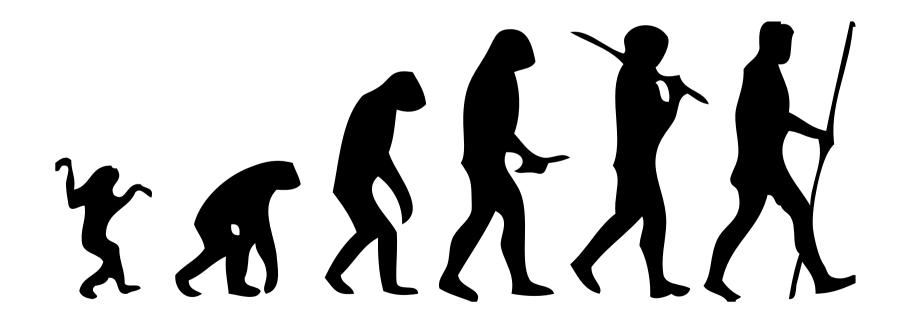




Manejando el mundo con la Raspberry Pi (RPi)



2015/4/10 Àngel Perles





Contenido

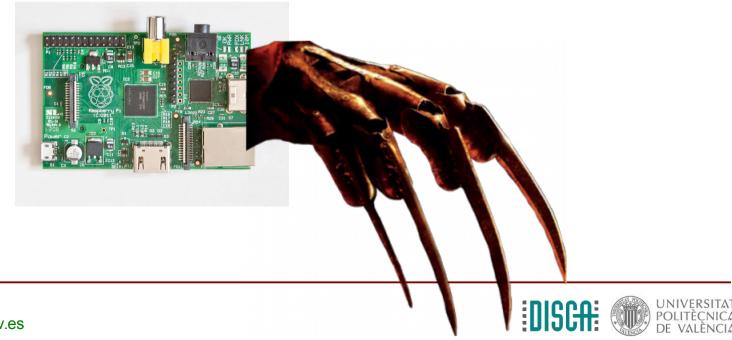
- Objetivo
- Opciones de conexión
 - bajo nivel: GPIO, SPI, I2C, UART
 - CSI
 - DSI
 - USB
- Un caso práctico
 - El problema
 - El montaje
 - El software
 - WiringPi
 - Probando





Objetivo

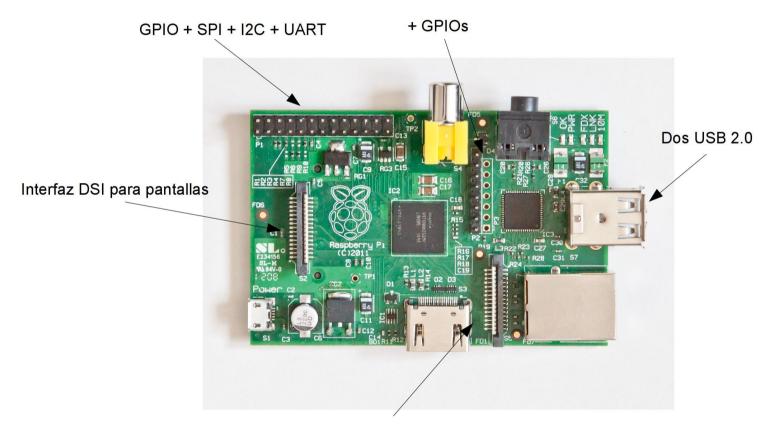
- Tener criterio para seleccionar el tipo de dispositivos adecuado a cada interfaz de la RPi
- Conocer básicamente software para acceder al subsistema de bajo nivel
- Practicar con un ejemplo hardware + software





Opciones de conexión

Estos son las opciones y sus conectores (RPi 1 mod.B ver. 2.0)



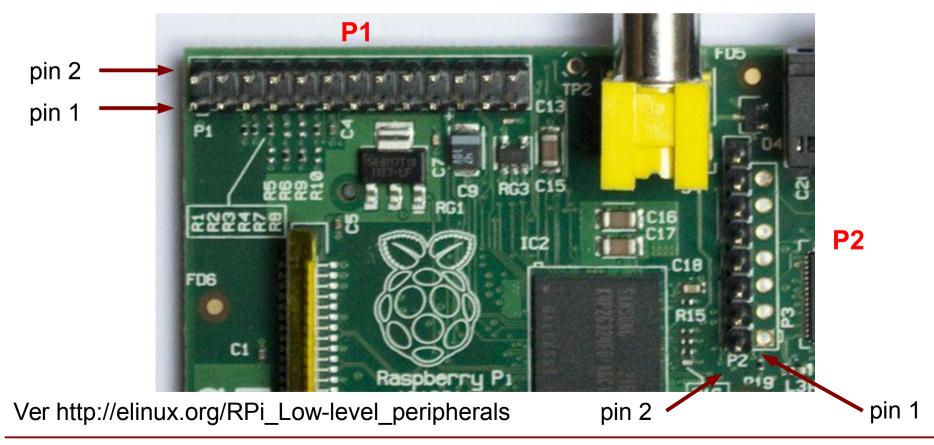
Interfaz CSI para cámaras

• Veamos qué podemos conectar. De "low-level" a "high-level" Recomendamos la "RPi 2 modelo B". Pero estas son las que tenemos.





- Los conectores P1 y P2 incorporan las interfaces de bajo nivel
 - General Pourpose Input/Out (GPIO)
 - SPI e I2C
 - UART







• Función asociada a los pines

%	P1	1: The Ma	in GP	Ю сог	nnector		
WiringPi Pin	BCM GPIO	Name	Header		Name	BCM GPIO	WiringPi Pin
		3.3v	1	2	5v		
8	Rv1:0 - Rv2:2	SDA	3	4	5v		
9	Rv1:1 - Rv2:3	SCL	5	6	0v		
7	4	GPIO7	7	8	TxD	14	15
		0v	9	10	RxD	15	16
0	17	GPI00	11	12	GPI01	18	1
2	Rv1:21 - Rv2:27	GPIO2	13	14	0v		
3	22	GPIO3	15	16	GPIO4	23	4
		3.3v	17	18	GPIO5	24	5
12	10	MOSI	19	20	0v		
13	9	MISO	21	22	GPI06	25	6
14	11	SCLK	23	24	CE0	8	10
		0v	25	26	CE1	7	11
WiringPi Pin	BCM GPIO	Name	Header		Name	BCM GPIO	WiringPi Pin

P5: Secondary GPIO connector (Rev. 2 Pi only)										
WiringPi Pin	BCM GPIO	Name	Header		Name	BCM GPIO	WiringPi Pin			
	-	5v	1	2	3.3v					
17	28	GPIO8	3	4	GPIO9	29	18			
19	30	GPIO10	5	6	GPIO11	31	20			
		0v	7	8	0v					
WiringPi Pin	BCM GPIO	Name	Header		Name	BCM GPIO	WiringPi Pin			





GPIO

- Para hacer entrada/salida digital: abierto/cerrado, blanco/negro, grande/pequeño
- Nivel "0" -> 0 voltios, Nivel "1" -> 3,3 voltios
- Pines configurables como salida o como entrada
- Pull-up, pull-down programable

Ideal para

- Leer (sensar) botones, sensores de presencia, finales de carrera, ...
- Escribir (actuar) LEDs, relés, electroválvulas, motores ...







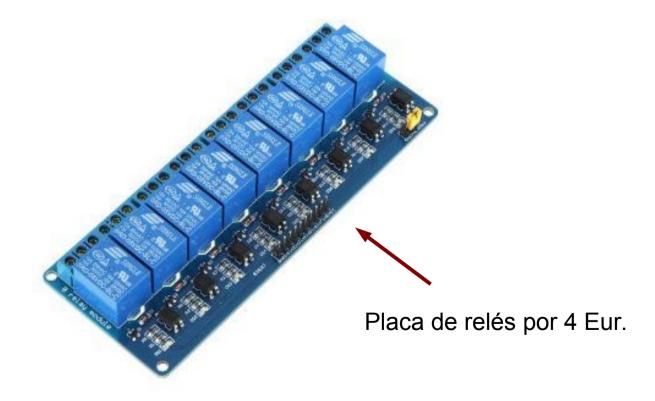






• GPIO

- PEGA: poca corriente de salida, unos 15 mA por pin
- Hace falta algún tipo de amplificación para las cargas grandes
- NO PROBLEM: montones de plaquitas en el mercado







GPIO

Como en esta edición hay mucho electrónico, vamos a trastear saltando al "Ejemplo práctico".

Después seguimos con las opciones.





I2C

- "Inter-integrated circuit" es un bus de datos serie
- Pines: SCL (serial clock), SDA (serial data) y GND
- Ideal para tener un chorro de sensores/actuadores sin requisitos de alta velocidad
 - Sensores digitales de temperatura, posición, magnetómetros, corriente
 - Actuadores digitales sencillos
 - Memorias donde se guardan pocas cosas (incluyendo tipo DNIe)
 - Conversores anológico-digitales, digitales-analógicos lentos
 - Reloj de tiempo real
- O lo montas tu o compras uno hecho. Miles en el mercado







RTC



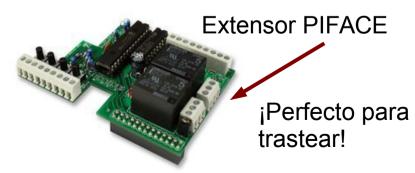


SPI

- Serial Peripheral Interconnect
- Pines: MOSI, MISO (master-slave input-output), CEx (chip enable), SCK (serial clock) y GND
- Pretende reemplazar buses paralelos con la ventaja de un serie sencillo
- Ideal para chips con relativa alta tasa de transferencia
 - Conversores analógico-digitales, digitales-analógicos rápidos, acelerómetros, giróscopos, memoria serie (las SD son memorias serie)
 - Pantallas TFT sencillas
 - Extensores de puertos. CAPES
- O lo montas tu o lo compras hecho. Miles en el mercado











UART

- Universal Asynchronous Receiver Transmiter
- El bus serie de toda la vida (RS-232, RS-485, ...)
- Desplazado por USB en el área de consumo
- Pero muy vivo en el área industrial
- Ideal para dispositivos industriales
 - GPS
 - Básculas industriales
 - Módulos GSM
 - Monederos electrónicos



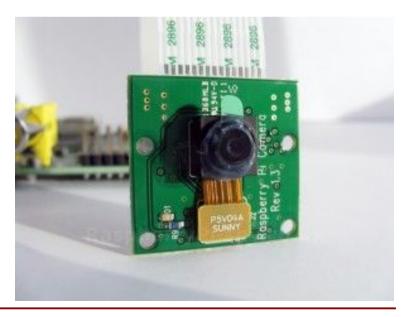






Opciones de conexión: CSI

- MIPI Camera Serial Interface
 - Otro estándar del MIPI
 - http://mipi.org/specifications/camera-interface
 - Aquí hay más suerte. Hay cámara oficial por 25 Eur.
 - 5 megapixels
 - hasta 1080p
 - driver V4L (Video for Linux) siiiiiií. Perfecto para OpenCV, ...
 - Pi Noir: versión infrarroja (es la otra sin filtro IR)







Opciones de conexión: DSI

- MIPI Display Serial Interface
 - Una interfaz para paneles LCD estandarizada por MIPI
 - http://mipi.org/specifications/display-interface
 - No está claro aún su uso oficial
 - Mejor buscar solución de terceros. Información oficial http://raspi.tv/2014/raspberry-pi-official-7-inch-dsi-prototype-preview











Opciones de conexión: USB

Universal Serial Bus 2.0 tipo host

- Universalmente conocido por la enorme diversidad de opciones
- Recordad lo del HUB
- Ideal para dispositivos "complejos": módem 3G, impresoras, Wi-Fi, Bluetooth, cámaras, etc.
- Y no tan complejos: discos duros, teclados, ratones, conversores USB a RS-232, RS-485, 1-wire, ...
- La PEGA son los drivers:
 - Asegurarse antes de que hay driver para el Linux "mainstream"
 - El driver debe ser de código abierto para que se pueda recompilar en la RPi







NOTA: Las marcas de estos dispositivos no se han elegido porque sí. <u>COMPRA</u> productos que soporten oficialmente Linux: HP, Logitech, Transcend, ...





Un caso práctico: el problema

- Quiero gestionar el llenado de un depósito. Tengo un sensor de llenado que hay que leer y una válvula que hay que controlar
- Parece lógico usar la GPIO



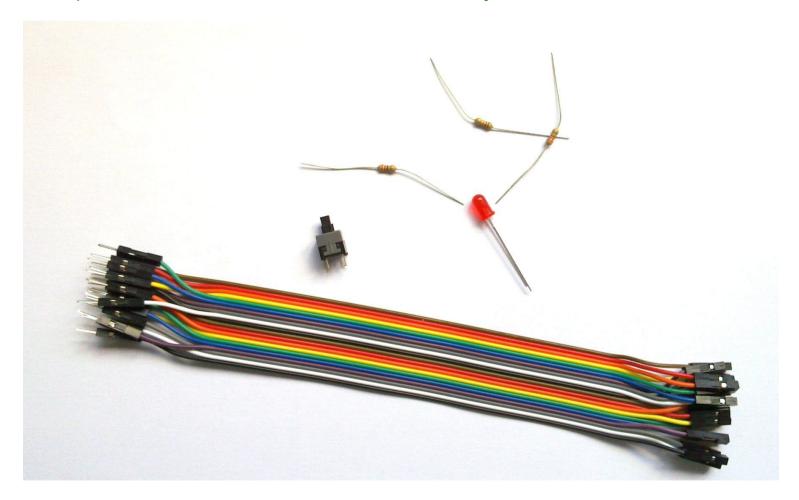






Un caso práctico: el montaje

- Ahora montamos las cosas (a lo pobre)
 - LED, pulsador, 2 resistencias de 220 Ohms y cables

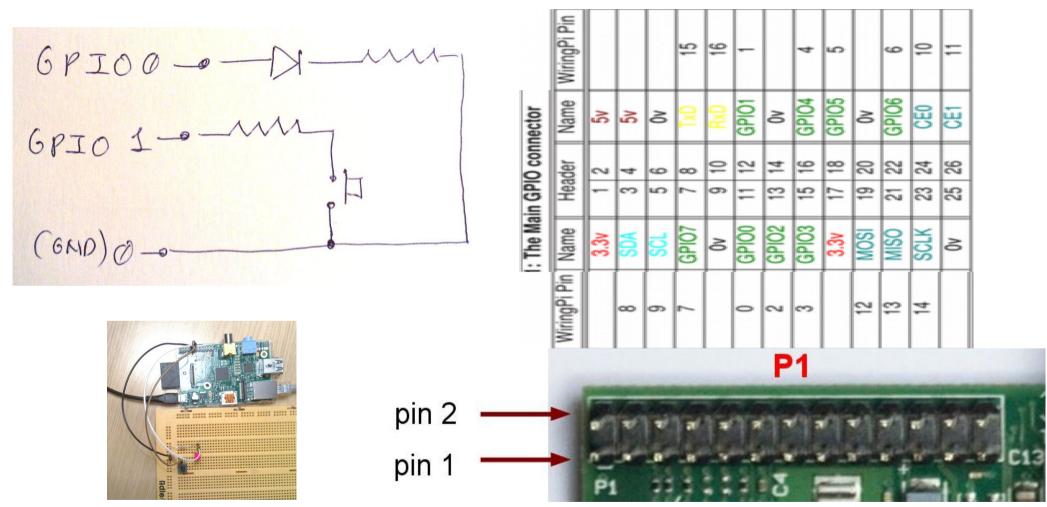






Un caso práctico: el montaje

APAGAR LA PI. QUITAR ALIMENTACIÓN. MONTAR CON CAUTELA



NOTA IMPORTANTE: Se monta así por motivos didácticos. Mala manera de hacer las cosas, pues se deberían proteger los pines.



Un caso práctico: el software

Vale ya he comprado, montado y conectado el cacharro



- MAL: primero nos aseguramos de que hay software para usarlos
 - O son drivers que ya vienen en Linux "mainstream"
 - O vienen en los repositorios generales o adicionales
 - O están en código fuente y hay que compilarlos
- En Linux, los dispositivos suelen mostrarse en /dev/*





Un caso práctico: software/wiringPi

- Primero nos debemos preocupar de la parte software
- Opción "a pelo". Acceso como archivos
 - /sys/class/gpio/gpio
 - \$ echo 1 > /sys/class/gpio/gpio0/value
 - ... y a complicarse la vida
- Opción "alguien ya se lo ha currado".
 - recomendable la biblioteca "wiringPi"
 - http://wiringpi.com/







Un caso práctico: wiringPi

wiringPi

- funciones para la GPIO, I2C, SPI
- soporta además las placas de extensión más populares: piface, gertboard, ...
- y viene listo para algunos chips típicos: sensores de temperatura, conversores AD, ...
- Viene en código fuente y está en un repositorio git
- ¿pega? ¡no!
 - \$ sudo apt-get install gcc
 - \$ sudo apt-get install git-core
 - \$ git clone git://git.drogon.net/wiringPi
 - \$ cd wiringPi
 - \$./build







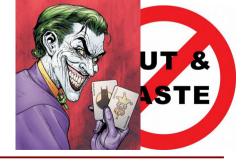
Un caso práctico: probando

- Veamos si va. Probad:
 - pi@raspberrypi ~ \$ gpio mode 0 out
 - pi@raspberrypi ~ \$ gpio write 0 1
 - pi@raspberrypi ~ \$ gpio write 0 0
- Si va, pues hacemos un pequeño script
 - pi@raspberrypi ~ \$ cd ~
 - pi@raspberrypi ~ \$ mkdir test_pins
 - pi@raspberrypi ~ \$ cd test_pins
 - pi@raspberrypi ~ \$ leafpad test_pins & (suponemos terminal dentro X o entubado X)
 - guardar como "test_pins"
 - pi@raspberrypi ~ \$ chmod 700 ./test_pins
 - pi@raspberrypi ~ \$./test_pins

```
#!/bin/sh

gpio mode 0 out
while true ; do
   echo "Escribo 1"
   gpio write 0 1
   sleep 1
   echo "Escribo 0"
   gpio write 0 0
   sleep 1
   done
```







Un caso práctico:probando

- Veamos el pulsador:
 - pi@raspberrypi ~ \$ gpio mode 1 up
 - pi@raspberrypi ~ \$ gpio read 1
 - y probar con el pulsador presionado o no





