MICROCONTROLLER BASED STEPPER MOTOR CONTROL SYSTEM

Kyi Kyi Khaing

Faculty of Computer System and Technology,

Myanmar Institute of Information Technology (MIIT), Mandalay

akyitaw07@gmail.com

ABSTRACT

The microcontroller based stepper motor control system controls the motor and any motor device according to its requirement for any industrial application. At the heart of the circuit is the AT mega 32 microcontroller which controls all its functions. The stepper motor can move a precise angle; the motor's position can be controlled without any feedback mechanism. Microcontroller can be used to apply different control signals to the motor to make it rotate according to the need of application. In this paper, we are going to rotate a stepper motor by using an AT mega 32 microcontroller. AT mega 32 has been programmed in order to rotate the stepper motor for various applications. The motor drive which acts as an interface between the microcontroller and the stepper motor. The microcontroller board is applying the external voltage of about 5V. From this it can be directly deliver the input to the stepper motor drive where it provides sufficient current to the stepper motor which tends to rotate the stepper motor. By programming the stepper drive, we can be able to control the speed of the stepper motor, direction of the stepper motor either in clockwise or anticlockwise direction. Motor drive protects the stepper motor from damages so that the stepper motor drive can be connected between the stepper motor and AT mega 32 microcontroller.

KEYWORDS

AT mega 32, Microcontroller, Stepper Motor, Preset, Virtual Terminal

1. INTRODUCTION

There are various AT mega microcontroller used for many applications. In this paper, we are going to use AT mega 32 microcontroller. AT mega 32 microcontroller has different numbers of analog and digital inputs. A specially provided USB cable which acts as an interface between the microcontroller and the computer. From the USB cable, we upload the program to the microcontroller. Based on the program done in the microcontroller, we can use the microcontroller for various applications such as rotating and to control the speed of the motor. The stepper motor is being connected with the digital input of the circuit. Reset button is provided in the microcontroller board to reset the program and also to upload the other program [1].

A stepper motor is a brushless, synchronous electronic motor that converts electrical pulses into mechanical movement. Every revolution of the stepper motor is divided into a discrete number of steps, and the motor must be sent a separate pulse for each step. The stepper motor can only a precise angle, the motor's position can be controlled without any feedback mechanism. As the electrical pulses increases in frequency, the step movement changes into continuous rotation, with the speed of rotation directly proportional to the frequency of the pulses. Step motors are used every day in both industrial and commercial applications because of their low cost, high reliability, high torque at low speeds and a simple, rugged construction that operates in almost any environment [2].

Stepper motor is a DC motor that divides the full rotation angle of 360° into number of equal steps. The motor is rotated by applying certain sequence of control signals. The speed of rotation can be changed by changing the rate at which the control signals are applied. Various stepper motors with different step angles and torque ratings are available in the market. Microcontroller can be used to apply different control signals to the motor to make it rotate according to the need of application [3].

2. IMPLEMENTATION OF THE DESIGN

This design is based on both hardware and software. For the design to be implemented, we will be using an ATMega32 microcontroller, interface with some other hardware components. The functional block diagram of the design is shown in Figure 2.1.

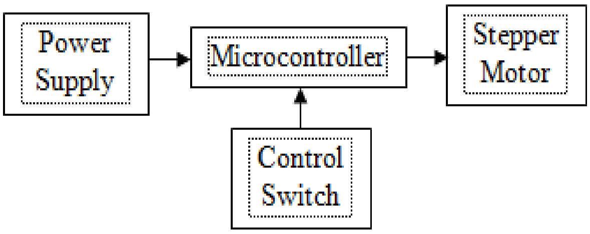

Figure 2.1 Functional Block Diagram of the Design

2.1 Types of Stepper Motor

Stepper motor is mainly classified into two types. These are unipolar stepper motor and bipolar stepper motor.

2.1.1 Unipolar Stepper Motor

A unipolar stepper motor has one winding with centre tap per phase. Each section of windings is switched on for each direction of magnetic field. Since in this arrangement a magnetic pole can be reversed without switching the direction of current, the communication circuit can be made very simple (e.g., a single transistor for each winding. Typically, given a phase, the centre tap of each winding is made common: giving three leads per phase and six leads for a typical two phase motor. Often, these two phase commons are internally joined, so the motor has only five leads.

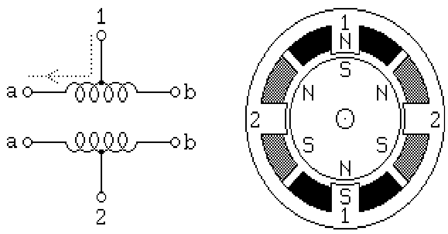

Figure 2.2 Unipolar Stepper Motor Windings

2.1.2 Bipolar Stepper Motor

The bipolar stepper motor has four wires coming out of it. Unlike unipolar steppers, bipolar steppers have no common centre connection. They have two independent sets of coils instead. We can distinguish them from unipolar steppers by measuring the resistance between the wires. We can identify the separate coils by touching the terminal wires together. If the terminals of a coil are connected, the shaft becomes harder to run.

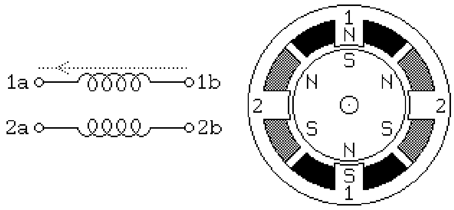

Figure 2.3 Bipolar Stepper Motor Windings

2.2 Step Sequence

Stepper motors can be driven in two different patterns or sequences. These are full step sequence and half step sequence [4].

2.2.1 Full Step Sequence

In the full step sequence, two coils are energized at the same time and motor shaft rotates. The order in which coils has to be energized is given in the Table 2.1.

Table2.1 Full Mode Sequence

Step	A	B	C	D
1	1	0	0	1
2	1	1	0	0
3	0	1	1	0
4	0	0	1	1

2.2.2 Half Step Sequence

In Half mode step sequence, motor step angle reduces to half the angle in full mode. So, the angular resolution is also increased i.e. it becomes double the angular resolution in full mode. Also in half mode sequence the number of steps gets doubled as that of full mode. Half mode is usually preferred over full mode. Table 2.2 shows the pattern of energizing the coils.

Table 2.2 Half Mode Sequence

Step	A	B	C	D
1	1	0	0	1
2	1	0	0	0
3	1	1	0	0
4	0	1	0	0
5	0	1	1	0
6	0	0	1	0
7	0	0	1	1
8	0	0	0	1

2.3 Step Angle

Step angle of the stepper motor is defined as the angle traversed by the motor in one step. To calculate step angle, simply divide 360 by number of steps a motor takes to complete on evolution. In half mode, the number of steps taken by the motor to complete one revolution gets doubled, so step angle reduces to half. Stepper motor rotating in full mode takes 4 steps to complete a revolution, so step angle can be calculated as:

$$\text{Step Angle } \emptyset = 360°/4 = 90°$$

So, the half mode step angle gets half so 45 degree.

3. Operation of the System

According to the circuit, the connections are done by connecting the stepper motor with the microcontroller interfaced by the stepper motor drive. Input and the output connections are properly done and therefore the connections are to be made easier by connecting using jumper cables. Program has been uploaded in the microcontroller. The power supply of the circuit is connected. When the 5V pin is connected with the motor manually, the microcontroller board gets triggered and further it sends the signal to the stepper motor drive. The stepper motor drive which amplifies the signal and sends to the stepper motor. The stepper motor receives the proper current and voltage and therefore the stepper motor can be rotated easily. By editing the program, the speed and direction of the stepper motor can be controlled.

3.1 Circuit Design

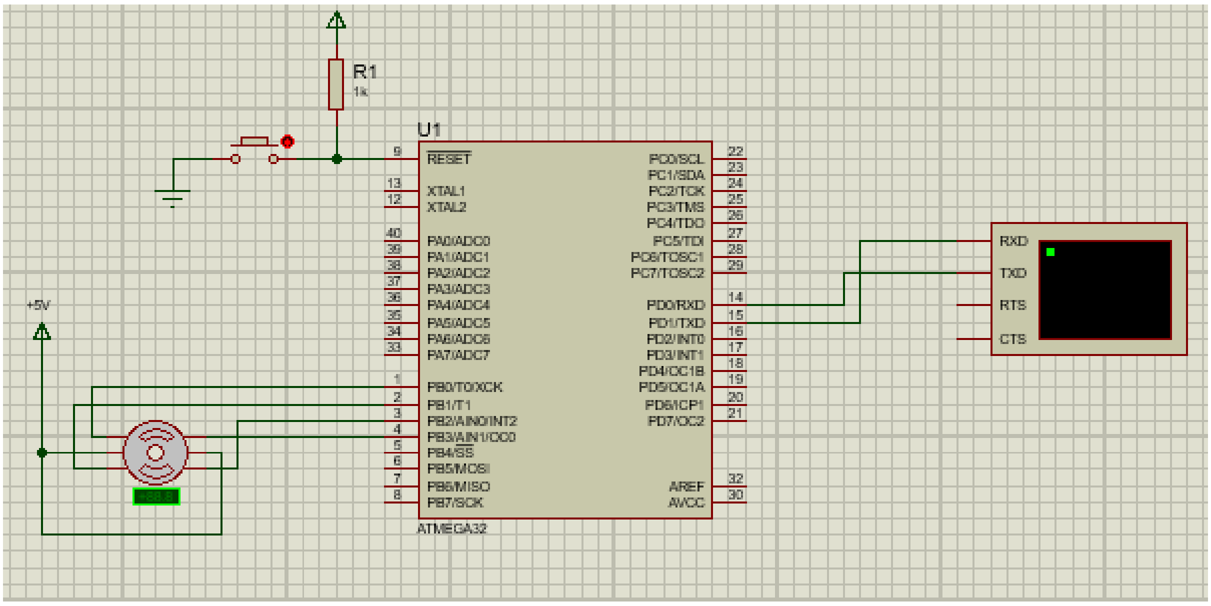

Figure 3.1: Schematic of the Design

3.2 Interfacing Stepper Motor with AT mega 32

In this paper, we are going to interface 6 wires unipolar stepper motor with ATmega32 microcontroller. Only four wires are required to control stepper motor. Two common wires of stepper motor are connected to 5V supply. Stepper motor is connected with PB0, PB1, PB2 and PB3 (PORTB) at AT mega 32 microcontroller. For changing direction of rotation, a reset pin is used (Pin 9), which is connected between the microcontroller and ground. Here, we use the simulation with the microcontroller and open the stepper motor using the password as shown in Figure 3.2. If we want to quit this project, we will press ‘q’ as shown in Figure 3.3.

3.3 Software and Coding

The following software's are used for programming and feeding in ATmega32 microcontroller.

1. Atmel Studio 7: Atmel Studio 7 is the development platform. Atmel studio is required to write the C-code and generate its HEX code.
2. Proteus 8 Professional: It is used to draw the design.

Coding

```
#define  F_CPU8000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <string.h>

#define baud9600
#define ubrr_value((F_CPU/(16UL*baud))-1)

volatile int temp,temp1;
char pw[]="1234";                //Password to open the Gate

//Function for USART
void usart_init();
void usart_tx(chardata);
unsigned char usart_rx();
void usart_tx_string(char*string);

//Function for Unipolar Stepper Motor
void motor();

int main(void)
{
        DDRD |= 0xff;            //Set the direction of TxD pin as output
        DDRD &= ~(1<<0);     //Set the direction of RxD pin as input
        DDRB = 0xff;            //Set the direction of Unipolar Stepper Motor

        usart_init();            //Initialize the USART
        int i =0;

        usart_tx_string("Unipolar Stepper Motor - Project\rVietnamese Group");
        _delay_ms(500);

        while(1)
        {
                        usart_tx(12); //Clear Virtual Terminal Screen
                        int count=0;
                        usart_tx_string("Enter password to open Gate");
                        for i=0;i<strlen(pw);i++)
                        {
                                temp = usart_rx();
                                usart_tx(temp);
                                if(temp != pw[i])
                                {
                                        count++;
                                }
                                if(temp==13)
                                        break;
                        }
                        if((count==0)&&(i=strlen(pw)))
                        {
```

```
                        usart_tx_string("\rGate is openning...");
                        motor ();    //Open the Gate
                    }
                    else
                    {
                        usart_tx_string("\r Password is not correct");
                    }
                    usart_tx_string("\r If you want to quit please press 'q'...");
                    temp1=usart_rx();
                    if(temp1=='q')break;
        }
        usart_tx(12);
        usart_tx_string("Project End...");
}

void usart_init()
{
        //Baudrate:9600
        UBRRH=(ubrr_value>>8);
        UBRRL=ubrr_value;
        //Enable Transmiter and Receiver
        UCSRB|=(1<<TXEN)|(1<<RXEN);
        //Character size:8bits
        UCSRC|=(1<<URSEL)|(1<<UCSZ1)|(1<<UCSZ0);
}

void usart_tx(chardata)
{
        while(!(UCSRA&(1<<UDRE)));
        UDR=data;
}
void usart_tx_string(char*string)
{
        while(*string)
        {
                usart_tx(*string++);
                _delay_ms(50);
        }
        usart_tx('\r');
}
unsigned char usart_rx()
{
        while(!(UCSRA&(1<<RXC)));
        return UDR;
}
void motor()
{
        PORTB=0x08;
        _delay_ms(100);
        PORTB=0x0C;
        _delay_ms(100);
        PORTB=0x04;
        _delay_ms(100);
        PORTB=0x06;
        _delay_ms(100);
        PORTB=0x02;
        _delay_ms(100);
        PORTB=0x03;
        _delay_ms(100);
        PORTB=0x01;
```

```
        _delay_ms(100);
        PORTB=0x09;
}
```

3.4 Burning the Code

The hex file is generated with the same name as program using an AT mega 32 microcontroller. This program is transferred to flash memory of microcontroller. A programmer can be used to burn the program. The program is burnt into microcontroller. The burner uses SPF port of microcontroller to load the program.

Steps:

1. Open Atmel studio. Copy the program source code along with its attachment libraries and headers. Save and build the main program source code in Atmel studio. To set board and frequency go to Project- Configuration options in Atmel studio. Select device as AT mega 32 and set frequency as 8MHz.
2. Connect USB programmer to AT mega 32 board. Open AVRdude software. In configuration settings, select AT mega 32 as microcontroller.
3. In memories section of AVRdude, browse for hex code generated for the main source code.
4. Also, check if the write button is ticked. If not, tick it. After uploading hex code. Press execute to burn the code into AT mega 32 board.
5. If everything is uploaded perfectly, then stepper motor rotes from 0˚ to 180˚ and vice versa.

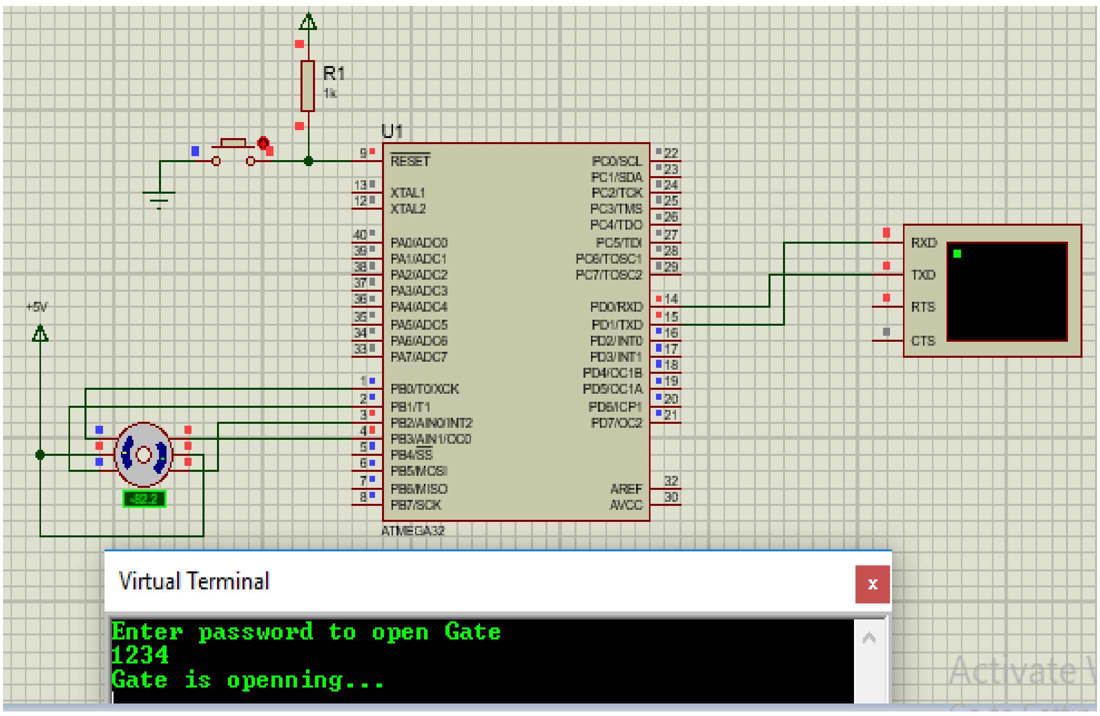

Figure 3.2 Open the Stepper Motor (Gate) Using the Password

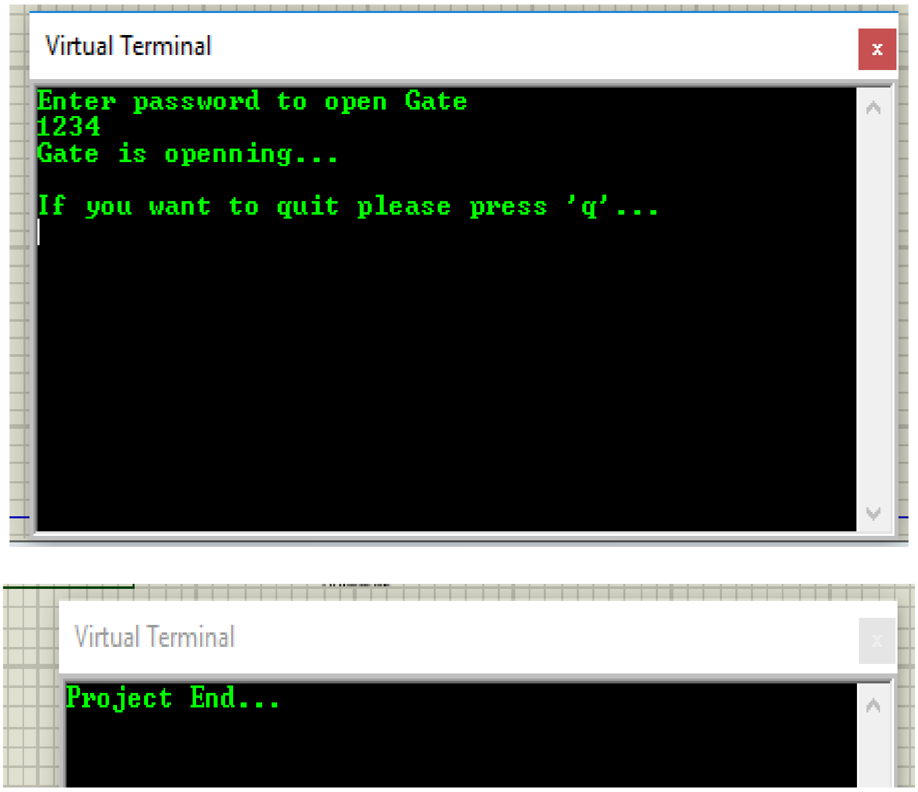

Figure 3.3 Quit the Project

4. Conclusion

Microcontroller based stepper motor control system is a simple and useful circuit which can be used to control the motor. In this paper, this system is only simulation. In general, the stepper motor can be rotated directly by connecting the motor with the power supply. By creating a C program where the stepper motor can be rotated by using the AT mega 32 microcontroller. The speed and the rotary directions can be controlled by editing the C program. These C programming stepper motors can be mainly applicable in the industries for the automation process. Generally stepper motors can be used for various applications. But by programming and rotating the stepper motor, it can be applicable for automation process in the industries.

Acknowledgement

The success and final outcome of this paper is required a lot of guidance and assistance from my teacher. I respect and thank Mr. G.R. Sinha, Professor, Myanmar Institute of Information Technology (MIIT), Mandalay (Adjunct Professor, IIIT Bangalore, India). I would like to special thanks to Mr. Kiran Kumar, Professor, CDAC (Central Development of Advance Computation), NOIDA, India.

References

[1] ATmega32 Datasheet, http://www.atmel.com

[2] https://en.wikipedia.org/wiki/Stepper_motor

[3] http://rickruling.blogspot.com/ interfacing-unipolar-stepper –motor.html.

[4] https://www.elprocus.com / stepper- motor - control- using - avr - microcontroller

[5] https://vshamu.wordpress.com/2011/03/23/stepper-motor-interfacing-with-microcontroller/

Authors

Kyi Kyi Khaing completed her Ph.D specialising in computer hardware technology from the University of Computer Studies, Yangon in 2007. She works in Faculty of Computer System and Technology (FCST) at Myanmar Institute of Information Technology (MIIT), Mandalay.