

Laboratorio No. 09

Diseño programas en lenguaje Ensamblador para el control de circuitos.

I. Competencias a desarrollar

Diseñar un programa básico en lenguaje ensamblador del procesador ARM que implemente la secuencia asignada.

II. Instrucciones

Esta actividad se desarrollará en los grupos de trabajo para el Proyecto 03.

Su catedrático asignará a cada grupo, uno de los temarios. y diseñarán y construirán un circuito. Por medio del control de puertos GPIO de la Raspberry PI y de programación, implementarán un circuito físico con la secuencia asignada por su catedrático.

III. Temarios

1. Juego: corrimiento de bits activos

Desarrollar un programa en lenguaje Assembler ARM, generar la secuencia mostrada en la siguiente figura por medio del control de los puertos GPIO. El cambio de estado se realizará cada 1.5 segundos.



Figura 1.

2. Juego: corrimiento de bits inactivos

Desarrollar un programa en lenguaje Assembler ARM, generar la secuencia mostrada en la siguiente figura por medio del control de los puertos GPIO. El cambio de estado se realizará cada 0.8 segundos.



Figura 2.

3. Juego: Acumulador de bits

Desarrollar un programa en lenguaje Assembler ARM, generar la secuencia mostrada en la siguiente figura por medio del control de los puertos GPIO. El cambio de estado se realizará cada 2.0 segundos.



Figura 3.

4. Juego: Gusanito

Desarrollar un programa en lenguaje Assembler ARM, generar la secuencia mostrada en la siguiente figura por medio del control de los puertos GPIO. El cambio de estado se realizará cada 1.0 segundos.



Figura 4.

5. Juego: acumulador de segmentos

Desarrollar un programa en lenguaje Assembler ARM, generar la secuencia mostrada en la siguiente figura por medio del control de los puertos GPIO. El cambio de estado se realizará cada 1.0 segundos.



Figura 5.

6. Simon says - Letras

Desarrollar un programa en lenguaje Assembler ARM, generar la secuencia mostrada en la siguiente figura por medio del control de los puertos GPIO. El cambio de estado se realizará cada 1.0 segundos.

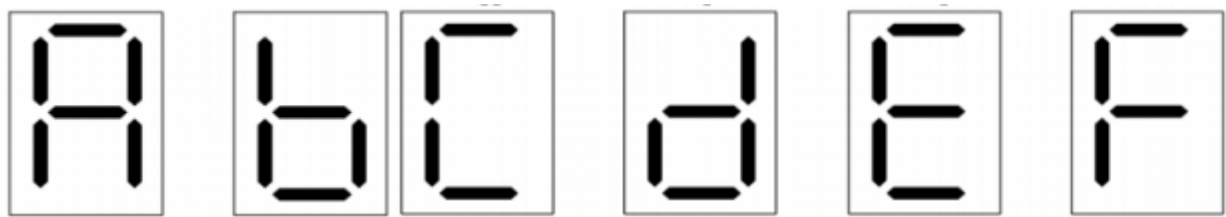


Figura 6.

7. Simon says - números

Desarrollar un programa en lenguaje Assembler ARM, generar la secuencia mostrada en la siguiente figura por medio del control de los puertos GPIO. El cambio de estado se realizará cada 1.0 segundos.

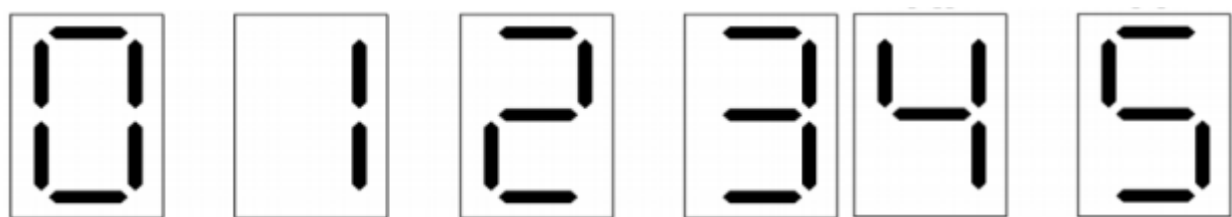


Figura 7.

8. Simon says - vocales

Desarrollar un programa en lenguaje Assembler ARM, generar la secuencia mostrada en la siguiente figura por medio del control de los puertos GPIO. El cambio de estado se realizará cada 1.0 segundos.

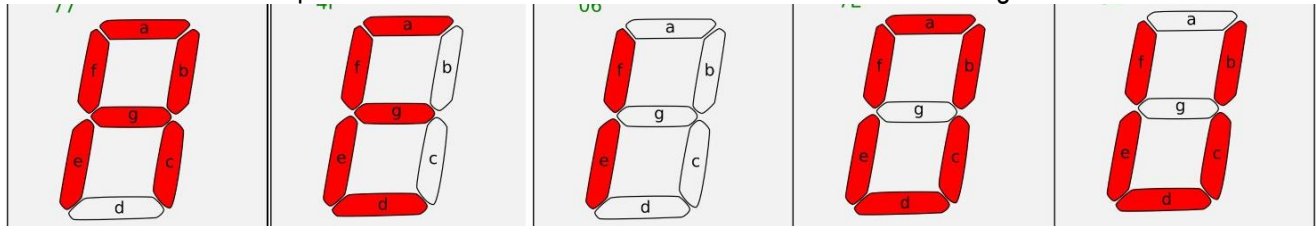


Figura 8.

9. Simon says - colores

Desarrollar un programa en lenguaje Assembler ARM, generar la secuencia mostrada en la siguiente figura por medio del control de los puertos GPIO. El cambio de estado se realizará cada 1.0 segundos.

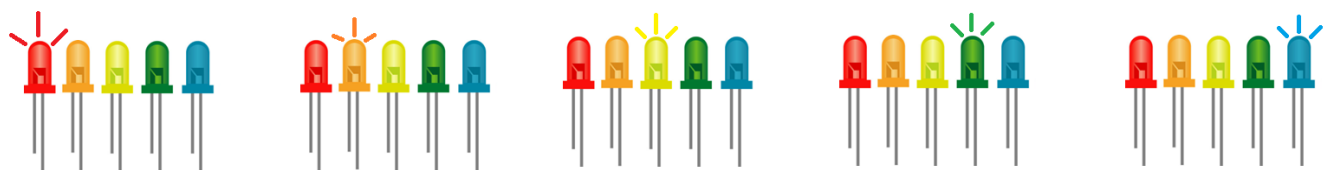


Figura 9.

La secuencia se detendrá al presionar la letra “q”.

10, 11 y 12 Implementación de una línea de producción de harina de trigo (3 grupos de 2 personas c/uno).

Diseñar y programar el funcionamiento de los puertos GPIO de la Raspberry en lenguaje ensamblador ARM, para el control de una cadena de tres máquinas industriales

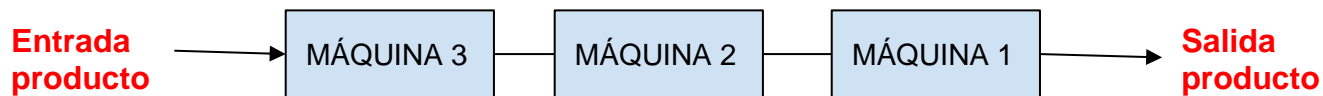


Figura 10.

Funcionamiento: tomar en cuenta que la secuencia de arranque en automático es la siguiente:

Al presionar la tecla “a”, se produce el arranque de las máquinas (secuencia de activación para todas las máquinas): primero se enciende la máquina 1, luego la máquina 2 y por último la máquina 3.

Al presionar la tecla “p”, se detienen de las máquinas (secuencia de desactivación para todas las máquinas), primero se apaga la máquina 3, luego la máquina 2 y por último la máquina 1.

Puede utilizar 3 leds (1 led para representar cada máquina). 1 led activado representa una máquina encendida.

13 Carrito inteligente seguidor de línea.

Diseñar y programar el funcionamiento de los puertos GPIO de la Raspberry en lenguaje ensamblador ARM, para el control de un carro dirigido mediante un seguidor de línea negra.

Al presionar la tecla “l”, se produce el movimiento de un motor (que produce el giro hacia la izquierda)

Al presionar la tecla “r”, se produce el movimiento de un motor (que produce el giro hacia la derecha)

Al presionar la tecla “s”, se detienen los motores.

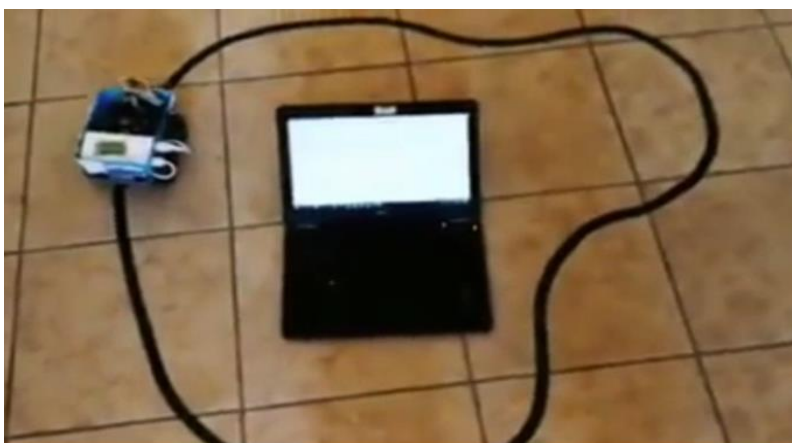


Figura 11.

14 Caja fuerte con código de activación.

Deberá ingresarse por medio de teclado un número de 4 bits.

El número ingresado en el DipSwitch deberá desplegarse en 4 displays de 7 segmentos conectados a los puertos GPIO



Figura 12.

15 Contador de personas

Se desea contar la cantidad de personas que ingresan a un restaurante.
Cada vez que se presiona la tecla “↑” se incrementa un contador de 0 a 9. El conteo debe ser cíclico.
El resultado debe mostrarse en un display de 7 segmentos.



IV. Evaluación

Todos los integrantes del grupo deben presentar el circuito y programa funcional.

Criterios de evaluación	Nivel 3 Experto	Nivel 2 Aprendiz	Nivel 1 Novato
Funcionamiento del Programa 60%	El programa funciona con todos sus requerimientos: ingreso de datos, despliegue de resultados correctos y salida correcta al sistema operativo 60%	El programa funciona en al menos el 60% de sus requerimientos de funcionamiento. 35%	El programa funciona en menos del 50% de sus requerimientos. 20% o menos
Programación defensiva 10%	Los programas tienen muy buena programación defensiva, proporcionando mensajes oportunos ante situaciones inesperadas. 10%	Los programas tienen programación y proporciona algunos mensajes oportunos ante situaciones inesperadas. 5%	Los programas tienen muy poca o ninguna programación defensiva, y proporciona pocos o ningún mensaje oportuno ante situaciones inesperadas. 0%
Circuito 20%	El circuito es funcional y ordenado 20%		
Documentación 10%	La documentación incluye encabezado y comentarios representativos en los bloques de código más importantes. Los nombres de las variables son significativos. 10%	Falta documentación en el encabezado o en bloques de código Los nombres de las variables son medianamente significativos. 5%	Falta gran parte de la documentación del código Los nombres de las variables no expresan ningún significado. 0%