MICROPROCESADORES

PRÁCTICA 4

AUTÓMATAS CONTROLADOS POR EVENTOS

DISEÑO FINAL

TITULACIONES DE GRADO DE LA
ETSIS DE TELECOMUNICACIÓN

DEPARTAMENTO DE INGENIERÍA TELEMÁTICA Y ELECTRÓNICA

UNIVERSIDAD POLITÉCNICA DE MADRID

PRIMAVERA 2020 – 2021

ÍNDICE

1. INTRODUCCIÓN 3
2. FUNCIONALIDAD MÍNIMA 4
3. OBJETIVOS DE LA PRÁCTICA 6
4. FUNCIONALIDAD OPCIONAL 7
   4.1. Reinicio del sistema 8
   4.2. Nivel de alerta 8
   4.3. Comunicaciones serie 8
   4.4. Nueva calibración 9
5. MEMORIA 9
6. SUBIDA DE RESULTADOS A MOODLE 10

1. INTRODUCCIÓN

En este enunciado se describen las especificaciones del sistema que debe diseñarse en la práctica 4.

El sistema final es de suficiente complejidad como para que sea necesario abordarlo empleando la técnica de máquinas de estados finitos (autómatas) controladas por eventos. El estudiante tendrá en cuenta que será obligatorio emplear varios autómatas trabajando conjuntamente para completar el sistema.

Esta práctica tiene tres sesiones. Solo se contemplan actividades previas para la primera de ellas, como se detallará más adelante. Esta actividad previa tiene un peso de 10 puntos sobre 100 en la calificación global de la práctica.

La práctica propone una funcionalidad mínima que es obligatorio implementar y 4 opciones diferentes que pueden añadirse al diseño. La funcionalidad mínima tiene un peso de 40 puntos sobre la calificación global de la práctica, mientras que las opciones adicionales tienen un peso global de 30 puntos.

A diferencia de las anteriores prácticas, en esta el estudiante deberá escribir una **memoria final** de la misma (con un peso de 20 puntos) que incluirá, al menos:

- **Descripción de los autómatas** empleados en el diseño, lo que incluirá:
  - diagramas de estados de los mismos;
  - descripción de los eventos o mensajes leídos por cada autómata (y qué periférico o autómata los genera);
  - descripción de los mensajes generados por cada autómata (y a qué otros autómatas van dirigidos);
  - relación de las acciones realizadas en cada estado (Moore) o transición (Mealy).
- **Relación de opciones** que se han implementado y descripción del enfoque empleado para realizarlas.
- **Descripción de las tácticas** empleadas para codificar aquellas partes del diseño no implementadas mediante máquinas de estados finitos.

No incluya el código final en la memoria, incluya solo aquellos fragmentos de código que necesite para ilustrar alguna particularidad del funcionamiento.

El circuito final hará uso de todos los recursos *hardware* empleados hasta el momento (pulsadores, LED, *display* de 7 segmentos, LDR y sensor telemétrico ultrasónico).

Para la realización de esta práctica no se permite el uso de la biblioteca de funciones *sw_tick_serial* ni de otras bibliotecas que no hayan sido codificadas directamente por usted (excepto la propia biblioteca *mbed*, claro está). Téngalo en cuenta para no incurrir en casos de plagio. Abundando en lo mismo, cada pareja debe trabajar independientemente de las demás, la similitud excesiva de código de diferentes parejas dará lugar a la aplicación de lo dispuesto en la normativa de la UPM respecto a casos de plagio en pruebas de evaluación.

2. Funcionalidad mínima

Se implementará un medidor de objetos brillantes, respecto a un valor de referencia calibrado, y lejanos empleando, respectivamente, la resistencia LDR y el sensor telemétrico ultrasónico. Las especificaciones del sistema son:

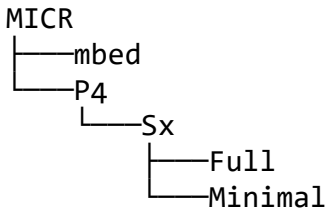
1. Tras la inicialización del sistema este permanecerá apagado, con todos los LED y el *display* apagados (segmentos y cátodos permanecerán en reposo) y sin actividad en la señal de *trigger* del sensor.
2. Tras una pulsación de duración mayor a 1 s sobre el pulsador derecho, el sistema se encenderá, mostrando en el *display* el mensaje «On» al máximo brillo posible, los LEDS permanecerán apagados. Se permanecerá en esta situación hasta que se realice una pulsación, no necesariamente mayor a 1 s, sobre el pulsador izquierdo. Una pulsación larga (de más de 1 s) sobre el pulsador derecho hará que el sistema se apague.
3. Tras dicha pulsación sobre el pulsador izquierdo se mostrará el mensaje «-A» en el *display*. A la vez, se encenderán los LEDS central y derecho. En ese momento, se realizarán 3 medidas de luminosidad con una espera entre ellas de 500 ms.
4. Tras la última medida el sistema calculará la media de las tres medidas, se almacenará en una variable de resolución uint_16 llamada brightness_calibration y volverá al estado descrito en

el punto 2. Además, a partir de este momento el brillo del *display* será proporcional al valor de brightness_calibration.

5. Tras una nueva pulsación del pulsado izquierdo, y habiendo comprobado que se ha realizado la calibración, el sistema encenderá los LEDS central y derecho durante 2 segundos, medirá la luminosidad y si el valor medido es superior o igual a brightness_calibration se incrementará en uno el valor de una variable llamada brightness_object. Además, se mostrará en el *display* el valor de dicha variable durante dos segundos y se volverá al estado descrito en el punto 2. Si el valor medido fuera inferior a brightness_calibration se mostrará en los displays «--», durante 2 segundos, se apagarán los LEDS central y derecho, y se pasará al siguiente punto.
6. En este momento se iniciará la toma de dos medidas de distancia independientes, es decir, hasta que no se tenga la distancia medida de la primera medida no se iniciará la segunda, dejando un tiempo entre medidas de 2 segundos. En el momento que una medida esté disponible se mostrará en el *display*. En función del resultado obtenido se tomarán las siguientes acciones:
    - Si la segunda medida resulta en un valor superior a la primera se incrementará en uno el valor de una variable far_object, se mostrará el valor de dicha variable en el *display* durante 2 segundos y se volverá al estado descrito en el punto 2.
    - En caso contrario se mostrará el mensaje «--» e, igualmente, se volverá al punto 2.
    - Si en algún momento la medida fuera superior a 99 cm se volverá al paso 2 mostrando la leyenda «Fr» durante 2 segundos.
    - Si alguna medida realizada fuera errónea se volverá al paso 2 mostrando la leyenda «Er» durante 2 segundos.
7. El brillo de los LEDs al encenderse será el máximo posible.
8. Deberá dormirse al procesador siempre que sea posible.

3. OBJETIVOS DE LA PRÁCTICA

El objetivo de esta práctica es que aplique adecuadamente el diseño de sistemas reactivos basado en máquinas de estados finitos controladas por eventos y que interactúan entre ellas. Por ello es *obligatorio* que plantee su diseño basado en tal paradigma. En este sentido existe en *Moodle* un proyecto de *Keil µVision* que puede descargar y que puede emplear como base para esta práctica (carpeta Minimal). Dicho proyecto está dividido en cuatro máquinas de estados finitos (FSM), aunque cada una se entrega en forma de librería compilada —.h más .lib—. Se recomienda que estudie los .h de cada una de ellas para entender su interfaz y que vaya sustituyendo cada .lib (incluido el to_7seg.lib —que no es una FSM— suministrado) por su propio código para que se mantenga el funcionamiento del sistema completo y poder verificar fácilmente la funcionalidad de su código. No debe modificar los ficheros .h dados, y tendrá que emplear los mensajes que en ellos se describen, con los significados ahí descritos y sin necesidad ni posibilidad de añadir nuevos. La jerarquía de carpetas debe ser:



donde mbed se refiere a la librería mbed de la que ya dispone por las prácticas anteriores.

Las FSM son:

- Gestión de los pulsadores. Sería la primera que implemente y pruebe. Debe gestionar los posibles rebotes en los pulsadores izquierdo y derecho, y generar los mensajes necesarios para informar a otras FSM del sistema de que se ha producido una pulsación larga en el pulsador derecho o de cualquier duración en el izquierdo. No es necesario que contemple la posibilidad de que se activen varios pulsadores a la vez.
- Gestión del sensor telemétrico ultrasónico. Esta FSM se encargará del control del sensor telemétrico ultrasónico. Lanzará una medida

cuando otra FSM así lo indique y, al terminar la medida, informará del resultado de la misma mediante mensajes.
- Gestión del *display* de 7 segmentos. Se encargará de la multiplexación del *display* y del control de su brillo. Será gobernado desde otras FSM mediante mensajes que le indiquen si debe modificar el mensaje visualizado (y su valor concreto) y también de cuándo debe encender o apagar el *display*.
- Control. Esta será la última FSM que implemente y pruebe. Desde ella se gobernarán los LED y las demás FSM del sistema para que éste funcione según lo descrito en las especificaciones.

El código completo debe estar convenientemente modularizado, existiendo al menos un fichero `.cpp` (con sus correspondientes ficheros de cabeceras) por cada FSM del sistema. Las complejidades de cada una de las cuatro FSM anteriores —expresadas en números de línea de código— son similares.

Al inicio de la primera sesión de laboratorio deberá entregar al docente —en papel— los diagramas de estados de los autómatas que pretende emplear para resolver la funcionalidad. Dicha descripción incluirá también la relación y descripción de eventos y mensajes (con sus parámetros, si fuese el caso) empleados.

Termina aquí la descripción de la funcionalidad mínima requerida y de las actividades previas a la primera sesión presencial.

4. Funcionalidad opcional

Se relacionan a continuación cuatro opciones que puede añadir al funcionamiento mínimo descrito en la sección 2, Funcionalidad mínima. Cada una de ellas tiene un peso de 12 puntos en la calificación de la práctica. Puede implementar todas las que considere oportuno, aunque la calificación global de estas opciones nunca será superior a 30 puntos.

4.1. Reinicio del sistema

Se modificará el funcionamiento del sistema de modo que estando el sistema de medida encendido, y después de 30 s sin acción ninguna en el estado 2 **por parte del usuario**, volverá automáticamente al paso 1 descrito en la sección 2.

4.2. Nivel de alerta

Si en algún momento cualquiera de las variables brightness_object y/o far_object superase un valor de 5 el LED izquierdo comenzaría a parpadear con una frecuencia de 3Hz siempre que el sistema se encuentre en el estado descrito en el punto 2 de la sección 2. Esta situación solo se revertirá si el sistema se apaga (punto 1 de la sección 2), momento en el que el valor de ambas variables de pondrá a cero.

4.3. Comunicaciones serie

Se añadirá la opción de que el sistema envíe datos al PC a través del puerto serie (57 600–8N1). En este caso se enviará la siguiente información:

- Cuando se enciende el sistema se enviará el mensaje «*System on*».
- Al entrar en la fase 3, indicada en la sección 2, se enviará el mensaje «*Calibrating…*».
- En todo momento se mandará, cada 4 segundos, la siguiente información: "*Brightness calibration value = XX, brightness_object = Y, far_object = Z*"
- Si se muestra en los *displays* «Fr», se enviará «*Too far*».
- Cuando se apague el sistema se enviará el mensaje «*System off*».
- Siempre que el sistema esté apagado no se enviará información.

4.4. Nueva calibración

Si, estando el sistema encendido, en el estado descrito en el punto 2 del apartado 2, y habiéndose realizado ya la calibración del sistema, se pulsase el pulsador central se procederá a realizar una nueva calibración (punto 3 de la sección 2), poniendo a cero el valor de la variable brightness_calibration, y por tanto ajustando de nuevo el brillo del *display*. En este caso no se modificará el valor de las variables brightness_object ni far_object.

En los ficheros adjuntos a este enunciado encontrará un ejecutable —carpeta Full— que implementa todas estas opciones. No dude en probar su funcionamiento para familiarizarse con el comportamiento esperado.

5. MEMORIA

Como ya se ha comentado, deberá redactar una memoria de esta práctica (cuyo peso en la calificación de esta actividad será de 20 puntos). En la sección 1, Introducción, se describieron los contenidos de esta memoria, aunque por comodidad se repiten aquí:

- Descripción de los autómatas empleados en el diseño, lo que incluirá:
  - diagramas de estados de los mismos;
  - descripción de los eventos o mensajes leídos por cada autómata (y qué periférico o autómata los genera);
  - descripción de los mensajes generados por cada autómata (y a qué otro autómata van dirigidos);
  - relación de las acciones realizadas en cada estado (Moore) o transición (Mealy).
- Relación de opciones que se han implementado y descripción del enfoque empleado para realizarlas.
- Descripción de las tácticas empleadas para codificar aquellas partes del diseño no implementadas mediante máquinas de estados finitos.

No incluya el código final en la memoria, incluya solo aquellos fragmentos de código que necesite para ilustrar alguna particularidad del funcionamiento.

Esta memoria deberá subirse a *Moodle* en forma de fichero comprimido (*7-Zip*, una entrega por cada estudiante, ambos miembros de la pareja deberán subir el mismo fichero) conteniendo un único fichero en formato PDF (extensión .pdf) o Microsoft Word (extensiones .doc o .docx). La fecha límite para esta entrega será 48 horas después del límite impuesto para subir el proyecto.

6. Subida de resultados a Moodle

Recuerde que para la realización de esta práctica **no se permite el uso de la librería *sw_tick_serial* ni de otras librerías que no hayan sido codificadas directamente por usted** (excepto la propia librería *mbed*, claro está). Téngalo en cuenta para no incurrir en casos de plagio.

Tenga presente lo dicho en la normativa de la UPM acerca del plagio. **El código que presente debe ser original suyo.** Todas las entregas serán sometidas a inspección automática mediante el *software* antiplagio *Turnitin* integrado en la plataforma *Moodle*.

Tal como se ha descrito anteriormente, deberá subir a *Moodle*, con fecha límite 48 horas después del límite impuesto para la entrega del proyecto, la memoria de la práctica, en forma de fichero comprimido (*7-Zip*) conteniendo un único documento en formato PDF (extensión .pdf) o Microsoft Word (extensiones .doc o .docx).

En algún momento antes de la finalización de la tercera sesión presencial de la práctica deberá enseñar su sistema funcionando al docente. Tenga en cuenta que no será posible atender a todos los alumnos del grupo durante los últimos minutos de la sesión, por lo que es recomendable no esperar al último momento para enseñar el sistema. Una vez el docente lo valide, elimine todas las carpetas ~build y ~listings de todos los proyectos de *Keil µVision 5* de esta práctica. Borre también todos los ejecutables de ejemplo (.bin) y librerías (.lib) que se han suministrado y que no estén en uso y, solo entonces, comprima la carpeta MICR\P4 en formato .7z (nivel de

compresión Ultra). Suba este archivo comprimido a *Moodle* en el enlace correspondiente (una entrega por cada estudiante, ambos miembros de la pareja deberán subir el mismo fichero). La fecha límite para esta entrega el día de la tercera sesión a las 23:30 horas. Es decir, si su grupo de laboratorio es, por ejemplo, el M03M04 deberá entregar el código del proyecto, ese mismo martes antes de las 23:30 horas, y la memoria el jueves de esa misma semana antes de las 23:30 horas.

Si su diseño no pudiera ser recompilado posteriormente a partir de su entregable subido a* Moodle*, se entenderá que no ha realizado la entrega.

Terminan aquí las tareas a realizar para esta práctica.