36	0	OUT		GPI0.27	27		16	
- 1		26	25	GPI0.25	0UT	0	37	Ш	38	0	OUT		GPI0.28	28		20	
- 1		- 1		0v			39	Ш	40	1	OUT		GPI0.29	29		21	
+		+	+-	+		-+	+	-++-		+	+	+		+	+		+
	В	CM	wPi	Name	Mode	V	Ph	ysic	al	V	Mode	e	Name	wPi	-	BCM	
+		+	+-	+		-+	+-Pi	Zer	0	+	+	+		+	+		-+

It can be seen that this Pi has GPIO set to OUT and some to IN. I am using the Pi as a clock, so these are the lines going to my LED display.

Using gpio command

Before looking at the commands, there is a switch I want to talk about, this is -g. If you use the -g switch you are operating on BCM numbers, not PIN numbers. I always use BCM numbers, but some people prefer pin numbers.

To set the function of the GPIO pin, use gpio -g mode

```
using BCM number
gpio -g mode 18 out - Sets GPI018 to OUTput mode (GPI018 is physical pin 12)
using pin instead of bcm
gpio mode 12 out - Sets GPI018 to OUTput mode (GPI018 is physical pin 12)
Modes are In, Out, PWM, Up, Down and Tri.
```

Set GPIO pin to 1 or 0 $\,$

```
gpio -g write 18 0 - sets GPI018 to a zero.
gpio -g write 18 1 - sets GPI018 to a one.
```

From:

http://cameraangle.co.uk/ - WalkerWiki - wiki.alanwalker.uk

Permanent link:

 $http://camera angle.co.uk/doku.php?id=gpio_control_from_command_line$

Last update: 2023/03/09 22:35



http://cameraangle.co.uk/ Printed on 2024/05/23 06:03