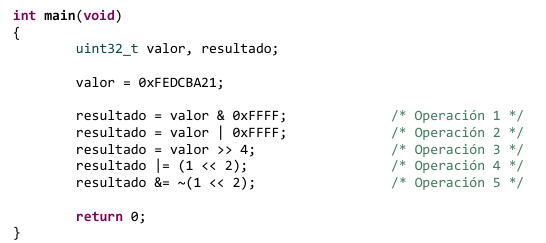
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2do.Parcial INFORMATICA II | Total Hojas | Hola Fin | PARCIAL | 24/Nov/2020 |
|  | 12hs. | # 2 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre y Apellido del alumno** | **Nº Legajo** | **Calificación** | **Docente a cargo / Firma** | |
|  |  |  |  |  |



**1.** Dado el siguiente código, indicar claramente cuál es el resultado de las Operaciones 1 a 5.

|  |  |
| --- | --- |
| OPERACION | RESULTADO |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |

**Nota**: Complete la tabla que se indica. Las 5 operaciones deben estar bien hechas. No se considerará puntaje parcial en este ejercicio. En su hoja de parcial (o en un documento de texto en la PC) debe haber constancia de cómo resolvió cada operación.

**2.** Realice todas las acciones necesarias para configurar como GPIO el P0.22 y P0.24 como salida y escriba un uno a sus pines asociados. Asimismo, configure P3.1 como entrada.

**Nota**: Absténgase de usar funciones utilitarias ya resueltas de acceso a HW. Haga uso de las hojas de datos. Procure optimizar el uso de líneas de código. COMENTE QUE HIZO EN CADA LINEA BREVEMENTE PERO QUE SE ENTIENDA.

**3.** Sabiendo que el microcontrolador posee un periférico denominado PWM cuyos registros se muestran, en parte, en la siguiente tabla, se pide que declare todo lo necesario para que desde un código escrito en C se pueda acceder a ellos a través de una estructura o a través de punteros, según su preferencia.

|  |  |
| --- | --- |
|  | Luego de cumplimentar la consiga Ud. debe ser capaz de escribir en su código:  PWM->IR = MASCARA\_INT;  PWM->TCR = MASCARA\_CTRL;  o  IR = MASCARA\_INT;  TCR = MASCARA\_CTRL;  *(siendo ambas MASCARAS #define de los cuales Ud. no debe ocuparse)* |

**4.** Ejercicio integrador**: Medidor de temperatura y humedad**





**max**

**max**

**min**

**min**

**reset**

**T**

**H**



El equipo consiste en un medidor de temperatura y humedad ambiente que muestra la medición actual de ambos parámetros en 2 displays de 7 segmentos cátodo común de dos dígitos cada uno (DSP0 ⇒ Temp y DSP1 ⇒ Hum). Los valores que el medidor puede mostrar son ÚNICAMENTE enteros.

El equipo, además, puede detectar y capturar máximos y mínimos de ambos parámetros Y cuenta con un teclado de 5 teclas que permiten acceder a dichos valores extremos de ambas magnitudes y además cuenta con un botón de reset (que reinicializa a cero los valores almacenados).

Los máximos y mínimos se deben mostrar por 2,5 segundos en el display correspondiente una vez oprimida la tecla respectiva. Luego, vuelve a mostrar el valor de la magnitud actual.

El dispositivo lee ambos parámetros de un sensor que tiene entradas analógicas para ambas magnitudes, o sea que requiere de dos canales del ADC para medirlos (use canal 1 ⇒ Temp y canal 5 ⇒ Hum).

Dado que tanto la humedad como la temperatura son magnitudes físicas que varían poco con el paso del tiempo, Ud., como diseñador, tiene la posibilidad de optar por cualquiera de las dos siguientes posibilidades:

1. Usar el método BURST para tomar automáticamente de a 50 lecturas y obtener el promedio de ellas[[1]](#endnote-1) .
2. Disparar (y consecuentemente leer) las conversiones de cada canal una vez por segundo. Dado que hay que convertir de a un canal por vez, le sugerimos que la separación entre mediciones de diferentes magnitudes sea de 500ms. (en criollo: cada medio segundo dispara las conversiones alternando entre el canal 1 y el 5)

El valor obtenido en cuentas sea cual fuere el método que use deberá ser formateado para mostrar en los displays desde la aplicación. Para ello, utilice la siguiente función primitiva: *uint8\_t format (uint32\_t)*, que recibe el valor filtrado de cada canal de ADC y devuelve un valor que se puede mostrar en los displays (transforma cuentas en valores numéricos enteros). *No debe ocuparse de su realización. solo úsela*.

El equipo cuenta con una interfaz bluetooth para comunicarse con una APP de smartphone que permite controlarlo y monitorearlo de manera remota. A los efectos del programa, la interfaz BT es un puerto serie.

Las mismas funciones que tiene el equipo desde su panel de control las tiene por interfaz serie. Los comandos que recibe son dos:

**$D#**; orden remota para que el medidor devuelva las lecturas de temperatura y humedad, así como los valores máximos y mínimos de ambos parámetros.

La respuesta del medidor es de la forma: **$DtahaTTHHtthh#**

*Donde todos los datos* ***son caracteres ASCII****. El paquete comienza con ´$´ y termina con ´#´. La letra ‘D’ indica que a continuación vienen los datos que son valores de dos cifras (en ASCII).*

*Datos: Primero se indica la temperatura actual (ta), luego la humedad actual (ha), luego la temperatura máxima (TT), la humedad máxima (HH), la temperatura mínima (tt) y la humedad mínima (hh).*

**$RH#**; orden remota para que el medidor resetee las humedades máximas y mínimas

La respuesta del medidor es de la forma: **$H00#**

**$RT#**; orden remota para que el medidor resetee las temperaturas máximas y mínimas

La respuesta del medidor es de la forma: **$T00#**

**Nota:** Los valores de temp y hum son siempre positivos. Tenga en cuenta que Ud. recibe ‘cuentas’ que son transformadas en valores enteros a través de *format* y que DEBE enviar ASCIIs. Recordar: ‘0’ = 48 en decimal.

Se pide realizar:

1. El ejercicio utilizando la técnica denominada Máquina de Estados, representando correctamente el/los diagramas de globos del sistema y su representación en C.
2. La isr del puerto serie y la función que decodifica la trama que se recibe.
3. ADC (inicialización e Isr)
4. La lectura (barrido) del teclado (No se pide debounce. Asuma que existe *TecladoSW().* Úsela).
5. La presentación de los datos en los displays (primitiva y firmware)

**Nota a:** Asuma que las siguientes funciones ya se encuentran hechas. Simplemente úselas si así lo requiere:

* void push (void) y uint16\_t pop (void) de la comunicación serie.

**Nota b:** Asuma que todos los pines y todos los periféricos (con excepción del ADC) ya fueron inicializados. Solo debe incluir a comienzos del *main()* la función *Inicialización()*. El systick interrumpe cada 2,5ms.

**Nota c:** Para el uso de los pines que necesite para teclado y display asuma que ya fueron hechas las respectivas macros del tipo:

#define digito0 PORT2,7

1. Si se anima, active y desactive el BURST una vez por segundo. Sería lo lógico para estas circunstancias. Sino, (más fácil) manténgalo activo y haga el promedio todo el tiempo. [↑](#endnote-ref-1)