

**MONITORING AND CONTROL OF THREE PHASE  
SERVODRIVE  
USING  
PLC AND SCADA**

A dissertation submitted in partial fulfillment  
of the requirement for the award of the degree of

**MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING  
(CONTROL AND INSTRUMENTATION)**

Submitted By

**POONAM JAYAL**

*College Roll No. 07/ME(C&I)/2009*

Under the Supervision of

**Dr. MADHUSUDAN SINGH**



**Department of Electrical Engineering**

**Delhi College Of Engineering**

**Delhi University**

**Delhi-110042**

**INDIA**

**2011**

## ***CERTIFICATE***

*This is to certify that **Poonam Jayal** (07/ME(C&I)/09) has carried out her major Project work in partial fulfilment for the degree of **Master of Engineering in Control and Instrumentation** on the topic “**Monitoring and control of Three Phase Servodrive using PLC and SCADA**” during the period July 2010 to June 2011, under my supervision and guidance and has completed the project to my satisfaction.*

*Dated:*

( ***Dr. Madhusudan Singh*** )  
Project Supervisor  
Department of Electrical Engg.  
DTU  
Delhi 110042

## **ACKNOWLEDGMENTS**

*I take this opportunity to express my deepest gratitude to Dr. **Madhusudan Singh**, my Project Supervisor, for his constant motivation, invaluable guidance and encouragement in my pursuit of exploring, understanding and implementing the technical concepts related to this project. It was indeed a privilege and honour to work under his supervision.*

*I would also like to sincerely thank **Mr. Manick Raj**, GE India, for taking time out of his busy schedule for responding to my system related queries from time to time. His inputs have been truly valuable.*

*I am also grateful to our respected Head of the Department **Prof Narendra Kumar** and the entire **Electrical Engineering Department staff** for their kind cooperation in rendering the necessary resources for the successful completion of this project.*

*I am also thankful to my friends **Saima, Nikhil and Sandeep** for their kind endeavours.*

*I owe my loving thanks to all my **family members** for their support and understanding through thick and thin, which gave me the inner strength to sail smoothly through this journey.*

*Special thanks to **Prof V.K Joseph**, Goa Engineering College, my mentor and guide since my undergraduate days, who in his special ways has always been a friend, philosopher and guide.*

*Last but not the least, my heartfelt Thanks to **the Almighty** for showing me the right path and being with me ALWAYS!!*

# CONTENTS

Certificate.....	ii
Acknowledgements.....	iii
Contents.....	iv
List Of Figures.....	viii
List Of Tables.....	xi
Abstract.....	xii

## **CHAPTER 1 INTRODUCTION 1-5**

- 1.1 General
- 1.2 State of Art in Three Phase Permanent Magnet AC Drive
- 1.3 Scope of Present work
- 1.4 Outline of the Thesis

## **CHAPTER 2 LITERATURE REVIEW 6-15**

- 2.1 General
- 2.2 Role of PLC in Automation
- 2.3 Permanent Magnet AC Drives
- 2.4 SCADA Applications
- 2.5 Conclusion

## **CHAPTER 3 HARDWARE DESCRIPTION 16-37**

- 3.1 General
- 3.2 Programmable Logic Controller
  - 3.2.1 Power Supply
  - 3.2.2 Input Module
  - 3.2.3 Output Module
  - 3.2.4 CPU
- 3.3 Three Phase AC Servomotor (PMSM)
  - 3.3.1 Modelling of the PMSM

- 3.3.2 Vector Control Theory of PMSM
- 3.3.3 Speed-Torque Characteristics of PMSM
- 3.4 Three phase Voltage Source Inverter fed PWM based AC Servo drive
- 3.5 SCADA (Supervisory Control and Data Acquisition System) as the Master station for control
- 3.6 Connection Diagram Of the PLC with the VersaMotion Servo Drive
- 3.7 Conclusion

**CHAPTER 4      AN OVERVIEW OF THE SYSTEM SOFTWARE      38-76**

- 4.1 General
- 4.2 The Proficy Machine Edition 5.90
  - 4.2.1 Proficy Machine Edition environment
  - 4.2.2 General Concepts about Machine Edition Projects
  - 4.2.3 Development of a PLC program in Proficy Machine Edition 5.9
- 4.3 VersaMotion Configuration Software
  - 4.3.1 General tools of Versamotion Configuration Software
  - 4.3.2 The Digital Inputs and Digital Outputs of the Servo Drive
  - 4.3.3 Running the Servo Motor under various Control Modes Using the Versamotion Configuration Software
    - 4.3.3.1 Speed Control Mode
    - 4.3.3.2 Position Control Mode
    - 4.3.3.3 Torque Control Mode
- 4.4 Proficy HMI/SCADA Cimplicity 7.5 Software
  - 4.4.1 Setting up a Cimplicity Project for controlling a system remotely
  - 4.4.2 Creating a Cimplicity Project for Monitoring the Servomotor Variables on the SCADA Screen
- 4.5 PLC Programming
  - 4.5.1 General
  - 4.5.2 Instruction Set Used for Programming GE Fanuc Series 90-30 PLC
  - 4.5.3 User References for PLC Series 90-30
  - 4.5.4 Important Features of the Ladder Diagram
- 4.6 Conclusion

**CHAPTER 5                    COMMUNICATION PROTOCOL IN PLC,                    77-88**  
**SERVODRIVE AND SCADA**

- 5.1 General
- 5.2 Serial Communication
  - 5.2.1 MODBUS
  - 5.2.2 Communication between MODBUS devices
  - 5.2.3 The Query-Response Cycle
  - 5.2.4 Serial Transmission Modes of Modbus Networks
- 5.3 ETHERNET
  - 5.3.1 IP Addressing
  - 5.3.2 Description of the Present Network Configuration
- 5.4 Conclusion

**CHAPTER 6                    EXPERIMENTAL RESULTS AND DISCUSSIONS                    89-109**

- 6.1 General
- 6.2 Software Implementations
  - 6.2.1 Ladder Programs for PMSM Control Using Proficy Machine Edition 5.90
    - 6.2.1.1 Ladder Logic to Start and Stop a Servomotor
    - 6.2.1.2 Ladder Logic to run Servomotor in Internal Position Control Mode
    - 6.2.1.3 Ladder Logic to run Servomotor in Internal Speed Control Mode
  - 6.2.2 Ladder Program for Servo Motor Control Using Cimplicity/HMI SCADA Edition 7.5
    - 6.2.2.1 Ladder Logic to develop animated screen for Start and Stop of Servomotor
    - 6.2.2.2 Bottle Filling, Capping and Labelling Application - A Case Study of PLC/SCADA based Virtual Monitoring System
    - 6.2.2.3 Test Tube Filling Application - A Case Study of PLC/SCADA based Virtual Monitoring System
- 6.3 Experimental Observations and Discussions
  - 6.3.1 Servomotor in Internal Speed Control Model
    - 6.3.1.1 Starting Characteristics of the PMSM Drive
    - 6.3.1.2 Characteristics of the PMSM Drive when it Stops
    - 6.3.1.3 Trajectory Tracking in Speed Control Mode

6.3.2 Input and Output Parameters of the PMSM Drive

6.4 Conclusion

<b>CHAPTER 7</b>	<b>MAIN CONCLUSION AND FUTURE SCOPE OF WORK</b>	<b>110-112</b>
7.1	Main Conclusion	
7.2	Future Scope of Work	
<b>REFERENCES</b>		<b>113-116</b>

## LIST OF FIGURES

3.1	System Layout Diagram.....
3.2	The real time control set - up which was configured and tested.....
3.3	Detailed System Block diagram.....
3.4	Basic block diagram of a PLC.....
3.5	Power Supply Front Panel.....
3.6	CPU 374 Front Panel Details.....
3.7	Cross-sectional views of two major types of PMAC motor construction.....
3.8	Phasor Diagram of PMSM.....
3.9	Speed-Torque Characteristics of the 1KW PMSM Motor Used in this Project.....
3.10	Internal architecture of PWM based servo-drive.....
3.11	PWM Scheme for Servodrive.....
3.12	Three Phase Inverter Bridge.....
3.13	PWM control of a three phase motor.....
3.14	Versamotion Servo drive front panel.....
3.15	Connection Diagram of PLC with AC Servodrive.....
4.1	Proficy Machine Toolbar Proficy Machine windows.....
4.3	New Project Start up window of Proficy machine.....
4.4	Naming the new project.....
4.5	Addition of GE-Fanuc PLC as the target.....
4.6	Configuration of the PLC Hardware.....
4.7	Development of Ladder program in Proficy Machine.....
4.8	Shortcut keys to go online.....
4.9	VersaMotion start-up windows (a,b).....
4.10	Servomotor parameter monitoring on Data scope.....
4.11	Parameter Configuration Window.....
4.12	Digital Input/Output Jog Control Box.....
4.13	Status monitor window.....
4.14	Block Diagram for Speed Control Mode.....
4.15	Block diagram for Position Control Mode.....
4.16	Block Diagram for Torque Control Mode.....
4.17	New Project creation in Cimplicity.....
4.18	Configuring Cimplicity Project Properties (a,b).....



4.19	Cimplicity Project Wizard.....	
4.20	Checking Device communication status using Quick View (a,b).....	
4.21	Creating the SCADA screen using CIMEDIT.....	
4.22	Configuring Cimplicity Project Properties.....	
4.23	Configuring a new Port.....	
4.24	Configuring Port Properties.....	
4.25	Configuring a new device.....	
4.26	Configuring Device Properties.....	
4.27	Creating a new device point.....	
4.28	Configuring the Point Properties.....	
4.29	Point Control Panel Window.....	
4.30	Tracking of Point status on the Point Control Panel.....	
4.31	Input table file operation.....	
4.32	Output table file operation.....	
5.1	MODBUS ISO/OSI Model.....	
5.2	The Query-Response Cycle of MODBUS protocol.....	
5.3	MODBUS Protocol Data Unit .....	
5.4	General MODBUS Message frame over serial line.....	
5.5	The Data Frame (For 8-bit data).....	
5.6	The MODBU-ASCII Message frame.....	
5.7	The MODBUS-RTU Message Frame.....	
5.8	IP Address Format.....	
5.9	Block Diagram showing Communication between all System components .....	
6.1	Flowchart for start and stop operation of PMSM.....	
6.2	Ladder logic for Start and Stop of Servomotor.....	
6.3	Flowchart for internal Position control mode of Servomotor.....	
6.4	Ladder Logic to operate a Servomotor in Internal Position control Mode (a-c).....	
6.5	Flowchart to run the servomotor in internal speed control mode.....	
6.6	Ladder Logic to run the servomotor in internal speed control mode.....	
6.7	SCADA based HMI Screen Developed for Motor ON/OFF Control.....	
6.8	Ladder logic for developing the screen of Fig. 6.7.....	
6.9	SCADA screen showing Animated Bottle Filling, Capping and Labelling Application.....	
6.10	Ladder logic for the developing the screen of Fig. 6.9 (a-c).....	

6.11	SCADA Screen showing animated TestTube filling Application.....	
6.12	Ladder logic for the developing the screen of Fig. 6.11.....	
6.13	Transient response of the PMSM drive during starting operation.....	
6.14	Transient Response of the PMSM drive during stopping operation.....	
6.15	Trajectory tracking of PMSM under no load condition .....	
6.16	Trajectory tracking of PMSM under loaded condition.....	
6.17	Magnified view of motor switching from rest to 1500rpm with motor at no load....	
6.18	Magnified view of motor switching from rest to 1500rpm with motor under load.....	
6.19	Input Voltage and input Line current of the PMSM Drive at 3000 rpm and under (a) No load (b) Loaded condition.....	
6.20	(a) PWM Voltage at PMSM Drive output at 3000 rpm under load.....	
6.20	(b) Current drawn by the PMSM Drive at 3000 rpm under load.....	

## LIST OF TABLES

Table 3.1: Power Supply Specifications.....	2
Table 3.2: Description of the LED indicators on the PLC.....	
Table 3.3: PLC Discrete Input Module Specification.....	
Table 3.4: PLC Discrete Output Module Specification.....	
Table 3.5: CPU (IC693CPU374) Specifications.....	
Table 3.6: AC Servomotor Specifications.....	
Table 3.7: AC Servodrive specifications.....	
Table 4.1: Detailed Description of the Digital Input Signals (DIs).....	
Table 4.2: Detailed Description of the Digital Output Signals (DOs).....	
Table 4.3: Different Modes of Servomotor operation.....	
Table 4.4: Digital Input Settings for Internal Speed Control Mode.....	
Table 4.5: Digital Output Settings for Internal Speed Control Mode.....	
Table 4.6: Parameter settings for Internal Speed Control Mode.....	
Table 4.7: Important parameters for Internal Position Control Mode.....	
Table 4.8: Digital Input Settings for Internal Position Control Mode.....	
Table 4.9: Digital Output Settings for Internal Position Control .....	
Table 4.10: Parameters for Internal Position Control Mode.....	
Table 4.11: Parameter Settings for Internal Position Control Mode.....	
Table 4.12: Important Parameters for Torque Control Mode.....	
Table 4.13: Digital Input Settings for Internal Torque Control Mode.....	
Table 4.14: Digital Output Settings for Internal Torque Control Mode.....	
Table 4.15: Parameter Settings for Internal Torque Control Mode.....	
Table 4.16: Important Instructions for GE Fanuc Series 90-30 PLC.....	
Table 5.1: Comparison of RS-232, RS-422 and RS-485 serial standards.....	
Table 5.2: Function Codes supported by MODBUS RTU slave.....	
Table 5.3: Identifying class of an IP Address.....	
Table 5.4: Ethernet Configuration used in the Project.....	
Table 6.1: Parameter settings for the Internal position control mode.....	
Table 6.2: Speed Control Mode Parameters.....	

## **ABSTRACT**

---

*The present work involves the study, development and implementation of a semi-automated system using a Programmable Logic controller for the control and monitoring of a Three phase Servodrive (Permanent Magnet Synchronous Motor drive). Three Phase Servodrives using PMSM are increasingly used for precision control applications in process industries due to their high torque to inertia ratio and operation at higher power factor as compared to Induction Motor. Most of the earlier research on PMSM focused on performance improvement through the use of advance control algorithms such as Vector control, sensorless control, direct torque control, etc. However modern industrial processes which involve multimotor drives require a co-ordinated sequential control through PLC and SCADA system. So far there has been very limited work on the PLC-SCADA control of a PMSM. The present work is an effort to explore this aspect of the PMSM control.*

*An integrated system comprising of a Workstation (PC), the PLC(GE-Fanuc), Permanent Magnet AC Servo Drive, three phase Servomotor(PMSM) and the software for Supervisory Control and Data Acquisition system are configured and developed in co-ordination with each other to ultimately control the three phase Permanent Magnet synchronous drive in multiple ways. The PMSM is configured and controlled in position and speed control modes. An effort is made to monitor the PMSM runtime variables like speed, torque and current. The necessary communication protocols are established between the various devices to ensure flawless communication. The real time Performance of the Permanent Magnet Synchronous motor is experimentally verified and captured on the Drive software. Various virtual animated screens are developed to explain the PMSM precise position control applications.*

# **CHAPTER I INTRODUCTION**

---

## **1.1 GENERAL**

This dissertation presents a PLC-SCADA based monitoring and control scheme for a three-phase Servo Drive system, which is basically a Permanent Magnet Synchronous Motor (PMSM) drive in the present set-up. Programmable Logic controllers (PLC) and Supervisory Control and Data Acquisition (SCADA) system have been an integral part of the industrial automation processes since 1960s and have evolved over the years with advances in technology. They enhance the quantity and quality of production. PLCs were developed in 1968 for the automotive manufacturing industry to provide a replacement for large relay based control panels. The development of the micro processor from the mid 1970's have allowed them to take on more complex tasks and larger functions as the speed of the processor increased. SCADA which is actually the combination of telemetry and data acquisition involves the collection of the information via a Remote Terminal Unit(RTU), transferring it back to the central site, carrying out any necessary analysis and control and then displaying that information on a number of operator screens or displays. The required control actions are then conveyed back to the process.

The overall system set up which forms an integral part of this dissertation work aims to provide a fairly good platform for developing the concepts, methodologies and the complete understanding of an industrial automated system comprising of the commonly used components like PLC, SCADA, Servo Drive and Motor along with a HMI device. The PLC proves to be a very versatile and effective tool in the control of the Three Phase Servo drive.

## **1.2 STATE OF ART IN THREE PHASE SERVO DRIVE**

Since technology for motion control of electric drives became available, the use of programmable logic controllers (PLCs) with power electronics in electric machines applications has been introduced in the manufacturing automation [24]. This use offers advantages such as lower voltage drop when turned on and the ability to control motors and other equipment with a virtually unity power factor [5]. Also the use of PLCs in automation processes diminishes production cost and increases quality and reliability To

obtain accurate industrial electric drive systems, it is necessary to use PLCs interfaced with power converters, personal computers and other electric equipments.

Some papers have already been published describing the control of AC/DC machines using a PLC based system. A PLC-based system for the control and monitoring of a Three-Phase Induction motor showed how the efficiency and speed regulation increased by using a PLC as compared to the conventional V/f control system [4]. The PLC was used to correlate and control the operational parameters to the speed set point requested by the user and continuously monitor the induction motor system during normal operation and under trip conditions. An industrial PLC was used for controlling stepper motors in a five-axis rotor position, direction and speed, reducing the number of circuit components, lowering the cost, and enhancing reliability[6]. In another research paper a PLC based power factor controller was proposed and experimentally verified for a Three phase induction motor [5]. There are very few references on the application of PLC and SCADA system for performance monitoring of PMSM drives. The present work is an attempt to fill this void and contribute significantly towards this area of research.

The present work focuses on a Permanent Magnet Synchronous Motor (PMSM) used in a Servo system. A Servo System is used to control the output, such as the rotation angle and position of the controlled object, to accurately follow the given instructions. For Servo system AC servo drives are preferred in spite of being more complicated and expensive than DC servos as the former lacks robustness and overload capability, has lower torque, and speed bandwidths and it requires regular brush and commutator maintenance. The machines used for AC servos are the Induction motors, Permanent Magnet Synchronous motors and Brushless DC motors. Due to its outstanding performances such as high resolution, wide speed range, low moment of inertia, lower torque ripple, high power density, high power factor, rugged construction, high torque capability, a higher-torque-to-inertia ratio, simple and easy maintenance and high efficiency the PMSM has become the mainstream motor of AC servo system.

There are two major classes of PMAC motor drives which can be characterized by the shapes of their respective back-EMF waveforms- sinusoidal and trapezoidal, the former being the Permanent Magnet Synchronous Motor (PMSM) drive and the latter being the Brushless DC (BLDC) motor drive. Under idealized conditions, each of these two

types of **PMAC** drives is capable of delivering perfectly smooth instantaneous torque waveforms. We have focused on the PMSM drive control in the present work.

The Permanent-Magnet Synchronous motor (PMSM) has numerous advantages over other machines that are conventionally used for AC servo drives. In fact recent research has shown that PMSM could become a serious competitor to the Induction Motor for servo applications. In Induction Motor (IM) drive, the stator currents contain both the magnetizing current and torque-producing current. The use of the permanent magnet in the rotor of the PMSM makes it unnecessary to supply the magnetizing current through the stator windings; the stator current need only be torque-producing. Hence for the same output, the PMSM will operate at a higher power factor (because of the absence of magnetizing current and no associated rotor loss) and will be more efficient than the IM. The conventional wound-rotor synchronous machine (SM), on the other hand, must have dc excitation on the motor, which is often supplied by brushes and slip rings. This implies rotor losses and regular brush maintenance, which implies poor efficiency. PMSM has a higher-torque-to-inertia ratio and power density when compared to the IM or the wound-rotor Synchronous Motor, i.e for the same output capacity in the above range the PMSM is smaller in size and lower in weight that makes it preferable for certain high performance applications like robotics and aerospace actuators. The PMSM Servo Drive system also finds wide applications in Computer Numerical Control (CNC) machine, automatic sewing machines and military operations like positioning and tracking control.

### **1.3 SCOPE OF PRESENT WORK**

- (a) A PLC-SCADA based control set up basically comprising of a GE-Fanuc PLC, a Three Phase servodrive, a 1 KW Permanent Magnet Synchronous Motor (Servomotor) and the workstation has been developed, configured and integrated together for the monitoring and control of the Servomotor.
- (b) Various control schemes have been used to operate the Servomotor in Speed and Position control modes of operation say using PLC programming, directly through the servodrive configuration software, through animated SCADA screens developed on the workstation etc.
- (c) The detailed mathematical model and the Vector control theory which forms the basis of the PMSM drive system has been developed and analysed.

- (d) An effort has been made to configure the SCADA software to monitor the run time values of the PMSM variables like speed, torque and current.
- (e) The communication standards and protocols used between the various components of the entire control set-up including the PC – PLC – PMSM Drive – SCADA – HMI have been completely identified and summarised.

#### **1.4 OUTLINE OF THE THESIS**

This dissertation is divided into various chapters. **Chapter 1** gives the general introduction about what the dissertation is all about and the earlier work done which motivated this thesis. **Chapter 2** presents a review of all the relevant research papers which were referred to in the process of pursuing this dissertation work. **Chapter 3** gives a detailed hardware description of the complete control set-up used for the monitoring and control of the three-phase servodrive. **Chapter 4** discusses the various softwares used in this dissertation and a step by step guide to their implementation. The description is made elaborate enough to enable a beginner to get a quick acquaintance and hands on with the various software platforms. **Chapter 5** summarizes the communication standards and protocols involved in the present control set up which gives a fair picture of the interconnection of the various components involved in this thesis. **Chapter 6** presents the experimental work, observations and results obtained in this complete process. **Chapter 7** finally presents the conclusion of the thesis and also proposes the future scope of work which can be carried out as an extension to this thesis. Finally the thesis completes with a detailed Reference list.



## **CHAPTER II LITERATURE REVIEW**

---

### **2.1 GENERAL**

This chapter presents a brief review of some research work in the area of process automation and control using PLC and SCADA system. This literature is broadly categorized in to three parts. The first part reviews some papers describing the Programmable Logic Controller's (PLC) role in various automation applications. PLC offers flexible and powerful switching strategies for methodical approach to industrial controls problems. PLC is a popular device and governs most of the industrial processes today. The second part presents a review of papers related to the Permanent Magnet Synchronous Motor (PMSM) Drives, which is the main control system to be monitored. This literature discusses the control strategies, applications, conventional and newer methods of speed and position control of the PMSM and also throws some light on the functioning of an AC Servo drive system. The third part is basically devoted to review of papers on Supervisory Control and Data Acquisition (SCADA) in industries and its real time implementation which gives a fairly good idea about practical SCADA and its components and their integrated operation.

### **2.2 ROLE OF PROGRAMMABLE LOGIC CONTROLLER (PLC) IN AUTOMATION**

A **Firoozshahi et al. [1]** focuses on an innovative and intelligent monitoring system using a PLC for Water Treatment Plant that has been successfully designed, implemented and commissioned for a large Gas Refinery. Water treatment plants have to provide good water quality and at the same time low operational costs. Owing to various physical, chemical and biological interactions water treatment processes are often difficult to handle. The complete set up, design and installation of the entire control system and the configuration of the monitoring system has been adequately described. It has been tested that the PLC automation of the Water treatment plant has proved to be very operator friendly, efficient and economical.

**Anirudha Upadhyay et al. [2]** presents a fully automated solution for controlling chemical water treatment plant process using a PLC and its analysis in LabVIEW. The chemical treatment plant comprises of input pipeline embedded with flow meter sensors. A density meter sensor is placed just before the purifying unit. A PID, Cascade and Feed Forward controller along with reactor forms the water purifying unit. Output flow line also contains density meter and flow meter sensors. Full automation of plant is governed by PLC. A mathematical model used for controller design depicts all the aspects of the chemical water treatment plant. Closed loop process offers optimal efficiency by maintaining the concentration at set point by controlling the flow rate of water feed in purifier. The system implements PID coupled with Dead Time Compensator for maintaining the concentration of harmful effluents under optimum level. Feed forward controller anticipates the temperature of the reactor by measuring the input concentration. Also, a cascade controller is implemented to control the reactor core temperature by controlling the flow rate of cooling water. LabVIEW enables the finding of the values of controlling parameters and analysis of the real time situation to perfection.

**I. Colak et al. [3]** illustrates a protection scheme to protect induction motors against possible failures thereby increasing reliability, efficiency and performance of the system. The proposed approach is a sensor-based technique, therefore, currents, voltages, speed and temperature values of the induction motor were measured with sensors and whenever any fault is detected during operation of the motor, PLC controlled on-line operation system activates immediately. The motor protection achieved is much reliable than classical techniques and can be applied even to larger motors. The protection system developed shows good performance in detecting the faults and in clearing them.

**Maria G. Ioannides et al. [4]** presents a PLC-based monitoring and control system for a three-phase induction motor. It also describes the complete design and implementation of the configured hardware and software. A speed sensor is employed for speed feedback, a current sensor for load current feedback and an additional current sensor is attached to stator circuits. The PLC correlates the operational parameters to the speed requested by the user and monitors the system during normal operation and under trip conditions. A control program has been developed, in accordance of which PLC continuously monitors the inputs and activates the outputs accordingly. Tests of the induction motor system driven by inverter and controlled by PLC prove a higher accuracy in speed regulation as

compared to a conventional V/ f control system. The efficiency obtained by using PLC control is increased as compared to the open-loop configuration of the induction motor fed by an inverter .Thus, PLC proves themselves as a very versatile and effective tool in industrial control of electric drives.

**A.R.Ali et al. [5]** proposes a power factor controller for a 3-phase induction motor, utilizing Programmable Logic Controller. This paper focuses on the implementation of a model for a PLC based power factor controller to improve the power factor of a 3-phase induction motor. Voltage to frequency ratio has been maintained constant to obtain maximum torque. The power factor controller hardware comprises of S7-PLC, a 3-phase squirrel cage induction motor coupled with a dc shunt generator and an electronic conditioning circuit. Features like interlocking, i.e., disabling the controller until it detects currents, voltage, frequency and power factor angles, switch failure detection, monitoring voltage to frequency ratio constant during correcting the power factor, independent control of reactive currents in each phase and maximum compensation even when switch failure occurs, have all been incorporated in the software program. It was also proposed that the control strategy developed may be adopted for power factor and protection control. Such controller will not introduce any harmonics in the system and may prove to be cost effective.

**A.S.Zein El Din et al. [6]** illustrates an industrial PLC used for controlling five-axis rotor position, direction and speed, thereby decrementing circuit components and cost and at the same time enhancing reliability. Ladder diagram has been used to control robot through PLC. The robustness of the PLC controller has been achieved by rotating the motor in both clockwise as well as counter-clockwise directions. Consequently, the PLC controller proves to be a simple tool for controlling robot by extremely simple algorithm

**Robert A. Hanna et al. [7]** presents development and design of Programmable Logic Controller (PLC) programs for optimising the operation of electrical distribution system of a refinery. The power distribution of the refinery faces problems of severe voltage dips or short duration power interruptions due to which on-line starters de-energize and motors stop resulting in plant upset. In order to minimize the impact of momentary power disturbances, it is highly required that motors used for critical services be restarted automatically and sequentially. This paper presents typical utility incoming power disturbance investigation, criteria for selecting critical motors and development and

design of PLC programs. The use of PLC in the system makes it automatic, hence solving major problems.

**Akram Hossain et al. [8]** presents the design, implementation and testing of hardware and software interfaces for a basic control problem involving the regulation of water level and the temperature level in a tank with an Allen Bradley 5/25 PLC forming an integral part of the process control loop. The system also comprises of a PC workstation to communicate with the PLC processor module through data highway for programming and controlling. The workstation is used for graphical display of process variables, data acquisition and much other information for control purpose. The control problem is tackled by using cascaded and feedforward control strategy. RTD is used as the temperature sensor and a level transmitter for the water level measurement. The hardware and software interface among the process control Unit, an Allen Bradley PLC 5/25, analog input/output modules, 115VAC input/output modules and a data highway module are successfully established. The ladder program for the control scheme is written and implemented. This paper provides an excellent tool for teaching the basic aspects of the modern industrial control process.

### **2.3 PERMANENT MAGNET SYNCHRONOUS MOTOR (PMSM) DRIVES**

**Hassan. M. Kamel et al. [9]** presents a fuzzy logic controller (FLC) for speed control of permanent magnet synchronous motor (PMSM). It is a known fact that in order to meet the high performance tracking criteria the speed controller in a drive should enable the drive to follow any reference speed, taking into account the effects of load impact, saturation and parameter variation. Conventional controllers such as P, PI and PID controllers need accurate mathematical models describing the dynamics of the system under control and are difficult to design in the absence of one. In this paper the proposed FLC is used to overcome the nonlinearity problem of PMSM. The specified rules and their membership functions of the proposed fuzzy system are represented. The fuzzy rules and the membership functions are tuned to give the best performance. Also the dynamic response of PMSM with the proposed controller is studied and experimentally verified for the starting process and under different load disturbances and its effectiveness is compared with that of the conventional PI controller using a Matlab/Simulink model. The response of fuzzy logic controller is found to be better than that of a PI controller.

**M. F. Moussa et al. [10]** proposes a unity power factor control (UPFC) drive for the PMSM. The UPF control drive technique is explained and designed to achieve this constraint, by two methods : controlling the d-axis stator current  $i_d$  and controlling the space angle of the command stator current. The drive system is built using MATLAB-SIMULINK software. The performance of UPFC is compared with the conventional decoupled vector drive and is evaluated in both steady-state condition and transient condition using computer simulation. It is concluded that UPFC provides an extension of the constant torque region, resulting in higher output power of the PMSM drive, which is desirable in many applications requiring extended speed range at rated motor torque. However, this drive is not optimal in terms of efficiency which will be less than that obtained from conventional decoupled vector control drive for the same torque. Therefore, before reaching the rated speed, the conventional decoupled vector control is preferable, whereas, the UPF control is optimal to have a wider range of speed operation (above the base speed of the conventional vector control) and hence, provides an extension of the constant torque region.

**Ravindra Kumar Sharma et al. [11]** describes the concept of vector control in adequate detail and implements it in real time on a PMSM to obtain linear dynamics similar to that of a DC motor. The objective of the control scheme is to achieve very fast response. The performance of PMSM drive system under different speed reference input is evaluated on the basis of simulation and experimental results. All the experiments have been done on PMSM with the help of ADMC-401 motor control DSP and an inverter circuit. The speed and current controller are then designed. The nonlinear equations of the PMSM, current and speed controller equations and real time model of the inverter switches and vector control are used in the simulation. The time varying abc currents are made stationary using Reverse Park Transformation, to simplify the calculation of PI controller's constants. Clarke and Park transformations are also discussed in adequate detail.

**Kai Sun et al. [12]** presents an application of the PMSM to drive air-condition compressor for electric vehicle in high speed applications. Unlike the conventional PMSM drive characteristics, the requirements from the drive for this application are slightly different. Firstly since the PMSM is installed inside the compressor full of corrosive liquid, the position sensor can not be mounted on it. Hence a sensorless control algorithm must be employed. Secondly, the PMSM is required to be working in high

speed range to increase power density and avoid the vibration and noise caused by pulsated load torque in low speed area. Hence a model reference adaptive system (MRAS) method is used to estimate rotor position to realize sensorless drive. Also a feedforward compensation strategy with maximum torque per ampere (MTPA) control and flux weakening control is proposed for high speed stable operation of PMSM-compressor drive with low power loss. The basic control technique of this drive system is rotor flux oriented vector control. A double mode over-modulation method suitable for compressor drive is proposed to increase motor voltage instead of conventional SVPWM in high speed range. The sensorless over-modulation control algorithm of PMSM-compressor drive has been evaluated through experiments on a real air-conditioner for electric vehicles. A 6kW PMSM is integrated with a compressor. A PWM inverter with CPU V850IA4 based control circuit is used to drive the PMSM. Control program is edited and compiled by using PC and downloaded into CPU through a communication adaptor. Simulation and experimental results demonstrate the feasibility and effectiveness of these proposed control strategies

**S. Z. Jiang et al. [13]** puts forward the concept of AC Servomotor system using PMSM. Two control structures of AC Servomotion system, ie. the distributed and the concentrated structures are compared and it is indicated that the distributed control structure is an ideal choice of initial developing stage of PMSM servo system. In the concentrated control structure, the controller is responsible for the calculation and analysis of the received commands and sensor signals, then controlling the driver and power devices. In the distributed control structure, the controller only functions as the input and display. The microcontroller in the driver plays the main controller's role. It controls not only the ac servo motor, but also the power devices and other drivers. Finally a design implementation of the PMSM servo motion is given which provides a very good basic understanding of how the various components of the AC Servo system are integrated with each other .

**Tae-Won Chun et al. [14]** proposes a method to control the position of a PMSM rapidly and precisely for an embroidery machine. The method suggests adjusting the proportional gain in the position controller to the position error to obtain fast transient response of position without any overshoots. The PMSM drive system is composed of a 4-pole 0.2 KW, 2000 rpm PMSM fed by a current controlled PWM inverter, current

transducer and an incremental encoder. The performance of the PMSM drive system using adjustable P-gain algorithms is verified through computer simulations, which reveal that the PMSM reaches the reference position rapidly without any overshoot.

**Shashank Wekhande et al. [15]** proposes a new encoding scheme based on the Vernier principle, which greatly enhances the feedback and resolution (i.e. providing the shaft position information with utmost accuracy). A brief discussion about the conventional rotary encoders i.e. Absolute and Incremental position encoders is also presented. The incremental encoders are simpler but provide a relative displacement only and are prone to noise. Absolute shaft position encoders require large number of sensors and tracks for higher resolution. A closed loop Servo drive is fabricated for rotating shaft through desired angular displacement and after comparing it with a gray disk encoder it is verified that the Vernier encoder gives better accuracy and prevents continuous oscillation of motor shaft due to large error steps. It is concluded that Vernier encoders provide higher resolution with relatively less expensive sensors, smaller diameter and less precision workmanship.

**Guo Qingding et al. [16]** describes a fully digital implementation of a Permanent Magnet Synchronous motor AC servo system based on self tuning PID control. The whole system is realized by an INTEL8031 and a DSP TMS320C25. It involves a 2.5 KW PMSM motor with a rated speed of 2000 rpm. Such drive systems find wide applications in robotics and numeric machine tools. A self tuning PID controller enables the system to achieve good performance by adjusting the parameters and thereby compensating for the disturbances. In this paper the regulation of the PID parameters is based on the IATE optimal rule for the speed loop and the DSP TMS320C25 with its excellent I/O capabilities and calculation functions carries out the control system algorithm and hence results in a system that has dynamic response, precise control and flexibility. A step response of speed is experimentally performed on the PM machine and the given system shows to have a good dynamic response compared to the analog system.

**Uk-Youl Huht et al. [17]** illustrates a sliding mode controller used for speed control of AC servomotor. Sliding mode controller can be characterized by high accuracy, fast response and robustness. An integral compensator has been added to reduce steady state error and to provide continuous torque reference. The acceleration which is necessary to get the sliding plane is estimated by an observer from actual speed and an integral

compensator. Sliding surface is included in control input to enhance the robustness, insensitivity to sudden change of load and improvement in transient response without increasing sliding mode controller gain. The proposed controller is implemented by DSP (digital signal processor). The effectiveness of the proposed control scheme for speed controller is confirmed through the real-time experimental results. It is also verified that speed response has less chattering to the load variation

**P. Freere et al. [18]** reiterates the fact that PM synchronous motors (PMSMs) can achieve the servo performance characteristics of dc machines using Vector control. This control technique uses the instantaneous rotor position to calculate the required stator currents which are oriented at some specified angle with respect to the rotor. Current controlled inverters are therefore used to force the actual current to follow the commanded current. An experimental evaluation of the different current controllers is made namely hysteresis, delta, ramp comparison and space vector current controllers. An evaluation based on average switching frequency and rms current error (the rms of the difference between the actual and commanded currents ) has been included.

**Pragasam Pillay et al. [19]** describes the d-q model of the PMSM drive system and presents the complete modeling, simulation, and analysis of the drive system. The application of Vector Control technique to the PMSM is also described which forms the basis of a PMSM drive system. This control technique basically transforms a PMSM into an equivalent separately excited DC motor by decoupling the field current (exciting current) and torque current so that independent flux and torque controls are possible. Also the structure of the entire PMSM drive system is explained which comprises of the machine, speed and position feedback, speed and current controllers, and inverter. The use of hysteresis as well as PWM current controllers used in the PMSM drive to generate the required stator currents is also discussed. Digital computer simulations of the entire drive system are also shown.

**P. Pillay et al. [20]** describes the detailed method to derive a mathematical model of the permanent magnet motor drives including the Permanent Magnet Synchronous motor (PMSM) and the Brushless DC motor (BLDC). The PMSM has a sinusoidal back emf and requires sinusoidal stator currents to produce constant torque while the BDCM has a trapezoidal back emf and requires rectangular stator currents to produce constant torque. The PMSM is very similar to the wound rotor synchronous machine except that the



PMSM that is used for servo applications does not have any damper windings and excitation is provided by a permanent magnet instead of a field winding. Hence the  $d, q$  model of the PMSM can be derived from the well-known model of the synchronous machine with the equations of the damper windings and field current dynamics removed. Because of the nonsinusoidal variation of the mutual inductances between the stator and rotor in the BDCM, it is also shown in this paper that no particular advantage exists in transforming the abc equations of the BCDM to the  $d, q$  frame. Hence the solution of the original abc equations is proposed for the BDCM. These models can be used to examine both the transient and steady state behavior of PM motor drive systems.

## **2.4 SCADA IMPLEMENTATIONS**

**Mihai Iacob et al.[21]** presents the design and implementation of a SCADA system for a central heating and power plant ( South Timisoara), meant to supervise and control field distributed electric devices using Siemens software and equipment. The old equipments were replaced by using PLCs, servers, modern approaches regarding network equipments and topologies and flexible monitoring stations. Remote actions and uninterrupted monitoring were made possible due to redundant servers and web-based applications via OPC server and web server. The system was shown to have various advantages like remote and safe operation and monitoring from anywhere in the world, evolution charts for one week, archives, easy to interpret alarm system and most importantly- flexibility, scalability and powerful modular structure.

**Cristina Anita Bejan et al.[22]** presents through this paper a platform for imparting theoretical and practical concepts to students in Supervisory Control and Data Acquisition (SCADA) systems. The laboratory work package is focused to gradually develop applications using an integrated automation system from Siemens -Totally Integrated Automation Democase with distributed peripheral employing Profibus and Ethernet communications. The final application is targeted to control an induction motor with associated frequency touch-screen human machine interface (HMI) to program the motor speed and to show variables, trends and alarms. It is believed that Laboratory works will provide the student competency, experience in identification, selection and programming of needed equipments to make a fully operational application.

**Mini J. Thomas et al.[23]** presents a detailed report about the design, commissioning and functioning of a state-of-the-art SCADA/EMS laboratory facility at Jamia Milia Islamia, New Delhi, India, designed with a purpose to function as a research and training center for utilities, faculty members and students. This lab provides hands on learning experience on SCADA system, and its applications to the management, supervision and control of an electric power System. One of the unique features of the SCADA/EMS laboratory, that makes it the only one of its kind, is the use of a distributed processing system, which supports a global database. Various research activities like adaptive and intelligent control of integrated power systems, preprocessing of data at the RTU level using Fuzzy Logic and Fuzzy-Genetic algorithm, substation automation etc. are already being supported by this laboratory. The various components of SCADA systems such as the master station, RTU, different communication channels and a variety of field equipments are available in the laboratory. The data acquisition is with time stamping, which leads to sequence of events monitoring. A 3-phase transmission line model with on-load tap changer and static VAR acts as one of the field equipments.

## **2.5 CONCLUSION**

An extensive overview of the available literature of Permanent Magnet Synchronous drives and role of PLC and SCADA in industrial processes has been summarized in this chapter, After going through this literature it is noted that only limited number of references are available on PLC application in Drives whereas there is no work reported on application of PLC and SCADA system for performance monitoring of PMSM drives.

## CHAPTER III

### HARDWARE DESCRIPTION

---

#### **3.1 GENERAL**

The PLC-based system in the present work is implemented and tested to control and monitor a PMSM Drive. The Servodrive basically acts as a mediator between the Field device (PMSM) and the system Controller (PLC). An encoder internally fitted with the Servomotor sends the pulses to the Servodrive which are then processed within the drive and finally serve as the input to the PLC (Digital Outputs). The PLC then processes these inputs and takes an appropriate action based on the logic developed for the motor control. The PLC output then becomes the drive input (Digital Inputs) which after being reprocessed within the drive brings about the necessary control of the motor (speed, position or torque control).

The complete hardware comprises of a Programmable Logic Controller, a 1KW Permanent Magnet Synchronous Motor (PMSM), A Three phase Voltage Source Inverter fed PWM based PMSM drive and SCADA (Supervisory Control and Data Acquisition system) software installed in the PC as the mainframe master station for control. Fig. 3.1 shows a layout of the present scheme representing all the above components and Fig. 3.2 gives the true image of the actual set-up configured and tested in the laboratory.

It can be seen that the Three Phase Servodrive front panel has three connectors- CN1, CN2 and CN3. CN1 is a 50 pin I/O connector which sends the drive signals to the PLC. CN2 is the encoder interface and is connected to the incremental encoder inherently present on the shaft within the motor shaft assembly. It taps the feedback signals (position pulses) from the Servomotor and gets them into the drive for further processing. An Analog to Digital converter is present within the drive which converts the analog signal to digital signals so that they can be processed by the controller within the drive. Also there is a Digital to Analog converter within the Drive which sends the necessary control signals for the motor control from the Drive. CN3 connector is for serial communication. It is connected to the serial port of the PC when the Servodrive has to communicate with

it and is connected to the serial port of the HMI panel when HMI applications are to be implemented to operate the drive.

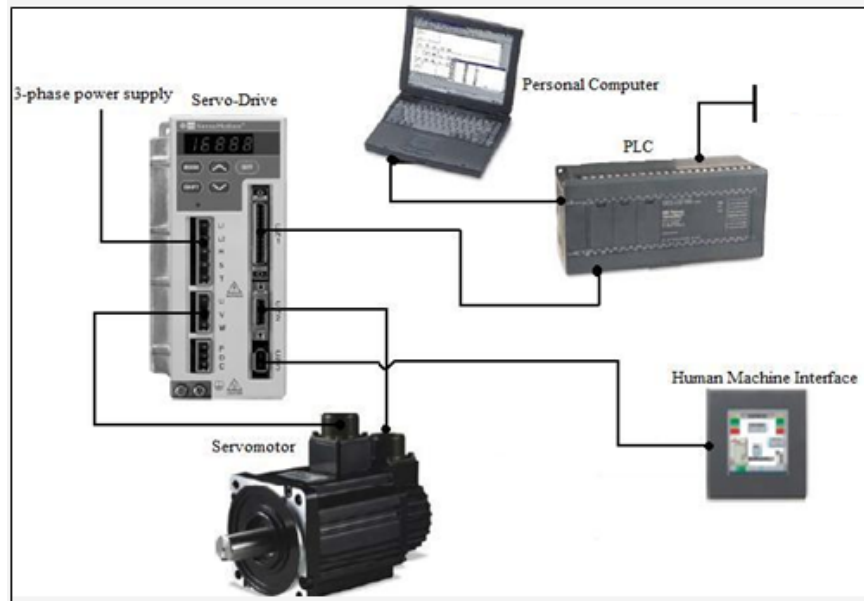


Fig 3.1 System Layout Diagram

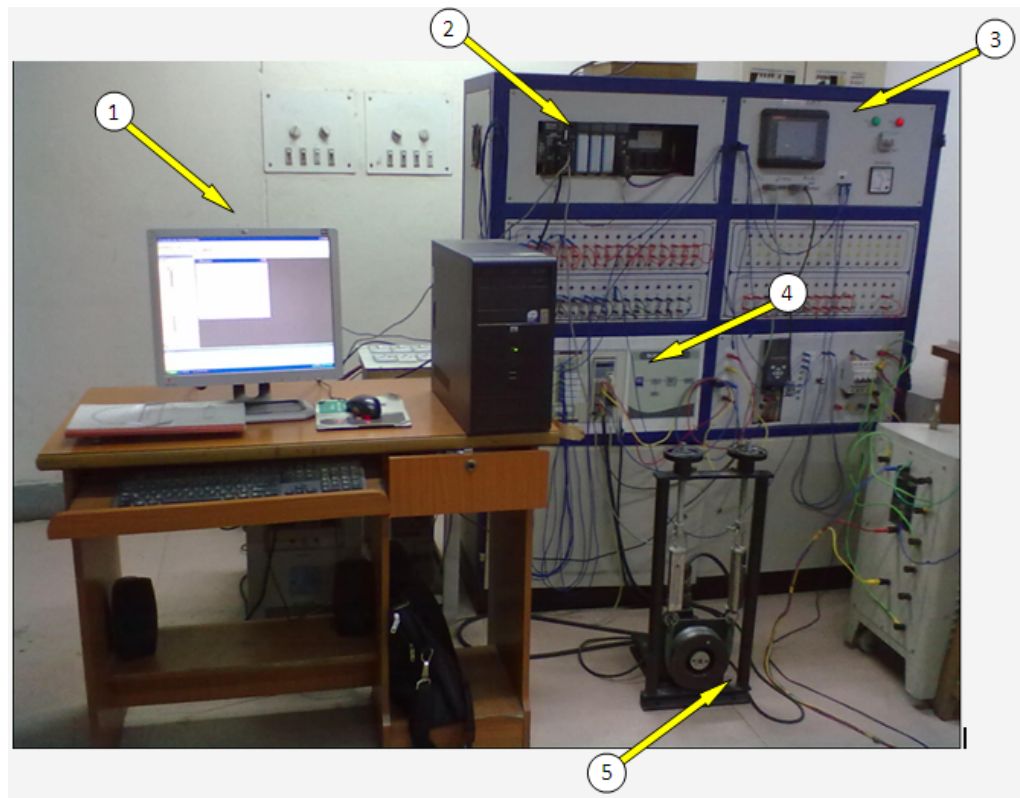


Fig 3.2 The real time control set up which was configured and tested

(1- PC, 2-PLC, 3-HMI, 4- Servodrive , 5- Servomotor(PMSM))

Fig. 3.3 gives a detailed layout of the hardware setup including the detailed interconnections, detailed drive block diagram and the important communication channels used between the components

### 3.2 PROGRAMMABLE LOGIC CONTROLLER (PLC)

A programmable logic controller (PLC) or simply programmable controller is a digital computer which accepts inputs from switches and sensors, stores instructions using programmable memory, evaluates these in accordance with a stored program, executes specific functions that include ON/OFF control, timing, counting, sequencing, etc. and generates outputs to control machines and processes involved. Fig. 3.4 shows a block diagram of the internal architecture of the PLC.

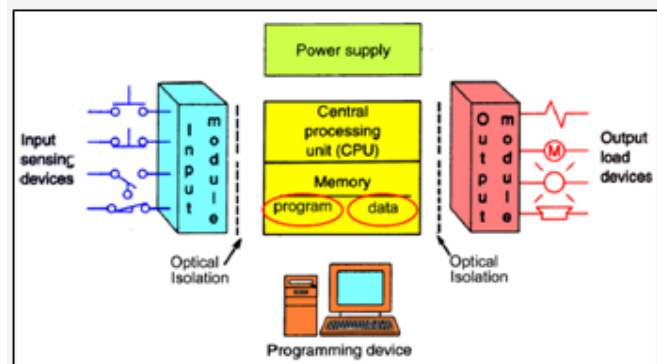


Fig.3.4 Basic block diagram of a PLC

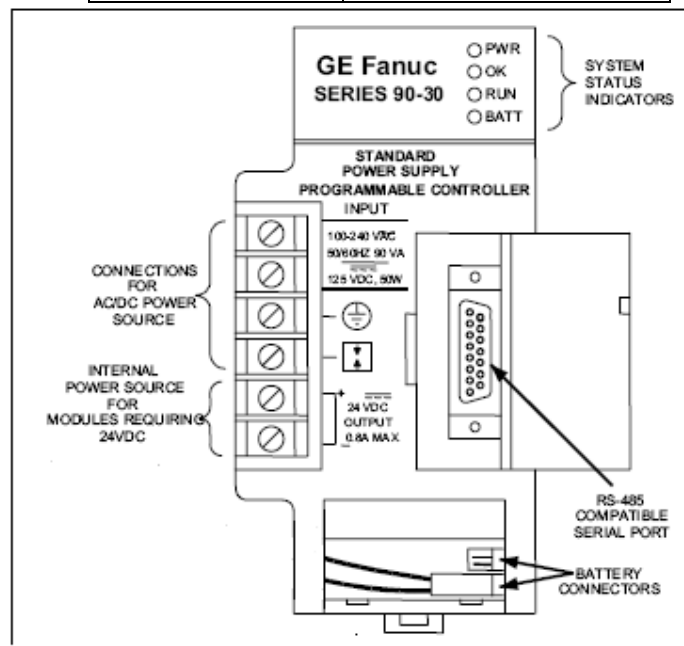
The PLC used in this project is the **GE Fanuc 90-30 series**. The basic parts of a series 90-30 PLC are Power Supply, CPU, Discrete Input Module and Discrete Output Module

#### 3.2.1 Power Supply

This can be built into the PLC or be an external unit. Series 90-30 power supplies are modular types that plug into the left slot of all 90-30 baseplates. The specifications of the power supply module used by the PLC in this Project work is listed in table 3.1 and Fig.3.5 shows a layout and the port connector diagram of the power supply [29]. Table 3.2 lists the utility of each indicator LED provided on the PLC, which basically indicates the various states of the PLC operation.

**Table 3.1 Power Supply Specifications**

<b>Catalog Number</b>	IC693PWR330
<b>Nominal Rated Voltage</b>	120/240 VAC or 125V DC
<b>Input Voltage Range</b>	85-264 VAC (100VA)
<b>(Input Power)</b>	100-300 VDC (50 W)
<b>Output Voltage</b>	5V DC (30 W Max)
<b>(Output Power)</b>	24V DC Relay ( 15W Max) 24V DC Isolated ( 20W Max)



**Fig.3.5 Power Supply Front Panel**

<b>PWR</b>	Provides an indication of the operating state of the power supply.
<b>OK</b>	It is steady <b>ON</b> if the PLC is operating properly, and <b>OFF</b> if a problem is detected by the PLC.
<b>RUN</b>	It is steady <b>ON</b> when the PLC is in the <b>RUN</b> mode
<b>BATT</b>	It will be <b>ON</b> if the memory backup battery voltage is too low to maintain the memory under a loss of power condition; otherwise it remains <b>OFF</b>

**Table 3.2 Description of the LED indicators on the PLC**

### 3.2.2 Input Module

Series 90-30 discrete input modules convert AC and DC power levels from user devices to the logic levels required by the PLC. An optical coupler provides isolation between the incoming power and the logic circuitry. The input module is designed to have either positive logic or Negative logic characteristics. Current into an input point results in a

logic 1 in the input status table (%I). Table 3.3 lists the details of the specifications of the Digital Input module of the PLC [31].

**Table 3.3 PLC Discrete Input Module Specification**

<b>Catalog Number</b>	IC693MDL645
<b>Number of Points</b>	16 ( One group with a single common)
<b>Input Voltage</b>	24 VDC
<b>Input Current</b>	7 mA (typical) at rated voltage
<b>ON state voltage</b>	11.5 to 30 Volts DC
<b>OFF State Voltage</b>	0 to 5 Volts DC
<b>ON State Current</b>	3.2 mA Min
<b>OFF State Current</b>	1.1 mA Max
<b>Response Time</b>	7 on/7 off (msec)

### 3.2.3 Output Module

Series 90-30 discrete output modules convert logic levels into AC or DC power levels required for driving user supplied devices such as motor starters, solenoids, indicators etc.. Either a power semiconductor or an electromagnetic relay switches each output point This module provides 16 output points in two groups of eight with a common power output terminal for each group. This output module is designed to have positive logic characteristics in that it sources current to the loads from the user common or positive power bus. Power to operate the field devices must be supplied by the user. ). Table 3.4 lists the details of the specifications of the Digital Output module of the PLC [31].

**Table 3.4 PLC Discrete Output Module Specification**

<b>Catalog Number</b>	IC693MDL740
<b>Number of Points</b>	16 ( Two groups of 8 outputs each)
<b>Output Voltage Range</b>	12-24 VDC
<b>Output Current</b>	0.5 A Max. Per point & 2A Max. Per common
<b>Output Voltage Drop</b>	1 V Max.
<b>OFF state Leakage</b>	1 mA Max.
<b>Response Time</b>	2 ON/ 2 OFF (msec)

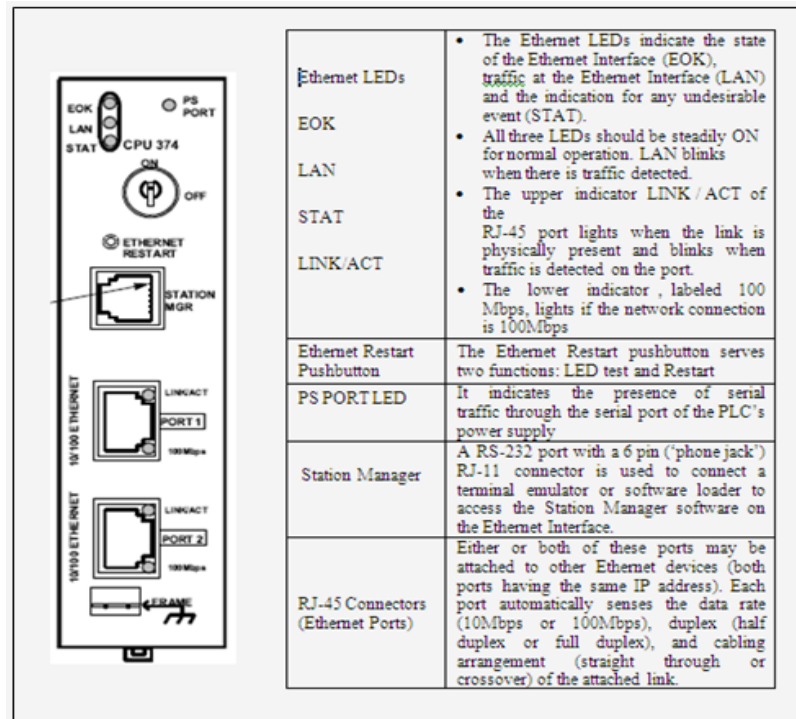
### 3.2.4 Central Processing Unit

The Series 90-30 PLC uses a Modular CPU. The modular CPUs consist of a CPU, memory, and associated integrated circuit chips soldered to circuit board which is

mounted in a plug-in module. Table 3.5 shows the detailed CPU specifications of Series 90-30 PLC and Fig.3.5 shows the front panel details of CPU374 [29]

**Table 3.5 CPU (IC693CPU374) Specifications**

<b>Processor Speed</b>	133 MHz
<b>Processor Type</b>	Embedded 586
<b>Execution time per 1K Boolean contacts/coils</b>	0.15 milliseconds
<b>Type of memory storage</b>	RAM and Flash
<b>User Memory (Total)</b>	240KB (245,760) Bytes
<b>Discrete Input Points - %I</b>	2,048 (fixed)
<b>Discrete Output Points - %Q</b>	2,048 (fixed)
<b>Internal Coils - %M</b>	4,096 bits (fixed)
<b>Register Memory - %R</b>	Configurable 128 to 32,640 words
<b>Timers/Counters</b>	>2,000 (depends on available user memory)
<b>Load Required from Power Supply</b>	7.4 watts of 5VDC
<b>Built-in Serial Ports</b>	None. Supports RS -485 port on power supply.
<b>Protocol Support</b>	SNP and SNPX on power supply RS -485 port
<b>Built-in Ethernet Communications</b>	Ethernet (built-in) – 10/100 base-T/TX Ethernet Switch





**Fig. 3.6 Front Panel details of CPU374**

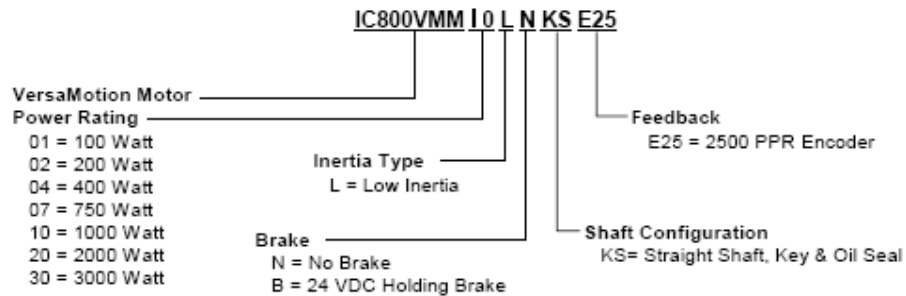
### **3.3 THREE PHASE PERMANENT MAGNET SYNCHRONOUS MOTOR (PMSM) USED AS SERVOMOTOR**

The experimental setup used in this project work consists of a three phase 1kW, AC servomotor, Permanent Magnet Synchronous Motor(PMSM), with an internally fitted encoder and amplifier. The servomotor is not fed directly, rather it is fed through a PWM based Versamotion Servo drive, which provides high frequency control voltage to motor stator windings. A 2500 pulses per revolution (ppr) or 10,000 Quadrature counts/revolution, incremental encoder is used for position feedback. This is equivalent to a resolution of 0.036 degrees at the motor shaft. Table 3.6 shows the specifications of the Three Phase Servomotor used for the present application [28].

**Table 3.6 AC Servomotor Specifications**

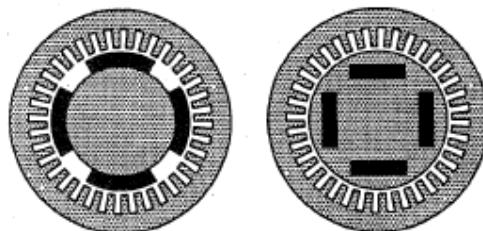
Part No.	IC800VMM10LNKSE25A (No brake)
Rated Output (KW)	1.0
Rated Current(A)	7.3
Maximum Current ( A)	21.9
Rated Torque(Nm)	3.18
Peak Torque (Nm)	9.54
Rated Speed (RPM)	3000
Maximum Speed (RPM)	5000
Rotor Inertia ( MI) ( $\text{kg}\cdot\text{m}^2 \times 10^{-4}$ )	2.65
Armature Resistance (Ohm)	0.20
Armature inductance(mH)	1.81
Torque constant-KT ( N.m/A)	0.43
Electrical time constant (ms)	9.3
Voltage constant-KE (mV/rmp)	16.8
Operating Temperature	0 to 40 <sup>0</sup> C
Motor Body Length(mm)	154
Weight (Kg)	4.3
Shaft Diameter (mm)	22

A-----Insulation class



VersaMotion is a family of cost effective AC servo motors and amplifiers ranging in power rating from 100 W to 3 kW and a continuous torque range from 0.3 Nm to 19.1 Nm (2.7 to 169 in-lb). Motors and amplifiers must be used as matched sets based on the power rating. A 100 Watt motor must be matched with a 100 Watt amplifier. The motors support a peak (intermittent) torque rating of 3 times the continuous rating which is sufficient for most application requirements. The wide range of torque and motor power provided by VersaMotion allows us to optimize performance, size and cost for our application [28].

PMSMs are favoured for high speed Servo applications because of their high efficiency, power density, high torque-to-inertia ratio which makes them a suitable choice for variable speed direct-drive operations. The permanent magnet AC servo motor has a rugged construction. The stator of AC servo motor consists of stacked steel laminations with winding placed in the slots that are axially cut along the inner periphery. The stator has three symmetrical windings, which are internally connected in a wye configuration. The rotor contains the permanent magnets. Rotors for sinusoidal PMAC machines can be designed using either surface-mounted or buried (interior) magnet configurations [25], as shown in Fig.3.6. The rotor of the servo motor is very long with a small radius. This enables high dynamics of the motor with good precision and high torque.



**Fig.3.7 Cross-sectional views of two major types of PMAC motor construction.**  
 (a) Surface-mounted magnets. (b) Buried (interior) magnets.  
 Solid black areas identify magnet positions.

### 3.3.1 Mathematical Model of the PMSM

Permanent Magnet Synchronous Motors are usually modeled in the d-q reference frame fixed to the rotor. In this synchronous rotating reference frame, the d-axis is oriented along the permanent magnet flux, whose angle in the stator reference frame is  $\theta_m$  in electrical radians.

The following assumptions are made in the derivation:

- 1) Saturation is neglected although it can be taken into account by parameter changes;
- 2) The induced EMF is sinusoidal;
- 3) Eddy currents and hysteresis losses are negligible;
- 4) There are no field current dynamics;
- 5) There is no cage on the rotor.

With these assumptions, the stator  $d, q$  equations of the PMSM in the rotor reference frame are [20]

$$v_d = R i_d + \frac{d\lambda_d}{dt} - \omega_e \lambda_q \quad (1)$$

$$v_q = R i_q + \frac{d\lambda_q}{dt} + \omega_e \lambda_d \quad (2)$$

where  $\lambda_d = L_d i_d + \lambda_{pm}$  (3)

and  $\lambda_q = L_q i_q$  (4)

$v_d$  and  $v_q$  are the  $d, q$  axis voltages,  $i_d$  and  $i_q$  are the  $d, q$  axis stator currents,  $L_d$  and  $L_q$  are the  $d, q$  axis inductances, and  $\lambda_d$  and  $\lambda_q$  are the  $d, q$  axis stator flux linkages, while  $R$  and  $\omega_e$  are the stator resistance and inverter frequency respectively,  $\lambda_{pm}$  is the flux linkage due to the rotor magnets linking the stator and  $p$  is the derivative operator

The electric torque is

$$T_e = 3P [\lambda_{pm} i_q - \lambda_q i_d] / 2 \quad (5)$$

and the equation for the motor dynamics is

$$J \frac{d\omega_r}{dt} = T_e - T_L - B \omega_r \quad (6)$$

$P$  is the number of pole pairs,  $T_L$  is the load torque,  $B$  is the damping coefficient,  $\omega_r$  is the rotor speed and  $J$  the moment of inertia. The inverter frequency is related to the rotor speed as

$$\omega_e = p \omega_r \quad (7)$$

$\psi_d, \psi_q, i_d,$  and  $i_q$  are state variables. For dynamic simulation, the equations of the PMSM presented in (1) to (6) must be expressed in state space form as

$$\dot{\psi}_d = (\omega_r \psi_q - \omega_r \psi_d + \omega_r \psi_q) / \sigma \quad (8)$$

$$\dot{\psi}_q = (\omega_r \psi_d + \omega_r \psi_q - \omega_r \psi_d) / \sigma \quad (9)$$

$$\dot{i}_d = (\omega_r \psi_q - \omega_r \psi_d + \omega_r \psi_q) / J \quad (10)$$

For a balanced system the total power input to the machine in terms of d-q variables is

$$Power = 3 (\omega_r \psi_d i_d + \omega_r \psi_q i_q) / 2 \quad (11)$$

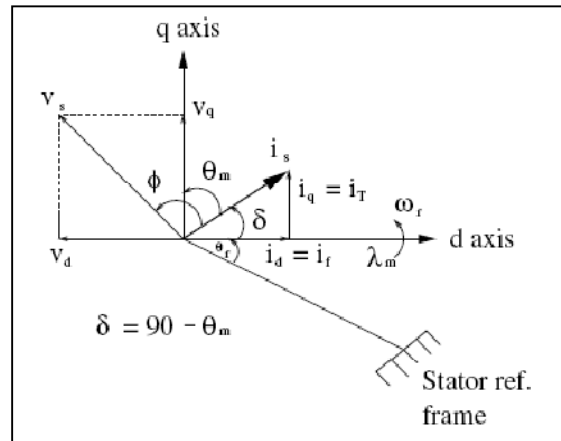
### 3.3.2 Vector control Principle

The Vector control Algorithm of the Permanent Magnet Synchronous Motor is the theoretical foundation of Electric Drive. This control technique basically transforms a PMSM into an equivalent separately excited DC motor. Separately-excited DC drives are simpler and have good dynamic response because they independently control flux, which, when maintained constant, contributes to an independent control of torque. This is made possible with separate control of field and armature currents which in turn control the field flux and the torque independently. Thus in Vector control, also called Field-oriented control, the field current (exciting current) and torque current are decoupled, so the independent flux and torque controls are possible as in DC motor.

A vector control refers to the magnitude and phase of the controlled variables. Matrix and vectors are used to represent the control quantities (voltages, currents, flux etc). This method takes into consideration the mathematical equations that describe the motor dynamics. This approach needs more calculations than a standard control scheme. It can be solved by the use of a calculation unit included in specialized digital signal processor (DSP).

Fig.3.8 shows a vector diagram of the PMSM. Phase  $a$  is assumed to be the reference. The instantaneous position of the rotor (and hence rotor flux) is at  $\theta_r$  angle from phase  $a$ . The application of vector control, so as to make it similar to a DC machine, demands that the quadrature axis current  $i_q$  be in quadrature to the rotor flux. Consequently  $i_d$  has to be along the rotor flux since in the reference used  $i_d$  lags  $i_q$  by 90 degree. If  $i_d$  is in the same direction as the rotor flux, the d-axis stator flux adds to the rotor flux that leads to

increase in the net air gap flux. On the other hand if  $i_d$  is negative then the stator d-axis flux is in opposite to that of the rotor flux resulting in a decrease in air gap flux. The PMSMs are designed such that the rotor magnet alone is capable of producing the required air gap flux up to the rated speed. Hence  $i_d$  is normally zero in the constant torque mode of operation [11].



**Fig.3.8 Phasor Diagram of PMSM**

From phasor diagram we get

$$\begin{bmatrix} i_q \\ i_d \end{bmatrix} = i_s \begin{bmatrix} \sin \delta \\ \cos \delta \end{bmatrix}$$

$i_q$  = Torque-producing component of stator current =  $i_T$

$i_d$  = Flux-producing component of stator current =  $i_f$

The electric torque equation (5) becomes :

$$T_e = K i_q = K i_T \quad (12)$$

where,

$$K = 3P \quad (13)$$

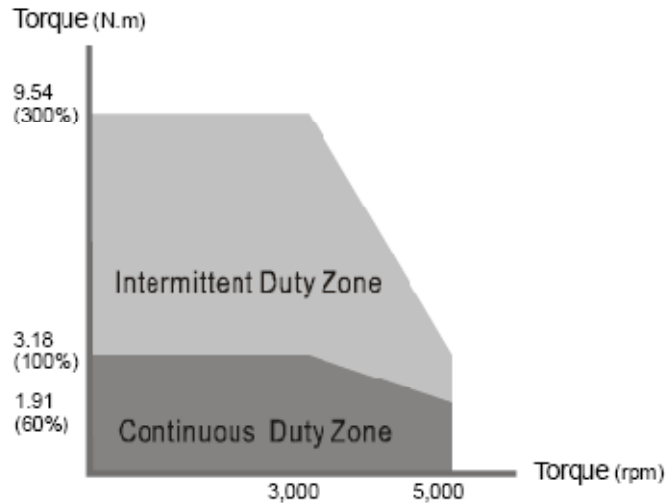
Hence the electric torque depends only on the quadrature axis current and a constant torque is obtainable by ensuring that  $i_q$  is constant.

Hence with the application of vector control, independent control of the torque ( $i_q$ ) and flux ( $i_d$ ) producing currents are possible. The equations for controller design then are (8), (9), (10) and (12). These equations basically describe a voltage source inverter fed PMSM. The technological development of high switching frequency, high current rating transistors has realized high bandwidth current sources that have lead to improved servo performance in terms of torque and speed bandwidths.

### 3.3.3 Speed-Torque Characteristics of PMSM motor

Rated Torque is the maximum torque that the machine can produce continuously at rated speed and lower, and is limited by motor heating. This rated torque is given the value of 100% torque. Likewise, rated speed is the highest speed at which the rated torque is available. The motor can continuously run faster than rated speed, but the torque available drops significantly the faster the motor runs. Fig.3.9 gives the exact Speed-Torque characteristics of the 1KW PMSM motor used in this Project. The rated speed of this motor is 3000 rpm and the maximum attainable speed is 5000 rpm. Likewise, the rated torque of this motor is 3.18Nm and the peak torque is 9.54 Nm.

The speed-torque curve has two regions – Continuous and Intermittent. If the combination of torque and speed required by the motor falls in the continuous region, the motor can produce that torque and speed continuously without overheating the motor. But if a torque and speed combination falls in the intermittent region, the motor can produce that torque and speed only for a limited amount of time, which if exceeded can overheat the motor. To prevent the damage due to overheating, modern amplifiers automatically disable the motor and enter an alarm state if the time limit is exceeded. In this case the drive displays an alarm **ALE 06** for **Overload** (ie. when the servomotor and the drive is overloaded) and **ALE 07** for **Overspeed** ( ie. when the motor speed exceeds the normal speed of operation). For applications when short bursts of high torque are required such as during acceleration or deceleration, the motor can be run in the intermittent region



**Fig.3.9 Speed-Torque Characteristics of the 1KW PMSM Motor Used in this Project**

### **3.4 THREE PHASE VOLTAGE SOURCE INVERTER FED PWM BASED AC SERVO DRIVE**

The Servo system works using feedback loops. The Servo system used in this setup employs a PWM based Servo drive. The drive is equipped with **serial communication** and a display screen to set and read the parameters of the drive. The specifications of the AC Servodrive are shown in table Table 3.7 [29].

**Table 3.7 AC Servodrive specifications**

Part No.	IC800VMA102
Power Rating (KW)	1.0
Power Supply	
Ø Phase / Voltage	Three Phase or Single Phase, 220VAC
Ø Voltage Range	Three Phase: 170-255VAC
Ø Frequency Range	50/60 Hz $\pm$ 5%
Encoder Resolution	2500 ppr Encoder(10000 counts/rev)
Control of Main Circuit	SVPWM control
Tuning modes	Auto/Manual
Communication Interface	RS-232/RS-422/RS-485

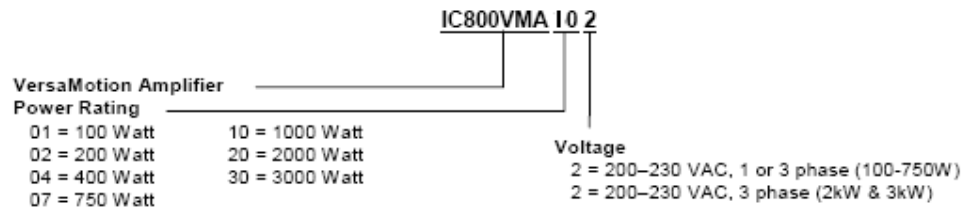
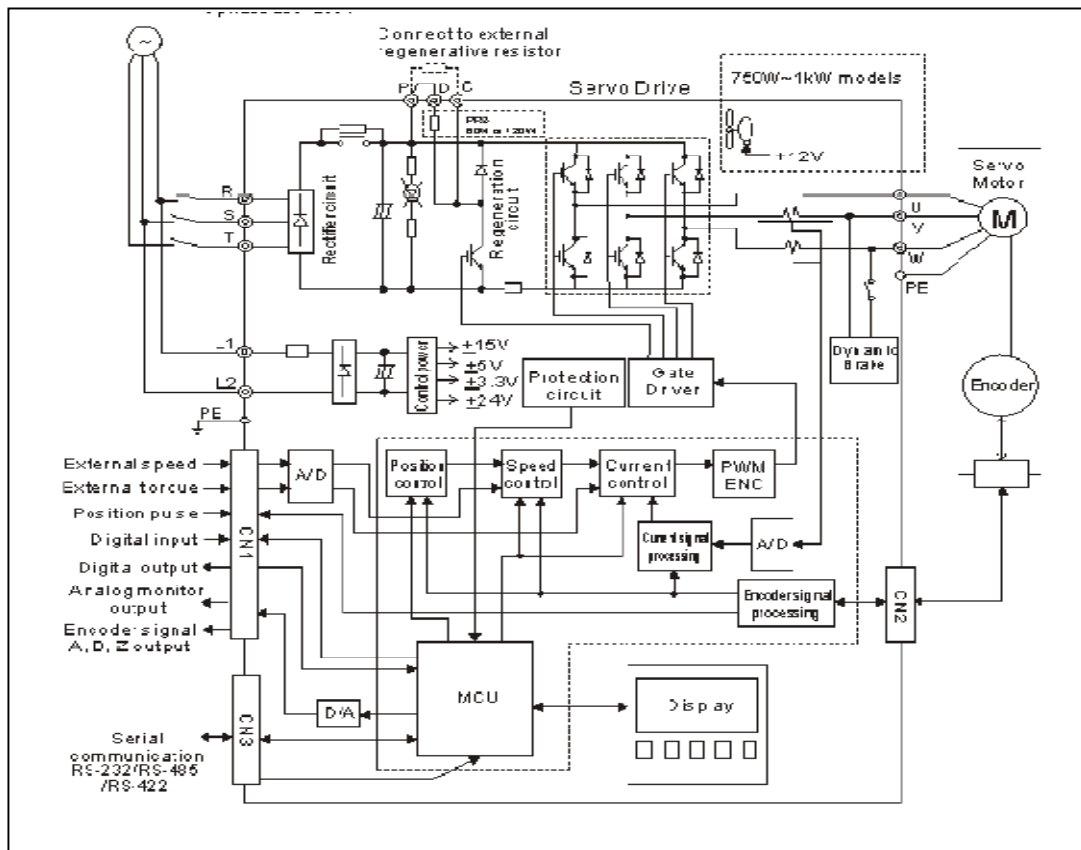


Fig. 3.10 shows a detailed internal block diagram of the Servodrive [29]. Three phase AC power is fed to AC servomotor through a **Voltage Source Inverter** fed servo-drive. The **rectifier circuit** converts it into DC voltage which is then fed to **three phase inverter** through a **DC link** (choke coil) and **regenerative resistor**. The DC link eliminates ripples from the converted voltage and regenerative resistor is meant for dissipation of regenerative energy produced during braking. The drive has a protection circuit which protects motor against undue overloading, faults, etc. The closed loop control circuits for control of position, speed and current (torque) are also present within the drive. These circuits convert the acquired data to suitable pulses (position mode), suitable current (torque mode) or an adjustable voltage (speed mode) as required for controlling the motor operation. The pulses to the gate circuitry of IGBTs are controlled by these circuits and in turn AC servomotor is controlled. The drive has a display screen to view and change various parameters as required.

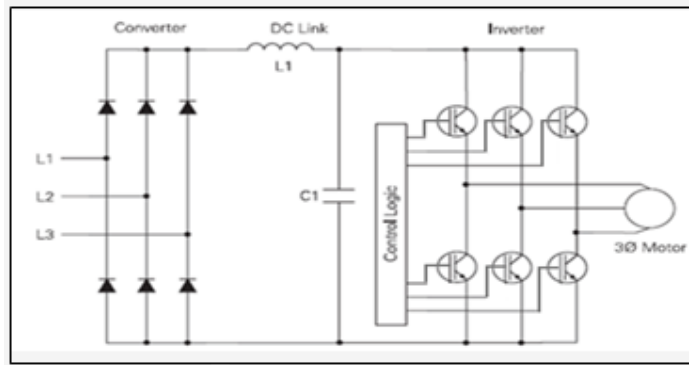
The three common inverter types are: variable voltage inverter (VVI), current source inverter (CSI), and pulse width modulation (PWM). Almost all AC drives convert AC to DC, and then through various switching techniques invert the DC into a variable voltage or variable frequency output. The technique used in this drive is pulse width modulation (PWM) technique.





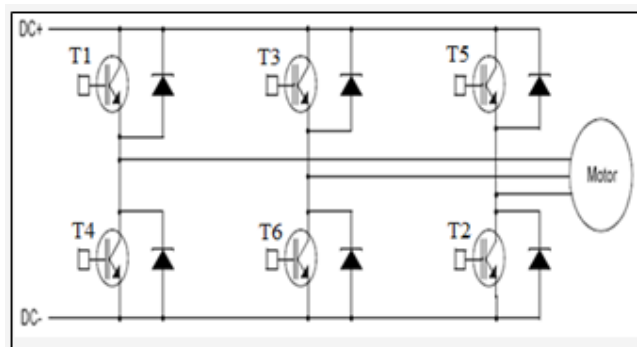
**Fig.3.10 Internal architecture of PWM based servo -drive.**

Output voltage and frequency applied to motor are controlled by control logic and inverter section of PWM based drive. The inverter section consists of six switching devices connected in a proper sequence as shown in the Fig. Various switching devices used for this purpose include thyristors, bipolar transistors, MOSFETs and IGBTs. In the present case IGBTs are used as switching devices. A voltage source power inverter is used to convert DC to required AC voltage and frequency. Fig. 3.11 shows that the power inverter has six switches that are controlled in such a way so as to generate an AC output from the DC input. The control logic generated by microcontroller unit controls the gate circuitry of these six switches. The phase voltage is determined by the duty cycle of the PWM signals. At a time, maximum of three switches will be ON, either one upper and two lower switches, or two upper and one lower switch. When switches are ON, current begins to flow into the motor stator windings. Since motor stator windings are highly inductive in nature, they hold electric energy in the form of current. This current needs to be dissipated while switches are OFF.



**Fig.3.11 PWM Scheme for AC Servodrive**

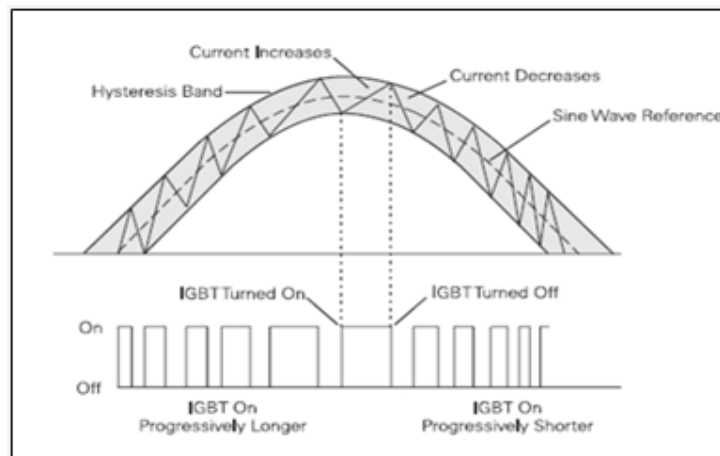
Diodes connected across the switches give a path for the current to dissipate when the switches are OFF. These diodes are also called freewheeling diodes as shown in Fig.3.12. Upper and lower switches of the same limb should not be switched ON at the same time. This will prevent the DC bus supply from being short-circuited. A dead time is given between switching OFF the upper switch and switching ON the lower switch and vice versa in order to ensure that both switches are not conductive when they change states from ON to OFF, or vice versa.



**Fig.3.12 Three Phase Inverter Bridge**

IGBTs are capable of switching ON and OFF several thousand times per second. When a positive voltage (typically +15 VDC) is applied to the gate, IGBT turns ON. This is similar to closing of a switch. Current flows between the collector and emitter. The device is turned OFF as soon as the positive voltage is removed from the gate. During the OFF state IGBT gate voltage is normally held at a small negative voltage (-15 VDC) to prevent the device from turning ON. An IGBT or any other type of switching device can be switched ON by connecting motor to the positive DC voltage and soon current starts flowing from converter into the motor. Fig. 3.13 shows hysteresis band current PWM control of three phase motor.

Initially IGBT is switched ON for a short period of time, allowing only a small amount of current to build up in the motor and then it is switched OFF. Later, IGBT is switched ON and left ON for progressively longer periods of time; allowing current to build up to higher levels until current in the motor reaches a peak value. IGBT is then again switched ON for progressively shorter periods of time, which results in decreased current build up in the motor. The negative half of the sine wave is generated by switching an IGBT connected to the negative terminal of the converted DC voltage and then switching it ON and OFF similarly as done in the previous case i.e., in case of IGBT connected to positive terminal.



**Fig.3.13 PWM control of a three phase motor**

The Versamotion Servo Drive used for Servo Motor control in this Project is based on a similar PWM principle and there is a reduction in torque pulsations and low speed motor cogging, since more sinusoidal current output is produced by the PWM.

Fig. 3.14 gives the front panel details of the Versamotion Servo Drive [29]. The connectors . CN2 and CN3 can be clearly seen which are important in connecting the drive to the PLC, the Encoder and the PC or HMI respectively. Also the Three Phase supply input is given through terminals R, S and T and the Servo Motor is connected through U,V and W terminals. The Servo Drive can be directly operated through the keypad present on the front panel.

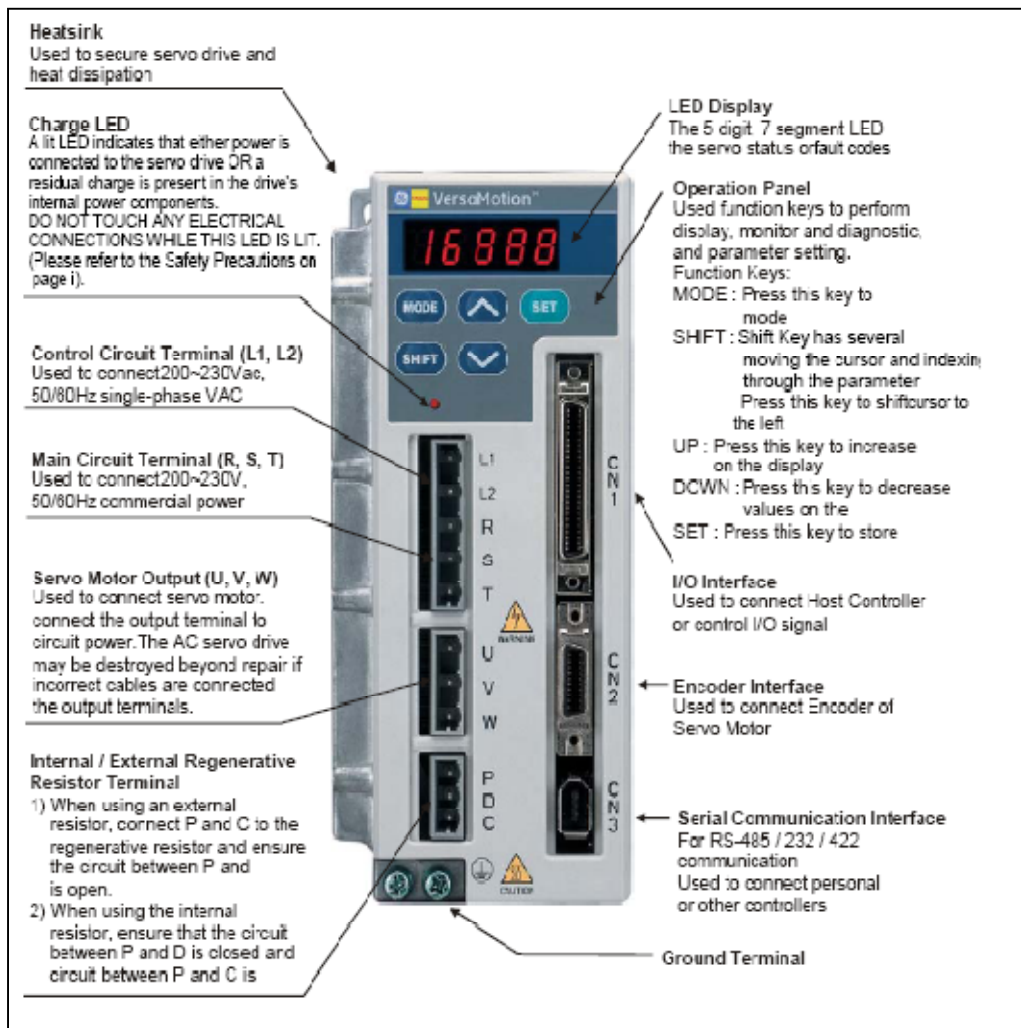


Fig.3.14 Versamotion Servo drive front panel

### 3.5 SCADA (Supervisory Control and Data Acquisition) AS THE MASTER STATION FOR CONTROL

SCADA is the abbreviation for Supervisory Control And Data Acquisition. It generally refers to an industrial control system: a computer system monitoring and controlling a process. The process can be industrial, infrastructure or facility based as described below:

- Ø **Industrial** processes include those of manufacturing, production, power generation, fabrication, and refining, and may run in continuous, batch, repetitive, or discrete modes.

- Ø **Infrastructure** processes may be public or private, and include water treatment and distribution, wastewater collection and treatment, oil and gas pipelines, electrical power transmission and distribution, and large communication systems.
- Ø **Facility** processes occur both in public facilities and private ones, including buildings, airports, ships, and space stations. They monitor and control HVAC, access, and energy consumption.

In short, SCADA can be defined as a system that gets data about a system under supervision in order to control that system. A typical SCADA System usually consists of the following subsystems:

- Ø A **Human-Machine Interface** or HMI is the apparatus which presents process data to a human operator, and through which the human operator monitors and controls the process.
- Ø A **supervisory (computer) system**, gathering (acquiring) data on the process and sending commands (control) to the process
- Ø **Remote Terminal Units** (RTUs) connecting to sensors in the process, converting sensor signals to digital data and sending digital data to the supervisory system.
- Ø **Communication** infrastructure connecting the supervisory system to the Remote Terminals Units

Data acquisition begins at the RTU or PLC level and includes meter readings and equipment status reports that are communicated to SCADA as required. The communication system, which can be wire, fibre optic, radio, telephone line, microwave or even satellite, provides the pathway for communication between the master station and the remote sites. Specific protocols and error detection procedures are used for optimum and efficient data transfer. Data is then compiled and formatted by the Master station in such a way that a control room operator using the HMI can make supervisory decisions to adjust or override normal RTU (PLC) controls.

SCADA systems typically implement a distributed database, commonly referred to as a **tag database**, which contains data elements called **tags** or **points**. A point represents a single input or output value monitored or controlled by the system. Points can be either "hard" or "soft". A hard point represents an actual input or output within the system, while a soft point results from logic and math operations applied to other points. Points are normally stored as value-timestamp pairs: a value and the timestamp when it was recorded or calculated. A series of value-timestamp pairs gives the history of that point.

It's also common to store additional metadata with tags, such as the path to a field device or PLC register, design time comments, and alarm information.

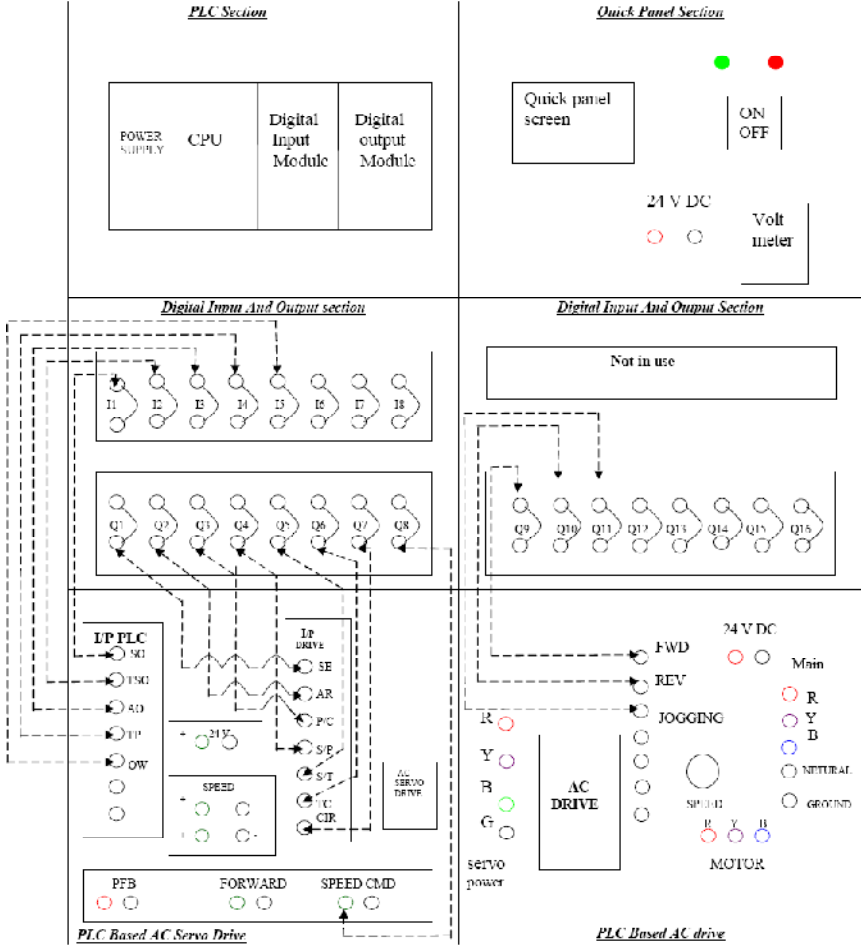
A properly designed SCADA system reduces labor costs by minimizing site visits for inspection, data collection/logging and making adjustments. SCADA provides supervision in form of latest graphical animations both virtual and real time as well as data trends of all other processes on the window of personal computer or the screen of HMI. User defined controls like (start, stop, speed set) can also be developed on the software window in order to control the system remotely. Using SCADA software makes monitoring of numerous control processes dynamic and easy. Typical legacy SCADA protocols include Modbus RTU, RP-570, Profibus and Conitel. These communication protocols are all SCADA-vendor specific but are widely adopted and used. Standard protocols are IEC 60870-5-101 or 104, IEC 61850 and DNP3. These communication protocols are standardized and recognized by all major SCADA vendors. Many of these protocols now contain extensions to operate over TCP/IP. It is good security engineering practice to avoid connecting SCADA systems to the Internet so the attack surface is reduced.

In this project AC Servomotor will also be controlled through SCADA software namely GE FANUC's Proficy HMI/SCADA Cimplicity 7.5. This software proves to be excellent software for development of animation windows for easy control of the system remotely and it also provides provision for development of SCADA scheme for the system involved.

### **3.6 CONNECTION DIAGRAM OF THE PLC WITH THE VERSAMOTION THREE PHASE SERVO DRIVE**

Fig. 3.14 gives the connection diagram of the PLC with the Digital Inputs and Digital output terminals of Versamotion Drive for this project. Jumper wires are used to connect the terminals. The Digital outputs from the Drive serve as the input to the PLC (I1,I2,I3,I4 and I5) and the output terminals of the PLC serve as the Digital inputs to the Servodrive (Q1 to Q8). Q9 to Q11 are reserved for the Induction Motor drive . The PLC section is given an external supply. The Drive section is fed from the Three Phase AC

supply through an electromagnetic contactor. The servomotor is connected via the CN2 terminal of the drive and the PC is connected via the CN3 connector for serial communication with the drive.



**Fig.3.15 Connection Diagram of PLC with AC Servodrive**

### **3.7 CONCLUSION**

This chapter described the complete hardware set-up used in the present project and the also the functions of all individual components. PLC is the main control and decision making device which directs the AC Servodrive as per the decisions or rules defined in the ladder logic of the PLC. Supervisory control is being exercised through PLC by monitoring the status of the drive through the available built in communication bus. HMI is a human operator interface for local control and/or monitoring of drive status through a touch button screen.



## CHAPTER IV

### OVERVIEW OF THE SYSTEM SOFTWARE

---

#### 4.1 GENERAL

A PLC and SCADA based monitoring and control systems for Permanent Magnet Synchronous motor have been successfully implemented and commissioned in the laboratory. The drive system provides precise position and speed control and also enables an online monitoring of drive speed, torque and current through SCADA system. The following software has been used in the control and monitoring of the Permanent Magnet Synchronous Motor.

- a) **Proficy Machine Edition 5.90** : To control the Servomotor directly through the ladder programs written for the **PLC- Servo drive - Servomotor** interface [28],[31]
- b) **VersaMotion Configuration Software**: To directly control the Servomotor through the VersaMotion Servo drive settings and monitor the Servomotor parameters. The Servo drive settings can be changed directly through the software installed on the PC or through a digital keypad provided on the Drive itself. [29]
- c) **Proficy HMI/SCADA Cimplicity 7.5 Software** : To control the Servomotor remotely through a SCADA screen developed using the Proficy HMI/SCADA Cimplicity 7.5 software. [34],[35]

#### 4.2 PROFICY MACHINE EDITION 5.90

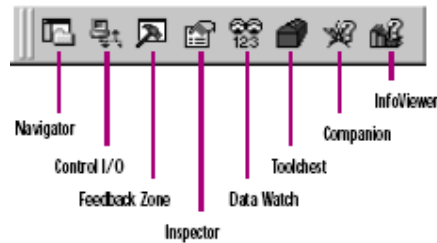
The Proficy Machine Edition 5.90 package provides all the tools necessary to create powerful control and HMI (human machine interface) applications for a variety of run-time targets. Machine Edition supports various products out of which the following products have been used in this Project:

- **Logic Developer – PLC**, a software tool for programming the full line of GE Fanuc Controllers. This software package provides all the tools necessary to create powerful control Applications by providing the means to configure the Controller hardware or remote I/O, to create and edit logic, upload and download projects, and monitor and debug the execution of control programs.

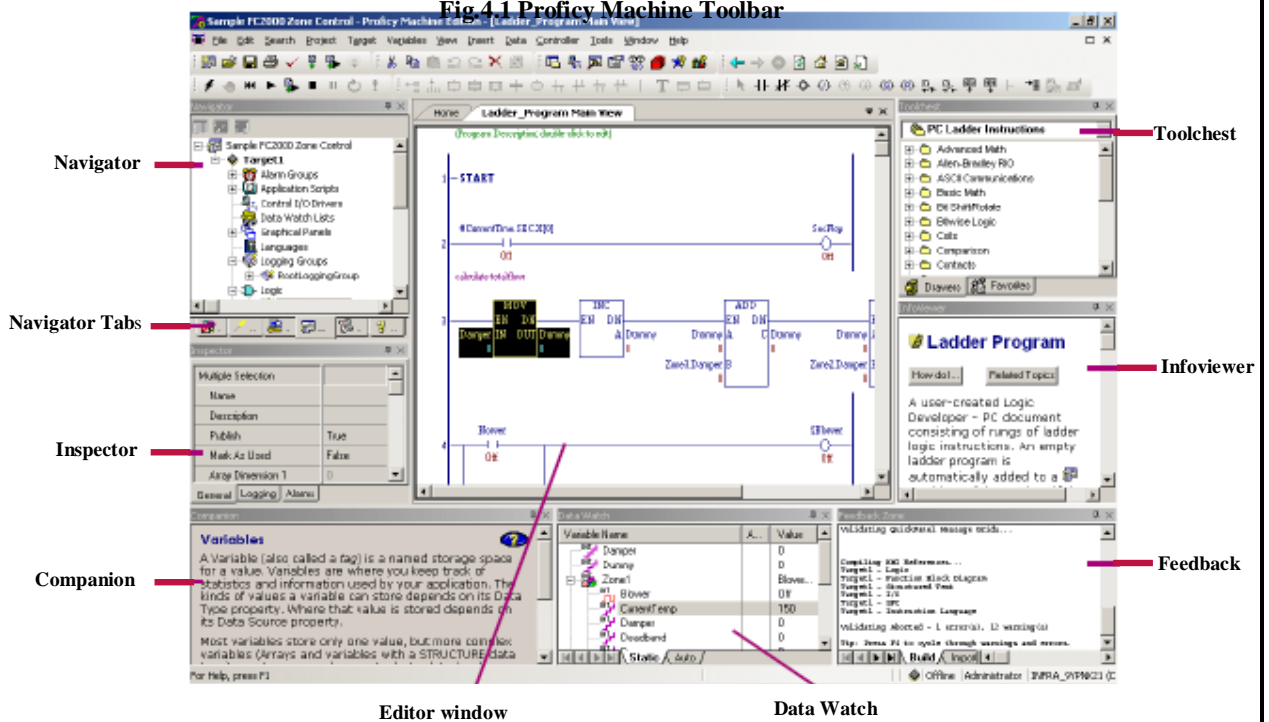
- **View**, for creating HMIs for Windows NT/2000/XP and Quick Panel units. View is the human-machine interface (HMI) component of Machine Edition, providing the means to create graphical panels, write scripts, configure alarm etc. in order to organize, track, display, and affect data from a plant or process.

#### 4.2.1 The Proficy Machine Edition Environment

Proficy Machine edition provides a fully-integrated environment because all the products share the same project database and the same set of tools thereby wasting no time in synchronizing data points between applications and also providing a consistent user interface throughout the development process. The toolbar is shown in Fig. 4.1



**Fig 4.1 Proficy Machine Toolbar**



**Fig.4.2 Proficy Machine windows**

The toolbar leads us to the following windows in Proficy machine as shown in Fig. 4.2

**Navigator:** Organizes and displays project information in a tree structure. It also adds targets and components to a project, creates scripts and variables, and opens editor and more.

**Companion:** A dynamic help window which displays brief information about whatever item we are working with, anywhere in Machine Edition like the meaning of a property, the purpose of a node in the Navigator, or an explanation of errors generated during validation etc.

**InfoViewer :** An embedded Web browser that provides more detailed and procedural help. It is launched whenever you access help topics from the InfoView Tab, the help index, or by pressing F1 on a selected item.

**Inspector :** Displays the properties of a selected object in Machine edition and offers a platform for the editing of these properties

**Data Watch:** The Data Watch (available only while editing a project) is a debugging tool that displays the current status of our Project's variables. It can also be used to monitor and edit real-time values of the Variables. Watch lists can be imported, exported, or saved with a project.

**Toolchest :** Contains preconfigured objects that you can drag into Machine Edition projects like a ladder logic instruction.

**Feedback Zone:** The Feedback Zone (available only while editing a project) is an interactive window that displays output information generated by Machine Edition components. With the Feedback Zone, you can keep track of project information, locate validity errors within your project, display generated reports, and more.

#### **4.2.2 General Concepts about Machine Edition Projects**

During development, our automation application is called a *project*. Each Machine Edition project is made up of *targets* and (sometimes) *components*. A target represents the hardware platform where the finished project runs, such as the PLC, a QuickPanel Unit etc. Targets further comprise of nodes which represent the different components associated with a project. Components add specific capabilities to a target.

The various components in a project may include: HMI (with View) and/or Logic (with Logic Developer - PLC). Note that View Stations do not support Logic components. For example, with View, we can add an HMI component to Windows NT/2000/XP and

Windows CE targets, used for creating human-machine interfaces. With Logic Developer - PLC, we can add a Logic component to Windows NT/2000/XP and Windows CE targets. The Control I/O Drivers, Ladder\_Program, Logic and SFC folders appear when a Logic component is added to Target1 (when using Logic Developer - PLC).A project can have multiple targets of different types with various components running on each target.

#### 4.2.3 Development of a PLC program in Proficy Machine Edition 5.9

1. Start Machine Edition from the Start menu | **Programs** | **GE Fanuc** | **Proficy Machine Edition** OR directly double click on the shortcut to Proficy Machine Edition on the desktop.
2. Create a project using a template. In the Manager tab of the Navigator, right-click **My Computer** and choose **New**. The New Project dialog box appears (Fig 4.3)

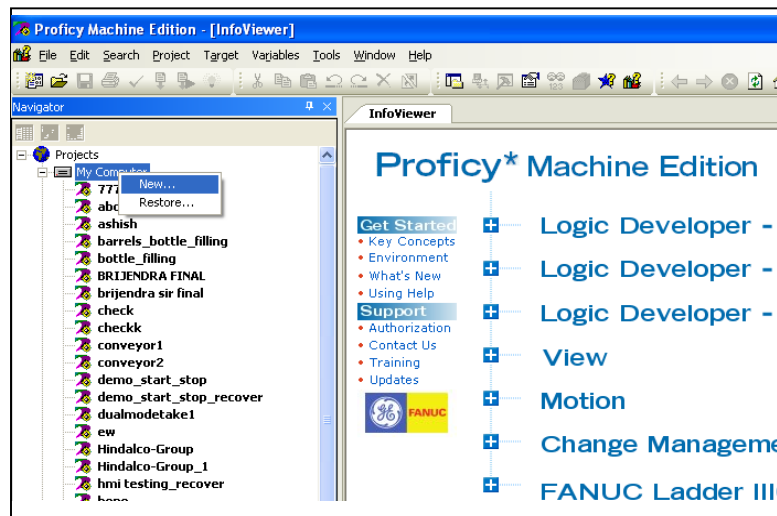


Fig 4.3 New Project Start up window of Proficy machine

3. In the New Project dialog box, enter a name for the new project (say NEWPROJ), select project template as GE Fanuc Series 90-30 PLC, and then click **OK** (Fig. 4.4)

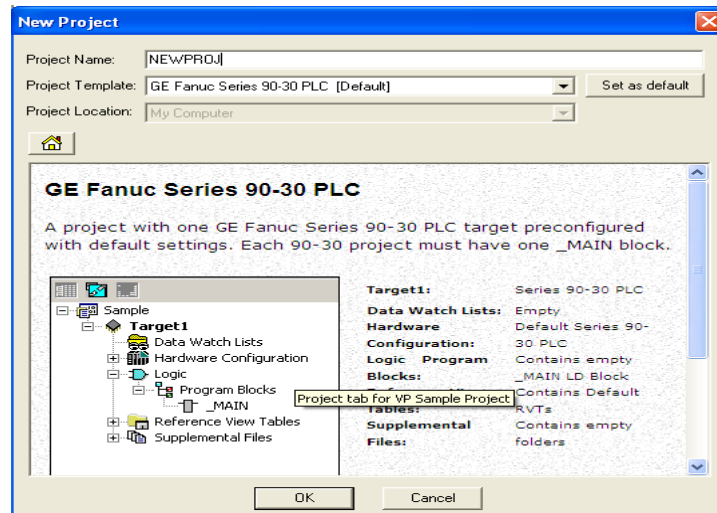


Fig 4.4 Naming the new project

- In the Navigator window, right click on the NEWPROJ. Select **Add target** and then in the GE Fanuc Controller option, select **Series 90-30 PLC**. A new target appears in the Project tab of the Navigator. When expanded, the new target contains a default Hardware Configuration, logic blocks, and miscellaneous components.(Fig. 4.5)

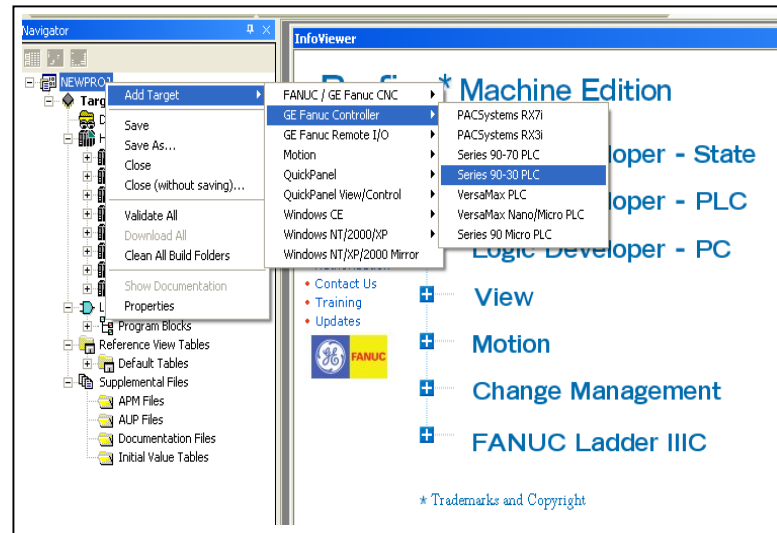
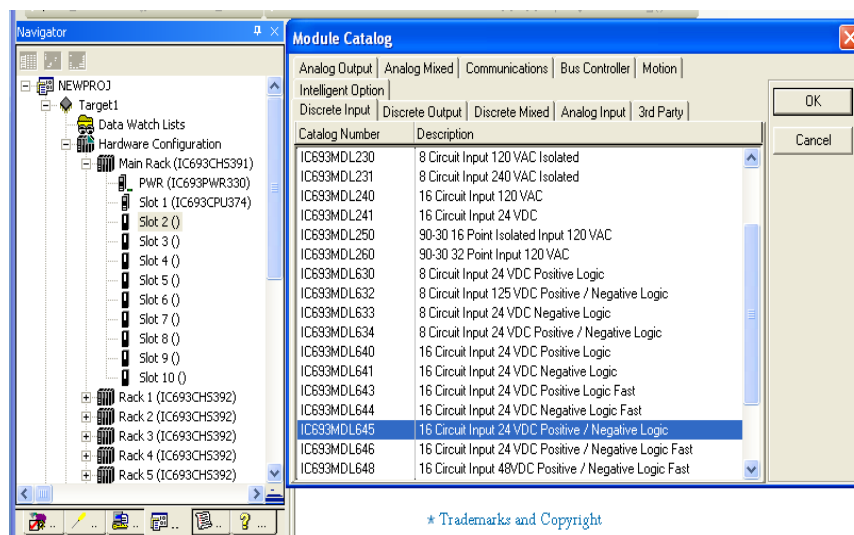


Fig. 4.5 Addition of GE-Fanuc PLC as the target

- In the Navigator window right click on the Target and click on **Properties**. This opens the properties of the Target in the inspector window. One of the important properties is to configure the **mode of communication** between Logic developer-PLC and the Controller, which can be either **Ethernet or COM1**. If Ethernet is chosen,

then the IP Address of the target controller too should be specified. This IP address has to be first set up in the controller settings too (discussed ahead). If COM1 is selected then ensure that the serial cable (black cable) coming from the PLC is connected to the PC COM1 port. (Note: If COM1 port is used for communication then the Versamotion configuration software can not be run simultaneously as both of them share the same port. Hence we prefer to use Ethernet communication as it is fast and also allows the simultaneous operation of the Versamotion Software along with PLC programming with Logic Developer –PLC).

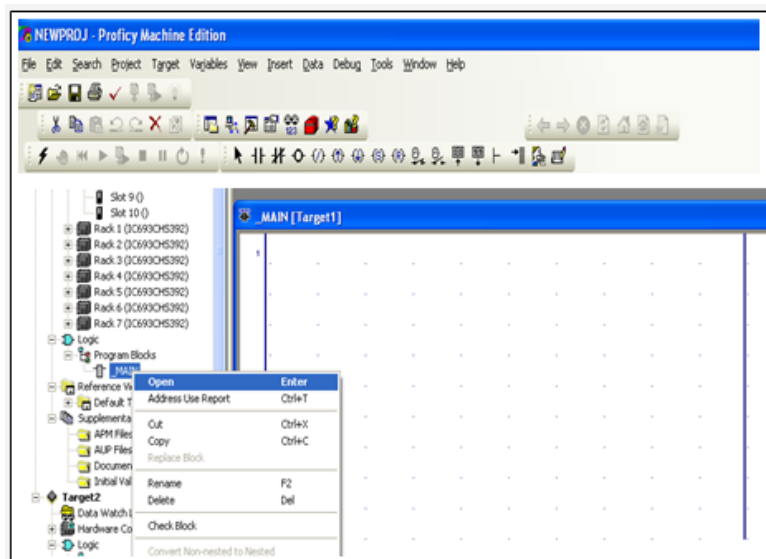
- Next, the hardware settings for the PLC in use ( i.e GE Fanuc 90-30) have to be done. In the Navigator expand the **Hardware Configuration** and expand the **Main Rack**. Double. A list of slots appears under the Main Rack which correspond to the various PLC components. Add and configure each slot one by one. For eg. click the Power supply slot, press the right mouse button and choose **Replace Module**. The Module Catalog dialog box will appear (Fig. 4.6). Select Power Supply model IC693PWR330, then click **OK**. Double click on the Power Supply slot to configure/Read its parameters. Similarly configure the other slots. The first slot is reserved for the Power Supply module (IC693PWR330), second for the CPU (IC693CPU374), third for the Digital Input module (IC693MDL645), fourth and fifth slots for Digital Output module (IC693MDL740) and last slot for the Bus controller module (IC693PBM200). Two.



**Fig. 4.6 Configuration of the PLC Hardware**

Digital Output slots are used because the experimental setup involves two output modules. In the CPU configuration do not forget to configure the Ethernet IP address in the parameter editor | Ethernet tab

7. Next left click on **Program blocks** | **Main** under the **Logic blocks** in the Navigator window (Fig. 4.7). This opens up the screen for developing the ladder program by ‘picking and dragging’ all the components of the ladder program for example, NO/NC contacts, Timers, relay coils, connectors etc. from the toolbar of the given screen.



**Fig. 4.7 Development of Ladder program in Proficy Machine**

8. Save the ladder program by clicking on **File** | **Save** or simply by clicking on the ‘floppy symbol’ for save. Give the program a suitable name.
9. To Run the saved program the following sequence is followed:

#### **Going online to the controller**

- (a) If the name of the target we want to go online to does not appear in bold characters, right-click the target and choose **Set as Active Target**. Bold characters indicate that this is the active target.
- (b) Right-click the active target and choose **Go Online** or Click the thunderbolt on the Online toolbar



**Fig. 4.8 Shortcut keys to go online**

(c) Follow the sequence from 1 through 4 in Fig. 4.8 to download to a GE Fanuc Controller. The download process builds and validates all run-time files necessary for a Target to perform its role in a completed project.

**Going Offline**

Click on 5 which stops the project and then click on 1 to disconnect from the controller.

**4.3 VERSAMOTION CONFIGURATION SOFTWARE**

Versamotion servo configuration tool, provided with Proficy Machine Edition 5.9, can be used for monitoring, analysis and control of various Servomotor parameters online i.e. when the Servomotor is running.

**4.3.1 General tools of Versamotion Configuration Software**

Open Proficy Machine edition > **Utilities** and from there select the option of **Versa motion Servo Configuration** tool. To establish a connection with Servo drive we have to do the following settings as shown in the dialog box below (19200, 8, N, 2)(Fig 4.9a,b)

COM Port Information	
Port Number	Com1
Station	2
Protocol	6 : 8,N,2 ( RTU )
Baud Rate	2 : 19200

**Fig 4.9(a)**



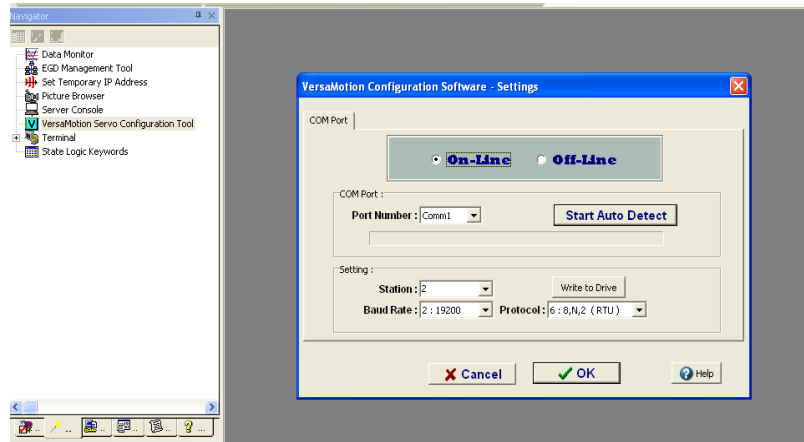



Fig. 4.9 (b)

Fig.4.9 (a), (b) Versamotion start-up windows

Following tools of Versamotion have been used in the Control and Monitoring of the Servomotor:

- a) **Data scope** : The data scope helps in graphical display of internal signals, similar to a digital oscilloscope. The data scope has three independent channels. The data on scope window can be printed or saved as a scope file (.scp). The data scope has zoom and trigger functions also. The Figure below shows data scope monitoring the servo-drive variables i.e. speed input command (rpm), Torque input command and motor rotation speed (rpm)

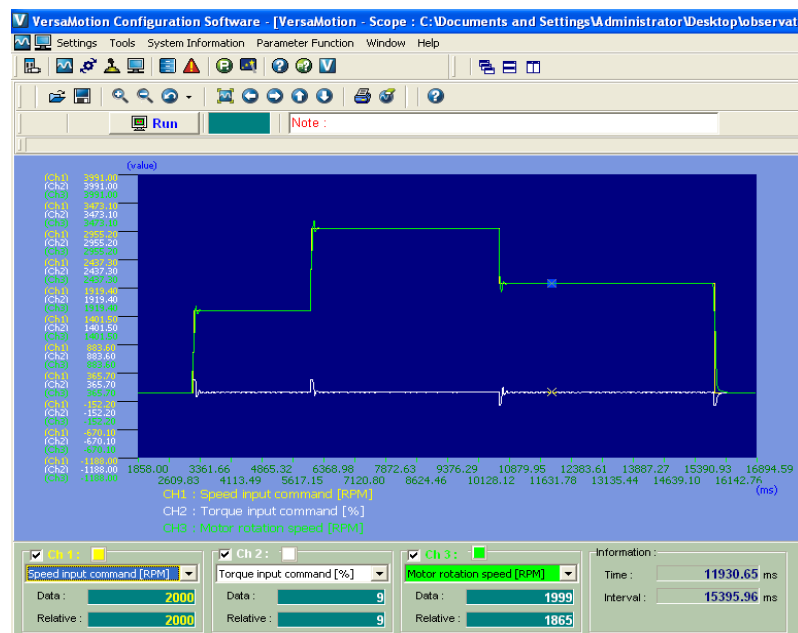

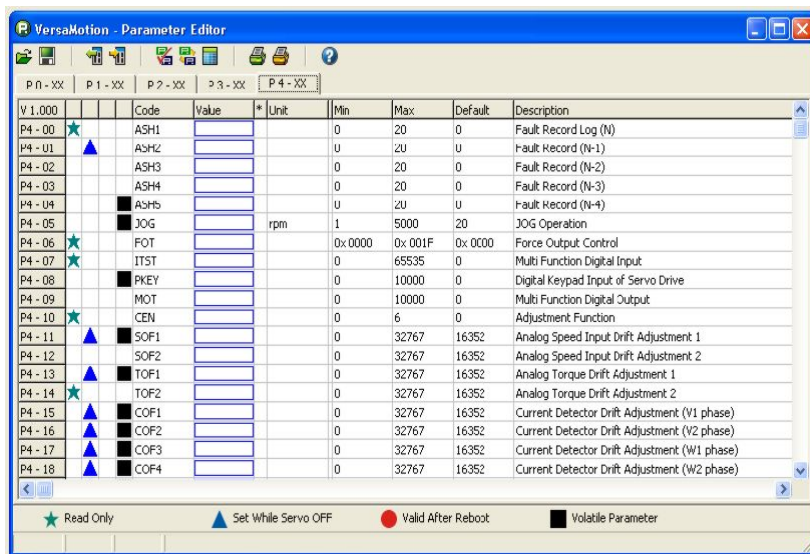




Fig.4.10 Servomotor parameter monitoring on Data scope


- b) **Parameter Editor**  : The Parameter Editor enables the user to read the status of all the parameters (P0,P1,P2 & P3) of the servo drive directly on the PC as well as allows the user to enter the new values of the parameters as per the required mode of control. Most parameters use the default values and can be printed or saved to a file for reuse. Changing values of certain servo-drive parameters needs power reboot to be given to servo-drive, while some parameters can be changed just by retyping .Some other parameters cannot be altered at all and retain default factory setting values.



 Read Parameters

 Write Parameters

**Fig.4.11 Parameter Configuration Window**

- c) **Digital IO / Jog Control Box**  This option allows the Control and verification of the digital I/O signals in the servo drive. It shows both the digital inputs (DIs) and digital outputs (DOs) including their individual settings and contact type. The central part of this dialog box shows the On/Off status of the DI/DO signals. The buttons on the right side are used to enable the DI and DO control via Versamotion software. The user may use these buttons to control the DI/DO signals while the Versamotion software is connected.

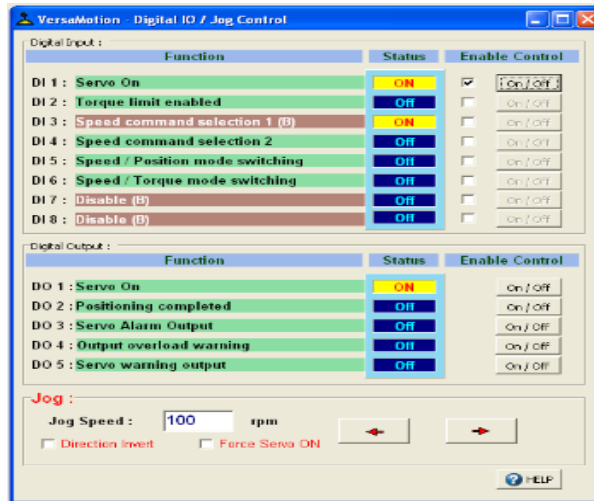



Fig.4.12 Digital IO / Jog control dialog box window

d) **Status Monitor**  : This option provides the user to monitor the servo status in the runtime mode. The exact values of the motor variables like motor feedback pulse and rotation number, speed input command and motor rotation speed value, etc. will be displayed in the status monitor dialog box shown below. In the dialog box, left side displays the monitored items, i.e. the setting value of parameter P0-02. The right side displays the **value** of the monitored item.

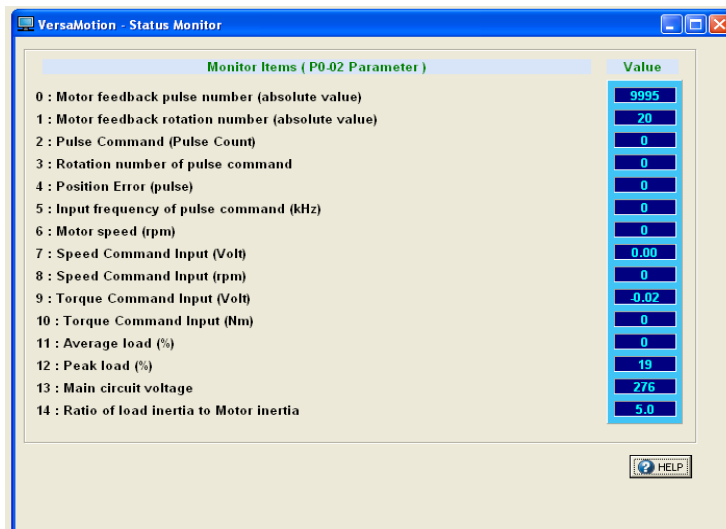


Fig.4.13 Status monitor window

### 4.3.2 The Digital Inputs and Digital Outputs of the Servo Drive

The VersaMotion Configuration software predominantly works based on the Digital Input and Output signals present on the CN1 Interface connector. The CN1 connector is a 50 pin connector and it provides access to three signal groups:

§ General interface for the analog speed and torque command, encoder reference signal from the motor, pulse / direction inputs, and reference voltages.

§ Eight programmable Digital Inputs (DI), can be set using parameters P2-10 — P2-17

§ Five programmable Digital Outputs (DO), can be set using parameters P2-18 — P2-22  
The function of each I/O point can be configured by the user as per control requirements. There are 45 input options and 11 output options present as shown in Fig.6 below. The figure explains input/output configuration as described by VersaMotion servo configuration tool and indicates some online monitored parameters of servomotor.

These digital input/ output options are used in combinations of seven and five respectively, depending on the type of application being developed and it as well depends on control mode in which motor is to be run. For example, for developing a conveyor type of application ,which needs high precision relating position, certain commands like command triggered (CTRG) are necessary to be activated, otherwise motor is unable to run in that particular mode.

**Table 4.1 Detailed Description of the Digital Input Signals (DIs)**

PLC OUTPUT	DI SIGNAL	SIGNAL CODE	PARAMETER VALUE	CONTROL MODE	CONTROL PARAMETER
Q1	DI1	SON	01	ALL	P2-10
Q2	DI2	CCLR	04	Pt	P2-11
		CTRG	08	Pr,Pr-S,Pr-T	
		TRQLM	09	S,Sz	
		SPDLM	10	T,Tz	
Q3	DI3	POS0	11	Pr	P2-12
		SPD0	14	S,Sz	
		TCM0	16	Pt,T,Tz,Pt-T,Pr-T,S-T	
Q4	DI4	POS1	12	Pr-S,Pr-T	P2-13
		SPD1	15	Pt-S,Pr-S,S-T	
		TCM1	17	Pt,T,Tz,Pt-T,Pr-T,S-T	
Q5	DI5	ARST	02	ALL	P2-14
Q6	DI6	CWL	22	Pt,Pr,S,Sz,T,Tz	P2-15
Q7	DI7	S-P	18	Pt-S,Pr-S	P2-16
		S-T	19	S-T	
		T-P	20	Pt-T,Pr-T	
Q8	DI8	EMGS	21	ALL	P2-17

SON	Servo ON	TCMO	Torque Command Selection
CCLR	Pulse Clear	ARST	Reset
CTRG	Command triggered	CWL	Reverse Inhibit Limit

TRQLM	Torque Limit Enabled	S-P	Position / Speed mode switching; OFF:SPEED,ON:POSITION
SPDLM	Speed Limit Enabled	S-T	Speed/Torque mode switching; OFF:SPEED,ON:TORQUE
POSO	Position Command Selection	T-P	Torque/Position mode switching; OFF:TORQUE,ON:POSITION
SPDO	Speed Command Selection	EMGS	Emergency Stop

**Table 4.2 Detailed Description of the Digital Output Signals (DOs)**

PLC INPUT	DO SIGNAL	SIGNAL CODE	PARAMETER VALUE	CONTROL MODE	CONTROL PARAMETER
I1	DO1	SRDY	O1	ALL	P2-18
I2	DO2	ZSPD	O3	ALL	P2-19
I3	DO3	TSPD	O4	ALL	P2-20
		HOME	O9	Pt,Pr	
I4	DO4	TPOS	O5	Pt,Pr,Pt-S,Pt-T,Pr-S,Pr-T	P2-21
		BRKR	O8	ALL	
I5	DO5	ALRM	O7	ALL	P2-22

SRDY	Servo Ready
ZSPD	Zero Speed
TSPD	Speed reached
TPOS	Positioning completed
ALRM	Servo Alarm Output (Servo fault)
HOME	Home Completed

### 4.3.3 Operation of PMSM under various Control Modes Using the Versamotion Configuration Software

Table 4.3 summarises all the possible modes in which the Servo Motor can be run using the VersaMotion Configuration Software:

**Table 4.3 Different Modes of Servomotor operation**

Mode	Code	Description
External Position Control	Pt	External position control for the servo motor is achieved via an external pulse command.

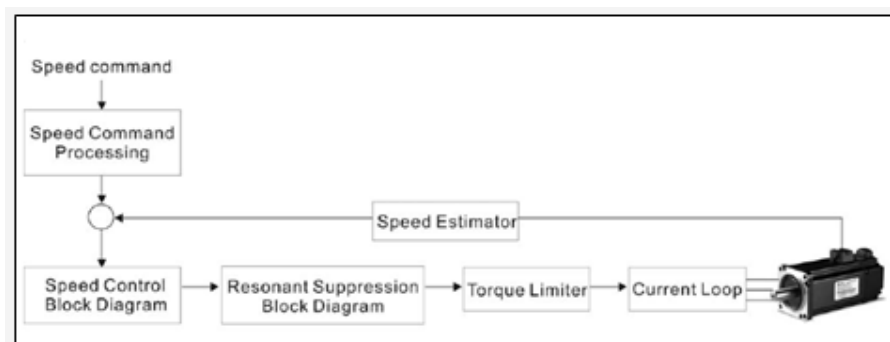
Internal Position Control	Pr	Internal position control for servo motor is achieved by eight commands stored in the servo controller and executed by digital inputs.
Speed Control	S	Speed control for servo motor is provided from an external analog -10 to +10 VDC command.
Internal Speed Control	Sz	Internal speed control for servo motor is provided only by parameters set within the controller. Digital Inputs control the internal speed parameters. (A maximum of three speeds can be stored internally).
Torque Control	T	Torque control for the servo motor is provided from an external analog -10 to +10 VDC command.
Internal Torque Control	Tz	Internal torque control for the servo motor is provided only by parameters set within the controller. Digital Inputs control the internal torque parameters. (A maximum of three torque levels can be stored internally).
Dual Modes		
External Position-Speed	Pt-S	Either Pt or S control mode can be selected via Digital Inputs.
External Position-Torque	Pt-T	Either Pt or T control mode can be selected via Digital Inputs.
Internal Position-Speed	Pr-S	Either Pr or S control mode can be selected via Digital Inputs.
Internal Position-Torque	Pr-T	Either Pr or T control mode can be selected via Digital Inputs.
Speed - Torque	S-T	Either S or T control mode can be selected via Digital Inputs.

#### 4.3.3.1 Speed Control Mode

The speed control mode is usually used for applications that require precision speed control such as robot manipulator, computer numeric control milling machine (CNC), etc. In this mode servomotor can be made to run from zero to full speed, i.e. variable speed with very high torque. Fig.4.14 shows a block diagram of the speed control mode.

The speed control command sources can be:

- a) **External** analog signal or speed command.(S)
- b) **Internal** parameters: P1-09 to P1-11 (Sz)



**Fig.4.14 Block Diagram for Speed Control Mode**

**Algorithm for Speed Control Mode Using the Versamotion Configuration**

1. Set the value of parameter P1-01 (CTL:Control Mode and Output direction)to 102.Cycle the drive power off and then ON for the new value of P1-01 to become effective.
2. Set the relevant Digital inputs for speed control mode using the following parameter values

**Table 4.4 Digital Input Settings for Internal Speed Control Mode**

Digital Input	Parameter Value	Code	Function Description
DI1	P2-10=101	SON	Servo On
DI2	P2-11=109	TRQLM	Torque limit enabled
DI3	P2-12=114	SPD0	Speed command selection 0
DI4	P2-13=115	SPD1	Speed command selection 1
DI5	P2-14=102	ARST	Reset
DI6	P2-15=0	Disabled	This DI function is disabled
DI7	P2-16=0	Disabled	This DI function is disabled
DI8	P2-17=0	Disabled	This DI function is disabled

3. Set The Digital Outputs as follows:

**Table 4.5 Digital Output Settings for Internal Speed Control Mode**

Digital Output	Parameter Value	Code	Function Description
DO1	P2-18=101	SRDY	Servo Ready
DO2	P2-19=103	ZSPD	Zero Speed
DO3	P2-20=104	TSPD	Target Speed
DO4	P2-21=108	BRKR	Brake
DO5	P2-22=107	ALRM	Alarm

4. In addition to the above Digital Inputs the following internal parameters for speed command also have to be set.

**Table 4.6 Parameter settings for Internal Speed Control Mode**

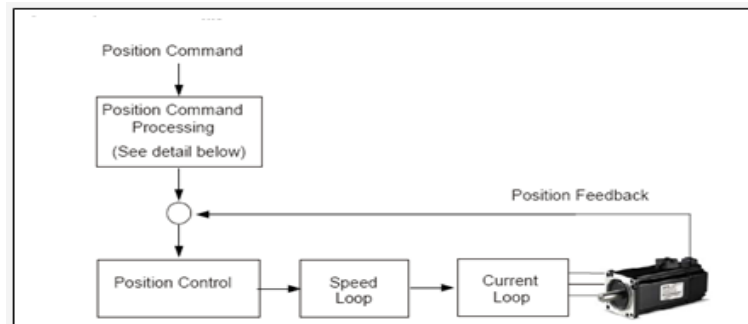
Speed Command	CN1 DI Signal		Command Source		Content	Range
	SPD1	SPD0				
S1	0	0	Mode	S	External Analog Signal	Voltage between Vref-Gnd
				Sz	N/A	Speed command is

					zero	
S2	0	1	Internal Parameters	P1-09	0-5000 rpm	
S3	1	0		P1-10	0-5000 rpm	
S4	1	1		P1-11	0-5000	

#### 4.3.3.2 Position Control Mode

The position control modes i.e., external position control or internal position control mode are usually used for applications requiring precision positioning, such as bottle filling applications, packaging industries, rotary drilling machine, metal profile cutting machine etc.

In order to provide convenient position control using servo drive, eight internal preset parameters are present. There are two methods for setting internal parameters. The first method is to set different position commands into eight internal parameters before operation and then use digital inputs POS0-POS2 to perform positioning control. The other method is to use serial communication to change values of eight internal parameters. Fig. 4.15 shows a block diagram of the Position control mode.



**Fig.4.15 Block diagram for Position Control Mode**

#### Algorithm for Position Control Mode Using the Versamotion Configuration software

1. Set the value of parameter P1-01 (CTL:Control Mode and Output direction)to 101(for internal Position control) OR to 100 (for external position control) depending on the case .Cycle the drive power off and then ON for the new value of P1-01 to become effective.
2. Set the following parameters for the position control mode (Both Pr and Pt). If not set , they take up the default values.

**Table 4.7 Important parameters for Internal Position Control Mode**

Parameter	Code	Function Description
P1-02	PSTL	Speed and Torque limit
P1-46	GR3	Encoder output Pulse Number



P1-55	MSPD	Maximum speed Limit
P1-12 to P1-14	TQ1- TQ3	1 <sup>st</sup> -3 <sup>rd</sup> Torque command
P2-50	CCLR	Pulse Deviation Clear Mode

3. (i) Set the following Digital inputs for Internal Position Control mode

**Table 4.8 Digital Input Settings for Internal Position Control Mode**

Digital Input	Parameter Value	Code	Function Description
DI1	P2-10=101	SON	Servo On
DI2	P2-11=108	CTRG	Command Triggered
DI3	P2-12=111	POS0	Position command selection 0
DI4	P2-13=112	POS1	Position command selection 1
DI5	P2-14=113	POS2	Position command selection 2
DI6	P2-15=0	Disabled	This DI function is disabled
DI7	P2-16=0	Disabled	This DI function is disabled
DI8	P2-17=0	Disabled	This DI function is disabled

(ii) Set the following Digital outputs:

**Table 4.9 Digital Output Settings for Internal Position Control Mode**

Digital Output	Parameter Value	Code	Function Description
DO1	P2-18=101	SRDY	Servo Ready
DO2	P2-19=103	ZSPD	Zero Speed
DO3	P2-20=109	HOME	Home Completed
DO4	P2-21=105	TPOS	Positioning Completed
DO5	P2-22=107	ALRM	Alarm

( iii ) In addition to the above DI and DOs the following parameters are to be set for Internal Position Control

**Table 4.10 Parameters for Internal Position Control Mode**

Parameter	Code	Function Description
P1-15 to P1-30	POS1H to POS8H	1st to 8th Position command for rotation
	POS1L to POS8L	1st to 8th Position command for pulse
P2-36 to P2-43	POV1 to POV8	Moving speed setting of 1st to 8th position
P1-33	POSS	Position Control Mode (0=Absolute,1=Relative)

The Odd Position Parameters (ie. P1-15, P1-17 etc. till P1-29) correspond to the number of complete rotations and the Even Parameters (ie. P1-16, P1-18 etc. till P1-30) correspond to the number of additional pulses representing any fraction of a motor revolution (pulses). Digital I/O signals (POS0 to POS2 and CTRG) are used to select one of the eight position commands and then trigger the execution of that command as shown in the table .In this table value “0” indicates that the contact is in OFF state (Normally open) and “1” indicates that contact is in ON state (Normally closed). CTRG in the given table represents the input used to trigger the move selected using the POS0, POS1 and POS2 inputs. The move is triggered on the rising edge of this digital input.

**Table 4.11 Parameter Settings for Internal Position Control Mode**

Position Command	POS2	POS1	POS0	CTRG	Parameters	Moving speed setting Parameter	Description
Internal Position 1	0	0	0	•	P1-15	P2-36	Rotation No. (+/- 3000)
					P1-16		Pulse No. (+/- max cnt)
Internal Position 2	0	0	1	•	P1-17	P2-37	Rotation No. (+/- 3000)
					P1-18		Pulse No. (+/- max cnt)
Internal Position 3	0	1	0	•	P1-19	P2-38	Rotation No. (+/- 3000)
					P1-20		Pulse No. (+/- max cnt)
Internal Position 4	0	1	1	•	P1-21	P2-39	Rotation No. (+/- 3000)
					P1-22		Pulse No. (+/- max cnt)
Internal Position 5	1	0	0	•	P1-23	P2-40	Rotation No. (+/- 3000)
					P1-24		Pulse No. (+/- max cnt)
Internal Position 6	1	0	1	•	P1-25	P2-41	Rotation No. (+/- 3000)
					P1-26		Pulse No. (+/- max cnt)
Internal Position 7	1	1	0	•	P1-27	P2-42	Rotation No. (+/- 3000)
					P1-28		Pulse No. (+/- max cnt)
Internal Position 8	1	1	1	•	P1-29	P2-43	Rotation No. (+/- 3000)
					P1-30		Pulse No. (+/- max cnt)

### 4.3.3.3 Torque Control Mode

The torque control mode is usually used for applications that require accurate control of torque such as military applications involving faster torque control, etc. In this control mode high torque can be achieved, which is more than positive rated torque that is obtained in control modes described above (i.e., position control mode and speed control mode). Fig. 4.16 shows a block diagram of the torque control mode.

Versamotion servo drives support two kinds of command sources in torque control mode: external analog signal or internal torque parameters. Torque command sources include:

- a) External analog signal or torque command.(T)
- b) Internal parameters: P1-12 to P1-14. (Tz)

The command source selection is determined by the Digital Inputs (DI) on connector CN1.

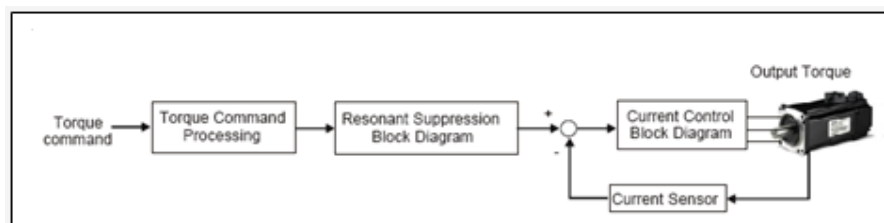


Fig.4.16 Block Diagram for Torque Control Mode

### Algorithm for Torque Control Mode Using the VersaMotion Configuration software

1. Set the value of parameter P1-01 (CTL:Control Mode and Output direction) to 105 (for internal Position control) OR to 103 (for external position control) as the case may be. Cycle the drive power off and then ON for the new value of P1-01 to become effective
2. Set the following parameters for the Torque control mode ( Both Tz and T ). If these parameters are not set they take up the default values.

**Table 4.12 Important Parameters for Torque Control Mode**

Parameter	Code	Function Description
P1-02	PSTL	Speed and Torque limit
P1-46	GR3	Encoder output Pulse Number
P1-55	MSPD	Maximum speed Limit
P1-09 to P1-11	TQ1-TQ3	1 <sup>st</sup> -3 <sup>rd</sup> Speed command

3. (i) Set the following Digital inputs for Torque Control mode

**Table 4.13 Digital Input Settings for Internal Torque Control Mode**

(ii)

Digital Input	Parameter Value	Code	Function Description
DI1	P2-10=101	SON	Servo On
DI2	P2-11=110	SPDLM	Speed Limit Enabled
DI3	P2-12=116	TCM0	Torque Mode Selection 1
DI4	P2-13=117	TCM1	Torque Mode Selection 2
DI5	P2-14=102	ARST	Reset
DI6	P2-15=0	Disabled	This DI function is disabled
DI7	P2-16=0	Disabled	This DI function is disabled
DI8	P2-17=0	Disabled	This DI function is disabled

Set the

following Digital Outputs:

**Table 4.14 Digital Output Settings for Internal Torque Control Mode**

Digital Output	Parameter Value	Code	Function Description
DO1	P2-18=101	SRDY	Servo Ready
DO2	P2-19=103	ZSPD	Zero Speed
DO3	P2-20=104	TSPD	Target Speed
DO4	P2-21=108	BRKR	Brake
DO5	P2-22=107	ALRM	Alarm

- (iii) In addition to the above DIs the following parameters are to be set:

**Table 4.15 Parameter Settings for Internal Torque Control Mode**

Torque Command	CNI DI Signal		Command Source			Content	Range
	TCM1	TCM0					
T1	0	0	Mode	T	External Analog Signal	Voltage between T-REF-GND	+/-10V
				Tz	N/A	Torque command is zero	0
T2	0	1	Internal Parameter			P1-12	+/- 300%
T3	1	0				P1-13	+/- 300%
T4	1	1				P1-14	+/- 300%

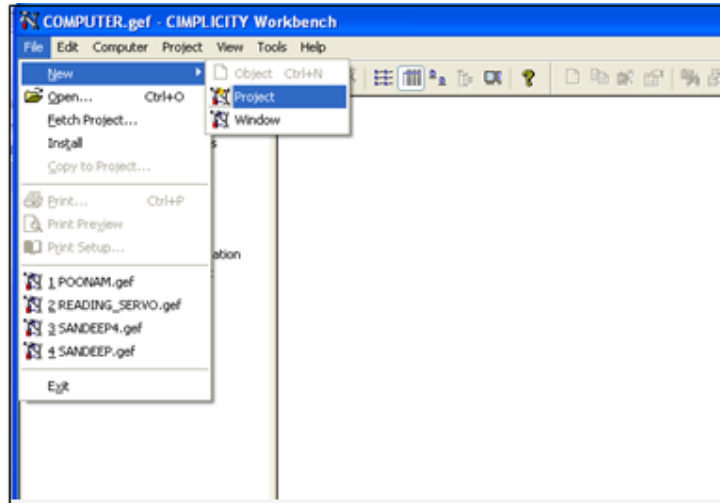
#### **4.4 Proficy HMI/SCADA Cimplicity 7.5 Software**

GE-Fanuc’s Cimplicity HMI Plant edition 7.5 software is used for remote monitoring and control of the system, ie it is a SCADA software. CIMPLICITY is based on a client-server architecture consisting of servers and viewers. Servers are responsible for the collection and distribution of data. Viewers connect into servers and have full access to the collected data for viewing and control actions. Servers and viewers can be easily networked together to seamlessly share data without the need to replicate point database from node to node. For example, points are configured once and only once on a server. Screens can be developed and stored in a single location on the network and accessed by any other CIMPLICITY display on the network.

This software is used to develop user defined controls like start, stop, speed control, etc. on the software window to control the system remotely. SCADA screens can be developed which make monitoring of processes very easy. Also data acquisition of all the device parameters( in the form of graphs trends, excel sheet, tables etc.) too can be done on the window of personal computer or screen of HMI.

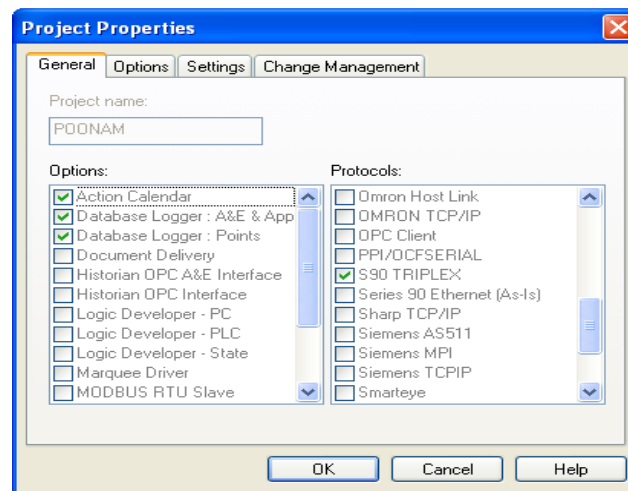
##### **4.4.1 Setting up a Cimplicity Project for controlling a system remotely**

- (1) To create a new Project, open the Cimplicity workbench window. Go to **File** option and select **New Project** as shown in Fig. 4.17.

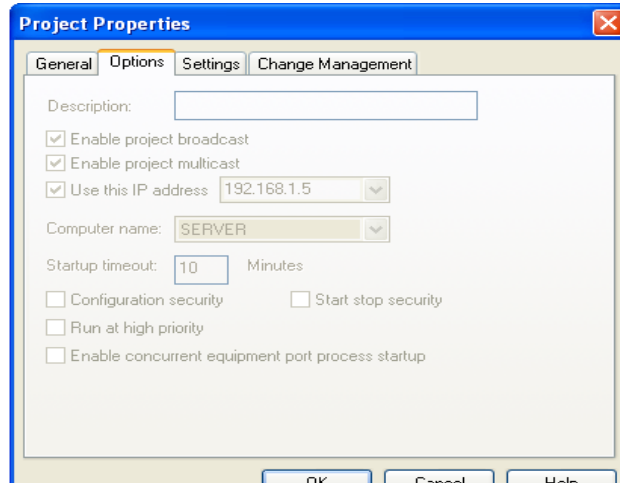


**Fig.4.17 New Project creation in Cimplicity**

(2) Open the Project **Properties** dialog box by clicking Project > Properties on the workbench toolbar to configure the project properties shown in 4.19 (a) and (b)



**Fig. 4.18 (a)**

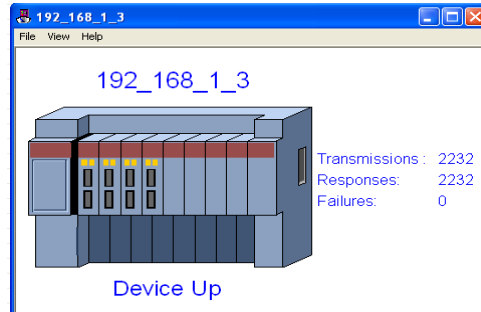


**Fig.4.18 (b) Configuring Cimplicity Project Properties**

- (3) Next step is to configure the device and the port (of the device) used for the project. A **device** is anything that can communicate point data to CIMPLICITY HMI software. CIMPLICITY HMI software can read data from and write data to devices. A **port** is a communication "socket" that connects one or more devices to the computer. In this case PLC is the Device and it communicates with the SCADA software installed in the PC via the Ethernet through the port Master S90\_TRIO and using the protocol S90 TRIPLEX.
  
- (4) The configuration of the Device, Port and the communication points can be done either by the user by right clicking on the **Device**, **Port** and **Points** on the left pane of the workbench or automatically using the Cimplicity Wizard (This is explained in more detail in 5.3.2). The Cimplicity **Project wizard window** automatically guides the user to configure the device.(Fig.4.20)







**Fig. 4.20 (b)**

**Fig.4.21 (a) and (b) Checking Device communication status using Quick View**

If communications cannot be established for a device on the port, the device is considered to be down, and a `DEVICE_DOWN` alarm is generated. Once a device is down, periodic attempts are made to resume connections to it. For this the number of scans to wait before attempting to reconnect to a device after a communications error has been detected have to be entered as 'Retry counts' in the Port Properties .

- (5) Click on **Screens** on the left pane of Cimplicity workbench. The following screen appears on which we can create a graphical interactive window by simply selecting and dragging the desired objects from the **Object Explorer** to the screen .
  
- (6) Then we need to configure the **Properties** of each object on the screen by right clicking on it. Say if a motor has to be animated, ie to be shown rotating, first select the motor, then right click on it, click on Properties. As shown in Fig. 4.21 a dialog box of the following type opens up. Enter the Point ID or expression that we want the rotation to reflect. This expression should relate the object rotation with the relevant timer variable used in the PLC program Use the **Quick Offset** option to select the centre of rotation. Fill the 'Min' and 'Max' values to set the angle of rotation. A '+' is for clockwise and '-' for anticlockwise rotation. Similarly there are properties associated with each object which should be configured depending on the animated screen requirements

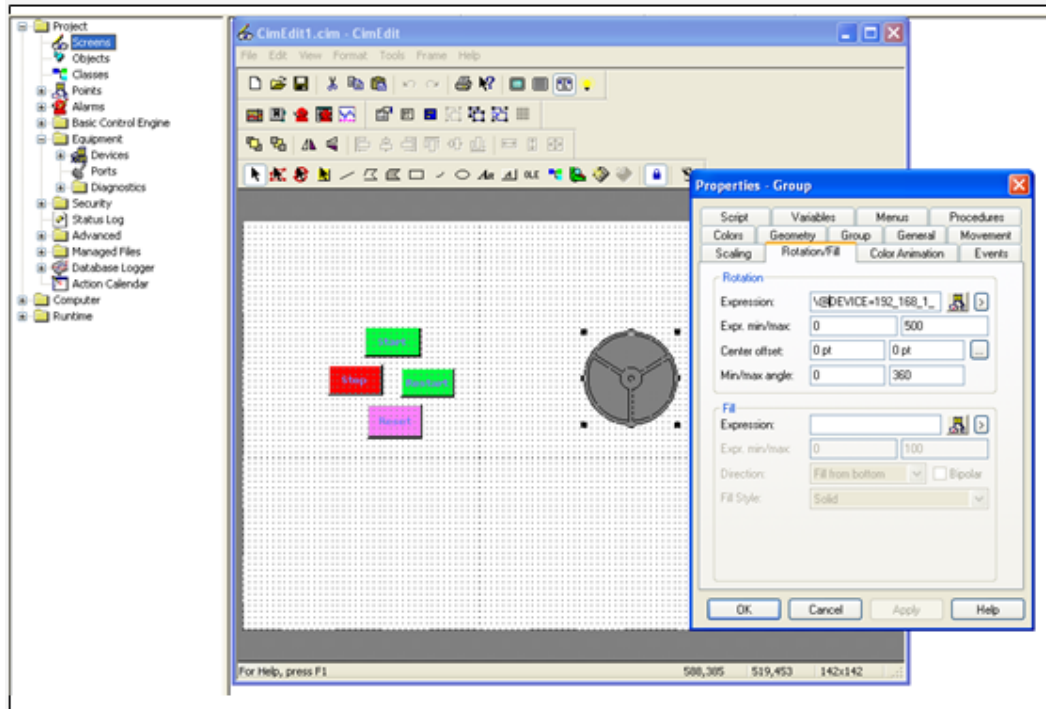



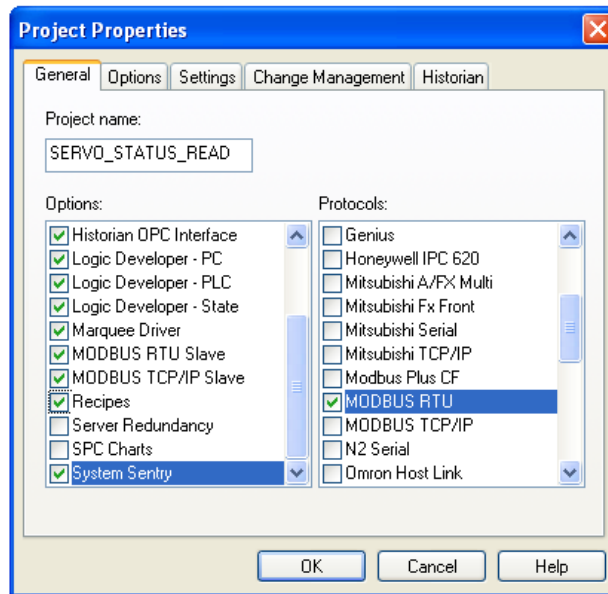
Fig. 4.21 Creating the SCADA screen using CIMEDIT

- (7) A PLC program should be written in Proficy Machine Edition to achieve the desired animation for the screen.
- (8) Finally to get an animated screen, first run the PLC program and validate it for errors as was done in the Proficy Machine ladder logic window. Once the program is successfully enabled then simultaneously open the Cimplicity screen. Click the **Test Screen** button  on the Standard Toolbar. The CimView screen displays and the animation begins by clicking on an appropriately configured 'Start' button.

**Note:** The instant animation of any object can be viewed by just configuring the object properties and then clicking on the 'Test Screen' button. But to run a completely animated application on the screen we need to write a PLC program relating all the objects with each other through the program logic.

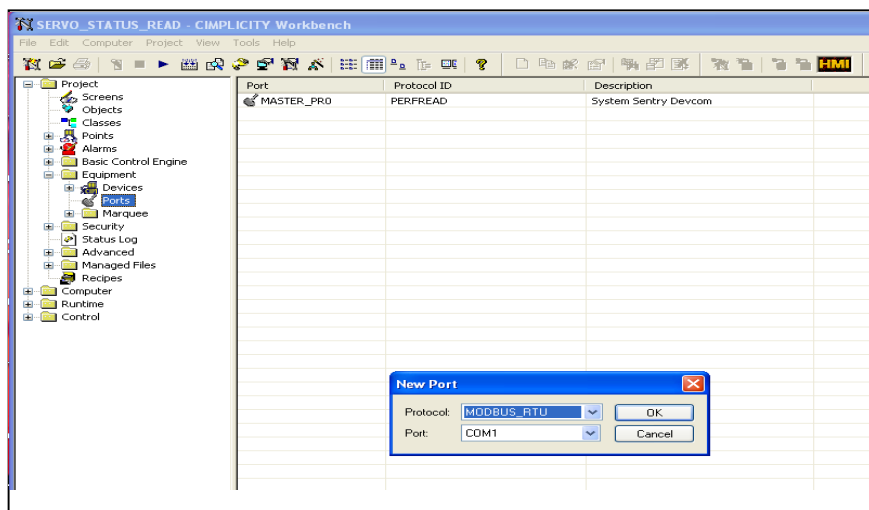
#### 4.4.2 Creating a Cimplicity Project for Monitoring the Servomotor Variables ( eg, Speed) on the SCADA Screen

- (1) To create a new Project, open the Cimplicity workbench window. Go to **File** option and select **New Project**
- (2) Fill the Project **Properties** dialog box as shown in Fig. 4.22

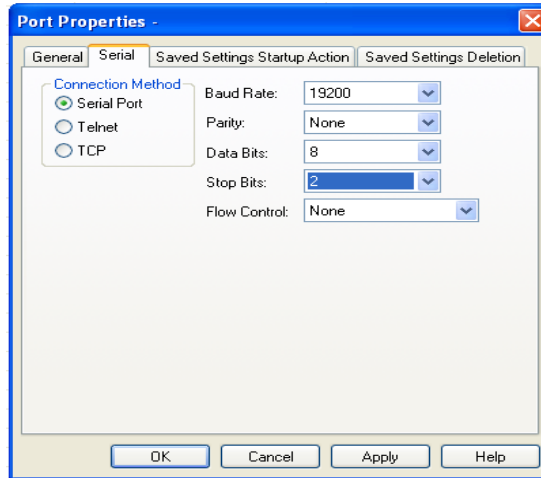


**Fig. 4.22 Configuring Cimplicity Project Properties**

- (3) Next we click on **Equipment** in the left pane and configure the port. **COM1** is used as a port. It is through this port that the Versamotion drive is connected to the PC and the motor parameters will be read on SCADA. Right Click on **Port** > **New** and set the dialog boxes as shown in Fig. 4.23 and Fig. 4.24

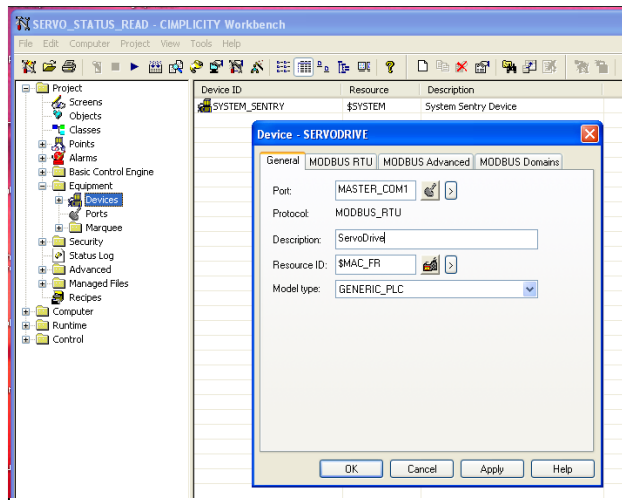


**Fig 4.23 Configuring a new port**

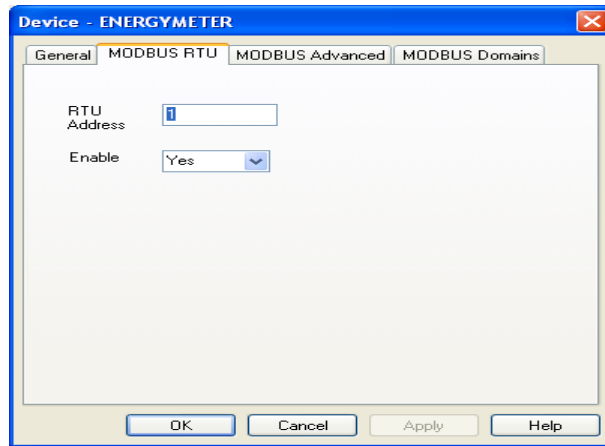


**Fig.4.24 Configuring Port Properties**

- (4) After configuring the port, the device will be configured .Under **Equipment** right click on **Device> New** and set the dialog boxes as shown in Fig. 4.25 and Fig. 4.26.



**Fig.4.25 Configuring a new Device**

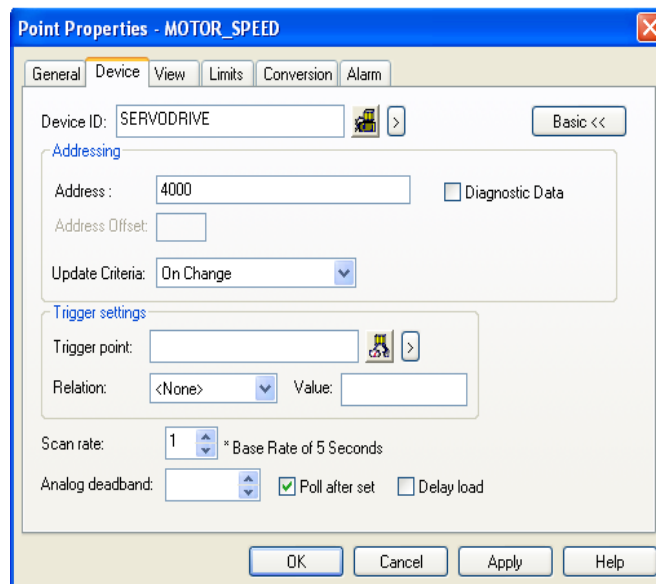


**Fig. 4.26 Configuring Device Properties**

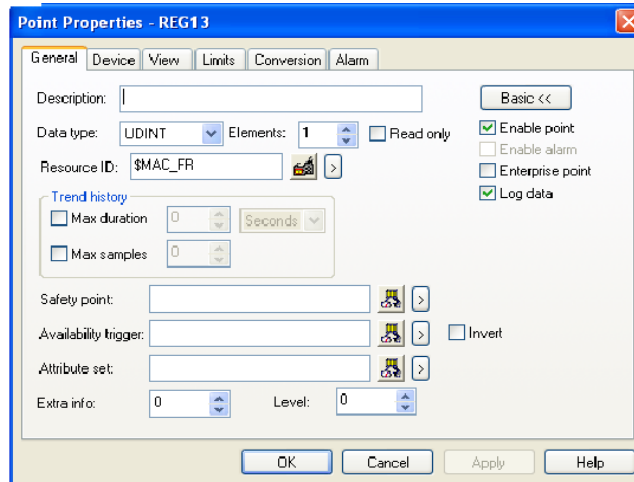
- (5) After configuring the Port and Devices we need to configure the device points for data acquisition, Points are configured using the MODBUS addresses of the Versamotion Drive parameters which are to be monitored. For example if the speed of the servomotor is to be monitored using the status monitor parameter P0-04 (communication address 0004H) then the MODBUS address for P0-04 is calculated as:  $0004H = 4$  (in Binary)

The Modbus address will then be  $400001 + 4 = 400005$  (ie convert the Hex address

to Binary and then add 400001 to that to get the MODBUS address). The dialog boxes are too be filled as shown in Fig. 4.27 and Fig.4.28.

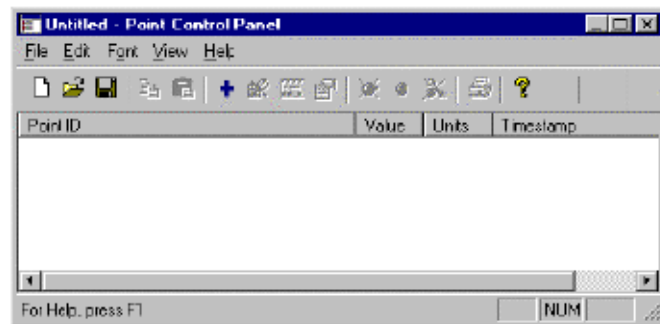


**Fig.4.27 Creating a new device point**



**Fig.4.28 Configuring the Point Properties**

- (6) After point configuration a screen can be designed on the SCADA software for displaying the status of the Versmotion Drive parameters (by configuring graphical blocks using configured points) or the desired points can be read directly from the **point control panel**.
- (7) To open the point control panel click on the **Runtime Folder** in the left pane of the workbench and under that is the **Point Control Panel** option. An empty Point Control Panel opens as shown in Fig. 4.29.



**Fig.4.29 Point Control Panel Window**

- (8) To monitor the configured points, click on the **Points** in the left pane of the workbench. Then copy and paste the desired points which you want to monitor into the Point Control Panel. The Point Control Panel then displays the runtime values of the desired points as shown in Fig.4.30.

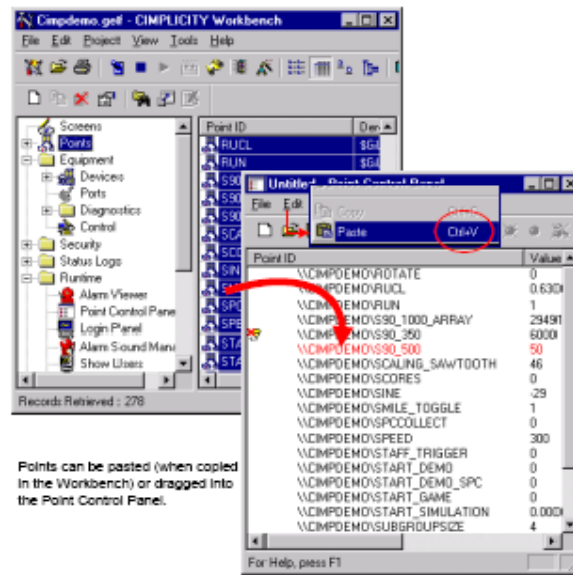


Fig.4.30 Tracking of Point status on the Point Control Panel

## 4.5 PLC PROGRAMMING

### 4.5.1 General

PLC programming basically refers to the method of developing a soft tool to communicate information to the PLC. Various methods of programming PLCs are

- (a) **Ladder Diagrams (LD):** This method uses graphic symbols similar to relay schematic circuit diagrams. It consists of two vertical lines representing the power rails. Circuits are connected as horizontal lines between these two vertical rails. They are very easy to understand and develop.
- (b) **Function Block diagram (FBD) :** When a ladder diagram is composed of ladder logic instructions in the form of building blocks to create simple executable units of logic, it is called a FBD. Every instruction is depicted as a functional block with the function name written in it.
- (c) **Instruction List (IL):** Instruction list is a programming language using mnemonic abbreviations of Boolean logic operations.
- (d) **Sequential Function Chart (SFC):** These are more powerful versions of simple flowcharts.
- (e) **Structured Text(ST) :** Structured Text programming is a more modern programming Language similar to languages such as BASIC.

In the present work all programs were developed using Ladder Diagrams(LD) on the Proficy Logic Developer -PLC. Hence in the coming paragraphs more details are given about the ladder programming of the PLC 90-30. Other programming options are also offered by Proficy.

The Ladder Programming is done on the computer on the Proficy Logic Developer-PLC platform. It provides an easy ‘drag and drop’ facility to develop a complete Ladder diagram rung by rung. After the Program is developed on the PC it is downloaded to the target controller either using a serial connection via the COM1 port or using the RJ-45 Ethernet port on the PLC. For Logic Developer - PLC to communicate with a target Controller, a connection must be properly configured (Discussed earlier in this Chapter )

#### 4.5.2 Instruction Set Used for Programming GE Fanuc Series 90-30 PLC

The instruction set of the GE-Fanuc 90-30 comprises of various instructions. But table 4.16 lists only the instructions used in the present work for programming. Some additional points are also listed about the usage of the Timer instruction

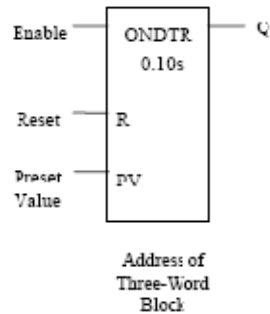
**Table 4.16 Important Instructions for GE Fanuc Series 90 -30 PLC [31]**

<b>Contacts</b>	<b>Name</b>	<b>Description</b>
— —	Normally Open contact (NO)	A normally open contact acts as a switch that passes power flow if the associated reference is ON (at logic 1).
— /—	Normally Closed contact(NC)	A normally closed contact acts as a switch that passes power flow if the associated reference is OFF (at logic 0).
<b>Coils</b>		
—( )—	Normally open coil	A coil sets a discrete reference ON while it receives power flow. It is non-retentive
—(/)—	Negated Coil	A negated coil sets a discrete reference ON when it does not receive power flow. It is not retentive
—(•)—	Positive Transition coil	If the reference associated with a positive transition coil is OFF, when the coil receives power flow it is set to ON. Any contacts associated with that coil will change state for one PLC scan (sweep). This coil can be used as a one-shot.
—(M)—	Retentive Coil	Like a normally open coil, the retentive coil sets a discrete reference ON while it receives power flow. The state of the retentive coil is retained across power failure.
—(R)—	Reset coil	The RESET coil sets a discrete reference OFF if the coil receives power flow. The reference remains OFF until the reference is reset by another coil



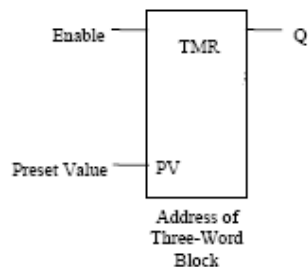
Timers		
ONDTR	Retentive ON - Delay Timer	A retentive on-delay timer (ONDTR) increments while it receives power flow and holds its value when power flow stops
TMR	Simple ON-delay Timer	The simple on-delay timer (TMR) function increments while it receives power flow and resets to zero when power flow stops

a) ONDTR is represented by the following Block



- Ø **Address of Three-Word Block:** The ONDTR uses three consecutive words (registers) of %R memory to store the Current value ( word 1),Preset value (word 2) and the Control word (word 3). The Control word is read not written to.
- Ø **Enable:** When this input receives power flow the Timer starts functioning or starts accumulating time. At any instant thereafter the Timer value is called the Current Value (CV).
- Ø **Reset:** This is the Reset input, which when receives power flow resets the current timer value to zero (i.e Q=0)
- Ø **Preset Value:** This is the value to which the timer is pre-set to obtain the desired delay by multiplying it with the appropriate **time base** value.
- Ø **Q:** This is the timer output which gets energized (i.e goes high) when the CV becomes equal to or more than the PV.

b) TMR represents a normal timer with only an Enable input and no Reset input. It is represented by the following block



- c) ONDTR and TMR may be counted in tenths (0.1), hundredths (0.01) or thousandths (0.001) of a second. The term in the bracket is called the **Time Base**. Depending on the Time Base selected we can have ONDTR\_TENTHS, ONDTR\_HUNDS etc. The total delay time (in seconds) set for the timer is given by Preset Value (PV) multiplied by the Time Base. Say for eg. a PV value of 100 set for an ONDTR\_Tenths (Time Base of 0.1) gives a delay of 10 Seconds.
- d) The main point of difference between ONDTR and TMR is that when **Enable** is disabled in ONDTR, output Q stops at the current accumulated value and when **Enable** is again activated, output Q starts accumulating from the last retained value. But in case of a TMR, if the Enable input is disabled, Q becomes zero.

#### 4.5.3 User References for PLC Series 90-30

The inputs and outputs are all identified by their addresses, the notation used depending on the PLC manufacturer. In the PLC 90-30 following types of references have been used :

- Ø **%I (Discrete Input Points):** The %I prefix represents input references. This prefix is followed by the reference's address in the input table (for example, %I00021). %I references are located in the input status table (Fig.4.31), which stores the state of all inputs received from input modules during the last input scan. %I data can be retentive or non-retentive. %I basically points to the Input Relays (transistors) which physically exist and receive signals from the outside world through switches, sensors, etc.

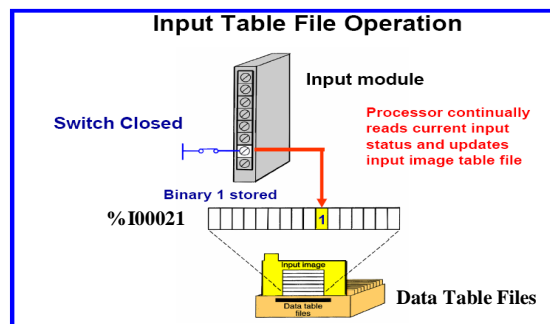


Fig. 4.31 Input table file operation

- Ø **%Q (Discrete Output Points)** : The %Q prefix represents physical output references. The %Q prefix is followed by the reference's address in the output table (for example, %Q00016). %Q references are located in the output status table (Fig. 4.32), which stores the state of the output references as last set by the application program. This output status table's values are sent to output modules during the output scan. %Q basically points to the Output Relays (transistors, relays or TRIACs) which physically exist and send on/off signals to output devices like solenoids, lights, etc connected to the output module.

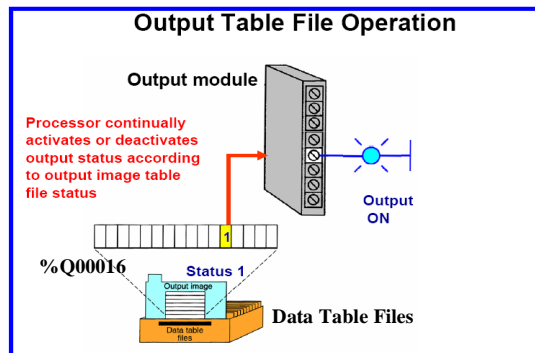


Fig. 4.33 Output table file operation

- Ø **% M (Internal Coils)**: It is normally used to represent internal references ie. Internal Relays which do not exist as real-world switching devices but are merely bits in the storage memory used to behave like relays. This prefix is followed by the register address of the reference (for example, %M0002)
- Ø **%R (Register Memory)**: The prefix %R is used to assign system register references, which will store program data such as the results of calculations. This prefix is followed by the register address of the reference (for example, %R0001)

**Specifications for CPU-374**

%I	2,048 (fixed)
%Q	2,048 (fixed)
%M	4,096 bits (fixed)
%R	Configurable 128 to 32,640 words

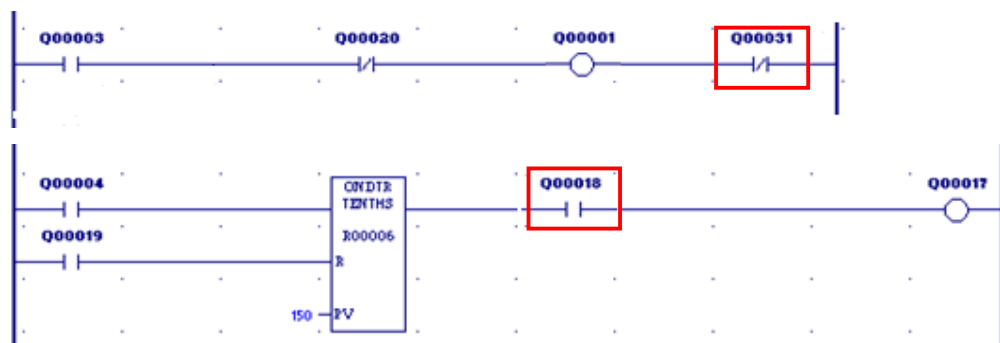
**4.5.4 Important Features of the Ladder Diagram**

Following are some points worth remembering before developing a ladder diagram in Proficy Machine for GE Fanuc Series 90-30 PLC:

- Ø In a Ladder Diagram the power flows from left to right.
- Ø In general, ladder logic is solved from left to right and top to bottom, although this flow direction can be altered temporarily by subroutine calls and jumps.
- Ø The ladder diagrams are based on a **COIL-CONTACT** relationship like relays. That is, every coil is associated with a contact (NO, NC or both). These contacts open or close depending on the status of the coil associated with them. For eg. a NO contact closes and a NC contact opens when the coil associated with them is energized. As soon as the coil de-energises the contacts come back to their original state.
- Ø Usually every rung terminates with a coil (output)



- Ø Each rung should contain at least one output coil and can have maximum up to eight coils
- Ø It is advisable to avoid the use of an output coil more than once in the program but a particular input can appear in more than one rung of a ladder.
- Ø A contact **can not be** placed on the right of the output coil or as the output of a timer.



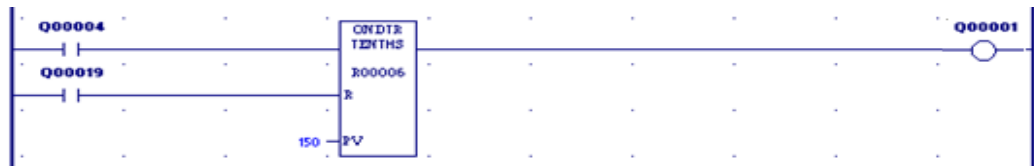
Ø In the following rung say Q00001 is activated by closing of M00001 contact. Then Q00001 can be deactivated either by opening of M00001 (with Q00023 remaining closed) or by the opening of Q00023 (with M00001 being closed) or with both M00001 and Q00023 being open (AND Logic).



Similarly in the following rung Q00001 is activated 50 secs after Q00003 closes and enables the Timer R00007. After that Q00001 can be deactivated by opening Q00003.



In the following rung we have an ONDTR. Here also the activation of Q00001 is similar to the above case, ie. Q00001 gets activated 15 secs after Q00004 closes and enables the Timer. But the deactivation of Q00001 can not be done just by disabling Q00004, as this timer has a tendency to retain its earlier value on being disabled and so Q00001 remains energized. The only way to deactivate Q00001 is by closing the RESET contacts i.e Q00019.



## **4.6 CONCLUSION**

The chapter presented an elaborate description of the various softwares that were analysed, configured and implemented for the control and monitoring of the Three Phase Permanent Magnet Synchronous AC drive. The software platforms which were mainly discussed are the Proficy Machine Edition 5.90 for Ladder Programming, the Versamotion configuration software for the Servodrive and the Cimplicity/HMI edition 7.50 for SCADA /HMI implementation. The project windows developed on these software platforms for various modes of the PMSM control and monitoring have been successfully implemented. The fundamentals of the Ladder logic Program have also been discussed.

## CHAPTER V

### COMMUNICATION PROTOCOL IN PLC, SCADA AND SERVODRIVE

---

#### 5.1 GENERAL

The protocols and communication media employed in the present set-up has been represented in the form of a block diagram at the end of this chapter in Fig.5.9 and the details of the main modes of communication like Serial Communication and ETHERNET are discussed in the following paragraphs. This chapter aims to provide an insight into how the various components have been networked with each other to result in an integrated system.

#### 5.2 SERIAL COMMUNICATION

Serial communication implies data transmission bit by bit through twisted-pair cables over long distances. RS-232, RS-422 and RS-485 are some popular standards for serial communication. RS-232 i.e., Recommended Standard 232, is a standard for serial binary data signals connecting between a DTE, Data Terminal Equipment and a DCE, Data Circuit-terminating Equipment. It is commonly used in computers for serial communication ports. In RS-232, user data is sent as a time-series of bits. Both **synchronous** and **asynchronous** transmissions are supported by this standard. The data communication links used with peripheral equipment can be **unidirectional** or **bidirectional**. If a peripheral is strictly either an input or an output device, then data transmission occurs in only one direction. In this case, a unidirectional serial signal line is all that is required to complete the link. Devices that serve as both input and output devices (e.g., video terminals) require bidirectional links. There are two ways to achieve this bidirectional communication. First, a single data line can be used as a shared communication line. The data can be sent in either direction, but only in one direction at a time. This operation is known as **half duplex**. If simultaneous bidirectional communication is required, two lines can connect the PLC to the peripheral. One line would be assigned permanently as an input, while the other would be a permanent output. This mode is known as **full duplex**.

**RS-232** ports are suitable for short distance point-to-point connections up to about 25 meters in length, but not for longer lines or **multidrop** (multipoint i.e. multiple devices

connected to the same signal cable) connections. The specification recommends limiting the data rate to 19,200 bits per second) or less, but rates up 115,200 bps are frequently used with short cables (typically about 2 meters).RS-422 is a bidirectional extension of RS232 for industrial environments.**RS-442** supports higher data rates and longer distances than RS-232. A 100-ohm nominal impedance is recommended for twisted pair circuits in cables, and 100-ohm terminating resistors are recommended for the receiving end of each circuit. Some RS-422 ports support multidrop operation. However, this capability is not guaranteed by the standard. Thus RS-485 standard has effectively replaced RS-422 because it adds guaranteed multidrop employing the Master-Slave paradigm (one master and n fixed address slaves). RS-485 uses 120-ohm cable and terminating resistors.The details of these three standards are summarized in table 5.1.

**Table 5.1 Comparison of RS-232, RS-422 and RS-485 serial standards**

S.No	Properties	RS-232	RS-422	RS-485
1	Mode of Operation	half duplex full duplex	half duplex	half duplex
2	Maximum No. of Drivers	1	1	32
3	Maximum No. Of Receivers	1	10	32
4	Network topology	point-to-point	multidrop	multipoint
5	Max. Distance	15 m	1200 m	1200 m
6	Max speed at 12 m Max speed at 1200 m	20 kbps 1 kbps	10 Mbps 100 kbps	35 Mbps 100 kbps
7	Max slew rate	30 V/•s	N/A	N/A
8	Receiver input resistance	3.7 k•	4 k•	12 k•
9	Driver load impedance	3.7 k•	100 •	54 •
10	Receiver input sensitivity	±3 V	±200 mV	±200 mV
11	Receiver input range	±15 V	±10 V	-7.12 V
12	Max driver output voltage	±25 V	±6 V	-7.12 V
13	Min driver output voltage (with load)	±5 V	±2.0 V	±1.5 V

Other than the interface it is necessary to exercise control of the flow of data between two devices which are communicating. For this there are a set of rules termed as PROTOCOL, which are followed by both the communicating devices.Two widely used Serial communication protocols are MODBUS-RTU and MODBUS-ASCII.

### 5.2.1 MODBUS

MODBUS is basically an open, serial communication protocol based on the master/slave architecture developed by Modicon in 1979. The bus consists of a master station, controlling the communication, and of a number of slave stations.

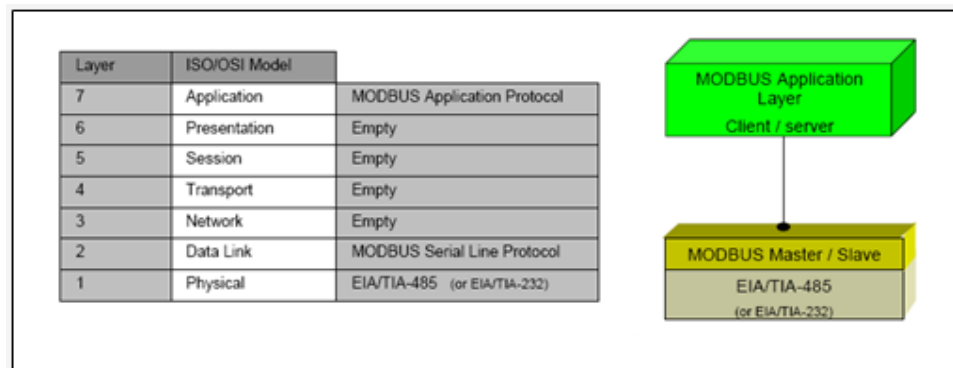


MODBUS is an application layer messaging protocol, positioned at level 7 of the OSI model that provides client/server communication between devices connected on different types of buses or networks. MODBUS is used to monitor and program devices; to communicate intelligent devices with sensors and instruments; to monitor field devices using PCs and HMIs. MODBUS is an ideal protocol for RTU applications where wireless communication is required.

Basically Modbus has been implemented and used over all types of physical links (wire, fibre, and radio) However the two basic types of implementations are:

1. **Serial** (Asynchronous Master/Slave)
2. **TCP/IP** (Client/Server)

Fig.5.1 gives a general representation of MODBUS serial communication stack compared to the 7 layers of the OSI model.



**Fig.5.1 MODBUS ISO/OSI Model**

MODBUS application layer messaging protocol, positioned at level 7 of the OSI model, provides client/server communication between devices connected on buses or networks. On MODBUS serial line the client role is provided by the Master of the serial bus and the Slaves nodes act as servers.

### 5.2.2 Communication between MODBUS devices

MODBUS devices communicate using a master-slave technique in which only one device (the master) can initiate transactions (called queries). The other devices (slaves) respond

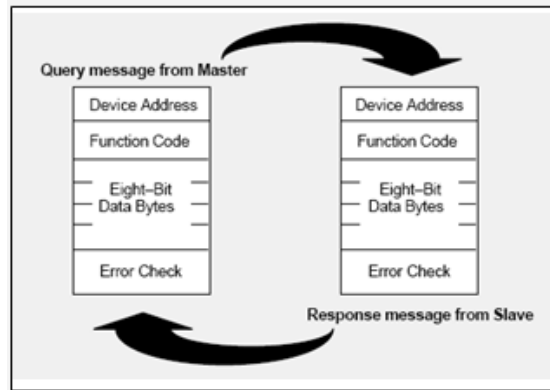
by supplying the requested data to the master, or by taking the action requested in the query. Only one master (at the same time) is connected to the bus, and one or several (247 maximum number) slaves nodes are also connected to the same serial bus. A slave is any peripheral device (I/O transducer, valve, network drive, or other measuring device), which processes information and sends its output to the master using MODBUS. Masters can address individual slaves (unicast mode), or can initiate a broadcast message to all slaves. Slaves return a response to all queries addressed to them individually, but do not respond to broadcast queries. All devices must accept the broadcast for writing function. The address 0 is reserved to identify a broadcast exchange. The slave nodes never communicate with each other.

All communications activity on the loop occurs in the form of "PACKETS", a packet being simply a serial string of 8-bit bytes. The maximum number of bytes contained within one packet is 255.

### **5.2.3 The Query-Response Cycle**

**(a) The Query:** The function code in the query tells the addressed slave device what kind of action to perform. The data bytes contain any additional information that the slave will need to perform the function. For example, function code 03 will query the slave to read holding registers and respond with their contents. The data field must contain the information telling the slave which register to start at and how many registers to read. The error check field provides a method for the slave to validate the integrity of the message contents.

**(b) The Response:** If the slave makes a normal response, the function code in the response is an echo of the function code in the query. The data bytes contain the data collected by the slave, such as register values or status. If an error occurs, the function code is modified to indicate that the response is an error response, and the data bytes contain a code that describes the error. The error check field allows the master to confirm that the message contents are valid. This is depicted in Fig.5.2

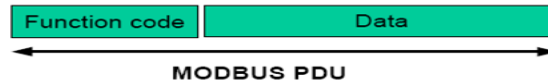


**Fig.5.3 The Query-Response Cycle of MODBUS protocol**

The MODBUS addressing space comprises 256 different addresses.

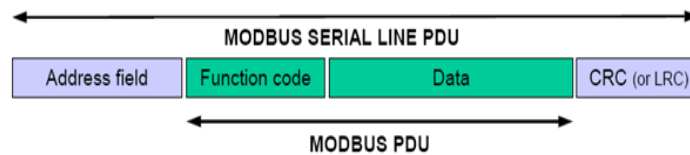
0	From 1 to 247	From 248 to 255
Broadcast address	Slave individual addresses	Reserved

The Address 0 is reserved as the broadcast address. All slave nodes must recognise the broadcast address. The MODBUS Master node has no specific address, only the slave nodes must have an address. This address must be unique on a MODBUS serial bus. The MODBUS application protocol defines a simple **Protocol Data Unit (PDU)** independent of the underlying communication layers as shown in Fig.5.3.



**Fig.5.3 Modbus Protocol Data Unit**

The mapping of MODBUS protocol on a specific bus or network introduces some additional fields on the **Protocol Data Unit**. The client that initiates a MODBUS transaction builds the MODBUS PDU, and then adds fields in order to build the appropriate communication PDU as shown in Fig.5.4.



**Fig.5.4 General Modbus Message frame over serial line**

On MODBUS Serial Line, the **Address field** only contains the slave address in the range of 1-247 either in the form of two characters (in ASCII mode), or 8 binary bits (in RTU Mode). A master addresses a slave by placing the slave address in the address field of the message. When the slave returns its response, it places its own address in the response address field to let the master know which slave is responding. The **function code** indicates to the server what kind of action to perform. The function codes supported by MODBUS RTU are listed in table 5.2. The function code can be followed by a **data field** that contains request and response parameters. **Error checking** field is the result of a "Redundancy Checking" calculation that is performed on the message contents. Two kinds of calculation methods are used depending on the transmission mode that is being used (RTU or ASCII).

**Table 5.2 Function Codes supported by MODBUS RTU slave**

Function Code	Register Type	Max. No. of coils/Registers
1	Read Coil status	2000
2	Read Discrete Input status	2000
3	Read Holding Registers	125
4	Read Input Registers	125
5	Write Single Coil	1
6	Write Single Holding Register	1
15	Write Multiple Coils	800
16	Preset Multiple Holding Registers	100

The maximum number of coils or registers that may be specified or included in a single MODBUS message is also mentioned in the above table. MODBUS devices usually include a Register Map. MODBUS functions operate on register map registers to monitor, configure, and control module I/O. One should refer to the register map for the concerned device to gain a better understanding of its operation.

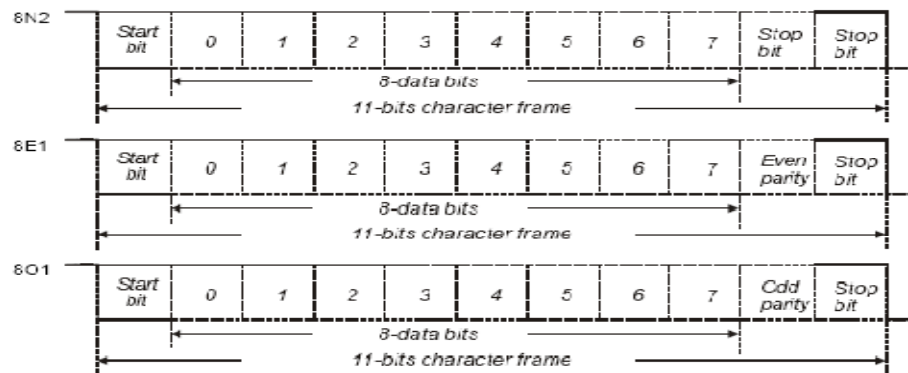
#### **5.2.4 Serial Transmission Modes of Modbus Networks**

The transmission mode defines the bit contents of the message bytes transmitted along the network, and how the message information is to be packed into the message stream and decoded. The mode of transmission is usually selected along with other serial port communication parameters (baud rate, parity, etc.) as part of the device configuration.

There exist two transmission modes, which differ in encoding, framing and checksum:

(a) **Modbus ASCII (American Standard Code for Information Interchange)**

Each 8-bit byte in a message is sent as two ASCII characters, representing the hexadecimal notation of the byte (i.e. characters 0–9, A–F). The error checksum is represented by a longitudinal redundancy check (LRC; 1 byte) and messages start with a colon (:, 3AH), and end with a carriage return – line feed pair ("CRLF", 0D0AH). The main advantage of this mode is that it allows time intervals of up to one second to occur between characters without causing an error. The 8-bit data frame and the complete message frame format for this mode are depicted in Fig. 5.6 and Fig.5.7 respectively.



**Fig.5.5 The Data Frame (For 8-bit data)**

START	ADDRESS	FUNCTION	DATA	LRC CHECK	END
1 CHAR :	2 CHARS	2 CHARS	n CHARS	2 CHARS	2 CHARS CRLF

**Fig.5.6 The MODBUS-ASCII Message frame**

**(b) Modbus RTU (Remote terminal Unit)**

Each 8-bit data in a message contains two 4-bit hexadecimal characters. The message frame as shown in Fig. 5.8 has an error checksum represented by a cyclic redundancy check (16 bit CRC; 2 byte) and messages start and end with a silent interval of at least 3.5 character times. This is most easily implemented as a multiple of character times at the baud rate that is being used on the network (shown as T1-T2-T3-T4). The maximum pause that may occur between two bytes is 1.5 character times. The main advantage of this mode is that its greater character density increases throughput over ASCII mode at the same baud rate. Networked devices monitor the network bus continuously, including

during the 'silent' intervals. When the first field (the address field) is received, each device decodes it to find out if it is the addressed device.

START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1-T2-T3-T4	8 BITS	8 BITS	$n \times 8$ BITS	16 BITS	T1-T2-T3-T4

**Fig.5.7The MODBUS-RTU Message Frame**

The entire message frame must be transmitted as a continuous stream. If a silent interval of more than 1.5 character times occurs before completion of the frame, the receiving device flushes the incomplete message and assumes that the next byte will be the address field of a new message.

### **5.3 ETHERNET**

Ethernet is the most widely implemented LAN architecture which has been popular for several decades as it is relatively inexpensive and reasonably fast. It was developed by two individuals at Xerox PARC -- Bob Metcalfe and D.R. Boggs in 1972 and specifications based on this work appeared in IEEE 802.3 in 1980.

An Ethernet LAN typically uses coaxial cable, twisted pair wires or more recently optical fibres. Ethernet is also used in wireless LANs. The most commonly installed Ethernet systems are called 10BASE-T and provide transmission speeds up to 10 Mbps. Fast Ethernet or 100BASE-T provides transmission speeds up to 100 megabits per second and is typically used for LAN backbone systems, supporting workstations with 10BASE-T cards. Gigabit Ethernet provides an even higher level of backbone support at 1000 megabits per second (1 gigabit or 1 billion bits per second). 10-Gigabit Ethernet provides up to 10 billion bits per second.

Ethernet uses the CSMA/CD access method to regulate communication among the nodes. Devices are connected to the cable and compete for access using a Carrier Sense Multiple Access with Collision Detection (CSMA/CD) technique. In other words, CSMA/CD is a set of rules determining how network devices respond when two devices attempt to use a data channel simultaneously (called a collision). This standard enables devices to detect a collision. After detecting a collision, a device waits a random delay time and then

attempts to re-transmit the message. If the device detects a collision again, it waits twice as long to try to re-transmit the message.

Ethernet basically uses a bus, star or tree topology.

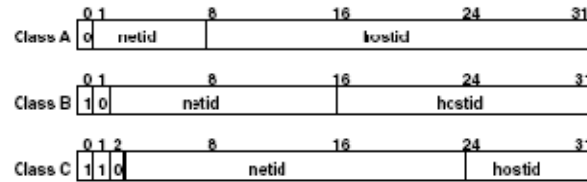
- Ø **Bus topology:** All devices are connected to a central cable, called the bus or backbone. Bus networks are relatively inexpensive and easy to install for small networks. Ethernet systems use a bus topology.
- Ø **Star topology:** All devices are connected to a central hub. Star networks are relatively easy to install and manage, but bottlenecks can occur because all data must pass through the hub.
- Ø **Tree topology:** A tree topology combines characteristics of linear bus and star topologies. It consists of groups of star-configured workstations connected to a linear bus backbone cable. These topologies can also be mixed. For example, a bus-star network consists of a high bandwidth bus, called the backbone, which connects collections of slower-bandwidth star segments.

Most manufacturers who offer Ethernet compatibility to implement supervisory functions over equipment controlling plant floor functions use a transmission control protocol/internet protocol (TCP/IP) for layers 3 and 4 of the OSI model. Some PLC manufacturers offer programmable controllers with TCP/IP over-Ethernet protocol built into the PLC processor. This allows the PLC to connect directly to a supervisory Ethernet network.

### **5.3.1 IP Addressing**

Each TCP/IP **node** on a network must have a unique **IP address**. The TCP/IP Ethernet Interface is such a node, as is a PC running TCP/IP. There may be other nodes on the network that are not involved with communications to the PLCs, but no matter what their function, each TCP/IP node must have its own IP address. It is the IP address that identifies each node on the IP network (or system of connected networks). The term “host” is often used to identify a node on a network. The IP

address is 32 bits long and has a netid part and a hostid part. Each network is a Class A, Class B or Class C network.



**Fig.5.8 IP Address Format**

In general, the **netid** part is assigned by the Internet authorities and the **hostid** part is assigned by the local network administrator. The class of network determines the number of hosts that can be supported. A Class A network can support  $2^{24} - 2$  (16,777,214) hosts, Class B,  $2^{16} - 2$  (65,534) hosts, and Class C,  $2^8 - 2$  (254) hosts. The minus 2 refers to host numbers reserved for the network itself and the local broadcast. Each node on the same physical network must have an IP address of the same class and must have the same netid. Each node on the same physical network must have a different hostid thus giving it a unique IP address.

IP addresses are written in “dotted-decimal” format as four decimal integers (0-255) separated by periods where each integer gives the value of one byte of the IP address. For example, the 32-bit IP address

00001010 00000000 00000000 00000001 is written as 10.0.0.1

One can determine the class of an IP address by examining the first integer in its dotted-decimal IP address and comparing with the range of values in table 5.3

**Table 5.3 Identifying class of an IP Address**

Range of first integer	Class
0-127	A
128-191	B
192-223	C
224-239	D (Reserved for Multicast Use)
240-255	E ( Reserved for Experimental Use)

### 5.3.2 Description of the Present Network Configuration

Ethernet connection exists between PC, PLC and HMI. The Series 90-30 PLC CPU374 module and the QuickPanel unit contains an embedded TCP/IP Ethernet



Interface with an Auto-sensing 10 Base T / 100 Base TX, RJ-45 Port . Other characteristics are summarized in the table 5.4.

**Table 5.4 Ethernet Configuration used in the Project**

<b>Port/connector</b>	RJ-45 ( 2 in PLC , 1 in HMI)
<b>Speed</b>	10 Base T/ 100 Base Tx ( 10 Mbps/100 Mbps)
<b>Type of Network cable used</b>	Twisted Pair
<b>Cable length</b>	Upto 100 m (PLC), Upto 30 m (HMI)
<b>Transmission mode</b>	Duplex ( Half/Full)
<b>IP Address</b>	192.168.1.3 (PLC), 192.168.1.3 (HMI),192.168.1.5 (PC)
<b>Topology</b>	Star
<b>No. of Hubs</b>	One
<b>No. of Nodes</b>	4
<b>No. of Transceivers used</b>	Nil
<b>Protocol used</b>	SRTCP

#### **5.4 CONCLUSION**

This chapter summarises the communication ports, standards and **protocols** used between the various components of the present set-up including the PC – PLC – ServoDrive – Servomotor – SCADA and HMI . It was seen that mostly serial communication and Ethernet were employed as the communication channels, Ethernet offering a much higher speed.

## CHAPTER VI EXPERIMENTAL RESULTS AND DISCUSSIONS

### 6.1 GENERAL

This chapter summarises the complete experimental work done on the system hardware and software, the observations made and a brief discussion on their results. The experimental work involved development of ladder programs for the control of Servomotor (PMSM), running it in various modes and capturing its runtime characteristics.

### 6.2 SOFTWARE IMPLEMENTATIONS

#### 6.1.1 Ladder Programs for control of Permanent Magnet Synchronous Motor using Proficy Machine Edition 5.90

##### 6.1.1.1 Ladder Logic to Start and Stop a Servo Motor

This program starts a servomotor and after a specified period of time (set by timer R00005) automatically stops the motor. The motor then restarts after a specified period of time (set by timer R00001). Fig. 6.1 shows the flowchart for this program. Since we are running the motor in the internal speed control mode Q3 (Digital Input 3) and Q4(Digital Input 4) terminals of the PLC output module are assigned SPD0 and SPD1 respectively. We are running the motor at the speed setting configured by P1-09 for which Q3=1 and Q4=0. So only Q3 is shown in the ladder program. Q1 (Digital Input 1) has to be turned on at the start of every program to ensure that control power has been supplied to the servodrive and it is ready to run.

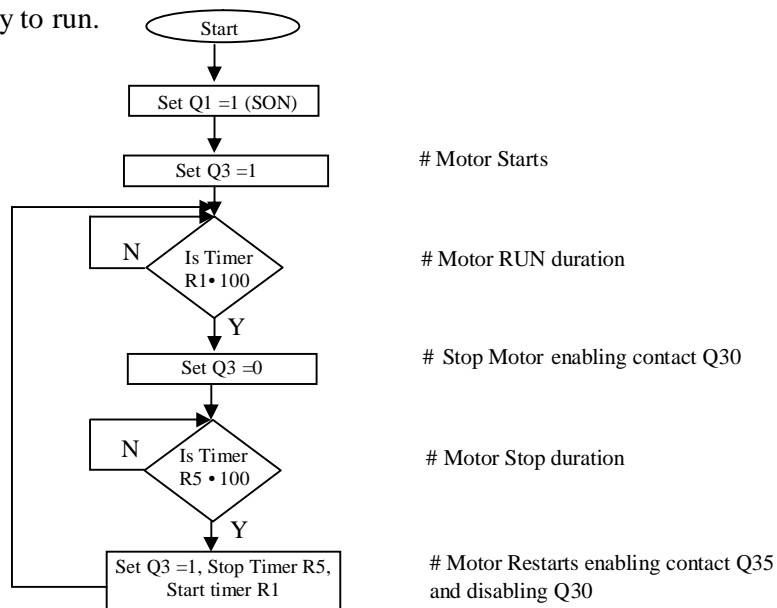


Fig. 6.1 Flowchart for start and stop operation of PMSM

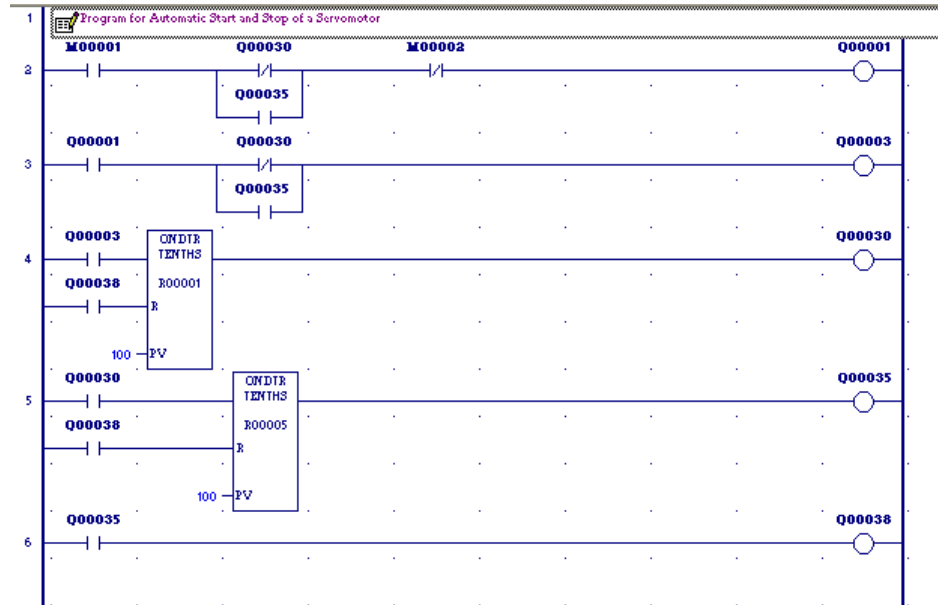


Fig.6.2 Ladder logic for Start and Stop of Servomotor

### 6.1.1.2 Ladder Logic to Run Servomotor in Internal Position Control Mode

This Program is designed to enable the user to obtain five different positions (Internal Positions)of the Servo Motor in succession, with a predefined time interval between the various positions.The positions chosen are from Position 8 to Position 4.The settings for the various desired positions can be set using the **Parameter Editor Window** of VersaMotion Configuration software. Also the DI and DO settings are to be done as shown in table 6.1.The position settings configured in the table are explained as follows:

- Ø Position 8 setting corresponds to 4 rotations at a speed of 100 rpm.
- Ø Position 7 setting corresponds to a movement of 90 degrees at a speed of 100 rpm from the position last reached by the motor in Position 8.
- Ø Position 6 corresponds to 3 complete rotations and a further movement of 45 degrees corresponding to the 1250 pulses at a speed of 200 rpm
- Ø Position 5 and 4 have been set for 1 and six rotations respectively at speeds of 100 rpm and 200 rpm respectively.

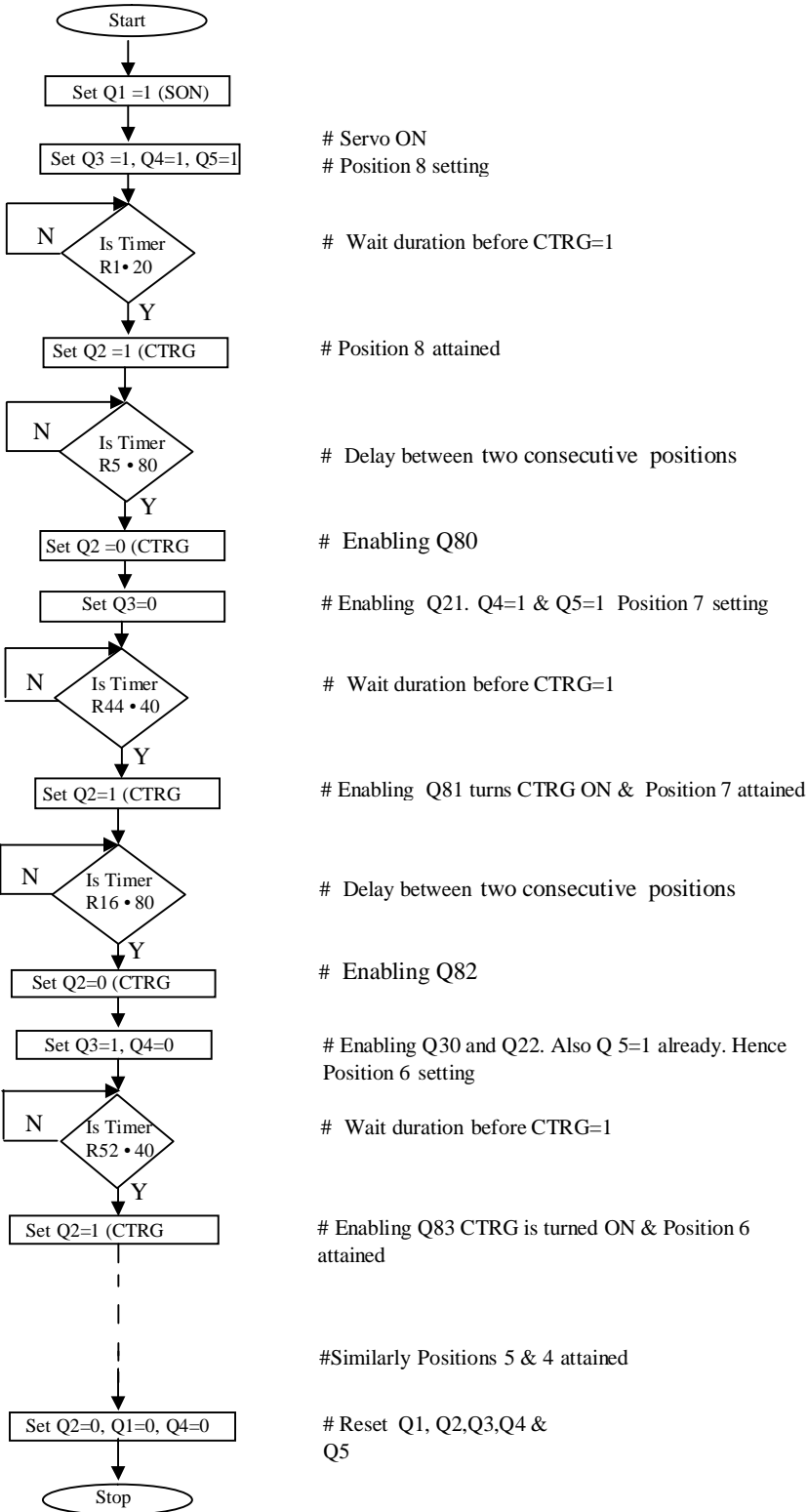
**Table 6.1 Parameter settings for the Internal position control mode**

	Pos 0 (Q3)	Pos 1 (Q4)	Pos 2 (Q5)	CTRG (Q2)	Parameter setting	Position settings	Speed setting for each position (rpm)
Internal Pos 8	1	1	1	•	P1-29	4	P2-43=100
					P1-30	0	
Internal Pos 7	0	1	1	•	P1-27	0	P2-42=100
					P1-28	2500	
Internal Pos 6	1	0	1	•	P1-25	3	P2-41=200
					P1-26	1250	
Internal Pos 5	0	0	1	•	P1-23	1	P2-40=200
					P1-24	0	
Internal Pos 4	1	1	0	•	P1-21	6	P2-39=100
					P1-22	0	

- Ø The incremental encoder that is installed within the Servomotor set up has a resolution of 10,000 quadrature counts per revolution which implies a resolution of 0.036 degrees at the shaft. That is one pulse (set by the even parameters say P2-28) corresponds to a rotation of 0.036 degrees and hence 2500 pulses correspond to a rotation of 90 degrees).
- Ø Remember to set Parameter P1-33 1 ( For Incremental Encoder).

As seen from table 6.1, for internal position control mode Q2, Q3, Q4 and Q5 terminals of the PLC output module are configured as Digital Inputs CTRG, POS0, POS1 and POS2 respectively. Hence to attain any one of the positions from Pos 4 to Pos 8 ( in the present program) we need to set the corresponding Digital inputs for that position as mentioned in table 6.1. For eg. for Pos 8 the settings have to be Q3=Q4=Q5=1 and then a control trigger has to be given in the program by making Q2=1. After the desired position is attained and a new position is desired, then Q2 has to be first reset and then again set as Q2=1 to provide a new trigger, after the Q3-Q5 settings of the new position are done. A fixed amount of delay is provided between two consecutive positions using ON-Delay timers (ONDTR). Normal timers (TMR) could have also provided the required delay , but it was found that to reset various contacts, which is required at many steps in the program, the RESET input of the ONDTR proved to be quite useful. Although at some places TMRs are also used . The detailed flowchart and the ladder

program for the internal position control mode have been shown in Fig. 6.3 and Fig.6.4(a-c) respectively.



Note: Similar Pattern follows for Position 5 and 4 also. Hence they are not shown

Fig. 6.3 Flowchart for internal Position control mode of Servomotor

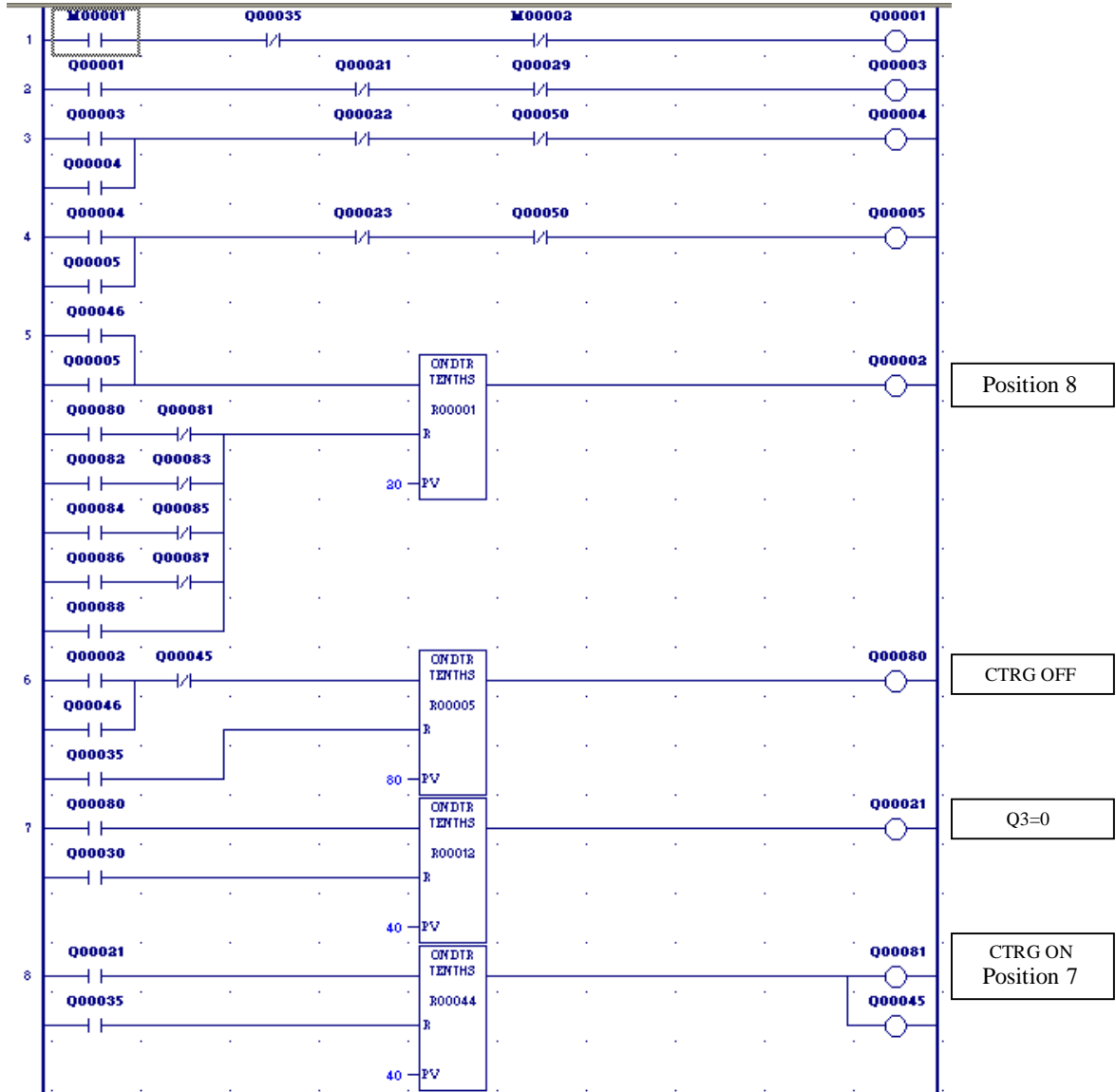


Fig. 6.4 (a) Ladder Logic to operate a Servomotor in Internal Position control Mode

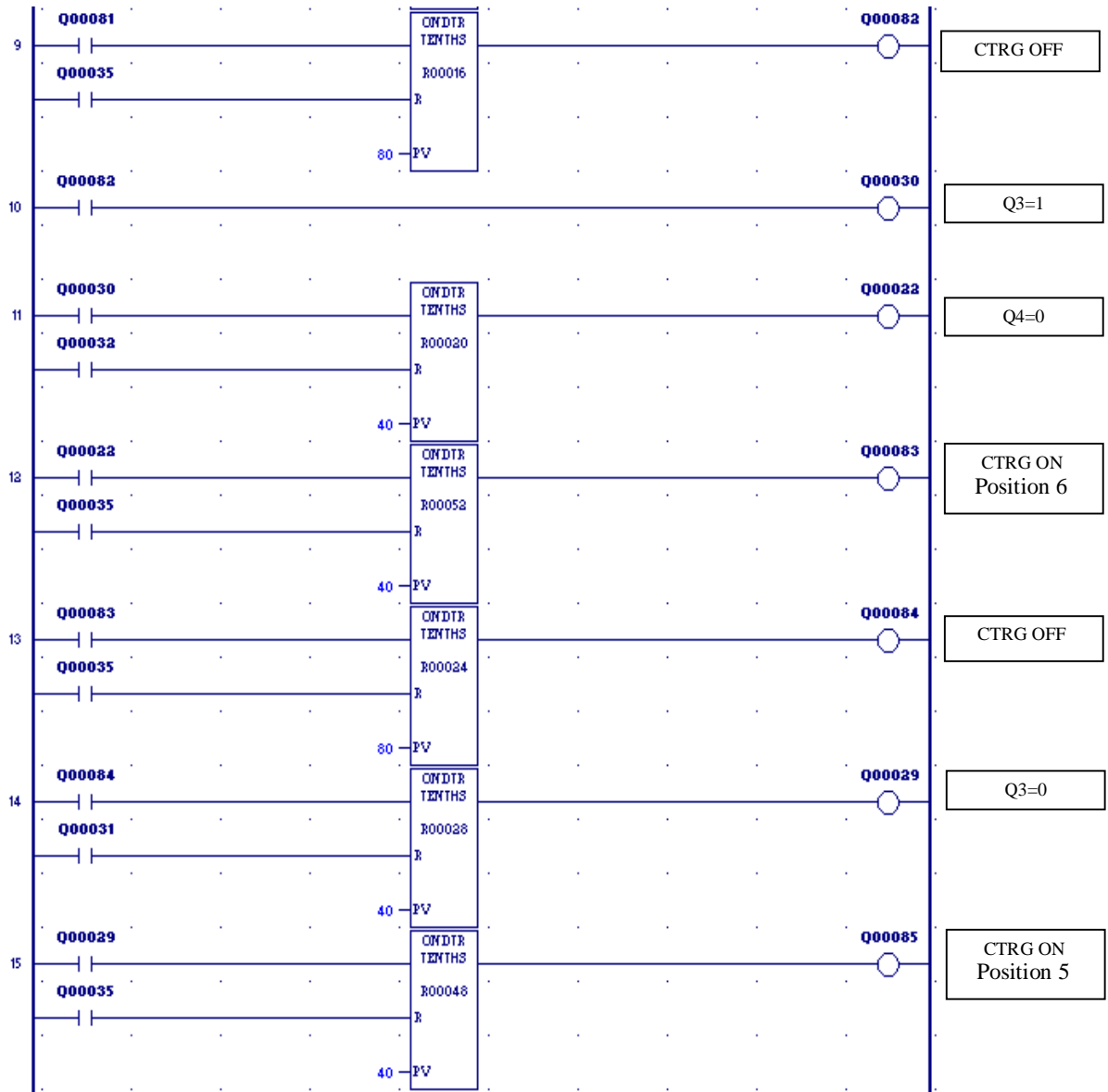


Fig. 6.4 (b) Ladder Logic to run a Servomotor in Internal Position control Mode

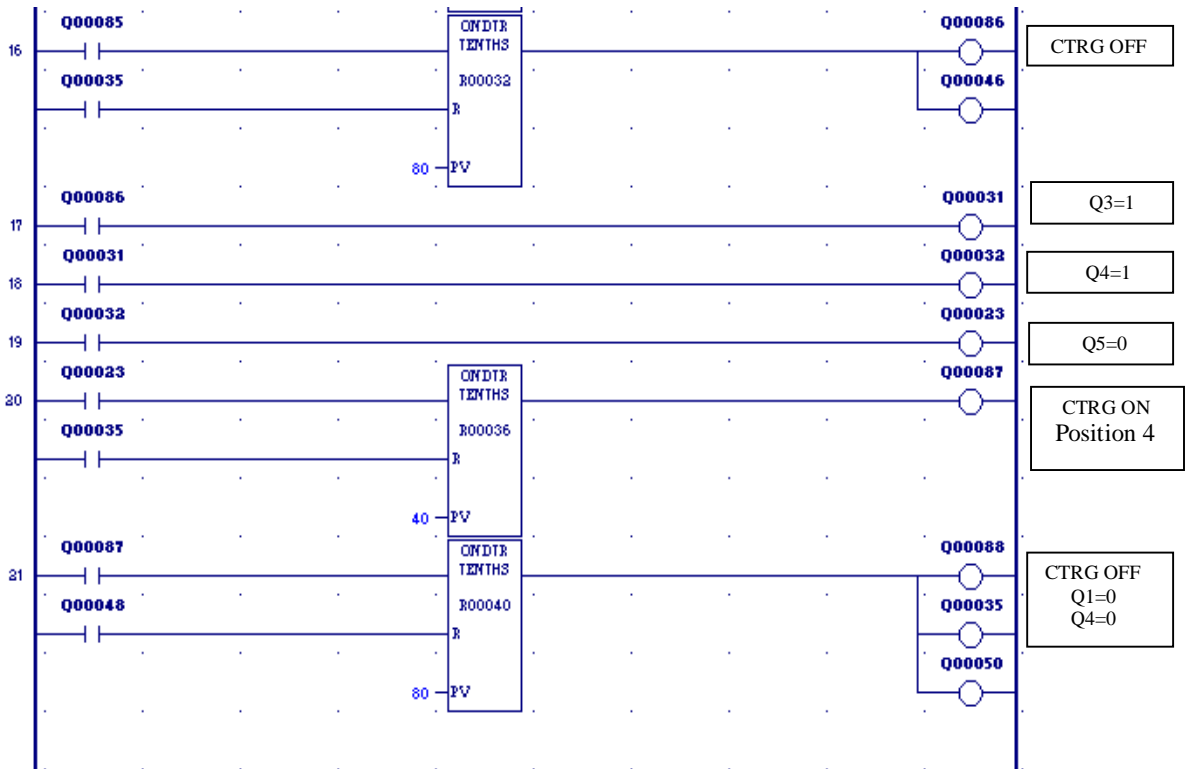


Fig. 6.4 (c) Ladder Logic to run a Servomotor in Internal Position control Mode

### 6.1.1.3 Ladder Logic for Internal Speed control Mode

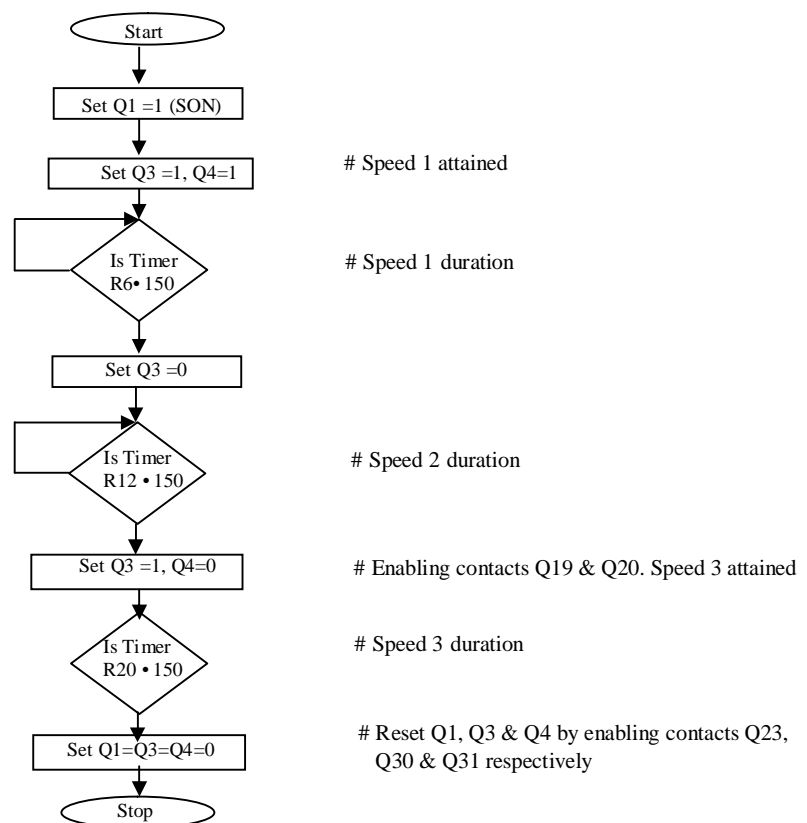
This program enables the user to obtain three different speeds (Internal speed ) in succession for a predefined period. The various speed settings can be done using the VersaMotion Configuration Software based on the user requirements. Also set the Digital Inputs P2-10 to P2-17 and the Digital Outputs P2-18 to P2-22 for the Speed mode (table 4.4 and 4.5). On doing so we get Q3 and Q4 as the Digital Inputs SPD0 and SPD1 respectively so they are used in the Program. Speed settings are done as shown in table 6.2.

Table 6.2 Speed Control Mode Parameters

	SPD 0 (Q3)	SPD 1 (Q4)	Parameter setting	Speed (rpm)
Internal Speed 3	1	1	P1-11	1500
Internal Speed 2	0	1	P1-10	3000
Internal Speed 1	1	0	P1-09	2000



The flowchart and the ladder program to run the servomotor in three different speed settings, in the internal speed control mode has been shown in Fig. 4.5 and Fig.4.6 respectively. As explained in 6.1.1.2, the ladder logic shows that to attain different speeds, the Digital inputs SPD0 (Q3) and SPD1 (Q4) have to be set as per table 6.2 and unlike the position control mode there is no additional trigger input which has to be activated. Three On-Delay Timers (ONDTR) have been used to provide a specific delay between two consecutive modes.



**Fig. 6.5 Flowchart to run the servomotor in internal speed control mode**

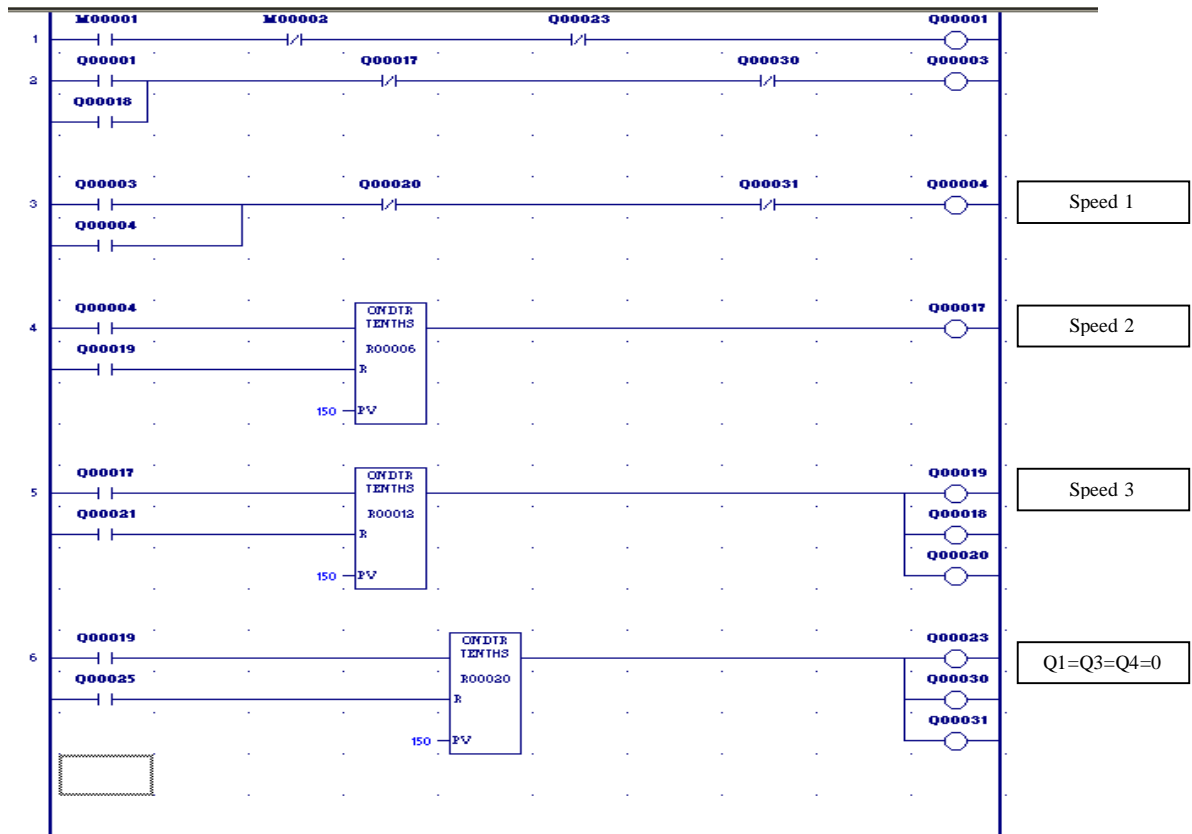


Fig. 6.6 Ladder Logic to run the servomotor in internal speed control mode

## 6.2.2 LADDER PROGRAMS FOR SERVO MOTOR CONTROL USING CIMPLICITY/HMI SCADA 7.5 EDITION

These programs are also written using Proficy Machine Edition 5.9 and then run simultaneously from the SCADA screen developed using Cimplicity/HMI 7.5 software.

### 6.2.2.1 Ladder Logic to develop animated screen for Start and Stop of Servomotor

This Program enables the user to control the Start and stop of the Servomotor through the graphics created on the computer screen (Fig. 6.7) using the SCADA software Cimplicity/HMI Machine Edition 7.5. This type of motor control is helpful when the operator has to control the motor from a remote location directly through the screen of the workstation. The motor is run in the internal speed mode by configuring the necessary parameters (table 4.4, 4.5 and 4.6). The ladder logic developed is similar to the

explanation given in 6.2.1.1 with some additional rungs added to attain the animation of the motor on the PC screen (TMR R00007) and has been shown in Fig.6.8.

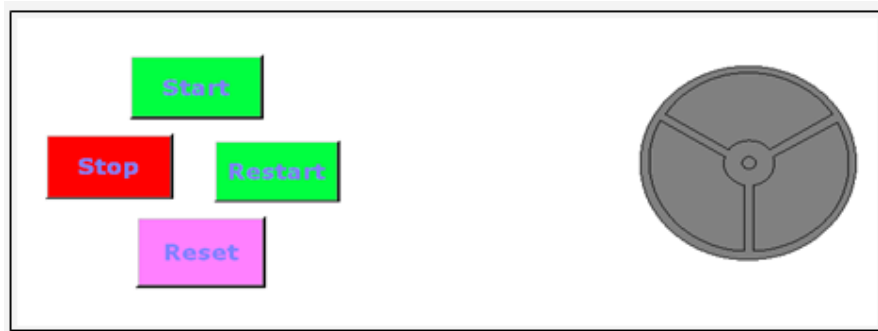


Fig.6.7 SCADA/HMI based Screen Developed for Motor ON/OFF Control

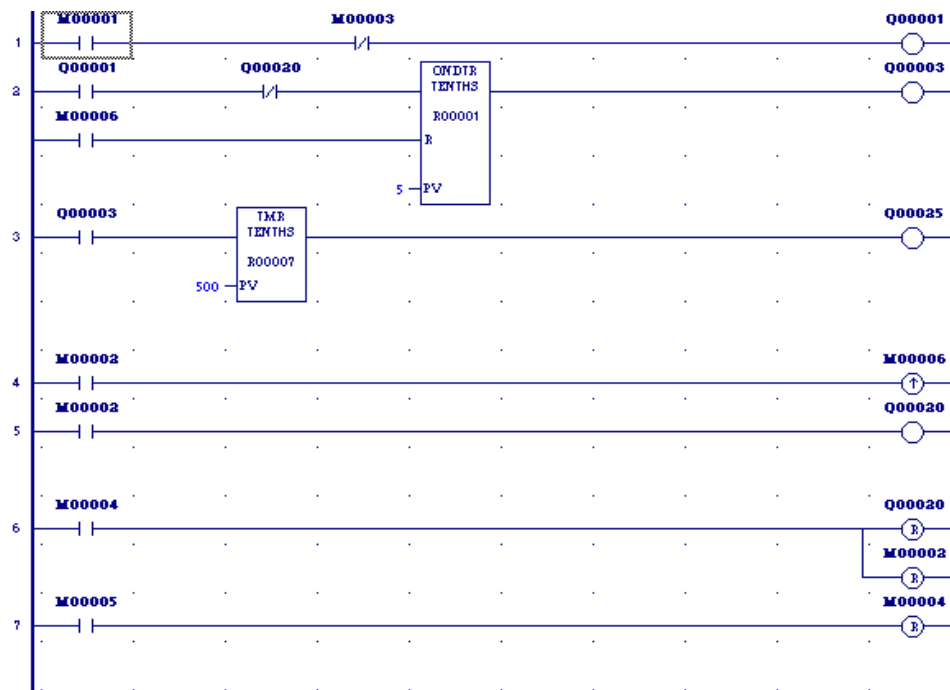


Fig.6.8 Ladder logic for the SCADA/HMI based Screen for Motor ON/OFF Control

### 6.2.2.2 Bottle Filling, Capping and Labelling Application - A Case Study of PLC/SCADA based Virtual Monitoring System

This Program shows the implementation of the position control program discussed in section 6.2.1.2. The program is used for developing an animated screen (Fig. 6.9) in synchronism with the servomotor for showing the applications of filling, capping and labelling of bottles actually happening in industries. In this case the Servomotor is

assumed to be running the Conveyor belt carrying the bottles. The program has been used for three different positions of the Servomotor, the desired number of rotations being set for each position to achieve precise filling, capping and labelling of the bottles. In the present case only one bottle has been shown going through the three different stages but in practice bulks of such bottles are processed in batches in the industries. Certain additions have been made to the ladder logic of Fig.6.4(a-c) control program to obtain the necessary animations on the SCADA software. The ladder logic is given in Fig.6.10 (a-c).

The ladder logic is developed for attaining three different positions (Pos 8 to Pos 6). All three positions are set to different number of motor rotations as per table 4.11. The number of rotations should be such that the Pos 8 setting rotates the motor driving the conveyor in such a manner that after completely attaining Pos 8, the motor stops and so does the conveyor and the bottle is brought at the filling end. After being sensed by a position sensor a filling mechanism is activated (not covered in detail in the present case) which fills the bottle to the required level. In the meantime the trigger input CTRG (Q2) is deactivated to prepare for the next position. An appropriate delay time is set in the program between two consecutive positions. Then after the Digital inputs Q3, Q4 and Q5 are set for Pos 7, CTRG is again set and the motor tries to follow the Pos 7 setting. The number of rotations for Pos 7 should be such that after completely attaining Pos 7 the filled bottle stops at the capping end. Again after being sensed by an appropriate sensor the capping lever is pulled down (mechanism not covered in detail here). Similarly at the end of Pos 6 the bottle is brought under the labelling head.

Hence for such applications it is the PMSM characteristics such as high resolution, low moment of inertia, lower torque ripple, a higher-torque-to-inertia ratio and a high torque to current ratio that enable it to be used in such precise position applications.

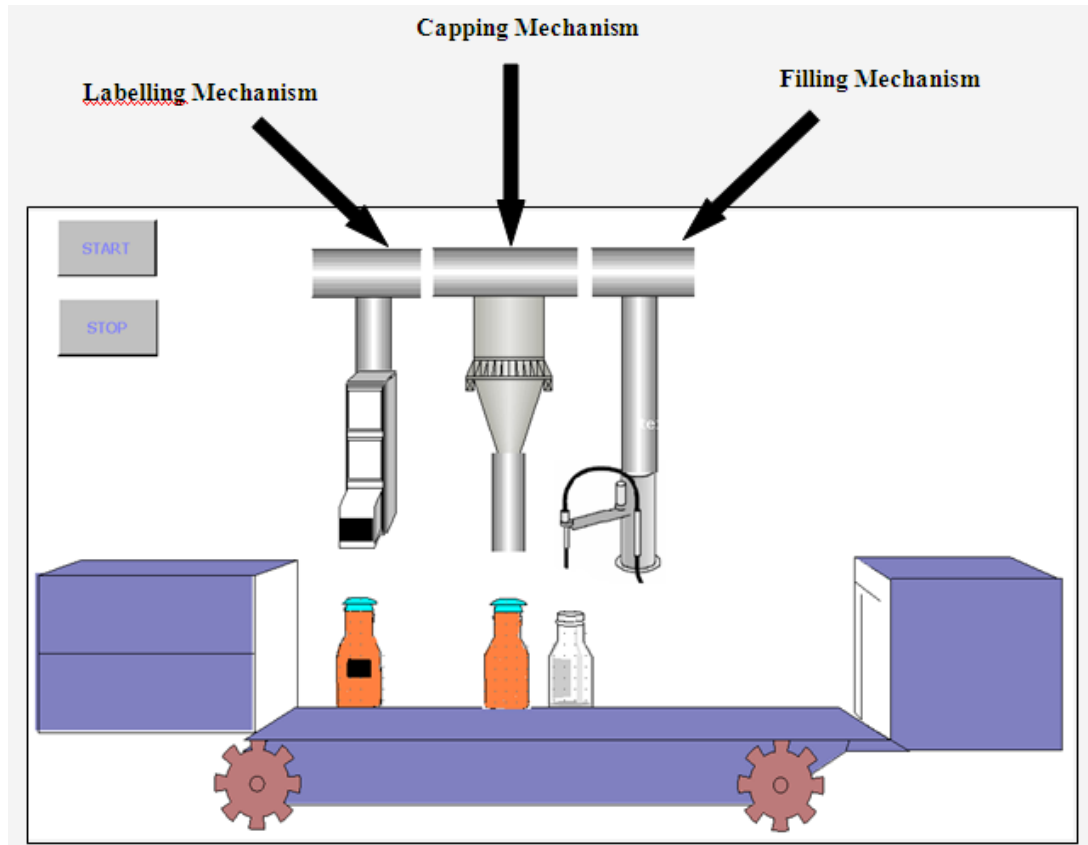
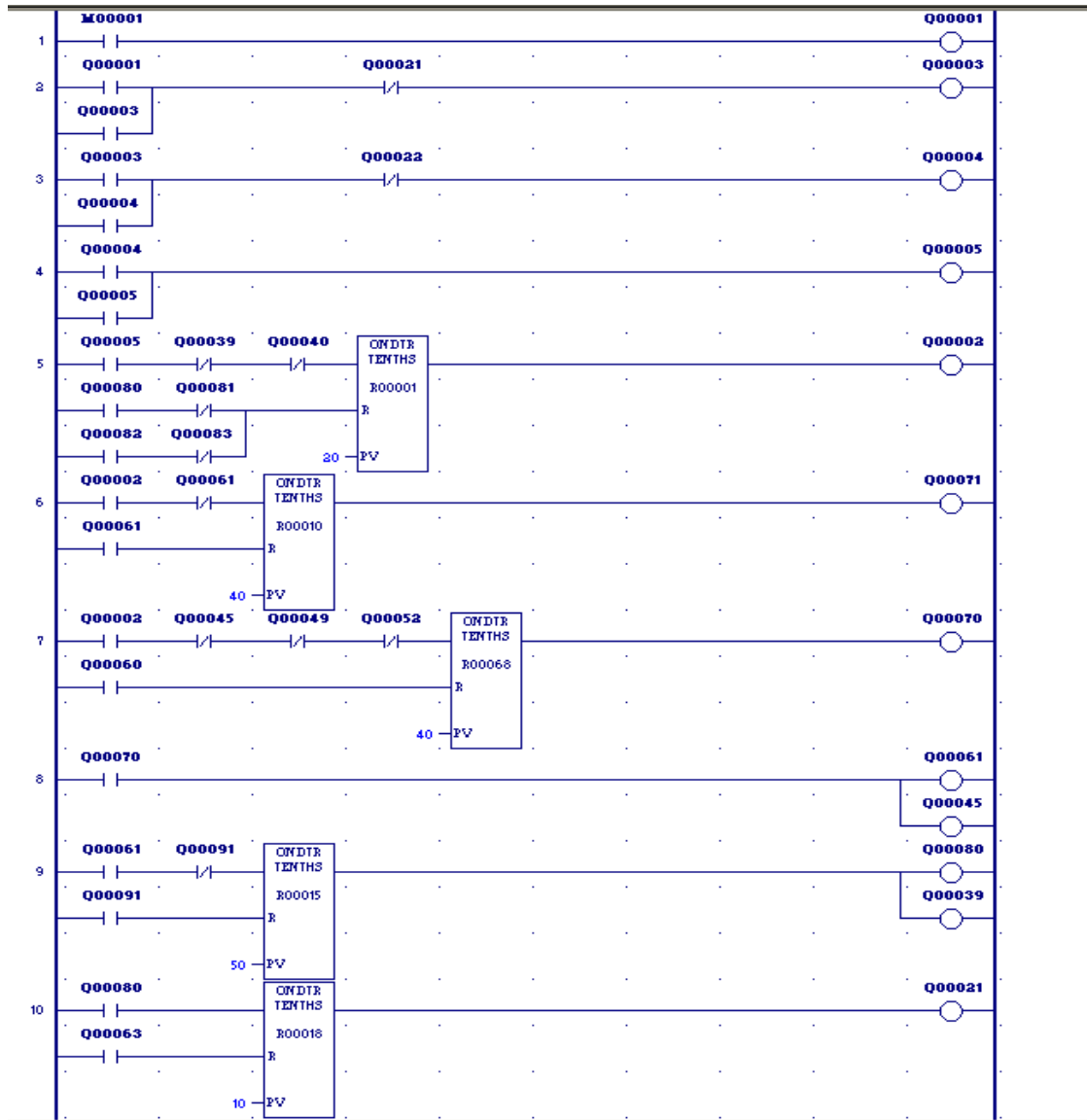
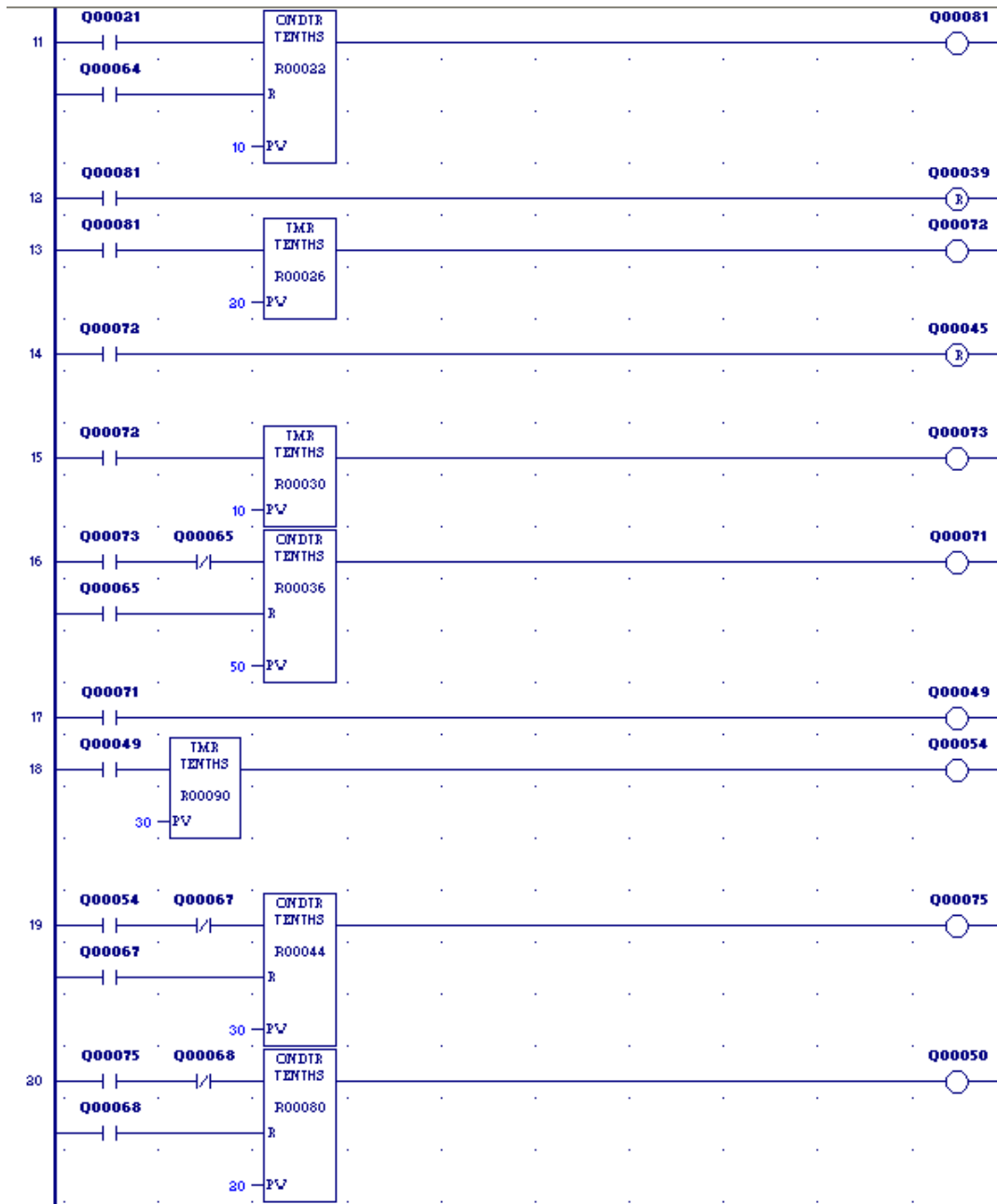


Fig.6.9 SCADA screen showing Animated Bottle Filling, Capping and Labelling Application



**Fig.6.10(a) Ladder logic for the developing a SCADA screen showing Animated Bottle Filling, Capping and Labelling Applications**



**Fig.6.10(b) Ladder logic for the developing a SCADA screen showing Animated Bottle Filling, Capping and Labelling Applications**

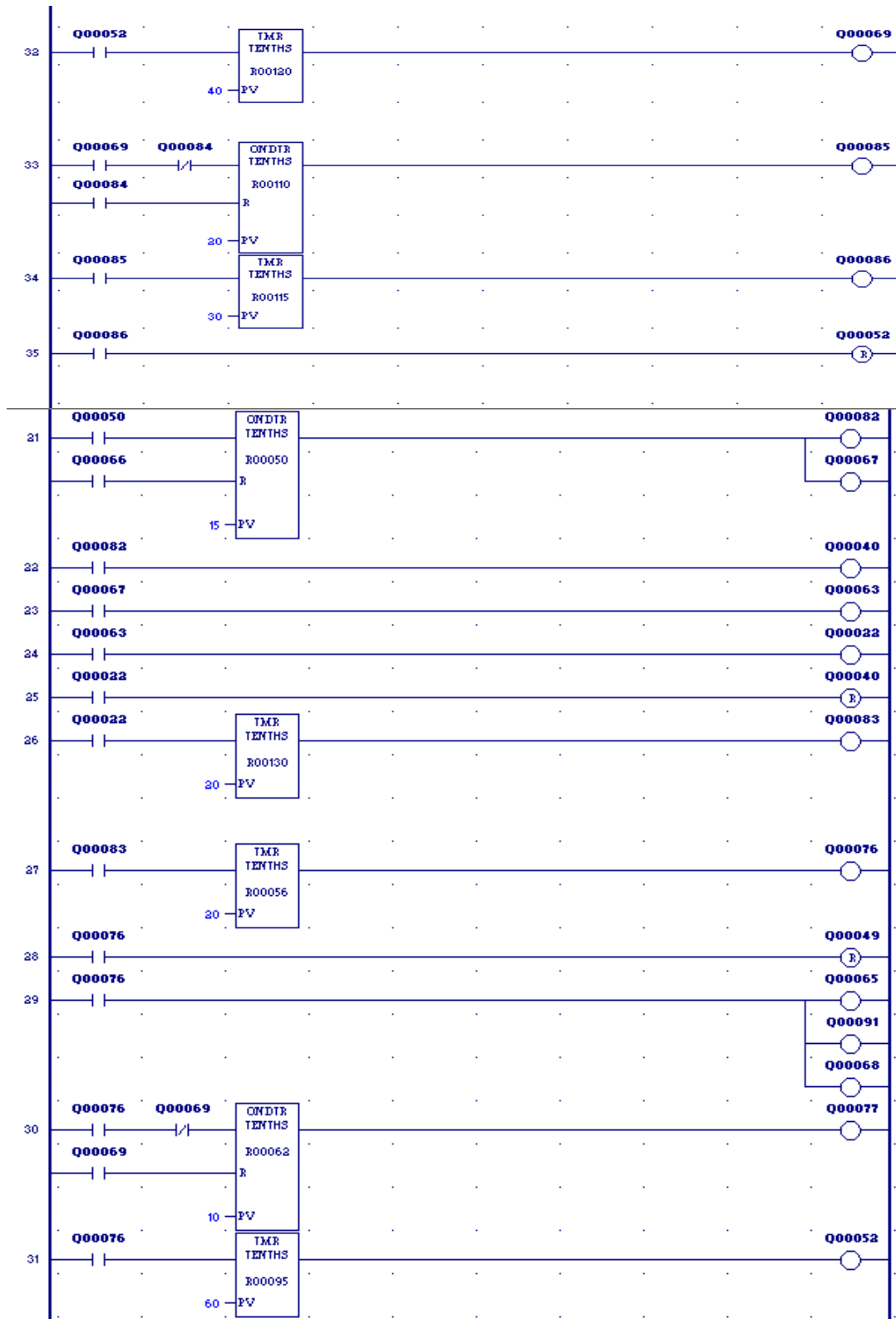


Fig.6.10(c) Ladder logic for the developing a SCADA screen showing Animated Bottle Filling, Capping and Labelling Applications



### 6.2.2.3 Test Tube Filling Application – A Case Study of PLC/SCADA based Virtual Monitoring System

A virtually animated screen is developed for a Test Tube filling application (Fig. 6.11), usually found in pharmaceutical companies. In this the servomotor is assumed to be rotating the disc carrying the test tubes. It is also a position control kind of an application. The ladder logic is shown in Fig. 6.12.

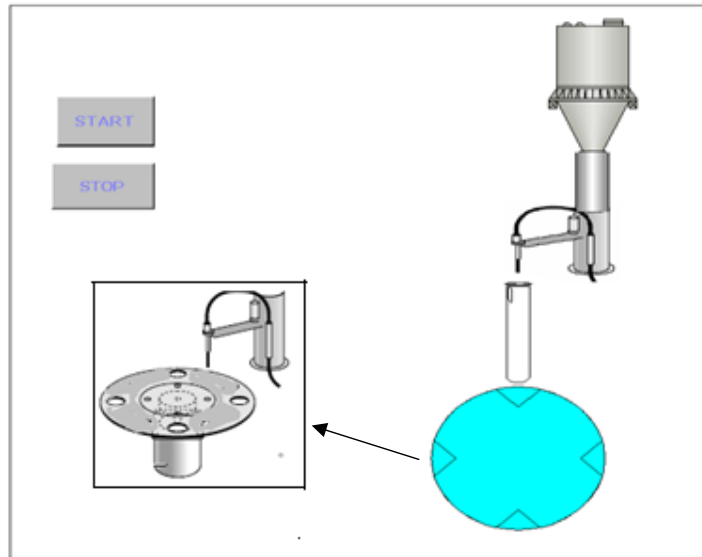


Fig.6.11 SCADA Screen showing an imated Test Tube filling Application

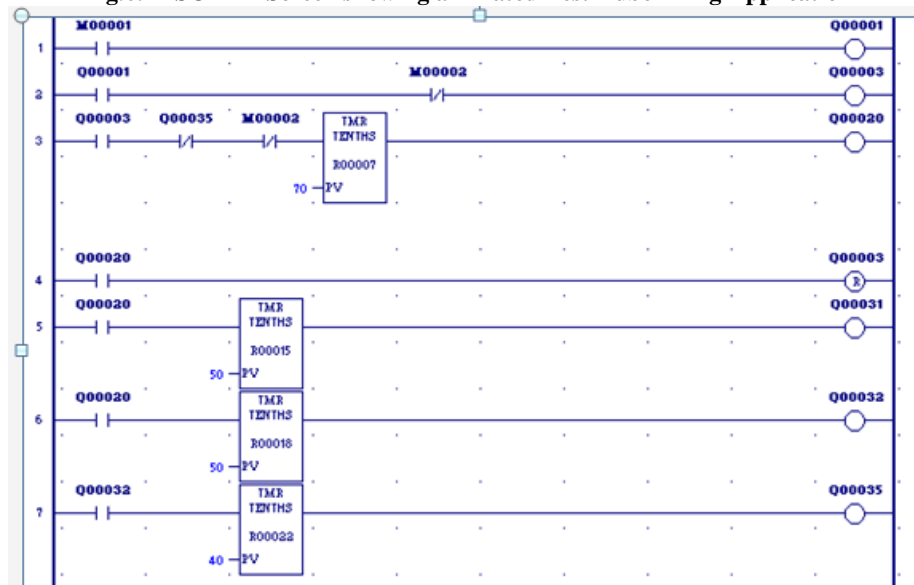


Fig.6.12 Ladder Logic for developing a Screen showing an imated Test Tube filling Application

## 6.3 EXPERIMENTAL RESULTS AND DISCUSSIONS

### 6.3.1 Servomotor in Internal Speed Control Mode

The following results were obtained on the Datascope window of the Versamotion Configuration software. The Servodrive was configured in the internal Speed control

mode by setting the appropriate parameters w.r.t tables 4.9, 4.10 and 4.11. The Datascope has a provision of monitoring three variables simultaneously through the three independent channels

### 6.3.1.1 Starting Characteristics of the PMSM Drive

The reference speed (Speed Input Command) was set to 2000 rpm and then the actual motor rotation speed of the servomotor was plotted on the datascope and it was observed (Fig. 6.13) that the motor attains the reference speed of 2000 rpm within a short duration of 187.20ms from the starting point. However it was also observed that the reference speed shows an instant rise in speed from rest to the set speed, whereas the actual motor rotation speed revealed a sluggish behavior. This is owing to the finite motor inertia ( $2.65 \times 10^{-4} \text{ Kg-m}^2$ ) and the armature inductance (1.81 mH) of the PMSM. These values are fairly small and hence it was concluded that the PMSM offers a good dynamic response with a small overshoot.

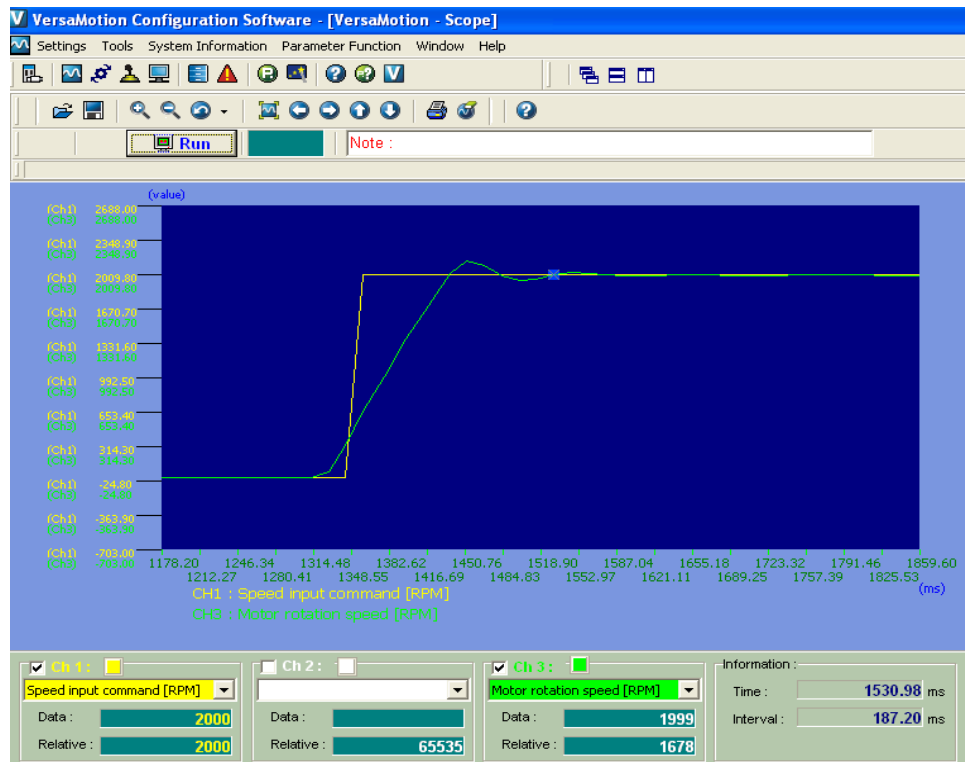


Fig. 6.13 Transient response of the PMSM drive during starting operation

### 6.3.1.2 Characteristics of the PMSM Drive when it Stops

After running for a small duration the PMSM is switched off and it is observed that it takes 437 ms to switch from 2000rpm to zero speed. Again the sluggish behavior is notable due to the reasons pointed out in 6.3.1.1. But overall the system response is fast as shown in Fig. 6.14.

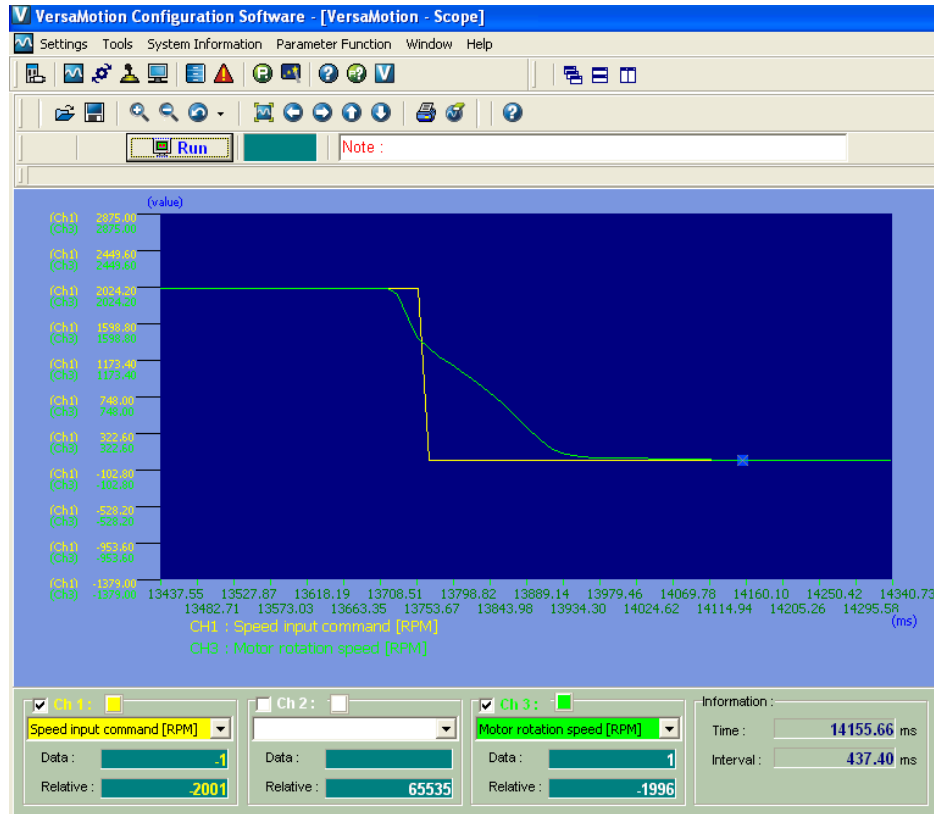


Fig. 6.14 Transient Response of the PMSM drive during Stopping operation

### 6.3.1.3 Trajectory Tracking in Speed Control Mode

The Servomotor was switched from rest to a speed of 1500 rpm, then to 3000 rpm, and further the speed was lowered to 2000 rpm and finally brought to rest. The variation of the actual speed of the drive has been captured on the Versmotion Datascope window. This was done at no load (Fig. 6.15) and then with the motor under loaded condition (Fig.6.16). It was observed that the Servomotor reached the actual speed precisely and quickly in both the cases but took comparatively more time to reach the reference speed in the loaded condition.

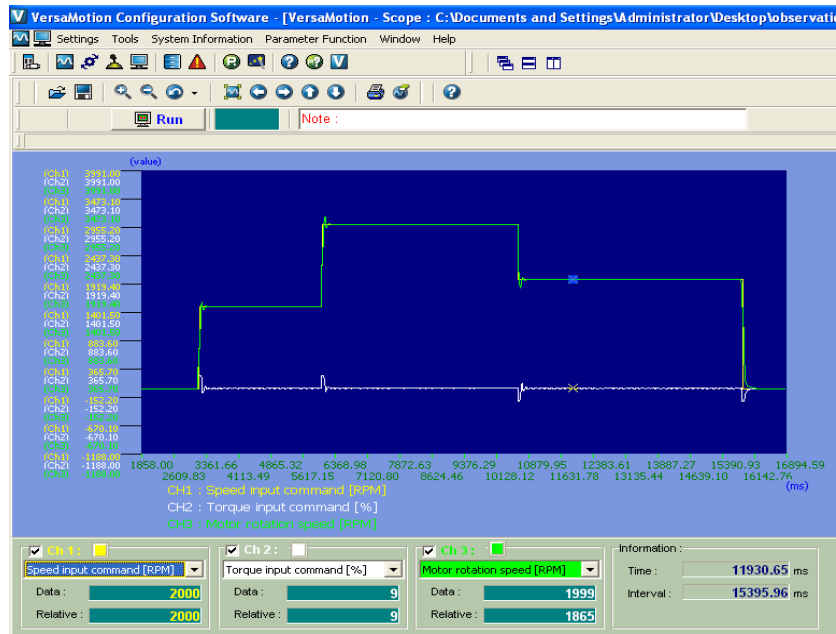


Fig.6.15 Trajectory tracking of PMSM under no load condition

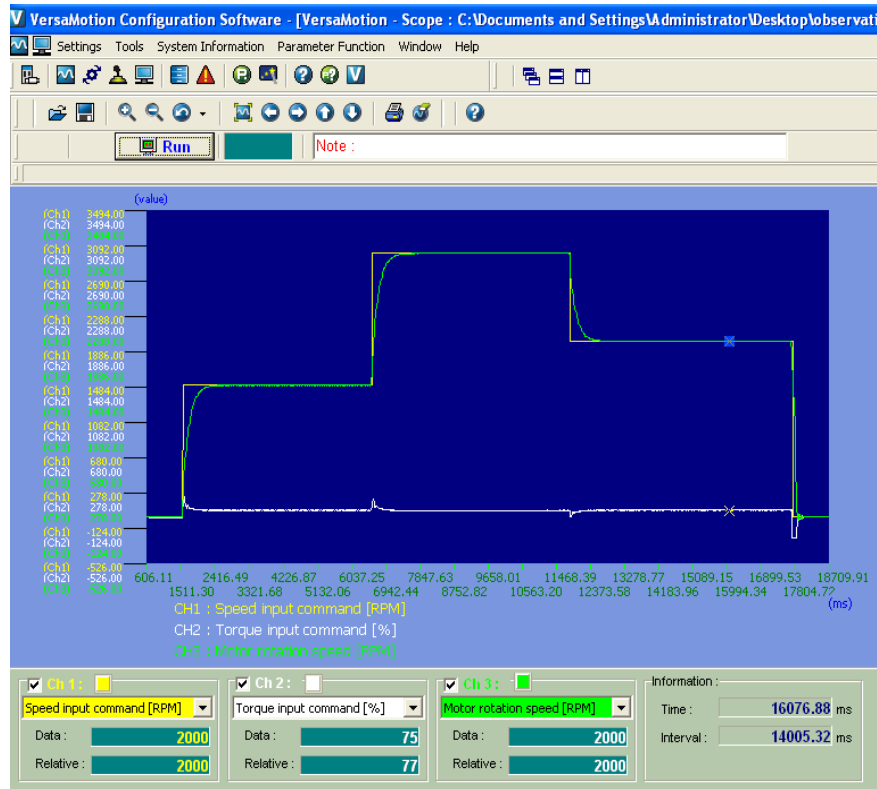


Fig. 6.16 Trajectory tracking of PMSM under load ed condition

The magnified views of the first switching instant, i.e from rest to a speed of 1500 rpm under the no-load condition and the loaded conditions of the motor is shown in Fig.6.17 and Fig. 6.18 respectively. Also the principle of Vector control is evident from these figures because even when the speeds were varied, the torque remained almost constant, showing only a small flicker at the switching points. It was also observed that the flicker was more prominent with the switching of speeds at no load.

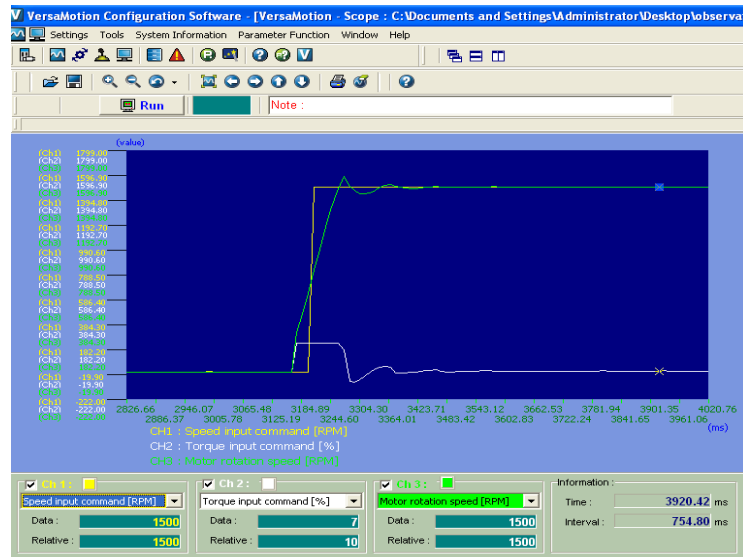
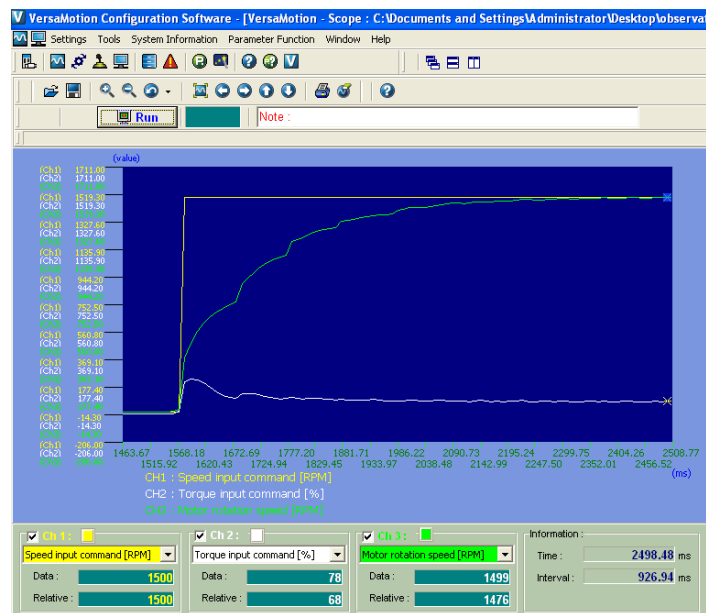


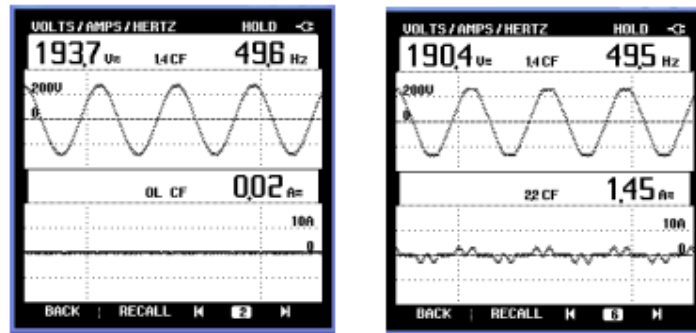
Fig.6.17 Magnified view of motor switching from rest to 1500rpm with motor at no load



6.18 Magnified view of motor switching from rest to 1500rpm with motor under load

### 6.3.2 Input and Output Parameters of the PMSM Drive

(a) Three Phase input supply voltage (line voltage) measured at R, S, T terminals of the AC drive with the motor at no load and running at 3000 rpm is shown in Fig.6.19 (a). It is observed that the PMSM drive draws AC voltage and converts it into DC through the diode rectifier. Under no load the input current drawn is too small and not shown clearly by the power analyser. Then the Servodrive is loaded and a significant input current is drawn which is actually the input line current of the diode bridge rectifier and is a pulsed waveform as shown in Fig. 6.19(b).



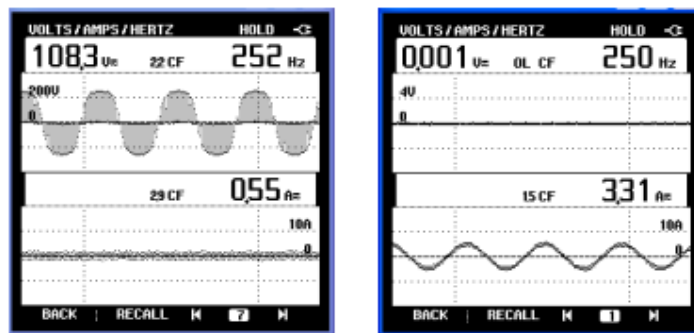
(a)

(b)

**Fig.6.19 (a) Input Voltage and input Line current of the PMSM Drive at 3000 rpm and no load**

**(b) Input Voltage and input Line current of the PMSM Drive at 3000 rpm under load**

(b) Fig.6.20 (a) shows the PWM Voltage at the drive output i.e terminals U,V,W which then act as the input terminals for the Servomotor. Fig. 6.20(b) shows the sinusoidal current drawn by the PMSM when fed from the PWM voltage. All these measurements are done at the motor rated speed of 3000 rpm with the motor under load condition.



(a)

(b)

**Fig.6.20 (a) PWM Voltage at PMSM Drive output at 3000 rpm under load**

**(b) Current drawn by the PMSM Drive at 3000 rpm under load**

## CHAPTER VII

### MAIN CONCLUSION AND FUTURE SCOPE OF WORK

---

#### 7.1 CONCLUSION

The present work was motivated to develop an online scheme to Monitor and Control a Three Phase Servodrive using PLC and SCADA.

A detailed study of the **Permanent Magnet Synchronous Motor (PMSM)**, ie the Servomotor used in this set-up was done including its Construction, Working, Control strategies adopted, specific features, Application areas and also some Problem areas. A detailed literature survey was done and all the important points were summarised.

A thorough study of all the **hardware components** was done including their specifications, functioning and overall performance. Three **softwares** platforms namely Proficy Machine Edition 5.90, Versamotion Configuration Software and the Proficy HMI/SCADA Cimplicity 7.5 Software were comprehended, analysed and implemented step by step.

A 1KW , 3000rpm, Permanent Magnet Servodrive was fully automated using PLC and AC Servo Drive and with the help of SCADA system its remote start and stop operations too were demonstrated. An effort was also made to configure the PLC device and its ports in SCADA to monitor the realtime values of the PMSM motor variables like Speed, current and Torque.

The necessary settings and software implementation of the algorithms needed for Position, Speed and Torque control operation of the drive were developed and tested for realtime implementation of the Servodrive.

The Servodrive used in this set-up offered various control modes of motor operation. The configuration and settings to run the Servomotor in **two control modes** (Speed and Position) were done systematically and its characteristics were captured on the datascopes and interpreted successfully. The Versamotion configuration software allowed eight preset internal positions to be attained by the Servomotor , with the user having to change the Digital Input settings everytime to change from one position setting to the other, through the JOG CONTROL window. This was quite time consuming. Hence a **Ladder program** was developed and verified in Proficy Machine which enabled the Servomotor to obtain five **different positions** in succession with a specified time

interval between two consecutive positions, the position settings being done all in one go in Versamotion Parameter Editor window. Similarly another Ladder program was developed and verified for obtaining three **different speeds** (in succession with a specified time interval between the different speeds) for Internal speed control mode.

Various methods of controlling the Three Phase Servomotor were designed and tested successfully in the lab. These methods included **Servomotor control** using the Versamotion configuration software, using Ladder Programming, using the HMI screen control, using the SCADA animated screen and directly using the digital keypad of the Servodrive.

Two case studies were done by creating virtually animated screens for PMSM precision position filling applications.

Other than practically implementing the above schemes an important aspect of the present work which was successfully carried out was to completely identify, study and summarise the communication ports, standards and **protocols** used between the various components of the entire laboratory set-up including the **PC – PLC – Drive – Servomotor – SCADA - HMI**.

A complete study and practical hands on the PLC and the servodrive operation have imparted a fairly good idea about the industrial automation systems.



## 7.2 FUTURE SCOPE OF WORK

The following aspects can be explored as an extension to this dissertation work.

Firstly, **External** Speed and Position control schemes can be tried for the Servomotor using the Servodrive. Secondly **dual modes** of Servomotor operation can be verified and analysed like Internal/External Position and Speed mode (Pr-S,Pt-S), Internal/External Position and Torque mode (Pr-T,Pt-T), Speed and Torque Mode (S-T). Thirdly an attempt can be made to develop a **MATLAB** Simulation of the present control scheme which will enable the user to verify the Servomotor characteristics using MATLAB. Fourthly, more emphasis can be given to the **real-time monitoring** of processes using the SCADA software to obtain complete control of a real-time process including monitoring of more variables and incorporating some error control element too.

Also the **Programming methodology** used in this dissertation work is Ladder Programming. The Proficy Machine Edition also offers various other programming platforms like FBD, ST , IL etc. Programs can be developed using these platforms. Last but not the least **expert systems** like Fuzzy logic, Neural Network and Genetic Algorithms can be used to develop some control strategy for the present scheme.

\*\*\*\*\*

## REFERENCES

1. Amir Firoozshahi and Seyeon Kim, "Intelligent and innovative monitoring of water treatment plant in large Gas Refinery" IEEE International Conference on Control Applications , pp. 269-273,September 2010.
2. Anirudha Upadhyay and Abhishek Agarwal, "Controller design and analysis for automation of Chemical Water Treatment system" 10th International Conference on Control, Automation, Robotics and Vision, pp. 1625-1629, December 2008.
3. I. Colak, R. Bayindir, A. Bektas, I.Sefa and G. Bal, "Protection of Induction Motor Using PLC" International Conference on Power Engineering, Energy and Electrical Drives,pp.96-99, April 2007.
4. Maria G. Ioannides, "Design and implementation of PLC-based monitoring control system for induction motor" IEEE Transactions on Energy Conversion, vol. 19, No.3,pp. 469-476 , September 2004.
5. A. R. Al-Ali, M. M. Negm, and M. Kassas, "A PLC based power factor controller for a 3-phase induction motor" in Proceedings of Conference Records IEEE Industry Applications, vol. 2, pp. 1065–1072, 2000.
6. A. S. Zein El Din, "High performance PLC controlled stepper motor in robot manipulator" in proceedings of IEEE International Symposium on Industrial Electronics, vol.2, pp. 974–978,1996.
7. Robert A. Hanna, Peter Bulmer and Rasul Kohistani, "Minimizing refinery upset during power interruptions using PLC control" in record of conference papers of Petroleum and Chemical Industry Conference, pp.185-195,1992.
8. Akram Hossain and Muhammed H.Rashid, "The hardware and software interface of a programmable logic controller to an industrial grade process control system" Conference Record of the IEEE Industry Applications Society Annual Meeting,vol.2,pp.1862-1868,1990.

9. H. M. Kamel, H. M. Hasanien, and H.E. A. Ibrahim, "Speed control of permanent magnet synchronous motor using fuzzy logic controller, pp.1587-1591, 2009.
10. M.F. Moussa, A. Helal, Y. Gaber, and H.A. Youssef, "Unity Power Factor control of permanent magnet motor drive system 12th International Middle-East Power System Conference, pp. 360-367, 2008.
11. R.K. Sharma, V. Sanadhya, L. Behera and S. Bhattacharya, "Vector control of a permanent magnet synchronous motor" Annual IEEE India Conference, vol.1, pp.81-86,2008.
12. Kai Sun, Kui Liu and Lipei Huang, "Control strategy of PMSM drive in high speed operation for air-condition compressor" 34th Annual Conference of IEEE on Industrial Electronics, pp. 1137-1142, 2008.
13. S.Z. Jiang, "Development and Implementation of AC Servo Motion System in proceedings of the 8th International conference on Electrical Machines and Systems, vol.2, pp.1569-1572, 2005.
14. Tae-Won Chun, Jung-Ryol Ahn, Hong-Hee Lee, Heung-Gun Kim and Eui-Cheol Nho, "Development of servo motor drive systems for embroidery machines" in the 8th Russian–Korean International symposium on Science and Technology, vol.1, pp.310-314, 2004.
15. S.Wekhande and V. Agarwal, "High resolution absolute position Vernier shaft encoder suitable for high performance PMSM servo drive" 29th Annual IEEE Power Electronics Specialists Conference, vol.1, pp.119 – 124, 1998.
16. Guo Qingding, Wang Limei and Luo Ruifu, "Completely digital PMSM servo system based on new self-tuning PID algorithm and DSP" in proceedings of the IEEE international conference on Industrial Technology, pp. 71-75, 1996.
17. Uk-Youl Huh and Je-Hie Lee, "A modified sliding mode speed control scheme for AC servo motor" in the proceedings of the 21st IEEE International Conference on Industrial Electronics, control and Instrumentation, vol.2, pp.730-735,1995.

18. P. Freere and P. Pillay, "Design and evaluation of current controllers for PMSM drives 16th Annual Conference of IEEE on Industrial Electronics Society, vol.2,pp.1193-1198, 1990.
19. P. Pillay and R. Krishnan"Modeling, simulation, and analysis of permanent-magnet motor drives-I: The Permanent-Magnet Synchronous motor drive" IEEE Transactions on Industry applications,vol.25,No.2,pp.265-273, March/April1989.
20. P. Pillay and R. Krishnan, "Modeling of permanent magnet motor drives" IEEE Transactions on Industrial Electronics,vol.35,No.4,pp.537-541, Nov.1988.
21. M. Iacob, G.-D. Andreescu, and N. Muntean, "SCADA system for a central heating and power plant" in Proceedings of 5th International Symposium on Applied Computational Intelligence and Informatics , pp. 159–164, May 2009
22. Cristina Anita Bejan, Mihai Iacob and Gheorghe-Daniel Andreescu, "SCADA Automation System Laboratory, Elements and Applications " IEEE, pp.181-186,2009
23. Mini S. Thomas, Pramod Kumar, V.K. Chandna, "Design, development, and commissioning of a supervisory control and data acquisition (SCADA) laboratory for research and training" IEEE transactions on Power System,vol.19,No. 3,pp.1582-1588, August 2004.
24. G. Kaplan, "Technology 1992 Industrial electronics" IEEE Spectr., vol.29, pp. 47–48, Jan. 1992
25. Thomas M. Jahns and Wen L. Soong, "Pulsating torque minimization techniques for Permanent Magnet AC Motor Drives – A Review" IEEE transactions on Industrial Electronics, vol.43, No.2,pp.321-330, April 1996
26. Bimal K. Bose, Modern Power Electronics and AC Drives, 1<sup>st</sup> ed.,Pearson Education, Inc, 2003.
27. Gopal K. Dubey, Fundamentals of Electrical Drives, 2<sup>nd</sup> ed.,Narosa Publishing House,2007.
28. Proficy Logic Developer-PLC, User's Manual, GFK-1918 ,Version 5.90,Nov. 2008.
29. VersaMotion Servo Motors and Amplifiers User's Manual, GFK-2480, Dec.2007.

30. Series 90-30 Installation and Hardware User's Manual, GFK-0356Q, Aug 2002.
31. Series 90-30/20/Micro Programmable Controllers Reference Manual, GFK-0467M, May 2002.
32. Series 90-30 PLC I/O Module Specifications, GFK-0898F, July 2000.
33. TCP/IP Ethernet Communications for the Series 90 PLC User's Manual, GFK-1541B, May 2002.
34. Cimplicity HMI Plant Edition CimEdit Operation Manual, GFK-1396F, July 2001.
35. Cimplicity HMI Plant Edition Base system Device communication User's Manual, GFK- 1181J, July 2001.