

DT-SENSE

TEMPERATURE SENSOR

Quick Start

Trademarks & Copyright

AT, IBM, and PC are trademarks of International Business Machines Corp. Windows is a registered trademark of Microsoft Corporation. Pentium is a trademark of Intel Corporation. CodeVisionAVR is copyright by Pavel Haiduc, HP InfoTech s.r.l. BASCOM-51 and BASCOM-AVR are copyright by MCS Electronics. I²C is a registered trademark of Philips Semiconductors. DT-51 is a trademark of Innovative Electronics.

1. PENDAHULUAN

DT-SENSE TEMPERATURE SENSOR merupakan sebuah modul sensor cerdas yang mampu mengukur temperatur 0 – 100°C atau 32 – 212°F menggunakan sensor LM35DZ. Selain itu modul sensor cerdas ini dapat berfungsi sebagai kendali temperatur mandiri secara ON/OFF mengikuti *setpoint* temperatur yang kita tentukan. Modul sensor ini dilengkapi dengan antarmuka UART TTL dan I²C. Contoh aplikasi DT-SENSE TEMPERATURE SENSOR antara lain sebagai termometer udara *indoor/outdoor*, kendali temperatur, atau aplikasi-aplikasi lain yang memerlukan informasi temperatur.

2. SPESIFIKASI DT-SENSE TEMPERATURE SENSOR

Spesifikasi DT-SENSE TEMPERATURE SENSOR sebagai berikut:

- Mampu mengukur temperatur 0 – 100°C atau 32 – 212°F dengan ketelitian 1°.
- Pin Input/Output kompatibel dengan level tegangan TTL dan CMOS.
- Dilengkapi dengan antarmuka UART TTL dan I²C.
- Tersedia 1 jalur output kendali ON/OFF.
- Tersedia 1 jalur output fungsi umum.
- Tersedia 1 jalur input fungsi umum.
- Sumber catu daya menggunakan tegangan 4,8 – 5,4 VDC.

3. SISTEM YANG DIANJURKAN

Sistem yang dianjurkan untuk penggunaan DT-SENSE TEMPERATURE SENSOR adalah:

Perangkat keras:

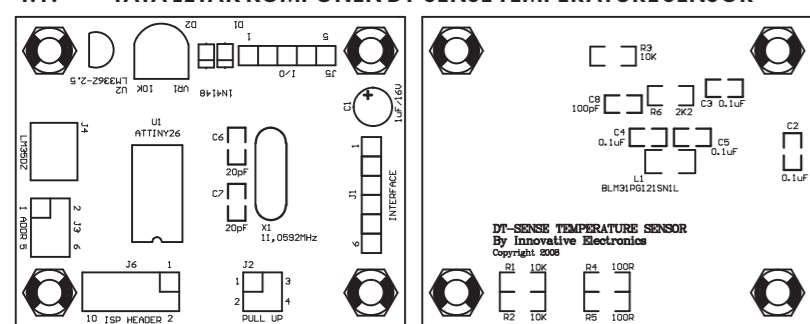
- PC AT Pentium® IBM Compatible dengan port Serial (COM1/COM2) dan Paralel (LPT).
- DT-51 Minimum System, DT-51 Low Cost Series, atau DT-AVR Low Cost Series.
- CD-ROM Drive dan Hard disk.

Perangkat lunak:

- Sistem operasi Windows® 98 SE.
- BASCOM-8051®, BASCOM-AVR®, atau CodeVisionAVR®.
- File yang ada pada CD program: CONTOH.PRJ, CONTOH.C, MANUAL DT-SENSE TEMPERATURE SENSOR, dan QUICK START DT-SENSE TEMPERATURE SENSOR.

4. PERANGKAT KERAS DT-SENSE TEMPERATURE SENSOR

4.1. TATA LETAK KOMPONEN DT-SENSE TEMPERATURE SENSOR



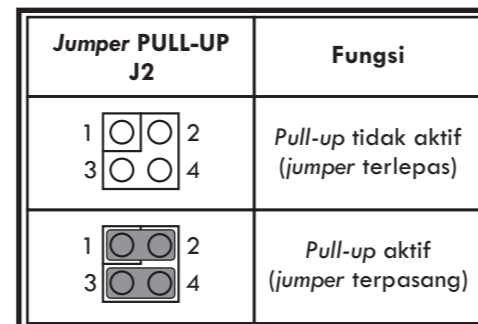
PCB-PCB awal menggunakan tulisan pada bagian atas dan bawah yang berbeda (lihat Manual di CD).

4.2. KONEKTOR DAN PENGATURAN JUMPER

Konektor INTERFACE (J1) berfungsi sebagai konektor untuk catu daya modul, antarmuka UART TTL, dan antarmuka I²C.

J1		
Pin	Nama	Fungsi
1	GND	Titik referensi untuk catu daya input
2	VCC	Terhubung ke catu daya (4,8 – 5,4 Volt)
3	RX TTL	Input serial level TTL ke modul
4	TX TTL	Output serial level TTL dari modul
5	SDA	I ² C-bus data input / output
6	SCL	I ² C-bus clock input

Jumper PULL-UP (J2) berfungsi untuk mengaktifkan resistor *pull-up* untuk pin SDA dan SCL pada antarmuka I²C.



Penting!

Apabila lebih dari satu modul dihubungkan pada I²C-bus maka jumper J2 (SCL/SDA) salah satu modul saja yang perlu dipasang.

Jumper ADDR (J3) berfungsi untuk mengatur alamat I²C dari modul DT-SENSE TEMPERATURE SENSOR.

J3			Alamat I ² C	
(A2)Pin 5-6	(A1)Pin 3-4	(A0)Pin 1-2	Alamat Tulis I ² C	Alamat Baca I ² C
■	■	■	E0H	E1H
■	■	■	E2H	E3H
■	■	■	E4H	E5H
■	■	■	E6H	E7H
■	■	■	E8H	E9H
■	■	■	EAH	EBH
■	■	■	ECH	EDH
■	■	■	EEH	EFH

Keterangan:

■ : jumper terpasang

Konektor LM35DZ (J4) sebagai konektor untuk sensor LM35DZ.

J4		
Pin	Nama	Fungsi
1	VCC	Output tegangan catu daya 5 Volt ke sensor
2	Vout	Tegangan output dari sensor
3	GND	Titik referensi catu daya ke sensor

Konektor I/O (J5) sebagai konektor untuk input atau output digital.

J5		
Pin	Nama	Fungsi
1	VCC	Output tegangan catu daya 5 Volt
2	OUT	Pin output kendali ON/OFF
3	GPO	Pin general purpose output
4	GPI	Pin general purpose input
5	GND	Titik referensi catu daya

5. KENDALI ON/OFF DENGAN DT-SENSE TEMPERATURE SENSOR

Modul DT-SENSE TEMPERATURE SENSOR memiliki 1 pin output (OUT) yang status logikanya akan berubah-ubah, sesuai dengan temperatur hasil pembacaan sensor LM35DZ dan batas atas serta batas bawah temperatur yang telah ditentukan. Pin output ini dapat dihubungkan dengan aktuator (*heater* atau *cooler*) sehingga modul DT-SENSE TEMPERATURE SENSOR ini dapat berfungsi sebagai pengendali temperatur secara mandiri.

Tiap satu periode *sampling*, modul DT-SENSE TEMPERATURE SENSOR akan membaca temperatur melalui sensor LM35DZ secara

otomatis dan kemudian mengubah status logika pin output kendali ON/OFF sesuai dengan aturan berikut:

- Jika temperatur lebih kecil dari pada batas bawah temperatur, maka pin output akan mengeluarkan logika *high* (Normal Logic) / *low* (Reverse Logic).
- Jika temperatur lebih besar dari pada batas atas temperatur, maka pin output akan mengeluarkan logika *low* (Normal Logic) / *high* (Reverse Logic).
- Jika temperatur sama dengan atau berada di antara batas atas dan batas bawah temperatur, maka logika pin output tidak berubah (jika sebelumnya *high*, maka akan tetap *high* atau jika sebelumnya *low* akan tetap *low*).

Lama periode *sampling*, nilai batas atas, nilai batas bawah, dan mode logika (Normal/Reverse), dapat diatur melalui antarmuka UART TTL atau I²C dengan menggunakan perintah-perintah yang akan dibahas pada **bagian 6.3**.

Ilustrasi cara kerja kendali ON/OFF menggunakan modul DT-SENSE TEMPERATURE SENSOR dapat dilihat pada Manual di CD.

6. PERANGKAT LUNAK DT-SENSE TEMPERATURE SENSOR

DT-SENSE TEMPERATURE SENSOR memiliki antarmuka UART TTL dan I²C yang dapat digunakan untuk menerima perintah atau mengirim data.

6.1. ANTARMUKA UART TTL

Parameter komunikasi UART TTL adalah sebagai berikut:

- 38400 bps
- 8 data bit
- 1 stop bit
- tanpa *parity* bit
- tanpa *flow control*

Semua perintah yang dikirim melalui antarmuka UART TTL dimulai dengan mengirim 1 byte data yang berisi **<nomor perintah>** dan (jika diperlukan) 1 byte data parameter perintah.

Jika perintah yang telah dikirimkan merupakan perintah yang meminta data dari modul DT-SENSE TEMPERATURE SENSOR, maka DT-SENSE TEMPERATURE SENSOR akan mengirimkan data melalui jalur TX TTL.

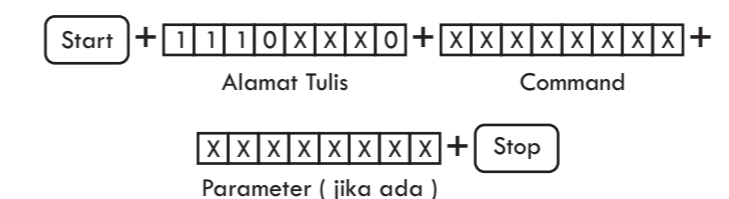
Perintah dan parameter yang bisa digunakan dapat dilihat pada **bagian 6.3**.

6.2. ANTARMUKA I²C

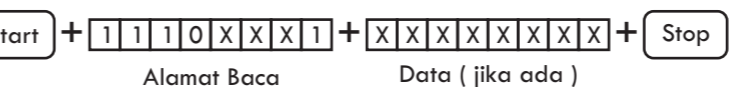
Modul DT-SENSE TEMPERATURE SENSOR memiliki antarmuka I²C. Pada antarmuka I²C ini, modul DT-SENSE TEMPERATURE SENSOR bertindak sebagai *slave* dengan alamat sesuai dengan telah ditentukan sebelumnya melalui pengaturan *jumper* (lihat **bagian 4.2**). Antarmuka I²C pada modul DT-SENSE TEMPERATURE SENSOR mendukung *bit rate* sampai dengan maksimum 50 kHz.

Semua perintah yang dikirim melalui antarmuka I²C diawali dengan *start condition* dan kemudian diikuti dengan pengiriman 1 byte alamat modul DT-SENSE TEMPERATURE SENSOR. Setelah pengiriman alamat, selanjutnya *master* harus mengirim 1 byte data yang berisi **<nomor perintah>** dan (jika diperlukan) 1 byte data parameter perintah. Selanjutnya, setelah seluruh parameter perintah telah dikirim, urutan perintah diakhiri dengan *stop condition*.

Berikut urutan yang harus dilakukan untuk mengirimkan perintah melalui antarmuka I²C.



Jika perintah yang telah dikirimkan merupakan perintah yang meminta data dari modul DT-SENSE TEMPERATURE SENSOR, maka data-data tersebut dapat dibaca dengan menggunakan urutan perintah baca. Berikut urutan yang harus dilakukan untuk membaca data dari DT-SENSE TEMPERATURE SENSOR.



Perintah dan parameternya yang bisa digunakan dapat dilihat pada **bagian 6.3**.

6.3. COMMAND SET

Berikut ini daftar lengkap perintah-perintah dalam antarmuka UART dan I²C. Beberapa parameter juga akan disimpan dalam EEPROM

modul DT-SENSE TEMPERATURE SENSOR. Parameter tersebut akan dibaca saat DT-SENSE TEMPERATURE SENSOR baru dinyalakan.

6.3.1. READ TEMPERATURE C

Fungsi	Untuk membaca temperature dalam satuan °C
Command	00H
Parameter	—
Respon	<temperaturC> 0 – 100 → data temperatur dalam °C
Delay antara Command dan Respon	15 ms
Keterangan	—

Contoh dengan antarmuka UART:

User : 00H
DT-SENSE : <temperaturC>

Contoh *pseudo code* C dengan antarmuka I²C (misalkan alamat = E0H):

```
i2c_start(); // Start Condition
i2c_write(0xE0); // Tulis ke modul
// TEMPERATURE SENSOR
i2c_write(0x00); // Perintah "Read Temperature C"
i2c_stop(); // Stop Condition

delay_ms(15); // delay 15 ms

i2c_start(); // Start Condition
i2c_write(0xE1); // Baca ke modul
// TEMPERATURE SENSOR
temp = i2c_read(0); // Data temperatur
i2c_stop(); // Stop Condition
```

6.3.2. READ TEMPERATURE F

Fungsi	Untuk membaca temperature dalam satuan °F
Command	01H
Parameter	—
Respon	<temperaturF> 32 – 212 → data temperatur dalam °F
Delay antara Command dan Respon	15 ms
Keterangan	—

Contoh dengan antarmuka UART:

User : 01H
DT-SENSE : <temperaturF>

Contoh *pseudo code* C dengan antarmuka I²C (misalkan alamat = E0H):

```
i2c_start(); // Start Condition
i2c_write(0xE0); // Tulis ke modul
// TEMPERATURE SENSOR
i2c_write(0x01); // Perintah "Read Temperature F"
i2c_stop(); // Stop Condition

delay_ms(15); // delay 15 ms

i2c_start(); // Start Condition
i2c_write(0xE1); // Baca ke modul
// TEMPERATURE SENSOR
temp = i2c_read(0); // Data temperatur
i2c_stop(); // Stop Condition
```

6.3.3. READ TOP SETPOINT

Fungsi	Untuk membaca batas atas <i>setpoint</i> temperatur (°C) yang tersimpan dalam EEPROM
Command	02H
Parameter	—
Respon	<batasAtas> 0 – 255 → data batas atas <i>setpoint</i> temperatur (°C)
Delay antara Command dan Respon	15 ms
Keterangan	—

Contoh dengan antarmuka UART:

User : 02H
DT-SENSE : <batasAtas>

Contoh pseudo code C dengan antarmuka I²C (misalkan alamat = E0H):

```
i2c_start(); // Start Condition
i2c_write(0xE0); // Tulis ke modul
// TEMPERATURE SENSOR
i2c_write(0x02); // Perintah "Read Top Setpoint"
i2c_stop(); // Stop Condition

delay_ms(15); // delay 15 ms

i2c_start(); // Start Condition
i2c_write(0xE1); // Baca ke modul
// TEMPERATURE SENSOR
batasA = i2c_read(0); // Data batas atas
i2c_stop(); // Stop Condition
```

6.3.4. READ BOTTOM SETPOINT

Fungsi	Untuk membaca batas bawah <i>setpoint</i> temperatur (°C) yang tersimpan dalam EEPROM
Command	03H
Parameter	—
Respon	<batasBawah> 0–255 → data batas bawah <i>setpoint</i> temperatur (°C)
Delay antara Command dan Respon	15 ms
Keterangan	—

Contoh dengan antarmuka UART:

User : 03H
DT-SENSE : <batasBawah>

Contoh pseudo code C dengan antarmuka I²C (misalkan alamat = E0H):

```
i2c_start(); // Start Condition
i2c_write(0xE0); // Tulis ke modul
// TEMPERATURE SENSOR
i2c_write(0x03); // Perintah "Read Bottom Setpoint"
i2c_stop(); // Stop Condition

delay_ms(15); // delay 15 ms

i2c_start(); // Start Condition
i2c_write(0xE1); // Baca ke modul
// TEMPERATURE SENSOR
batasB = i2c_read(0); // Data batas bawah
i2c_stop(); // Stop Condition
```

6.3.5. SET TOP SETPOINT

Fungsi	Untuk mengatur batas atas <i>setpoint</i> temperatur (°C)
Command	04H
Parameter	<batasAtas> 0–255 → data batas atas <i>setpoint</i> temperatur (°C)
Respon	—
Delay antara Command dan Respon	—
Keterangan	~ Agar nilai batas atas dianggap <i>valid</i> dan tersimpan ke EEPROM, maka nilai batas atas harus lebih besar dari nilai batas bawah.

Contoh dengan antarmuka UART untuk mengatur agar batas atas *setpoint* temperatur menjadi 50°C (50 = 32H):

User : 04H 32H

Contoh pseudo code C dengan antarmuka I²C (misalkan alamat = E0H):

```
i2c_start(); // Start Condition
i2c_write(0xE0); // Tulis ke modul
// TEMPERATURE SENSOR
i2c_write(0x04); // Perintah "Set Top Setpoint"
i2c_write(0x32); // Data batas atas
i2c_stop(); // Stop Condition
```

6.3.6. SET BOTTOM SETPOINT

Fungsi	Untuk mengatur batas bawah <i>setpoint</i> temperatur (°C)
Command	05H
Parameter	<batasBawah> 0–255 → data batas bawah <i>setpoint</i> temperatur (°C)
Respon	—
Delay antara Command dan Respon	—

Keterangan	~ Agar nilai batas bawah dianggap <i>valid</i> dan tersimpan ke EEPROM, maka nilai batas bawah harus lebih kecil dari nilai batas atas.
------------	---

Contoh dengan antarmuka UART untuk mengatur agar batas bawah *setpoint* temperatur menjadi 45°C (45 = 2DH):

User : 05H 2DH

Contoh pseudo code C dengan antarmuka I²C (misalkan alamat = E0H):

```
i2c_start(); // Start Condition
i2c_write(0xE0); // Tulis ke modul
// TEMPERATURE SENSOR
i2c_write(0x05); // Perintah "Set Bottom Setpoint"
i2c_write(0x2D); // Data batas bawah
i2c_stop(); // Stop Condition
```

6.3.7. SET SAMPLING

Fungsi	Untuk mengatur periode <i>sampling</i> kendali ON/OFF
Command	06H
Parameter	<periodeSampling> 1–250 → periode <i>sampling</i> kendali ON/OFF (dalam satuan 100ms). Range yang diperbolehkan 100 ms – 25 s.
Respon	—
Delay antara Command dan Respon	—
Keterangan	~ Jika perintah ini diterima, maka parameter akan disimpan pada EEPROM.

Contoh dengan antarmuka UART untuk mengatur agar lama periode *sampling* kendali ON/OFF menjadi 1 detik (10 x 100 ms) sehingga nilai variabel <periodeSampling> adalah 10 (10 = 0AH):

User : 06H 0AH

Contoh pseudo code C dengan antarmuka I²C (misalkan alamat = E0H):

```
i2c_start(); // Start Condition
i2c_write(0xE0); // Tulis ke modul TEMPERATURE SENSOR
i2c_write(0x06); // Perintah "Set Sampling"
i2c_write(0x0A); // Data periode (10 x 100 ms)
i2c_stop(); // Stop Condition
```

6.3.8. SET MODE

Fungsi	Untuk mengatur mode logika pin output kendali ON/OFF (Out)
Command	07H
Parameter	<mode> 0 → mode <i>Normal Logic</i> 1 → mode <i>Reverse Logic</i>
Respon	—
Delay antara Command dan Respon	—
Keterangan	~ Jika perintah ini diterima, maka parameter akan disimpan pada EEPROM.

Contoh dengan antarmuka UART untuk mengatur agar pin output kendali ON/OFF bekerja pada mode **Normal Logic**:

User : 07H 00H

Contoh pseudo code C dengan antarmuka I²C (misalkan alamat = E0H):

```
i2c_start(); // Start Condition
i2c_write(0xE0); // Tulis ke modul TEMPERATURE SENSOR
i2c_write(0x07); // Perintah "Set Mode"
i2c_write(0x00); // Data pilihan mode "Normal logic"
i2c_stop(); // Stop Condition
```

6.3.9. CLEAR GPO

Fungsi	Untuk mengatur agar pin GPO memiliki logika low
Command	08H
Parameter	—
Respon	—
Delay antara Command dan Respon	—

Keterangan	~ Hasil perintah ini tidak disimpan di EEPROM. ~ Saat modul DT-SENSE TEMPERATURE SENSOR baru dinyalakan, pin general purpose output akan memiliki logika <i>low</i> .
------------	--

Contoh dengan antarmuka UART:

User : 08H

Contoh pseudo code C dengan antarmuka I²C (misalkan alamat = E0H):

```
i2c_start(); // Start Condition
i2c_write(0xE0); // Tulis ke modul
// TEMPERATURE SENSOR
i2c_write(0x08); // Perintah "Clear GPO"
i2c_stop(); // Stop Condition
```

6.3.10. SET GPO

Fungsi	Untuk mengatur agar pin GPO memiliki logika <i>high</i>
Command	09H
Parameter	—
Respon	—
Delay antara Command dan Respon	—
Keterangan	~ Hasil perintah ini tidak disimpan di EEPROM. ~ Saat modul DT-SENSE TEMPERATURE SENSOR baru dinyalakan, pin general purpose output akan memiliki logika <i>low</i> .

Contoh dengan antarmuka UART:

User : 09H

Contoh pseudo code C dengan antarmuka I²C (misalkan alamat = E0H):

```
i2c_start(); // Start Condition
i2c_write(0xE0); // Tulis ke modul
// TEMPERATURE SENSOR
i2c_write(0x09); // Perintah "Set GPO"
i2c_stop(); // Stop Condition
```

6.3.11. READ GPI

Fungsi	Untuk membaca logika pin GPI
Command	0AH
Respon	<logikalInput> 0 → GPI berlogika <i>low</i> 1 → GPI berlogika <i>high</i>
Delay antara Command dan Respon	15 ms
Keterangan	~ Jika pin GPI ini tidak dihubungkan ke manapun, maka pin ini akan memiliki logika <i>high</i> karena pin ini memiliki resistor <i>internal pull-up</i> .

Contoh dengan antarmuka UART:

User : 0AH
DT-SENSE : <logikalInput>

Contoh pseudo code C dengan antarmuka I²C (misal alamat = E0H):

```
i2c_start(); // Start Condition
i2c_write(0xE0); // Tulis ke modul
// TEMPERATURE SENSOR
i2c_write(0x0A); // Perintah "Read GPI"
i2c_stop(); // Stop Condition

delay_ms(15); // delay 15 ms

i2c_start(); // Start Condition
i2c_write(0xE1); // Baca ke modul
// TEMPERATURE SENSOR
logika = i2c_read(0); // Data logika pin GPI
i2c_stop(); // Stop Condition
```

7. PROSEDUR PENGUJIAN

1. Hubungkan sensor LM35DZ ke konektor J4 modul DT-SENSE TEMPERATURE SENSOR.
2. Hubungkan sumber catu daya 5 Volt ke modul DT-SENSE TEMPERATURE SENSOR.
3. Jika modul DT-SENSE TEMPERATURE SENSOR berhasil melakukan power-up reset dengan baik, maka pin GPO (J5 pin 3) akan memiliki nilai sekitar 0 Volt dan pin GPI (J5 pin 4) akan memiliki nilai sekitar 5 Volt.

4. Kirimkan perintah READ TEMPERATURE C melalui antarmuka UART TTL.
5. Modul DT-SENSE TEMPERATURE SENSOR akan mengambil data temperatur melalui sensor LM35DZ dan mengirimkan hasilnya melalui antarmuka UART TTL. Untuk suhu ruangan biasa, nilai temperatur berkisar antara 20 hingga 30 °C.

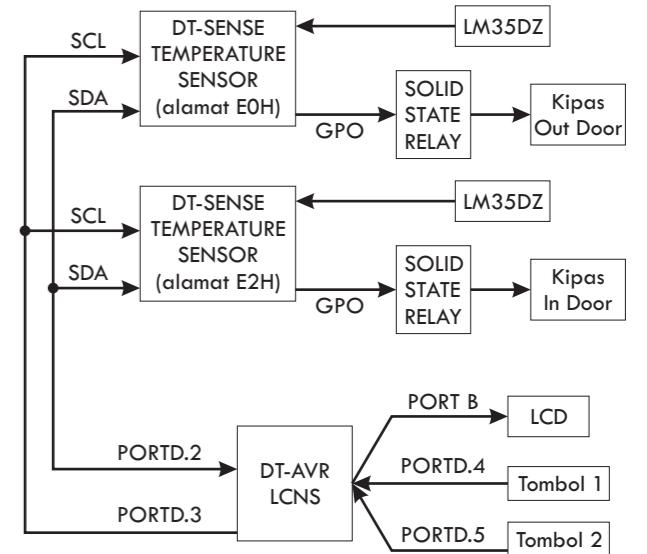
8. CONTOH APLIKASI DAN PROGRAM

Sebagai contoh aplikasi, dimisalkan 2 buah modul DT-SENSE TEMPERATURE SENSOR digunakan untuk memantau temperatur dalam ruangan (*indoor*) dan temperatur luar ruangan (*outdoor*) dengan antarmuka I²C. Modul DT-AVR Low Cost Nano System (LCNS) digunakan sebagai *master*. DT-AVR LCNS bertugas untuk mengirimkan perintah pembacaan dan menampilkan ke LCD. Selain memantau temperatur, modul DT-AVR LCNS digunakan untuk mengendalikan kipas *outdoor* dan kipas *indoor* secara terpusat menggunakan 2 buah tombol.

Berikut koneksi antara modul-modul yang digunakan:

DT-AVR LCNS	Terhubung ke
Port D.2	SDA DT-SENSE TEMPERATURE SENSOR
Port D.3	SCL DT-SENSE TEMPERATURE SENSOR
Port D.4	Tombol <i>active low</i> (salah satu kakinya dihubungkan ke ground)
Port D.5	Tombol <i>active low</i> (salah satu kakinya dihubungkan ke ground)
Port B.0	RS (LCD Pin 4)
Port B.1	R/W (LCD pin 5)
Port B.2	E (LCD pin 6)
Port B.4	DB4 (LCD pin 11)
Port B.5	DB5 (LCD pin 12)
Port B.6	DB6 (LCD pin 13)
Port B.7	DB7 (LCD pin 14)

DT-SENSE TEMPERATURE SENSOR	Terhubung ke
J4	SENSOR
GPO (J5 pin 3)	Solid State Relay



Sebagai contoh program untuk aplikasi di atas, pada CD yang disertakan pada saat pembelian modul DT-SENSE TEMPERATURE SENSOR disertakan program contoh.c yang ditulis dengan menggunakan CodeVisionAVR 1.25.2 versi evaluasi. Pada program tersebut, DT-AVR LCNS akan mengirimkan perintah *Read Temperature C* ke DT-SENSE TEMPERATURE SENSOR *outdoor* (alamat E0H) dan DT-SENSE TEMPERATURE SENSOR *indoor* (alamat E2H). Setelah data temperatur *outdoor* dan *indoor* diperoleh, maka DT-AVR LCNS akan menampilkannya data tersebut di LCD. Setelah menampilkan data ke LCD, DT-AVR LCNS akan memeriksa apakah ada penekanan Tombol 1 atau Tombol 2. Jika Tombol 1 ditekan, maka DT-AVR LCNS akan mengirimkan perintah untuk menyalakan atau mematikan kipas *outdoor*. Demikian juga jika Tombol 2 ditekan, maka DT-AVR LCNS akan mengirimkan perintah untuk menyalakan atau mematikan kipas *indoor*.

Terima Kasih atas kepercayaan Anda menggunakan produk kami, bila ada kesulitan, pertanyaan atau saran mengenai produk ini silakan menghubungi technical support kami :

support@innovativeelectronics.com