PC Based Speed Control of Stepper Motor Using Wired Communication

[1]Jeetender Singh Chauhan, [2]Atul Kumar Pandey, [3]Gyan Prabhakar
[1]Dept. of Instrumentation & Control Engineering, Graphic Era University, Dehradun, India
[2,3]Dept. of ECE, Saroj Institute of Technology & Management, Lucknow, India

Abstract
Stepper motors are used in many devices and appliances that are part of our everyday life. Sensing variables such as position, velocity or current for the purpose of control is a common problem in many industrial drive applications Sensing signals that truly represents system variables, such as absolute shaft position, may be a difficult either because of cost or physical limitations. In such cases we must estimate all or some of the missing variables from limited measurements that may be noisy. The objective of this project is to design and implement a Microcontroller circuit to control the stepper motor via keypad. This enables the Microcontroller circuit to control the speed and step angle of the stepper motor. When user input different commands through the keypad. This project gives exact concept of interfacing a high voltage electrical device or DC / AC motor to high sensitive personal computer system. The developed system we can develop the GUI to monitor and control the speed of stepper motor. The project can be divided into two element which hardware and software. The hardware of the proposed system and interfacing with computer using RS232 serial communication port. We are using the RS-232 as the communication medium between PC and controller.

Keywords
Stepper Motor, RS232, PC, MAX 232

I. Introduction

Stepper motor filled a unique niche in the motor control world. These motors are mainly used in measurement and control applications. Sample applications include ink jet printers, CNC machines and volumetric pumps. Several features common to all stepper motors make them ideally suitable for these types of applications. Stepper motors are brushless. The commutator and brushes of conventional motor are some of the most failure prone components and they create electrical arcs that are undesirable or dangerous in some environments. Stepper motor also will not turn at a speed regardless of a load as long as the load does not exceed the torque rating of the motor. Open loop position of stepper motors move in quantified increments of steps. Holding torque characteristic is able to hold the shaft stationary. Stepper Motors come in a variety of sizes, and strengths, from tiny floppy disk motors to huge machinery steppers. The Pulse-Width-Modulation (PWM) in microcontroller is used to control duty cycle of stepper motor drive.PWM is an entirely different approach to controlling the speed of a stepper motor. Power is supplied to the motor in square wave of constant voltage but varying pulse-width or duty cycle. Duty cycle refers to the percentage of one cycle during which duty cycle of a continuous train of pulses The term duty cycle describes the proportion of 'on' time to the regular interval or 'period' of time; a low duty cycle corresponds to low power, because the power is off for most of the time. Duty cycle is expressed in percent, 100% being fully on.

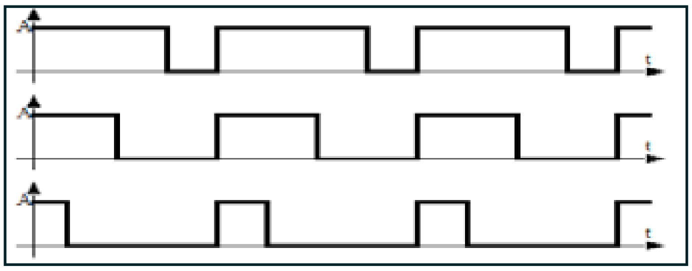

Fig. 1: Wave form at Different Duty Cycles

II. Hardware Development

Block diagram of the proposed system is shown below:

A. Power Supply Modules

This module is basically designed to achieved 5V, 500mA.This consists of a transformer which is used to step down the AC voltage, IN4007 diodes used to form a bridge rectifier to convert AC to DC, capacitor 1000uF which used as a filter circuit, 7805 regulator to obtain a 5V at the output of the regulator, 330 ohm resistance, LED as indicator.

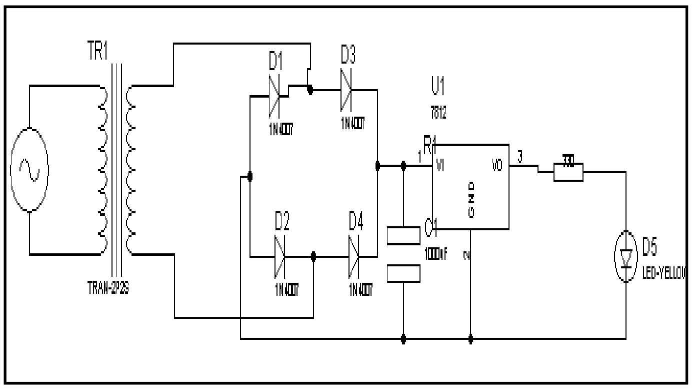

Fig. 2: Diagram of Supply Section

B. Microcontroller (ATmega16)

There is a whole wide range of microcontroller available in the market. But this particular project is developed using AVR series of microcontroller (ATMEGA16) because of its inbuilt ADC port and its variable frequency.ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves throughputs approaching 1 MIPS per MHz, allowing the system designed to optimize power consumption versus processing speed. Further it also minimizes the cost of this personal area network.

C. USB to Serial Cable

To interfaces the coordinator node with the other nodes.

D. MAX-232(level converter IC)

MAX232 is a dual driver/receiver IC that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply.Each receiver converts EIA-232 inputs to 5-V TTL/

CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept ±30-V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. This can be made to work with the help of a few capacitors attached to it.

E. LCD Display

The LCD(liquid crystal display) unit receives character codes (8 bits per character) from a microprocessor or microcomputer, latches the codes to its display data RAM (80-byte) DD RAM for storing 80 characters, transforms each character code into a 5'7 dot-matrix character pattern, and displays the characters on its LCD screen. We are 16*2 LCD's which have 16 columns and 2 rows with 16 hardware pins connected as pin 1,3and 16 are connected to ground, pin 2 and 15 are connected to +5v pin 3 ,4,5 are RS ,RW and enable respectively enable pin is always low . data pins of LCD are 11,12,13,14 which are used for 4 bit parallel communication.

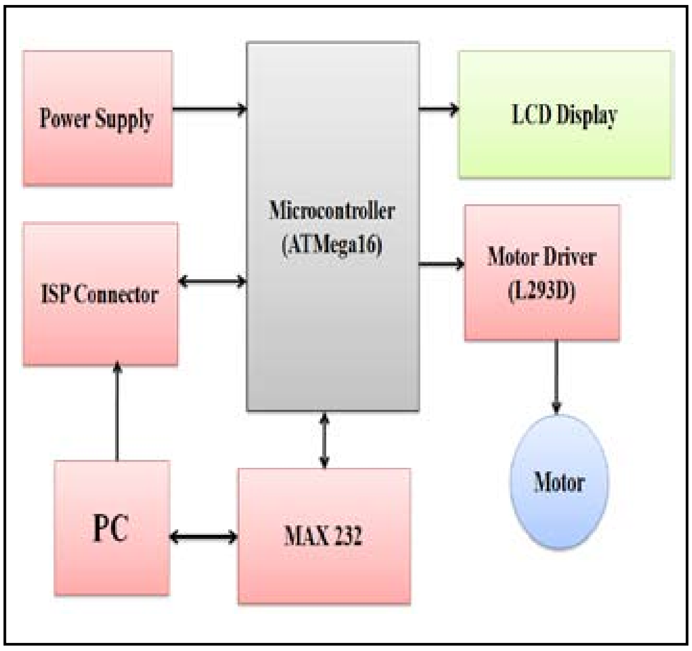

Fig. 2: Block Diagram of System

F. DB9

It is 9 pin male / female connector. In DB9 9 represent total number of pins and D represents the two parallel rows of pins that are in the shape of D alphabet.

G. L293D (Motor Driver IC)

This IC is high voltage high current four channel driver designed to accept DTL or TTL logic. This can provide 600mA output current capability per channel and providing 1.2 peak output current (non repetitive) per channel and also have internal over temperature protection. It consists of a Half H Bridge to provide high current in order to drive motors.

H. RS-232

An RS-232 port was once a standard feature of a personal computer for connections to modems, printers, mice, data storage and other peripheral devices.RS-232 as the communication medium between PC and controller.

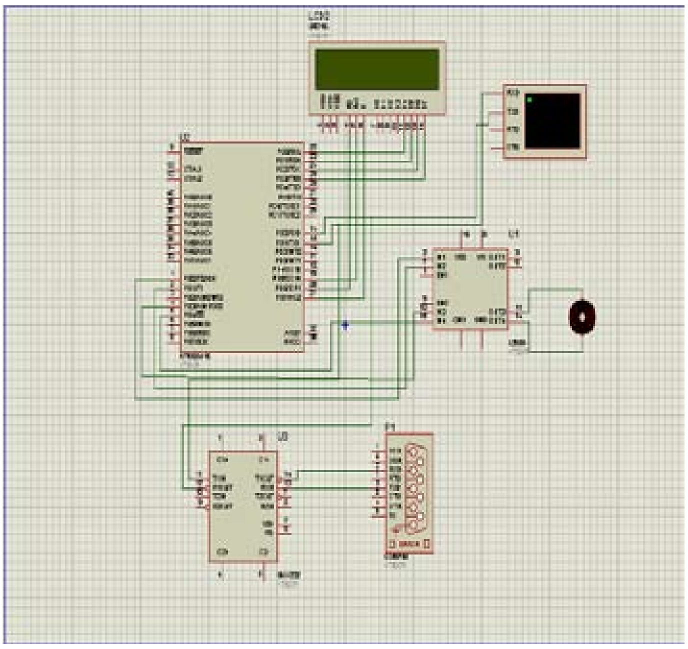

Fig. 3: Simulation Diagram of System

III. Software Development

Microcontroller, when it is used to operate as a wireless network involves following steps:

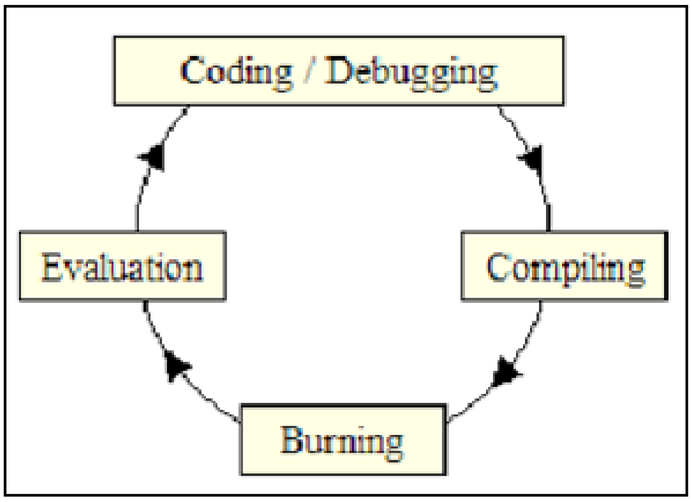

Fig. 4: Steps for Software Development

1. Coding / Debugging

Coding or debugging is one in a high-level language (such as c or java). Compiler for a high level language helps to reduce production time. To program the microcontrollers WinAVR [2] was used using C language.The source code has been commented to facilitate any occasional future improvement and maintenance.WinAVR is a suite of executable, open source software development tools for the Atmel AVR series of RISC microprocessors hosted on the Windows platform. It includes the GNU GCC compiler for C and C++. WinAVR contains all the tools for developing on the AVR. This includes AVR-gcc (compiler), AVR-gdb (debugger) etc.

2. Compiling

After compiling the program, it is converted to machine level language in the form of o's ans1's.This file is called as the Hex file and is saved with the extension (.Hex). The compiler also generates errors in the program which should be removed for proper execution of the program.

3. Burning

Burning the machine language (hex) file into the microcontroller's program memory is achieved with a dedicated programmer, which attaches to a PC's peripheral. PC's serial port has been used for the purpose. For this purpose Ponyprog programmer was used to burn the machine language file into the microcontroller's program memory. Pony prog is serial device programmer software with a user-friendly GUI framework for Windows95/98/ME/NT/2000/XP and Intel Linux. Its purpose is reading and writing every serial device. It supports I²C Bus, Micro wire, SPI EEPROM, and the Atmel AVR and Microchip PIC microcontroller.The microcontrollers were programmed in approximately two seconds with a high speed-programming mode. The program memory, which is of Flash type, has, just like the EEPROM, a limited lifespan. On AVR microcontroller family it may be reprogrammed up to a thousand times without any risk of data corruption Atmega16 Programmer (ISP) which is used to burn the program into AVR microcontrollers.

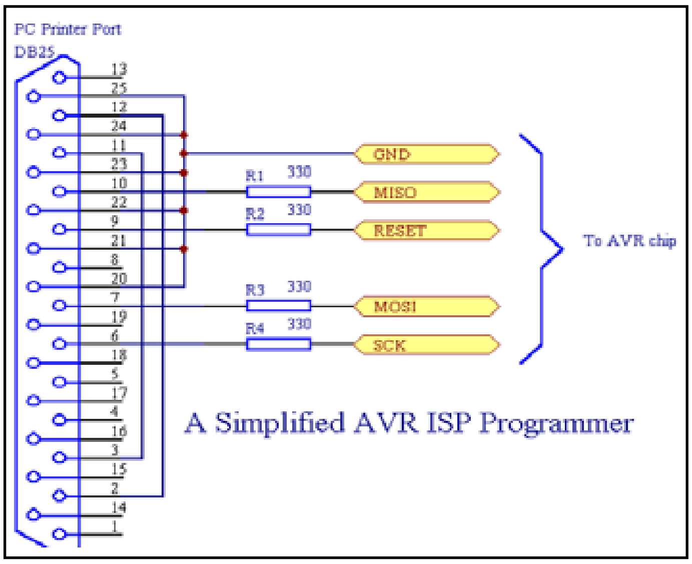

Fig. 5: Snapshot of Robokits AVR Programmer Window to Erase, Write and Verify .h File Generated by AVRstudio4 to Target

4. Evaluation

If the system performs as desired by the user and performs all the tasks efficiently and effectively the software development phase is over and the project is ready to be installed in any of the industrial sites as a personal area network. If not, the entire process is repeated again to rectify the errors. One of the difficulties of programming microcontrollers is the limited amount of resources the programmer has to deal with. In PCs resources such as RAM and processing speed are basically limitless when compared to microcontrollers. In contrast to a PC, the code on microcontrollers should be as low on resources as possible, but being cost effective and power efficient makes it a better option.

In the programming of the proposed system is used the following .c and .h file

(i). lcd.c

This c file contains the code for control of functionality of the attached LCD module. The code controls the initialization of the LCD, data writing on the LCD, and also the movement, characteristics and location of the cursor. It offers the facility to write data on the LCD character-by-character or string-wise.

The command set used in the software is based on the command set used in the LCD based on Hitachi HD44780 ICs. This file contain INitlcd( ), remove ( ), display ( ) and displayint ( ).

(i) to initialize the LCD:

```
Void INitlcd( )
{
//This function Initializes the lcd module
  must be called before calling lcd related functions

          Arguments:
          style = LS_BLINK,LS_ULINE(can be "OR"ed for
combination)

          LS_BLINK :The cursor is blinking type

          LS_ULINE :Cursor is "underline" type else
"block" type
}
```

(ii) to display strings to LCD :

```
Void display( const char  *data)
{
//This function Writes a given string to lcd at the current cursor
location.
Arguments:
          msg: a null terminated string to print
```

(ii). lcd.h

This header file contains all the constant variable values and names of the subroutines used by various files used in the software. It clearly indicates which variable can be used as a global variable and which of the subroutines can be used across the software files.

(iii). Usart_lib.c

This file contains the code for controlling the USART of ATMEGA'S. This is contain three major functions USARTInit ( ), USARTReadChar ( ) and USARTWriteChar ( ).

Initialization of USART:

This function will initialize the USART.

```
void USARTInit(uint16_t ubrr_value)
{
UBRR= ubrr_value;   //Set Baud rate
UCSRC=(1<<URSEL)|(3<<UCSZ0);// Set Frame Format
UCSRB=(1<<RXEN)|(1<<TXEN);// //Enable The receiver and
transmitter
}
```

Reading From The USART :

This function will read data from the USART.

```
char USARTReadChar()
{
    while(!(UCSRA & (1<<RXC))) //Wait until a data is
available
{
    //Do nothing
}
   return UDR;   //Now USART has got data from host and is
available is buffer
}
Writing to USART:
void USARTWriteChar(char data)
{
while(!(UCSRA & (1<<UDRE)))   //Wait until  the transmitter
is ready
  {
     //Do nothing
   }
UDR=data;   //Now write the data to USART buffer
}
```

(4) Adc.c- This file contains the code for controlling the ADC of ATMEGA'S. This is contain two major functions initializeADC( ), int ReadADC(uint8_t ch). This helps us to read various sensors .

(i)-Initialization of ADC:

```
Initialize ADC()
  {
    ADMUX=(1<<REFS0);// For Aref=AVcc;
    ADCSRA=(1<<ADEN)|(7<<ADPS0);
   }
```

(ii) Read data from ADC:

```
Int ReadADC(uint8_t ch)
   {
//Select ADC Channel ch must be 0-7

//Start Single conversion

 //Wait for conversion to complete

//Clear ADIF by writing one to it

return(ADC);
```

(v). Functions used in Program

The code which is used to program the controller include some functions as :

(i) to provide delay in the program

```
     Void delay (unsigned char value)
      {
       For(unsigned int i=0;i<value;i++)
            {
            _delay_ms(1);
```

(ii) controlling of the motor

```
Void motor(char data)
{
Switch(data)
{
Case 'a': motor at 100%;
               Break;

Case 'b': motor at 75%;
               Break;

Case 'c': motor at 50%;
               Break;

Case 'd': motor at 25 %;
               Break;

Default: motor at 0%;
               Break;
 }}
```

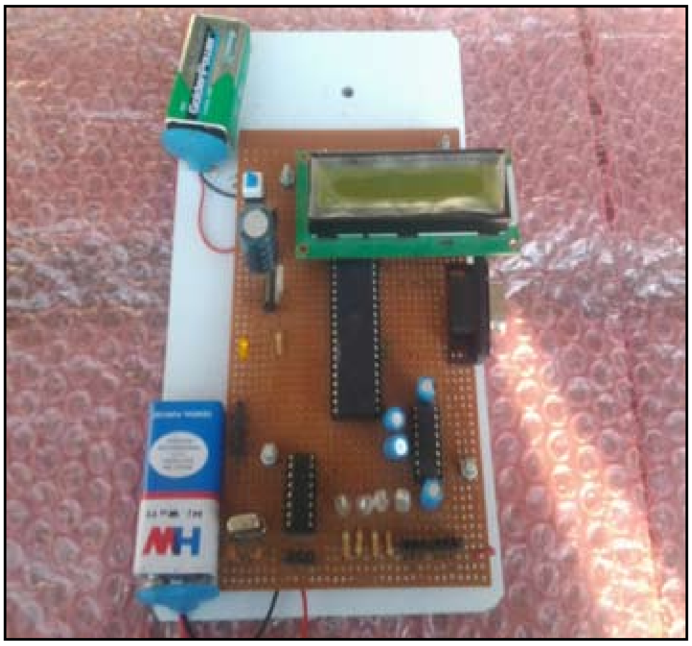

Fig. 7: View of Hardware Module

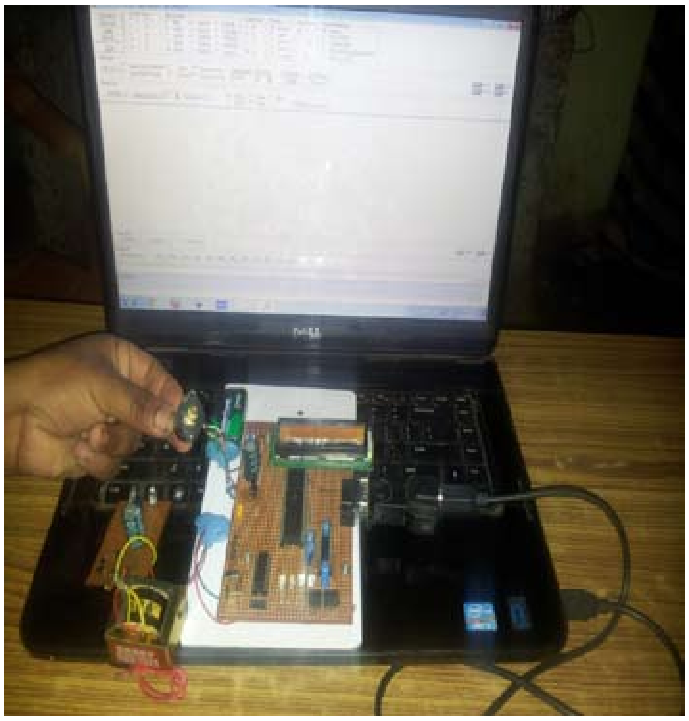

Fig. 8: View of Complete System

III. Conclusion

Stepper motor is widely used in industrial applications, office equipments, medical instruments and automobiles etc. Due to the high simplicity, low cost, high dependability and most importantly it can be controlled in an open loop system the designing of a sustainable system to control the speed of a Stepper Motor through

PC was successfully implemented in this paper. The paper provides a platform for further advancement in the field of industrial use of stepper motors.

References

[1] A. R. Alae, M. M. Negm, M. Kassas,"A PLC Based Power Factor Controller for a 3-Phase Induction Motor", IEEE Transactions on Energy Conversion, USA, 1065-1071, 2000. M. G. Ioannidis, Design and Implementation of PLC-Based Monitoring Control System for Induction Motor, IEEE Transactions on Energy Conversion, 19, No. 3, USA, 2004.

[2] M. G. Ioannidis,"Design and Implementation of PLC-Based Monitoring Control System for Induction Motor", IEEE Transactions on Energy Conversion, 19, No. 3, USA, 2004

[3] S. M. Bashi, I. Aris, S.H. Hamad,"Development of Single Phase Induction Motor Adjustable Speed Control Using M68HC11E-9 Microcontroller", Journal of Applied Sciences 5 (2), pp. 249-252.

[4] Kumara MKSC, Dayananda PRD, Gunatillaka MDPR, Jayawickrama SS,"PC based speed controlling of a dc motor", A fmal year report University of Moratuwa Illiniaus USA, 2001102.

[5] J. Chiasson,"Nonlinear Differential-Geometric Techniques for Control of a Series DC Motor", IEEE Transactionson Control Systems Technology, Vol. 2, pp. 35-42, 1994.

[6] Yodyium Tipsuwan, Mo-Yuen Chow,"FuzzyLogic microcontroller implementation for DC motor speed control", IEEE Transactions on Power Electronics, Vol. 11, No. 3, June 1999, pp. 1271-1276.

Jeetender Singh Chauhan received his B.Tech degree in Electronic & Communication from Sagar Institute of Technology and Management Barabanki U.P., India and pursuing M.Tech. in Instrumentation and Control Engineering from Graphic Era University Dehradun, Uttarakhand, India.

Atul kumar Pandey received his B.Tech degree in Electronic & Communication from Sagar Institute of Technology and Management Barabanki U.P., India and pursuing M.Tech. in Instrumentation and Control Engineering from Graphic Era University Dehradun, Uttarakhand, India. He is currently attached with Saroj Institute of Technology & Management Lucknow, India in the department of Electronics & Communication.

Gyan Prabhakar received his M.Tech (Micro-electronics/VLSI Design) from Punjab University Chandigarh (five Star Central University). I did my post graduation (M.Sc in Electronics) from C.S.J.M. University Kanpur. I have 2.7 years Industry experience & 2.5 year teaching experience. He is currently attached with Saroj Institute of Technology & Management Lucknow, Head of Electronics & Communication department.