STEPPER MOTOR INTERFACE BOARD

Model No : (VBMB - 13A)

Application Manual

Version 1.0

Technical Clarification /Suggestion :

✍ / ☎

Technical Support Division,

Vi Microsystems Pvt. Ltd.,

Plot No :75, Electronics Estate,

Perungudi, Chennai - 600 096, INDIA.

Ph: 91- 44-2496 1842, 91-44-2496 1852

Mail : sales@vimicrosystems.com,

Web : www.vimicrosystem.com

CONTENTS

CHAPTER - 1 INTRODUCTION TO 8055

1-1 Introduction 1 - 1

CHAPTER - 2 INTRODUCTION TO STEPPER MOTOR

2.1 Stepper Motor 2

2.2 Constructional Features 2

2.3 Wave Scheme (Unipolar Operation) 4

2.4 2Phase Scheme 6

2.5 Half Stepping Scheme 8

CHAPTER - 3 HARDWARE DESCRIPTION

3.1 Introduction 12

3.2 Address Decoding 12

3.3 Driving Circuitry 13

3.4 Operating Voltage Selection 14

CHAPTER - 4 INSTALLATION PROCEDURE 16

CHAPTER - 4 SOFTWARE EXAMPLES 18

LIST OF APPENDIX

Appendix - A Circuit Diagram 23

Appendix - B Connector Details 25

Appendix - C Component Layout 27

Appendix - D PROGRAMS IN 8086/88 ASSEMBLY LANGUAGE 29

Appendix - E PROGRAMS IN 8086/88 LCD TRAINER 33

PREFACE

VBMB-013A Technical reference manual is intended to present to the user all the details regarding the stepper motor and various modes with which it can be operated. VBMB-013A board is a microprocessor based stepper motor controller, which is capable of demonstrating the various modes of stepper motor operations.

The manual has been organised in such a way as to enable the user to learn about the aspects of stepper motor in a systematic fashion. Starting from the basics of the stepper motor, all the details have been provided in an order of gradually increasing complexity.

For those users who have been wondering about the advantages of the board, CHAPTER 1 is the place to start with.

Chapter 2 furnishes information regarding stepper motor and its operations.

Chapter 3 equips user with the circuit description of this board.

Chapter 4 illustrates the procedure to interface VBMB-13A to any your microprocessor trainer kit.

Chapter 5 contains the software examples for interfacing VBMB-13A board to microprocessor trainer kit.

Appendices contain all the relevant material, including the circuit diagram, the component layout, chip details etc.

We hope this board will be of great use to study the concept of stepper motor operation. Any suggestions to improve this product are greatly appreciated.

Write to:

The Customer-Support Division,
Vi Microsystems Pvt. Ltd.,
Plot No.75, Electronics Estate,
Perungudi, Chennai - 600 096.

Phone: (044) 496 1842, 496 1852.
Fax: (044) 496 1536.
E-mail: vimicro@vsnl.com

CHAPTER - 1

INTRODUCTION

We have introduced a series of microprocessor Trainers based on different microprocessors such as 8085A, Z-80, 6502, 8086, 8088, 8097, 8031 and we provide sufficient software and hardware experiments. Also we have a range of add-on-boards.

These add-on-boards are meant for beginners who desire to get into hardware interfacing and hardware oriented software.

VBMB-13A is one of the add-on-boards which enable a beginner to get familiarised with interfacing stepper motors to any of our microprocessor based trainer kits. This board supports stepper motor, ranging from 2 to 20Kg with operating voltages 6, 12 & 24V. The supply can be given externally. Two stepper motors of different voltages can be connected to this board. The following sections cover the details of stepper motors, hardware descriptions of this board and software listing for doing experiments in any of our microprocessor kits.

This board can be interfaced to any computer using our PC-Add-on-card VAD-107.

CHAPTER - 2

INTRODUCTION TO STEPPER MOTOR

2.1 STEPPER MOTOR:

A motor in which the rotor is able to assume only discrete stationary angular position is a stepper motor. The rotary motion occurs in a stepwise manner from one equilibrium position to the next.

Stepper motor control is a very popular application of microprocessor in control area. They are widely used in (simple position control systems in the open and closed loop mode) a variety of applications such as computer peripherals (printers, disk driver etc.) and in the areas of process control machine tools, medicine, numerically controlled machines and Robotics.

2.2 CONSTRUCTIONAL FEATURES :

A Stepper motor could be either of the reluctance type or of the permanent magnet type (PM). A PM stepper motor consists of multi phase stator and two part permanent magnet rotor. The VR stepper motor has un magnetised rotor. PM stepper motor is the most commonly used type. The basic two phase stepper motor consists of two pairs of stator poles. Each of the four poles has its own winding. The excitation of any one winding generates a north pole (N), a south pole (S) gets induced at the diametrically opposite side.

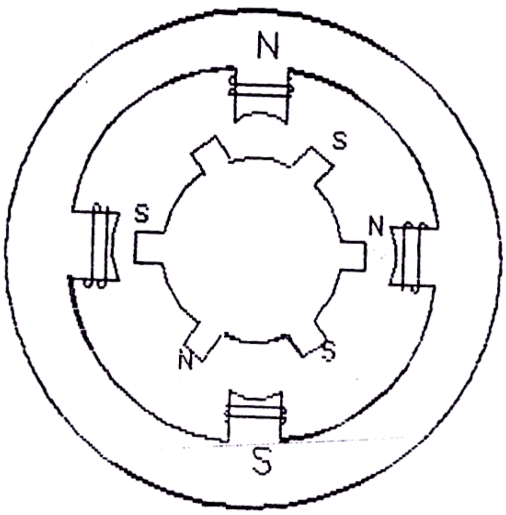


Fig 2-1 Stepper Motor Cross-sectional view

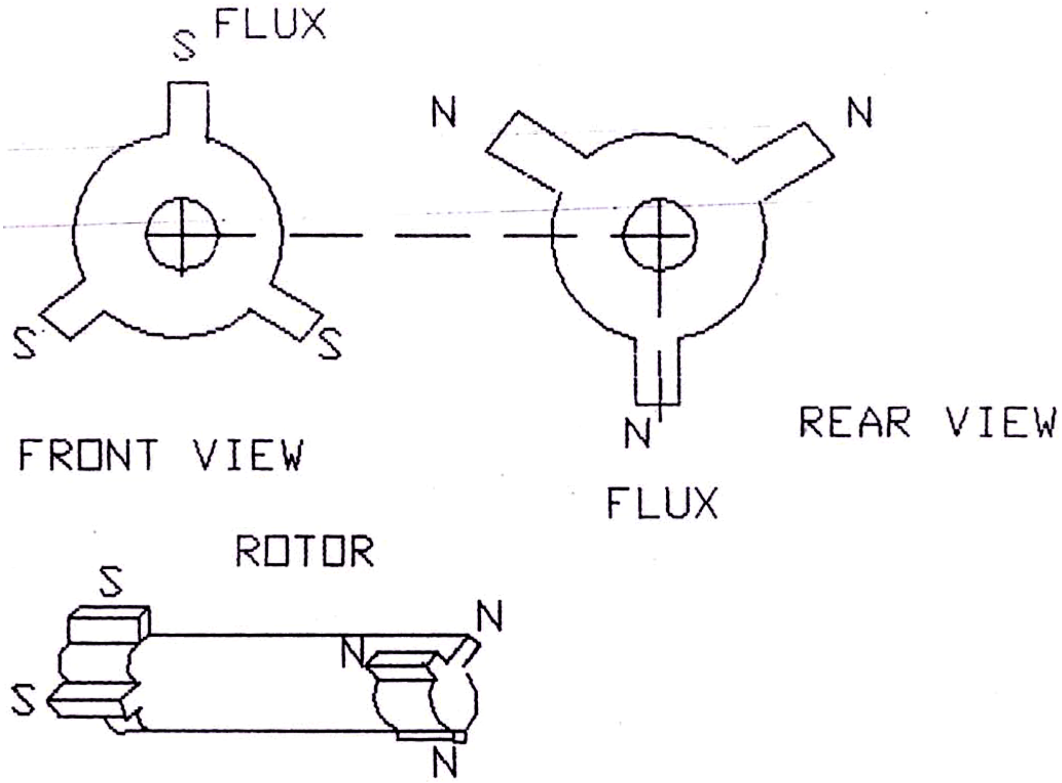


Fig.2-2 Typical Stepper Motor Rotor

As shown in the figure 2-1 the four pole structure is continuous with the stator frame and the magnetic field passes through the cylindrical stator annular ring. The rotor magnetic system has two end faces. The left face is permanently magnetised as south pole and the right face as north pole. The south pole structure and the north pole structure possess similar pole faces. The north pole structure is twisted with respect to the south pole structure so that south pole comes precisely between two north poles. The north pole structure is offset with respect to the south pole structure by one pole pitch. The cross sectional view is shown in figure 2-2. In an arrangement where there are four stator poles and three pairs of rotor poles, there exists 12 possible stable positions in which a south pole of the rotor can lock with a north pole of the stator. From this it can be noted that the step size is,

$$\frac{360°}{Ns \times Nr}$$

Where

Ns is the No. of stator poles.
Nr is the No. of pairs of rotor poles.

Generally step size of the stepper motor depends upon NR. These stable positions can be attained by simply energising the winding on any one of the stator poles with a DC. There are three different schemes available for "stepping" a stepper motor. These are:

(a) Wave scheme

(b) 2-phase scheme and

(c) Half stepping or mixed scheme.

These three schemes are described in detail in the following sections.

2.3 WAVE SCHEME (UNIPOLAR OPERATION) :

The stepper motor windings A1, B1, A2, B2 can be cyclically excited with a DC current to run the motor in the clockwise direction. By reversing the phase sequence as A1, B2, A2, B1, we can obtain anticlockwise stepping.

2.3.1 STEPPING DIAGRAM :

Consider the four possible rotor positions and the corresponding stator excitations, illustrated in Fig.2-3. In step (a), it is seen that the rotor pole S1 locks with the stator pole A1(N) and the rotor pole N1 locks with the stator pole A2(S'). Under these conditions, it is obvious that the winding A1 must have been energised. A2 (S') is an induced south pole.

Next, A1 is de-energised and B1 is turned on. The nearest rotor pole S2 locks with B1(N) and the corresponding rotor pole N2 locks with the induced pole B2(S'). This is illustrated in step (b) of Fig.2.3.

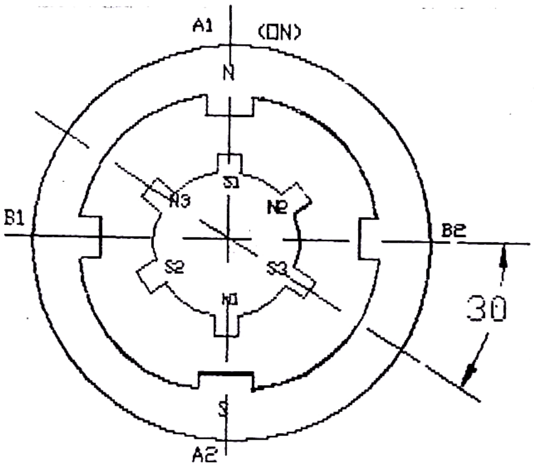


Fig.2-3 Simple Wave Scheme for Stepper Motor (Cont...)

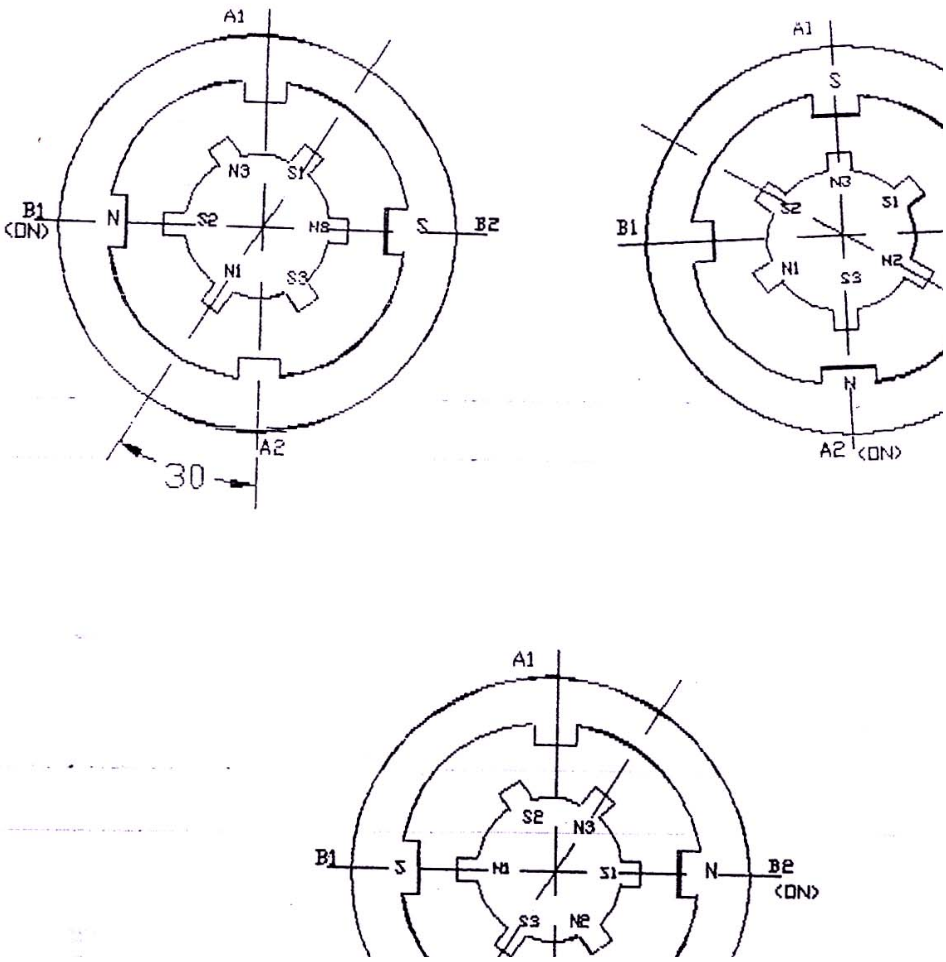


Fig.2-3 Simple Wave Scheme for Stepper Motor

When A2 alone is energised next, the nearest rotor pole S3 locks with A2(N) (step (c) of Fig.2.3).

B2 alone is energised next and the nearest rotor pole S1 locks with B2(N). Totally 12 such steps cause a displacement of 360°. The step angle is 30° (mechanical).

The switching sequence for the wave mode of excitation is given in Table 2-1.

Table 2.1 Wave switching scheme

Anticlockwise					**Clockwise**				
Step	A1	A2	B1	B2	Step	A1	A2	B1	B2
1	1	0	0	0	1	1	0	0	0
2	0	0	0	1	2	0	0	1	0
3	0	1	0	0	3	0	1	0	0
4	0	0	1	0	4	0	0	0	1

2.4 2-PHASE SCHEME :

In this scheme, any two adjacent stator windings are energised. There are two magnetic fields active in quadrature and none of the rotor pole faces can be in direct alignment with the stator poles. A partial but symmetric alignment of the rotor poles is of course possible.

Typical equilibrium conditions of the rotor when the windings on two successive stator poles are excited are illustrated in Fig.2.4. In step (a), A1 and B1 are energised. The pole-face S1 tries to align itself with the axis of A1 (N) and the pole face S2 with B1(N). The north pole N3 of the rotor finds itself in the neutral zone between A1(N) and B1(N). S1 and S2 of the rotor, position themselves symmetrically with respect to the two stator north pole.

Next, when B1 and A2 are energised, S2 tends to align with B1(N) and S3 with A2(N). Of course, again under equilibrium conditions, only partial alignment is possible and N1 finds itself in the neutral region, midway between B1(N) and A2(N) [Step (b)]. In step (c), A2 and B2 are on. S3 and S1 tend to align with A2(N) and B2(N), respectively, with N2 in the neutral zone. Step (d) illustrates the case when A1 and B2 are on.

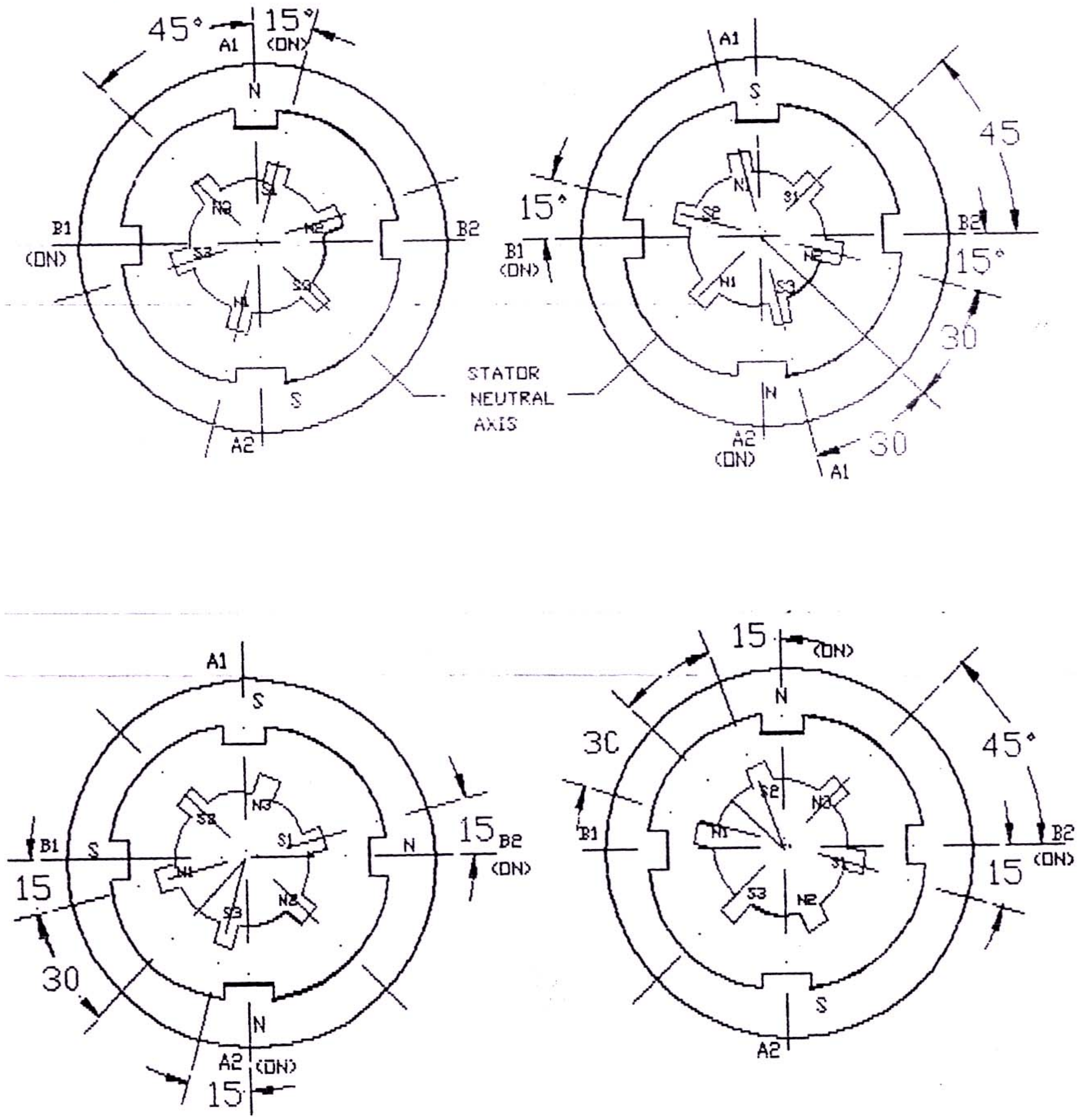


Fig.2-4 2-Phase Drive Scheme

The step angle is 30° as in the wave scheme. However, the rotor is offset by 15° in the two-phase scheme with respect to the wave scheme. A total of 12 steps are required to move the rotor by 360° (mechanical). Two-phase drives produce more torque than the wave drives.

The switching sequence for the 2-phase scheme is given in Table 2.2.

Table 2.2 2-Phase switching scheme

Clockwise					**Anti clock wise**				
Step	A1	A2	B1	B2	Step	A1	A2	B1	B2
1	1	0	0	1	1	1	0	1	0
2	0	1	0	1	2	0	1	1	0
3	0	1	1	0	3	0	1	0	1
4	1	0	1	0	4	1	0	0	1

2.5 HALF STEPPING SCHEME :

The wave scheme as well as the 2-phase scheme give steps of size 30° for the stepper motor under consideration. However, there is an offset of 15° between these two schemes. By interleaving these two schemes, the step size can be reduced to 15°, thereby improving the accuracy of the stepper motor. The half stepping scheme is a mixture of the wave scheme and the 2-phase scheme [Fig.2.5 (a) to (i)].

The switching sequence is:

i. A1 (ON)
ii. A1 and B1 (ON)
iii. B1 (ON)
iv. B1 and A2 (ON)
v. A2 (ON)
vi. A2 and B2 (ON)
vii. B2 (ON)
viii. B2 and A1 (ON)
ix. A1 (ON), etc.

Eight steps are required to move the shaft by 120° and 24 steps for one complete revolution. By reversing the switching sequence, we can reverse the direction of rotation. One major disadvantage of the half stepping scheme is torque fluctuations. This is because the aligning torque for the wave scheme is different from that for the 2-phase scheme.

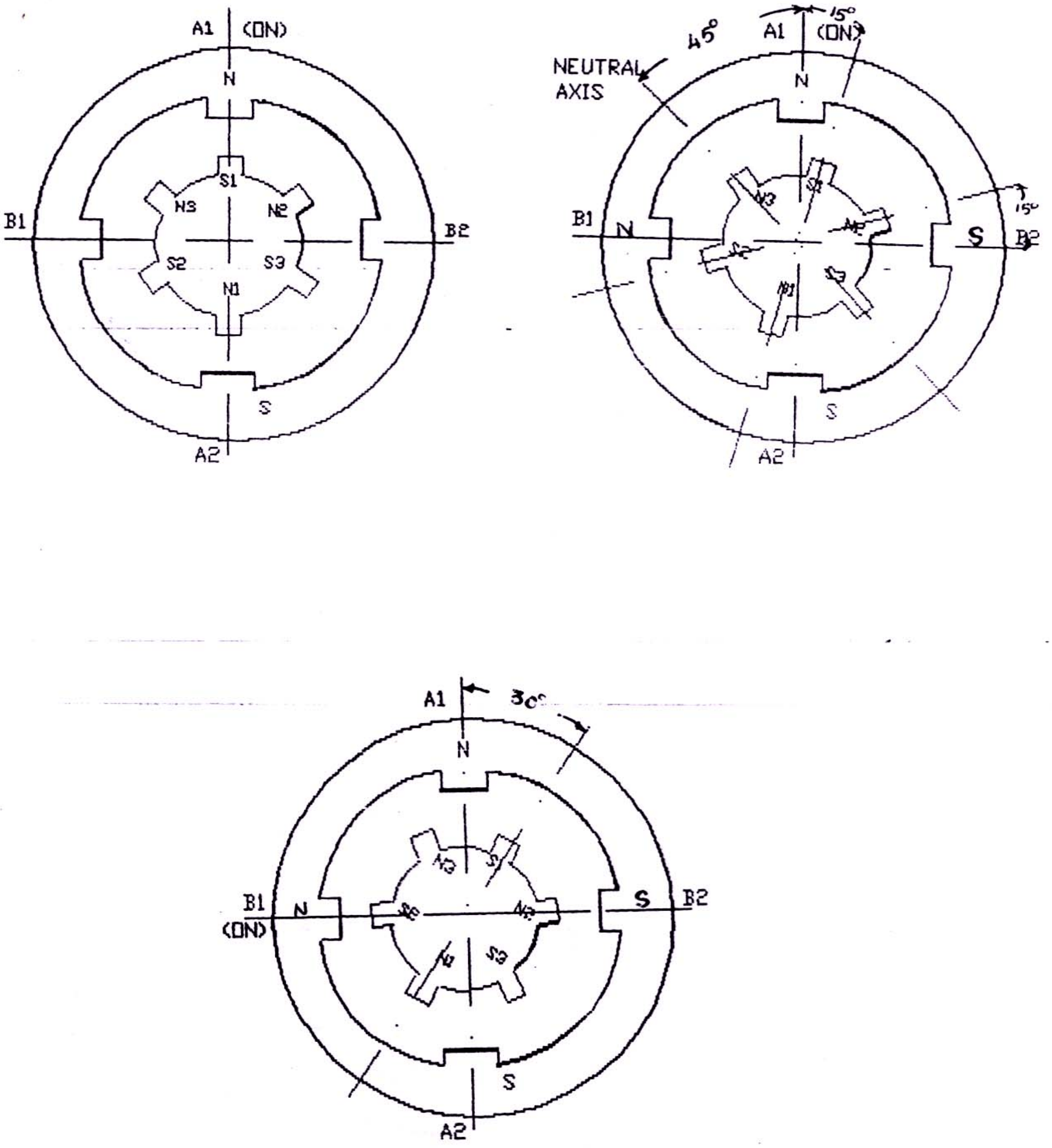


Fig.2-5 Half Stepping Scheme (Cont...)

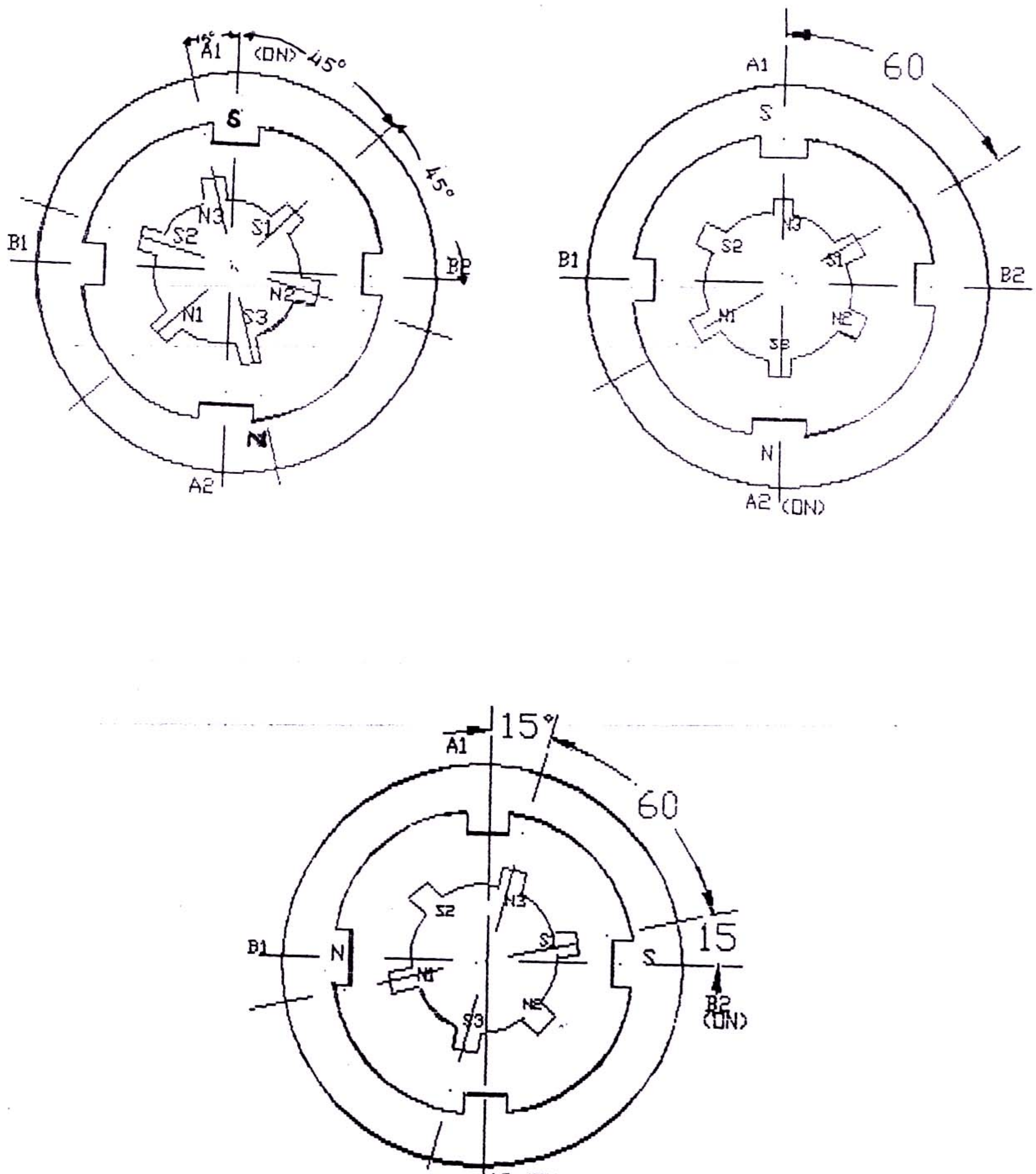


Fig.2-5 Half Stepping Scheme (Cont...)

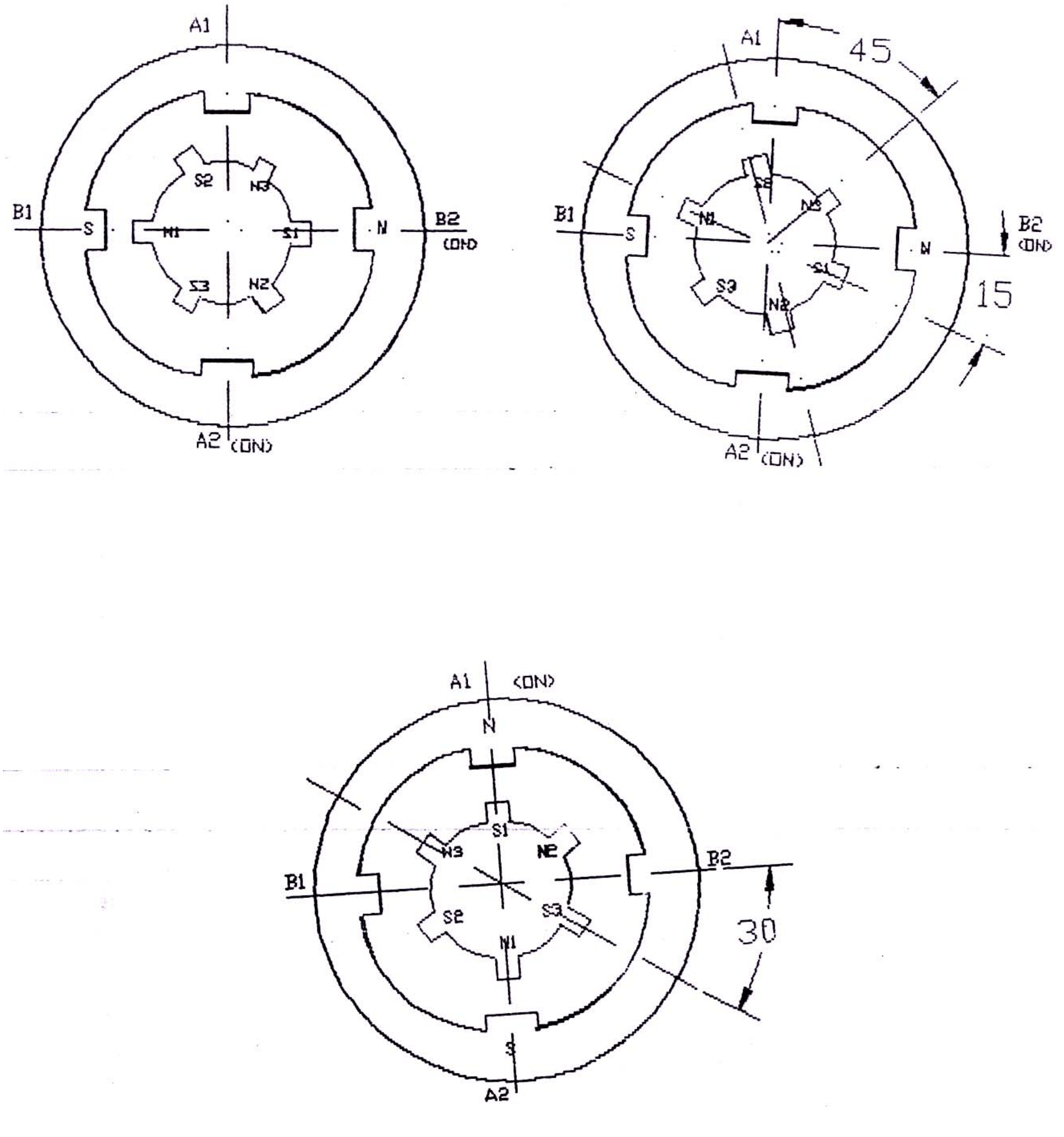


Fig.2-5 Half Stepping Scheme

CHAPTER - 3

HARDWARE DESCRIPTION

3.1 INTRODUCTION

Among the Vi range of add-on-cards, the VBMB-13A supports the stepper motor interface. This board can be plugged into the VXT Bus in any EB kits using 50 core cable and into the VME Bus of any other kits using our VXT Bus extender. This board consists of 3 sections as following.

1. Address decoding.

2. Stepper motor driving circuitry.

3. Operating voltage selections.

3.2 ADDRESS DECODING

The 74138 chip (U2) generates the address decoding logic to generate the device select pulses, CS1 and CS2 for selecting the ICs 74175 (U3 &U5). The 74175 (U3 & U5) latches the data bus to the stepper motor driving circuitry. Address lines A7 & A6 from VXT Bus is NANDed and the O/P is connected to 138 enable signal. Similarly IOW & IOR signals are NANDed and the NAND gate O/P is connected to Pin 6 of 74138, Pin 4 is grounded.

74175 at U3 is selected with the address

A7	A6	A5	A4	A3	A2	A1	A0	
1	1	0	0	0	X	X	X	= C0 [HEX]

74175 at U4 is selected with the address

A7	A6	A5	A4	A3	A2	A1	A0	
1	1	0	0	1	X	X	X	= C8 [HEX]

74125 is selected with the address

A7	A6	A5	A4	A3	A2	A1	A0	
1	1	0	1	0	0	X	X	= D0 [HEX]

A7	A6	A5	A4	A3	A2	A1	A0	
1	1	0	1	1	X	X	X	= D8 [HEX]

The circuit diagram is given in Appendix-A.

3.3 DRIVING CIRCUITRY

Stepper motor requires logic signals of relatively high power. In this board the silicon darlington pair (TIP 122) transistors are used to supply that required power. The driving pulses are generated by the interface circuit. The input for the interface circuit are TTL pulses generated under software control using a microprocessor trainer kit. The TTL levels of pulse sequence from the data bus is translated to high voltage output pulses using a buffer 7407 with open collector.

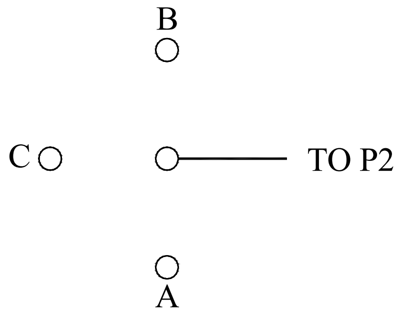
The darlington pair transistor (TIP 122) drive the stepper motor as they withstand higher current. A 220 ohm resistor and an IN4148 diode are connected between the power supply and darlington pair collector for supporting flyback current.

The data lines D0-D3 and D4-D7 are used to drive the 8 TIP 122 available on this board as shown in the Appendix-A. The four Collector points of each TIP 122 are brought to two 5 pin connectors P2 & P3 to connect two different stepper motors. With this board it is possible to connect stepper motor of torque ranging from 2Kg to 20Kg with operating voltage of 12V, 24V & 6V.

3.4 OPERATING VOLTAGE SELECTION

As explained earlier it is possible to connect two stepper motors of different operating voltage. This is made possible by jumpers J1 to J3.

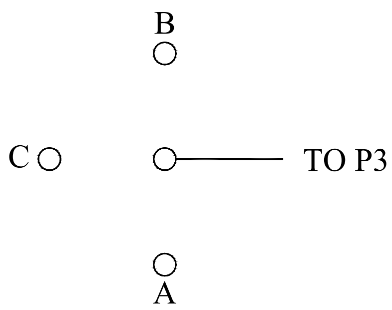
JUMPER J1:



This jumper selects the voltage for driving stepper motor connected at connector P2.

POSITION OF J1	VOLTAGE SELECTED
A	12V
B	6V
C	24V

JUMPER J2:

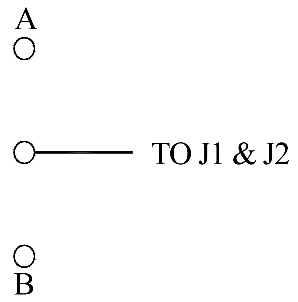


This jumper selects the voltage for driving stepper motor connected at connector P3.

POSITION OF J2	VOLTAGE SELECTED
A	12V
B	6V
C	24V

JUMPER J3:

This jumper should be strapped when using stepper motor of operating voltage 12V.



POSITION OF J3	VOLTAGE SELECTED
A	12V from VXT BUS
B	12V Given externally

Strap J3 in A position when 12V from microprocessor trainer kit is used.

Strap J3 in B position when external 12V is used.

CHAPTER - 4

INSTALLATION PROCEDURE

In the previous chapter we have studied about the circuit description. In this chapter let us see how to connect the VBMB-13A to our microprocessor kit.

The figure 4.1 shows how to connect a VBMB-13A board to the EB trainer kit.

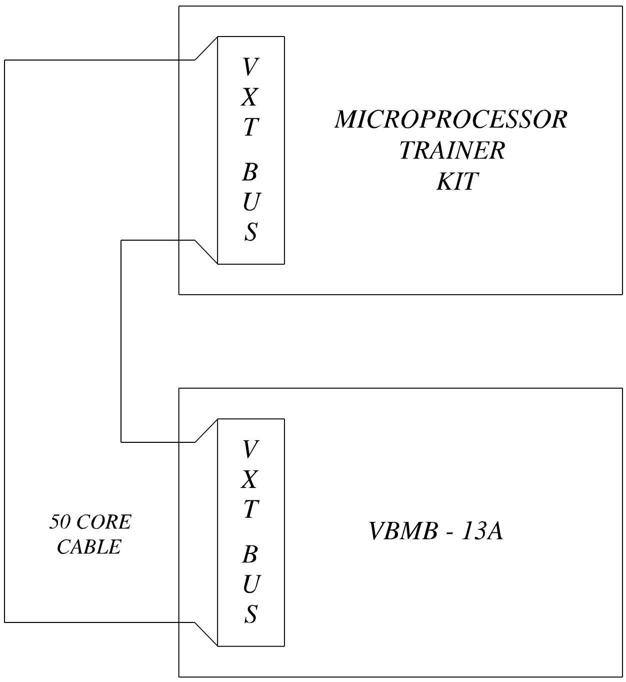


Figure 4-1.

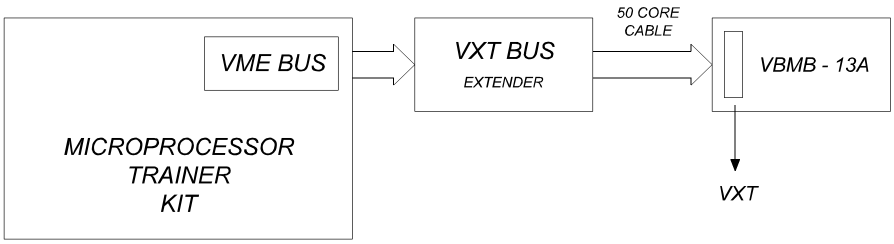


Figure 4-2.

Figure 4.2 shows how to connect a VBMB-13A board to kits other flash EB model.

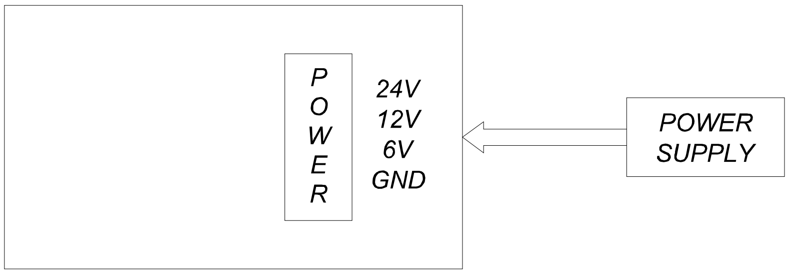


Figure 4.3.

Figure 4.3 shows how to connect a VBMB-13A to power supply. For different voltages the jumpers should be placed as explained in the previous chapter.

CHAPTER - 5

SOFTWARE EXPERIMENTS

This section explains the software listing using 8085 based trainer kit.

EXAMPLE 5-1:

Aim:

To run a stepper motor at different speed in two directions.

PROGRAM

```
4100   21 1A 41       START:     LXI    H,LOOK UP
4103   06 04                     MVI    B, 04
4105   7E             REPT:      MOV    A,M
4106   D3 C0                     OUT    0C0H
4108   11 03 03                  LXI    D, 0303H
410B   00             DELAY      NOP
410C   1B                        DCX    D
410D   7B                        MOV    A,E
410E   B2                        ORA    D
410F   C2 0B 41                  JNZ    DELAY
4112   23                        INX    H
4113   05                        DCR    B
4114   CZ 05 41                  JNZ    REPT
4117   C3 00 41                  JMP    START
411A   09 03 06 0C    LOOK UP:   DB     09 05 06 0A
```

RESULT:

Enter the above program starting from location 4100. Execute the same. The stepper motor rotates. Speed can be varied by varying the count at DE pair. Direction can be varied by entering the data in the look-up table in the reverse order.

EXAMPLE 5-2:

Aim:

To run a stepper motor for required angle within 360°, which is equivalent to 256 steps.

PROGRAM

```
4101   OE HEX DATA                 MVI     C, HEX DATA
4102   21 20 41       START:       LXI     H, LOOK UP
4105   06 04                       MVI     B, 04
4107   7E             REPT:        MOV     A,M
4108   D3 C0                       OUT     C0
410A   OD                          DCR     C
410B   CA 24 41                    JZ      END
410E   11 03 03                    LXI     D, COUNT
4111   00             DELAY:       NOP
4112   1B                          DCX     D
4113   7B                          MOV     A,E
4114   B2                          ORA     D
4115   C2 11 41                    JNZ     DELAY
4118   23                          INX     H
4119   05                          DCR     B
411A   C2 07 41                    JNZ     REPT
411D   C3 02 41                    JMP     START
4120   09 05 06 0A    LOOK UP:     DB      09 05 06 0A
4124   76             END:         HLT
```

RESULT:

Enter the above program and execute it. By converting the required step in decimal to hex and entering the hex data at 4101 the motor rotates for so much steps and then stops.

EXAMPLE 5-3:

Aim:

To run a stepper motor in two ports.

PROGRAM

```
5000                            ORG     5000H
5000  21 1B 50      START:      LXI     H,LOOKUP
5003  06 04                     MVI     B,04
5005  7E            REPT:       MOV     A,M
5006  D3 C0                     OUT     0C0H
5008  D3 C8                     OUT     0C8H
500A  11 03 03                  LXI     D,0303H
500D  1B            DELAY:      DCX     D
500E  7B                        MOV     A,E
500F  B2                        ORA     D
5010  C2 0D 50                  JNZ     DELAY
5013  23                        INX     H
5014  05                        DCR     B
5015  C2 05 50                  JNZ     REPT
5018  C3 00 50                  JMP     START
501B  09 05 06 0A   LOOKUP:     DB      09H,05H,06H,0AH
501F                            END
```

RESULT:

Enter the program starting from 5000H onwards. Connect the stepper motors in the two ports. Execute the program. Now it can be seen that the two stepper motors runs in the forward direction simultaneously.

EXAMPLE 5-4:

Aim:

To run Stepper Motor in both Forward and Reverse directions with delay.

PROGRAM

```
4100                                ORG     4100H
4100  0E 20         START:          MVI     C,20H
4102  21 3F 41      FORWD:          LXI     H,FORLOOK
4105  CD 21 41                      CALL    ROTATE
4108  0D                            DCR     C
4109  C2 02 41                      JNZ     FORWD
410C  CD 35 41                      CALL    STOP
410F  0E 20                         MVI     C,20H
4111  21 43 41      REVES:          LXI     H,REVLOOK
4114  CD 21 41                      CALL    ROTATE
4117  0D                            DCR     C
4118  C2 11 41                      JNZ     REVES
411B  CD 35 41                      CALL    STOP
411E  C3 00 41                      JMP     START
4121  06 04         ROTATE:         MVI     B,04H
4123  7E            REPT:           MOV     A,M
4124  D3 C0                         OUT     0C0H
4126  11 03 03                      LXI     D,0303H
4129  1B            LOOP1:          DCX     D
412A  7B                            MOV     A,E
412B  B2                            ORA     D
412C  C2 29 41                      JNZ     LOOP1
412F  23                            INX     H
4130  05                            DCR     B
4131  C2 23 41                      JNZ     REPT
4134  C9                            RET
4135  11 FF FF      STOP:           LXI     D,FFFFH
4138  1B            LOOP2:          DCX     D
4139  7B                            MOV     A,E
413A  B2                            ORA     D
413B  C2 38 41                      JNZ     LOOP2
413E  C9                            RET
413F  09 05 06 0A   FORLOOK         DB      09H,05H,06H,0AH
4143  0A 06 05 09   REVLOOK         DB      0AH,06H,05H,09H
4147                                END
```

RESULT:

Enter the program starting from 4100H location and connect the stepper motor in port 1. Execute the program. Now you can see that the stepper motor runs in forward direction and reverse direction continuously with a delay.

EXERCISES: 5-5

1. Write a program to run two stepper motors at a time continuously.

NOTE :

The port address of 2nd stepper motor driving circuitry is C8.

2. Write a program to run the stepper motor in other two modes as explained in chapter 2.

NOTE :

The Look-up table data should be changed accordingly.

3. Write a program to run the Stepper motor for any no. of steps and to stop it.

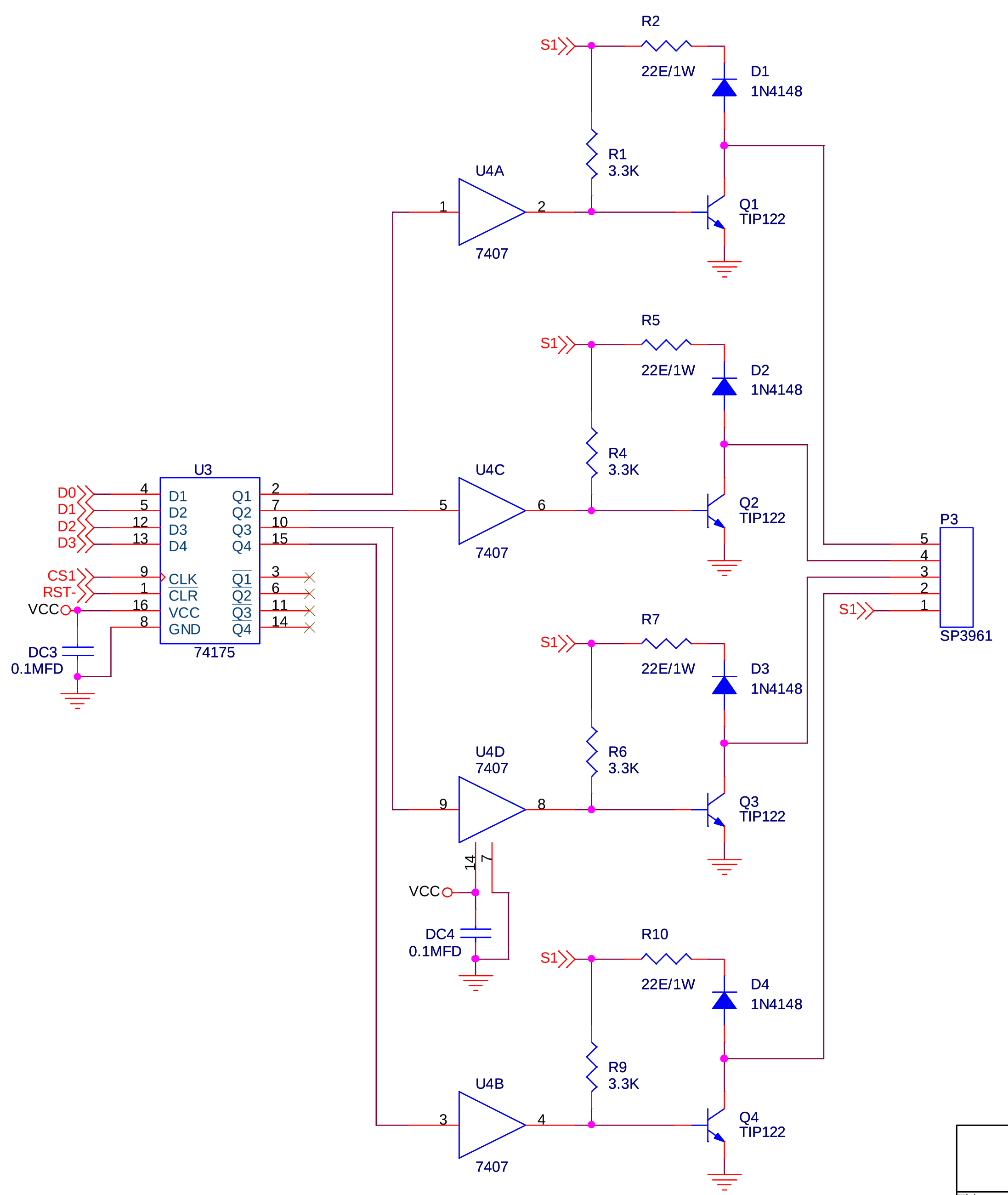
4. Write a program to make the stepper motor to rotate for required angle and to wait for required time and again to rotate for required angle and so on.

CONCLUSION

We have converted most of the hardware and software sections in details. Thus we come across this section.

APPENDIX-A

CIRCUIT DIAGRAM



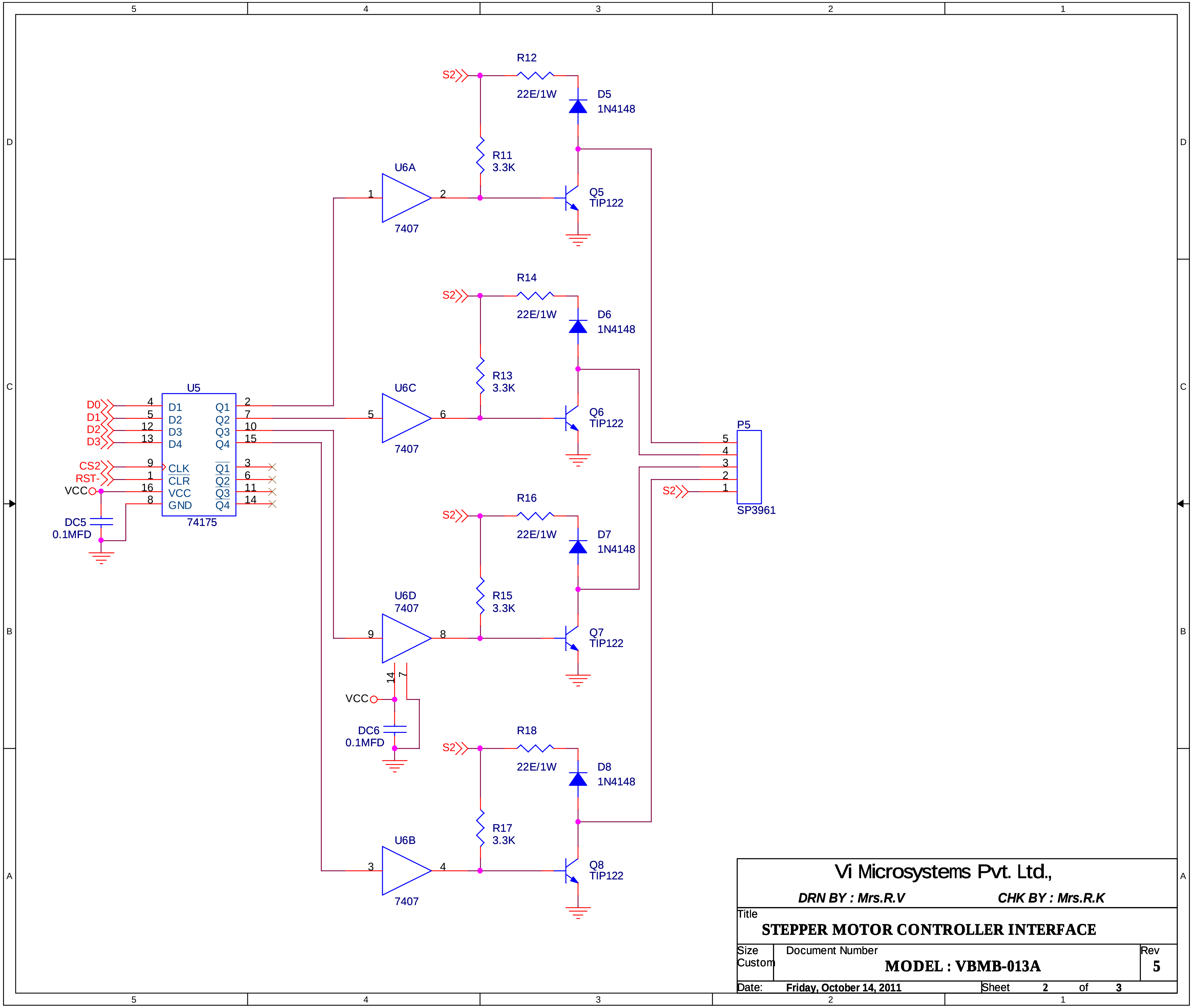
Vi Microsystems Pvt. Ltd.,

DRN BY : Mrs.R.V   CHK BY : Mrs.R.K

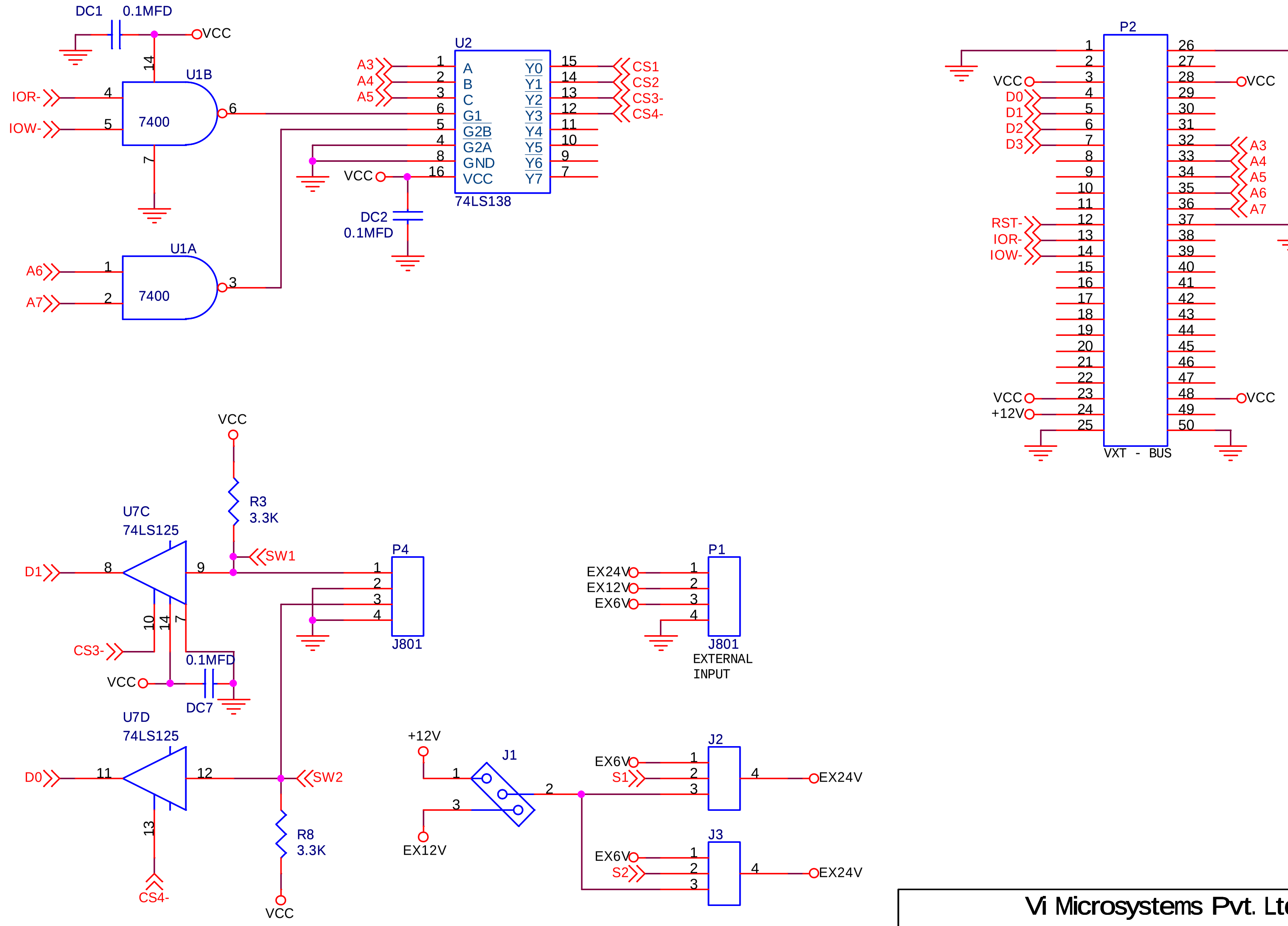
Title: STEPPER MOTOR CONTROLLER INTERFACE

Size	Document Number	Rev
Custom	MODEL : VBMB-013A	5

Date: Friday, October 14, 2011   Sheet 1 of 3



Vi Microsystems Pvt. Ltd.,		
DRN BY : Mrs.R.V	CHK BY : Mrs.R.K	
Title: **STEPPER MOTOR CONTROLLER INTERFACE**		
Size: Custom	Document Number: **MODEL : VBMB-013A**	Rev: **5**
Date: Friday, October 14, 2011	Sheet 2 of 3	



Vi Microsystems Pvt. Ltd.,

DRN BY : Mrs.R.V *CHK BY : Mrs.R.K*

Title

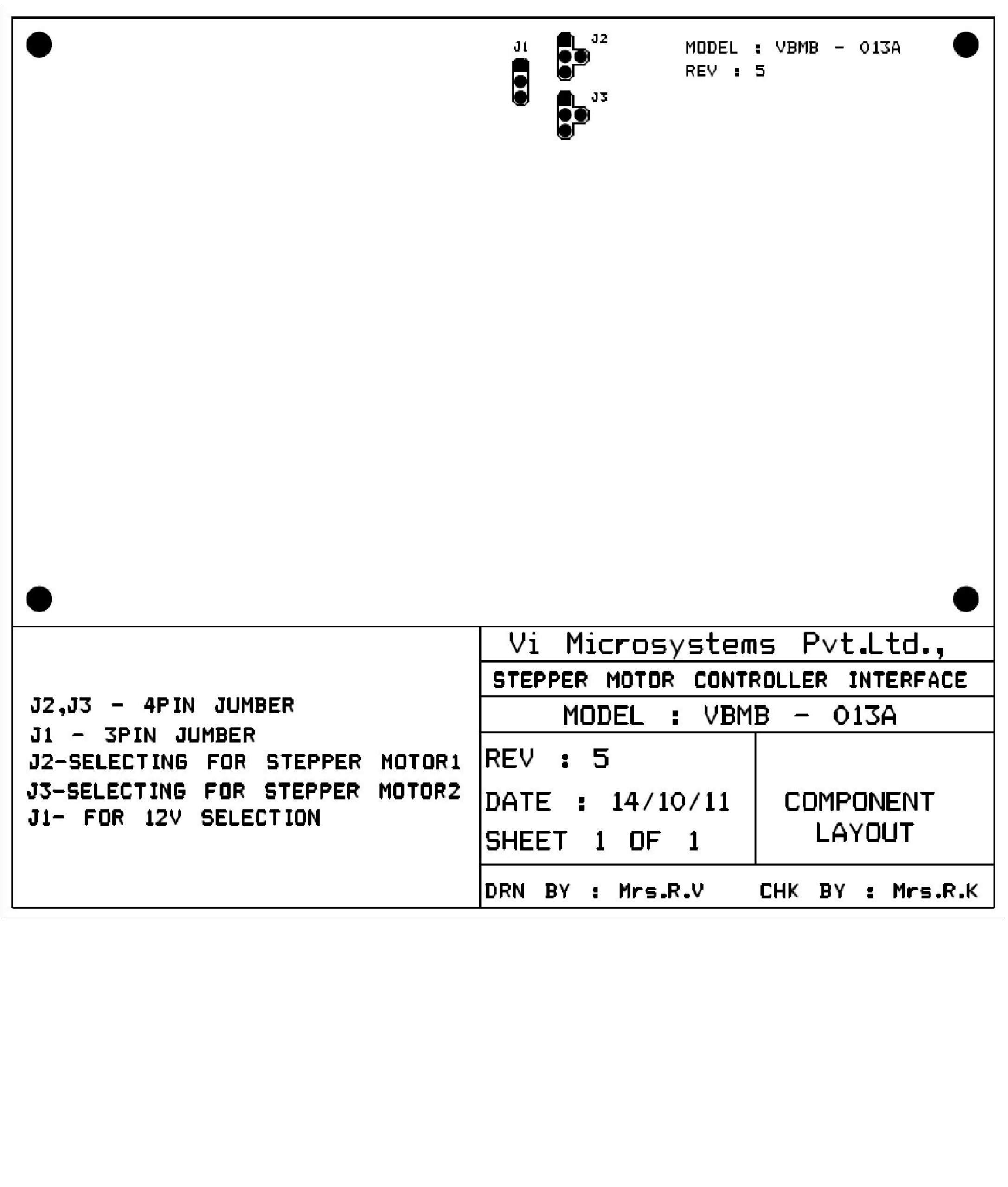
STEPPER MOTOR CONTROLLER INTERFACE

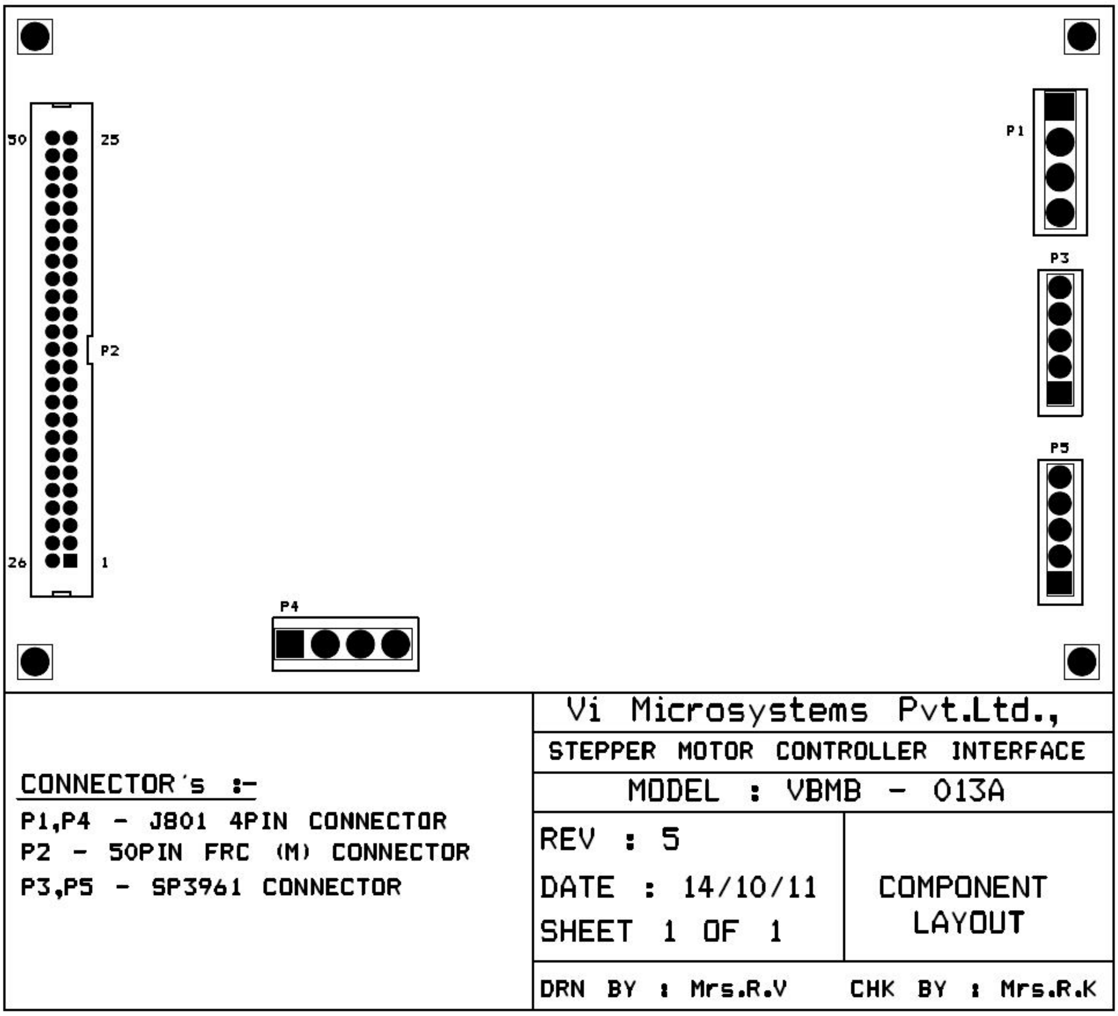
Size	Document Number	Rev
A	**MODEL : VBMB-013A**	**5**

Date: Friday, October 14, 2011 Sheet 3 of 3

APPENDIX-B

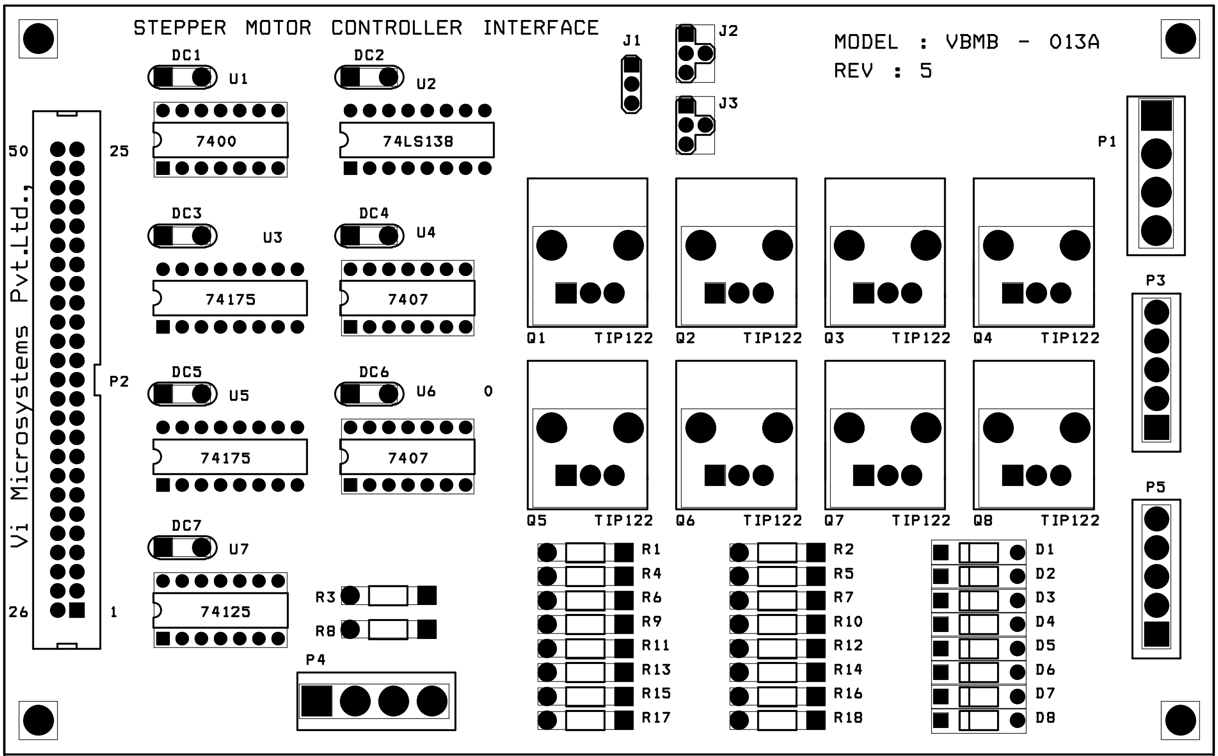
CONNECTOR DETAILS




50
25
P1
P3
P2
P5
26
1
P4
Vi Microsystems Pvt.Ltd.,
STEPPER MOTOR CONTROLLER INTERFACE
MODEL : VBMB - 013A
CONNECTOR's :-
P1,P4 - J801 4PIN CONNECTOR
P2 - 50PIN FRC (M) CONNECTOR
P3,P5 - SP3961 CONNECTOR
REV : 5
DATE : 14/10/11
SHEET 1 OF 1
COMPONENT LAYOUT
DRN BY : Mrs.R.V
CHK BY : Mrs.R.K

APPENDIX-C

COMPONENT LAYOUT



LIST OF COMPNENT's :-

IC's :-

U1 - 7400 (14PIN IC BASE)
U2 - 74LS138 (16PIN IC BASE)
U3,U5 - 74175 (16PIN IC BASE)
U4,U6 - 7407 (14PIN IC BASE)
U7 - 74125 (14PIN IC BASE)

CONNECTOR's :-

P1,P4 - J801 4PIN CONNECTOR
P2 - 50PIN FRC (M) CONNECTOR
P3,P5 - SP3961 CONNECTOR

RESISTOR's :-

R1,R4,R6,R9,R11 - 3.3K
R13,R15,R17 - 3.3K
R2,R5,R7,R10,R12 - 22E/1W
R14,R16,R18 - 22E/1W
R3,R8 - 3.3K

OTHER COMPONENT's :-

(DC1-DC7) - 0.1MF DISC CAPACITOR
(D1-D8) - 1N4148 DIODE
(Q1-Q8) - TIP122 WITH HEATSINK
J1 - 3PIN JUMBER
J2,J3 - 4PIN JUMBER

Vi Microsystems Pvt.Ltd.,	
STEPPER MOTOR CONTROLLER INTERFACE	
MODEL : VBMB - 013A	
REV : 5 DATE : 14/10/11 SHEET 1 OF 1	COMPONENT LAYOUT
DRN BY : Mrs.R.V	CHK BY : Mrs.R.K

APPENDIX-D

PROGRAMS IN 8086/88 ASSEMBLY LANGUAGE

EXAMPLE-1: To run a stepper motor at different speed.

```
1000                PORT1       EQU     OC0H
1000                            ORG     1000H
1000  BF 14 10      BEGIN:
1003  B1 04         START:      MOV     DI, OFFSET TABLE
1005  8A 05                     MOV     CL, 04
1007  E6 C0         LOOP1:      MOV     AL, [DI]
1009  BA 1010                   OUT     PORT1, AL
100C  4A                        MOV     DA, 1010H
100D  75 FD                     JNZ     DELAY
100F  47                        INC     DI
1010  E2 F3                     LOOP    LOOP1
1012  EB EC                     JMP     START
1014                TABLE:
1014  09 05 06 0A               DB      9,5,6,0AH
```

RESULT:

Enter the above program starting from location 1000. Execute the same. The stepper motor rotates. Speed can be varied by varying the count at dx register pair. Direction can be varied by entering the data in the look-up table in the reverse order.

EXAMPLE-2:

To run a stepper motor for required angle within 460°, which is equivalent to 256 steps.

```
                   PORT1     EQU     0C0H
1000                         ORG     1000H
1000               BEGIN:
1000  B3 45                  MOV     BL,45H
1002  BF 1B 10     START:    MOV     DI,OFFSET TABLE
1005  B1 04                  MOV     CL,04
1007  8A 05        REPT:     MOV     AL,[DI]
1009  E6 C0                  OUT     PORT1,AL
100B  FE CB                  DEC     BL
100D  74 0B                  JZ      END
100F  BA 1010                MOV     DX,1010H
1012  4A           DELAY:    DEC     DX
1013  75 FD                  JNZ     DELAY
1015  47                     INC     DI
1016  E2 EF                  LOOP    REPT
1018  EB E8                  JMP     START
101A  F4           END:      HLT
101B   TABLE:
101B  09 05 06 0A            DB      9.5,6,0AH
```

RESULT:

Enter the above program and execute it. By converting the required step in decimal to hex and entering the hex data at 1001 the motor rotates for so much steps and then stops.

EXAMPLE-3:

To run a stepper motor in both ports.

```
                        PORT1       EQU   0C0H
                        PORT2       EQU   0C8H
1000                                ORG   1000H
1000                    BEGIN:
1000  BF 16 10          START:      MOV   DI, OFFSET TABLE
1003  B1 04                         MOV   CL,04
1005  8A 05             REPT:       MOV   AL,[DI]
1007  E6 C0                         OUT   PORT1,AL
1009  E6 C8                         OUT   PORT2,AL
100B  BA 1010                       MOV   DX,1010H
100E  4A                DELAY:      DEC   DX
100F  75 FD                         JNZ   DELAY
1011  47                            INC   DI
1012  E2 F1                         LOOP  REPT
1014  EB EA                         JMP   START
1016  TABLE:
1016  09 05 06 0A                   DB    9,5,6,0AH
```

RESULT:

Enter the program starting from 1000 onwards. Connect the stepper motors in the two ports. Execute the program. Now it can be seen that the two stepper motors runs in the forward direction simultaneously.

EXAMPLE-4:

To run the stepper motor in both forward and reverse direction with delay.

```
                        PORT1       EQU     0C0H
1000                                ORG     1000H
1000                    BEGIN:
1000    B3 20           START:      MOV     BL,20H
1002    BF 37 10        FORWD:      MOV     DI,OFFSET FORW
1005    E8 1800                     CALL    ROTATE
1008    FE CB                       DEC     BL
100A    75 F6                       JNZ     FORWD
100C    E8 21 00                    CALL    DELAY
100F    B3 20                       MOV     BL,20H
1011    BF 3B 10        REVER:      MOV     DI,OFFSET REV
1014    E8 09 00                    CALL    ROTATE
1017    FE CB                       DEC     BL
1019    75 F6                       JNZ     REVER
101B    E8 12 00                    CALL    DELAY
101E    EB E0                       JMP     START
1020    B1 04           ROTATE:     MOV     CL,04
1022    8A 05           REPT:       MOV     AL,[DI]
1024    E6 C0                       OUT     PORT1,AL
1026    BA 1010                     MOV     DX,1010H
1029                    LOOP1:
1029    4A                          DEC     DX
102A    75 FD                       JNZ     LOOP1
102C    47                          INC     DI
102D    E2 F3                       LOOP    REPT
102F    C3                          RET
1030    BA FFFF                     MOV     DX,0FFFFH
1033    4A              DELAY:      DEC     DX
1034    75 FD                       JNZ     DELAY
1036    C3                          RET
1037    09 05 06 0A     FORW:       DB      9,5,6,0AH
103B    0A 06 05 09     REV:        DB      0AH,6,5,9
```

RESULT:

Enter the program starting from 1000 onwards. Execute the program after connecting the stepper motor in port 1. Execute the program. Now you can see that the stepper motor runs in forward and reverse direction continuously with a delay.

APPENDIX-E

PROGRAMS IN 8086/88 LCD TRAINER

EXAMPLE-1: To run a stepper motor at different speed.

1000	C7 C7 18 10	MOV	DI, 1018
1004	C6 C1 04	MOV	CL, 04
1007	8A 05	MOV	AL, [DI]
1009	E6 C0	OUT	C0, AL
100B	C7 C2 10 10	MOV	DX, 1010
100F	4A	DEC	DX
1010	75 FD	JNZ	100F
1012	47	INC	DI
1013	E2 F2	LOOP	1007
1015	E9 E8 FF	JMP	1000
1018	09 05 06 0A	TABLE	09 05 06 0A

RESULT:

Enter the above program starting from location 1000. Execute the same. The stepper motor rotates. Speed can be varied by varying the count at dx register pair. Direction can be varied by entering the data in the look-up table in the reverse order.

PROGRAMS IN 8086/88 LCD TRAINER

EXAMPLE-2: To run a stepper motor for required angle within 360°, which is equvalent to 256 steps.

1000	C6 C3 45	MOV	BL, 45
1003	C7 C7 20 10	MOV	DI, 1020
1007	C6 C1 04	MOV	CL, 04
100A	8A 05	MOV	AL, [DI]
100C	E6 C0	OUT	C0, AL
100E	E6 CB	DEC	BL
1010	74 0D	JZ	101G
1012	C7 C2 10 10	MOV	DX, 1010
1016	4A	DEC	DX
1017	75 FD	JNZ	1016
1019	47	INC	DI
101A	E2 F0	LOOP	100A
101C	E9 E8 FF	JMP	1003
101F	F4	HLT	
1020	09 05 06 0A	TABLE	09 05 06 0A

RESULT:

Enter the above program execute it. By converting the required step in decimal to hex and entering the hex data at 1001 the motor for so much steps and then stops.

PROGRAMS IN 8086/88 LCD TRAINER

EXAMPLE-3: To run a stepper motor in both ports.

```
1000  C7  C7  20  10      MOV     DI, 1020
1004  C6  C1  04          MOV     CL, 04
1007  8A  05              MOV     AL, [DI]
1009  E6  C0              OUT     C0, AL
100B  E6  C8              OUT     C8, AL
100D  C7  C2  10  10      MOV     DX, 1010
1011  4A                  DEC     DX
1012  75  FD              JNZ     1011
1014  47                  INC     DI
1015  E2  F0              LOOP    1007
1017  E9  E8  FF          JMP     1000
1020  09  05  06  0A      TABLE   09  05  06  0A
```

RESULT:

Enter the above program starting from location 1000. Execute the same. The stepper motor rotates. Speed can be varied by varying the count at dx register pair. Direction can be varied by entering the data in the look-up table in the reverse order.

PROGRAMS IN 8086/88 LCD TRAINER

EXAMPLE-4: To run the stepper motor in both forward and reverse direction with delay.

```
1000  C6  C3  20           MOV       BI, 20
1003  C7  C7  3F  10       MOV       DI, 103F
1007  E8  1B  00           CALL      1025
100A  FE  CB               DEC       BL
100C  75  F5               JNZ       1003
100E  E8  2A  00           CALL      103B
1011  C6  C3  20           MOV       BL, 20
1014  C7  C7  43  10       MOV       DI, 1043
1018  E8  0A  00           CALL      1025
101B  FE  CB               DEC       BL
101D  75  75               JNZ       1014
101F  E8  19  00           CALL      103B
1022  E9  DB  FF           JMP       1000
1025  C6  C1  04           MOV       CL, 04
1028  8A  05               MOV       AL, [DI]
102A  E6  C0               OUT       C0, AL
102C  C7  C2  10  10       MOV       DX, 1010
1030  4A                   DEC       C0, AL
1031  75  FD               JNZ       1030
1033  47                   INC       DI
1034  E2  F2               LOOP      1028
1036  C3                   RET
1037  C7  C2  FF  FF       MOV       DX, 0FFFF
103B  4A                   DEC       DX
103C  75  FD               JNZ       103B
103E  C3                   RET
103F  09  05  06  0A       FORWARD   DATA
1043  0A  06  05  09       REVERSE   DATA
```

RESULT:

Enter the above program starting from location 1000 onwards. Execute the program after connecting the stepper motor in port 1. Execute the program. Now you can see that the stepper motor runs in forward and reverse direction continuously with a delay.

APPENDIX -F

SOFTWARE EXAMPLES IN 8031 ASSEMBLY LANGUAGE

EXAMPLE-1:

AIM:

TO RUN A STEPPER MOTOR AT DIFFERENT SPEED

```
4100                        ORG     4100H
4100  90 45  00  START:     MOV     DPTR, #4500H
4103  78 04                 MOV     R0, #04
4105  E0         JO:        MOVX    A, @DPTR
4106  C0 83                 PUSH    DPH
4108  C0 82                 PUSH    DPL
410A  90 FF  C0             MOV     DPTR, #FFC0H
410D  7A 04                 MOV     R2, #04H
410F  79 0F                 MOV     R1, #0FH
4111  7B 0F      DLY1:      MOV     R3, #0FH
4113  DB FE      DLY:       DJNZ    R3, DLY
4115  D9 FA                 DJNZ    R1, DLY1
4117  DA F8                 DJNZ    R2, DLY1
4119  F0                    MOVX    @DPTR, A
411A  D0 82                 POP     DPL
411C  D0 83                 POP     DPH
411E  A3                    INC     DPTR
411F  D8 E4                 DJNZ    R0, JO
4121  80 DD                 SJMP    START
4123                        END
4500  09, 05, 06, 0A TABLE: DB      09,05,06,0A
```

EXAMPLE-2:

AIM:

TO RUN A STEPPE MOTOR FOR REQUIRED ANGEL WITHIN 360°, WHICH IS EQUIVALENT TO 256 STEPS.

```
4100                         ORG      4100H
4100  7C 05                  MOV      R4, #FF       ;Hexdata for 360°
4102             START:
4102  90 41 14               MOV      DPTR, #LOOKUP
4105  78 04                  MOV      R0, #04
4107             JO:
4107  E0                     MOVX     A, @DPTR
4108  C0 83                  PUSH     DPH
410A  C0 82                  PUSH     DPL
410C  90 FF C0               MOV      DPTR, #FFC0H
410F  F0                     MOVX     @DPTR, A
4110  DC 06                  DJNZ     R4, CALL
4112             HLT:
4112  80 FE                  SJMP     HLT
4114             LOOKUP:
4114  09 05 06 0A            DB       09H, 05H, 06H, 0AH
4118             CALL:
4118  7A 03                  MOV      R2, #03
411A             DLY2:
411A  79 FF                  MOV      R1, #FFH
411C             DLY1:
411C  7B FF                  MOV      R3, #FFH
411E             DLY:
411E  DB FE                  DJNZ     R3, DLY
4120  D9 FA                  DJNZ     R1, DLY1
4122  DA F6                  DJNZ     R2, DLY2
4124  D0 82                  POP      DPL
4126  D0 83                  POP      DPH
4128  A3                     INC      DPTR
4129  D8 DC                  DJNZ     R0, JO
412B  80 D5                  SJMP     START
412D                         END
```

EXAMPLE - 3

AIM :

TO RUN A STEPPER AT DIFFRENT SPEED IN BOTH PORTS

4100				
4100	90 45 00	START :	MOV	DPTR, #4500H
4103	78 04		MOV	R0, #04H
4105	E0	J0 :	MOVX	A, @DPTR
4106	FC		MOV	R4, A
4107	C0 83		PUSH	DPH
4109	C0 82		PUSH	DPL
410B	90 FF C0		MOV	DPTR, #FFC0H
410E	F0		MOVX	@DPTR, A
410F	12 41 23		LCALL	DELAY
4112	90 FF C8		MOV	DPTR, #FFC8H
4115	EC		MOV	A, R4
4116	F0		MOVX	@DPTR, A
4117	12 41 23		LCALL	DELAY
411A	D0 82		POP	DPL
411C	D0 83		POP	DPH
411E	A3		INC	DPTR
411F	D8 E4		DJNZ	R0, J0
4121	80 DD		SJMP	START
4123		DELAY :		
4123	7A 04		MOV	R2, #04H
4125		L0 :		
4125	79 FF		MOV	R1, #FFH
4127		L1 :		
4127	7B FF		MOV	R3, #FFH
4129		L2 :		
4129	DB FE		DJNZ	R3, L2
412B	D9 FA		DJNZ	R1, L1
412D	DA F6		DJNZ	R2, L0
412F	22		RET	
4130			END	
4500	09, 05, 06, 0A	TABLE:	DB	09, 05, 06, 0A

PROGRAM FOR FORWARD & REVERSE ROTATION IN 8051

PAGE1

```
4100                                   ORG      4100H

                    $MOD52
4100  7C33          START :            MOV      R4, #33H
4102  904144        L2 :               MOV      DPTR, #FORWARD
4105  12411C                           LCALL    L1
4108  DCF8                             DJNZ     R4, L2
410A  124113B                          LCALL    DELAY
410D  7C33                             MOV      R4, #33H
410F  904148        L3 :               MOV      DPTR, # REVERSE
4112  12411C                           LCALL    L1
4115  DCF8                             DJNZ     R4, L3
4117  12413B                           LCALL    DELAY
411A  80E4                             SLMP     START
411C  7804          L1 :               MOV      R0, #04H
411E  E0            LOOP :             MOVX     A, @DPTR
411F  C083                             PUSH     83H
4121  C082                             PUSH     82H
4123  90FFC0                           MOV      DPTR, #OFFCOH
4126  7A04                             MOV      R2, #04H
4128  7905          L7 :               MOV      R1, # 05H
412A  7BFF          L6 :               MOV      R3, # 0FFH
412C  DBFE          L4 :               DJNZ     R3, L4
412E  D9F6                             DJNZ     R1, L6
4130  DAF6                             DJNZ     R2, L7
4132  F0                               MOVX     @DPTR, A
4133  D082                             POP      82H
4135  D083                             POP      83H
4137  A3                               INC      DPTR
4138  D8E4                             DJNZ     R0, LOOP
413A  22                               RET
413B  7D01          DELAY :            MOV      R5, #01H
413D  7A05          L9 :               MOV      R2, #01H
413F  DAFE          L8 :               DJNZ     R2, L8
4141  DDFA                             DJNZ     R5, L9
4143  22                               RET
4144  0905060A      FORWARD :          DB 09H, 05H, 06H, 0AH
414B  0A060509      REVERSE :          DB 0AH, 06H, 05H, 09H
```