



de KITS *Application Note*

AN9 - 8 Channel Temperature Display

Oleh: Tim IE

Temperatur / Suhu merupakan salah satu parameter input yang paling sering dijumpai pada sistem kontrol. Catatan aplikasi kali ini membahas tampilan untuk 8 sensor suhu dengan menggunakan :

- 1 unit DT-51 MinSys Ver 3.0
- 1 unit DT-51 KND
- 2 unit DT-51 I²C ADDA
- 2 unit Temperature Sensor 7 Segment pada DT-51 KND (Keypad & 7 Segment Display) digunakan untuk menampilkan 8 channel sensor suhu secara bergantian. Hubungan antar board & DT-51 ditunjukkan pada gambar 1. Sebagai contoh hanya digunakan 2 sensor.

Beberapa hal yang harus diperhatikan :

- Hubungan antara DT-51 MinSys dengan DT-51 I²C ADDA dan DT-51 KND dapat dilihat pada buku manual masing-masing.
- Tegangan 5V DC dapat diambil dari Port Control DT-51 MinSys Ver 3.0. Perhatikan polaritasnya, jangan sampai terbalik.
- Board #0 DT-51 I2C ADDA
Jumper alamat (J1-J3) diset pada alamat 0 (default), J4 dan J5 untuk resistor pull up dipasang.
- Board #1 DT-51 I2C ADDA
Jumper alamat (J1-J3) diset pada alamat 1, J4 dan J5 untuk resistor pull up tidak dipasang.
- Hubungkan kabel sensor #0 ke board 0 DT-51 I2C ADDA (merah +, hitam -, dan putih AI0)
- Hubungkan kabel sensor #1 ke board 1 DT-51 I2C ADDA (merah +, hitam -, dan putih AI3)
- Setelah semua siap maka download file demo7.hex ke DT-51 MinSys Ver 3.0

Maka DT-51 KND akan menampilkan suhu setiap channel secara bergantian. Contoh channel 0

0 = 0 2 7 °C

kemudian channel 1

1 = 0 2 8 °C

demikian seterusnya sampai channel 7.

Hubungan antara channel dengan Analog Input (AI) DT-51 I2C ADDA dapat dilihat pada tabel berikut :

Channel	Analog Input
0	AI0 Board #0
1	AI1 Board #0
2	AI2 Board #0
3	AI3 Board #0
4	AI0 Board #1
5	AI1 Board #1
6	AI2 Board #1
7	AI3 Board #1

Pada contoh hanya dipasang 2 sensor suhu, yaitu channel 0 dan channel 7. Analog input yang tidak terpakai ada baiknya dihubungkan dengan ground supaya menampilkan 0 0 0.

Sensor suhu

Software

Software untuk aplikasi ini cukup sederhana, dengan menggunakan rutin-rutin yang telah tersedia untuk DT-51 KND maupun I2C ADDA.

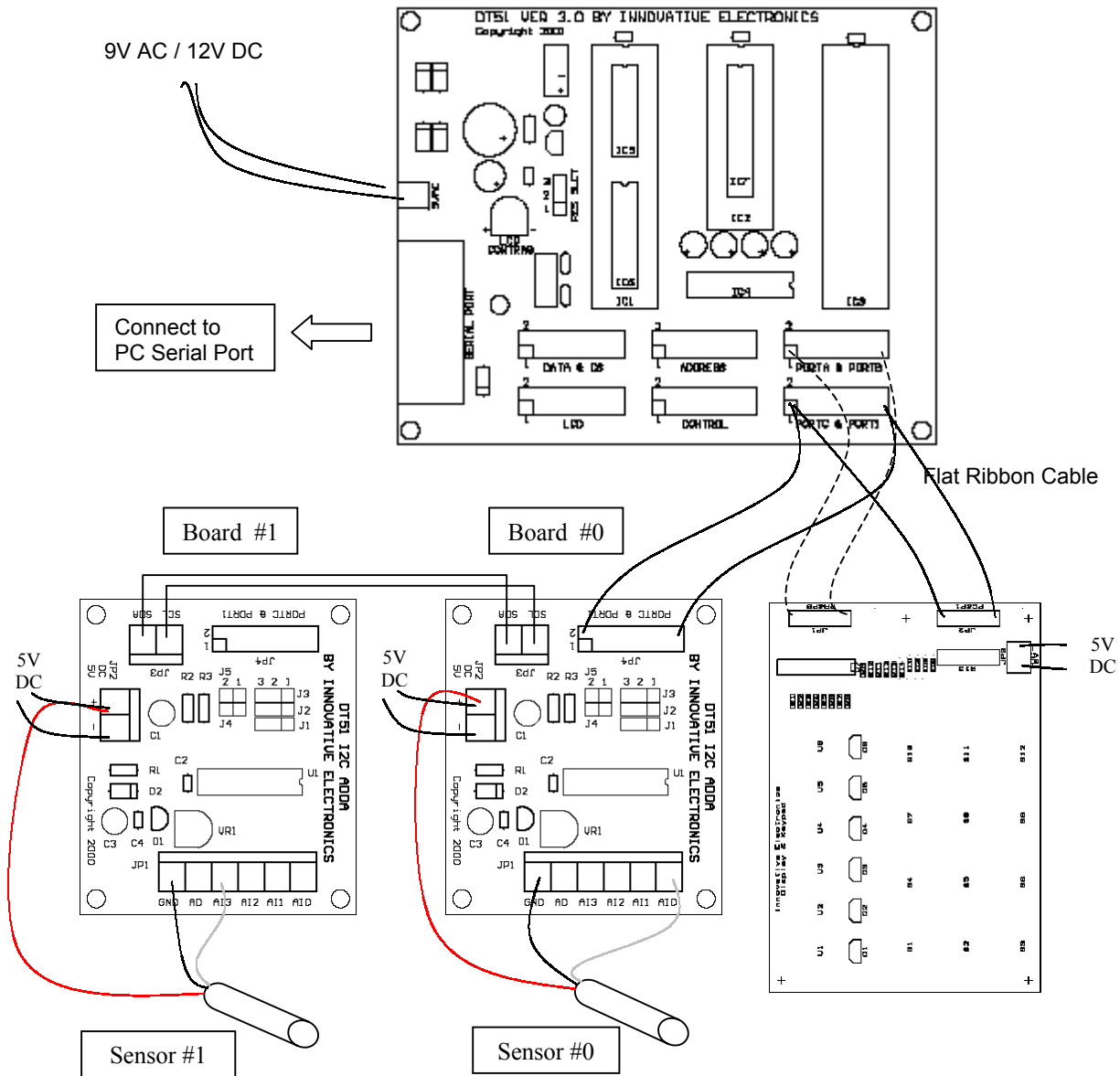
Listing program DEMO7.ASM dapat dilihat pada halaman 2. Output sensor berupa tegangan 10 mV / °C, jangkauan sensor 0 - 100°C. Sehingga tegangan output sensor berkisar 0 - 1 volt. Jangkauan input I2C ADDA ialah 0-2,5 volt. Jadi sensor bisa langsung dihubungkan ke analog input.

Konfigurasi analog input yang digunakan ialah single-ended input, dengan persamaan konversi ADC untuk kode output N (biner) :

$$N = \frac{V_{AIN}}{V_{REF}} \times 256$$

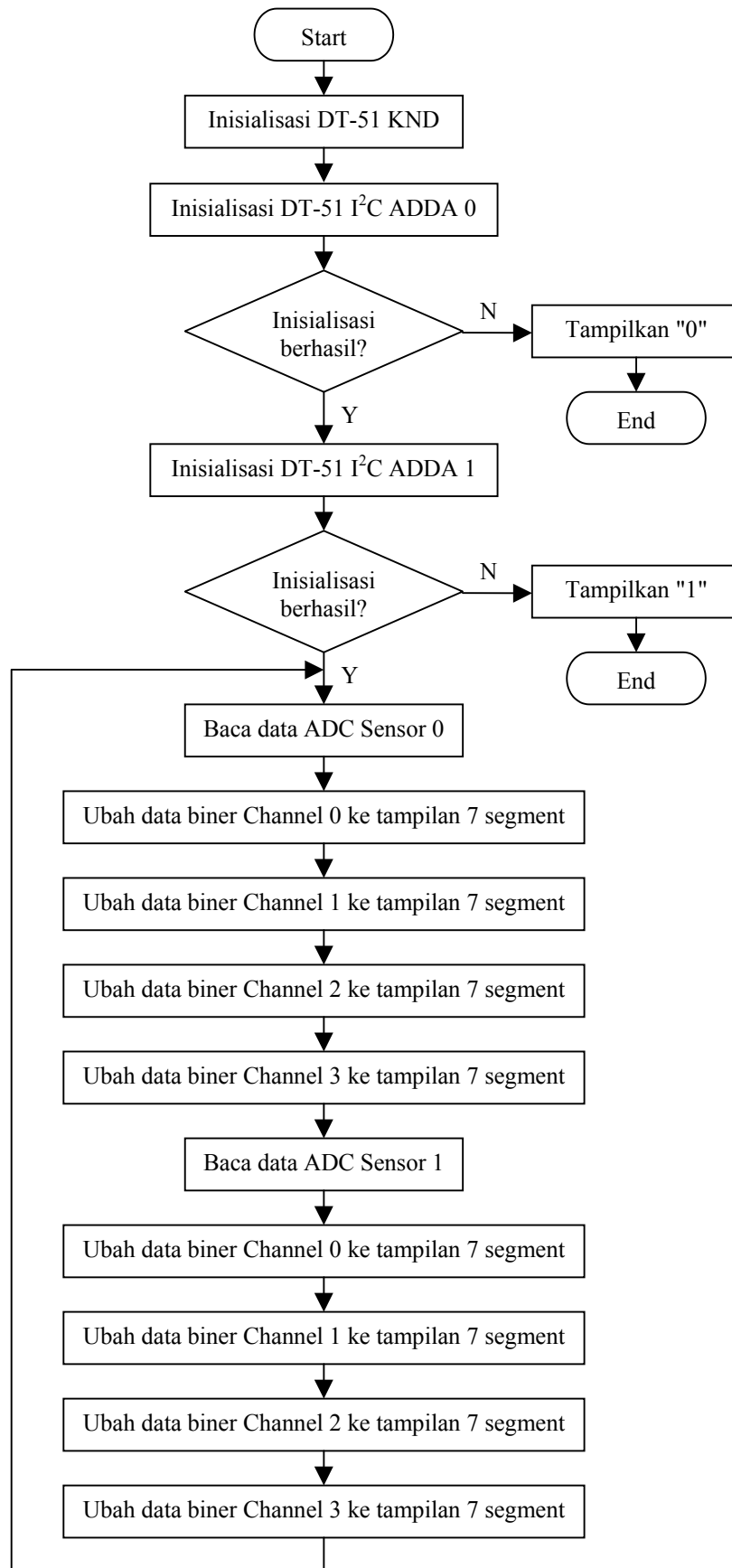
Tegangan referensi V_{REF} diset pada 2,56 volt agar resolusi tiap bit menjadi 10 mv, sehingga tidak diperlukan perhitungan tambahan pada program.

Rutin **BINTO7SMAN** digunakan untuk mengubah bilangan biner menjadi desimal kemudian ditampilkan ke 7 Segment.



Gambar 1. Hubungan Antar Board & Temperature Sensor

Flowchart program dapat dilihat pada gambar 2.



Gambar 2. Flowchart Program

Listing Program DEMO7.ASM (juga terdapat pada AN9.zip):

```
$MOD51
$title(DT-51 I2C ADDA AN-8 Temp Display)

;-----
;Sample Program Using DT-51 KND Routine
;    Without External Interrupt 1
;-----

REG6          EQU          0006H
REG7          EQU          0007H

HUND          DSEG
DATA          22H
TENONE       DATA          23H

CSEG
ORG           4000H
LJMP         Start

ORG           400BH
LJMP         Scanning

ORG           4100H
$INCLUDE (KNDINT.ASM)
$INCLUDE (ADDA.ASM)

;-----
;Convert Bin To 7 Segment Display Code
;Input  : A <- Temp. Bin
;Output : DisTemp <- Display Temperature
;-----
BINTO7SMAN:  PUSH          ACC
              PUSH          PSW
              MOV           B,#100
              DIV          AB
              ACALL         WRITE7S
              MOV           A,#10
              XCH           A,B
              DIV          AB
              ACALL         WRITE7S
              MOV           A,B
              ACALL         WRITE7S
              POP           PSW
              POP           ACC
              RET

;-----
;          Delay Proc.
;-----
Delay:       PUSH          REG6
              PUSH          REG7
              MOV           R2,#0FFH
Del:         MOV           R3,#0FFH
              DJNZ         R3,$
              DJNZ         R2,Del
              POP           REG7
              POP           REG6
              RET

Start:      MOV           P1,#0FFH
              MOV           SP,#53H
```

```

        LCALL      InitKND
        LCALL      CursorOFF
        LCALL      VirtualDis
        LCALL      ClrDisMem

I2CADDA_0:  MOV      Flag,#00H
            MOV      Ch0,#00H
            MOV      Ch1,#00H
            MOV      Ch2,#00H
            MOV      Ch3,#00H
            MOV      A,#00H
            MOV      Mode,#0
            MOV      Channel,#0
            SETB     AutoInc
            LCALL    InitADDA
            JB       InitDone,I2CADDA_1
            MOV      A,#0
            ACALL    WRITE7S
            AJMP     $

I2CADDA_1:  MOV      A,#1
            LCALL    InitADDA
            JB       InitDone,NxtStep

ErrMessage: MOV      A,#1
            ACALL    WRITE7S
            AJMP     $

NxtStep:
Module:    MOV      R5,#0
;          MOV      R7,#0
Ulang:    MOV      R6,#4
          MOV      R7,#0
          MOV      A,R5
          LCALL    ReadADC

Loop:
          LCALL    ClrDisMem
          MOV      A,R5
          MOV      B,#4
          MUL      AB
          ADD      A,R7
          LCALL    WRITE7S
          MOV      A,#18H
          LCALL    WRITE7S
          CJNE     R7,#0,Channel1
          MOV      A,Ch0
          LCALL    BINTO7SMAN
          AJMP     Next

Channel1:  CJNE     R7,#1,Channel2
          MOV      A,Ch1
          LCALL    BINTO7SMAN
          AJMP     Next

Channel2:  CJNE     R7,#2,Channel3
          MOV      A,Ch2
          LCALL    BINTO7SMAN
          AJMP     Next

Channel3:  MOV      A,Ch3
          LCALL    BINTO7SMAN

Next:     MOV      A,#19H
          LCALL    WRITE7S
          LCALL    Delay
          INC      R7
          DJNZ     R6,Loop
          INC      R5
          CJNE     R5,#2,Ulang
          AJMP     Module
          END

```