RESUMEN DE TRABAJOS PRÁCTICOS Y TEMAS DICTADOS EN LA MATERIA OPTATIVA “LA PC COMO CONTROLADORA DE PROCESOS”

2010

FaMAF

ALUMNOS:

Gómez Adrian

Volarik Mariano

Monti Raúl

PROFESORES:

Marqués Carlos

Zaninetti Walter

A continuación se hace una breve descripción de cada trabajo práctico realizado durante el semestre. Todos ellos fueron presentados previamente por los profesores de la cátedra, junto con el contenido teórico necesario para realizarlos.

**Práctico 1**

Detectar disponibilidad y correcto funcionamiento del Puerto Paralelo

Es recomendable para cada programa que utilice un puerto, primero

comprobar la disponibilidad del puerto antes de comenzar a utilizarlo. Dando

lugar a un programa más robusto.

Una comprobación sencilla e (casi) infalible, está dada por los siguientes pasos:

1 - Enviar un dato cualquiera almacenado en la variable 'first\_data\_out' al puerto

de datos del LPT.

2 - Leer el puerto de datos del LPT y almacenar el valor en la variable 'first\_data\_in'

3 - Comparar los valores 'first\_data\_out' y 'first\_data\_in', si ambos son iguales,

repetir el paso 1 con el nuevo valor 'snd\_data\_out' = COMPLEMENTO(first\_data\_out).

Realizar nuevamente la comprobación. Si ambos son iguales, se puede concluir

que el puerto paralelo se encuentra disponible para usarlo, de lo contrario,

no está disponible o no funciona correctamente.

El práctico 1 de la materia, implementa esta comprobación.

**Práctico 2**

Leer el registro STATUS del LPT1 y escribir el valor leído en el registro DATA

Mediante la rutina

dato\_in = inportb(LPT1\_DIR + 1)

leemos el registro STATUS del LPT1 (0x378 + 1) y almacenamos el valor leído en la

variable 'dato\_in'.

El bit 7 (Bit BUSY) del registro STATUS es invertido, lo cual lo invertimos mediante

una máscara con el valor 0x80.

Luego tomamos los 4 bits de mayor peso del valor leído (y con bit 7 complementado),

mediante la división por 16.

Con el array tabla[], convertimos el valor para ser mostrado un display 7 segmentos

conectado al registro DATA.

Finalmente enviamos el valor tabla[dato\_in] al registro DATA.

**Práctico 3**

Comunicación entre 2 PC's mediante el puerto serie.

Si dos PC’s conectadas a través del puerto serie, acuerdan un protocolo de transmisión,

(bit de paridad, longitud del mensaje, Divisor Latch), pueden establecer una comunicación

sencilla.

El cable cruza las líneas Tx con Rx y viceversa, de modo que la línea de salida de un host

sea la línea de entrada del otro. De igual manera la línea Rx.

Ambos HOST setearon el puerto serie con los siguientes valores:

- 9600 Baudios

- 8 bits de mensajes

- Sin paridad

- 1 Stop bit

El programa entra en un bucle, verificando si hay datos en el buffer de entrada y mostrándolos

en pantalla en el caso encontrar algunos.

En el mismo bucle detecta si el buffer de salida está lleno, si es así, transmite los datos.

**Práctico 4**

Control de un motor PaP mediante el puerto paralelo

Un motor PaP consta de 4 bobinas que son controladas desde afuera a través de un cable por bobina.

A diferencia de un motor DC, se necesita una secuencia de impulsos distintos para girar el rotor.

Existen distintos modos para hacer girar el motor (PASO NORMAL, WAVE DRIVE, MEDIO PASO),

nosotros implementamos el modo PASO NORMAL que consiste en la siguiente secuencia

Motor:

Bobina 1

Bobina 2 (X) Bobina 4

Bobina 3

Secuencia de encendido de bobinas:

1 - Bobina 1 y Bobina 2

2 - Bobina 2 y Bobina 3

3 - Bobina 3 y Bobina 4

4 - Bobina 4 y Bobina 1

Sentido anti horario: Secuencia = 1, 2, 3, 4, 1, 2, 3, 4, 1, ...

Sentido horario: Secuencia = 1, 4, 3, 2, 1, 4, 3, 2, 1, ...

Si los cables del motor se conectan a los 4 bits de menor peso del registro DATA del LPT,

la secuencia se corresponde con los siguientes valores:

1 - 0xc

2 - 0x09

3 - 0x03

4 - 0x06

Estos valores están almacenados en el arreglo 'comando[]'.

**Práctico 5**

Control de un motor PaP mediante el puerto serie

En éste proyecto, disponemos un driver para el motor PaP a través del puerto serie.

Una vez seteado el puerto serie, los comandos que acepta el drive permiten definir

el sentido de giro y la cantidad de pasos.

**Práctico 6**

Lectura de una termocupla a través del puerto paralelo

Disponemos de un driver con una termocupla.

El driver se comunica a través de la PC mediante el puerto serie.

Para leer la temperatura medida por el sensor, enviamos el valor 't' y

luego el valor ' ' (espacio) para leer el número correspondiente a las

centenas del valor del sensor, otro ' ' más para las decenas y por último

otro ' ' para las unidades.

Una vez obtenida el valor de las centenas, decenas y unidades, procedemos

a mostrarlo por la pantalla.

**Práctico 7**

Lectura a través del puerto paralelo el valor de un conversor AD

Un generador de señales se conecta a la entrada del conversor AD y la salida

del AD se conecta al registro DATA del puerto paralelo.

Cada cierto intervalo de tiempo, se lee el puerto de datos del LPT y se

imprime el valor en pantalla, que corresponde con la señal digitalizada

del conversor AD.

**Práctico 8**

Conexión de un conversor DA al registro de datos del puerto paralelo para generar

una señal analógica con forma de diente de sierra

Disponemos de un conversor DA con una resolución de 8 bits conectado al puerto paralelo.

Enviamos a través del registro de datos los valores:

0 1 2 .. 255 0 1 2 .. 255 ...

La salida del DA se conecta a un osciloscopio, donde podemos ver que la señal analógica

generada tiene forma de diente de sierra.

**Práctico 9**

Conexión de un conversor DA al registro de datos del puerto paralelo para generar

una señal analógica con forma de triangulo

Análogo al práctico 8, con la diferencia que los valores de salidas son:

0 1 2 .. 255 254 ... 2 1 0 1 2 3 .. 255 254 ... 2 1 0 ....

generando así otra forma de señal analógica.

**Práctico 10**

Conexión de un conversor DA al registro de datos del puerto paralelo para generar

una señal analógica con forma de seno

Análogo a los prácticos 8 y 9. Los valores de salida conforman una señal sinusoidal

128\*sin(1) + 128, 128\*sin(2) + 128, 128\*sin(3) + 128, ...

La señal está deslazada hacia arriba (+ 128), para localizar el 0 en la mitad del rango

de valores de resolución del DA.

También está multiplicada por el factor 128, para cubrir todo el rango de resolución del DA.

**Práctico 11**

Implementación de un conversor AD Simple Rampa usando el puerto paralelo y un conversor DA

Implementamos usando la PC un conversor AD con algoritmo Simple Rampa.

El algoritmo es el siguiente:

1 - Se toma una muestra de la señal de entrada (simulando un módulo Sample And Hold)

2 - Se inicializa una variable contador con el valor 0

3 - El valor de la variable contador se transforma a través del DA en una señal analógica

y se compara con el valor de la muestra.

Si son distintas, se incrementa el valor del contador, y se repite el paso Nº 3.

Si son iguales, el valor del contador se corresponde con el valor analógico de la señal

de entrada, terminando el proceso de digitalización y tomando una muestra para comenzar

otro proceso.

Si la variable contador alcanzó su rango máximo sin que la comparación se cumpla,

se toma éste valor máximo como el valor digital de la muestra de entrada y se termina el proceso.

**Práctico 12**

Implementación de un conversor AD con algoritmo de Aproximaciones Sucesivas

usando el puerto paralelo y un conversor DA.

Análogo al Práctico 11, pero con distinto algoritmo.

Se utiliza el algoritmo de aproximaciones sucesivas, que tiene mayor velocidad

de conversión que el algoritmo de Simple Rampa.

El algoritmo es el siguiente:

1 - Se toma una muestra 'in\_value' de la señal de entrada (analógica)

2 - 'value' = b7 b6 b5 b4 b3 b2 b1 b0 = 0 (Digital)

3 - i = 7

3.1 Se setea el valor bi en 1 y se compara ANALOGICO('value') con 'in\_value'

Si ANALOGICO('value') == 'in\_value' -> Fin del proceso

Si ANALOGICO('value') > 'in\_value' -> bi = 0

Sino bi = 1.

Si i == 0 -> Fin

Se repite el paso 3.1 con i = i - 1

Cuando el proceso termina, 'value' tiene el valor digitalizado correspondiente

al valor analógico 'value\_in'.

**Práctico 13**

Implementación de un osciloscopio sencillo usando el puerto paralelo

Se utiliza un conversor AD. El valor leído del conversor, es graficado en pantalla, como

una función de voltaje, en el dominio del tiempo.

Se toman muestras cada cierto intervalos y se grafica cada una.

Formando la imagen en la PC correspondiente a la señal de entrada.

**Práctico 14**

Mantener una temperatura deseada, mediante el sistema de control Si / No

Es una implementación del sistema de control más sencillo.

A través de un sensor de temperatura, con interface por el puerto serie, medimos la

temperatura interna de un habitáculo pequeño.

Dentro del mismo habitáculo, se encuentra una lámpara conectada también al puerto

serie, que con 2 comandos podemos prenderla y apagarla.

Para mantener una temperatura deseada, primero se lee el sensor de entrada, si

la temperatura leída es menor que la temperatura deseada, enviamos el comando

para encender la lámpara un instante fijo, si la temperatura es mayor, la lámpara

permanece apagada.

Se repite el proceso indefinidamente, logrando mantener la temperatura del habitáculo a

niveles muy cercano al deseado.

**Práctico 15**

Mantener una temperatura deseada, mediante el sistema de control PWM

El práctico se asemeja al práctico 14, pero posee un sistema de control más refinado.

La lámpara ahora no permanecerá encendida (o apagada), un instante de tiempo fijo.

La duración del tiempo de encendido (o apagado), se calcula dinámicamente en base

a la diferencia entre temperatura efectiva y temperatura actual.

Es un sistema de control PWM (Pulse-width modulation), el pulso de encendido / apagado,

no tiene una duración fija, depende de parámetros.

**Práctico 16**

Seguidor de luz

Tenemos un dispositivo, que gira sobre su propio eje, con 2 sensores de luz, conectado

mediante una interface al puerto serie de la PC.

La lectura de los sensores, devuelve valores directamente proporcionales a la cantidad de luz

que incide sobre ellos. De ésta forma, podemos saber aproximadamente cuál es el ángulo

de incidencia del rayo de luz sobre ello.

Si los valores leídos de los sensores, entonces incide con mayor magnitud la luz en uno

de ellos. Si es así, mediante comandos enviados por el puerto serie, podemos hacer girar

su propio eje, logrando mantener el haz de luz, perpendicular a ellos (Cuando ambos sensores

devuelven el mismo valor).

**Trabajo Final**

Controlador de sistema de emisión de señales digitales y adquisición para sistema de resonancia magnética nuclear.

Consistió en realizar el software pertinente para el manejo de una placa compuesta por un programador de pulsos, un generador de señales digitales dds, y un conversor A/D tipo flash de 2 canales de 12 bits. El trabajo consiste en una interfaz para el usuario, el programa que maneja los chips mencionados, y un ploteador para representar los resultados gráficamente en pantalla. La comunicación con la placa se realizó vía puerto paralelo aplicando así los conocimientos adquiridos durante los prácticos de la materia.