

Examen final

(Duración: 2 horas y 30 minutos)

Nombre: _____

Grupo: _____

INSTRUCCIONES

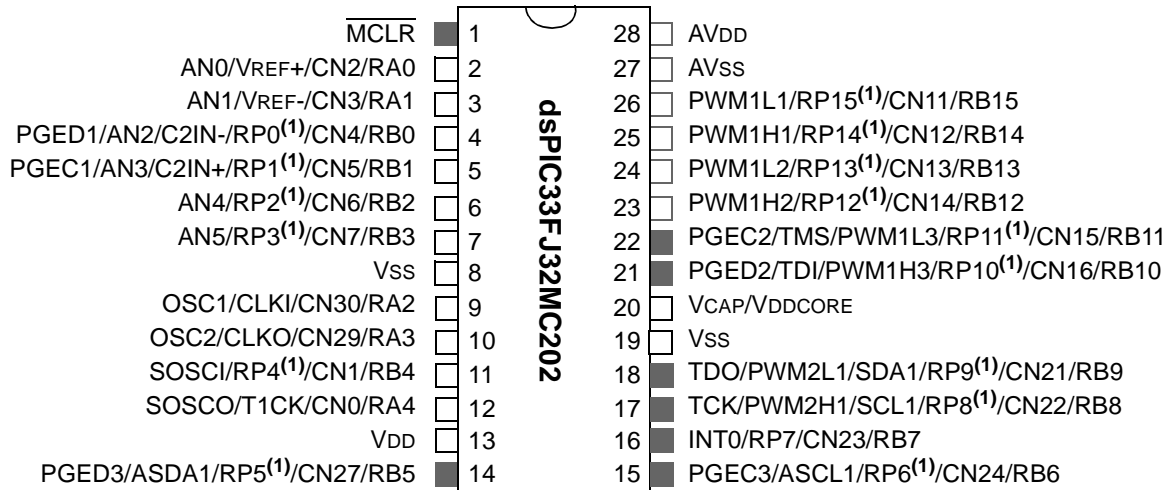
- No abra el cuadernillo hasta que se le indique.
- El examen consiste en un sistema electrónico que se ha dividido en dos problemas con un valor total de **diez puntos**.
- Se recomienda leer el enunciado completo antes de empezar a resolver el examen.
- Cada uno de los problemas deberá realizarse en una **hoja de solución diferente**.
- En la página 3 y siguientes encontrará un **formulario** que puede resultarle útil.
- Se permite el uso de **calculadora** y de una **hoja con anotaciones** sobre acondicionamiento de sensores.
- No desgrape el cuadernillo. Puede utilizar las últimas páginas como **borrador**.
- Si lo prefiere, puede resolver el examen a **lápiz**.
- Deberá emplear **máscaras** para leer o escribir los bits de los registros. La única excepción son los bits asociados a las interrupciones, con los que puede utilizar el acceso a nivel de bit.

Calificación:

FORMULARIO

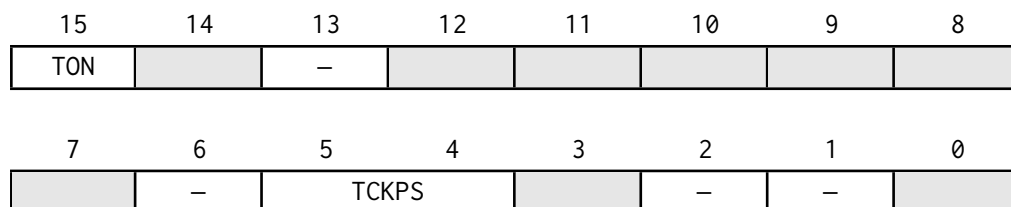
Microcontrolador dsPIC33FJ32MC202

Pines



Timers¹

- TMRx: Cuenta del timer
- PRx: Valor final de la cuenta
- TxCON: Registro de configuración
 - TON: Bit de encendido
 - 1 = Enciende el timer
 - 0 = Apaga el timer
 - TCKPS: Preescalado
 - 3 = 1:256
 - 2 = 1:64
 - 1 = 1:8
 - 0 = 1:1



- Función de atención a la interrupción

```
void __attribute__((interrupt, no_auto_psv)) _TxInterrupt(void);
```

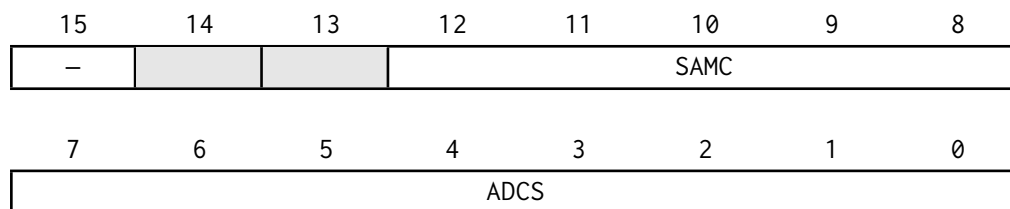
¹x indica el número de Timer (1, 2 ó 3).

Conversor analógico-digital

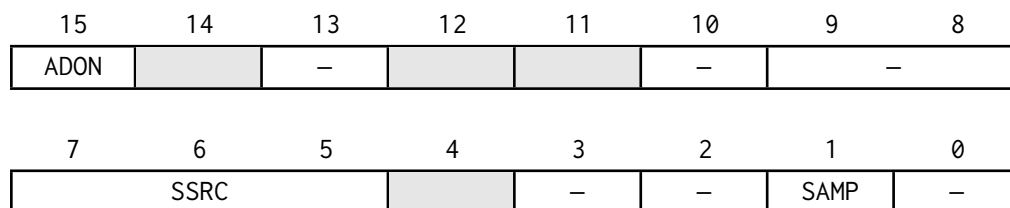
- AD1PCFGL: Registro para configurar los pines como analógicos o digitales
- AD1CHS0: Permite seleccionar el pin analógico que se va a convertir
- ADC1BUF0: Almacena el resultado de la última conversión realizada
- AD1CON3: Registro de configuración #3
 - ADCS: Preescalado
 - T_{AD} = Periodo del conversor A/D
 - TCY: Periodo del oscilador interno

$$T_{AD} = TCY \cdot (ADCS + 1)$$

- SAMC: Número de ciclos de muestreo



- AD1CON1: Registro de configuración #1
 - ADON: Arranca el periférico pero no empieza a muestrear
 - 1 = Enciende el módulo
 - 0 = Apaga el módulo
 - SSRC: Indica cuándo comienza la conversión
 - 7 = Automáticamente al terminar de muestrear
 - SAMP: Inicio del muestreo (se borra automáticamente)
 - 1 = Empezar a muestrear



- Función de atención a la interrupción

```
void __attribute__((interrupt, no_auto_psv)) _ADC1Interrupt(void);
```

Generador de PWM²

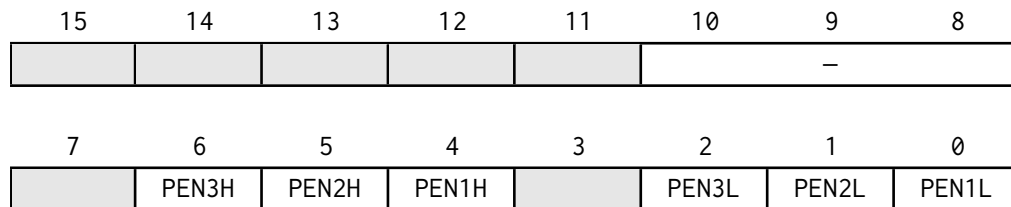
- PxTPER: Permite configurar el periodo de la señal (15 bits)

$$\text{PxTPER} = \text{Periodo} \cdot \text{FCY} \cdot \text{preescalado} - 1$$

- PxDCy: Factor de servicio de la señal del pin PWMxHy (16 bits)
 - Si es igual que PxTPER equivale a un 50 % de factor de servicio

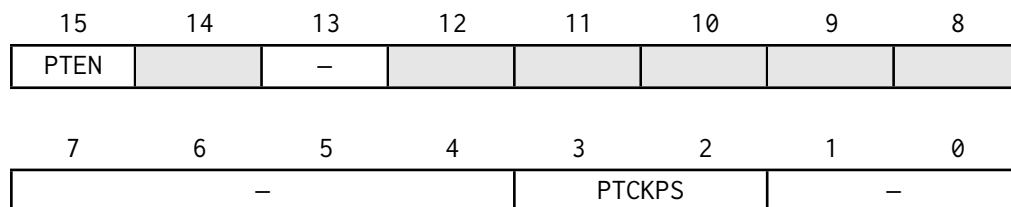
- PWMxCON1: Registro de control de puertos

- PENyH: Habilita el puerto PWMxHy
1 = Salida de PWM
0 = Entrada o salida digital
- PENyL: Habilita el puerto PWMxLy
1 = Salida de PWM
0 = Entrada o salida digital



- PxTCON: Registro de configuración

- PTEN: Bit de encendido
1 = Enciende el módulo
0 = Apaga el módulo
- PTCKPS: Preescalado para calcular la frecuencia de la señal
3 = 1:64
2 = 1:16
1 = 1:4
0 = 1:1



²x indica el número de generador de PWM (1 ó 2) e y el pin dentro de cada generador.

Registros de control de interrupciones

- IFS0: Banderas de interrupción

15	14	13	12	11	10	9	8
		AD1IF	-	-	-	-	T3IF
7	6	5	4	3	2	1	0
T2IF	-	-		T1IF	-	-	-

- IEC0: Registro para habilitar la atención de interrupciones

15	14	13	12	11	10	9	8
		AD1IE	-	-	-	-	T3IE
7	6	5	4	3	2	1	0
T2IE	-	-		T1IE	-	-	-

- IPCx: Prioridad de interrupciones

- IPC0

15	14	13	12	11	10	9	8
	T1IP					-	
7	6	5	4	3	2	1	0
	-					-	

- IPC1

15	14	13	12	11	10	9	8
	T2IP					-	
7	6	5	4	3	2	1	0
	-					-	

- IPC2

15	14	13	12	11	10	9	8
	-					-	
7	6	5	4	3	2	1	0
	-					T3IP	

- IPC3

15	14	13	12	11	10	9	8
						-	
7	6	5	4	3	2	1	0
	AD1IP					-	

Los productores de un famoso concurso televisivo de cocina están interesados en instalar un **sistema de extracción de humos y extinción de incendios** en el plató de cara a la próxima edición que se estrena esta Navidad. El objetivo es garantizar la calidad del aire y evitar que algún concursante manzanas calcine los estudios de la cadena en un descuido.

Especificaciones de *hardware*

Los sensores y actuadores del sistema (véase la Figura 1) se conectarán a un microcontrolador dsPIC33FJ32MC202 de la siguiente forma:

- Pines 2 y 3: Sensores de temperatura RTD (convenientemente acondicionados) para medir la temperatura en cada una de las dos zonas (A y B, respectivamente) en que se divide el plató.
- Pin 6: Un detector de humo que mida la obscuración³ en un rango del 0 al 2%.
- Pin 7: Un pulsador que permite reiniciar el sistema una vez sofocado un incendio grave al que sólo pueden acceder los bomberos.
- Pin 21: Un ventilador de corriente continua alimentado a 24 V para evacuar el humo.
- Pin 22: Una luz LED para indicar la salida de emergencia.
- Pin 23: Un altavoz para hacer sonar una señal de alarma en caso de incendio.
- Pines 24 y 25: Rociadores para pulverizar agua sobre las zonas A y B, respectivamente, cuando se detecte fuego. Se controlan a través de unos relés integrados en los dispositivos, de modo que su manejo será mediante señales digitales (se activan a nivel alto).

Especificaciones de *software*

- En ausencia de fuego, si la obscuración es inferior al 0,2%, el ventilador permanecerá apagado.
- A partir de ese umbral, el ventilador girará a una velocidad proporcional al valor medido. Por ejemplo, con una obscuración del 1,1% girará a la mitad de su velocidad máxima y con una obscuración del 2%, se moverá a máxima velocidad.
- Si la temperatura supera los 50°C en cualquiera de las zonas significará que se ha declarado un incendio y se activarán los rociadores de ese sector para intentar sofocarlo.
- Los rociadores son independientes, lo cual implica que aunque ya se haya disparado el rociador de una zona, si se supera el umbral en el otro sector, también deberá activarse su rociador.
- Si la temperatura desciende por debajo de 35°C antes de que transcurra un minuto, el fuego se considerará bajo control, se continuará refrescando durante otro minuto y después se apagará el rociador de la zona correspondiente.
- En caso contrario, se iniciará el protocolo de evacuación: se encenderán las luces de emergencia, arrancarán los rociadores de la otra zona para refrescarla (si no lo habían hecho anteriormente) y se hará sonar una alarma bitonal (un segundo a 1 kHz, un segundo a 200 Hz).
- Una vez declarado el estado de emergencia, sólo los bomberos podrán desactivar el sistema de extinción (rociadores, luz de emergencia y altavoz apagados) accionando el pulsador del pin 7.

³La obscuración es una unidad que indica la reducción de visibilidad. A mayor obscuración, más concentración de humo.

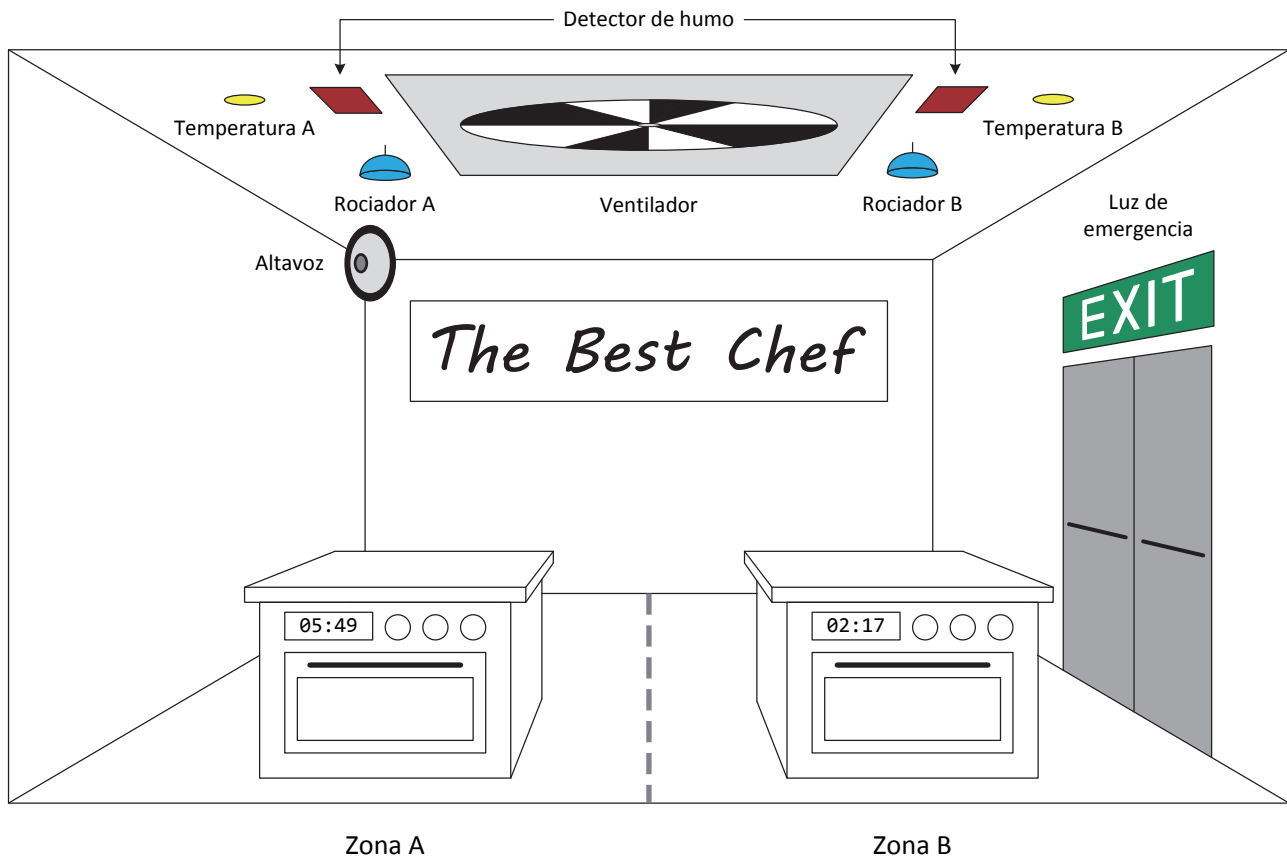


Figura 1. Diagrama del plato una vez instalado el sistema.

Problema 1. Acondicionamiento de sensores (4 puntos)

1.1. Sensor de temperatura RTD (2 puntos)

Para la medida de temperatura se han escogido sensores RTD del tipo Pt100 cuya curva de calibración es la siguiente:

$$R(T) = R_0 \cdot (1 + \alpha \cdot \Delta T), \quad \Delta T = T - T_0 \quad R_0 = 100 \, \Omega, \quad \alpha = 3,85 \cdot 10^{-3} \, ^\circ\text{C}^{-1}, \quad T_0 = 0 \, ^\circ\text{C}, \quad I_{\max} = 1 \, \text{mA}$$

- (1,5 puntos)** Diseñe un circuito de acondicionamiento que permita medir temperaturas entre 15°C y 65°C con la mayor sensibilidad posible. Puede emplear amplificadores operacionales, amplificadores de instrumentación de ganancia 10V/V, fuentes de tensión de valores razonables y las resistencias que necesite. Suponga que ambos tipos de amplificadores son ideales y pueden alimentarse con tensión simple (entre 0 y 3,3V) o simétrica (entre -3,3 y 3,3V); deberá indicar las tensiones de alimentación de cada amplificador que utilice.
- (0,5 puntos)** Justifique su solución en base a la característica del sensor.

1.2. Detector de humo (2 puntos)

Se ha decidido hacer un detector de humo por reflexión difusa. Este tipo de dispositivos consta de una fuente emisora de luz (normalmente un diodo LED infrarrojo) y de un fotorreceptor (por ejemplo, un fotodiodo) colocados dentro de una cámara opaca, que permite la entrada de humo, pero no de luz exterior. Además, el emisor y el receptor están separados por un cuerpo negro que impide que la luz del emisor llegue directamente al receptor. A medida que aumenta la concentración de humo en la cámara, la luz del emisor se empieza a reflejar en las partículas en suspensión haciendo que parte de ella llegue al receptor.

Como el plató es grande se quiere medir la concentración de humo en dos puntos y obtener una **salida proporcional a la media**, para lo cual se va a emplear el circuito de la Figura 2. Los fotodiodos P_1 , P_2 y P_3 son todos iguales, tienen una corriente de oscuridad de $0,5 \mu\text{A}$ y una sensibilidad de 3 nA/lux . Además, P_1 está situado en una cámara que garantiza oscuridad total, se sabe que $R_g = 1 \text{ M}\Omega$ y que 25 lux equivalen al 1% de obscuración. Puede suponer que los amplificadores operacionales son ideales y que sus tensiones de saturación coinciden con las de alimentación.

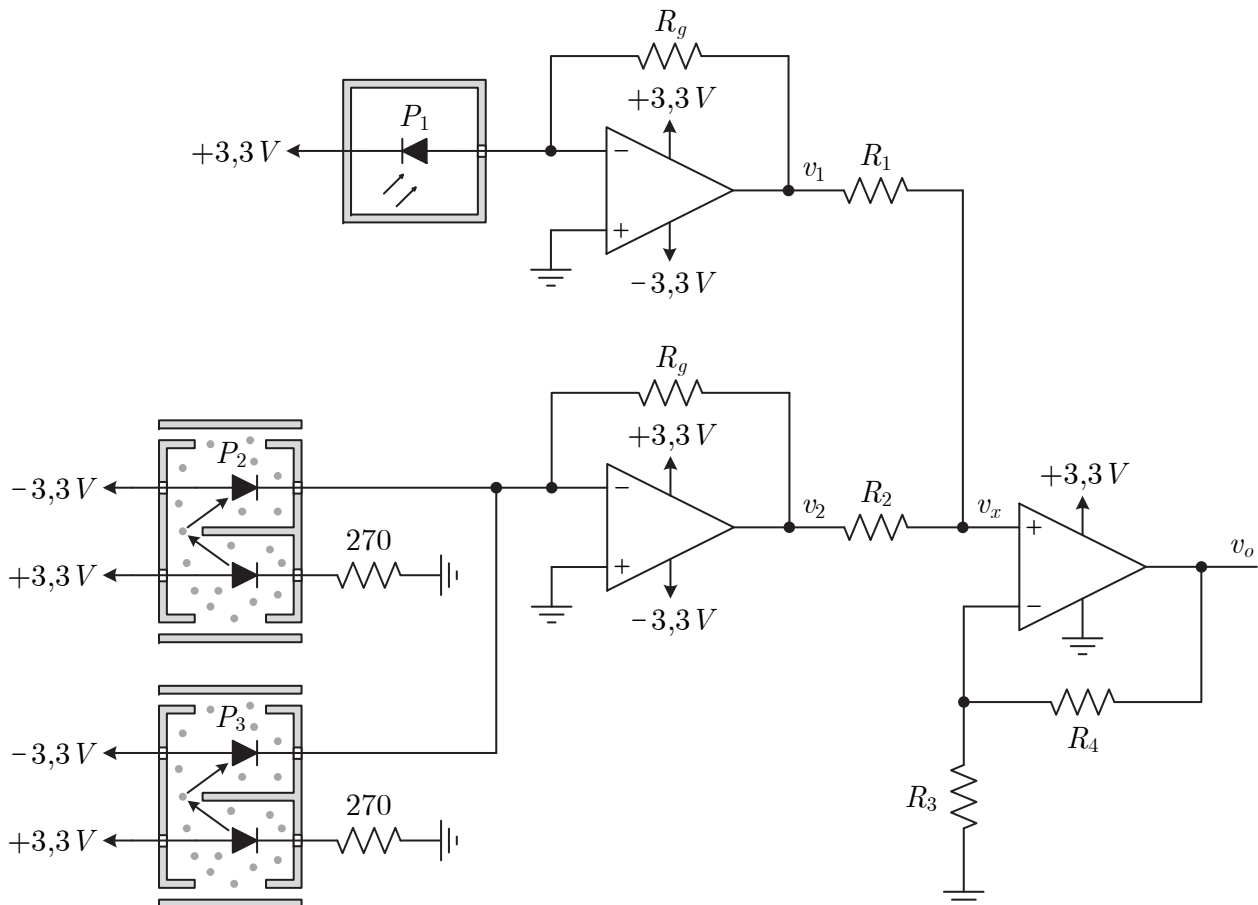


Figura 2. Detector de humo.

- (1 punto) Determine el valor de R_1 y R_2 para anular el efecto de la corriente de oscuridad.
- (1 punto) Calcule R_3 y R_4 para obtener la máxima sensibilidad posible a la salida en el rango del 0 al 2% de obscuración media.

Problema 2. Sistema digital (6 puntos)

- (0,5 puntos)** Dibuje el esquema del sistema electrónico siguiendo las especificaciones de *hardware*. Deberá incluir los circuitos de acondicionamiento de las entradas y salidas (sin especificar los valores de los componentes), así como su conexión con el microcontrolador. Como ya ha diseñado previamente los medidores de humo y temperatura, puede tratarlos como cajas negras, al igual que los rociadores, que vienen listos para usar de fábrica.
- (0,5 puntos)** Dado que necesita medir varias señales analógicas, programe un *driver* para realizar conversiones analógico-digitales a 500 ksps más fácilmente. No hace falta que permita gestionar todos los pines analógicos; basta con que pueda manejar los que necesita.
- (1 punto)** Describa brevemente cómo piensa resolver el problema: dibuje una máquina de estados, indique cuántos timers necesita, de qué periodo. . .
- (4 puntos)** Escriba un programa para que el sistema funcione según las especificaciones de *software*. Utilice el *driver* que se adjunta a continuación para controlar el ventilador.

motor.h

```
#ifndef _MOTOR_H
#define _MOTOR_H

#define PWM_L1 0 // Pin 26
#define PWM_L2 1 // Pin 24
#define PWM_L3 2 // Pin 22
#define PWM_H1 4 // Pin 25
#define PWM_H2 5 // Pin 23
#define PWM_H3 6 // Pin 21

/**
 * Configura un pin como salida de PWM.
 *
 * @param pin Emplear las constantes [PWM_L1, PWM_H3].
 */
void configurarPinMotor(unsigned char pin);

/**
 * Inicializa el generador de señales PWM a 24,743 kHz.
 */
void inicializarMotor(void);

/**
 * Establece el factor de servicio de una señal de PWM.
 *
 * @param pin Índice del pin PWM [1, 3].
 * @param dc Factor de servicio (%).
 */
void dcMotor(unsigned char pin, unsigned int dc);

#endif
```