Cursos Abiertos de Programación de Sistemas Embebidos

Nivel 1 - Introducción a la Programación de Microcontroladores

Modelado y Máquina de Estados Finitos

Profesores: Ing. Eric Pernia, Dr. Ing. Pablo Gomez

ACSE

Asociación Civil para la Investigación, Promoción y Desarrollo de los Sistemas Electrónicos Embebidos

CIAA

Temario

- Modelado en sistemas embebidos.
- Máquina de Estados Finitos.
- Implementación de MEF en C.

Modelado en sistemas embebidos

¿Qué es un Modelo?

- Un **modelo** es una **representación simplificada de un sistema** que contempla las propiedades importantes del mismo desde un determinado punto de vista.
- El uso de modelos es una actividad arraigada en técnicos e ingenieros y probablemente tan antigua como la propia ingeniería.
- Los modelos de mayor utilidad se caracterizan por ser:
  - **Abstractos.** Enfatizan los aspectos importantes del sistema y eliminan los aspectos irrelevantes.
  - **Comprensibles.** Expresados en forma fácilmente perceptible por los observadores.
  - **Precisos.** Representan fielmente el sistema modelado.
  - **Predictivos.** Pueden utilizarse para responder cuestiones sobre el sistema Modelado.
  - **Económicos.** Mucho más baratos de construir y estudiar que el propio Sistema.

Modelado en sistemas embebidos

¿Qué es un Modelo?

- Los modelos empleados para crear software para sistemas embebidos, además de servir para **lograr un conocimiento más profundo del problema**, también se utilizan como elementos de entrada para la **generación de código**.

- La mayoría de los enfoques actuales en el desarrollo de software basado en modelos coinciden en:

  - Utilizar una **representación gráfica del sistema a desarrollar**.
  - Describir el sistema con un cierto grado de **abstracción.**
  - Generar **código ejecutable para el sistema embebido partiendo del propio modelo**.

Máquina de Estados Finitos (MEF)

- Un **modelo** altamente utilizado es el de **máquina de estados finitos.**
- Constituye una herramienta gráfica utilizada tradicionalmente para modelar el comportamiento de sistemas electrónicos e informáticos.
- Una MEF (o FSM en inglés) es un **modelo computacional**, basado en la **teoría de autómatas**, que se utiliza para describir sistemas cuyo comportamiento depende de los eventos actuales y de los eventos que ocurrieron en el pasado.
- En cada instante de tiempo la máquina se encuentra en un estado, y dependiendo de las entradas, actuales y pasadas, que provienen del ambiente, la máquina cambia o no cambia de estado, pudiendo realizar acciones que a su vez influyen en el ambiente.

Máquina de Estados Finitos (MEF)

- Existen muchos **sistemas** que pueden describirse como un conjunto de estados finitos.
- Un **programa** puede estructurarse de acuerdo a los estados propuestos.
- Son propensos a caber en esta metodología los programas que interactúan con el usuario (por ejemplo, teclado, display, data entry, alarmas, configuración, etc).
- Un programa que posee una “maquina de estados finitos” es **fácil de mantener** ya que se pueden agregar o quitar estados sin modificar el resto.

Máquina de Estados Finitos (MEF)

- El modelado puede hacerse utilizando "diagrama de estados" o "tabla de estados".
- La máquina de estados va cambiando de estados según indiquen los "flags" de estado.
- Existen reglas bien definidas para cambiar de estado.
- Cada transición implica diferentes respuestas del sistema.
- Existen dos tipos de implementaciones de MEF:
  - **Moore.**
  - **Mealy.**

  } Difieren en la forma en que se produce la salida.

Máquina de Estados Finitos (MEF)

Moore

- La salida del sistema depende del estado actual.
- El siguiente estado depende de la entrada y del estado actual.
- Puede haber múltiples estados con la misma salida, pero cada uno de ellos es un estado diferente.
  - Por ejemplo, un controlador de Semáforos.

Máquina de Estados Finitos (MEF)

Mealy

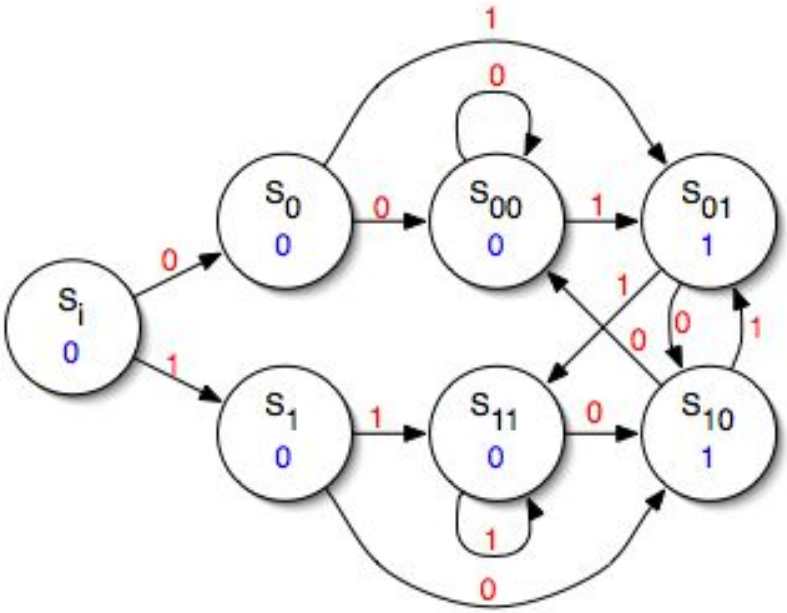
- La salida depende del estado actual y la entrada.
- El siguiente estado depende de la entrada y del estado actual.
- Las diferentes salidas son necesarias para el cambio de estado.
  - Por ejemplo, el movimiento de un robot donde las salidas producen un cambio en el estado (en movimiento-detenido-parado-sentado).

Máquina de Estados Finitos (MEF)

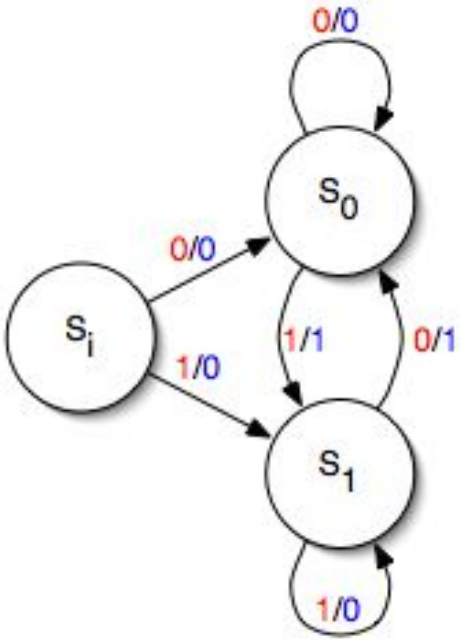
Ejemplo:

- Detector de flanco: la salida es la XOR de las dos entradas más recientes.

Moore:



Mealy:



Estados
Entradas
Salidas

Máquina de Estados Finitos (MEF)

En general en Sistemas Embebidos la transición entre estados se puede dar por:

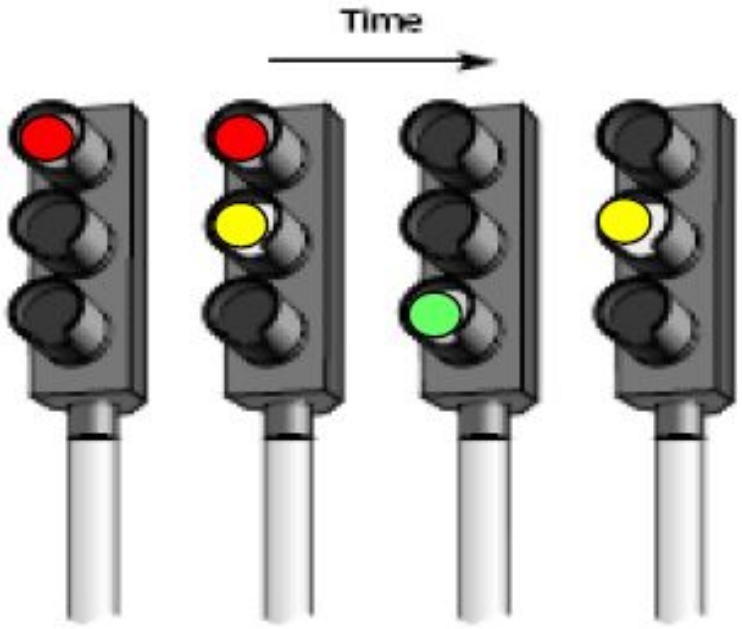
- **Tiempo:** depende solamente del paso del tiempo.
  - Ej: en el estado A la función a se ejecuta por x seg, luego se pasa al estado B en donde la función b se ejecuta por y seg y así sucesivamente.
- **Tiempo-entradas:** la transición entre estados depende del tiempo y de las entradas.
  - Ej: el sistema se mueve del estado A al B si una entrada particular se recibe dentro de un intervalo de tiempo.
- **Entradas-Salidas:** la transición entre estados depende de las entradas (sensores) y las salidas (actuadores).

Máquina de Estados Finitos (MEF)

Ejemplo 1: Semáforo

Estados:

- 1-rojo
- 2-rojo-amarillo
- 3-verde
- 4-amarillo



Máquina de Estados Finitos (MEF)

¿Qué tipo de MEF utilizaría para describirlo, Moore o Mealy?

Máquina de Estados Finitos (MEF)

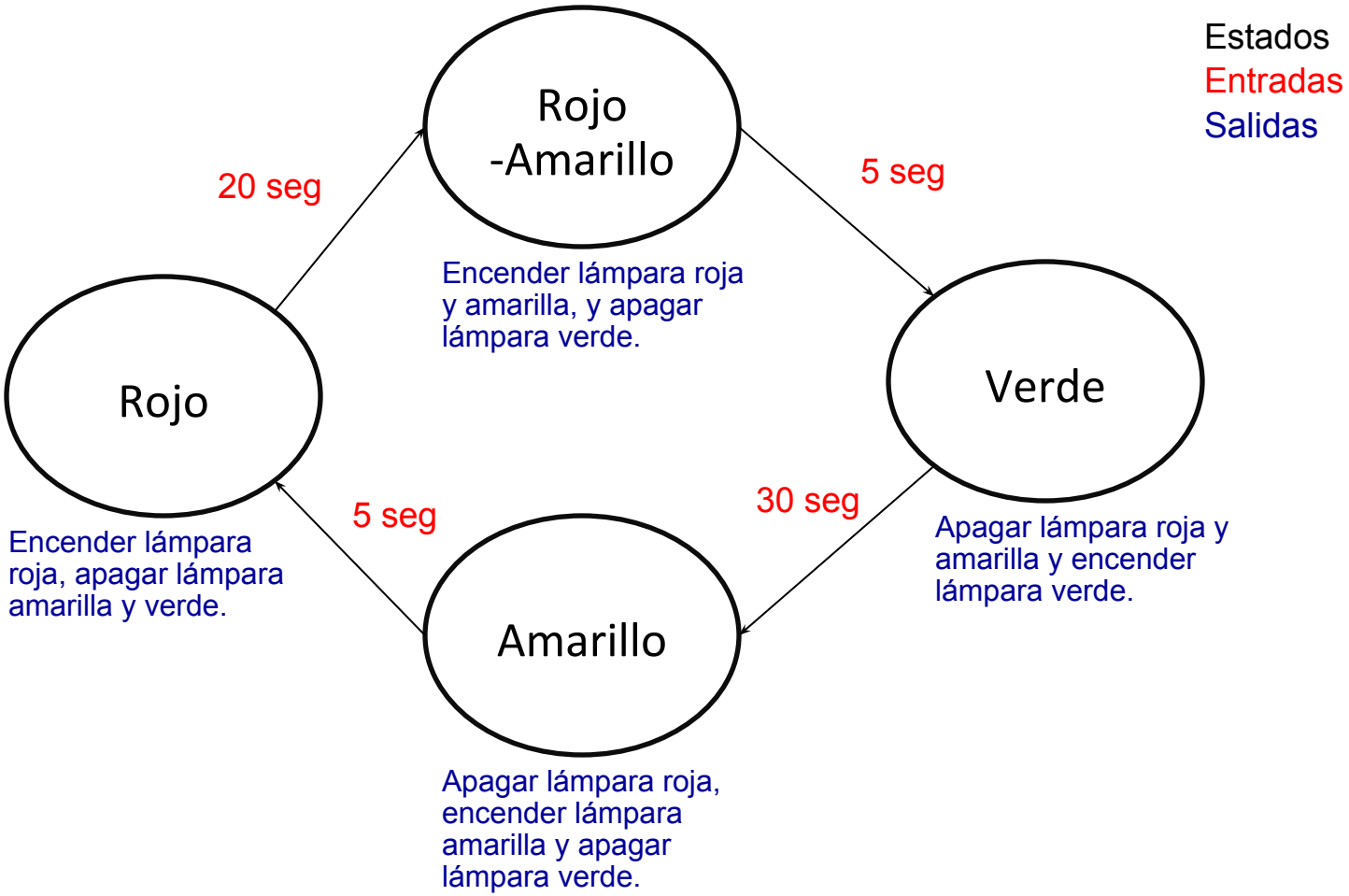
Podemos utilizar ambos modelos, en particular el de Moore, veamos el por qué:

- ¿Cuál es la entrada?
  - Tiempo transcurrido.
- Analicemos las posibles salidas del sistema:
  - Si estoy en estado Rojo. Encender lámpara roja, apagar lámpara amarilla y verde.
  - Si estoy en estado Rojo-Amarillo. Encender lámpara roja y amarilla, y apagar lámpara verde.
  - Si estoy en estado Amarillo. Apagar lámpara roja y amarilla y encender lámpara verde.
  - Si estoy en estado Verde. Apagar lámpara roja, encender lámpara amarilla y apagar lámpara verde.
- Vemos que la salida depende únicamente del **estado actual**. *Condición de Moore.*
- Los cambios de estado dependen del **estado actual** y del **tiempo transcurrido (entrada).** *Condición de Moore o Mealy.*

  ➔ *Cumple ambas condiciones de Moore.*

Máquina de Estados Finitos (MEF)

Diagrama de estados del ejemplo de semáforo:



Implementaciones de MEF en C

- Implementación con switch-case o múltiples if:

```
// Nuevo tipo de datos enumerado llamado estadoMEF
typedef enum{ESTADO_INICIAL, ESTADO_1, ESTADO_2, ESTADO_N} estadoMEF;

// Variable de estado (global)
estadoMEF  estadoActual;

// Prototipos de funciones
void InicializarMEF(void );
void ActualizarMEF(void);
```

Implementaciones de MEF en C

```
// Programa principal
int main (void){
        ...
        InicializarMEF();
        ...
        while(1){
                ...
                ActualizarMEF();
                ...
        }
        return 0;
}

// Función Inicializar MEF
void InicializarMEF(void){
        estadoActual = ESTADO_INICIAL;
}
```

Implementaciones de MEF en C

```
// Función Actualizar MEF
void ActualizarMEF(void){
    switch (estadoActual) {
        case ESTADO_INICIAL:{
            // Actualizar salida en el estado
            salidas = valores;
            // Chequear condiciones de transición de estado
            if(condicionesDeTransición == TRUE){
                // Cambiar a otro estado
                estadoActual = ESTADO_N;
            }
        }
        break;
        case ESTADO_1:{
            ...
        }
        break;
```

Implementaciones de MEF en C

```
            case ESTADO_N:{
                ...
            }
            break;
            default:{
                //Si algo modificó la variable estadoActual
                // a un estado no válido llevo la MEF a un
                // lugar seguro, por ejemplo, la reinicio:
                InicializarMEF();
            }
            break;
        }
}
```

Implementaciones de MEF en C

- Ahora vamos a programarlo, ¡Manos a la Obra!

Bibliografía

- Clases de la asignatura “Sistemas Digitales” de la UNQ. Profesores: Ing. José Juarez e Ing. Eric Pernia.

- “Seminario de Sistemas embebidos” de la FI-UBA. Profesor Ing. Juan Manuel Cruz.

- Cohortes anteriores de CAPSE

¡Muchas gracias!

Seguinos:

/ProyectoCIAA

/ProyectoCIAA

@ProyectoCIAA

www.proyecto-ciaa.com.ar