

PERCOBAAN 5

Tujuan :

1. Mahasiswa memahami Penggunaan SPC Motor Stepper Mikrokontroler AT89S51 secara paralel maupun I2C-BUS.

Teori Dasar :

SPC MOTOR STEPPER

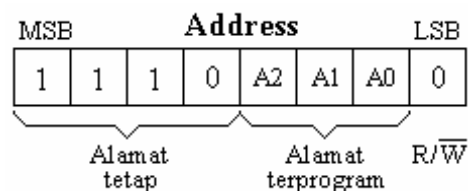
Smart Peripheral Controller / SPC STEPPER MOTOR merupakan pengontrol motor stepper yang menggunakan I²C-bus sebagai jalur pengiriman data sehingga dapat lebih menghemat dan mempermudah pengkabelan, selain itu SPC STEPPER MOTOR dapat digunakan secara paralel. Contoh aplikasi dari SPC STEPPER MOTOR adalah untuk robot dan sumber gerak lainnya.

1. SPESIFIKASI SPC STEPPER MOTOR

Spesifikasi SPC STEPPER MOTOR sebagai berikut :

- Komplitabel penuh dengan DT-51 Minimum System Ver 3.0.
- (DT51 adalah alat pengembangan mikrokontroler keluarga MCS51 yang sederhana, handal, dan ekonomis. DT51 berbentuk sistem minimum dengan komponen utamanya mikrokontroler 89C51.)
- Hanya perlu 2 jalur kabel untuk interface dengan mikroprosesor / mikrokontroler lain.
- Dapat digunakan untuk unipolar atau bipolar stepper motor.
- Dapat digunakan pada I2C-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.

2. Pengalamatan

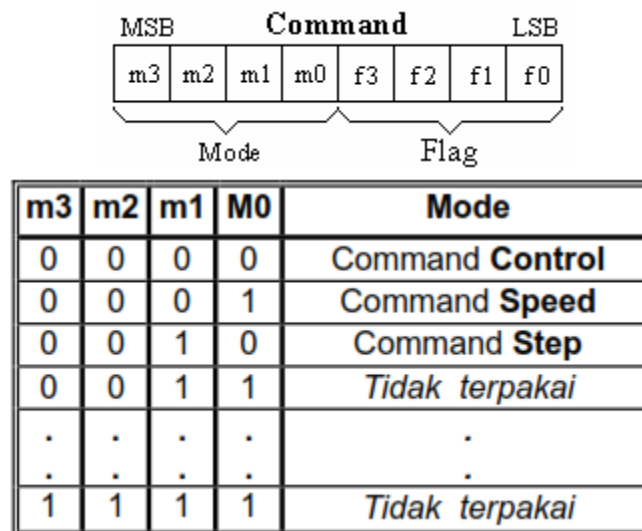


Pengalamatan memanfaatkan register : **AddressI2C** dengan alamat memory 2Fh

Semua penggunaan dari I2C-bus selalu diawali dengan pengalamatan. Pada pengalamatan itu sendiri dibedakan menjadi tiga bagian : alamat tetap, alamat

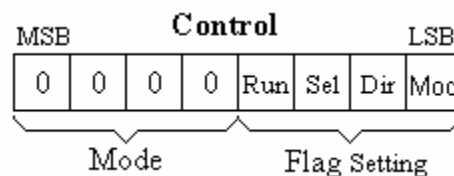
terprogram, dan Read/Write (R/W). SPC STEPPER MOTOR selalu menggunakan alamat tetap dengan nilai “1110”, dan Read/Write selalu bernilai Write (“0”) sedangkan alamat terprogram digunakan untuk memberikan alamat terhadap modul sesuai dengan kehendak pemakai. Alamat terprogram diatur dengan cara mengubah setting jumper (dapat dilihat pada **bagian 6.2**) sehingga pada jalur I2C yang sama dengan alamat tetap yang sama (“1110”) dapat digunakan 8 buah modul secara bersamaan dengan membedakan alamat terprogram.

3. Command



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. Command Control



Control	Setting (H/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.

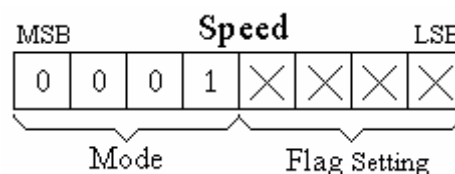
- Yang dimaksud dengan Full adalah suatu mode motor stepper dimana per langkahnya merupakan pergeseran maksimum antara rotor terhadap stator sehingga mode Full merupakan mode tercepat untuk melakukan pergeseran/perputaran namun memiliki kekurangan dari segi torsi yang dihasilkan .
- Yang dimaksud dengan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

Command Speed

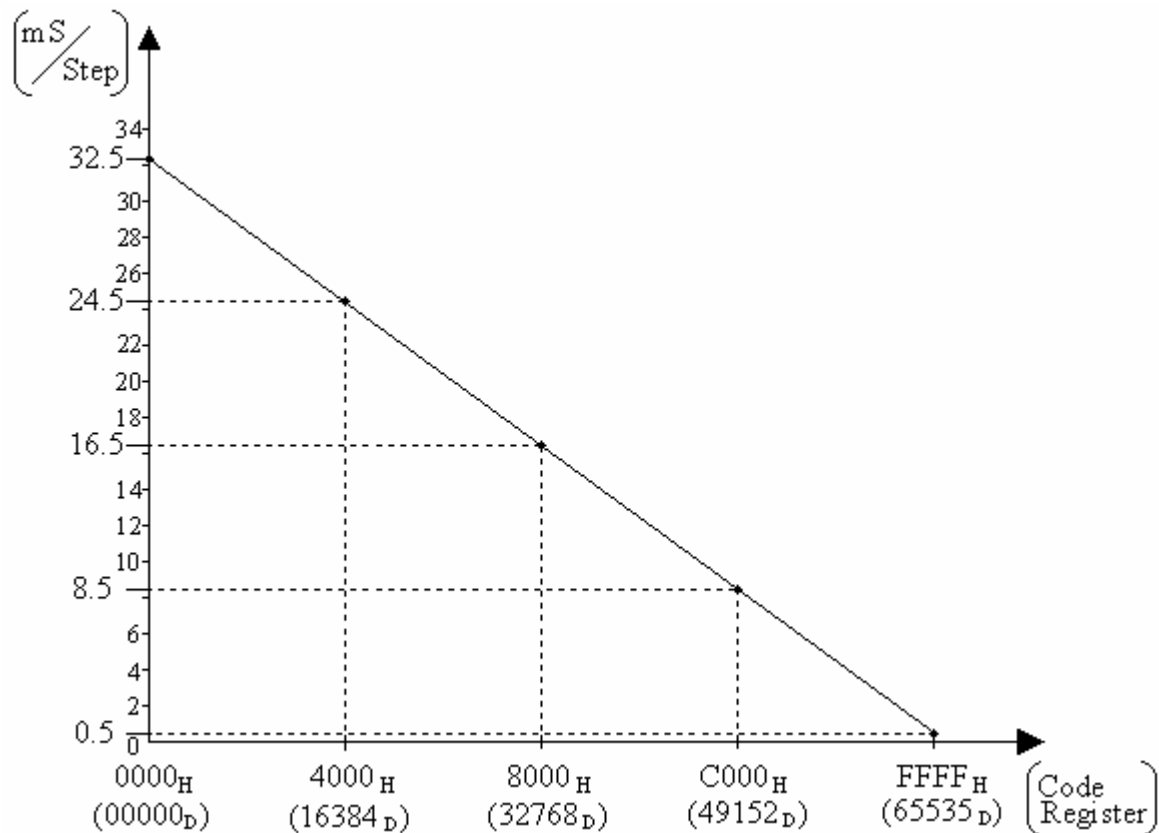


Command Speed memanfaatkan register : **StepSpeedH** dan **StepSpeedL**

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

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

Contoh aplikasi :

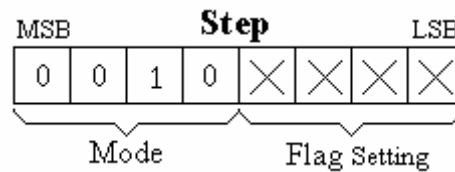
Bila dikehendaki motor stepper bergerak dengan waktu 10mS per step maka didapatkan:

Code = $-2048(10-32.5) = 46080$ (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.

5. 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

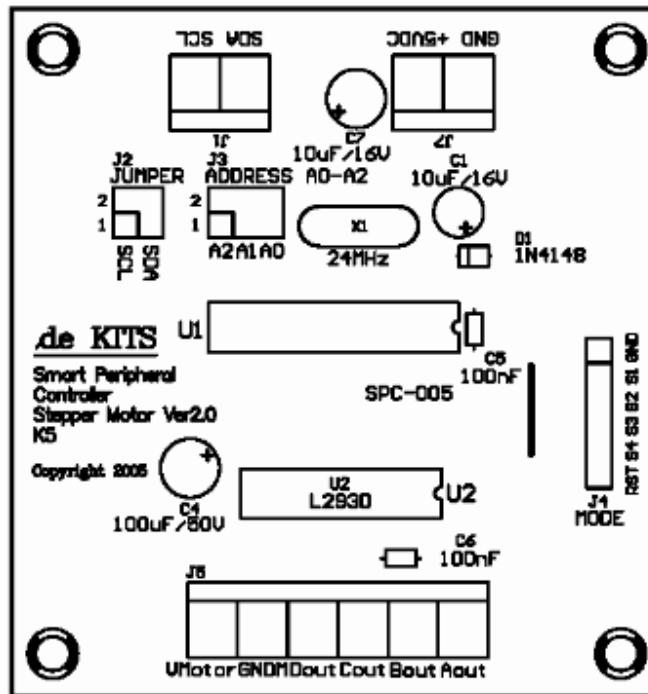
Command Step merupakan command ketiga dan dalam penggunaannya harus dikirimkan Command Step “00100000” terlebih dahulu kemudian diikuti oleh StepH dan StepL. Namun user hanya perlu mengisi StepH, StepL saja dan secara otomatis dalam pengiriman akan ditambahkan Command Speed “00100000”. Command step digunakan untuk memberikan besaran langkah yang harus ditempuh oleh motor stepper dengan mengirimkan 2 byte kode langkah. Dari 0000h untuk berhenti (tidak ada langkah) sampai FFFEh 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.

6. TATA LETAK KOMPONEN SPC STEPPER MOTOR



7. 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-DOSTM atau PC-DOSTM
- Assembler MetaLink ASM51
- File-file yang ada pada pada disket/CD program.

8. 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 ² C Bus	DT-51 Minimum System PORT C & PORT 1	SPC STEPPER MOTOR J1
SCL	Pin 15 (Port 1.6)	SCL
SDA	Pin 16 (Port 1.7)	SDA

Catu daya 5V DC dihubungkan dengan konektor J7 (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.

9. SETTING JUMPER

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

J3 (A2)	J3 (A1)	J3(A0)	Alamat Terprogram	
■	■	■	0	000
■	■		1	001
■		■	2	010
■			3	011
	■	■	4	100
	■		5	101
		■	6	110
			7 (default)	111

Keterangan :

: jumper tersambung (ON) ■

Jumper J2 (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 STEPPER MOTOR dihubungkan pada I2C bus maka jumper J2 (SCL/SDA) salah satu board saja yang perlu dipasang.

10. 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.

11. 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 table kegunaan dari pin – pin tersebut:

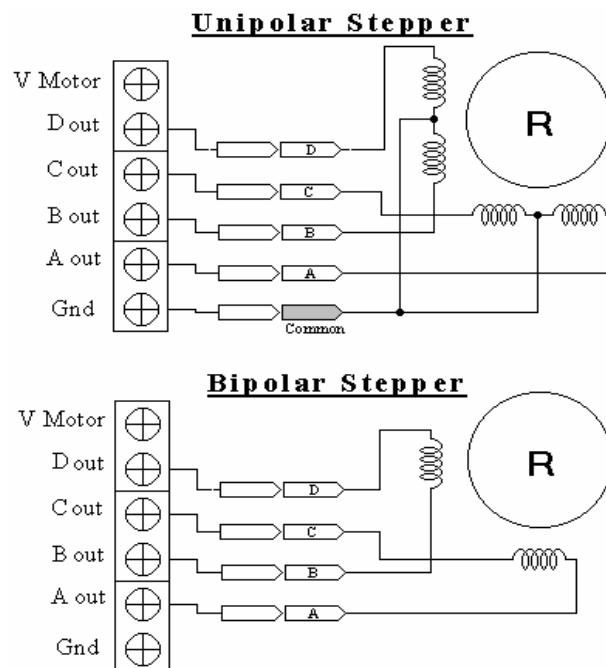
Pin	Name	Setting	Fungsi
S1	Sel	Bi / $\overline{\text{Uni}}$	Untuk tipe motor stepper yang digunakan : <i>Motor stepper Bipolar beri logika '1' (high)</i> <i>Motor stepper Unipolar beri logika '0' (low)</i>
S2	Dir	CW / $\overline{\text{CCW}}$	Untuk arah putaran motor stepper : <i>CW (searah jarum jam) 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 ‘I2C’ dan ‘Paralel’ maka yang menjadi prioritas adalah I2C, setelah perintah I2C 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 relative 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 $(800 \text{ langkah} / 200\text{Hz})$ 4 detik kemudian diberikan pulsa reset.



[illegible]

Alat-alat :


1. 1 Set IBM PC dengan Sistem Operasi Windows,
2. Software M-IDE51,
3. ISP Flash Programmer,
4. 1 Set Mikrokontroler Atmel 89C51,
5. 1 Set Downloader K-51.

Bagian 1

Modul 5.1 A Penggunaan SPC Stepper Motor Secara Paralel.

1. Bukalah **M-IDE 51** yang ada di desktop kemudian klik **New** untuk membuat lembaran kerja baru. Ketiklah Program dan setelah itu **Save as** program tersebut dalam bentuk **.asm** (Contoh : ketik : **PARALEL.asm**)
2. Kemudian ketikkanlah program dibawah ini:

```
PROGRAM
$mod51
org 0h
    mov p2,#0ffh ;Mereset Semua Kondisi Pada S1, S2, S3, S4
    setb p2.0     ;Bipolar
    clr p2.1      ;CW: 1, CCW: 0
    clr p2.2      ;Full: 1, Hall: 0
start: setb p2.3   ;_____/_----- Memberikan Pulsa High
    acall delay
    clr p2.3      ;-----\_____ Memberikan Pulsa Low
    acall delay
    sjmp start
delay: mov r5,#10
dly2:  mov r7,#0ffh
    djnz r7,$
    djnz r6,dly2
    ret
end
```

3. Setelah selesai, klik **F9** atau klik symbol seperti ini  (Running Program/Build Current File) untuk mengetahui program tersebut terjadi **error** atau **no error**.
4. Setelah program berhasil **tanpa error**, hubungkan ISP K-51 USB ke PC. Kemudian bukalah aplikasi **ISP_PROG v1.4**.
Setelah itu ikuti langkah-langkah berikut:
 - a. Pilihlah **READ DEVICE SIGNATURE** lalu pilih **RUN** ;berfungsi untuk mengecek apakah downloader sudah terhubung ke PC atau belum terhubung ke PC?

Alat-alat :

1. 1 Set IBM PC dengan Sistem Operasi Windows,
2. Software M-IDE51,
3. ISP Flash Programmer,
4. 1 Set Mikrokontroler Atmel 89C51,
5. 1 Set Downloader K-51.

Bagian 2

Modul 5.2 A Penggunaan SPC Stepper Motor Secara Paralel.

1. Bukalah **M-IDE 51** yang ada di desktop kemudian klik **New** untuk membuat lembar kerja baru. Ketik program dan setelah itu **Save as** program tersebut dalam bentuk **.asm** (Contoh : ketik : **EXAMPLE.asm**)
2. Kemudian ketikkanlah program dibawah ini:

```

                                PROGRAM
;=====;
;   EXAMPLE for MOTOR STEPPER   ;
;-----;
;File Name:  EXAMPLE.ASM        ;
;Include File: ENG_I2C.INC      ;
;           STEPPER.INC        ;
;Date   :    06/11/02          ;
;=====;
$MOD51           ;Deklarasi awal program
    CSEG
    ORG    0H
    AJMP   Start
    ORG    100H
    $INCLUDE(ENG_I2C.INC)
    $INCLUDE(STEPPER.INC)
Start:
    MOV    P1,#0FFh             ;reset SDA SCL bersamaan
    ACALL  DelayB
Loop:
    MOV    AddressI2C,#11101110B ;akses SPC dgn address default (111)
    MOV    stepcontrol,#00001111B ;Run,Bi,CW,Full
    MOV    StepSpeedH,#000H
    MOV    StepSpeedL,#000H      ;Min Speed
    MOV    StepH,#000H
    MOV    StepL,#018H           ;180 degree
    ACALL  Stepnit               ;Proses Steppernya
    MOV    P1,#0FFh             ;reset SDA SCL bersamaan
```




```
ACALL DelayA

MOV    AddressI2C,#11101110B ;akses SPC dgn address default (111)
MOV    stepcontrol,#00001101B ;Run,Bi,CCW,Full
MOV    StepSpeedH,#0B0H
MOV    StepSpeedL,#000H      ;Med Speed
MOV    StepH,#000H
MOV    StepL,#030H          ;90 degree
ACALL StepInit
MOV    P1,#0FFh
ACALL DelayB
AJMP   Loop

DelayA:  PUSH 07h      ;Menyimpan data pada R5, R6, R7 agar tidak hilang
        PUSH 06h
        PUSH 05h
        MOV   R5,#00DH ;aslinya 00Dh= 0.85ms
DelayA1: MOV   R6,#0FFH
DelayA0: MOV   R7,#0FFH
        DJNZ  R7,$
        DJNZ  R6,DelayA0
        DJNZ  R5,DelayA1
        POP  05h
        POP  06h
        POP  07h
        RET

DelayB:  PUSH 07h
        PUSH 06h
        PUSH 05h
        MOV   R5,#004H
DelayB1: MOV   R6,#0FFH
DelayB0: MOV   R7,#0FFH
        DJNZ  R7,$
        DJNZ  R6,DelayB0
        DJNZ  R5,DelayB1
        POP  05h
        POP  06h
        POP  07h
        RET

END
```

3. Setelah semua program diketik dan di save as dengan **EXAMPLE.asm** langkah selanjutnya adalah membuka program **ENG_I2C.INC** dan **STEPPER.INC**. lalu, klik **F9** atau klik symbol seperti ini  (Running Program/Build Current File) untuk mengetahui program tersebut terjadi **error** atau **no error**.
4. Apabila program berhasil **tanpa error**, hubungkan ISP K-51 USB ke PC. Kemudian bukalah aplikasi **ISP_PROG v1.4**.
Setelah langkah-langkah diatas diikuti, ikuti langkah berikut :
 - a. Pilihlah **READ DEVICE SIGNATURE** lalu pilih **RUN** ;berfungsi untuk melakukan cek apakah downloader sudah terhubung ke PC atau belum terhubung ke PC? Setelah aplikasi berhasil mendeteksi device K-51 langkah selanjutnya adalah FLASHING program yang kalian buat.
 - b. Klik **File** kemudia pilih **Load FLASH File** kemudian open file yang kalian buat. Setelah itu pilih **CHIP ERASE** dan **WRITE FLASH** kemudian klik **RUN**.
5. Langkah selanjutnya setelah program berhasil didownload, hubungkanlah beberapa komponen berikut:
 - a. P1.0 to S1
 - b. P1.1 to S2
 - c. P1.2 to S3
 - d. P1.3 to S4
 - e. P1.4 to RST
 - f. P1.6 to SCL
 - g. P1.7 to SDA
 - h. GND (slave) to GND (master)
 - i. VCC +5 (slave) to VCC +5 (master)
 - j. VMotor (slave) to VCC +12 (adapter)
 - k. GNDMotor (slave) to GND (master/adapter)
6. Amati pergerakan alat dan buatlah analisisnya.

Kesimpulan/Analisa dan Hasil Output:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

1. Buatlah program dengan ketentuan:
 - a. Jika ditekan P3.3 maka Motor Stepper akan bergerak CW, FULL STAPE, SPEED MAX dan 360 derajat.
 - b. Jika ditekan P3.2 maka Motor Stepper akan bergerak CCW, HALF STAPE, SPEED MIN dan 180 derajat.

[illegible]

Nama Assisten :		
BAGIAN 1	BAGIAN 2	MANDIRI
Tanggal Periksa :		

[illegible]

