

TP1
EJERCICIO ENTREGABLE**METODOLOGIA:**

Según programa analítico de la materia aprobado en el consejo en 2008, los contenidos mínimos son:

- *Caracterización y componentes de hardware y software de un Sistema Embebido. Arquitecturas de microcontroladores. Microprogramación de la CPU y de los sistemas periféricos.*

Respecto al programa analítico los contenidos del TP1 son:

- *1-Introducción a los Sistemas Embebidos. Caracterización y componentes de Hardware y Software de un sistema embebido.*
- *2-Tipos de arquitecturas. Implementaciones: 8bits/32Bits. Von Neumann / Harvard. Buses. Puertos de Entrada – Salida..*
- *3- Programación en lenguaje de ensamblador. Programación en C. Ambientes de Desarrollo. Facilidades para la programación de periféricos.*

Contenido del TP1:

El TP1 contendrá en primer lugar un cuestionario sobre los temas dados según 1,2 y 3 y que los alumnos deben re-leer directamente de la bibliografía. En segundo lugar una guía con ejercicios para que los alumnos aprendan a usar las herramientas y hagan sus primeros pasos en base a ejemplos. Estos ejemplos proveen una base para el repaso del lenguaje C, el manejo de los puertos Entrada-salida del MCU y el manejo de las herramientas de desarrollo. En tercer lugar, el TP posee un ejercicio entregable a resolver con kit (o simulador).

Cómo evaluamos el TP1:

El cuestionario se evaluará en el examen escrito es decir se evalúan los conceptos aprendidos.

En cuanto a la parte práctica, por medio de la entrega de ejercicios (con informe) evaluamos las siguientes capacidades:

- *Manejar el lenguaje C para programar un MCU y su relación con el lenguaje ensamblador.*
- *Manejar las herramientas para hacer un nuevo proyecto, editar, compilar, reconocer errores y bajar a placa o simular. Depurar, ejecutar paso a paso, corregir errores de funcionamiento.*
- *Conocer la arquitectura del MCU y como trabajar con los puertos entrada-salida.*
- *Comprender la temporización de un programa mediante retardos bloqueantes y cómo afecta al tiempo de respuesta de los eventos respecto al usuario.*
- *Comprender los problemas típicos de interconexión de los puertos con periféricos externos digitales o analógicos.*
- *Comprender como desarrollar una interfaz de usuario mínima para interactuar con el MCU (Leds y pulsadores).*

EJERCICIOS PARA ENTREGAR c/ INFORME (Vencimiento: 7/4)**Control de periféricos externos con puertos de entrada/salida.**

1. Se desea conectar 8 diodos LED de diferentes colores al puerto B del MCU. Realice el esquema eléctrico de la conexión en Proteus. Calcule la resistencia serie para que la corriente por cada terminal no supere la capacidad de corriente de cada salida individual y de todas las salidas en funcionamiento simultáneo.
2. Realice un programa que permita activar los bits del puerto B uno a la vez, en el siguiente orden: 0-1-2-3-4-5-6-7 y luego que vuelva a empezar la secuencia.
3. Realice una función que implemente un retardo de 100ms para adecuar la visualización al usuario. Simule con el debugger cuantos ciclos de reloj consume el algoritmo de retardo y ajuste el tiempo en ms, suponiendo un reloj del sistema de 8MHz. Compare su solución con la función de biblioteca `_delay_ms`.

4. Se desea conectar un pulsador a una entrada digital del MCU. Muestre los posibles esquemas para conectar un pulsador y determine el algoritmo más adecuado para detectar cuando el pulsador se presiona y cuando se suelta en cada caso. ¿Qué es una resistencia de pull up? ¿y de pull down? Investigue sobre el efecto de rebote que producen los pulsadores e implemente un método para eliminar este efecto por software (puede encontrar información útil en <http://www.ganssle.com/debouncing.htm>).
5. Mediante el pulsador cambie el sentido de activación de la secuencia anterior con un pulso en el bit 0 del puerto C (simule el presionar y soltar el pulsador). Es decir, el usuario puede cambiar a la secuencia: 7-6-5-4-3-2-1-0 y viceversa cada vez que presiona y suelta el pulsador.
6. Saque conclusiones sobre el funcionamiento del programa, sobre las ventajas y desventajas de utilizar un retardo bloqueante y cómo afecta el tiempo de respuesta del programa ¿qué sucede si se deja presionado constantemente el pulsador?

El informe a entregar debe contener (*Ver ejemplo de informe.pdf en web/descargas/reglamento*):

- **Interpretación:** trate de expresar con sus palabras que interpreta de los enunciados por ejemplo: ¿qué se debe hacer? ¿qué no se debe hacer? ¿qué datos dispone? ¿qué datos necesita?, ¿qué periféricos hay involucrados? ¿qué eventos hay involucrados? ¿qué acciones se solicitan? entre otras. Realice una interpretación para cada requerimiento. Si surgen dudas, consulte!!! Una mala interpretación lleva a una mala solución.
- **Resolución del problema:** trate de explicar claramente cuáles son los razonamientos que aplica para llegar a las distintas soluciones que cubren los requerimientos de los problemas. Algunos puntos a considerar en la explicación:
 - Describir el funcionamiento de los periféricos y su configuración para resolver los problemas.
 - Explicar mediante pseudocódigo los algoritmos **propuestos** para cada ejercicio, describa la arquitectura de software utilizada, cuáles son las tareas y la temporización de las mismas. Indique cómo es la descomposición en funciones.
- **Validación:** Realice una explicación sobre la simulación. Adjunte capturas de pantallas de la simulación para justificar que cumple con los requerimientos. Muestre capturas del debugger o de los ciclos de reloj para justificar que se cumple la temporización del programa. Fundamente las conclusiones.
- **Código:** Muestre el Código en C luego de haber realizado el modelado y la explicación. Recuerde utilizar comentarios para documentar el mismo e indentar adecuadamente. El programa deberá coincidir con el pseudocódigo definido en el paso anterior.