

FINAL INFORMATICA II	Total Hojas	Duración de todo el final 19:00 a 21:00	11/02/2025
----------------------	-------------	---	------------

Nombre y Apellido	Nº Legajo	Calificación

Se desea desarrollar en C++ el software de una placa de control para un colorímetro industrial. El mismo consta de un sensor de color RGB analógico, un teclado con 12 teclas, un display color y un puerto de comunicación serial.

El colorímetro tiene el siguiente procedimiento para realizar la medición correctamente.

- 1- Se debe colocar la muestra en el recipiente medidor y cerrar el compartimiento (hay un sensor que detecta que está perfectamente cerrado)

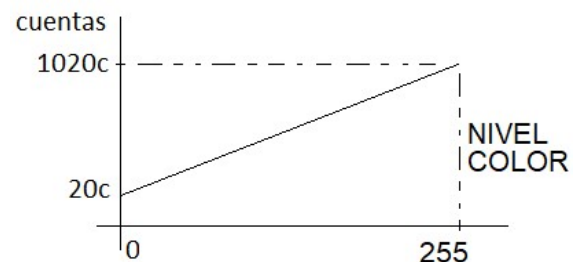
Si el compartimiento está abierto se debe encender un led rojo de advertencia. El mismo se apagará al cerrar correctamente.

- 2- Selecciona el modo de medición adecuado en el colorímetro por medio de 2 teclas donde puedes elegir entre medición de reflectancia o transmitancia.

- 3- Presionar el botón de inicio de medición.

Hay que tener en cuenta que la medición consta de 3 colores (RGB, rojo, verde y azul) Cada salida analógica del sensor está conectada a un canal del ADC del microcontrolador. (El sensor tiene la siguiente curva de color vs tensión)

- 4- Una vez finalizada la medición se visualizará en el display y se enviará a la PC por medio del puerto serial



1 byte '>'	1 byte MODO DE MEDICION	1 byte NIVEL ROJO	1 byte NIVEL VERDE	1 byte NIVEL AZUL	1 byte CHECKSUM (suma de los bytes anteriores)	1 byte '<'
----------------------	-----------------------------------	-----------------------------	------------------------------	-----------------------------	---	----------------------

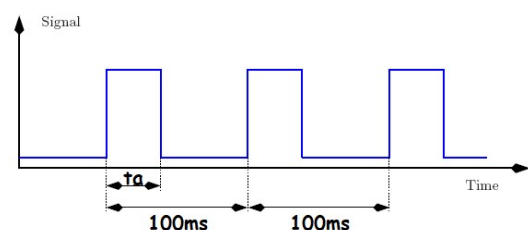
La PC responderá con una cadena en caso de una correcta recepción, en caso de no recibir la trama en 20 segundos se deberá reenviar nuevamente. (Máximo 3 veces)

1 byte '>'	1 byte '1' = RECEPCION EXITOSA	1 byte '<'
----------------------	--	----------------------

Se pide realizar las siguientes funciones:

1. Implementar todas las funciones de control de la máquina de estado que controle el equipo.
2. Implementar todas las funciones que intervienen para enviar y recibir la trama por puerto serie. (Ya está todo inicializado a 9600bps)
3. Implementar la función lectura del ADC para los 3 canales (RGB) por interrupción. ¿Desde dónde la llamaría? No hace falta filtrar e inicializar.
4. Implementar el barrido por hardware del teclado suponiendo que solamente dispone de 8 pines.
5. Como futura mejora se está proponiendo controlar la intensidad del display por medio de PWM, para esto el jefe de proyectos propuso generar un nuevo driver en c++, donde la idea sea crear una clase GpioPWM la cual se deba heredar la funcionalidad actual de GPIO y de Callback.

Se pide implementar la clase GpioPWM con los siguientes métodos:



```

void PWM(void); // Esta función se debe llamar cada 1ms para generar el PWM
GpioPWM(uint32_t port, uint32_t pin); // Constructor, donde se pasa el puerto y pin del gpio
void Set(uint32_t porcentaje); // Función que se llama para ir modificando el ciclo de actividad
// Los rangos son de 0 a 100. 0 = apagado ; 100 = full actividad
uint32_t Read( void ); // Función que devuelve el porcentaje de ciclo de actividad actual
void Callback( void ); // Función de callback

```

Además se pide sobrecargar el método = para setear el ciclo de actividad

NOTAS:	<ul style="list-style-type: none"> • No se pide el antirrebote • No se pide inicializar nada de HW • Si bien el teclado tiene varias teclas más, no se pide la resolución de lo que no está escrito.
--------	---