

Grado en Ingeniería Electrónica Industrial y Automática

Informática Industrial y Comunicaciones

Examen ExtraOrdinario – Julio 2020

- 1) 1.5 puntos Se considera que un motor de corriente continua ha llegado a un tope si, dados n valores temporales de medición de su velocidad y corriente consumida, se cumple que:
- La velocidad es decreciente en valor absoluto: $|v_{k-i}| \leq |v_{k-i-1}| \quad \forall i=0 \dots n-2$
 - La corriente es creciente en valor absoluto: $|I_{k-i}| \geq |I_{k-i-1}| \quad \forall i=0 \dots n-2$

Realizar una función que, dadas dos tablas de números reales (del mismo tamaño) que representan los valores temporales de su velocidad y corriente recientes, devuelva el carácter 'T' si el motor ha llegado a un tope, o el carácter 'X' en caso contrario.

- 2) 1.5 puntos Realizar una función `ActivarSemaforo()` que recibe 3 valores enteros, que indican (verdadero/falso) el estado de tres elementos (`LSclosed`, `LSopen`, `Tope`), y devuelve un valor entero con los bits siguientes:
- Si `LSopen` es verdadero y `Tope` es verdadero, bit 3 del resultado a 1, bits 4 y 5 a 0.
 - Si no, si `LSclosed` es verdadero y `Tope` es verdadero, bit 5 del resultado a 1, bits 3 y 4 a 0
 - Si no, si `Tope` es falso, bit 4 del resultado a 1, bits 3 y 5 a 0
 - Si ninguna de las anteriores, bits 3, 4 y 5 a 1.

Ej: `ActivarSemaforo(1,0,1)` → devuelve 00..00100000 (bit 5 a 1, bits 3 y 4 a 0)

- 3) 1.5 puntos Realizar una función que, dada una cadena de caracteres, añada a un archivo llamado "comandos.log" dicho texto junto a la fecha y hora actuales, obtenidas de una función ya disponible:

```
void GetDateTime(int tabla[]);
```

La cual rellena los elementos `tabla[0]` a `tabla[5]` con el año, mes, día, hora, minuto, segundo respectivamente.

- 4) 1.5 puntos Se dispone de una librería con funciones para realizar E/S analógica en un computador (ejemplo, simulador del trabajo). Describir muy brevemente:

4.a) Qué tipos de archivo (extensión y contenidos) debe proporcionar dicha librería.

4.b) Qué archivo del entorno de programación Qt Creator se debe modificar para que nuestro programa pueda utilizar las funciones de esta librería.

4.c) Qué debe contener xxxxx en la línea siguiente en dicho archivo:

```
INCLUDEPATH += xxxxxx
```

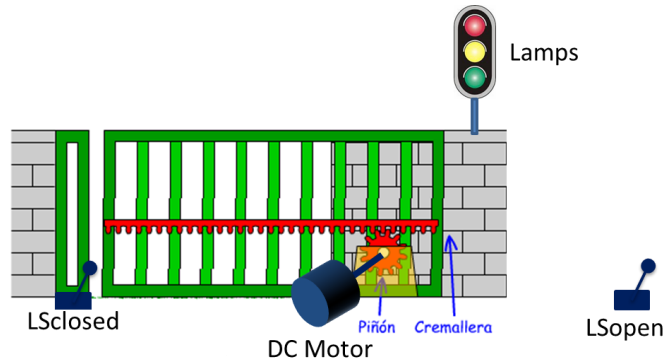
Grado en Ingeniería Electrónica Industrial y Automática

Informática Industrial y Comunicaciones

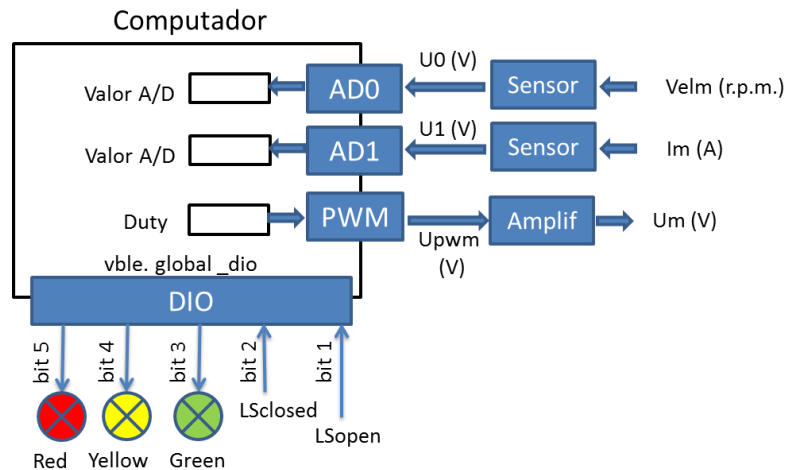
Examen ExtraOrdinario – Julio 2020

- 5) Se desea realizar el control de un mecanismo de apertura de puerta como el que se indica en la figura:

4 puntos



Para lo cual se ha conectado un computador con las E/S siguientes:



Funciones de E/S disponibles en librería:

```
void SetTimer(int timeout_ms, void (*FnCallback)());
int ReadAD(int n_channel);
void SetPWMDuty(float duty_0_a_1);
int _global_dio; // vble global para E/S digital
```

Especificaciones de las E/S analógicas:

Conversores AD0, AD1: Tensión de entrada -0V a 5V, 8 bits

Sensor pos portón: 0mm → 0V
4000mm → 5V

Sensor intensidad: -5 A → 1V
5A → 4V

Amplif. PWM: -24VDC para duty=0, +24VDC para duty=1

Grado en Ingeniería Electrónica Industrial y Automática

Informática Industrial y Comunicaciones

Examen ExtraOrdinario – Julio 2020

Las condiciones de funcionamiento son las siguientes:

En main()

- Se solicita al operador el número de valores anteriores de velocidad y corriente del motor necesarios para chequear detención del motor por llegada a tope.
- Se asigna memoria para valores temporales y se inicializan los valores de velocidad y corriente a 0.
- Se activa una interrupción temporizada cada 100 ms (función SetTimer).
- Se realiza un bucle de solicitud de cadena de caracteres, que pueden contener:
 - OPEN=valor_de_Um_en_Volt
 - CLOSE=valor_de_Um_en_Volt

Cuando se recibe una cadena, se actualizan las variables necesarias para que la rutina de interrupción para saber cuál es el último comando introducido y la tensión deseada.

Si la cadena no es válida, se actualizan las variables necesarias para que la rutina de interrupción detecte comando inválido y pare el movimiento del portón.

Además de lo anterior, se guarda en un archivo “movimientos.log” el comando introducido con su fecha y hora, mediante la función del ejercicio 3.

En la interrupción temporizada se realizan los siguientes pasos:

- Desplazamiento de tablas temporales.
- Lectura de AD0 y AD1 utilizando 8 bits de conversión A/D, y cálculo de valores actuales Pos_k y Im_k .
- Cálculo de velocidad del portón Vel_k por derivación de la señal de posición Pos_k
- Cálculo de detención del motor por llegada a *Tope* a partir de Vel_k e Im_k mediante función del ejercicio 1.
- Activar luces del semáforo según estado de los switches *LSopen*, *LSclosed* y el valor *Tope* calculado anteriormente, mediante función del ejercicio 2.
- Establecer tensión del motor en función del último comando recibido y del estado de *Tope*:
 - Si último comando=OPEN Y no se ha detectado *Tope*, aplicar duty PWM para conseguir tensión U_m indicada en comando OPEN.
 - Si último comando=CLOSE Y no hay *Tope*, aplicar duty PWM para conseguir tensión $-U_m$ indicada en comando CLOSE
 - Si ninguna de las anteriores se cumple → aplicar duty PWM para conseguir tensión 0.

Grado en Ingeniería Electrónica Industrial y Automática

Informática Industrial y Comunicaciones

Examen ExtraOrdinario – Julio 2020

Condiciones de programación:

Para la realización de las funciones de los ejercicios 1 a 3 es necesario determinar qué parámetros hay que pasar y qué valor devolver.

Para el ejercicio 5, utilizar las funciones realizadas en ejercicios previos (aunque no se hayan hecho). Se pueden realizar más funciones si se consideran útiles.

El nombre de archivo y el texto de cada línea nunca superarán 80 caracteres.

Algunas funciones de C:

```
int atoi(const char* cadena);
double atof(const char* cadena);
double strtod(const char* cadena, char** final);
int strlen(const char* cadena);
char* strcpy(char* dst, const char* src);
char* strncpy(char* dst, const char* src, int n);
char* strcat(char* dst, const char* src);
char* strncat(char* dst, const char* src, int n);
int strcmp(const char* cad1, const char* cad2);
int strncmp(const char* cad1, const char* cad2, int n);
char* strchr(const char* cad, char c);
char* strstr(const char* cad, const char* busca);
FILE* fopen(const char* nombre, const char* modo);
char* fgets(char* cad, int tam_max, FILE* f);
int fputs(const char* cad, int tam_max, FILE* f);
int fscanf(FILE* f, const char* fmt, ...);
int fprintf(FILE* f, const char* fmt, ...);
int fclose(FILE* f);
double fabs(double f);
```