



ROB 2019-20

Informatika Fakultatea, EHU  
Konputagailuen Arkitektura eta Teknologia Saila

## ROBÓTICA, SENSORES Y ACTUADORES

### PRÁCTICA 1.b

#### Programación de threads en la Raspberry Pi2

##### OBJETIVOS

Aprender a

- programar los pines GPIO de la Raspberry Pi para realizar funciones de entrada/salida mediante threads e interrupciones



##### MATERIAL NECESARIO

###### Hardware

- Raspberry Pi 2
- Protoboard, LEDs, pulsador...

###### Software

- NetBeans 8.0.2
- Librerías WiringPi

###### Manuales (en Moodle)

- RaspberriPi para RSA
- API WiringPi : <http://wiringpi.com/reference/>

## 1. Uso de las librerías

En esta segunda parte de la práctica 1 se hace uso de las siguientes funciones:

- `int piHiPri (int priority) ;`
- `int waitForInterrupt (int pin, int timeOut) ;`
- `int wiringPiISR (int pin, int edgeType, void (*function)(void)) ;`
- `int piThreadCreate (name) ;`
- `piLock (int keyNum) ;`
- `piUnlock (int keyNum) ;`

Hay más información sobre estas funciones en la documentación sobre WiringPi disponible en eGela y en: <http://wiringpi.com/reference/>

## 2. Desarrollo de la práctica

### a. Programación con threads de Wiring Pi

- Crear un proyecto NetBeans para controlar las E/S de la GPIO
- Conectar un LED a la GPIO de la Raspberry Pi
- Escribir un programa que cree dos *threads*: uno que encienda el led y otro que lo apague.
  - Acceder a la(s) variable(s) compartida(s) en exclusión mutua.
  - Introducir esperas suficientes para que dé tiempo a ver el efecto de cada thread.
  - Asignar más prioridad relativa al proceso que enciende el led. Después asignar más prioridad relativa al proceso que apaga el led ¿hay alguna diferencia?

### b. Programación con threads e interrupciones de Wiring Pi

- Conectar un LED y un pulsador a la GPIO de la Raspberry Pi
- Escribir un programa que cree dos threads: uno que espere una interrupción producida por el pulsador y otro que encienda o apague el led cuando se produce la interrupción.
  - Acceder a la(s) variable(s) compartida(s) en exclusión mutua.
  - Introducir esperas suficientes para que dé tiempo a ver el efecto de cada thread.
  - Asignar más prioridad relativa al proceso que atiende la interrupción. Después asignar más prioridad relativa al proceso que enciende, apaga el led ¿hay alguna diferencia?

## 3. Resultados

- a. Antes de desmontar los circuitos, es necesario hacer una demostración al profesor del funcionamiento de todos los programas
  - b. Cada grupo redactará y subirá a eGela un breve informe con el siguiente contenido:
    - Código fuente **comentado** de **todos los programas**
    - Informe de la práctica redactado según la plantilla “Informe de la Práctica 1.b” disponible en eGela
- Comprimir todos los ficheros en un único zip con el siguiente nombre:
- **Practica1b\_GrupoXY\_Apellido1\_Apellido2**