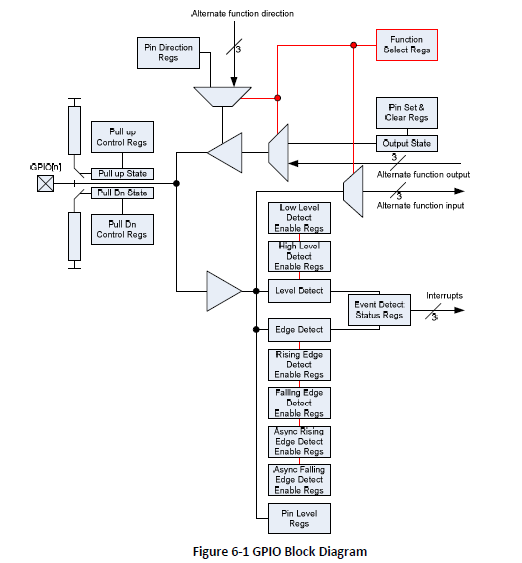
|  |  |
| --- | --- |
| PROGRAMA ACADÉMICO: **Ingeniería Informática** | |
| Módulo: **Laboratorio de Arquitectura de Hardware** | |
| Área : **Infraestructura** | Docentes: **Carlos Andrés Madrigal** |

1. **Practica #2. Guía de Laboratorio N. 2. Módulo de Puertos GPIO, Interrupciones y Retardos de Tiempo.**
2. **Objetivo de Aprendizaje**

* Comprender la arquitectura del manejo de los puertos de entrada y salida GPIO.
* Comprender la arquitectura del manejo de las interrupciones.
* Comprender la arquitectura del manejo del timer.

1. **Fundamentos Teóricos**

**Modulo GPIO: General Purpose Input/Ouput**

****

Este módulo permite controlar dispositivos de entrada y salida desde los pines de la Raspberry. La versión 3 de la Raspberry pi cuenta con 40 pines que pueden ser usados como entrada, salida o alguna función alternativa.



Hay 2 maneras de numerar los pines de I/O en la Raspberry Pi 3 usando la librería RPi.GPIO. Usando el sistema de numeración BOARD, los cuales son los pines físicos del conector de la tarjeta y el otro es usando la numeración BCM que es referida al SoC Broadcom. Por facilidad se recomienda trabajar con numeración BOARD.

**Multiplexación De Señales en los Puertos.**

Para optimizar el funcionamiento en un empaquetado tan pequeño y con pocos pines es necesario multiplexar las señales en los pines para tener en esta mayor funcionalidad. Cabe resaltar que el chip SoC BCM2837 tiene 51 líneas de propósito general, las cuales no todas están disponibles en la tarjeta Raspberry pi3.

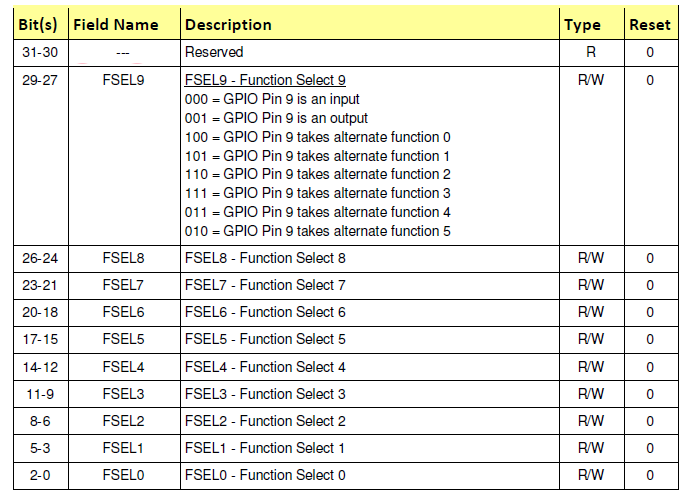


*Para mayor información revisar el datasheet del SoC BCM2835/2836/2837*

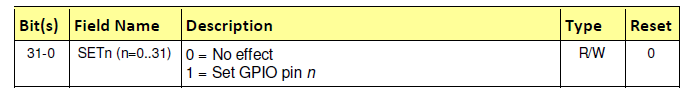
**REGISTROS DE CONFIGURACIÓN**

El GPIO tiene 41 registros de configuración, cada uno de 32 bits. Acá explicaremos los más relevantes.

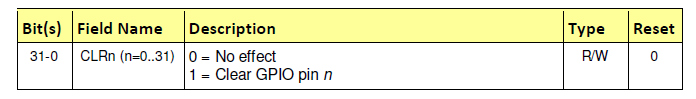
**GPIO Function Select Registers (GPFSELn) (0-5):** Registro de selección de función del GPIO. Es usado para definir la operación de los pines I/O. Cada uno de los pines tiene al menos 2 alternativas de uso.



**GPIO Pin Output Set Registers (GPSETn) (0-1): Registro de Fijado Set a la Salida.** Coloca en 1 los pines del GPIO configurados en este registro.

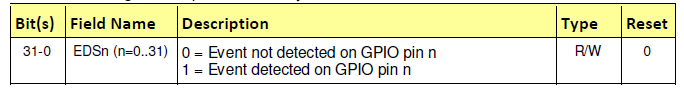


**GPIO Pin Output Clear Registers (GPCLRn) (0-1): Registro de Fijado Clear a la Salida.** Coloca en 0 los pines del GPIO configurados en este registro.

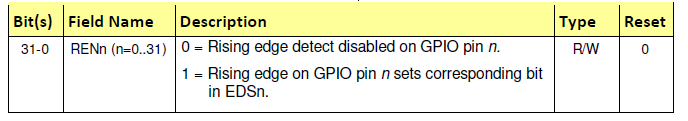


**GPIO Pin Level Registers (GPLEVn)(0-1): Registro de Nivel Actual del Pin.** Retorna el valor actual del pin.

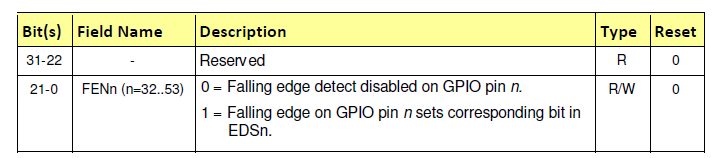
**GPIO Event Detect Status Registers (GPEDSn) (0-1): Registro de Estados de Eventos del GPIO:** Almacena el nivel y el flanco de un evento sobre el GPIO. Este registro puede generar una petición de interrupción al controlador de interrupciones del procesador. Los bits de este registro son borrados escribiendo sobre ellos 1.



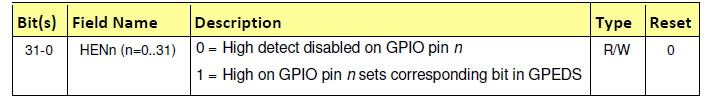
**GPIO Rising Edge Detect Enable Registers (GPRENn) (0-1): Registro de Habilitación de Detección de Flanco de Subida.** Permite definir los pines a los cuales se les habilita la detección de eventos por flanco de subida. La detección de flancos es sincronizada con el sistema de clock del procesador.



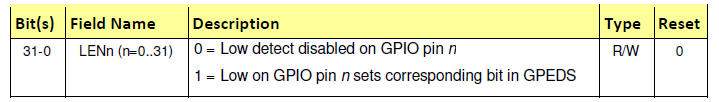
**GPIO Falling Edge Detect Enable Registers (GPFENn) (0-1): Registro de Habilitación de Detección de Flanco de Bajada.** Permite definir los pines a los cuales se les habilita la detección de eventos por flanco de bajada. La detección de flancos es sincronizada con el sistema de clock del procesador.



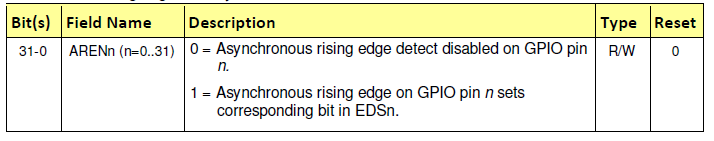
**GPIO High Detect Enable Registers (GPHENn)(0-1): Registro de Habilitación de Detección de Nivel Alto.** Permite definir los pines a los cuales se les habilita la detección de eventos por nivel alto. La detección de flancos es sincronizada con el sistema de clock del procesador.



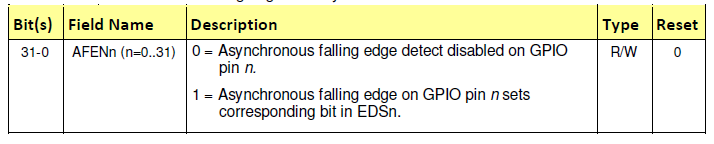
**GPIO Low Detect Enable Registers (GPLENn)(0-1): Registro de Habilitación de Detección de Nivel Bajo.** Permite definir los pines a los cuales se les habilita la detección de eventos por nivel bajo. La detección de flancos es sincronizada con el sistema de clock del procesador.



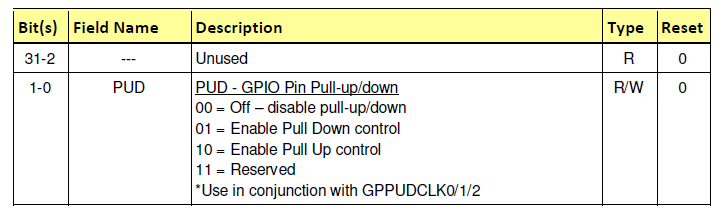
**GPIO Asynchronous rising Edge Detect Enable Registers (GPARENn)(0-1): Registro de Habilitación de Detección de Flanco de Subida Asíncrono.** Permite definir los pines a los cuales se les habilita la detección de eventos por flanco de subida de manera asíncrona. Esto hace que señales con flancos muy cortos de duración puedan ser detectados.

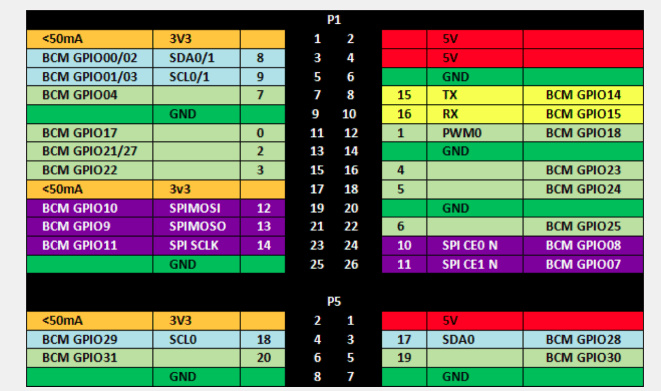
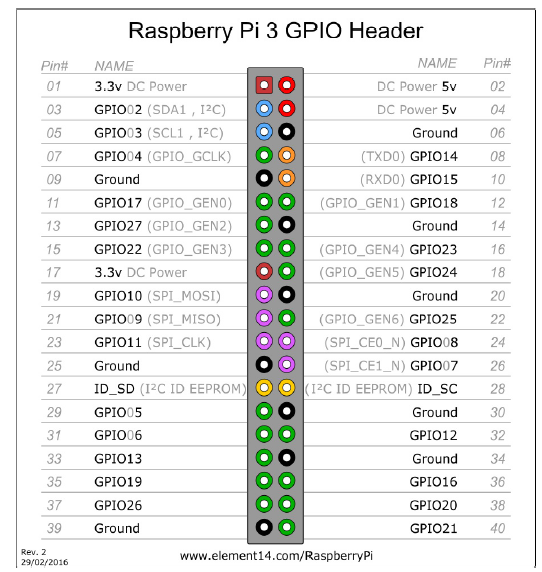


**GPIO Asynchronous Falling Edge Detect Enable Registers (GPAFENn)(0-1): Registro de Habilitación de Detección de Flanco de Bajada Asíncrono.** Permite definir los pines a los cuales se les habilita la detección de eventos por flanco de bajada de manera asíncrona. Esto hace que señales con flancos muy cortos de duración puedan ser detectados.



**GPIO Pull-up/down Register (GPPUD): Registro de Habilitación del Dispositivo de Pull del GPIO.**

****



**Configuración GPIO con la librería RPi.GPIO**

* Tipo de Numeración

Configura la numeración a usar:

GPIO.setmode(GPIO.BCM)

GPIO.setmode(GPIO.BOARD)

* Entrada/Salida

Configura un canal específico como entrada o como salida

GPIO.setup(canal, GPIO.IN)

GPIO.setup(canal, GPIO.OUT)

GPIO.setup(11, GPIO.IN)

GPIO.setup(12, GPIO.OUT)

Es posible inicializar la salida de un pin desde su configuración

GPIO.setup(canal, GPIO.OUT, initial=GPIO.HIGH)

Configurar más de un canal al tiempo.

canales = [11,12]

GPIO.setup(canales, GPIO.OUT)

Generando Salidas

GPIO.output(canal, GPIO.LOW)

GPIO.output(canales, GPIO.LOW)

GPIO.output(canales, (GPIO.LOW, GPIO.HIGH))

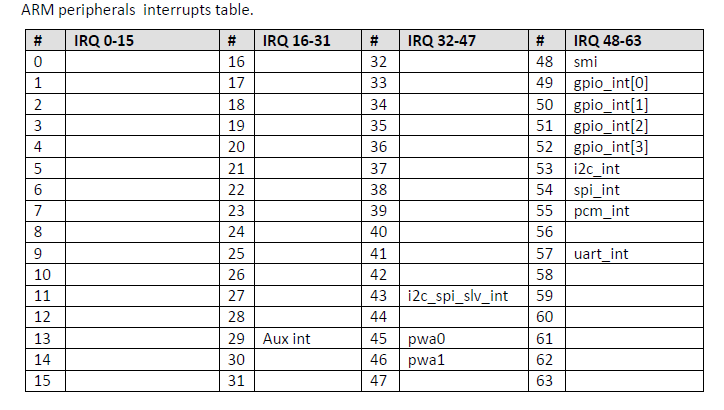
Limpiar Canales

GPIO.cleanup()

**INTERRUPCIONES**

as interrupciones son eventos que pueden ser internos o externos al procesador que paran la ejecución del programa para saltar a atender una rutina de atención de la interrupción. Las interrupciones del procesador están vectorizadas, lo que significa que tienen una jerarquía. Esto con el fin, por si se presentan 2 interrupciones al mismo tiempo, poder ejecutar primero la de mayor prioridad.

El ARM tiene 2 tipos de fuentes de interrupciones. Una por los periféricos de la GPU y los otros por los de periféricos del ARM. Estos son algunas de las interrupciones.



Las interrupciones del GPIO son controladas a través de los registros **GPAFENn, GPARENn, GPLENn, GPFENn, GPHENn, GPRENn, GPEDSn.** Sin embargo su utilización mediante el modulo RPi.GPIO es más simple.

**Resistencias Internas de Pull Up/Down**

Es posible habilitar internamente resistencias de pull up o pull down sobre pines que han sido configurados como de entrada de datos. Esto se realiza a través del registro GPPUD como se vio anteriormente.

Usando la librería Rpi.GPIO se realiza de la siguiente manera:

GPIO.setup(canal, pull\_up\_down=GPIO.PUD\_UP)

GPIO.setup(canal, pull\_up\_down=GPIO.PUD\_UP)

**Habilitando las Interrupciones por flanco.**

Para detectar el cambio de estado en una entrada existen 3 funciones en la librería RPi.GPIO. La sensibilidad del evento puede programarse para GPIO.RISING, GPIO.FALLING, GPIO.BOTH

* GPIO.wait\_for\_edge() bloquea la ejecución del programa hasta que un flanco es detectado. Lo cual significa que se puede convertir en un ciclo infinito si no llega un flanco al procesador.

GPIO.wait\_for\_edge(canal, GPIO.RISING)

# Espera 10seg por un flanco de subida

Event= GPIO.wait\_for\_edge(canal, GPIO. RISING, timeout=10000)

if Event is None:

print('Tiempo completado')

else:

print('Flanco detectado')

* GPIO.Event\_detected() Permite detectar un evento por interrupción. Esta hecho para ejecutarse en un ciclo
* GPIO.add\_event\_detect(canal,GPIO.RISING,callback=interruptGPIO) Permite detectar un evento por interrupción y genera un llamado callback a una función.

def interruptGPIO(canal):

print('Flanco detectado')

GPIO.add\_event\_detect(canal, GPIO. RISING, callback= interruptGPIO)

**Antirrebotes por software**

Debido a que al presionar un pulsador, se puede generar varios flancos, esto por ser un elemento mecánico que genera rebotes. Es posible eliminar los rebotes por software como sigue:

GPIO.add\_event\_detect(canal, GPIO. RISING, callback= interruptGPIO, bouncetime=200)

El tiempo hace que cuando se genere un flanco se ignoren los flancos por 200ms.

**Recursos**

Computador con acceso a internet.

Single Board Computer Raspberry pi 3

Cable MiniUSB de alimentación de 3A.

Tarjeta SD de 16GHz con el sistema operativo raspbian instalado.

Cable conversor HDMI-VGA

4 Displays 7 Segmentos de ánodo común.

Pulsador

Pinzas, cortafrío y cables.

Resistencias

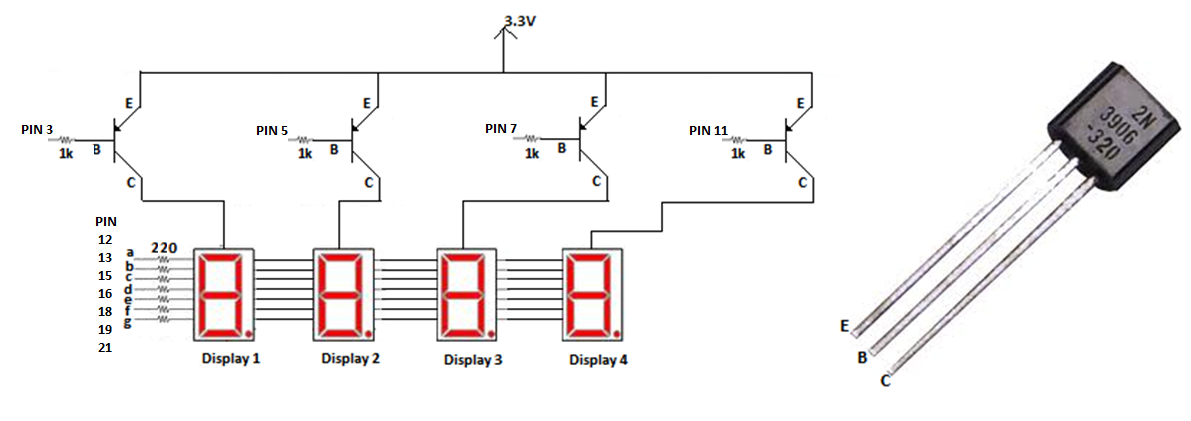
Protoboard

Ayuda: http://raspi.tv/2015/how-to-drive-a-7-segment-display-directly-on-raspberry-pi-in-python

1. **Guía de desarrollo**

**PROCEDIMIENTO**

1. Desarrollar un programa que permita visualizar los números del 0 al 9 con retardos de tiempo en un display 7 segmentos de ánodo común.
2. Generar un contador de interrupciones y su visualización por 4 display 7 segmentos.
3. Programar un reloj de horas y minutos y visualizar por 4 displays 7-segmentos de ánodo común de manera multiplexada, si se presiona un pulsador se debe reiniciar el conteo.



1. Se debe diseñar un sistema que permita elegir entre 5 tareas a realizar.
2. Visualizar Reloj.
3. Actualizar Reloj.
4. Ingresar Alarmas.

El usuario podrá escoger cada una de las tareas mediante los switches SW0 y SW1 de la siguiente forma:

|  |  |  |
| --- | --- | --- |
| SW0 | SW1 | TAREA |
| Off | Off | 1 |
| On | Off | 2 |
| Off | On | 3 |

**Tarea 1. Visualizar Reloj.** En esta tarea se visualiza la hora actual en el formato militar.

**Tarea 2. Actualizar Reloj.** Para cambiar la hora solo se hace en formato militar. Cuando se esté en esta tarea los displays deben titilar de acuerdo al display a cambiar, primero debe empezar a titilar el display de las decenas de la hora, si pasados 5 segundos no ha ocurrido ningún cambio en este display o si se presiona el pulsador ubicado en uno de los pines configurado como interrupción 2 veces en un intervalo menor a 300ms, debe empezar a titilar el segundo display correspondiente a las unidades de la hora y así sucesivamente para los demás displays. Tener en cuenta que los cambios son solo para 1 display y que cuando se desea cambiar el primer display correspondiente a las decenas de la hora, este display podrá cambiar a 0,1 ó 2 dependiendo del valor del segundo display, de lo contrario solo podrá cambiar a 0 ó 1 dependiendo del valor del segundo display. De igual forma para los demás displays. El valor se cambia mediante el número de pulsadas de un pulsador ubicado en uno de los pines configurado como interrupción.

**Tarea 3. Ingresar Alarma.** En esta opción se permite el ingreso de 20 alarmas. Cuando el reloj alcance el valor de cada alarma, se debe generar un sonido que lo indique por un tiempo de 27 seg. Este sonido tendrá periodo de 4.5seg y un tiempo en alto de 2seg. Si el usuario presiona el pulsador de la interrrupción antes de los 27 seg el sonido debe parar. Para cambiar la alarma se realiza de la misma forma que en cambiar hora, con la diferencia que antes de empezar a titilar el display de las decenas de la hora, se le da un tiempo de 5 seg al usuario para que cambie de número de alarma.

Nota: El manejo de los displays debe ser multiplexado. Cuando se menciona presionar el pulsador, hace referencia a presionar un pulsador ubicado en uno de los pines configurado como interrupción. Cuando se este cambiando la hora en un display, este debe titilar.

1. **Entregables**

**Informe**

* Este debe ser desarrollado de acuerdo al formato IEEE para la redacción de artículos científicos. "www.ieee.org/documents/MSW\_USltr\_format.doc"
* Se debe adjuntar archivo \*.DSN de la simulación.

1. **Criterios de Evaluación**

|  |  |
| --- | --- |
| **Aspecto a evaluar** | **Desarrollo del laboratorio y preguntas en el desarrollo** |
| **Escala de Valoración** | **Criterios** |
| 5 | Demuestra total comprensión del problema. Todos los requerimientos de la tarea están incluidos en la desarrollo de la práctica. |
| 4 | Demuestra considerable comprensión del problema. Todos los requerimientos de la tarea están incluidos en la desarrollo de la práctica. |
| 3 | Demuestra comprensión parcial del problema. La mayor cantidad de requerimientos de la tarea están incluidos en la desarrollo de la práctica. |
| 2 | Demuestra poca comprensión del problema. Muchos de los requerimientos de la tarea faltan en el desarrollo de la práctica.. |
| 1 | No comprende el problema |
| 0 | No responde. No intentó hacer la tarea. |

|  |
| --- |
| **Observaciones:** |
|  |
|  |
|  |
|  |
|  |
|  |