

**IMPLEMENTATION OF ONLINE TEMPERATURE
CONTROLLER BASED ON LabVIEW**

*Thesis submitted in partial fulfillment
of the requirements for the Degree of*

**Master of Technology
in
Electronics and Instrumentation Engineering**

By

Abhyarthana Bisoyi

Roll No: 211EC3377



Department of Electronics & Communication Engineering

National Institute of Technology, Rourkela

Odisha- 769008, India

May 2013

**IMPLEMENTATION OF ONLINE TEMPERATURE
CONTROLLER BASED ON LabVIEW**

*Thesis submitted in partial fulfillment
of the requirements for the Degree of*

**Master of Technology
in
Electronics and Instrumentation Engineering**

By
Abhyarthana Bisoyi
Roll No: 211EC3377

Under the Supervision of
Dr. Umesh Chandra Pati



**Department of Electronics & Communication Engineering
National Institute of Technology, Rourkela
Odisha- 769008, India
May 2013**



Department of Electronics & Communication Engineering
National Institute of Technology, Rourkela

CERTIFICATE

This is to certify that the Thesis Report entitled “**IMPLEMENTATION OF ONLINE TEMPERATURE CONTROLLER BASED ON LabVIEW**” submitted by **Ms. ABHYARTHANA BISOYI** bearing roll no. **211EC3377** in partial fulfillment of the requirements for the award of Master of Technology in Electronics and Communication Engineering with specialization in “**Electronics and Instrumentation Engineering**” during session 2011-2013 at National Institute of Technology, Rourkela is an authentic work carried out by her under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.



Dr. Umesh Chandra Pati

Associate Professor

Dept. of Electronics and Comm. Engineering

National Institute of Technology

Rourkela-769008

Place:

Date:

*Dedicated to my
family and friends*

ACKNOWLEDGEMENT

The satisfaction and euphoria on the successful completion of any task would be incomplete without mentioning the people who made it possible whose constant guidance and encouragement crowned out effort with success.

I would like to express my heartfelt gratitude to my esteemed supervisor, **Dr. Umesh Chandra Pati** for his technical guidance, valuable suggestions, and encouragement throughout the experimental and theoretical study and in preparing this thesis. It has been my honour to work under his guidance, whose expertise and discernment were keys in the completion of this project.

I am grateful to the **Dept. of Electronics & Communication Engineering**, for giving me the opportunity to execute this project, which is an integral part of the curriculum in M.Tech programme at the National Institute of Technology, Rourkela.

Many thanks to my friends who are directly or indirectly helped me in my project work for their generous contribution towards enriching the quality of the work. I would also express my obligations to Prof. S. K. Patra, Prof. K. K. Mahapatra, Prof. S. Meher, Prof. T. K. Dan, Prof. S. K. Das, and Prof. Poonam Singh.

This acknowledgement would not be complete without expressing my sincere gratitude to my family for their love, patience, encouragement, and understanding which are the source of my motivation and inspiration throughout my work. Finally I would like to dedicate my work and this thesis to my parents and my brother.

Date:

Place:

Abhyarthana Bisoyi

Roll No: 211EC3377

Dept. of ECE

NIT,Rourkela

ABSTRACT

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) is the software which gives virtual existence of hardware, reduces its cost and hence termed as Virtual Instrumentation. This thesis presents the implementation of ON/OFF and PID controller for controlling the temperature of a heating element inside a wooden box with the help of LabVIEW. In this software, DataSocket Protocol and Transmission Control Protocol (TCP) are used for developing an online transmission process between client and server. Client has control over the set point and Server has control over the temperature. With the help of internet protocol, client provides the value of set point according to which the control actions are taken by the server. The paper also includes discussions regarding the advantages and disadvantages of TCP/IP.

Written communication in LabVIEW refers to communication via internet where the user can write messages to another user. Generally written communication is not found in any programming related software but LabVIEW makes this possible using internet protocol. This helps in easy and fast transmission of data in form of messages between the client and server PC(s). After the transmission process gets over, the transmitted data is used for implementing PID controller. The various responses in PID controller are obtained by varying the tuning parameters. The responses are with respect to three conditions in the controller. On studying the response curves, parameters like setting time, rise time and maximum overshoot are obtained. A comparison table is obtained for these parameters and helps in deciding to choose the values for tuning parameters of the controller.

TABLE OF CONTENTS

Particulars	Page No.
Acknowledgements.....	i
Abstract.....	ii
Table of Contents.....	iii
List of Figures.....	vi
List of Tables.....	viii
Abbreviations.....	ix
1. INTRODUCTION.....	2
1.1 Overview.....	2
1.2 Motivation of Work.....	3
1.3 Background Literature Survey.....	4
1.4 Objective of Work.....	5
1.5 Thesis Outline.....	6
2 THERMAL PROCESS.....	8
2.1 Proposed System.....	8
2.2 Temperature Controllers.....	11
2.2.1 Steps to control temperature.....	11
2.3 ON/OFF Controller.....	12
2.3.1 Application of ON/OFF controller.....	14
2.4 PID Controller.....	14
2.4.1 Proportional.....	15
2.4.2 Integral.....	16
2.4.3 Derivative.....	16
2.4.4 Stability.....	16
2.4.5 Manual tuning.....	16
2.4.6 Characteristics of PID controller.....	17

2.4.7	Steps for designing a PID controller.....	17
2.4.8	PID palette in LabVIEW.....	18
3	IMPLEMENTATION OF DATASOCKET PROTOCOL.....	20
3.1	VI in LabVIEW.....	20
3.1.1	Front panel.....	20
3.1.2	Block diagram.....	21
3.2	DataSocket Overview.....	22
3.3	Using DataSocket in LabVIEW.....	22
3.3.1	DataSocket open.....	23
3.3.2	DataSocket close.....	24
3.3.3	DataSocket write.....	24
3.3.4	DataSocket read.....	24
3.4	Internet Communication Using DataSocket Protocol.....	25
3.4.1	Server DataSocket program implementation.....	25
3.4.2	Client Data Socket program implementation.....	25
3.4.3	Algorithm for implementation of ON/OFF controller.....	25
3.4.4	Algorithm for manual tuning of PID controller.....	27
3.5	Results and Discussion.....	27
3.5.1	Implementation of ON/OFF controller using Data Socket Protocol.....	27
3.5.2	Implementation of PID controller using DataSocket protocol.....	33
3.5.3	Implementation of multiple client-server communication.....	37
4	IMPLEMENTATION OF TCP/IP.....	41
4.1	Internet Protocol.....	41
4.2	User Datagram Protocol.....	41
4.3	Transmission Control Protocol.....	41
4.3.1	Using TCP connections in LabVIEW.....	41
4.3.2	Algorithm for using TCP/IP:.....	43

4.3.3	Advantages of TCP/IP	44
4.3.4	Disadvantage of TCP/IP	44
4.4	Online Written Communication.....	44
4.4.1	Algorithm for written communication.....	44
4.5	Results and Discussion.....	45
4.5.1	Implementation of ON/OFF controller using TCP/IP	45
4.5.2	Implementation of PID controller using TCP/IP	46
4.5.3	Implementation of online written communication.....	48
5	CONCLUSION	51
5.1	Limitations of the Thesis.....	51
5.2	Scope for Future Research	52
	References.....	54
	Dissemination.....	56

LIST OF FIGURES

Figure No.	Page No.
Figure 1.1: Block diagram of a control system.....	2
Figure 2.1: Block Diagram of the Proposed System.....	9
Figure 2.2: Flow Diagram of the project	10
Figure 2.3: General Block Diagram.....	12
Figure 2.4: Basic block diagram of a conventional PID controller	15
Figure 2.5: PID Palette.....	18
Figure 3.1: Front Panel of LabVIEW	20
Figure 3.2: Block Diagram of LabVIEW	22
Figure 3.3: DataSocket Server	23
Figure 3.4: DataSocket Open.....	23
Figure 3.5: DataSocket Close	24
Figure 3.6: DataSocket Write	24
Figure 3.7: DataSocket Read	24
Figure 3.8: Simple DataSocket Protocol Implementation	28
Figure 3.9: Output when $SP < PV$ and PV is Within Limit.....	28
Figure 3.10: Output when $SP < PV$ and PV is Out of Limit.....	29
Figure 3.11: Output when $SP > PV$ and PV is Within Limits	30
Figure 3.12: Output when $SP > PV$ and PV is Out of Limits	31
Figure 3.13: Output when $SP = PV$	31
Figure 3.14: Upper portion of the block diagram of server	32
Figure 3.15: Lower portion of the block diagram of server.....	32
Figure 3.16: Front Panel of PID controller	33
Figure 3.17: Front panel of client (PID controller).....	34
Figure 3.18: Block diagram of client (PID controller)	34
Figure 3.19: Graph showing under damped response.....	35
Figure 3.20: Graph showing critically damped response.....	36
Figure 3.21: Graph showing over damped response.....	36
Figure 3.22: Flow of data among 2 clients and a server	37
Figure 3.23: Front Panel of Server.....	38
Figure 3.24: Front Panel of Client1	38
Figure 3.25: Front Panel of Client2	39

Figure 4.1: TCP/IP Listen Icon.....	42
Figure 4.2: TCP/IP Write Icon.....	42
Figure 4.3: TCP/IP Read Icon.....	43
Figure 4.4: Front Panel of ON/OFF controller Server.....	46
Figure 4.5: Block Diagram of ON/OFF controller Client.....	46
Figure 4.6: Front Panel of PID controller Server.....	47
Figure 4.7: Front Panel of PID controller Client	47
Figure 4.8: Front Panel of Communicating Server.....	48
Figure 4.9: Front Panel of Communicating Client.....	49

LIST OF TABLES

Table No.	Page No.
Table 2.1: Effect of PID Controllers on Closed-Loop System	17
Table 3.1: Comparison between response curves	37

LIST OF ABBREVIATIONS

LabVIEW	Laboratory Virtual Instrument Engineering Workbench
DAQ	Data Acquisition
DAS	Data Acquisition System
PC	Personal Computer
DS	Data Socket
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
IP	Internet Protocol
VI	Virtual Instrument
PV	Process Variable
SP	Set Point
DSTP	Data Socket Transfer Protocol
NI	National Instrument
PID	Proportional Integral Derivative
PI	Proportional Integral
PD	Proportional Derivative
FTP	File Transfer Protocol
ANFIS	Adaptive Network-Based Fuzzy Inference System
NCS	Networked Control System
OLE	Object Linking and Embedding
OPC	OLE for Process Control
HTTP	Hyper Text Transfer Protocol
LAN	Local Area Network
DNS	Domain Name System
TC	Temperature Controller
CGI	Common Gateway Interface

Chapter 1

Introduction

Overview

Motivation

Background of Literature Survey

Objective of Work

Thesis Outline

1. INTRODUCTION

1.1 Overview

Control system is a system that controls a variable by using error-sensing through a closed loop. The variable is known as controlled variable that must be maintained or controlled at some desired value. The desired value of the controlled variable is known as set point. The variable used to maintain the controlled variable at its set point is known as manipulated variable. A closed loop refers to the condition in which the controller is connected the process comparing the set point with the controlled variable and determining corrective action. The objective of a control system is to use the manipulated variable and to maintain the controlled variable at its set point despite of disturbances. Disturbance is any variable that may cause controlled variable to deviate away from set point. Figure 1.1 represents the block diagram of a control system.

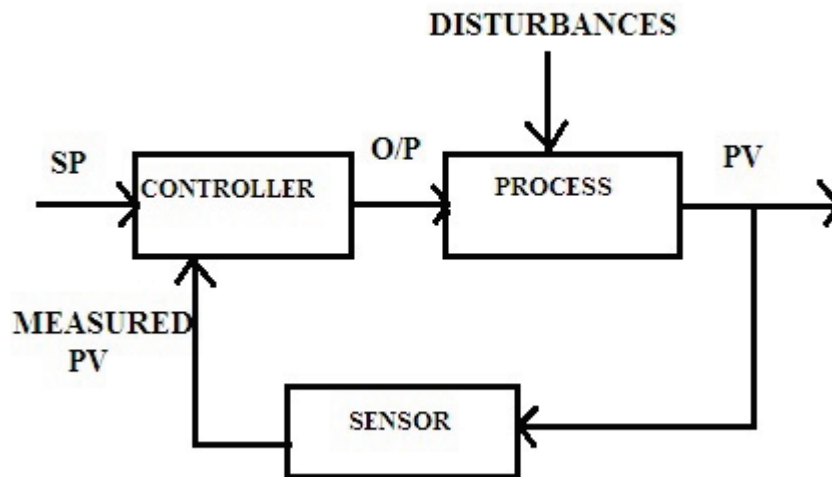


Figure 1.1: Block diagram of a control system

The basic controllers used to control a process variable at its set point are ON/OFF, P, PI, PID and PID fuzzy. There are various networking protocols used in LabVIEW (Laboratory Virtual Instrument Engineering Workbench) such as DS (DataSocket Protocol), TCP (Transmission Control Protocol), and UDP (User Datagram Protocol) which provides online facilities to the users. These are the basic tools for communication over internet.

Mainly controllers with combination of Proportional, Integral and Derivative are more effective than other controllers. As conventional controllers such as P, PI, PD, PID, Otto-Smith, all their different types and realizations, and other controller types have been counted on years. It is important for all conventional controllers to know a mathematical model of the process in order to design a controller. Unconventional controllers develop new approaches in the controller design where mathematical model of a process is generally is not needed. Examples of unconventional controller are a fuzzy controller and neuro or neuro-fuzzy controllers.

LabVIEW is a graphical programming environment which reduces the cost by giving a virtual existence of hardware, and hence termed as Virtual Instrumentation. This environment best suits for high-level or system-level design. The basic difference between “natural instrumentation” and “virtual instrumentation” is the software part of a virtual instrumentation. The software replaces complex and expensive equipment by simple and less expensive hardware. The program in this software consists of two parts: front panel and block diagram. Front panel shows the control and indicator while the main programming part is done in the block diagram which is usually kept up to user. Controls are the inputs and indicators are the outputs. Various operations have been done relating these controls and indicators.

1.2 Motivation of Work

In the academic and industrial communities, remote real-time control of processes is receiving considerable attention. In industries, when an instrument acquires some quantity and then measures it, its value has to be kept at some limit otherwise it may cause some problem. Hence the controllers are needed to solve the purpose. The controllers may function in controlling controlled variable such as temperature, pressure, flow, level, humidity, etc. Using LabVIEW software it becomes easy to design a controller. In earlier years, much research has been done on the controllers but not on the various internet protocols found in LabVIEW. The internet protocols can be used in order to develop communication among the users.

Data Acquisition (DAQ) is the process of measuring and processing any electrical or physical quantity such as voltage, temperature, humidity, pressure, etc. A “server-client” relationship is formed using these protocols. A server is a user who acquires and measures the

variable, performs operations and if requires any help then sends a request to the client. Client is a user who accepts the request from server and provides service in return. The protocols help in long distance communication by making it very easy and fast. The combination of controllers and protocols is more helpful in industries. The client and server are connected using DAQ.

Depending upon the system to be controlled, the user will choose the type of control for the process. Among the controllers, ON/OFF controller is the basic one that has been implemented followed by PID in order to compare. PID controller is best suited for industrial application since it gives zero offset because of I-component and better stability because of other components. Various technologies are developed in order to perform real time control using internet based technology. LabVIEW is one of the software packages used in process control application. It uses various protocols such as TCP/IP, DataSocket, etc. that allows remote control using internet. Many universities have developed internet based process control laboratories for the students that would be helpful for distance education.

1.3 Background Literature Survey

The literature survey begins with study of ON/OFF controller. In 2003, E. Rézaei [1] and in 2009, K. Perna [2] proposed a system where the temperature of a bulb inside a wooden box has been controlled by using ON/OFF controller, with the help of DS Protocol and also developed the algorithm. In 2011, A.O. Neaga, et al. [3] proposed a system to control liquid nitrogen to its set point using ON/OFF controller, but there are certain defects in the control loop. In 2011, R. Dutta [4] proposed a system where the temperature of a set of light bulbs inside a wooden box is controlled using DS Protocol and digital temperature sensor. In 2007, S. Murthy [5] described in their paper about protocols and the way these protocols help in improving networking capabilities of many measurement and automation applications. LabVIEW encourages such protocols in control applications.

In 2007, S. Zhong, et al. [6] proposed a self-tuning algorithm for PID parameter using method of recursion least square. In 2009, A. Nazir, et al. [7] presented a genetic algorithm for tuning PID parameters which reduced the computational requirement. In 2009, J. Liu, et al. [8] presented a real time development system showing the control of position in a dc servo motor using PID controller. He described about various PID algorithms to control position using DAQ.

In 2010, S. Ravi, et al. [9] developed an ANFIS controller for plastic extrusion system which gave better performance than fuzzy logic and PID controller. F. Faizan, et al. [10] implemented a discrete PID on pendulum with an idea to balance an inverted pendulum using PIC microcontroller. In 2011, S. K. Sahoo, et al. [11] discussed in their paper regarding the modulus hugging approach for designing PI and PID controller in order to control the speed of DC motors using various algorithms. In 2011, V. Kumar, et al. [12] discussed a comparative study on performance of various fuzzy PID with other fuzzy controllers. In 2011, J.H. Zhang, et al. [13] proposed an acquisition diagnostic system for remote ground equipment using LabVIEW. Hai-bo Lin [14] introduced an intelligent temperature controller where the simulation results show intelligent temperature controller has good control performance and higher accuracy using PID algorithm.

In 2010, S. Mohammadi, et al. [15] developed a fuzzy based PID controller that helps in reducing the packet loss while TCP communication. In 2003, D. Poulton, et al. [16] discussed in their paper about the fading effects of rain on TCP/IP packet based satellite links and proposed a PID delay controller based on non-uniform sampling filtering. In 2006, A. Wei, et al. [17] discussed in their paper about TCP in Networked Control System (NCS) and performance of different types of data result. In 2011, F. Xiao, et al. [18] introduced in their paper about CAN-TCP/IP gateway showing the application of webserver in that field. In 2012, Y. Li, et al. [19] described about vulnerabilities of TCP/IP by exposing it to attacks.

1.4 Objective of Work

The objective intends to measure and control the temperature of a heating element inside a box using controllers. This is performed by using LabVIEW software, by developing a “Client” and a “Server”. They communicate using DataSocket and TCP protocol where the server controls the temperature using ON/OFF and PID controller and the client has an option to decide the set point. The various advantages of internet protocols in LabVIEW have been discussed. The advantages include communication like chatting, sending information, etc. Client and server are two users that communicate using TCP/IP for sending their data. These data are used further in implementation of controllers. Here PID controller has been implemented and the various

conditioned outputs are obtained. A comparison table is obtained with different outputs of the controller.

1.5 Thesis Outline

Including the introductory chapter, the thesis is divided into 6 chapters. The organization of the thesis is presented below.

Chapter 2 - Thermal Process

The proposed system has been discussed in this chapter. The flow diagram of the process has been described with brief explanation on it. Introduction to the two controllers has been given.

Chapter 3 - Implementation of DataSocket Protocol

In this chapter, the control of temperature using DataSocket protocol has been discussed. The implementation algorithms of the two controllers have been described. The simulation results have been discussed. Communication of two clients with one server has been discussed.

Chapter 4 - Implementation of TCP/IP

In this chapter, the control of temperature using Transmission Control Protocol has been discussed. The implementation algorithms for the protocol have been described. The chapter also includes the simulation results obtained.

Chapter 5 – Conclusion

The overall conclusion of the thesis is presented in this chapter. It also contains some limitations of the thesis and future research topics.

Chapter 2

Thermal Process

Proposed System

Temperature Controllers

ON/OFF Controller

PID Controller

2 THERMAL PROCESS

This chapter discusses the proposed system, introduction to the two controllers, ON/OFF and PID controller and their algorithm.

2.1 Proposed System

The main objective is to maintain the temperature inside a wooden box which is heated by a heating element i.e. a bulb, at some desired value of set point which is selected by Client. The heating element is covered by a wooden box and on heating; the temperature is sensed by temperature sensor and is transmitted to the server PC with the help of DAQ card. Since, the output of sensor is in voltage form, the server PC converts it to temperature using Equation 2.1.

$$PV = 6. Vin + 60 \quad (2.1)$$

Where, PV is Process Variable

Vin is voltage measured by sensor

This value of temperature i.e. process variable (PV) is sent to client PC. The client PC on receiving PV sends set point (SP) value to the server PC. Depending upon the SP, ON/OFF and PID controller take necessary actions. The actions are in the form of signals that control the operation of fan. When the temperature crosses the neutral zone, controller turns a fan ON or OFF accordingly. Neutral zone is the zone bounded by low and high limit.

Figure 2.1 shows the proposed system block diagram. It shows the client server relationship along with the communication path between them.

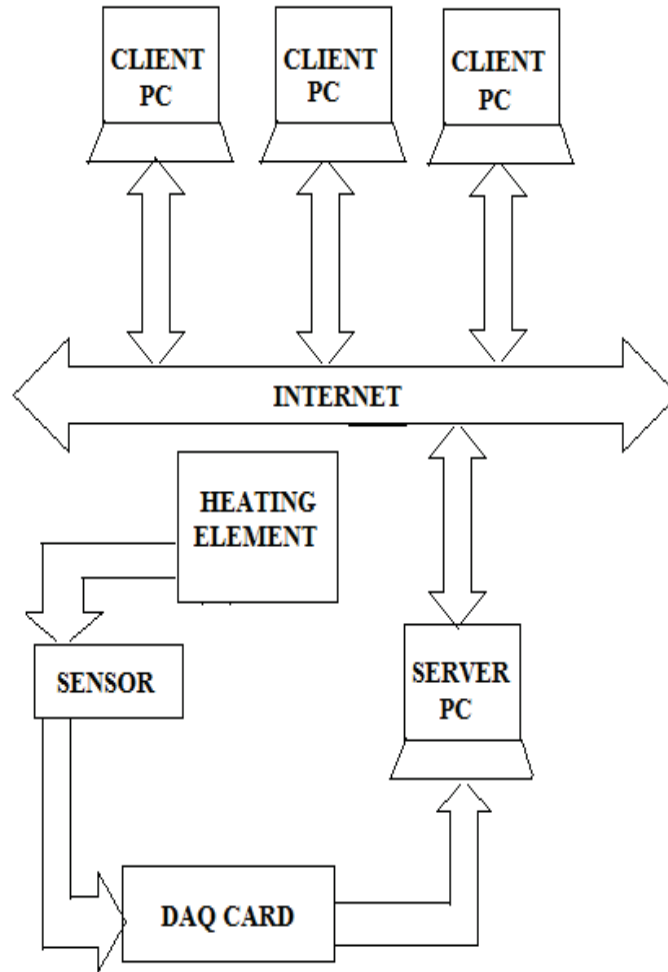


Figure 2.1: Block Diagram of the Proposed System

Figure 2.2 shows the flow diagram of the project including the involvement of proposed system. The transmission of information between client and server has been done by using TCP/IP and DataSocket Protocol. The approach for the goal can be easily summarized by a flow chart by which we can see the further proceedings. It just gives a generalized view but not specific for each controller. There is a difference between the two paths after “Data received by PC/Server” block. The first path can be used when hardware modeling is involved. The second path has been implemented in the software part.

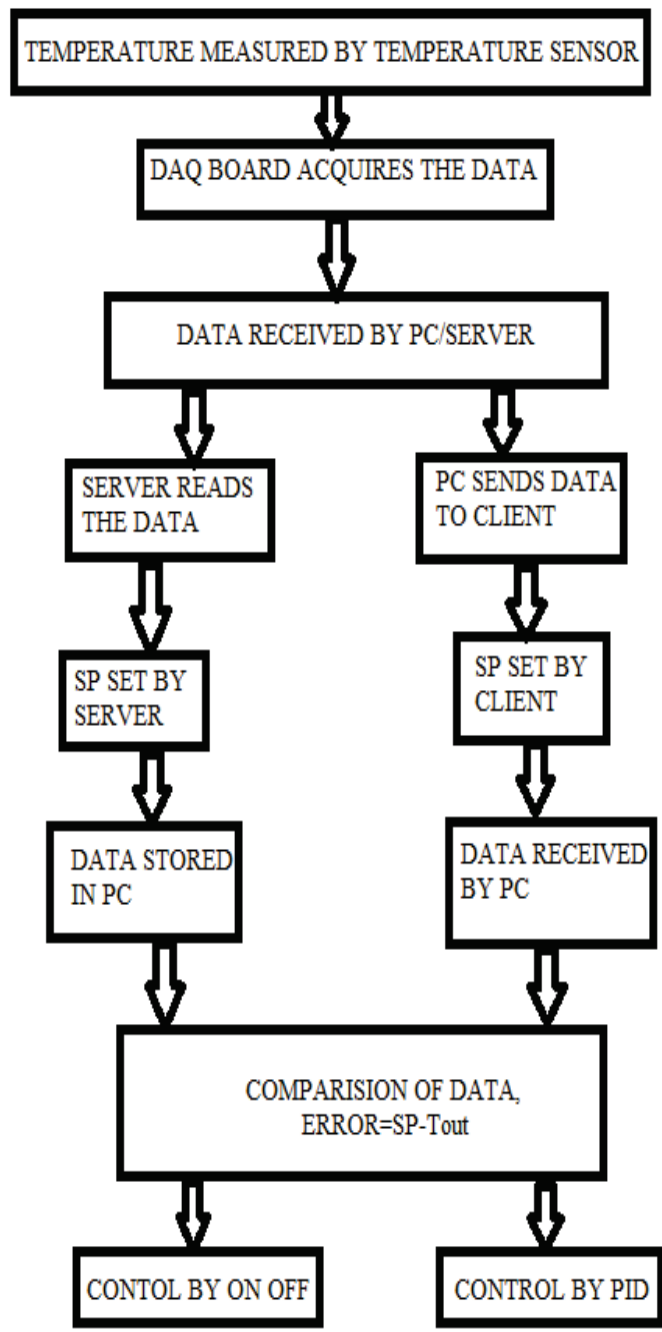


Figure 2.2: Flow Diagram of the project

2.2 Temperature Controllers

Temperature controller (TC) is an instrument that controls temperature. The temperature sensor senses the temperature and the value is given as an input to the TC. The temperature controller has an output that is connected to a control element for instance a heater or fan.

In order to accurately control a process temperature without an operator involvement, a temperature control system depends upon a controller, which uses a temperature sensor (Thermocouple or RTD) as an input. It compares the observed temperature with the desired temperature (set point), and obtains an output to a control element. The controller is just a part of the whole control system, but the entire system is after selecting the proper controller. The following points should be noted while selecting a controller:

- Type of sensor and the temperature range.
- Type of output required.
- Control algorithm needed.
- Number of output and its type.

In various applications, the above points will not be enough to measure a quantity. So it is needed to control a quantity. Either the quantity has to stay constant at some fixed value or the quantity can be varied in some manner decided by user. The control action has to be decided first.

2.2.1 Steps to control temperature

The temperature is the physical quantity that has to be controlled. The following steps are carried out in order to control temperature:

- The first step for controlling temperature is to measure it and to observe the value it has. So it is important to know the temperature.
- The Second step after measurement of temperature is to compare it with the desired temperature (Set Point) that is the temperature needed. It is essential to verify if the value is too high or too low and check the range.
- The last step after comparison is the decision of control action to be taken. There are various ways to control a variable.

When a control system is build up, it is useful to visualize how the signals are produced and the way it affects the system. The designers of a control system design a block diagram to show the signals flow in a control system. Figure 2.3 shows the block diagram of a typical control system.

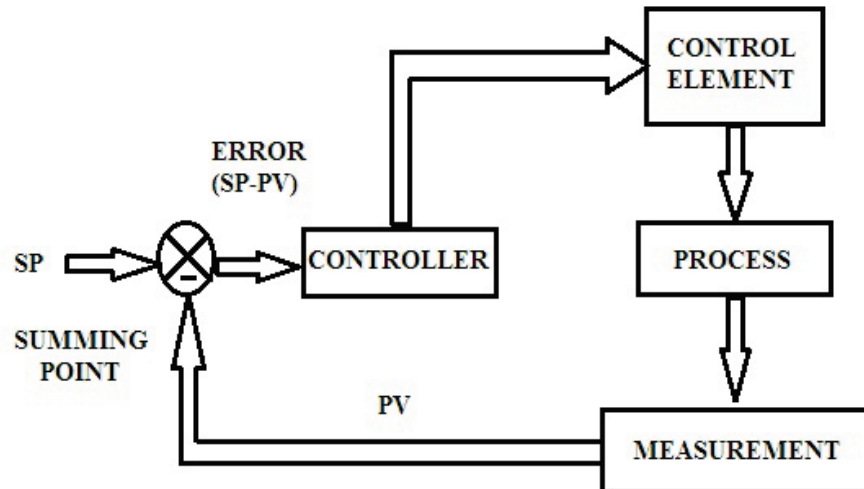


Figure 2.3: General Block Diagram

The block diagram consists of the following components:

- Process
- Measurement
- Error Detector
- Controller
- Control Element

2.3 ON/OFF Controller

An ON/OFF controller is a basic temperature controller. The output from the controller is either ON or OFF without any middle state. An ON/OFF controller will respond to the output only when the temperature deviates from the set point. An ON/OFF controller drives the manipulated variable from a fully closed condition to fully open depending upon the position of the controlled variable with respect to the set point. For controlling heat, the output is on when the temperature is less than the set point, and off when above SP. Since the temperature deviates from the set

point in order to change the output response, the temperature of the process will be monitored continuously.

If sudden actions are taken by heating switches turning ON and OFF at the instant the measured temperature deviates from the set point, then the system starts “chattering” repeatedly and then the system wouldn't last for a long time. In order to avoid this, practical ON/OFF controllers have a dead band around the set point. The dead band is also known as proportional band. When the measured value lies within this band the controller does not take any action. But when the value moves outside this band the action is taken. The reason behind this is to introduce set of oscillation for the value of controlled variable, which means that at larger dead band the amplitude obtained is higher and frequency is lower.

ON/OFF differential prevents chattering or making fast, if the deviation above and below the set point occurs very rapidly. The difference in temperature needs exceeding the set point by a certain amount before the output turns OFF or ON continuously. The demerit of ON OFF Controller is the occurrence of “Hunting”. This “ON and OFF” switching continues all the time, around the set point temperature. After the “Overshoot” and “Undershoot” the process “stabilizes” and “oscillates” around the Set point value. This is called “Hunting”. To prevent “oscillation” at set point Temperature the ON and OFF switching point are NOT the same. Otherwise the controller would not know if it has to switch ON or OFF at set point. The difference between the ON-switching point and the OFF point is called the “Hysteresis”. It is represented in equation 2.2.

$$u(t) = \begin{cases} U_{\max} ; \forall e(t) > 0 \\ U_{\min} ; \forall e(t) < 0 \end{cases} \quad (2.2)$$

Where:

$e(t)$ – control error (for unit feedback).

$u(t)$ – control signal (controller output).

ON/OFF controller have only two possible control signal values irrespective of the value of control error. Hence this controller is the simplest one among other temperature controllers. Depending on the error, whether it is positive or negative, the control signal $u(t)$ can

have two values, U_{\max} (high level) or U_{\min} (low level). The objective behind this way of control is that the two control levels attain desired value of controlled variable in shortest time possible. When the process has a positive static gain, high level control signal (U_{\max}), will increase the value of controlled variable. A drawback of control is that the control signal on oscillating may cause control variable to oscillate around the set point and at times there is no solution to this problem. The process is forced to oscillate since $u(t)$ is never zero, it is either U_{\max} or U_{\min} . The only way to avoid these forced oscillations is to reduce the gain for small values of error, $e(t)$ which can be achieved by introducing a proportional mode that remains active for certain values of error.

2.3.1 Application of ON/OFF controller

ON/OFF controller is mostly used where an accurate control is not demanded. It is implemented in systems which cannot handle heavy amount of energy to turn ON and OFF, where the mass of the system more, that temperatures change extremely slowly. One special kind of ON/OFF management used for alarm could be a limit controller. This controller uses a latching relay that should be manually reset, and is employed to close up a method once an exact temperature is reached.

2.4 PID Controller

PID controllers are the foremost widely-used style of controller for industrial application. It is conjointly referred to as “Three-term controller”. They are structurally easy and exhibit strong performance over large varies of operating conditions. Within the absence of the entire data of the method these varieties of controllers are the foremost economical of selections. The three main parameters concerned are Proportional (P), Integral (I) and Derivative (D). The proportional half is liable for following the required set-point, whereas the integral and derivative half accounts for the buildup of past errors and therefore the rate of modification of error within the method severally.

Derivative mode improves stability of the system and allows increase in gain K_p and decrease in integral time constant T_I , that will increases speed of the controller response. PID controller is employed once coping with higher order electrical phenomenon processes once

their dynamic is not like a dynamics of an integrator.

There are various methods for tuning a PID controller:

1. Manual Tuning.
2. Ziegler-Nichols Method

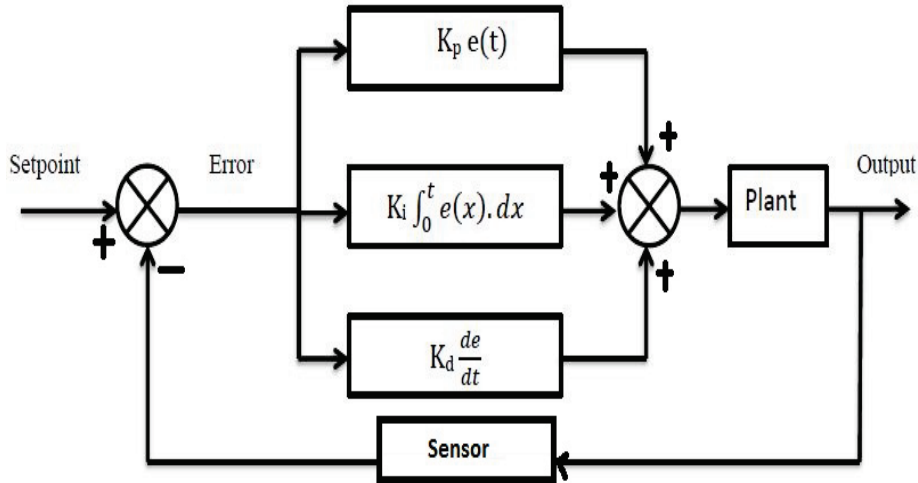


Figure 2.4: Basic block diagram of a conventional PID controller

For the PID controller represented in Figure 2.4,

Output of PID controller is represented by equation (2.3),

$$u(t) = K_p e(t) + K_i \int_0^{t_1} e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (2.3)$$

Where,

Error, $e(t)$ = Set point- Plant output

K_p = proportional gain, K_i = integral gain, K_d = derivative gain

Problem in structure of a controller arises during designing a control system, defining structure and controller parameters and tuning them.

2.4.1 Proportional

The proportional term is given by equation 2.4,

$$P_{out} = K_p e(t) \quad (2.4)$$

If K_p is high then the system becomes unstable. If it is too low then control action may not be effective for responding to disturbances.

2.4.2 Integral

The integral term is given by equation 2.5,

$$I_{out} = K_i \int_0^t e(\tau) d\tau \quad (2.5)$$

The integral term eliminates the steady-state error by the movement of the process towards set point. But it may cause overshoot so an additional component is required.

2.4.3 Derivative

The derivative term is given by equation 2.6:

$$D_{out} = K_d \frac{de(t)}{dt} \quad (2.6)$$

The derivative term reduces the overshoot and improves the stability of the process. But it slows the transient response of the controller.

2.4.4 Stability

If the tuning parameters of PID controller are not chosen properly, the process may become unstable and the instability is caused by excess gain. PID controller algorithm is given by equation 2.7,

$$u(t) = K \left[e(t) + \frac{1}{T_i} \int_0^{t_1} e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad (2.7)$$

$$\text{Or, } u(t) = Ke(t) + \frac{K}{T_i} \int_0^{t_1} e(\tau) d\tau + KT_d \frac{de(t)}{dt}$$

2.4.5 Manual tuning

The proportional, integral and derivative terms must be individually adjusted or "tuned" to a particular system using trial and error. It provides accuracy and stability for the process.

2.4.6 Characteristics of PID controller

Table 2.1 shows the comparison between rise time, overshoot, settling time, steady state error for different tuning parameters of PID controller. . Effects of each of controllers K_p , K_d , and K_i on a closed-loop system are summarized in the table shown below. Changing one of these variables can change the effect of the other two. For this reason, the table should only be used as a reference while determining the values for K_i , K_p and K_d .

Table 2.1: Effect of PID Controllers on Closed-Loop System

PARAMETERS	RISE TIME	OVERSHOOT	SETTLING TIME	S-S ERROR
K_p	Decrease	Increase	Small Change	Decrease
K_i	Decrease	Increase	Increase	Eliminate
K_d	Small Change	Decrease	Decrease	Small Change

A proportional controller can reduce the rise time and steady-state error but cannot remove it. An integral controller eliminates the steady-state error, but makes the transient response worse. A derivative controller improves the stability of the system, reduces the overshoot, and improves the transient response

2.4.7 Steps for designing a PID controller

While designing a PID controller for a given system, the following steps are followed in order to obtain a desired response.

- The open-loop response is obtained.
- K_p is added to improve the rise time.
- K_d is added to improve the overshoot.
- K_i is added to eliminate the steady-state error.
- Each of K_p , K_i , and K_d is adjusted until a desired overall response is obtained.

2.4.8 PID palette in LabVIEW

The figure 2.5 shows the functions of the PID Palette. Following functions are found the Control Palette:

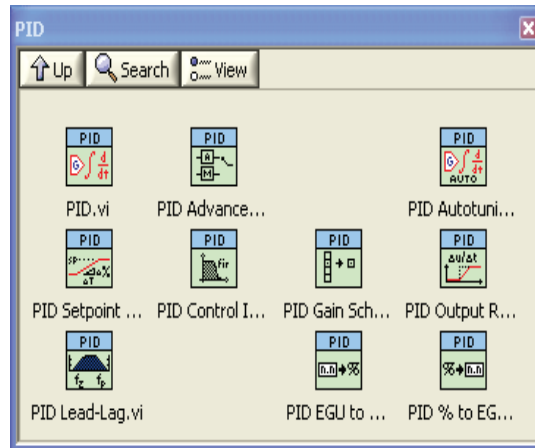


Figure 2.5: PID Palette

Chapter 3

Implementation of DataSocket Protocol

VI LabVIEW

DataSocket Overview

Using DataSocket in LabVIEW

Internet Comm. Using DataSocket

Results and Discussion

3 IMPLEMENTATION OF DATASOCKET PROTOCOL

LabVIEW is different from other software packages as it uses symbols rather than any textual language for programming purpose. LabVIEW programs are called Virtual Instruments (VIs) because replaces original instruments with the blocks which carry the same operation as that of the instrument. Any instrumentation application will have two components, namely, the user interface and the underlying code.

3.1 VI in LabVIEW

The VI consists of two divisions as follows:

- Front Panel
- Block Diagram

3.1.1 Front panel

The front panel of LabVIEW is known as “panel” in short. It is a LabVIEW term for user interaction interface. The panel contains two objects namely controls and indicators. The control is a LabVIEW front panel entity which takes an input to the code while the indicator is an entity which displays the output from the code. Figure 3.1 shows the front panel of a VI.

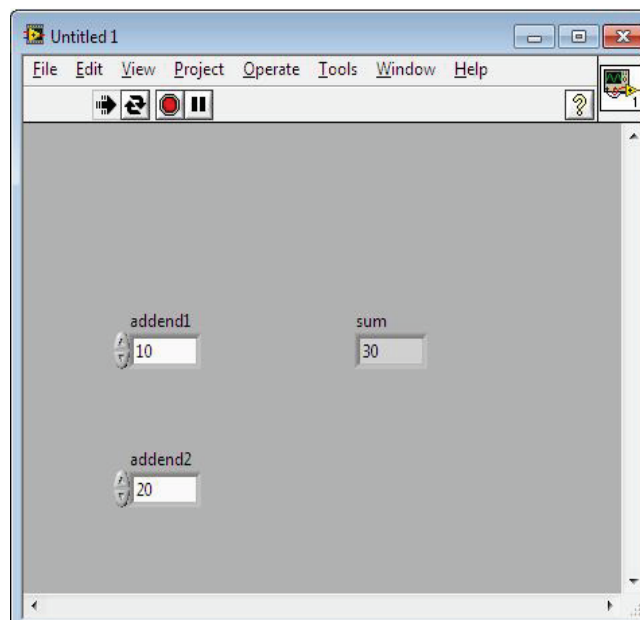


Figure 3.1: Front Panel of LabVIEW

In short the front panel of LabVIEW has got the following features,

- Provides user interface.
- Shows Controls or Indicators.
- Highly customizable.

3.1.2 Block diagram

The block diagram of LabVIEW is otherwise known as “diagram” in short. It is the “backplane” of the LabVIEW program where the actual code resides. Any control or indicator placed on the front creates an icon on the block diagram which acts as a connection between them. The main programming part of VI is done here. The controls can be differentiated from the indicators by their thicker border and contains an arrow on the right side of the block.

Even the software gives facility to make sub VI(s) by which the efficiency and optimization increases. National Instruments provides various toolkits by which a user can understand the software easily. Figure 3.2 shows the block diagram of a VI where the sum of two numbers is found out.

In short the block diagram of LabVIEW has got the following features,

- It consists of the actual program.
- This part is invisible to user.
- User reads from left to right just like a book.

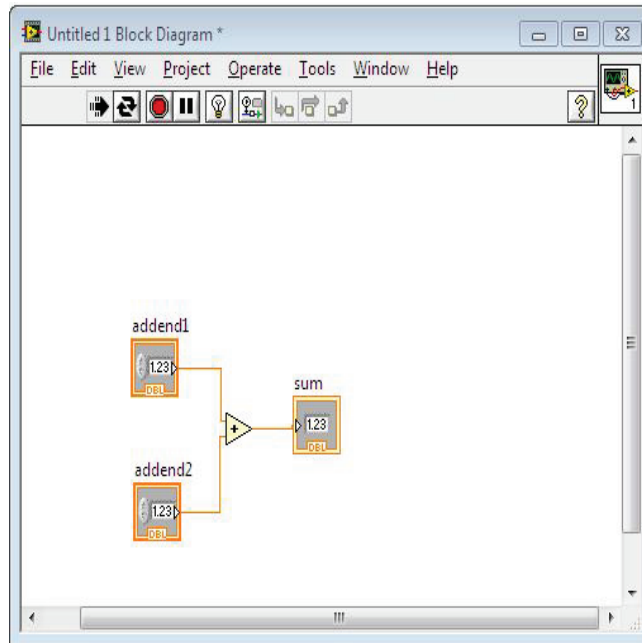


Figure 3.2: Block Diagram of LabVIEW

3.2 DataSocket Overview

DataSocket is a TCP/IP tool developed by National Instruments for networked data transfers. It is an online programming technology based on TCP/IP that makes an easier way of data exchange between computers and applications. In earlier days, various protocols, hardware and software were developed for the applications. Usually in testing and manufacturing industries problems arise due to the integration of hardware and software component of a system. This problem forced a programmer to link between the protocols to transfer data, which required lots of development, time and resources. As a solution to this, National Instruments developed DataSocket. It helped in publishing/sharing live data over a network and provides an interface for many protocols.

3.3 Using DataSocket in LabVIEW

The tool that is devised by NI for networked data acquisition is Data Sockets. With development in LabVIEW for networking and distributed control, Data Sockets helped in the recent versions of LabVIEW. However, it does not form a part of the standard networking tools. So in order to use Data Socket in LabVIEW the user has to start the DataSocket Server in the data socket network. Figure 3.3 shows the DataSocket server.

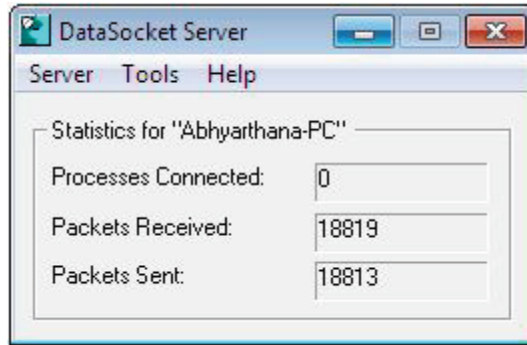


Figure 3.3: DataSocket Server

DataSocket is compatible with any form of data type. The DataSocket sub-palette is the Data Communication menu for a user. There are four functions found in this palette and they are:

- DataSocket Open
- DataSocket Close
- DataSocket Read
- DataSocket Write

The natural sequence of the program will be Open, Close, in between a series of read and write operations. The variant sub menu is used to handle data transfers with variants in a bidirectional manner. Variant is a special type of data type that is created for Data Socket operations. It has got close relation with String data type.

3.3.1 DataSocket open

This block can be used only once in each VI. Figure 3.4 shows the block for DataSocket Open.

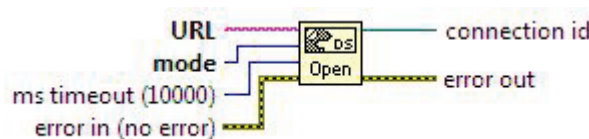


Figure 3.4: DataSocket Open

The inputs and outputs of the block can be seen in the diagram. The URL is the site for the connection, for host system `dstp://localhost` can be used. The mode is an integer which defines the mode of connection, '1' is for write, '2' is for read, '3' is for buffered read and '4' is for buffered read/write. The output of the blocks is connection ID that is generated.

3.3.2 DataSocket close

This block can be used only once in each VI. Figure 3.5 shows the block for DataSocket Close.

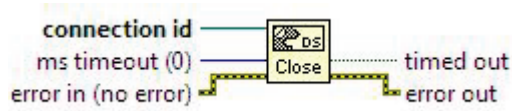


Figure 3.5: DataSocket Close

The connection ID generated from DataSocket Open is given as an input to this block. The output timed out is optional.

3.3.3 DataSocket write

Figure 3.6 shows the block for DataSocket Write.

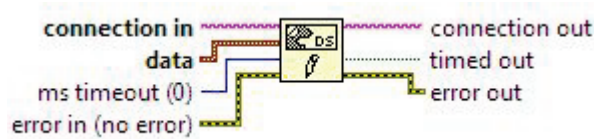


Figure 3.6: DataSocket Write

The connection for this block can be defined either through a string or the connection id. This block writes the data to the variable that is present on the other side connected in.

3.3.4 DataSocket read

Figure 3.7 shows the block for DataSocket Read.

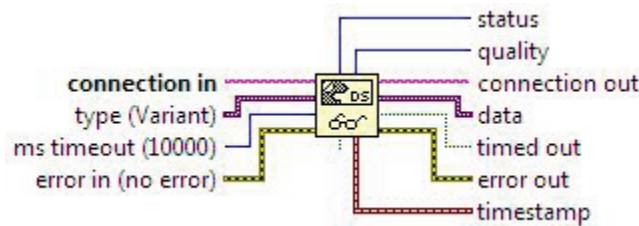


Figure 3.7: DataSocket Read

The output of this block is data where the value of variable is assigned. The input to this block is connection in where the data exchange process is carried out.

3.4 Internet Communication Using DataSocket Protocol

Data Socket (DS) is a communication protocol in LabVIEW. It permits reading, writing, and sharing of live data between applications and/or different data sources and destinations. DS uses an URL format similar to http, which is called dstp (Data Socket Transfer Protocol). Dstp allows two way communications and data exchange between applications. This protocol is used in both server and client applications. LabVIEW has several built-in Sub VIs that can be used within systems have permission to write, create, and read data on the DS Server.

3.4.1 Server DataSocket program implementation

The Data Socket protocol based program in the server computer opens bi-directional communication using DS Open Connection sub VI. The server with URL given as `dstp://localhost/file_name` receives set point value from the client. The server sends the process temperature value to the client. The Low-Limit and the High-Limit of the temperature control neutral zone are also set from the server front panel. The client controls the set-point on the server.

3.4.2 Client Data Socket program implementation

The client computer also opens a communication using DS Open sub VI. The server with URL as `dstp://ip_address of server/file_name` is selected which allows connection between clients and the server using DS protocol to interchange the information. The client sends set-point information to the server as a double precision value. The server sends the process temperature value to the client computer. The Data Socket connection is closed using the DS Close Connection sub VI.

3.4.3 Algorithm for implementation of ON/OFF controller

The logic behind the implementation of ON/OFF controller using DataSocket Protocol is described as follows:

1. The calculated error is compared with upper limit and lower limit.
2. Accordingly, the indicators show the state and action that has been taken. The state is seen through the “Within Limit” and “Out of Limit” indicator. The “Fan ON” and “Fan OFF”

indicators show the control action that is to be taken.

3. The comparison has been done with following three conditions.

A). If $\text{Error} > 0$ AND $\text{PV} > \text{Low Limit}$ then the indicators states are:

Cooling Fan ON: OFF

Cooling Fan OFF: ON

Within Limit: ON

Out of Limit: OFF

Digital I/O: OFF

If condition is not satisfied, then the indicators states are:

Cooling Fan ON: ON

Cooling Fan OFF: OFF

Within Limit: OFF

Out of Limit: ON

Digital I/O: ON

B). If $\text{Error} < 0$ AND $\text{Tout} < \text{High Limit}$, then the indicators states are:

Cooling Fan ON: ON

Cooling Fan OFF: OFF

Within Limit: ON

Out of Limit: OFF

Digital I/O: ON

If condition is not satisfied, then the indicators states are:

Cooling Fan ON: OFF

Cooling Fan OFF: ON

Within Limit: OFF

Out of Limit: ON

Digital I/O: OFF

C). If $\text{Error} = 0$ then the indicators states are:

Cooling Fan ON: OFF

Cooling Fan OFF: OFF

Within Limit: ON

Out of Limit: OFF

Digital I/O: OFF

If condition is not satisfied, then the indicators states are,

Cooling Fan ON: OFF

Cooling Fan OFF: ON

Within Limit: OFF

Out of Limit: ON

Digital I/O: OFF

3.4.4 Algorithm for manual tuning of PID controller

Using the following steps, manually the PID tuning parameters has been set and the output response is obtained.

1. First T_i and T_d values has been set to zero.
2. K_C has been increased until the output of the loop oscillates.
3. K_C has been set to approximately half of that value for a "quarter amplitude decay" type response.
4. T_i has been increased until any offset is corrected for the process.
5. K_C has been increased if required but too much K_C causes excessive response and overshoot.

3.5 Results and Discussion

The various simulation results of implementation of ON/OFF and PID controller using Data Socket protocol are discussed in this section.

3.5.1 Implementation of ON/OFF controller using Data Socket Protocol

Figure 3.8 shows the front panel and block diagram of a client server communication using Data Socket Protocol. The client1 VI acts as server and client2 VI acts as client. The server is run first followed by client so that the client responds to the request of the server. The URL given by server for sending Process Variable (PV) is `dstp://localhost/client1` and for receiving Set Point (SP) is `dstp://localhost/client2`. The URL given by client for sending SP is

dstp://localhost/client1 and for receiving PV is dstp://localhost/client2.

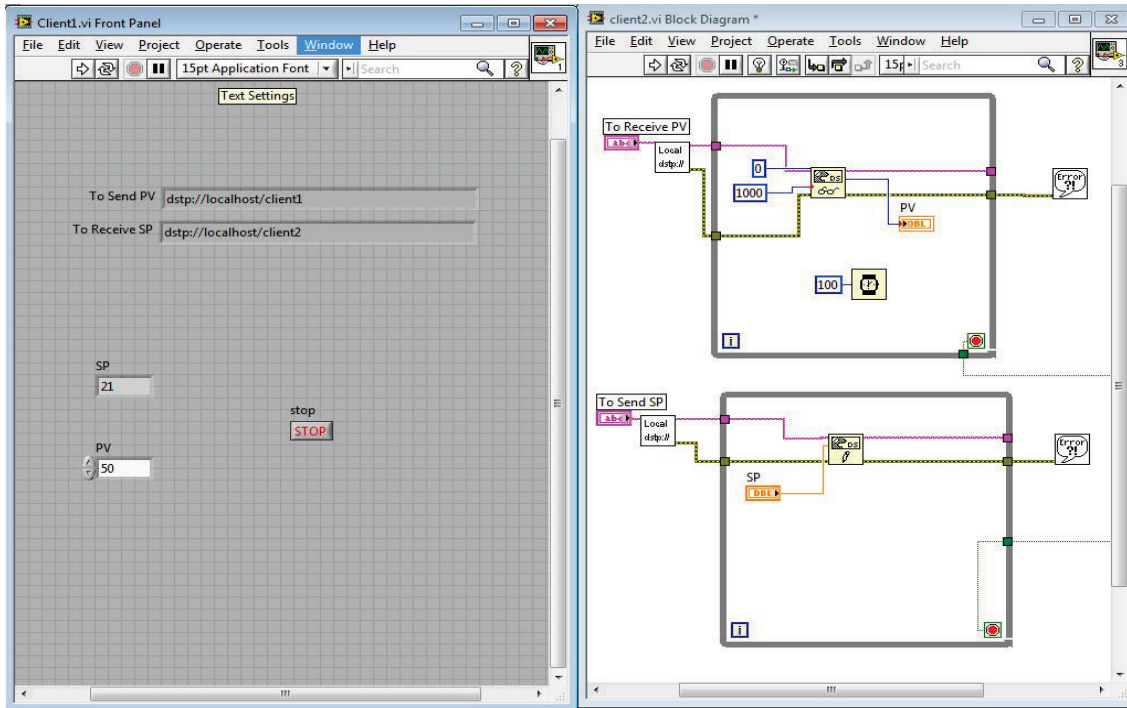


Figure 3.8: Simple DataSocket Protocol Implementation

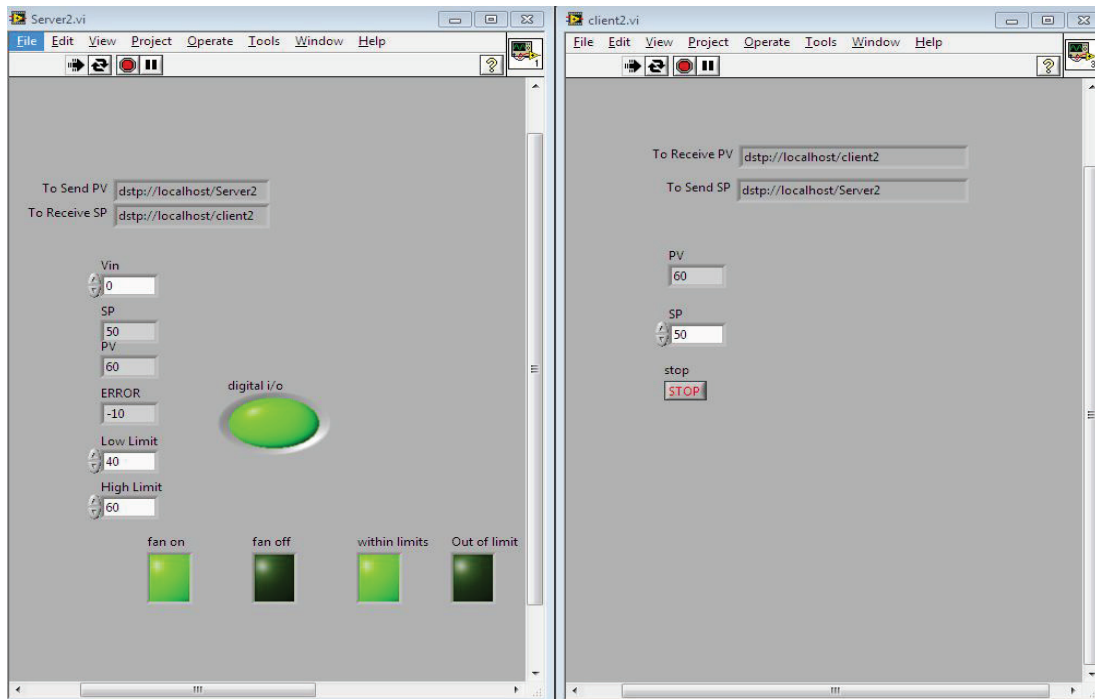


Figure 3.9: Output when $SP < PV$ and PV is Within Limit

Figure 3.9 shows the front panel of server on left side and client on right side. The lower and

upper limit for temperature has been chosen as 40 and 60 respectively by the server PC. The V_{in} is 0, corresponding PV obtained is 60. The client sets the SP as 50 and the value of error has been calculated as -10. The output is obtained for the condition $SP < PV$ and PV is Within Limit. The state of the indicators can be seen in the figure. Even if the PV is within limit still it is deviated from the set point. So the Cooling Fan is turned ON in order to bring back to its desired value.

Figure 3.10 shows the front panel of server on left side and client on right side. The lower and upper limit for temperature has been chosen as 40 and 60 respectively by the server PC. The V_{in} is 5, corresponding PV obtained is 90. The client sets the SP as 50 and the value of error has been calculated as -40. The output is obtained for the condition $SP < PV$ and PV is Out of Limit. The state of the indicators can be seen in the figure. Since the value of PV is not in the range the “Out of Limit” is turned ON while others remain OFF.

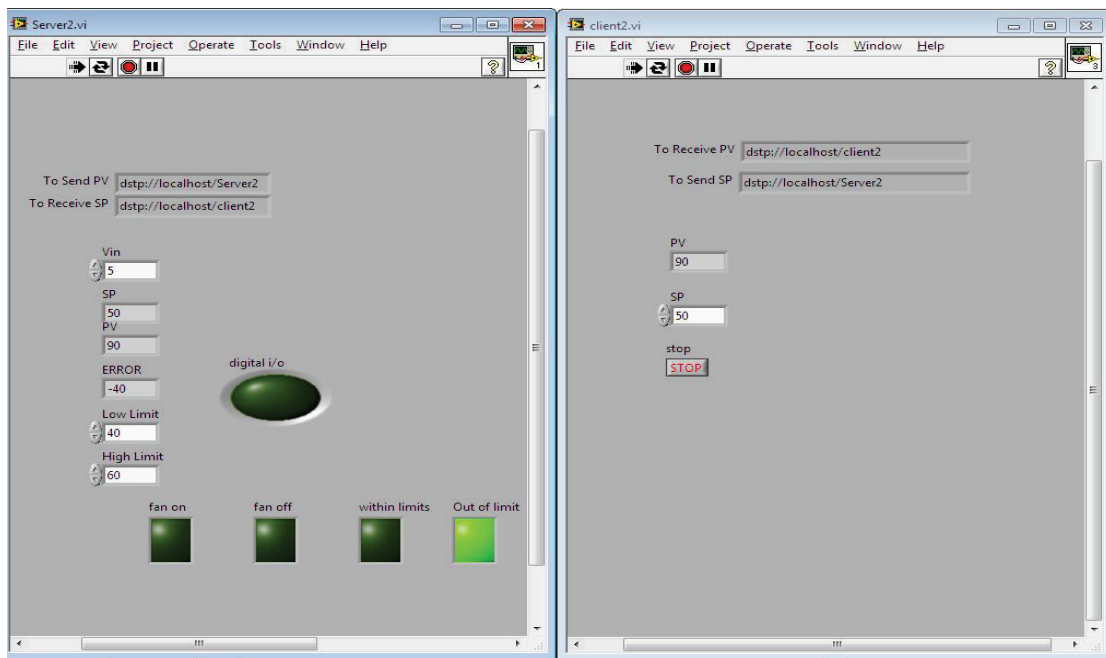


Figure 3.10: Output when $SP < PV$ and PV is Out of Limit

Figure 3.11 shows the front panel of server client for the condition when $SP > PV$ and PV is within the limit. The V_{in} is -2, corresponding PV obtained is 48. The client sets the SP as 50 and the value of error has been calculated as 2. The output is obtained in the form of indicators that can be seen in the figure. Since the value of PV is within limit but has not crossed SP, the cooling fan remains OFF.

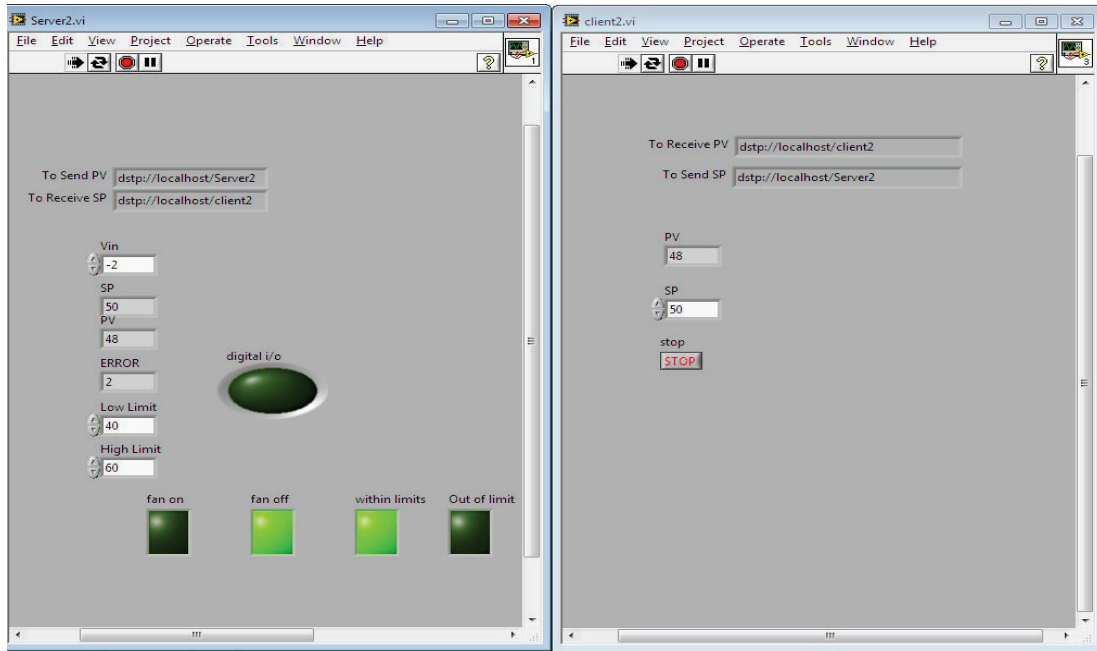


Figure 3.11: Output when $SP > PV$ and PV is Within Limits

Figure 3.12 shows the front panel of server client for the condition when $SP > PV$ and PV is out of limit. The V_{in} is -5, corresponding PV obtained is 30. The client sets the SP as 50 and the value of error has been calculated as 20. The output is obtained in the form of indicators that can be seen in the figure. Since the value of PV is below the lower limit the “Out of limit” indicator turns ON while the other remain OFF.

Figure 3.13 shows the front panel of server client for the condition when $SP = PV$. The SP set by client is 60, V_{in} is 0, corresponding PV obtained is 60. Since the client has set the SP to 60, the value of error calculated is 0. The output is obtained in the form of indicators that can be seen in the figure. Since there is no deviation of PV from the SP , no control actions are taken. Only the “Within limit” indicator turns ON since the PV does not lie outside the dead band set by the server. Other indicators remain OFF.

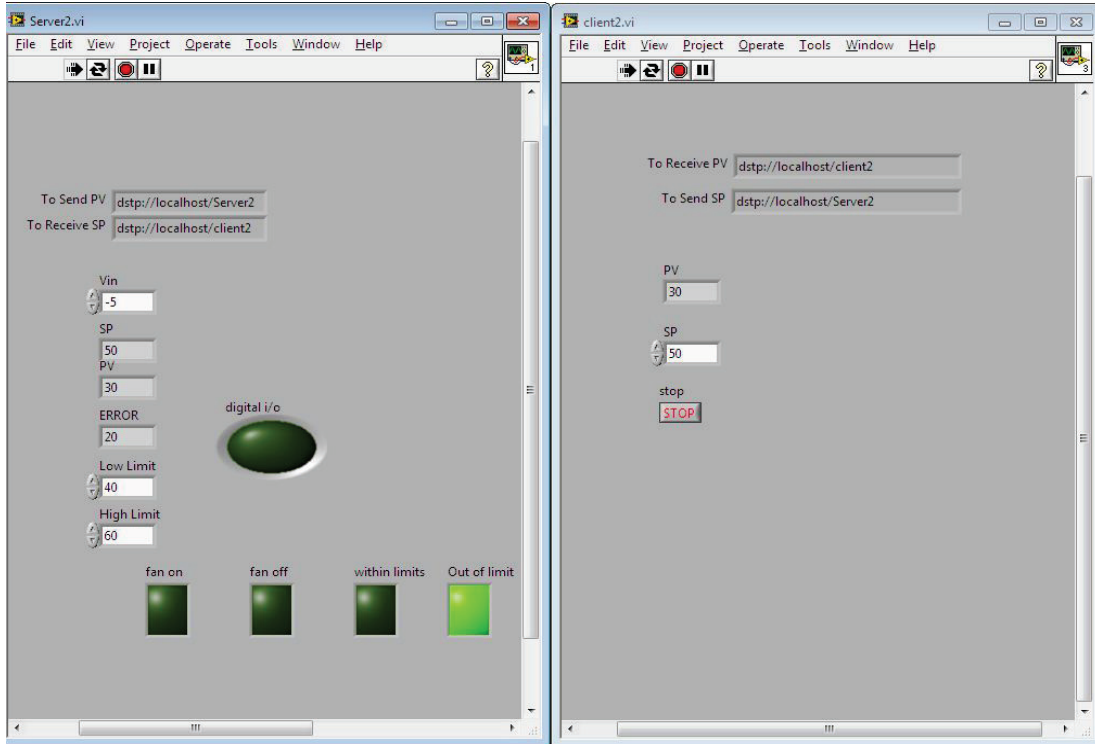


Figure 3.12: Output when $SP > PV$ and PV is Out of Limits

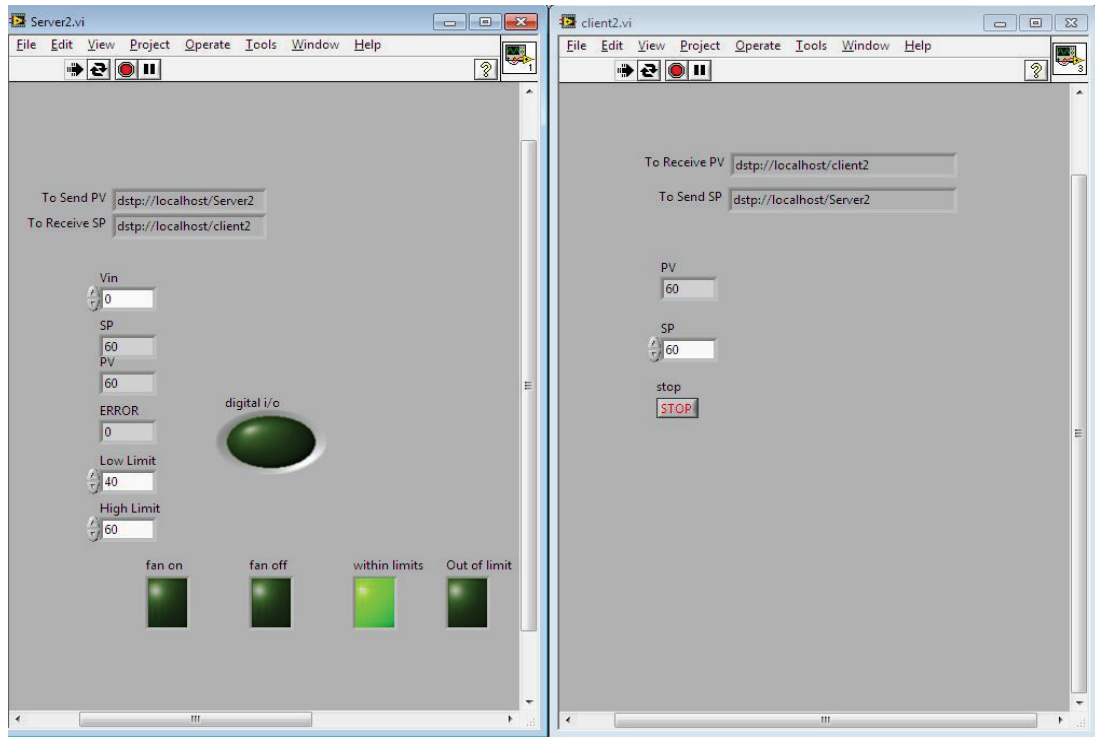


Figure 3.13: Output when $SP = PV$

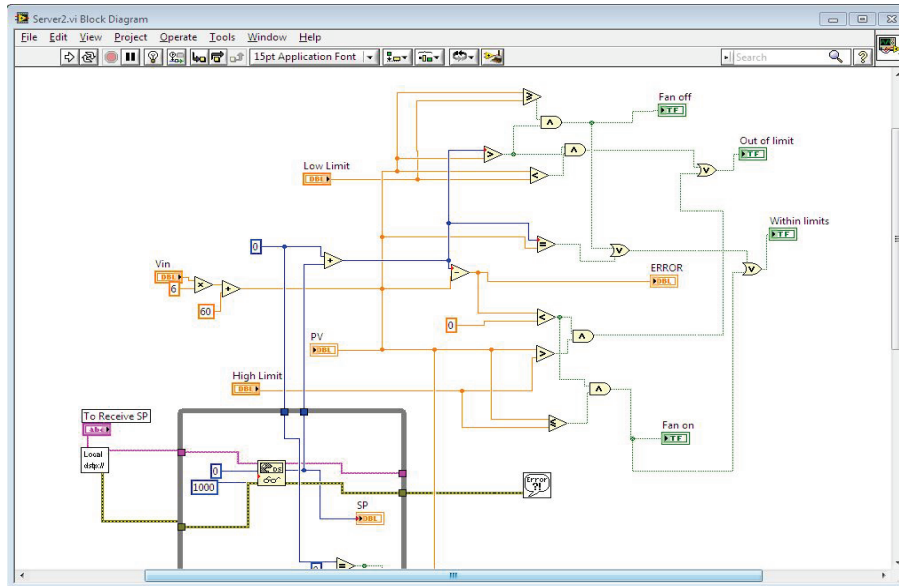


Figure 3.14: Upper portion of the block diagram of server

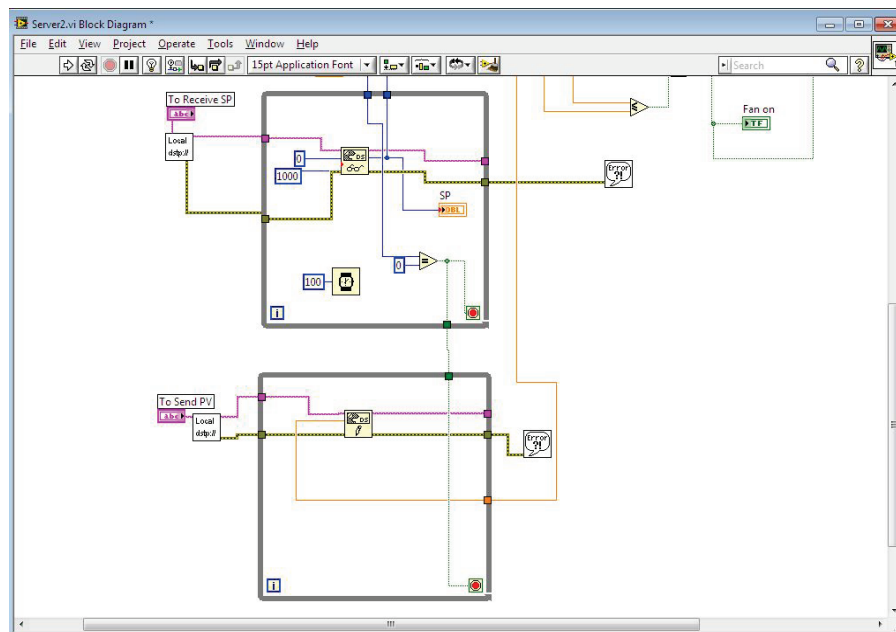


Figure 3.15: Lower portion of the block diagram of server

Figure 3.14 shows the upper portion of the block diagram of server while figure 3.15 shows the lower portion. The combination of both figures forms the complete block diagram of server. The upper portion shows the operation of ON/OFF controller and lower portion shows the connections for Data Socket protocol communication with client.

3.5.2 Implementation of PID controller using DataSocket protocol

Figure 3.16 shows the front panel of PID controller using DS protocol. The URL for sending PV to client is `dstp://localhost/PID_DS_SER` and that of receiving SP is `dstp://localhost/client2`. The SP set by the client is 60. In order to obtain response curve, the tuning parameters are set to different values. In the given figure the value of PV is controlled up to 60.02. This is a close approximation obtained.

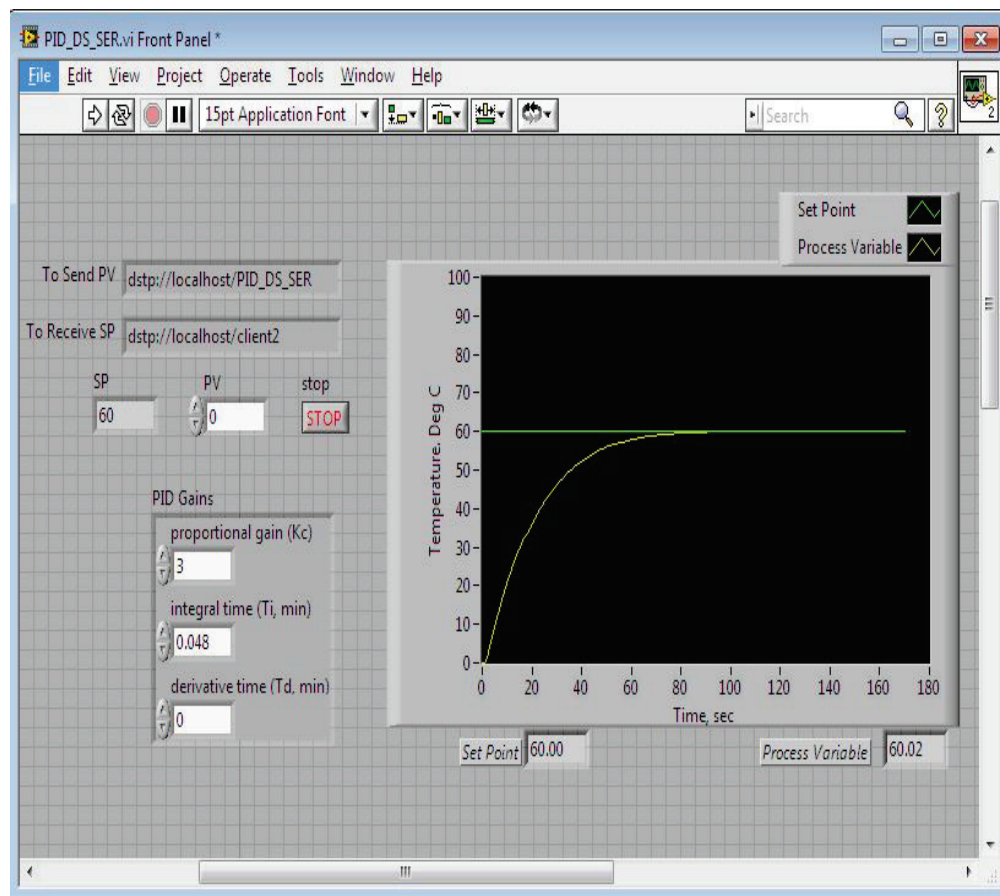


Figure 3.16: Front Panel of PID controller

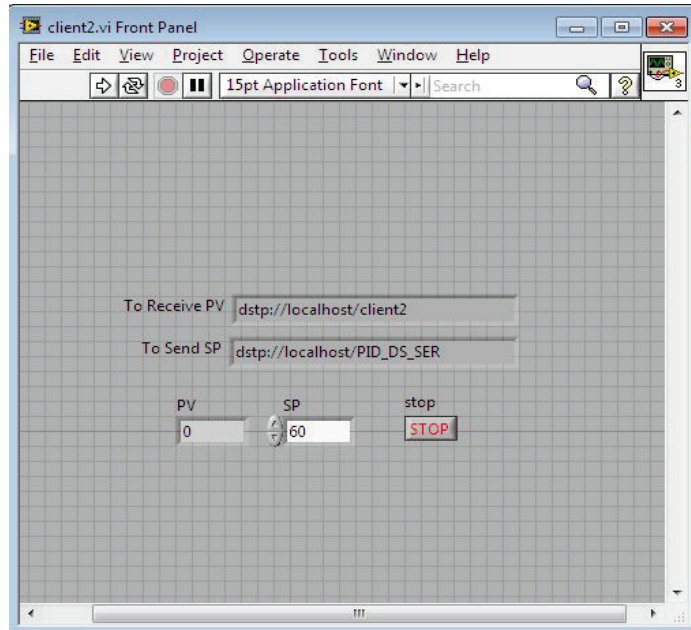


Figure 3.17: Front panel of client (PID controller)

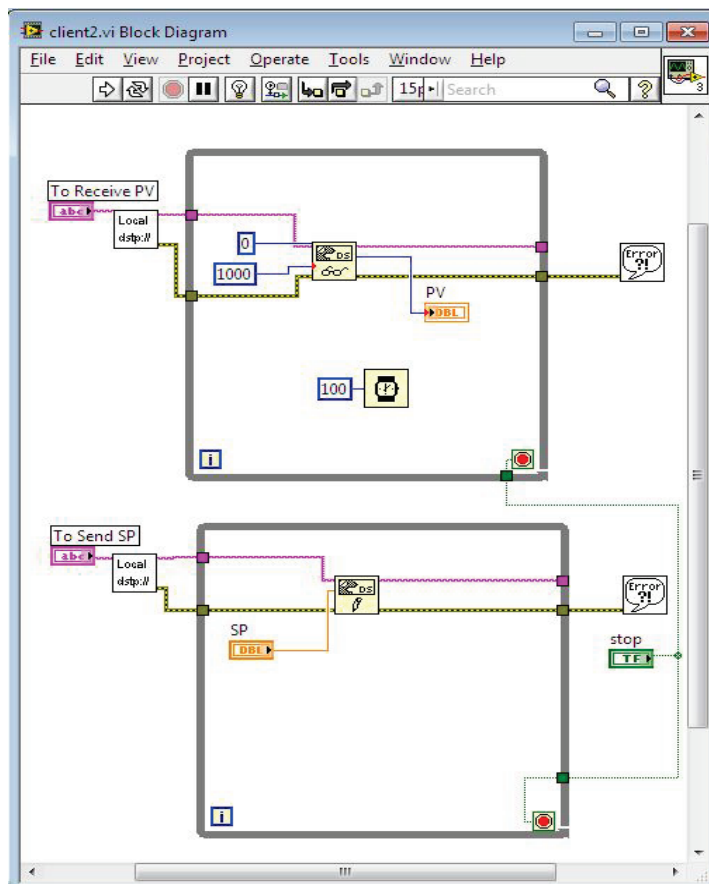


Figure 3.18: Block diagram of client (PID controller)

Figure 3.17 shows the front panel of the client and figure 3.18 shows its block diagram. The communication of client and server is similar to that of the ON/OFF controller. The only difference lies in the operation of controller.

There are three conditions of response and they are, critically damped, over damped and under damped. These responses have been obtained using manual tuning of PID tuning parameters. When K_C , T_i and T_d have been tuned to 3, 0.006 and 0 respectively, underdamped response have been obtained. The graph is shown in Figure 3.19. The value of process variable obtained was 60.05. An overshoot peak is observed and immediately follows undershoot which tries to stabilize the curve.

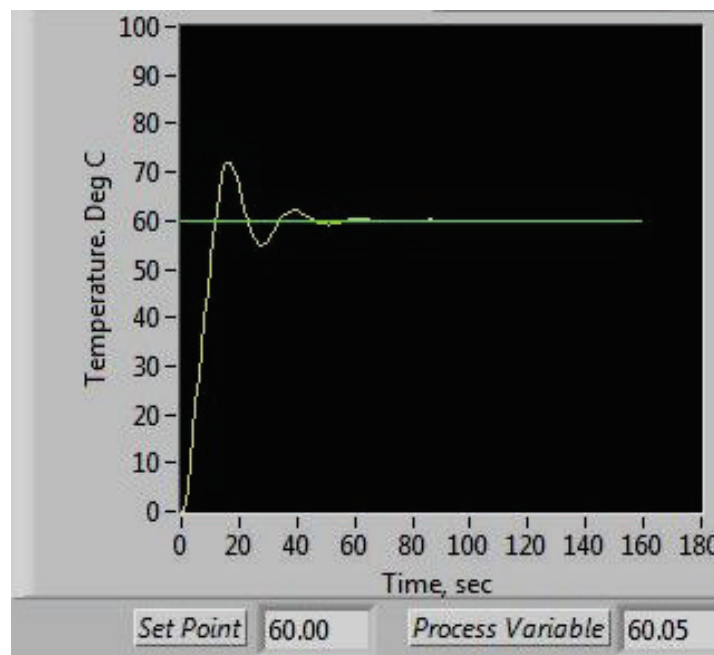


Figure 3.19: Graph showing under damped response

Figure 3.20 shows the critically damped response of PID controller. This is the manual tuning results where the values of tuning parameters of PID controller, K_C , T_i and T_d have been tuned to 3, 0.048 and 0 respectively. It is clearly seen that the PV is controlled to its SP value with less time. The value of process variable obtained was 60.

When K_C , T_i and T_d have been tuned to 3, 0.08 and 0 respectively, overdamped response have been obtained. The graph is shown in Figure 3.21. The value of process variable obtained was 60.06. For individual response, their corresponding rise time, settling time, peak overshoot

have been obtained.

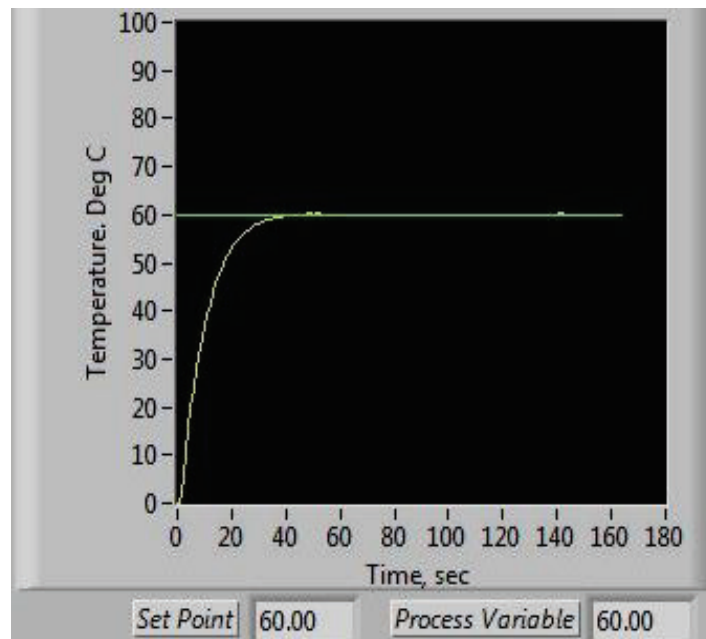


Figure 3.20: Graph showing critically damped response

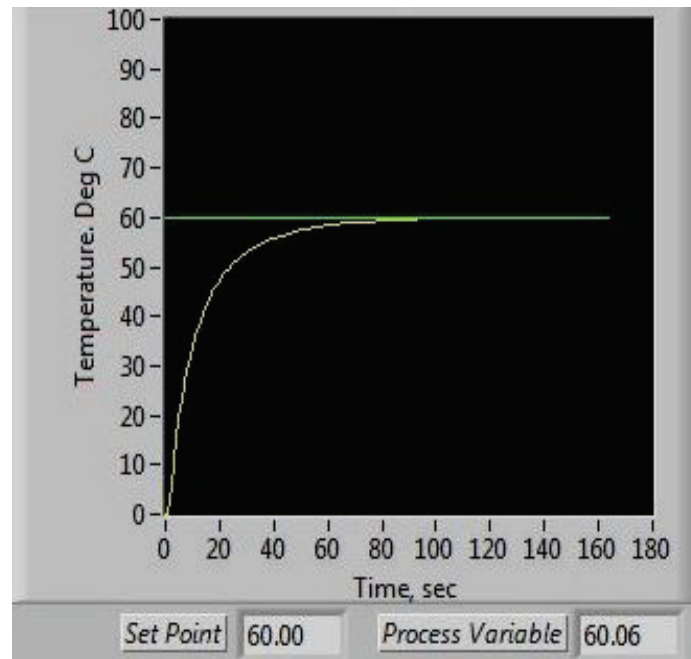


Figure 3.21: Graph showing over damped response

Table 3.1 shows the comparison between the three response curves with respect to their curve

characteristics. So, the final values for PID parameters were chosen as 3, 0.048 and 0 respectively.

Table 3.1: Comparison between response curves

Response Curve	Output at different conditions		
	Rising time	Settling time	Maximum Overshoot
Critically damped	21	40	0
Over-damped	35	72	0
Under-damped	10	56	12

3.5.3 Implementation of multiple client-server communication

Multiple communications refers to a relation where more than one connection is present. In this case, two client users are present and they communicate with a single server. Figure 3.22 shows the flow of data among the three users acting as clients and server. The server at first initiates the request for data.

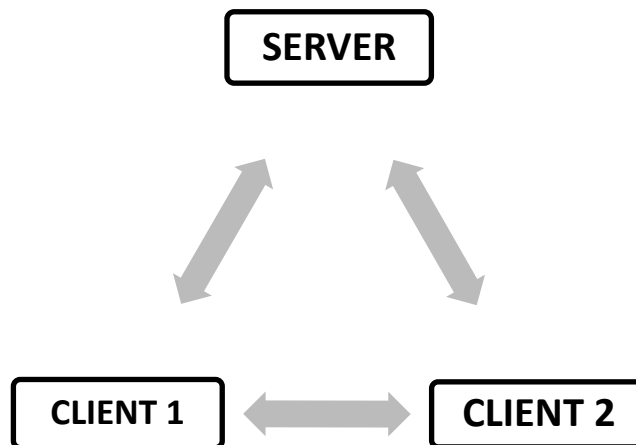


Figure 3.22: Flow of data among 2 clients and a server

Figure 3.23 shows the front panel of server. The server sends the value of PV to client1 and waits for value of SP from client 2.

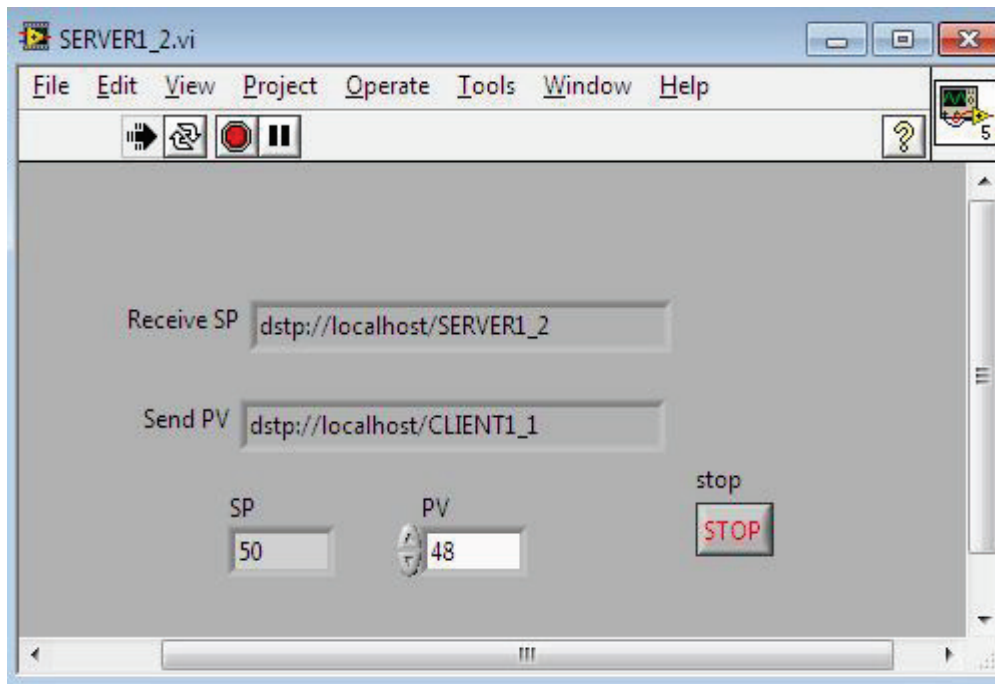


Figure 3.23: Front Panel of Server

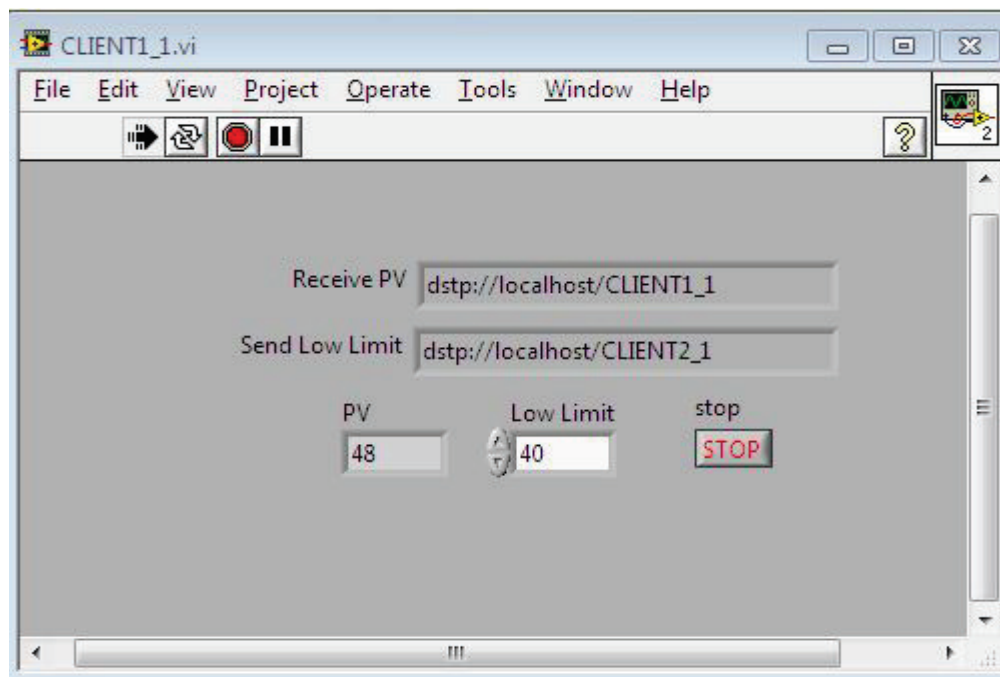


Figure 3.24: Front Panel of Client1

Figure 3.24 shows the front panel of client1. The client receives the value of PV from server and then sends the value low limit to client 2. The function of client 1 is to decide the value for lower limit of temperature band.

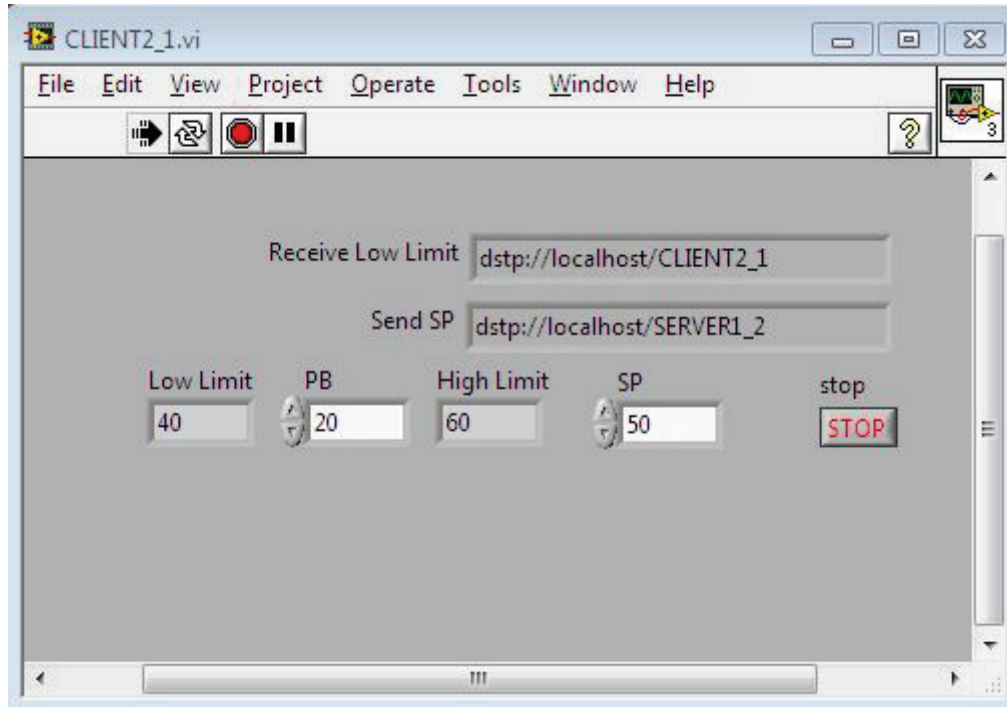


Figure 3.25: Front Panel of Client2

Figure 3.25 shows the front panel of client2. This client receives the value of lower limit from client 1 and then sends the value set point to server. The function of client 2 is to decide the value for upper limit of temperature band as well as the set point. The value of upper limit is obtained by the following formula,

$$\text{Upper limit} = \text{Lower limit} + \text{Dead band} \quad (3.1)$$

On obtaining upper limit, the client has got both the limits and can decide the set point.

Chapter 4

Implementation of TCP/IP

*Internet Protocol
User Datagram Protocol
Transmission Control Protocol
Online Written Communication
Results and Discussion*

4 IMPLEMENTATION OF TCP/IP

The two controllers are already described briefly in chapter 3. The algorithm and the basic operations remain same as discussed earlier. This chapter deals with implementation of these two controllers using TCP/IP.

4.1 Internet Protocol

The various tools for communication over networks are Internet Protocol (IP), User Datagram Protocol (UDP), and Transmission Control Protocol (TCP). A datagram contains of data and a header containing source and destination addresses. IP develops the path for the datagram to transmit over the network and reach to the specified destination. IP is not reliable and provides duplicity in transmission of data. A programmer usually prefers TCP or UDP over IP.

4.2 User Datagram Protocol

UDP provides simple, low-level communication between processes on computers. Processes communicate by sending datagrams to a destination computer or port. A port is the location where data has to be sent. IP handles the computer-to-computer delivery. Once the datagram reaches the destination computer, UDP moves the datagram to its destination port. If the destination port is not open, UDP discards the datagram. UDP shares all the delivery problems of IP. UDP is used in applications where reliability is not critical.

4.3 Transmission Control Protocol

TCP/IP is an Internet Protocol (IP). The TCP part of the name refers to Transmission Control Protocol. TCP/IP provides a reliable way of transmitting and receiving data over a network.

4.3.1 Using TCP connections in LabVIEW

In order to use TCP/IP the following steps have been followed in LabVIEW.

- A TCP/IP Listen has been established. This is found on the Functions Communication TCP palette. Figure 4.1 is the icon for the TCP/IP Listen.



Figure 4.1: TCP/IP Listen Icon

- The TCP/IP Listen listens over the network for another computer trying to connect over the port specified in the integer input at the left side of the icon. That port number must be a number not used by other parts of the system, and here port number 2055 has been used arbitrarily.
 - When the TCP/IP Listen hears another computer trying to connect over the correct port, it establishes a Connection ID, and an Error Signal, and those outputs are used in later TCP/IP vi blocks.
 - Another TCP/IP Listen cannot be used with the same port number once the connection has been established.
- User can use either of the following vi(s).
 - TCP/IP Write - which writes data to the other computer over the network. Figure 4.2 is the icon.

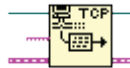


Figure 4.2: TCP/IP Write Icon

- ❖ The TCP/IP Write uses the Connection ID, which it also passes to the next vi, and the Error signal which it also passes to the next vi.
 - ❖ The input is a string so it is necessary to convert it to a string.
 - ❖ User can specify a timeout, and can get information on the number of bytes written over the network with this vi.
- TCP/IP Read - which reads bytes from the remote computer. Figure 4.3 is the icon.
 - ❖ There are the usual inputs, the Connection ID and the Error Signal.

-
- ❖ The integer input (blue line at left) is the number of bytes that the user wants to read.
 - ❖ The output is the string composed of the bytes that are read.

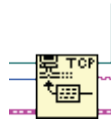


Figure 4.3: TCP/IP Read Icon

4.3.2 Algorithm for using TCP/IP:

By convention, the “client” is the user that initiates the call and requests for information, and the “server” is the user that answers the call and dispenses information. But this is a general not a rule, the connection can work both ways.

Next target is to obtain the server’s IP number for the client. If it is a local area network (LAN), the address is fixed as (192.168.xxx.xxx) and a constant is stored somewhere.

Following are the Vis found in a TCP palette:

- TCP Open Connection
- TCP Close Connection
- TCP Write
- TCP Read
- TCP Create Listener
- TCP Wait on Listener

For the client side,

- In Client PC, TCP Open has been called with the IP Number, the port number that has been connected with a suitable timeout.
- For TCP Write, a string and Connection ID has been provided that is obtained from TCP Open.
- The SP that is in integer format has been converted to string by using typecast. TCP Read has been called and is provided with the connection id (obtained from TCP Open), a mode, and a number of bytes to read.

-
- When transmission is over, TCP Close has been called to close the connections.

For the server side,

- The TCP Create listener has been called and this provides a listener id. In a while loop; TCP listener has been called using the listener id that is just obtained.
- When a connection comes in, the listener becomes active, then the TCP Read and TCP Write has been called.
- When transmission is over, TCP Close has been called, and the loop is repeated, to wait for another connection to come in.

4.3.3 Advantages of TCP/IP

- Variable length messages allows clients to negotiate the transfer of information
- Data that is transferred is buffered by receiver
- Servers can be built where desired

4.3.4 Disadvantage of TCP/IP

- At times it can be complicated to setup

TCP provides the most reliable data transmission and allows multiple and simultaneous communications.

4.4 Online Written Communication

Chatting has been always done via internet but not in any software. LabVIEW offers the facility to chat and communicate in software. Here the internet protocol helps in communication that is TCP). The main objective is to show the various advantages of internet protocols in LabVIEW. The advantages include communication like chatting, sending information, etc. Client and Server are two users that communicate using TCP/IP for sending their data.

4.4.1 Algorithm for written communication

The following steps have been taken to send a single string.

-
- At first, the length of the string has been found out.
 - The length of the string is a number, and that number is converted to a string so that it can be sent further.
 - This length is sent to the user using a TCP/IP Write.
 - Then, the string is sent.
 - When the transmission is over, connection has been closed.

The following steps have been taken to receive a single string.

- The first string received has the data for the string length of the string needed. The length of the first string received has been assumed to be 2.
- The length of the string is a string; it is converted to a integer and is used in the second TCP/IP Read.
- When transmission is over, connection has been closed using TCP/IP Close Connection.

4.5 Results and Discussion

The various simulation results of implementation of ON/OFF and PID controller using TCP/IP are discussed in this section.

4.5.1 Implementation of ON/OFF controller using TCP/IP

The front panel of the ON/OFF controller server is shown in Figure 4.4 and the block diagram is shown in Figure 4.5. In this case, the value of PV is greater than SP, the calculated error is positive and hence the “fan OFF” indicator is turned ON. Since PV is less than the high limit, the “within limits” indicator is turned ON while the rest two indicators remain OFF.

The TCP/IP icons can be seen in the block diagram. The port address is 2055 which is given as input to the TCP OPEN CONNECTION in Client side and as input to TCP LISTEN icon in Server side.

In this case, the value of PV is greater than SP, the calculated error is positive and hence the “fan OFF” indicator is turned ON. Since PV is less than the high limit, the “within limits” indicator is turned ON while the rest two indicators remain OFF. The TCP/IP icons can be seen in the block diagram. The port address is 2055 which is given as input to the TCP OPEN

CONNECTION in Client side and as input to TCP LISTEN icon in Server side.

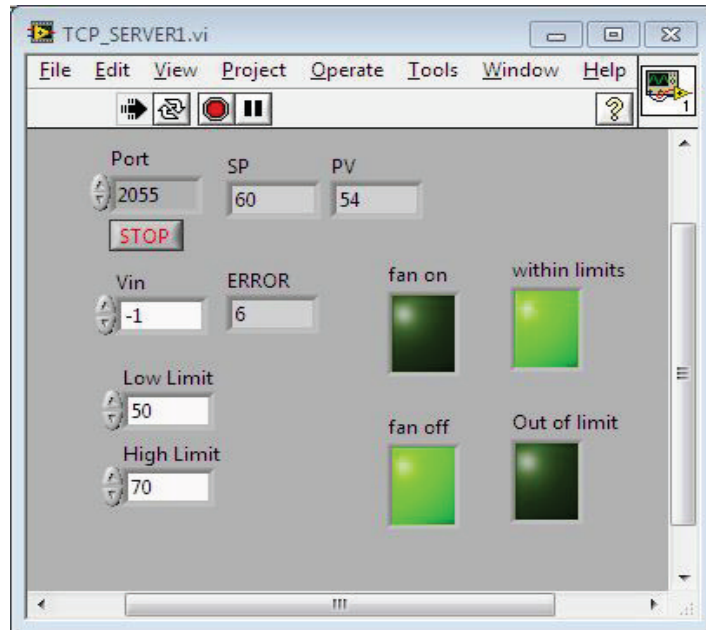


Figure 4.4: Front Panel of ON/OFF controller Server

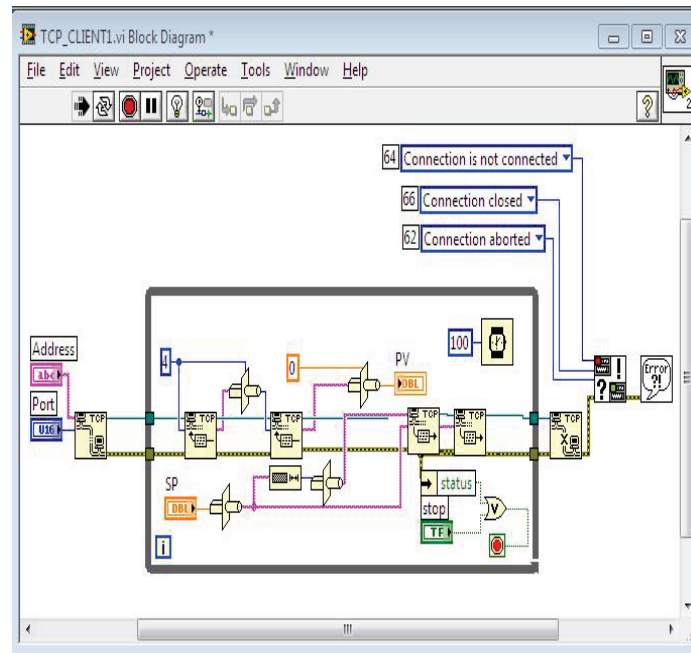


Figure 4.5: Block Diagram of ON/OFF controller Client

4.5.2 Implementation of PID controller using TCP/IP

The front panel of the PID controller server using TCP protocol is shown in Figure 4.6 where the

set point is set to 60. The front panel of PID controller client is shown in Figure 4.7. In this case, also the PV value is sent to the client and the client sets the SP. This is the manual tuning results where the values of tuning parameters of PID controller, K_C , T_i and T_d has been tuned to 3, 0.048 and 0 respectively. It is clearly seen that the PV is controlled to its SP value with less time.

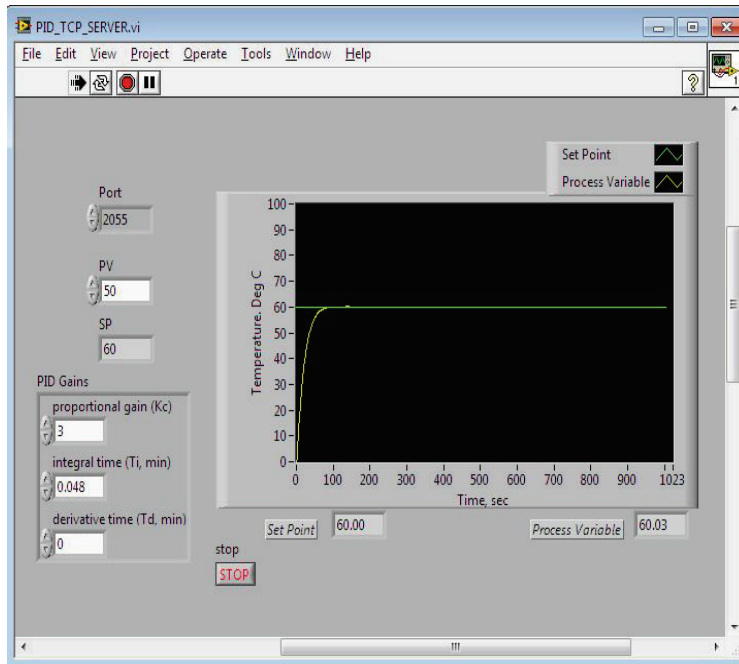


Figure 4.6: Front Panel of PID controller Server

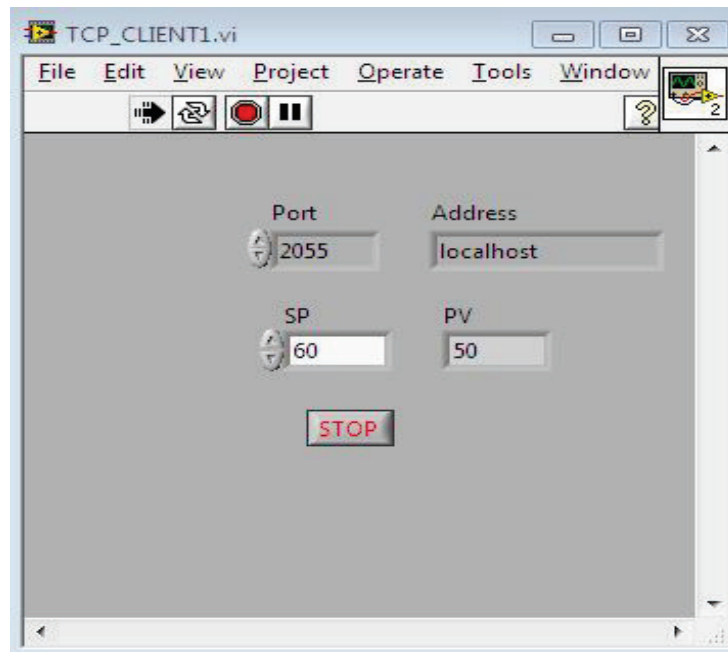


Figure 4.7: Front Panel of PID controller Client

4.5.3 Implementation of online written communication

The front panel of communicating server is shown in Figure 4.8. The server on finding the value of PV sends it to client by writing a message as “PV=50” in “Server Typing” box found in the front panel. On pressing enter while running condition, this message can be seen in “Server’s Message” which shows the message server typed. This message is sent to client with port number 6342 and can be seen in “Server’s Reply” box. The next message sent by server is “SP=?” and on pressing enter reaches client.

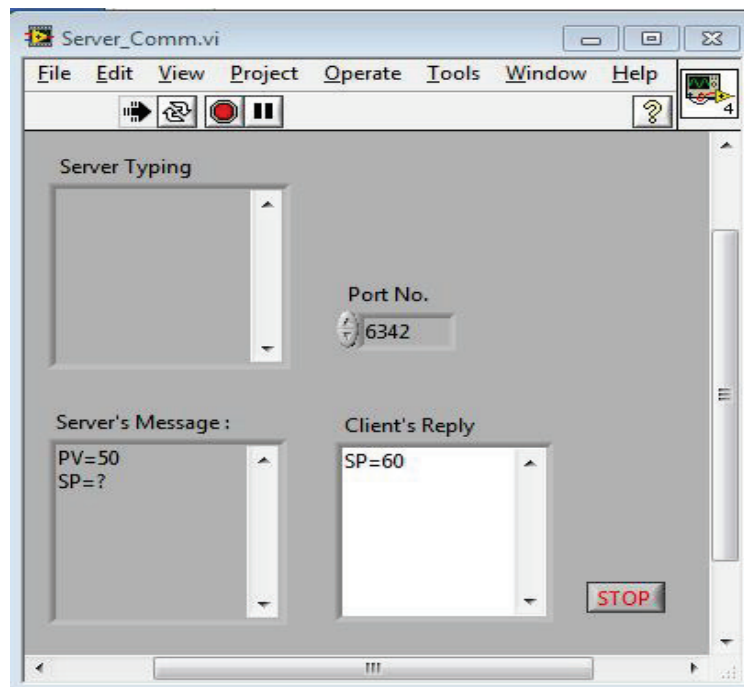


Figure 4.8: Front Panel of Communicating Server

The front panel of communicating client is shown in Figure 4.9. The client on receiving the value of PV sends the value of SP to server by writing a message as “SP=60” in “Client Typing” box found in the front panel. On pressing enter while running condition, this message can be seen in “Client’s Message” which shows the message you typed. This message is sent to server with port number 6342 and can be seen in “Client’s Reply” box.

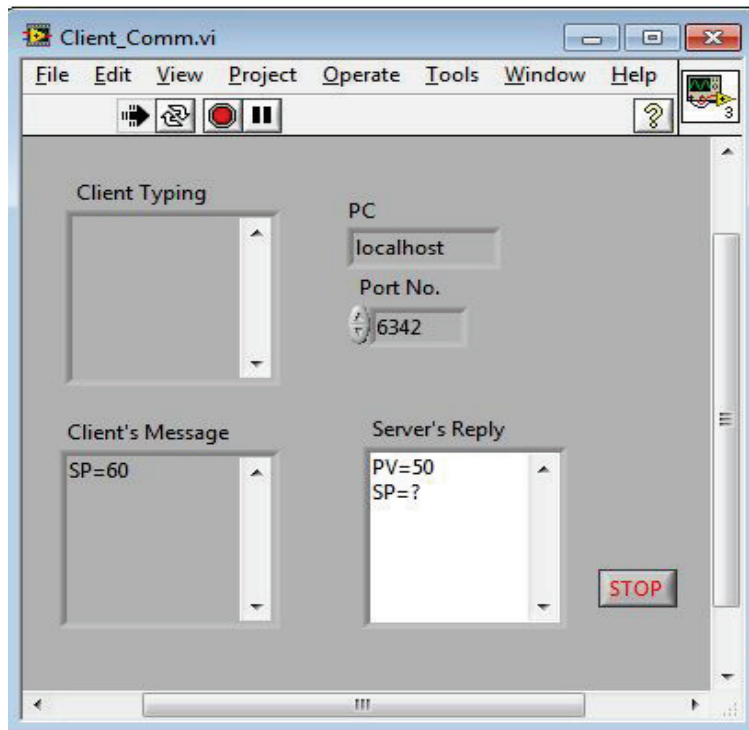


Figure 4.9: Front Panel of Communicating Client

Chapter 5

Conclusion

Limitation of the thesis
Scope for future research

5 CONCLUSION

The thesis shows the applications of internet protocols in LabVIEW, to control the temperature of a heating body. The software has got wide applications in industries for various purposes including generation of C and VHDL code. It is very easy to use since it provides blocks which give a virtual existence of instrument. This chapter concludes the thesis, provides the limitations and the future scope of the project. In chapter 3, algorithm for the two controllers has been discussed. The implementation of these algorithms has been properly done and corresponding simulation results has been obtained.

DataSocket is an extremely powerful way to transfer data and to create easy connections between computers. The DataSocket Read and DataSocket Write VIs support various protocols including DSTP, OPC, Lookout, HTTP, FTP and File retrieval. Through the use of front panel Datasocket connections, a user can quickly and easily establish a connection to a data item. This capability will improve the networking capabilities of many measurement and automation applications. DataSocket is an easy-to-use, high-performance programming tool that is designed specifically for sharing and publishing live data in measurement and automation applications between different applications and between machines across the Internet. DataSocket for LabVIEW simplifies live data exchange between different applications on one computer or between computers connected through a network.

Although ON/OFF is a very cheap form of control, it is rarely used in process control applications because of the oscillation which causes in the controlled and manipulated variables. In a connected process these oscillations would propagate right through the system and causes problems. The thesis brings the applications of LabVIEW to perform online operations. In chapter 4, online written communication that is “Chatting” has been discussed.

5.1 Limitations of the Thesis

The limitations of the thesis can be the following:

- The simulation results are obtained in software only, no hardware is involved.
- New technology for controllers is not implemented.
- Techniques like FTP, HTTP; etc which is now-a-days used for fast online transmission of

information is not implemented.

5.2 Scope for Future Research

A continuous development of new control algorithms insure that the time of PID controller has not past and that this basic algorithm will have its part to play in process control in foreseeable future. It can be expected that it will be a backbone of many complex control systems. The work can be extended to following areas,

- Implementation of hardware which would help in industrial application.
- Controlling temperature by using fuzzy PID controller.
- The application of TCP/IP can be extended to multiple accesses by client and server, service tasks, etc.

The graphs obtained can be published in web browser using HTTP, can be sent from one user to another by using FTP (File Transfer Protocol), e-mails, etc.

References

-
- [1] E.K. Rézaei and S.R. Kolla, “Internet-based ON/OFF controller using LabVIEW”, Proceedings of 2003 American Society for Engineering Education Annual Conference & Exposition, American Society for Engineering Education, 2003.
- [2] K. Prema, N. Kumar, and K.A. Sunitha, “Online temperature control based on virtual instrumentation”, Proceedings of International Conference on Control, Automation, Communication and Energy Conservation, (INCACEC), pp. 1–4, June 2009.
- [3] A.O. Neaga, C. Festila, E.H. Dulf, R. Both, T. Szelitzky, et al., “Monitoring and control of liquid nitrogen level of the 13c separation column using ni pxi-1031”, Proceedings of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), pp. 35–40, May 2011.
- [4] R. Dutta, and P. Ghosh, “Implementation of a Smart Temperature Controller in Real Time using LabVIEW”, International Journal of Smart Sensors and Ad Hoc Networks, (IJSSAN) ISSN No. 2248-9738, Vol. 1, Issue 2, 2011.
- [5] S.S. Murthy, R.K. Mittal, A. Dwivedi, G. Pavitra, and S. Choudhary, “Online performance monitoring and testing of electrical equipment using virtual instrumentation”, Proceedings of 7th International Conference on Power Electronics and Drive Systems,(PEDS), pp. 1608–1612, Nov. 2007.
- [6] J. H. Zhang and X. M. Liu. “An acquisition system for remote diagnostics of airport special equipment based on LabVIEW”, Proceedings of International Conference on Computational and Information Sciences (ICCIS), pp. 744–746, Oct.2011.
- [7] A.J.A. Nazir, Gautham, R. Surajan, and L. Binu, “A simplified genetic algorithm for online tuning of PID controller in LabVIEW,” Proceedings of World Congress on Nature Biologically Inspired Computing, pp.1516–1519, 2009.
- [8] J. Liu, P. Zhang, and F. Wang, “Real-time dc servo motor position control by PID controller using LabVIEW”, Proceedings of International Conference on Intelligent Human-Machine Systems and Cybernetics, (IHMSC), vol. 1, pp. 206–209, Aug. 2009.
- [9] S. Ravi, and P. Balakrishnan, “Modeling and control of an ANFIS temperature controller for plastic extrusion process,” Proceedings of IEEE International Conference on Communication Control and Computing Technologies, pp. 314-320, 2010.

-
- [10] F. Faizan, F. Farid, M. Rehan, S. Mughal, M.T. Qadri, "Implementation of discrete PID on Inverted pendulum," Education Technology and Computer (ICETC), Proceedings of 2nd International Conference on , pp. 22-24, June 2010
- [11] S. K. Sahoo, R. Sultana, and M. Rout, "Speed control of dc motor using modulus hugging approach", Proceedings of International Conference on Sustainable Energy and Intelligent Systems (SEISCON), pp. 523–528, July 2011.
- [12] V. Kumar, K. Rana, and V. Kumar, "Real time comparative study of the performance of FPGA based PID and fuzzy controllers for a rectilinear plant," Proceedings of India International Conference on Power Electronics, pp. 1–7, 2011.
- [13] S. Zhong, K. Chunpeng, Z. Weina and X. Dawei, "Research of PID parameter self-tuning applied in temperature control system," Proceedings of 8th International Conference on Electronic Measurement and Instruments, vol. 1, pp. 360-363.
- [14] Hai-bo Lin, "A kind of intelligent temperature controller use PID algorithm to realize," Electrical and Control Engineering (ICECE), 2011 International Conference on, pp.2838,2840, Sept. 2011
- [15] S. Mohammadi, H. Pour, M. Jafari, and A. Javadi, "Fuzzy-based PID active queue manager for TCP/IP networks," Proceedings of 10th International Conference on Information Sciences Signal Processing and their Applications, pp. 434 –439, 2010.
- [16] D. Poulton, O. Herrero, and J. Oksman, "Non- uniformly sampled power control scheme for TCP based data transfers on a faded satellite link," Proceedings of 10th International Conference on Telecommunications, vol. 1, pp. 1590– 1595, 2003.
- [17] A. Wei, Y. Chen, and J. Wu, "Simulation Study of TCP/IP Communication Based on Networked Control Systems," Proceedings of the 6th World Congress on Intelligent Control and Automation, vol. 1, pp. 4479-4483, 2006.
- [18] F. Xiao, and Z. Zhu, "Research of embedded web server based on can-TCP/IP gateway," Proceedings of International Conference on Intelligence Science and Information Engineering, pp. 279–281, 2011.
- [19] Y. Li and K. Jiang, "Prospect for the future internet: A study based on TCP/IP vulnerabilities", Proceedings of International Conference on Computing, Measurement, Control and Sensor Network (CMCSN), pp. 52–55, July 2012.

DISSEMINATION

Journals:

1. Abhyarthana Bisoyi and U. C. Pati, "Implementation of ON/OFF and PID controller using TCP Protocol Based on Virtual Instrumentation", Published for *International Journal of Advanced Computer Research*, Mar. 2013.
2. Abhyarthana Bisoyi and U. C. Pati, "Online Written Communication using Internet Protocol based on Virtual Instrumentation", Accepted for publication in *Journal of Instrument Society of India*, Sep. 2013.

Conference:

1. Abhyarthana Bisoyi and U. C. Pati, "Implementation of ON/OFF and PID controller using TCP Protocol Based on Virtual Instrumentation", *Proceedings of International Conference on Advance Computing and Communication*, Ranchi, pp. 23-27, 16-17 Mar. 2013. (Published)