



RASPBERRY PI ++

Página Oficial rpiplus.sourceforge.net
Experiencias y Experimentos rpiplus.blogspot.com

viernes, 7 de junio de 2013

LIBRERIA WIRING PI

[WiringPi](#) es una librería escrita en C y liberada bajo licencia GNU LGPLv3, que puede ser empleada en varios lenguajes de programación, además de C y C++, con alguna pequeña modificación en forma de adaptación.

Su principal uso es en la programación de periféricos a través de los 26 pines de General Purpose Input Output (GPIO).

WiringPi, además, ofrece un comando que permite programar y configurar los pines de la GPIO, pudiendo efectuar la lectura y escritura de los pines desde la línea de comandos o incluso, incorporándola en un Shellsript.

Instalación utilizando GIT

Git es un software de control de versiones, libre y de código abierto, diseñado para manejar todo tipo de proyectos, desde pequeños a grandes con rapidez y eficiencia.

En caso de no estar instalado GIT se debería instalar. Para ello en primer lugar hay que actualizar los repositorios y posteriormente instalarlo.

```
sudo apt-get update sudo apt-get install git-core
```

Posteriormente hay que obtener la librería mediante GIT

```
git clone git://git.drogon.net/wiringPi
```

Una vez descargado, hay que ir al directorio donde se encuentra y obtener una versión actualizada.

```
cd wiringPi
git pull origin
```

A continuación, se debe ejecutar el script de creación e instalación en el directorio de WiringPi.

```
./build
```

El nuevo script se encarga de compilar y ejecutar todo, sin que sea necesaria la intervención del usuario.

Descargando la librería.

Se va a proceder a descargar la aplicación en el directorio temporal /tmp.

```
cd /tmp
wget http://project-downloads.drogon.net/files/wiringPi.tgz
```

A continuación hay que desempaquetar el fichero.

```
tar xzf wiringPi.tgz
```

Experiencias

▼ [2013](#) (19)

► [septiembre](#) (1)

► [julio](#) (1)

▼ [junio](#) (13)

[DEFENSA DEL TRABAJO FINAL DE CARRERA](#)

[PANTALLA LCD](#)

[TECLADO HEXADECIMAL 4X4](#)

[PROGRAMANDO UN 7 SEGMENTOS DE 4 BIT](#)

[PROGRAMANDO UN BOTÓN](#)

[PROGRAMANDO LED](#)

[MATERIALES DE BRICO HARDWARE](#)

[PROGRAMANDO GPIO](#)

[VOLTAJES EN LA RASPBERRY PI](#)

[CONFIGURANDO ACCESO REMOTO](#)

[EXPLORANDO EL HARDWARE DE RASPBERRY](#)

[LIBRERIA WIRING PI](#)

[COMPILAR ENSAMBLADOR ARM Y C](#)

► [mayo](#) (1)

► [abril](#) (3)

Contribuyentes

- [Francisco J. Martinez](#)
- [Miguel Alvarez](#)

En este momento, se va a proceder a instalar las librerías relacionadas con wiringPi.

```
cd wiringPi/wiringPi
make
sudo make install
```

A continuación, se va a instalar la herramienta gpio, que permite en línea de comandos trabajar con los pines.

```
cd ../gpio make sudo make install
```

Finalmente, se van a instalar los ejemplos.

```
cd ../examples
make
```

Utilización

Para poder utilizar esta librería es necesario indicar que se va a emplear, en la primera parte del programa se incorpora la cabecera de la librería (**#include <wiringPi.h**) y en la compilación se especifica que utilice la misma (**gcc -o pruebas -l wiringPi pruebas.c**).

Principales constantes y funciones de la librería

Funciones constructoras

Hay tres funciones constructoras que permiten trabajar con la librería Wiring Pi, todas ellas devuelven -1 en caso de que se produzca un error.

- **int wiringPiSetup(void)** : Inicializa Wiring Pi y emplea el convenio de numeración de Wiring Pi.

A continuación se muestra la tabla resumen de los pines GPIO para la versión 1.

Ping wiringPi	BCM GPIO	Name	Header	Name	BCM GPIO	Ping wiringPi
-	-	3.3v	1 2	5v	-	-
8	R1:0/R2:2	SDA0	3 4	DNC	-	-
9	R1:1/R2:3	SCL0	5 6	0v	-	-
7	4	GPIO7	7 8	TxD	14	15
-	-	DNC	9 10	RxD	15	16
0	17	GPIO0	11 12	GPIO1	18	1
2	R1:21/R2:27	GPIO2	13 14	DNC	-	-
3	22	GPIO3	15 16	GPIO4	23	4
-	-	DNC	17 18	GPIO5	24	5
12	10	MOSI	19 20	DNC	-	-
13	9	MISO	21 22	GPIO6	25	6
14	11	SCLK	23 24	CE0	8	10
-	-	DNC	25 26	CE1	7	11

Para la revisión 2

Ping wiringPi	BCM GPIO	Name	Header	Name	BCM GPIO	Ping wiringPi
-	-	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	-	-

- **int wiringPiSetupGpio(void)** : Idéntica a la anterior, pero utiliza la numeración de pines de Broadcom (BCM GPIO).
- **int wiringPiSetupSys(void)** : En este caso utiliza la interfaz /sys/class/gpio, en vez de trabajar directamente sobre el Hardware.

Funciones generales

- **void pinMode(int pin, int mode)** : Sirve para especificar el pin (primer argumento) y modo (segundo argumento), que puede ser entrada (INPUT), salida (OUTPUT) o salida PWM (PWM_OUTPUT).
- **void digitalWrite(int pin, int value)** : Se utiliza para poner un pin, que previamente ha sido configurado como OUTPUT a dos

posibles valores 1 (HIGH) o 0 (LOW).

- **void digitalWriteByte(int value)** : Permite escribir en los 8 pines de la GPIO un valor.
- **void pwmWrite(int pin, int value)** : Escribe el valor del registro PWM (segundo argumento) al pin indicado (primer argumento). El valor suministrado debe estar comprendido entre 0 y 1024, además, hay que tener en cuenta que sólo soporta PWM el pin BCM_GPIO 18.
- **int digitalRead(int pin)** : devuelve el valor leído en el pin indicado (único argumento), que puede ser 1 (HIGH) o 0 (LOW).
- **void pullUpDnControl(int pin, int pud)** : establece sobre el pin indicado (primer argumento) el modo de tensión o resistencia, elevar a 3v3 (PUD_UP), tirar a tierra (PUD_DOWN) o ni elevar ni disminuir (PUD_OFF).

Funciones de control PWM

Las siglas PWM corresponden a **pulse-width modulation**, en castellano, Modulación por ancho de pulsos, consistente en cambiar el ciclo de una señal periódica. No se pueden utilizar cuando se trabaja en modo sistema (mode sys).

- **void pwmSetMode(int mode)** : El generador puede trabajar en dos modos balanceado (PWM_MODE_BAL), que es el que utiliza por defecto Raspberry Pi o marca espacio (PWM_MODE_MS) .
- **void pwmSetRange(unsigned int range)** : Establece el valor máximo del rango del registro PWM, establecido por defecto a 1024.
- **void pwmSetClock(int divisor)** : Sirve para establecer el divisor del reloj PWM.

Funciones de tiempo

- **unsigned int millis(void)** : Devuelve el numero de milisegundos que han transcurrido desde que se ha invocado a una función Wiring Pi.
- **void delay(unsigned int howLong)** : Provoca la pausa del programa durante al menos howLong milisegundos.
- **void delayMicroseconds(unsigned int howLong)** : Análogo al anterior, pero en este caso se establecen microsegundos.

Program/Thread Priority

- **int piHiPri(int priority)** : sirve para establecer la prioridad del programa o thread, que oscila entre 0 ,valor establecida por defecto y 99 valor máximo y permite la planificación en tiempo real. El valor devuelto es 0 en caso de éxito y -1 en caso de error. Sólo los programas ejecutados como root pueden cambiar su prioridad.

Funciones de interrupciones

- **int waitForInterrupt(int pin, int timeOut)** : permite establecer un tiempo de espera en milisegundos (segundo argumento) para una interrupción definida sobre un pin (primer argumento), en el caso de definir como timeOut -1 esperará indefinidamente. El valor devuelto es 0 en caso de éxito y -1 en caso de error. Esta función está obsoleta y se recomienda utilizar la que se presenta a continuación.
- **int wiringPiISR(int pin, int edgeType, void (*function)(void))** : mediante esta function se puede establecer un manejador sobre un pin, además se debe especificar si se detecta mediante un flanco de bajada (INT_EDGE_FALLING) , un flanco de subida (INT_EDGE_RISING), en ambos flancos (INT_EDGE_BOTH) o el pin no ha sido inicializado (INT_EDGE_SETUP). El valor devuelto es 0 en caso de éxito y -1 en caso de error.

Funciones de programación concurrente

Wiring Pi permite la utilización de hilos POSIX, así como mecanismos de

exclusión mutua (mutex).

- **int piThreadCreate(void * (* fn) (void *))** : Crea un Thread de una función que ha sido declarada previamente mediante PI_THREAD. La declaración sería similar a PI_TREAD(miThread){ código } y posteriormente, esta función será el argumento de la función. El valor devuelto es 0 en caso de éxito y -1 en caso de error.
- **void piLock(int keyNum)** : bloquea una clave (keyNum) cuyo valor va de 0 a 3, cuando otro thread intenta bloquear a la clave, queda a la espera de que sea liberado.
- **void piUnlock(int keyNum)** : Permite desbloquear una clave definida como en la función anterior.

Otras funciones

- **int piBoardRev(void)** : indica la versión de la placa sobre la que se está trabajando.
- **int wpiPinToGpio(int wPiPin)** : transforma el número de pin según el convenio de Wiring Pi al convenio de BCM GPIO.
- **void setPadDrive(int group, int value)** : establece la "fuerza" de los drivers para un grupo de pins.

Publicado por Miguel Alvarez en [22:02](#)

6 comentarios:



Jose de Sabadell 17 ago. 2013 12:23:00

Muy bueno este documento. Buen trabajo y felicidades por el trabajo.

[Responder](#)

[Respuestas](#)



Miguel Alvarez 13 sept. 2013 17:16:00

muchas gracias José, uno de los objetivos que perseguíamos es que fuera de utilidad a otras personas. Un saludo

[Responder](#)



Carlos Moran Encalada 7 feb. 2015 16:32:00

Buenas tardes es un muy buen post, tengo una inquietud que radica en el uso de simulink con la raspberry pi para poner en funcionamiento este es necesario tener instalado la libreria wiringpi en matlab/simulink de ser asi como lo deberia instalar ya que tengo instalado el programa de matlab/simulink en una màquina que tiene el sistema operativo windows y de verdad no se como instalar esta libreria.

[Responder](#)



Miguel Alvarez 7 feb. 2015 21:06:00

No había oído hablar de Simulink. Intenta agregar la librería (puedes descargarla desde <http://wiringpi.com/download-and-install/> donde pone Plan B). Un saludo

[Responder](#)



Carlos Moran Encalada 23 feb. 2015 20:02:00

Buenas tardes lo que quería saber es que si para usar la tarjeta de raspberry Pi en Matlab/Simulink necesito la libreria de wiringPi . Cabe recordar que estoy usando Matlab/Simulink en Windows 7. Muchas gracias.

[Responder](#)



Ramiro Seliman 29 oct. 2017 13:47:00

Hola , como estas? , quisiera saber si esta libreria se puede instalar en w7 y Lazarus , estoy tratando de hacer compilacion cruzada a Rasbperry pi 3 ...Mil Gracias

[Responder](#)

Introduce tu comentario...

Comentar como: Cuenta de Goo ▼

Publicar

Vista previa

[Entrada más reciente](#)

[Página principal](#)

[Entrada antigua](#)

Suscribirse a: [Enviar comentarios \(Atom\)](#)

Se distribuyen los materiales bajo licencia Copyleft. Tema Sencillo. Con la tecnología de [Blogger](#).