MICROCONTROLLER BASED SERVO MOTOR CONTROL SYSTEM

Phyu Phyu Shein[1], Tin Tin Nwet[2], Kyi Kyi Khaing[3]
[1]ITSM Dept. Computer University(Dawei), Myanmar
[2]FCST Dept. Computer University(Taungoo), Myanmar
[3]FCST Dept. MIIT(Mandalay),Myanmar

Abstract

The microcontroller based servo 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 servo 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 servo motor by using an AT mega 32 microcontroller. AT mega 32 has been programmed in order to rotate the servo motor for various applications. The motor drive which acts as an interface between the microcontroller and the servo motor. The microcontroller board is applying the external voltage of about 5V. From this it can be directly deliver the input to the servo motor drive where it provides sufficient current to the servo motor which tends to rotate the servo motor. By programming the servo drive, we can be able to control the speed of the servo motor, direction of the servo motor either in clockwise or anticlockwise direction. Motor drive protects the servo motor from damages so that the servo motor drive can be connected between the servo motor and AT mega 32 microcontroller.

Keywords -- *AT mega 32, Microcontroller, Servo Motor, Reset, Trigger*

I. 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 analogue 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 servo 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].

II. IMPLEMENTATION OF THE DESIGN

This design is based on both hardware and software. For the design to be implemented, we will be using an AT mega32 microcontroller, interface with some other hardware components. The functional block diagram of the design is shown in Figure 2.1.

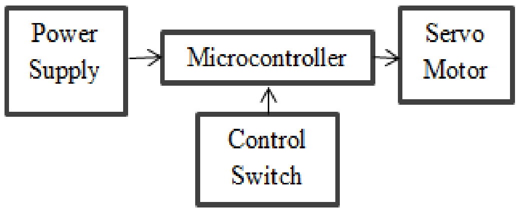

Fig. 2.1 Functional Block Diagram of the Design

Fig. 2.2 Servo motor

A servo system that has the highest available feedback resolution and minimal compliance and/or backlash, can obtain, 1-15:1 and higher inertia ratios for many applications. These deliver the best operational efficiency with little additional risk. Higher inertia ratios become increasingly dependent upon the application, not only relative to the desired machine specification, but also in relation to the mechanism's compliance and backlash. For Direct-Drive systems, the required stiffness often demands larger shafts and bearings, and even the compliance of the machine fixture/stand holding the motor comes into play. [2]

Coordination of Axes

Applications requiring coordination between axes can benefit from servo-controlled systems due to their tight synchronization and high BW capability which allows for fast correction against signal disturbances and/or command changes. Properly sized open-loop servo motor systems, will stay in sync without any confirmation feedback, but are limited to point-to-point moves with only the possibility of sequential or pseudo coordination between the commanded axes. [3]

Cabling & Motor-Drive Adjustments

One change that improves reliability and maintenance in servos has been the reduction in the number of wires necessary between the power and feedback devices.

Manufacturers have taken much of the guesswork out of servo tuning (motor-drive-mechanism compensation) of the closed-loop system and determining when a system needs maintenance. Automated or calculated tuning techniques and built-in diagnostic programs help simplify this requirement for the user. In addition, most servo drives can use traditional Step and Direction inputs that have been used to interface to steppers for many years. Servos utilizing this capability are in a position mode that eliminates the potential loss or addition of commanded motor steps.

This brings us to one of the most common issues with step motor systems when run on the edge of their capability, the loss and/or addition of motion steps relative to the number of commanded steps. The problem is most noticeable during acceleration and/or deceleration. The loss of steps typically results from too large inertia (affecting acceleration) or higher-than-desirable friction. In contrast, the addition of steps typically results from noise or to too large inertia (affecting deceleration). Due to the accumulation of the additional (+) and loss (-) motion steps, it can take hours before some manufacturing tolerance is exceeded.

Nonetheless, servo motors are still simpler having fewer wires to connect with minimal amounts of motor-drive adjustment to get a system up and running.

III. OPERATION OF THE SYSTEM

According to the circuit, the connections are done by connecting the servo motor with the microcontroller interfaced by the servo 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 servo motor drive. The servo motor drive which amplifies the signal and sends to the servo motor. The servo motor receives the proper current and voltage and therefore the servo motor can be rotated easily. By editing the program, the speed and direction of the servo motor can be controlled.

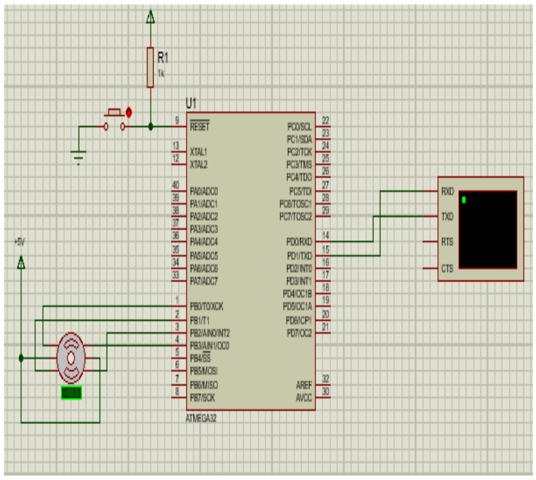

Fig. 2.3: Schematic of the Design

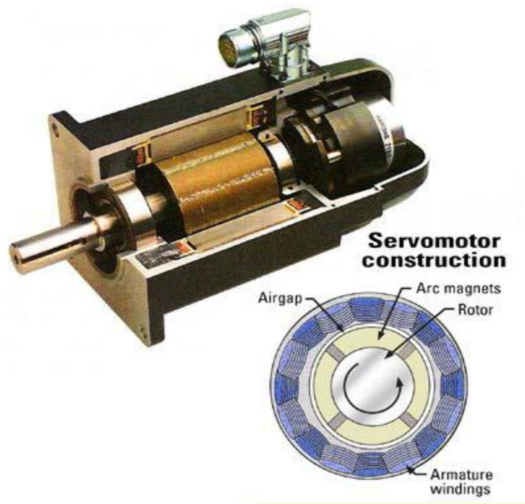

Fig 2.4 Servomotor construction

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 servo motor rotes from 0˚ to180˚ and vice versa.

IV. CONCLUSION

Microcontroller based servo 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 servo motor can be rotated directly by connecting the motor with the power supply. By creating a C program where the servo 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 servo motors can be mainly applicable in the industries for the automation process. Generally servo motors can be used for various applications. But by programming and rotating the servo 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 would like to special thanks to Mr. Kiran Kumar, Professor, CDAC (Central Development of Advance Computation), NOIDA, India.

REFERENCES

[1] ATmega32Datasheet,http://www.atmel.com
[2] https://en.wikipedia.org/wiki/Servo_motor
[3] Technology & Capability for the best performance of Servo and Stepper Motor