2° parcial INFORMATICA II	Total Hojas	Duración de todo el parcial 8:30 a 12:00	17/11/2015

Nombre y Apellido	Nº Legajo	Calificación

PARTE1:

Figura 1

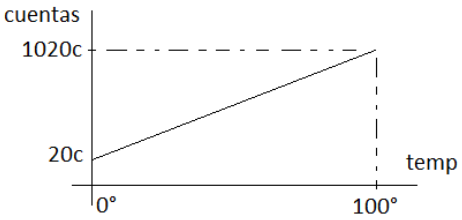

Figura 2

Se pide desarrollar UN equipo de sensado de temperatura que está conectado en una red, considerar que hay más equipos conectados a la misma red y desde una central se solicitan los datos de temperatura de cada uno de ellos. Cada equipo tiene un número de identificación (ID) único de un byte que sirve como identificación (CONSIDERAR QUE ES UN DEFINE).

CENTRAL ➔ EQUIPO: La central envía un comando de lectura cada 1 segundo con el siguiente formato.

CARÁCTER INICIO '>'	ID (1 BYTE)	COMANDO LECTURA '0x02' (1 BYTE)	CARÁCTER FIN '<'

EQUIPO ➔ CENTRAL: La respuesta del equipo hacia la central

CARÁCTER INICIO '>'	ID (1 BYTE)	TEMPERATURA MEDIDA (2 BYTEs) (00.0º a 99.9º)	CARÁCTER FIN '<'

Además cada equipo deberá encender una alarma externa conectada a una GPIO cuando la temperatura medida supera un valor de temperatura que se setea desde la central, por debajo de esta permanece apagada la alarma. (1=encendido , 0=apagado)

CENTRAL ➔ EQUIPO: La central envía el siguiente comando para setear la temperatura de corte de los equipos

CARÁCTER INICIO '>'	ID (1 BYTE)	COMANDO ESCRITURA '0x03' (1 BYTE)	TEMPERATURA DE CORTE (2 BYTEs) (00.0º a 99.9º)	CARÁCTER FIN '<'

EQUIPO ➔ CENTRAL: La respuesta del equipo hacia la central

CARÁCTER INICIO '>'	ID (1 BYTE)	COMANDO ESCRITURA '0x03' (1 BYTE)	TEMPERATURA DE CORTE SETEADA (2 BYTEs) (00.0º a 99.9º)	CARÁCTER FIN '<'

El sensor de temperatura esta conectado al ADC0 del microcontrolador, es lineal y cumple con la curva de la figura 2.

Se pide realizar las siguientes funciones de la central:

- Realizar la función "main".
- Realizar todas las funciones para controlar la recepción y la transmisión de las tramas. (NO USAR POOLING)
- Realizar la función de interrupción del puerto serie. (Ya fue realizada la función de inicialización).
- Realizar la función que controla la alarma.
- Realizar la interrupción y conversión del ADC. (Ya fue realizada la función de inicialización) (NO USAR POOLING)

PARTE2:

Una pinturería industrial prepara sus pedidos a través de un equipo informatizado. El mismo está basado en un sistema escrito con funciones hechas en C++. De ese sistema se utilizaran dos clases: a saber: clase **color** y clase **preparación:**

- **clase Color**: Tiene las variables miembro correspondiente a los colores primarios **R ,G , B** (rojo –verde – azul respectivamente) , cuyo contenido está en función de la proporción necesaria (porcentual), para obtener colores

complementarios. Además posee otras variables miembro en donde se almacenará el **costo** por litro de dicha combinación y el nombre (**name**) de la mezcla asignado (ej: Turquesa , Violeta , Rojo, etc)

PUNTO 1) Se pide que complete la clase color (todos los puntos señalados con flechas celestes) y codifique sus funciones miembro en los archivos correspondientes " .h" y " .cpp" .

```
                                                    // Color.h
class Color  {
        protected:
                int             R,G,B;          // porcentuales por litro de preparacion
                char*           name;           // permite acceder al nombre del color
        private:
                float costo;                    // costo por litro
        public:
        /*Función miembro inline que al ejecutarse, instanciando un objeto de la clase,  inicializa por defecto sus variables R, G,B  a valores triviales o través de sus argumentos*/

        /*Función miembro que al ejecutarse, instanciando un objeto de la clase,  inicializa sus variables R, G,B  con los valores del parámetro que recibe como argumento, siendo este del tipo color&*/

        /*Función miembro inline que al ejecutarse, al ser "abandonado" un objeto de la clase, libere los  recursos que se hubieran tomado del sistema*/

                void        set_name ( tipo ..? );      // Permite asociar un nombre a una combinación
                tipo ..?    get_name ( void );          // Permite acceder al  nombre de una combinación
                void        set_color ( int , int , int );  // set RGB
                int         get_color  ( int );         // Retorna el contenido de R, G o B –
                                                        // utilice macros como argumento
                void        set_costo ( float );        // Setea costo por litro de la combinación
                float       get_costo ( void);          // Retorna el  costo
                color & operator = ( color& );          // Sobrecarga de operador =

        /* Función crea_color  tal que permita ejecutar desde el programa principal (main) :
        C.crear_color (A, B) ; siendo A , B , C objetos de la clase, donde las variables miembro de C serán el promedio (despreciando decimal) y el nombre la combinación su concatenación */

        /*sobrecarga del operador << para permitir la impresión por consola de objetos de tipo color.
        (uso: cout << A;) siendo A un objeto color.*/
};
```

- **PUNTO 2)** Se pide que desarrolle la clase **Preparacion** (derivada de la clase color) y sus funciones miembro en los archivos correspondientes " .h" y .cpp", teniendo en cuenta las siguientes consideraciones:

```
        miembros:
        float   costo;          // costo de la preparación.
        int     litros;         // cantidad a preparar
        xxxx    cant;           // cantidad  de objetos preparación creados

        //constructor parametrizado cuyos dos argumentos sean del tipo color& (color a preparar) e int (litros), que garantice la inicializacion de todos los miembros.

        // Destructor

        void    set_costo ( float );    // Setea el costo por litro , y actualiza el de preparación
        float   get_costo ( );          // Retorna el costo de la preparación

        /*sobrecarga del operador = .  ¡OJO! Debe optimizar su código*/

        /*sobrecarga del operador <<.  ¡OJO! Debe optimizar su código*/
```