

Organización de Computadoras Entregable final	Facultad de Ingeniería y Tecnologías Universidad Católica del Uruguay
Nombre:	Fecha:

Entregable Final

La finalidad de este trabajo es integrar varios de los conceptos y habilidades adquiridas durante el curso. Se trabajará sobre el microcontrolador PIC en lenguaje assembler.

Objetivo

El objetivo del proyecto es desarrollar el software de un dispositivo instalable en campo que registre la temperatura y almacene un histórico de medidas tomadas cada un intervalo fijo de tiempo. Una vez retirado, el dispositivo tendrá la capacidad de transmitir las medidas almacenadas a través del puerto serial.

Por simplicidad, el software se desarrollará y probará en la placa de desarrollo. El sensor de temperatura se simulará utilizando el potenciómetro conectado a la placa.

Detalles

Conversión A/D y Buffer circular (41 pts)

- Se utilizarán únicamente los 8 bits más significativos del conversor A/D.
- Las medidas se tomarán cada 10 segundos, y se deberán guardar las últimas 10 medidas tomadas utilizando un buffer circular.
- Las medidas no se pueden perder al desconectar el equipo, por lo que se deberán guardar en memoria no volátil. El equipo debe seguir registrando medidas al conectarse nuevamente.

Protocolo entre el PIC y el PC (32 pts)

Los parámetros de la comunicación serial serán 8N1 a 9600bps. El PIC responderá a los comandos que le lleguen desde el PC.

- Cuando el PIC reciba el carácter 'A' deberá responder con el valor actual de temperatura.
- Cuando el PIC reciba el carácter 'H' deberá responder con toda la lista almacenada de temperaturas, ordenadas desde la más reciente a la más antigua. Para facilitar la visualización de la lista en un terminal, el PIC enviará un carácter de salto de línea al final de cada temperatura.

En todos los casos, la temperatura se representará en formato hexadecimal mediante dos caracteres ASCII.

Otros (27 pts)

- Inicializar entradas y salidas correctamente (9 pts)
- Lograr conversiones de 8 bits (9 pts)
- Utilizar interrupciones periódicas para determinar los 10 segundos (9 pts)

Informe, comentarios y defensa

El informe sobre el proyecto final **no es obligatorio**, siempre y cuando se entregue el **código perfectamente formateado y comentado**. Si se desea, se puede entregar un breve informe sobre el proyecto, por ejemplo, con el diagrama de flujo usado. Es útil a veces notificar a los docentes de alguna funcionalidad faltante, que funciona bajo ciertas condiciones o que está limitada por alguna razón, así como de las posibles causas o potencial solución.

La defensa es de carácter opcional y podrán asistir únicamente aquellos estudiantes que no hayan conseguido

el derecho a examen con las notas de parcial y el presente proyecto final. En la defensa se harán preguntas sobre el funcionamiento del presente programa, así como de conceptos teóricos o generales del PIC. Un buen desempeño en la defensa posibilita el derecho a examen.

Adicionales

El proyecto tiene un total de 100 puntos pero no tienen que necesariamente conseguirse con la especificación planteada anteriormente. Entendemos que algunos individuos pueden tener interés en explorar otros componentes de la placa y por lo tanto se decidió agregar los siguientes 5 objetivos secundarios, completamente opcionales, a total discreción de cada equipo.

Investigar el uso del buzzer de la placa (24 pts)

La placa de desarrollo cuenta con un buzzer conectado al PIC que permite hacer sonidos. Utilícelo para generar 2 alertas sonoras, de distinto tono, para cuando se hace una consulta por el puerto serie. Un tono se utilizará cuando la consulta recibida es válida y otro tono para cuando es incorrecta. La opción correcta no sólo admite el carácter mencionado en el cuerpo principal del obligatorio sino también su contraparte en minúscula. La duración y frecuencia de los tonos será elegida por los estudiantes, pero intente que el sonido de la consulta incorrecta suene reprobatorio. Está terminantemente prohibido utilizar rutinas de larga duración en la rutina de atención a la interrupción de cualquier dispositivo.

Normalización de temperatura (32 pts)

Supongamos que el sensor de temperatura tiene un rango de 0°C a 100°C, que corresponden a 0V y 5V respectivamente. Al consultar con el carácter 'A' se responderá el valor actual de la temperatura como se especifica en el texto original del obligatorio. Sin embargo, cuando se consulta con una letra 'a' minúscula, se debe responder con el valor, decimal en grados Celsius, por ejemplo:

3.16 V resulta en una conversión $162_d = 0xA2$ y corresponde a 63°C.

Al enviar 'A' por el puerto serie se debe responder "A2" mientras que al enviar 'a' la respuesta será "63°C"

No se puede resolver evaluando todos los casos uno por uno y produciendo la respuesta deseada.

Investigar el uso del LED RGB de la placa (8 - 32 pts)

La placa cuenta con un led RGB que se puede utilizar para dar una alerta visual del rango de temperatura que nos encontramos. La idea es mostrar distintos colores en función de la temperatura registrada. Para ninguna de las siguientes opciones está permitido evaluar todos los casos uno por uno (switch)

Opción 1 (8 pts)

Dividir el resultado conversión (el valor de la temperatura) en al menos 3 zonas, de menos a más temperatura: Azul, Verde y Rojo.

Opción 2 (16 pts)

Dividir la escala en 3 zonas: Verde, Amarillo y Rojo.

Opción 3 (32 pts)

Dividir la escala en 256 zonas en un gradiente de color desde el Verde puro en 0, haciéndose más amarillento hasta el amarillo puro en 128 y haciéndose más rojizo hasta el rojo puro en 255.

Display de 7 segmentos para mostrar datos de control (16 pts)

Utilizar el display de 7 segmentos para mostrar los siguientes datos en formato hexadecimal de dos dígitos:

- 1) Total de medidas registradas desde el inicio.
- 2) Cantidad de medidas guardadas (hasta 10)
- 3) Valor actual de la temperatura.
- 4) Tiempo restante hasta registrar la próxima medida.
- 5) Valor de la última medida guardada.

Cada una de las opciones se activa mediante los botones RB0 hasta RB4.

Verificación de consistencia de memoria no volátil (16 pts)

La idea es utilizar un mecanismo más robusto que el valor de una posición de memoria para determinar la validez de los valores guardados en la memoria. Es muy común usar un valor verificador que resulta de operar con todos los datos registrados, pudiendo así establecer cuando los valores leídos de memoria son válidos o no. Los datos que se contemplan en esta verificación son todos aquellos que pueden haber sido leídos o escritos para registrar medidas, incluido cualquier contador o puntero, pero sin contemplar el byte de verificación (o asumirlo 0x00)

Este adicional consiste en implementar el chequeo de la validez de los datos calculados mediante el uso de un CRC simplificado de 8 bits (o CRC8) Existe vasta información en Internet de cómo implementar este algoritmo de verificación. Tenga en cuenta que hay algunos algoritmos que calculan el resultado utilizando solamente operaciones entre bytes (más lento pero ocupa menos memoria de programa) y otros que utilizan “lookup tables” (más rápido pero ocupa un poco más de memoria de programa) La elección del algoritmo está a cargo de los estudiantes.

Luego de esto, cuando la rutina esté implementada, cualquier cambio en la memoria no volátil será acompañada por un cálculo del CRC. Dicho valor se guardará en memoria no volátil, con el resto de los datos. Al cargarse los datos de la memoria se calculará el CRC y se comparará con el CRC guardado en memoria. Si los CRC coinciden, se considerará que los valores de la memoria no volátil están sanos. De lo contrario, se procederá a descartar lo registrado en la memoria.