

DT-BASIC DT-BASIC *Application Note*

AN137 – Kontrol Motor dengan DT-BASIC

Oleh: Tim IE

Aplikasi ini menjelaskan penggunaan modul DT-BASIC *series* yang menggunakan bahasa pemrograman PBASIC™ untuk mengendalikan Motor DC dengan bantuan SPC DC Motor. Modul *controller* yang digunakan di sini adalah DT-BASIC Nano System, dan programnya ditulis dengan bantuan *software compiler* BASIC Stamp® Editor®. Modul DT-BASIC Nano System dan SPC DC Motor dihubungkan secara paralel maupun secara I²C. Kedua jenis antarmuka ini didukung oleh modul SPC DC Motor.

Komponen yang diperlukan dalam aplikasi ini:

- 1 buah DT-BASIC Nano System (dapat diganti dengan DT-BASIC Micro System / Mini System)
- 1 buah SPC DC Motor
- 1 buah motor DC yang telah dilengkapi dengan *speed / rotary encoder*

Adapun blok diagram sistem secara keseluruhan adalah sebagai berikut:



Gambar 1
Blok Diagram AN137

Hubungan detail antara modul-modul tersebut adalah sebagai berikut:

DT-BASIC Nano System (J2) / Micro System (J3) / Mini System (J1)	SPC DC Motor
GND (Pin 1 / Pin 2)	GND (J1)
5VDC (Pin 19)	+5V (J1)
P6* (Pin 9)	S1 (J5 - Pin 1)
P4* (Pin 7)	S2 (J5 - Pin 2)
P2* (Pin 5)	S3 (J5 - Pin 3)
P0* (Pin 3)	S4 (J5 - Pin 4)

Tabel 1

Hubungan DT-BASIC Nano System / Micro System / Mini System Dengan SPC DC Motor Secara Paralel

DT-BASIC Nano System (J2) / Micro System (J3) / Mini System (J1)	SPC DC Motor
GND (Pin 1 / Pin 2)	GND (J1)
5VDC (Pin 19)	+5V (J1)
P8* (Pin 11)	SDA (J2)
P9* (Pin 12)	SCL (J2)

Tabel 2

Hubungan DT-BASIC Nano System / Micro System / Mini System Dengan SPC DC Motor Secara I²C

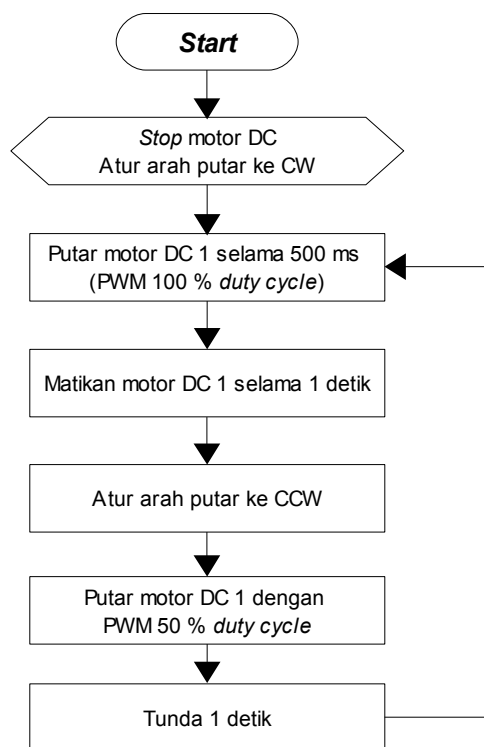
Hubungkan VMotor (J6) dengan sumber tegangan sesuai dengan kebutuhan tegangan motor yang digunakan. Misalnya motor DC yang digunakan memerlukan tegangan 12 Volt DC, maka hubungkan VMotor (J6) dengan kutub positif sumber tegangan 12 Volt DC dan GND (J6) dengan *ground* sumber tegangan 12 Volt DC. Selanjutnya hubungkan kutub positif motor DC pada M1+ dan kutub negatif motor DC pada M1-. Jika *rotary*

encoder membutuhkan catu daya 5V DC maka *input* catu daya dari *rotary encoder* dapat dihubungkan ke jalur catu daya +5V (J1) SPC DC Motor. Hubungkan keluaran dari *rotary encoder* ke IN1 (J6) SPC DC Motor. Pasang *jumper* pada *address* A1-A2 (J4), serta pasang semua *jumper pull up* pada SDA dan SCL (J3). *Jumper address* A0 (J4) tidak perlu dipasang karena alamat I²C untuk SPC DC Motor yang digunakan dalam program aplikasi ini adalah E2H. Pin yang diberi tanda * (P0, P2, P4, P6, P8 dan P9) tidak mutlak dan dapat diganti dengan pin lain dengan cara mengubah program.

Gunakan kabel serial untuk menghubungkan modul DT-BASIC Nano System / Micro System / Mini System ke *COM port* komputer. Kabel ini digunakan untuk melakukan *download* program ke DT-BASIC *series* dan untuk menerima data yang dikirimkan oleh DT-BASIC *series* (hanya untuk contoh program dengan komunikasi I²C). *Download* program dapat dilakukan dengan bantuan BASIC Stamp[®] Editor[®], dan data yang dikirimkan oleh DT-BASIC *series* dapat ditampilkan menggunakan jendela “Debug Terminal” dari BASIC Stamp[®] Editor[®].

Setelah semua rangkaian dan sumber tegangan terhubung dengan tepat sesuai dengan tabel di atas, programlah DC_PARALEL.BS2 (untuk komunikasi secara paralel) atau DC_I2C.BS2 (untuk komunikasi secara I²C) ke DT-BASIC Nano System dengan menekan tombol F9 atau CTRL+R pada program BASIC Stamp[®] Editor[®] (secara otomatis akan muncul jendela “Debug Terminal” untuk program DC_I2C.BS2). Agar program DC_PARALEL.BS2 atau DC_I2C.BS2 dapat berjalan pada modul DT-BASIC Micro System / Mini System, maka ubahlah *STAMP directive* (lihat *listing* program pada baris pertama) sesuai 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 DC_PARALEL.BS2 adalah sebagai berikut:



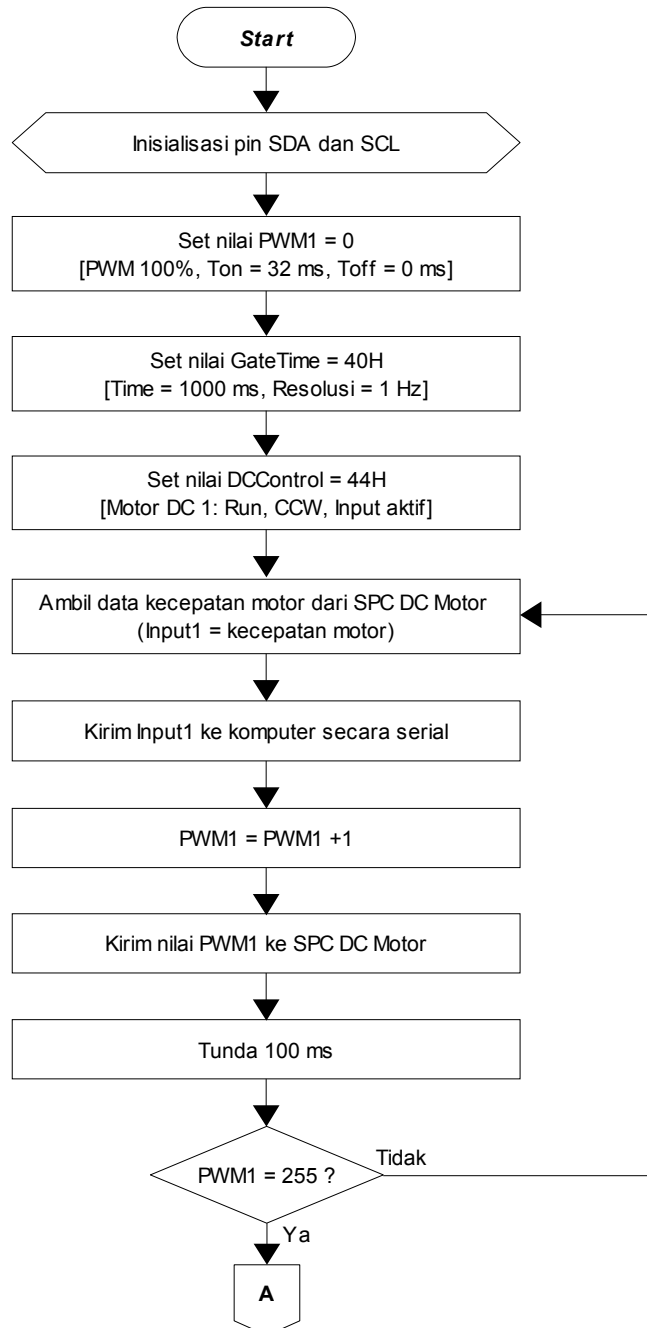
Gambar 2
Flowchart Program Untuk Hubungan Antar Modul Secara Paralel

Program utama DC_PARALEL.BS2 akan diproses sebagai berikut:

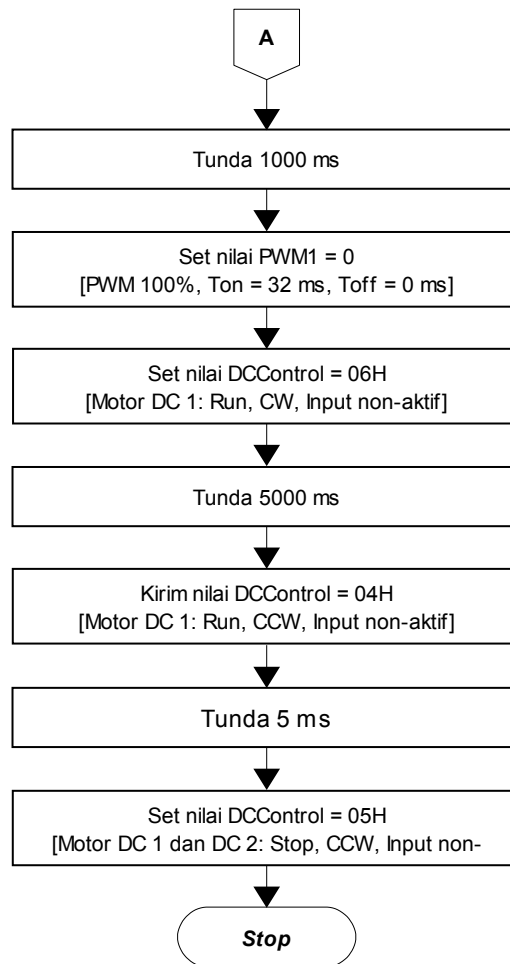
1. Proses yang pertama kali dilakukan adalah inialisasi awal motor DC 1 (Stop / Run, diatur pada pin S1) dan memilih arah putaran motor (CW / CCW, diatur pada pin S2), sedangkan untuk motor DC 2 selalu dalam keadaan berhenti. Semua pin J5 (pin S1 - S4) diberi logika ‘1’ (*high*) sehingga motor DC 1 dan motor DC 2 dalam keadaan berhenti (*Stop*) serta arah putaran searah jarum jam (CW).
2. Setelah proses inialisasi selesai, berikutnya program akan menjalankan (*Run*) motor DC 1 searah jarum jam (CW) dengan memberikan logika ‘0’ pada pin S1. Motor DC 1 akan berputar selama 500 mili detik.
3. Setelah program memutar motor DC 1 searah jarum jam selama 500 mili detik, program akan menghentikan motor DC 1 selama 1000 mili detik.

4. Kemudian program akan membalik arah putaran motor DC 1 (berputar berlawanan arah jarum jam, CCW) dengan memberi logika '0' pada pin S2. Setelah itu program akan memberikan *sinyal* PWM 50% dengan frekuensi ± 100 Hz sebanyak 100 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 5 mili detik dan berhenti selama 5 mili detik).
5. Setelah langkah pada nomor 4 selesai, program akan berhenti selama 1000 mili detik dan selanjutnya program akan kembali ke langkah 2.

Flowchart dari program **DC_I2C.BS2** adalah sebagai berikut:



Gambar 3
Flowchart Program Untuk Hubungan Antar Modul Secara I²C



Gambar 4
Flowchart Program Untuk Hubungan Antar Modul Secara I²C (Lanjutan Dari Gambar 3)

Program utama (DC_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 I²C SPC DC Motor yaitu E2H.
2. Proses selanjutnya yaitu 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 = 32 mili detik dan Toff = 0 mili detik.
3. Setelah pengiriman nilai PWM1 selesai, program mengirimkan nilai GateTime sebesar 40H. Nilai GateTime 40H pada SPC DC Motor mempunyai arti waktu penghitungan kecepatan putaran motor selama 1 detik dengan resolusi penghitungan putaran motor 1 Hz.
4. Setelah program mengirimkan nilai GateTime, program akan mengirimkan nilai DCControl sebesar 44H. Nilai 44H mempunyai arti SPC DC Motor akan menjalankan motor DC 1 dengan putaran berlawanan arah jarum jam (CCW) dan menghitung kecepatan putaran motor DC 1.
5. Proses berikutnya adalah membaca kecepatan putaran motor DC 1 dan hasil pembacaan disimpan pada variabel Input1 (InputL1 dan InputH1). Kemudian program akan mengirimkan variabel Input1 ke komputer (dapat dibaca melalui Debug Terminal) dalam bentuk desimal.
6. Proses berikutnya adalah menambah variabel PWM1 dengan nilai 1 dan mengirimkannya ke SPC DC Motor. Semakin besar nilai PWM1 maka kecepatan putar motor semakin lambat, sebab nilai Ton (waktu hidup motor) semakin kecil. Setelah selesai mengirimkan nilai PWM1, program akan berhenti selama 100 mili detik (*delay*).
7. Langkah 5 dan 6 dilakukan terus-menerus sampai nilai variabel PWM1 sudah maksimum (255d) sehingga motor DC 1 berhenti berputar. Proses berikutnya adalah melakukan *delay* 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 selesai mengirimkan variabel PWM1, dilakukan pengiriman nilai DCControl 06H (motor DC 1 berputar searah jarum jam dengan input non-aktif). Ini berarti SPC DC Motor akan

menjalankan motor DC 1 dengan PWM 100% searah jarum jam. Setelah itu *delay* selama 5 detik agar motor berputar dengan kecepatan penuh.

8. Proses berikutnya yaitu melakukan pengereman (*brake*) secara cepat dengan memutar arah putaran motor secara terbalik selama 5 mili detik dan mematikan putaran motor (*Stop*). Proses ini dilakukan dengan mengirimkan nilai DCControl 04H (motor DC 1 berputar berlawanan arah jarum jam dengan *input non-aktif*) kemudian *delay* selama 5 mili detik. Sebelum putaran motor berbalik arah dilakukan pengiriman nilai DC Control 05H (motor DC 1 dan motor DC 2 Stop) untuk mematikan motor DC 1.

Listing program DC_I2C.BS2 terdapat pada **AN137.ZIP**.

Dalam **AN137.ZIP** terdapat tambahan contoh program yaitu BASIC_DC.BS2 yang berisi gabungan program DC_PARALEL.BS2 dan DC_I2C.BS2. Pada program BASIC_DC.BS2, program DC_PARALEL.BS2 (dengan adanya perubahan program) dijalankan terlebih dahulu kemudian program DC_I2C.BS2. Jadi SPC DC Motor dikontrol dengan menggunakan jalur komunikasi paralel kemudian menggunakan jalur komunikasi I²C.

Selamat berinovasi!

All trademarks, trade names, company names, and product names are the property of their respective owners.
All softwares are copyright by their respective software publishers and/or creators.