

<b>INFORMATICA II</b>	SUBTEMA	TEMA #	PARCIAL	<b><u>24/07/2021</u></b>
	<b>MdE</b>	<b>8</b>	<b># 1</b>	

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.	<b>MdE</b>	<b>35hs.</b>	<b>3hs. (*)</b>	ejercicio práctico de modelización y codificación	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.

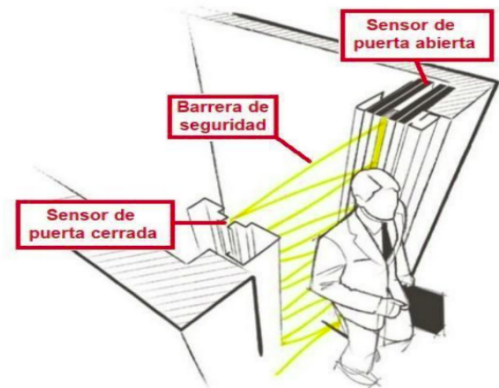
NOTA1: Renombre el archivo ZIP con su APELLIDO y NOMBRE.

NOTA2: El ZIP debe incluir los fuentes C del proyecto realizados sobre *MCUXpresso* (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

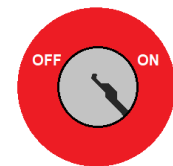
Se requiere implementar el control de una puerta automática de ascensor usando un sistema embebido. La puerta cuenta con 3 modos de operación: automático, ascensorista e incendio. En cada modo la puerta tendrá un comportamiento diferente, razón por la cual cada modo tendrá su propia máquina de estados. En este examen se le solicitará que gestione el cambio de modo de operación, pero que solo implemente el modo ascensorista.



### Modos de operación

Para cambiar el modo de operación se cuentan con dos llaves del tipo "On/Off" conectadas a dos entradas digitales de la placa base.

- Llave Incendio:
  - "1", opera en modo incendio.
  - "0", opera en modo ascensorista o automático según el estado de la otra llave.
- Llave Modo:
  - "1", opera en modo automático.
  - "0", opera en modo ascensorista.



**INCENDIO**



**MODO**

Importante: El cambio de modo se hace efectivo solamente cuando la puerta está completamente cerrada.

### Modo ascensorista:

1. Cuando el ascensor llega al piso destino (en ese momento la entrada digital asociada se pone en 0), se debe esperar a que el ascensorista oprima el botón de abrir puerta. Una vez que lo haga, la misma debe abrirse (La salida correspondiente se pondrá en 1 para abrir la puerta).
2. Si la puerta está completamente abierta (sensor de puerta abierta == 0) y el ascensorista presiona el botón de cerrar, la puerta debe cerrarse.
3. Si la puerta se está abriendo y el ascensorista presiona el botón de cerrar, la puerta debe detenerse, esperar 100 milisegundos y luego comenzar a cerrarse.
4. Si la barrera de seguridad es interrumpida mientras la puerta se está cerrando, entonces, la puerta debe detenerse inmediatamente, esperar 100 milisegundos y comenzar a abrirse nuevamente. La barrera de seguridad es un sensor conectado a una entrada digital. Si está en 0 indica que no hay ninguna persona entrando o saliendo, y si está en 1 indica que hay alguien bloqueando la puerta.

**Observaciones:** La barrera de seguridad y el botón de abrir puerta siempre tienen más prioridad que el botón de cerrar puerta.

### Reintentos y casos de falla:

- Si la puerta no finaliza la apertura total en 10 segundos, debe volver a cerrarse y hacer un reintento. Si nuevamente no finaliza la apertura en 10 segundos, se pondrá en modo de falla y permanecerá allí hasta que se reinicie el sistema.
- Si la puerta no finaliza el cierre total en 10 segundos, debe volver a abrirse y hacer un reintento. Si nuevamente no finaliza el cierre en 10 segundos, se pondrá en modo de falla y permanecerá allí hasta que se reinicie el sistema.

**Nota:** Por simplicidad, se recomienda que en estos dos casos no se consideren ni los botones de abrir y cerrar ni la barrera de seguridad. En un sistema real estarían considerados, pero complejiza el modelado y no tiene importancia a los efectos del examen.

**Para realizar el sistema cuenta con las siguientes constantes que definen cada entrada o salida para ser utilizadas por las funciones que se describen más abajo:**

Sensor / Actuador	Etiqueta (#define)	Entrada/Salida	Observaciones
Llego al piso destino	PISO	Entrada digital0	Entrada == 0 -> Llego al piso de destino
Sensor de puerta cerrada	P_CERRADA	Entrada digital1	Entrada == 0 -> Se cerró la puerta completamente
Sensor de puerta abierta	P_ABIERTA	Entrada digital2	Entrada == 0 -> Se abrió la puerta completamente
Botón de cerrar puerta	B_CERRAR	Entrada digital3	Entrada == 0 -> Botón presionado
Botón de abrir puerta	B_ABRIR	Entrada digital4	Entrada == 0 -> Botón presionado
Barrera de seguridad	BARRERA	Entrada digital5	Entrada == 0 -> la puerta está libre (nadie está pasando) Entrada == 1 -> la puerta esta obstruida
Modo Incendio	INDENDIO	Entrada digital 6	Entrada en 1 -> modo incendio
Modo Automático/ascensorista	AUT	Entrada digital 7	Entrada == 0 -> modo ascensorista Entrada == 1 -> modo automático
Abrir puerta	ABRIR	Salida digital 0	Salida == 1 -> Acciona el motor para abrir puerta
Cerrar puerta	CERRAR	Salida digital 1	Salida == 1 -> Acciona el motor para cerrar puerta
Detener puerta			Si AbrirPuerta = 0 y CerrarPuerta = 0, la puerta se detiene

#### **Funciones:**

//Pone la salida asignada en el primer argumento en el estado que se recibe en el segundo argumento.  
**void Salidas ( unsigned char nro\_SD , unsigned char estado );**

//Devuelve el estado de la entrada digital que se recibe como parámetro  
**unsigned char Entradas ( unsigned char nro\_ED );**

//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)**

//Inicializa el Hardware del sistema:

**void InicializarHW ( void );**

#### **Se pide:**

Modelizado: Entregable -> proyecto de uModel Factory.

- Utilizando el “uModel Factory”, modelice el sistema utilizando diagramas de estado. Se recomienda modelizar la gestión de los modos de operación con una máquina de estados y el comportamiento del modo ascensorista en otra.

Implementación: Entregable -> proyecto en MCUXpresso con resolución del examen que utiliza las funciones de acceso a hardware detalladas más arriba.

- Implemente la función main.

- Implemente el código para la gestión del cambio de estado.
- Implemente el código para el modo ascensorista