ASYNCHRONOUS DATA TRANSMISSION FOR MOTOR CONTROL VIA THE INTERNET

Chandra R. Sekhar, Omer Farook, Jai P. Agrawal, Essaid Bouktache, J. Spader, T. Webb
Purdue University Calumet, Hammond, IN 46323.

ABSTRACT

This paper describes a senior design project of a real-time asynchronous data transfer utilizing a client -server architecture and the Ethernet LAN topology. This project was executed using two independent software programs and a stand-alone server. These two programs function together to control a stepper motor via Internet. LabVIEW software is used to monitor, control, and store the specific control parameters such as direction, motor speed, and other measurable attributes of the motor. The digital data from the LabVIEW was then transmitted from the computer, over the Internet through the use of a DataSocket server. The remote host computer used a Visual Basic Program to process, monitor, and control the motor functions and serve as the remote front end. The remote user can view direction, motor speed etc., and can manipulate those variables to change the motor operation. The control signals were transmitted back to the originating computer, by the use of DataSocket API's. In turn, LabVIEW in the interfacing computer will carryout the specified changes made from the remote front end control panel.

I. INTRODUCTION

The project's objective was to develop an asynchronous data transmission for controlling a motor via the Internet. This method is equally applicable for controlling a motor driven instrument or a process from a remote location, monitoring and control of multiple motors/processes from a centralized remote location through the Internet or local area networks. Another application is distance learning classes with virtual laboratory in the design, testing and trouble-shooting in many electrical areas.

This project incorporates (a) A stepper motor, (b) LabVIEW[3], a software residing in a computer at the primary site for data acquisition and control, (c) A server to store data and the controlling software (d) A remote site computer with programs written in visual basic to display a control panel where the transmitted parameters could be adjusted. Refer Fig. 1.

II. SYSTEM DESCRIPTION

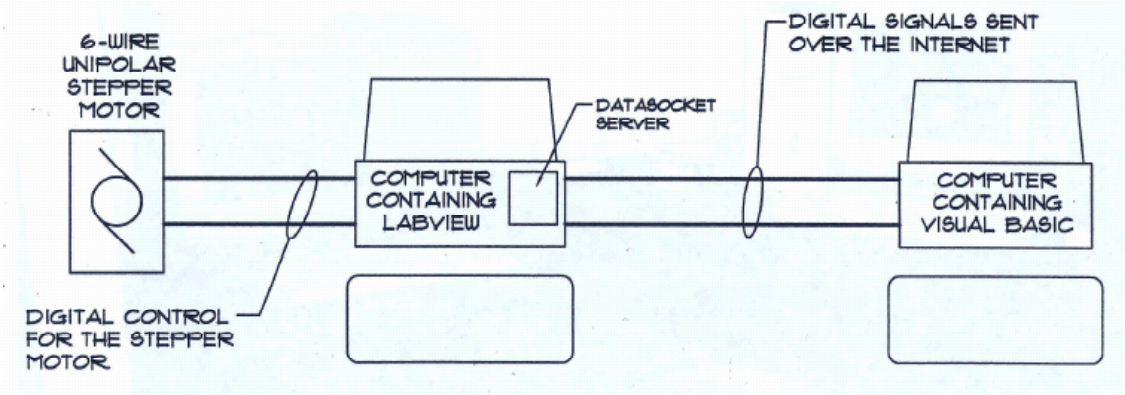


Fig. 1 System diagram

2.1 STEPPER MOTOR INTERFACE

We selected a 6-wire uni-polar stepper motor, sth-39d103-01: 12 V/ph, 16 A/ph and 1.8 deg/step. The six wires of the stepper motor were directly connected to the 5804 BiCMOS II Unipolar Stepper-Motor Translator/Driver [5], which supplies 1.25A and 35V per phase. This driver will allow the user to utilize the sequencing logic. The uni-polar stepper motor translator/driver was connected to DAQ, the data acquisition card in LabVIEW[3]. The Driver is essential for the operation of the motor since the output current from the DAQ card is insufficient to run the motor.

The current was stepped up to 1 amp in order to properly control the motor. The DAQ card was connected to the input/output slots of the computer. The driver was connected to the DAQ card through seven wires. Five wires are used to control the direction, enable output, step input, half-step and full step. The sixth and seventh wires were connected to +5 V and ground, refer to Fig. 2. The five connections to the DAQ card were programmed in LabVIEW[3] explained in the following section 2.2. LabVIEW's configuration menu allows the user to designate port A (0-4) for the binary control of the motor.

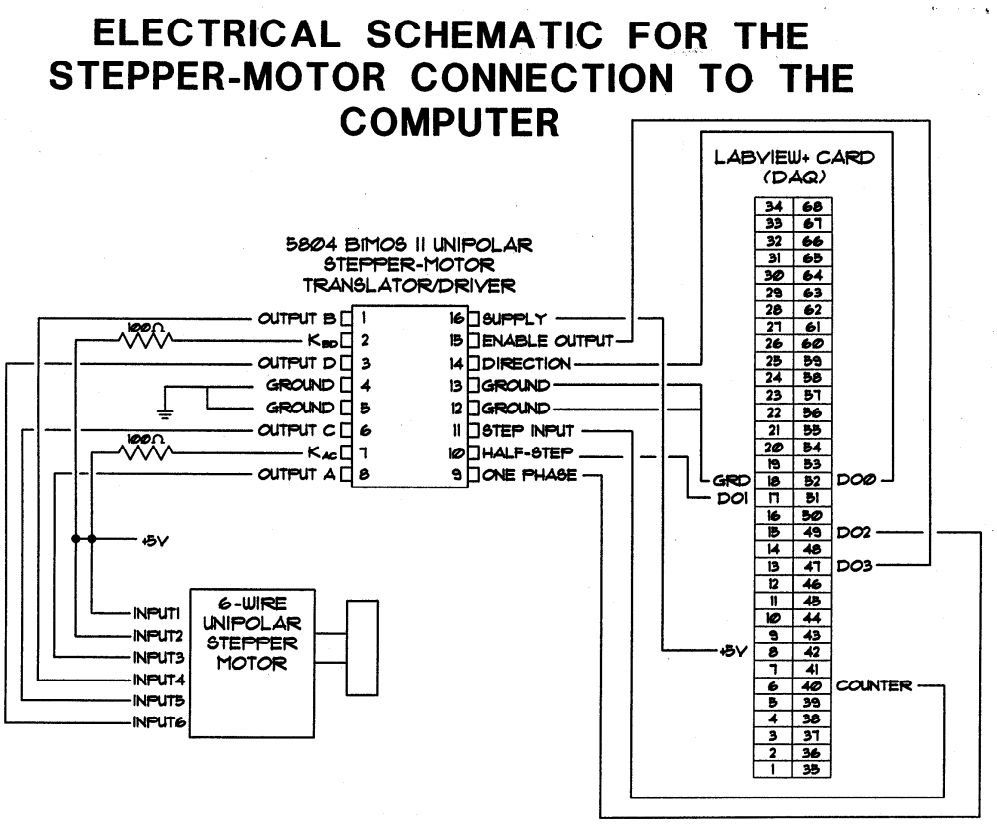
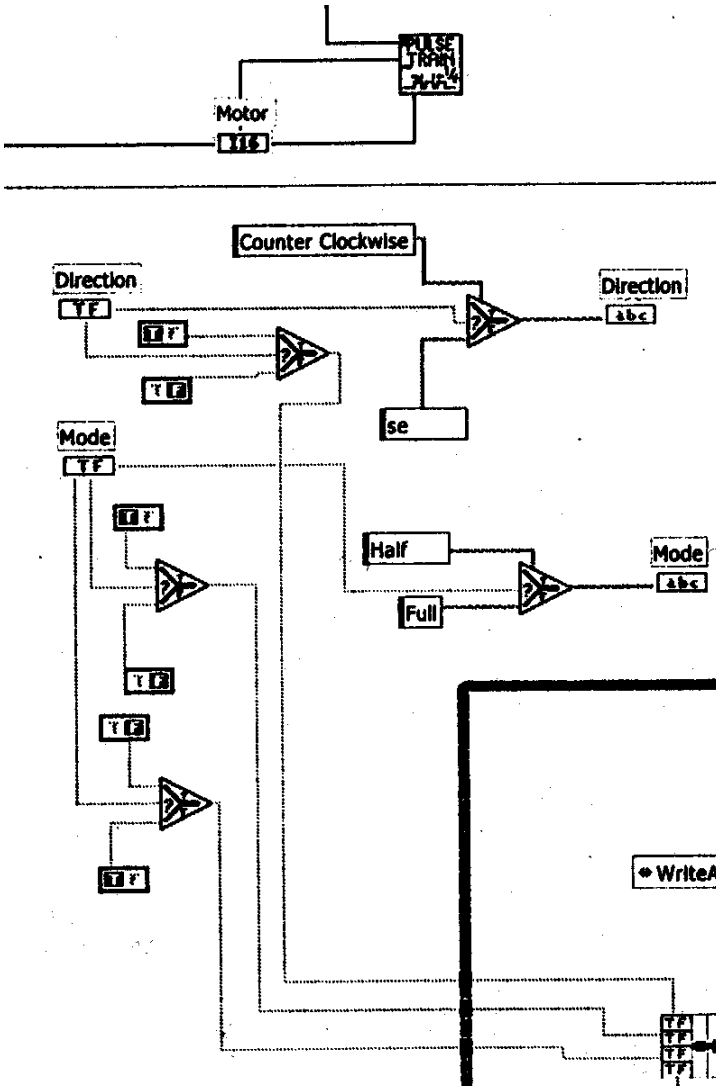


Fig. 2 Motor Control Schematic

2.2 LABVIEW PROGRAMMING

LabVIEW is an user friendly graphical program that allows the user to acquire Data from a source, use the built in programs to analyze and control the data. The input variables can be sent over the Internet for real time manipulation by a remote user. LabVIEW is a program much like C++ builder where there is a user interface which is designed through the use of visual development tools. LabVIEW differs from C++ in that the tools are graphically programmed, as opposed to being text based programming. The graphical program can then be modified and manipulated to perform various functions.

Our LabVIEW program controls the start/stop function, direction, full step input, half step input and motor speed. The start/stop function is controlled by a *while* loop that runs until stop is pressed on the front panel. The front panel will display motor status, which will state if the motor is running or idle. The program is then paused until it starts again. The mode and direction use a Boolean switch, which will feed a true or false value to the selectors depending on the operation chosen. The clock-wise direction is denoted by "1" and the counter clock-wise by "0". The full-wave mode is denoted by "1" and the half-wave mode by "0". These were controlled using two separate true and false symbols. This was done so each true-false symbol could be sent directly to the digital output port. The final output sent to the digital port is the clock signal.



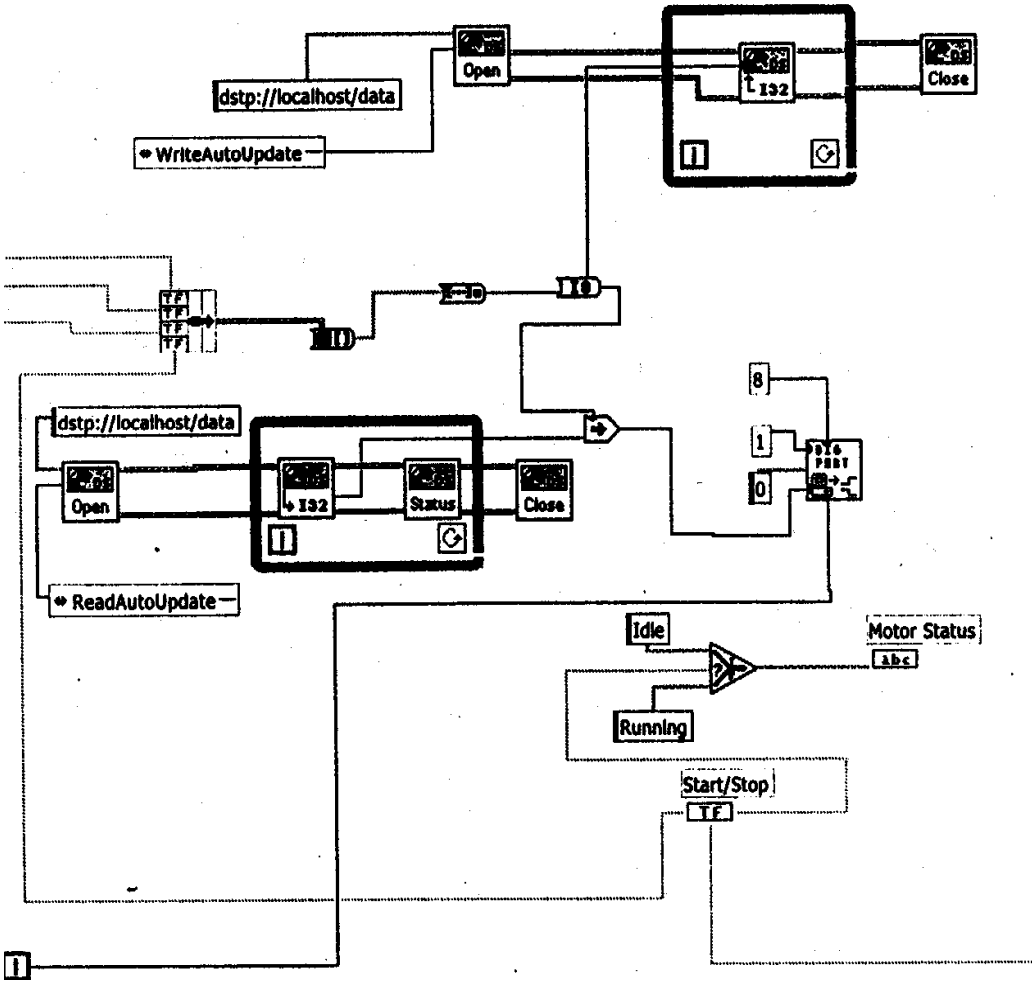


Fig. 3 LabView program

A pulse train was used in order to continuously pulse the step input. The frequency is altered on the front panel and is then inputted into the pulse train function in order to control the motor speed. This pulse train function can be controlled.

The final step to the program was to access the Internet. First the open DataSocket symbol was put in place and the proper URL's and read-write functions were provided as input. This function will access the DataSocket server in order to read or write from it. Then the DataSocket writer was put in to place and connected to the variable that was being sent over the Internet. The variable will initially be sent to the DataSocket server then in turn the DataSocket server will access the Internet and make the variables remotely accessible. Next the DataSocket close symbol was put into place and attached to the DataSocket writer. This function is essential, because it closes the DataSocket
connection after the program has been stopped. Refer to the specifications section for LabVIEW graphical program[6], see Fig. 3.

III. DATA TRANSMISSION

The attributes of the stepper motor are controlled using LabVIEW through the interface I/O card. In order to read and manipulate the data over the Internet, software programs in the client and server computers need to interact with each other, to provide remote site control. A DataSocket Server is created in LabVIEW to store real-time files for controlling the stepper motor. These LabVIEW files can be accessed after log on, and from then on can be manipulated from the DataSocket server, by a DataSocket API. A DataSocket is a single, unified, end user API (Application Programming Interface) based on the URLs for connecting to measurement and automation data located anywhere, be it on a local computer or anywhere on the Internet [6]. It is a protocol-independent, language-independent, and OS-independent API designed to simple binary publishing. Essentially,
DataSocket is a technology that allows you to send and receive data over the network from a variety of software platforms.

A DataSocket server is a standalone application that runs on a computer and will handle client connections. The client connections may write data to the server (known as DataSocket publishers) or read data (known as DataSocket subscribers). The DataSocket server automatically handles the underling network connections and data packet transmission, making it transparent to the clients. The DataSocket clients are implemented in Visual Basic [1]. This DataSocket API is used to talk to the server. The DataSocket package contains definitions of interfaces, classes and exceptions. We have used the ActiveX control in Visual Basic to control the DataSocket. It is used to access the
DataSocket Server and granted the rights to publish to and subscribe to the DataSocket Server, in effect either reading the motor's attributes half-step or full step, speed, and state (on/oft) or changing them. The value of a DataSocket determines the following actions of the DataSocket Server: Reserved, Read, *ReadAutoUpdate*, Write, and *WriteAutoUpdate*. These functions basically perform the same operation as their names specify .The programmer can set the DataSocket value to any of these values listed and subsequently determine what the remote user is capable of doing to the LabVIEW program files at the DataSocket Server.

IV. THE REMOTE USER PROGRAM

Visual Basic was opted as the remote user program. The ActiveX controls permit easy interaction between the LabVIEW and Visual Basic. The front panel, refer Fig. 4, created in Visual Basic resembles the panel created in LabVIEW. In the program code, different components were dropped down onto the ActiveX control form in Visual Basic. These components were assorted: DataSockets, buttons, text boxes, and labels. Each of the components' code was changed to perform the desired functions. The DataSocket Server component on the Visual Basic side was designed to connect to, read from and write to the DataSocket Server. The DataSocket Server was setup for readautoupdate
and writeautoupdate. These values allow the server to show or change different motor attributes on the LabVIEW front panel for controlling the stepper motor.

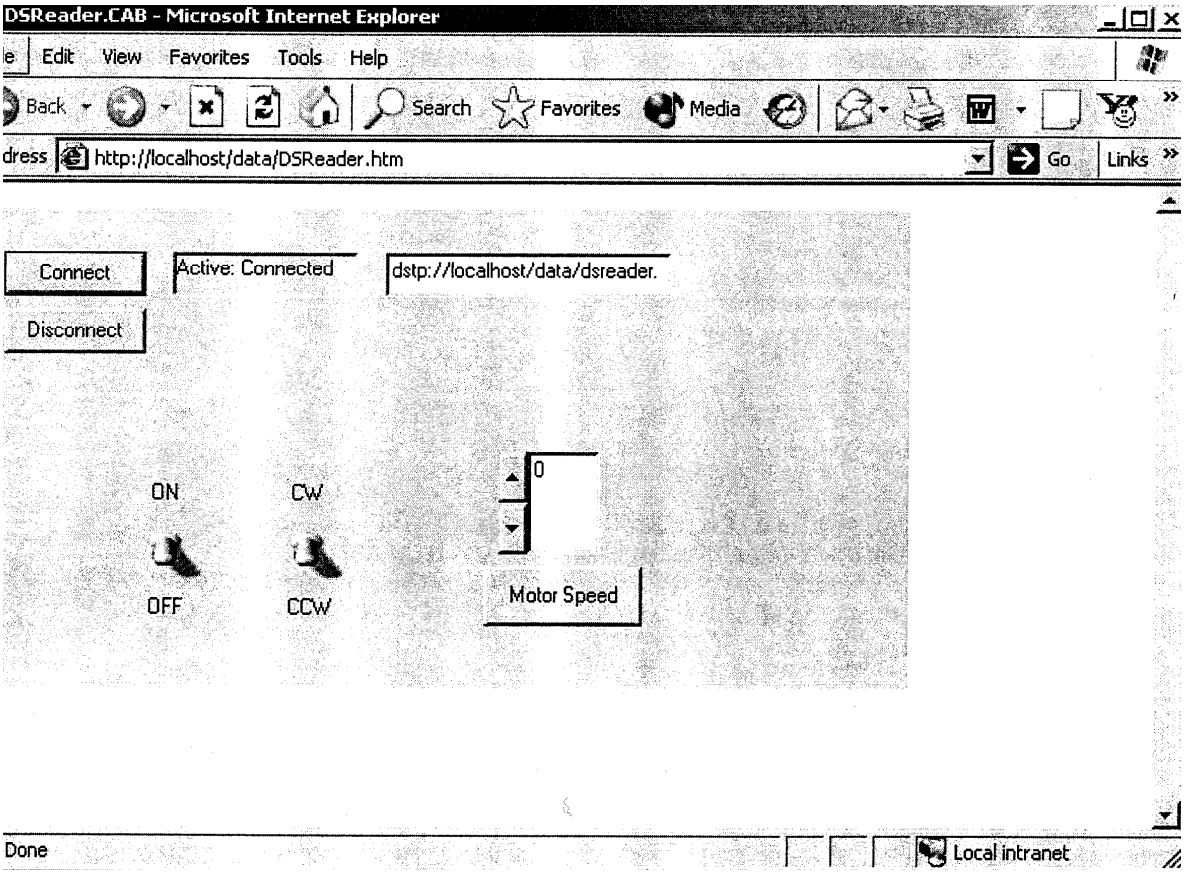


Fig. 4 Front Panel on the Remote terminal

VISUAL BASIC[1] CODE DESCRIPTION

Visual Basic is a graphical programming language. An ActiveX Control Form was created. Two DataSocket server components 1 and 2 were placed on the ActiveX Control Form. DataSocket Server 1 component is designated to show the status of the DataSocket server running on the local host. The DataSocket Server 2 component is used to control the LabVIEW front panel. Their designations can be viewed in the program code. The DataSocket Server 1 component is set as ***readautoupdate*** while the DataSocket Server 2 component is set as ***writeautoupdate***. The label 1 is placed on the ActiveX Control Form. This field remains blank until the application is connected to the DataSocket Server. This label tells the user the status of the DataSocket server. The TextArea 1 is placed on the ActiveX Control Form. This field displays the location of the DataSocket server files. The On/Off attribute code is located in the CWButton1 area of the code and the CW /CCW attribute is located in the CWButton2 area of the code. The CWButton1 and 2 data type is Boolean.

In the code, the CWButtons1 and 2 (On/Off and CW /CCW) were set to equal an async. Type constant. Once the value is changed on the VB control panel viewed over the Internet (automatically checked when updated set by ***writeautoupdate***) the program will then

call the DataSocket server 2 component and update it with the new value. This value is then sent over the Internet to the DataSocket server. After receiving the value, it is integrated into the LabVIEW program by use of the DataSocket server components in the LabVIEW program. The received value is used in the LabVIEW program. This value alters the front panel in the LabVIEW as well as the attribute of the running motor.

```
Private Sub Conamandl Click(}

CTjDataSocketl.ConnectTo Text1.Text, cwdsReadAutoUpdate

End Sub
Private Sub Conmand2_C Click(


CWDataSocketl.Disconnect
End Sub

Private Sub Conamand3_Click ( )
CWDataSocket2.ConnectTo NumEditl, cwdsWriteAutoUpdate
End Sub


Private Sub CWButtoni ValueChanged(ByVal Value As Boolean)
CWButtonl = AsyncTypeConstants
CWDataSocket2.ConnectTo ActiveControl, cwdslriteAutoUpdate


End Sub

Private Sub CWButton2 ValueChanged(ByVal Value As Boolean)
 ~
 •
CWButton2 = AsyncTypeConstants
CWDataSocket2.ConnectTo ActiveControl, cwdsJriteAutoUpdate
End Sub
```

```
Private Sub CWDataSocketl OnStatusUpdated(ByVal Status As Long, ByVal Err
StatusLbl.Caption = Message
End Sub


Private Sub CWDataSocket2_OnDataUpdated(ByVal Data As CWDSLib.CWData)
CWNumEditl.Value = Value
CWButtonl.DataBindings = AsyncTypeConstants
CWButton2.DataBindings = AsyncTypeConstants
End Sub


Private Sub CWDataSocket2 OnStatusUpdated(ByVal Status As Long, ByVal Err
StatusLb2.Caption = Message
End Sub


Private Sub CWNumEditl_ValueChanged(Value As Variant, PreviousValue As Vs
CTjJNumEditl = Value
CWDataSocket2.ConnectTo CWNuiEditi, cwdsWriteAutoUpdate


End Sub
```

```
Private Sub stal Click()

End Sub
```

III. CONCLUSION

Asynchronous data transmission for motor control via the Internet can be a very powerful tool. Our project enables the user to control a motor or a machine from a remote location. Some examples of these applications are: Monitoring tools (such as cameras, medical equipment, sensors), which can be supervised and controlled from a remote location, controlling machinery from long distances, and distance learning classes.

Using the technology available today, two software applications can negotiate over long distances, to produce a desired end result. We have used Visual Basic and LabVIEW. Both are robust, object-oriented, and multi-threaded programs. Visual Basic permits multiple concurrent threads of execution running simultaneously. Multithreading enables the spinning off background tasks, management of the simultaneous streams of input and the management of a user interface. Both LabVIEW and Visual Basic have been interfaced with Internet.

In this project we have used LabVIEW to control a stepper motor at the server site. The LabVIEW contains extensive network support features. The LabVIEW program files are stored in a DataSocket server, accessed and manipulated by a remote user who has been granted publishing rights. We have used ActiveX controls in the Visual Basic program, residing in the remote client computer to interface with the LabVIEW program in the server site.

REFERENCES

[1]. Visual Basic 6 Unleashed Professional Reference Edition, Thayer Rob, Indianapolis: Sams Publishing, 1999.

[2]. Applied Data Communications, Goldman, James E., and Phillip T. Rawles, John Willy & Sons, 2001.

[3]. LabVIEW Data Aquisition Basics Manual, Travis, Jeffrey, National Instruments, 1996.

[4]. Northwestern. Mechatronics. 1 Dec. 2001 <http:1 Imechatronics.mech.north western. Edu/mechatronicsl designr stepper_intro

[5]. Allegro Microsystems, Inc. Data Sheets. 1 Dec. 2001

[6]. National Instruments. LabVIEW. 1 Dec. 2001 http://www.ni.com

BIOGRAPHY

CHANDRA R. SEKHAR is a member of the faculty of the Electrical Engineering Technology at Purdue University Calumet. Professor Sekhar earned a Bachelor.s Degree in Chemistry from the University of Madras (India), a Diploma in Instrumentation from Madras Institute of Technology and Master.s Degree in Electrical Engineering from University of Pennsylvania. Professor Sekhar.s primary teaching and research focus is in the areas of Biomedical and Process Control Instrumentation and Clinical Engineering.

OMER FAROOK is a member of the faculty of the Electrical Engineering Technology Department at Purdue University Calumet. Professor Farook received the Diploma of Licentiate in Mechanical Engineering and BSME in 1970and 1972 respectively. He further received BSEE and MSEE in 1978 and 1983 respectively from Illinois Institute of Technology. Professor Farook.s current interests are in the areas of Embedded System Design, Hardware Software Interfacing, Digital Communication, Networking, C++ and Java Languages.

JAI AGRAWAL is an Associate Professor with joint assignment in Electrical Engineering Technology and Electrical & Computer Engineering. He received his PH.D. in Electrical Engineering from University of Illinois, Chicago, in 1991, M.S. and B.S. also in Electrical Engineering from I.I.T. Kanpur, India in 1970 and 1968 respectively. Professor Agrawal has worked recently for two years in optical networking industry in the Silicon Valley in California. Professor Agrawal is the Founder Advisor to Agni Networks Inc., San Jose, California. His expertise includes optical networking at Physical and Data link layers, optical and WDM interface, SONET and Gigabit Ethernet and analog electronic systems. He is the author of a Textbook in Power Electronics, published by Prentice-Hall. His professional career is equally divided in academia and industry. He has authored several research papers in IEEE journals and conferences.

ESSAID BOUKTACHE is a member of the faculty of the Electrical Engineering Technology Department at Purdue University Calumet. Dr. Bouktache received his MS and Ph. D in Electrical Engineering from the Ohio State University in 1980 and 1985, respectively. His research and teaching interests include Digital Signal Processing, Computer Networks, and Digital Communications. Professor Bouktache has been with Purdue since 1992 and is a member of IEEE and ASEE. He has several publications to his credit.

J. SPADER received her B.S. in Electrical Engineering Technology from Purdue University Calumet in 2003. She is currently working as an independent consultant. Her current interests reside in software design using C++ and in embedded systems.

T. WEBB received his B.S. in Electrical Engineering Technology from Purdue University Calumet in 2003. He currently is working software/hardware engineer. His current interests reside in control systems, programming in C++ and Assembly.