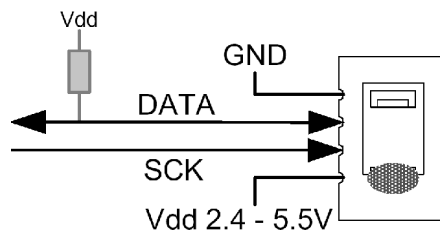


Revisiones	Fecha	Comentarios
0	07/02/07	port de CAN-047

Si bien existen numerosas alternativas para la medición de temperatura ambiente, la medición de la humedad relativa ambiente resulta ser algo complicada. En ambos casos, lograr una determinada precisión implica disponer de sensores caros y mucho cuidado en la sección analógica y la placa de circuito impreso; hecho que se ve potenciado si el sensor debe estar a una cierta distancia del circuito de procesamiento, y/o en ambientes hostiles. Los sensores combinados de humedad y temperatura ambiente SHT-71, desarrollados por la firma Sensirion, constan de un par de sensores y conversores A/D, circuitería de calibración, y compensación. La información es presentada por una interfaz serie, de modo que el sistema de medición se desentiende del tema ruidos, linealización analógica, calibración, e impedancias. Para más detalles, y código ejemplo en C, le recomendamos referirse a la CAN-028. En esta oportunidad, desarrollamos código para poder leer los sensores con los micros de Holtek, bajando notablemente el costo para aplicaciones que requieran gran cantidad de sensores remotos.

Hardware

La conexión es sumamente simple, requiere dos pines de I/O, uno con capacidad tristate u open collector para los datos; el otro entrega el clock que controla el timing de esos datos, para ambos sentidos de la comunicación



Software

El fabricante nos provee código en C como para ser compilado en un procesador compatible MCS51, sin demasiado esfuerzo, es posible portarlo a assembler para cualquier otra arquitectura, tarea la cual hemos desarrollado para Holtek, en la forma de una pequeña biblioteca de funciones.

El software a continuación se encarga de la lectura de la información provista por el sensor. No realizamos ningún tipo de procesamiento de la misma, tarea la cual dejamos a criterio del lector. En nuestro caso, y según veremos en la CAN-058, reportaremos este valor a un sistema central basado en Rabbit, el cual hará la linealización, corrección y presentación de los datos. Para tareas locales que no requieran gran precisión, el usuario puede procesar la información que colecta esta nota como un número entero de 16-bits, a su gusto.

Comenzamos por las funciones de lectura y escritura de un byte en el sensor.

```

; Envía un byte
; Dato en A
; Devuelve resultado en el flag de carry (C)
; C -> Error
s_write_byte:
    mov datas,a                ; Store data
    mov a,8                    ; 8 bits
    mov bitcntr,a
swbll: rlc datas                ; C = MSB (shift bit for output)

```

CAN-057, Sensores de Humedad y Temperatura SHT-71 con Holtek

```

        clr SCK                                ; CLK = 0
        nop
        clr DAT                                ; DATA = 0 (open-collector)
        nop
        set DATT                               ; pull-up
        snz C                                  ; if C, DATA = pull-up
        clr DATT                               ; otherwise DATA = 0
        nop
        set SCK                                ; CLK = 1
        sdz bitcntr
        jmp swbll
        clr SCK                                ; CLK = 0
        nop
        set DATT                               ; pull-up
        nop
        set SCK                                ; CLK = 1 (9th clk)
        rrca DATPORT                          ; check ack (DATA will be pulled down by SHT11)
        clr SCK                                ; CLK = 0
        ret

; Recibe un byte
; Devuelve el dato en A
s_read_byte:
        clr datas                              ; DATA = 0
        mov a,8                                ; 8 bits
        mov bitcntr,a
srbll:  set SCK                                ; CLK = 1
        nop
        rrca DATPORT                          ; read data in C
        rlc datas                              ; and into register (MSB first)
        clr SCK                                ; CLK = 0
        sdz bitcntr
        jmp srbll
        clr DAT                                ; DATA = 0 (open-collector)
        nop
        sz ACK                                 ; if need to ACK
        clr DATT                               ; DATA=0 (ACK)
        nop
        set SCK                                ; CLK = 1 (9th clk)
        nop
        clr SCK                                ; CLK = 0
        nop
        set DATT                               ; pull-up
        mov a,datas
        ret

```

Como habrán deducido, la selección de los pines a emplear se realiza mediante macros, las cuales pueden redefinirse dentro del programa:

```

; DAT debe ser el LSB, de otro modo hay que modificar el programa
DAT      EQU      PB.0
DATPORT EQU PB
DATT     EQU PBC.0
SCK      EQU PB.1

```

```
SHT_TMOU equ 2000
```

```

;
; addr  command  r/w
STATUS_REG_W EQU 006h ; 000 0011 0
STATUS_REG_R EQU 007h ; 000 0011 1
MEASURE_TEMP EQU 003h ; 000 0001 1
MEASURE_HUMI EQU 005h ; 000 0010 1
SHT_RESET    EQU 01Eh ; 000 1111 0

```

```

shtvars .section 'data'
datas  db ?
bitcntr db ?
value  db 2 dup(?)
chksum db ?
flags  db ?

```

```
ACK EQU flags.0
```

CAN-057, Sensores de Humedad y Temperatura SHT-71 con Holtek

Luego seguimos por el proceso de inicialización.

```

; communication reset: DATA-line=1 and at least 9 SCK cycles followed by transstart
;
; DATA: _____|_____|_____
;
; SCK : __|__|__|__|__|__|__|__|__|__|__|_____|_____|_____|_____
; Call first time, or when you sense errors
s_connectionreset:
    clr SCK                                ; CLK = 0
    mov a,9                                ; 9 SCK cycles
    mov bitcntr,a
scrll: set SCK                               ; CLK = 1
    nop
    clr SCK                                ; CLK = 0
    sdz bitcntr
    jmp scrll
; follow through transmission start
;generates a transmission start
;
; DATA: _____|_____|_____
;
; SCK : __|__|__|__|_____|_____
s_transstart:
    clr SCK                                ; CLK = 0
    nop
    set SCK                                 ; CLK = 1
    nop
    clr DAT                                 ; DATA = 0 (open-collector)
    nop
    clr DATT                                ; DATA=0
    nop
    clr SCK                                 ; CLK = 0
    nop
    set SCK                                 ; CLK = 1
    nop
    set DATT                                ; pull-up
    nop
    clr SCK                                 ; CLK = 0
    ret

```

A continuación, un juego de rutinas que nos permite iniciar una medición, interrogar al sensor el estado de la misma, y obtener el resultado.

```

; inicia medición de temperatura
s_measure_t:
    call s_transstart                       ; transmission start
    mov a,MEASURE_TEMP
    jmp s_write_byte                        ; send command to sensor

; inicia medición de humedad
s_measure_rh:
    call s_transstart                       ; transmission start
    mov a,MEASURE_HUMI
    jmp s_write_byte                        ; send command to sensor

; Llamar hasta que regresa sin carry (C=0), o transcurre demasiado tiempo
s_measure_wait:
    rrca DATPORT                            ; read data in C
    ret

; Obtiene resultados
s_measure_get:
    set ACK                                  ; ACK
    call s_read_byte                        ; read the first byte (MSB)
    mov value[1],a
    call s_read_byte                        ; read the second byte (LSB)
    mov value,a
    clr ACK                                  ; no ACK
    call s_read_byte                        ; read checksum (8-bit)
    mov chksum,a
    ret

```

CAN-057, Sensores de Humedad y Temperatura SHT-71 con Holtek

Dado que el sensor, según la medición de que se trate, puede demorar más de 200ms en realizarla, y sabiendo que nos lo indicará poniendo el pin DATA en estado bajo, lo que haremos es interrogarlo periódicamente; esta es la razón por la cual partimos la tarea en tres subrutinas, de modo que el programa principal puede iniciar una medición, realizar sus tareas normales e ir interrogando periódicamente al sensor, para finalmente obtener el resultado cuando éste le indica que está disponible:

```
initsn call s_connectionreset      ; inicializa al sensor
      ; ...
      call s_measure_t             ; inicia medición de temperatura
      sz C                        ; verifica si hubo error (reset) o espera
      jmp hubounerror             ; OK, empezó la medición
      ; ...

      call s_measure_wait         ; pregunta si ya terminó
      sz C                        ; verifica respuesta
      jmp hayqueesperar           ; espera, verifica si ya esperó demasiado
      call s_measure_get          ; obtiene resultado
      mov A,value                 ; y lo guarda en su destino
      mov temp,A
      mov A,value[1]
      mov temp[1],A
```

Las rutinas desarrolladas fueron escritas teniendo en cuenta las especificaciones del fabricante, para 4MHz de clock. La cantidad de calls anidadas se comprobó para un HT48E30. Para otras aplicaciones, se sugiere al interesado revisar que se cumplan los tiempos detallados en la hoja de datos del SHT-71 y la cantidad de calls permitidos por el hardware stack del micro de Holtek utilizado.

El archivo adjunto contiene la totalidad de los listados.