

SPC

SMART PERIPHERAL CONTROLLER

DC MOTOR

Trademarks & Copyright

AT is a trademark of International Business Machines Corp.
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

Daftar Isi

1. Pendahuluan	3
1.1. Spesifikasi Eksternal SPC DC MOTOR	3
1.2. Spesifikasi Internal SPC DC MOTOR	3
1.2.1. Pengalamatan	4
1.2.2. Command	4
1.2.2.1. Command Control	5
1.2.2.2. Command PWM	6
1.2.2.3. Command GateTime	7
1.2.2.4. Command Input	8
1.3. Spesifikasi I2C Bus	8
1.3.1. Kondisi Start dan Stop	9
1.3.2. Transfer Data	9
1.4. Tata Letak Komponen SPC DC MOTOR	11
2. Sistem yang Dianjurkan	11
2.1. Hubungan DT-51 Minimum Sistem dengan SPC DC MOTOR	11
2.2. Setting Jumper	12
2.3. Eskpansi SPC DC MOTOR	12
2.4. Penggunaan SPC DC MOTOR secara Paralel	12
2.5. Penyambungan SPC DC MOTOR dengan Motor DC	13
2.6. Mencoba SPC DC MOTOR dengan EXAMPLE.HEX	14
3. Perangkat Lunak SPC DC MOTOR	14
3.1. Driver dan Rutin	14
3.2. Contoh Aplikasi dan Program	17
3.3. Kerangka Program	18
Lampiran	20
Skema SPC DC MOTOR	20
Protokol SPC DC MOTOR	21

1. PENDAHULUAN

Smart Peripheral Controller / SPC DC MOTOR merupakan pengontrol motor DC yang menggunakan I²C-bus sebagai jalur penyampaian data sehingga dapat lebih menghemat dan mempermudah pengkabelan. SPC DC MOTOR ini dilengkapi dengan prosedur **input** sehingga dapat mengetahui kecepatan motor pada saat tertentu, juga dilengkapi dengan prosedur **brake** yang dapat menghentikan motor secara cepat. Selain itu SPC DC MOTOR dapat digunakan secara paralel. Contoh aplikasi dari SPC DC MOTOR adalah untuk robot, dan sumber gerak lainnya.

1.1 SPESIFIKASI EKSTERNAL SPC DC MOTOR

Spesifikasi Eksternal SPC DC MOTOR sebagai berikut :

- Kompatibel penuh dengan DT-51 Minimum System Ver 3.0.
- Hanya perlu 2 jalur kabel untuk interface dengan mikroprosesor / mikrokontroler lain.
- Mempunyai 2 buah pengontrol motor DC yang dapat bekerja secara bersama-sama.
- Masing-masing pengontrol motor DC dilengkapi dengan prosedur input dan brake.
- Dapat dikontrol secara I²C-bus maupun paralel.
- Pengaturan kecepatan motor menggunakan metode Pulse Width Modulation (PWM).
- Semua pin-pin kontrol paralel diakses dengan taraf logika TTL.
- Dilengkapi dengan jumper untuk setting alamat, sehingga bila menggunakan I²C bus dapat di-ekspan sampai 8 board (16 buah motor DC) tanpa tambahan perangkat keras.
- Tersedia prosedur siap pakai untuk aplikasi SPC DC MOTOR.

1.2 SPESIFIKASI INTERNAL SPC DC MOTOR

Dalam penggunaan dari SPC DC MOTOR akan dikenal adanya tiga layer (lapisan) penggunaan:

Pertama	:	I ² C Engine Layer
Kedua	:	I ² C Protocol Layer
Ketiga	:	I ² C Application Layer

I²C Engine Layer adalah lapisan yang mengurus kegiatan dari tiap bit yang akan diterima atau yang akan dikirim.

Bagian ini tidak boleh diubah kecuali untuk keperluan khusus. Bagi pengguna yang belum mahir dan berpengalaman tidak dianjurkan untuk mengubah bagian ini.

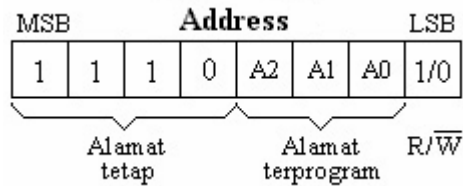
I²C Protocol Layer adalah lapisan yang terletak satu lapis lebih tinggi dari *I²C Engine Layer* dan dipergunakan untuk mengatur semua lalu lintas data dan sudah tersusun sesuai dengan kegunaan menjadi paket Sub-rutin.

Bagian ini tidak boleh diubah kecuali untuk keperluan khusus. Bagi pengguna yang belum mahir dan berpengalaman tidak dianjurkan untuk mengubah bagian ini.

I²C Application Layer adalah lapisan terluar yang dipergunakan untuk berinteraksi secara langsung dengan user / pengguna. Bagian ini tidak boleh diubah kecuali untuk keperluan khusus. Bagi pengguna yang belum mahir dan berpengalaman tidak dianjurkan untuk mengubah bagian ini.

Protokol dari SPC DC MOTOR dapat dilihat pada lampiran.

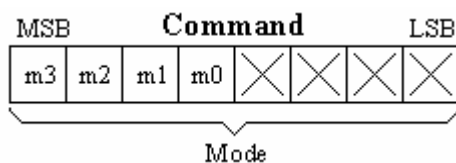
1.2.1 Pengalamatan



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

Semua penggunaan dari I²C-bus selalu diawali dengan pengalamatan. Pada pengalamatan itu sendiri dibedakan menjadi tiga bagian : alamat tetap, alamat terprogram, dan Read/Write (R/W). SPC DC MOTOR selalu menggunakan alamat tetap dengan nilai “1110”, sedangkan untuk alamat terprogram digunakan untuk memberikan alamat terhadap modul sesuai dengan kehendak pemakai. Alamat terprogram diatur dengan cara mengganti setting jumper (dapat dilihat pada **bagian 2.2**) sehingga pada jalur I²C yang sama dengan alamat tetap yang sama (“1110”) dapat digunakan 8 buah modul secara bersamaan dengan membedakan alamat terprogram. Bagian Read/Write (R/W) bernilai “1” jika Master I²C (DT-51 MinSys / mikrokontroler lain) akan membaca data dari Slave I²C (SPC DC Motor) dan bernilai “0” jika DT-51 MinSys / mikrokontroler lain akan menulis data ke SPC DC Motor.

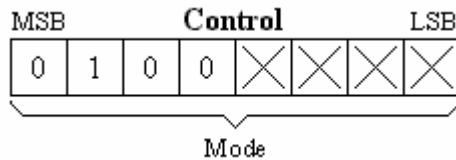
1.2.2 Command



m3	m2	m1	m0	x	x	x	x	Mode
0	0	0	0	X	X	X	X	Tidak terpakai
0	0	0	1	X	X	X	X	Command GateTime
0	0	1	0	X	X	X	X	Command PWM1
0	0	1	1	X	X	X	X	Command PWM2
0	1	0	0	X	X	X	X	Command Control
0	1	0	1	X	X	X	X	Command Input
0	1	1	0	X	X	X	X	Tidak terpakai
.
1	1	1	1	X	X	X	X	Tidak terpakai

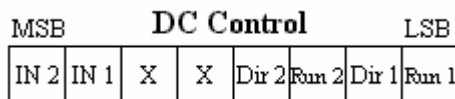
Pada perintah command terdapat bagian utama yaitu Mode. Mode digunakan untuk memilih perintah selanjutnya yang akan diberikan pada device sesuai dengan pilihan mode yang diberikan. Pada command memiliki 16 kemungkinan mode, namun pada SPC DC MOTOR ini hanya digunakan 5 mode saja.

1.2.2.1 Command Control



Command Control memanfaatkan register : *DCCControl*

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



Control	Setting (H/ L)	Fungsi
Run 1	Stop/ Run	Untuk menjalankan dan menghentikan motor DC 1 : Stop beri logika '1' (high) Run beri logika '0' (low)
Dir 1	CW / CCW	Untuk arah putaran motor DC 1 : CW (searah jarum jam) beri logika '1' (high) CCW (berlawanan arah jarum jam) beri logika '0' (low)
Run 2	Stop / Run	Untuk menjalankan dan menghentikan motor DC 2 : Stop beri logika '1' (high) Run beri logika '0' (low)
Dir 2	CW / CCW	Untuk arah putaran motor DC 2 : CW (searah jarum jam) beri logika '1' (high) CCW (berlawanan arah jarum jam) beri logika '0' (low)
In 1	On / Off	Untuk mengaktifkan dan menon-aktifkan input motor DC 1 Mengaktifkan beri logika '1' (high) Menon-aktifkan beri logika '0' (low)
In 2	On / Off	Untuk mengaktifkan dan menon-aktifkan input motor DC 2 Mengaktifkan beri logika '1' (high) Menon-aktifkan beri logika '0' (low)

Pengiriman Command Control diikuti dengan pengiriman DCCControl. Namun dalam Application Layer, user hanya perlu mengisi DCCControl. Command Control akan ditambahkan secara otomatis. DCCControl digunakan untuk mengatur semua kegiatan dari motor DC.

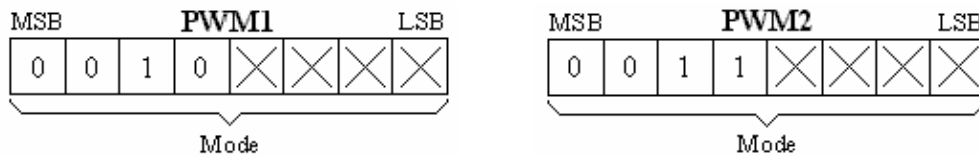
- Jika salah satu dari **IN 1** atau **IN 2** aktif, maka SPC DC MOTOR akan menghitung pulsa kecepatan motor DC pada input yang aktif. Perhitungan ini akan diperbarui setiap periode (ditentukan oleh command **GateTime**) sampai input dinon-aktifkan.
- Jika **IN 1** dan **IN 2** aktif secara bersamaan, maka SPC DC MOTOR akan menghitung pulsa kecepatan motor DC 1 dan pulsa kecepatan motor DC 2 secara bergantian.

Contoh Aplikasi :

Bila ingin menjalankan motor DC 1 dengan arah searah jarum jam dan ingin mengetahui kecepatan dari motor DC 1 maka register DCCControl dapat diisi dengan nilai '01000110b atau setara dengan '46h'.

Bila ingin membuat motor DC berhenti maka Control Run 1 harus dibuat 'high', yaitu: '01000111b'.

1.2.2.2 Command PWM



Command PWM memanfaatkan register : PWM1

Memanfaatkan alamat memory 3Ah atau dengan nama lain *BufferOut2* untuk PWM1

Command PWM memanfaatkan register : PWM2

Memanfaatkan alamat memory 3Bh atau dengan nama lain *BufferOut3* untuk PWM2

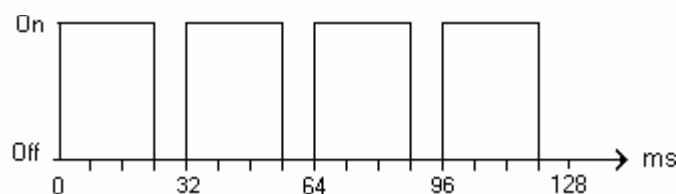
Pengiriman Command PWM diikuti dengan pengiriman PWM1 dan/atau PWM2. Namun dalam Application Layer, user hanya perlu mengisi PWM1 dan/atau PWM2. Command PWM akan ditambahkan secara otomatis. Nilai PWM1 dan PWM2 digunakan untuk mengatur kecepatan putaran motor DC, dengan cara menghidupkan dan mematikan motor DC secara bergantian dalam satu periode (32 ms) secara terus menerus. PWM hanya akan berfungsi jika motor DC dalam keadaan 'Run'.

Nilai dari PWM ini dapat diatur mulai dari 0 sampai 255 (FFh). Berikut ini adalah rumus perhitungan PWM dalam satu periode :

Ton	= 32 ms	untuk	PWM = 0
Toff	= 0 ms		
Ton	= (255 – PWM) * 0.125 ms	untuk	1 < PWM ≤ 255
Toff	= 32 ms – Ton		

Sebagai contoh, jika nilai PWM diset pada posisi 63d (3Fh), maka motor DC secara periodik berada pada posisi "On" selama 24 ms, dan pada posisi "Off" selama 8 ms.

Berikut ini adalah timing diagram dari nilai PWM 63d (3Fh).



Ada 2 buah register yang digunakan untuk mengatur setting PWM. Register

PWM1 digunakan untuk mengatur setting PWM motor DC 1 dan register **PWM2** digunakan untuk mengatur setting PWM motor DC 2.

Contoh Aplikasi :

Bila dikehendaki motor DC 1 berjalan dengan PWM 75% dan motor DC 2 berjalan dengan PWM 30%, maka perhitungannya sebagai berikut :

❖ Motor DC 1

$$\begin{aligned} T_{on} &= 75\% * 32 \text{ ms} \\ &= 24 \text{ ms} \end{aligned}$$

Dari rumus di atas maka diperoleh

$$24 \text{ ms} = (255 - PWM_1) * 0.125 \text{ ms}$$

isi register PWM1 = '**63d**'

❖ Motor DC 2

$$\begin{aligned} T_{on} &= 30\% * 32 \text{ ms} \\ &= 9,6 \text{ ms} \end{aligned}$$

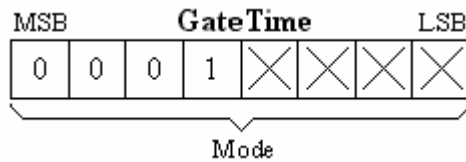
Dari rumus di atas maka diperoleh

$$9,6 \text{ ms} = (255 - PWM_2) * 0.125 \text{ ms}$$

$$PWM_2 = 178,2$$

Dibulatkan ke bilangan desimal terdekat maka isi register PWM2 = '**178d**'

1.2.2.3 Command GateTime



Command GateTime memanfaatkan register : *GateTime*

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

GateTime	Time	Resolusi
80 H	2000 ms	0.5 Hz
40 H	1000 ms	1 Hz
20 H	500 ms	2 Hz
10 H	250 ms	4 Hz
08 H	125 ms	8 Hz
04 H	62.5 ms	16 Hz
02 H	31,25 ms	32 Hz
01 H	15,625 ms	64 Hz

Pengiriman Command GateTime diikuti dengan pengiriman GateTime. Namun dalam Application Layer, user hanya perlu mengisi GateTime. Command GateTime akan ditambahkan secara otomatis. GateTime digunakan untuk mengatur besarnya waktu yang dibutuhkan untuk menghitung banyaknya pulsa kecepatan motor DC setiap periode. Ada delapan nilai

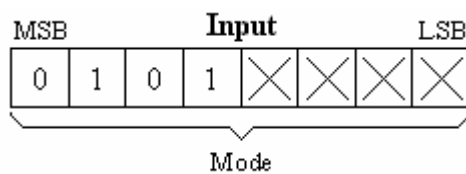
GateTime yang dapat digunakan, seperti yang terlihat pada tabel di atas. Semakin besar nilai GateTime, perhitungan pulsa kecepatan motor DC akan semakin akurat, namun waktu yang dibutuhkan untuk menghitung kecepatan dalam satu periode lebih lama.

Default GateTime dari SPC DC MOTOR ini adalah '08h'.

Contoh :

Jika GateTime diberi nilai '20h', maka waktu yang dibutuhkan untuk menghitung pulsa kecepatan motor DC adalah 500 ms, dan kesalahan perhitungannya adalah ± 1 Hz.

1.2.2.4 Command Input



Command Input memanfaatkan register : **InputH1, InputL1, InputH2, InputL2.**

Memanfaatkan alamat memory 30h atau dengan nama lain *BufferIn0* untuk InputH1

Memanfaatkan alamat memory 31h atau dengan nama lain *BufferIn1* untuk InputL1

Memanfaatkan alamat memory 32h atau dengan nama lain *BufferIn2* untuk InputH2

Memanfaatkan alamat memory 33h atau dengan nama lain *BufferIn3* untuk InputL2

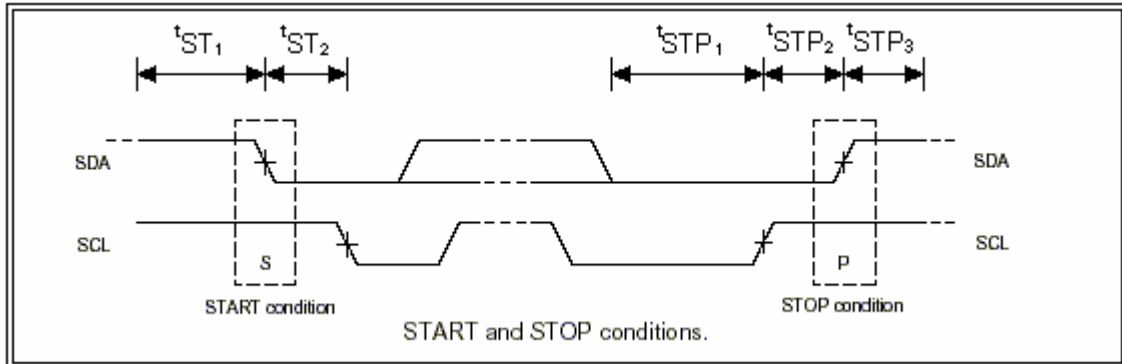
Pengiriman Command Input diikuti dengan pembacaan InputH1 dan InputL1 dan/atau InputH2 dan InputL2. Namun dalam Application Layer, user hanya perlu membaca InputH1 dan InputL1 dan/atau InputH2 dan InputL2. Command Input akan ditambahkan secara otomatis. InputH1, InputL1, InputH2, dan InputL2 digunakan untuk menyimpan hasil perhitungan pulsa kecepatan putaran motor DC dalam satu detik. Command Input hanya dapat digunakan, jika motor DC dilengkapi dengan data input kecepatan putaran motor yang berupa pulsa TTL. Perhitungan dilakukan saat terjadi transisi dari high menjadi low dari data input kecepatan. Semakin cepat putaran motor, maka pulsa yang dihasilkan akan semakin tinggi frekuensinya. Sebuah motor DC dalam satu putaran bisa menghasilkan lebih dari satu pulsa tergantung dari spesifikasi motor DC tersebut.

Pada SPC DC MOTOR ini, pulsa maksimum yang dapat dihasilkan dalam satu detik adalah 65.535 (16 bit), yang disimpan didalam dua register yaitu **InputH1** (bit 8–15) dan **InputL1** (bit 0–7) untuk motor DC 1 dan **InputH2** (bit 8–15) dan **InputL2** (bit 0-7) untuk motor DC 2.

1.3 SPESIFIKASI I²C BUS

Berikut akan dijelaskan mengenai cara kerja dari komunikasi I²C Master (DT-51 Minimum System) – Slave (SPC DC MOTOR). Tugas dari master adalah mengontrol semua komunikasi yang dilakukan, seperti mengatur pulsa clock pada jalur SCL, bit data pada SDA, dan memeriksa atau mengirimkan acknowledge.

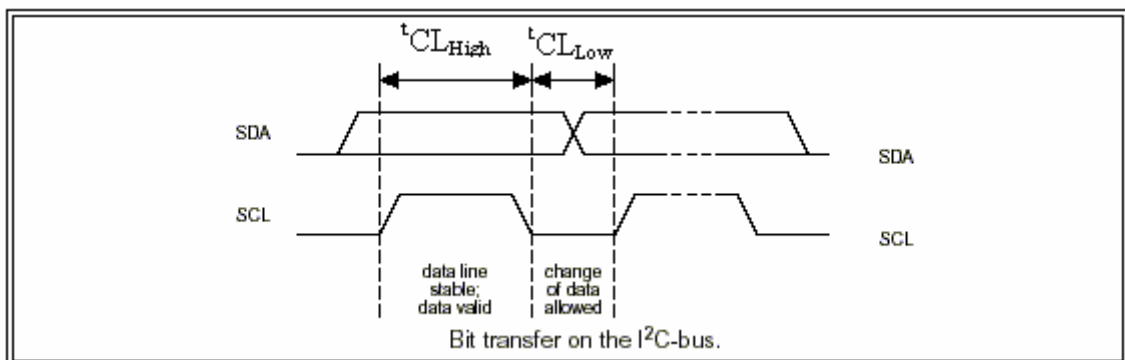
1.3.1. Kondisi Start dan Stop



Kondisi Start dalam I²C selalu dilakukan dengan cara memberikan level High ("1") pada jalur SCL kemudian pada jalur SDA terjadi transisi turun (High ke Low).

Kondisi Stop dalam I²C selalu dilakukan dengan cara memberikan level High ("1") pada jalur SCL kemudian pada jalur SDA terjadi transisi naik (Low ke High).

1.3.2. Transfer Data



Transfer data dapat dilakukan setelah memberikan kondisi **start** dan perubahan bit data (SDA) hanya boleh terjadi / diijinkan saat jalur SCL pada kondisi Low ("0"). Transfer data ini biasanya sebesar 8 bit yang dilakukan secara streaming (Bit Streaming), kemudian dilengkapi dengan pengiriman bit acknowledge.

Mode Write

Saat master I²C berada pada kondisi **Write** (bit $\overline{R/\overline{W}}$ pada **AddressI2C** =

“0”), maka master I²C akan mengirimkan 8 bit data ke slave I²C, kemudian master akan menunggu adanya bit acknowledge dari slave, apabila tidak diterima bit acknowledge maka slave tidak menerima data (bit streaming) secara lengkap (miss/hilang) atau terjadi salah alamat (alamat tetap atau terprogram).

Mode Read

Saat master I²C berada pada kondisi **Read** (bit R/W pada **AddressI²C** = “1”), maka master I²C akan menerima 8 bit data dari slave I²C yang alamatnya sesuai dengan alamat master I²C, kemudian master I²C akan mengirimkan bit acknowledge untuk memberitahukan kepada slave apakah ingin mengambil data berikutnya atau tidak.

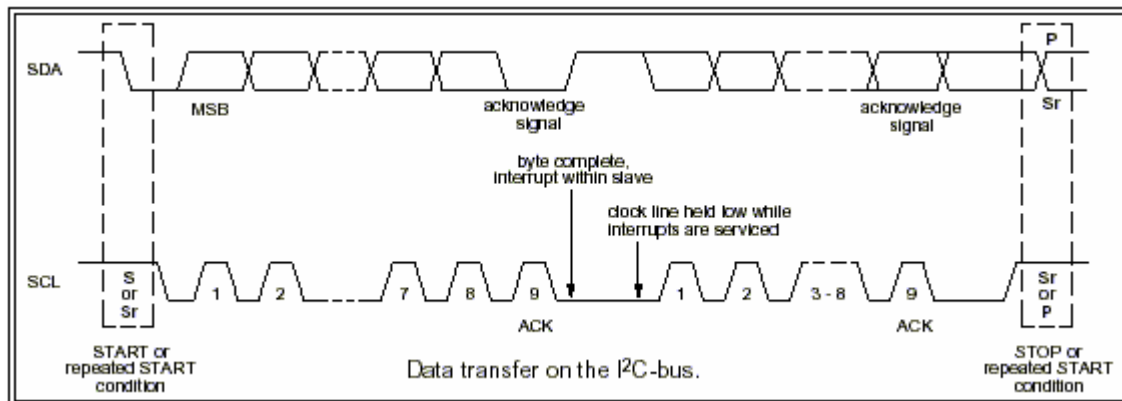


Diagram di atas adalah diagram lengkap tentang transfer data melalui I²C.

Symbol	Parameter	Min	Units
^t ST ₁	Time before START action	15	μs
^t ST ₂	Time after START action	7	μs
^t STP ₁	Hold time for prepare STOP action	14	μs
^t STP ₂	Time before STOP action	7	μs
^t STP ₃	Time after STOP action	7	μs
^t CL _{High}	Time High for data hold	10	μs
^t CL _{Low}	Time Low for change data	12	μs

Untuk menghubungkan SPC DC MOTOR dengan DT-51 Minimum System dianjurkan untuk menggunakan kabel pita (flat ribbon cable). Hubungannya ditunjukkan pada tabel berikut :

I ² C Bus	DT-51 Minimum System PORT C & PORT 1	SPC DC MOTOR J7
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 J1 (Supply). Perhatikan polaritasnya jangan sampai terbalik, karena dapat mengakibatkan kerusakan.

Penting !

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

2.2 SETTING JUMPER

Alamat terprogram setiap board SPC DC MOTOR ditentukan oleh setting jumper J4.

J4 (A2)	J4 (A1)	J4(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 J3 (Pull up SCL/SDA) digunakan untuk resistor pull up SDA (I²C bus data input / output) dan SCL (I²C bus clock input). Apabila lebih dari satu board SPC DC MOTOR dihubungkan pada I²C bus maka hanya perlu memasang jumper J3 pada salah satu board saja.

2.3 EKSPANSI SPC DC MOTOR

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

- Setiap board harus mempunyai alamat terprogram yang berbeda, ditentukan oleh jumper J4 (A0/A1/A2).
- Jumper J3 pada salah satu board saja yang dipasang.

2.4 PENGGUNAAN SPC DC MOTOR SECARA PARALEL

SPC DC MOTOR dapat digunakan secara paralel dengan cara mengatur pin-pin S1, S2, S3, dan S4 yang ada pada board SPC DC MOTOR.

Berikut adalah tabel kegunaan dari pin-pin tersebut:

Pin	Name	Setting	Fungsi
S1	Run 1	Stop / Run	Untuk menjalankan atau mematikan motor DC 1 Stop beri logika '1' (high) Run beri logika '0' (low)
S2	Dir 1	CW / CCW	Untuk arah putaran motor DC 1 CW (searah jarum jam) beri logika '1' (high) CCW (berlawanan arah jarum jam) beri logika '0' (low)
S3	Run 2	Stop / Run	Untuk menjalankan atau mematikan motor DC 2 Stop beri logika '1' (high) Run beri logika '0' (low)
S4	Dir 2	CW / CCW	Untuk arah putaran motor DC 2 CW (searah jarum jam) beri logika '1' (high) CCW (berlawanan arah jarum jam) beri logika '0' (low)

- Secara default jika pin-pin S1, S2, S3, dan S4 tersebut tidak dihubungkan (Floating/mengambang) maka akan selalu berlogika "high".
- Untuk dapat menjalankan SPC DC MOTOR secara paralel, setting kedua motor DC pada register **DCControl** harus dalam keadaan Stop.
- Untuk pengaturan PWM secara paralel, dapat dilakukan dengan cara memberi pulsa secara periodik pada pin S1 atau S3 dengan frekuensi maksimal 10 KHz.

Contoh Aplikasi :

Bila diinginkan motor DC 1 dijalankan secara paralel dengan PWM 50% dan putaran searah jarum jam, maka pin S2 diberi logika '1' dan pin S1 diberi sinyal kotak dengan periode high dan periode low yang sama besarnya.

2.5 PENYAMBUNGAN SPC DC MOTOR DENGAN MOTOR DC

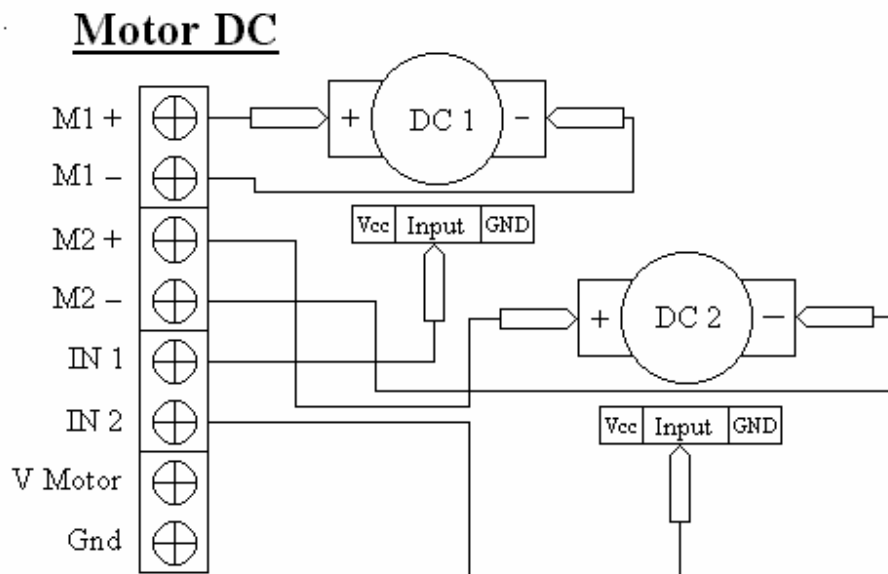
Dalam penyambungan motor DC dengan modul SPC DC MOTOR perlu diperhatikan tipe dari motor DC yang akan dipergunakan.

Modul SPC DC MOTOR dapat dipergunakan untuk dua buah motor DC yang mempunyai tegangan kerja yang sama. Modul SPC DC MOTOR ini dapat digunakan baik untuk motor DC yang mempunyai data input kecepatan maupun yang tidak mempunyai data input kecepatan. Untuk motor DC yang tidak mempunyai data input kecepatan, prosedur Input dan Brake yang terdapat pada SPC DC MOTOR ini tidak dapat digunakan.

Berikut adalah cara pemasangan dari kedua buah motor DC yang dilengkapi dengan data input.

- ❑ Modul SPC DC MOTOR dapat dipergunakan untuk motor DC dengan tegangan kerja dari 5 Volt sampai dengan 36 Volt.
- ❑ Arus RMS maksimum untuk modul SPC DC MOTOR adalah 600 mA.
- ❑ Arus impuls tak berulang maksimum untuk modul SPC DC MOTOR adalah 1.2 A.
- ❑ Sudah dilengkapi dioda clamp secara internal.
- ❑ Hubungkan catu daya positif (+) untuk motor DC pada *Vmotor* dan catu daya negatif (-) pada *GND*.
Tegangannya harus sesuai dengan tegangan kerja motor.
- ❑ Untuk motor DC 1, sambungkan kutub positif motor DC pada M1+ dan

- kutub negatif motor DC pada M1- serta data input pada IN1 secara benar.
- Untuk motor DC 2, sambungkan kutub positif motor DC pada M2+ dan kutub negatif motor DC pada M2- serta data input pada IN2 secara benar. (lihat gambar).



2.6. MENCOBA SPC DC MOTOR DENGAN EXAMPLE.HEX

- ◆ Hubungkan DT-51 Minimum System dengan SPC DC MOTOR (lihat bagian 2.1)
 - ◆ Hubungkan SPC DC MOTOR dengan motor DC (lihat bagian 2.5)
 - ◆ Setting alamat SPC DC MOTOR pada alamat terprogram ke-7 (default)
 - ◆ Download EXAMPLE.HEX yang terdapat pada disket/CD
 - ◆ Motor DC 1 akan bergerak secara Clockwise (CW) dengan PWM 100 % dan input 1 aktif sedangkan motor DC 2 akan bergerak secara Counter Clockwise (CCW) dengan PWM 50 %.
- Setelah 5 detik, motor DC 1 dihentikan dengan menggunakan prosedur Brake sedangkan motor DC 2 dihentikan secara manual (tanpa prosedur Brake). Demikian seterusnya.

3. PERANGKAT LUNAK SPC DC MOTOR

3.1 DRIVER dan RUTIN

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

- Internal RAM alamat 21h bit 0 dan 1
- Internal RAM dengan alamat 2Fh – 3Fh, dan 40h - 43h
- 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 13 buah register yang terdiri dari:

AddressI2C	GateTime	PWM1	PWM2	DCCControl
InputH1	InputL1	InputH2	InputL2	
BrakeH1	BrakeL1	BrakeH2	BrakeL2	

Kegunaan dari register-register tersebut dapat dilihat pada **bagian 1.2**.

Dari register tersebut akan digunakan dalam 9 rutin penting berikut :

DCInit

Fungsi : Untuk menginisialisasi SPC DC Motor.

Input : AddressI2C, GateTime, PWM1, PWM2 dan DCCControl

Output : Flag FAck

Keterangan :

- ❖ Rutin ini digunakan untuk memberikan nilai awal atau inisialisasi tanpa menjalankan motor DC, yaitu dengan memberi logika high '1' pada setting Run dari DCCControl.
- ❖ Rutin ini dapat juga digunakan untuk menjalankan rutin SetGateTime, SetPWM1, SetPWM2 dan SetControl dalam satu buah rutin.
- ❖ Jika **Flag FAck** bernilai '1' maka SPC DC MOTOR siap untuk digunakan.

Metode : Isi register AddressI2C, GateTime, PWM1, PWM2 dan DCCControl sesuai dengan kebutuhan kemudian panggil rutin DCInit.

SetGateTime

Fungsi : Mengatur besarnya waktu yang dibutuhkan untuk menghitung pulsa input kecepatan motor DC dalam satu periode.

Input : AddressI2C dan GateTime

Output : Flag FAck

Keterangan : Tabel pengaturan nilai register GateTime ini dapat dilihat pada **bagian 1.2.4**.

Metode : Isi register AddressI2C dan GateTime sesuai dengan kebutuhan kemudian panggil rutin SetGateTime.

SetPWM1

Fungsi : Mengatur kecepatan putaran motor DC 1

Input : AddressI2C dan PWM1

Output : Flag FAck

Keterangan : Perhitungan kecepatan putaran motor DC 1 dapat dilihat pada **bagian 1.2.3**.

Metode : Isi register AddressI2C dan PWM1 sesuai dengan kebutuhan kemudian panggil rutin SetPWM1.

SetPWM2

Fungsi : Mengatur kecepatan putaran motor DC 2

Input : AddressI2C dan PWM2

Output : Flag FAck

Keterangan : Perhitungan kecepatan putaran motor DC 2 dapat dilihat pada **bagian 1.2.3**.

Metode : Isi register AddressI2C dan PWM2 sesuai dengan kebutuhan kemudian panggil rutin SetPWM2.

SetControl

Fungsi : Untuk menjalankan atau menghentikan motor DC, mengubah arah putaran motor DC, dan mengaktifkan perhitungan pulsa input kecepatan pada motor DC.

Input : AddressI2C dan DCControl

Output : Flag FAck

Keterangan :

- ❖ Rutin ini digunakan untuk mengatur semua aktivitas dari SPC DC MOTOR.
- ❖ Tabel dari nilai register DCControl ini dapat dilihat pada **bagian 1.2.2**

Metode : Isi register AddressI2C dan DCControl sesuai dengan kebutuhan kemudian panggil rutin SetControl.

GetInput1

Fungsi : Menyimpan hasil perhitungan pulsa input kecepatan motor DC 1.

Input : AddressI2C

Output : InputH1 dan InputL1

Keterangan : Hasil dari rutin ini disimpan pada register **InputH1** untuk bit 8-15 dan register **InputL1** untuk bit 0-7.

Metode : Isi register AddressI2C sesuai dengan alamat kemudian panggil rutin GetInput1.

GetInput2

Fungsi : Menyimpan hasil perhitungan pulsa input kecepatan motor DC 2.

Input : AddressI2C

Output : InputH2 dan InputL2

Keterangan : Hasil dari rutin ini disimpan pada register **InputH2** untuk bit 8-15 dan register **InputL2** untuk bit 0-7.

Metode : Isi register AddressI2C sesuai dengan alamat kemudian panggil rutin GetInput2.

Brake1

Fungsi : Untuk menghentikan motor DC 1 secara cepat.

Input : AddressI2C, BrakeH1 dan BrakeL1

Output : InputH1, InputL1

Keterangan :

- ❖ Fungsi ini hanya bisa digunakan, jika motor DC 1 dilengkapi dengan data input kecepatan putaran motor.
- ❖ Motor DC 1 akan berhenti jika pulsa input kecepatan lebih rendah dari input brake.
- ❖ Nilai pulsa input kecepatan terakhir sebelum motor DC 1 berhenti disimpan pada register InputH1 dan InputL1.

Metode : Isi register AddressI2C, **BrakeH1** untuk bit 8-15 dan **BrakeL1** untuk bit 0-7 sesuai dengan kebutuhan kemudian panggil rutin Brake1.

Brake2

Fungsi : Untuk menghentikan motor DC 2 secara cepat.

Input : AddressI2C, BrakeH2 dan BrakeL2

Output : InputH2, InputL2

Keterangan :

- ❖ Fungsi ini hanya bisa digunakan, jika motor DC 2 dilengkapi dengan data input kecepatan putaran motor.
- ❖ Motor DC 2 akan berhenti jika pulsa input kecepatan lebih rendah dari input brake.
- ❖ Nilai pulsa input kecepatan terakhir sebelum motor DC 2 berhenti disimpan pada register InputH2 dan InputL2.

Metode : Isi register AddressI2C, **BrakeH2** untuk bit 8-15 dan **BrakeL2** untuk bit 0-7 sesuai dengan kebutuhan kemudian panggil rutin Brake2.

3.2 CONTOH APLIKASI DAN PROGRAM

Bila dikehendaki modul SPC DC MOTOR dengan alamat terprogram ke-5 menjalankan motor DC 1 yang mempunyai input kecepatan dengan arah searah jarum jam (CW) dengan setting PWM 75% dan menjalankan motor DC 2 yang tidak mempunyai input kecepatan dengan arah berlawanan jarum jam (CCW) dengan PWM 50%.

Setelah 5 detik, arah motor DC 1 berubah menjadi berlawanan jarum jam (CCW) dengan PWM 25% dan arah motor DC 2 menjadi searah jarum jam (CW) dengan PWM 100%.

Lima detik kemudian motor DC 2 berhenti, dan 5 detik kemudian motor DC 1 juga berhenti.

Listing program untuk kasus diatas:

```
$MOD51
CSEG
ORG 4000H
LJMP Start

ORG 4100H
$INCLUDE(ENG_I2C.INC) ;Driver untuk semua produk
;SPC I2C (HARUS DITULISKAN
;TERLEBIH DAHULU SEBELUM
;DCMOTOR.INC)
$INCLUDE(DCMOTOR.INC) ;Driver SPC DC MOTOR

Delay5s:
MOV R5, #28H
D1: MOV R6, #0FFH
D2: MOV R7, #0FFH
DJNZ R7, $
DJNZ R6, D2
DJNZ R5, D1
RET

Start:
MOV SP, #50H
```

```

MOV   AddressI2C,#11101010B   ;memasukkan alamat i2c
MOV   DCControl,#01000010B   ;memasukkan nilai DCControl
MOV   PWM1,#03FH             ;memasukkan PWM 1 = 75%
MOV   PWM2,#07FH             ;memasukkan PWM 2 = 50%
ACALL DCInit                 ;memanggil rutin DCInit
ACALL Delay5s                ;Delay 5 detik

MOV   PWM1,#0BFH             ;memasukkan PWM 1 = 25%
ACALL SetPWM1                ;memanggil rutin SetPWM1
MOV   PWM2,#00H              ;memasukkan PWM 2 = 100%
ACALL SetPWM2                ;memanggil rutin SetPWM2
MOV   DCControl,#01001000B   ;memasukkan nilai DCControl
ACALL SetControl             ;memanggil rutin SetControl
ACALL Delay5s                ;Delay 5 detik

MOV   DCControl,#10001100B   ;memasukkan nilai DCControl
ACALL SetControl             ;memanggil rutin SetControl
ACALL Delay5s                ;Delay 5 detik

ACALL GetInput1              ;memanggil rutin GetInput1
MOV   BrakeH1,#20H           ;\memasukkan input Brake 1
MOV   BrakeL1,#00H           ;/
ACALL Brake1                 ;memanggil rutin Brake1
END

```

3.3 KERANGKA PROGRAM

Bagi user yang ingin membuat program aplikasi SPC DC MOTOR dengan menggunakan rutin yang sudah ada maka 2 driver berikut harus dimasukkan (include) : *ENG_I2C.INC* dan *DCMOTOR.INC*

ENG_I2C.INC merupakan driver yang akan selalu digunakan untuk setiap aplikasi Smart Peripheral Controller (SPC) yang menggunakan I²C-Bus.

DCMOTOR.INC merupakan driver yang khusus digunakan untuk SPC DC MOTOR.

Kerangka pemrograman SPC DC MOTOR menggunakan Assembler MetaLink ASM51[©] sebagai berikut :

```

;-----
;FILE TEMPLATE UNTUK SPC I2C BUS
;DENGAN DT-51 MINSYS
;-----
$MOD51

      CSEG
      ORG   4000H
      LJMP  START

      ORG   4100H
      $INCLUDE (ENG_I2C.INC)           ;DRIVER UNTUK SEMUA PRODUK
                                       ;I2C BUS (HARUS DITULISKAN
                                       ;TERLEBIH DAHULU SEBELUM
                                       ;DCMOTOR.INC)
      $INCLUDE (DCMOTOR.INC)         ;DRIVER SPC DC MOTOR

START:
      .
      .
      .
      END

```

- ◆ *Terima Kasih atas kepercayaan Anda menggunakan produk kami, bila ada kesulitan, pertanyaan atau saran mengenai produk ini silahkan menghubungi technical support kami :*
support@innovativeelectronics.com
-

LAMPIRAN

