

**PRACTICA Nro. 2****CUESTIONARIO****Ejercicio No 1: Teclado Matricial**

- a) Explique cómo construir un teclado con 16 pulsadores dispuestos en forma matricial (4x4). Realice un esquema de conexionado para controlarlo con el puerto B del uC. Explique cómo configurar la dirección de los bits del puerto y cuáles requieren resistencias de Pull-ups.
- b) Realice una función en C con el siguiente prototipo `uint8_t KEYPAD_Scan(uint8_t *key)`, que permita encuestar el teclado para determinar si se presionó una tecla y el valor de la misma. La función debe retornar 0 si no se presionó ninguna tecla o 1 si se presionó alguna. El valor de la tecla (o carácter ASCII) debe retornarse por el puntero pasado como parámetro (\*key). ATENCION: el diagrama de conexionado será el del kit utilizado en clase.

**Ejercicio No 2: Display LCD**

- a) Vamos a utilizar el LCD Winstar1602 de 16 caracteres y dos líneas para visualizar información. Investigue con la hoja de datos como funciona este periférico, realice un resumen de sus características y de los comandos necesarios para su operación.
- b) Haga el diagrama esquemático en Proteus que incluya el LCD y el teclado matricial del kit utilizado en clase.
- c) Verifique que los niveles lógicos de la interfaz del periférico sean compatibles con los niveles de los puertos I/O del MCU.
- d) Explique cómo funcionan las funciones de biblioteca para el control del LCD que utilizaremos en la práctica (provistas en los archivos `lcd.h` y `lcd.c`). Particularmente, las funciones: `LCDinit()`, `LCDgotoxy()` y `LCDstring()`.

**Ejercicio No 3: Interrupciones**

- a) Explique cuál es la secuencia de pasos que realiza el MCU cuando recibe una señal de pedido de interrupción. ¿cómo se retorna a la ejecución normal del programa?
- b) Explique para qué sirve el bit I que se encuentra en el SREG. ¿Cuál es el valor por defecto luego del RESET? ¿Cuáles son las instrucciones que permiten modificarlo?
- c) Explique que es un RESET, las distintas maneras por las que puede generarse y la función del registro *MCU Status Register*.
- d) Explique qué es la latencia de una interrupción y cuáles son los valores mínimos que puede tomar medida en ciclos de reloj.
- e) Explique qué entiende por interrupciones anidadas. ¿Es posible por defecto?
- f) Investigue sobre la configuración y el uso de los terminales de interrupción Externa (INTx). ¿Cuál es la diferencia entre configurar la interrupción por flanco o por nivel?
- g) Interrupciones desde los puertos de Entrada/Salida (*PIN Change Interrupts*).
- Describa que terminales I/O pueden generar interrupciones por cambio de estado
  - Investigue como se configuran estas interrupciones en los puertos de Entrada/Salida
  - Describa cuales son los vectores de interrupción asociados y sus prioridades relativas.
  - Comente las diferencias más importantes con las interrupciones del punto f).

**Ejercicio No 4: Temporizadores**

- a) Describa los componentes principales del módulo TIMER0 y explique sus modos de funcionamiento.
- b) Explique cómo funciona el prescalador (Prescaler) y obtenga una expresión para calcular la frecuencia de reloj del contador ( $CLK_{TN}$ ) y la frecuencia de desborde (Overflow) en función de las distintas opciones de fuentes de reloj. ¿Cómo se borra la bandera de desborde?
- c) Si se tiene una frecuencia de  $CLK_{IO}$  de 8MHz, establezca con que valores debería inicializar los registros pertinentes del TIMER0 para realizar un retardo de 1ms. ¿es posible generar un retardo de 1000 ms?
- d) Si se tiene una frecuencia de  $CLK_{IO}$  de 8MHz, configure TIMER0 para generar una interrupción periódica de 1ms.

**Ejercicio No 5: Temporizador Asincrónico–RTC (Contador de Tiempo Real)**

- a) Analice las diferencias y similitudes entre TIMER0 y TIMER2.
- b) Si se usa el cristal externo de 32.768kHz, determine con que valores debería inicializar los registros pertinentes del módulo para obtener una interrupción periódica de 10 ms.
- c) Busque información sobre los circuitos integrados IC: DS1302 y DS3231 ¿Qué funcionalidad proveen? ¿Cómo se conectan a un MCU? ¿Qué ventajas le encuentra a su uso?

**Ejercicio No 6: Implementación de Maquinas de Estado Finitos (MEF)**

- a) ¿Qué es un modelo de Estados Finitos? ¿Qué elementos contiene? ¿Cómo se especifica?
- b) ¿Cuál es la diferencia entre el modelo propuesto por Moore y el de Mealy?
- c) Describa diferentes maneras para implementar en C un modelo MEF, ¿Cómo conviene declarar las variables de estados? De un ejemplo resumido del código para una posible implementación.
- d) ¿Cómo se inicializa una MEF? ¿Cómo se actualiza una MEF?
- e) Analice diferencias y similitudes de una implementación MEF en lenguaje C y en lenguaje VHDL partiendo del mismo modelo de la MEF.

**Ejercicio No 7: Planificación Cooperativa de Tareas y Modularización**

- a) Explique en qué consiste un esquema de planificación cooperativo con interrupción periódica de timer.
- b) Determine el periodo de la interrupción para la planificación de tres tareas periódicas x(), y(), z() con periodos 10, 25 y 50ms respectivamente. ¿Qué condiciones tiene que asumir para que el funcionamiento del sistema esté garantizado y sea confiable?
- c) Explique mediante pseudocódigo como implementar el planificador y despachador de las tareas y muestre con un diagrama temporal un ejemplo de la ejecución de las mismas.
- d) En el caso de que haya superposición de tareas en el mismo tick de sistema, explique cómo puede hacerse la planificación del inciso b) para que no haya superposición en la ejecución de x(), y(), z().
- e) Explique qué entiende por modularización de software y qué ventajas tiene.
- f) Explique la manera de modularizar archivos en C a partir del concepto de clases tomado de la P.O.O.
- g) Explique cómo se emplea el modificador `static` con las funciones, variables y constantes para modularizar un programa en C.
- h) Explique que son las funciones privadas y las funciones públicas de un módulo, de un ejemplo de implementación de cada una.

**Ejercicio No 8: Sistema de generación y distribución de reloj**

- a) Explique cuáles son las posibles fuentes de reloj y sus rangos de frecuencia para que funcione el MCU.
- b) Explique cómo se distribuyen internamente las señales de reloj para la CPU y los distintos periféricos según las diferentes opciones (diagrama en bloques de la distribución interna de las señales de reloj).
- c) Investigue sobre cómo utilizar la biblioteca de funciones de bajo consumo (sleep.h).
- d) Explique cuál es la configuración del reloj del sistema dispuesta desde fábrica y que tolerancia tiene la frecuencia del mismo.

**EJERCICIOS PARA REALIZAR CON SIMULADOR****Ejercicio No 9: Teclado y LCD**

- a) Realizar un programa para probar las funciones vistas en los ejercicios 1 y 2, es decir el programa debe ejecutar la función `KEYPAD_Scan` para determinar si se presionó una tecla del teclado matricial y en caso afirmativo mostrarla en el LCD (formato ASCII). El programa puede validarse con el simulador y con el kit utilizado en clase.

**Ejercicio No 10: Temporizadores**

- a) Realice un programa para hacer parpadear un LED conectado en el bit 0 del puerto B con 10Hz de frecuencia utilizando la interrupción del TMER0. Verifique la frecuencia (o el periodo) con el contador de

ciclos del simulador y con el Osciloscopio de Proteus. ¿Qué error se comete en la frecuencia y de qué depende? ¿afecta la latencia de la interrupción?

- b) Realizar en base al TIMER0, una función retardo (bloqueante) cuyo parámetro de entrada sea el número entero de milisegundos que se desea esperar. El rango de valores permitidos deber ser de 1 a 255mseg. Verifique que funciona en el ejemplo del TP1.

### Ejercicio No 11: MEF

a) Una forma de implementar un modelo MEF en software es mediante la sentencia switch-case. Utilice como ejemplo el diagrama de estados del controlador de un semáforo y explique este método describiendo como implementar las variables de estados, las entradas, las salidas y las transiciones.

b) Realice un programa para implementar el inciso anterior suponiendo que las luces del semáforo son controladas por los bits 6,7 y 8 del puerto B.

c) Realice un modelo MEF (diagrama de estados) que corresponda a un algoritmo de detección de un pulsador con anti-rebote. Explique el modelo desarrollado, la elección de los estados, entradas, salidas y transiciones.

d) Implemente el modelo de MEF del inciso anterior en una función que se pueda utilizar para consultar el estado de un pulsador conectado a un bit de un puerto de entrada.

### Ejercicio No 12: Planificación de tareas con RTI

a) Realizar el ejercicio entregado en el TP1 utilizando una interrupción periódica de TIMER0 para planificar y temporizar las tareas involucradas (no deberá usar más los retardos bloqueantes).

### Ejercicio No 13: Reloj Digital

a) Realizar la implementación básica de un reloj digital utilizando el MCU y un display LCD como el mostrado en la fig. 1. El formato a mostrar del reloj debe ser HH:MM:SS. Utilizar TIMER2 para generar una interrupción periódica que permita la actualización del reloj y del LCD cada 1 seg.



Figura 1

## EJERCICIO PARA REALIZAR CON EL KIT Y ENTREGAR CON INFORME

Se especificará en un documento anexo próximamente.