

DT-SENSE

AN224 – Antarmuka DT-Sense Gas Sensor menggunakan Arduino™ UNO

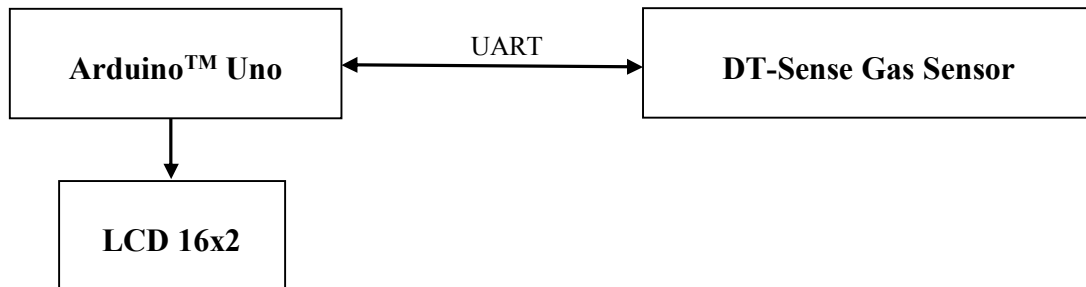
Oleh: Tim IE

Pengukuran kualitas udara dan kadar gas tertentu pada suatu area atau ruangan sekarang bisa kita ukur menggunakan modul DT-Sense Gas Sensor *series*. Ada beberapa jenis gas sensor yang bisa kita gunakan dalam mengukur kadar gas maupun kualitas udara. Antarmuka dan komunikasi pada modul DT-Sense Gas Sensor *series* semuanya sama, ada dua cara antarmuka yang bisa kita pake diantaranya adalah antarmuka menggunakan komunikasi serial UART dan antarmuka menggunakan komunikasi I²C. Pada AN kali ini kita akan membahas kedua antarmuka tersebut pada modul *board* Arduino™ Uno.

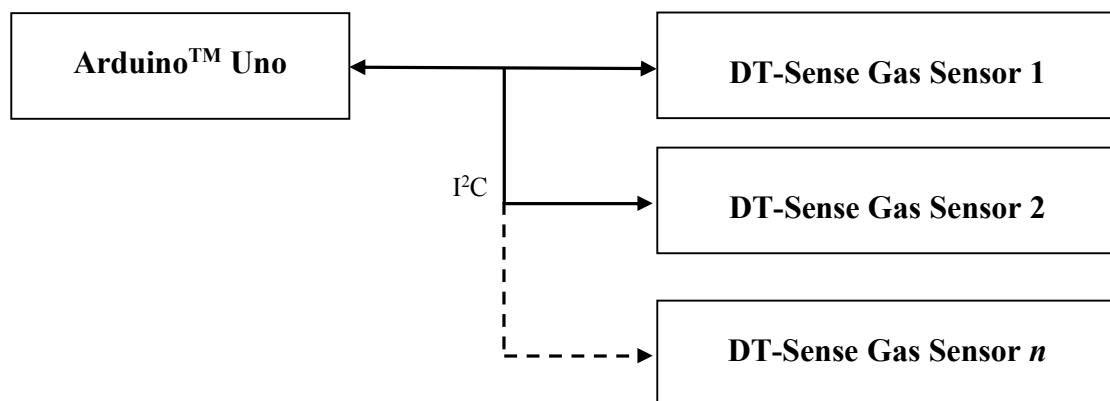
Berikut adalah modul yang digunakan pada aplikasi ini:

- 1 unit modul board Arduino™ Uno R3.
- 1 unit modul DT-SENSE CARBON MONOXIDE SENSOR.
- 1 unit modul DT-SENSE CARBON DIOXIDE SENSOR.
- 1 *power supply* 5V.
- 1 *power supply* 12V.
- 1 LCD 16x2.
- 1 Potensiometer 10K ohm.
- Kabel jumper.

Blok diagram sistem adalah sebagai berikut:



Gambar 1
Blok Diagram Antarmuka UART

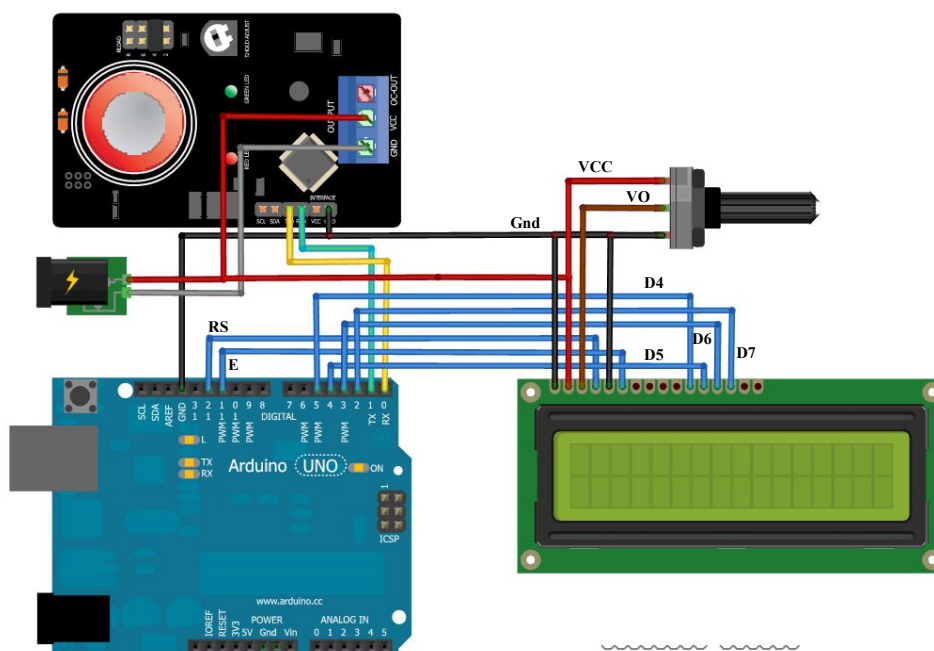


Gambar 2
Blok Diagram Antarmuka I²C

Hubungan kabel antar modul pada aplikasi ini adalah sebagai berikut:

DT-SENSE GAS SENSOR	ARDUINO™ UNO	LCD 16x2
GND (J3 pin 1)	GND	GND, RW
RX TTL(J3 pin 3)	1 (TX)	-
TX TTL (J3 pin 4)	0 (RX)	-
VCC (J3/J5)	-	VCC
-	12	RS
-	11	E
-	5	D4
-	4	D5
-	3	D6
-	2	D7

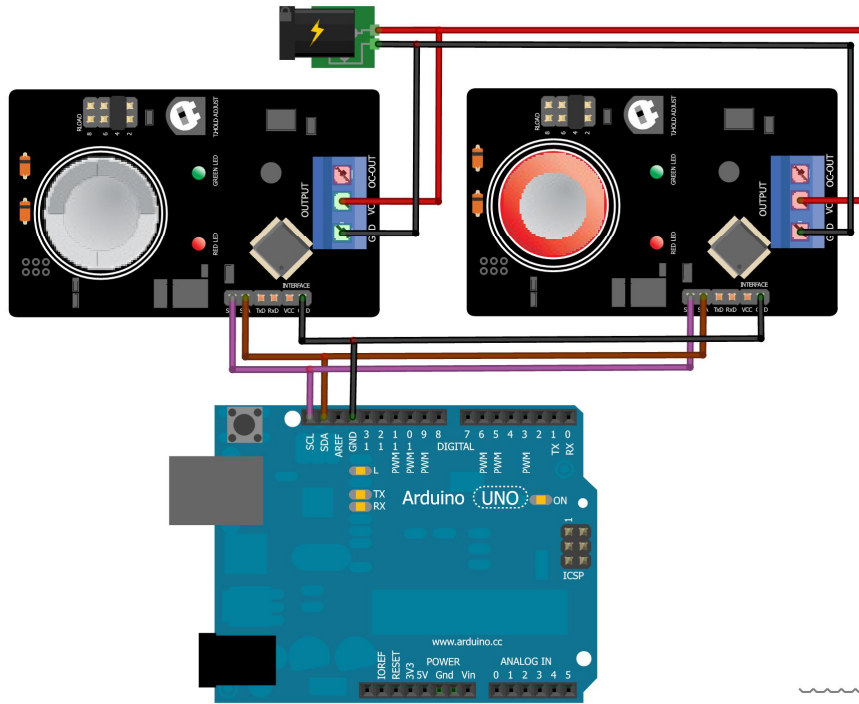
Tabel 1
Hubungan Kabel Antarmuka UART



Gambar 3
Ilustrasi Koneksi Antar Modul Menggunakan Komunikasi UART

DT-SENSE GAS SENSOR	ARDUINO™ UNO
GND (J3 pin 1)	GND
SDA (J3 pin 5)	SDA
SCL (J3 pin 6)	SCL

Tabel 2
Hubungan Kabel Antarmuka I²C



Gambar 4
Ilustrasi Koneksi Antar Modul Menggunakan Komunikasi I²C

Catu daya yang digunakan ada 2 macam: catu daya 12 VDC untuk modul Arduino™ Uno sedangkan catu daya 5 VDC untuk modul DT-Sense Gas Sensor (DT-SENSE CARBON MONOXIDE SENSOR atau DT-SENSE CARBON DIOXIDE SENSOR). Kita bisa saja menggunakan keluaran tegangan 5V dari modul Arduino™ Uno untuk memberi tegangan 5V pada modul DT-Sense Gas Sensor, tetapi untuk waktu penggunaan yang lama dan terus menerus, modul DT-Sense Gas Sensor akan sangat membebani rangkaian regulator pada modul Arduino™ Uno yang bisa menyebabkan panas.

Setelah semua modul terhubung, periksa kembali semua koneksi pin. Pastikan koneksi VCC dan GND tidak terhubung singkat. Untuk koneksi menggunakan komunikasi I²C, pasang *jumper pull-up* pada salah satu modul saja bila antarmuka lebih dari satu modul DT-Sense Gas Sensor.

Apabila semua koneksi sudah dipastikan benar, hubungkan kabel USB dari PC ke Arduino Uno, selanjutnya pastikan LED power pada Arduino Uno menyala dan PC sudah mengenali modul Arduino Uno.

Untuk koneksi menggunakan komunikasi UART, buka *file* "uart.ino" pada *folder* "DT-Sense uart" menggunakan program Arduino IDE.

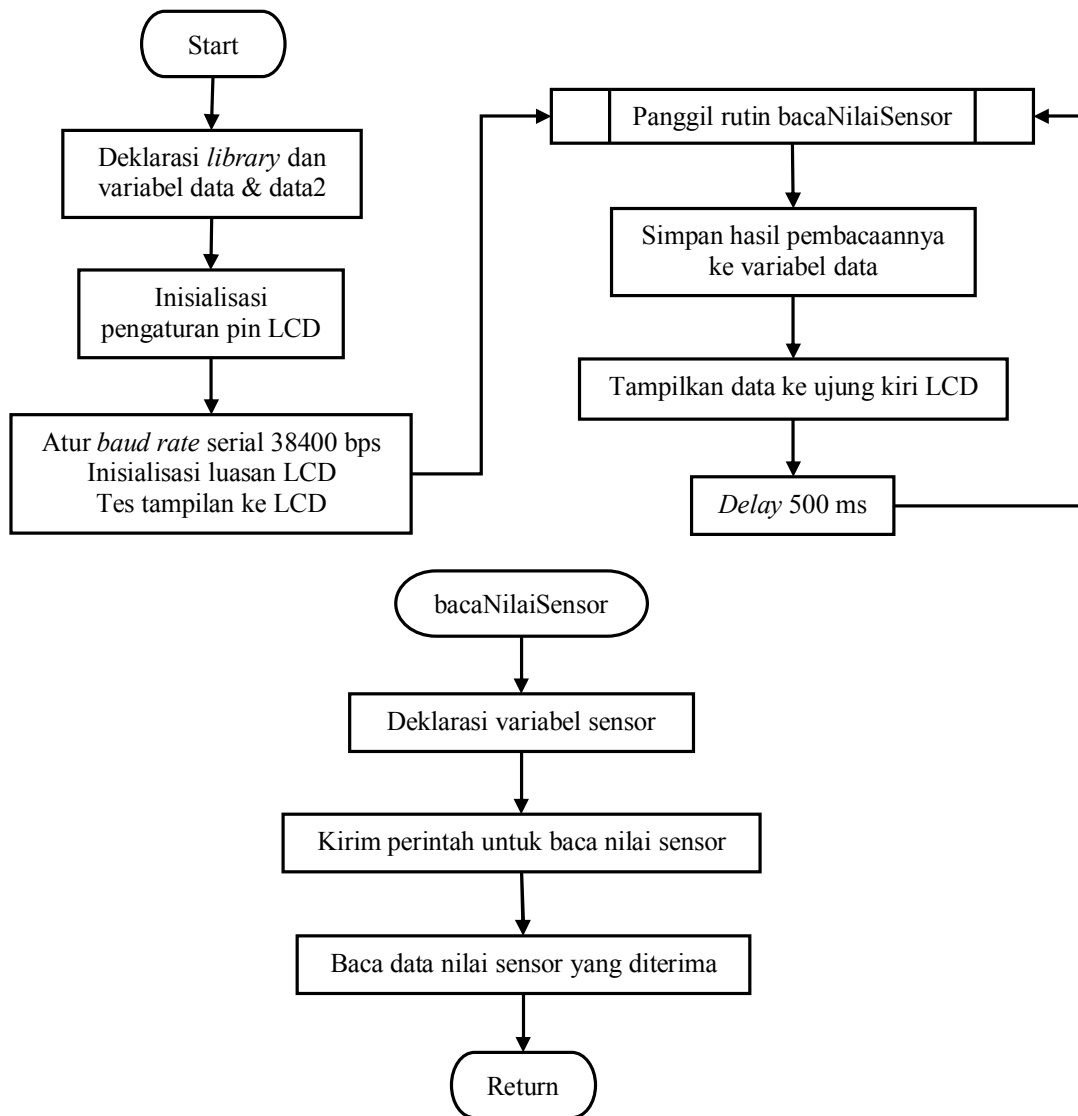
Untuk koneksi menggunakan komunikasi I²C, gantikan *folder* "i2cmaster" pada *folder* "DT-Sense i2c" ke *folder library* yang biasanya terdapat pada "C:\Program Files\Arduino\libraries". *Folder* "i2cmaster" berisi *library* I2C Master buatan Peter Fleury (<http://homepage.hispeed.ch/peterfleury/avr-software.html>) yang telah kami sesuaikan untuk Arduino dan Uno. Lalu buka *file* "i2c.ino" pada *folder* "DT-Sense i2c\i2c" menggunakan program Arduino IDE.

Lakukan *upload sketch* ke Arduino Uno dan tunggu beberapa saat. Apabila instalasi *driver* sudah benar dan PC dapat mengenali Arduino Uno, maka program akan di-*upload* secara otomatis.

Untuk koneksi menggunakan komunikasi UART, data pembacaan sensor gas akan ditampilkan pada layar LCD 16x2.

Untuk koneksi menggunakan komunikasi I²C, data pembacaan sensor gas akan ditampilkan melalui Serial Monitor pada program Arduino IDE. Dengan model antarmuka menggunakan komunikasi I²C ini kita bisa dengan lebih mudah menghubungkan secara paralel lebih dari satu modul DT-Sense Gas Sensor.

Flowchart dari `uart.ino` adalah sebagai berikut:



Gambar 5
Flowchart `uart.ino`

Penjelasan untuk alur program `uart.ino`:

1. Program menggunakan *library* `LiquidCrystal` bawaan Arduino.
`#include <LiquidCrystal.h>`
2. Setelah deklarasi *library* yang digunakan, program akan mendeklarasikan variabel `data` dan `data2`.
`int data, data2;`
3. Inisialisasi pin LCD dijalankan untuk menentukan pin yang digunakan
`LiquidCrystal lcd(12, 11, 5, 4, 3, 2);`
4. Program juga akan menentukan *baud rate* komunikasi serial menjadi 38400 bps, menentukan ukuran LCD, dan menguji tampilan LCD.

```

void setup() {
  Serial.begin(38400);
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("Test, Baca!");
}

```

- Pada bagian loop, program akan memanggil rutin `bacaNilaiSensor()` untuk mendapatkan hasil pembacaan. Hasil tersebut disimpan dalam variabel `data`.

```
void loop() {
    data = bacaNilaiSensor();
```

- Lalu program akan menampilkan variabel `data` ke LCD.

```
// set the cursor to column 0, line 1
lcd.setCursor(0, 1);
lcd.print("Data = ");
lcd.print(data/1000 %10 + 0x30);
lcd.print(data/100 %10 + 0x30);
lcd.print(data/10 %10 + 0x30);
lcd.print(data %10 + 0x30);
```

- Sebelum program kembali ke langkah 5, program akan menjalankan waktu tunda sebesar 500 ms sebagai jeda antar pembacaan.

```
delay(500);
```

```
}
```

- Rutin `bacaNilaiSensor` diawali dengan deklarasi variabel sensor.

```
unsigned int sensor;
```

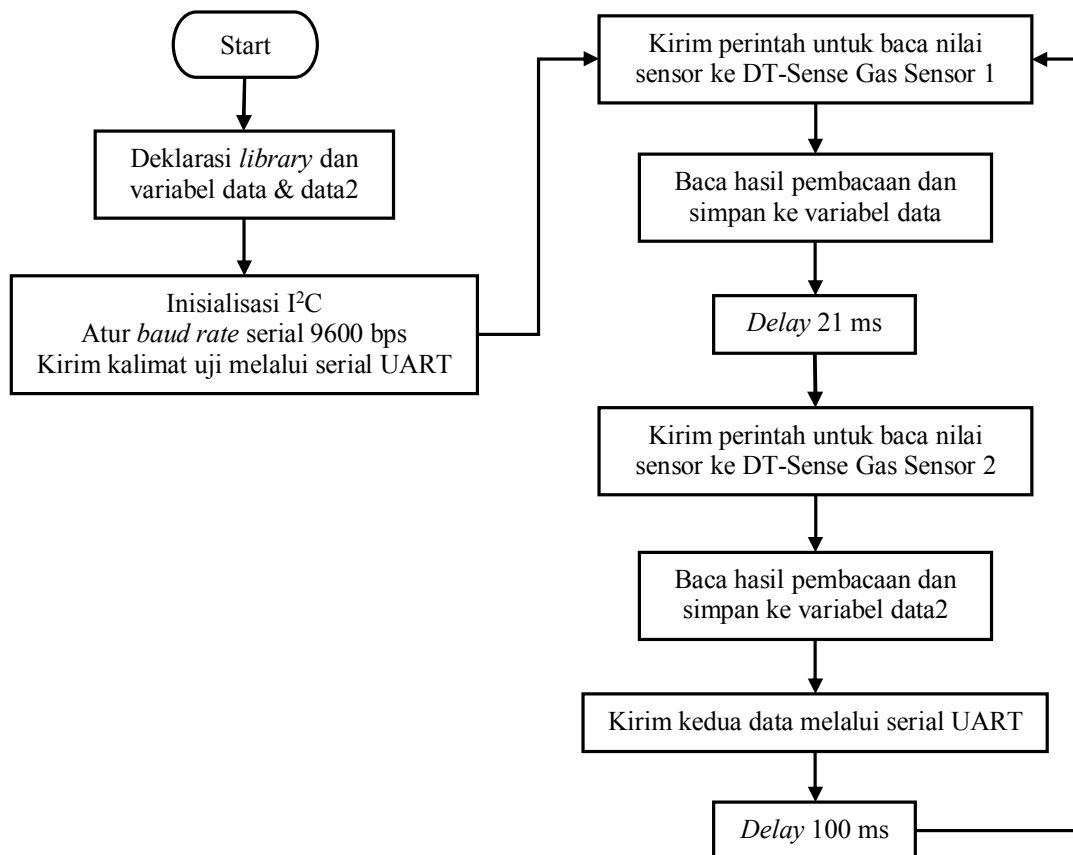
- Lalu program akan mengirimkan perintah `0x41` ke DT-Sense Gas Sensor untuk mendapatkan hasil pembacaan sensor.

```
Serial.write(0x41);
```

- Hasil pembacaan akan disimpan pada variabel sensor.

```
sensor=Serial.read();
sensor = sensor*256 + Serial.read();
```

Flowchart dari `i2c.ino` adalah sebagai berikut:



Gambar 6
Flowchart i2c.ino Bagian 1

Penjelasan untuk alur program i2c.ino:

1. Program menggunakan *library* i2cmaster buatan Peter Fleury (<http://homepage.hispeed.ch/peterfleury/avr-software.html>) yang telah kami sesuaikan untuk Arduino dan Uno.

```
#include <i2cmaster.h>
```

2. Setelah deklarasi *library* yang digunakan, program akan mendeklarasikan variabel data dan data2.

```
int data, data2;
```

3. Program juga akan melakukan inisialisasi jalur komunikasi I²C, menentukan *baud rate* komunikasi serial menjadi 9600 bps, dan mengirimkan kalimat uji melalui serial UART.

```
void setup()
{
  i2c_init();           //I2C init
  Serial.begin(9600);
  Serial.println(" Test Baca Sensor..."); //send data to Serial
}
```

4. Pada bagian loop, program akan mengirimkan perintah untuk mendapatkan hasil pembacaan sensor pertama. Hasil tersebut disimpan dalam variabel data. Setelah proses pembacaan ini, program memberi waktu jeda 21 ms sebelum menjalankan perintah berikutnya.

```
void loop()
{
  i2c_start(0xE0); // Alamat I2C modul gas Sensor A
  i2c_write(0x41);
  i2c_stop();
  delay(10);
  i2c_start(0xE1);
  data = (int) i2c_readAck();
  data = data*255 + i2c_readNak();
  i2c_stop();

  delay(21);
}
```

5. Program akan melanjutkan dengan mengirimkan perintah untuk mendapatkan hasil pembacaan sensor kedua. Hasil tersebut disimpan dalam variabel data2.

```
i2c_start(0xE2); // Alamat I2C modul gas Sensor B
i2c_write(0x41);
i2c_stop();
delay(10);
i2c_start(0xE3);
data2 = (int) i2c_readAck();
data2 = data2*255 + i2c_readNak();
i2c_stop();
```

6. Setelah kedua data diperoleh, keduanya akan dikirimkan ke komputer melalui serial UART. Dan sebelum program kembali ke langkah 4, program akan menjalankan waktu tunda sebesar 100 ms sebagai jeda antar pembacaan.

```
Serial.print("Gas Sensor A (Data) : ");
Serial.println(data, DEC); //send data gas sensor A to Serial
Serial.print("Gas Sensor B (Data2) : ");
Serial.println(data2, DEC); //send data2 gas sensor A to Serial
delay(100);
}
```

Listing program aplikasi ini terdapat pada AN224.zip.

Selamat berinovasi!

*All trademarks, trade names, company names, and product names are the property of their respective owners.
All softwares are copyright by their respective software publishers and/or creators.*