

Guía de Laboratorio N°2

Manejo de E/S: Puertos e Interrupciones sobre RB-Pi

1. Objetivos

- Adquirir experiencia real en la programación de plataformas basadas en microprocesadores no convencionales (MIPS, ARM, etc.)
- Que el alumno adquiera la habilidad de consultar manuales, hojas de datos de microprocesadores y documentación de librerías de software disponibles.
- Específico: Aplicar conocimientos previamente adquiridos sobre técnicas de manejo de periféricos (E/S programada y mediante interrupción) en un sistema real a bajo nivel.

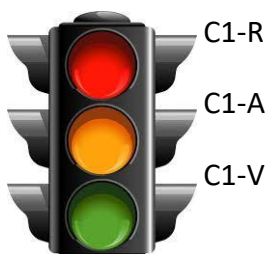
2. Desarrollo

Elementos de hardware necesarios:

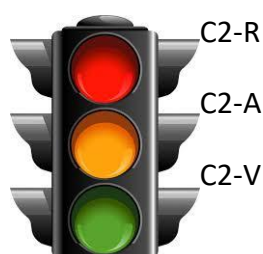
- Placa Raspberry Pi 1
- 6 Módulos para conexión GPIO del tipo Led
- 1 Módulo para conexión GPIO del tipo Switch

Desarrollar un proyecto en lenguaje C que implemente, mediante la utilización de los recursos de hardware GPIO y Timer de una placa Raspberry Pi (Rpi), un **controlador de semáforo de secuencia programable** de 2 tiempos. El controlador posee 2 canales (C1 y C2) de 3 salidas cada una (R, A, V) (correspondientes a 2 semáforos de 3 luces c/u: rojo, amarillo y verde) y un canal de entrada tipo pulsador (WS), tal como se muestra en el diagrama.

Semáforo 1 (C1)



Semáforo 2 (C2)



Switch (SW)



El controlador debe soportar la carga de **3 modos de funcionamiento normales** y un **modo de emergencia**. La secuencia de encendido de luces y sus tiempos de duración, debe ser configurada para cada modo normal. Solo el modo de emergencia tiene funcionamiento fijo: encendido intermitente de las luces amarilla de ambos canales en intervalos de tiempo de 1seg (1seg ON -> 1seg OFF -> ...). Las estructuras de datos utilizadas para cargar las secuencias y duraciones de los diferentes modos son libres.

Las reglas de funcionamiento generales son:

- El controlador ejecuta en forma indefinida la secuencia establecida por el modo actual (modo 1, modo 2 o modo 3).
- El cambio del modo actual se realiza pulsando una vez el SW (modo1 → modo 2 → modo3 → modo1...)
- El controlador debe corroborar EN FORMA PERMANENTE que nunca se haya establecido una combinación **NO PERMITIDA** de luces:
 - 1 - Luces verdes de ambos canales encendidas en simultáneo
 - 2- Cualquier combinación de una luz verde de un canal y una luz amarilla del otro canal encendidas en simultáneo.
 - 3 – Todas las luces apagadas de ambos canales en simultáneo.
- Si se detecta la existencia de una combinación NO PERMITIDA de luces, el controlador debe pasar inmediatamente a modo de emergencia, y ejecutar esta secuencia de forma indefinida (reset o apagado).

3. Conexionado

Señal	Tipo	Puerto RB-Pi	Nivel Lógico
C1-R	Salida (OUT)	GPIO 11	"1"-> Encendido "0"-> Apagado
C1-A	Salida (OUT)	GPIO 10	
C1-V	Salida (OUT)	GPIO 17	
C2-R	Salida (OUT)	GPIO 7	
C2-A	Salida (OUT)	GPIO 24	
C2-V	Salida (OUT)	GPIO 18	
SW	Entrada (IN)	GPIO 14	"0"-> Suelto "1"-> Presionado

4. Recomendaciones

El proyecto debe incluir las operaciones de configuración inicial de hardware y control de los recursos utilizados. El control de tiempos de cada ciclo debe ser preciso y es considerado como un parámetro crítico del sistema, por lo que debe implementarse mediante la utilización de un módulo Timer (ARM Timer) por interrupción.

El estado del SW y el de los canales de salida pueden procesarse mediante software "pooling"

Se tendrá en especial consideración la eficiencia en términos del tamaño del código generado.

5. Referencias

Documentación, información adicional y herramientas (compiladores) disponibles en:

Tutorial: *Raspberry-Pi Bare Metal Programming in C*

<http://www.valvers.com/open-software/raspberry-pi/step01-bare-metal-programming-in-cpt1/>