Speed and Position Control of Stepper Motor using 8051 Microcontroller

Amit Kumar[1] Ramjee Prasad Gupta[2]
[1]PG Scholar [2]Assistant Professor
[1,2]Department of Electrical Engineering
[1,2]BIT Sindri, Jharkhand, India

Abstract— Speed control of machine is the most vital and important part in any industrial organization. This research work focuses on speed and position control of stepper motor using 8051 microcontroller. The speed of stepper motor has been controlled by the delay between consecutive pulses of stepper motor phases provided by microcontroller. Actually speed is inversely proportional to delay between pulses of consecutive phases and Position of stepper motor has been controlled by the number of revolutions. In this project work step angle of 15 degree per step has been taken when motor is moving in clockwise direction and 7.5 degree per step when moving in anticlockwise direction. The direction of movement of stepper motor depends on opening and closing of switch. When switch is closed, motor moves in clockwise direction and when switch is opened, motor moves in anticlockwise direction. Furthermore, variable resistor connected to pin no. 6 of ADC has been varied in clockwise direction due to which speed increases and when varied in anticlockwise direction speed decreases. In this project work infrared sensor has been used to detect speed of stepper motor in rpm. Speed in rpm & step angle in degree has been displayed on LCD screen and output pulses can be observed on CRO. The control program has been written in assembly language and Kiel compiler has been used to convert this control program into executable file or in a HEX code. This hex code has been burnt into microcontroller by Flash Magic software.

Key words: P89C51RD2 Microcontroller, Darlington pair driver for stepper motor, ADC0804, LCD, LED, Switch

I. Introduction

Stepper motors with microcontrollers are used in numerous applications where there is need of controlling the motion. This is due to robust structure of stepper motor, absence of brushes and better control capability when they are used with microcontrollers. These motors have excellent response to starting, stopping and reversing. The microcontroller is used for sending pulses to the stepper motor drive. Since microcontroller is not able to provide sufficient current to drive stepper motor so stepper motor driver is used. The microcontroller is used both to control the speed as well as position of stepper motor. The speed of stepper motor is directly proportional to the frequency of drive input pulses which is controllable with the help of microcontrollers and the rotation is proportional to the no of output pulses. In this project we are controlling speed by providing delay by analog to digital converter and direction is controlled by a switch.

II. Methodology

The project work has been divided into two parts. In the first part simulation is done using proteus simulation software and in second part waveform of input pulse is verified by hardware.

The system is designed using Philips (P89C51RD2) 8-bit microcontroller. In this system a bipolar stepper motor having 6-wires arrangement has been used as shown in fig.1 with following specification: Maximum voltage 12v Dc, Step angle: 7.5 degree, Steps per revolution: 48. Here centre tap arrangement has been taken into consideration. In this type of arrangement common wires are connected to 12v supply and the two ends of each winding are alternatively grounded to reverse the direction of the field provided by that winding.

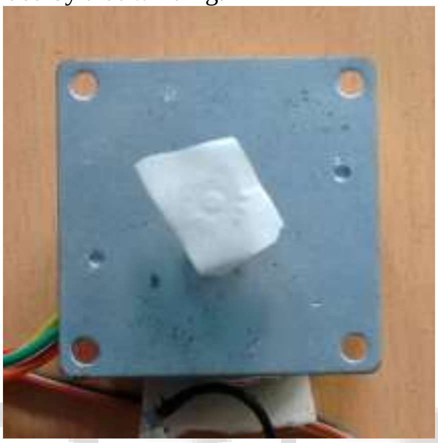

Fig. 1 Bipolar stepper motor

Crystal oscillator of frequency 11.059 MHZ has been used to produce clock frequency for microcontroller. Since microcontroller is unable to provide sufficient current to stepper motor hence L293D driver has been used in software implementation. However in hardware model darlington pair driver has been used [4].

The 8-pin (pin no. 11 to 18) of ADC0804 has been connected to port 0 of microcontroller and a variable resistor is connected to pin no. 6 of ADC to provide pulse delay stepper motor [3].

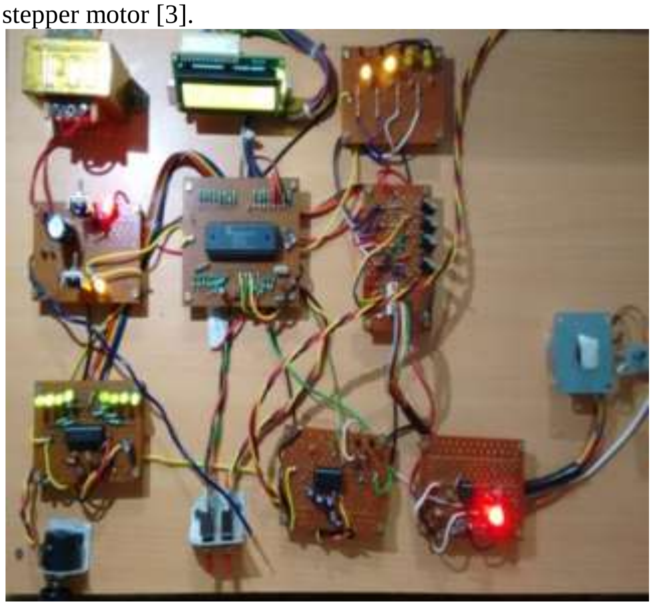

Fig. 2: Microcontroller based stepper motor hardware model

To display direction of movement, step angle and total no. of step moved by motor a 16*2 character LCD has been used. The 8-pin (pin no. 7 to 14) of LCD has been connected to port 1 and pin no. 4,5& 6 of LCD are connected to P3.4, P3.6 & P3.7 of microcontroller.

To control direction of movement of motor a switch has been connected to p3.2 of microcontroller.

P2.0 to P2.3 of microcontroller has been used to send control signal to stepper motor. Fig. 2 shows hardware of microcontroller based stepper motor [2].

III. STEPPING SEQUENCE

Stepper motors can be driven in two different pattern:

– Full step sequence
– Half step sequence

A. *Full step*

In this type of sequence two coils are energized at the same time and motor shaft rotates accordingly. The stepping sequence for clockwise and anticlockwise movement is given in table 1.

Clockwise Rotation				Anticlockwise Rotation			
Coil A	Coil B	Coil C	Coil D	Coil A	Coil B	Coil C	Coil D
1	0	0	1	0	0	1	1
1	1	0	0	0	1	1	0
0	1	1	0	1	1	0	0
0	0	1	1	1	0	0	1

Table 1: Full Step

B. *Half Step*

It is the combination of full drive and half drive. In this mode one coil and two coils are energized at a time alternatively [1][5]. Stepping sequence for clockwise and anticlockwise movement is given in table 2.

Clockwise Rotation				Anticlockwise Rotation			
Coil A	Coil B	Coil C	Coil D	Coil A	Coil B	Coil C	Coil D
1	0	0	1	0	0	0	1
1	0	0	0	0	0	1	1
1	1	0	0	0	0	1	0
0	1	0	0	0	1	1	0
0	1	1	0	0	1	0	0
0	0	1	0	1	1	0	0
0	0	1	1	1	0	0	0
0	0	0	1	1	0	0	1

Table 2: Half Step

IV. SIMULATION USING SOFTWARE

A. *Introduction:*

To implement this project work three software have been used. These are:

– Kiel
– Proteus
– Flash Magic

1) *Kiel*

Kiel compiler has been used to convert high level language into Hex code.

2) *Proteus*

It has been used to simulate the result in software.

3) *Flash Magic*

It has been used to burn Hex code into microcontroller.

B. *Stepper Motor Without Switch*

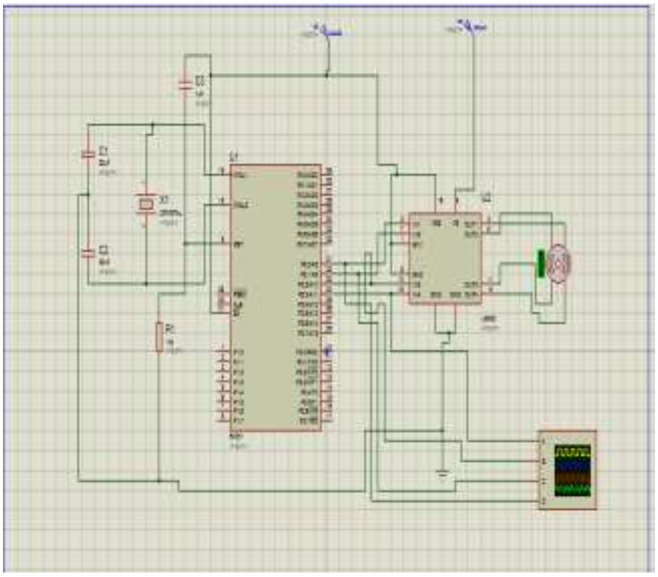

Fig. 3: Circuit diagram of microcontroller based stepper motor

1) *Simulation Result using Proteus*

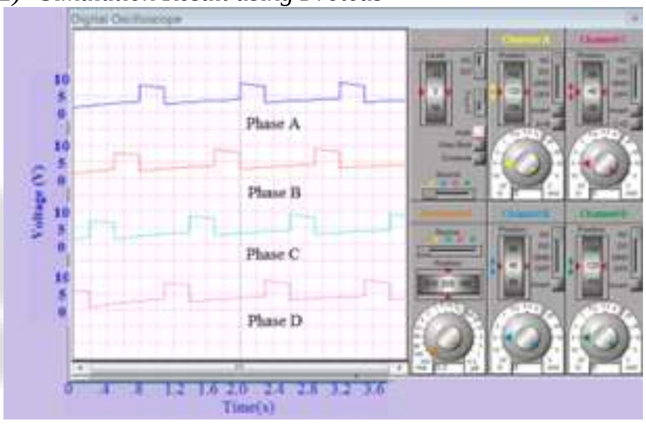

Fig. 3: waveform of input pulse given to stepper motor for anticlockwise movement

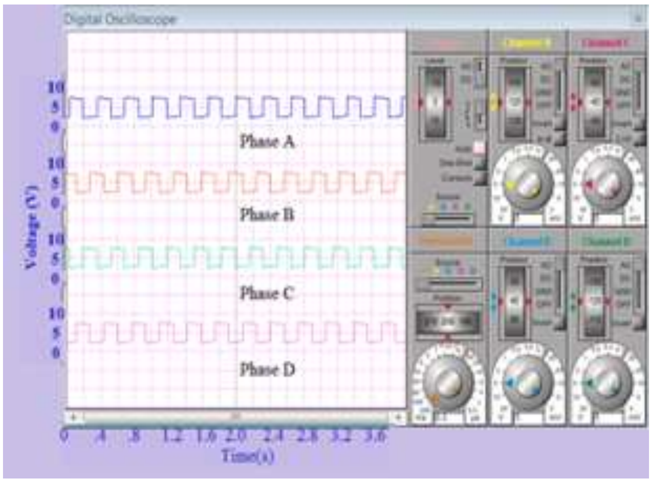

Fig. 4: waveform of input pulse given to stepper motor for anticlockwise movement

Observation: The on-time pulse width of fig. 3 is more than fig. 4, hence speed of stepper motor is higher for fig. 4.

C. Stepper Motor with Switch

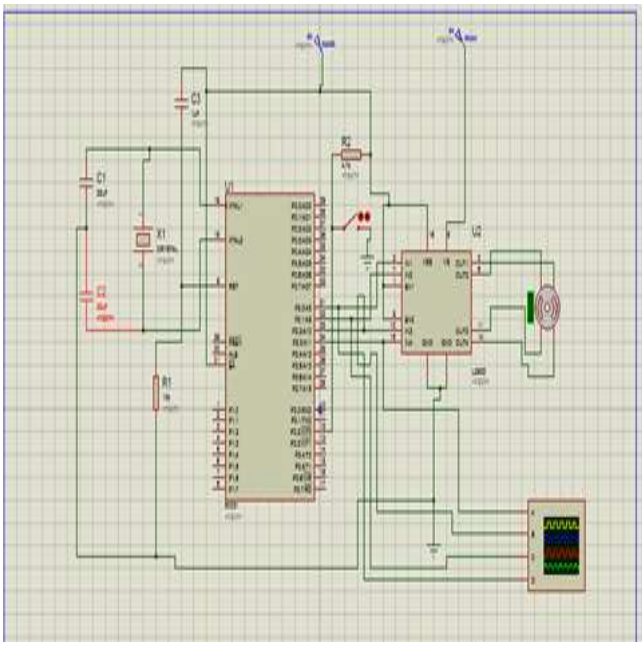

Fig. 5: Circuit diagram of stepper motor with switch

1) Simulation result using proteus

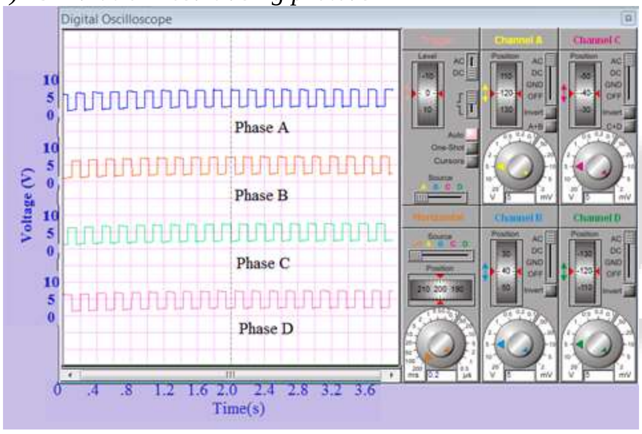

Fig. 6: waveform of input pulse given to stepper motor for anticlockwise movement

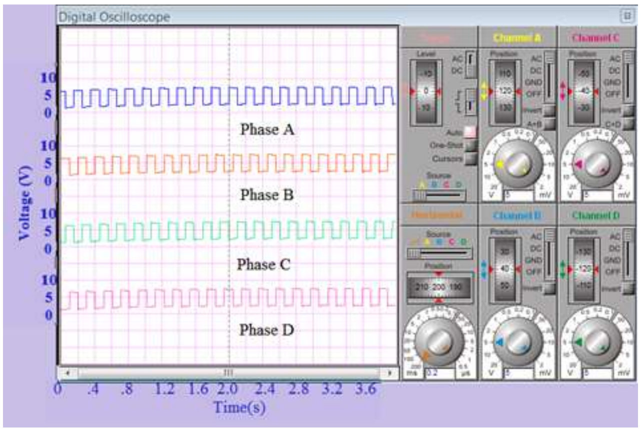

Fig. 7: waveform of input pulse given to stepper motor for clockwise movement

Observation: In fig. 6 Input has been given as 1100, 1001, 0011 and 0110 to Phase A, B, C and D of stepper motor respectively, hence movement of motor is anticlockwise. On other hand, in fig. 7 input has been given as 1100, 0110, 0011 and 1001 to Phase A, B, C and D respectively, hence movement of motor is clockwise.

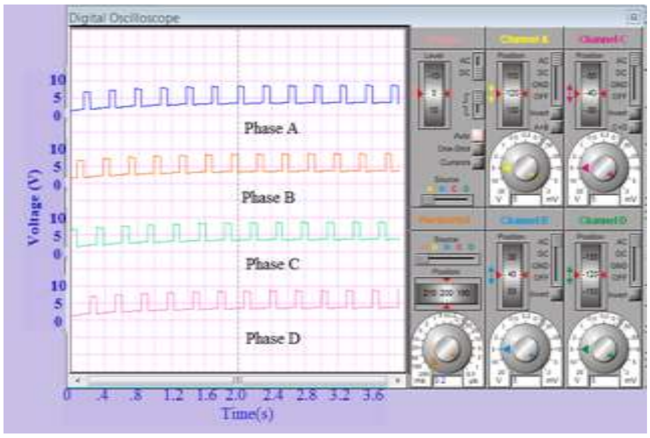

Fig. 8: waveform of input pulse given to stepper motor for anticlockwise movement

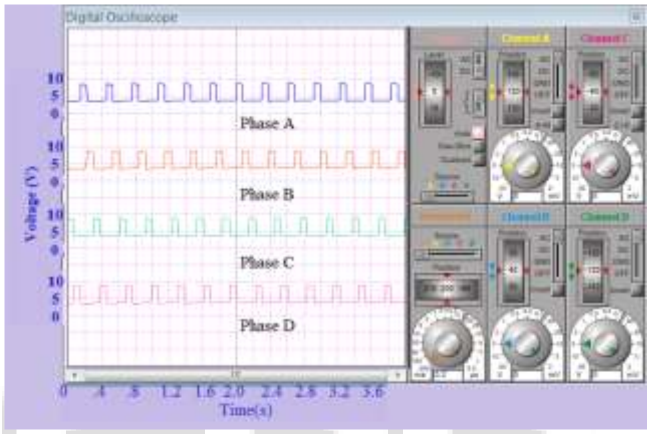

Fig. 9: waveform of input pulse given to stepper motor for clockwise movement

Observation: From fig. 6 and fig. 8, it has been observed that on time Pulse width of fig. 6 is more than that of fig. 8, hence speed of motor will be less for fig. 6.

D. Stepper motor with switch and LCD display

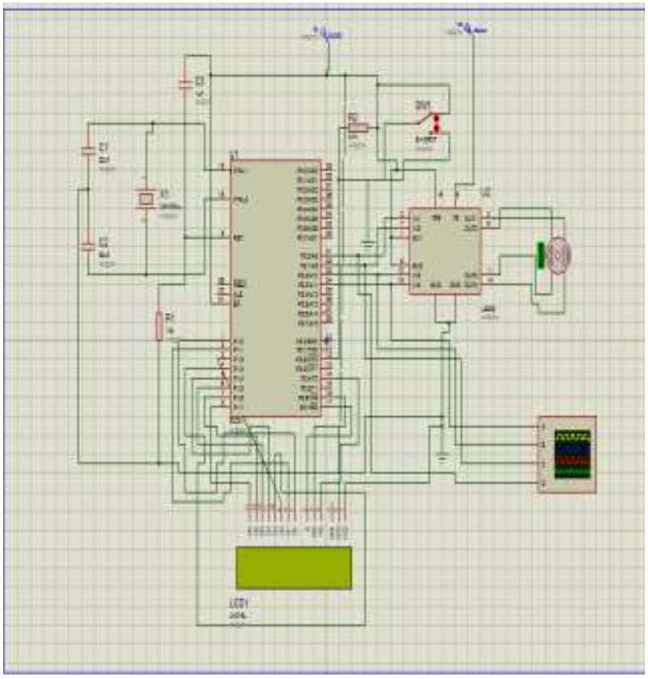

Fig. 10: Circuit diagram of stepper motor with switch and LCD display

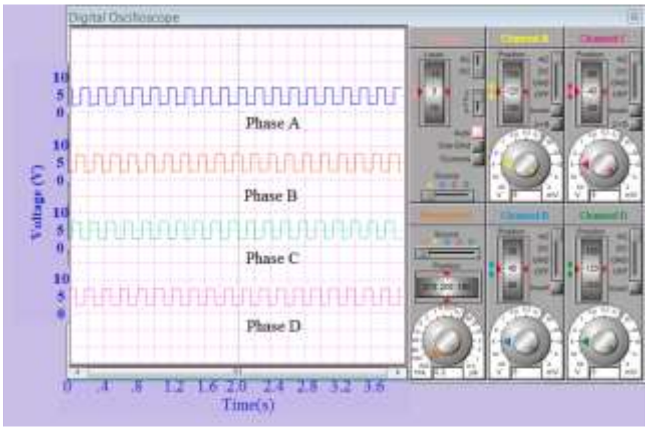

Fig. 11: waveform of input pulse given to stepper motor for anticlockwise movement

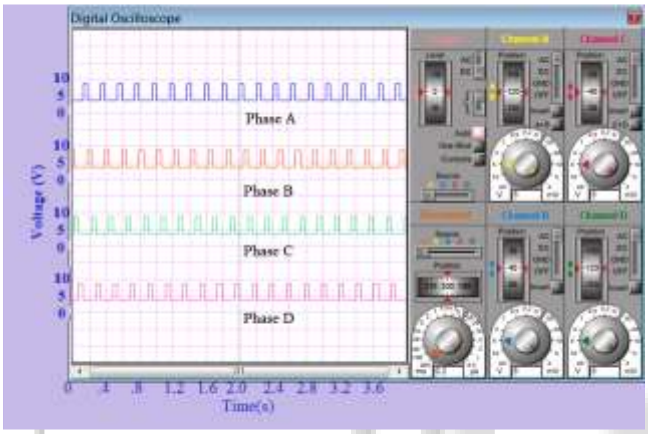

Fig. 12: waveform of input pulse given to stepper motor for clockwise movement

Observation: From fig. 11 and fig. 12 it has been observed that duty cycle for on time for fig. 11 and fig. 12 is not same, this is due to different step angles in two cases. For clockwise movement of stepper motor steep angle is 15 degree and that of for anticlockwise movement step angle is 7.5 degree.

V. EXPERIMENTAL RESULT

A. Anticlockwise Movement of Stepper Motor (Speed=50 rpm)

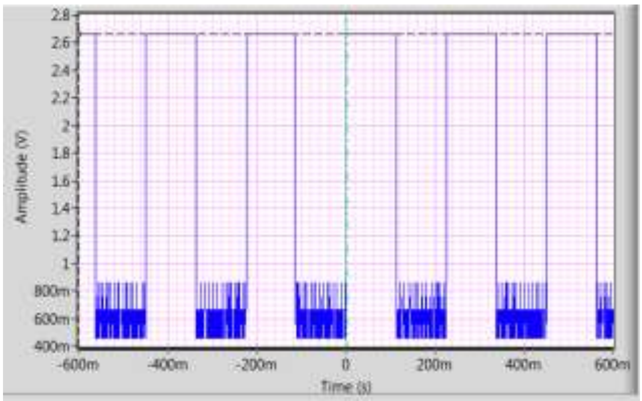

Fig. 13: Waveform of input pulse given to phase A of stepper motor

Parameter	Value	Unit
Maximum Voltage	2.20	V
Minimum Voltage	0.00	V
Peak to Peak Voltage	2.20	V
High voltage	2.20	V
Low Voltage	0.00	V
Amplitude	2.20	V
Average Voltage (On time)	1.00	V
RMS Voltage (Off time)	1.40	V
Period	225.00	Ms
Frequency	4.43	Hz
Cycle Average Voltage	1.20	V
Cycle RMS Voltage	1.40	V
Duty cycle (On time)	49.89	Percent
Duty Cycle (Off time)	50.11	Percent
On time Pulse Width	112.50	Ms

Table 3: Measurement data for Phase A

B. Clockwise movement of stepper motor (Speed=100 rpm)

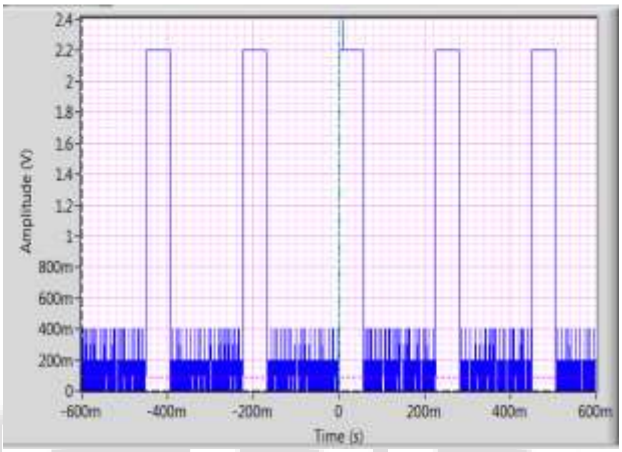

Fig. 14: Waveform of input pulse given to phase A of stepper motor

Parameter	Value	Unit
Maximum Voltage	2.20	V
Minimum Voltage	0.00	V
Peak to Peak Voltage	2.20	V
High voltage	2.20	V
Low Voltage	0.00	V
Amplitude	2.20	V
Average Voltage (On time)	.600	V
RMS Voltage (Off time)	1.00	V
Period	225.50	Ms
Frequency	4.43	Hz
Cycle Average Voltage	.600	V
Cycle RMS Voltage	1.00	V
Duty cycle (On time)	25.06	Percent
Duty Cycle (Off time)	74.94	Percent
On time Pulse Width	56.50	Ms

Table 4: Measurement data for Phase A

Observation: From fig. 13 and fig.14 it has been observed that pulse widths for anticlockwise and clockwise movement are not same. This is due to fact that step angle for movement in both directions is not same. Here 15 degree is taken as step angle for clockwise movement and that of for anticlockwise is 7.5 degree. The time to move one step in anticlockwise direction is same as the time to 2 step in clockwise direction, hence speed in case of clockwise movement is more.

C. Comparison between Simulated System and Prototype Development Model

Micro-controller	80C51	P89C51RD2	P89C51Rd2 belongs to 80C51 microcontroller family.
Stepper motor	4-wire	6-wire	Two extra wire of 6-wire stepper motor have been connected to 5v.
Stepper motor driver	L293D	Darlington pair Transistor	L293D driver supply current upto 500mA while darlington pair upto 1A.
LCD	16*2 character	16*2 character	Used to display direction of movement, step angle and no. of steps moved by stepper motor.
LED	Not used	Used	Showing bit patterns of 4-phases.
ADC	Not used	ADC0804	The set voltage is converted into corresponding 8-bit digital output.
Variable resister	Not used	Used	To control the speed of stepper motor.
Serial Interface unit	Not used	Max232	To establish communication between Microcontroller & PC.
IR sensor	Not used	Used	To sense rpm of stepper motor.
Switch 1	Not used	SPDT	In programming mode Hex code is burnt into microcontroller IC and in normal mode circuit works according to hex code burnt into microcontroller.
Switch 2	On/off	SPDT	SPDT switch is more reliable.

Table 5: Comparison between Simulated System and Prototype Development Model

VI. CONCLUSION

This research work focuses on the speed control of stepper motor using 8051 microcontroller which has been simulated using proteus software and later its performance has been investigated using hardware model. On the basis of experimental result it has been concluded that more precise and accurate speed control has been achieved by stepper motor which is most important for industrial applications. The speed of the motor is inversely proportional to the pulse delay given to the four phases of stepper motor.

REFERENCES

[1] Mazidi Muhammad Ali "The 8051 Microcontroller and Embedded Systems", Prentice Hall of India, New Delhi, India, pp. 492-507, 2012.
[2] Khalaf Azzeddine Ferrah and Jehad AL Bani Younes "Sensorless speed and position estimation in a stepper motor". IEEE transaction on industrial electronics, pp. 66-78, March 2010.
[3] Chin Ming Hsu and Hui Mei Chao "Constructing microcontroller teaching tool with the integration of hardware and software technology". International conference on new trends in information and service science, pp. 270-276, 2009.
[4] Peter Crnosija "Microcontroller implement of optional algorithms for closed loop control of hybrid stepper motor driver". IEEE transaction on industrial electronics, Vol. 47, No. 6, pp.1319-1324, 2000.
[5] Rass Bannatyne "Introduction to microcontrollers". IEEE transaction on education, pp. 250-254, Fab. 1998.

