



**PROGRAMMING AND  
INTERFACING OF MACHINE  
VISION BASED MATERIAL  
SORTING SYSTEM**



**A PROJECT REPORT**

*Submitted by*

<b>R.KEERTHANA</b>	<b>1010104020</b>
<b>M.K.RAKESH</b>	<b>1010104037</b>
<b>SRIRAM.J</b>	<b>1010104049</b>

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

*in*

**MECHATRONICS ENGINEERING**

**KUMARAGURU COLLEGE OF TECHNOLOGY,  
COIMBATORE - 641 049**

**(An Autonomous Institution Affiliated to Anna University, Chennai)**

**APRIL - 2014**

**KUMARAGURU COLLEGE OF TECHNOLOGY,  
COIMBATORE - 641 049  
(An Autonomous Institution Affiliated to Anna University, Chennai)**

**BONAFIDE CERTIFICATE**

Certified that this project report “**PROGRAMMING AND INTERFACING OF MACHINE VISION BASED MATERIAL SORTING SYSTEM**” is the bonafide work of “**R.KEERTHANA, M.K.RAKESH, SRIRAM.J**” who carried out the project work under my supervision.

**SIGNATURE**

**Dr.S.A.PASUPATHY  
HEAD OF THE DEPARTMENT**

DEPARTMENT OF MECHATRONICS  
ENGINEERING  
KUMARAGURU COLLEGE OF  
TECHNOLOGY  
COIMBATORE-49.

**SIGNATURE**

**SUPERVISOR  
Mr.A.RAMKUMAR  
ASSISTANT PROFESSOR  
DEPARTMENT OF  
MECHATRONICS ENGINEERING  
KUMARAGURU COLLEGE  
OF TECHNOLOGY  
COIMBATORE-49.**

Submitted for the end semester examination held on 23/04/2014

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

We are grateful to thank **DR.R.S.KUMAR, Principal, KUMARAGURU COLLEGE OF TECHNOLOGY** for his extensive support in bringing out the project.

Our sincere thanks to **Dr. S.A.PASUPATHY, Head of Department, Department of Mechatronics Engineering**, for his encouragement and interest towards the project.

We thank our project coordinator, **Mr. PREM ANAND**, Assistant Professor, Department of Mechatronics Engineering for her continuous support towards finishing our project.

We wish to deliver our deep sense of gratitude to our project guide, **Mr.A.RAMKUMAR**, Assistant Professor, Department of Mechatronics Engineering, for her valuable guidance and advice towards our project to make it as a successful innovation.

We would like to express our gratitude towards **our department faculty** and others who helped and delivered their valuable suggestions to complete this project.

Last and literally, we express our gratefulness to **our beloved parents and friends** for aiding us with all the requirements and providing us all the encouragement and support throughout our project work.

## **ABSTRACT**

This project primarily focuses on Industrial Automation. It discusses about setting up a real-time Industrial process which deals with Autonomous Material Sorting system. The programming and interfacing of various control modules of this system is explained in this project. The major interface is between **LabView**, for image processing and **PLC** which controls the various inputs/outputs of the industrial setup. The various components used for the interfacing are Programmable Logic Controllers (PLC) from DELTA and related software package; National Instrument's LabVIEW with the associated Data Logging and Supervisory Control Toolset (DSC). The communication protocols experimented in this project are MODBUS and Serial Communication. The final actuation of the robot arm IRB1410 is through the robot controller IRC-5.

**Keywords: PLC, IRC5, Interfacing, LABVIEW PLC Communication**

## TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	<b>ABSTRACT</b>	<b>iv</b>
	<b>LIST OF TABLE</b>	<b>vii</b>
	<b>LIST OF FIGURES</b>	<b>viii</b>
	[	
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 LITERATURE REVIEW	3
	1.2 INDUSTRIAL AUTOMATION	4
	1.2.1 TYPES OF INDUSTRIAL AUTOMATION	
<b>2</b>	<b>PROJECT DOMAIN</b>	
	2.1 MATERIAL TENDING	6
	2.2 FLOOR SETUP	7
<b>3</b>	<b>COMPONENTS AND MODULES</b>	
	3.1 PC WITH LABVIEW	9
	3.2 PROGRAMMABLE LOGIC CONTROLLER (PLC)	9
	3.3 ABB IRB 1410 ROBOTIC ARM	13
	3.3.1 ROBOT KINEMATICS	14
	3.3.2 IRC 5 CONTROLLER	17
	MODULE	
	3.3.2.1 KEY FEATURES	17
	3.3.2.2 MOTION CONTROL COMMANDS	20

<b>4</b>	<b>INTERFACING AND CONTROL EXPERIMENTS</b>	
4.1	INTERFACE MODES	22
4.2	MODBUS RTU	23
	4.2.1 USING ABB AC 31 PLC	26
4.3	SERIAL COMMUNICATION	28
4.4	ARDUINO COMMUNICATION	29
<b>5</b>	<b>SOFTWARE AND PROGRAMMING</b>	
5.1	SOFTWARE	30
	5.1.1 DELTA-WPL SOFT	30
	5.1.2 ROBOT STUDIO	
	5.1.2.1 FLEX PENDANT	34
	5.1.2.2 STUDIO EDITOR	35
5.2	PLC PROGRAMMING	36
5.3	IRC-5 PROGRAMMING	37
<b>6</b>	<b>CONCLUSION</b>	
6.1	PROJECT RESULTS	40
6.2	PROBLEMS ENCOUNTERED	41
6.3	FUTURE SCOPE	41
<b>7</b>	<b>APPENDIX</b>	42
<b>8</b>	<b>REFERENCES</b>	51

## LIST OF TABLES

<b>TABLE NO.</b>	<b>TABLE</b>	<b>PAGE NO.</b>
Table 3.1	PLC Modules	11
Table 3.2	Inputs/Outputs of PLC	11
Table 3.3	Specifications of IRB1410	16
Table 3.4	Specifications of IRC-5 Controller	19
Table 4.1	MODBUS Function codes	24
Table 4.2	MODBUS Error codes	25
Table 4.3	Serial Parameters	28

## LIST OF FIGURES

<b>FIGURE NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
Figure1.1	Structure of Industrial automation	3
Figure1.2	Automation Types	5
Figure2.1	Signal Flow	7
Figure2.2	Floor set up using Robot Studio	8
Figure3.1	PLC Product profile and Outline	12
Figure3.2	Programming Cable	13
Figure3.3	IRB 1410- Specifications	15
Figure3.4	Motion commands	21
Figure4.1	MODBUS Transmission	23
Figure4.2	MODBUS RTU Network	25
Figure4.3	MODBUS using AC31 GRAF	26
Figure4.4	MODBUS using 907AC31 GRAF	27
Figure4.5	Flow loop	29
Figure5.1	PLC communication – step 1	30
Figure5.2	PLC communication – step 2	31
Figure5.3	PLC communication – step 3	31
Figure5.4	PLC communication – step 4	32
Figure5.5	PLC communication – step 5	33
Figure5.6	Studio editor	35
Figure5.7	Ladder diagram	36

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 LITERATURE REVIEW**

#### **MACHINE VISION AND OBJECT SORTING: PLC COMMUNICATION WITH LABVIEW USING OPC[1]**

In this thesis Pokharel demonstrated a machine vision application for the quality control of products in an industry using a miniature production line station. This was constructed and automated using STEP 7 software and Simatic based Programmable Logical Controller (PLC). The machine vision part of the thesis was done in NI LabVIEW. Object sorting using machine vision was accomplished using pattern matching algorithm of machine vision. When the object sorting application runs, the camera acquires the image of the object into LabVIEW. The vision application analyses the image, and sends an electrical signal to the sorter if the acquired image matches the template image. The communication between the PLC and LabVIEW was carried out using NI OPC (OLE for Process Control), an add-on for LabVIEW. For the SIEMENS PLC, a bus coupler was networked using PROFIBUS cable.

The OPC driver integrated in Labview was used successfully to communicate between PLC and LabView, for which OPC communication Protocol was used.

#### **APPLICATION OF MODBUS TO COMMUNICATE THE PLC AND LAB VIEW FOR REAL TIME PROCESS CONTROL.[2]**

A similar Communication between LabView and PLC was done in this paper where a SCADA architecture was developed instead of a proprietary technology to develop and simulate an industrial process. This was done by using

three SCADA Layers utilizing LabView and PLC. MODBUS communication protocol was employed to solve compatibility problem raised from different vendors' tools. Delta DVP-14SS PLC to the PC with LabVIEW 2011 via MODBUS for controlling water level. WPLSoft programming software was used to program the PLC and communication was achieved via RS232 port. In the MODBUS communication between LabView and PLC, LABView acts as master and PLC acts as slave. The water level control was simulated and controlled in the LABView VI itself.

### **MECHATRONIC SYSTEMS CONTROL BASED ON SCADA SYSTEM, OPC SERVER AND LABVIEW[3]**

Another SCADA system used LAN network to connect different PLCs and each PLC's unique IP address present was tracked to communicate over the LAN via TCP/IP Ethernet. A big advantage of the solution presented in this paper is the possibility to control systems which are realized with different types of PLCs and with PLCs made by different producers, possibility to control systems in the local area network as well as control and monitoring of mechatronic systems by Internet regardless of geographic position of the operator and process. Fusion of these two software packages (LabView and PLC) forms a powerful SCADA system which enables access to users worldwide by the Internet. This system provides monitoring of all analog and digital signals which are significant to users, as well as controlling mechatronic systems in real time.

# 1.2 INDUSTRIAL AUTOMATION

Industrial Automation is the muscle that moves the Industry today. It isa set of technologies that result in significant operation of machines without human intervention. It achieves performance superior to manual operation.

Industrial Process without human intervention is the need of the hour technology, which sustainably accelerates material, energy, manpower and infrastructure aspects. Industrial Automation is the building block for setting up a Real-Time system. The Structure of the Industrial automation is illustrated in Figure1.1

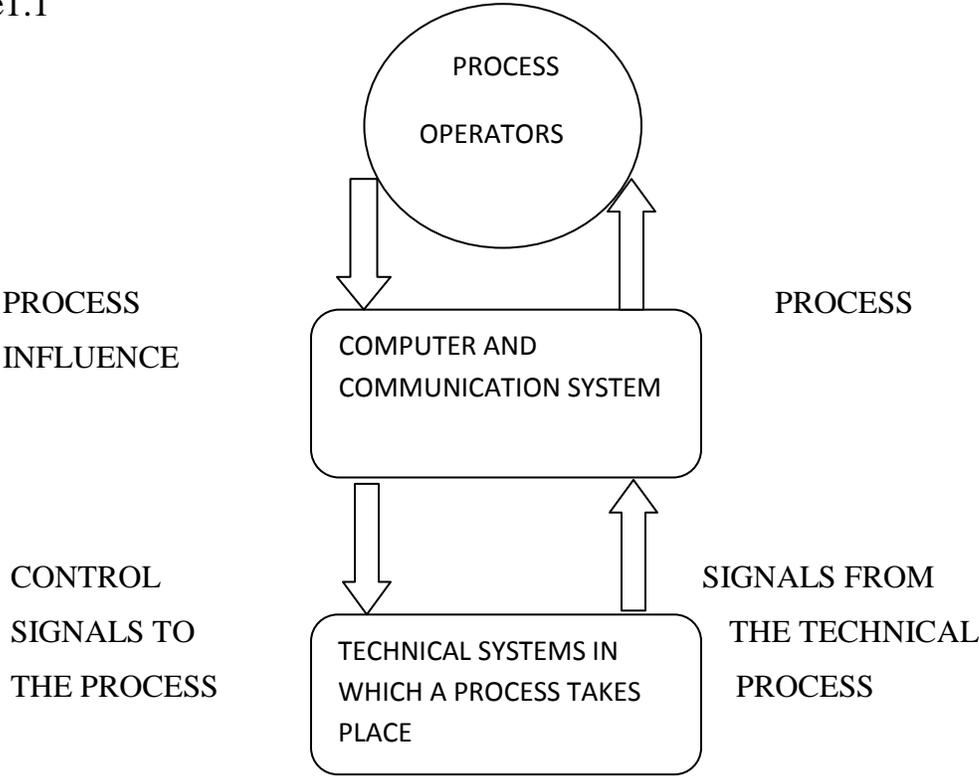


Figure1.1 Structure of Industrial automation

## 1.2.1 TYPES OF INDUSTRIAL AUTOMATION

Industrial automation is usually one of these four types

1. Fixed
2. Programmable
3. Flexible
4. Integrated

### **Fixed automation (Hard automation):**

Fixed automation refers to the use of special purpose equipment to automate a fixed sequence of processing or assembly operations. Each of the operation in the sequence is usually simple, involving perhaps a plain linear or rotational motion or an uncomplicated combination of two. It is relatively difficult to accommodate changes in the product design. This is called hard automation.

### **Programmable automation:**

In programmable automation, the production equipment is designed with the capability to change the sequence of operations to accommodate different product configurations. The operation sequence is controlled by a program, which is a set of instructions coded. So that, they can be read and interpreted by the system. New programs can be prepared and entered into the equipment to produce new products.

**Flexible automation (Soft automation):**

Flexible automation is an extension of programmable automation. A flexible automation system is capable of producing a variety of parts with virtually no time lost for changeovers from one part style to the next. There is no lost production time while reprogramming the system and altering the physical set up.

**Integrated Automation:**

Integrated automation involves production and management integration. It is a sophisticated system, which uses computer communications with advanced optimized algorithms. This type of automation is applied in chemical process automation.

The various automation types comparison is illustrated in Figure 1.2 [9]

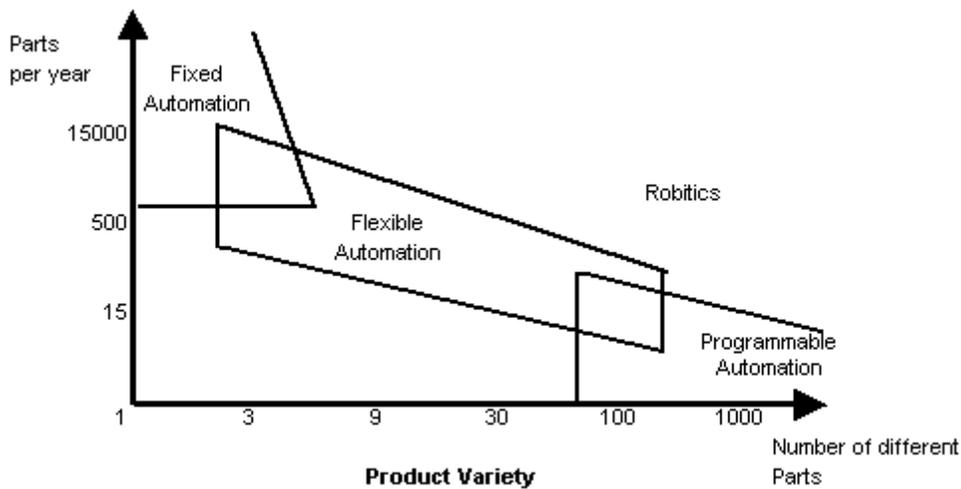


Fig.1.2 Automation types

## **CHAPTER 2**

### **PROJECT DOMAIN**

This Industrial automation process focuses on Autonomous Material Tending. Since it is a fixed continuous process, it falls under the type Fixed Automation technology. It aims to bridge the gap between a Semi-Autonomous and a Fully-Autonomous Bot.

#### **2.1 MATERIAL TENDING**

The process is a machine vision based material sorting/tending process. Continuous monitoring of the products, moving on the conveyor, is enabled using Microsoft Lifecam. Decision making is done by using LabVIEW, based on template and color of the product. A Programmable Logic Controller (PLC)- Delta DVP-16SX11 series is the central control module used. An ABB IRB 1410 Robotic arm is the actuator used for material sorting (pick and place). Control signals (LabVIEW – PLC - ABB IRC 5 Controller) flowing through a closed loop controls the entire process. The total signal flow of the process is illustrated in the Figure2.1

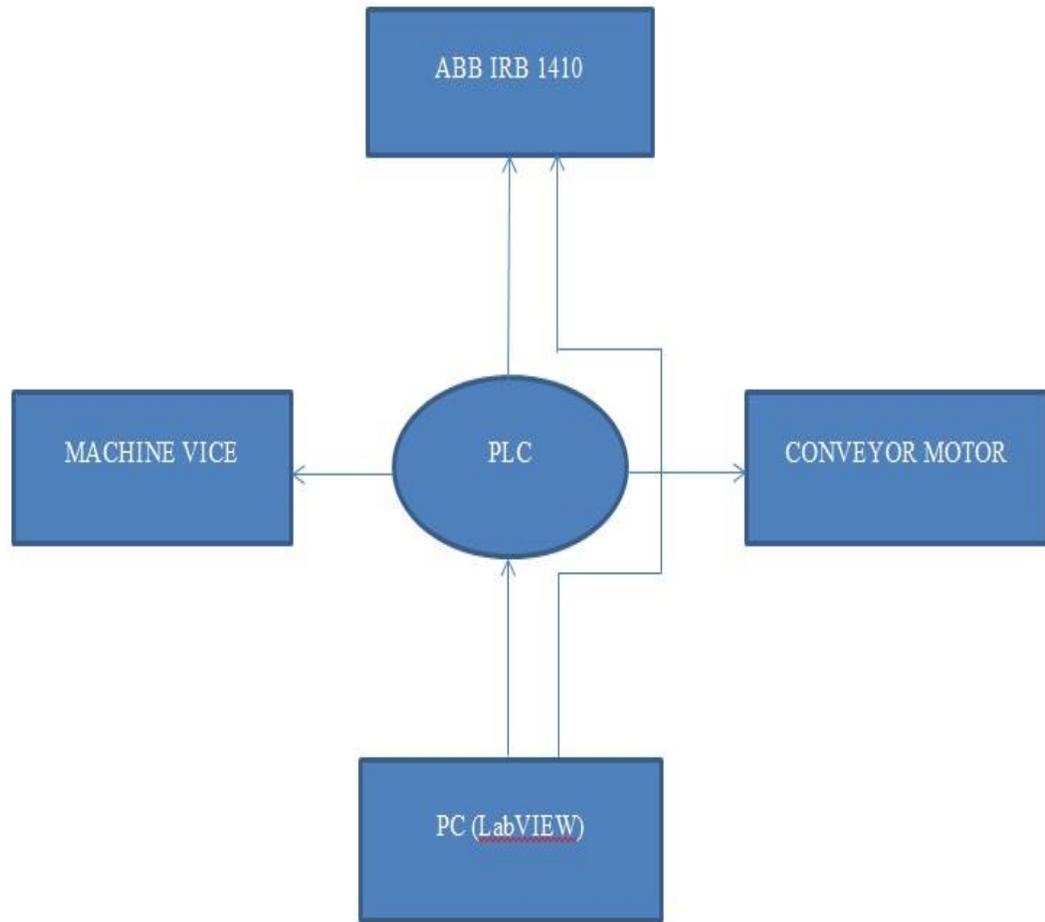


Figure2.1 Signal Flow

## 2.2 FLOOR SET UP

This Real-Time process involves setting up of the following:

1. A conveyor with payloads. An AC (Alternating Current) Induction motor drives the conveyor, with payloads.
2. A CNC like set up using a Machine Vice with automatic door open-close system.
3. Permanent slots/seats for the final product selected using machine vision.
4. A Programmable Logic Controller (Delta DVP-SX series) with its I/O extension module.

5. ABB IRB 1410 Industrial Robotic Arm. It is the final actuator layer which picks and places the selected product.

The floor/arena should be such that it accommodates all the required hardware components and modules, in a continuous run. The floor setup is illustrated in figure 2.2

**NOTE:**

It is best recommended to install the **ABB robot** in the **centre** of the arena and all other components and modules around it. It is made sure that the process components are **in the reach** of the robot

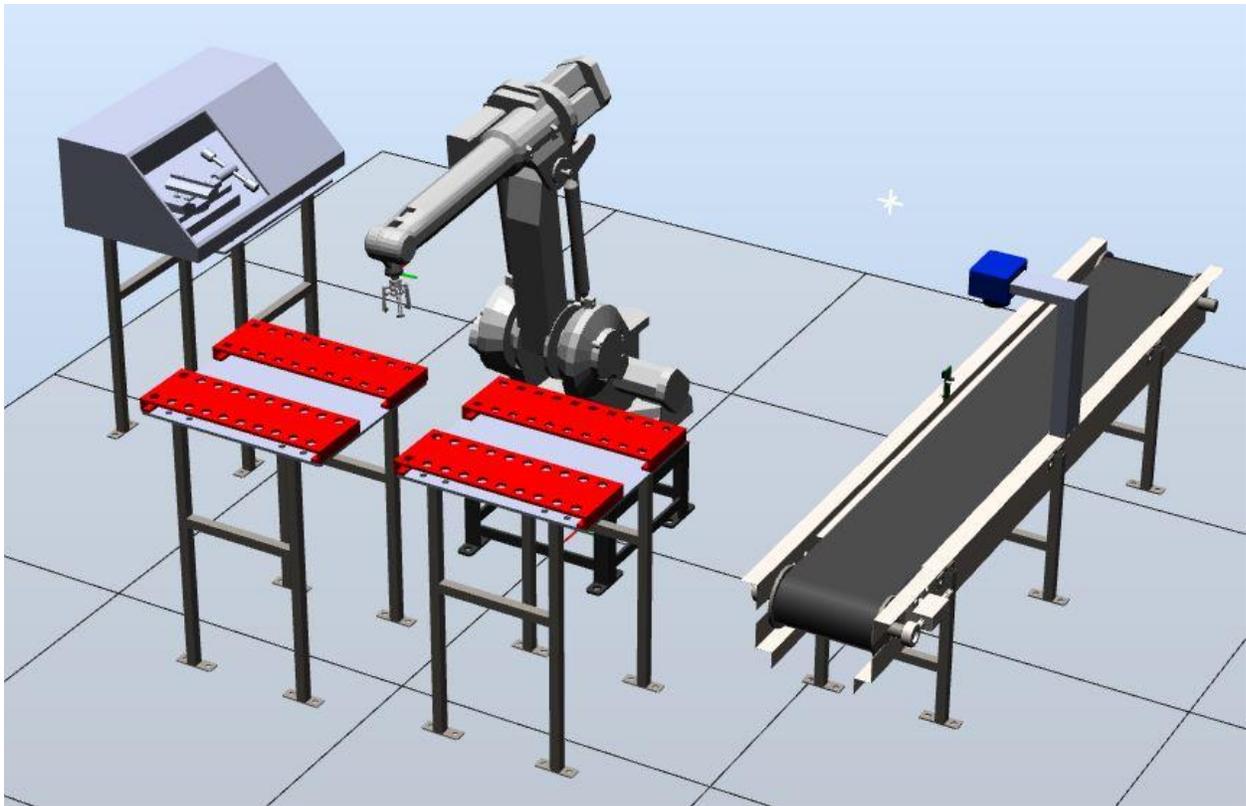


Figure2.2 Snapshot of Floor Set up using Robot Studio

## **CHAPTER 3**

### **COMPONENTS AND MODULES**

The Components and modules required for Programming and Interfacing the Autonomous Material Tending process are detailed below:

#### **3.1 PC WITH LABVIEW**

NI's (National Instrument's) LabVIEW is the software used for Image Processing. It continuously processes the images of the product captured. Hence, a PC (Personal Computer) or Laptop is used. This PC should possess the following requisites:

- Compatible with LabVIEW
- Suitable configuration as to process image continuously
- Serial Port (RS-232/RS-485) / Ethernet socket is mandatory. The Signals after decision making are sent via any of these ports.

#### **3.2 PROGRAMMABLE LOGIC CONTROLLER (PLC)**

PLC (Programmable Logic Controller) is an electronic device, previously called "sequence controller". In 1978, NEMA(National Electrical Manufacture Association) in the United States officially named it as "Programmable Logic Controller". PLC reads the status of the external input devices, e.g. keypad, sensor, switch and pulses, and execute by the microprocessor logic, sequential, timing, counting and arithmetic operations according to the status of the input signals as well as the pre-written program stored in the PLC. The generated output signals are sent to output devices as the switch of a relay, electromagnetic valve, motor drive, and control of a machine or operation of a procedure for the purpose of machine automation or processing procedure. The peripheral devices (e.g. personal

computer/handheld programming panel) can easily edit or modify the program and monitor the device and conduct on-site programmaintenance and adjustment. The widely used language in designing a PLC program is the ladder diagram.[7]

### **The Working Principles of Ladder Diagram**

The ladder diagram was a diagram language for automation and it is the oldest and most widely adopted language in automation. In the initial stage, there were only A (normally open) contact, B (normally closed) contact, output coil, timer and counter are the sort of basic devices on the ladder diagram. After the invention of programmable logic controllers (PLC), the devices displayable on the ladder diagram are added with differential contact, latched coil and the application commands which were not in a traditional power panel, for example the addition, subtraction, multiplication and division operations.

The working principles of the traditional ladder diagram and PLC ladder diagram are basically the same. The only difference is that the symbols on the traditional ladder diagram are more similar to its original form, and PLC ladder diagram adopts the symbols that are easy to recognize and shown on computer or data sheets. In terms of the logic of the ladder diagram, there are combination logic and sequential logic.

In this process automation, the PLC acts as the major interface between LabView and IRC-5 controller.

The functions of the PLC is to

- Communicate with LabView i.e., acquire the result of image processing.
- According to the LabView signal and sensor input, control the Conveyor motor, Chuck door.

The steps involved in the PLC Programming are

- To communicate with LabView via MODBUS protocol
- To develop ladder diagram for controlling IRC-5 Controller, conveyor motor, chuck door, sensor and vice.

The PLC used here is Delta's DVP PLC which contains two modules. They are tabulated in the table 3.1

PLC NAME	FUNCTION	INPUTS		OUTPUTS	
DVP10SX11R	PLC MPU WITH RS 232 AND RS 485 COMMUNICATION PORTS	Digital input	Analog Input	Digital Output	Analog Output
		4	2	2	2
DVP16SP11R	DIGITAL I/O MODULE	8		8	

Table 3.1 PLC Modules

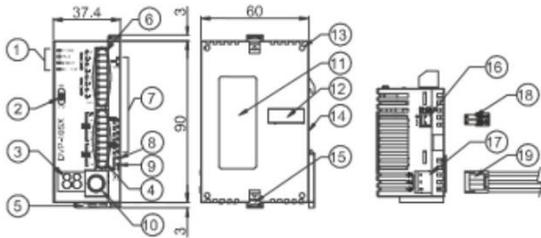
The various inputs/outputs connections of the DELTA PLC are in the table 3.2

ITEM	FUNCTION	INPUT/OUTPUT CONNECTION
LabView	Obtain signals of Image processing(Machine Vision End)	Input VIA RS485 connection utilizing MODBUS
Chuck door	To close the door of the vice for machining	Output relay of PLC
Bench Vice	To fit the product for machining	
Sensor	To monitor the position of the product in the conveyor	
Conveyor Motor	To run the conveyor belt	

Table 3.2 Input/Outputs of PLC

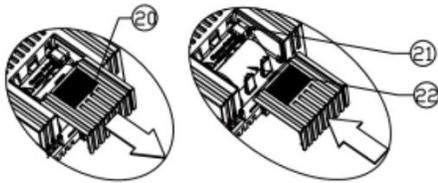
The PLC product profile is illustrated in figure 3.1[10]

2.2 Product Profile and Outline



Units: mm

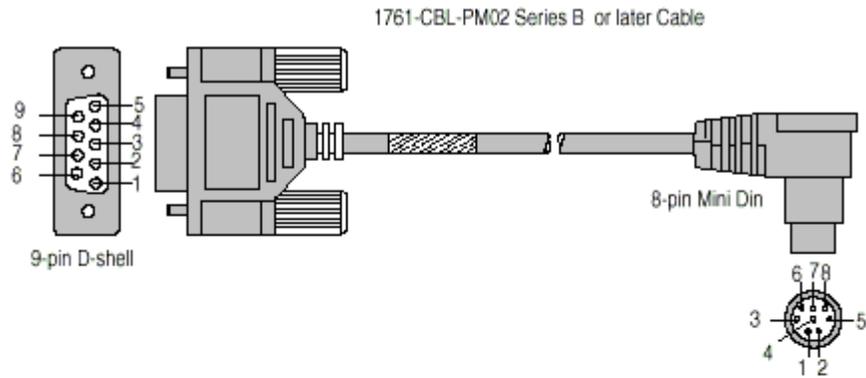
※ Battery replacement: Please change the battery within 3 minutes, or the internal data of the PLC (including the program area, RTC and latched registers) could be lost or destroyed.



- |    |  |
|----|--|
| 1  | Status indicator: POWER, RUN, ERROR, BAT.LOW |
| 2  | RUN/STOP switch                              |
| 3  | 2-digital 7-segment display                  |
| 4  | Digital I/O terminal                         |
| 5  | DIN rail clip                                |
| 6  | Analog I/O terminals                         |
| 7  | I/O point indicators                         |
| 8  | COM1 (RS-232) (Rx) indicator                 |
| 9  | COM2 (RS-485) (Tx) indicator                 |
| 10 | COM1 (RS-232) Communication port (Slave)     |
| 11 | Nameplate                                    |
| 12 | Expansion port                               |
| 13 | Mounting hold of the expansion unit          |
| 14 | DIN rail (35mm)                              |
| 15 | Expansion unit clip                          |
| 16 | COM2 (RS-485) Communication port             |
| 17 | DC Power input                               |

Figure3.1 PLC Product profile and outline

The PLC program from the WPLSoft is flashed via RS232 Cable. The cable used is 9 pin D Shell to 8 pin Mini DIN. The PIN connection is illustrated in Figure 3.2[11]



1761-CBL-PM02 Series B (or equivalent) Cable Wiring Diagram

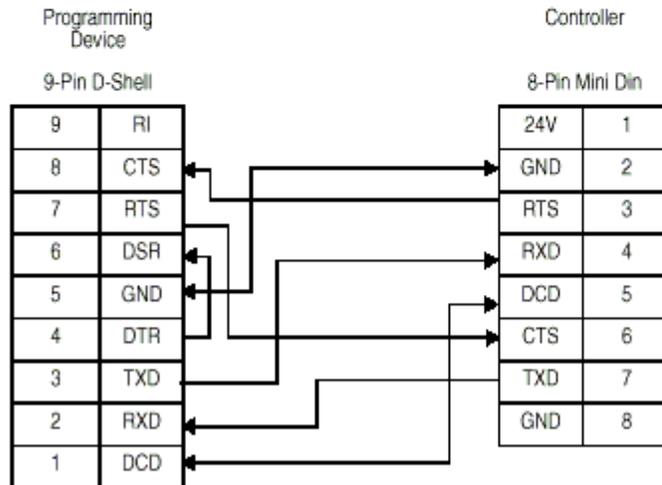


Figure 3.2 Programming Cable

### 3.3 ABB IRB 1410 ROBOTIC ARM

An industrial robot is a general-purpose, computer-controlled manipulator consisting of several rigid links connected in series by revolute or prismatic joints. One end of the chain is attached to a supporting base, while the other end is free and equipped with a tool to manipulate objects or perform assembly tasks. The motion of the joints results in relative motion of the links. Mechanically, a robot is composed of an arm, wrist and tool. The work volume is the sphere of influence of a robot whose arm can deliver the wrist subassembly unit to any point within the

sphere. The arm subassembly generally can move within 6 degrees of freedom (6DOF).

The wrist subassembly unit usually consists of **three rotary motions**.

These motions are defined as -

- Pitch,
- Yaw,
- Roll.

The ability to adjust the process speed and position facilitates to achieve optimum process efficiency with little or no rejects.

### **3.3.1 ROBOT KINEMATICS**

Robot arm kinematics deals with the analytical study of the geometry of motion of a robot arm with respect to a fixed reference co-ordinate system. This system is without regard to the force / moments that cause the motion. Thus, **kinematics** deals with the analytical description of the spatial displacement of the robot as a function of time, in particular the relations between the joint-variable space and the position and orientation of the end-effector of a robot arm

The two fundamental concepts with respect to robot arm kinematics are -

- Direct kinematics
- Inverse kinematics

A systematic and generalized approach which utilizes *matrix algebra* to describe and represent the spatial geometry of the links of a robot arm by systematically establishing a co-ordinate system (body-attached frame) to each link of an articulated chain. This method uses a 3x3 homogeneous transformation matrix to describe the special relationship between two adjacent mechanical links and reduces the direct kinematic problem to finding an equivalent 3x3 homogeneous transformation matrix. Thus, through sequential transformations, the

end-effector expressed in the “hand co-ordinates” can be transformed and expressed in the “base co-ordinates” which make up the inertial frame of this dynamic system.

The IRB 1410 robot is proven mainly in arc welding applications. It gives you fast and reliable work cycles that boost productivity and provides outstanding performance, ensuring short payback times. The robot has a handling capacity of 5Kg at the wrist with an additional 18Kg load for application equipment on the upper arm. The **end effectors** can be customized according to the process needs.

Here, in this process, we use a pneumatic gripper to handle objects. IRB 1410 is known for its **stiff** and **robust** construction. The robot has a large working area and a long reach. With a compact design and slim wrist, high performance is possible even in difficult operations. The specification, working range and load diagram of The ABB IRB1410 is illustrated in Figure.3.3. The specifications of IRB 1410 is tabulated in the table 3.3

Specification

Robot version	Handling capacity	Reach	Mounting	Protection options
IRB 1410	5 kg	1.44 m	Floor	Dry environment

Working range and load diagram

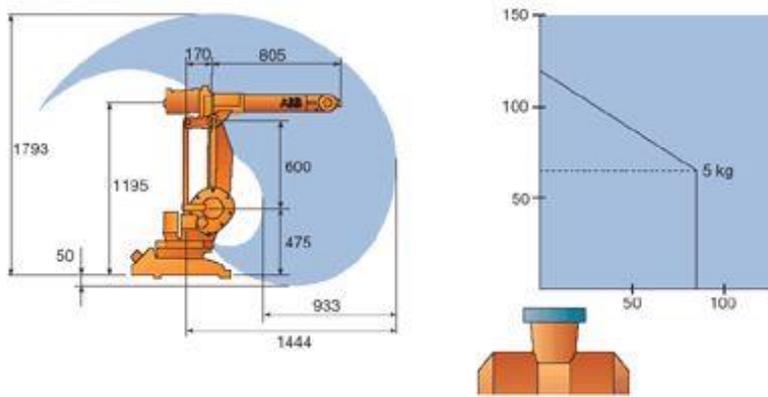


Figure3.3 IRB1410-Specifications

Working Range	
Robot Handling Capacity	5 KG
Reach of the axis	1.44 m
Supplementary load: on axis 3 on axis 1	18 kg 19 kg
Number of axes	6
Robot manipulator	6
External devices	6
Integrated signal supply	12 signals on upper arm
Integrated air supply	Max. 8 bar on upper arm
IRC5 Controller variants	Single cabinet, Compact
Performance	
Position repeatability	0.05 mm (average result from I/O test)
Movements	IRB 1410
Max. TCP velocity	2.1 m/s
Continuous rotation of axis	6
Electrical connections	
Supply voltage	200-600 V, 50/60 Hz
Rated power- Transformer rating	4 kVA/7.8 kVA with external axes
Physical	
Robot mounting	Floor
Dimensions	620 x 450 mm
Robot base	
Weight of the Robot	225 kg

Table 3.3 Specifications of IRB 1410

### **3.3.2 IRC 5 CONTROLLER MODULE**

The IRC 5 robot controller extends the comprehensive IRC 5 control family bringing all the benefits of the world leading robot controller including superior path motion, user friendly flex pendant programming, flexible RAPID language and powerful communication capabilities. The specifications of IRC-5 controller are given in table 3.4

#### **3.3.2.1 KEY FEATURES**

##### **SAFETY**

Operator safety is a central quality of the IRC 5, fulfilling all relevant regulations with good measure. Electronics position switches and safe move represent a new generation of safety, enabling more flexible safety concepts.

##### **MOTION CONTROL**

Based on advanced dynamic modeling, the IRC 5 optimizes the performance of the robot for the physically shortest possible cycle time (Quick move) and precise path accuracy.

##### **MODULARITY**

The IRC 5 comes in different variants in order to provide a cost-efficient solution for every need. The ability to stack modules on top of each other, put them side by side or distributed in the cell, is a unique feature, leading to optimization of foot print and cell layout.

##### **FLEXPENDANT**

The flex pendant is characterized by its clear touch screen based design and 3D joystick for intuitive interaction. Powerful customized application support enables leading of tailor-made applications, e.g. operator screens, thus eliminating a separate operator HMI.

## **RAPID PROGRAMMING LANGUAGE**

RAPID programming provides the perfect combination of simplicity, flexibility and power. It is a truly unlimited language with support for structured programs, shop floor language and advanced features. It also incorporates powerful support for many process applications.

## **COMMUNICATION**

The IRC 5 supports the state-of-the-art field buses for I/O and is a well-behaving node in any plant network. Sensor interface, remote disk access and socket messaging are examples of the many powerful network features. The IRC 5 controller includes communication modes like DeviceNet™, PROFINET, PROFIBUS and Ethernet/IP™.

## **ROBOT STUDIO**

A powerful PC tool for working with IRC 5 data on-lines as well as off-lines. In off-line mode, Robot Studio provides a perfect digital copy of the robot system together with strong programming and simulation features.

## IRC-5 CONTROLLER SPECIFICATIONS

Control Hardware	Multi-processor system PCI bus Pentium® CPU Flash disk for mass memory USB memory interface
Control software	Object-oriented design High-level RAPID programming language PC-DOS file format
Supply Voltage	1 phase 220/230 V, 50-60 Hz  Integrated transformer or direct mains connection 1 phase 220/230 V, 50-60 Hz (for Compact Controller only)
<b>Machine Interfaces</b>	
Inputs/outputs	Up to 8192 signals
Digital	24V DC or relay signals
Analogue	2 x 0-10V , 3 x ± 10V , 1 x 4-20Ma
Serial channel	1 x RS 232/RS 422 with adapter
Network	Ethernet(10/100 Mbits per second)
Fieldbus Master	DeviceNet™ PROFINET PROFIBUS DP Ethernet/IP™
Fieldbus Slave	DeviceNet™ PROFINET PROFIBUS DP Ethernet/IP™

	Allen-Bradley Remote I/O CC-link
<b>User Interfaces</b> Control panel	On cabinet or remote
FlexPendant	Weight 1 kg Graphical color touch screen Joystick Emergency stop Hot plug Support for right and left-handed operators USB Memory support
Sensor Interfaces	Search stop with automatic program shift Seam/contour tracking Conveyor tracking Machine vision Force Control

Table 3.4 Specifications of IRC-5 Controller

### 3.3.2.2 MOTION CONTROL COMMANDS

Path motion forms the main component of any industrial robot. Figure illustrates the move command structure for an ABB robot controller, which serves the robot manipulator to specific path co-ordinates that have been predefined or calculated from robot kinematics algorithms. The Industrial robot MOVE command is structured as illustrated in Figure 3.4

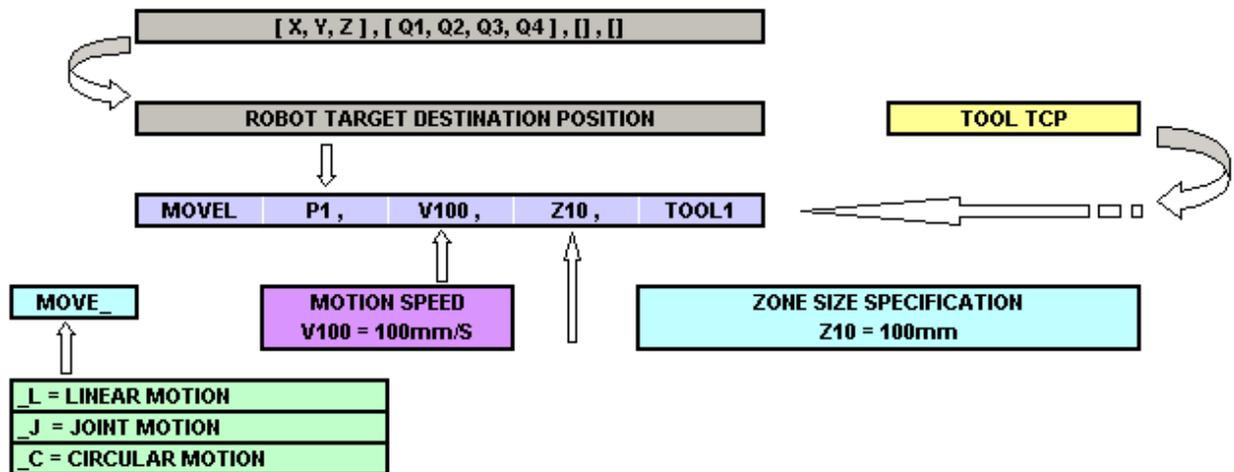


Figure 3.4 Motion commands

MOVE Command consists of the following components:

- MOVEL = Linear path motion
- P1 = the destination position to which the robot is to move
- V100 = Motion Speed
- Z10 = Zone size (accuracy) , i.e. how close the robot must be to the destination
- position before it can start to move towards the next position
- TOOL1 = Current Tool Position (TCP)

## **CHAPTER 4**

### **INTERFACING AND CONTROL**

Interfacing and control of various components and modules in the process enables the flow of control signals through that interfaced loop. The central control module-PLC is interfaced with the PC (Personal Computer) and IRC 5 controller. Gathering the communication signals, PLC controls other actuators. This process is done using the two methods detailed below:

#### **4.1 INTERFACE MODES**

For the process automation here, there are three functions to be done as part of the interface. i.e,

- Acquire image processing signals from LABView
- Communicate to the IRC-5 Controller
- Control the other outputs viz. conveyor motor, sensor, chuck door and bench vice.

#### **LABVIEW COMMUNICATION**

The data from LabView has to be communicated to the IRC-5 controller ultimately so that the robotic arm can take control for the pick and place operation. This is done either

- Via PLC by using MODBUS protocol (RS485) using WPLSoft

OR

- Directly to the IRC-5 Controller via serial communication using Flex Pendant and RAPID instructions

Both the methods are explained in this thesis.

## 4.2 MODBUS RTU

MODBUS was developed by Modicon (now Schneider Electric) in 1979, as a means for communicating with many devices over a single twisted pair wire. The original scheme ran over RS232, but was adapted to run on RS485 to gain faster speed, longer distances and a true multi-drop network. MODBUS quickly became a de facto standard in the automation industry.

The MODBUS protocol follows a master/slave architecture, where a master will request data from the slave and can also ask to perform some action. The master initiates a process by sending a function code that represents the type of transaction to perform. During communication on a MODBUS network, the protocol determines how each controller will know its device address, recognize a message addressed to it, determine the kind of action to be taken, and extract any data or other information contained in the message. A MODBUS RTU Connection is given in Figure 4.2 [4]

The master typically is a PLC (Programmable Logic Controller), PC, DCS (Distributed Control System) or RTU (Remote Terminal Unit). MODBUS RTU slaves are often field devices, all of which connect to the network in a multi-drop configuration. [4]

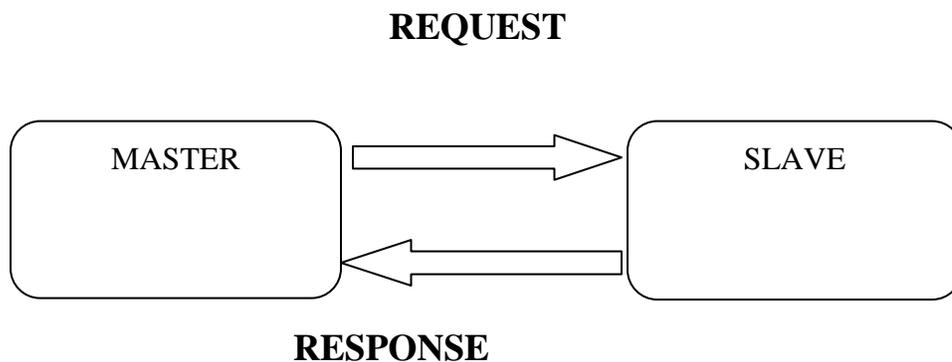


Fig 4.1 MODBUS TRANSMISSION

The MODBUS® frames transmitted by the master contain the following information [5]

- The MODBUS® address of the interrogated slave (1 byte)
- The function code defining the master request (1 byte)
- The data to be exchanged (N bytes)
- The CRC16 checksum (2 bytes)

The frame has a maximum length of 240 bytes, thereby enabling the exchange of a maximum of 100 data words or 255 binary data.

In the table 4.1 are the MODBUS codes and their description:

Function codes		Description
In Hexadecimal	In decimal	
01 or 02	01 or 02	Read n bits
03 or 04	03 or 04	Read n words
05	05	Write a bit
06	06	Write a word
07	07	Fast reading of 8 bits
08	08	Diagnosis / Initialization
0F	15	Write n bits
10	16	Write n words

Table 4.1 MODBUS Function codes

The error codes are given in the table 4.2

ERROR CODES	DESCRIPTION
01	Unknown Function Code
02	Address Error
03	Data Error
09	Time out
10	Checksum Error

Table 4.2 MODBUS Error codes

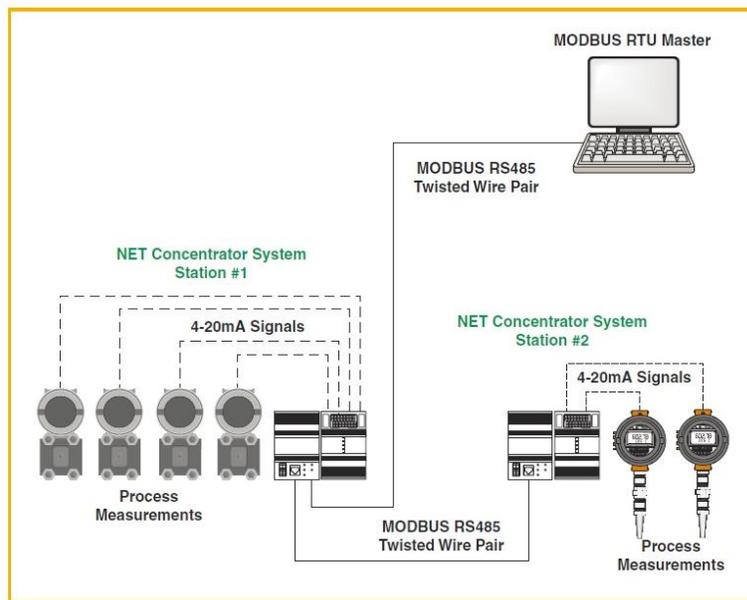


Figure4.2 A MODBUS RTU network

## 4.2.1 USING ABB AC 31 PLC

ABB Advant Controller (AC) 31 has 8 digital inputs and 6 digital outputs. It facilitates three modes of communication namely ASCII, MODBUS and Programming. Experimentation was carried out with **ABB AC 31** in MODBUS RTU, using **AC 31 GRAF** as shown in figure4.3.[8]

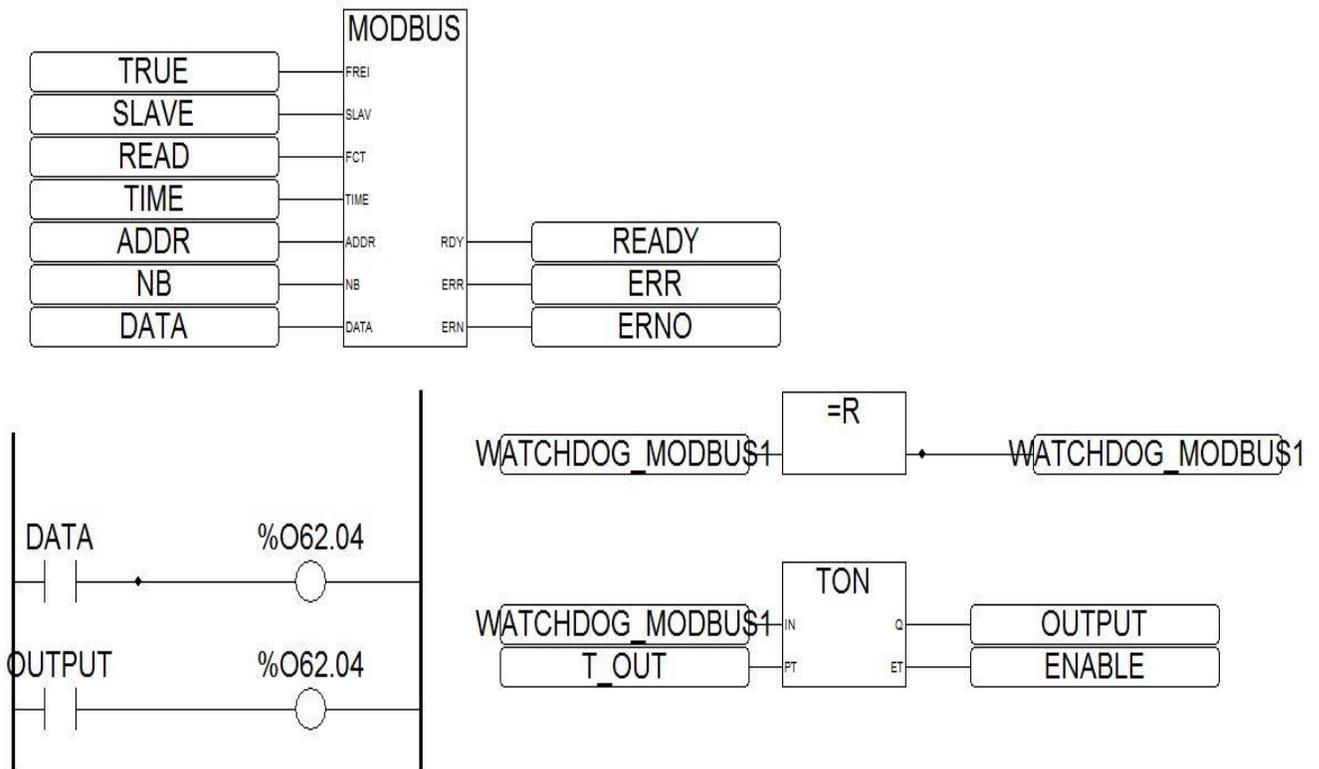


Fig4.3 SNAPSHOT OF MODBUS using AC 31 GRAF

ABB AC 31 is also compatible with the software package **907AC1131**. The MODBUS experimentation integrating bridge PC and PLC is as shown in figure4.4 below

```

0001 PROGRAM PLC_PRG
0002 VAR
0003   EN AT %MX0.0: BOOL;
0004   COM AT %MW1000.0: INT := 1;
0005   SLAVE AT %MW1000.1: INT := 13101;
0006   FUNC AT %MW1000.2: INT := 01;
0007   TOUT AT %MW1000.3: INT := 2;
0008   ADDR AT %IW1000.0: INT := 0000;
0009   NB AT %MW1000.4: INT := 1;
0010   DATA AT %QX0.0: BOOL;
0011   ERR AT %QX0.1: BOOL;
0012   ERNO AT %QW1000.0: INT;
0013   MOD_MAST AT %MW3000.6: MODMASTB;
0014   TON: TON;
0015   Y: TIME;
0016   OUTPUT AT %QX62.1: BOOL;
0017   INPUT AT %MX255.8: BOOL;
0018 END_VAR

```

MOD\_MAST

EN-EN	RDY-EN
COM-COM	ERR-ERR
SLAVE-SLAV	ERNO-ERNO
FUNC-FCT	
TOUT-TIMEOUT	
ADDR-ADDR	
NB-NB	
DATA-DATA	

```

0002

```

ton

ONLINE | OV | READ

Fig4.4 SNAPSHOT OF MODBUS using 907AC1131

### 4.3 SERIAL COMMUNICATION

**IRC 5 controller** enables serial communication with other devices via RS-232. There are four possible ways to communicate via serial channels:

- Messages can be output to the teach pendant display and the user can answer questions, such as about the number of parts to be processed.
- Character-based information can be written to or read from text files on mass memory. In this way, for example, production statistics can be stored and processed later in a PC. Information can also be printed directly on a printerconnected to the robot.
- Binary information can be transferred between the robot and a sensor.
- Binary information can be transferred between the robot and another computer,for example, with a link protocol.

Typical parameters of serial communication are shown in table 4.3

<b>Parameters</b>	<b>Values</b>
Baud rate	9600
Data bits	8
Stop bits	1
Parity bits	None

Table 4.3 Serial Parameters

#### 4.4 ARDUINO COMMUNICATION

An Arduino board consists of an Atmel 8-bit AVR microcontroller with complementary components to facilitate programming and incorporation into other circuits. Official Arduino's have used the megaAVR series of chips, specifically the ATmega8, Atmega168, Atmega328, Atmega1280, and Atmega2560. Most boards include a 5 volt linear regulator and a 16 MHz crystal oscillator.

Arduino UNO (ATMega328P-PU) is used for interfacing LabVIEW with the Programmable Logic Controller (PLC). The control signals flowing in a loop are shown in the figure 4.3 below.

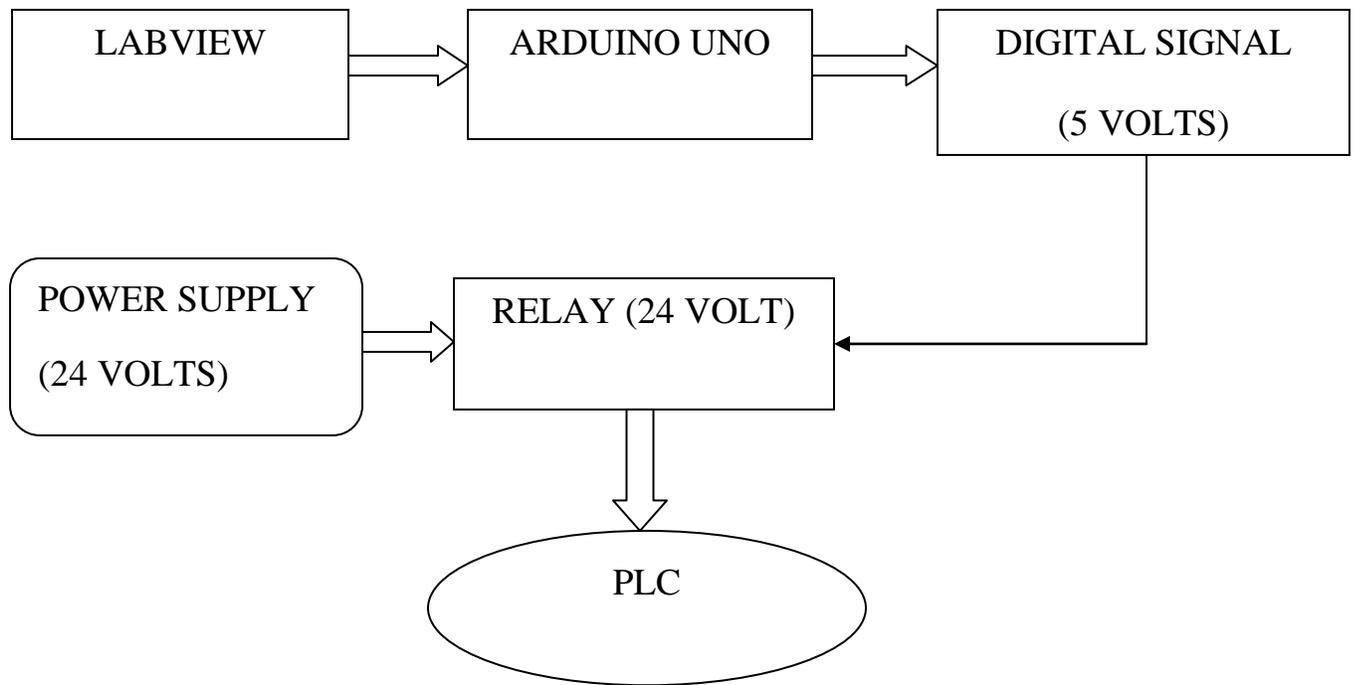


Figure 4.5 Flow Loop

# CHAPTER 5

## SOFTWARE AND PROGRAMMING

### 5.1 SOFTWARE

The various software packages used for the communication are DELTA WPLSoft which is used for the PLC Programming, ROBOT STUDIO for IRB1410 Robot Programming.

#### 5.1.1 DELTA-WPL SOFT

DELTA's WPLSoft is a freeware which is given by DELTA for programming its own PLCs. The programming is done in ladder logic diagram. There are two main parts of programming. One is communication mode and the other ladder diagram to control various inputs/outputs.

For the communication programming inWPLSoft, the following steps are followed. Delta WPLSoft is opened and New project is created.

1. The PLC Model name is selected in the first window.

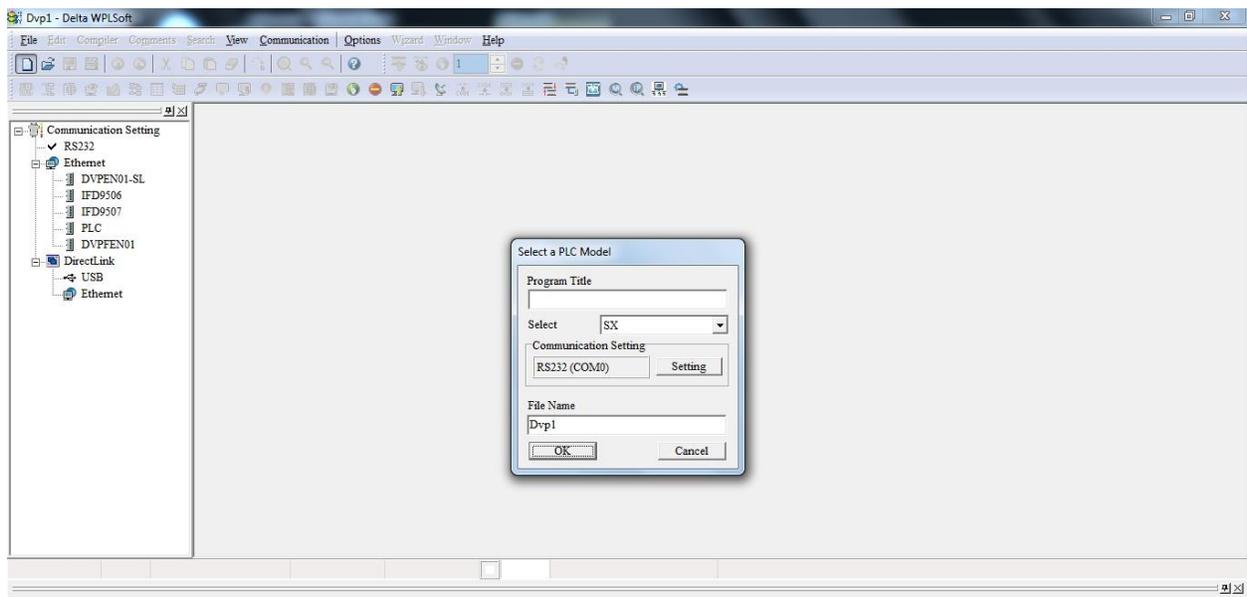


Figure 5.1 PLC Communication Step1

## 2. The communication settings are configured.

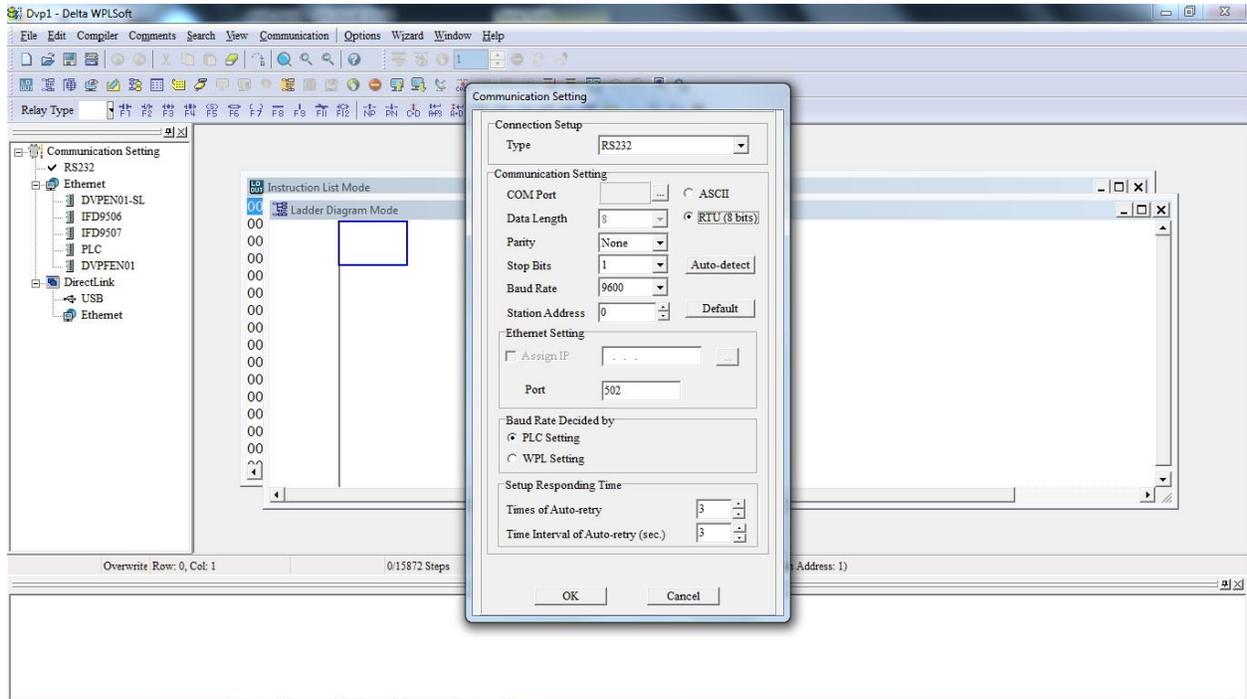


Figure 5.2 PLC Communication Step2

## 3. Communication program is selected to configure the communication model using the RS485(COM2) connection of the PLC.

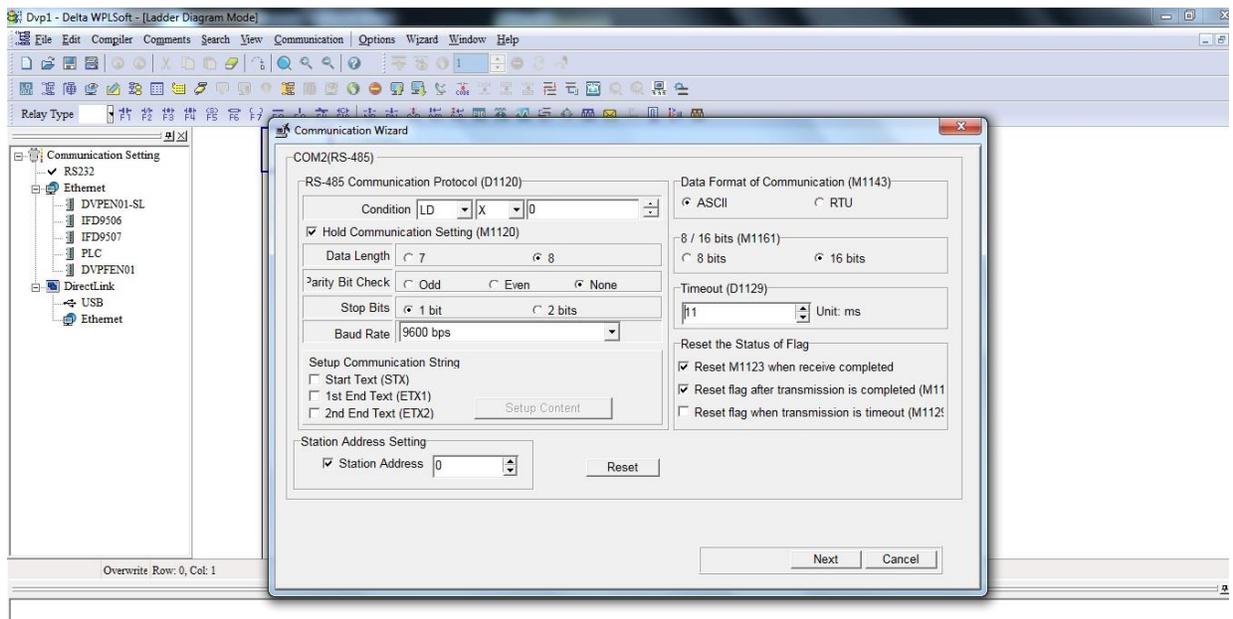


Figure 5.3 PLC Communication Step3

4. Then, the type of protocol to be used, here MODBUS, is selected.

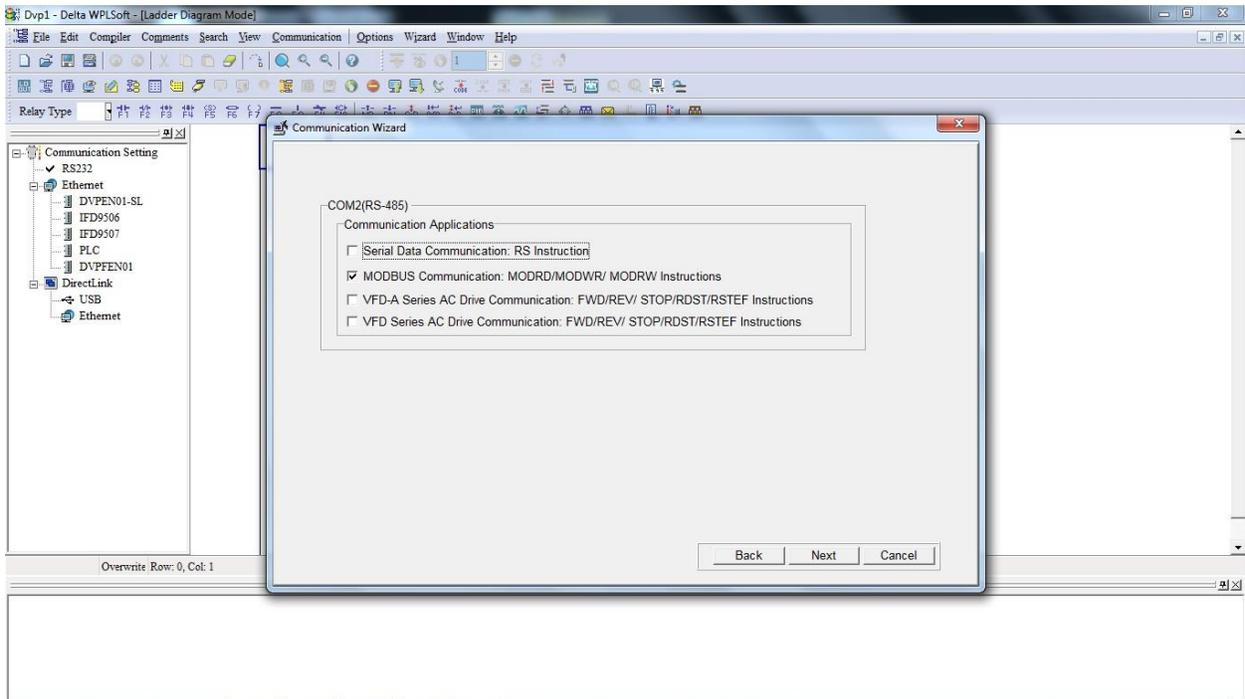


Figure 5.4 PLC Communication Step4

5. Finally, the program is generated using the various instructions.

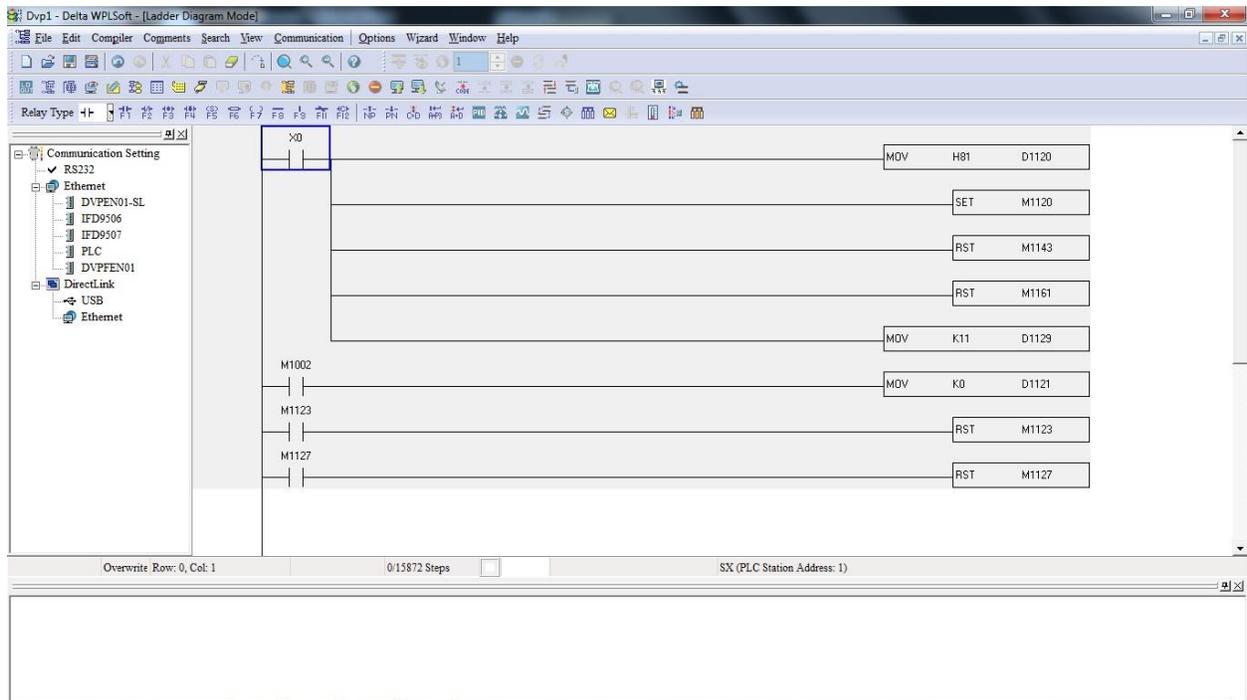
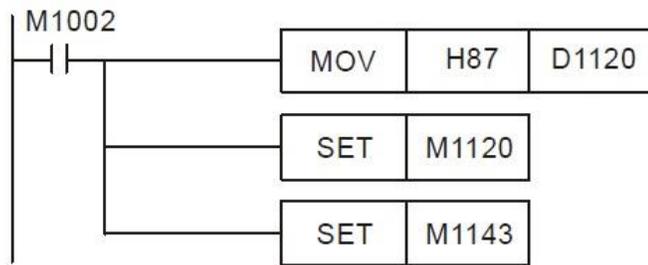


Figure 5.5 PLC Communication Step5

### Modifying COM2 communication format:

1. Add the below instructions on top of the program to modify the communication format of COM2. When PLC switches from STOP to RUN, the program will detect whether M1120 is ON in the first scan. If M1120 is ON, the program will modify the communication settings of COM2 according to the value set in D1120.
2. Set the flag M1143 ON to enable RTU mode or OFF to enable ASCII mode.
3. Modify COM2 communication format to ASCII mode, 9600bps, 7 data bits, even parity, 1 stopbits (9600, 7, E, 1)

**COM2:**



The various instructions and registers accessed are given in Appendix 3. [6]

The program is flashed into the PLC module via RS232 cable.

## **5.1.2 ROBOT STUDIO**

### **5.1.2.1 FLEX PENDANT**

The decision whether to use character-based or binary information is dependent on how the equipment with which the robot communicates handles that information. A file, for example, can have data that is stored in character-based or binary form.

If communication is required in both directions simultaneously, binary transmission is necessary.

Each serial channel or file used must first be opened. On doing this, the channel/file receives a descriptor that is then used as a reference when reading/writing. The teach pendant can be used at all times and does not need to be opened.

The flex pendant instructions used in the programming are given in Appendix 1.



## 5.2 PLC PROGRAMMING

### PLC PROGRAM FOR CONTROLLING I/O

After the communication via RS485 is done and already shown in chapter 4, the next step in PLC logic is to control the inputs and outputs. The signal from the communication part is stored in register X0. Thus this is accessed to control the other outputs such as the conveyor motor, IRC-5, bench vice and vice door. The Ladder logic program is given in figure 5.2

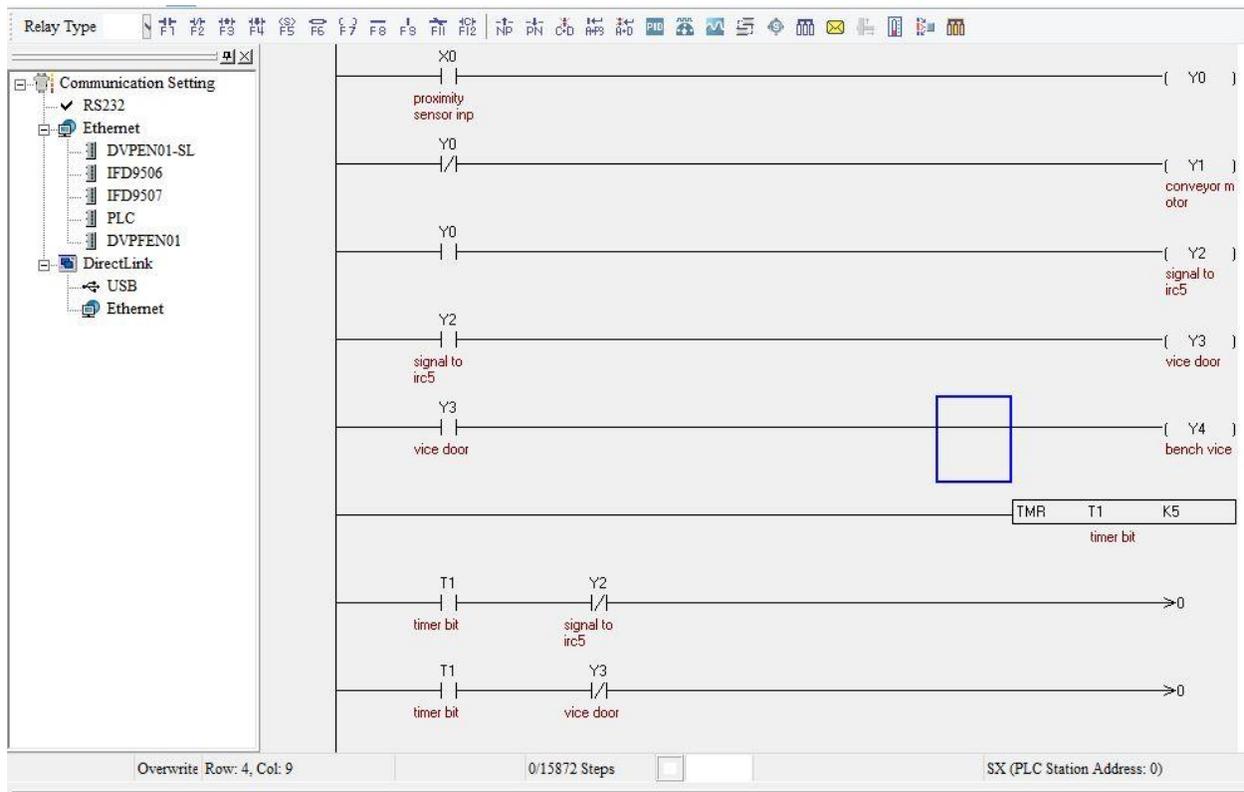


Figure 5.7 SNAPSHOT OF Ladder Diagram

### 5.3 IRC-5 PROGRAMMING

MODULE MainModule

VAR iodev channel;

PERS num reg\_new:=20;

PERS string string1:="";

VAR signaldo DO2;

PROC main()

Open "COM1",channel\Read\Bin;

ClearIOBuff channel;

string1:=ReadStrBin(channel,reg\_new\Time:=5);

TPWrite string1;

Close channel;

ERROR

IF ERRNO=ERR\_DEV\_MAXTIME THEN

TPWrite " data Not received ";

ENDIF

IF string1="rec" THEN

MoveJ[[223.69,1172.98,737.00],[0.486723,-  
0.494155,0.597299,0.402673],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,  
9E+09]], v1000, z50, tool0;

!move to the target object

SetDO DO2, 1;

!gripper close and pick

MoveJ[[-13.22,-1226.56,817.27],[0.42813,0.435824,0.601851,-0.514332],[-  
2,-1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v1000, z50, tool0;

!move to the target position

SetDO DO2, 0;

!gripper close and place

MoveL[[-10.28,-1076.89,873.60],[0.42824,0.435696,0.601944,-0.514241],[-  
2,-1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v1000, z50, tool0;

!arm backward movement

WaitTime 5;

MoveJ[[-13.11,-1252.03,802.76],[0.42824,0.435696,0.601944,-  
0.514241],[-2,-1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v1000,  
z50, tool0;

!arm forward movement

SetDO DO2, 1;

!gripper close and pick

MoveJ[[1141.13,108.74,853.17],[0.612147,-0.412649,0.617971,-  
0.270388],[0,-1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v1000,  
z50, tool0;

!arm movement towards object holder

SetDO DO2, 0;

!gripper open and place

MoveJ[[717.97,71.05,1313.46],[0.770942,-0.304889,0.428135,-  
0.359711],[0,-1,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v1000,  
z50, tool0;

!arm moves to the home position

ENDIF

ENDPROC

ENDMODULE

## **CHAPTER 6**

### **CONCLUSION**

This chapter discusses the final project results. It will also illustrate the accomplishments and contributions of this study with regards to the objectives set out in Chapter 1. Problems that were encountered during the project's development as well as future extensions to the project will be highlighted.

#### **6.1 PROJECT RESULTS**

Integration and control of an autonomous robot was created to provide industry with a more flexible approach to remote automated robot systems. A standard industrial ABB robot, PLC and bridge PC were used to test the project results.

The entities were linked together by the following ways:

- 1. PC AND PLC:** The Bridge PC and PLC were linked by using MODBUS protocol. Both the entities are equipped with RS-232 terminals, which are used for MODBUS serial communication as detailed in **chapter 4.2**. Software to enable communication between the bridge PC and robot controller was also established
- 2. PC AND ROBOT CONTROLLER:** The Bridge PC and the robot controller are interfaced via open serial communication. The platform for communication were established using LabVIEW (in PC) and Flexpendant (in robot controller). Software to enable communication between the bridge PC and robot controller was also established. (RAPID programming).

## 6.2 PROBLEMS ENCOUNTERED

Various problems were encountered while interfacing the modules:

- INTERFACING OF ABB AC 31 WITH PC:

The ABB PLC (AC 31) enables MODBUS communication. When experimentation was made interface via MODBUS RTU, Handshaking problems were encountered, since there was **only one serial port** (RS-232) for program uploading and LabVIEW communication.

- INTERFACING OF ROBOT CONTROLLER WITH PC:

The robot controller (IRC 5) enables serial communication to interface with bridge PC. When experimentation was made to integrate them, the serial port (COM1) failed to bind for several times, in spite of software enabling.

## 6.3 FUTURE SCOPE OF PROJECT

- Profile recognition of the object to be sorted, using advanced image processing techniques.
- PC based automated robot trajectory system, by tracking the object coordinates using the vision system based CCD camera.
- The Ethernet communication environment provides the ideal platform for real-time position control. By adding additional intelligence software to the current system the robot can automatically follow a moving object and make intelligent decisions dependent on the vision feedback.

**APPENDICES**  
**APPENDIX A**  
**FLEX PENDANT INSTRUCTIONS**

**Communicating using the teach pendant**

Instruction	used to:
<i>TPErase</i>	Clear the teach pendant operator display
<i>TPWrite</i>	Write text on the teach pendant operator display
<i>ErrWrite</i>	Write text on the teach pendant display and simultaneously store that message in the program's error log.
<i>TPReadFK</i>	Label the function keys and to read which key is pressed
<i>TPReadNum</i>	Read a numeric value from the teach pendant
<i>TPShow</i>	Choose a window on the teach pendant from RAPID

**Reading from or writing to a character-based serial channel/file**

Instruction	Used to:
<i>Open</i>	Open a channel/file for reading or writing
<i>Write</i>	Write text to the channel/file
<i>Close</i>	Close the channel/file
Function	Used to:
<i>ReadNum</i>	Read a numeric value
<i>ReadStr</i>	Read a text string

## **Communicating using binary serial channels/files**

Instruction	used to
<i>Open</i>	Open a serial channel/file for binary transfer of data
<i>WriteBin</i>	Write to a binary serial channel/file
<i>WriteStrBin</i>	Write a string to a binary serial channel/file
<i>Rewind</i>	Set the file position to the beginning of the file
<i>Close</i>	Close the channel/file
Function	Used to:
<i>ReadBin</i>	Read from a binary serial channel

## **Data for serial channels**

Data type	Used to define:
<i>IODEV</i>	A reference to a serial channel/file, which can then be used for reading and writing

## **APPENDIX B**

### **PLC REGISTERS AND RELAYS**

X(Input Relay)	Bit memory represents the physical input points and receives external input signals. X0~X7,X10~X17...X377
Y(Output Relay)	Bit memory represents the physical output points and saves the status to be refreshed to physical output devices. Y0~Y7,Y10~Y17...Y377
M(Internal Relay)	Bit memory indicates PLC status. M0, M1,M2...M4095
S(Step Relay)	Bit memory indicates PLC status in Step Function Control (SFC) mode. If no STL instruction is applied in program, step point S can be used as an internal relay M as well as an annunciator. S0, S1,S2...S1023
T(Relay)	Bit, word or double word memory used for timing and has coil, contact and register in it. When its coil is ON and the set time is reached, the associated contact will be energized. Every timer has its resolution (unit:1ms/10ms/100ms). T0, T1,T2...T255

## C(Counter)

Bit, word or double word memory used for counting and has coil, contact and register in it. The counter counts once (1 pulse) when the coil goes from OFF to ON. When the predefined counter value is reached, the associated contact will be energized. There are 16-bit and 32-bit high-speed counters available for users.

C0, C1, C2...C255

## D(Data Register)

Word memory stores values and parameters for data operations. Every register is able to store a word (16-bit binary value). A double word will occupy 2 consecutive data registers.

D0, D1, D2...D4999

## E, F(Index register)

Word memory used as a modifier to indicate a specified device (word and double word) by defining an offset. Index registers not used as a modifier can be used as general purpose registers.

**E0 ~ E7 and F0 ~ F7.**

**APPENDIX C**  
**REGISTERS AND INSTRUCTIONS ACCESSED IN DELTA PLC**  
**PROGRAMMING**

**M1002**

Enables single positive pulse for the first scan when PLC RUN is activated.  
Used to initialize registers, outputs, or counters when RUN is executed.

**M1120**

Retaining the communication setting of COM2 (RS-485), modifying D1120 will be invalid when M1120 is set.

**M1123**

Data receiving completed. When data receiving of communication instructions is completed, M1123 will be ON. Users can process the received data when M1123 is ON. When data processing is completed, M1123 has to be reset by users.

**M1127**

COM2 (RS-485) data sending/receiving/converting completed. RS instruction is NOT supported. Supported communication instructions: MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW

**M1143**

For COM2(RS-485), ASCII/RTU mode selection (OFF: ASCII; ON: RTU)

**M1161**

8/16 bit mode (ON = 8 bit mode)

**D1120** COM2 RS485 Communication Protocol

# APPENDIX D

## ARDUINO PROGRAMMING

### Source Code For Labview Arduino Integration

```
/*  
*****  
  
** Functions  
  
*****  
*****/  
  
// Writes Values To Digital Port (DIO 0-13). Pins Must Be Configured As Outputs  
Before Being Written To  
  
void writeDigitalPort(unsigned char command[])  
{  
  
    digitalWrite(13, (( command[2] >> 5) & 0x01) );  
  
    digitalWrite(12, (( command[2] >> 4) & 0x01) );  
  
    digitalWrite(11, (( command[2] >> 3) & 0x01) );  
  
    digitalWrite(10, (( command[2] >> 2) & 0x01) );  
  
    digitalWrite(9, (( command[2] >> 1) & 0x01) );  
  
    digitalWrite(8, (command[2] & 0x01) );  
  
    digitalWrite(7, (( command[3] >> 7) & 0x01) );  
  
    digitalWrite(6, (( command[3] >> 6) & 0x01) );  
  
    digitalWrite(5, (( command[3] >> 5) & 0x01) );  
  
}
```

```

digitalWrite(4, (( command[3] >> 4) & 0x01) );
digitalWrite(3, (( command[3] >> 3) & 0x01) );
digitalWrite(2, (( command[3] >> 2) & 0x01) );
digitalWrite(1, (( command[3] >> 1) & 0x01) );
digitalWrite(0, (command[3] & 0x01) );

}

// Reads all 6 analog input ports, builds 8 byte packet, send via RS232.

void analogReadPort()
{
    // Read Each Analog Pin

    int pin0 = analogRead(0);

    int pin1 = analogRead(1);

    int pin2 = analogRead(2);

    int pin3 = analogRead(3);

    int pin4 = analogRead(4);

    int pin5 = analogRead(5);

    //Build 8-Byte Packet From 60 Bits of Data Read

    char output0 = (pin0 & 0xFF);

    char output1 = ( ((pin1 << 2) & 0xFC) | ( (pin0 >> 8) & 0x03) );

    char output2 = ( ((pin2 << 4) & 0xF0) | ( (pin1 >> 6) & 0x0F) );

```

```
char output3 = ( ((pin3 << 6) & 0xC0) | ( (pin2 >> 4) & 0x3F) );
```

```
char output4 = ( (pin3 >> 2) & 0xFF);
```

```
char output5 = (pin4 & 0xFF);
```

```
char output6 = ( ((pin5 << 2) & 0xFC) | ( (pin4 >> 8) & 0x03) );
```

```
char output7 = ( (pin5 >> 6) & 0x0F );
```

```
// Write Bytes To Serial Port
```

```
Serial.print(output0);
```

```
Serial.print(output1);
```

```
Serial.print(output2);
```

```
Serial.print(output3);
```

```
Serial.print(output4);
```

```
Serial.print(output5);
```

```
Serial.print(output6);
```

```
Serial.print(output7);
```

```
}
```

```
// Synchronizes with LabVIEW and sends info about the board and firmware  
(Unimplemented)
```

```
void syncLV()
```

```
{
```

```
Serial.begin(DEFAULTBAUDRATE);
```

```
i2cReadTimeouts = 0;  
  
spiBytesSent = 0;  
  
spiBytesToSend = 0;  
  
Serial.flush();  
  
}
```

## REFERENCES

- [1] B. Pokharel, "MACHINE VISION AND OBJECT SORTING: PLC Communication with LabVIEW using OPC," 2013.
- [2] A. S. Ashtekar, B. J. Parvat, and C. B. Kadu, "Application of MODBUS to Communicate the PLC and Lab VIEW for Real Time Process Control."
- [3] S. Aleksandrov, Z. Jovanović, S. Nikolić, and S. Čajetinac, "Mechatronic systems control based on SCADA system, OPC server and LabView," *FACTA Univ. Ser. Autom. Control Robot.*, vol. 10, no. 2, pp. 189–198, 2011.
- [4] J. McConahaV, "Using Modbus for Process Control and Automation," *measurements*, vol. 100, p. 1, 2011.
- [5] Modbus Application Protocol Specification V1.1b, Modbus-IDA, December 2006, pp. 1-51, [Online] available at <http://www.modbus-IDA.org>.
- [6] [www.kalatec.com.br/plc/MANUAL/EXEMPLOS\\_DE\\_APLICACAO-CLP\\_DVP.pdf](http://www.kalatec.com.br/plc/MANUAL/EXEMPLOS_DE_APLICACAO-CLP_DVP.pdf)
- [7] [www.delta.com.tw/product/em/control/plc/download/manual/Delta\\_PLC-Program\\_O\\_EN\\_20130530.pdf](http://www.delta.com.tw/product/em/control/plc/download/manual/Delta_PLC-Program_O_EN_20130530.pdf)
- [8] [www05.abb.com/global/scot/scot397.nsf/veritydisplay/a7fa1d41a1a75ad2c1257c21005399c8/\\$file/2CDC120067M0202.PDF](http://www05.abb.com/global/scot/scot397.nsf/veritydisplay/a7fa1d41a1a75ad2c1257c21005399c8/$file/2CDC120067M0202.PDF)
- [9] <http://dspace.nmmu.ac.za:8080/xmlui/bitstream/handle/10948/213/WillDJ.pdf?sequence=1>

[10]<http://www.delta.com.tw/product/em/control/plc/download/manual/SX%20Instruction%20Sheet-English-20060112.pdf>

[11] <http://www.automation-drive.com/EX/05-14-07/1761-cbl-pm02.gif>