

Revisiones	Fecha	Comentarios
0	07/02/07	port de CAN-048

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 HT48E30, empleando lo desarrollado en CAN-056 y CAN-057. 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 resetear al sensor si transcurre demasiado tiempo sin obtener resultados. Si bien hay diversidad 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. El lector puede emplear el método que más le agrada 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                ; Esperar al arrancar antes de
        call trw_init               ; inicializar el módulo (devuelve A=0)
        mov a,0                     ; selecciona master
        call trw_address            ; Setea master address (para responder)
        clr shtstatus              ; inicializa máquina de estados
        clr shttime                 ; resetea timeout

main:   call trw_setrx               ; Habilita rx
mloop: call trw_getpacket           ; msg ?
        snz ACC.0                  ; A!=0, (sólo miro bit 0)
        jmp nomsg                  ; no, salteo
        call TXdelay               ; espera que el master entre en recepción
        mov a,shtstatus            ; estado del sensor
        mov messge,a
        mov a,temp                 ; última medición de temperatura
        mov messge[1],a
        mov a,temp[1]
        mov messge[2],a
        mov a,humi                 ; última medición de humedad
        mov messge[3],a
        mov a,humi[1]
        mov messge[4],a
        call trw_sendpacket        ; contesta
        call TXdelay               ; espera >3ms a que salga la Tx
        jmp main

```

CAN-058, Sensores remotos de Humedad y Temperatura SHT-71 con TRW-2.4G

```

nomsg:      mov a,shtstatus
            addm a,PCL                ; salta según estado
            jmp shtrst                ; reset del sensor
            jmp shtmeat               ; espera de temperatura
            jmp shtmeah               ; espera de humedad
            jmp shtslp                ; descanso entre mediciones

shtrst:
            clr shtstatus              ; inicia
            call s_connectionreset     ; reset sensor

shtstrt:
            call s_measure_t           ; mide temperatura
            snz C                      ; resetea sensor si hay error
            inc shtstatus              ; espera resultado
            mov a,toconst              ; constante de tiempo de espera máximo
            mov shttime[1],a
            jmp mloop

shtmeat:
            call chktmout
            sz C                      ; resetea si hay timeout
            jmp shtrst
            call s_measure_wait
            sz C                      ; regresa si debe esperar
            jmp mloop
            call s_measure_get         ; obtiene resultados
            mov a,value
            mov temp,a
            mov a,value[1]
            mov temp[1],a
            call s_measure_rh         ; mide humedad
            sz C                      ; resetea si hay error
            jmp shtrst
            inc shtstatus              ; siguiente
            mov a,toconst              ; constante de tiempo de espera máximo
            mov shttime[1],a
            jmp mloop

shtmeah:
            call chktmout
            sz C                      ; resetea si hay timeout
            jmp shtrst
            call s_measure_wait
            sz C                      ; regresa si debe seguir esperando
            jmp mloop
            call s_measure_get         ; obtiene resultados
            mov a,value
            mov humi,a
            mov a,value[1]
            mov humi[1],a
            inc shtstatus              ; pasa a estado de descanso
            mov a,sconst               ; constante de tiempo de descanso
            mov shttime[1],a
            jmp mloop

shtslp:
            call chktmout
            snz C                      ; regresa si debe seguir esperando
            jmp mloop
            clr shtstatus
            jmp shtstrt                ; inicia un nuevo ciclo

chktmout:
            clr C
            sdz shttime
            ret
            sdz shttime[1]
            ret
            set C
            ret

```

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.

```

constate {

```

CAN-058, Sensores remotos de Humedad y Temperatura SHT-71 con TRW-2.4G

```
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++;
}
```