

# DT-AVR DT-AVR *Application Note*

## AN74 – Econo Digital Thermometer

Oleh: Tim IE

DS1822 merupakan termometer digital buatan Dallas Semiconductor dengan antarmuka 1-Wire<sup>®</sup>, divais ini hanya memerlukan 1 pin port I/O mikrokontroler. Dengan menggunakan divais ini, membuat termometer digital menjadi lebih mudah dan ekonomis. Karena divais ini tidak memerlukan ADC, tidak memerlukan rangkaian pengkondisi sinyal, dan hanya memakai 1 pin port I/O mikrokontroler. Bahasa yang digunakan adalah C dengan bantuan CodeVisionAVR<sup>®</sup>.

Komponen yang diperlukan:

- 1 bh DS1822 Econo 1-Wire<sup>®</sup> Digital Thermometer
- 1 bh DT-AVR Low Cost Micro System / Low Cost Nano System
- 1 bh LCD karakter 16 x 1 (kompatibel dengan driver HD44780)

Adapun blok diagram sistem secara keseluruhan adalah sebagai berikut:

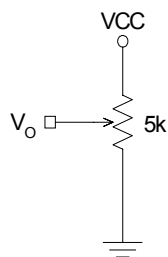


Gambar 1  
Blok Diagram AN74

Hubungan antara modul-modul tersebut adalah sebagai berikut:

DT-AVR Low Cost Micro System / Low Cost Nano System	LCD Karakter
GND	GND
V <sub>cc</sub>	V <sub>cc</sub>
-	V <sub>0</sub> *
PortB.0	RS
PortB.1	R/W
PortB.2	E
PortB.4	DB4
PortB.5	DB5
PortB.6	DB6
PortB.7	DB7

Tabel 1  
Hubungan DT-AVR Low Cost Micro System / Low Cost Nano System dengan LCD Karakter



**Gambar 2**  
Hubungan Pin V<sub>O</sub> LCD ke VR 5k ohm

DS1822	DT-AVR Low Cost Micro System / Low Cost Nano System
V <sub>DD</sub>	V <sub>CC</sub>
DQ	PortB.3
GND	GND

**Tabel 2**  
Hubungan DT-AVR Low Cost Micro System / Low Cost Nano System dengan DS1822

Pin LCD yang tidak tertulis pada tabel 1 tidak digunakan, karena menggunakan komunikasi data 4-bit. Sedangkan pin LCD yang diberi tanda \* (V<sub>O</sub>) dihubungkan ke VR 5k ohm sesuai Gambar 2 dan digunakan untuk mengatur kontras LCD.

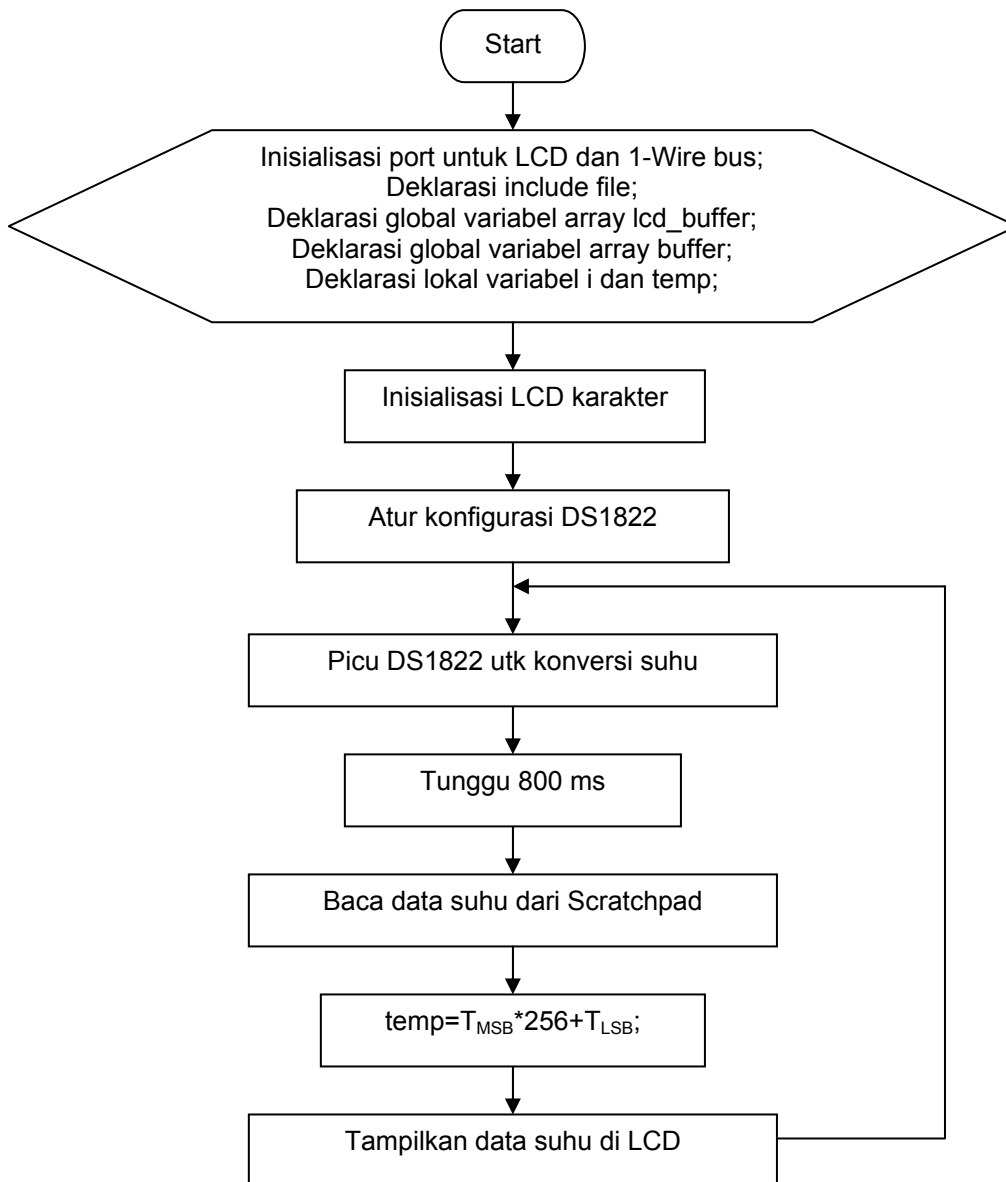
Setelah semua rangkaian dan sumber tegangan terhubung dengan tepat, programlah "anm3g3.C" (dengan meng-*compile/make* program tersebut terlebih dahulu pada *project* "ANm3g3.prj") ke DT-AVR Low Cost Micro System dengan AVR In System Programmer.

Untuk DT-AVR Low Cost Nano System atau tipe mikrokontroler AVR<sup>®</sup> yang lain:

1. Ubah tipe mikrokontroler pada menu Project – Configure – C Compiler – Chip (misalnya AT90S2313).
2. Ubah baris `#include <mega8535.h>` sesuai dengan tipe mikrokontroler yang digunakan (misalnya 90s2313.h).
3. *Compile/Make* ulang program tersebut.

Agar dapat menggunakan DT-AVR Low Cost Nano System diperlukan CodeVisionAVR<sup>®</sup> versi penuh, bukan evaluasi. Karena pada CodeVisionAVR<sup>®</sup> versi evaluasi, untuk chip AT90S2313 ukuran programnya dibatasi sampai dengan 512 word (keterangan ini terdapat di README.TXT bawaan CodeVisionAVR<sup>®</sup>).

**F**lowchart dari sistem ini adalah sebagai berikut:



**Gambar 3**  
**Flowchart Program**

**P**rogram akan diproses sebagai berikut:

1. Program menentukan port yang dipakai untuk LCD dan 1-wire bus, yaitu portB.[0..2] & portB.[4..7] untuk LCD dan portB.3 untuk 1-wire bus.
2. Deklarasi *include file* yang berisi rutin-rutin *built-in* yang akan digunakan, yaitu sebagai berikut:  
File lcd.h → untuk rutin-rutin LCD,  
File delay.h → untuk rutin-rutin delay,  
File stdio.h → untuk rutin-rutin I/O standar C compiler,  
File 1wire.h> → untuk rutin-rutin 1-wire bus.
3. Deklarasi variabel global, yaitu:  
lcd\_buffer merupakan array bertipe char, digunakan untuk data tulisan yang akan ditampilkan di LCD,  
buffer merupakan array bertipe unsigned char, untuk menyimpan data scratchpad DS1822.
4. Deklarasi variabel lokal, yaitu:  
i bertipe unsigned char, akan digunakan sebagai indeks array sekaligus tanda suhu (+ atau -),  
temp bertipe int, akan digunakan untuk data suhu yang telah diolah.

5. Kemudian program melakukan inialisasi LCD. Lalu mengatur konfigurasi DS1822 dengan alur berikut:  
 Pertama tulis konfigurasi byte pada scratchpad DS1822:  
 Inialisasi divais 1-wire → tulis 0xCC (lewati ROM) → tulis 0x4E (perintah tulis scratchpad) → tulis batas atas alarm (+125) → tulis batas bawah alarm (-55) → tulis byte konfigurasi (0x7F untuk resolusi 12-bit).  
 Kemudian kopi byte konfigurasi dari scratchpad 1822 ke EEPROM 1822:  
 Inialisasi divais 1-wire → tulis 0xCC (lewati ROM) → tulis 0x48 (perintah untuk kopi scratchpad).

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	R1	R0	1	1	1	1	1

**Gambar 4**  
**Format Register Konfigurasi DS1822**

R1	R0	Resolution	Max Conversion Time
0	0	9-bit	93.75ms ( $t_{CONV}/8$ )
0	1	10-bit	187.5ms ( $t_{CONV}/4$ )
1	0	11-bit	375ms ( $t_{CONV}/2$ )
1	1	12-bit	750ms ( $t_{CONV}$ )

**Gambar 5**  
**Pilihan Resolusi DS1822**

6. Program lalu memicu DS1822 untuk memulai konversi suhu, dengan alur berikut:  
 Inialisasi divais 1-wire → tulis 0xCC (lewati ROM) → tulis 0x44 (perintah untuk convert T).
7. Kemudian program menunggu selama 800 ms sebagai toleransi untuk waktu konversi DS1822 (=750 ms @12 bit). Lalu program membaca data suhu dari *scratchpad* DS1822 dengan alur berikut:  
 Inialisasi divais 1-wire → tulis 0xCC (lewati ROM) → tulis 0xBE (perintah untuk baca *scratchpad*). → baca 9 byte data *scratchpad* DS1822 secara berurutan dari byte 0 sampai byte 8 dan masukkan ke buffer.

**SCRATCHPAD (Power-up State)**

byte 0	Temperature LSB (50h)	} (85°C)
byte 1	Temperature MSB (05h)	
byte 2	T <sub>H</sub> Register or User Byte 1*	
byte 3	T <sub>L</sub> Register or User Byte 2*	
byte 4	Configuration Register*	
byte 5	Reserved (FFh)	
byte 6	Reserved (0Ch)	
byte 7	Reserved (10h)	
byte 8	CRC*	

\*Power-up state depends on value(s) stored in EEPROM

**Gambar 6**  
**Urutan Scratchpad DS1822**

8. Setelah itu program mengambil data suhu dari buffer[0] (=byte 0) dan buffer[1] (=byte 1) kemudian menggabungkannya dan menempatkan datanya ke temp, dengan rumus berikut:  
 $Temp = (buffer[1] * 256) + buffer[0]$

9. Apabila data suhu negatif maka program mengubahnya menjadi positif dan variabel i diisi karakter '-' untuk kemudian ditampilkan ke LCD. Dan bila data suhu positif, program tidak mengubahnya dan variabel i diisi karakter '+'.
10. Setelah itu program menampilkan data suhu tersebut ke LCD dengan susunan berikut:  

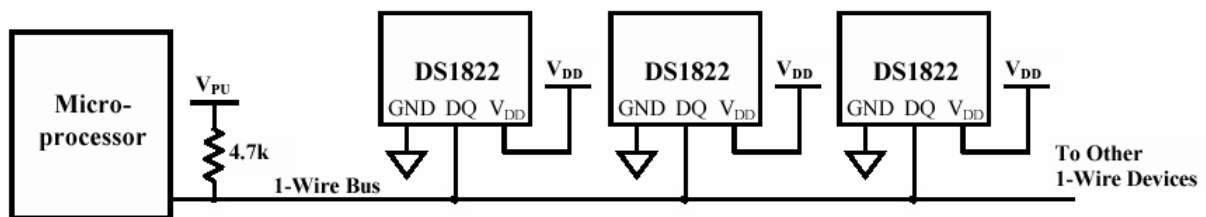
$$T = [ i ] [ temp/16 ] . [ temp \% 16 * 625 ] ^\circ C$$
 dimana i menunjukkan tanda + atau -, temp/16 merupakan angka desimal di depan koma sedangkan temp%16\*625 merupakan angka di belakang koma. Perhitungan ini menyesuaikan register temperatur DS1822.

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
LS Byte	$2^3$	$2^2$	$2^1$	$2^0$	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$
	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
MS Byte	S	S	S	S	S	$2^6$	$2^5$	$2^4$

**Gambar 7**  
**Format Register Temperatur DS1822**

11. Kemudian program kembali ke langkah 6, demikian seterusnya.

Aplikasi 1-Wire<sup>®</sup> menjadi sangat mudah dengan menggunakan bahasa pemrograman CodeVisionAVR. Dan bila dikehendaki aplikasi ini dapat diekspansi dengan menambah beberapa divais 1-Wire<sup>®</sup> lain dengan koneksi seperti pada Gambar 8, tentunya dengan sedikit mengubah program yaitu menambah sistem pengalamatan ROM 1-Wire<sup>®</sup>.



**Gambar 8**  
**Hubungan Beberapa Divais 1-Wire<sup>®</sup>**

Listing program terdapat pada **AN74.ZIP**.

Selamat berinovasi!

CodeVisionAVR is copyright by Pavel Haiduc, HP Info Tech.  
 1-Wire is registered trademark of Dallas Semiconductor.