

Grado en Ingeniería Electrónica y Automática

Informática Industrial y Comunicaciones

Examen Ordinario – Junio/Julio 2021

Ejercicio 1): Realizar una función que, dada una tabla de valores reales, un valor real de comparación, 2 puntos y dos valores enteros $p1$ y $p2$, calcule el mínimo y el máximo de los datos de la tabla y devuelva:

- Un valor entero con todos sus bits a 0 si tanto el mínimo como el máximo se encuentran en el intervalo alrededor del 1% del valor de comparación.

Si no se cumple lo anterior:

- Un valor entero con únicamente el bit de peso $p1$ a 1, si el mínimo y el máximo se encuentran en el intervalo alrededor del 10% del valor de comparación.

Si no se cumple lo anterior:

- Un valor entero con únicamente el bit de peso $p2$ a 1.

Ejemplos: ($cmp=10.0 \rightarrow$ Intervalo 1% = [9.9 a 10.1] , intervalo 10% = [9.0 a 11.0])

$t = [9.97, 10.02, 10.01]$, $cmp=10.0$, $p1=2$, $p2=5$ \rightarrow Devuelve binario 00...000000

$t = [9.97, 10.12, 10.01]$, $cmp=10.0$, $p1=2$, $p2=5$ \rightarrow Devuelve binario 00...000100

$t = [9.97, 11.12, 10.01]$, $cmp=10.0$, $p1=2$, $p2=5$ \rightarrow Devuelve binario 00...100000

Ejercicio 2): Realizar una función que, dada una cadena de caracteres y un valor entero, devuelva el 2 puntos número real que se encuentra dentro de la cadena en el ítem indicado por el valor entero (comenzando la cuenta en 0). Los ítems de la cadena de caracteres estarán separados por el carácter ':'. La función devolverá 0 si no hay un número real en la posición indicada, o no hay dicha posición en la cadena.

Ejemplos:

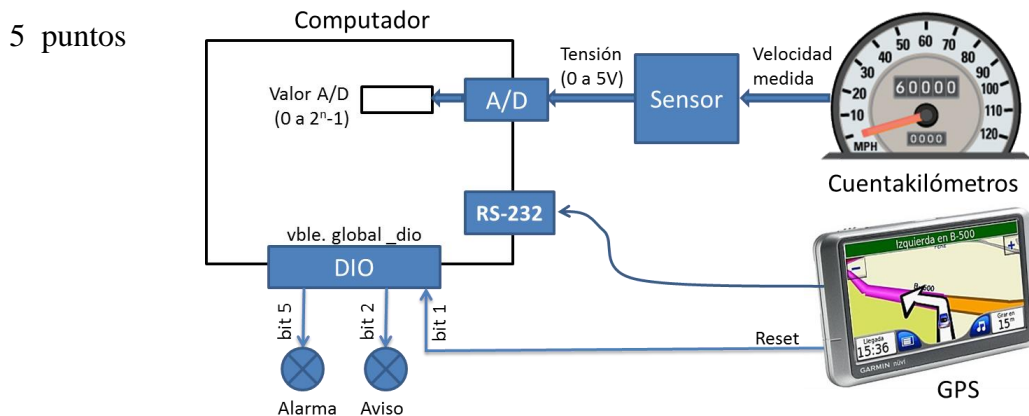
$MiFuncionEj2("3.5 : 4.7 : a.2 : 5.4 : 6.3",0)$; \rightarrow Devuelve el nº real 3.5

$MiFuncionEj1("3.5 : 4.7 : a.2 : 5.4 : 6.3",3)$; \rightarrow Devuelve el nº real 5.4

$MiFuncionEj2("3.5 : 4.7 : a.2 : 5.4 : 6.3",2)$; \rightarrow Devuelve el nº real 0.0

$MiFuncionEj2("3.5 : 4.7 : a.2 : 5.4 : 6.3",5)$; \rightarrow Devuelve el nº real 0.0

Ejercicio 3): Se desea realizar un sistema de control para comprobar la exactitud de la medición de 5 puntos distancia del cuentakilómetros de un vehículo. Para ello, se dispone del montaje siguiente:



Grado en Ingeniería Electrónica y Automática

Informática Industrial y Comunicaciones

Examen Ordinario – Junio/Julio 2021

La velocidad medida por el cuentakilómetros es captada por un sensor que entrega una tensión entre 1V (para 0Km/h) y 4V (para 180Km/h), y adquirida con un conversor A/D 0-5V de 12 bits. Se obtendrá la posición del cuentakilómetros por integración de esta medida.

El GPS está conectado por RS-232 al computador, en el que aparece como un dispositivo que se debe abrir y leer como un archivo de texto de nombre “/dev/ttyUSB0”; en dicho archivo, cada segundo se recibirá una cadena de caracteres de longitud máxima 50, con el siguiente formato:

`campo0 : campo1 : campo2 : campo3 : campo4 : campo5 \n`

Dónde:

`campo4` es la posición en Km desde el comienzo del viaje (ej: 75.2).

y el nº de caracteres de cada campo puede variar.

La comprobación de funcionamiento se realizará de la siguiente manera:

- ❑ Se solicitará por teclado al usuario un número entero indicando el número de valores de posición (medida a través del cuenta-kilómetros) a utilizar en los cálculos (n_pos).
- ❑ Se asignará memoria dinámica para una tabla de n_pos elementos que albergue las medidas de posición más recientes.
- ❑ Tras las inicializaciones: abrir archivo/dispositivo GPS, inicializar tablas temporales, etc., la obtención de datos del GPS se realiza mediante un bucle llamando a la función `fgets()` en `main()`. Una vez obtenida la cadena, se debe extraer la posición en Km a partir del campo 4 usando la función del ejercicio 2.
- ❑ En una interrupción temporizada, cada 200 ms:
 - Se comprueba el bit de peso 1 de `_dio`. Cuando el usuario pone a cero el contador de distancia del GPS se activa el bit de peso 1 de `_dio`. En ese instante, se debe reiniciar la medida de posición a cero, así como la cuenta de tiempo en segundos; sólo si ha habido esta activación en algún momento, se realizan los pasos siguientes de comprobación:
 - Se adquiere la velocidad actual del cuentakilómetros (en Km/h) a través de la lectura del canal A/D.
 - Se integra la velocidad para obtener la posición actual por el cuentakilómetros, mediante integración trapezoidal. Se añade esta posición a la tabla de valores.
 - Llamando a la función del ejercicio 1, si el mínimo o máximo de las n_pos últimas posiciones medidas están afuera del intervalo $MedidaGPS \pm 10\%$ se activará el bit de alarma en la variable global `_dio`; si están afuera del intervalo $MedidaGPS \pm 1\%$ (función ejercicio 1) se activará el bit de aviso en la variable global `_dio`; si ninguna de las anteriores se cumple, se borrarán ambos bits de `_dio`. El resto de bits de `_dio` no debe modificarse en ningún caso.
 - Se añade a un archivo de texto una línea con la información siguiente:
`t_seg , pos_min , pos_max , pos_GPS , AVISO ON/OFF, ALARMA ON/OFF`

Grado en Ingeniería Electrónica y Automática

Informática Industrial y Comunicaciones

Examen Ordinario – Junio/Julio 2021

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);

// Variable global que almacena los contenidos de las salidas digitales
int _dio;
```

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 – Junio/Julio 2021

APELLIDOS Y NOMBRE: _____

Ejercicio 4 (responder aquí):

1 punto (0.25 pto cuestión acertada, -0.1 pto cuestión fallada, 0 pto cuestión no contestada)

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 1 si la función Suma() está correctamente realizada
- x vale 12 si la función Suma() está correctamente realizada
- El valor de x depende del valor del puntero *datos*, es indeterminado
- x vale 2 si la función Suma() está correctamente realizada

b) ¿Qué ocurre con el código siguiente para convertir los valores de una tabla en sus cuadrados?

```
void Cuadrados(const int* t,int n)
{
    int i;
    for (i=0;i<n;i++)
        t[i] *= t[i];
}
...
int datos[4]={0,2,4,6};
Cuadrados(datos[],4);
```

- No se puede compilar, la declaración del puntero t no puede ser const
- No se puede compilar, está mal la llamada con el formato Cuadrados(datos[],4);
- Las dos anteriores son ciertas
- Ninguna de las anteriores es cierta

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}
- $2^{32}-1$