

Proyecto 4

Competencias a desarrollar

Diseñar e implementar un programa en lenguaje ensamblador del *Raspberry Pi* que tenga comunicación con circuitos digitales sencillos.

Especificaciones

1. El proyecto debe realizarse en **parejas** escogidas por los estudiantes, el docente asignará los temarios de forma aleatoria.
2. **Fecha entrega y presentación: 23 y 24 de mayo** en los periodos de clase.

NOTA: La mínima evidencia de copia entre grupos conlleva un 0 (cero) como nota.

Materiales

- Raspberry Pi, cargador y tarjeta SD con Raspbian instalado
- *Protoboard* (1 galleta mínimo)
- Cables de conexión para *protoboard*
- Jumpers macho-macho y macho-hembra
- Corta alambre, pela alambre, pinzas, alicates
- Los leds necesarios, cada uno con resistencia de 270Ω
- Los push-button (botones) necesarios, cada uno con resistencia del $10k\Omega$
- De acuerdo al temario:
 - Display 7 segmentos con sus respectivos convertidores de BCD a siete segmentos cátodo común SN74LS48 o ánodo común SN74LS47
 - Bocina
 - Servomotor
 - Motor stepper

Los lugares donde pueden conseguir los componentes son:

- **Electrónica ABP:** <http://www.electronicabp.com>. Calzada San Juan 30-81 Z7, Plaza Florencia Local 7. Tel. 2474 1522
- **Electrónica CEF:** 3 Av 12-52 Zona 1. Tel 2220 5623
- **ABC Electrónica:** <https://www.abcelectronica.net>. 11 avenida 15-71 zona 1. Tel 2221 4265
- **Servicios Técnicos Sagastume S.A:** Blvd Los Próceres 12-25 Z-10. Tel 2333 5941

Condiciones de Entrega

Cada grupo entregará lo siguiente en la actividad de *Canvas* correspondiente:

1. **Programas fuente:** todos los archivos que contienen los programas fuente.
2. **Video:** enlace al video que muestre el funcionamiento del circuito armado y conectado a la Raspberry Pi, de acuerdo al temario asignado. En el video deben estar los integrantes del grupo explicando y demostrando el funcionamiento del proyecto (no el código del programa).
3. **Documento de análisis y diseño:** No imprima este documento, solo envíelo a la actividad de *Canvas*. El documento **debe ser PDF** e incluir lo siguiente:
 - a. **Una tabla con la estructura de su programa:** la tabla debe incluir los archivos fuente realizados por el grupo para el proyecto (extensión .s) y sus respectivas subrutinas, explicando los parámetros de entrada, la función que realiza y los parámetros de salida de cada una. No es necesario agregar los programas de las bibliotecas de la U. de Cambridge.
 - b. **Diagrama de flujo y/o Algoritmo narrativo** general del funcionamiento de su proyecto, utilizando la simbología estándar de diagramas y/o algoritmos.
 - c. **Diagrama del circuito** diseñado para el proyecto, realizado en alguna herramienta de software tipo *multiSim* o similar.
 - d. **Bibliografía** de mínimo 2 sitios confiables, citada de acuerdo al formato APA o IEEE.
4. **Diario reflexivo:** Al finalizar el proyecto cada estudiante de forma individual deberá contestar el Diario Reflexivo.

Instrucciones

Este proyecto 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 “segundos”

Diseñar y construir un programa que controle un circuito que sea un reloj despertador de “Segundos” (0 a 60) de 2 dígitos. Los segundos debe ser mostrados 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.

Al iniciar el programa mostrará dos opciones: programar los segundos de la alarma por medio de hardware (botones) o por medio del programa. En la primera opción (por hardware), el reloj debe tener como mínimo tres botones, para poder configurar los segundos en cada dígito y establecer la alarma a ciertos segundos (botón para determinar el dígito a cambiar, botón para aumentar el dígito de forma circular (de 0 a 9, regresando a 0), botón para iniciar la alarma en 0). En la segunda opción (por programa), se solicitará al usuario que ingrese los segundos de la alarma por medio del teclado. Al dar enter, la alarma inicia en 0. En ambos casos, los display 7 segmentos irán mostrando el avance hasta llegar al segundo programado como alarma.

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.

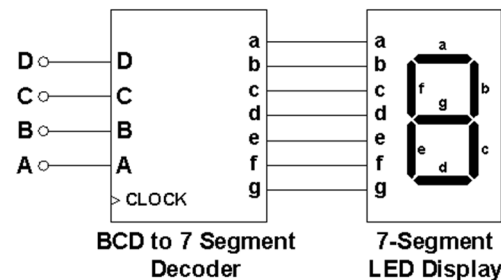


Figura 1. Funcionamiento de un BCD con un 7 segmento. (1)

2. Timer: contador descendente.

Diseñar y construir un programa que controle un circuito que sea un “Timer” de dos dígitos, mostrados en dos display de 7 segmentos. Al iniciar el programa mostrará dos opciones: programar el Timer por medio de hardware (botones) o por medio del programa. En la primera opción (por hardware), el valor del “timer” y el inicio del mismo, debe ser configurado utilizando mínimo con tres botones: uno para configurar el primer dígito, otro para el segundo y el tercero para comenzar el conteo regresivo. En la segunda opción (por programa), se solicitará al usuario que ingrese el valor del Timer por medio del teclado. Al dar enter, la alarma inicia en el valor programado. En ambos casos, los display 7 segmentos irán mostrando el conteo regresivo hasta llegar a 0.

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 display 7 segmentos, 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.

3. Piano

Su objetivo será construir un pequeño piano digital con la *Raspberry Pi*. Para ello deberá conectar 7 botones y un *buzzer* al GPIO. Cada botón representa una nota distinta de la escala musical (do, re, mi, fa, sol, la, si) y cada una de estas notas tiene una frecuencia que la caracteriza. Debe investigar las frecuencias de las notas y cada vez que se presiona el botón correspondiente a la nota, esta debe sonar en el *buzzer*. Además de lo anterior el piano debe tener la capacidad de grabar una melodía y poder reproducirla.

Extra: Implementar botón de pausa para parar la reproducción de la melodía y botón reset para comenzar la melodía desde el principio

4. Controlador de motores Stepper

Diseñar y construir un programa que controle circuito que pueda controlar un motor de paso unipolar. Al iniciar el programa el motor empezará a moverse con la secuencia de paso doble a una velocidad predefinida por los programadores, y mostrará dos opciones: programar el motor por medio de hardware (botones) o por medio del programa. En la primera opción (por hardware), se utilizan tres botones: uno para aumentar la velocidad de rotación del motor, otro para disminuir la velocidad de rotación del motor y el tercero para cambiar la dirección del motor. En la segunda opción (por programa), se solicitará al usuario que cambie la velocidad del motor y la dirección por medio del teclado.

La figura 4 muestra la secuencia que debe enviarse al motor para generar la secuencia de paso doble y la figura 5 muestra la conexión del motor. **Puntos extras:** si se almacenan 5 velocidades diferentes del motor y por medio de botones se activan dichas velocidades.

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

↓ Clockwise Rotation

Alternate Full Step Sequence
(Provides more torque)

Figura 4. Secuencia de paso doble

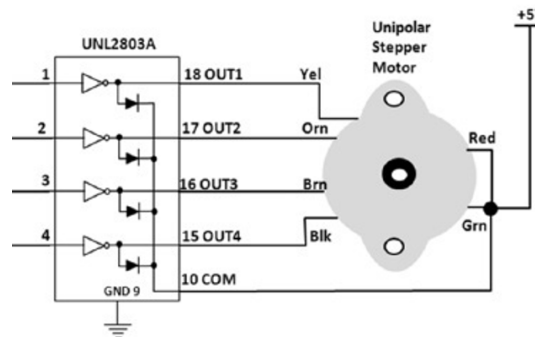


Figura 5. Conexión de un Stepper Unipolar de 4 fases usando un driver UNL2803A.

Se sugiere mostrar las secuencias enviadas (1a, 1b, 1c, 1d) al motor por medio de 4 leds, para apreciar que se están generando de manera correcta.

5. Simón dice

Su objetivo será implementar un juego de “Simón dice”. Para ello deberá utilizar 4 botones, 4 LEDs de colores diferentes correspondientes a los 4 botones y un botón para forzar un reinicio en el juego. El juego empieza encendiendo un LED al azar. Al apagarse el LED el jugador debe presionar el botón correspondiente a ese LED. Con cada turno finalizado el programa genera de forma aleatoria una nueva posición y muestra la secuencia completa (incluyendo la nueva posición). En los turnos del usuario, con cada tecla que presione debe encenderse por un tiempo corto su LED correspondiente para indicarle al usuario que presionó correctamente el botón. Si el usuario ingresa una posición de la secuencia de forma incorrecta, deberá indicárselo encendiendo todos los LEDs. Al finalizar el juego, debe esperar un tiempo y reiniciarse automáticamente. Considere que cada vez que sea turno del usuario puede presionar el botón de reinicio y el juego debe empezar nuevamente. NOTA: el juego tiene turnos indefinidos, termina cuando pierde.



Extra:

- Desplegar la cantidad de movimientos de la secuencia actual en dos *display* de 7 segmentos.
- Implementar diferentes tonos mientras se muestra la secuencia.

6. ALU

Su objetivo será construir una ALU con la *Raspberry Pi*. Para ello deberá conectar 4 botones a los puertos GPIO. Cada switch representará una operación aritmética o lógica: AND, OR, ADD (sin overflow), SUB. El programa funciona en el siguiente orden:

- El usuario debe ingresar por medio del teclado dos números enteros entre 0 y 9.
- Luego de ingresar los operandos, debe presionar el botón correspondiente a la operación aritmética o lógica que desea efectuar (la operación se ejecuta al presionar el botón).
- El resultado en decimal deberá ser desplegado en un display de siete segmentos, y resultado en binario en el monitor.

7. Número y dirección de revoluciones de un stepper

Su objetivo será controlar el número de revoluciones y la dirección de giro de un servomotor, por programa o por botones. Al iniciar el programa el motor girará 3 vueltas a la derecha usando la secuencia de paso completo (Figura A) y mostrará dos opciones: configurar el motor por medio de hardware (botones) o por medio del programa. En la primera opción (por hardware), se utilizan tres botones: uno para aumentar de forma circular entre 1 y 9 el número de revoluciones del motor, otro para modificar la dirección de rotación del motor y el tercero para guardar los cambios e iniciar la rotación. En la segunda opción (por programa), se solicitará al usuario que cambie el número de revoluciones y la dirección por medio del teclado.

Puntos extra: 1) mostrar el número de revoluciones y la dirección en un display de 7 segmentos. 2) agregar repetición de la secuencia de rotación y dirección un número de veces entre 1 y 9, dejando una pausa de 2 segundos entre cada repetición.

Clockwise Rotation ↓

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 A. Secuencia de paso doble

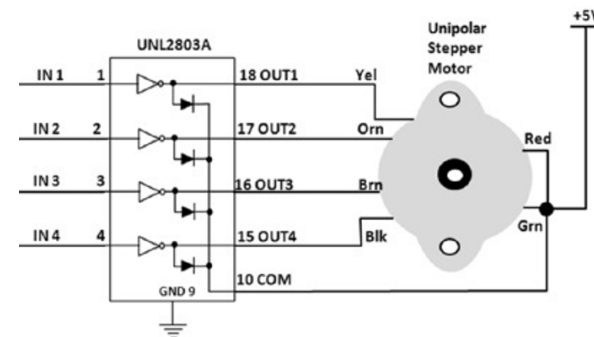


Figura B. Conexión de un Stepper Unipolar de 4 fases usando un driver UNL2803A

Escala valorativa de Evaluación

Cada miembro del grupo presentará una parte del proyecto, teniendo una nota individual, la cual servirá para determinar el **porcentaje de conocimiento que tenga de la TOTALIDAD del mismo**. Este porcentaje se determinará por medio de la realización de dos o tres preguntas acerca del funcionamiento o bases teóricas de cualquier parte del proyecto. Por ejemplo, si la nota obtenida en el proyecto es de 80 puntos y el estudiante posee un 50% de conocimiento del proyecto, la nota será de 40 puntos. Además, tome en cuenta que si un integrante del grupo no llega a clase el día de la presentación, su nota individual del proyecto será cero.

CRITERIOS		DESCRIPCIÓN	PUNTOS
Funcionamiento total del proyecto		La integración del circuito con el programa cumple con todos los requerimientos solicitados en el proyecto. Adjuntaron el video del funcionamiento explicado por los dos integrantes en la actividad de <i>Canvas</i> .	60
Orden y cableado del circuito		El orden del cableado del circuito permite la fácil identificación de las rutas de datos . Todas las conexiones están completamente ordenadas , se encuentran sobre la superficie del <i>protoboard</i> y no utilizan jumpers provisionales , solo los necesarios para conectar los puertos digitales GPIO.	5
Documentación y orden de los programas fuente (.s)		La documentación incluye encabezado y comentarios representativos en los bloques de código más importantes, indicando claramente el objetivo del proyecto y detalles de cómo conectar el circuito . Los nombres de las variables son significativos. La presentación del programa es muy clara y ordenada, y utiliza una tabulación adecuada	10
Uso de subrutinas		Los programas fuente están totalmente estructurados con subrutinas que siguen el estándar ABI para paso de parámetros y uso de registros y están en un archivo independiente al programa principal.	10
Documento de análisis y diseño	Tabla	El documento incluye una tabla que menciona los archivos fuente realizados por el grupo para el proyecto (.s) y sus respectivas subrutinas explicando los parámetros de entrada, la función que realiza y los parámetros de salida de cada una. No es necesario agregar los programas de las bibliotecas de la U. de <i>Cambridge</i> .	3
	Diagrama de flujo	El documento contiene el diagrama de flujo y/o algoritmo narrativo del funcionamiento del proyecto y utiliza la simbología estándar de diagramas y/o algoritmos.	3
	Diagrama del circuito	El documento contiene el diagrama del circuito diseñado para el proyecto, realizado en alguna herramienta de software tipo <i>multiSim</i> o similar, y es presentado con claridad y excelente resolución.	3
	Bibliografía	El documento incluye la bibliografía de al menos dos sitios confiables, con el formato APA o IEEE.	1
Reflexión individual		El estudiante realizó una reflexión completa del proyecto, con aportes significativos para mejorar su aprendizaje en/antes de la fecha solicitada y contestó todas las preguntas planteadas.	5
TOTAL			100

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].