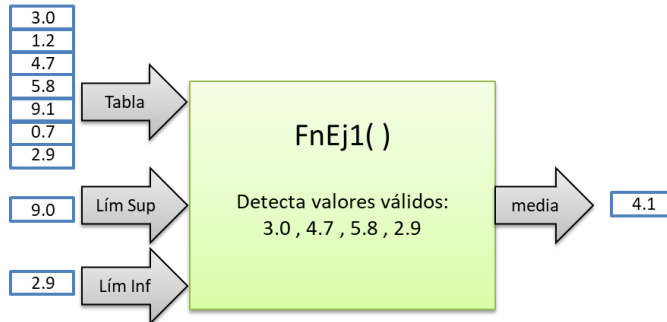


# Grado en Ing<sup>a</sup> Electrónica Industrial y Automática

## Informática Industrial y Comunicaciones

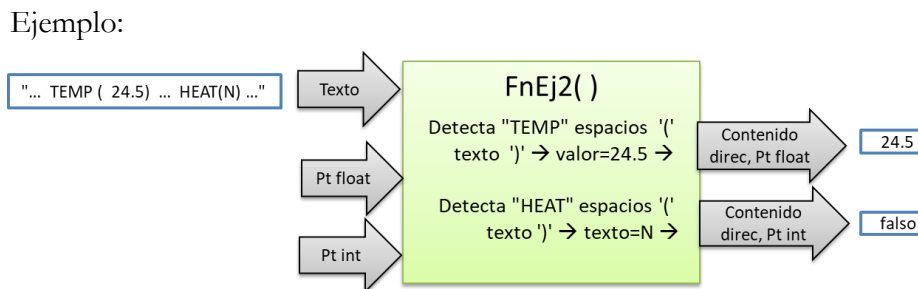
### Examen Ordinario – Julio 2023

- 1) Realizar una función que, dada una tabla de valores reales, y dos valores límite (inferior y superior) para los datos válidos en dicha tabla, devuelva el valor de la media de los elementos de la tabla que están entre dichos límites (ambos incluidos). Ejemplo:



- 2) Realizar una función que recibe como parámetro una cadena de caracteres, y devuelve dos valores **mediante parámetros tipo puntero**, en función del contenido de la cadena:

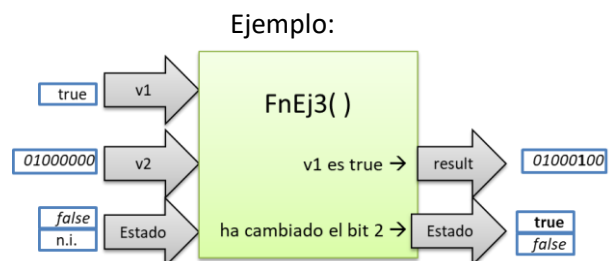
- Si la cadena contiene **en cualquier lugar** el texto *TEMP ( t\_obj )* (donde *t\_obj* es un número real, y puede haber espacios en blanco a ambos lados de cada paréntesis), se devuelve el valor *t\_obj* como float en el primero de los parámetros tipo puntero.
- Si la cadena contiene **en cualquier lugar** el texto *HEAT ( y\_o\_n )* (donde *y\_o\_n* debe ser el carácter Y ó el carácter N, y puede haber espacios en blanco a ambos lados de cada paréntesis), se devuelve true (si es Y)/false (si no es Y) como entero en el segundo de los parámetros tipo puntero.
- Si la cadena no contiene alguno de los textos, o el formato de éste es incorrecto, el valor correspondiente no se modifica.



- 3) Realizar una función que, dados dos valores enteros, devuelve el segundo de ellos modificado según las condiciones siguientes:

int FnCompara(int v1,int v2,int estado[2]);

- Si *v1* es false, devuelve el valor *v2* pero con su bit de peso 2 puesto a 0.
- Si *v1* es true, devuelve el valor *v2* pero con su bit de peso 2 tomando el valor de la operación lógica :  $v2_{bit6} \text{ OR } v2_{bit4}$
- Si, por las razones anteriores, el bit de peso 2 cambia de valor, se desplaza la tabla *estado[ ]* y se añade en el 1<sup>er</sup> elemento el nuevo valor de dicho bit (true/false).

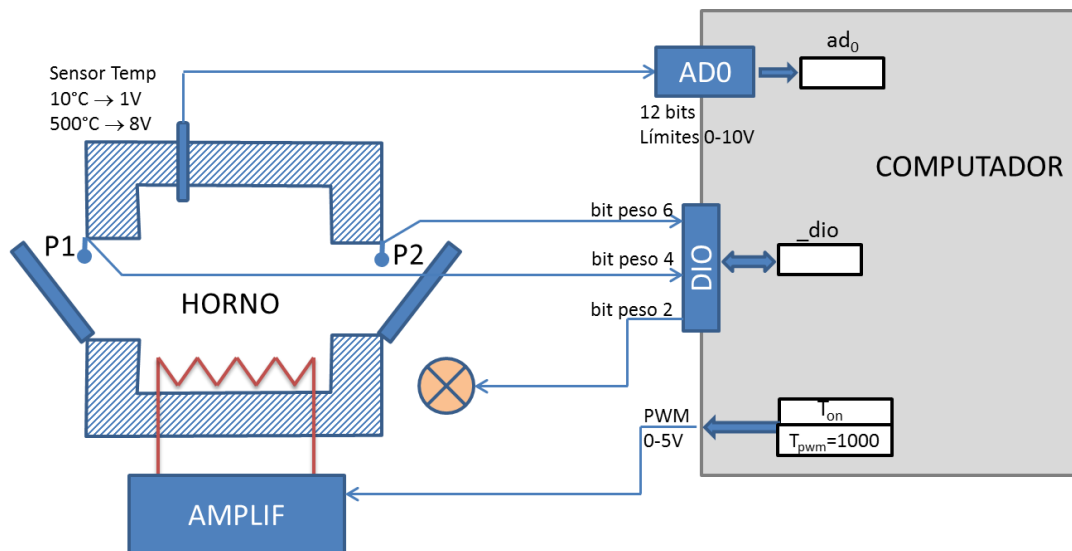


# Grado en Ing<sup>a</sup> Electrónica Industrial y Automática

## Informática Industrial y Comunicaciones

### Examen Ordinario – Julio 2023

- 4) Se desea controlar en tiempo real la temperatura de un horno de doble puerta, mediante un sistema como el mostrado en la figura:  
5 puntos



Las condiciones de funcionamiento son las siguientes:

- ❖ El horno se calienta mediante una resistencia, a la que se aplica una tensión controlable a través de una señal PWM 0-5V. Sólo se debe calentar cuando ambas puertas del horno (P1 y P2) estén cerradas (valor 1 en los bits correspondientes de la variable global `_dio`).
- ❖ El usuario puede fijar una temperatura objetivo en cualquier momento mediante la introducción por teclado de un texto con el formato:

`TEMP ( t_obj ) ; HEAT ( y_n )`

donde `t_obj` es un valor real en °C, e `y_n` puede tomar los valores Y (yes) o N (no).

Inicialmente, la temperatura objetivo es 0°C, y calentamiento está en modo N (no).

- ❖ Se mide la temperatura **cada 2 seg** mediante un sensor de temperatura y un convertor A/D (ver características de ambos en la figura). Se considera una temperatura válida si está entre 10 y 500 °C. Si la media `Tavg` de las temperaturas válidas **del último minuto** (que estén entre 10°C y 500°C, usar función ejercicio 1) es mayor que la temperatura objetivo `Tobj`, y además la última orden de teclado indicaba `HEAT=Y`, y además las puertas P1 y P2 están cerradas, se procederá al calentamiento aplicando una señal PWM cuyo valor será:

$$\text{PWM(\%)} = \mathbf{b_0} * ( Tobj - Tavg )$$

En caso de no cumplirse las condiciones anteriores, se aplicará una PWM de 0.

Si la temperatura medida supera 50°C y además hay alguna puerta abierta, se genera un aviso activando el bit de peso 2 de la variable `_dio`.

# Grado en Ing<sup>a</sup> Electrónica Industrial y Automática

## Informática Industrial y Comunicaciones

### Examen Ordinario – Julio 2023

---

La organización del programa será la siguiente:

- ❖ En el programa principal, se solicita al usuario el nombre de un archivo de texto, y se van leyendo líneas del mismo hasta que una contenga el texto:

$b_0 = \text{número\_real}$

Una vez encontrada, se asigna ese valor real a la constante  $b_0$  y se finaliza la lectura del archivo.

A continuación, se inicializan el resto de variables y tablas necesarias, y se arranca el temporizador ( $T=10$  seg).

Finalmente, el programa principal queda en un bucle esperando la introducción de órdenes por teclado y reaccionando según su contenido (función ejercicio 2).

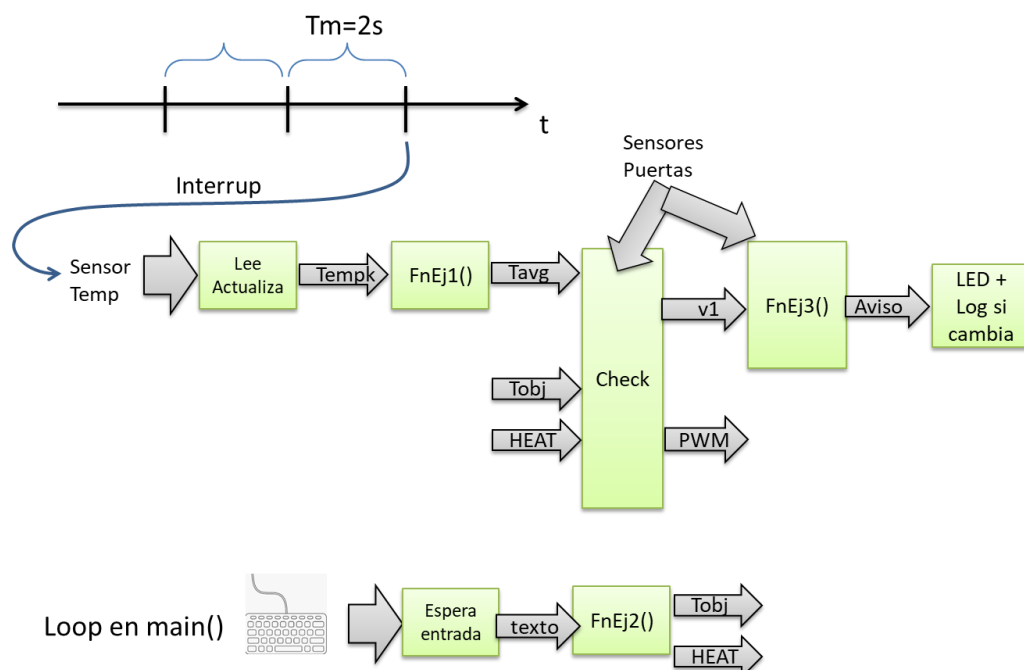
- ❖ En la función de interrupción (cada 2 seg) se adquiere un nuevo valor de temperatura, se actualiza(n) la(s) tabla(s) necesaria(s), se calcula la media de las últimas temperaturas válidas (función ejercicio 1), se comprueba si se cumplen las condiciones para calentar, se genera la PWM adecuada y se activa/apaga la señal de aviso según se cumplan sus condiciones (función ejercicio 3, donde  $v_1 \equiv \text{temp supera los } 50^\circ\text{C}$ ).

Si el estado de alarma ha cambiado, se añade un texto indicativo a un archivo "log.txt":

30/05/2023 15:30 → Comienza alarma, temp media 75°C y puerta abierta

30/05/2023 16:20 → Fin alarma, temp 72°C y puertas cerradas

En la librería auxiliar (ver a continuación) se dispone de una función para escribir fecha y hora, además de otras utilidades.



# Grado en Ing<sup>a</sup> Electrónica Industrial y Automática

## Informática Industrial y Comunicaciones

### Examen Ordinario – Julio 2023

---

#### Librería auxiliar ya disponible:

```
int LeerCanalAD(int n_canal); // Obtiene valor de canal A/D, 12 bits 0/10V
void InitTemporizador(int tm_ms,void (*FnISR)()); // Arranca temporizador con interrup.
int _dio; // Variable global conectada a la E/S digital
void FechayHora(FILE* fid); // Escribe en fecha y hora actual como texto en archivo
float ProductoEscalar(const float* v1,const float* v2,int n); // Calcula prod. escalar
void InitTabla(float* t,int n); // Inicializa valores de una tabla a 0
void DesplazaTabla(float* t,int n); // Desplaza los valores de una tabla
```

#### 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
int strcmp(const char* c1,const char* c2); // Compara cadenas, devuelve 0 si iguales
int strncmp(const char* c1,const char* c2,int n); // Compara cadenas hasta max n
// caract, devuelve 0 si iguales
char* gets(char* destino); // Lee cadena de consola, almacena en destino
void* malloc(int n_bytes); // Asigna memoria para n bytes
void free(void* ptr); // Libera memoria asignada
FILE* fopen(const char* nombre,const char* modo); // Abre stream
char* fgets(char* dest,int n_max,FILE* fid); // Lee línea de stream de texto
int fscanf(FILE* fid,...); // Lee datos de stream de texto con formato
int fprintf(FILE* fid,...); // Escribe datos en stream de texto con formato
void fclose(FILE* fid); // Cierra stream
```

# Grado en Ing<sup>a</sup> Electrónica Industrial y Automática

## Informática Industrial y Comunicaciones

Examen Ordinario – Julio 2023

---

APELLIDOS Y NOMBRE: \_\_\_\_\_

**Cuestiones** (responder aquí y entregar esta hoja con nombre y apellido):

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

- a) ¿Qué contiene el archivo .pro en un proyecto de Qt-Creator ?
- Un texto que indica los archivos envueltos en la compilación del programa y las opciones de compilación y enlazado.
  - Las direcciones de red donde se encuentran los archivos de encabezado .h compartidos con otros proyectos.
  - Las funciones de C compartidas con otros proyectos.
  - El código binario del ejecutable de la aplicación, excepto las librerías de enlace dinámico.
- b) En una comunicación serie, cuando quiere usarse para equipos alimentados a  $\pm 12V$ , 5V ó 3.3V :
- El estándar indica que el emisor utilice en la línea TxD los niveles de tensión 0V = 0 lógico, 5V = 1 lógico , que deben interpretados por el receptor en la línea RxD.
  - El estándar indica que el emisor utiliza en la línea TxD los niveles de tensión según su alimentación, y el receptor se debe adecuar a la interpretación de los mismos en su línea RxD.
  - El estándar indica que emisor y receptor usen diferentes líneas TxD y RxD para cada nivel de tensión.
  - El usuario debe asegurarse que los niveles de tensión TxD del emisor y RxD del receptor son iguales, no hay un estándar único que los defina.
- c) Un servidor DNS y un servidor DCHP en una red :
- Son lo mismo (siglas en inglés / en español)
  - Ambos enrutan los mensajes hacia el servidor Internet, pero con diferentes protocolos (TCP/UDP).
  - Tienen tareas distintas, pero ambas relacionadas con la gestión de direcciones IP.
  - Ambos sirven el contenido de las páginas web, uno de forma segura y el otro sin seguridad.

d) ¿Qué hace el código siguiente?

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

- Escribe **Saludo** si el usuario introduce por teclado un texto que **comience** por la palabra **HOLA**
- Escribe **Saludo** si el usuario introduce por teclado cualquier texto que **contenga** la palabra **HOLA**
- Escribe **Saludo** si el usuario introduce por teclado cualquier texto **más largo** que la palabra **HOLA**
- Ninguna de las anteriores