



FACULTAD DE INGENIERÍA DEL EJÉRCITO
UNIVERSIDAD DE LA DEFENSA NACIONAL

MATERIA: Técnicas Digitales II

CURSO: IV - Plan 2007 - Electrónica

TRABAJO PRACTICO Nro.: 2

TITULO: *“Manejo de periféricos y dispositivos externos”*

DOCENTE: Ing. Daniel Steiner

TRABAJOS PRACTICOS: Ing. Ariel Dalmas Di Giovanni

AÑO: 2019.

BIBLIOGRAFIA:

- Manual del Microcontrolador
- Hojas de datos de periféricos a implementar.
- Apuntes de clases

INTRODUCCIÓN

En las distintas aplicaciones donde se utilizan microcontroladores es frecuente la necesidad de interactuar con distintos dispositivos externos como ser: tipos de sensores (digitales, analógicos, etc), actuadores (*steppers*, servomecanismos, etc), conversores AD y DA, etc.

A su vez, es también común la necesidad de contar con una interfaz con el usuario a través de botones, llaves y presentar la información en forma visual mediante luces, *displays*; y también estímulos sonoros, etc.

Por otro lado, los microcontroladores están integrados por diferentes periféricos que mejoran la escala de integración de un sistema e incrementa la versatilidad de aplicaciones en las que se puede utilizar un mismo microcontrolador.

ORGANIZACIÓN

Los equipos electrónicos de ensayo y medición que intervienen en la práctica serán determinados por los alumnos, con guía y discusión por parte del PJTP.

El alumno recibirá y/o propondrá un problema a resolver, con un detalle de qué funciones debe implementar y la lista de requisitos a cumplir.

El alumno analizará las diferentes soluciones y determinará cómo lograrlo, desarrollando el hardware y software aplicando su propia inventiva.

OBJETIVO

Que el alumno aplique los conceptos dados en las clases teóricas y diseñe el/los circuito/s de interfaces conveniente/s para implementar con un microcontrolador la solución al problema planteado. Que analice y justifique que periféricos se utilizarán para dar solución al problema.

Que utilice un *timer* interno como base de tiempo del sistema.

Que diseñe, implemente y depure el *software* del microcontrolador.

DESARROLLO

Se cubrirán los siguientes aspectos:

- Esquemáticos del circuito.
- Diagramas de flujo de instrucciones.
- Codificación en ASSEMBLER.
- Ensayos en simulador – PC.
- Armado del circuito práctico (opcional).
- Mediciones y verificación práctica (opcional).

EQUIPOS

A continuación se listan los equipos necesarios para la práctica:

- PC, con IDE instalado.
- Programador de microcontrolador.
- Material y equipamiento para verificación en laboratorio (opcional).

CONCLUSIONES

El alumno deber elaborar un informe sobre la metodología empleada, y los resultados obtenidos según los siguiente tópicos:

- Módulos de Software.
- Hardware. (Cálculos, esquemáticos y justificaciones).
- Documentación.
- Pruebas y ensayos.
- Conclusiones.

ANEXO- Enunciado: Generador de burst

Se necesita generar un *burst* de pulsos rectangulares. La frecuencia del *burst* podrá ser de 2Hz o de 10Hz. El ancho de cada pulso del *burst* es de 1ms.

El usuario configura mediante pulsadores la cantidad de pulsos que tendrá el *burst*, la cantidad de pulsos debe presentarse mediante un *display* de 7 segmentos. La cantidad máxima es de 9 pulsos.

El sistema cuenta con una llave de disparo que genera la secuencia configurada en forma periódica.

Detalle de la operatoria:

Una llave para configurar la frecuencia y una llave de comienzo/detención de generación de señal.

Dos pulsadores para configurar la cantidad de pulsos del burst. Uno que incremente la cuenta y otro que la decremente. Internamente el sistema controlará la cantidad máxima de pulsos admisible según sea la frecuencia configurada. Una vez disparado el sistema, no se podrán modificar los parámetros.

El display de 7 segmentos deberá mostrar el valor de pulsos pre-configurados.

Se indicará mediante un led parpadeante cuando el sistema está disparado.

El período de parpadeo será de:

- 500 ms para cuando la frecuencia del *burst* es de 2Hz.
- 250 ms para cuando la frecuencia del *burst* es de 10Hz.

Si el sistema no está disparado, el led debe estar apagado.

Consideraciones:

- Diseñe un código no bloqueante.
- Los tiempos deberán derivarse de un *timer* del microcontrolador.