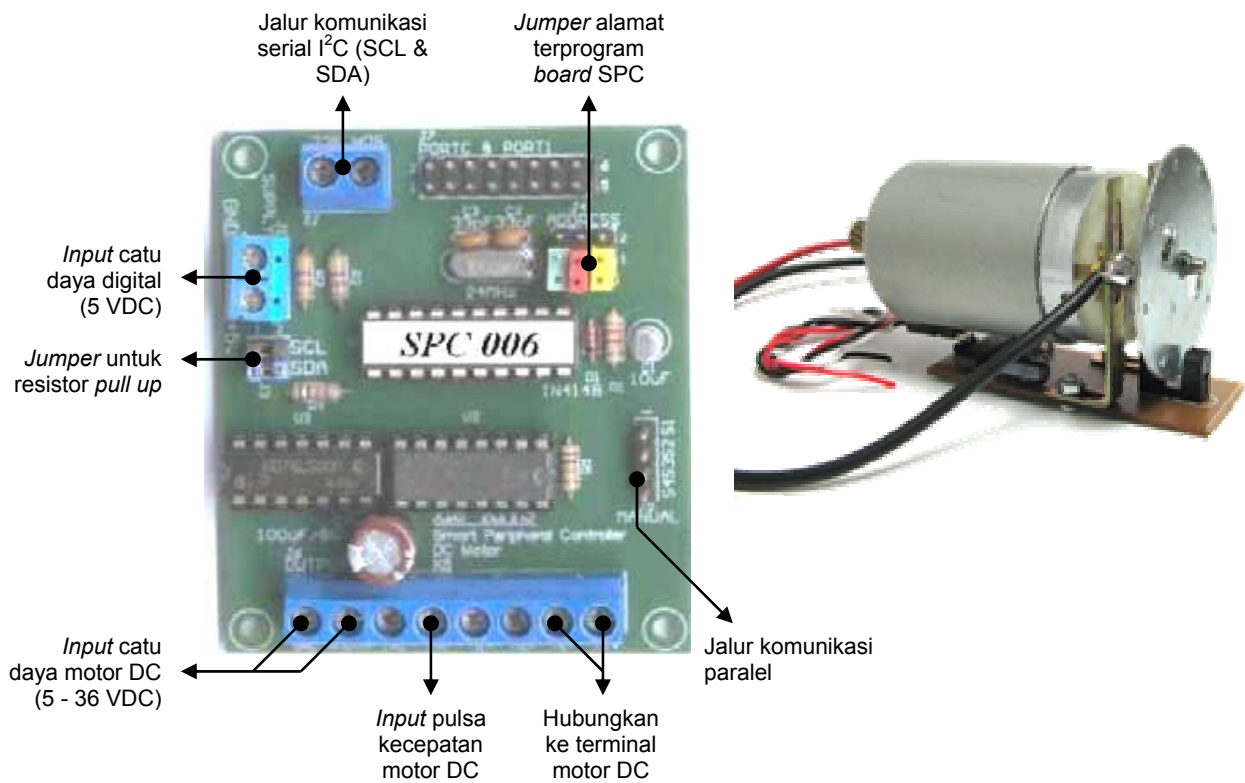


Sebuah motor DC seringkali digunakan sebagai divais penggerak dalam aplikasi robotika karena harganya relatif murah, tetapi sayangnya untuk mengendalikan motor DC ini relatif rumit. Artikel berikut ini akan menunjukkan bahwa mengendalikan motor DC dengan bantuan SPC DC Motor adalah sangat sederhana dan mudah. Modul mikrokontroler yang akan digunakan adalah DT-AVR Low Cost Nano System / Low Cost Micro System dan bahasa pemrograman BASIC dengan *compiler* BASCOM-AVR[®]. Aplikasi ini akan memberi contoh komunikasi SPC DC Motor dengan DT-AVR Low Cost Series, baik secara serial I²C maupun paralel. Aplikasi ini dapat dikembangkan menjadi suatu sistem pengendali motor DC sebagai penggerak dalam sebuah peralatan otomatis maupun robot.

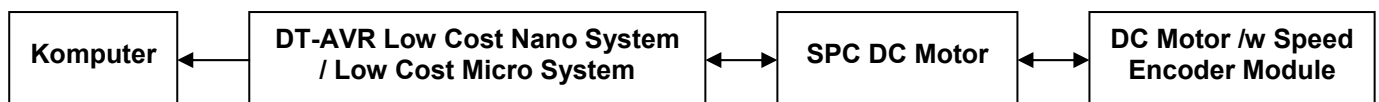
Komponen yang diperlukan:

- 1 DT-AVR Low Cost Nano System / Low Cost Micro System
- 1 SPC DC Motor
- 1 DC Motor /w Speed Encoder Module



Gambar 1
SPC DC Motor dan DC Motor /w Speed Encoder Module

Adapun blok diagram sistem secara keseluruhan adalah sebagai berikut:



Gambar 2
Blok Diagram AN81

Hubungan antara modul-modul tersebut secara I²C adalah sebagai berikut:

DT-AVR Low Cost Nano System / Low Cost Micro System	SPC DC Motor
+5VDC	+5V (J1)
GND	GND (J1)
PB.4*	SCL (J2)
PB.5*	SDA (J2)

* Pin ini tidak mutlak dan dapat diganti pin lain tetapi harus mengubah program

Tabel 1
Hubungan DT-AVR Low Cost Nano System / Low Cost Micro System dengan SPC DC Motor secara Serial I²C

Pasanglah *jumper* J3 dari SPC DC Motor untuk memberi resistor *pull up* pada jalur komunikasi serial I²C (SCL dan SDA). Dan aturlah juga *jumper* J4 dari SPC DC Motor agar alamat terprogram *board* SPC DC Motor menjadi 111b. Hubungkan DC Motor /w Speed Encoder Module ke SPC DC Motor sesuai dengan petunjuk dalam manual SPC DC Motor.

Gunakan kabel serial DT-AVR Low Cost Nano System / Low Cost Micro System untuk menghubungkan modul dengan komputer.

Setelah semua rangkaian dan sumber tegangan terhubung dengan benar, programlah I2CDCMOTOR.BIN atau I2CDCMOTOR.HEX ke dalam DT-AVR Low Cost Nano System (berbasis AT90S2313) dengan menggunakan DT-HiQ AVR In System Programmer atau divais ISP programmer lain dengan konektor *header* 10-pin standar Atmel.

Hubungan antara modul-modul tersebut secara Paralel adalah sebagai berikut:

DT-AVR Low Cost Nano System / Low Cost Micro System	SPC DC Motor
+5VDC	+5V (J1)
GND	GND (J1)
PB.4*	Run1 (S1-J5)
PB.5*	Dir1 (S2-J5)
PB.6*	Run2 (S3-J5)
PB.7*	Dir2 (S4-J5)

* Pin ini tidak mutlak dan dapat diganti pin lain tetapi harus mengubah program

Tabel 2
Hubungan DT-AVR Low Cost Nano System / Low Cost Micro System dengan SPC DC Motor secara Paralel

Untuk komunikasi secara paralel, hubungan antara DT-AVR Low Cost Nano System / Low Cost Micro System dan SPC DC Motor mengikuti Tabel 2. Setelah semua rangkaian dan sumber tegangan terhubung dengan benar, programlah PARALLELDCMOTOR.BIN atau PARALLELDCMOTOR.HEX ke dalam DT-AVR Low Cost Nano System (berbasis AT90S2313) dengan menggunakan DT-HiQ AVR In System Programmer atau divais ISP programmer lain dengan konektor *header* 10-pin standar Atmel.

Catatan:

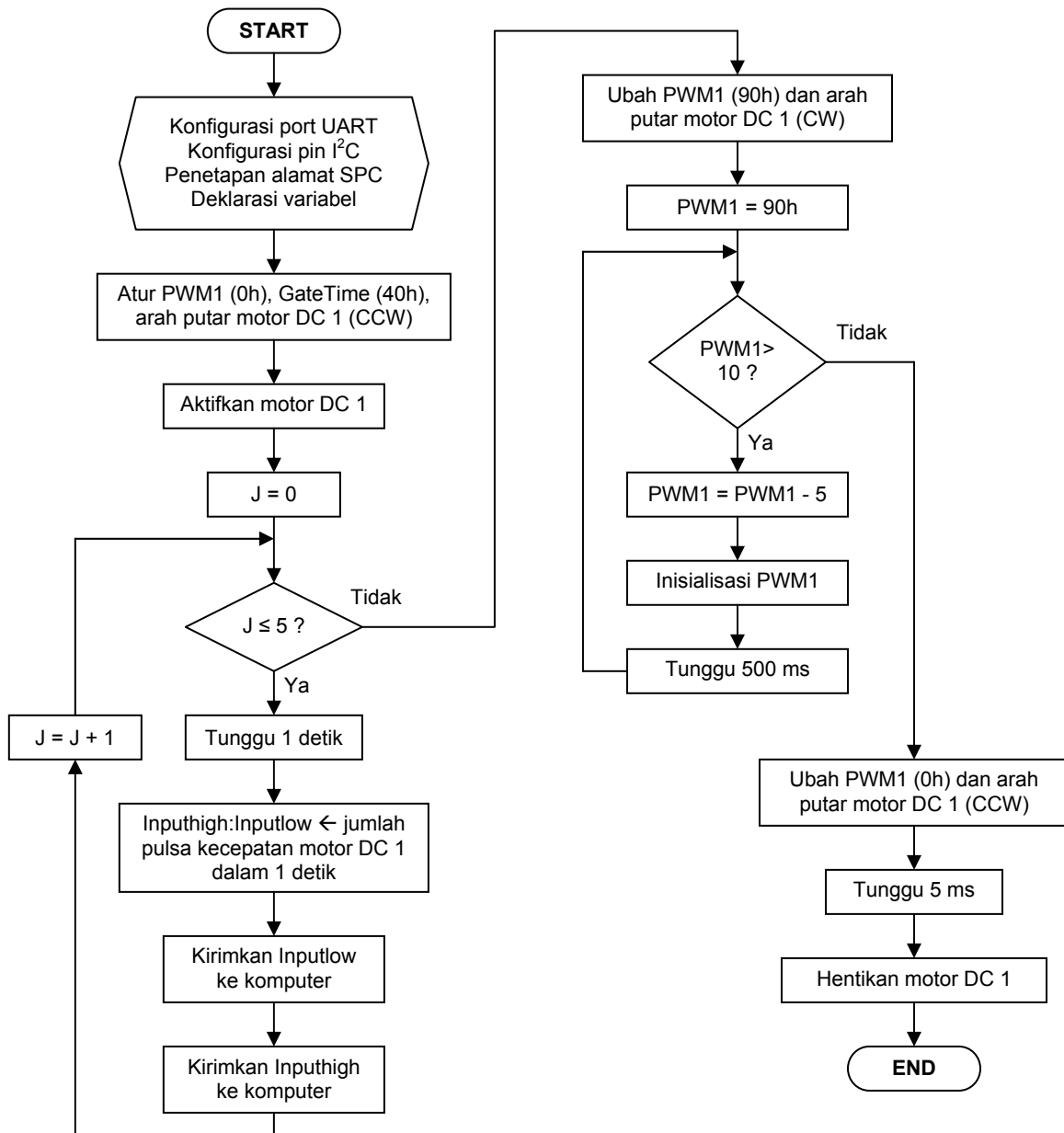
Dalam aplikasi ini menggunakan DT-AVR Low Cost Nano System (dengan mikrokontroler AT90S2313) sehingga I2CDCMOTOR.BIN / I2CDCMOTOR.HEX maupun PARALLELDCMOTOR.BIN / PARALLELDCMOTOR.HEX yang disertakan dalam **AN81.ZIP** hanya berlaku untuk AT90S2313.

Apabila menggunakan mikrokontroler tipe lain, bukalah I2CDCMOTOR.BAS atau PARALLELDCMOTOR.BAS menggunakan BASCOM-AVR[®] dan ubahlah baris pertama dalam program tersebut:

- \$regfile = "2313def.dat" '→ untuk AT90S2313
- \$regfile = "8535def.dat" '→ untuk AT90S8535
- \$regfile = "m8535.dat" '→ untuk ATmega8535 (DT-AVR Low Cost Micro System)

agar sesuai dengan tipe mikrokontroler yang akan digunakan. Lalu *compile* ulang program sehingga menghasilkan *file* dengan ekstensi *.bin* atau *.hex* yang sesuai dan dapat diprogram ke dalam modul mikrokontroler.

Flowchart program untuk komunikasi secara Serial I²C adalah sebagai berikut:



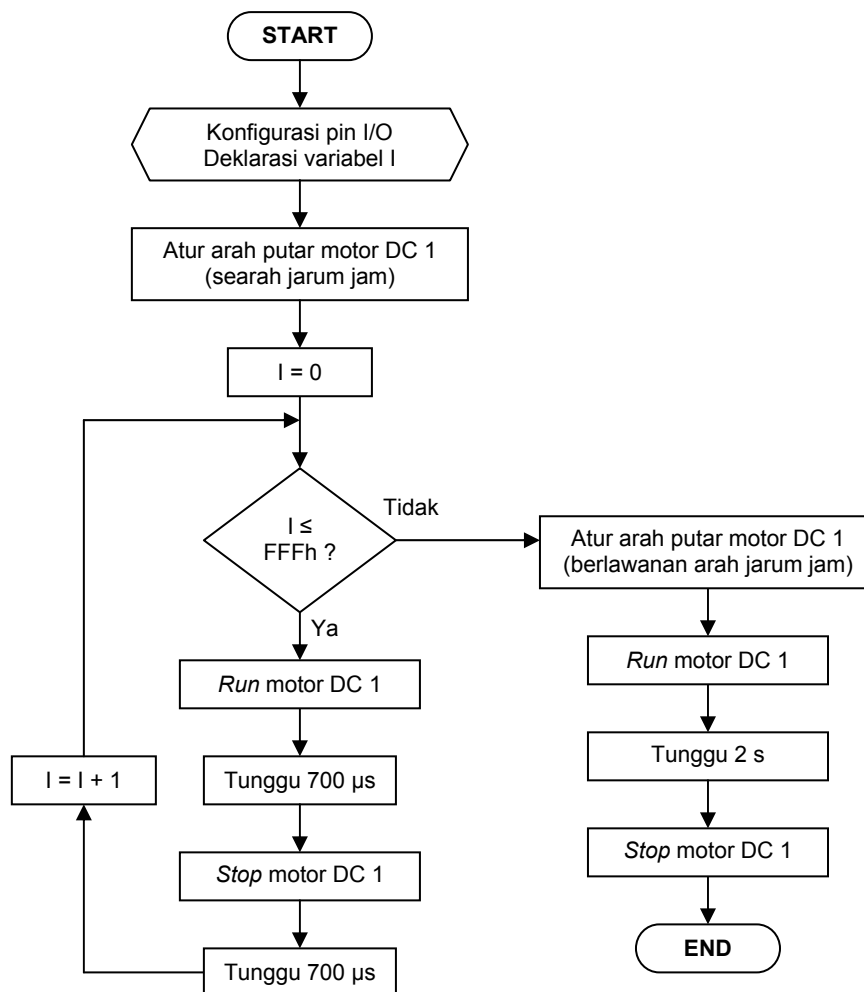
Gambar 3
Flowchart Program untuk Komunikasi secara Serial I²C

Program utama untuk komunikasi secara serial I²C akan diproses sebagai berikut:

1. Program melakukan konfigurasi port UART dengan *baudrate* 9600 bps, 8-bit data, 1 *stop* bit, tanpa bit *parity*. Dan menentukan definisi pin SDA dan SCL sebagai *input / output* untuk komunikasi serial I²C. Serta penetapan alamat SPC DC Motor yaitu EEh (alamat terprogram 111b).
2. Kemudian program melakukan deklarasi variabel *Sub_data*, *Inputlow*, *Inputhigh*, *I*, *J*. Fungsi dari masing-masing variabel tersebut yaitu:
Sub_data adalah variabel yang digunakan untuk data *input / output* pada sub rutin (misalnya: *Init_pwm1*).
Inputhigh adalah variabel untuk 8-bit data MSB dari jumlah pulsa kecepatan motor DC dalam 1 detik.
Inputlow adalah variabel untuk 8-bit data LSB dari jumlah pulsa kecepatan motor DC dalam 1 detik.
I, *J* adalah variabel yang digunakan dalam proses *looping*.

3. Program menjalankan motor DC 1 dengan kecepatan penuh (PWM1=0h), berlawanan arah jarum jam, dan resolusi pengukuran pulsa kecepatan motor DC sebesar 1Hz (GateTime=40h).
4. Program mengambil data jumlah pulsa kecepatan motor DC 1 dalam 1 detik dan dikirimkan ke komputer. Hal ini dilakukan sebanyak 6 kali dengan selang waktu 1 detik. Data tersebut dapat dilihat pada layar komputer dengan bantuan HyperTerminal[®] atau Terminal[®] (baudrate 9600 bps, 8-bit data, 1 stop bit, tanpa bit parity).
5. Program mengubah arah putar motor DC 1 menjadi searah jarum jam dan dengan kecepatan agak pelan (PWM1=90h).
6. Naikkan kecepatan motor DC 1 dari PWM1=90h sampai PWM1=10h secara perlahan (dengan resolusi penurunan PWM1 sebesar 5h dan selang waktu 500 ms tiap perubahannya).
7. Program mengubah lagi arah putar motor DC 1 menjadi berlawanan arah jarum jam dan dengan kecepatan penuh (PWM1=0h).
8. Setelah menunggu 5 ms, program menghentikan motor DC 1.

Flowchart program untuk komunikasi secara Paralel adalah sebagai berikut:



Gambar 4
Flowchart Program untuk Komunikasi secara Paralel

Program untuk komunikasi secara Paralel akan diproses sebagai berikut:

1. Program menentukan definisi pin I/O untuk komunikasi Paralel. Dan mendeklarasikan variabel I yang akan digunakan dalam proses *looping*.
2. Menjalankan motor DC 1 searah jarum jam dengan kecepatan setengah dari kecepatan penuh ($T_H=700 \mu s$, $T_L=700 \mu s$), selama $1,4 \times 1000h$ mili detik.
3. Menjalankan motor DC 1 berlawanan arah jarum jam dengan kecepatan penuh selama 2 detik.
4. Menghentikan motor DC 1.

Listing program terdapat pada **AN81.ZIP**.

Selamat berinovasi!

BASCOM-AVR is copyright by MCS Electronics
I2C is a Registered Trademark of Philips Semiconductors
HyperTerminal is a copyright by Hilgraeve Inc.
Terminal is a copyright by Bray++