

Revisiones	Fecha	Comentarios
0	07/09/05	

En esta nota, les proponemos un simple y vistoso controlador industrial/campestre, el cual permite graficar en un display color en tiempo real el avance de las muestras de temperatura y humedad entregadas por un SHT-71, a la vez que monitorea que éstas no excedan determinados límites. De excederse esos límites, el controlador encenderá un ventilador o un calefactor, según corresponda, cuya velocidad o intensidad serán proporcionales a la magnitud del exceso.

Tanto la medición actual como un pequeño histórico, pueden observarse en una página web, donde además pueden cambiarse los valores de los límites fijados de temperatura y humedad.

Historia:

- En la CAN-031 les presentamos los nuevos displays color LCD de 320x240.
- En las CAN-035 y CAN-036 vimos la forma de utilizar un controlador de Epson, el S1D13706, para que se encargue de todo lo relacionado con el manejo del display, mientras que nosotros nos limitamos a decirle como lo tiene que hacer, y darle la información a mostrar.
- En la CAN-037 mejoramos la performance con un clock más rápido.
- En la CAN-028 vimos como usar los sensores de humedad y temperatura SHT-71.

Descripción del proyecto

El hardware está basado en lo anteriormente desarrollado en las notas de aplicación mencionadas, realizamos pequeñas modificaciones para que pudieran convivir entre sí, dado que en algunas hemos tomado algunos atajos como escribir directamente en todo un port en vez de manejar los bits individuales, para ganar performance. La interfaz con el SHT-71 se porta fácilmente a R3000, el cual empleamos debido a que el controlador del display funciona a 3,3V como máximo.

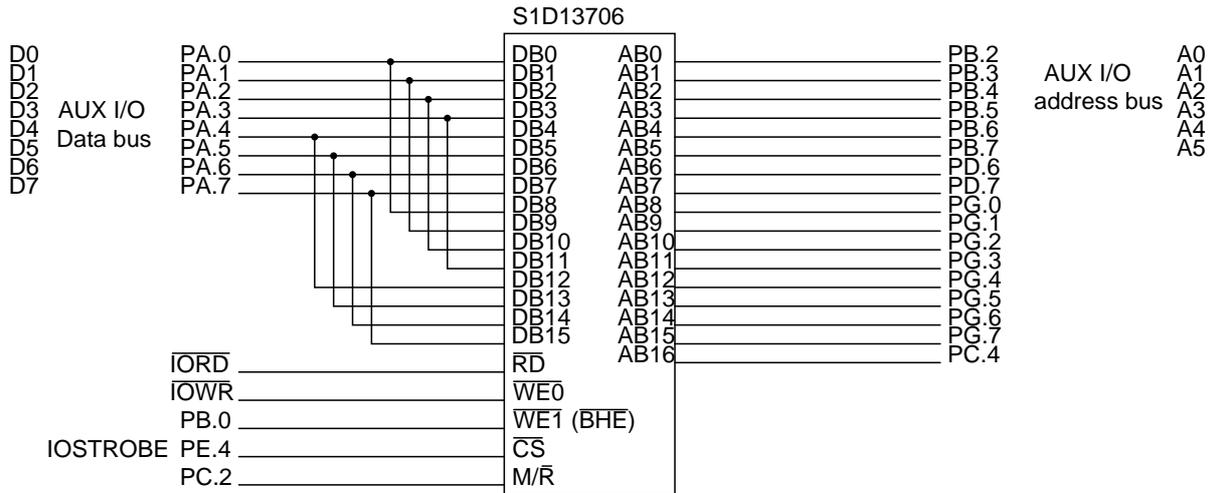
Como comentáramos en la introducción, la idea del proyecto es tomar muestras periódicas de humedad y temperatura, y compararlas contra dos límites preestablecidos de cada una, uno alto y uno bajo. Mientras los valores obtenidos estén dentro de los rangos autorizados, mantendremos tanto ventilador como calefactor apagados. Si alguna de las señales sale fuera de rango, actuaremos sobre el ventilador o calefactor, según corresponda calentar o enfriar. Para esta demo, no hemos tomado acción sobre la humedad, si bien graficamos ambas en pantalla, solamente actuamos ante variaciones de temperatura. Como también comentáramos, el accionamiento del dispositivo corrector será proporcional, es decir, el ventilador girará más rápido cuanto más se haya excedido la temperatura, y el calefactor calentará más cuanto más por debajo del límite inferior se esté. Para que ésto funcione en la vida real, necesitaremos de algún dispositivo que responda al valor medio de la tensión generada por PWM. En esta demo utilizamos MOSFETs, que nos permiten colocar alguna resistencia y/o motor de continua; sin que con esto pretendamos calentar todo un silo o enfriar un galpón industrial, sino cubrir los alcances de una nota de aplicación.

En cuanto a la interfaz de control, queda como tarea para el lector el desarrollo de la misma.

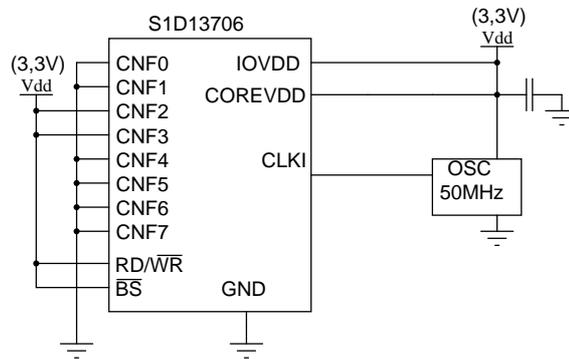
Hardware

El hardware que se armó se basa en una placa desarrollada para la evaluación del controlador, la cual se describe en el Comentario Técnico CTC-027 y puede adquirirse en Cika Electrónica. Los diagramas esquemáticos coinciden con lo detallado en CAN-035, con la salvedad del clock de 50Mhz (según CAN-037) y el agregado del SHT-71 y las salidas de control del calefactor y ventilador. Por conveniencia, los reproducimos aquí con las modificaciones apropiadas:

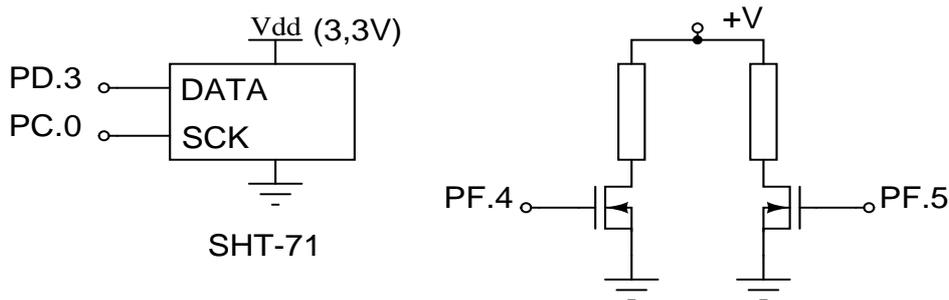
CAN-039, Control de temperatura/humedad con display color, SHT-71 y Rabbit



Detalle de conexionado con el oscilador de 50MHz:



Conexionado del SHT-71 y las salidas PWM



Software

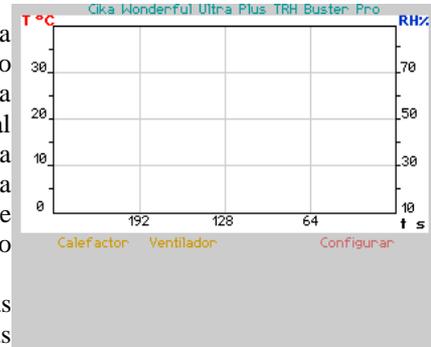
En las notas de aplicación mencionadas pueden obtener toda la información de cómo controlar el display y el SHT-71. En esta sección nos limitaremos a describir los detalles particulares de este proyecto. Por simplicidad, incluiremos sólo aquellas partes del listado que sean relevantes, la totalidad del mismo puede consultarse en el archivo que acompaña a esta nota de aplicación.

Graficaremos dos señales, una de color rojo (temperatura) y otra de color azul (humedad). A su vez el display presenta una grilla con las unidades y muestra los límites. El gráfico de ambas señales se desplaza de izquierda a derecha, como si fuera un graficador; para mantener el dibujo por detrás de la grilla, como si fuera un

osciloscopio, evitaremos dibujar la señal sobre aquellos puntos que correspondan a la grilla o los límites. Para que las señales se vean claramente, haremos una cruz en cada posición graficada, lo cual tiene la ventaja de generar un trazo grueso cuando las variaciones son pequeñas, como es de esperarse, y permitir la visualización conjunta de la grilla o límites y la señal, en puntos donde coinciden. El efecto de desplazamiento hacia la izquierda lo obtenemos almacenando la información en un buffer circular, borrando la información vieja (ploteando en color de fondo), agregando la información nueva a la cabeza del buffer y volviendo a plotear, con lo cual el gráfico resulta automáticamente desplazado.

Para simplificar las cosas, diseñamos el "frente" con un programa gráfico y plasmamos en él la grilla, las escalas y los nombres, de modo de no necesitar siquiera manejar textos, como puede observarse en la imagen que acompaña este párrafo. Los límites los graficamos al comenzar, y cada vez que recibamos un comando de actualización por la página web. El defecto de esta elección, es que la grilla resulta borrada en los lugares originales de los límites cambiados, lo cual puede subsanarse re-ploteando la grilla o la imagen completa, cosa que no hemos hecho.

Mediante imágenes (dibujos) de LEDs encendido y apagado tomadas de las samples de Rabbit, indicaremos en el display el estado de las salidas, superponiéndolas a la imagen anterior. Mediante la animación (envío alternado de una u otra imagen) podemos simular el parpadeo de estos "LEDs" de nuestro "panel frontal"



Algoritmos

Para ubicar un punto dentro del área destinada, nos acomodamos a nuestra nueva "pantalla", en la cual seguimos teniendo una distancia de 320 bytes entre puntos verticales (aunque disponemos de menos puntos), pero horizontalmente tenemos menos puntos y un desplazamiento. El área elegida resultó ser de 256x127, con un desplazamiento que sitúa al punto (0;0) del gráfico en las coordenadas (287;142) del display. Dado que la escala del gráfico se distribuye de menor a mayor de derecha a izquierda y de abajo hacia arriba, debemos realizar una inversión tanto horizontal como vertical (el punto (0;0) del display está en la posición superior izquierda y el del gráfico en la inferior derecha). Tomando en cuenta los offsets de nuestro gráfico en particular, y las escalas utilizadas, las ecuaciones para calcular un punto en pantalla para graficar temperatura serán:

$$address = 320 \cdot y + 287 - x; y = 142 - \left(\frac{127}{40} \cdot T \right)$$

Mientras que las de humedad serán:

$$address = 320 \cdot y + 287 - x; y = 142 - \left(\frac{127}{80} \cdot (RH - 10) \right)$$

Por conveniencia, calculamos primero $y=f(T)$ e $y=f(RH)$, las guardamos en el buffer circular, y luego al plotear calculamos $address=f(x,y)$, donde x es el número de muestra tomada. La cruz se hace plotear además $address+1$, $address-1$, $address+320$ y $address-320$, siempre que quepan dentro del gráfico.

A la vez, guardamos los valores originales en otro buffer, y al juntar 256 muestras las bajamos a la flash. Dado que esto es una demo, no nos preocupamos demasiado por llenar el "disco", lo cual puede ocurrir si se pretende emplear esto en la vida real.

La utilización del PWM es más que sencilla, para inicializarlo y mantener apagadas las salidas, simplemente hacemos:

```
pwm_init(10000L);           // setea frecuencia 10KHz
pwm_set(0,0,0);            // apaga 0
pwm_set(1,0,0);            // apaga 1
```

luego, cuando detectamos que la temperatura excede el límite, calculamos el valor a setear basándonos en un mínimo absoluto (por debajo de él el ventilador no gira) y un máximo correspondiente a la resolución del generador de PWM (1024) y el punto de máxima velocidad, que en este caso hacemos coincidir con un límite de alarma. El calefactor tendrá su máximo en 0°. La siguiente es la tarea que actualiza las activaciones y alarmas a cada segundo:

```
costate {
    // main control loop
```

CAN-039, Control de temperatura/humedad con display color, SHT-71 y Rabbit

```
if(curT>=thi){
  heater=0;
  fan=(int)(FAN_MIN+((FAN_MAX-FAN_MIN)*(curT-thi))/(t2hi-thi));
  if(fan>FAN_MAX)
    fan=FAN_MAX;
  if(curT>=t2hi)          //alarma
    alarm=1;
  }
  else {
    fan=alarm=0;
    if(curT<=tlo){
      heater=(int)(HEATER_MIN+((HEATER_MAX-HEATER_MIN)*(tlo-curT))/10);
      if(heater>HEATER_MAX)
        heater=HEATER_MAX;
    }
    else heater=0;
  }

  pwm_set(0,fan,0);
  pwm_set(1,heater,0);
  while(ambisi)          // no modifica buffer mientras está mostrando en web
    yield;
  datastuff(curT,curRH);
  waitfor(DelayMs(1000)); // update every second
}
```

En el listado completo que acompaña esta nota puede consultar todas las demás funciones.

Una simulación del modelo terminado, similar a lo que en la vida real pudieron apreciar los asistentes al Seminario Rabbit Buenos Aires 2005, se vería más o menos, así:

