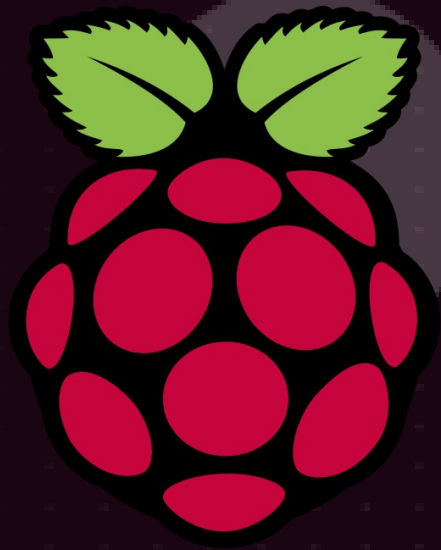


Framboise pour le printemps

A la découverte des systèmes embarqué Raspberry pi et Orange pi.



David Bieder
Ubuntu Tunisian LoCo Team



Définition:

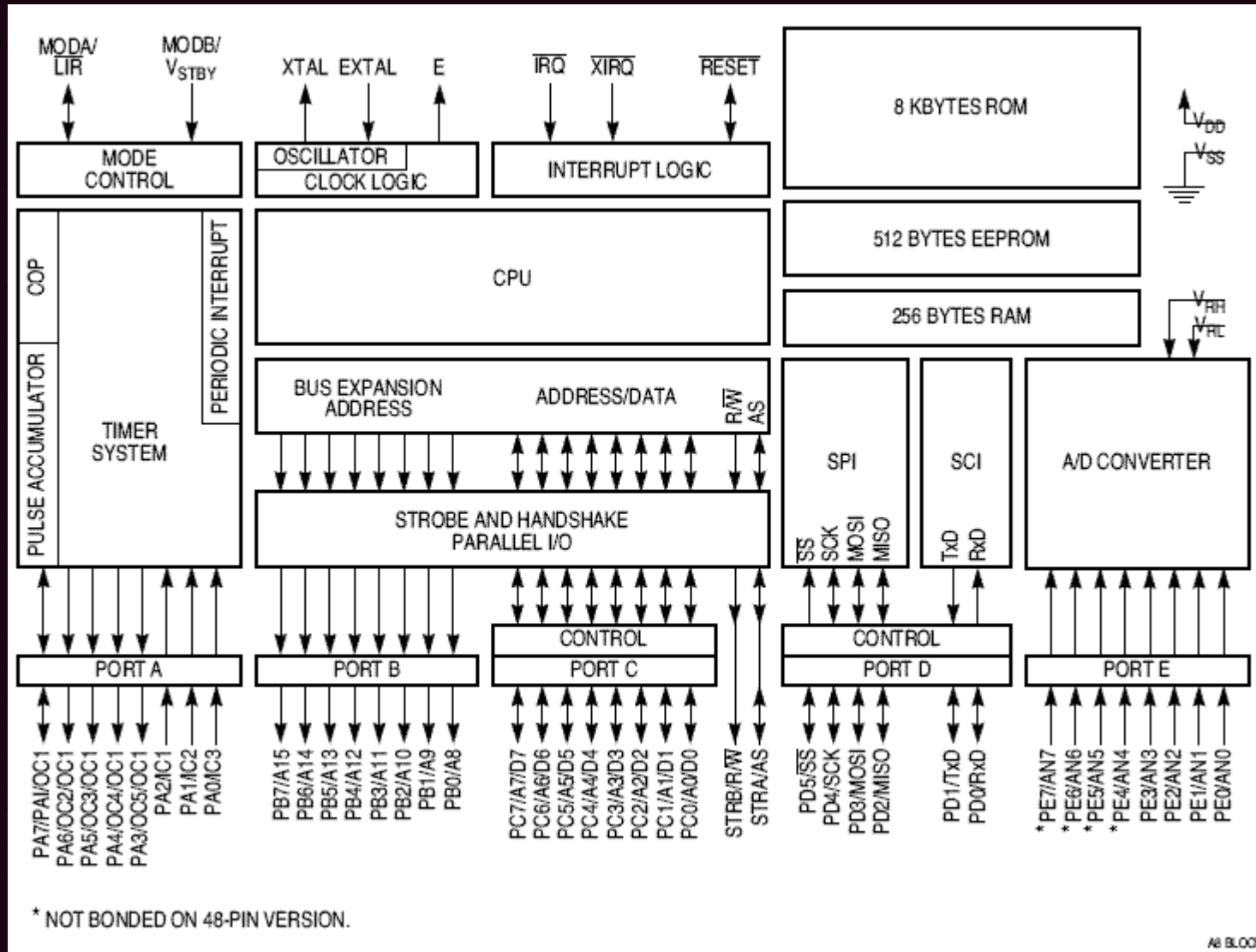


Un système embarqué est défini comme un système électronique et informatique autonome, souvent temps réel, spécialisé dans une tâche bien précise. Le terme désigne aussi bien le matériel informatique que le logiciel utilisé. Ses ressources sont généralement limitées. Cette limitation est généralement d'ordre spatial (encombrement réduit) et énergétique (consommation restreinte).

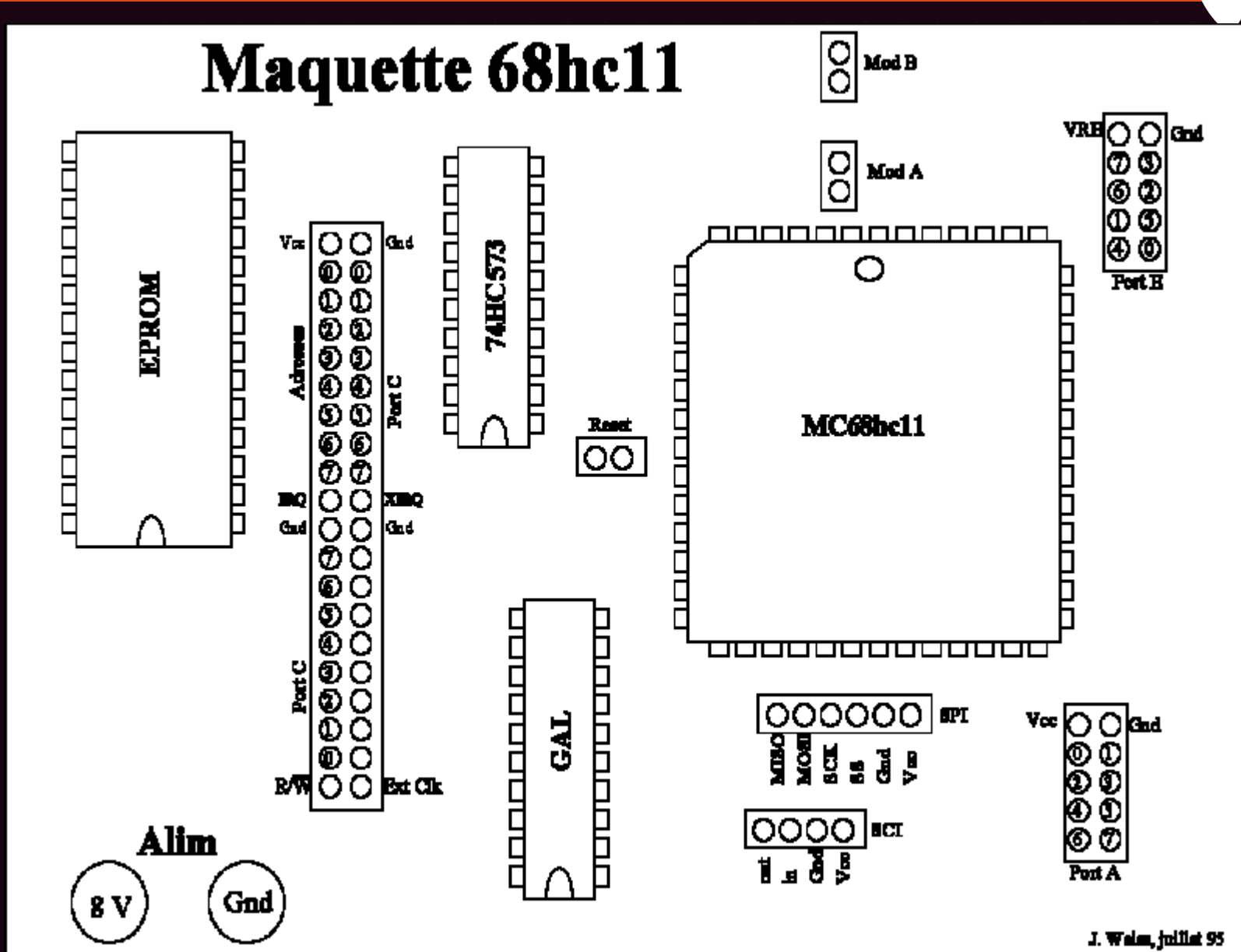
Petit histoire du silicium.



- 6809 et 68hc11



Et une pi primitive:



Histoire et processeur.



- Motorola
 - 68xx(M05...)
 - 680xx(Atari St, Amiga, Mac...)
- Intel/AMD
 - Pc, Mac...
- Arm
 - Système embarqué
 - Smartphone
 - Contrôleur

Architecture



- CISC
 - complex instruction set computer
 - Ex: X86
- RISC
 - reduce instruction set computer
 - Ex: gamme Arm
- DSP
 - Digital signal processing (produit de convolution)
 - Ex: Nvidia, Ati

ARM



Sessions

- ARM1 (1985) : premier prototype de cœur ARM
- Famille ARM2 (1987) : premier ARM commercialisé dans l'Archimedes d'Acorn : Pipeline 3 niveaux, adressage sur 24 bits alignés (16 registres de 32 bits soit 64 Mio adressables), 8 MHz puis 12 MHz
- Famille ARM3 : Interface FPU, fréquence 25 puis 33 MHz, 4K cache
- Famille ARM4
- Famille ARM4T
- Famille ARM5TE (2000) : arrivée de Thumb et de fonctionnalités de DSP
- Famille ARM5TEJ (2000) : arrivée de Jazelle
- Famille ARM6 sorti en 1990
- avec Jazelle (2001)
- ARM1136J(F)-STM (été 2002)
- ARM1156T2(F)-S (2003)
- ARM1176JZ(F)-S (2003)
- Famille ARM7 :
- ARM720T (MMU)
- ARM7TDMI
- ARM7TDMI-S
- ARM7EJ-S : DSP et Jazelle
- Famille ARM9 (5 niveaux de pipeline sur les entiers, MMU) : ARM920T (double cache de 16 ko) et ARM922T (double cache de 8 ko)
- Famille ARM9E
- ARM946E-S : DSP, double cache, MPU, 1 port AHB
- ARM926EJ-S: DSP, double cache, MMU, 2 ports AHB
- ARM966E-S : DSP, double cache, MPU, 1 ports AHB
- Famille ARM10E
- ARM1020E : DSP, double cache de 32 ko, MMU
- ARM1022E : identique au ARM1020E, sauf le double cache de 16 ko
- ARM1026EJ-S
- Famille ARM11 : SIMD, Jazelle, DSP, Thumb-2
- ARM1136JF-S : FPU
- ARM1156T2-S
- ARM1156T2F-S : FPU
- ARM1176JZ-S
- ARM1176JZF-S : FPU

ARM suite

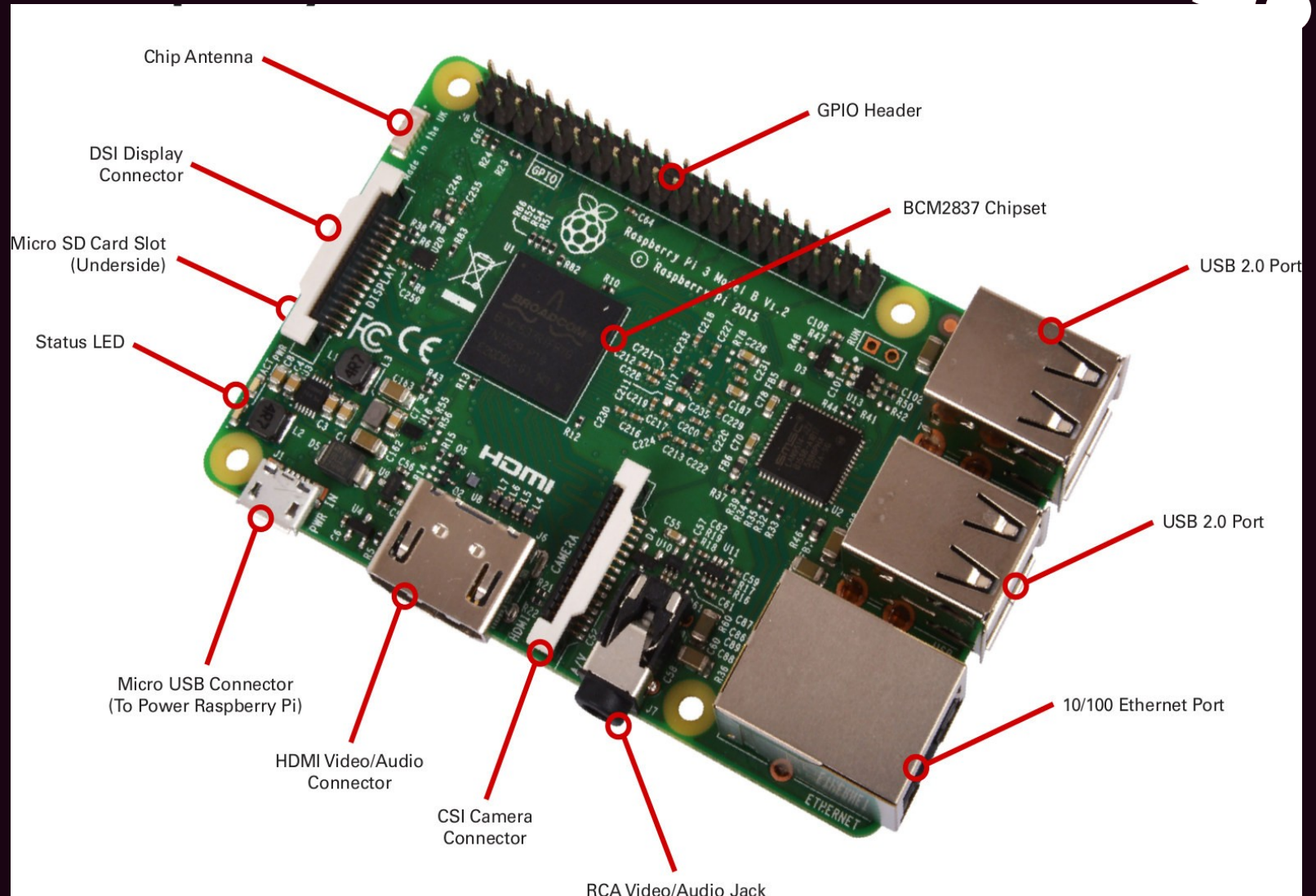


- Famille Cortex-A, processeur applicatif: Architecture ARMv7-A, SIMD, Jazelle, DSP, Thumb-2
- Cortex-A5
- Cortex-A5 MPCore : Cortex-A5 version multiprocesseur (1 à 4 CPU)
- Cortex-A7 MPCore : Cortex-A7 multiprocesseur (1 à 8 CPU)
- Cortex-A8
- Cortex-A9
- Cortex-A9 MPCore : Cortex-A9 version multiprocesseur (1 à 4 CPU), 45, 32 nm et 28 nm jusqu'à 2 GHz (3,1 GHz dans certaines conditions 19)
- Cortex-A12 MPCore, jamais produit, finalement remplacé par le Cortex-A17
- Cortex-A15 MPCore : Cortex-A15 multiprocesseur (1 à 4 CPU), 45, 32 et 28 nm (20 nm projeté), jusqu'à 2,5 GHz
- Cortex-A17 MPCore
- Cortex-A32, processeur 32 bits compatible avec le jeu d'instruction 64 bits d'ARMv8.
- Famille Cortex-R, processeur temps-réel: Architecture ARMv7-R
- Cortex-R4
- Famille Cortex-M, processeur embarqué : Architecture ARMv6-M et ARMv7-M
- Cortex-M0
- Cortex-M0+
- Cortex-M1
- Cortex-M3
- Cortex-M4
- Cortex-M7
- Famille ARMv8, processeurs 64 bits).
- Cortex-A35
- Cortex-A53
- Cortex-A57
- Cortex-A72
- Cortex-A73

Raspberry Pi



Elle a tout d'une grande.

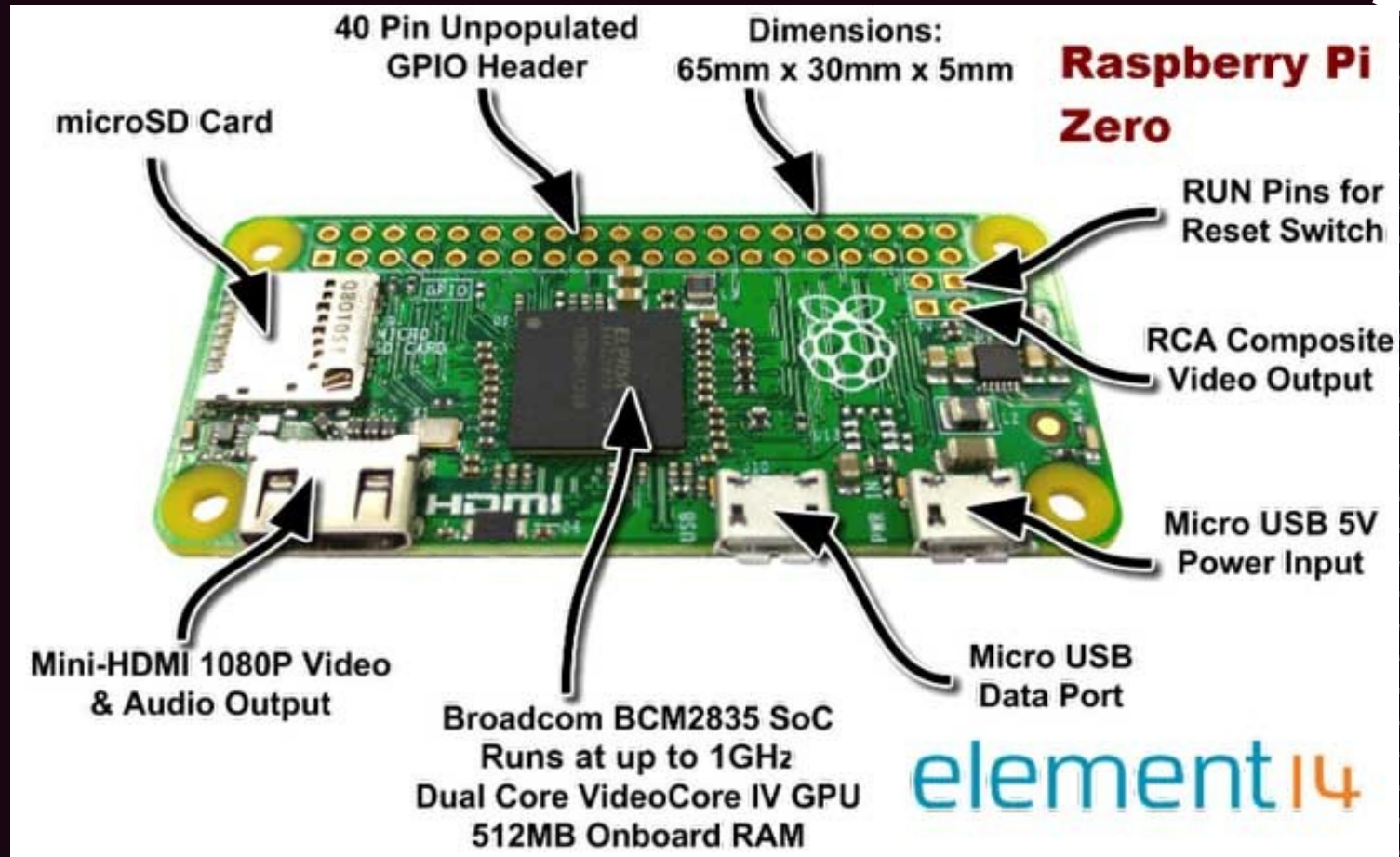


Pour la Pi3

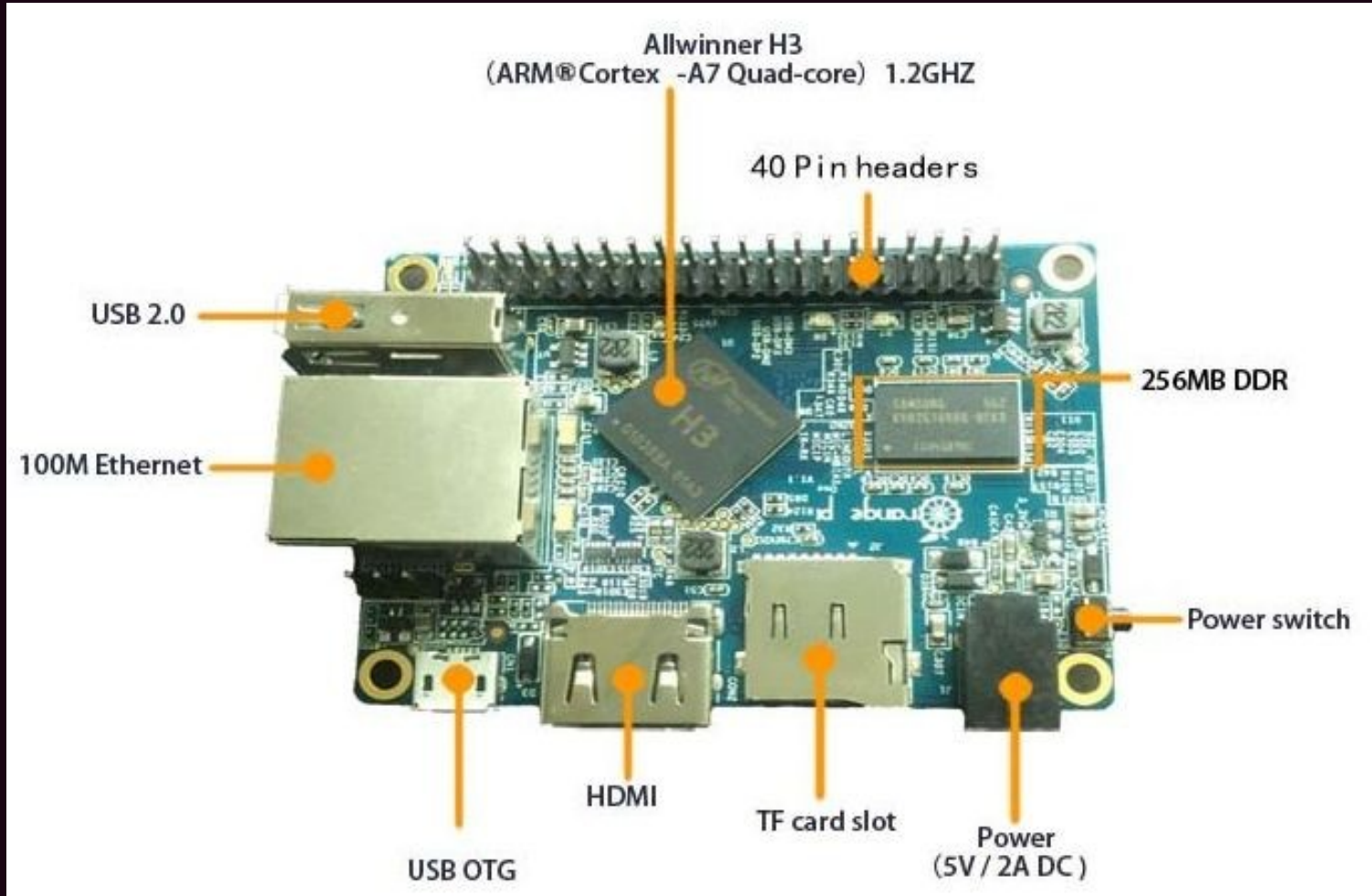


- Architecture: armv7l
- Byte Order: Little Endian
- CPU(s): 4
- On-line CPU(s) list: 0-3
- Thread(s) per core: 1
- Core(s) per socket: 4
- Socket(s): 1
- Model name: ARMv7 Processor rev 4 (v7l)
- CPU max MHz: 1200,0000
- CPU min MHz: 600,0000
-

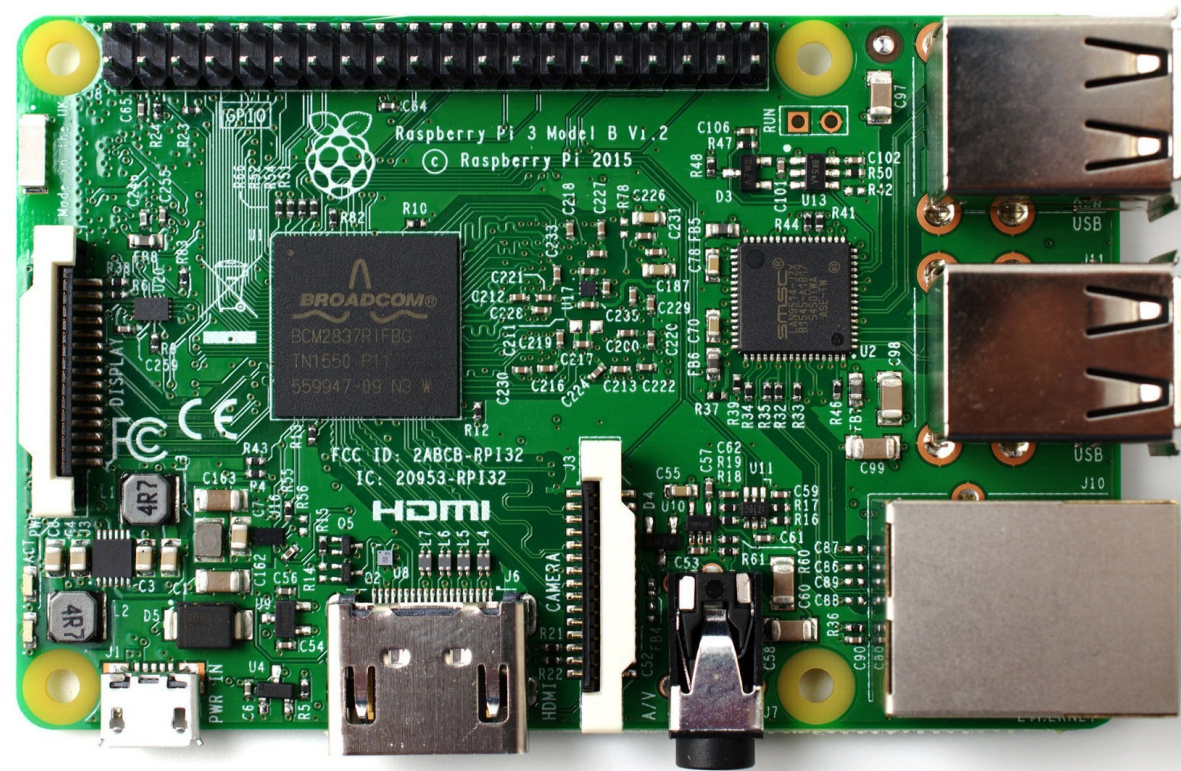
La version Zero



Orange pi one



Que intérêt?



Pour l'Orange Pi



- Architecture: armv7l
- Byte Order: Little Endian
- CPU(s): 4
- On-line CPU(s) list: 0-3
- Thread(s) par cœur : 1
- Cœur(s) par socket : 4
- Socket(s): 1
- CPU max MHz: 1200,0000
- CPU min MHz: 480,0000

GPIO



Alternate Function					Alternate Function
	3.3V PWR	1		2	5V PWR
I2C1 SDA	GPIO 2	3		4	5V PWR
I2C1 SCL	GPIO 3	5		6	GND
	GPIO 4	7		8	UART0 TX
	GND	9		10	UART0 RX
	GPIO 17	11		12	GPIO 18
	GPIO 27	13		14	GND
	GPIO 22	15		16	GPIO 23
	3.3V PWR	17		18	GPIO 24
SPI0 MOSI	GPIO 10	19		20	GND
SPI0 MISO	GPIO 9	21		22	GPIO 25
SPI0 SCLK	GPIO 11	23		24	GPIO 8
	GND	25		26	GPIO 7
	Reserved	27		28	Reserved
	GPIO 5	29		30	GND
	GPIO 6	31		32	GPIO 12
	GPIO 13	33		34	GND
SPI1 MISO	GPIO 19	35		36	GPIO 16
	GPIO 26	37		38	GPIO 20
	GND	39		40	GPIO 21
					SPI0 CS0
					SPI0 CS1
					SPI1 CS0
					SPI1 MOSI
					SPI1 SCLK

GPIO 26 et 40



Raspberry Pi B
Rev 2 P1 GPIO Header

	Pin No.		
3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7

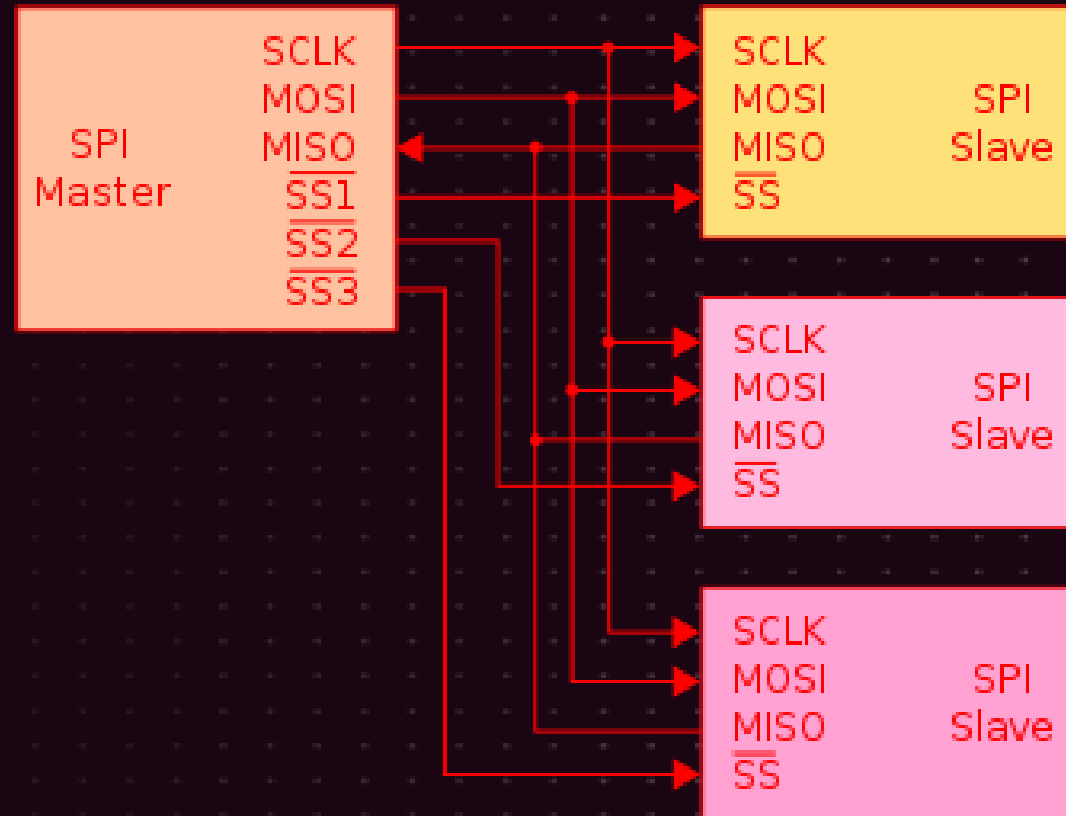
Key

Power +	UART
GND	SPI
I²C	GPIO

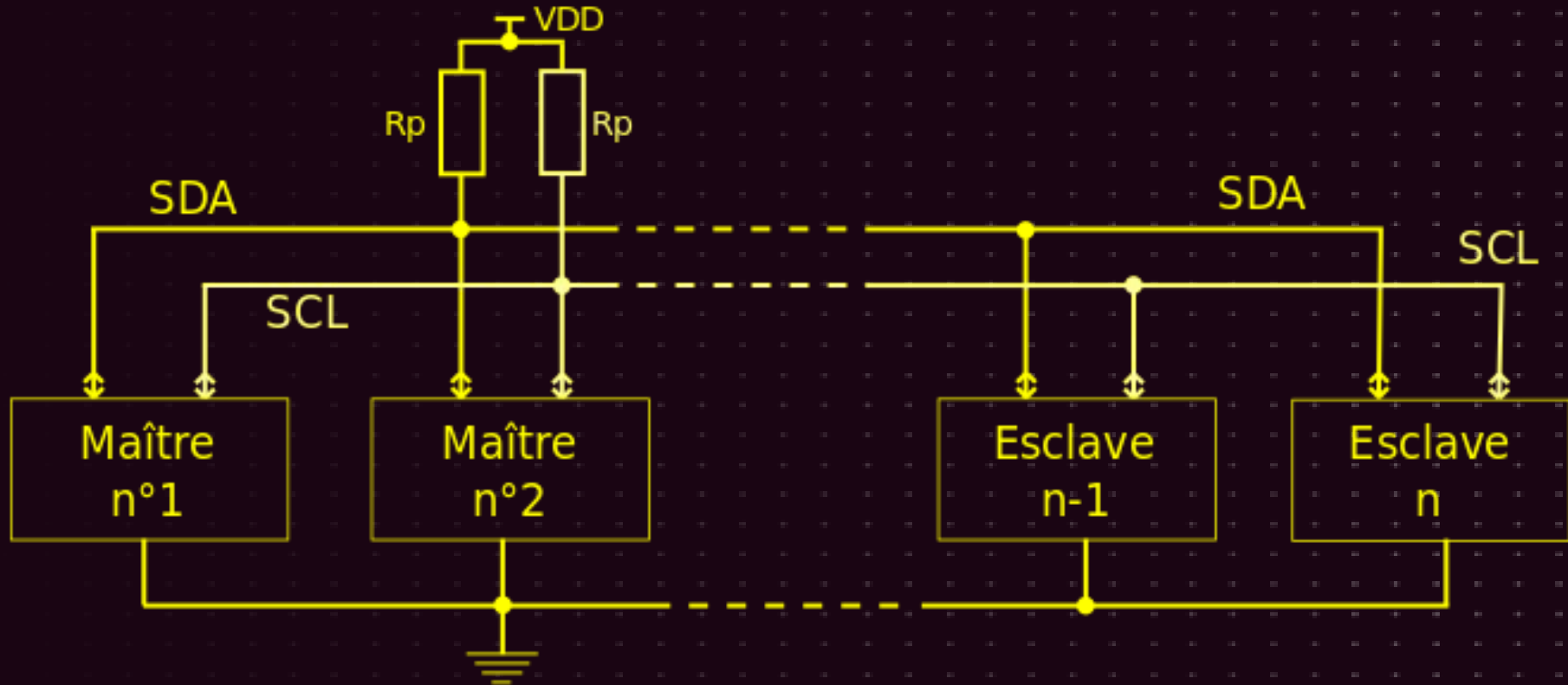
Raspberry Pi B+
B+ J8 GPIO Header

	Pin No.		
3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
DNC	27	28	DNC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

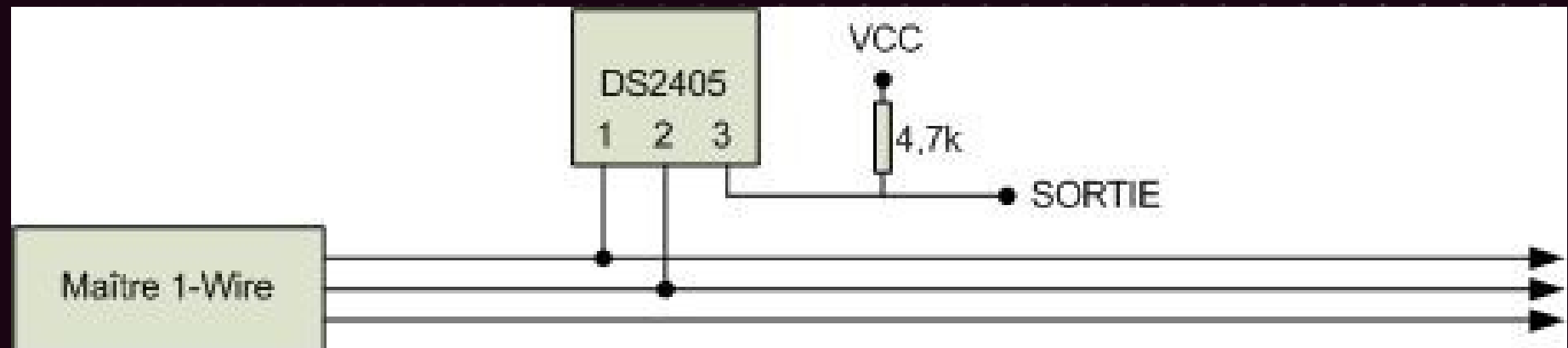
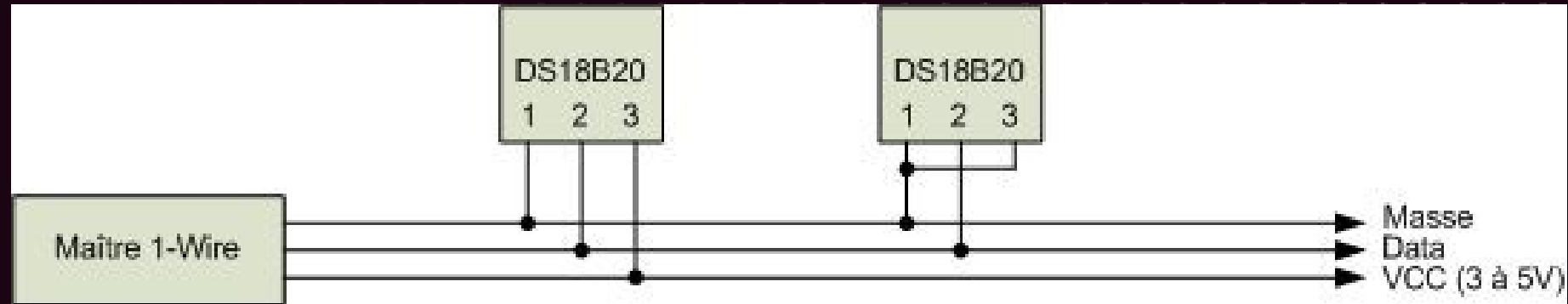
Bus SPI



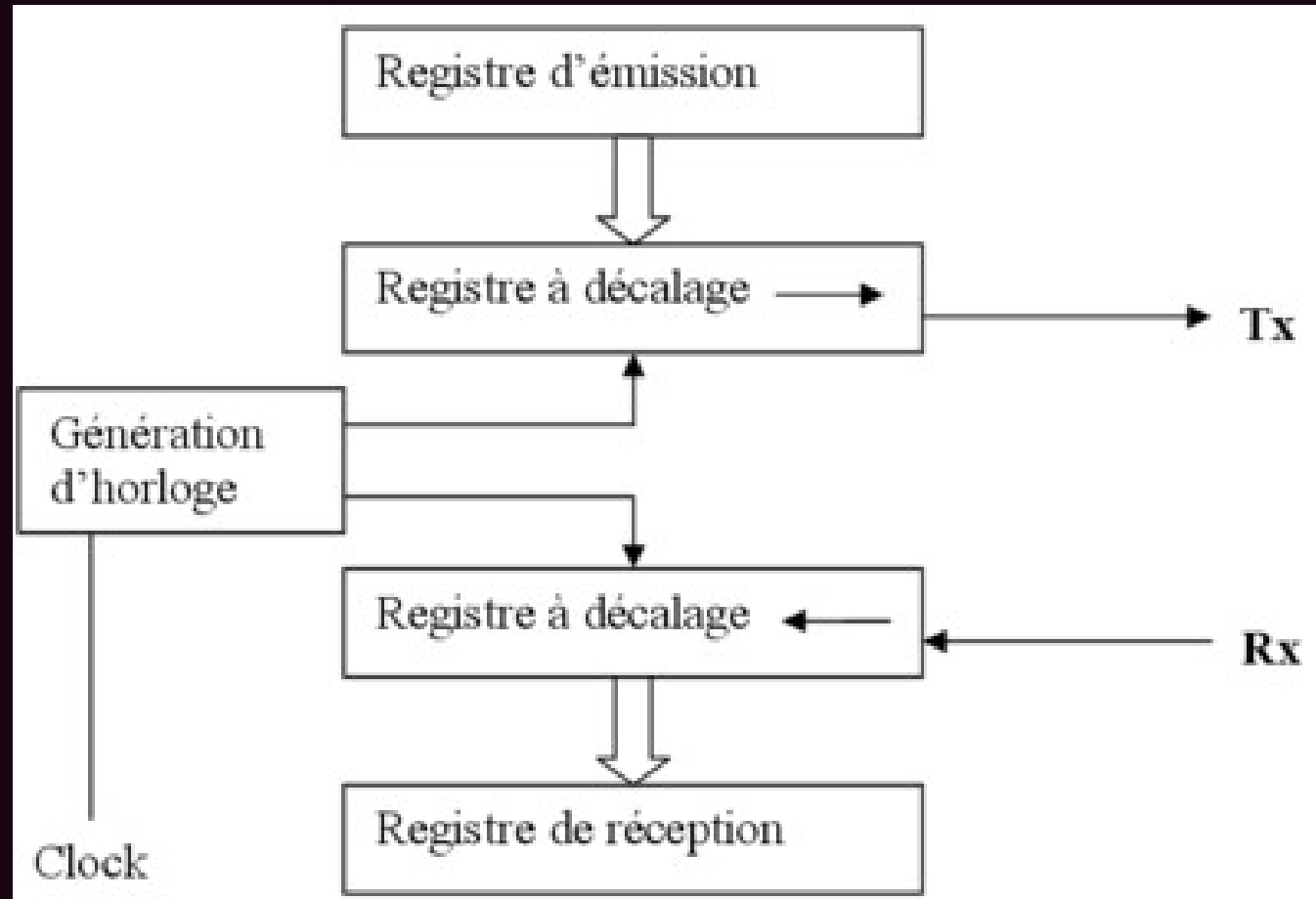
IC²



1-wire



UART



La numérotation



- GPIO.BOARD en noir (numérotation native)
- GPIO.BCM fait le liens avec la version 26 connecteur.

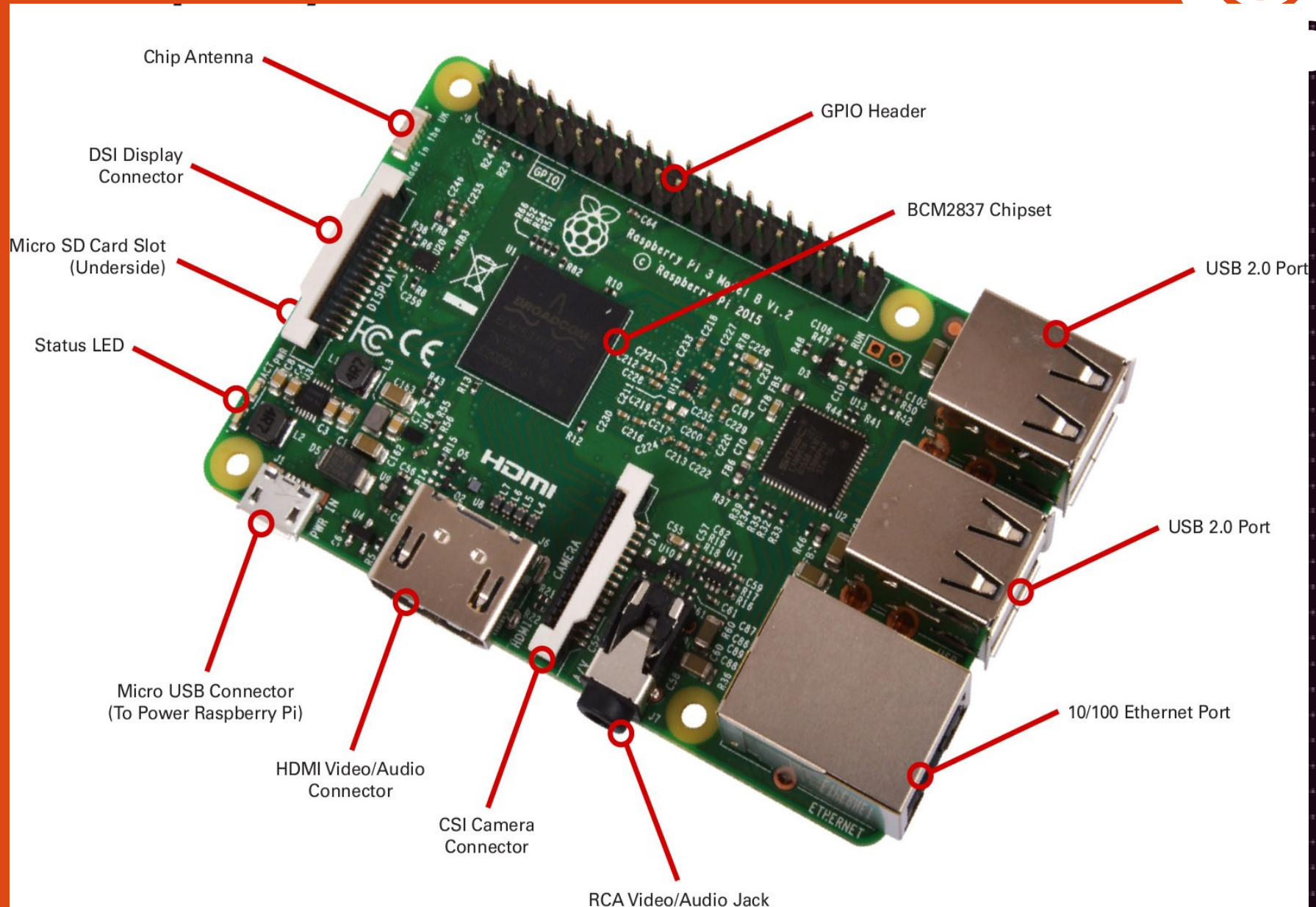


Installer Ubuntu

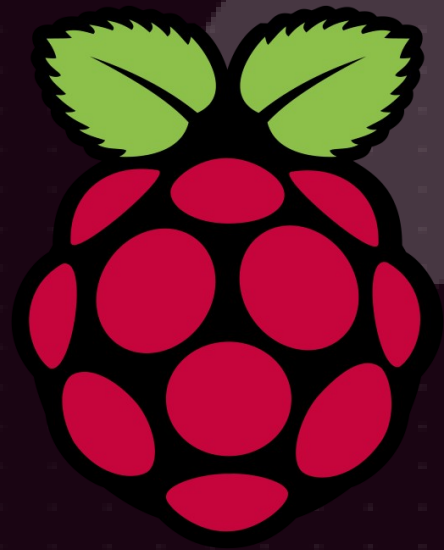


- `sudo apt-get install gddrescue xz-utils`
- `unxz ubuntu-mate-16.04.2-desktop-armhf-raspberry-pi.img.xz`
- `sudo ddrescue -D --force ubuntu-mate-16.04.2-desktop-armhf-raspberry-pi.img /dev/sdx`

Et démarrer



Parti Atelier pratique des pi3.



Jouer avec les pins!

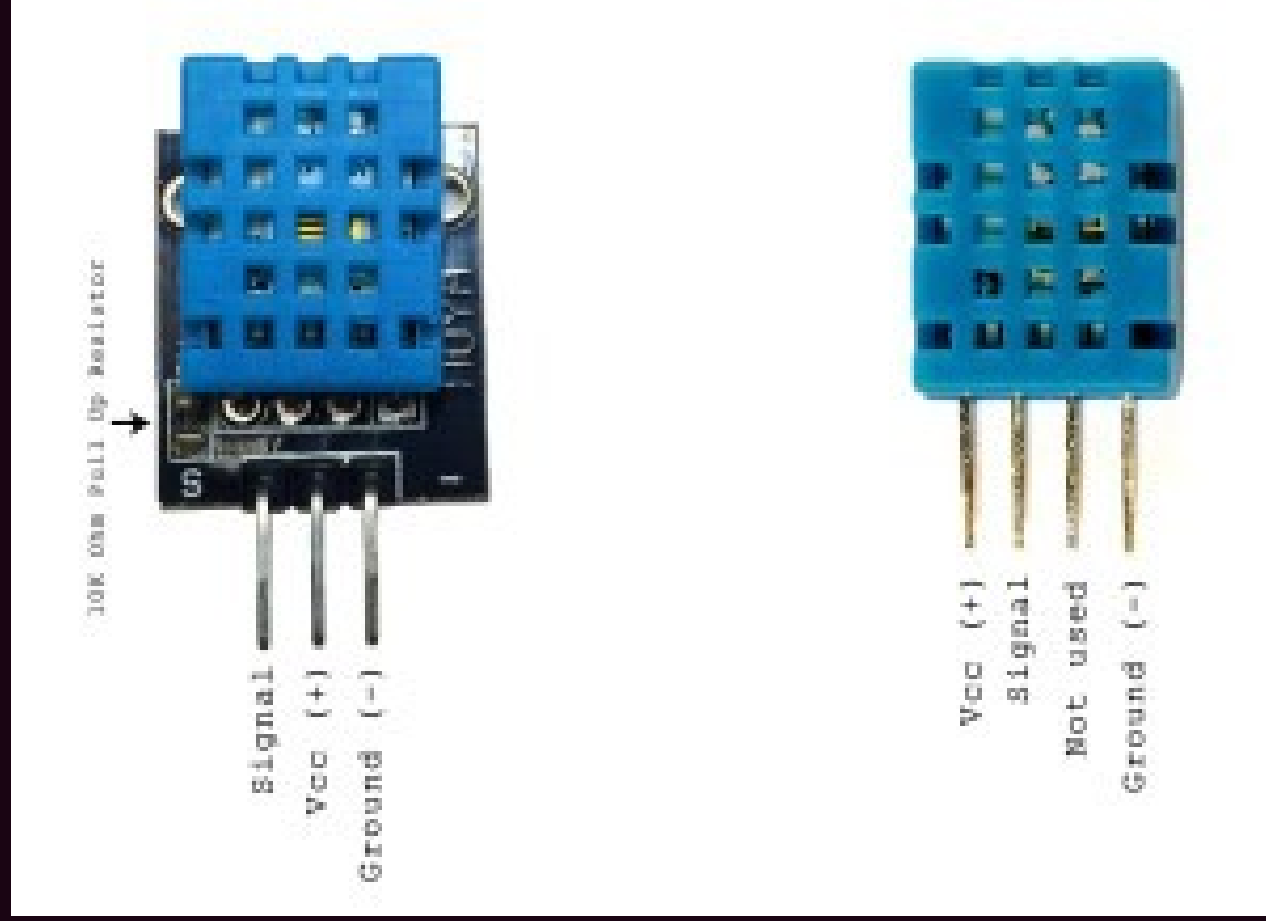


Météo

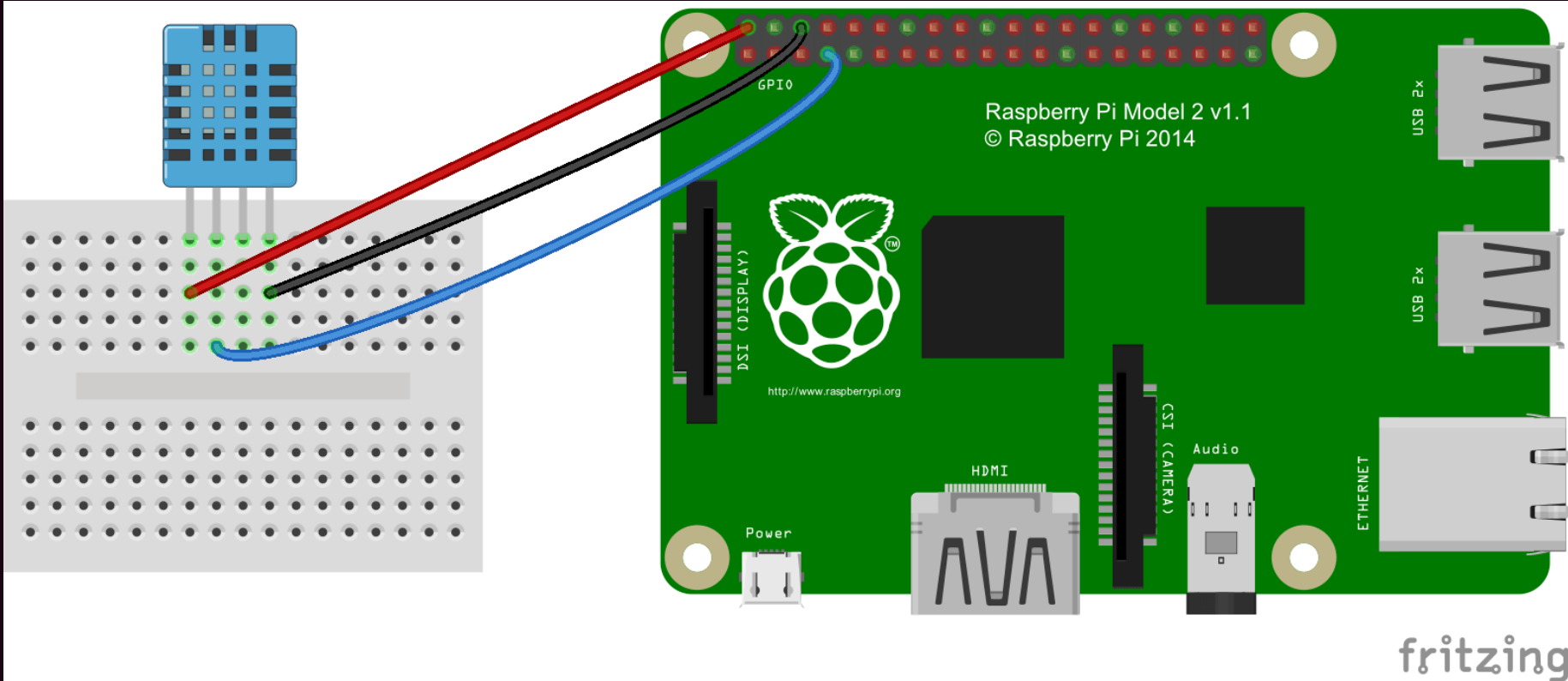


- Réaliser un prise de température et d'humidité.
- Installation du capteur.
- Prise d'information et traitement.
-

DHT11



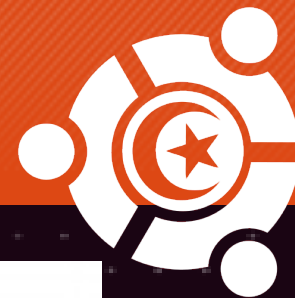
Cablage.



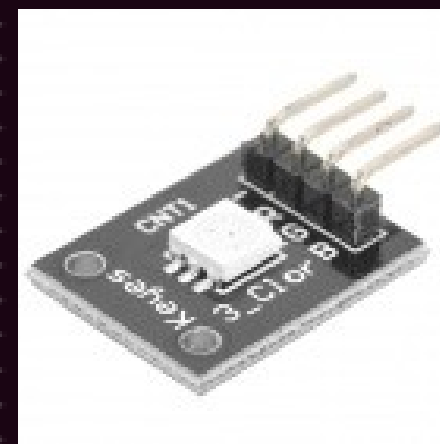
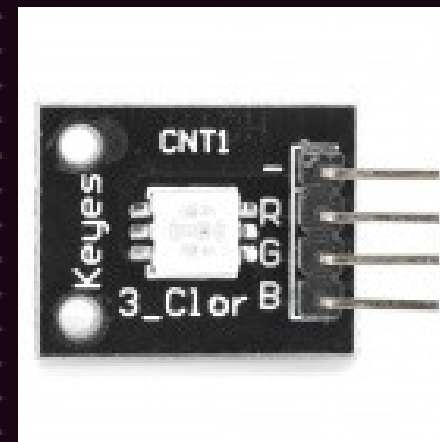
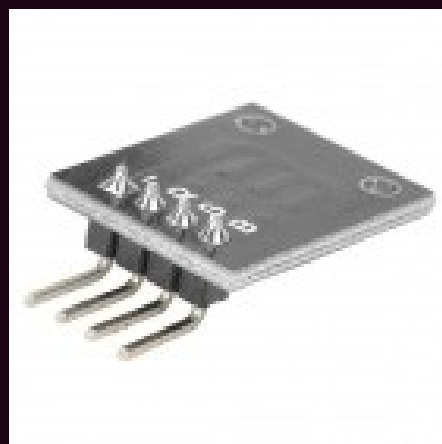
Codage python



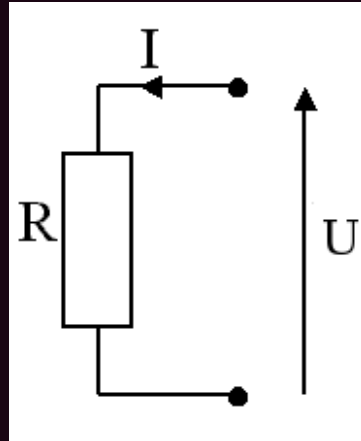
- `root@david-desktop:/home/david# python`
- `Python 2.7.12 (default, Nov 19 2016, 06:48:10)`
- `[GCC 5.4.0 20160609] on linux2`
- Type "help", "copyright", "credits" or "license" for more information.
- `>>> import sys`
- `>>> import Adafruit_DHT`
- `>>> humidite, temperature = Adafruit_DHT.read_retry(11, 4)`
- `>>> humidite`
- `69.0`
- `>>> temperature`
- `17.0`
-



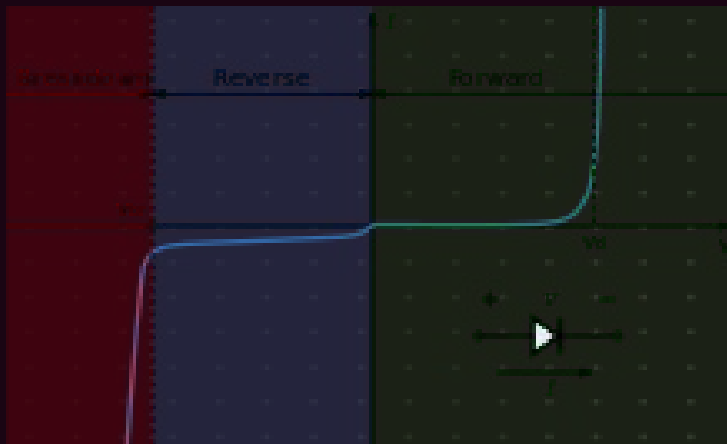
- Utilisation d'un led tri color.



Loi d'ohm



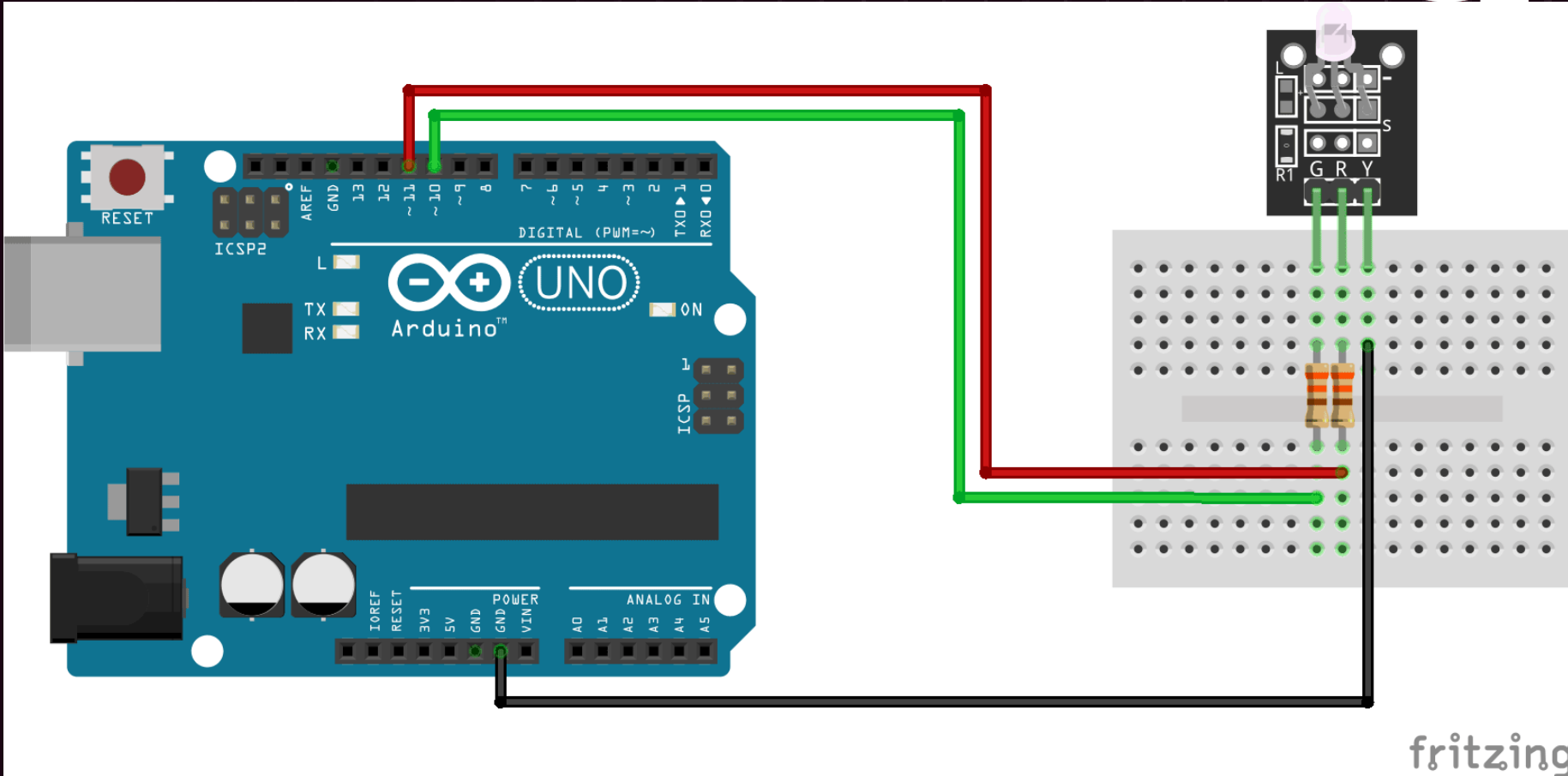
- $U = RI \rightarrow R = U/I$



- Donc pour I:

- $R = (U - U_{\text{led}})/I$

Cablage pour deux couleurs



Code python



```
• root@david-desktop:/home/david# python
• Python 2.7.12 (default, Nov 19 2016, 06:48:10)
• [GCC 5.4.0 20160609] on linux2
• Type "help", "copyright", "credits" or "license" for more information.
• >>> from RPi import GPIO
• >>> GPIO.setwarnings(False)
• >>> GPIO.setmode(GPIO.BOARD)
• >>> GPIO.setup(36, GPIO.OUT)
• >>> GPIO.setup(38, GPIO.OUT)
• >>> GPIO.setup(40, GPIO.OUT)
• >>> GPIO.output(40, GPIO.LOW)
• >>> GPIO.output(38, GPIO.LOW)
• >>> GPIO.output(36, GPIO.LOW)
• >>> GPIO.output(38, GPIO.HIGH)
• >>> GPIO.output(40, GPIO.HIGH)
• >>> GPIO.output(40, GPIO.HIGH)
• >>> GPIO.output(36, GPIO.HIGH)
```

Et merci de votre accueil.

