

DT-51

DT-51 *Application Note* AN107 – Aplikasi SPC DC Motor dengan Bahasa C MCS-51®

Oleh: Tim IE

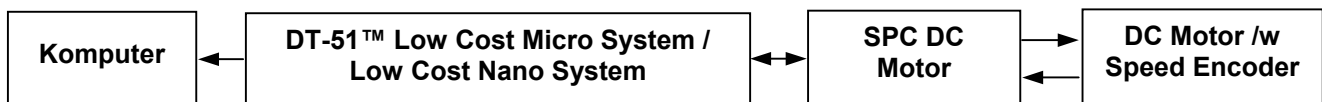
Sudah banyak AN yang membahas aplikasi menggunakan SPC DC Motor, mengapa perlu satu lagi? Sebab AN yang satu ini berbeda, yaitu menggunakan bahasa C (uC/51®, MCS-51® C compiler buatan Wickenhäuser) dan di dalamnya disertakan pula rutin-rutin (*engine*) untuk komunikasi I²C (*file Eng_I2C.h*) yang akan membantu dalam melakukan komunikasi dengan SPC DC Motor secara I²C. Selain komunikasi I²C, aplikasi ini juga memberikan contoh penggunaan SPC DC Motor dalam komunikasi paralel.

Dalam aplikasi ini hanya diberikan contoh dengan menggunakan sebuah motor DC saja sehingga penggunaan SPC DC motor kelihatannya kurang efektif. Tetapi bila aplikasi ini telah dikembangkan untuk mengendalikan beberapa motor DC maka kita dapat melihat bahwa dengan komunikasi I²C kita dapat mengendalikan hingga 16 buah motor DC dengan hanya 2 jalur I/O saja. Selain itu beban mikrokontroler utama (DT-51™ Low Cost Micro System / Low Cost Nano System) akan lebih ringan karena sebagian tugas untuk mengendalikan motor DC telah ditangani oleh SPC DC Motor.

Komponen yang diperlukan:

- 1 DT-51™ Low Cost Micro System / Low Cost Nano System,
- 1 SPC DC Motor,
- 1 Motor DC yang dilengkapi dengan Speed Encoder.

Adapun blok diagram sistem secara keseluruhan adalah sebagai berikut:



Gambar 1
Blok Diagram AN107

Hubungan antara modul-modul tersebut adalah sebagai berikut:

DT-51™ Low Cost Micro System / Low Cost Nano System	SPC DC Motor
GND (J10)	GND (J1)
VOUT (J10)	+5V (J1)
P1.0* (J4)	S1 (J5)
P1.1* (J4)	S2 (J5)

* pin ini dapat diganti dengan pin I/O lainnya tetapi tentunya juga harus mengubah alokasi pin pada program

Tabel 1

Hubungan DT-51™ Low Cost Micro System / Low Cost Nano System dengan SPC DC Motor secara Paralel

DT-51™ Low Cost Micro System / Low Cost Nano System	SPC DC Motor
GND (J10)	GND (J1)
VOUT (J10)	+5V (J1)
P3.2* (J6)	SDA (J2)
P3.3* (J6)	SCL (J2)

* pin ini dapat diganti dengan pin I/O lainnya tetapi tentunya juga harus mengubah alokasi pin pada program

Tabel 2

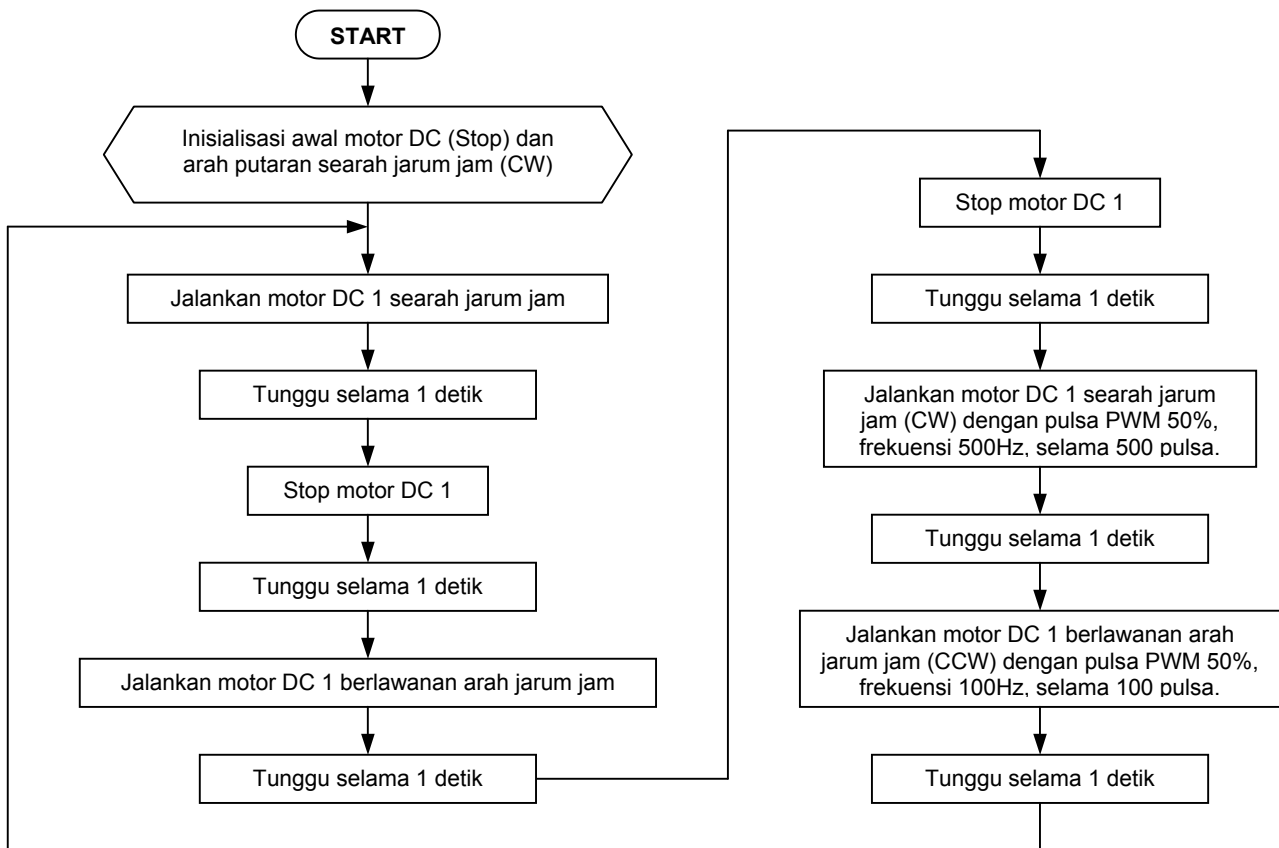
Hubungan DT-51™ Low Cost Micro System / Low Cost Nano System dengan SPC DC Motor secara I²C

Pada modul SPC DC Motor hubungkan terminal VMotor (J6) dengan sumber tegangan yang sesuai dengan tegangan kerja motor DC yang digunakan, misalnya motor DC yang digunakan 12 volt maka hubungkan terminal VMotor (J6) dengan sumber tegangan 12 volt. Kemudian hubungkan kutub positif motor DC pada terminal M1+ dan kutub negatif motor DC pada terminal M1-. Hubungkan keluaran *Speed Encoder* ke terminal IN1 (J6) SPC DC Motor. Lepaslah *jumper* J4 (*address*, A0-A2) SPC DC Motor sehingga alamat I²C untuk modul SPC DC Motor menjadi EEH. Sedangkan semua jumper J3 (SDA dan SCL) SPC DC Motor dipasang.

Gunakan kabel serial DT-51™ Low Cost Series untuk menghubungkan modul DT-51™ dengan komputer. Pada contoh program dengan komunikasi I²C, kabel ini digunakan komputer untuk menerima data yang dikirimkan oleh DT-51™ Low Cost Micro System / Low Cost Nano System. Pada komputer data ini ditampilkan dengan bantuan program Terminal[®], dengan konfigurasi UART pada *baud rate* 9600 bps, 8 bit data, tanpa bit *parity*, 1 bit stop dan tanpa *flow control*.

Setelah semua rangkaian dan sumber tegangan terhubung dengan tepat, programlah PROGRAM.HEX pada direktori "Program_PARALEL" (untuk komunikasi secara paralel) atau pada direktori "Program_I2C" (untuk komunikasi secara I²C) ke DT-51™ Low Cost Micro System dengan bantuan **DT-HiQ AT89S In System Programmer** atau divais *in-system programmer* berkonektor standar Atmel lainnya yang mendukung AT89S51. Jika menggunakan DT-51™ Low Cost Nano System maka gunakan **DT-HiQ Programmer** atau divais *parallel programmer* lainnya untuk melakukan pemrograman AT89C2051.

Flowchart program untuk komunikasi secara **paralel** adalah sebagai berikut:

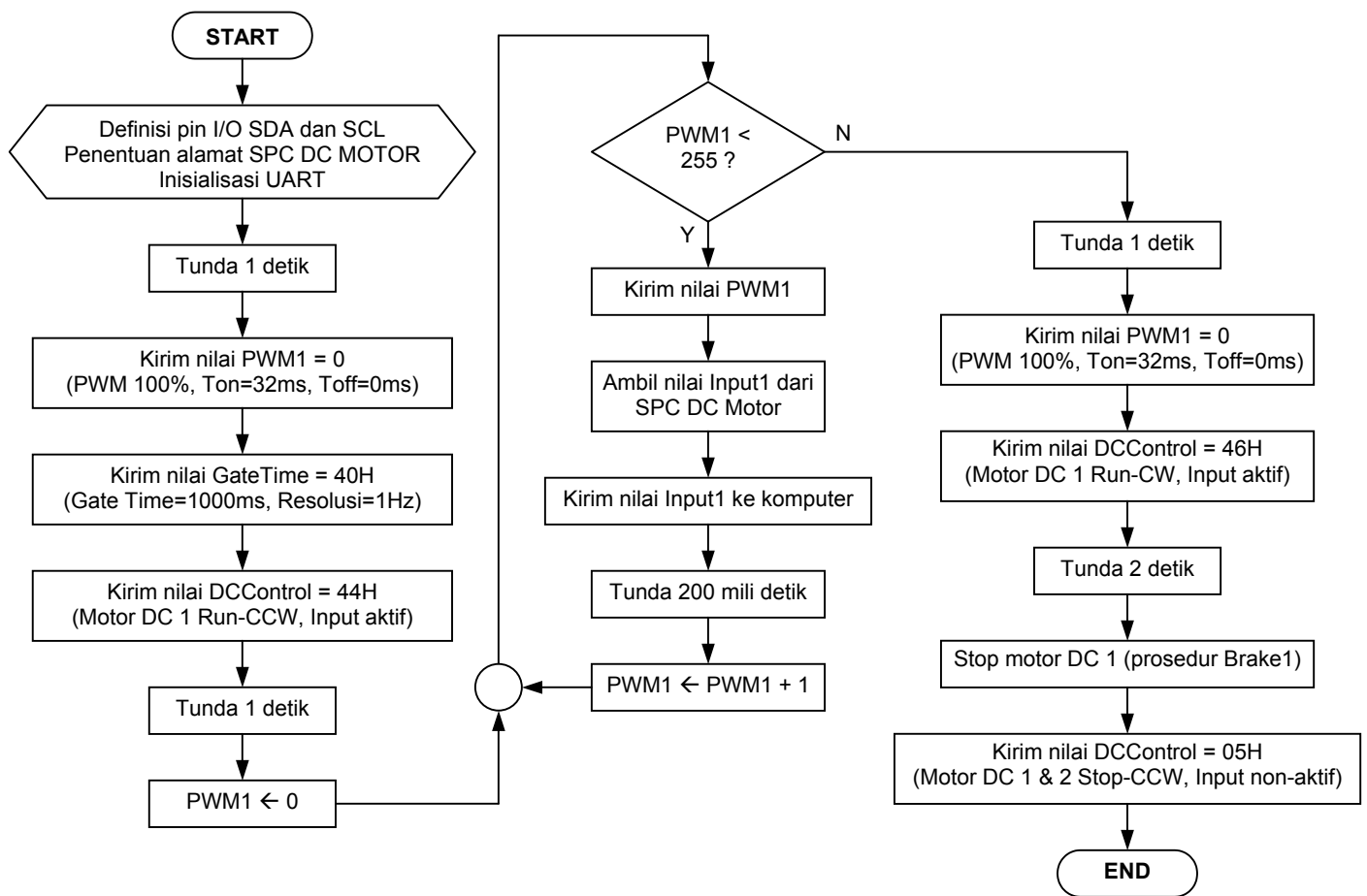


Gambar 2
Flowchart Program untuk Hubungan antar Modul secara Paralel

Program utama untuk komunikasi secara **paralel** akan diproses sebagai berikut:

1. Proses yang pertama kali dilakukan adalah alokasi pin I/O yaitu P1.0 sebagai S1, P1.1 sebagai S2, P1.2 sebagai S3, dan P1.3 sebagai S4. Pada awal program semua pin ini (S1-S4) diberi logika '1' (*high*) sehingga motor DC 1 dan motor DC 2 dalam keadaan Stop (berhenti) dan CW (arah putaran searah jarum jam).
2. Berikutnya program akan menjalankan motor DC 1 searah jarum jam dengan memberikan logika '1' pada pin S1. Motor DC 1 berputar selama 1 detik, kemudian program akan menghentikan motor DC 1 selama 1 detik. Catatan: arah putaran motor yang sebenarnya adalah bergantung pada motor DC yang digunakan.
3. Setelah itu program menjalankan motor DC 1 berlawanan arah jarum jam dengan memberikan logika '0' pada pin S1. Motor DC 1 berputar selama 1 detik, kemudian program akan menghentikan motor DC 1 selama 1 detik.
4. Kemudian program akan membalik arah putaran motor DC 1 (CW) dengan memberi logika '1' pada pin S2. Setelah itu program akan memberikan pulsa PWM 50% dengan frekuensi 500 Hz sebanyak 500 pulsa pada pin S1. Sehingga motor DC 1 akan berjalan lebih lambat daripada sebelumnya karena kondisi logika pada pin S1 berbentuk pulsa (motor berputar selama 1 ms dan berhenti selama 1 ms). Setelah pemberian pulsa sebanyak 500 kali selesai maka program akan berhenti selama 1 detik.
5. Proses berikutnya sama dengan langkah pada nomor 4 tetapi motor DC 1 berputar berlawanan arah jarum jam dengan pulsa PWM 50% pada frekuensi 100 Hz sebanyak 100 pulsa.
6. Program kembali ke langkah 2.

Flowchart program untuk komunikasi secara I²C adalah sebagai berikut:



Gambar 3
Flowchart Program untuk Hubungan antar Modul secara I²C

Program utama untuk komunikasi secara I²C akan diproses sebagai berikut:

1. Proses yang pertama kali dilakukan adalah menentukan definisi pin I/O SDA dan SCL untuk komunikasi serial I²C serta membuat konstanta untuk alamat SPC DC Motor yaitu pada alamat EEH. Kemudian inisialisasi komunikasi serial UART yaitu pada *baud rate* 9600 bps, 8 bit data, tanpa bit *parity*, 1 bit *stop*, dan tanpa *flow control*. Kemudian program menunggu selama 1 detik.
2. Proses selanjutnya adalah mengirimkan nilai PWM1 dengan nilai awal 0 ke SPC DC Motor. Ini berarti SPC DC Motor akan menjalankan motor DC 1 dengan PWM 100%, Ton = 32ms dan Toff = 0ms.
3. Kemudian program mengirimkan nilai GateTime sebesar 40H. Ini berarti SPC DC Motor akan melakukan penghitungan kecepatan putaran motor tiap jeda waktu 1 detik dengan resolusi 1 Hz.
4. Setelah itu program mengirimkan nilai DCControl sebesar 44H. Artinya SPC DC Motor akan menjalankan motor DC 1 dengan putaran berlawanan arah jarum jam dan menghitung kecepatan putaran motor DC 1. Kemudian program akan berhenti selama 1 detik. Catatan: arah putaran motor yang sebenarnya adalah bergantung pada motor DC yang digunakan.
5. Pada proses berikutnya program memberi variabel PWM1 nilai nol.
6. Program mengirimkan nilai PWM1 ke SPC DC Motor dan membaca kecepatan putaran motor DC 1. Hasil pembacaan kecepatan putaran motor DC 1 ini dikirimkan ke komputer dalam bentuk desimal (dapat dibaca melalui program Terminal[®]). Setelah itu program berhenti selama 200 mili detik.
7. Kemudian program menambah variabel PWM1 dengan nilai 1. Langkah 6 dan 7 ini dilakukan terus-menerus hingga nilai variabel PWM1 = 255 sehingga motor DC 1 akan berhenti berputar.
8. Jika nilai variabel PWM1 = 255 maka proses berikutnya adalah program menunda selama 1 detik agar motor DC 1 terlihat tidak berputar. Setelah itu program mengirimkan variabel PWM1 dengan nilai 0 ke SPC DC MOTOR (PWM 100%), setelah itu dilakukan pengiriman nilai DCControl 46H. Ini berarti SPC DC Motor akan menjalankan motor DC 1 dengan PWM 100% searah jarum jam dan membaca masukan dari *speed encoder*. Setelah itu delay selama 2 detik agar motor berputar dengan kecepatan penuh.

9. Proses berikutnya yaitu melakukan pengereman (*brake*) pada motor DC 1 dengan menggunakan prosedur Brake1.
10. Proses yang terakhir yaitu menghentikan semua motor DC dengan mengirimkan nilai DCControl sebesar 05H.

Untuk melakukan kompilasi program komunikasi secara I²C dengan menggunakan *compiler uC/51*[®], lakukan langkah berikut:

1. Buatlah *workspace* baru dengan cara:
 - a. Jalankan program “**MakeWiz**”.
 - b. Pada tab “**General**” tekan tombol “**New *.MAK file...**” untuk membuat MAKE *file* baru.
 - c. Pada tab “**Components**” tekan tombol “**Add Source File ...**” untuk menambahkan file sumber yang akan dikompilasi yaitu *file* “**program.c**” (program utama) dan “**ENG_I2C.s51**” (rutin I²C).
 - d. Pada *combo box* “**Target File (*.BIN)**” pilih *file* .BIN yang akan dihasilkan, yaitu “**program.bin**”.
 - e. Pada tab “**C-Compiler**” isilah *CPU-Speed* sesuai dengan *hardware* yang digunakan.
 - f. Pada tab “**Linker**” tentukan *ROM Start* dan *External RAM Start* bila ada.
 - g. Pada tab “**Misc**” pilih *Write JFE-Workspace File* dan pilih *Gen. HEX-File*.
 - h. Tekan tombol “**Save Changes**” lalu tekan tombol “**Cancel/Exit**”.
2. Lakukan kompilasi *workspace* dengan cara:
 - a. Jalankan program “**JFE**”.
 - b. Pilihlah menu **File** → **Open Workspace** untuk membuka *workspace* yang baru saja dibuat tadi.
 - c. Lalu tekan tombol “**Make**” untuk melakukan kompilasi program. Setelah itu pada direktori di mana *file* program.c berada akan muncul *file* baru bernama **program.HEX**.

Listing program komunikasi I²C terdapat pada direktori “Program_I2C” dalam **AN107.ZIP**, sedangkan *listing* program komunikasi **paralel** terdapat pada direktori “Program_PARALEL” dalam **AN107.ZIP**.

Selamat berinovasi!

μC/51 is copyright by Wickenhäuser Elektrotechnik.
I²C is a registered trademark of Philips Semiconductors.
MCS-51 is a registered trademark of Intel Corp.
DT-51 is a trademark of Innovative Electronics.