

# Jarvis de JB...



Mon jarvis est composé de:

- 2 leds pour les yeux pour me dire qu'il attend un ordre suite au magic word et aussi pour me faire un clin d'oeil...
- 1 pour le nez, cette dernière doit rester allumée tant que jarvis est en fonctionnement.

Je suis équipé d'un Raspberry pi III mais tout cela fonctionne aussi sur le II.

Mon micro est incorporé dans le haut parleur c'est sa soucoupe volante lol, avec un Jabra410 voir ici pour l'acheter: <http://www.onedirect.fr/produits/gn-netcom-jabra/jabra-speak-410>. Il est super sensible même de loin et la qualité sonore est au top !

- Lorsque jarvis est lancé le nez est éclairé.
- Lorsque jarvis est arrêté le nez est éteint.
- Lorsqu'il a entendu le magic word ses yeux clignotent.
- Lorsque le temps d'écoute est fini ses yeux s'éteignent.
- A la demande, il sait me faire le clown ou des clins d'œil.

# Fonctionnement du gpio:

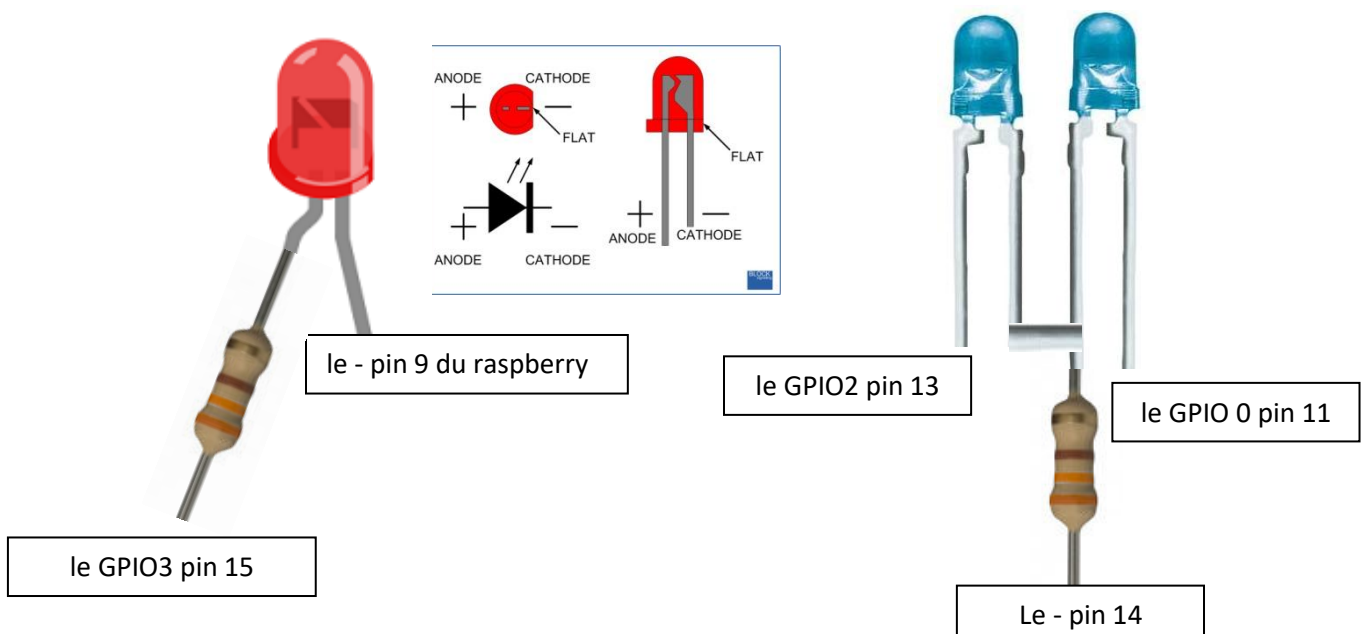


Raspberry Pi 3 Model B (J8 Header)					
GPIO#	NAME			NAME	GPIO#
	3.3 VDC Power	1		5.0 VDC Power	2
8	GPIO 8 SDA1 (I2C)	3		5.0 VDC Power	4
9	GPIO 9 SCL1 (I2C)	5		Ground	6
7	GPIO 7 GPCLK0	7		GPIO 15 TxD (UART)	15
	Ground	9		GPIO 16 RxT (UART)	16
0	GPIO 0	11		GPIO 1 PCM_CLK/PWM0	1
2	GPIO 2	13		Ground	14
3	GPIO 3	15		GPIO 4	4
	3.3 VDC Power	17		GPIO 5	5
12	GPIO 12 MOSI (SPI)	19		Ground	20
13	GPIO 13 MISO (SPI)	21		GPIO 6	6
14	GPIO 14 SCLK (SPI)	23		GPIO 10 CE0 (SPI)	10
	Ground	25		GPIO 11 CE1 (SPI)	11
30	SDA0 (I2C ID EEPROM)	27		SCL0 (I2C ID EEPROM)	31
21	GPIO 21 GPCLK1	29		Ground	30
22	GPIO 22 GPCLK2	31		GPIO 26 PWM0	26
23	GPIO 23 PWM1	33		Ground	34
24	GPIO 24 PCM_FS/PWM1	35		GPIO 27	27
25	GPIO 25	37		GPIO 28 PCM_DIN	28
	Ground	39		GPIO 29 PCM_DOUT	29

Pour avoir ce tableau  
 taper au clavier = gpio readall

A gauche le Raspberry, à droite la correspondances avec les Pin numéroté. En Vert ce sont les broches que l'on peut utiliser pour nos bidouilles.

Deux résistances de 330 ohms (orange, orange, marron)



# Fonctionnement des entrées / sorties du GPIO.

Raspberry Pi 3 Model B (J8 Header)				
GPIO#	NAME		NAME	GPIO#
	3.3 VDC Power	1		2
8	GPIO 8 SDA1 (I2C)	3		4
9	GPIO 9 SCL1 (I2C)	5		6
7	GPIO 7 GPCLK0	7		8
	Ground	9		10
0	GPIO 0	11		12
2	GPIO 2	13		14
3	GPIO 3	15		16
	3.3 VDC Power	17		18
12	GPIO 12 MOSI (SPI)	19		20
13	GPIO 13 MISO (SPI)	21		22
14	GPIO 14 SCLK (SPI)	23		24
	Ground	25		26
30	SDA0 (I2C ID EEPROM)	27		28
21	GPIO 21 GPCLK1	29		30
22	GPIO 22 GPCLK2	31		32
23	GPIO 23 PWM1	33		34
24	GPIO 24 PCM_FS/PWM1	35		36
25	GPIO 25	37		38
	Ground	39		40

Trop de chiffres tue les chiffres... simplifions

Ce que vous devez retenir c'est que pour activer un port **VERT uniquement**, il faut lui demander gentiment comme ceci:

- Je veux que le GPIO 2 soit une sortie il faudra écrire:

**gpio mode 2 out** (pour commander une led par exemple)

*ou une entrée pour un bouton poussoir par exemple.*

**gpio mode 4 in** (voir les dernière pages pour mon détecteur PIR)

- Ensuite lui dire Je veux l'actionner (mettre un 1=ON)

**gpio write 0 1**

ou l'éteindre (mettre un 0=OFF)

**gpio write 0 0**

## Dans Jarvis-commands :

```
*LIN*D*OEIL*==say "Voici mon plus beau clein d'oeil...";gpio write 2 0; sleep 0.8; gpio write 2 1
```

```
*LIGN*DES*EUX*==say "Comme ca ? "; gpio write 2 0; gpio write 0 0; sleep 0.8; gpio write 2 1;gpio write 0 1
```

```
*NEZ*== [ `gpio read 3` = 0 ] && (say "Ne me prend pas pour un clown ! "; gpio write 3 1) || (say "Ok je m'efface"; gpio write 3 0)
```

```
*OUV*ES*EUX*==say "Ha je te voix mieux comme ça"; gpio write 2 1; gpio write 0 1
```

```
*FER*ES*EUX*==say "Je vais m'endormir..."; gpio write 2 0; gpio write 0 0
```

# Mes Hooks Jarvis:

*Rappel:      Gpio 0 et 2=Yeux      GPIO3=NEZ*

## **entering\_cmd**

```
$verbose && jv_debug "DEBUG: entering_cmd hook"
```

```
gpio mode 2 out
```

```
gpio mode 0 out
```

```
gpio write 2 1
```

```
gpio write 0 1
```

```
sleep 0.2
```

```
gpio write 2 0
```

```
gpio write 0 0
```

```
sleep 0.2
```

```
gpio write 2 1
```

```
gpio write 0 1
```

```
sleep 0.2
```

```
gpio write 2 0
```

```
gpio write 0 0
```

```
sleep 0.2
```

```
gpio write 2 1
```

```
gpio write 0 1
```

## **exiting\_cmd**

```
$verbose && jv_debug "DEBUG: exiting_cmd hook"
```

```
gpio mode 2 out
```

```
gpio mode 0 out
```

```
gpio write 2 0
```

```
gpio write 0 0
```

## **program\_exit**

```
$verbose && jv_debug "DEBUG: program_exit hook"
```

```
(( $1 )) && say "J'ai rencontré une erreur, relancez-moi!"
```

```
gpio write 0 0
```

```
gpio write 2 0
```

```
gpio write 3 0
```

## **program\_startup**

```
$verbose && jv_debug "DEBUG: program_startup hook"
```

```
gpio mode 3 out
```

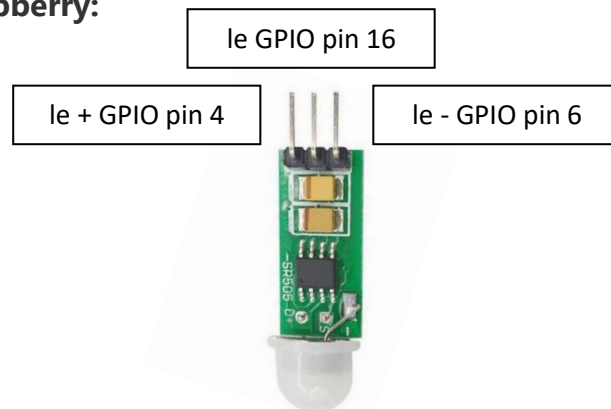
```
gpio write 3 1
```

# Pour les entrées du port GPIO il faut ce module:



The screenshot shows the AliExpress product page for the HC-SR505 Mini Infrared PIR Motion Sensor Module. The product is a small green PCB with a white PIR sensor dome. The page includes the AliExpress logo, a search bar, and navigation links. The product details show a price of €1,30 per piece, a 5.0 star rating from 14 votes, and 64 orders. The shipping is free to France via China Post Ordinary Small Packet Plus, with a delivery time of 22-40 days. The page also features a 'Passer la souris dessus pour zoomer' (Click on the mouse to zoom) prompt and a 'Q' icon for zooming.

## Connexion au Raspberry:



## Configuration:

gpio mode 4 in  
gpio read 4

Réponse: 1 ou 0

Si il détecte quelqu'un il y aura 1 (10 secondes de délais automatique entre les mesures du PIR)  
Puis si il détecte aucun mouvement il y aura 0

**Dans Jarvis: Utiliser le jarvis-plug-PIR (*En cours de finalisation...*)**

<https://github.com/Jean-Bernard-Hallez/jarvis-plug-PIR>