

Máster en Mecatrónica

Computadores y programación

Examen Ordinario – Junio 2023

- 1) Realizar una función que recibe como parámetros dos cadenas de caracteres y devuelve un número entero. La primera cadena de caracteres tendrá el formato:

<<PARAM_A=valor_a>> <<PARAM_B= valor_b>> etc etc <<PARAM_X= valor_x>>

Donde *PARAM_?* puede ser cualquier texto, y *valor_?* es un número entero.

La función debe buscar en qué campo se encuentra la 2ª cadena pasada como parámetro y, si la encuentra, devolver el entero valor correspondiente. En caso de no encontrarla o que no esté en el formato adecuado, la función devolverá -1.

Puede haber espacios en blanco en los siguientes lugares de la 1ª cadena:

- Antes y después de "<<"
- Antes y después de ">>"
- Antes y después de "="

Ejemplo de llamada:

```
char cadena[]="<<ESTACION=3>> <<CANAL_AD=2>> <<PESO_DI=4>>";
int canal_ad, peso_di;
canal_ad=FnEjercicio1(cadena, "CANAL_AD"); // canal_ad pasa a valer 2
peso_di=FnEjercicio1(cadena, "PESO_DI"); // peso_di pasa a valer 4
```

- 2) Realizar una función que recibe como parámetros una tabla de reales que contiene datos temporales de caudal en litros/min y un valor real que indica el periodo de muestreo (en seg) en que han sido adquiridos, y devuelve un valor real que contiene:

- Un valor negativo si alguno de los valores de la tabla es menor que cero.
- El valor acumulado en litros si todos los valores de la tabla son mayores o iguales a cero

- 3) En una ciudad se dispone de varias estaciones meteorológicas, cada una de ellas equipada con sensor de lluvia que genera una señal analógica entre 20mA (no hay lluvia) y 0mA (lluvia máxima=5 litros/hora), y un sensor digital todo/nada que indica si la estación está operativa. Las señales de todas las estaciones son cableadas hacia un equipo con 8 entradas A/D 0-5V de 12 bits y una entrada digital DI de 8 bits (ver figura). El equipo dispone también de una salida digital DO de 8 bits.

Se debe realizar un programa que pida por teclado una cadena de caracteres con el formato del ejercicio 1, y a partir de ella determine el canal_ad y el peso_di a los que la estación de interés está conectada (usar función ejercicio 1).

A continuación, cada minuto (utilizar interrupción temporizada), el programa debe:

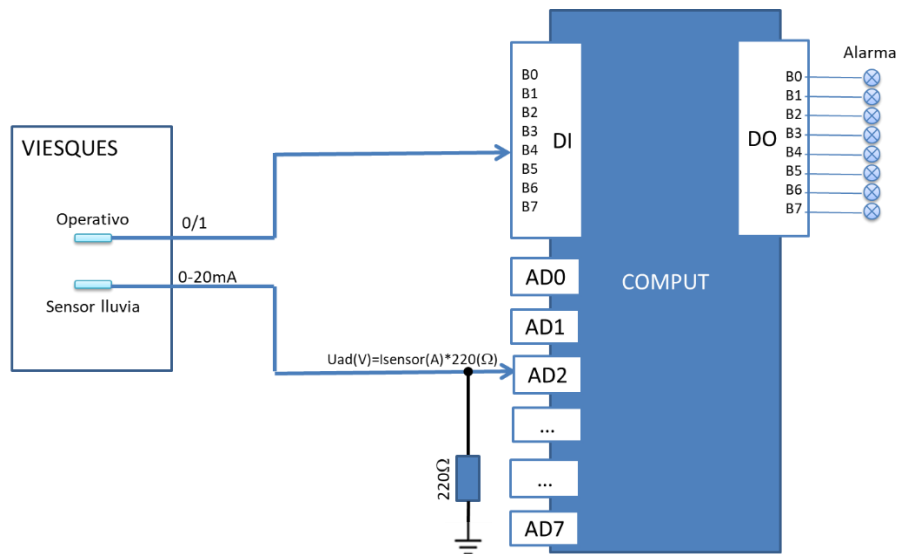
- Adquirir las señales de la estación deseada (medida analógica de caudal, y estado operativa/no). Si la estación no está operativa, el valor de caudal asignado a ese instante será -1.
- Calcular la lluvia acumulada (en litros) en la última hora. Si la estación está no operativa en algún momento de esa hora, el resultado no debe ser tenido en cuenta (usar función de ejercicio 2).
- Si la estación supera un valor máximo de lluvia acumulada en la última hora (valor inicial 1.5 litros), se generará una alarma activando en la salida DO el bit del peso correspondiente a la estación.

Máster en Mecatrónica

Computadores y programación

Examen Ordinario – Junio 2023

Durante la ejecución del programa, el operador deberá poder cambiar en cualquier momento (por teclado) el valor máximo de lluvia acumulada en la última hora, para activar una alarma.



Funciones y variables globales disponibles para E/S en “funciones_ES.h”:

```
void InitTemporizador(int T_ms,void (*FnISR)() );  
// Lanza una temporización por interrupción.  
// La función de interrupción debe declararse como: void MiFn();  
void StopTemporizador();  
// Desactiva la temporización.  
void Sleep(int tiempo_ms);  
// Retarda la ejecución durante el tiempo indicado  
int LeerCanalAD(int n_canal_ad);  
// Obtiene el valor de la conversión A/D del canal indicado (12 bits)  
  
int _di,_do; // Variables globales asociadas a los bits de E/S digital
```

Máster en Mecatrónica

Computadores y programación

Examen Ordinario – Junio 2023

1 punto

APELLIDOS Y NOMBRE: _____

Cuestiones: (0.25 ptos Acertada, -0.15 ptos Fallada, 0 ptos No Contestada)

a) ¿Cuál de estas afirmaciones es **falsa** respecto a una salida PWM ?

- El valor instantáneo varía automáticamente entre 0 y 1 a impulsos de un reloj
- El valor promedio depende de la relación entre T_{on} y T_{total} del ciclo
- Se debe amplificar para conseguir la tensión y corriente necesaria para el accionamiento al que se conecte
- Nunca puede utilizarse para LED

b) Si, en el ejercicio 3, la temporización se realizase bajo interrupción:

- Todas las variables deberían pasar a ser globales
- No se podría utilizar la función $FnEj1()$ tal y como está realizada
- Las dos anteriores son ciertas
- Ninguna de las anteriores es cierta

c) ¿Cuánto vale la variable j tras ejecutar el código siguiente? Justificar la respuesta.

```
#define N 8
int j;
for (j=0 ; j<N ; j++)
    ;
```

- j vale 8
- j vale 7
- No se puede usar j fuera del bucle
- No se puede compilar, el bucle for está mal construido

d) ¿Qué hace el código siguiente?

```
int main()
{
    char txt[40];
    printf("Introduzca texto: ");
    gets(txt);
    if (txt>="HOLA")
        printf("Saludo\n");
    return 0;
}
```

- Escribe en pantalla **Saludo** si el usuario introduce por teclado un texto que comience por la palabra **HOLA**
- Escribe en pantalla **Saludo** si el usuario introduce por teclado cualquier texto que contenga la palabra **HOLA**
- Escribe en pantalla **Saludo** si el usuario introduce por teclado cualquier texto que **no** contenga la palabra **HOLA**
- Ninguna de las anteriores

Máster en Mecatrónica

Computadores y programación

Examen Ordinario – Junio 2023

Algunas funciones y definiciones de las librerías estándar de C:

Algunas funciones de C:

```
int atoi(const char* cad); // Devuelve entero equivalente a cadena
double atof(const char* cad); // Devuelve real equivalente a cadena
double strtod(const char* cad,char** next); // Id. a atof() y guarda en next puntero
// a final de conversión
int strlen(const char* cadena); // Devuelve longitud de cadena
char* strcpy(char* dst,const char* src); // Copia cadena fuente en destino
char* strncpy(char* dst,const char* src,int n); // Id. Máximo n caracteres
char* strcat(char* dst,const char* src); // Concatena cadena Fuente a destino
char* strncat(char* dst,const char* src,int n); // Id. Máximo n caracteres
char* strchr(const char* cad,char c); // Busca caracter en cadena, devuelve puntero
// a la primera ocurrencia o NULL si no está
char* strstr(const char* cad,const char* busca); // Id. buscando cadena
char* gets(char* destino); // Lee cadena de consola, almacena en destino
```