



# Tecnológico de Monterrey

## Escuela de Ingeniería y Ciencias

Carrera:	Ingeniería en sistemas digitales y robótica	Materia:	Sistemas embebidos
Nombre del alumno:	Carlos Jesús Salguero Rosales Jorge Martínez Hernández Sherlyn Quetzal López Hernández Ricardo García Sedano	Semestre:	Octavo semestre
Nombre del docente:	Héctor Eduardo de Cos Cholula		
Práctica No.	5	Nombre de la práctica:	Máquina de estados Moore (Semáforos Crucero)

**Tema:** Máquina de estados Moore

### Introducción:

Dentro de la teoría de la computación, una máquina de estados es una modelización conceptual de un problema. No obstante, muchas veces suelen confundirlo como una máquina física. Dentro de este problema, se encuentra nuestra variable(s) que se mantiene en el punto inicial. No obstante, esta variable(s) se mantendrá ahí hasta que reciba un estímulo inicial del mundo exterior. Todas sus funciones pueden ser exactamente listadas, en el cual contiene todos los estados y sus condiciones para que estos se desarrollen. Con el desarrollo de los sistemas embebidos, las máquinas de estado finito se han convertido en una poderosa herramienta para el desarrollo de controladores digitales.

En este contexto, existen diferentes tipos de máquinas de estado. Una de ellas fue nombrada por el profesor estadounidense de matemáticas y ciencias de la computación Edward F. Moore. Él fue el primero en utilizar una FSM (Finite State Machine/Máquina de estados finito) el cual llamó Machine Moore / Máquina de estados de Moore. Moore fue un pionero de la vida artificial y de la teoría de la computabilidad. La mayor parte de los sistemas digitales se encuentran diseñados como sistemas de pulso de reloj. Es decir, son una forma reducida de las máquinas de Moore, pero... ¿qué son?



Ilustración 1.- Ejemplo básico de una máquina de estados finito



# Tecnológico de Monterrey

## Escuela de Ingeniería y Ciencias

Una máquina de estados finito de Moore consiste en que las salidas se encuentran determinadas por la función de los estados actuales (A diferencia de las máquinas de estado de Mealy). Dentro de su diagrama, esta máquina proporciona una señal de salida por cada estado que contiene. Una máquina de estados finito Moore contiene: un conjunto de estados finitos, un conjunto finito de entradas, un conjunto finito de salidas, una función de transición (este define a cada estado y a una entrada al siguiente estado) y una función de salida (este maneja a cada estado a su salida).

s0: Estado inicial (pelota parada)

i1: Entrada 2 (patada)

s1: Estado 2 (pelota aplastada)

o0: Salida del estado 2

s2: Estado 3 (pelota en movimiento)

o1: Salida del estado 3

i0: Entrada 1 (puñetazo)

Máquina de Moore:

Estado\entrada	i0 = 0	i1 = 1
s0	s1/x	s2/x

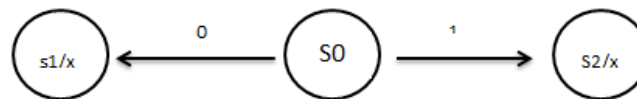


Ilustración 2.- Ejemplo de la máquina de estados moore que simula el movimiento de una pelota que recibe o no un estímulo.

Para el desarrollo de esta práctica, usaremos el software LabView 2015. Este entorno de programación se utilizará para simular cada uno de los estados de dos semáforos. Dado que un solo estado contiene un estado verde, un estado amarillo, un estado de apagado y un estado rojo, se realizará una máquina de estados que simule dos semáforos que trabajen a la vez. Además, con la tarjeta myRio 1900, se realizará un circuito que simulé ambos semáforos con 6 led (con los colores verde, amarillo y rojo).

Como se mencionó anteriormente, una tarjeta myRio 1900 es un dispositivo que es un sistema embebido. Este realizará la simulación física de la práctica a realizar. Esta tarjeta ofrece una solución embebida por conexión física y por wi-fi. Así mismo incluye entradas analógicas, salidas analógicas, líneas de E/S digitales, LEDs, un botón, un acelerómetro interno, un FPGA Xilinx y un procesador dual-core ARM Cortex-A9. En algunos casos también incluye un soporte para Wi-fi, cuya función se usará en prácticas futuras.

Las máquinas de estados, en realidad, se encuentran en cualquier parte de este mundo, pero su complejo significado no permite que las personas lo vean así. Un ejemplo básico de esto es un semáforo. Dentro de un semáforo, se encuentran 5 estados fundamentales, cuyo estímulo es el tiempo. Es decir, cada determinado tiempo, cada uno de los estados de esta máquina de estados ira cambiando, dependiendo de X tiempo. Por lo tanto, para el desarrollo de



# Tecnológico de Monterrey

## Escuela de Ingeniería y Ciencias

esta práctica se realizarán dos semáforos. Esto con la finalidad de manejar máquina de estados dentro de un entorno embebido.

### Objetivo de la práctica:

Para el desarrollo de la práctica, se realizará la implementación de dos semáforos con una máquina de estados de Moore. Esto por medio del entorno gráfico Labview que simula las condiciones de cada uno de los estados de cada semáforo. Además, con la tarjeta myRIO 1900, se realizará la implementación física de los dos semáforos mediante seis LED y seis resistencias de 1k. No obstante, el objetivo principal es que el alumno comprenda conceptos de una máquina de estados de Moore aplicados al software LabVIEW 2015 y el embebido myRIO 1900.

### Equipo necesario y material de apoyo:

El equipo necesario para el desarrollo de esta práctica es la siguiente:

- Computadora con el simulador LabVIEW 2015.

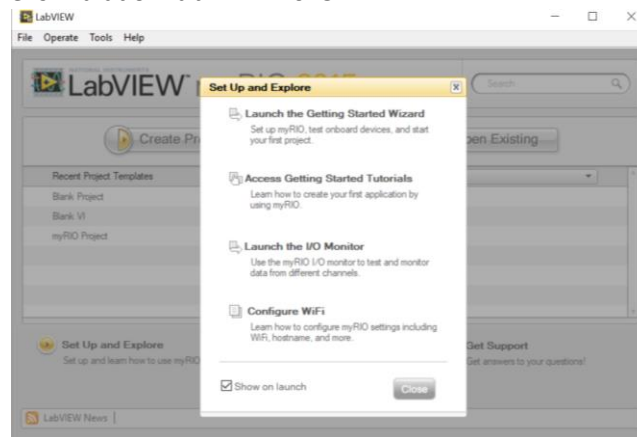


Ilustración 3.- laptop con el software LabView 2015

- Tarjeta myRIO 1900

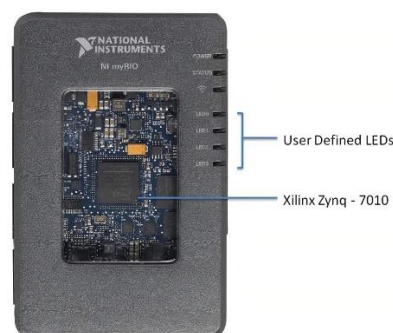


Ilustración 4.- Tarjeta myRio 1900



# Tecnológico de Monterrey

## Escuela de Ingeniería y Ciencias

- LED



Ilustración 5.- Focos LED

- 6 resistencia 1k



Ilustración 6.- Ilustración gráfica de una resistencia de 1k

### Procedimiento:

1. Empezamos abriendo un proyecto nuevo en LabVIEW 2015. En el Front Panel, damos clic derecho, y en la ventana "Controls" seleccionamos la opción "String & Path". Ahí, nos aparecerán varias opciones. No obstante, se seleccionará la opción "String indicator".

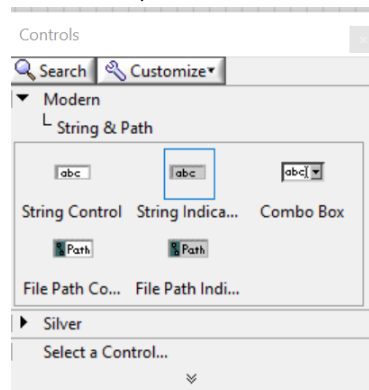


Ilustración 7.- Ventana en LabVIEW 2015 que muestra que opción se debe elegir

2. En el elemento seleccionado (enumerador), se le dará clic derecho. En la ventana que se abra, se seleccionará la opción de "Advanced", y ahí se seleccionará la opción "Customize...". Cuando se le dé clic, saldrá una nueva ventana con el puro enumerador. En



# Tecnológico de Monterrey

## Escuela de Ingeniería y Ciencias

esa ventana se editará este elemento. Arriba de la ventana, a un lado de la imagen de una llave, se le dará clic a esa opción. De esa manera, se podrá elegir entre una de las 3 opciones que aparecen. La que importa es la que se llama "Strict Type Def." Con esto, se le indica al programa que se está seleccionando un tipo muy definido de valores.

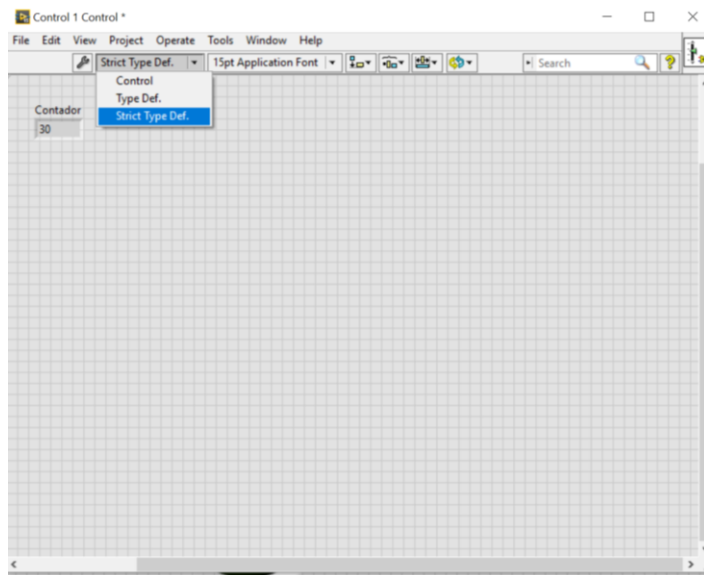


Ilustración 8.- Imagen que muestra la ventana que aparece al darle clic derecho al elemento del enumerador

3. En el cuadro gris del enumerador, se le dará clic derecho, y se seleccionará la opción de "Edit items". Se abrirá una ventana llamada "Enum Properties: Enum". Aquí se podrá agregar los 5 estados necesarios para cada uno de los dos semáforos.

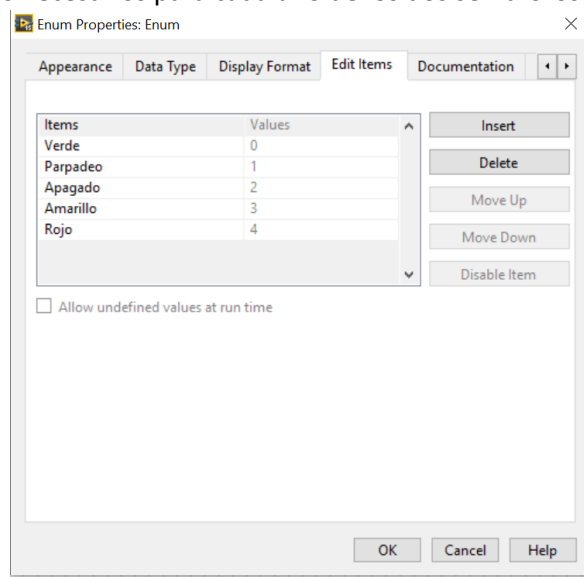


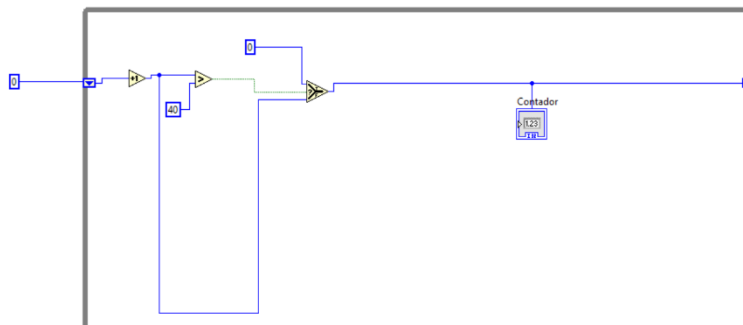
Ilustración 9.- Ventana que muestra todos los estados en el elemento "String indicator". Aquí se podrá eliminar o insertar nuevos elementos.



# Tecnológico de Monterrey

## Escuela de Ingeniería y Ciencias

4. En el block diagram, se selecciona el elemento que incluye a los estados. Con clic derecho, se selecciona la opción de "Change to indicator" para cambiar el tipo de elemento. En el block diagram, se da clic derecho, y se selecciona la opción "Structures". Ahí, se selecciona la opción "While Loop". Se mete el indicador dentro del while. En la orilla del while, se da clic derecho, y se seleccionará la opción "Add Shift Registers". Esto se hará para crear el contador que definirá los estados.
5. Afuera del while, se agregará una constante, y se conectará a la entrada del Shift Register. Dentro del while, se le dará clic derecho para abrir la ventana "Functions". Se seleccionará la opción "Numeric", y dentro del while se agregará el elemento "Increment". También se agregarán los elementos "Select" y "Greater?" que se encontrarán en la opción "Comparison". Se conectará una constante, y se le agregará el número 40. El número 40 indica la duración de todo un semáforo. Este se conectará al elemento "Greater?". Por el otro lado, al select se le agregará otra constante conectada a la entrada "t". Finalmente, la salida del "Select" se conectará a la salida del "Shift register".
6. Para agregar el contador dentro del loop, se seleccionará el cable con clic derecho. Ahí, seleccionará la opción "Create". Ahí se seleccionará el elemento "Indicator". Una vez que aparece este elemento, se le cambia el nombre a "Contador". Ya se encuentra listo la señal de control.



*Ilustración 10.- Gráfico que muestra la conexión final del shift register y los elementos "Select", "Greater?" y el contador.*

7. Se dará clic derecho en el block diagram, y en la ventana "Functions", se seleccionará la opción de "Structures". Aquí se seleccionará el elemento "Case Structure". En la orilla del while, se le dará clic derecho, y se agregará otro "Shift register". Se selecciona el elemento "Estados", y ahí se le dará clic derecho. Se selecciona la opción "Create", y se seleccionará el elemento "Constant". Este elemento se pondrá afuera, y se conectará al shift register. De este shift register, se conectará al "Case Structure". De esta conexión, se conectará al elemento estados (String indicator). Se seleccionará con clic derecho el case structure. Aquí se seleccionará la opción "Add Case For Every Value". Con esto, cada uno de los estados aparecerán en el case structure.
8. Dentro del case structure, se agregará los elementos equal y select. Al equal, se conectará otro constant salido del contador y un constant. Este constant determinará la duración de cada caso. Por ejemplo, el segundo semáforo tendrá un constant de valor 29. Esto indica



# Tecnológico de Monterrey

## Escuela de Ingeniería y Ciencias

- que hasta que el contador dé 29, el estado de ese semáforo será verde. Después cambiará a apagado.
9. En el elemento "select", a la entrada "t" se conectará una constant salido de los estados anteriormente creado. En la entrada s estará conectado el contador anteriormente mencionado. Finalmente, en la entrada "f" se conectará otro constant que salió del estado. La salida del elemento "Select" se conectará al shift register que sale en el case structure.
  10. Por último, se agregará los 6 Round LED en el Front Panel que simularán los LED de los semáforo en el simulador. En el block diagram, se crearán una constante True-False para cada LED. Las constante se meterán adentro del Case Structure, y desde ahí se conectarán a los LED. De esa manera, le indicará dependiendo del estado, si uno de los 3 led (de cada semáforo) se enciende o se apaga.
  11. Se conectará el pin 31, 29, 27, 25, 23, 21 con uno de los seis cable macho-hembra al ánodo del LED y la resistencia se conectará al cátodo del LED. Con el programa funcionando, podrá encender el LED o apagarlo dependiendo del estado en que se encuentre en LabView.

### Pruebas:

En las siguiente imágenes se mostrará cómo se desarrollaron la máquina de Moore en cada uno de los semáforos. Finalmente, en la última imagen se mostrará el funcionamiento de la práctica.

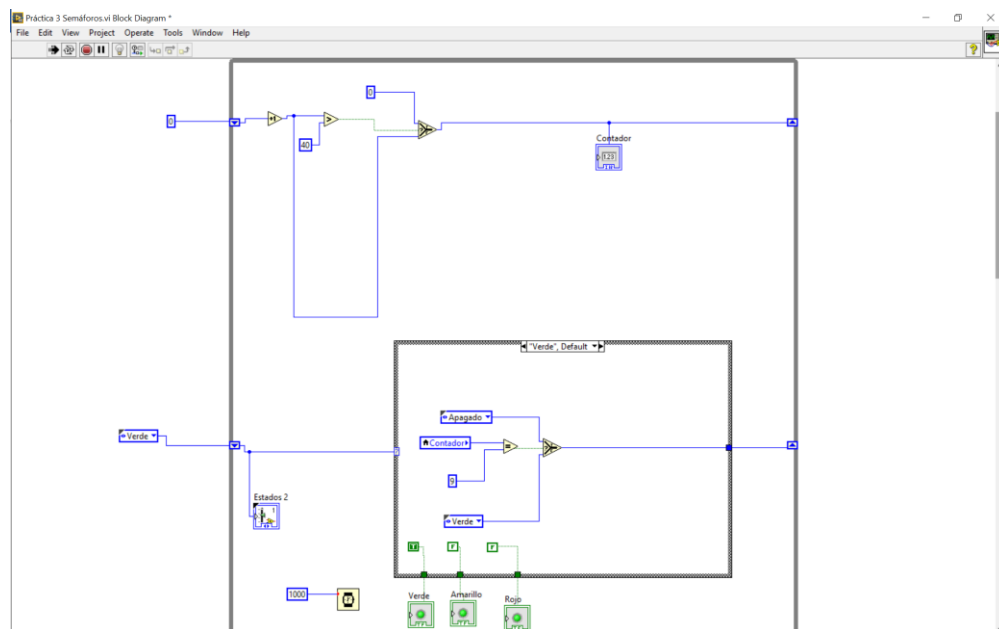


Ilustración 11.- Ilustración que muestra el diagrama del primer semáforo



# Tecnológico de Monterrey

## Escuela de Ingeniería y Ciencias

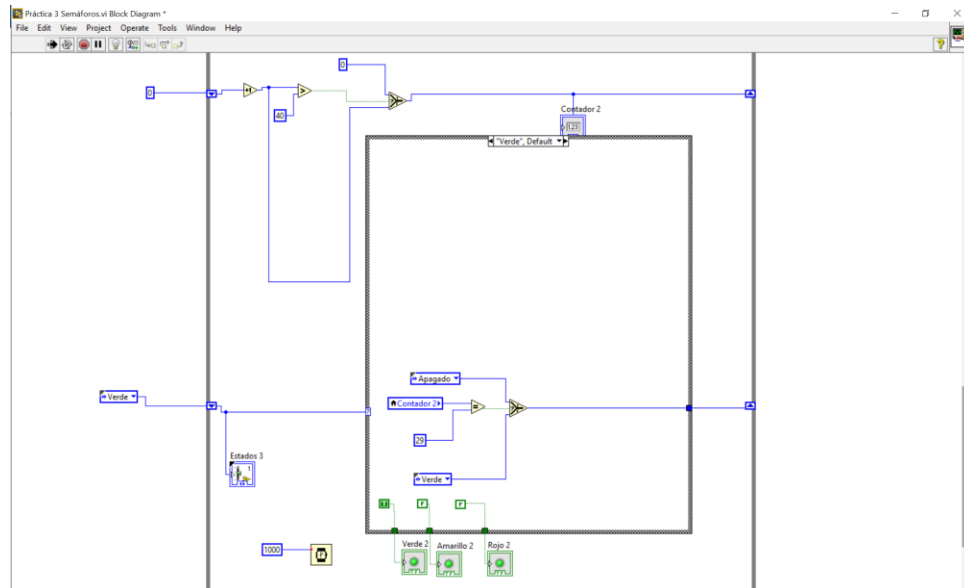


Ilustración 12.- Ilustración que muestra el diagrama del segundo semáforo

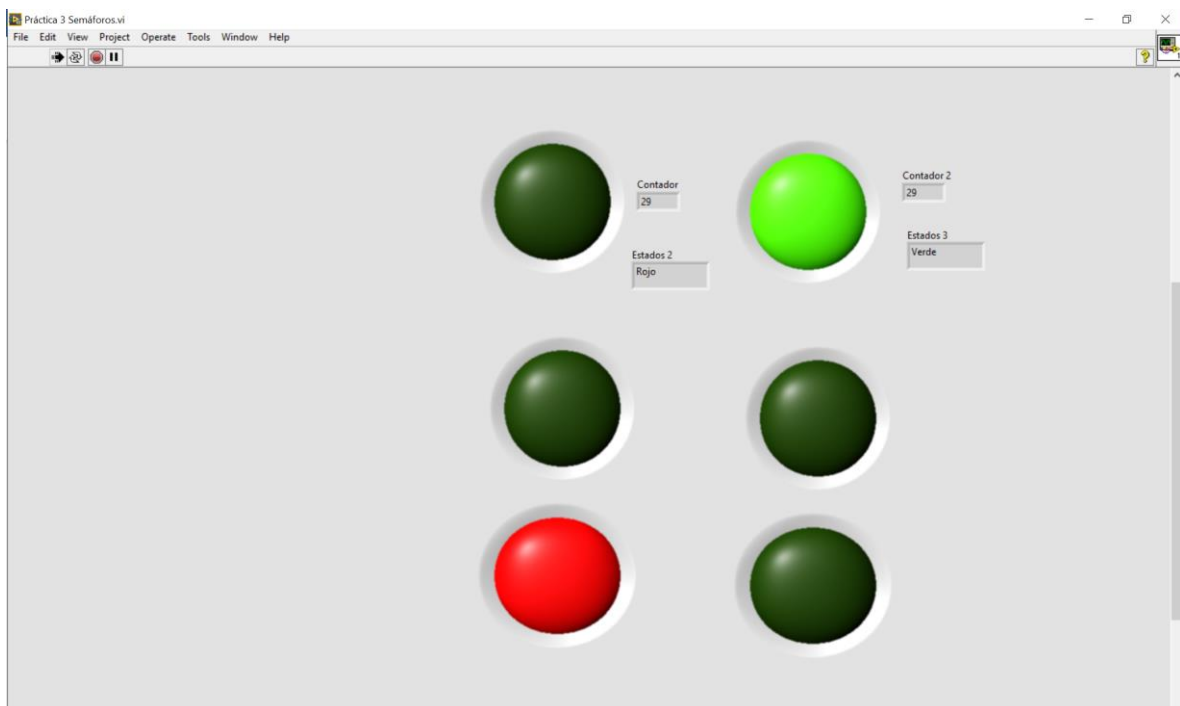


Ilustración 13.- Esta ilustración muestra el correcto funcionamiento de la práctica en labVIEW 2015

**Resultados:**

Elaboró: Los borreguitos

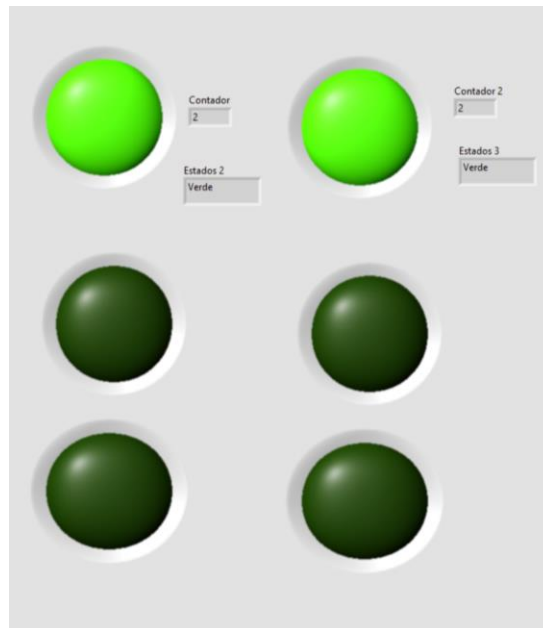




# Tecnológico de Monterrey

## Escuela de Ingeniería y Ciencias

A continuación, se muestran las condiciones iniciales de esta práctica en la siguiente foto. Como se puede ver, se inicia ambos contadores en 1. Además, ambos semáforos inician en verde. No obstante, cuando llegue a 10 el primer contador, este cambiará a apagado en el primer semáforo. Cuando llegue a 16 el contador, el primer semáforo cambiará al estado “Amarillo”. Finalmente, cuando el contador llegue a 21, el estado del primer semáforo será rojo. Sin embargo, el segundo semáforo se mantendrá en verde... hasta que llegué a 30. Aquí, el segundo cambiará a estado “Apagado”.



*Ilustración 14.- Aquí inicia la simulación de ambas máquinas. Se puede ver que ambos inician en verde en el 1.*



# Tecnológico de Monterrey

## Escuela de Ingeniería y Ciencias

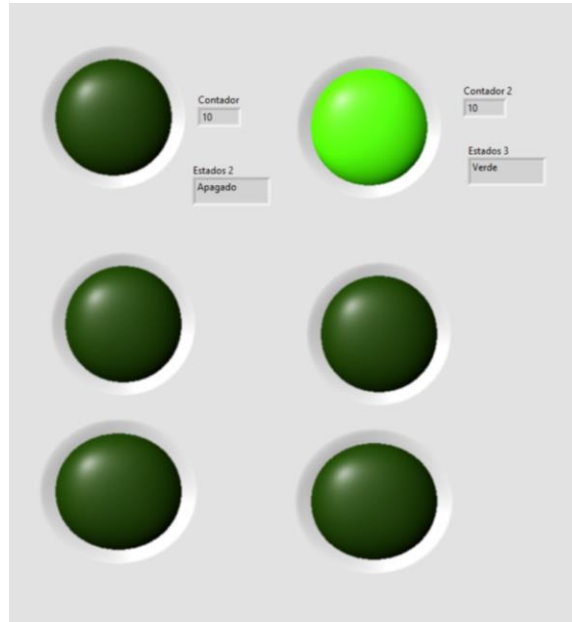


Ilustración 15.- Cuando el contador llegue a 10, el primero semáforo cambiará a estado apagado. El segundo se mantendrá en verde.

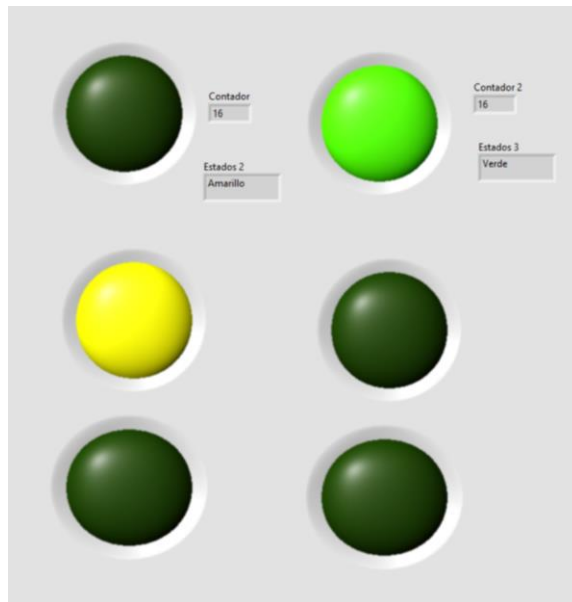
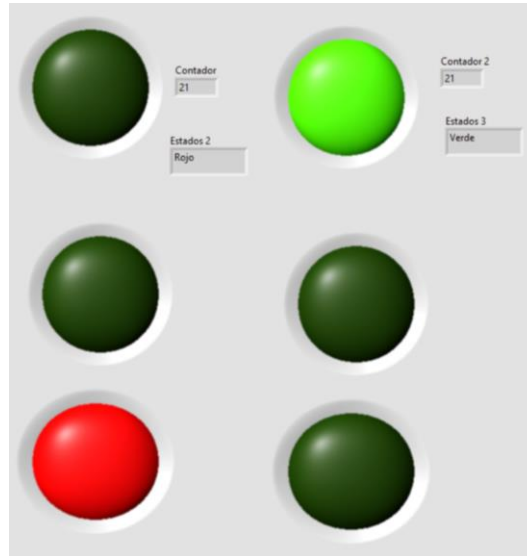


Ilustración 16.- Cuando el contador llegue a 16, el primer semáforo cambiará su estado a "Amarillo". El primer se mantendrá en verde.

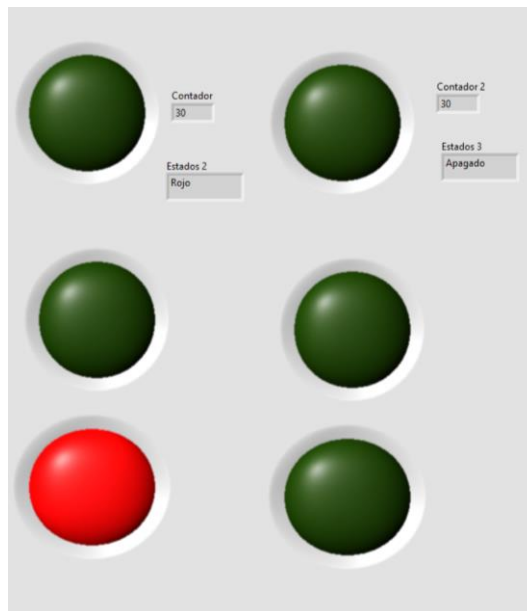


# Tecnológico de Monterrey

## Escuela de Ingeniería y Ciencias



*Ilustración 17.- Cuando el contador llegué a 21, el primer semáforo cambiará su estado a rojo. El segundo semáforo se mantendrá igual.*



*Ilustración 18. Cuando el contador llegue a 30, el primer semáforo se mantendrá en rojo. No obstante, el segundo semáforo cambiará a "Apagado".*



# Tecnológico de Monterrey

## Escuela de Ingeniería y Ciencias

### Semáforo

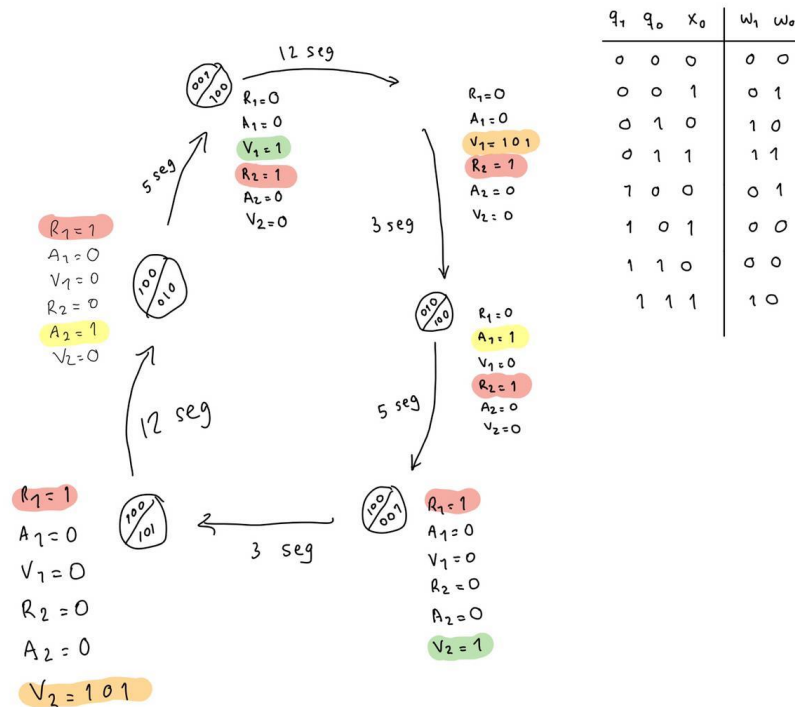


Ilustración 19.- Máquina de estados que muestra el funcionamiento de ambos semáforos

Como se puede ver, la práctica fue un éxito. Para una mejor visualización de la práctica, se recomienda verificar el siguiente video: [https://drive.google.com/drive/folders/1i6geQ9avNm-GoTrvRI\\_3iP6TcdTr9VcB?usp=sharing](https://drive.google.com/drive/folders/1i6geQ9avNm-GoTrvRI_3iP6TcdTr9VcB?usp=sharing)

### Conclusiones y comentarios:

Como se mencionó anteriormente, el alumno pudo comprender el uso de las máquinas de estado de Moore en el entorno gráfico de LabVIEW y en simulado físicamente en la tarjeta myRIO 1900. Gracias a esta práctica, los miembros del equipo pudieron familiarizarse con el desarrollo de este tipo de máquina de estado junto con LabVIEW. La práctica resultó un poco compleja, a diferencia de las prácticas anteriores.

En conclusión, la evidencia mostrada anteriormente muestra que la práctica se desarrolló con éxito. Con la imagen del diagrama de la máquina de Moore, se puede verificar que la práctica se resolvió correctamente. La parte física no presentó grandes complicaciones. El equipo seguirá practicando en el uso del software y tarjeta para poder desarrollar problemas más complejos.

### Referencias:



# Tecnológico de Monterrey

## Escuela de Ingeniería y Ciencias

Ballesteros, D. (2008). Modulación pwm en fpga basado en máquinas de estado finito. *Scientia et technica*, 14(38), 421-426. <https://www.redalyc.org/pdf/849/84903874.pdf>

Chitiva Estupiñan, D., & Duarte Amezquita, J. Material de Apoyo Basado en la Equivalencia de Circuitos Digitales, Lógica de Contactores y Máquinas de Estado Finito. <https://repository.udistrital.edu.co/handle/11349/5989>

Fernández, G., & Torres, L. (1994). Autómatas celulares estocásticos. *Lecturas matemáticas*, 15(2), 167-191. <https://dialnet.unirioja.es/descarga/articulo/7172213.pdf>