

Grado en Ingeniería Electrónica y Automática

Informática Industrial y Comunicaciones

Examen Ordinario - Mayo 2014

- 1) Realizar una función que, dado un carácter que indica potencia de 10 en el Sistema Internacional (**M**ega, **K**ilo, **H**ecto, **D**eca, **d**eci, **c**enti, **m**ili, **u** - micro), devuelva como real el factor de multiplicación correspondiente:

1 punto

Ejemplo:

Potencia_SI('H') → devuelve 100.0 (float)

Algoritmo sugerido: realizar una tabla constante de char con los caracteres, y otra de float con sus factores. A continuación, buscar el índice del carácter en la 1ª tabla, y devolver el valor correspondiente al mismo índice en la 2ª tabla.

- 2) Realizar una función que, dado un valor de presión en las unidades indicadas mediante una cadena de caracteres, devuelva el valor convertido a otras unidades (indicadas mediante una cadena de caracteres). El formato de la función será:

1.5 puntos

```
float ConversorPresion(float valor,const char* unid_entrada,const char* unid_salida);
```

Donde unid_entrada y unid_salida pueden ser las cadenas: "bar" , "atm" ó "Pa", cada una de las cuales puede estar precedida por un carácter de potencia de 10 en el S.I. No es necesaria la comprobación de errores.

Se suponen definidas las constantes siguientes:

```
#define BAR_EN_UNA_ATM    1.01325F
#define BAR_EN_UN_PA     1.0e-5F
```

Ejemplo:

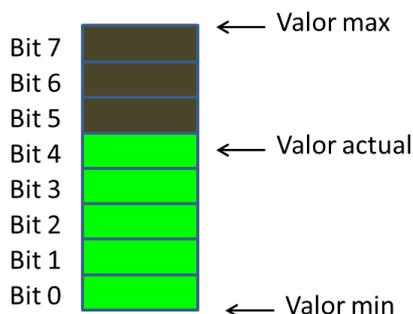
ConversorPresion(2.54F,"mbar","Pa") → devuelve 254.0 (float)

Algoritmo sugerido:

- Realizar una función auxiliar que devuelva el factor de conversión a bar de las unidades indicadas: float FactorConvABar(const char* unid);
- En la función auxiliar, buscar la cadena "bar" , "atm" ó "Pa" en la cadena unid, para obtener el factor de conversión adecuado. Si la cadena encontrada está al principio de la cadena unid, ya está. Si no, usar el 1er carácter de la cadena unid para modificar el resultado según el factor de potencia de 10 en el S.I.
- Utilizando la función FactorConvABar(), convertir de las unidades de entrada a bar, y a continuación de bar a las unidades de salida.

- 3) Realizar una función que, dado un valor real y sus límites mínimo y máximo, devuelva como entero el valor de los bits de una barra de LED que muestren el nivel de ese valor:

1.5 puntos



Algoritmo sugerido:

- Calcular paso por cada división
- Para cada bit:
 - Activar si $v_{actual} > v_{min} + i_{bit} * paso$
 - Desactivar en caso contrario

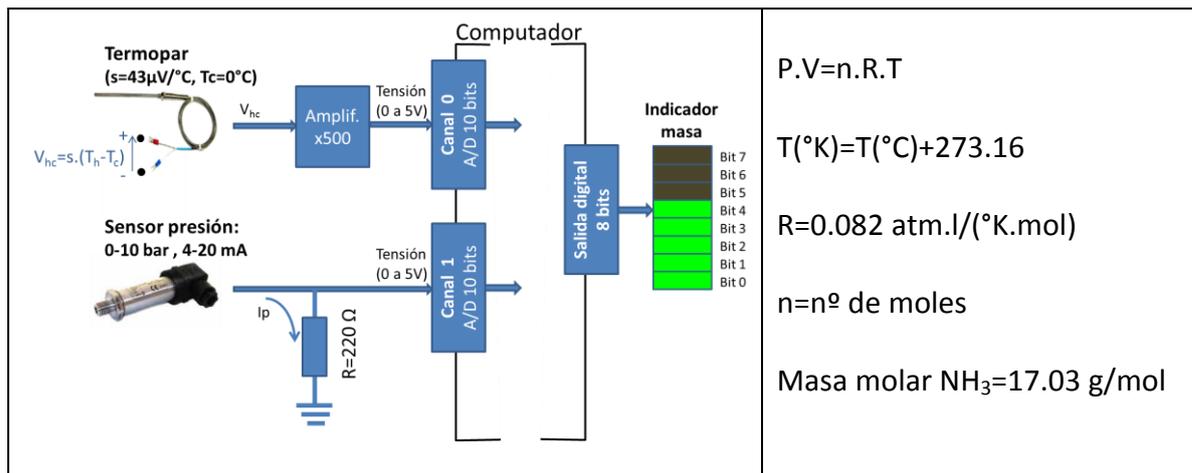
Grado en Ingeniería Electrónica y Automática

Informática Industrial y Comunicaciones

Examen Ordinario - Mayo 2014

- 4) Se desea actualizar **cada 10 seg** en una barra de LED la masa media **en el último minuto** del gas NH₃ contenido en un depósito de 500 litros. Para ello, se dispone del siguiente montaje que permite medir la presión y temperatura del depósito, a partir de las cuales se calcula el n° de moles de gas, y con él su masa.

4 puntos



Las condiciones de funcionamiento son las siguientes:

- Se dispone de las siguientes declaraciones de funciones de E/S en “io.h”:

```
void InitTemporizador(int T_ms,void (*FnCallback)() );
// Lanza una temporización por callback.
// La función callback debe declararse como: void MiFn();

int ValorAD(int n_canal,int n_bitsAD);
// Obtiene el valor de la conversión A/D del canal deseado con el n° de bits
// de digitalización indicado.

int* SalidaDigital();
// Obtiene la dirección de la variable (puerto de E/S) conectado con la
// salida digital (barra de LEDs).
```

- Se desea utilizar conversiones A/D de 10 bits y **Tm=1seg**.
- La sensibilidad del termopar es de 43 µV/°C. La temperatura Tc es 0°C.
- El sensor de presión genera una corriente según una relación lineal con límites 4mA (para P=0bar) y 20mA (para P=10bar). Se convierte la corriente a tensión mediante una resistencia de 220 Ω.
- Se deben usar las funciones de ejercicios anteriores, disponibles en “aux.h”.
- En el programa principal, por este orden, se solicita al usuario por teclado los valores de masa_min y masa_max para la barra de LEDs, se inicializan las variables necesarias (asignación dinámica de memoria para tablas), se ordena el inicio de la temporización (asíncrona por callback), y se queda a la espera de final con getchar().
- En la función callback (cada 1 seg) se adquieren los valores medidos por los sensores y se realiza el cálculo de masa, se añade a la tabla temporal de masas medidas, y una de cada 10 veces se calcula la media de la tabla y se actualiza el valor de la barra de LEDs.

Grado en Ingeniería Electrónica y Automática

Informática Industrial y Comunicaciones

Examen Ordinario - Mayo 2014

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);
int sprintf(char* dest, const char* fmt, ...);
int sscanf(const char* src, const char* fmt, ...);
FILE* fopen(const char* filename, const char* mode);
char* fgets(char* dest, int tam_max, FILE* f);
int fprintf(FILE* fid, const char* fmt, ...);
int fscanf(FILE* fid, const char* fmt, ...);
int fclose(FILE* fid);
void* malloc(int tam);
void free(void* pt);
```