Cika Electrónica

Nota de Aplicación: CAN-048
Título: **Sensores remotos de Humedad y Temperatura SHT-71 con TRW-2.4G**
Autor: Sergio R. Caprile, Senior Engineer

Revisiones	Fecha	Comentarios
0	07/11/05	

En esta oportunidad, desarrollamos código para poder emplear los SHT-71 como sensores remotos de temperatura y humedad. Los mismos reportarán a un sistema central mediante transceptores TRW-2.4G. Los remotos operan sobre PIC16F630, empleando lo desarrollado en CAN-046 y CAN-047. El master, opera sobre un módulo Rabbit, aprovechando el desarrollo de la CAN-045.

Para detalles sobre hardware y las rutinas de bajo nivel de SHT-71 y/o TRW-2.4G, remítase a las notas de aplicación mencionadas. El contenido de esta nota se refiere a la operación del sistema en sí, y al código necesario para poder realizar la tarea mencionada.

Desarrollo propuesto

El sistema central, basado en Rabbit, se ocupará de interrogar a cada uno de sus remotos y realizar la linealización de los sensores, a fin de obtener una mayor precisión. Cada sensor remoto obtiene el resultado de la medición en forma de un número entero de 16-bits, el cual puede procesar para la toma de decisiones locales. Al recibir el requerimiento del master, procederá a entregar los valores que tenga corrientes, junto con el estado interno de operación, a fin de que el master pueda determinar la operatividad del remoto.
A los fines prácticos, los remotos funcionan sobre una máquina de estados, la cual se encarga de realizar la tarea de inicialización o reset del sensor, el inicio de la medición, la espera de los resultados y tomar la decisión de resetar al sensor si transcurre demasiado tiempo sin obtener resultados. Si bien hay diverisdad de formas de llevar cuenta del tiempo, elegimos una forma sencilla como contar la cantidad de iteraciones de la secuencia de instrucciones, a fin de no complicar esta nota con la inicialización y manejo de interrupciones de timers en PIC. El lector puede emplear el método que más le agrade si no gusta del utilizado o se le complica al agregar una mayor cantidad de tareas en los remotos.

A continuación, veremos el programa principal de los remotos:

```
        call TXdelay                ; Espera al arrancar antes de
        call trw_init               ; inicializar el módulo TRW-2.4G
        call trw_address            ; Setea la dirección del master (para responder)
        clrf shtstatus              ; inicializa máquina de estados
        clrf shttime                ; resetea timeout

main    call trw_setrx              ; Habilita recepción
mloop   call trw_getpacket          ; mensaje ?
        iorlw 0                     ; ?
        btfsc STATUS,Z
        goto nomsg                  ; no, mira el sensor
        call TXdelay                ; sí, espera que el master entre en recepción (3ms)
        movf shtstatus,W            ; estado
        movwf message
        movf temp,W                 ; última medición de temperatura
        movwf message+1
        movf temp+1,W
        movwf message+2
        movf humi,W                 ; última medición de RH%
        movwf message+3
        movf humi+1,W
        movwf message+4
        call trw_sendpacket         ; contesta
        call TXdelay                ; espera >3ms a que salga el paquete
        goto main
```

```
; esta porción de código no debe cruzar un espacio de 256 bytes (culpa de PCLATH y PCL)
nomsg   movlw HIGH nomsg              ; parte alta
        movwf PCLATH
        movf shtstatus,W              ; salta según estado
        addwf PCL,f
        goto shtrst                   ; reset del sensor
        goto shtmeat                  ; espera de temperatura
        goto shtmeah                  ; espera de humedad
        goto shtslp                   ; descanso entre mediciones

shtrst  clrf shtstatus                ; inicia
        call s_connectionreset        ; reset sensor
shtstrt call s_measure_t              ; mide temperatura
        btfss STATUS,C                ; reset si hay error
        incf shtstatus,f              ; pasa a estado de espera
        movlw toconst                 ; constante de tiempo de espera máxima
        movwf shttime+1
        goto mloop
shtmeat call chktmout
        btfsc STATUS,C                ; reset si hay timeout
        goto shtrst
        call s_measure_wait           ; mira si el sensor terminó
        btfsc STATUS,C                ; retorna si debe seguir esperando
        goto mloop
        call s_measure_get            ; obtiene resultados
        movf value,W
        movwf temp
        movf value+1,W
        movwf temp+1
        call s_measure_rh             ; mide humedad
        btfsc STATUS,C                ; reset si hay error
        goto shtrst
        incf shtstatus,f              ; pasa a estado de espera
        movlw toconst                 ; constante de tiempo de espera máxima
        movwf shttime+1
        goto mloop
shtmeah call chktmout
        btfsc STATUS,C                ; reset si hay timeout
        goto shtrst
        call s_measure_wait
        btfsc STATUS,C                ; retorna si debe seguir esperando
        goto mloop
        call s_measure_get            ; obtiene resultados
        movf value,W
        movwf humi
        movf value+1,W
        movwf humi+1
        incf shtstatus,f              ; pasa a estado de descanso
        movlw slconst                 ; constante de tiempo de descanso
        movwf shttime+1
        goto mloop
shtslp  call chktmout
        btfss STATUS,C                ; retorna si debe seguir esperando
        goto mloop
        clrf shtstatus
        goto shtstrt                  ; comienza un nuevo ciclo

chktmout:
        bcf STATUS,C
        decfsz shttime,f
        return
        decfsz shttime+1,f
        return
        bsf STATUS,C
        return
```

A continuación, un ejemplo de como el sistema central interroga a uno de los remotos y linealiza la información. Las rutinas de SHT-71 son las publicadas en la CAN-028, el código de TRW-24G utiliza la library desarrollada en la CAN-045.

```
        costate {
                printf("Sent poll\n");
                wfd i=trw_poll(raddr2,msg,100);                    // pide y espera
```

```
        if(i>0){
               humi_val=(float)(*(int *)(&msg[3]));         // obtiene y pasa a float
               temp_val=(float)(*(int *)(&msg[1]));         // obtiene y pasa a float
               calc_sht11(&humi_val,&temp_val);             // calcula
               printf("\ttemperatura: %5.1fC, humedad: %5.1f%%\n",temp_val,humi_val);
               waitfor(DelayMs(1000));
        }
        num++;
}
```