

# Definición de máquinas de estados, eventos y acciones en pequeños procesadores

Aitzol Zuloaga, Jaime Jiménez, Jesús Lázaro, Carlos Cuadrado, Unai Bidarte

Departamento de Tecnología Electrónica

Universidad del País Vasco (UPV/EHU)

Bilbao, España

Email: {aitzol.zuloaga, jaime.jimenez, jesus.lazaro, carlos.cuadrado, unai.bidarte}@ehu.es

**Resumen**—Las máquinas de estados son herramientas fundamentales para el desarrollo de sistemas digitales de diversa índole. En los cursos de "Electrónica Digital", las máquinas de estado se utilizan en su versión conceptualmente más simple para desarrollar sistemas medianamente complejos. Posteriormente, en el curso de "Sistemas Digitales", las máquinas de estado se amplían para incluir nuevos conceptos propios de los sistemas basados en procesadores. En este trabajo se presenta un procedimiento para la enseñanza de las máquinas de estados y máquinas de eventos normalizando los conceptos y los procedimientos para su correcta implementación en procesadores. El objetivo final es presentar a los estudiantes un decálogo que les permita el diseño profesional de sistemas de funcionamiento eficiente y robusto.

**Keywords**—Máquinas de estado, procesadores, funcionamiento robusto, método de diseño.

## I. INTRODUCCION

Este trabajo analiza diferentes conceptos involucrados en el diseño de máquinas de estado y eventos. Se enmarca principalmente en la asignatura "Sistemas Digitales", aunque debe ser considerado como una continuación de los primeros conceptos introducidos en la asignatura de "Electrónica Digital". En concreto, estas asignaturas se imparten para la titulación de Grado en Ingeniería en Tecnología de Telecomunicación en la Escuela de Ingeniería de Bilbao [1].

La utilización de las estructuras de máquinas de estados y eventos [2], representa la forma más idónea y profesional de realización de proyectos basados en pequeños procesadores. Además son el puente entre la enseñanza de la electrónica digital y la enseñanza de los sistemas operativos utilizados en procesadores de gran capacidad.

Por otro lado, el uso de pequeños procesadores a modo de afición por un gran número de personas y la gran difusión por Internet de pequeños programas, crea una disyuntiva a la hora de decidir los tópicos que deben ser tratados en un curso profesional. Por ello, para la asignatura de "Sistemas Digitales", nos hemos decantado por enseñar cómo diseñar y realizar estructuras de programación más complejas y robustas que son utilizadas en el ámbito profesional. Por ello, dentro de la temática abordada, están las máquinas de estados y eventos, dejando en un segundo plano detalles tan específicos como la programación de un determinado periférico de un procesador específico. Estos últimos se encuentran por doquier en los manuales de los procesadores, libros, revistas y páginas web. Sin embargo, las máquinas de estados y eventos para pequeños

procesadores no se encuentran extensamente abordadas en ningún medio.

Es cierto que en los cursos de Electrónica Digital, las máquinas de estado son de obligado estudio, y ello se corrobora en las Guías Docentes de diversas titulaciones impartidas en universidades españolas. Sin embargo, la evolución del tema para el ámbito de la programación de pequeños procesadores y microcontroladores en lenguajes de bajo nivel, es una gran ausente en las Guías Docentes en las universidades españolas. Con la información escrita en libros e Internet, sucede algo similar: se encuentra mucha información al respecto para la aplicación en circuitos digitales y la programación de aplicaciones avanzadas, pero casi nada para la utilización en pequeños procesadores con lenguajes de bajo nivel. Una honrosa excepción podemos encontrarla en el libro de M. Lynch [3] para cursos en la Universidad de Florida

Las máquinas de estado y eventos, al ser estructuras muy concretas, permiten al profesorado una evaluación más sistemática y realista del desempeño del estudiante, sin dejar de lado el hecho de que se dota de una poderosa herramienta para el futuro profesional del estudiante en aras de mejorar la calidad de los productos diseñados.

En este documento se presentarán primero las definiciones de los conceptos fundamentales: los estados, los eventos y las acciones. Posteriormente, se estudiará la "máquina de eventos" en su forma más simple. A partir de ésta se pasará a estudiar la "máquina de estados y eventos" y, por último, se estudiará la "máquina de estados de evento único". Esta división del estudio en tres máquinas distintas se realiza para facilitar la asimilación, por parte del estudiante, de las variantes más comunes en que es posible encontrar estas máquinas. En el estudio de las diferentes máquinas se introduce al estudiante de una manera escalonada en los mencionados conceptos.

Cada una de estas máquinas tiene su utilidad particular en determinadas aplicaciones y permiten soluciones rápidas, de funcionamiento robusto y de fácil adaptación. En la sección correspondiente a cada una de ellas se perfilará el tipo de aplicaciones a las cuales se dirige.

La herramienta más importante en el desarrollo de las máquinas de estados y eventos son los diagramas que conceptualizan los mismos. En el estudio se ha hecho especial hincapié en seguir una serie de reglas para realizar los diagramas de estados, eventos y acciones, con el objetivo de detectar los posibles errores en el desarrollo de las máquinas y a la vez

permitir una fácil y eficiente implementación en el marco de los microcontroladores.

Por último, en el estudio se establece la conversión de los diagramas de estado, eventos y acciones en diagramas de flujo para que puedan ser implementados en cualquiera de los lenguajes destinados a procesadores. En el caso en particular de las asignaturas en la Escuela de Ingeniería de Bilbao, se utiliza el lenguaje ensamblador de los procesadores PIC16 dado que permite entender mejor el procesador desde una perspectiva de máquina electrónica [4,5,6]. El lenguaje ensamblador del PIC16 tiene un reducido conjunto de instrucciones y es especialmente indicado para el desarrollo de máquinas de estados y eventos.

Entre los principales objetivos establecidos para la docencia de la asignatura de Sistemas digitales están los siguientes:

- Conocer los principios básicos de funcionamiento interno de un procesador desde el punto de vista de máquina electrónica.
- Ampliar los conceptos fundamentales de las máquinas de estado, adquiridos en Electrónica digital, con las nuevas posibilidades que brinda el uso de la programación.
- Interiorizar la necesidad de utilizar estructuras de programación profesionales para el desarrollo de sistemas de funcionamiento eficiente y robusto.

## II. ESTADOS, EVENTOS Y ACCIONES

### A. Estados

En su forma más simple, un "estado" es un comportamiento determinado de un sistema [7]. Por ejemplo, en una lavadora, podemos definir estados tales como "cargando agua", "lavando" o "centrifugando".

Es difícil transmitir al estudiante novel la estabilidad o quietud del concepto de "estado". En el ejemplo anterior, cuando una lavadora está "lavando" pensamos en que algo se mueve, sin embargo, el módulo de control simplemente está quieto, dando paso de corriente al motor por alguna conexión específica.

En los sistemas digitales el estado es un número almacenado en un registro o memoria: "la variable de estado". Eventualmente pueden existir tantos estados como el mayor número almacenable en la variable de estado.

Gráficamente, un estado se representa como una pelota, con un número asociado. Usamos el término "pelota" para dar la idea de ser una esfera tridimensional sólida donde no hay cambios ni movimientos. Por lo general, buscando la máxima simplificación, los estados se numeran consecutivamente comenzando por el estado número cero (0). También, para una fácil comprensión por el diseñador, a cada estado se le asocia un nombre, a ser posible corto.

Con frecuencia un sistema puede presentar varios estados. Éstos se representan gráficamente como un conjunto de pelotas al cual podemos denominar "espacio de estados" (Fig. 1).

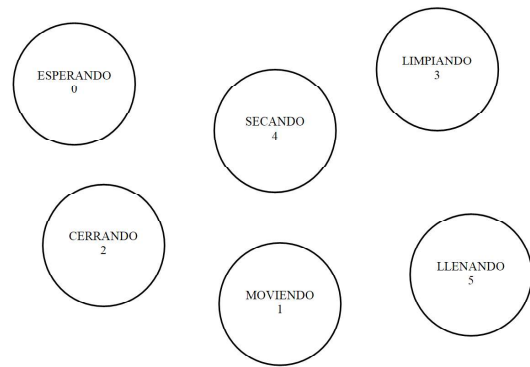


Fig. 1. Ejemplo de un espacio de estados con el número y el nombre del estado.

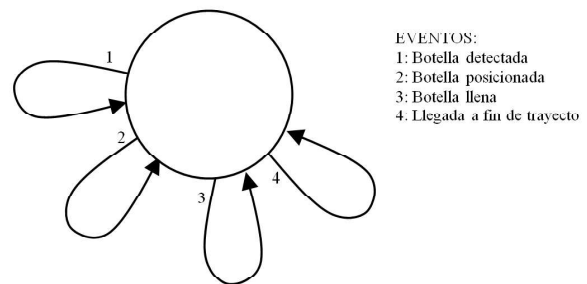


Fig. 2. Ejemplo de un diagrama de eventos.

### B. Eventos

Un "evento" es la ocurrencia de un estímulo al sistema que provocará que éste realice ciertas operaciones [8]. Por ejemplo, en un teléfono, la pulsación de cualquiera de sus teclas producirá determinadas señales sobre la línea telefónica. En este caso los eventos serán "presionada tecla 1", "presionada tecla 2", etc.

Al igual que los estados, los eventos se representan con números enteros consecutivos. Sin embargo, hay dos características que los diferencian. El evento cero (0) se suele utilizar para indicar que no ha ocurrido evento alguno. Además, los eventos se producen instantáneamente y desaparecen en cuanto se procesan, no quedan almacenados en una variable como ocurre con los estados.

Gráficamente, podemos representar los eventos como flechas saliendo de una pelota. Esta representación no suele ser habitual pero permite una uniformidad en el desarrollo del tema de máquinas de estados y eventos. Cada flecha tiene un número de evento asociado y escrito en el punto de salida de la flecha (Fig. 2). Para facilitar la formación del estudiante, se recomienda que se representen todos los eventos a excepción del evento cero. Si bien los eventos pueden tener un nombre o frase que los identifique, por simplicidad, en el diagrama sólo se representan con números. Los nombres o frases pueden colocarse a modo de leyenda al lado del gráfico.

En un sistema los eventos se recogen de diversas fuentes como pueden ser un teclado, un temporizador o un conjunto de interruptores. En todos ellas las rutinas generarán un número distinto de cero, el evento, sólo cuando ocurra un cambio. Así,

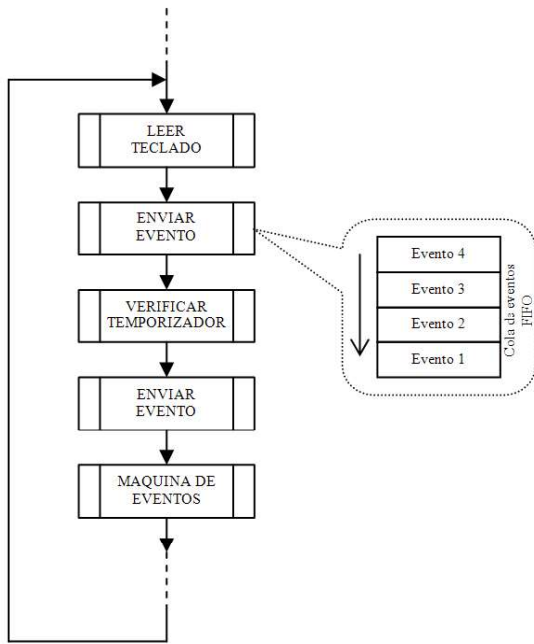


Fig. 3. Recolección de eventos y procesamiento de los mismos en el lazo del programa principal.

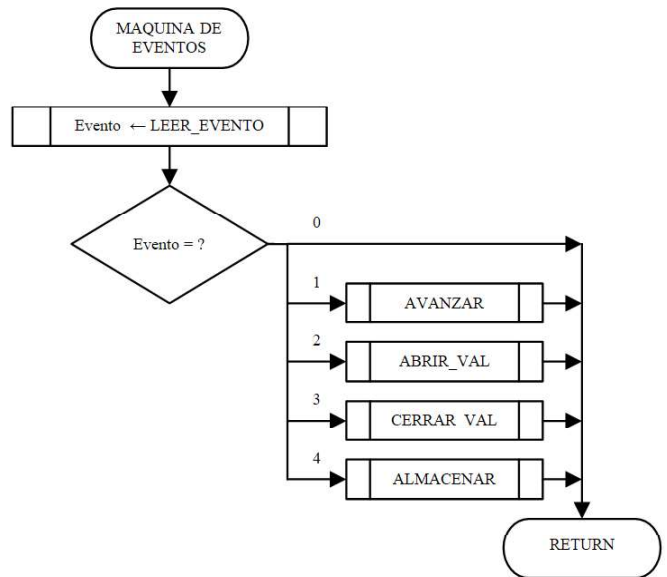


Fig. 5. Diagrama de flujo de una máquina de eventos.

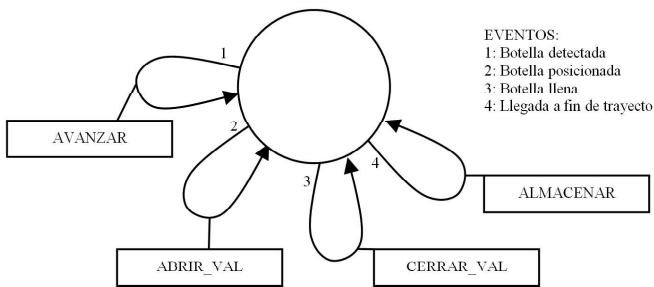


Fig. 4. Diagrama de eventos con acciones.

por ejemplo, la rutina de lectura de teclado producirá un evento en los instantes en que se presiona o deja de presionar una tecla, una rutina de temporización producirá un evento sólo en el instante que se alcance un determinado valor de tiempo, etc.

En el lazo principal del programa, los eventos recolectados por las diferentes rutinas de supervisión son almacenados por una rutina de "envío de eventos" (por lo general, en una cola FIFO) para que posteriormente sean procesados por las máquinas de eventos (Fig. 3).

C. Acciones

Las "acciones" son rutinas que efectúan las diferentes operaciones que se requieren en el sistema a consecuencia de los eventos ocurridos.

Las acciones se representan como rectángulos conectados a las flechas de los eventos. Dentro del rectángulo se pone un nombre corto para identificar la rutina de acción (Fig. 4).

```

;*****
; Máquina de eventos
; (Para ubicar en posiciones de 000h a 0FFh)

MAQEVE:
CALL LEER_EVENTO ; Se lee evento
ADDWF PCL,F      ; Se salta a la rutina de acción
RETURN           ; Evento 0: No hay evento
GOTO AVANZAR    ; Evento 1: Botella detectada - Avanzar
GOTO ABRIR_VAL  ; Evento 2: Botella posicionada - Abrir válvula
GOTO CERRAR_VAL ; Evento 3: Botella llena - Cerrar válvula
GOTO ALMACENAR ; Evento 4: Llegada a fin trayecto - Almacenar
    
```

Fig. 6. Máquina de eventos desarrollada en lenguaje ensamblador para procesadores PIC16.

III. MÁQUINAS DE EVENTOS

Las máquinas de eventos son programas que permiten ejecutar acciones ante la ocurrencia de un evento.

En su forma más simple, las máquinas de eventos se utilizan para desarrollar sistemas donde un evento siempre ejecuta una misma acción. Son muchas las aplicaciones que siguen esta forma de operar: calculadoras, teléfonos simples, organillos, etc. Son aplicaciones que pueden ser descritas con frases como "con este botón hace esto, con este otro botón hace esto otro".

El diagrama de flujo de la máquina de eventos ejemplificada por la Fig. 4 (una máquina de rellenado de botellas) se representa en la Fig. 5. En ese diagrama se observa la rutina que lee los eventos de la cola de eventos (LEER\_EVENTO). Además, se observa la característica básica de las máquinas de eventos: el evento cero representa la ausencia de eventos y por tanto no hay nunca acción asociada al mismo. Obsérvese también que en el diagrama de flujo se utilizan los mismos nombres utilizados en el diagrama de eventos para las acciones.

La implementación de las máquinas de eventos varía de un procesador a otro o de un lenguaje de programación a otro. Sin embargo, la implementación puede hacerse con gran eficiencia

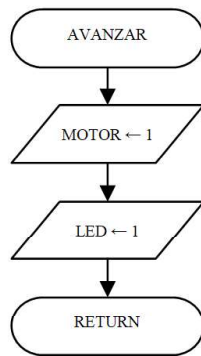


Fig. 7. Las acciones suelen ser rutinas sencillas.

en la mayoría de los procesadores si se siguen ciertas pautas:

- Seleccionar números enteros consecutivos a partir del uno (1) para especificar los eventos.
- Utilizar un solo nombre para cada acción aunque la misma reúna en sí varias acciones más pequeñas.
- Utilizar sólo acciones que no reciban parámetros.

En la Fig. 6 puede observarse la implementación básica de una máquina de eventos en los microcontroladores de 8 bits PIC16. Esta implementación, aparte de sencilla, es muy eficiente dado que sólo requiere 3 ciclos de instrucción para acceder a la rutina de acción, independientemente del evento que trate.

Las acciones, por lo general, suelen ser rutinas relativamente sencillas, que en el caso de los pequeños microcontroladores puede ser escribir algo en pantalla y accionar algún dispositivo externo (Fig. 7).

Es importante también señalar que la rutina de lectura de eventos, utilice o no el sistema de colas, debe retirar el evento leído, es decir, "consumir el evento". De esta forma se evitan los problemas ocasionados por eventos circulantes.

De esto también se desprende la conclusión de que en una acción nunca se debe volver a preguntar por el evento. Se supone que esto ha sido una pregunta previamente procesada por la máquina de eventos.

Esto no excluye casos en los que se requiera cierta información adicional, por ejemplo en el teclado de una calculadora. En ella tenemos dos grandes grupos de eventos: las teclas numéricas y las teclas de operación. En estos casos pueden definirse los eventos en dos dimensiones: el evento propiamente dicho y el dato del evento. Así, en la máquina de eventos se determina si fue un evento producido por una tecla numérica o una tecla de operación y, en la acción correspondiente a una tecla numérica, se almacena el dato del evento (el valor numérico).

#### IV. MÁQUINAS DE ESTADO Y EVENTOS

Los sistemas más modernos, donde es posible ejecutar y configurar múltiples aspectos de su funcionamiento, utilizan la combinación de los conceptos de estado y eventos. Un ejemplo muy familiar de estos sistemas lo tenemos en los

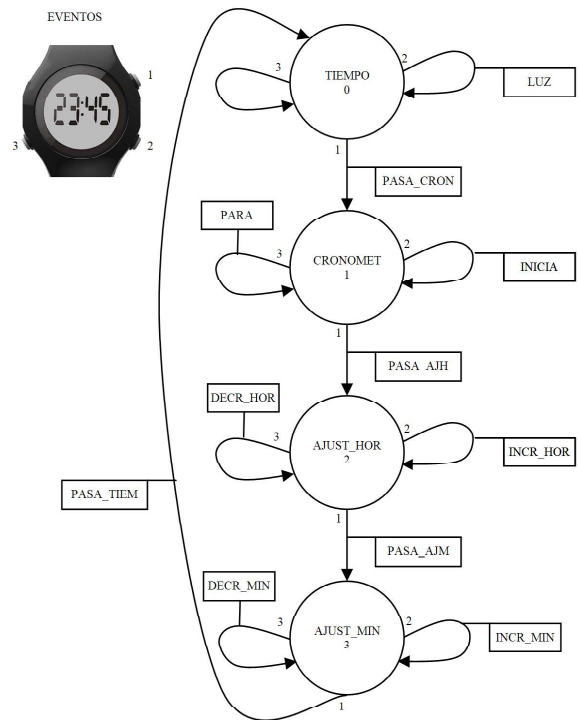


Fig. 8. Diagrama de estados, eventos y acciones en un reloj digital.

relojes digitales de pulsera donde, utilizando muy pocas teclas, disponemos de una muy amplia gama de funciones. Siguiendo este ejemplo, se puede desarrollar un diagrama donde se combinen los estados, los eventos y las acciones (Fig. 8).

Tal y como se observa en la Fig. 8, el diagrama de estados, eventos y acciones es la superposición de los diagramas de eventos para cada estado del sistema en un único diagrama. La primera característica a señalar es que los eventos en cada estado ejecutan diferentes acciones, por ello, un mismo botón o pulsador puede tener una función distinta en cada estado. Esto es la base de los sistemas actuales que trabajan con menús.

El mayor problema que presentan los estudiantes al desarrollar diagramas de estados, eventos y acciones es el cambio de paradigma de pensamiento con respecto al clásico desarrollo de programas lineales que utilizan contadores para el control de ramificaciones [9]. El problema conceptual reside en que los contadores son en esencia máquinas de estado en su forma más rudimentaria y el lidiar con varias máquinas de estado complica los programas, limita la modularidad y hace a los sistemas más propensos a fallos.

El diagrama de flujo de la máquina de estados es muy parecido a la máquina de eventos. Para el ejemplo que se viene utilizando se presenta en la Fig. 9. Lo primero que se observa, a diferencia de la máquina de eventos, es que el estado es una variable y no se obtiene de una cola por medio de una rutina. En segundo lugar, se observa que el estado cero (0) tiene una rutina asociada, algo que no ocurre con las máquinas de eventos.

Por otro lado, en las ramas de la máquina de eventos están las rutinas de acción. En la máquina de estados, las rutinas en las ramas son los nombres de las máquinas de eventos

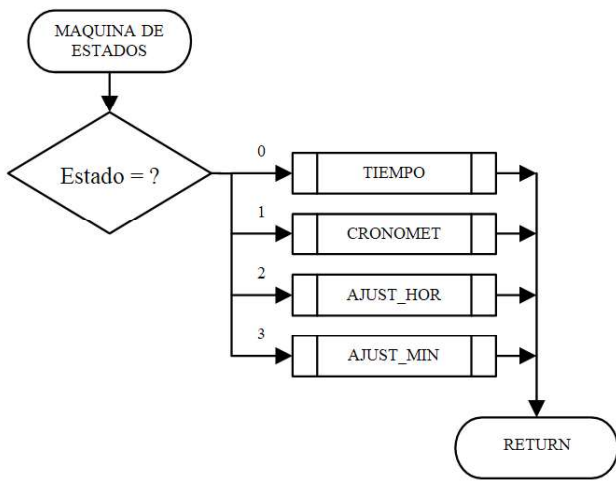


Fig. 9. Diagrama de flujo de la máquina de estados de un reloj digital.

```

;*****
; Máquina de estados
; (Para ubicar en posiciones de 000h a 0FFh)

MAQEST:
MOVWF Estado      ; Se lee estado
ADDWF PCL,F       ; Se salta a maquina de eventos correspondiente
GOTO TIEMPO       ; Estado 0: Mostrar tiempo
GOTO CRONOMET     ; Estado 1: Mostrar cronometro
GOTO AJUST_HOR    ; Estado 2: Mostrar ajuste de horas
GOTO AJUST_MIN    ; Estado 3: Mostrar ajuste de minutos
    
```

Fig. 10. Máquina de estados desarrollada en lenguaje ensamblador para procesadores PIC16.

correspondientes a cada estado (Fig. 11). Obsérvese que se denominan con el mismo nombre del diagrama de estados.

Nuevamente, la implementación varía de un procesador a otro o de un lenguaje de programación a otro, pero suele ser bastante similar a la implementación de la máquina de eventos, tal y como puede verse en la Fig. 10 para un procesador PIC16. Nuevamente, la selección de la rutina de máquina de eventos correspondiente se selecciona en tan solo 3 ciclos de instrucción.

Tal y como se observó en el espacio de estados de la Fig. 1, se disponía de un universo de modos o formas de operar del sistema, pero no había ninguna representación sobre la manera de pasar de un modo a otro. Ahora, el uso de los eventos permite cambiar de modo, o lo que es lo mismo, cambiar de estado. En este punto, es preciso señalar que el responsable de cambiar el estado es la acción, cambiando el valor de la variable de estado (Fig. 12).

V. MÁQUINAS DE ESTADOS DE EVENTO ÚNICO

Existen otras aplicaciones para las cuales las máquinas de estados tienen una utilidad especial. Son aplicaciones relacionadas con interfaces de comunicación. En ellas, por lo general, se requiere cumplir con ciertos tiempos y pasar por una serie de etapas o estados. El único evento involucrado es el tiempo. Las máquinas de estado para este tipo de aplicaciones las denominaremos "máquinas de estados de evento único".

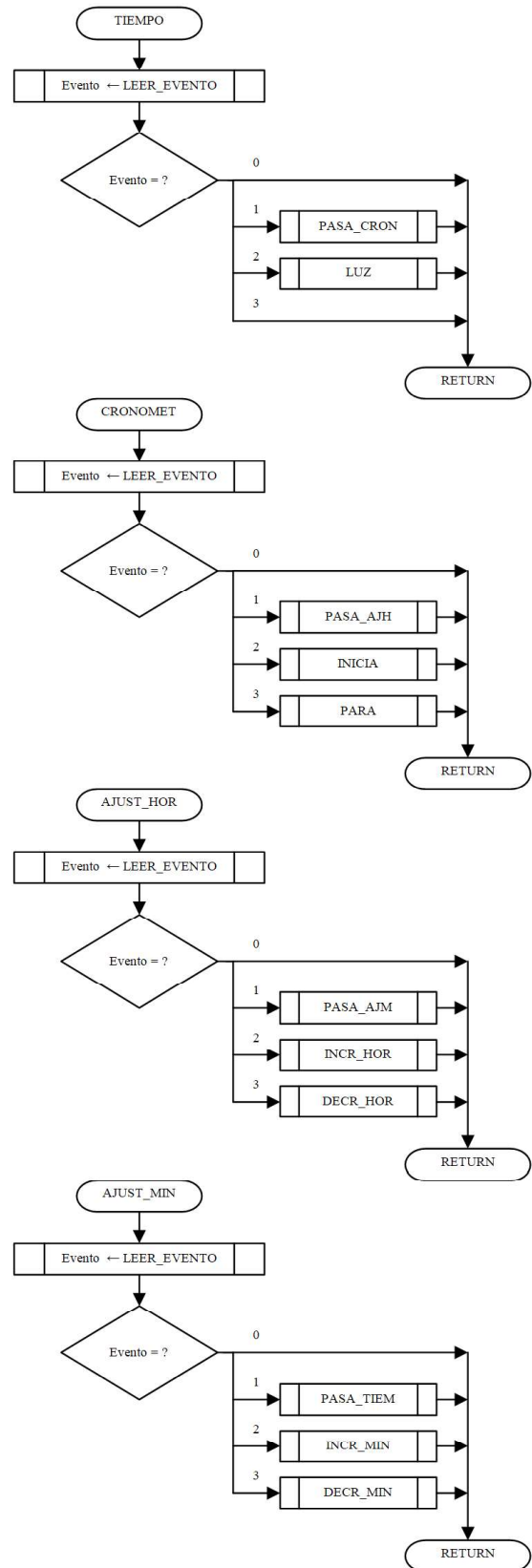


Fig. 11. Diagrama de flujo de una máquina de estados.



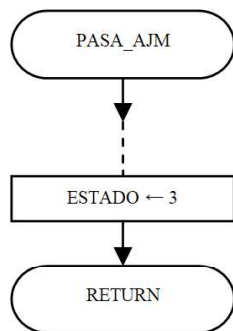


Fig. 12. El cambio de estado se realiza en las acciones.

Las máquinas de estado de evento único, a diferencia de las máquinas de eventos y las máquinas de estado y eventos, se ejecutan desde una interrupción de tiempo periódica (Fig. 13).

Es cierto que, hoy día, los microcontroladores cuentan con varios periféricos de comunicaciones, sin embargo, en muchos casos se requieren formatos de comunicación distintos a los comúnmente disponibles o se requieren más periféricos de los que se disponen en el microcontrolador. Las máquinas de estado de evento único pueden gestionar estos casos.

El diagrama de una máquina de estados de evento único es parecido a los diagramas de las máquinas de estado y eventos, con la gran diferencia de que sólo una flecha de evento sale de cada pelota de estado. En la Fig. 14 se representa un diagrama de estado de evento único de un transmisor serie sencillo. Las flechas representan el evento tiempo: la interrupción periódica. Hay una excepción: la flecha dibujada con línea punteada que sale del estado de "reposo". Esta línea punteada señala que el arranque de la comunicación se realiza por una intervención no sincronizada en la interrupción. El cambio de estado es realizado en algún punto del programa principal (Fig. 15) fuera de la interrupción.

Otra característica que distingue a las máquinas de estado de evento único es que en las ramas del diagrama de flujo se llama directamente a las acciones (Fig. 16) y no a máquinas de eventos. Además, el hecho de que la salida del estado de reposo se realice por una rutina externa, hace que la rama correspondiente al estado cero (0) no posea ninguna acción.

De la Fig. 16 se desprende también otra característica que facilita la reutilización de las acciones para distintos estados. Esto puede ser hecho siempre que se planifique correctamente la asignación del número asociado a cada estado y es válido tanto para las máquinas de estados y eventos como para las máquinas de estado de evento único. En el ejemplo de la Fig. 14, los estados son consecutivos. La rutina de acción de transmisión de bits TX\_BIT pudiera ser desarrollada de la forma mostrada en la Fig. 17. Obsérvese que la extracción del bit de dato se hace apuntando con el estado y que el cambio de estado se realiza con un incremento de la variable de estado. En cualquier caso es importante que en las acciones no se pregunte nuevamente por el estado, ya que se supone que esa discriminación ya está hecha en la máquina de estados.



Fig. 13. Ubicación de la máquina de estados de evento único en la interrupción de tiempo.

## VI. CASOS ESPECIALES

Las máquinas de estado y eventos admiten muchas variantes que permiten utilizarlos en muchas aplicaciones de diversa índole. A continuación se explicitan unos casos especiales.

### A. Eventos ramificados

Hay aplicaciones, como aquellas que exigen una clave de acceso, que después de recibir una cierta cantidad de dígitos se validan los dígitos recibidos. Con frecuencia estas validaciones se confunden con "estados de validación", lo cual es incorrecto. Una validación es una acción y en ella, dependiendo de la validación o no, se decide cuál va a ser el próximo estado (Fig. 18).

### B. Eventos de tiempo

Otro caso que se da con frecuencia en las máquinas de estado y eventos, es la combinación de eventos producidos por teclas y eventos producidos por temporizadores. Este es el caso de relojes y crono-termostatos. En este tipo de dispositivos además de tener varios eventos generados por las diferentes teclas, se tiene un evento que marca el paso de los segundos. Para estos casos, los diferentes estados del sistema deben contemplar la actualización de los contadores de tiempo en, prácticamente, todos los estados.

### C. Supervisiones de tiempo

Otra aplicación donde también surgen eventos relacionados con el tiempo son aquellas en el que se quiere supervisar un determinado tiempo. Por ejemplo, una calculadora que se apaga luego de estar un tiempo sin que se presione una tecla. En estos casos, las acciones asociadas a los eventos de tecla llaman a una rutina que inicializa un contador de tiempo. Cuando se concluye el tiempo asignado al contador, se genera el evento de "fin de supervisión de tiempo", que será atendido por la acción correspondiente en el estado correspondiente.

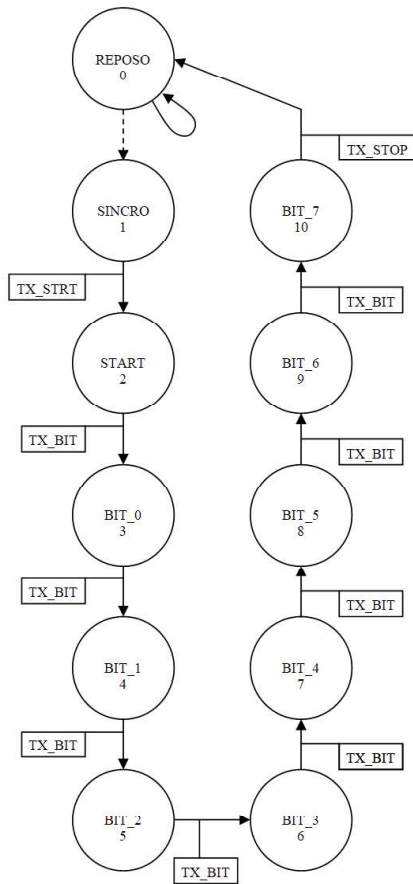


Fig. 14. Diagrama de estados de evento único de un transmisor serie.

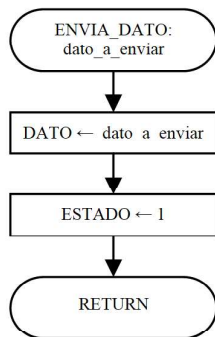


Fig. 15. Activación de la máquina de estados de evento único por rutina en el lazo principal.

VII. DECÁLOGO

A manera de resumen se presenta en esta sección un pequeño decálogo de pautas que deben tomarse en cuenta en el diseño de las máquinas de estados, eventos y acciones:

- Debe determinarse el tipo de máquina más adecuado al funcionamiento deseado del sistema.
- Se diseña el diagrama de estados-eventos-acciones del sistema.

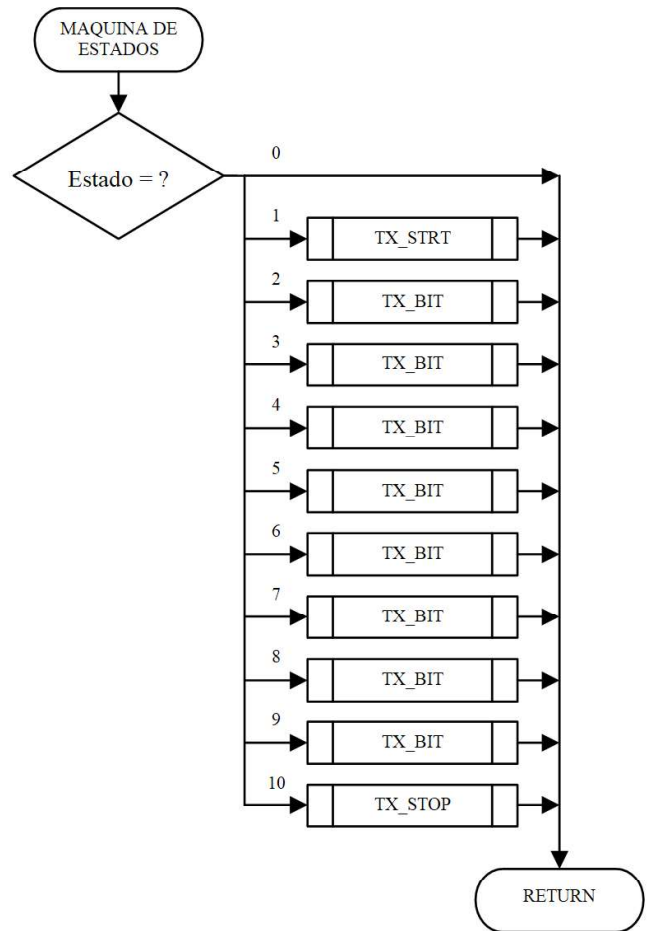


Fig. 16. Diagrama de flujo de una máquina de estados de evento único para un transmisor serie.

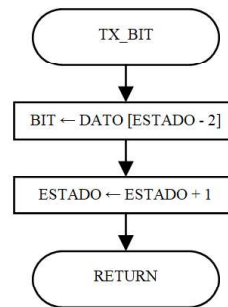


Fig. 17. Diagrama de flujo de una rutina de acción que puede ser reutilizada en diferentes estados.

- Se numeran todos los estados comenzando por el cero (0). De ser posible, la numeración debe ser consecutiva en las ramas que presente el diagrama. Esto ayuda a la simplificación de ciertas acciones.
- En cada estado deben representarse todos los eventos existentes salvo el evento cero (0). Este último debe omitirse para minimizar posibles confusiones.
- Si se obtiene algún estado del cual no pueda pasarse a

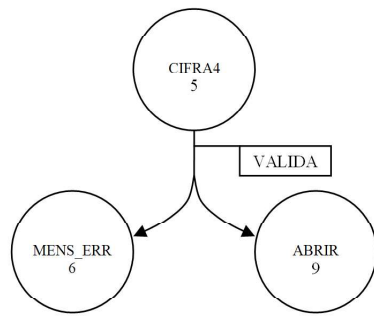


Fig. 18. Los eventos se ramifican a causa de una acción.

otro estado por medio de uno de los eventos definidos, es muy posible que el "estado" no sea tal o que falta definir algún nuevo evento.

- Es posible que los eventos que no obliguen a un cambio de estado no tengan acción asociada, pero es imperativo que los eventos que generen cambios de estado tengan asociada una acción (porque el cambio de estado lo realiza la acción).
- En los eventos que presenten ramificaciones, la rutina de acción realiza la ramificación, por tanto, al dibujar el rectángulo de acción debe hacerse antes de la ramificación de la flecha de evento.
- Desarrollar el diagrama de flujo de la máquina de estados. En el caso de una máquina de estados y eventos, se sugiere poner a cada ramificación el mismo nombre utilizado para el estado en el diagrama. En el caso de una máquina de estados de evento único, los nombres de las ramificaciones serán los nombres de las acciones. El número de ramificaciones debe coincidir con el número de estados en los diagramas.
- Las acciones nunca reciben ni retornan parámetros directamente (en lenguaje C serían tipo "void" en cuanto a entrada o salida de parámetros).
- Si por alguna razón aparece la necesidad de utilizar un contador en las acciones, es muy probable que el diagrama de estados y eventos sea incorrecto. Un contador es una máquina de estados encubierta y pueden producirse problemas de sincronización entre las dos máquinas de estado.
- La existencia de variables lógicas (banderas) en las acciones también puede ser considerado como un contador de un bit y por tanto se aplica lo expresado en el párrafo anterior.
- Las máquinas de estado y eventos son estructuras donde se detecta el valor tanto del estado como del evento, por lo que resulta una posible incoherencia preguntar en las acciones por estas variables.

## VIII. CONCLUSIONES

Con frecuencia, los cursos de Sistemas Digitales se centran en cursos excesivamente teóricos sin aplicación práctica, o excesivamente prácticos dedicados a un procesador o plataforma

en particular. El estudio de las máquinas de estados y eventos en tales cursos, además de ser una herramienta que los estudiantes pueden utilizar con utilidad en su ejercicio profesional, permite organizar el curso sobre unas estructuras que son útiles para cualquier procesador y lenguaje de programación.

Por otro lado, las máquinas de estados y eventos, permiten evaluar con mayor objetividad el desempeño del alumno, porque son estructuras muy concretas que deben cumplir ciertas normas para que puedan ser utilizados en grupos de trabajo, competencia ésta que es muy exigida por el sistema educativo [10] y las empresas.

También, el hecho de centrar el curso de Sistemas Digitales, en la utilización de un procesador o plataforma en particular, explicar un lenguaje y dejar que los estudiantes desarrollen alguna aplicación, encierra grandes riesgos: una gran cantidad de aplicaciones de dudosa calidad disponibles en Internet que favorece el plagio y la creación de malos hábitos en el desarrollo de programas.

## AGRADECIMIENTOS

El trabajo descrito en esta publicación ha sido generado y patrocinado por el Departamento de Educación, Política Lingüística y Cultura del Gobierno Vasco en base a las ayudas para apoyar las actividades de grupos de investigación del sistema universitario vasco IT978-16 y por el Ministerio de Economía y Competitividad a través del proyecto de investigación TEC2017-84011 y los fondos FEDER.

## REFERENCIAS

- [1] BOPV. RESOLUCIÓN de 20 de diciembre de 2010, de la Vicerrectora de Ordenación Académica, de la Universidad del País Vasco/Euskal Herriko Unibertsitatea, por la que se procede a la publicación del plan de estudios del Grado en Ingeniería Técnica de Telecomunicación de la Universidad del País Vasco/Euskal Herriko Unibertsitatea. Boletín Oficial del País Vasco, nº 44 de 4 de marzo de 2011.
- [2] A. Zuloaga, A. Astarloa, *Sistemas de procesamiento digital*, Madrid: Delta Publicaciones, 2008.
- [3] M. A. Lynch, *Microprogrammed State Machine Design*. Boca Raton: CRC Press, 1993.
- [4] A. Zuloaga, A. Astarloa, *Sistema Digital PICTOR*, Bilbao: Servicio de publicaciones de la Escuela de Ingeniería de Bilbao, 2013
- [5] A. Zuloaga, *Laboratorio de Sistemas Digitales*, Bilbao: Servicio de publicaciones de la Escuela de Ingeniería de Bilbao, 2013
- [6] A. Zuloaga, J. Jiménez, I. Kortabarria, J. Andreu, " PICTOR: circuito impreso para montar un sistema digital básico," , pp. 315- 319, XI Congreso de Tecnologías, Aprendizaje y Enseñanza de la Electrónica (TAAE 2014), Sevilla (España), Junio 11-13, 2014.
- [7] "State (computer science)", Wikipedia, 2018. [En línea] Disponible en: [https://en.wikipedia.org/wiki/State\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/State_(computer_science)) [Accedido: 20/02/2018].
- [8] "Event (computing)", Wikipedia, 2018. [En línea] Disponible en: [https://en.wikipedia.org/wiki/Event\\_\(computing\)](https://en.wikipedia.org/wiki/Event_(computing)) [Accedido: 20/02/2018].
- [9] J. Jiménez, C. Cuadrado I. Kortabarria, J. Andreu, A. Zuloaga, , "Dificultades al aprender a especificar máquinas de estados en sistemas microprogramados," , pp. 428, XII Congreso de Tecnologías, Aprendizaje y Enseñanza de la Electrónica (TAAE 2016), Sevilla (España), Junio 22-24, 2016.
- [10] BOE. Orden CIN/352/2009, de 9 de febrero, por la que se establecen los requisitos para la verificación de los títulos universitarios oficiales que habiliten para el ejercicio de la profesión de Ingeniero Técnico de Telecomunicación, 20 de febrero de 2009.