

Pemrograman dengan bahasa tingkat tinggi (*high level language*) memberikan keleluasaan dan kemudahan dalam mengimplementasikan algoritma program dari yang sederhana hingga sangat rumit. Aplikasi berikut akan memberikan contoh penggunaan modul DT-51™ Low Cost Micro System yang akan diprogram dengan bahasa pemrograman C ( $\mu\text{C}/51^{\circ}$ , Wickenhäuser) untuk mengendalikan SPC Stepper Motor. Hubungan antara kedua modul tersebut adalah secara paralel dan secara I<sup>2</sup>C (terdapat 2 buah program), sesuai fitur antarmuka yang dimiliki oleh modul SPC Stepper Motor. Alur singkat dari program AN ini adalah mikrokontroler memberikan perintah ke SPC Stepper Motor, lalu SPC Stepper Motor mengaktualkan perintah tersebut pada motor stepper. Komponen yang diperlukan dalam aplikasi ini adalah:

- 1 buah DT-51™ Low Cost Micro System (dapat diganti dengan DT-51™ Low Cost Nano System)
- 1 buah SPC Stepper Motor dan motor stepper tipe bipolar

Adapun blok diagram sistem secara keseluruhan adalah sebagai berikut:



Gambar 1  
Blok Diagram AN138

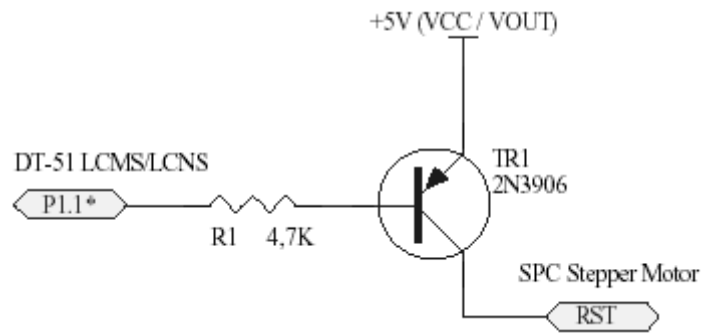
Hubungan antara modul-modul tersebut adalah sebagai berikut:

DT-51™ Low Cost Micro System (J4) / Low Cost Nano System (J6)	SPC Stepper Motor
GND (Pin 1)	GND (POWER)
VCC (Pin 2)	+5V (POWER)
P1.0* (Pin 3)	S1 (J4 - Pin 2)
P1.2* (Pin 5)	S2 (J4 - Pin 3)
P1.4* (Pin 7)	S3 (J4 - Pin 4)
P1.6* (Pin 9)	S4 (J4 - Pin 5)

Tabel 1  
Hubungan DT-51™ Low Cost Micro System / Low Cost Nano System dengan SPC Stepper Motor Secara Paralel

DT-51™ Low Cost Micro System (J6) / Low Cost Nano System (J7)	SPC Stepper Motor
GND (Pin 1)	GND (POWER)
VCC (Pin 2)	+5V (POWER)
P3.2* (Pin 5)	SDA (J1)
P3.3* (Pin 6)	SCL (J1)

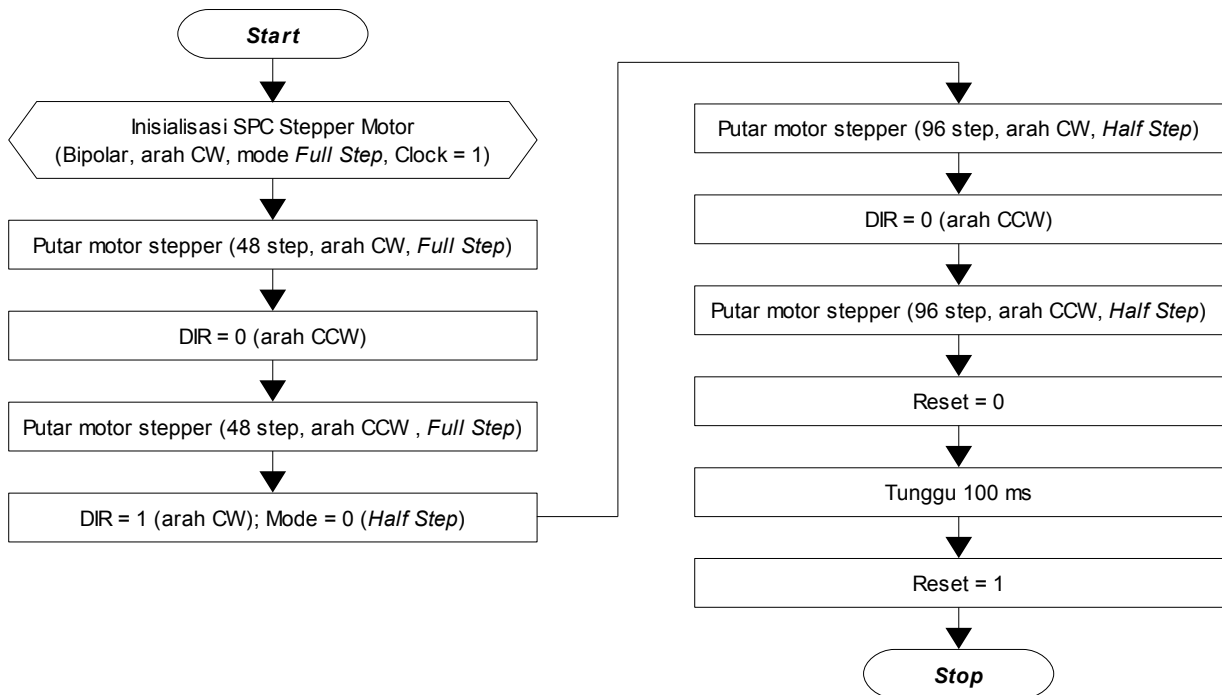
Tabel 2  
Hubungan DT-51™ Low Cost Micro System / Low Cost Nano System dengan SPC Stepper Motor secara I<sup>2</sup>C



**Gambar 2**  
**Rangkaian “Reset” Dari DT-51™ Low Cost Micro System / Low Cost Nano System ke SPC Stepper Motor Untuk Antarmuka Paralel**

Pin yang bertanda \* pada Tabel 1 dan Tabel 2 tidak mutlak dan bisa diganti dengan pin yang lain dengan cara mengubah program. Pada SPC Stepper Motor hubungkan VMotor (J5) ke terminal biru +5V (POWER). Pasang semua *jumper* pada Address A0-A2 (J3) serta semua *jumper* resistor *pull up* untuk SDA dan SCL (J2). Alamat I<sup>2</sup>C modul SPC Stepper Motor yang digunakan dalam program aplikasi ini adalah E0H. Motor stepper yang digunakan mempunyai spesifikasi tegangan 5 Volt, bertipe *bipolar*, dan memiliki resolusi putar 7,5° /step. Jika menggunakan DT-51™ Low Cost Nano System maka pasanglah *jumper* J8 DT-51™ Low Cost Nano System untuk memberi *pull up* pada jalur P1.0. Setelah semua rangkaian dan sumber tegangan terhubung dengan benar sesuai tabel di atas, programlah PROGRAM.HEX dari direktori “Program\_PARALEL” (untuk komunikasi secara paralel) atau dari direktori “Program\_I2C” (untuk komunikasi secara I<sup>2</sup>C) ke DT-51™ Low Cost Micro System / Low Cost Nano System dengan DT-HiQ AT89S In System Programmer atau *programmer* lain yang sesuai.

**F**lowchart program untuk komunikasi secara paralel adalah sebagai berikut:



**Gambar 3**  
**Flowchart Program Antarmuka Paralel**

**P**rogram utama (komunikasi secara PARALEL) akan diproses sebagai berikut:

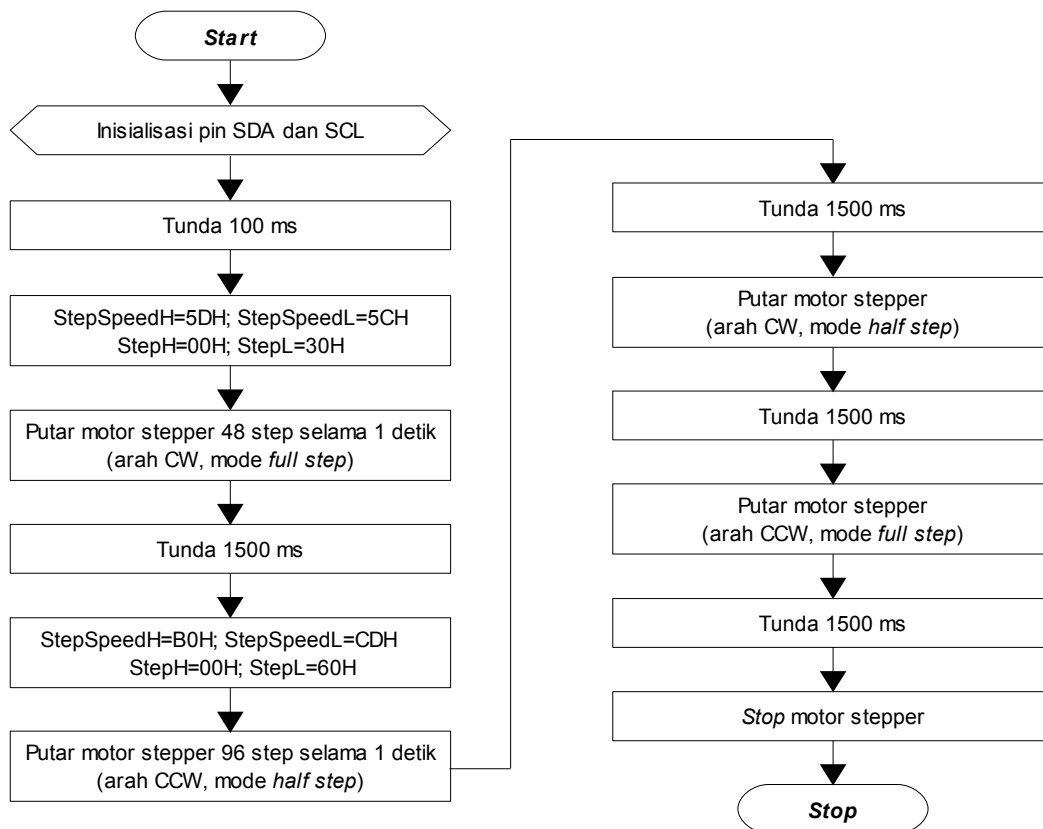
1. Proses yang pertama kali dilakukan adalah inisialisasi untuk memilih tipe motor stepper yang digunakan (*Bipolar / Unipolar*, diatur pada pin S1), arah putaran motor (CW / CCW, diatur pada pin S2), mode putaran motor (*Full step / Half step*, diatur pada pin S3) dan kondisi logika pada pin “clock” (pada pin S4). Semua

pin S1 - S4 (J5) diberi logika '1' (*high*) sehingga dipilih motor stepper tipe bipolar, arah putaran searah jarum jam (CW), mode putaran secara *full step*, dan logika pada pin "clock" adalah '1'.

2. Setelah proses inialisasi selesai, berikutnya program akan menggerakkan motor stepper agar berputar searah jarum jam (CW) secara *full step* dengan memberikan *clock* sebanyak 48 kali (satu putaran). Setiap pemberian sinyal *clock* ke SPC Stepper Motor, diberikan *delay* selama 30 mili detik.
3. Setelah program memutar motor stepper searah jarum jam secara *full step* selesai, program akan membalik arah putaran motor stepper (berputar berlawanan arah jarum jam, CCW) dengan memberi logika '0' pada pin S2. Kemudian program akan memberikan sinyal *clock* sebanyak 48 kali (satu putaran) untuk memutar motor stepper berlawanan arah jarum jam.
4. Proses berikutnya hampir sama dengan langkah pada nomor 2 (berputar searah jarum jam), tetapi mode putaran motor diubah menjadi *half step* dengan jumlah *clock* sebanyak 96 kali (satu putaran). Mode putaran diatur secara *half step* dengan cara memberi logika '0' pada pin S3.
5. Setelah program pada langkah nomor 4 selesai, kembali program membalik arah putaran motor stepper (berputar berlawanan arah jarum jam) dan memberikan sinyal *clock* sebanyak 96 kali (satu putaran).
6. Setelah selesai memutar motor stepper, program akan memberikan pulsa "RESET" (melalui pin RST) pada SPC Stepper Motor untuk melepas keadaan "lock" dari motor stepper.

Program di atas akan menjalankan motor stepper agar berputar searah jarum jam dan berlawanan arah jarum jam secara *full step* dan *half step*. Jika program tersebut selesai dijalankan maka posisi poros motor stepper akan kembali pada posisi semula.

**F**lowchart program untuk komunikasi secara I<sup>2</sup>C adalah sebagai berikut:



**Gambar 4**  
**Flowchart Program Antarmuka I<sup>2</sup>C**

**P**rogram utama (komunikasi secara I<sup>2</sup>C) akan diproses sebagai berikut:

1. Proses yang pertama kali dilakukan adalah menentukan definisi pin I/O SDA dan SCL untuk komunikasi serial I<sup>2</sup>C serta membuat konstanta untuk alamat SPC Stepper Motor yaitu E0H. Setelah itu program akan menunggu 100 mili detik untuk memberi waktu *start-up* pada SPC Stepper Motor.

2. Berikutnya program akan menggerakkan motor stepper agar berputar searah jarum jam (CW) secara *full step*. Motor stepper akan bergerak 48 step (satu putaran) dalam waktu 1 detik. Agar SPC Stepper Motor dapat menggerakkan motor stepper 48 step dalam waktu 1 detik maka diperlukan pengaturan nilai StepSpeedH, StepSpeedL, StepH, dan StepL. Nilai-nilai tersebut dikirimkan ke modul SPC Stepper Motor terlebih dahulu dan setelah itu nilai StepControl sebesar 0FH (*bipolar, CW, full step*) dikirimkan agar motor stepper berputar. Proses pengiriman semua nilai-nilai tersebut dilakukan melalui prosedur StepInit. Satu putaran motor stepper dengan spesifikasi 7,5° /step dibutuhkan 48 step (360° ÷ 7,5°) pada mode *full step*. Nilai 48 dalam format desimal ini adalah setara dengan 30 heksadesimal. Satu putaran (48 step) dapat ditempuh dalam waktu 1 detik jika motor digerakkan dengan kecepatan 0,02083 detik/step (1 detik ÷ 48 step) atau setara 20,83 ms/step. Dengan demikian nilai StepSpeed dapat ditentukan sebagai berikut:  

$$\text{StepSpeed} = -2048 * (20,83 - 32,5) = 23900 \text{ (dalam desimal)}$$
 Nilai StepSpeed 23900 dalam format desimal ini setara dengan 5D5C heksadesimal. Jadi nilai StepSpeedH = 5DH, StepSpeedL = 5CH, StepH = 00H, StepL = 30H, dan StepControl = 0FH.
3. Setelah menggerakkan motor stepper agar berputar searah jarum jam dengan mode *full step*, program akan menunggu (*delay*) 1,5 detik sehingga motor stepper terlihat berhenti berputar sesaat ( $\pm 1/2$  detik).
4. Setelah eksekusi *delay* selesai, program akan menggerakkan motor stepper berlawanan arah jarum jam (CCW) dengan mode *half step*. Motor stepper akan digerakkan 96 step (satu putaran) dalam waktu 1 detik. Agar SPC Stepper Motor dapat menggerakkan motor stepper 96 step dalam waktu 1 detik maka diperlukan pengaturan nilai StepSpeedH, StepSpeedL, StepH, StepL, dan StepControl seperti pada langkah nomor 2.  
 Satu putaran motor stepper pada mode *half step* membutuhkan 2x jumlah step pada mode full step, dalam hal ini 96 step (60H). Satu putaran dapat ditempuh dalam waktu 1 detik jika motor digerakkan dengan kecepatan 10,42 ms/step (1 detik ÷ 96 step) dan nilai StepSpeed dapat ditentukan sebagai berikut:  

$$\text{StepSpeed} = -2048 * (10,42 - 32,5) = 45260,8 \text{ (dalam desimal)}$$
 Nilai StepSpeed 45261 desimal setara dengan B0CDH. Jadi nilai StepSpeedH = B0H, StepSpeedL = CDH, StepH = 00H, StepL = 60H dan StepControl = 0CH (*bipolar, CCW, half step*).
5. Program akan menunggu selama 1,5 detik sehingga motor stepper akan terlihat berhenti berputar sesaat.
6. Kemudian program membalik arah putaran motor stepper menjadi searah jarum jam dengan mengirimkan nilai StepControl sebesar 0EH (*bipolar, CW, half step*).
7. Program akan menunggu selama 1,5 detik, tetapi motor akan terus berputar karena nilai StepH dan StepL diisi dengan nilai FFH.
8. Program membalik lagi arah putaran motor stepper menjadi berlawanan arah jarum jam dan mengubah mode putaran menjadi *full step* dengan mengirimkan nilai StepControl sebesar 0DH (*bipolar, CCW, full step*).
9. Program akan menunggu selama 1,5 detik dan motor akan terus berputar (StepH dan StepL berisi FFH).
10. Proses yang terakhir adalah menghentikan putaran motor stepper dengan menggunakan prosedur StepStop yang akan mengirimkan nilai-nilai StepSpeedH = 00H, StepSpeedL = 00H, StepH = 00H, StepL = 00H, dan StepControl = 0FH ke SPC Stepper Motor.

**L**isting program terdapat pada **AN138.ZIP**.

**S**elamat berinovasi!

µC/51 is a copyright by Wickenhäuser Elektrotechnik.  
DT-51 is a trademarks of Innovative Electronics.