

Controlled Rotation of the Shaft of a Stepper Motor with 8051 Microcontroller

Dr. D. Chinni Krishna

Associate Professor and HOD of Physics, Bhavan's New Science College, Narayanaguda, Hyderabad-500029, Telangana State, India

ABSTRACT: With a view to control the direction, speed, angle of rotation and Bidirectional rotation of the shaft of a stepper motor, the stepper motor was interfaced with 8051 microcontroller through 8255 PPI ports and rotating the shaft by supplying the required data by executing 8051 assembly language programs. The Hexadecimal data 07, 0B, 0D and 0E was given as inputs of stepper motor through amplifiers to rotate the shaft in a clock wise direction, while 07, 0E, 0D and 0B was given through the amplifiers to produce opposite magnetic poles to rotate the shaft of a stepper motor in an anti clock wise direction. The speed was controlled by setting different values of the registers used in the delay subroutine. The lesser values of delay registers will increase the speed of the shaft of the stepper motor. To rotate the shaft of a stepper motor with a specified angle, a separate register was used to supply the above datas for required times to produce the suitable angle. An effort were made to rotate the shaft of the motor in both clock wise and anti clock wise alternatively with a specific angle by combining the assembly language programs of both clock wise and anti clock wise direction rotations at one place with an appropriate count value.

Key Words: Stepper motor, 8051 microcontroller, Interfacing, Time delay generation, Assembly language program

I. INTRODUCTION

Stepper motor is a specially designed DC motor that translates electrical pulses in to mechanical movement by giving excitation pulses to the phase windings as given in [1]. They cannot be driven by just connecting the positive and negative leads of the power supply. They are driven by a stepping sequence which is generated by a controller as given in articles and projects [2-5]. The motor moves in steps according to this sequence. Stepper motors are used in industrial automation, Elevators [6], disk drives, dot matrix printers and robotics as given in [7] due to their ability to move in steps. The practical interfacing circuit diagram of the stepper motor through port C of 8255 Programmable peripheral interface and through 8051 was shown in Fig. 1.

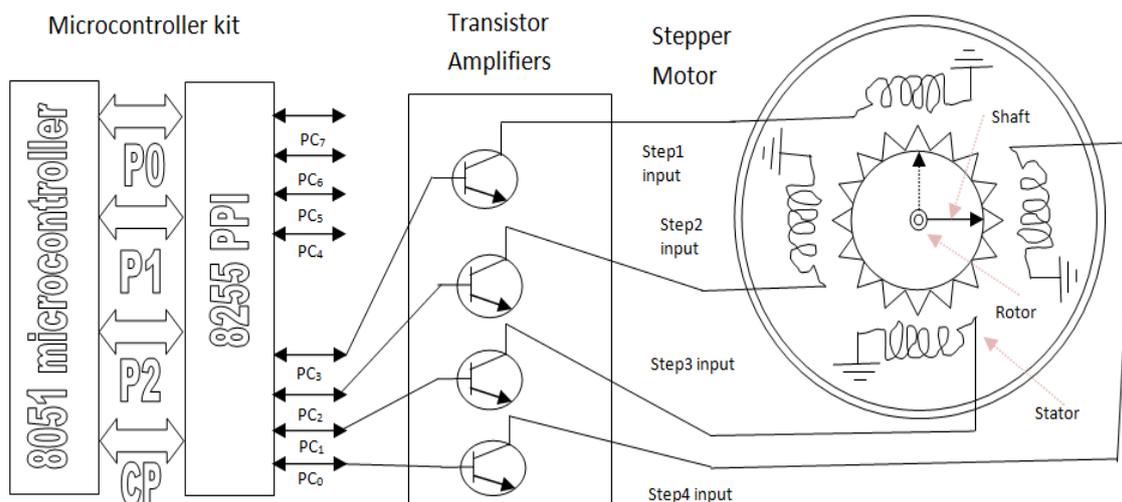


Fig. 1. Interfacing diagram of Stepper motor through transistor amplifiers and 8051 Microcontroller

II. WORKING PRINCIPLE OF STEPPER MOTOR

It is working on the principle of electromagnetic induction. The stepper motor consists of two major parts. They are stator and rotor. The stator is designed with induction coils of opposite phase, which acts as variable magnetic Pole (opposite coils have opposite phase by centre tapped ground, which was not shown in figure 2), while the rotor consists of many numbers of permanent magnetic teeth and each tooth is acting as a

magnetic Pole. If more the teeth, the angle taken by a step will be less. The direction of rotation is dictated by the electric pulses supplied to stator poles. The stator poles are determined by the current sent through the wire coils. As the direction of the current is changed, the polarity is also changed causing reverse rotation of the motor as given in a text book [8]. The required data to be transmitted through port C from the 8051 microcontroller along with their amplified output data for clock wise rotation was given in Table 1. Similarly to rotate the shaft in anti clock wise direction the data should be applied in the manner along with their amplified signals was given in Table 2. The higher byte was not connected to stepper motor; therefore its value was taken as 0 in all datas.

Table 1. Stepper motor code sequence to rotate the shaft in clock wise direction

PC3	PC2	PC1	PC0	Hex code	Step1 input	Step2 input	Step3 input	Step4 input
0	1	1	1	07	1	0	0	0
1	0	1	1	0B	0	1	0	0
1	1	0	1	0D	0	0	1	0
1	1	1	0	0E	0	0	0	1

Table 2. Stepper motor code sequence to rotate the shaft in anti clock wise direction

PC3	PC2	PC1	PC0	Hex code	Step1 input	Step2 input	Step3 input	Step4 input
0	1	1	1	07	1	0	0	0
1	1	1	0	0E	0	0	0	1
1	1	0	1	0D	0	0	1	0
1	0	1	1	0B	0	1	0	0



Fig 2. Physical appearance of interfacing diagram of stepper motor with 8051 microcontroller (a)with out executing the assembly language program and (b) with executing the program

To carry out the experiment to rotate the shaft of the stepper motor, the stepper motor trainer kit was purchased from the Physitech Electronics, Secunderabad, Telanagana state, India., and the 8051 microcontroller trainer kit was taken from Advanced Electronics, Banagalore, Karnataka, India., and the details were given in a lab manual [9]. The practical circuit diagram without executing and with executing the assembly language program was given in Fig. 2a&b. It was observed from Fig. 2a that no LED was glowing, but in Fig. 2b, three LEDs are in the glowing state and only one LED was in the off state and through it the required data was reached to the stepper motor coil. Sending a single data for longer time may produce heat in the coil and chance of damaging the circuit. Therefore precautions were taken to maintain smaller time delay between each data.

III. ASSEMBLY LANGUAGE PROGRAMS OF 8051 MICROCONTROLLER TO ROTATE THE SHAFT OF A STEPPER MOTOR CONTINUOUSLY IN A GIVEN DIRECTION (CLOCK WISE OR ANTI CLOCK WISE) WITH A SPECIFIED SPEED.

An assembly language program of 8051 was written as shown in Table 3 to transmit the data continuously to port C of 8255 PPI through 8051 to rotate the shaft of stepper motor in clock wise direction as specified in Table 1. The speed (1 rotation per minute, 2 rotations per minute, 3 rotations per minute etc.;;) can be controlled by the values of the registers used in the delay program as given in Table 8, which was written

from the memory address FF00H onwards. In the similar way an assembly language program of 8051 was written and shown in Table 4 to transmit Table 2 data continuously to rotate the shaft in an anti clock wise direction.

Table 3: Rotating the stepper motor shaft in a clock wise direction

Address	Label field	Mnemonic field	Comments field
8000		MOV A,#80	; (A) = 80H, Load control word in Accumulator
8002		MOV DPTR,#2023	; (DPTR) = 2023H, the control port address of 8255
8005		MOVX @DPTR,A	; Transfer control word to control port
8006		MOV DPTR,#2022	; (DPTR) = 2022H, the port C address of 8255
8009	REPEAT:	MOV A,#07	; (A) = 07, the data 1 to rotate the shaft of the stepper motor
800B		MOVX @DPTR,A	; Transfer data 1 to port C
800C		LCALL DELAY	; Call delay sub routine to generate time delay
800F		MOV A,#0B	; (A) = 0B, the data 2 to rotate the shaft of the stepper motor in clock wise direction
8011		MOVX @DPTR,A	; Transfer data 2 to port C
8012		LCALL DELAY	; Call delay sub routine to generate delay
8015		MOV A,#0D	; (A) = 0D, the data 3 to rotate the shaft of the stepper motor in clock wise direction
8017		MOVX @DPTR,A	; Transfer data 3 to port C
8018		LCALL DELAY	; Call delay sub routine to generate time delay
801B		MOV A,#0E	; (A) = 0E, the data 4 to rotate the shaft of the stepper motor in clock wise direction
801D		MOVX @DPTR,A	; Transfer data 4 to port C
801E		LCALL DELAY	; Call delay sub routine to generate time delay
8021		SJMP REPEAT	; Short jump to repeat the shaft rotation in clock wise

Table 4: Rotating the stepper motor shaft in an anti clock wise direction

Address	Label field	Mnemonic field	Comments field
8000		MOV A,#80	; (A) = 80H, Load control word in Accumulator
8002		MOV DPTR,#2023	; (DPTR) = 2023H, the control port address of 8255
8005		MOVX @DPTR,A	; Transfer control word to control port
8006		MOV DPTR,#2022	; (DPTR) = 2022H, the port C address of 8255
8009	REPEAT:	MOV A,#07	; (A) = 07, the data 1 to rotate the shaft of the stepper motor
800B		MOVX @DPTR,A	; Transfer data 1 to port C
800C		LCALL DELAY	; Call delay sub routine to generate time delay
800F		MOV A,#0E	; (A) = 0E, the data 2 to rotate the shaft of the stepper motor in an anti clock wise direction
8011		MOVX @DPTR,A	; Transfer data 2 to port C
8012		LCALL DELAY	; Call delay sub routine to generate time delay
8015		MOV A,#0D	; (A) = 0D, the data 3 to rotate the shaft of the stepper motor in an anti clock wise direction
8017		MOVX @DPTR,A	; Transfer data 3 to port C
8018		LCALL DELAY	; Call delay sub routine to generate time delay
801B		MOV A,#0B	; (A) = 0B, the data 4 to rotate the shaft of the stepper motor in anti clock wise direction
801D		MOVX @DPTR,A	; Transfer data 4 to port C
801E		LCALL DELAY	; Call delay sub routine to generate time delay
8021		SJMP REPEAT	; Short jump to repeat the shaft rotation in anti clock wise

IV. ROTATING THE SHAFT OF THE STEPPER MOTOR IN A FIXED GIVEN ANGLE:

The step angle is the minimum degree of degree of rotation associated with a single step. To complete one full rotation (360°), it takes 200 continuous steps in the given direction.

For 200 steps = 360° angle (one full circle)

For one step angle = 1.8°,

For four data one time = 7.2°,

To repeat these four data for 10 times gives an angle of 72° rotation in the specified direction. The number of times to be repeated these four data's can be taken as a count value and which will loaded in the register R3.

IV.1. Rotating the stepper motor shaft in clock wise direction with a given angle

The assembly language program of 8051 microcontroller to rotate the shaft of stepper motor in continuous clock wise direction will be slightly modified by introducing count value in R3 register before applying the data to the stepper motor. After completion of supplying all four datas to the stepper motor, R3 value will be decremented by 1. If R3≠0, then apply the four datas to repeat the stepper motor. If R3=0, then stop the program and check the position of the shaft to observe the angle of the shaft from its initial position. The assembly language program to achieve the angle of rotation was given in the Table 5.

Table 5: Assembly language program of 8051 Microcontroller to Rotate the stepper motor shaft in a clock wise direction with a given angle

Address	Label field	Mnemonic field	Comments field
8000		MOV A,#80	; (A) = 80H, Load control word in Accumulator
8002		MOV DPTR,#2023	; (DPTR) = 2023H, the control port address of 8255
8005		MOVX @DPTR,A	; Transfer control word to control port
8006		MOV DPTR,#2022	; (DPTR) = 2022H, the port C address of 8255
8009		MOV R3,#0D	; (R3) = 0DH, the value of count to call stepper motor datas for 13 times to get around 90° rotation
800B	REPEAT:	MOV A,#07	; (A) = 07, the data 1 to rotate the shaft of the stepper motor
800D		MOVX @DPTR,A	; Transfer data 1 to port C
800E		LCALL DELAY	; Call delay sub routine to generate 1 second delay
8011		MOV A,#0B	; (A) = 0B, the data 2 to rotate the shaft of the stepper motor in clock wise direction
8013		MOVX @DPTR,A	; Transfer data 2 to port C
8014		LCALL DELAY	; Call delay sub routine to generate 1 second delay
8017		MOV A,#0D	; (A) = 0D, the data 3 to rotate the shaft of the stepper motor in clock wise direction
8019		MOVX @DPTR,A	; Transfer data 3 to port C
801A		LCALL DELAY	; Call delay sub routine to generate 1 second delay
801D		MOV A,#0E	; (A) = 0E, the data 4 to rotate the shaft of the stepper motor in clock wise direction
801F		MOVX @DPTR,A	; Transfer data 4 to port C
8020		LCALL DELAY	; Call delay sub routine to generate 1 second delay
8023		DJNZ R3, REPEAT	; Decrement R3 and jump to REPEAT if R3≠0 to rotate the shaft for an angle of 90°
8025		LCALL 0003	; At memory 0003, program is available to end the assembly language program

IV.2. Rotating the stepper motor shaft in anti clock wise direction with a given angle

The assembly language program of 8051 microcontroller to rotate the shaft of stepper motor in continuous anti clock wise direction will be slightly modified by introducing count value in R3 register before applying the data to the stepper motor in anti clock wise direction. After completion of supplying all four datas to the stepper motor, R3 value will be decremented by 1. If $R3 \neq 0$, then apply the four datas again to the stepper motor to rotate in the same direction. If $R3=0$, then stop the program and check the position of the shaft to observe the angle of the shaft from its initial position. The assembly language program to achieve the angle of rotation was given in the Table 6.

Table 6: Assembly language program of 8051 to rotate the stepper motor shaft in an anti clock wise direction with a given angle

Address	Label field	Mnemonic field	Comments field
8000		MOV A,#80	; (A) = 80H, Load control word in Accumulator
8002		MOV DPTR,#2023	; (DPTR) = 2023H, the control port address of 8255
8005		MOVX @DPTR,A	; Transfer control word to control port
8006		MOV DPTR,#2022	; (DPTR) = 2022H, the port C address of 8255
8009		MOV R3,#0D	; (R3) = 0DH, the value of count to call stepper motor datas for 13 times to get around 90° rotation
800B	REPEAT:	MOV A,#07	; (A) = 07, the data 1 to rotate the shaft of the stepper motor
800D		MOVX @DPTR,A	; Transfer data 1 to port C
800E		LCALL DELAY	; Call delay sub routine to generate 1 second delay
8011		MOV A,#0E	; (A) = 0E, the data 2 to rotate the shaft of the stepper motor in an anti clock wise direction
8013		MOVX @DPTR,A	; Transfer data 2 to port C
8014		LCALL DELAY	; Call delay sub routine to generate 1 second delay
8017		MOV A,#0D	; (A) = 0D, the data 3 to rotate the shaft of the stepper motor in an anti clock wise direction
8019		MOVX @DPTR,A	; Transfer data 3 to port C
801A		LCALL DELAY	; Call delay sub routine to generate 1 second delay
801D		MOV A,#0B	; (A) = 0B, the data 4 to rotate the shaft of the stepper motor in anti clock wise direction
801F		MOVX @DPTR,A	; Transfer data 4 to port C
8020		LCALL DELAY	; Call delay sub routine to generate 1 second delay
8023		DJNZ R3, REPEAT	; Decrement R3 and jump to REPEAT if R3≠0 to rotate the shaft for an angle of 90°
8025		LCALL 0003	; End of the program

IV.3. Rotating the stepper motor shaft in clock wise and anti clock wise direction with a given angle

The assembly language program of 8051 Microcontroller to rotate the shaft of stepper motor in bidirectional rotation, both the programs of rotating the shaft in clock wise and Anti clock wise directions were mixed in a systematic manner as given in Table 7. The angle of rotation in both directions will be decided by the register value loaded in register R3. In both directions give same value for R3 to get equal angle of rotation in clock wise and anti clock wise rotations. To get unequal angle in each direction give unequal values for R3 register. The assembly language programs 8085 Microprocessor for similar type of clock wise and anti clock wise rotations of the shaft of stepper motor were given in a lab manual [10].

Table 7: Assembly language program of 8051 to Rotate the stepper motor shaft in clock wise & anti clock wise direction alternatively with a given angle

Address	Label field	Mnemonic field	Comments field
8000		MOV A,#80	; (A) = 80H, Load control word in Accumulator
8002		MOV DPTR,#2023	; (DPTR) = 2023H, the control port address of 8255
8005		MOVX @DPTR,A	; Transfer control word to control port
8006		MOV DPTR,#2022	; (DPTR) = 2022H, the port C address of 8255
8009	AGAIN:	MOV R3,#0D	; (R3) = 0DH, the value of count to call stepper motor datas for 13 times to get around 90° rotation
800B	REPEAT:	MOV A,#07	; (A) = 07, the data 1 to rotate the shaft of the stepper motor
800D		MOVX @DPTR,A	; Transfer data 1 to port C
800E		LCALL DELAY	; Call delay sub routine to generate 1 second delay
8011		MOV A,#0B	; (A) = 0B, the data 2 to rotate the shaft of the stepper motor in clock wise direction
8013		MOVX @DPTR,A	; Transfer data 2 to port C
8014		LCALL DELAY	; Call delay sub routine to generate 1 second delay
8017		MOV A,#0D	; (A) = 0D, the data 3 to rotate the shaft of the stepper motor in clock wise direction
8019		MOVX @DPTR,A	; Transfer data 3 to port C
801A		LCALL DELAY	; Call delay sub routine to generate 1 second delay
801D		MOV A,#0E	; (A) = 0E, the data 4 to rotate the shaft of the stepper motor in clock wise direction
801F		MOVX @DPTR,A	; Transfer data 4 to port C
8020		LCALL DELAY	; Call delay sub routine to generate 1 second delay
8023		DJNZ R3, REPEAT1	; Decrement R3 and jump to REPEAT1 if R3≠0 to rotate the shaft for an angle of 90°
8025		MOV R3,#0D	; (R3) = 0DH, the value of count to call stepper motor datas for 13 times to get around 90° rotation
8027	REPEAT2:	MOV A,#07	; (A) = 07, the data 1 to rotate the shaft of the stepper motor
8029		MOVX @DPTR,A	; Transfer data 1 to port C
802A		LCALL DELAY	; Call delay sub routine to generate 1 second delay
802D		MOV A,#0E	; (A) = 0E, the data 2 to rotate in an anti clock wise direction
802F		MOVX @DPTR,A	; Transfer data 2 to port C
8030		LCALL DELAY	; Call delay sub routine to generate 1 second delay
8033		MOV A,#0D	; (A) = 0D, the data 3 to rotate the shaft an anti clock wise
8035		MOVX @DPTR,A	; Transfer data 3 to port C
8036		LCALL DELAY	; Call delay sub routine to generate 1 second delay
8039		MOV A,#0B	; (A) = 0B, the data 4 to rotate in an anti clock wise direction
803B		MOVX @DPTR,A	; Transfer data 4 to port C
803C		LCALL DELAY	; Call delay sub routine to generate 1 second delay
803F		DJNZ R3, REPEAT2	; Decrement R3 and jump to REPEAT2 if R3≠0 to rotate the shaft for an angle of 90°
8041		SJMP AGAIN	; Short jump to AGAIN to repeat this bidirectional rotation process

Table 8: The Delay Routine

Address	Label field	Mnemonic field	Comments field
FF00	DELAY:	MOV R0,#01	; Load R0 register with 01
FF02	LOOP3:	MOV R1,#0FF	; Load R1 register with FFH
FF04	LOOP2:	MOV R2,#0FF	; Load R2 register with FFH
FF06	LOOP1:	DJNZ R2,LOOP1	; Decrement R2 by 1 and if it is not equal to 0 jump to LOOP1 (inner loop)
FF08		DJNZ R1,LOOP2	; Decrement R1 by 1 and if it is not equal to 0 jump to LOOP2 (middle loop)
FF0A		DJNZ R0,LOOP3	; Decrement R0 by 1 and if it is not equal to 0 jump to LOOP3 (outer loop)
FF0C		RET	; Return to the main program

V. CONCLUSIONS

1. Stepper motor translates electrical pulses in to mechanical movement by giving excitation pulses to the phase windings through 8051 Microcontroller.
2. The datas 07, 0B, 0D, 0E were applied continuously as inputs of stepper motor continuously with a small time delay in between the datas to rotate the shaft of the stepper motor in a clock wise direction and apply them in reverse order to rotate the shaft in an anti clock wise direction.
3. The speed of the rotation will be decided by the values loaded in registers used in time delay routine.
4. The step angle must be determined to rotate the shaft to a fixed angle.
5. One of the register must be loaded with a count value to repeat the four datas to rotate the shaft of stepper motor with some angle in a given direction.
6. Clock wise rotation and anti clock wise rotation were mixed in a single program to rotate the shaft of a stepper motor in bidirectional mode.

ACKNOWLEDGEMENTS

The author would like to express his sincere thanks to the Management of Bhavan's New Science College, Hyderabad, for their encouragement and support to undertake the present work and providing the facilities to interfacing of the stepper motor with 8051 Microcontroller, writing and executing of the assembly language programs in Final year of M.Sc. Applied Electronics Laboratories of Department of Physics. He also like to express his thanks to his colleagues, friends and family members for their support. Finally, the author would like to express his M.Sc. Applied Electronic students for interfacing, writing and executing the assembly language programs of 8051 microcontroller.

REFERENCES

- [1] Stepper Motors – Introduction and Working Principle published in embedded journal.com
- [2] Interfacing stepper motor with microcontroller [Stepper Motor Tutorial From Rikipedia Embedded Wiki.
- [3] Stepper Motor Control using 8051 Microcontroller OCTOBER 12, 2015, Electronics Hub. projects | tutorials | courses.
- [4] Interfacing Stepper Motor with 8051 using Keil C – AT89C51 By Ebin George 8051 Microcontroller, Tutorials
- [5] Interfacing of Stepper Motor, electrical4u.com
- [6] Poorvi Behre, Viveka Nema, Bhupendra Badonia, Int. J. of Sci. & Eng. Res. Vol. 4, Jan 2013.
- [7] Working with stepper motors, from *robot builder's bonanza, 2nd edition*, 2001 see www.robotoid.com
- [8] Muhammad Ali Mazidi, Jainice Gillispie Mazidi and Rolin D. McKinlay; The 8051 Microcontrollers and Embedded Systems using Assembly and C, Second Edition, Eastern Economy Edition-2006, Prentice Hall of India Private Limited, New Delhi-110001.
- [9] ALS-SDA-8051-MEL User's Manual, Advanced Electronics, Banagalore-560058, Karnataka, India.
- [10] Dr. D. Chinni Krishna and Prof. G. Pushpa Chakrapani
M. Sc. Physics, Laboratory Manual & Record on Memory Devices & Microprocessors of Dr.B.R. Ambedkar Open University,
Jubilee Hills, Hyderabad, Telangana State, India