

Grado en Ing^a Electrónica Industrial y Automática

Informática Industrial y Comunicaciones

Examen Ordinario – Enero 2022

- 1) Realizar una función que, dados como parámetros los valores reales (dL y dR) medidos por dos sensores, una tabla de n valores reales, y un valor mínimo, realice los siguientes cálculos:
- Desplazar 1 posición hacia abajo los valores de la tabla
 - Añadir al principio de la tabla la diferencia dL-dR
 - Si alguno de los valores (dL ó dR) es inferior al valor mínimo pasado como parámetro, devolver éste último multiplicado por 1 millón (1.0e6 en C)
 - Si no ocurre lo anterior, devolver la media ponderada de los valores de la tabla:

1.5 puntos

$$mp = \frac{\sum_i (n - i) * t[i]}{\sum_i (n - i)}$$

- 2) Realizar una función que, dados dos parámetros: una tabla de 2 valores enteros que reflejan el estado actual y anterior de 8 interruptores, y un entero adicional que indica los pesos a chequear mediante bits a 1, calcule y devuelva el peso del bit más alto de los pesos a chequear que ha cambiado de 0 a 1, o -1 si no ha cambiado ninguno de estos bits.

1 punto

Ejemplo:

	Peso bit				Peso bit			
	7	6	5	4	3	2	1	0
actual	?	0	?	?	1	1	?	?
anterior	?	1	?	?	0	0	?	?
chequear	0	1	0	0	1	1	0	0

Devuelve 3

Algoritmo sugerido: desde los pesos j=7 hasta 0, si se cumple:

“el bit de peso j está activo en chequear” AND “el bit de peso j está activo en actual”
AND “el bit de peso j no está activo en anterior”

entonces retornar el valor de j.

Si se llega al final del bucle, retornar -1.

- 3) Realizar una función que, dados como argumentos el nombre de un archivo y un *texto* a buscar, lea el archivo línea por línea y compruebe para cada línea si comienza con *texto* entre corchetes. En caso afirmativo, a continuación del [*texto*] debe haber un '=' (se permiten espacios antes y después del '=') y un valor real, que debe retornar la función. Si ninguna de las líneas contiene el texto, o no se encuentra el '=' después del mismo, se devolverá el valor 0.

1.5 puntos

En ambos casos, el archivo debe quedar cerrado al terminar la función.

Ejemplo:

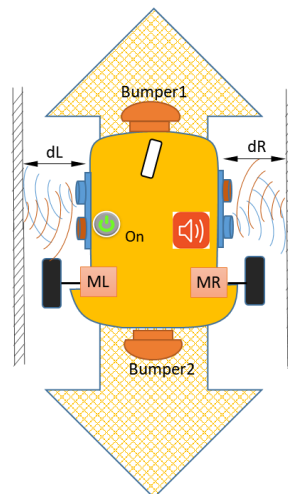
"datos.txt"	
[POS] =40.5	FuncionEj3("datos.txt", "VEL") → devuelve 75.2
[VEL] = 75.2	
[ACCEL] =10.5	FuncionEj3("datos.txt", "TEMP") → devuelve 0.0

Grado en Ing^a Electrónica Industrial y Automática

Informática Industrial y Comunicaciones

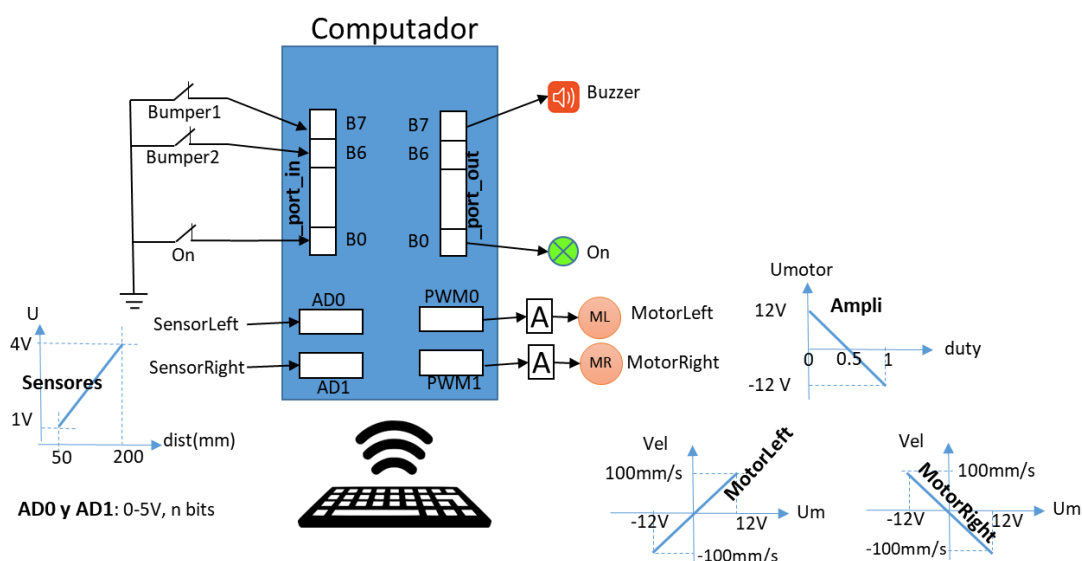
Examen Ordinario – Enero 2022

- 4) Se desea realizar un sistema de control para que un vehículo autónomo realice un trayecto rectilíneo entre dos paredes paralelas. Para ello, se dispone del montaje siguiente:
5 puntos



Dos motores (ML y MR) mueven sendas ruedas, mientras que una rueda loca permite el giro. Si todo está perfecto la trayectoria será recta y paralela a ambas paredes. Sin embargo, si algo no es perfecto, el vehículo se irá acercando a una de las paredes. Para corregirlo, se debe incrementar la velocidad del motor del lado del que la distancia se está decreciendo.

Se dispone de las siguientes conexiones del vehículo con el computador de control:



El accionamiento de los motores se realiza mediante la activación de sendas PWM según las gráficas mostradas a la derecha en la figura anterior.

La posición lateral se detecta mediante sendos sensores de ultrasonidos conectados a convertidores A/D de 12 bits, 0-5V. Los sensores son capaces de obtener una medida entre 50 y 200mm, generando una tensión entre 1 y 4V.

Los sensores Bumper1 y Bumper2, y el pulsador On, todos ellos de tipo todo/nada, están conectados al puerto digital accesible a través de la variable global `_port_in`. Se puede accionar el Led del pulsador On, y un avisador acústico, a través de sendas salidas tipo todo/nada conectadas al puerto digital accesible a través de la variable global `_port_out`.

Grado en Ing^a Electrónica Industrial y Automática

Informática Industrial y Comunicaciones

Examen Ordinario – Enero 2022

El funcionamiento deseado es el siguiente:

- Hay 4 estados para el sistema:
 - ESTADO_OFF: sin arrancar. Se desactivan el LED On y el zumbador, y se establece la consigna de velocidad de ambos motores a cero.
 - ESTADO_FWD: avanzando. Se activa el LED On y se desactiva el zumbador, y se establece la consigna de velocidad de ambos motores al valor VELMAX inicializado en main.
 - ESTADO_REW: retrocediendo. Se activa el LED On y se desactiva el zumbador, y se establece la consigna de velocidad de ambos motores al valor -VELMAX.
 - ESTADO_ERR: parado con error. Se activan el LED On y el zumbador, y se establece la consigna de velocidad de ambos motores a cero.

Se realizará una función **ActivarEstado(int estado)** que, cuando hay un cambio de estado, produce las modificaciones anteriores.

- En el programa principal, se solicitan por teclado al usuario:
 - El número de datos n a recordar para el cálculo de la media de distancias de sensores; con éste valor se asigna memoria dinámica para una tabla que recuerde el número de valores deseado.
 - El nombre de un archivo de texto, que se procesa con la función del ejercicio 3 para obtener el valor de la velocidad de avance (campo "VELMAX") y el límite de distancia a las paredes (campo "DMIN").
 - Se inicializan las variables y tablas necesarias, se activa ESTADO_OFF y se activa una interrupción temporizada cada 10ms. A continuación se ejecuta en un bucle:
 - Se realiza un bucle en el que se obtiene el nuevo valor de los bits del puerto de entrada, y mantiene el estado anterior mientras no ocurra ninguna activación en los bits de interés (función ejercicio 2). Si ocurre la activación de alguno de estos bits, en función de la activación ocurrida:
 - Si ha sido la tecla On, se pasa a ESTADO_FWD si el estado anterior era ESTADO_OFF, y a ESTADO_OFF en caso contrario.
 - Si ha sido el Bumper 1 y el estado era ESTADO_FWD, se activa ESTADO_REW.
 - Si ha sido el Bumper 2 y el estado era ESTADO_REW, se activa ESTADO_FWD.
- En la interrupción temporizada se realizarán, por este orden:
 - Si el estado es ESTADO_OFF o ESTADO_ERR, se termina.
 - Para ESTADO_FWD o ESTADO_REW:
 - Lectura de los valores de distancia en mm de los sensores ultrasónicos.
 - Aplicación del filtro (función ejercicio 1) para los n valores más recientes de la diferencia, y comprobación de cercanía a las paredes laterales
 - Si hay demasiada cercanía a las paredes laterales: Activación de ESTADO_ERR
 - Si no hay demasiada cercanía a las paredes laterales:
 - Corrección de velocidades de los motores a partir de las velocidades anteriores y de la diferencia de distancias media (función disponible en anexo).
 - Aplicación de PWM para las nuevas velocidades calculadas.

Grado en Ing^a Electrónica Industrial y Automática

Informática Industrial y Comunicaciones

Examen Ordinario – Enero 2022

ANEXO

Librería auxiliar:

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

```
// 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);

// Activación de PWM a partir del valor de duty (0.0 a 1.0)
void AplicarPWM(int nCanal,float duty);

// Variables globales para entradas y salidas digitales
int _port_in,_port_out;

// Función para la corrección de velocidades según la posición media
void CorrigeVelocidad(float* vLeft_mm_s,float* vRight_mm_s,float diff_d_mm);
```

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,...); // Escribre datos en stream de texto con formato
void fclose(FILE* fid); // Cierra stream
```

Grado en Ing^a Electrónica Industrial y Automática

Informática Industrial y Comunicaciones

Examen Ordinario – Enero 2022

APELLIDOS Y NOMBRE: _____

Ejercicio 5 (responder aquí):

1 punto

a) ¿ 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, escribir *datos[]* en la llamada a *Cuadrados()* es inválido
- Las dos anteriores son ciertas
- Ninguna de las anteriores es cierta

b) ¿ Cómo se pone a 0 el bit de peso *p* de la variable entera *v*, sin modificar el resto ?

- `v=1<<p;`
- `v=~1<<p;`
- `if (v & (1 <<p))`
- Ninguna de las anteriores

c) ¿ Cómo sabe un receptor RS-232 que el emisor envía un byte en una comunicación serie de 115200baud (ciclo reloj 8.68µs)?

- La línea RxD del receptor pasa de 1 a 0 (bit de start), el 1^{er} bit se debe leer 1.5*8.68µs después, el resto cada 8.68µs después del 1°.
- La línea CS del receptor es activada por el emisor. Se lee un bit cada 8.68µs.
- La línea CLK del receptor comienza a pulsar con ciclo 2*8.68µs. Se lee un bit en cada flanco ascendente de reloj.
- La línea CLK del receptor comienza a pulsar con ciclo 2*8.68µs. Se lee un bit en cada flanco (ascendente o descendente) de reloj.

d) ¿ Qué es un puerto IP ?

- Un valor de 32bits que se asigna a cada equipo conectado en una red
- Un texto separado por puntos que se asigna a cada equipo conectado en una red
- Un valor de 16bits que se asigna a un programa de un equipo conectado en una red
- La extensión de un proveedor de internet