

DT-BASIC DT-BASIC *Application Note*

AN134 – Pengendali Motor Stepper

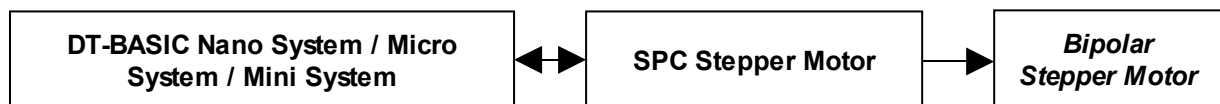
Oleh: Tim IE

AN berikut adalah contoh sederhana penggunaan DT-BASIC *series* untuk mengendalikan motor *stepper*. Pengendalian motor *stepper* akan dibantu oleh modul SPC Stepper Motor sehingga aplikasi ini menjadi sederhana dan mudah dibuat. Aplikasi ini dirancang menggunakan modul DT-BASIC Nano System dan pemrogramannya yang berbahasa PBASIC™ dibuat dengan bantuan *compiler* BASIC Stamp® Editor®. DT-BASIC Nano System dapat dihubungkan ke SPC Stepper Motor secara paralel maupun I²C. Kedua jenis antarmuka ini didukung oleh modul SPC Stepper Motor.

Modul dan komponen yang digunakan dalam AN ini adalah sebagai berikut:

- 1 unit DT-BASIC Nano System / Micro System / Mini System,
- 1 unit SPC Stepper Motor,
- 1 buah motor *stepper* tipe *bipolar*.

Adapun blok diagram sistem secara keseluruhan adalah sebagai berikut:



Gambar 1
Blok Diagram AN134

Hubungan antara modul-modul tersebut adalah sebagai berikut:

DT-BASIC Nano System (J2) / Micro System (J3) / Mini System (J1)	SPC Stepper Motor
GND (pin 1)	GND (Power)
5VDC (pin 19)	+5VDC (Power)
P6* (pin 9)	S1 (J4)
P4* (pin 7)	S2 (J4)
P2* (pin 5)	S3 (J4)
P0* (pin 3)	S4 (J4)
P1* (pin 4)	RST (J4)

* pin ini tidak mutlak dan dapat diganti pin lain dengan mengubah program

Tabel 1

Hubungan DT-BASIC Nano System / Micro System / Mini System dengan SPC Stepper Motor secara Paralel

DT-BASIC Nano System (J2) / Micro System (J3) / Mini System (J1)	SPC Stepper Motor
GND (pin 1)	GND (Power)
5VDC (pin 19)	+5VDC (Power)
P8* (pin 11)	SDA (J1)
P9* (pin 12)	SCL (J1)

* pin ini tidak mutlak dan dapat diganti pin lain dengan mengubah program

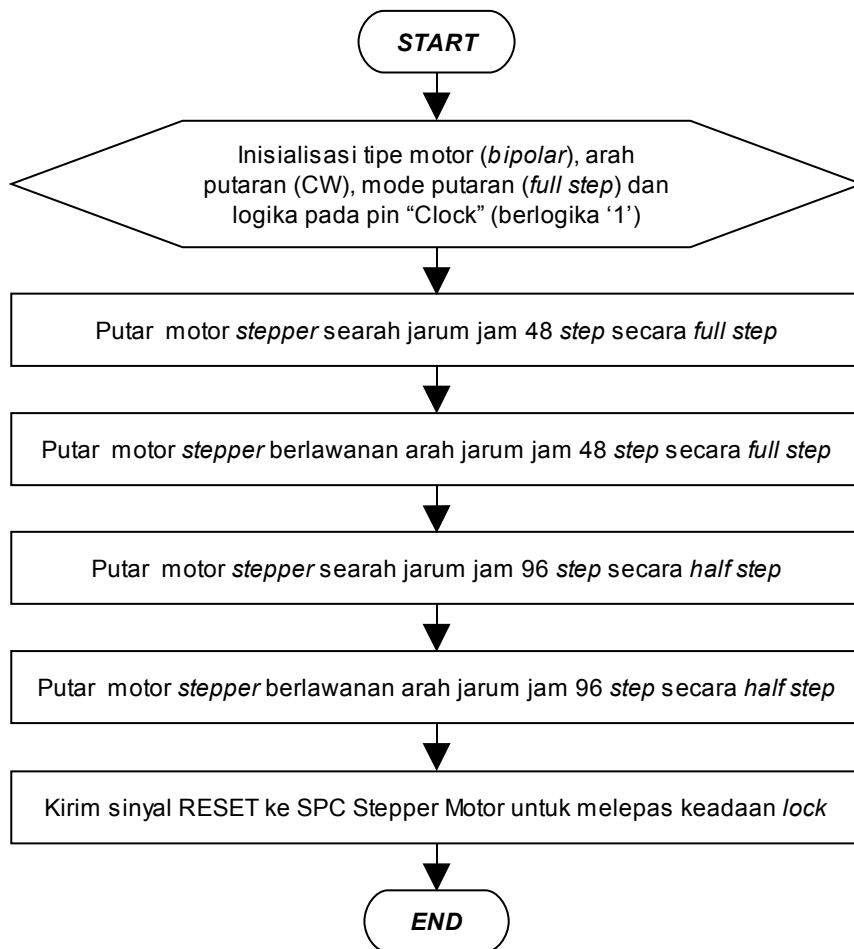
Tabel 2

Hubungan DT-BASIC Nano System / Micro System / Mini System dengan SPC Stepper Motor secara I²C

Pada modul SPC Stepper Motor hubungkan VMotor (J5) dengan +5VDC (Power). Pasang semua *jumper* pada pin *address* A0-A2 (J3) serta semua *jumper pull up* SDA dan SCL (J2). Alamat I²C untuk modul SPC Stepper Motor pada contoh aplikasi ini adalah E0H. Motor *stepper* yang digunakan dalam AN ini mempunyai spesifikasi: tegangan kerja 5 Volt, tipe *bipolar*, dan bergerak 7,5° / step (*full step*). Hubungkan motor *stepper* ke SPC Stepper Motor melalui terminal J5. Gunakan kabel serial DT-BASIC *series* untuk menghubungkan modul DT-BASIC Nano System / Micro System / Mini System ke komputer. Kabel ini sekaligus digunakan untuk melakukan *download* program PBASIC™ yang telah dibuat ke modul DT-BASIC Nano System / Micro System / Mini System melalui program BASIC Stamp® Editor®.

Setelah semua rangkaian dan sumber tegangan terhubung dengan benar, programlah STEP_PARALEL.BS2 (untuk komunikasi secara paralel) atau STEP_I2C.BS2 (untuk komunikasi secara I²C) ke DT-BASIC Nano System dengan menekan tombol F9 atau CTRL+R pada program BASIC Stamp® Editor®. Program STEP_PARALEL.BS2 atau STEP_I2C.BS2 dapat juga bekerja pada modul DT-BASIC Micro System / Mini System, tetapi *STAMP Directive* (lihat *listing* program pada baris pertama) perlu disesuaikan dengan modul DT-BASIC yang digunakan yaitu '{ \$STAMP BS2sx }' untuk DT-BASIC Micro System atau '{ \$STAMP BS2p }' untuk DT-BASIC Mini System.

Flowchart dari program STEP_PARALEL.BS2 adalah sebagai berikut:



Gambar 2
Flowchart Program Untuk Hubungan Antar Modul Secara Paralel

Program utama (STEP_PARALEL.BS2) 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 awal pin "Clock" (S4). Semua pin ini (pin S1-S4) diberi logika '1' (*high*), yang artinya menggunakan motor *stepper* tipe *bipolar*, berputar searah jarum jam (CW), mode putaran *full step* dan logika pada pin "Clock" adalah '1'.
2. Setelah proses inisialisasi selesai, berikutnya program akan menggerakkan motor *stepper* searah jarum jam (CW) secara *full step* dengan memberikan *clock* sebanyak 48 kali (satu putaran) pada pin S4. Setiap

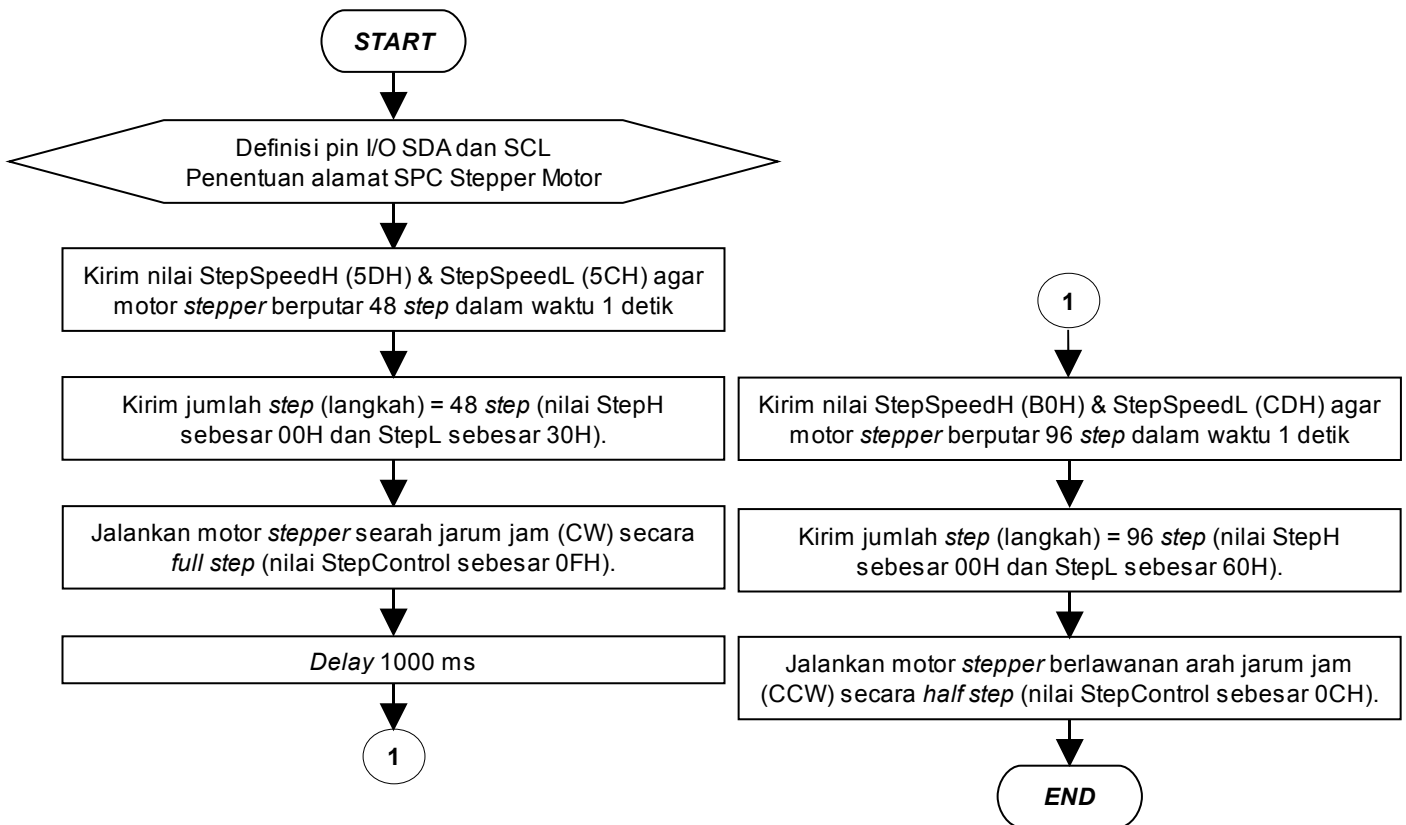
pemberian sinyal *clock* ke SPC Stepper Motor, diberikan *delay* 30 mili detik untuk menghasilkan pulsa kotak pada pin “clock” agar motor *stepper* dapat berputar dengan baik.

3. Setelah program selesai memutar motor *stepper* searah jarum jam secara *full step*, maka program akan membalik arah putaran motor *stepper* (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 adalah *half step* dengan jumlah *clock* sebanyak 96 kali (satu putaran). Pemilihan mode putaran motor *stepper* secara *half step* dengan memberi logika ‘0’ pada pin S3.
5. Setelah program pada langkah nomor 4 selesai, program akan 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 (pin RST) untuk melepas keadaan “lock” pada SPC Stepper Motor.

Program di atas akan menggerakkan motor *stepper* 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 seperti semula.

Listing program STEP_PARALEL.BS2 terdapat pada AN134.ZIP.

Flowchart dari program STEP_I2C.BS2 adalah sebagai berikut:



Gambar 3
Flowchart Program Untuk Hubungan Antar Modul Secara I²C

Program utama (STEP_I2C.BS2) 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 Stepper Motor yaitu pada alamat E0H.
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 menggerakkan motor *stepper* sebanyak 48 *step* dan dalam waktu 1 detik, maka diperlukan pengaturan nilai StepSpeedH, StepSpeedL, StepH, dan StepL. Nilai-nilai tersebut dikirimkan ke modul SPC Stepper Motor terlebih dahulu sebelum nilai Control sebesar 0FH (*bipolar, CW, full step*). Kalkulasi nilai StepSpeedH, StepSpeedL, StepH, dan StepL untuk motor *stepper* berspesifikasi 7,5°/step adalah:
 - Satu putaran dalam mode *full step* dibutuhkan $(360^\circ \div 7,5^\circ) \text{ step} = 48 \text{ step}$.
 - Nilai 48 *step* adalah dalam desimal yang setara dengan nilai 30 heksadesimal. Jadi nilai StepH = 00H dan StepL = 30H.
 - Waktu yang dibutuhkan per-*step*-nya adalah $1 \text{ detik} \div 48 \text{ step} = 0,02083 \text{ detik}$ atau setara dengan 20,83 mili detik.
 - Sehingga nilai StepSpeed adalah: $-2048 \times (20,83 - 32,5) = 23900$ (dalam desimal).
 - Nilai StepSpeed yang didapat adalah 23900 dalam desimal yang setara dengan nilai 5D5C heksadesimal. Jadi nilai StepSpeedH = 5DH dan StepSpeedL = 5CH.
3. Setelah program menggerakkan motor *stepper* searah jarum jam secara *full step*, program berhenti (*delay*) selama 1000 mili detik.
4. Setelah eksekusi program *delay* selesai, program akan menggerakkan motor *stepper* berlawanan arah jarum jam (CCW) secara *half step*. Motor *stepper* akan bergerak 96 *step* (satu putaran) dalam waktu 1 detik. Agar SPC Stepper Motor menggerakkan motor *stepper* sebanyak 96 *step* dan dalam waktu 1 detik maka diperlukan pengaturan nilai StepSpeedH, StepSpeedL, StepH, dan StepL. Nilai-nilai tersebut dikirimkan ke modul SPC STEPPER MOTOR terlebih dahulu sebelum nilai Control sebesar 0CH (*bipolar, CCW, half step*). Kalkulasi nilai StepSpeedH, StepSpeedL, StepH, dan StepL untuk motor *stepper* berspesifikasi 7,5° adalah:
 - Satu putaran dalam mode *half step* dibutuhkan $(360^\circ \div 7,5^\circ) \times 2 \text{ step} = 96 \text{ step}$.
 - Nilai 96 *step* adalah dalam desimal yang setara dengan nilai 60 heksadesimal. Jadi nilai StepH = 00H dan StepL = 60H.
 - Waktu yang dibutuhkan per-*step*-nya adalah $1 \text{ detik} \div 96 \text{ step} = 0,01042 \text{ detik}$ atau setara dengan 10,4 mili detik.
 - Sehingga nilai StepSpeed adalah: $-2048 \times (10,4 - 32,5) = 45260,8$ (dalam desimal).
 - Nilai StepSpeed yang didapat adalah 45261 dalam desimal yang setara dengan nilai B0CD heksadesimal. Jadi nilai StepSpeedH = B0H dan StepSpeedL = CDH.
5. Setelah satu detik motor *stepper* akan berhenti dan program akan berakhir.

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

Listing program STEP_I2C.BS2 terdapat pada **AN134.ZIP**.

Dalam **AN134.ZIP** terdapat tambahan contoh program yaitu BASIC_STEP.BS2 yang berisi gabungan program STEP_PARALEL.BS2 dan STEP_I2C.BS2. Pada program BASIC_STEP.BS2, proses dalam program STEP_PARALEL.BS2 akan dijalankan terlebih dahulu, setelah itu barulah proses dalam program STEP_I2C.BS2 dijalankan. Jadi dalam program ini SPC Stepper Motor akan dikontrol dengan menggunakan jalur komunikasi paralel kemudian menggunakan jalur komunikasi I²C. Jika program BASIC_STEP.BS2 selesai dijalankan, maka posisi poros motor *stepper* akan kembali seperti semula.

Selamat Berinovasi!

PBASIC is a trademark of Parallax, Inc.
BASIC Stamp is a registered trademark of Parallax, Inc.
BASIC Stamp® Editor is copyright by Parallax, Inc.