

# Laboratorio 2

---

## Competencias a desarrollar

Mediante esta actividad los estudiantes diseñarán, implementarán y depurarán un programa en lenguaje ensamblador del *Raspberry Pi* que tenga comunicación con circuitos digitales sencillos.

## Especificaciones

1. El laboratorio debe realizarse en parejas escogidas por los estudiantes. La mínima evidencia de copia entre grupos conlleva un 0 (cero) como nota.
2. **Fecha entrega y presentación:**
  - a. **Sección 10:** 8 de septiembre
  - b. **Sección 20:** 8 de septiembre
  - c. **Sección 30:** 9 de septiembre

## Condiciones de Entrega

Cada grupo deberá entregar:

1. **Programa fuente:** enviar a la actividad de *Blackboard* todos los archivos fuente en la fecha de entrega establecida.
2. **Video:** enviar el enlace al video que muestra el funcionamiento del circuito armado y conectado a la Raspberry Pi, de acuerdo al temario elegido. En el video deben estar los integrantes del grupo explicando el funcionamiento (no el programa).
3. **Diario reflexivo:** Al finalizar el laboratorio cada estudiante de forma individual deberá contestar el Diario Reflexivo.

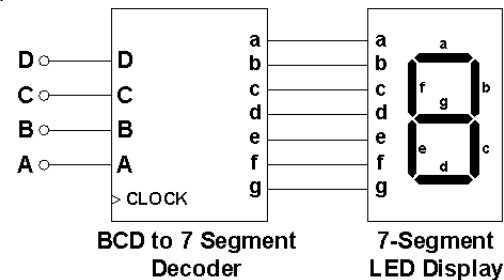
## Instrucciones

Su laboratorio consiste en elaborar un programa en lenguaje ensamblador de ARM en el *RaspberryPi* que tenga una interfaz con un circuito utilizando los puertos GPIO. No se requiere construir una interfaz gráfica (GUI) para interactuar con el usuario, el objetivo principal es que su programa funcione y el usuario se comunique con él utilizando el circuito. Para el diseño del circuito no se pueden utilizar compuertas lógicas, a menos que el temario lo indique.

## Temarios

### 1. Reloj despertador de “hora”

Diseñar y construir un circuito que sea un reloj despertador de “Hora”. La hora debe ser mostrada en utilizando dos 7 segmentos, de cátodo o ánodo común. Pueden utilizar el convertidor de BCD a siete segmentos cátodo común SN74LS48 o ánodo común SN74LS47. La forma en que funciona un BCD junto con un 7 segmento, se muestran en la figura 1. El reloj debe tener como mínimo tres botones, para poder configurar la hora y establecer la alarma a una cierta hora. La alarma puede ser representada con un LED que se encienda y apague, a una cierta frecuencia, cuando se active la alarma. Puntos extras si se incluye una bocina, y/o los minutos, y/o poder almacenar 3 diferentes alarmas.



**Figura 1.** Funcionamiento de un BCD con un 7 segmento. (1)

## 2. Control de intensidad de volumen de una bocina.

Diseñar y construir un circuito que maneje el volumen de una bocina. Este debe contar con tres botones como mínimo, para manejar la intensidad del sonido de la bocina, y poder guardar 3 porcentajes de volumen predefinidos. El porcentaje de la intensidad del tono que emita la bocina, debe ser representada en 4 leds, dando un rango de 0 a 15. Puntos extras desplegar el valor en 7 segmentos y/o almacenar 3 intensidades predefinidas.

## 3. Timer: contador descendente.

Diseñar y construir un circuito que sea un "Timer" de dos dígitos, mostrados en dos 7 segmentos. El valor del "timer" y el inicio del mismo, debe ser configurado utilizando mínimo tres botones, para poder configurar el valor y comenzar el conteo regresivo. Pueden utilizar el convertidor de BCD a siete segmentos cátodo común SN74LS48 o ánodo común SN74LS47. La forma en que funciona un BCD junto con un 7 segmento, se muestran en la figura 1. Al momento de llegar a "0" se debe mostrar mediante una señal luminosa. (ejemplo: un led o se enciende al mismo tiempo todos los segmentos de los displays). Puntos extras si se incluye una bocina, y/o almacenar distintas cuentas regresivas.

## 4. Control de posiciones de un Servomotor

Diseñar y construir un circuito que pueda controlar un número determinado de posiciones de un servo motor, como mínimo 4, utilizando dos botones para incrementar o disminuir dicha posición. Para alcanzar su objetivo debe investigar acerca del funcionamiento de los servomotores y la modulación por ancho de pulsos (Pulse Width Modulation, PWM, por sus siglas en inglés). En las figuras 5 y 6, se muestra un ejemplo del control de posiciones de un servo mediante PWM y sus ángulos respectivos, así como la conexión de un servo motor. Puntos extras si se pueden almacenar 4 distintas posiciones y/o mostrar la posición en display de 7 segmentos.

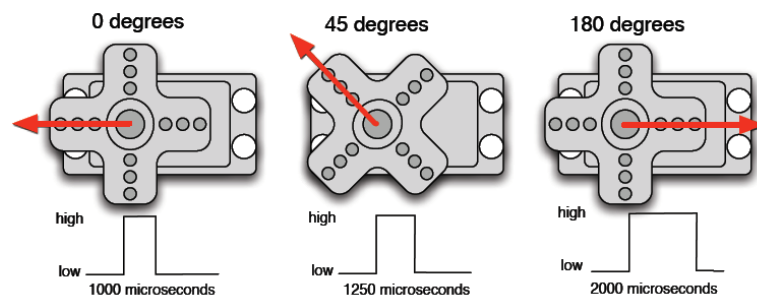


Figura 2. Ejemplo de como el PWM se relaciona con la posición de un servo (2)

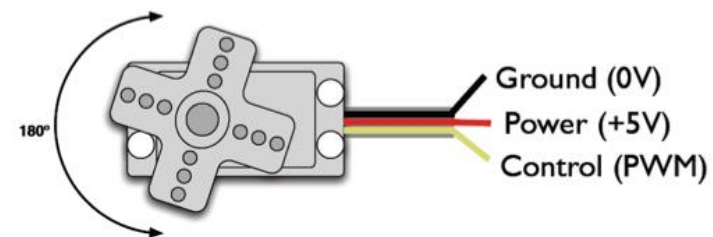


Figura 3. Ilustración del detalle de la alimentación y señal de control de un servo. (2)

## 5. Controlador de motores de medio paso OK

Diseñar y construir un circuito que pueda controlar un motor de paso unipolar. El circuito debe permitirle al operador hacer funcionar y controlar la dirección de un motor de paso con una secuencia de paso simple o paso medio (ver figura2) mientras este mantenga presionado un interruptor. El circuito debe poder mover un motor de paso, utilizando el circuito mostrado en la figura 4. Puntos extras si se puede variar la velocidad de rotación, y/o poder almacenar 5 posiciones del rotor, y/o escoger entre paso medio (figura 2) o paso doble (figura 3).

Index	1a	1b	2a	2b
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1
9	1	0	0	0
10	1	1	0	0
11	0	1	0	0
12	0	1	1	0
13	0	0	1	0
14	0	0	1	1
15	0	0	0	1
16	1	0	0	1

Half Step Sequence

Figura 4. Secuencia de paso simple

Index	1a	1b	2a	2b
1	1	0	0	1
2	1	1	0	0
3	0	1	1	0
4	0	0	1	1
5	1	0	0	1
6	1	1	0	0
7	0	1	1	0
8	0	0	1	1

Alternate Full Step Sequence  
(Provides more torque)

Figura 5. Secuencia de paso doble

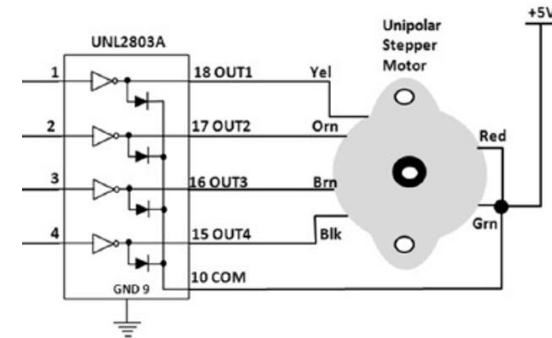


Figura 6. Conexión de un Stepper Unipolar de 4 fases usando un driver UNL2803A.

## Escala valorativa de Evaluación

CRITERIOS	DESCRIPCIÓN	PUNTOS
<b>Funcionamiento total del laboratorio</b>	La integración del circuito con el programa <b>cumple con todos los requerimientos solicitados</b> en el laboratorio. <b>Adjuntaron el video del funcionamiento explicado por los dos integrantes</b> en la actividad de <i>Blackboard</i> .	60
<b>Orden y cableado del circuito</b>	El orden del cableado del circuito permite la fácil <b>identificación de las rutas de datos</b> . Todas las <b>conexiones</b> están completamente <b>ordenadas</b> , se encuentran sobre la superficie del <i>protoboard</i> y <b>no utilizan jumpers provisionales</b> .	10
<b>Documentación y orden de los programas fuente (.s)</b>	La documentación incluye encabezado y comentarios representativos en los bloques de código más importantes, <b>indicando claramente el objetivo del laboratorio y detalles de cómo conectar el circuito</b> . Los nombres de las variables son significativos. La presentación del programa es muy clara y ordenada, y utiliza una tabulación adecuada	10
<b>Uso de subrutinas</b>	Los programas fuente están totalmente estructurados con <b>subrutinas</b> que siguen el estándar ABI para paso de parámetros y uso de registros.	10
<b>Reflexión individual</b>	El estudiante realizó una reflexión completa del laboratorio, con aportes significativos para mejorar su aprendizaje en/antes de la fecha solicitada y contestó todas las preguntas planteadas.	10
<b>TOTAL</b>		<b>100</b>

## Bibliografía

1. Macao.communications.museum. (2016). *Displays*. [online] Available at: <http://macao.communications.museum/eng/exhibition/secondfloor/MoreInfo/Displays.html> [Accessed 1 Sep. 2016].
2. Sweb.cityu.edu.hk. (2016). *SM2608 Hardware Hacking Workshop | U7*. [online] Available at: <http://sweb.cityu.edu.hk/sm2608/styled-6/> [Accessed 1 Sep. 2016].