

“Real Time Automated Control using PLC-VB Communication”

Prof.Mrs.S.S.Bidwai¹, Prof.Mr.V.B. Kumbhar², Prof.Mr. A.R.Nichal³

¹ADCET Ashta Electronics and Telecommunication Dept, Ashta, Dist-Sangli, India.

^{2,3}AITRC Electronics and Telecommunication Dept, Vita, Dist-Sangli, India.

Abstract

This paper is to enable the Delta PLC (Programmable Logic Control) DVP14SS to communicate with the Visual Basic 6.0. The communication between DVP14SS and Visual Basic 6.0 is via Modbus Serial Protocol. Computers are used as a link between humans and PLC systems as they have more graphics and visual capabilities. These are nothing but SCADA systems widely used for determining plant setups and displaying plant status on high quality screens. They also record/log the system data for long period. The SCADA software's are the software packages needs to be purchased from vendors and the cost depends on tag count. Visual Basic 6.0 platform can be used develop the SCADA application effectively. Using VB 6.0 we integrate software and hardware across spectrum of vendors easily. Here we show simple approach to communicate Delta PLC with visual Basic using MSComm control in visual basic. By means of Visual Basic cost effective solution is possible as Visual Basic we do not need to purchase licenses and is cheaper than SCADA packages. It also has the advantages like flexibility.

Keywords—*MSComm control and PLC, plc, communication between PLC and VB 6.0, DVP14SS.*

I. INTRODUCTION

Communication has become a major part of any process control automation system. Today PLC communication is as much for data acquisition as plant control. The objective of communication is to collect or transfer large / small amount of data. Over the course of time certain interfacing standards have been generated by industry in order to make communication between systems from two different manufacturers more simple. These standards typically defined the communications medium, transmission voltages, speed of communication, (baud rate). The first such real standard was RS232. This was written by the Electronic Industries Association (EIA). The main advantage of this being cost as RS232 interfaces are very simple and thus cheap, plus of course the fact that most PCs, which are used more and more in automation systems today, have at least one RS232 (serial) interface as a standard.

Programmable Logic Controllers (PLC's) are solid state devices using integrated circuits to

control process or machines. They can store instructions like sequencing counting, timing, arithmetic, data manipulation and communication [2]. A PLC is an example of a hard real time system since output results must be produced in response to input conditions within a bounded time, otherwise unintended operation will result. PLC reads the status of the external input devices, e.g. keypad, sensor, switch and pulses, and execute by the microprocessor logic, sequential, timing, counting and arithmetic operations according the status of the input signals as well as the pre-written program stored in the PLC [5]. The generated output signals are sent to output devices as the switch of a relay, electromagnetic valve, motor drive, control of a machine or operation of a procedure for the purpose of machine automation or processing procedure.

Visual Basic (VB) is developed by Microsoft Corporation and is most widely used development environment in the world. Visual Basic is tool provides platform to develop new optimum Windows/SCADA applications. Visual Basic can communicate with PLC with various ways. Visual Basic does not have code for communication protocol inbuilt. User need to write code for communication protocol or need to purchase ActiveX controls which are used to develop applications. ActiveX control with Libraries can be purchased to develop application. Here we are using MSComm ActiveX control freely available in VB to establish communication with delta PLC.

II.Method of Interfacing

Delta DVP14SS comes with serial port and the communication protocol is serial Modbus protocol which is a open protocol. Visual Basic acts as a Modbus Master device which issues request or commands (Query) on serial bus which is received by slave. PLC acts as a Modbus slave device and responds to the Master by sending message called response. Modbus Query and Response contains fields like device address, function code, data, and error check [1]. Figure 1 shows the system block diagram.

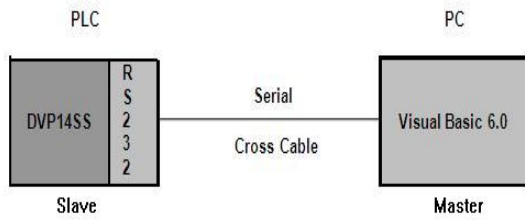


Fig 1 Block Diagram

Table 1 shows the function codes supported by the DVP14SS PLC. By using function code Master device can request data such as, Read coil status, Read input status, Read Holding Register, Force Single coils, etc. Slave device will repeat this function code in its reply to indicate that slave has received the function code and acted upon it [3]. Default protocol used by the delta plc is ASCII mode, 9600baud rate, 7 bit data length, even parity and 1 stop bit [4].

TABLE I
FUNCTION CODE FOR DELTA PLC

Code	Name	Description
01	Read Coil Status	S, Y, M, T, C
02	Read Input Status	S, X, Y, M, T, C
03	Read Holding Registers	T, C, D
05	Force Single Coil	S, Y, M, T, C
06	Preset Single Register	T, C, D
15	Force Multiple Coils	S, Y, M, T, C
16	Preset Multiple Register	T, C, D
17	Report Slave ID	None

Table 2 shows the Query format to read Input Status of contacts Y20 to Y47 from slave device address 01

TABLE II QUERY TO READ INPUT STATUS Y20 - Y47

Field Name	Example (Hex)
Heading	3A
Slave Address	01
Command code	02
Starting Address Hi	05
Starting Address Lo	14
Number of Points Hi	00
Number of Points Lo	25
Error Check (LRC)	BF

Table 3 shows the response from slave device to Query shown above.

TABLE III RESPONSE FROM SLAVE

Field Name	Example (Hex)
Slave Address	01
Command code	02
Bytes Count	05
Data (Coils Y033...Y024)	CD
Data (Coils Y043...Y034)	6B
Data (Coils Y053...Y044)	B2
Data (Coils Y063...Y054)	0E
Data (Coils Y070...Y064)	1B
Error Check (LRC)	E5

The PLC receives the messages without a communication error, but cannot handle it, an exception response will return to the master device. In the exception response, the most significant bit of the original command code is set to 1, and an exception code explains the condition that caused the exception is returned. Table 4 shows message response with exception code. [3] [4]

TABLE IV EXCEPTION RESPONSE FROM SLAVE

Field Name	Example (Hex)
Heading	3A
Slave Address	01
Function	81
Exception Code	02
Error Check (LRC)	7C

Table 5 shows different exception codes available in DVP14SS PLC [3] [4].

TABLE IIIV EXCEPTION CODES

Code	Meaning
01	Illegal command code: The command code received in the command message is not available for the PLC.
02	Illegal device address: The device address received in the command message is not available for the PLC.
03	Illegal device value: The device value received in the command message is not available for the PLC.

07	<p>Check Sum Error Check if the check Sum is correct Illegal command messages The command message is too short. Command message length is out of range.</p>
----	---

Visual Basic has ActiveX MSComm control 6.0 to send and receive ASCII characters on communication port. MSComm control added through the project component menu as shown in fig 2.

MSComm control is a drag and drop tool and can be accessed, handled with its properties and event handler.

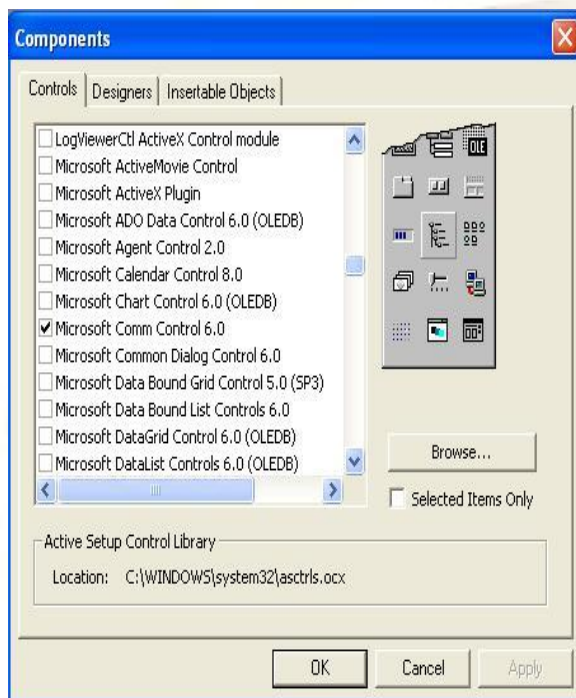


Fig 2 MSComm Control

Fig 3 shows the various properties of MSComm control.

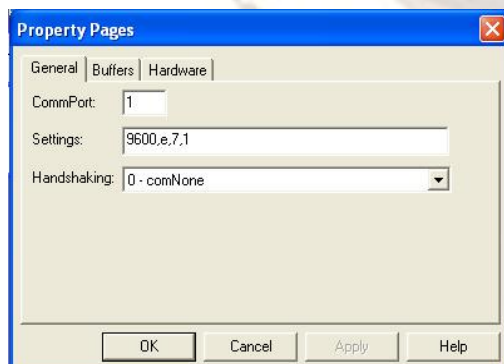


Fig 3 MSComm Control Property

III.IMPLEMENTATION

Following steps are implemented to obtain desired objective.

A. Knowing the Modbus Protocol for Delta PLC

DVP14SS series MPU support Modbus ASCII/RTU communication format with speed of up to 115,200bps. and modification on data length (data bits, parity bits, stop bits). Modbus communication format, CRC calculation, Modbus addressing for various internal devices of PLC are the key factor in the interface. Communication parameters for COM can be manipulated by writing ladder program. Here we use default protocol settings hence we do not alter PLC program. [4]

B. Writing Visual Basic code for MSComm Control

Designing windows form in visual basic as shown in fig 4. Form contains controls like Timer, MSComm Control, Button, textbox etc.

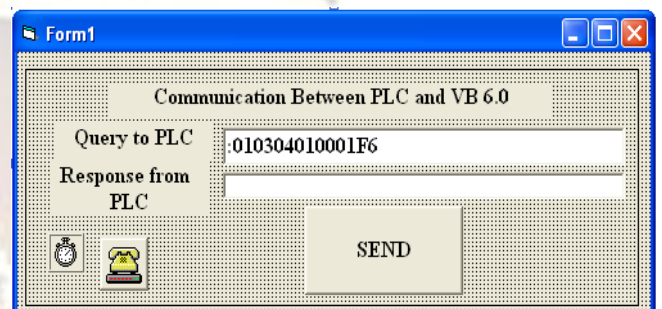


Fig 4 Form

OnComm () event in MSComm control is used to receive response from slave PLC.

```
Dim strRx As String
Select Case MSComm1.CommEvent
Case comEvReceive
Case comEvEOF
strRx = MSComm1.Input
```

Similarly MSComm.output property is used to send ASCII characters through serial port.

```
For Index = 0 To Len(txtWrite.Text) - 1
sendArr(Index) = Asc(Mid(txtWrite.Text, Index + 1, 1))
Next
```

```
sendArr(Index) = &HD
sendArr(Index + 1) = &HA
strRx = ""
MSComm1.Output = sendArr
Do While True
If MSComm1.InBufferCount > 0 Then
strRx = strRx + MSComm1.Input
If Len(strRx) > 2 Then
If Asc(Mid(strRx, Len(strRx) - 1, 1)) = &HD
And Asc(Mid(strRx, Len(strRx), 1)) = &HA Then
Text1.Text = Mid(strRx, 1,
Len(strRx) - 2)
Exit Do
End If
End If
```

