INFORMATICA II	SUBTEMA	TEMA #	PARCIAL	24/07/2021
	MdE	11	# 1	

Estimado alumno,

A partir de la siguiente página encontrará el enunciado de su examen. Le recordamos que ya no puede cambiarlo y que ya está corriendo el reloj.

La realización del ejercicio incluye la modelización del problema utilizando Máquinas de Estados (MdE) y su codificación en C.

Para la realización del modelo sugerimos utilice uModel Factory (uMF) pero la codificación debe ser realizada por Ud. sin utilizar la herramienta de codificación de uMF.

Al respecto, tenga presente que en caso de entregar una codificación en C realizada con uMF NO SERA TENIDA EN CUENTA.

Ud. deberá resolver el ejercicio en su computadora. Luego, deberá subir comprimidos los archivos realizados (preferentemente zip). Si lo desea, puede agregar un documento de texto con explicaciones que quiera hacerle llegar a su profesor.

Le recordamos:

	Fecha	tópico	Examen disponible durante	Tiempo para hacerlo desde bajada	Modalidad	Entregables
3	Sáb 24/7 desde las 8am hasta Dom25 – 19hs.	**MdE**	35hs.	3hs. (*)	ejercicio práctico de modelización y codificaión	Archivo .zip (ver NOTA 1 y 2)

(*) La duración nominal del examen se cuenta desde el momento en que el alumno comienza a realizar el examen dentro del Aula Virtual (AV). La cuenta del tiempo la lleva la plataforma Moodle automáticamente. Si bien es cierto habrá 35hs a partir del horario de comienzo para comenzar la evaluación, y que las evaluaciones en sí pueden realizarse en 3 horas, debe tenerse presente que, a las 18:59:59 del domingo quedará bloqueada la subida del archivo final. En criollo: Si va a tomar el examen de MdE a las 18hs del domingo 25 de julio, solo tendrá una hora para entregarlo.

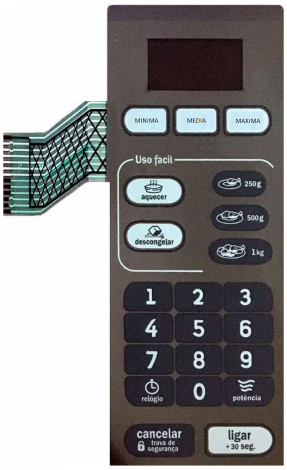
NOTA1: Renombre el archivo ZIP con su APELLIDO y NOMBRE.

NOTA2: El ZIP debe incluir los fuentes C del proyecto realizados sobre *MCUXpreso* (o sobre otro editor de texto de su preferencia) + el archivo .umf correspondiente a la/s MdE realizada/s en *uModel Factory* + archivo de texto con aclaraciones, de considerarlo necesario.

Cátedra de Info2

Nombre y Apellido	Nº Legajo	Calificación

figura1



Se desea desarrollar el software de una placa de control para un microondas (Figura 1).

El microondas tiene las siguientes funciones que se deben contemplar en el código:

1- ***Seteo lento:***
   a. Se selecciona el nivel de cocción, que pueden ser 3 (mínima, media y máxima) y se seleccionan por medio de tres botones.
   b. Se ingresa el tiempo con los pulsadores numéricos (considere solamente segundos) (mínimo 5 segundos y máximo 8000 segundos)
   c. Se presiona el pulsador de "Encendido" para finalizar, de esta forma se activara el circuito externo

2- ***Seteo rápido:***
   a. Cada vez que se presiona el pulsador "Encendido" sin que se haya seteado tiempo se debe incrementar el tiempo en 30 segundos y activar el circuito externo a máxima potencia.

Para la implementación de la lectura del teclado se deberá llamar a la función **unsigned char TECLADO_Leer(void)** que devolverá el código de la tecla presionada o el código NO_KEY en caso de que no se presione ninguna.

La tecla CANCELAR cancela todo y vuelve todo al sistema a su estado inicial, y el tiempo a 0

Para activar el circuito de encendido se deberá llamar a la función **void UART_Enviar(unsigned char Nivel)**, con el campo nivel en valor 1, 2 o 3, según la potencia seleccionada. Para cancelar se deberá enviar el campo nivel en 0.

Una vez activado el sistema de calentamiento, se controlará que el mismo funcione correctamente, mediante un termostato colocado en el dispositivo. El mismo puede leerse con la función ***unsigned char Termostato (void)*** y deberá activarse luego de 30 segundos de iniciado el calentamiento (en tiempos de cocción menores a 30 segundos dicho termostato no tendrá relevancia). En caso de que el mismo no se active pasados los 30 segundos, se indicará que hay un error en el microondas, haciendo funcionar una sirena en forma intermitente (con períodos de encendido/apagado de 1 segundo), durante 5 segundos, para luego apagar el sistema y volver al estado inicial. La sirena podrá ser activada con la función ***void Buzzer (unsigned char estado)***.

#define	CODIGO_TECLA_0 a 9	0 a 9	#define	CODIGO_TECLA_MAXIMA	12
#define	CODIGO_TECLA_MINIMA	10	#define	CODIGO_TECLA_CANCELAR	13
#define	CODIGO_TECLA_MEDIA	11	#define	CODIGO_TECLA_ENCENDIDO	14
#define	NO_KEY	0xFF			

Se pide:

- Realizar el diseño de la(s) máquina(s) de estado que controlen el sistema.
- Implementar las mismas en C, incluyendo las funciones de inicialización que setean las variables y los estados que sean necesarios.

- Implementar un programa que invoque a la/s máquina/s de estado desarrollada/s en el punto anterior.

Nota: A continuación se detallan las funciones que tiene a su disposición, junto con una breve descripción de cada una:

- ***Entradas:***

  //Lectura del teclado. Cada vez que el operario presiona una tecla la función retorna el valor de la tecla presionada SOLO UNA VEZ (**no** continúa devolviendo el valor mientras la tecla está presionada, solo avisa que el operario presionó dicha tecla). En caso de que no haya sido presionada ninguna tecla, retornará NO_KEY

  unsigned char TECLADO_Leer (void);

  //Lectura del termostato. Cuando la temperatura alcanza la temperatura fijada por en usuario, la función devuelve un 1. En caso contrario, la función devuelve 0

  unsigned char Termostato (void);

- **Salidas:**

  //Función para fijar el valor de la potencia y comenzar el calentamiento. Recibe como argumento la potencia a la que debe funcionar el horno (1, 2 o 3)

  void UART_Enviar(unsigned char Nivel);

  //Operación de la sirena. Enviando un 1 se activa la alarma sonora, y enviando un 0 se desactiva la misma

  void Buzzer (unsigned char estado);

- ***Temporizaciones:***

  //Da comienzo al timer indicado por el argumento ***event***, que contará ***t*** tiempo, medido de acuerdo a la base indicada en el parámetro ***base*** (DEC, SEG o MIN). Finalizado el mismo se invocará a la función ***handler***, que usted deberá desarrollar.

  void TimerStart (unsigned int event, unsigned int t, void (*handler)(void) , unsigned int base)

  //Detiene el timer indicado por el argumento ***event***

  void Timer_Stop (unsigned int event)

- ***Inicialización del sistema:***

  //Inicializa el Hardware del sistema:

  void InicializarHW (void);

Desarrolle el código preferentemente sobre MCUXpresso, y el diseño de las máquinas de estado en uModel. Si necesita enviar aclaraciones a su profesor, escríbalas en un documento de texto. Comprima todos los archivos y súbalos al AV.