

DT-SENSE *Application Note*

AN140 - How 2 Use DT-SENSE USIRR with DT-AVR Low Cost Nano System

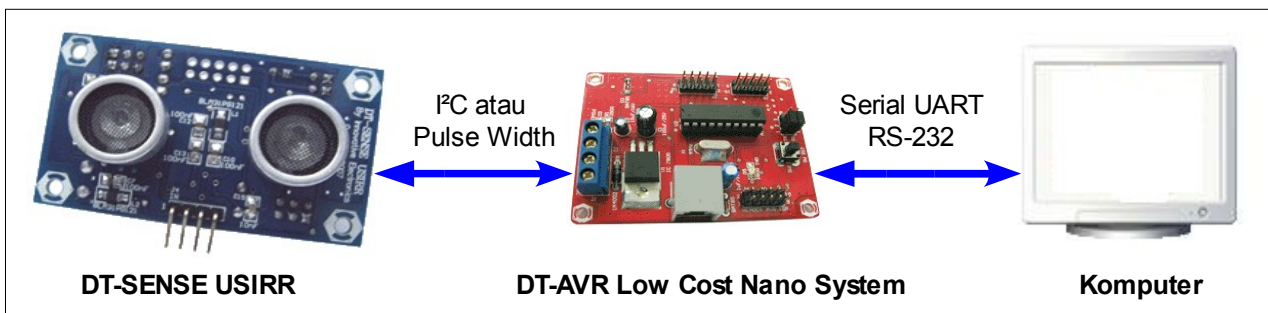
Oleh: Tim IE

Application note ini mengulas tentang cara penggunaan DT-SENSE UltraSonic and InfraRed Ranger (USIRR) yang baik dan benar dengan bantuan beberapa contoh program. Modul kontroler yang digunakan untuk berinteraksi dengan DT-SENSE USIRR ini adalah DT-AVR Low Cost Nano System. Program-program dalam AN ini ditulis dalam bahasa pemrograman C menggunakan IDE pemrograman CodeVisionAVR®.

Modul-modul yang diperlukan dalam aplikasi ini adalah:

- 1 buah DT-SENSE USIRR.
- 1 buah DT-AVR Low Cost Nano System (menggunakan AT90S2313).

Diagram blok sistem untuk contoh program dalam AN ini adalah sebagai berikut:



Gambar 1
Diagram Blok Sistem

Modul DT-SENSE USIRR dapat mengukur jarak melalui dua media ukur yaitu ultrasonik dan *infrared* (dengan tambahan sensor *infrared ranger* Sharp GP2D12). Sensor ultrasonik sudah terpasang pada modul, sedangkan sensor *infrared ranger* dapat ditambahkan dengan jumlah maksimal 2 buah. Pengambilan data pengukuran dapat dilakukan melalui 2 pilihan antarmuka yaitu melalui I²C atau *pulse width*. Dalam AN ini akan diberikan contoh penggunaan kedua media ukur (ultrasonik dan *infrared*) dan juga penggunaan kedua antarmuka (I²C dan *pulse width*) untuk pengambilan data pengukuran. Data pengukuran akan dikirim ke komputer secara serial UART RS-232 dan dapat ditampilkan di monitor komputer melalui program HyperTerminal® (atau sejenisnya).

Berikut daftar *jumper* dan *header* pada *board* DT-SENSE USIRR:

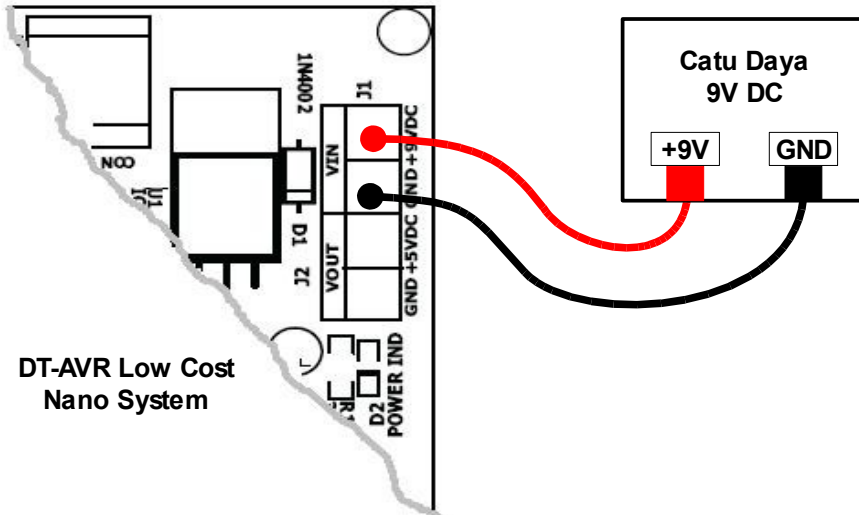
Nama	Jumlah Pin	Keterangan
J1	2	Jalur SDA (data) dan SCL (<i>clock</i>) untuk komunikasi I ² C
J3	6	<i>Jumper</i> untuk pengaturan alamat terprogram komunikasi I ² C
J4	4	Jalur VCC, GND, SIG, dan Busy/Ready
J5	3	<i>Header</i> untuk sensor <i>infrared ranger</i> 1 (VCC, GND, VOUT)
J6	3	<i>Header</i> untuk sensor <i>infrared ranger</i> 2 (VCC, GND, VOUT)
J7 & J8	4	<i>Jumper</i> untuk mengaktifkan R <i>pull up</i> pada jalur komunikasi I ² C

Tabel 1
Jumper Dan Header Pada Board DT-SENSE USIRR

Pembacaan data jarak melalui antarmuka *pulse width*. Percobaan pertama ini akan menggunakan antarmuka *pulse width* untuk pembacaan data jarak dari DT-SENSE USIRR. Antarmuka *pulse width* ini hanya membutuhkan 1 pin I/O dari DT-AVR Low Cost Nano System dan hanya menggunakan header J4 pada DT-SENSE USIRR. Berikut akan dijelaskan secara *step by step* untuk percobaan ini:

1. Menghubungkan catu daya.

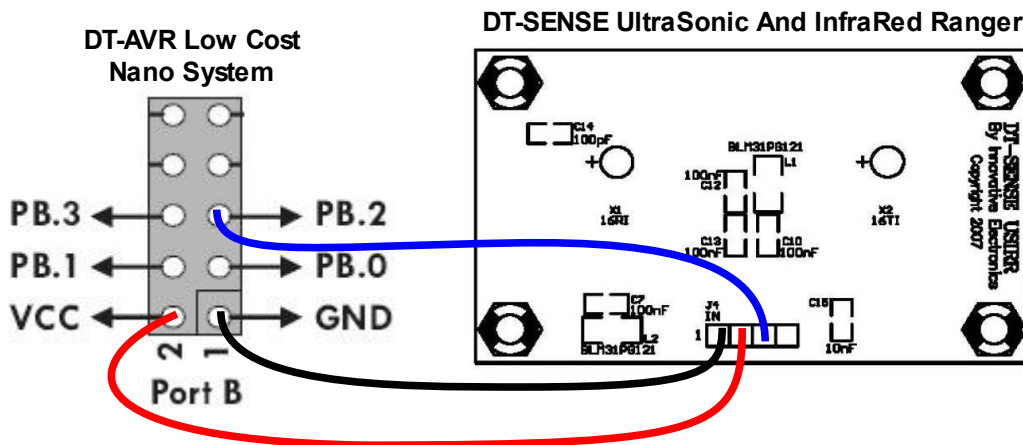
Modul yang perlu dihubungkan ke catu daya adalah DT- AVR Low Cost Nano System, sedangkan DT-SENSE USIRR dapat mengambil catu daya dari DT-AVR Low Cost Nano System melalui J7 (akan dijelaskan pada langkah berikutnya).



Gambar 2
Menghubungkan DT- AVR Low Cost Nano System ke Catu Daya 9V DC

2. Menghubungkan DT-SENSE USIRR.

Selain pin 3 (SIG) untuk antarmuka *pulse width*, pin 1 dan pin 2 header J4 DT-SENSE USIRR juga harus dihubungkan ke J7 (Port B) DT-AVR Low Cost Nano System untuk menyalurkan catu daya ke DT-SENSE USIRR.



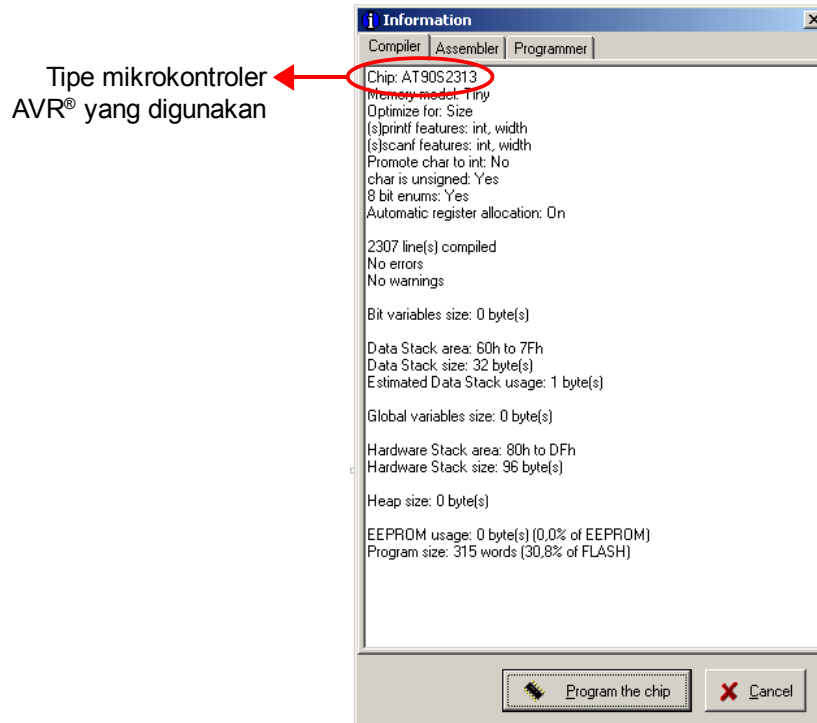
Gambar 3
Menghubungkan DT SENSE USIRR ke DT-AVR Low Cost Nano System

3. Setelah kedua modul di atas selesai dihubungkan dan siap, maka langkah selanjutnya adalah penulisan program ke DT-AVR Low Cost Nano System (mikrokontroler AT90S2313).

Program dituliskan ke DT-AVR Low Cost Nano System menggunakan DT-HiQ AVR USB ISP (atau AVR® *in-system programmer* lainnya). Langkah penulisan program ke DT-AVR Low Cost Nano System adalah sebagai berikut:

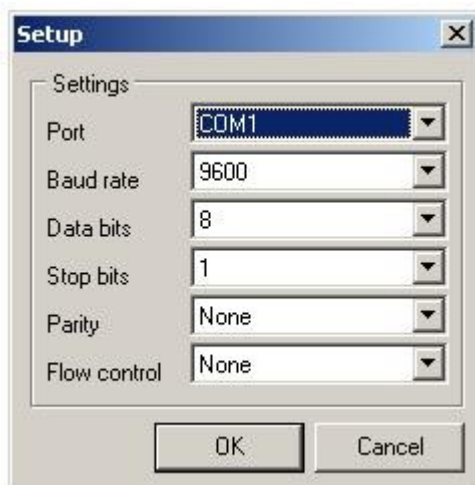
- ◆ Hubungkan DT-HiQ AVR USB ISP ke HEADER AVR ISP (J6) DT-AVR Low Cost Nano System. Lalu hubungkan DT-HiQ AVR USB ISP ke komputer melalui *port* USB.

- ◆ Lakukan pengaturan pada CodeVisionAVR® untuk *in-system programmer* yang digunakan.
- ◆ Bukalah *project file* **Demo_Pulse.prj** di CodeVisionAVR®. *Demo_Pulse.prj* ini terdapat pada direktori *pulse_width*.
- ◆ Pada *menu bar*, pilih menu “**Project**” - “**Configure**”. Jendela *dialog* baru akan muncul, lalu pilihlah tab “**After Build**” dan centang *checkbox* “**Program the Chip**”, serta klik tombol “OK”.
- ◆ Setelah itu tekan tombol *keyboard* **Shift + F9** atau pilih menu “**Project**” - “**Build**” dan akan muncul jendela *dialog* sebagai berikut:

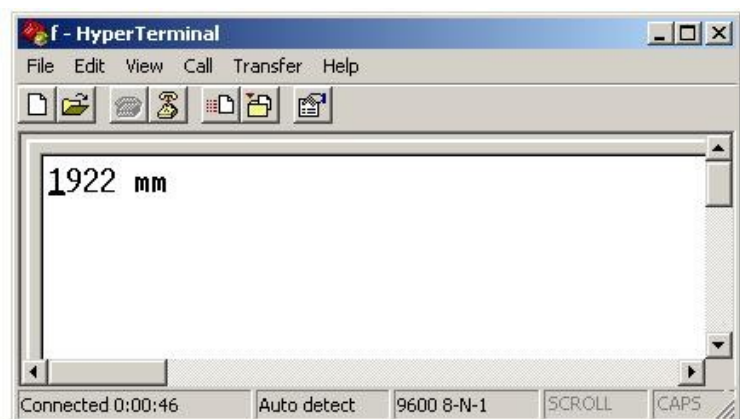


Gambar 4
Tampilan Jendela Pemrograman

- ◆ Klik tombol “**Program the chip**” untuk menulis program ke DT-AVR Low Cost Nano System.
- ◆ Jika program berhasil ditulis ke DT-AVR Low Cost Nano System maka DT-AVR Low Cost Nano System akan mengirimkan data pengukuran ke komputer melalui *port* serial. Data pengukuran ini dapat ditampilkan menggunakan *software* HyperTerminal® atau *tool* Terminal dari CodeVisionAVR®. Berikut pengaturan komunikasi serial *COM port* (sesuaikan nomor *port* dengan koneksi *hardware*) dan tampilan data jarak di jendela HyperTerminal®:



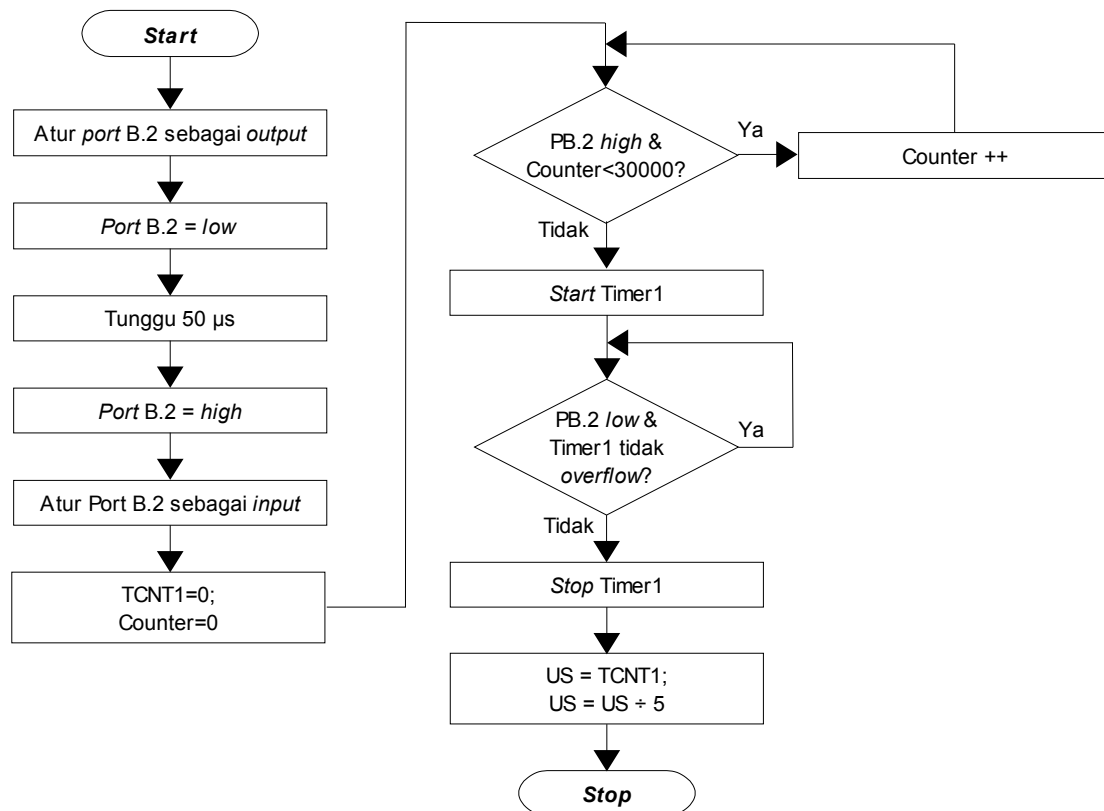
(a)



(b)

Gambar 5
(a) Pengaturan *COM port*, (b) Tampilan Data Jarak Dalam Satuan Milimeter

- ◆ Penjelasan program:
Pembacaan data jarak dengan antarmuka *pulse width* relatif sederhana. Pertama, mikrokontroler mengatur pin PB.2 sebagai *output* dan menggunakannya untuk mengirim pulsa *trigger* (logika *low*) selama 50 mikrodetik ke pin SIG. Setelah itu, pin PB.2 diatur lagi menjadi *input* dengan mengubah nilai *register* DDRB.2. Lalu program menunggu logika pada pin PB.2 (SIG) *low*. Pada saat kondisi pin SIG berubah ke logika *low*, program akan menjalankan Timer1. Timer1 akan dimatikan ketika kondisi logika pada pin SIG kembali ke logika *high*. Lama waktu pin SIG dalam kondisi logika *low* inilah yang menyatakan hasil pengukuran (jarak terukur) dengan konversi 1 mm tiap 10 mikrodetik. Pada program ini Timer1 diatur untuk menghitung lebar pulsa dengan nilai cacahan 2 mikrodetik. DT-AVR Low Cost Nano System menggunakan *crystal* 4 MHz sebagai sumber *clock* AT90S2313, sehingga register TCCR1B harus diisi dengan nilai 0x02 untuk memperoleh nilai cacahan 2 mikrodetik (keterangan lengkap mengenai pengaturan Timer1 terdapat di datasheet AT90S2313). Setelah Timer1 selesai mencacah maka register TCNT1 akan berisi data yang menyatakan jarak yang terukur dengan konversi 1 mm tiap 5 LSB. Misalnya data dalam register TCNT1 adalah 50 maka jarak yang terukur adalah 10 milimeter.



Gambar 6
Flowchart Program Untuk Antarmuka Pulse Width

Pembacaan data jarak melalui antarmuka *pulse width* dapat juga dilakukan tanpa menggunakan fungsi *timer*, yaitu dengan menggunakan metoda *delay*. Pembacaan data jarak menggunakan metoda *delay* kurang begitu akurat jika dibandingkan dengan pembacaan data menggunakan fungsi *timer*. Program `demo_pulse.c` dapat dimodifikasi untuk menerapkan metode *delay* dengan mengubah bagian berikut:

```

TCCR1B=0x02;          // start timer (timer period = 2µs)
while ((!SIG_in) && !(TIFR&0x80));
TCCR1B=0x00;          // stop timer
US = TCNT1;           // save timer value to variable US
US = US/5;
  
```

diubah menjadi:

```

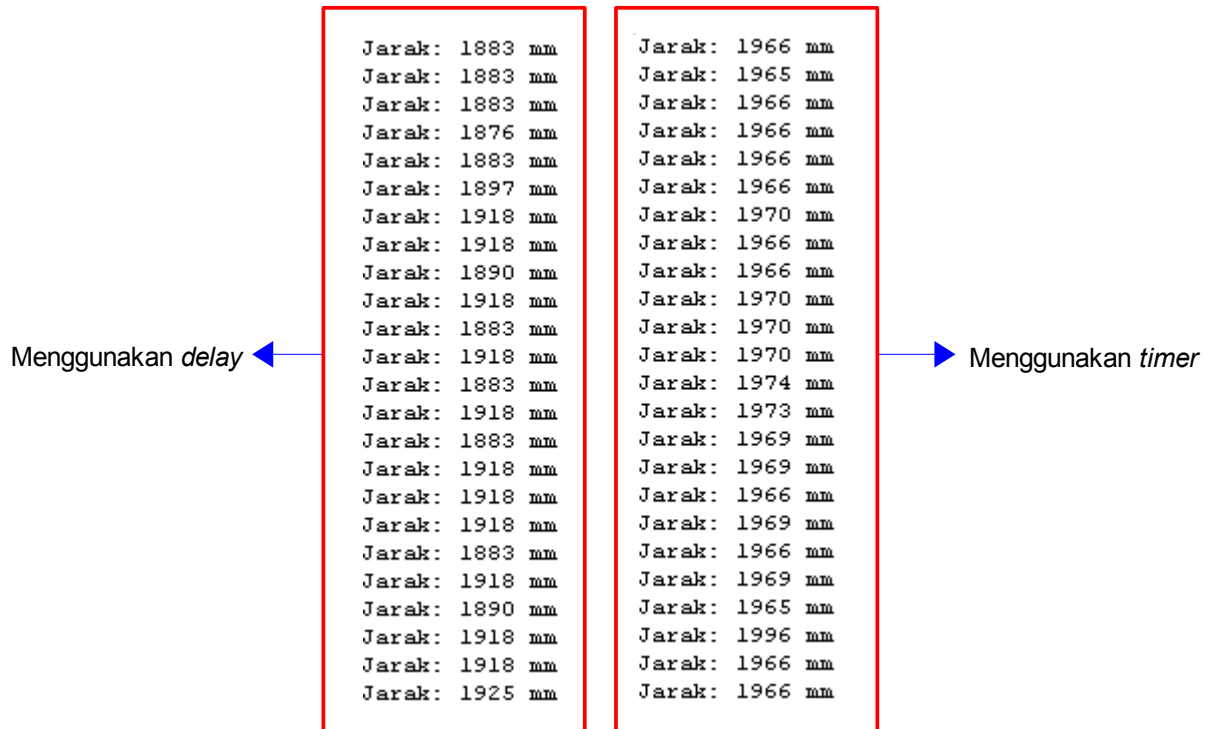
US=0;
while ((!SIG_in) && US<40000)
  
```

```

{
    delay_us (70);
    US++;
}
US = US*7;

```

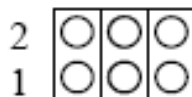
Pada bagian program yang diubah di atas, nilai variabel US dinaikkan setiap 70 mikrodetik sehingga data jarak yang terbaca harus dikali dengan 7. Perbandingan data hasil pengukuran menggunakan fitur *timer* dengan data hasil pengukuran menggunakan metoda *delay* adalah sebagai berikut:



Gambar 7
Perbandingan Hasil Pembacaan Data Jarak Antara Program Yang Menggunakan Delay Dengan Program Yang Menggunakan Timer

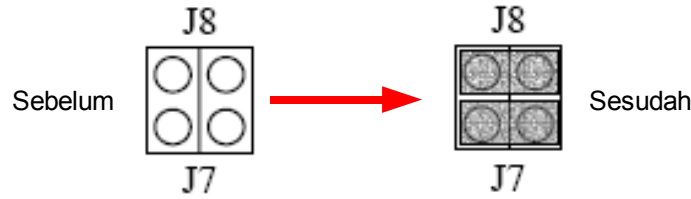
Setelah kita mencoba antarmuka *pulse width*, sekarang kita akan mencoba membaca data jarak hasil pengukuran DT-SENSE USIRR menggunakan antarmuka I²C. Pada percobaan dengan antarmuka I²C ini, kita perlu memperhatikan *jumper / header* J1, J3, J4, J7, dan J8 pada DT-SENSE USIRR. Berikut langkah-langkah untuk mencoba DT-SENSE USIRR dengan antarmuka I²C:

1. Alamat I²C DT-SENSE USIRR yang akan dipakai dalam program AN ini adalah 7 (*default*), sehingga pin J3 DT-SENSE USIRR dibiarkan saja (tidak diberi *jumper*).



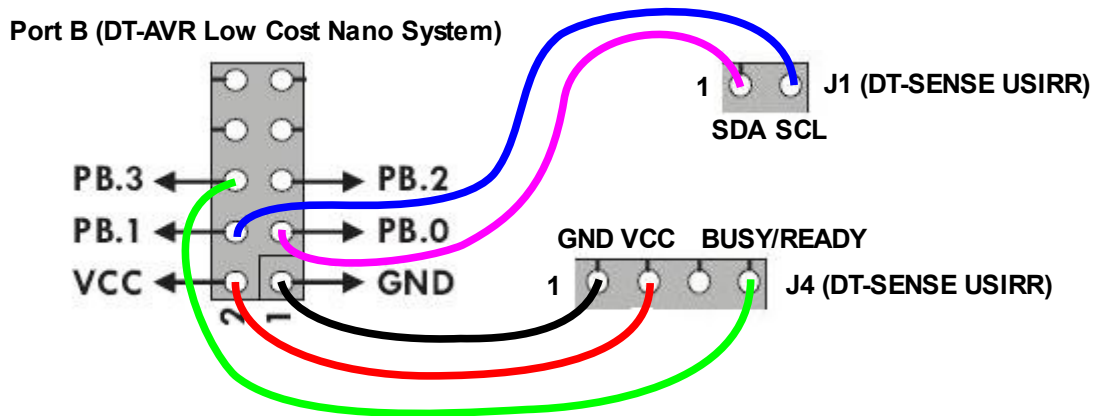
Gambar 8
Pin J3 DT-SENSE USIRR Tidak Diberi Jumper

- Mengaktifkan resistor *pull-up* untuk jalur SDA dan SCL dengan memasang *jumper* J7 dan J8.



Gambar 9
Pemasangan *Jumper* pada J7 dan J8 DT-SENSE USIRR

- Menghubungkan DT-SENSE UltraSonic and InfraRed Ranger (DT-SENSE USIRR) ke DT-AVR Low Cost Nano System:



Gambar 10
Hubungan DT-AVR Low Cost Nano System Dengan DT-SENSE USIRR Melalui Antarmuka I²C

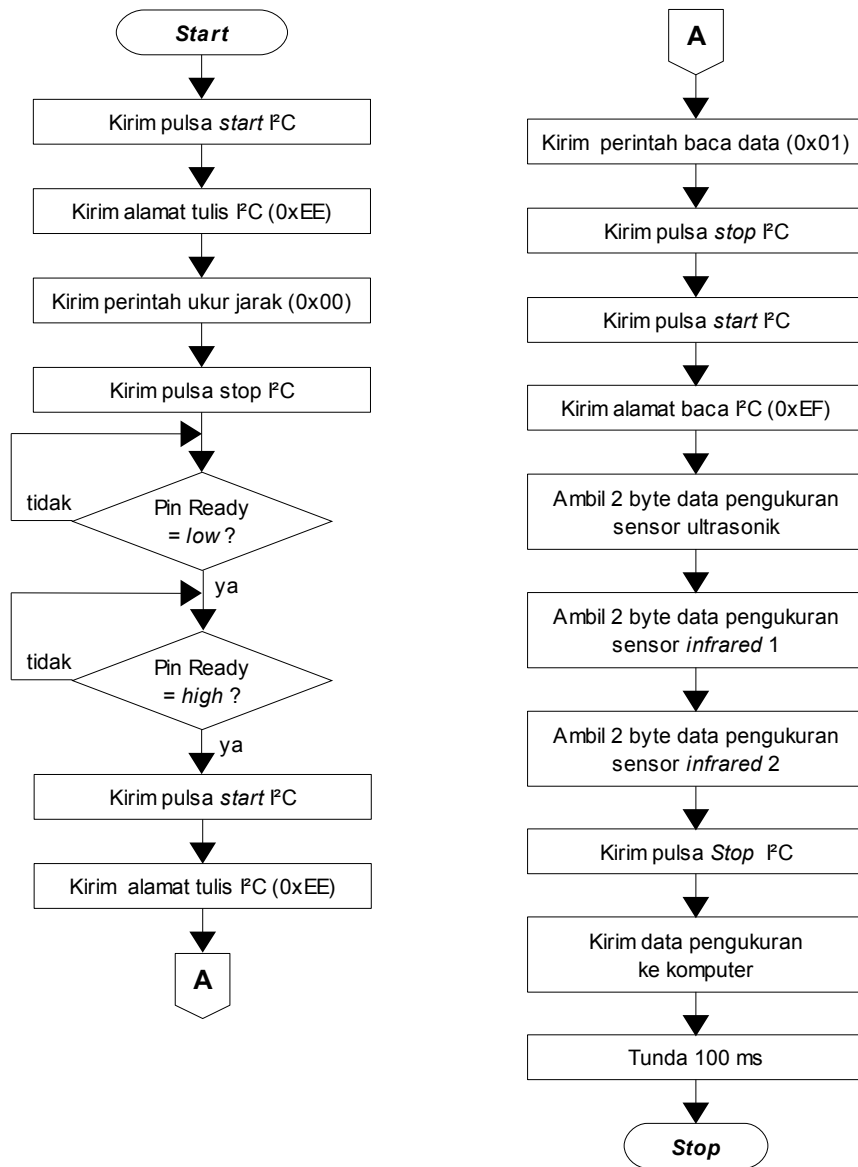
- Dalam percobaan ini akan ditambahkan 1 buah sensor *infrared ranger* pada J5 DT-SENSE UltraSonic and InfraRed Ranger. Pemasangan sensor ini adalah *optional*, pengukuran jarak sudah dapat dilakukan dengan hanya menggunakan media ultrasonik.



Gambar 11
Pemasangan Sensor *Infrared 1*

Jika seandainya diperlukan 2 buah sensor *infrared ranger*, maka pemasangan sensor *infrared ranger* yang kedua adalah serupa dengan gambar di atas namun dipasang pada J6.

- Program untuk mengambil data hasil pengukuran DT-SENSE UltraSonic and InfraRed Ranger melalui antarmuka I²C adalah berbeda dengan program yang menggunakan antarmuka lebar pulsa di atas. Data hasil pengukuran yang diperoleh melalui antarmuka I²C ini adalah data jarak siap pakai dalam satuan milimeter. *Flowchart* dari proses pembacaan data jarak dalam program untuk percobaan yang menggunakan antarmuka I²C ini terdapat pada Gambar 12.



Gambar 12
Flowchart Dari Proses Pembacaan Data Jarak Dalam Program Untuk Antarmuka I²C

6. Penjelasan program untuk antarmuka I²C ini adalah sebagai berikut:
- Pada setiap pengiriman perintah secara I²C, hal pertama yang dilakukan adalah mengirim pulsa *start* I²C dan mengakhiri perintah dengan mengirim pulsa *stop* I²C.
 - Pengiriman perintah untuk memulai pengukuran jarak adalah:


```

i2c_start();
i2c_write(0xEE); // --> 0xEE adalah alamat tulis I2C DT-SENSE USIRR dalam AN ini
i2c_write(0);    // --> parameter = 0 artinya perintah untuk memulai pengukuran jarak
i2c_stop();
      
```
 - Setelah itu program menunggu pin READY berlogika *low*, yang menandakan DT-SENSE USIRR sedang bekerja. Setelah pin READY bernilai *low*, program akan menunggu lagi hingga pin READY kembali bernilai *high* yang menandakan proses pengukuran sudah selesai.
 - Langkah selanjutnya adalah membaca data pengukuran dari sensor ultrasonik, *infrared 1*, dan *infrared 2*. Proses pembacaan data dimulai dengan:


```

i2c_start();
i2c_write(0xEE);
i2c_write(1);    // --> parameter = 1 artinya perintah untuk membaca data
i2c_stop();
      
```

 Pembacaan data jarak dilakukan secara berurutan, yaitu dimulai dari data pengukuran sensor

ultrasonik, diikuti data dari sensor *infrared* 1, dan yang terakhir adalah data dari sensor *infrared* 2. Lebar data pengukuran dari masing-masing sensor adalah 16 bit (2 byte) dan dikirimkan oleh DT-SENSE USIRR per byte, sehingga program harus melakukan proses baca 2 kali untuk tiap data pengukuran sensor. Contoh pembacaan data pengukuran sensor ultrasonik:

```
k=i2c_read(1); // --> pembacaan byte ke-1 (MSB)
US = k * 256; //
k=i2c_read(1); // --> pembacaan byte ke-2 (LSB)
US = US + k;
```

Byte pertama yang dikirimkan oleh DT-SENSE USIRR adalah bagian MSB dari data jarak, sedangkan byte ke-2 adalah bagian LSB dari data jarak. Oleh karena itu untuk mendapat hasil akhir dari data jarak, data byte ke-1 dikali 256 dan ditambah dengan byte ke-2.

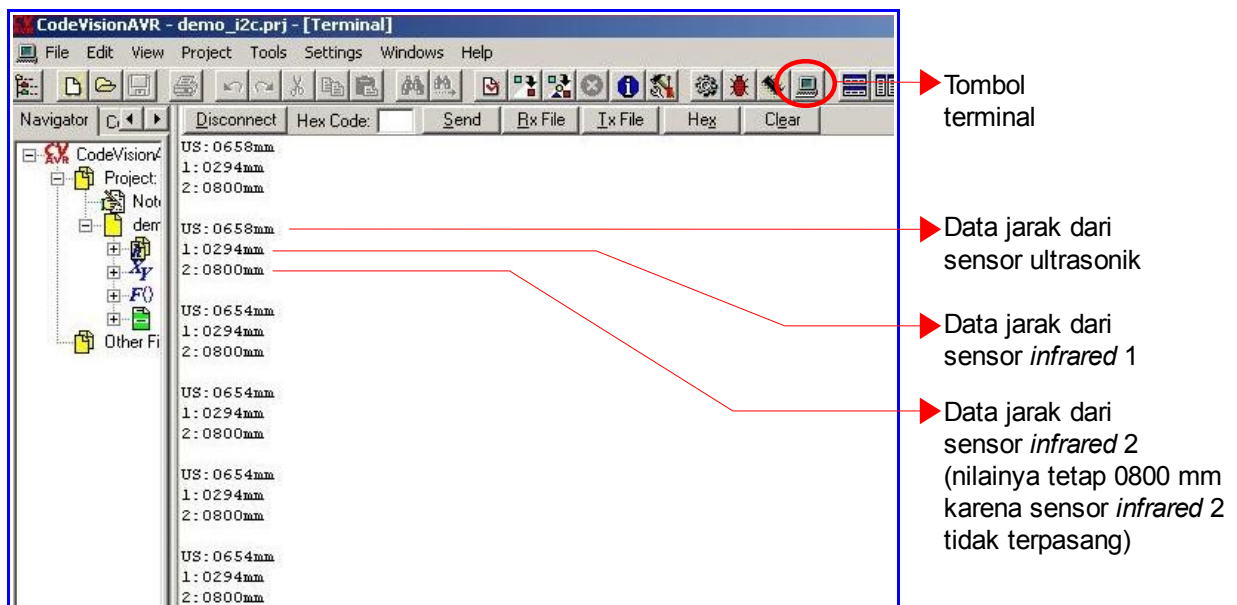
- (e) Data jarak yang diterima tadi akan dikirim oleh program ke komputer secara serial. Data jarak ini disimpan pada variabel US, IR1 dan IR2 yang bertipe *unsigned int* (0-65535). Banyak cara untuk mengirimkan data ke komputer, namun dalam program aplikasi ini data dikirim per karakter dengan menggunakan perintah *putchar()*. Contoh pengiriman data hasil pengukuran sensor ultrasonik:

```
putchar('U');
putchar('S');
putchar(':');
putchar((US/1000)%10 + 0x30); // --> ribuan
putchar((US/100)%10 + 0x30); // --> ratusan
putchar((US/10)%10 + 0x30); // --> puluhan
putchar(US%10 + 0x30); // --> satuan
```

Data jarak dikirim dalam satuan milimeter. Nilai hexadesimal '0x30' ditambahkan untuk mengubah nilai jarak menjadi karakter angka sesuai ASCII.

- (f) Waktu tunda untuk membaca data jarak selanjutnya adalah 100 mikro detik (tidak mutlak) dan program akan mengulangi keseluruhan proses ini lagi.

Bukalah *project file Demo_I2C.prj* pada CodeVisionAVR®, lalu tuliskan program tersebut ke DT-AVR Low Cost Nano System (mikrokontroler AT90S2313) sama seperti pada percobaan dengan antarmuka *pulse width*. Setelah DT-AVR Low Cost Nano System (AT90S2313) terisi program, aktifkan *tool* terminal CodeVisionAVR® dengan mengklik tombol bergambar komputer. Nilai-nilai parameter komunikasi serial diatur sama seperti pada HyperTerminal® yang diperlihatkan dalam Gambar 5. Hasil *running* program pembacaan jarak dengan antarmuka I²C di CodeVisionAVR® terdapat pada gambar di bawah ini:



Gambar 13
Tampilan Jendela Terminal CodeVisionAVR®

DT-SENSE UltraSonic and InfraRed Ranger memiliki sebuah *control register* dan 3 buah *register* kalibrasi. *Control register* akan menentukan mode operasi (*full*, *reduced*) serta menentukan sensor mana yang aktif (*ultrasonik*, *infrared* 1, *infrared* 2). *Control register* ini harus diisi agar DT-SENSE UltraSonic and InfraRed Ranger dapat

bekerja sesuai dengan kebutuhan pengguna. *Register* kalibrasi digunakan untuk melakukan kompensasi terhadap hasil pengukuran. Nilai yang tersimpan dalam *register* kalibrasi akan menentukan akurasi hasil pengukuran masing-masing sensor. Rutin penulisan nilai tertentu ke *control register* dan *register* kalibrasi adalah sebagai berikut:

```
void writeReg(uint8_t cReg,uint8_t cUS,uint8_t cIR1,uint8_t cIR2)
{
    i2c_start();
    i2c_write(0xEE);
    i2c_write(2);
    i2c_write(cReg);
    i2c_write(cUS);
    i2c_write(cIR1);
    i2c_write(cIR2);
    i2c_stop();
}
```

Parameter *cReg* adalah nilai *control register* yang akan dituliskan. Parameter *cUS*, *cIR1*, *cIR2* masing-masing adalah nilai kalibrasi sensor ultrasonik, *infrared 1*, dan *infrared 2* yang akan dituliskan. Secara *default* nilai kalibrasinya adalah *cUS=128*, *cIR1=128*, *cIR2=128*. Di dalam program AN ini, ketiga sensor diaktifkan dan DT-SENSE USIRR dioperasikan dalam mode *full operation*, sehingga nilai *cReg=0x0F*. Keterangan lebih lanjut tentang bagaimana mengatur nilai dari keempat parameter ini terdapat di sub bab 3.1.2.3 dalam manual DT-SENSE USIRR.

Listing program terdapat di *file AN140.ZIP*

Selamat berinovasi!

I²C is a registered trademark of Philips Semiconductors.
CodeVisionAVR is copyright by Pavel Haiduc, HP InfoTech s.r.l.
HyperTerminal is copyright by Hilgraeve Inc.
Atmel and AVR is a registered trademarks of Atmel Corporation.