Control de Temperatura Universal

Autor: Gaspar Daniel Gómez. – email: gaspargomez975@hotmail.com

Introducción:

Cuando nos enfrentamos ante el diseño de una aplicación de control automático, es muy frecuente cruzarnos con el control de la variable "Temperatura". Esta aplicación surgió inicialmente como un control de temperatura para un invernadero, pero como cada cultivo requiere de una temperatura que difiere del resto, el curso original de mi proyecto se vio desviado hacia una nueva aplicación, un controlador capaz de mantener la temperatura ambiente dentro de un rango preestablecido activando un elemento calefactor y desactivándolo de acuerdo a lo que requiera cada cultivo.-

Viéndolo desde este nuevo punto de vista este proyecto podría dedicarse a una infinidad de aplicaciones que requieran mantener una temperatura dentro de un rango determinado, como ser: Acondicionadores de ambientes, enfriadores, hornos, incubadoras, etcétera. Por este motivo se le agrego el titulo de "Universal".-

Objetivos:

Para empezar con este diseño me planteé los siguientes objetivos:

- ✓ Uso sencillo e intuitivo.-
- ✓ Monitorización constante.-
- ✓ Bajo costo.-
- ✓ Versátil.-
- ✓ Tamaño reducido.-

En que consiste el dispositivo?

El equipo cuenta con un "Cerebro central" que es el microcontrolador 68HC908QB8 de la firma Freescale y los periféricos mas importantes utilizados son: Un display LCD de 2 líneas con 16 caracteres cada una, un teclado de 4 teclas, un sensor de temperatura LM35 y la interfaz de potencia mediante relé.-

El microcontrolador: Como se dijo anteriormente el control de todo el sistema esta a cargo del microcontrolador 68HC908QB8 de solo 16 pines. Éste, a su vez, esta dividido en módulos (Conversor A/D, KBI (*"KeyBoard Interrupt"*), Puertos de I/O, modulo de comunicaciones seriales, entre otros.) por lo que la programación del mismo esta orientada a cada uno de estos módulos. El entorno de desarrollo utilizado fue CodeWarrior, este es mi primer proyecto en código C.-

Módulos utilizados:

- ✓ Conversor Analógico/Digital de 10 bits de resolución.-
- ✓ KBI (*"KeyBoard Interrupt"*).-
- ✓ Puertos de I/O.-
- ✓ TIM (*"Timer Interfaz Module").-*

Se selecciono este microcontrolador porque, para su reducido tamaño, es muy versátil, es un dispositivo de bajo costo y posee todos los módulos que para este diseño se necesitan, lo que lo convierte en idóneo para este desarrollo.-

Desarrollo: El primer paso fue comunicar el microcontrolador con el display LCD. Se utilizo el bus de datos con un tamaño de 4 bits ya que genera un ahorro notable de pines. Esto se hizo para generar funciones propias para el control de dicho display y optimizar los tiempos de envío de datos/comandos, como así también, de habilitación del display. Algunas de las funciones más importantes son:

LCD_enviar(): Esta función convierte un dato a enviar de 8 bits en dos envíos de 4 bits (Recordemos que cuando el display se comunica en bus de 4 bits se debe enviar primero los 4 bits mas significativos y, seguidamente, los cuatro bits menos significativos).-

```
//*****************************
void LCD_enviar (void){
aux1 = dato;                // El dato se guarda en
asm{                        // un registro auxiliar.-
  LSR aux1                  // Se desplaza a derecha
  LSR aux1                  // el registro auxiliar 4 veces.-
  LSR aux1
  LSR aux1
}
auxsend=PTB;                // Para no modificar el valor del
auxsend=auxsend&0xF0;       // puerto se lo trabaja en otra
auxsend= auxsend|aux1;      // variable, una vez "adaptado"
PTB=auxsend;                // se modifica con el nuevo valor
habilitar();                // y se habilita al LCD para tomarlo.-
dato=dato&0x0F;             // Se elimina la parte alta de "dato"
auxsend=PTB;                // y se repite el proceso.-
auxsend=auxsend&0xF0;
auxsend= auxsend|dato;
PTB=auxsend;

habilitar();
}
```

Antes de hacer un llamado a esta función se debe precargar el valor a enviar en la variable “dato”.-

Habilitar(): Si bien es una función que no ocupa mucho código, la cantidad de veces que se la invoca justifica su creación, la principal ventaja de esta función es el ahorro de memoria.-

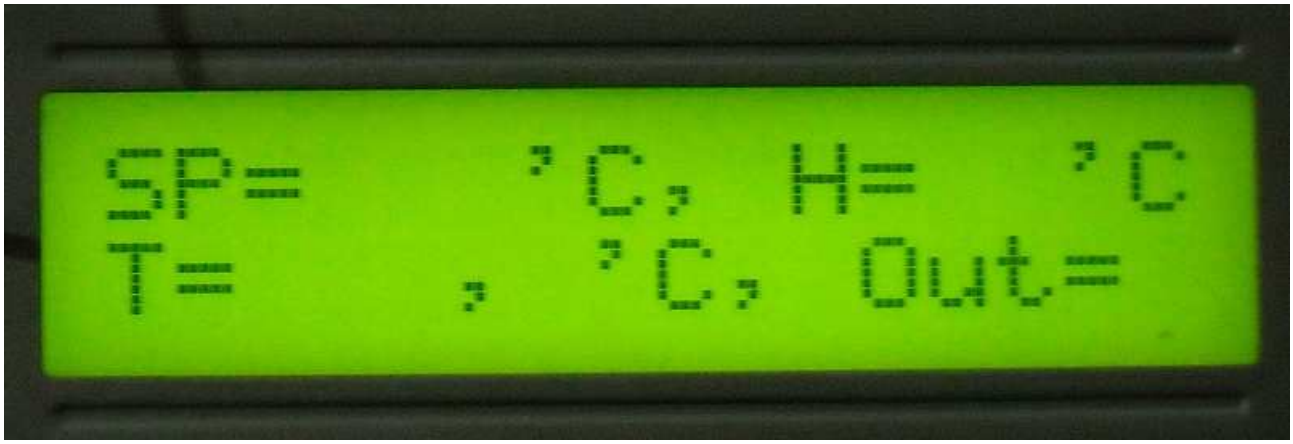
```
//*****************************
void habilitar (void){
  PTB_PTB4 = 1;       // Habilitacion en 1
  delay_5ms();        // Retarda 5ms
  PTB_PTB4 =0;        // Habilitacion en 0
}
```

linea1() y linea2(): Estas funciones son utilizadas para seleccionar la línea del LCD en la que se desea escribir. Para apuntar a la línea 1 se debe cargar “*dato*” con 40’h.-

```
//***************************
void linea2 (void){
  PTA_PTA1=0;       // Linea RS en bajo
  delay_5ms();      // indica que se envia
  dato=0xC0;        // un comando
  LCD_enviar();
  PTA_PTA1=1;
}
```

Una vez establecida la comunicación se optimizaron los tiempos para una visualización rápida y fiel.-

Para esta aplicación se trabajo al LCD en modo “plantilla”, este modo se caracteriza por instalar sobre el display los textos que este deberá mostrar y, a continuación, se lanza el programa principal que se encarga de apuntar a las direcciones de memoria donde serán ubicados los registros y datos obtenidos que se deseen mostrar.-



El display muestra:

- **SP:** *“Set Point”*. Es la temperatura que el usuario desea mantener.-
- **H:** *“Histéresis”.* Indica en nivel de histéresis aceptable para esa temperatura preestablecida.-
- **T:** *“Temperatura”*. Muestra la temperatura actual tomada por el sensor.-
- **Out:** *“Salida”.* Indica el estado actual de la salida (1= Activa, 0= Inactiva).-

El LM35 y el Conversor A/D:

Paso seguido se instalo el sensor de temperatura LM35 en el pin RA0 que pertenece al canal 0 (cero) del conversor A/D. Este microcontrolador posee un conversor A/D de aproximaciones sucesivas de 10 bits de resolución, más que suficiente para el caso que en esta oportunidad nos compete.

Para convertir el valor entregado por el sensor, se genero un contador interno que se utiliza para transformar de hexadecimal a milivots, se lo multiplica por 10 dando así el valor de temperatura real, ya que **el sensor LM35 entrega 10mV por grado centígrado**.

Una vez obtenido el valor de temperatura se lanza un contador que divide el valor obtenido en unidades, decenas, centenas y unidades de mil. Con estos valores se generan los caracteres ASCII necesarios para enviarlos al LCD y así se hace la visualización de temperatura.-

El rango de temperatura que se pueden monitorear va desde 2ºC hasta los 120ºC, se planteo este rango debido a que el proyecto se orienta al control de la temperatura ambiente pero, haciendo unas modificaciones de software, se podrían alcanzar

temperaturas mayores. Si se modifica el hardware también se podrían medir temperaturas por debajo de los 0ºC.-

El Programa Principal:

El programa principal comienza configurando el conversor A/D para realizar una sola conversión y espera a que esta termine. Una vez tomada la muestra se evalua el valor registrado y se guarda.

Se procede al muestreo de los registros *SetPoint (SP) e Histéresis (H),* enviando al LCD las direcciones que corresponden a la posición de los datos a mostrar. Por ejemplo, en el caso del SP se envía el dato 84'h que indica al LCD que debe posicionar el cursor en la línea 1, carácter 4. Este valor puede variar, ya que en el software, se incluyó una rutina que detecta si los valores de mas alto peso (Unidades de mil y centena) están en 0 (cero), de ser así, no es necesario mostrarlos, entonces se ajusta la posición de escritura.-

Al estar listo el muestreo de *SP* y *H*, se pasa a mostrar los valores del segundo renglón o línea.

Como se dijo anteriormente el valor convertido ah quedado guardado en una variable, se "desmenuza" este valor en sus respectivos componentes unidad, decenas, centenas y unidades de mil, y se lo presenta en el LCD comenzando el envío (previo posicionamiento del cursor) por las unidades de mil hasta las decenas, ya que el cursor se auto incrementa con cada dato recibido. Una vez enviadas las decenas se envía el valor 2C'h que representa la coma en ASCII. Paso seguido se envían las unidades y ya queda mostrado el valor de temperatura actual.-

Ahora estamos en condiciones de evaluar si la temperatura actual se encuentra dentro del rango deseado. Este proceso se realiza tomando el valor del contador utilizado para la conversión y comparándolo con otro valor más: Valor 1 = *SP – H.* Si la temperatura se encuentra por debajo de "Valor 1" la salida se activa y comienza un bucle hasta que la temperatura llegue al valor *SP*, sino, se continua con el muestreo y actualización de datos en el display.-

Una vez concluido un ciclo de programa principal la visualización es la siguiente:

El Teclado:

Como se puede observar el programa principal no tiene relación alguna con los eventos del teclado. Esto sucede debido a que el modulo *KBI* de este microcontrolador genera una interrupción cuando sucede un evento en los pines asociados. Dentro de la rutina de interrupción se atiende al teclado y sus eventos. Dentro de la interrupción se

verifica que pin fue el causante y de ahí se desprenden las rutinas que atiendan al mismo. Cada vez que sucede una interrupción por teclado se envía un pulso a un buzzer conectado en RB7 que informa que el dato ha sido tomado.-

Designación de teclas y ubicación en el microcontrolador:

- ✓ **SP +** (RA2)
- ✓ **SP -** (RA3)
- ✓ **H +** (RA4)
- ✓ **H -** (RA5)

SP +: Genera un incremento en el registro *SetPoint.-*
SP -: Genera un decremento en el registro *SetPoint.-*
H +: Genera un incremento en el registro *Histéresis.-*
H -: Genera un decremento en el registro *Histéresis.-*

Todos los incrementos y decrementos controlan que las variables no salgan de sus respectivos rangos. Para *SetPoint* el rango va desde 2 hasta 120 y para *histeresis* desde 1 hasta 9.-

Al utilizar el teclado con interrupciones hacemos que el programa principal sea mas fluido y menos engorroso.-

Conclusiones: El dispositivo diseñado evalúa una temperatura, la compara con un rango predeterminado y actúa en consecuencia. Con estas simples palabras podemos ver que es fácilmente aplicable a cualquier sistema que requiera mantener una temperatura estable, dentro de cierto rango. Si miramos el dispositivo detenidamente podemos observar también que los objetivos han sido cumplidos en su totalidad. Los costos del equipo no son importantes si se comparan con lo que cuesta un controlador comercial.-

En cuanto al uso podemos decir que es más que sencillo ya que, en solo 4 teclas, podemos presetear el rango que necesitamos y el sistema se encargara del control. Si el lector observa con detenimiento el plano adjunto del equipo se dará cuenta que solo queda libre un pin, ese pin llamado "*PTB5/**Tx**/AD9"* se dejo a propósito ya que a futuro planeo enviar la información registrada vía puerto serie a una PC donde, utilizando un software, se pueda llevar un registro mas detallado de las temperaturas medidas.-

Referencias:

MC68HC908QB8 Data Sheet M68HC908 Microcontroller Rev.1 6/3/2005.-
LM35 Precision Centigrade Temperature Sensors Data Sheet Nov. 2000.-