

Grado en Ingeniería Electrónica y Automática

Informática Industrial y Comunicaciones

Examen Ordinario – Enero 2020

- 1) Realizar una función que, dados como parámetros un dato medido (número real) y tres tablas (del mismo tamaño) de reales (mínimo medición, máximo medición y valor nominal), devuelva el valor nominal para el cual se cumple que la medición está entre el mínimo y máximo, o cero si no encuentra ninguna.
- 1.5 puntos

Ejemplo:

MEDIDAS MONEDAS EURO

Diam min (mm)	Diam max (mm)	Valor nominal €
21.20	21.30	0.05
19.70	19.80	0.10
22.20	22.30	0.20
24.20	24.30	0.50
23.20	23.30	1.00
25.70	25.80	2.00

EJEMPLOS:

FnEjercicio1(22.24, tablas de datos) → 0.20

FnEjercicio1(24.05, tablas de datos) → 0

- 2) Realizar una función que, dados como parámetros el nombre de un archivo de texto, un número entero y la dirección de una cadena de caracteres, busque en el archivo la línea que comienza por dicho número entero y obtenga de la misma el valor real y el texto que le acompañan.
- 1.5 puntos

Cada línea del archivo está formada por 3 valores separados por coma y espacio(s)

- Un valor entero
- Un valor real
- Un texto descriptivo

Ejemplo:

```
datos.txt
1 , 1.10 , BEBIDA1
17 , 0.60 , GOLOSINA2
25 , 1.25 , PATATAS
```

```
char articulo[20];
float valor;
```

```
valor=FuncionEj2("datos.txt",17,articulo); // valor=0.60, articulo → "GOLOSINA2"
```

La función devolverá valor cero y texto vacío si no encuentra el número entero

- 3) Realizar una función que, dados como parámetros un índice entero, una tabla de bytes y un carácter **A** (activar) o **D** (desactivar), active o desactive en dicha tabla el bit B correspondiente al índice, según la interpretación siguiente:
- 1 punto



El resto de bits de la tabla tienen que mantener su valor anterior.

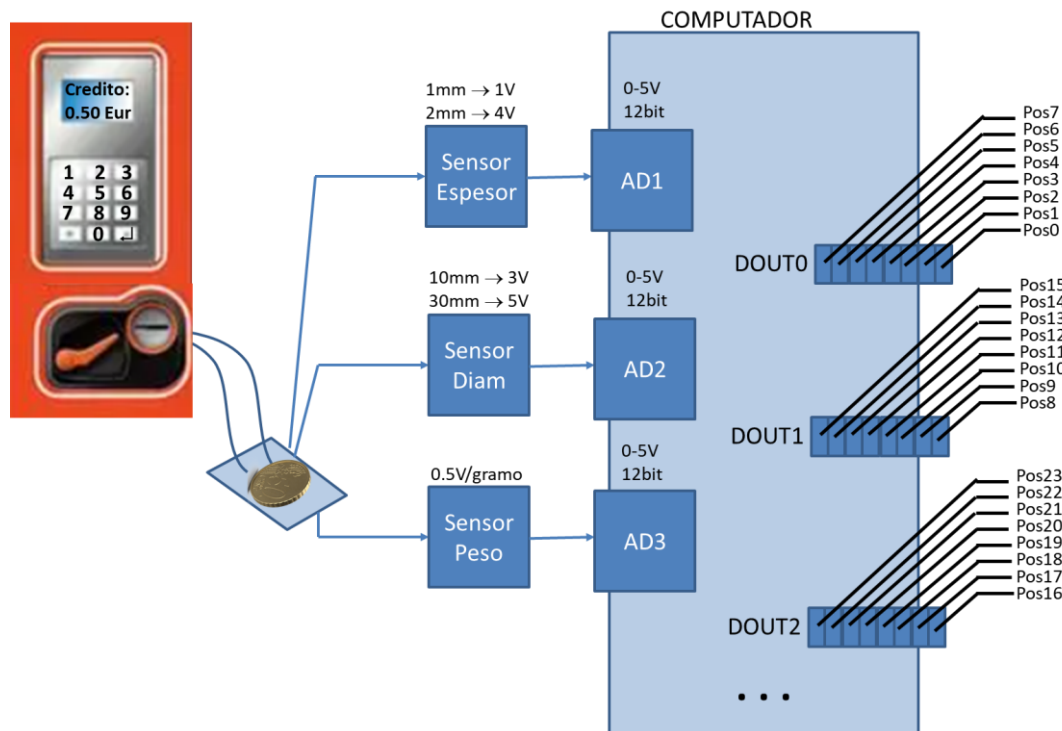
Grado en Ingeniería Electrónica y Automática

Informática Industrial y Comunicaciones

Examen Ordinario – Enero 2020

4) Se desea realizar un sistema de control para la venta automática utilizando monedas.

5 puntos



Como interfaz con el usuario, la máquina dispone de un teclado básico para la selección del artículo, y de un display LCD para la visualización.

Las monedas son detectadas por un sistema que utiliza tres sensores analógicos: espesor, diámetro y peso. Para detectar cada moneda se toman medidas de los tres sensores de forma permanente con periodo 10ms, y se obtiene el promedio de las 5 últimas medidas. Si los valores medios de espesor, diámetro y peso se corresponden con la misma moneda, se considera que es válida.

Cada vez que se detecta una moneda válida, se añade su valor al crédito total, se escribe el crédito en el display LCD, y se vacían las tablas de medidas para evitar detectar de nuevo la misma moneda.

Cuando el cliente pulsa una cadena de selección seguida de INTRO, se comprueba que el importe acumulado es mayor o igual al importe necesario (utilizando la función del ejercicio 2) y, en caso afirmativo, se abre durante 1 seg el dispensador correspondiente. Se utilizará la función del ejercicio 3 para determinar qué dispensador se debe abrir.

Organización del programa:

- ❑ El programa principal realiza las inicializaciones necesarias (valores y medidas de monedas usando función `GetValoresMonedasEuro()` del anexo, tablas temporales, etc.) y lanza una interrupción temporizada cada 10 ms. A continuación, realiza un bucle infinito en que espera la selección (índice) del usuario como un valor entero de teclado (`scanf`); cada vez que se introduce el índice, se determina el nombre e importe del artículo asociado utilizando la función del ejercicio 2 y, si el crédito disponible es suficiente:

Grado en Ingeniería Electrónica y Automática

Informática Industrial y Comunicaciones

Examen Ordinario – Enero 2020

- Se escribe en el display los datos del artículo seleccionado:



```
SELECCION: 25
PATATAS / 1.25 EUR
```

- Se evita que la función de interrupción siga acumulando crédito
 - Se ordena la activación del dispensador correspondiente durante 1 seg (usar Sleep). Se utiliza la función del ejercicio 3 para establecer el contenido de la variable global `_bytes_do`.
 - Se reduce el importe del artículo del crédito disponible
 - Se vuelve a permitir que la función de interrupción siga acumulando crédito
- En la interrupción temporizada se realizarán, por este orden:
- Comprobar que está permitida la acumulación de crédito. En caso afirmativo:
 - Desplazamiento de las tablas utilizadas para valores temporales de medidas (peso, espesor y diámetro).
 - Leer canales AD 1, 2 y 3 para obtener medidas actuales, añadir como datos más recientes de sus tablas y calcular media de últimos 5 datos.
 - Llamar a función ejercicio 1 para comprobar si hay moneda y cuál es su valor.
 - Si hay moneda nueva:
 - Añadir su importe al crédito
 - Borrar tablas de medidas de evolución temporal
 - Escribir en display LCD el saldo actual



```
CREDITO:
1.50 EUR
```

Grado en Ingeniería Electrónica y Automática

Informática Industrial y Comunicaciones

Examen Ordinario – Enero 2020

ANEXO

Librería auxiliar:

Se dispone de las siguientes funciones y declaraciones dentro de una librería FnAux (archivos FnAux.h y FnAux.lib):

```
// Inicialización de un temporizador que llame a la función indicada en
// FnCallback cada cierto intervalo (en ms)
void InitTemporizador(int tiempo_ms,void (*FnCallback)());

// Obtención del valor de conversión A/D del canal deseado utilizando
// nBits : resultado de 0 a 2nBits-1, para tensión de entrada 0 a 5V.
int ConversionAD(int nCanal,int nBits);

// Estructura para los valores mínimo y máximo de medidas de monedas
struct monedas
{
    int n_monedas;
    float *valores;
    float *diams_min_mm,*diams_max_mm;
    float *espesores_min_mm,*espesores_max_mm;
    float *pesos_min_g,*pesos_max_g;
};

// Función que asigna memoria para las tablas y rellena los datos de las
// monedas de euro (NO HAY QUE HACERLA, YA EXISTE)
struct monedas GetValoresMonedasEuro();

// Función para escribir en display LCD
void LCD_PutString(int fila,const char* texto);

// Variable global que almacena los contenidos de las salidas digitales
char* _bytes_do[5]; // Permite hasta 40 salidas diferentes
```

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
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 Ingeniería Electrónica y Automática

Informática Industrial y Comunicaciones

Examen Ordinario – Enero 2020

APELLIDOS Y NOMBRE: _____

Ejercicio 4 (responder aquí):

1 punto

a) ¿Cuánto vale la variable x tras ejecutar el código siguiente?

```
int Suma(const int* t,int n);          /* Suma los valores de una tabla
...                                  de n enteros */
int datos[4]={0,2,4,6};
int x;
x=Suma(datos+1,1);
```

- x vale 12 si la función Suma() está correctamente realizada
- x vale 1 si la función Suma() está correctamente realizada
- x vale 3 si la función Suma() está correctamente realizada
- x vale 2 si la función Suma() está correctamente realizada

b) ¿Cuánto vale la variable x tras ejecutar el código siguiente?

```
void Suma(const int* t,int n,float result); /* Suma los valores de una tabla
...                                       de n enteros */
int datos[4]={0,2,4,6};
float x ;
Suma(datos,4,x);
```

- x vale 12 si la función Suma() está correctamente realizada
- No se puede compilar, la tabla debería ser de float
- No se puede saber el valor de x , independientemente de lo que haga Suma()
- Debería declararse `float* x` y pasarle a la función `*x`. En ese caso, x vale 12 si la función Suma() está correctamente realizada.

c) ¿Cómo se sabe si el bit de peso P de la variable entera V está activo?

- `if (v &= 1<<p)`
- `if (p[v] << 1)`
- `if (v && (1 <<p))`
- `if (v & (1 <<p))`

d) ¿Cuántos equipos se pueden comunicar a través de un enlace serie RS-232 ?

- 2
- Infinitos
- 2^{32}
- $2^{32}-1$