

# Temperature Control using PLC and LabVIEW Integration

Sachin S. Nerkar

Assistant Professor, Instrumentation Engineering,  
Government College of Engineering, Jalgaon.  
Maharashtra, India

*e-mail:sachinnerkar@rediffmail.com*

**Abstract**— Remotely controlled applications over a wide area have been commonly used in the industries today. In most of the industries direct temperature measurement is not possible. Most of the processes for various applications require control action by using Programmable Logic Controller (PLC) or LabVIEW. In this paper we are monitoring and controlling the process by establishing communication between PLC and LabVIEW. This can be done by Ethernet module of PLC and used for remote control of process. Nowadays PLC can not only control by its original software but also by third party software such as LabVIEW. The model developed and mentioned in this paper uses LabVIEW software which provides programming, monitoring, logging and human machine interface for the temperature control of process. It is easy to integrate different hardware simultaneously which exploits the multitasking and multiprocessing environment of LabVIEW in order to achieve the control objective.

**Keywords**- Programmable Logic Controller, LabVIEW, Human machine interface.

\*\*\*\*\*

## I. INTRODUCTION

The modern sensing technology and control methods are under continuously updating due to innovation, where the real-time temperature control is demanding higher accuracy and faster response more than the earlier systems. Temperature control is widely used in production and industrial control processes in all aspects of controlling quality and maintaining quantity of production and market demands. The PLC based Industrial Temperature Control System: Design and Implementation. [3]

Programmable logic controller (PLC) have eased the engineering design and lessen materials required, it is because the entire design is implemented in software programming paradigm. PLC has been commonly used in the industry, including controlling induction motor inverter fed variable drive system. Design distance control machinery is now possible, even by using Ethernet as the communication device between the computer and the PLC. [1]

The convenience part of PLC is the accessibility and controllability by other software. The communication between LabVIEW and PLC is established in different methods. Modbus technology is one of them. We are proposing here Modbus Ethernet protocol based communication of PLC with the LabVIEW and process equipments in this paper.

Modbus TCP and Modbus Serial are two of the most commonly used industrial protocols/networks on the market. NI LabVIEW 11.0 introduced native Modbus TCP and Modbus Serial support on any Ethernet or serial port with two LabVIEW add-on modules: LabVIEW Real-Time and LabVIEW DSC. Either of these modules allows creating a Modbus TCP or Modbus Serial I/O server through a graphical configuration assistant. With just a few clicks of the mouse, one can create a Modbus Master or Slave and specify the different registers to read and write.[4]

Proper monitoring of process can maintain operations at an optimal level by identifying and correcting problems before they turn into significant system failures. SCADA systems have made substantial progress over the recent years in terms of functionality, scalability and performance. The MODBUS protocol is employed while communication among the multilayer SCADA architecture.

## II. PLC TO LABVIEW INTEGRATION

Both process and equipments manufacturers are under the pressure, to maximize efficiency, reduce downtime, and improve yield. Manufacturers recognize that the key to making these improvements is timely and accurate flow, logging, recording and analyzing the information. However, information development is complicated task because of mix of hardware and software that exists in plants and the limitation of traditional systems. We can incorporate LabVIEW into the existing system to add measurements and analysis to our processes to gather complex data and convert it into useful information. Additionally, through the openness of LabVIEW we can connect and provide the information where it is needed, whether that is in a database, available on a web based dashboard, or provided into the existing plant control hardware. We can even use LabVIEW to close feedback loops and perform complex control operations using the provided information. [5]

A personal computer (PC), with National Instrument LabVIEW Version 2011 is used to develop the HMI layer. LabVIEW VI can implement HMI functionalities such as sense the level, ON/OFF of Motor and Solenoid Valve etc. It gives us opportunity to show how Modbus solve this compatibility problem. The Modbus protocol is utilized [5] for data exchange between supervision control layer and RTU layer. LabVIEW works like a Master and PLC as Slave. The

automation system is developed based on LabVIEW, though the serial communication, issuing commands to the PLC. PLC will work on the control system, real-time access and store various sensor signals after receiving the control command from the PC. It was easy to develop an automated system using PLC and LabVIEW via Modbus. National Instruments provides a Modbus Library for LabVIEW that is design to facilitate communication to a device that can use a Modbus protocol.[6][8]

**A. Requirements for integration**

1. LabVIEW capable for interface with High-Speed Measurements from a Broad Range of Sensors.
2. LabVIEW capable for Sharing Data between PLCs and other Automation Devices.
3. LabVIEW for Developing Graphical User Interfaces.

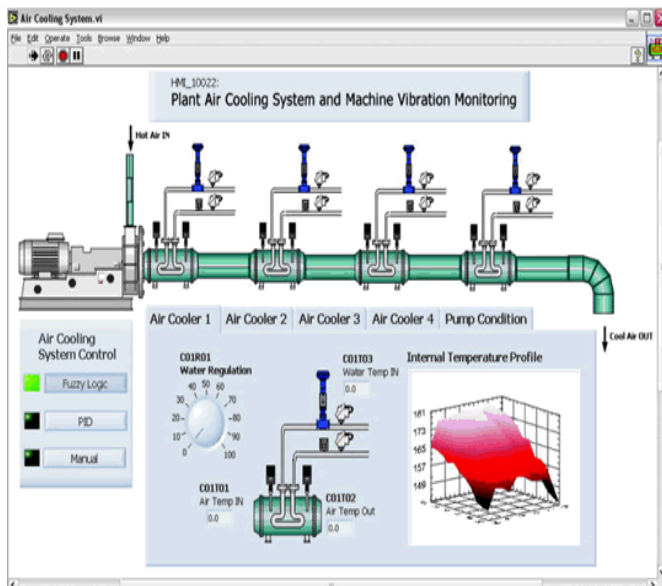


Figure 1: User Interface Development with LabVIEW

**B. Communication Procedure**

The LabVIEW Data logging and Supervisory Control (DSC) Module is used in this tutorial. This module includes tools for logging data to a networked historical database, real-time and historical trending, managing alarms and events, networking LabVIEW Real-Time targets and Modbus devices into one complete system, and adding security to user interfaces. With these features, LabVIEW becomes a powerful HMI/SCADA package for industrial control applications.[9]

**III. HARDWARE, SOFTWARE DESIGN REQUIREMENTS AND PROCEDURE**

- Windows 7/Vista/XP/Server 2003 R2/Server 2008 R2
- NI LabVIEW Full Development System and LabVIEW DSC

- Any Modbus supported PLC (eg. Schnedier)

**A. Creating a Modbus Master I/O Server**

In this section, a LabVIEW interface to the Modbus addresses called an I/O Server is created. The I/O Server automatically updates LabVIEW with the current values. [9]

1. In the Getting Started window of LabVIEW, Click File» New Project.  
 This opens a new LabVIEW Project.
2. If the Context Help window is not visible, press Ctrl + H to display the window. Keep this window open for helpful information about items under your cursor.
3. In the LabVIEW Project window, right-click My Computer and select New»I/O Server, as shown in Figure

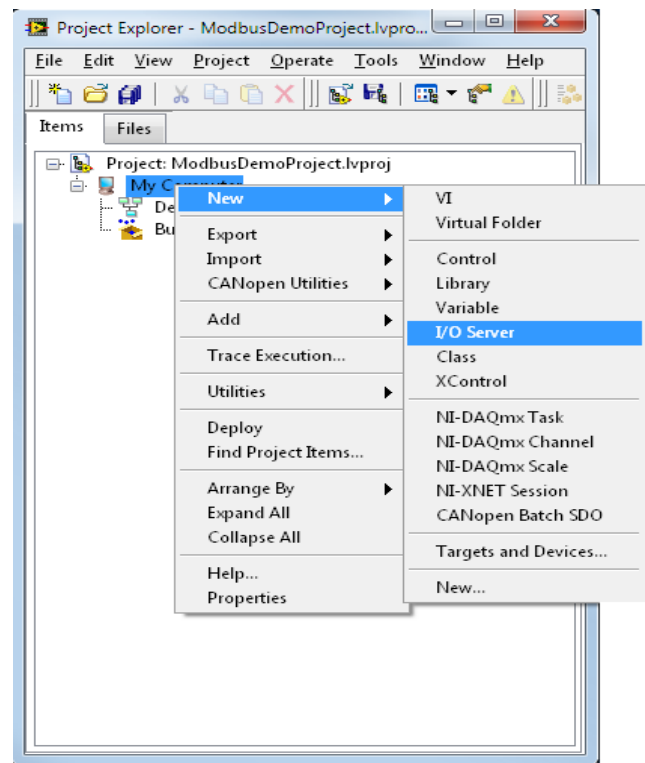


Figure 2: For creating a New I/O Server through the LabVIEW Project

3. Select Modbus in the Create New I/O Server Window and click Continue.
4. Select Modbus Ethernet from the Model drop down menu.
5. Enter the IP Address of your target Modbus Slave in the IP address field. In this example we are running the Modbus Slave on the same computer as the master, so we use the local host IP address, 127.0.0.1.
6. Select OK. A library is automatically created in your Project Explorer window to manage the I/O Server.

7. Save the project as Modbus Demo Project and the library as Modbus Demo Library by selecting File»Save All from the project explorer window.

**B. Bind Shared Variables to Modbus Addresses Through the I/O Server**

1. Right-click the newly created Modbus Demo Library and select Create Bound Variables...
2. In the Create Bound Variables window, expand the tree down to the Modbus1 in order to select the desired address range.

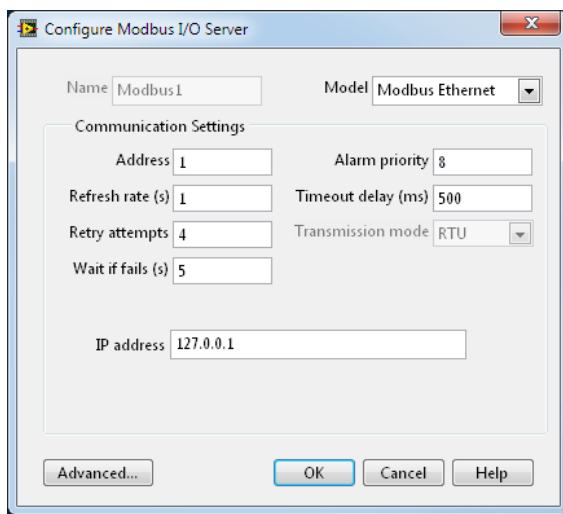


Figure 3: for configuring Modbus Master I/O server

3. Then select 000001-065535, and click on Add Range. Leave the default of Data Item as 000001 and Number of items as 1. Hit OK.

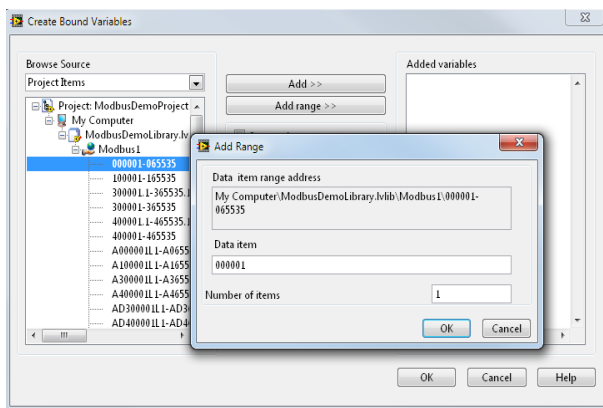


Figure 4 Select Modbus Addresses to Bind to Shared Variables

1. Your configuration window should now have variable 000001 in the **Added variables** section.
2. Click **OK**. This creates a shared variable that is bound to the Modbus address and loads it into the Multiple Variable Editor.
3. In the Multiple Variable Editor, select **Done**. This adds the new shared variables to the library that was created earlier.

4. In the Project Explorer right click on **000001** and select **Rename**. Rename the variable to **Coil1**.

You now have access to Modbus data natively in LabVIEW through the shared variables.

**C. Viewing Shared Variables with Distributed System Manager**

From the Project Explorer window, select Tools»Distributed System Manager. This opens a window that you can use to manage your shared variables in various ways (view, deploy, undeploy, etc.).

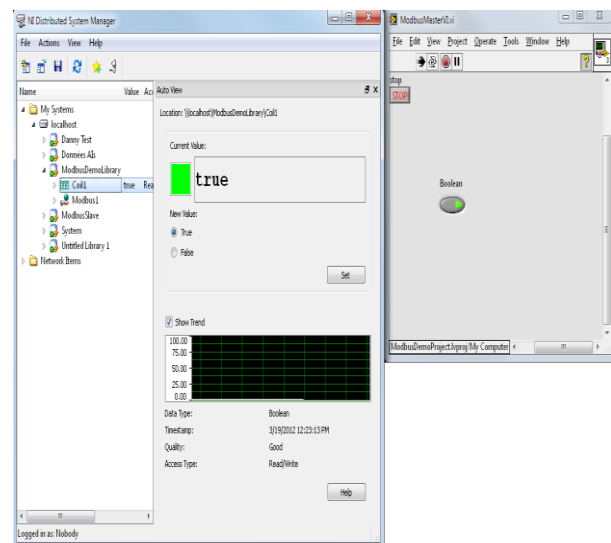


Figure 5: Viewing Coil1's value in Distributed System Manager

In the Tree pane of the Variable Manager, expand the local host item under the My Systems category. Locate and expand the Modbus Demo Library library, and select Coil1 to display Auto View on the right hand side of the window.[10]

You can now toggle the **Boolean** push button your front panel and see the result in Distributed System Manager like in Figure 5.

**IV. PROCESS MONITORING AND CONTROL**

We can integrate any Modbus communication supported PLC to LabVIEW in this setup I have used the Schneider Electric PLC and also used its analog card.

Here a process of simple reactor tank measurement and control operation is considered. In this process all necessary control signals are generated by PLC, but monitoring, Logging and recording the whole process from LabVIEW.[5]

Sequence of Operation:

1. When start button is pressed the analog value from R to V converter of RTD will be displayed in the form of bit register in PLC
2. In this condition Heater will be on.
3. As given set point is reached the bit value is reset and Heater is turned off.
4. Now if temp goes below set point bit value is set and heater is turned on this process is continuous and a continuous control and monitoring of temperature is achieved.

V. SCHNEIDER PLC PROGRAMMING

Analog value coming from the RTD (output resistance is converted into analog voltage by R to V converter) AI 0 is displayed in the input display and calibrating bit value in the PLC is showing in the PLC Program. As analog input value reach to set point i.e. bit value 1860 at 88 degree centigrade the heater is gets off. If the temperature of water gets reduced below 1860 bit value the Heater will be on. [10]

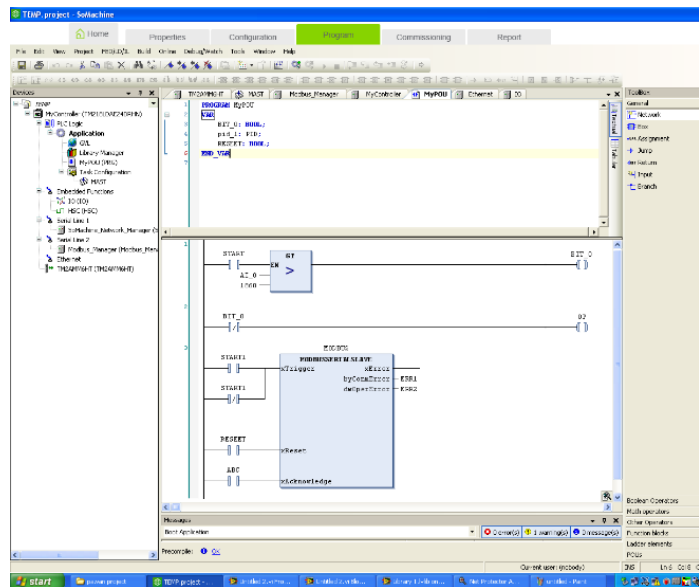


Figure 6: PLC program of temperature control process

The PLC program also contains the Modbus Serial Slave box which can communicate the PLC with Labview by using Ethernet TCP/RTU as shown in below figure.

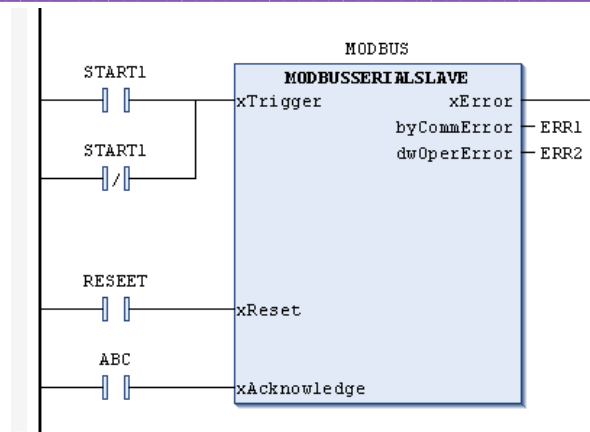


Figure 7: Modbus block in Schneider PLC.

In LabVIEW there is a Modbus demo library in which there is a wide range coils (data registers) are available. The data registers of range 000001-065535. From this range we can use the following range of data in LabVIEW as shown in figure8.

Useful range of coils is

000001-400001

A1L000001, A100001L1

A300001L1, A400001L1, AD300001

Now this coil's having the Boolean output this is converted in to binary by using the Boolean to binary converter as shown in block diagram of figure 9. These binary outputs then we have to convert into bit values by using multipliers constants and also this final will get the temperature indications shown in block diagram of coding in figure 8.

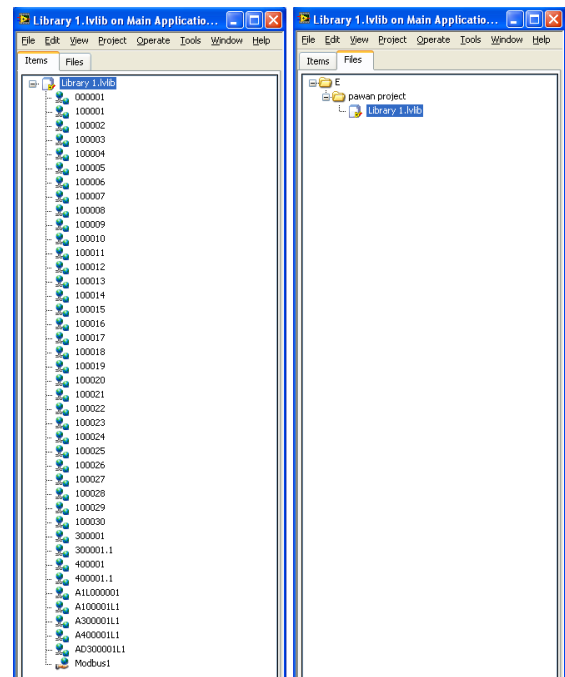


Fig. 8: Modbus Library and coil's used for communication

## VI. LABVIEW PROGRAMMING

LabVIEW is a powerful and versatile analysis and instrumentation software system for measurement and automation. It is used for communicating with devices such as process instruments, PLCs, smart sensors, or single-loop controllers. LabVIEW offers a variety of reliable and easy-to-use tools to meet any of the communication needs. LabVIEW supports Object linking and Embedding (OLE) for Process Control (OPC) for information exchange among disparate automation devices. LabVIEW includes compatibility with OPC Data Access 3. OPC specification increases performance and reliability for accessing real-time data from process control hardware and software. With the MODBUS library for National Instruments LabVIEW, or the built in MODBUS I/O server in the DSC module you can use any Ethernet or serial port as a Modbus TCP or Modbus serial master or slave. Using the Modbus I/O server with communication gateways, you can easily incorporate existing equipment on any industrial network to LabVIEW application. [9]

Figure 9 shows front panel of the LabVIEW where the temperature is measured by RTD and its analog value which is controlled by PLC, is displayed in LabVIEW by Modbus communication. The front panel also shows the status of Heater whether it is on or off. If heater is on then GREEN indication displayed and if heater is off then RED indication displayed. The temperature is also displayed in degree Celsius. This shows the temperature of the process. The graph shows the changing of temperature of amplitude versus time.[10]

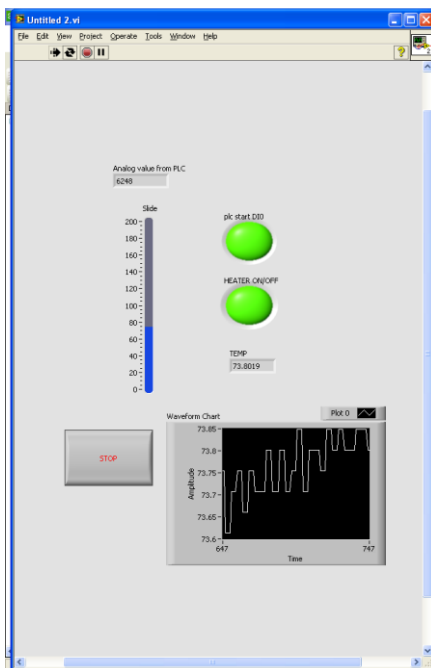


Figure 9: Front panel of LabVIEW coding for temperature monitoring and control.

The figure 10 shows the Block diagram of the Temperature control and monitoring system. The block diagram contains the graphical source code of a LabVIEW program.

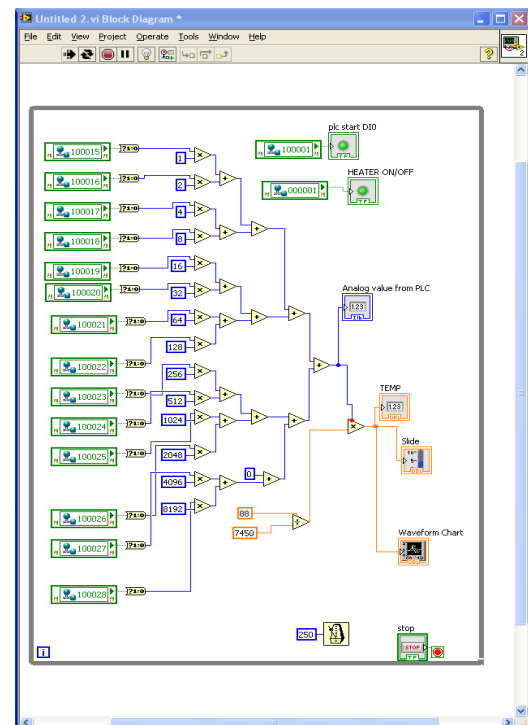


Figure 10: Block diagram of LabVIEW Program

## VII. RESULT

In the industrial control process, Temperature control is a very important especially in Interacting systems. It can have a very good effect in quality control of products and chemical processes. Figure 9 shows, attractive GUI based Front Panel for Temperature control application using PLC and LabVIEW integration. This Front Panel indicators turn ON and OFF as per the process changes. This shows the proper execution of LabVIEW program and obtains the desired results. These ON-OFF operations are visualized on the front panel in the form of ignition of LEDs and waveform chart. This represents the successful communication of the PC, PLC and Prototype setup using LabVIEW via Modbus.

## VIII. CONCLUSION

The LabVIEW and PLC integration for monitoring and controlling parameter provides high-speed monitoring and connectivity to enterprise level systems or plant. The schneider PLC is generating actual control signal but the entire things are monitored from the graphical user software like LabVIEW. The design of the temperature control system is successfully implemented in our Laboratory.

LabVIEW software enhances the performance of systems without spending development time and enables to handle the complexity of application. Thus the PLC can be

integrated with LabVIEW for controlling several real time applications in industry for effective control of parameters.

#### REFERENCES

- [1] Fanjie Wei, "The PLC-based Industrial temperature Control System: Design and Implementation", MATEC Web of Conferences 100:03031 · January 2017.
- [2] Gray Johanson, "Labview graphical programming", second edition, MC Graw Hill, New york, 1997.
- [3] Hossain A., Rashid Muhammed H., "The hardware and software interface of a programmable logic controller to an industrial grade process control system", Industry Applications Society Annual Meeting, IEEE, ISBN: 0-87942-553-9, Vol.-2, pp.- 1862 – 1868, 7-12 Oct. 1990.
- [4] "A case study of modern PLC and LabVIEW controls" Published in: FIW, 2011 7-8 Nov. 2011
- [5] Anjali S. Ashtekar, Bhagsen J. Parvat, Chandrakant B. Kadu, "Application of MODBUS to Communicate the PLC and LabVIEW for Real Time Process Control", IJESE ISSN: 2319-6378, Volume-1, Issue-11, September 2013.
- [6] Mohamed Endi, Y. Z. Elhalwagy and Attalla hashad, "Three-Layer PLC/SCADA System Architecture in Process Automation and Data Monitoring", IEEEEC – 2010, Volume 2, 978-1-4244-5586, pp 774-779.
- [7] Xiaolong Li, Munigala S., Qing-An Zeng, "Design and Implementation of a Wireless Programmable Logic Controller System", Electrical and Control Engineering (ICECE), International Conference, ISBN: 978-1-4244-6880-5, pp. 3138 – 3141, 25-27 June 2010.
- [8] Nargalkar Akshay, K. Uday Sravanth, Rahul Varanasi and J. Ankitha Reddy, "Real Time Automated Control of Industrial Processes with PLC –LABVIEW Communication", IJRSAT ISSN: 2319-2690 Issue-1, Volume-1, pp 035-038.
- [9] www.ni.com.
- [10] Schneider PLC Manual.