

# DT-SENSE

---

## FLAME DETECTOR

### **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.

I<sup>2</sup>C is a registered trademark of Philips Semiconductors.

DT-51 is a trademark of Innovative Electronics.

# Daftar Isi

---

<b>1</b>	<b>Pendahuluan.....</b>	<b>3</b>
1.1	Spesifikasi DT-SENSE FLAME DETECTOR.....	3
1.2	Sistem yang Dianjurkan.....	3
<b>2</b>	<b>Perangkat Keras DT-SENSE FLAME DETECTOR.....</b>	<b>4</b>
2.1	Tata Letak Komponen DT-SENSE FLAME DETECTOR.....	4
2.2	Konektor dan Pengaturan Jumper.....	5
<b>3</b>	<b>Perangkat Lunak DT-SENSE FLAME DETECTOR.....</b>	<b>6</b>
3.1	Antarmuka UART TTL.....	6
3.2	Antarmuka I <sup>2</sup> C.....	6
3.3	Command Set.....	7
3.3.1	Check Status.....	7
3.3.2	Goto Position.....	8
3.3.3	Read Raw.....	9
3.3.4	Autoscan.....	10
3.3.5	Read Autoscan.....	10
3.3.6	Get Position.....	11
3.3.7	Enable Servo.....	12
3.3.8	Disable Servo.....	12
3.3.9	Set Default.....	13
<b>4</b>	<b>Prosedur Pengujian.....</b>	<b>13</b>
<b>5</b>	<b>Contoh Aplikasi dan Program.....</b>	<b>14</b>
<b>Lampiran</b>		
A.	Skematik DT-SENSE FLAME DETECTOR.....	16

## 1. PENDAHULUAN

DT-SENSE FLAME DETECTOR merupakan sebuah sistem sensor cerdas yang mampu mendeteksi posisi nyala api dengan ketelitian tinggi (hingga nyala api sekecil cahaya lilin) menggunakan gabungan sensor mata api dan motor servo. Sistem ini terdiri dari sebuah sensor photodiode yang didesain untuk mendeteksi mata api dan sebuah modul berbasis mikrokontroler yang digunakan untuk mengatur kerja motor servo, mengambil *sampling* data sensor, dan mengatur antarmuka dengan sistem lain. Salah satu aplikasi DT-SENSE FLAME DETECTOR adalah robot pencari api (Kontes Robot Cerdas Indonesia).

### 1.1. SPESIFIKASI DT-SENSE FLAME DETECTOR

Spesifikasi DT-SENSE FLAME DETECTOR sebagai berikut:

- Mampu mendeteksi api sebuah lilin pada jarak maksimum 40 cm dari mata sensor.
- Pin Input/Output kompatibel dengan level tegangan TTL dan CMOS.
- Dilengkapi dengan antarmuka UART TTL dan I<sup>2</sup>C.
- Tersedia konektor untuk motor servo standar.
- Dapat mengatur kerja motor servo dengan ketelitian 1°.
- Sumber catu daya menggunakan tegangan 4,8 – 5,4 VDC.

### 1.2. SISTEM YANG DIANJURKAN

Sistem yang dianjurkan untuk penggunaan DT-SENSE FLAME DETECTOR 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.

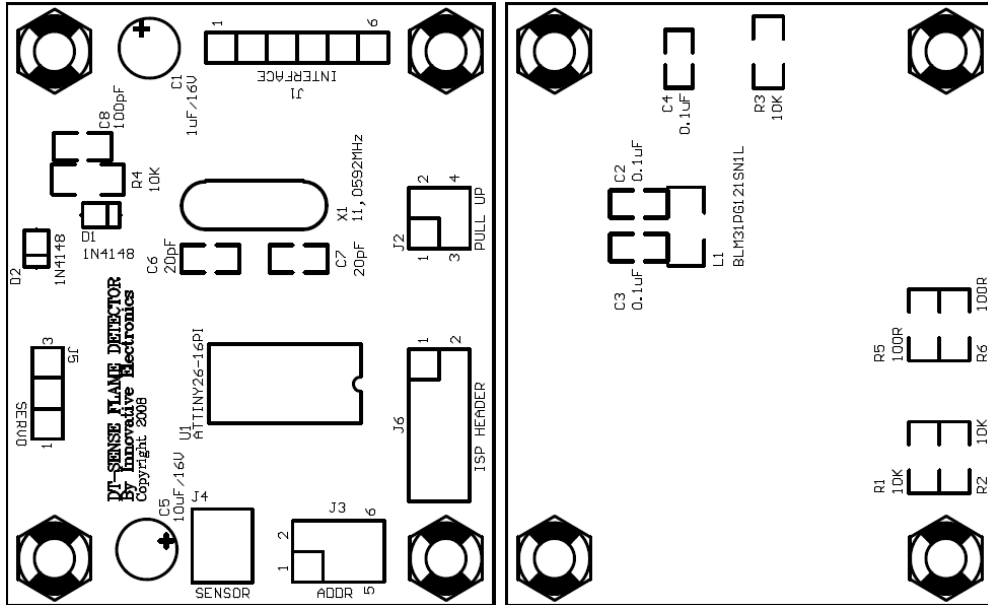
Perangkat lunak:

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

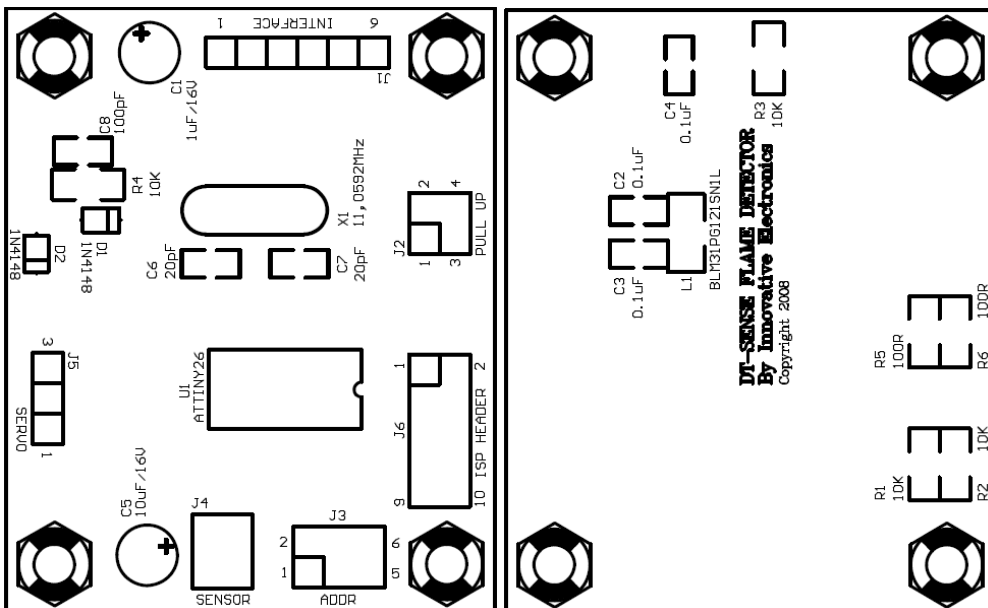
## 2. PERANGKAT KERAS DT-SENSE FLAME DETECTOR

### 2.1. TATA LETAK KOMPONEN DT-SENSE FLAME DETECTOR

PCB-PCB awal menggunakan tulisan pada bagian atas dan bawah sebagai berikut:



PCB-PCB berikutnya akan menggunakan tulisan sebagai berikut:



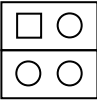

Rangkaian kedua jenis PCB adalah SAMA.

## 2.2. KONEKTOR DAN PENGATURAN JUMPER

Konektor INTERFACE (J1) berfungsi sebagai konektor untuk catu daya modul, antarmuka UART TTL, dan antarmuka I<sup>2</sup>C.

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 <sup>2</sup> C-bus data input / output
6	SCL	I <sup>2</sup> C-bus clock input

Jumper PULL-UP (J2) berfungsi untuk mengaktifkan resistor *pull-up* untuk pin SDA dan SCL pada antarmuka I<sup>2</sup>C.

Jumper PULL-UP J2	Fungsi
	<i>Pull-up</i> tidak aktif ( <i>jumper</i> terlepas)
	<i>Pull-up</i> aktif ( <i>jumper</i> terpasang)

### Penting !

Apabila lebih dari satu modul dihubungkan pada I<sup>2</sup>C-bus maka *jumper* J2 (SCL/SDA) salah satu modul saja yang perlu dipasang.

Jumper ADDR (J3) berfungsi untuk mengatur alamat I<sup>2</sup>C dari modul DT-SENSE FLAME DETECTOR.

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

Keterangan:

■ : *jumper* terpasang

Konektor SENSOR (J4) sebagai konektor untuk sensor mata api *photo IC diode* S10108.

Pin	Nama	Fungsi
1	Vout	Tegangan output dari sensor
2	VCC	Tegangan catu daya ke sensor

Konektor SERVO (J5) sebagai konektor untuk motor servo standar. Motor servo digunakan sebagaiudukan sensor sehingga mata sensor dapat menyapu sudut 180°.

Pin	Nama	Fungsi
1	Servo	Output pulsa (1 – 2 ms) ke motor servo
2	VCC	Tegangan catu daya ke motor servo
3	GND	Titik referensi catu daya ke motor servo

### 3. PERANGKAT LUNAK DT-SENSE FLAME DETECTOR

DT-SENSE FLAME DETECTOR memiliki antarmuka UART TTL dan I<sup>2</sup>C yang dapat digunakan untuk menerima perintah atau mengirim data.

#### 3.1. ANTARLUKA 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 FLAME DETECTOR, maka DT-SENSE FLAME DETECTOR akan mengirimkan data melalui jalur TX TTL.

Perintah dan parameter yang bisa digunakan dapat dilihat pada **bagian 3.3**.

#### 3.2. ANTARLUKA I<sup>2</sup>C

Modul DT-SENSE FLAME DETECTOR memiliki antarmuka I<sup>2</sup>C. Pada antarmuka I<sup>2</sup>C ini, modul DT-SENSE FLAME DETECTOR bertindak sebagai *slave* dengan alamat sesuai dengan telah ditentukan sebelumnya melalui pengaturan *jumper* (lihat **bagian 2.2**). Antarmuka I<sup>2</sup>C pada modul DT-SENSE FLAME DETECTOR mendukung *bit rate* sampai dengan maksimum 50 kHz.

Semua perintah yang dikirim melalui antarmuka I<sup>2</sup>C diawali dengan **start condition** dan kemudian diikuti dengan pengiriman 1 byte alamat modul DT-SENSE FLAME DETECTOR. Setelah pengiriman alamat, selanjutnya *master* harus mengirim 1 byte data yang berisi **<nomor perintah>** dan (jika diperlukan) 1



Contoh dengan antarmuka UART:

User : 00H  
DT-SENSE : <status>

Berikut ini contoh *pseudo code* C untuk menggunakan perintah ini dengan antarmuka I<sup>2</sup>C (misalkan alamat I<sup>2</sup>C = E0H):

```
i2c_start(); // Start Condition
i2c_write(0xE0); // Tulis ke modul FLAME DETECTOR
i2c_write(0x00); // Perintah "Check Status"
i2c_stop(); // Stop Condition

delay_ms(15); // delay 15 ms

i2c_start(); // Start Condition
i2c_write(0xE1); // Baca ke modul FLAME DETECTOR
status = i2c_read(0); // Data Status
i2c_stop(); // Stop Condition
```

### 3.3.2. GOTO POSITION

<b>Fungsi</b>	Untuk memutar motor servo ke sudut tertentu
<b>Command</b>	<b>01H</b>
<b>Parameter</b>	<Sudut> 0 – 180 → Posisi sudut motor servo yang diinginkan. <i>Range yang diperbolehkan 0° – 180°.</i>
<b>Respon</b>	-
<b>Delay antara Command dan Respon</b>	-
<b>Keterangan</b>	-

Contoh dengan antarmuka UART untuk memutar motor servo ke sudut 90° (90 = 5AH):

User : 01H 5AH

Berikut ini contoh *pseudo code* C untuk menggunakan perintah ini dengan antarmuka I<sup>2</sup>C (misalkan alamat I<sup>2</sup>C = E0H):

```
i2c_start(); // Start Condition
i2c_write(0xE0); // Tulis ke modul FLAME DETECTOR
i2c_write(0x01); // Perintah "Goto Position"
i2c_write(90); // Sudut = 90 derajat
i2c_stop(); // Stop Condition
```

### 3.3.3. READ RAW

<b>Fungsi</b>	Untuk membaca data mentah dari sensor di posisi sekarang
<b>Command</b>	<b>02H</b>
<b>Parameter</b>	-
<b>Respon</b>	<dataRaw> 0 – 255 → data mentah sensor hasil ADC 8 bit
<b>Delay antara Command dan Respon</b>	15 ms
<b>Keterangan</b>	-

Contoh dengan antarmuka UART:

User : 02H  
DT-SENSE : <dataRaw>

Berikut ini contoh *pseudo code* C untuk menggunakan perintah ini dengan antarmuka I<sup>2</sup>C (misalkan alamat I<sup>2</sup>C = E0H):

```
i2c_start();           // Start Condition
i2c_write(0xE0);      // Tulis ke modul FLAME DETECTOR
i2c_write(0x02);      // Perintah "Read Raw"
i2c_stop();           // Stop Condition

delay_ms(15);         // delay 15 ms

i2c_start();           // Start Condition
i2c_write(0xE1);      // Baca ke modul FLAME DETECTOR
rawData = i2c_read(0); // Data mentah sensor
i2c_stop();           // Stop Condition
```

### 3.3.4. AUTOSCAN

<b>Fungsi</b>	Untuk mulai proses <i>autoscan</i>
<b>Command</b>	<b>03H</b>
<b>Parameter</b>	-
<b>Respon</b>	-
<b>Delay antara Command dan Respon</b>	-
<b>Keterangan</b>	<ul style="list-style-type: none"> <li>● Pada proses <i>autoscan</i> sebaiknya sensor diletakkan di atas motor servo sehingga mata sensor dapat menyapu sudut 180°.</li> <li>● Selama modul DT-SENSE FLAME DETECTOR sibuk melakukan <i>autoscan</i>, maka hasil perintah Check Status akan menghasilkan data 00H.</li> <li>● Jika modul DT-SENSE FLAME DETECTOR telah selesai melakukan <i>autoscan</i>, maka hasil perintah Check Status akan menghasilkan data 01H jika api tidak ditemukan atau 02H jika api ditemukan.</li> <li>● Jika api <b>tidak</b> ditemukan, maka motor servo akan kembali ke posisi <i>default</i>.</li> <li>● Jika api ditemukan, maka motor servo akan menuju ke posisi api.</li> </ul>

Contoh dengan antarmuka UART:

User : 03H

Berikut ini contoh *pseudo code* C untuk menggunakan perintah ini dengan antarmuka I<sup>2</sup>C (misalkan alamat I<sup>2</sup>C = E0H):

```

i2c_start();           // Start Condition
i2c_write(0xE0);      // Tulis ke modul FLAME DETECTOR
i2c_write(0x03);      // Perintah "Autoscan"
i2c_stop();           // Stop Condition

```

### 3.3.5. READ AUTOSCAN

<b>Fungsi</b>	Untuk membaca hasil proses <i>autoscan</i>
<b>Command</b>	<b>04H</b>
<b>Parameter</b>	-
<b>Respon</b>	<p>&lt;posisi api&gt;  0 – 180 → Posisi api hasil <i>autoscan</i>.  Range yang diperbolehkan 0° – 180°.  255 → Jika api tidak ditemukan atau proses <i>autoscan</i> belum dijalankan.</p>
<b>Delay antara Command dan Respon</b>	15 ms
<b>Keterangan</b>	<ul style="list-style-type: none"> <li>● Jika perintah ini dikirimkan, maka modul DT-SENSE FLAME DETECTOR akan selalu mengirimkan hasil proses <i>autoscan</i> terakhir. Nilai ini tidak terpengaruh oleh perubahan posisi motor servo.</li> </ul>

Contoh dengan antarmuka UART:

User : 04H  
DT-SENSE : <posisi api>

Berikut ini contoh pseudo code C untuk menggunakan perintah ini dengan antarmuka I<sup>2</sup>C (misalkan alamat I<sup>2</sup>C = E0H):

```
i2c_start(); // Start Condition
i2c_write(0xE0); // Tulis ke modul FLAME DETECTOR
i2c_write(0x04); // Perintah "Read Autoscan"
i2c_stop(); // Stop Condition

delay_ms(15); // delay15 ms

i2c_start(); // Start Condition
i2c_write(0xE1); // Baca ke modul FLAME DETECTOR
posisi = i2c_read(0); // posisi api
i2c_stop(); // Stop Condition
```

### 3.3.6. GET POSITION

<b>Fungsi</b>	Untuk membaca posisi sudut motor servo sekarang
<b>Command</b>	<b>05H</b>
<b>Parameter</b>	-
<b>Respon</b>	<posisi servo> 0 – 180 → Posisi sudut motor servo sekarang. Range yang diperbolehkan 0° – 180°.
<b>Delay antara Command dan Respon</b>	-
<b>Keterangan</b>	-

Contoh dengan antarmuka UART:

User : 05H  
DT-SENSE : <posisi servo>

Berikut ini contoh pseudo code C untuk menggunakan perintah ini dengan antarmuka I<sup>2</sup>C (misalkan alamat I<sup>2</sup>C = E0H):

```
i2c_start(); // Start Condition
i2c_write(0xE0); // Tulis ke modul FLAME DETECTOR
i2c_write(0x05); // Perintah "Get Position"
i2c_stop(); // Stop Condition

delay_ms(15); // delay 15 ms

i2c_start(); // Start Condition
i2c_write(0xE1); // Baca ke modul FLAME DETECTOR
sudut = i2c_read(0); // Data posisi servo
i2c_stop(); // Stop Condition
```

### 3.3.7. ENABLE SERVO

<b>Fungsi</b>	Untuk mengaktifkan motor servo
<b>Command</b>	<b>06H</b>
<b>Parameter</b>	-
<b>Respon</b>	-
<b>Delay antara Command dan Respon</b>	-
<b>Keterangan</b>	<ul style="list-style-type: none"> <li>● Pada saat awal dinyalakan, motor servo selalu dalam kondisi aktif.</li> <li>● Jika motor servo diaktifkan, maka motor servo akan selalu berusaha mempertahankan posisi sudutnya sesuai dengan yang diperintahkan.</li> <li>● Sebelum menggunakan perintah-perintah yang menggunakan motor servo (Goto Position atau Autoscan), motor servo harus berada dalam kondisi aktif.</li> </ul>

Contoh dengan antarmuka UART:

User : 06H

Berikut ini contoh *pseudo code* C untuk menggunakan perintah ini dengan antarmuka I<sup>2</sup>C (misalkan alamat I<sup>2</sup>C = E0H):

```
i2c_start(); // Start Condition
i2c_write(0xE0); // Tulis ke modul FLAME DETECTOR
i2c_write(0x06); // Perintah "Enable Servo"
i2c_stop(); // Stop Condition
```

### 3.3.8. DISABLE SERVO

<b>Fungsi</b>	Untuk menonaktifkan motor servo
<b>Command</b>	<b>07H</b>
<b>Parameter</b>	-
<b>Respon</b>	-
<b>Delay antara Command dan Respon</b>	-
<b>Keterangan</b>	<ul style="list-style-type: none"> <li>● Pada saat awal dinyalakan, motor servo selalu dalam kondisi aktif.</li> <li>● Jika motor servo dinonaktifkan, maka motor servo tidak berfungsi dan dapat digerakkan dengan tangan.</li> <li>● Perintah ini dapat digunakan untuk menghemat penggunaan catu daya.</li> </ul>

Contoh dengan antarmuka UART:

User : 07H

Berikut ini contoh *pseudo code* C untuk menggunakan perintah ini dengan antarmuka I<sup>2</sup>C (misalkan alamat I<sup>2</sup>C = E0H):

```

i2c_start();           // Start Condition
i2c_write(0xE0);      // Tulis ke modul FLAME DETECTOR
i2c_write(0x07);      // Perintah "Disable Servo"
i2c_stop();           // Stop Condition

```

### 3.3.9. SET DEFAULT

<b>Fungsi</b>	Untuk menentukan posisi <i>default</i> motor servo
<b>Command</b>	<b>08H</b>
<b>Parameter</b>	<Sudut> 0 – 180 → Posisi <i>default</i> motor servo yang diinginkan. Range yang diperbolehkan 0° – 180°.
<b>Respon</b>	-
<b>Delay antara Command dan Respon</b>	-
<b>Keterangan</b>	<ul style="list-style-type: none"> <li>● Saat selesai melakukan proses <i>autoscan</i> dan tidak ditemukan api, maka motor servo akan diatur agar berada pada posisi <i>default</i>.</li> <li>● Posisi <i>default</i> ini akan disimpan pada EEPROM modul DT-SENSE FLAME DETECTOR.</li> <li>● Saat modul DT-SENSE FLAME DETECTOR baru dinyalakan, maka motor servo akan diatur agar berada pada posisi <i>default</i>.</li> </ul>

Contoh dengan antarmuka UART untuk mengatur agar posisi *default* motor servo adalah 90° (90 = 5AH):

User : 08H 5AH

Berikut ini contoh *pseudo code* C untuk menggunakan perintah ini dengan antarmuka I<sup>2</sup>C (misalkan alamat I<sup>2</sup>C = E0H):

```

i2c_start();           // Start Condition
i2c_write(0xE0);      // Tulis ke modul FLAME DETECTOR
i2c_write(0x08);      // Perintah "Set Default"
i2c_write(90);        // Sudut = 90 derajat
i2c_stop();           // Stop Condition

```

## 4. PROSEDUR PENGUJIAN

1. Pasang sensor ke motor servo.
2. Hubungkan motor servo ke konektor J5 DT-SENSE FLAME DETECTOR.
3. Hubungkan sensor ke konektor J4 DT-SENSE FLAME DETECTOR.
4. Hubungkan sumber catu daya ke modul DT-SENSE FLAME DETECTOR.
5. Jika modul DT-SENSE FLAME DETECTOR berhasil melakukan *power-up reset* dengan baik, maka motor servo akan bergerak ke posisi *default*.
6. Nyalakan lilin dan atur agar tinggi sesuai dengan tinggi mata sensor, dan letakkan pada jarak antara 5 – 40 cm di depan sensor dengan posisi terserah asalkan masih pada jangkauan sapuan motor servo dan sensor.

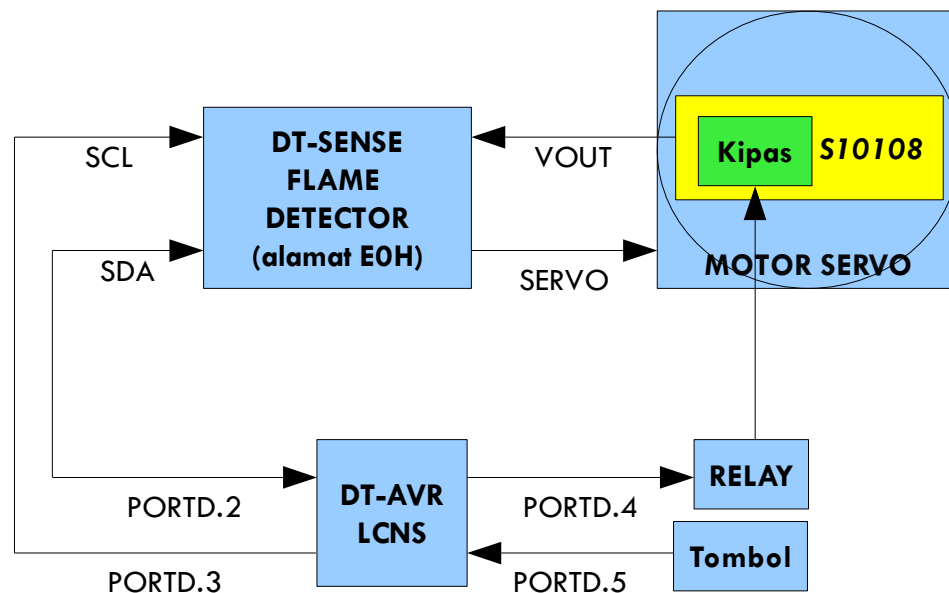
7. Kirimkan perintah *autoscan* melalui antarmuka UART TTL.
8. Modul DT-SENSE FLAME DETECTOR akan melakukan proses *autoscan* dan jika berhasil mendeteksi api lilin, maka mata sensor akan terarah menuju ke posisi api lilin.

## 5. CONTOH APLIKASI DAN PROGRAM

Sebagai contoh aplikasi, dimisalkan modul DT-SENSE FLAME DETECTOR digunakan untuk mencari posisi api lilin dengan antarmuka I<sup>2</sup>C. Modul DT-AVR Low Cost Nano System (LCNS) digunakan sebagai *master*. DT-AVR LCNS bertugas untuk mengirimkan perintah *autoscan* ke DT-SENSE FLAME DETECTOR dan membaca hasil *autoscan* serta mengaktifkan kipas untuk mematikan api jika api ditemukan. Berikut koneksi antara modul-modul yang digunakan:

DT-AVR LCNS	Terhubung ke
PORTD.2	SDA DT-SENSE FLAME DETECTOR
PORTD.3	SCL DT-SENSE FLAME DETECTOR
PORTD.4	Relay untuk mengaktifkan kipas
PORTD.5	Tombol <i>active low</i> (salah satu kakinya dihubungkan ke ground)

DT-SENSE FLAME DETECTOR	Terhubung ke
J4	Sensor
J5	Motor Servo



Sebagai contoh program untuk aplikasi di atas, pada CD yang disertakan pada saat pembelian modul DT-SENSE FLAME DETECTOR disertakan program contoh.c yang ditulis dengan menggunakan CodeVisionAVR 1.25.2 versi evaluasi.

Pada program tersebut, DT-AVR LCNS akan membaca kondisi tombol. Jika tombol ditekan, maka DT-AVR LCNS mengirimkan perintah *autoscan* ke DT-

SENSE FLAME DETECTOR (alamat I<sup>2</sup>C = E0H). Setelah proses *autoscan* selesai dan ditemukan ada api, maka DT-AVR LCNS mengirimkan perintah untuk membaca nilai api dan kemudian mengaktifkan kipas untuk mematikannya (arah angin kipas searah dengan mata sensor). Setelah kipas menyala selama 2 detik, DT-AVR LCNS akan mematikan kipas dan kemudian mengirimkan perintah untuk membaca nilai api. Jika selisih nilai api sekarang dan nilai api awal (sebelum kipas dinyalakan) masih lebih besar dari batas yang kita tentukan, maka kipas akan diaktifkan lagi.

- ◆ *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***

**LAMPIRAN A.**  
**Skematik DT-SENSE FLAME DETECTOR**

