

# SPC

SMART PERIPHERAL CONTROLLER

## STEPPER MOTOR

### Quick Start

#### Trademarks & Copyright

XT, AT, IBM, PC, and PC-DOS are trademarks of International Business Machines Corp.

MS-DOS is a registered trademark of Microsoft Corporation.

Pentium is a registered trademark of Intel Corporation.

MetaLink ASM51 is copyright by MetaLink Corporation

#### 1. PENDAHULUAN

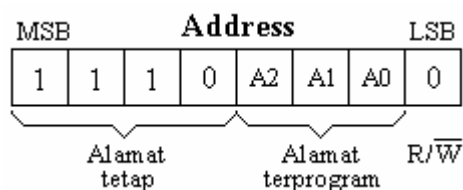
*Smart Peripheral Controller* / SPC STEPPER MOTOR merupakan pengontrol motor stepper yang menggunakan I<sup>2</sup>C-bus sebagai jalur pengiriman data sehingga dapat lebih menghemat dan mempermudah pengkabelan, selain itu SPC STEPPER MOTOR dapat digunakan secara paralel. Contoh aplikasi dari SPC STEPPER MOTOR adalah untuk robot dan sumber gerak lainnya.

#### 2. SPESIFIKASI SPC STEPPER MOTOR

Spesifikasi SPC STEPPER MOTOR sebagai berikut :

- Kompatibel penuh dengan DT-51 Minimum System Ver 3.0.
- Hanya perlu 2 jalur kabel untuk interface dengan mikroprosesor / mikrokontroler lain.
- Dapat digunakan untuk unipolar atau bipolar stepper motor.
- Dapat digunakan pada I<sup>2</sup>C-bus maupun paralel.
- Semua pin – pin paralel diakses dengan taraf logik TTL
- Dilengkapi dengan jumper untuk setting alamat, sehingga dapat di-ekspan sampai 8 board tanpa tambahan perangkat keras .
- Tersedia prosedur siap pakai untuk aplikasi SPC STEPPER MOTOR.

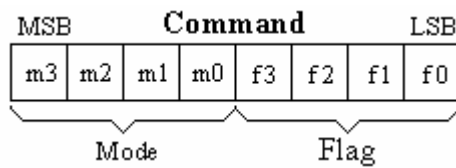
#### 3. Pengalamatan



Pengalamatan memanfaatkan register : *AddressI2C* dengan alamat memory 2Fh

Semua penggunaan dari I<sup>2</sup>C-bus selalu diawali dengan pengalamatan. Pada pengalamatan itu sendiri dibedakan menjadi tiga bagian : alamat tetap, alamat terprogram, dan Read/Write (R/W). SPC STEPPER MOTOR selalu menggunakan alamat tetap dengan nilai "1110", dan Read/Write selalu bernilai Write ("0") sedangkan alamat terprogram digunakan untuk memberikan alamat terhadap modul sesuai dengan kehendak pemakai. Alamat terprogram diatur dengan cara mengubah setting jumper (dapat dilihat pada **bagian 6.2**) sehingga pada jalur I<sup>2</sup>C yang sama dengan alamat tetap yang sama ("1110") dapat digunakan 8 buah modul secara bersamaan dengan membedakan alamat terprogram.

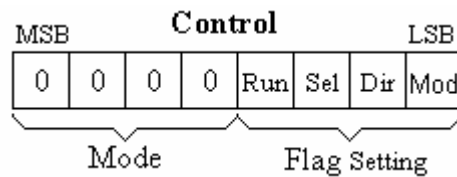
#### 4. Command



m3	m2	m1	M0	Mode
0	0	0	0	Command <b>Control</b>
0	0	0	1	Command <b>Speed</b>
0	0	1	0	Command <b>Step</b>
0	0	1	1	<i>Tidak terpakai</i>
.	.	.	.	.
.	.	.	.	.
1	1	1	1	<i>Tidak terpakai</i>

Command terdiri dari dua bagian utama yaitu Mode dan Flag. Mode digunakan untuk memilih perintah selanjutnya yang akan diberikan pada device sesuai dengan pilihan mode yang diberikan. Command memiliki 16 kemungkinan mode, namun pada SPC STEPPER MOTOR ini hanya digunakan 3 mode pertama saja. Flag digunakan untuk mengatur aktivitas / kegiatan dari setiap mode yang diberikan, pada mode – mode tertentu Flag tidak digunakan sehingga Flag dapat diisi sembarang (don't care / ' X '). Pada SPC STEPPER MOTOR ini Flag hanya dapat digunakan pada Command **Control**.

##### 4.1 Command Control



Control	Setting ( H/ $\bar{L}$ )	Fungsi
Run	Run / $\overline{\text{Stop}}$	Untuk menjalankan dan menghentikan motor stepper : <i>Run</i> beri logika '1' (high) <i>Stop</i> beri logika '0' (low)
Sel	Bi / $\overline{\text{Uni}}$	Untuk tipe motor stepper yang digunakan : <i>Motor stepper Bipolar</i> beri logika '1' (high) <i>Motor stepper Unipolar</i> beri logika '0' (low)
Dir	CW / $\overline{\text{CCW}}$	Untuk arah putaran motor stepper : <i>CW</i> (searah jarum jam) beri logika '1' (high) <i>CCW</i> (berlawanan arah jarum jam) beri logika '0' (low)
Mod	Full / $\overline{\text{Half}}$	Untuk mode putaran motor stepper : <i>Full*</i> (penuh) beri logika '1' (high) <i>Half*</i> (setengah) beri logika '0' (low)

Command Control memanfaatkan register : **StepControl**

Memanfaatkan alamat memory 38h atau dengan nama lain *BufferOut0*

Command control digunakan untuk mengatur semua kegiatan dari motor stepper.

\* Yang dimaksud dengan Full adalah suatu mode motor stepper dimana per langkahnya merupakan pergeseran maksimum antara rotor terhadap stator sehingga mode Full merupakan mode tercepat untuk melakukan pergeseran/perputaran namun memiliki kekurangan dari segi torsi yang dihasilkan .

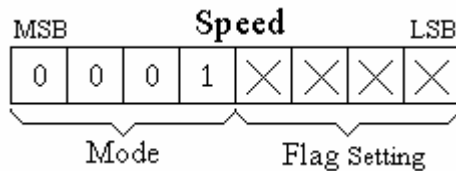
\* Yang dimaksud dengan Half adalah setiap pergeseran per langkah merupakan setengah dari pergeseran yang dihasilkan oleh mode Full, sehingga mode Half memiliki kecepatan yang lebih lambat dari mode Full namun memiliki torsi yang lebih kuat.

Contoh:

Bila ingin menjalankan motor stepper unipolar dengan arah searah jarum jam dan dengan mode half maka register StepControl harus di isi dengan '00001010b' atau setara dengan '0Ah'.

Bila ingin membuat berhenti maka control run harus dibuat 'low', yaitu: '00000010'.

## 4.2 Command Speed



Command Speed memanfaatkan register : *StepSpeedH* dan *StepSpeedL*

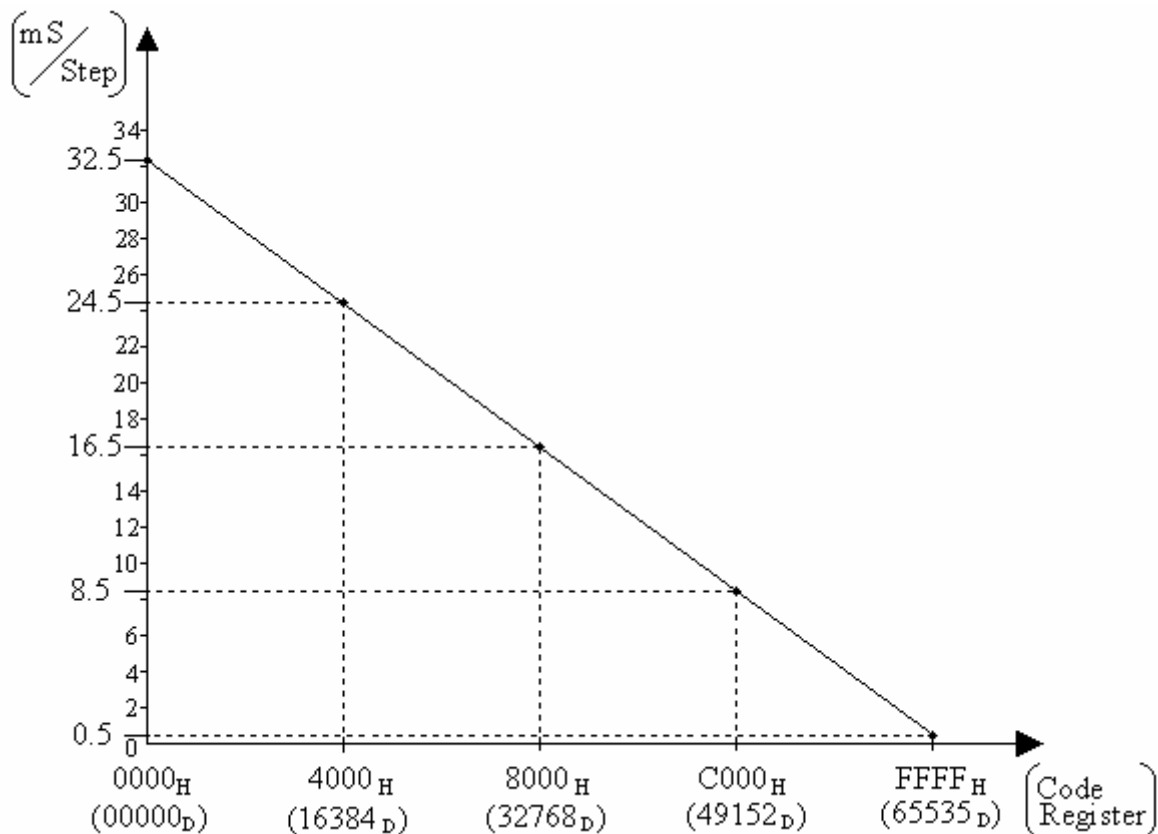
Memanfaatkan alamat memory 39h atau dengan nama lain *BufferOut1* untuk StepSpeedH

Memanfaatkan alamat memory 3Ah atau dengan nama lain *BufferOut2* untuk StepSpeedL

Command Speed merupakan command kedua dan dalam penggunaannya harus dikirimkan Command Speed "00010000" terlebih dahulu kemudian diikuti oleh StepSpeedH dan StepSpeedL. Namun user hanya perlu mengisi StepSpeedH, StepSpeedL saja dan secara otomatis dalam pengiriman akan ditambahkan Command Speed "00010000".

StepSpeedH dan StepSpeedL digunakan untuk memberikan besaran kecepatan yang dikehendaki dengan mengirimkan 2 byte kode kecepatan.

Dari 0000h untuk paling lambat sampai FFFFh untuk yang tercepat.



Grafik di atas menggambarkan waktu yang dibutuhkan (dalam milidetik) untuk menempuh satu step / satu langkah terhadap isi dari register StepSpeedH dan StepSpeedL.

Berdasarkan grafik diatas maka didapatkan suatu rumus

$$\left( \frac{\text{mS}}{\text{Step}} \right) = 32.5 - \frac{\text{Code}}{2048} \quad \text{atau} \quad \text{Code} = -2048 \left[ \left( \frac{\text{mS}}{\text{Step}} \right) - 32.5 \right]$$

**Contoh aplikasi :**

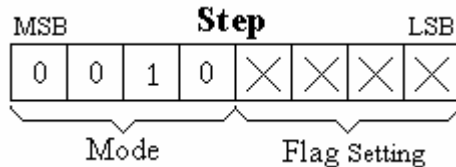
Bila dikehendaki motor stepper bergerak dengan waktu 10mS per step maka didapatkan :

$$\text{Code} = -2048(10-32.5) = 46080 \text{ (dalam desimal)}$$

Code yang didapat adalah 46080 dalam desimal sehingga harus diubah ke bentuk hexa, dengan hasil B400 dalam hexa. Maka register “StepSpeedH” berisi ‘B4h’ dan register “StepSpeedL” berisi ‘00h’.

- perlu diperhatikan kemampuan kecepatan dari motor stepper yang akan digunakan untuk mendapatkan torsi yang maksimal dan agar tidak terjadi slip.

**4.3 Command Step**



**Command Step memanfaatkan register : StepH dan StepL**

Memanfaatkan alamat memory 3Bh atau dengan nama lain *BufferOut3* untuk StepH

Memanfaatkan alamat memory 3Ch atau dengan nama lain *BufferOut4* untuk StepL

Command Step merupakan command ketiga dan dalam penggunaannya harus dikirimkan Command Step “00100000” terlebih dahulu kemudian diikuti oleh StepH dan StepL. Namun user hanya perlu mengisi StepH, StepL saja dan secara otomatis dalam pengiriman akan ditambahkan Command Speed “00100000”.

Command step digunakan untuk memberikan besaran langkah yang harus ditempuh oleh motor stepper dengan mengirimkan 2 byte kode langkah.

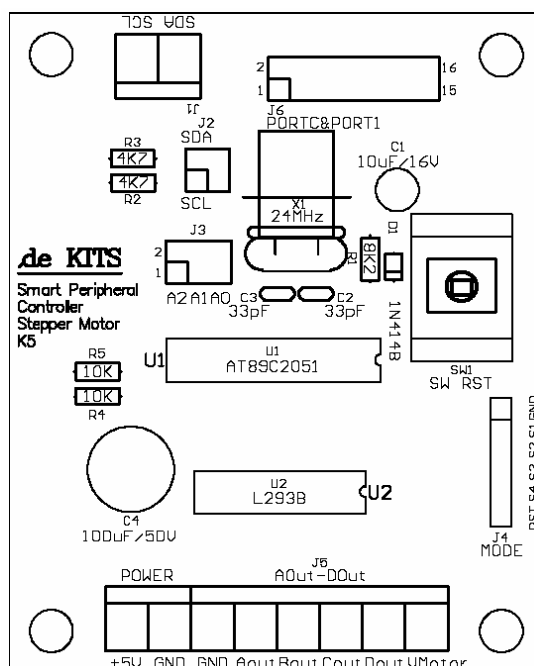
Dari 0000h untuk berhenti (tidak ada langkah) sampai FFEh untuk langkah terbanyak.

**Contoh aplikasi :**

Bila dikehendaki berjalan 400 langkah maka (400d setara 0190h) register “StepH” berisi ‘01h’ dan register “StepL” berisi ‘90h’.

- Fungsi tambahan dari command step adalah dapat membuat motor stepper untuk terus berputar tanpa henti dengan cara mengirimkan FFFFh.

**5. TATA LETAK KOMPONEN SPC STEPPER MOTOR**



## 6. SISTEM YANG DIANJURKAN

### Perangkat keras :

- PC XT / AT Pentium™ IBM Compatible dengan port serial (COM 1/ COM2).
- Board DT-51 Minimum System.
- Floppy Disk 3.5", kapasitas 1,44Mbytes atau CD-ROM Drive.
- Hard disk dengan kapasitas minimum 500Kbytes.

### Perangkat lunak :

- Sistem operasi MS-DOS™ atau PC-DOS™.
- Assembler MetaLink ASM51<sup>©</sup>
- File-file yang ada pada pada disket/CD program.

### 6.1 HUBUNGAN DT-51 MINIMUM SYSTEM DENGAN SPC STEPPER MOTOR

SPC STEPPER MOTOR bila dihubungkan dengan DT-51 Minimum System akan menghasilkan suatu sistem yang 'Smart'. Namun user tetap bisa menghubungkan SPC STEPPER MOTOR ini dengan sistem mikroprosesor / mikrokontroler yang lain atau difungsikan secara paralel (lihat **bagian 6.4**). Apabila Anda ingin menghubungkan SPC STEPPER MOTOR dengan sistem yang lain, kami sarankan untuk mempelajari skema SPC STEPPER MOTOR.

Untuk menghubungkan SPC STEPPER MOTOR dengan DT-51 Minimum System dianjurkan untuk menggunakan kabel pita (flat ribbon cable).

Hubungannya ditunjukkan pada tabel berikut :

I <sup>2</sup> C Bus	DT-51 Minimum System PORT C & PORT 1	SPC STEPPER MOTOR J6
SCL	Pin 15 (Port 1.6)	Pin 15 (Port 3.3)
SDA	Pin 16 (Port 1.7)	Pin 16 (Port 3.2)

Catu daya 5V DC dihubungkan dengan konektor J5 (Power). Perhatikan polaritasnya jangan sampai terbalik, karena dapat mengakibatkan kerusakan.

#### Penting !

Referensi ground (GND) antara modul SPC STEPPER MOTOR dengan DT-51 Minimum System harus sama.

### 6.2 SETTING JUMPER

Alamat terprogram setiap board SPC STEPPER MOTOR ditentukan oleh setting jumper J3.

J3 (A2)	J3 (A1)	J3(A0)	Alamat Terprogram	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	000
<input type="checkbox"/>	<input type="checkbox"/>		1	001
<input type="checkbox"/>		<input type="checkbox"/>	2	010
<input type="checkbox"/>			3	011
	<input type="checkbox"/>	<input type="checkbox"/>	4	100
	<input type="checkbox"/>		5	101
		<input type="checkbox"/>	6	110
			7 (default)	111

Keterangan :

: jumper tersambung (ON)

Jumper J2 (SCL/SDA) digunakan untuk resistor pull up SDA (I<sup>2</sup>C bus data input / output) dan SCL (I<sup>2</sup>C bus clock input). Apabila lebih dari satu board SPC STEPPER MOTOR dihubungkan pada I<sup>2</sup>C bus maka jumper J2 (SCL/SDA) salah satu board saja yang perlu dipasang.

### 6.3 EKSPANSI SPC STEPPER MOTOR


SPC STEPPER MOTOR dapat di-ekspan sampai 8 board. Beberapa hal yang perlu diperhatikan apabila menggunakan lebih dari satu board SPC Stepper MOTOR :

- Setiap board harus mempunyai alamat terprogram yang berbeda, ditentukan oleh jumper J3 (A0/A1/A2).
- Jumper J2 pada salah satu board saja yang dipasang.

#### 6.4 PENGGUNAAN SPC STEPPER MOTOR SECARA PARALEL

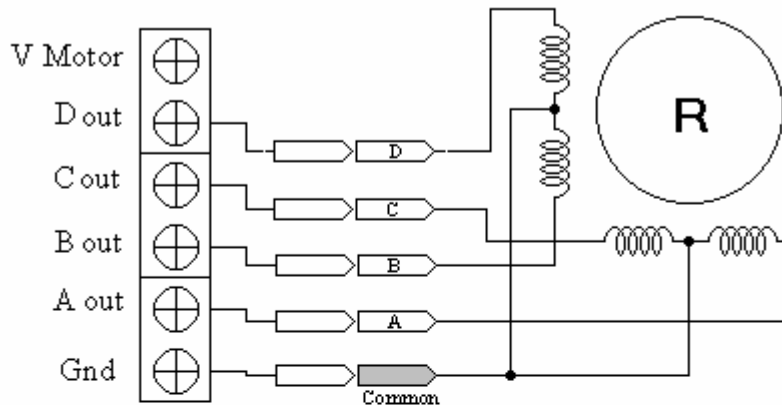
SPC STEPPER MOTOR dapat digunakan secara paralel dengan cara mengatur pin – pin S1, S2, S3, S4, RST yang ada pada board SPC STEPPER MOTOR. Berikut adalah tabel kegunaan dari pin – pin tersebut:

Pin	Name	Setting	Fungsi
S1	Sel	Bi / $\overline{\text{Uni}}$	Untuk tipe motor stepper yang digunakan : <i>Motor stepper Bipolar beri logika '1' (high)</i> <i>Motor stepper Unipolar beri logika '0' (low)</i>
S2	Dir	CW / $\overline{\text{CCW}}$	Untuk arah putaran motor stepper : <i>CW (searah jarum jam) beri logika '1' (high)</i> <i>CCW (berlawanan arah jarum jam) beri logika '0' (low)</i>
S3	Mode	Full / $\overline{\text{Half}}$	Untuk mode putaran motor stepper : <i>Full* (penuh) beri logika '1' (high)</i> <i>Half* (setengah) beri logika '0' (low)</i>
S4	Step	Clock	Untuk menjalankan motor stepper sebanyak jumlah clock
RST	Reset	RST	Untuk melepaskan motor stepper dari keadaan "lock"

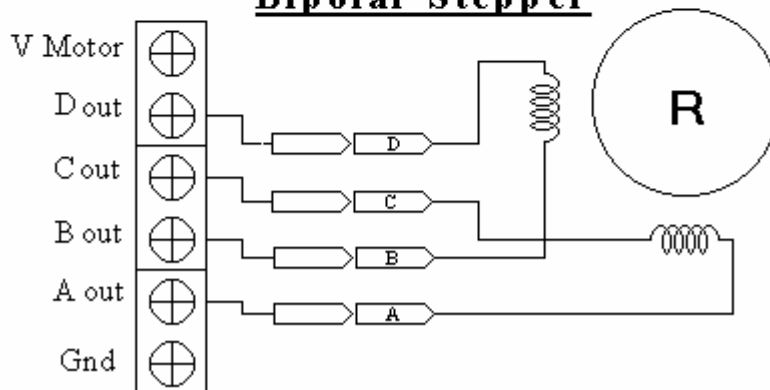
- Secara default jika pin – pin S1, S2, S3, S4 tersebut tidak dihubungkan maka berlogika “high” kecuali Pin RST akan berlogika “low”.
- Pin S4 (Clock) merupakan falling edge triggering ( transisi high ke low ) sehingga untuk menjalankan motor stepper beberapa langkah harus diberikan pulsa sebanyak jumlah langkah, bentuk pulsa yang dianjurkan adalah persegi dengan level TTL (“low”=0V – 0.8V dan “High”=2.5V – 5V).
- Pin RST merupakan pin yang digunakan untuk melepaskan motor stepper dari kondisi mengunci, untuk melakukan reset dilakukan dengan memberikan pulsa “high” satu kali (  ). Reset juga dapat dilakukan dengan menekan tombol reset (SW RST) yang ada pada board
- Apabila pada saat yang bersamaan terjadi pengaturan secara ‘I<sup>2</sup>C’ dan ‘Paralel’ maka yang menjadi prioritas adalah I<sup>2</sup>C, setelah perintah I<sup>2</sup>C selesai dilaksanakan maka perintah paralel baru dapat dilaksanakan.
- Setelah selesai memberikan pulsa secara paralel maka sebaiknya akhir dari pulsa adalah level “high”.
- **Penting !**  
Dalam pengontrolan secara paralel setiap selesai memberikan pulsa maka motor stepper akan selalu pada kondisi mengunci (lock), kondisi mengunci pada waktu yang relatif lama dapat membuat motor stepper atau modul SPC STEPPER MOTOR terbakar. Untuk itu setiap akhir dari pemberian pulsa maka harus diberikan pulsa reset atau menekan tombol Reset.  
Kecuali pada kondisi khusus di mana memanfaatkan kondisi mengunci.

Contoh: Apabila menghendaki motor stepper bergerak dengan kecepatan 200 langkah per detik dan menghasilkan 800 langkah maka pulsa persegi yang diberikan adalah 200 Hz selama (800langkah / 200Hz) 4 detik kemudian diberikan pulsa reset.

## Unipolar Stepper



## Bipolar Stepper



### 6.5 PENYAMBUNGAN SPC STEPPER MOTOR DENGAN MOTOR STEPPER

Dalam penyambungan motor stepper dengan modul SPC STEPPER MOTOR perlu diperhatikan tipe dari motor stepper yang akan dipergunakan, Modul SPC STEPPER MOTOR dapat dipergunakan untuk dua macam tipe motor stepper : Unipolar dan Bipolar. Berikut adalah cara pemasangan dari kedua tipe motor stepper tersebut.

- ❑ Modul SPC STEPPER MOTOR dapat dipergunakan untuk motor stepper dengan tegangan kerja dari 6 Volt sampai dengan 36 Volt.
- ❑ Arus RMS maksimum untuk modul SPC STEPPER MOTOR adalah 600 mA.
- ❑ Arus impuls tak berulang maksimum untuk modul SPC STEPPER MOTOR adalah 1.2 A.
- ❑ Sudah dilengkapi dioda clamp secara internal.
- ❑ Hubungkan catu daya positif (+) untuk motor stepper pada *VMotor* dan negatif (-) pada *GND*.  
Tegangannya harus sesuai dengan tegangan kerja motor.
- ❑ Sambungkan semua kabel A, B, C, D pada Bipolar Stepper atau A, B, C, D, dan Common pada Unipolar Stepper secara benar (lihat gambar).

### 6.6 MENCoba SPC STEPPER MOTOR DENGAN EXAMPLE.HEX

- ◆ Hubungkan DT-51 Minimum System dengan SPC STEPPER MOTOR (lihat bagian 6.1)
- ◆ Hubungkan SPC STEPPER MOTOR dengan motor stepper Bipolar (lihat bagian 6.5)
- ◆ Setting alamat SPC Stepper Motor pada alamat terprogram 7 (default)
- ◆ Download EXAMPLE.HEX yang terdapat pada disket/CD
- ◆ Motor Stepper akan bergerak dalam mode full, Clockwise (CW) 180°, Counter Clockwise (CCW) 90°, demikian seterusnya.

## 7. PERANGKAT LUNAK SPC STEPPER MOTOR

### 7.1 DRIVER DAN RUTIN

SPC STEPPER MOTOR dilengkapi dengan driver **STEPPER.INC** yang akan mempermudah user dalam pemrograman. **STEPPER.INC** menggunakan resource dari mikrokontroler 89C51 sebagai berikut :

- Internal RAM alamat 21h bit 0 dan 1.
- Internal RAM dengan alamat 2Fh – 3Fh
- P1.6 dan P1.7

Sehingga tidak boleh dipakai oleh user untuk keperluan lain, kecuali user mampu melakukan modifikasi pengaturan memori dengan benar.

Driver ini menggunakan 6 buah register yang terdiri dari:

AddressI2C	StepSpeedH	StepH
StepControl	StepSpeedL	StepL

Kegunaan dari keenam register dapat dilihat pada **bagian 4**.

Dari keenam register tersebut akan digunakan dalam tiga rutin penting berikut :

#### **StepInit**

Fungsi	: Untuk mengirimkan data yang ada pada keenam register SPC STEPPER MOTOR melalui I <sup>2</sup> C-bus.
Input	: AddressI2C, StepControl, StepSpeedH, StepSpeedL, StepH, StepL.
Output	: Flag FAck
Keterangan	: <ul style="list-style-type: none"><li>❖ Rutin ini dapat digunakan untuk memberikan nilai awal atau inisialisasi tanpa menjalankan motor stepper, yaitu dengan memberi logika low '0' pada setting Run dari StepControl. Apabila dikehendaki saat memanggil rutin StepInit dapat langsung menjalankan motor stepper maka setting Run pada StepControl harus berlogika high '1'.</li><li>❖ Rutin ini juga dapat digunakan untuk mengubah instruksi yang lama dengan yang instruksi baru walaupun instruksi yang lama belum selesai dieksekusi / atau sedang dalam proses eksekusi. Proses ini dapat dilakukan dengan cara isi instruksi yang baru pada register kemudian panggil rutin StepInit, maka saat itu pula SPC STEPPER MOTOR akan melaksanakan instruksi yang baru.</li><li>❖ Pada saat instruksi I<sup>2</sup>C sedang dieksekusi maka semua instruksi paralel tidak dapat dilaksanakan.</li><li>❖ Setelah Rutin StepStop dipanggil atau instruksi I<sup>2</sup>C selesai dieksekusi maka perintah paralel baru dapat dilaksanakan.</li><li>❖ Setelah instruksi I<sup>2</sup>C selesai dijalankan maka motor stepper tidak pada kondisi mengunci.</li></ul>
Metode	: Berikan semua nilai - nilai untuk AddressI2C, StepControl, StepSpeedH, StepSpeedL, StepH, StepL sesuai dengan kebutuhan, kemudian panggil sub rutin StepInit untuk mengirimkan data secara I <sup>2</sup> C.

#### **StepRun**

Fungsi	: Untuk membuat motor stepper berputar secara terus menerus sampai instruksi I <sup>2</sup> C yang baru diberikan.
Input	: AddressI2C
Output	: Flag FAck
Keterangan	: <ul style="list-style-type: none"><li>❖ Rutin ini digunakan untuk membuat agar motor stepper berputar secara terus menerus.</li><li>❖ Kecepatan, tipe, mode, arah dari motor stepper berdasarkan dari setting kontrol yang sudah ada sebelumnya.</li><li>❖ Rutin ini secara langsung memberikan nilai FFh pada kedua register StepH dan StepL yang akan membuat motor selalu berputar secara terus menerus.</li></ul>

❖ Pada saat rutin ini dipanggil maka semua instruksi paralel tidak dapat dilaksanakan.

Metode : Isi register AddressI2C sesuai dengan alamat kemudian panggil rutin StepRun.

### **StepStop**

Fungsi : Untuk membuat motor stepper berhenti .  
 Input : AddressI2C  
 Output : Flag FAck  
 Keterangan :

- ❖ Rutin ini digunakan untuk membuat agar motor stepper berhenti (stop) dan keluar dari instruksi I<sup>2</sup>C.
- ❖ Rutin ini juga dapat digunakan untuk menghentikan rutin StepRun.
- ❖ Setelah perintah ini atau instruksi I<sup>2</sup>C selesai dieksekusi maka perintah paralel baru dapat dilaksanakan.
- ❖ Dengan memberikan nilai 00h pada register StepH, StepL akan membuat motor berhenti.
- ❖ Rutin ini tidak dapat digunakan untuk menghentikan instruksi paralel.
- ❖ Setelah instruksi I<sup>2</sup>C selesai dijalankan maka motor stepper tidak pada kondisi mengunci.

Metode : Isi register AddressI2C sesuai dengan alamat kemudian panggil rutin StepStop.

## **7.2 CONTOH APLIKASI DAN PROGRAM**

Bila dikehendaki modul SPC STEPPER MOTOR dengan alamat terprogram ke-3 menggunakan motor stepper unipolar bergerak dalam mode half dengan arah berlawanan jarum jam (CCW) menempuh 5 putaran dalam waktu 4 detik maka register-register yang harus diisi adalah adalah:

*Pada umumnya motor stepper unipolar dengan mode full untuk menempuh satu putaran membutuhkan 200 step, sehingga untuk mode half dibutuhkan 400 step per satu putaran.*

Sedangkan untuk menempuh 5 putaran maka dibutuhkan 2000 step (5 putaran X 400 step per putaran) dalam 4 detik. 2000 (dalam desimal) step setara dengan 7D0 (dalam hexa) step Maka waktu yang dibutuhkan untuk per stepnya adalah 4 detik / 2000 = 0.002 detik atau setara 2 milidetik (2mS). Sehingga code yang didapatkan adalah:

$$\text{Code} = -2048(2-32.5) = 62464 \text{ (dalam desimal)}$$

Code yang didapat adalah 62464 dalam desimal sehingga harus diubah ke bentuk hexa, dengan hasil F400 dalam hexa.

Maka register – register yang harus diisi adalah:

AddressI2C = 11100110b = E0h  
 StepControl = 00001000b = 08h  
 StepSpeedH = F4h  
 StepSpeedL = 00h  
 StepH = 07h  
 StepL = D0h

### **Cuplikan Listing program untuk kasus diatas:**

```
MOV AddressI2C, #11100110B ;memasukan nilai alamat
MOV StepControl, #00001000B ;memasukan nilai Control
MOV StepSpeedH, #0F4H ;memasukan nilai Timer
MOV StepSpeedL, #000H
MOV StepH, #007H ;memasukan nilai langkah
MOV StepL, #0D0H
ACALL StepInit ;memanggil rutin StepInit
```

### **Catatan :**

- ◆ Bagi user yang ingin mempelajari lebih lanjut mengenai SPC STEPPER MOTOR dapat membaca MANUAL SPC STEPPER MOTOR.PDF serta contoh program EXAMPLE.ASM yang disertakan pada disket/CD.
- ◆ Technical Support : support@innovativeelectronics.com