MACHINE VISION AND OBJECT SORTING

PLC Communication with LabVIEW using OPC

HAMK
UNIVERSITY OF APPLIED SCIENCES

Bachelor's thesis

Degree Programme in Automation Engineering

Valkeakoski, May 2013

Bikarna Pokharel

HAMK
UNIVERSITY OF APPLIED SCIENCES

ABSTRACT

Automation Engineering
D.P. in Automation Engineering

Author Bikarna Pokharel **Year** 2013

Subject of Bachelor's thesis Machine Vision And Object Sorting. PLC Communication with LabVIEW using OPC

ABSTRACT

The main objective of this thesis was to demonstrate a machine vision (MV) application for the quality control of products in an industry using a miniature production line station. A project was commissioned by HAMK University of Applied Sciences, Valkeakoski, and conducted in the automation laboratory for this thesis.

A miniature production line was constructed and automated using STEP 7 software, and Simatic-300 based Programmable Logical Controller (PLC); both are manufactured by Siemens. The machine vision part of the thesis was done in NI LabVIEW, a graphical program development environment from National Instruments (NI). The communication between the PLC and LabVIEW was carried out using NI OPC (OLE for Process Control), an add-on for LabVIEW.

Basic theoretical background on machine vision is presented at the beginning of this thesis; this is followed by a description of the vision development module in LabVIEW, and a method to communicate between a PLC and LabVIEW. At the end, the program developed to demonstrate machine vision is described.

Object sorting using machine vision was accomplished using pattern matching algorithm of machine vision. A pattern image template was created and stored into the memory of computer. 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 thesis was successfully completed, and most of the objectives as to the machine vision application demonstration for object sorting were met. In addition, the communication between the PLC and LabVIEW was enabled using NI OPC.

Keywords Machine vision, Object sorting, NI vision, NI OPC communication.

Pages 50 p. + appendices 16 p.

CONTENTS

1 INTRODUCTION

Machine vision (MV) is a young discipline in the field of science and technology. It has emerged as a useful industrial tool for about 25 years and is growing at a higher speed. (Davies 2012, Foreword.)The applications of machine vision in industries have been typically seen in measurements, counting, quality control, object sorting, and robotic guidance. It has become a yielding tool in product inspection and analysis, because it reduces cost, effort, and time with a significant level of accuracy and reliability.

The development of machine vision technology directly relates to the development of optical systems. The development of modern time optics was in the late 1800 whereas in the ‘80s a new era of optics called digital image processing and machine vision systems began to flourish. (Vision Systems Oy. 2012.)

Digital image processing involves extraction of information from an image (Wisegeek, 2011. What is Machine Vision Image Processing?). it is the backbone of machine vision systems. Digital images are the combinations of different light intensity level. Each point in a digital image is a representation of pixel value that corresponds that corresponds to x- and y-coordinates in the image plane; it defines the intensity at that point.

A monochrome digital image generally contains 256 gray levels from 0 through 255; 0 corresponds to white color and 255 corresponds to black color. On the other hand, a photograph contains infinite gray levels. (Gonzalez, Woods & Eddins 2009.) Figure 1 shows the gray levels of a digital image (a) and a traditional photograph (b).

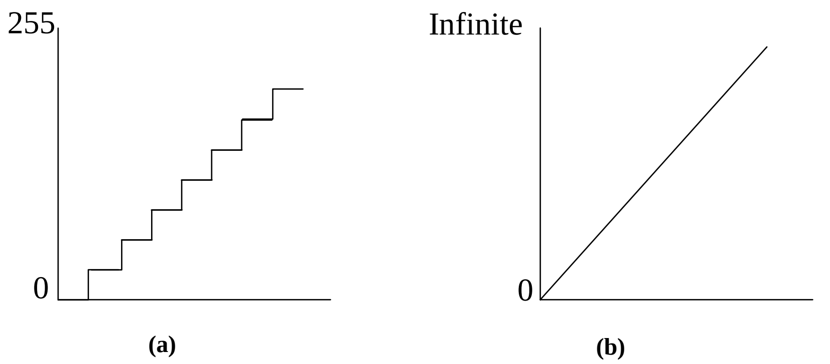


Figure 1 Gray levels of a digital image (a), and a photograph (b) (Pokharel, 2013)

Machine vision is meant for real-time image processing. This implies that, the image processing is done at the instant the image is acquired. Thus, it represents a real-world image in real time. It extracts the 3D information from the 2D image of the object.

Device and software communications are also widely practised in industries. With the advancement in communications and networking, different devices, instruments, and software in the industries these days can be interconnected. Today different communication protocols are being used in industries such as PROFIBUS, PROFINET, Ethernet, OPC, etc.

This thesis was conducted mainly to demonstrate machine vision tool for sorting of objects, and pattern-matching tool was used to sort the objects. Pattern matching is the first step in a machine vision application. The sorting of objects for this thesis project had only two demands; correct objects and damaged objects. So, it should be enough if the objects patterns are only matched. Pattern matching is easy to apply as it has been said by experts that pattern matching is not affected by lighting, motion, and noise.

The secondary part of the thesis was PLC communication with LabVIEW. To do so, an automation networking protocol called OPC was used. Many other ways can be applied to communicate between PLC and LabVIEW such as using other hardware communication devices, but the sorting station already had a PLC, and the input and output signal modules connected, use of a software protocol was found easier.

The forthcoming sections and sub-sections consist of issues that are related to machine vision components, machine vision process, developing a machine vision application, deploying the application to the desired control station, and a method for the industrial communication between PLC and LabVIEW.

1.1 Machine vision today

With the recent advancement in technology, machine vision can be applied to extract different properties of the objects such as their dimensions, areas, etc. The application of machine vision has been seen in medical, industrial, and security fields. The application made for one machine vision purpose might be also applied for a different but similar purpose; for example, the application made for object sorting can also be used for object tracking or geometric classification. Table 1 shows some machine vision applications in different fields.

Table 1 Some applications of machine vision

Medical Fields		Industrial Fields		Security fields	
1	Dermatological diagnosis	1	Object sorting	1	Face recognition
2	Treatment of skull abnormalities in children	2	Robotic guidance	2	Object tracking
		3	Object detection	3	Security administration and surveillance
		4	Object inspection		
		5	Pattern recognition		
		6	Edge detection		

Machine vision is a broad term. Many mathematical theories, image acquisition, image processing and analysis, etc. form the whole machine vision world. Therefore, to define machine vision is a difficult task, if these things are considered. Pages and pages could be written on these different topics. For this reason, the main concentration of this thesis document was to develop a simple machine vision application using software readily available in the market. The software used in this thesis project was LabVIEW. However, the basic requirements to understand a machine vision system are discussed in the forthcoming sections.

2 COMPONENTS OF MACHINE VISION SYSTEM

A machine vision system typically consists of machine vision software (machine vision tools) and a camera (image acquisition device). But many other things need to be considered for a machine vision system. Each of the components has its own significance. So none of the components can be isolated or segregated. The important components are discussed in brief in the following sub-sections.

2.1 Illumination

Illumination refers to the light sources that are available around the object being analysed. It is significant that the object(s) under analysis be clearly visible to the image acquisition device. It ensures that much of the information is retained in the acquired image, and no much image processing needs to be done; thus making the machine vision application simpler to develop.

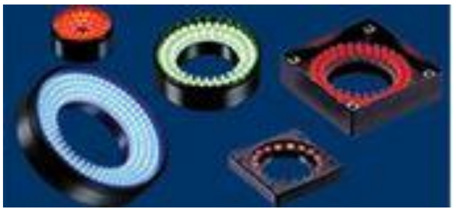
Illuminating object(s) does not mean availability of huge amount of light around the object; it refers the lights to be adjusted in a proper way. Proper illumination involves the right intensity and correct direction of light. It should be done in a way that shadow formation is checked and maximum contrasts can be achieved from the region of interest of the object(s). (Movimed custom imaging solutions, 2007.)

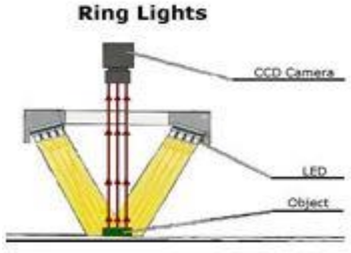
The light sources may be fluorescent lights or LED lights or halogen lights, etc. LED lights are more preferred over the other types of light sources, because of their long life and less energy consumption. Depending upon the arrangement of lights, illumination can be direct or indirect.

2.1.1 Direct illumination

Direct illumination involves illuminating specific regions of the object(s) being analysed such that those regions can be identified clearly. These regions may be the centre or edges of the object. Direct lighting can be obtained using ring lights, low angle ring lights or bar lights. (Moviemed imaging solutions, 2007.)

High-density LED ring lights are perfect for illumination if central region or the edges of the object needs to be analysed (Movimed imagine solutions, 2007). Figure 2 below shows ring lights for illuminating the central region of the object.

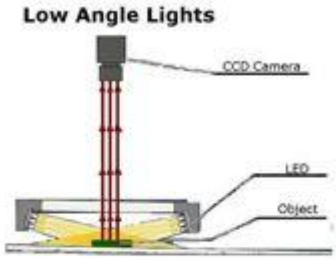




(a) **(b)**

Figure 2 Different ring lights (a) and ring- light assembly (b) (Movimed custom imaging solutions).

However for edge illumination, low angle ring lights should be considered. Figure 3 shows ring lights and their assembly.



(a) **(b)**

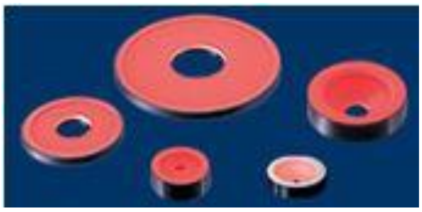
Figure 3 Different low- angle ring lights (a), and their assembly (b) (Movimed custom imaging solutions).

High intensity LED lights arranged to rectangular oblique illumination units can also be used for direct lighting instead of ring lights and low- angle ring lights as shown in Figure 4 (Movimed custom imaging solutions, 2007).

Figure 4 Bar lights for direct illumination (Movimed custom imaging solutions).

2.1.2 Indirect illumination

When the illumination is done all over the object, it is referred to as indirect lighting. Indirect lighting can be achieved in several ways such as using backlights, line lights, flat-ring lights, etc. Figure 5 below shows some indirect lighting source types.



(a)

(b)

(c)

Figure 5 Different types of indirect lighting sources; (a) Flat-ring lights, (b) Line lights, (c) Back lights (Movimed custom imaging solutions).

2.2 Image acquisition

Image acquisition is the most important part in a machine vision system. It involves capturing an image of the object to be analysed with the help of camera. Different types of cameras can be used for image acquisition; they can include an ordinary mobile camera, a typical digital camera, or even a webcam. But cameras that are tailored specially for industrial use are also available. Depending on the sensor technology used, different cameras can be classified into two categories as follows:

i. CMOS cameras
ii. CCD cameras

“The sensors could be matrix sensors or line sensors. An image sensor converts an optical image into an electronic signal” (Wikipedia, 2013. Image sensor).

Choosing a machine vision camera can be a difficult task. However, resolution, sensitivity, and type of camera-monochrome or color, should be considered when buying one. (ALLIED Vision Technologies GMBH, 2006.) Also, the interface the camera uses for communication should be considered. The available interfaces include USB, Ethernet, Firewire, etc.

2.2.1 CCD cameras

A CCD (Charge-coupled Device) camera uses the CCD sensor technology. The main features of these sensors (and hence cameras) are listed as follows. (Vision Systems Oy. 2012.)

i. The most common camera sensors.
ii. In-coming charges are stored.

iii. Equivalent to films of traditional film cameras.
iv. Consist of pixels with a typical size of 10µm x 10µm.
v. These are both color and monochrome.
vi. These are light-sensitive diode sensors.
vii. Each pixel has a micro-lens for focusing the light into the sensor surface.
viii. Disadvantage of these sensors is the possibility of over exposure.

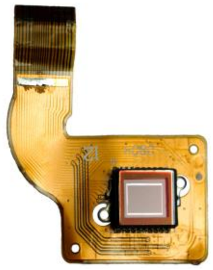


Figure 6 CCD image sensor on a flexible circuit board (Wikimedia Commons).

Figure 6 alongside shows a CCD sensor.

2.2.2 CMOS cameras

CMOS cameras are alternatives to CCD cameras. The main features of a CMOS sensor are listed as follows (Vision Systems Oy. 2012).

i. These sensors help to capture images at very high frame speed of more than 1000 frames/second.
ii. The manufacturing cost is very low.
iii. These sensors have low power consumption.
iv. Windowing, selecting a part of sensor, is possible with these sensors.
v. Disadvantage of these sensors is poor image quality.

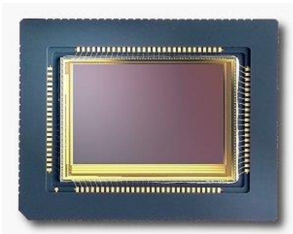


Figure 7 CMOS image sensor (Sudcamp. A Technology Hub).

Figure 7 alongside shows a CMOS sensor.

2.2.3 Line-scan cameras

When products are manufactured in a factory, they are frequently moved from one stage to another on a conveyor. Stopping the conveyor to acquire an image for inspection would impose unwanted design problems. So, the conveyor is adjusted to a reasonably uniform speed. (Davies 2012, 720.) For acquiring images in such situations, line-scan cameras are used. As factories often have many conveyor belts, these cameras must be widely used cameras in factories and industries. Figure 8 below shows the use of line-scan camera in a production line where the conveyor is running.

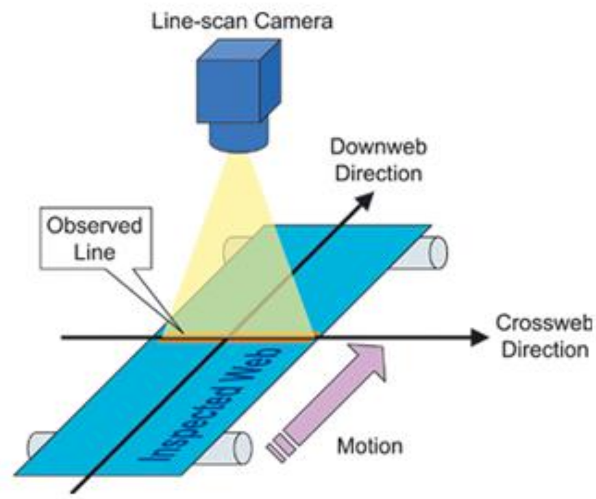


Figure 8 Use of Line-scan camera for object analysis over a conveyor (EURESYS ADR Technology: Advanced Downweb Resampling).

Line-scan cameras consist of row of photocell on a single integrated circuit sensor. As with the name of this type of cameras suggests, the lighting should be selected such that it follows the same kind of symmetry as the camera. The most obvious lighting would be long tube lights parallel to the line of camera and perpendicular to the direction of motion of conveyors. (Davies 2012, 730.)

2.3 Camera calibration

Camera calibration estimates the intrinsic and extrinsic parameters of a camera. Usually the camera parameters are represented in a 3X4 matrix called the camera matrix. It is directly dependent on the optical design of the camera. The intrinsic parameters are the focal length, pixel coordinates and principal points whereas the extrinsic parameters are relative position of camera, and rotational matrix and linear vector of camera. (Wikipedia, 2013. Camera resectioning.)

The method for calibrating a camera is not included in this document. But, the camera may be calibrated using the calibration algorithms available in the machine vision software.

2.4 Lens optics

The cameras for machine vision works on the basis of pin-hole camera model. This model assumes that the perspective rays pass through an infinitesimally small aperture at the front of the camera. But, in practice, the aperture must be larger to admit more light. Lenses are placed in the aperture to bundle of rays from each point on the scene onto the corresponding point in the image as indicated by the geometry of perspective projection. A lens gathers more light, allowing the camera to work with less ambient illumination or with a faster shutter speed, but the depth of field is limited. (Jain, Kasturi & Schunck 1995, 249.)

It might be necessary to calculate the depth of field in order to select a camera lens for a particular machine vision application. The lens equation and the image resolution are required for this calculation. The lens equation and image resolution are explained in brief in the following sub-sections. (Jain et al. 1995, 250.)

2.4.1 Lens equation

The lens equation (2.4.1) relates the distance z' (image distance) of the image plane from the centre of the lens (optical origin, O), the distance z (object distance) to the point in the scene, and the focal length fl of the lens as (Jain et al. 1995, 250)

$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{fl}. \quad (2.4.1)$$

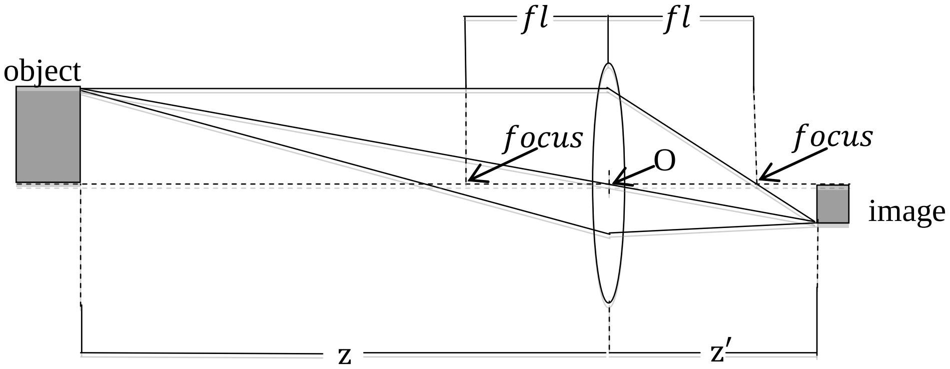


Figure 9 Mechanism of image formation by a lens (Pokharel, 2013)

Figure 9 above shows the general mechanism of image formation by lens.

In optical systems, the important variables are f-number, aperture size, luminance, focal length, field of view, and the depth of field.

2.4.2 f-number

'f-number' is defined as the ratio of focal length to the diameter of the aperture; aperture is only a hole that allows light to enter the lens (Basic Photography Tutorials, 2001). Normally, lenses have adjustable diaphragm as shown in Figure 10 that changes the size of the aperture and thus limits the objective size or aperture size (Vision Systems Oy. 2012.)

Figure 10 showing adjustable diaphragm (Pokharel, 2013)

2.4.3 Field of view

Field of view (FOV) may be defined as the extent to which a camera can view the real world image in an instant of time. It depends on the focal length of the lens. With smaller focal length, the angle of view is larger. This allows the lens to cover wider area of the image.

Figure 11 shows the schematic diagram of FOV. The region surrounded by the height and width of the scene falls inside the FOV.

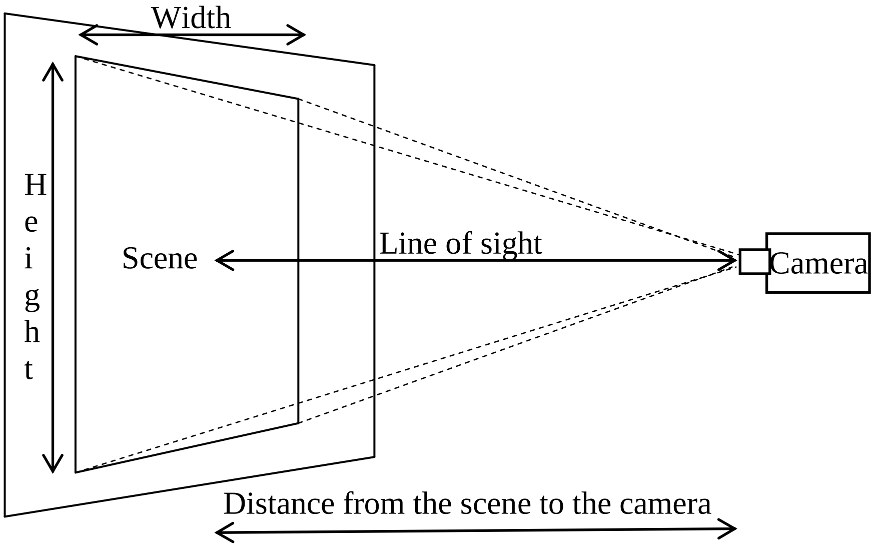


Figure 11 Showing the field of view of a camera lens (Pokharel, 2013)

2.4.4 Depth of field

Depth of Field (DOF) is the measure of the distance between the nearest and the farthest objects in a scene that can be captured by the camera and be acceptably sharp in the image (Wikipedia, 2013. Depth of Field). Figure 12 illustrates the DOF of a lens.

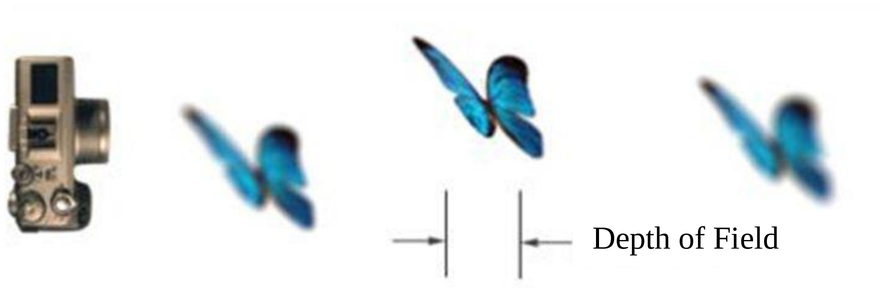


Figure 12 Depth of field of a camera (Wikipedia. Depth of Field)

In Figure 12 above, the area within the DOF appears sharp whereas the areas in front and back of it appear blurry.

It depends on the focal length and the aperture size of the imaging device. The larger the focal length, smaller is the DOF; and also the larger the aperture size, smaller is the DOF.

Although the lens can focus at one point at a time, the gradual decrease in sharpness occurs on both sides of the focus point. So, within DOF the unsharpness is unpredictable under normal conditions. (Wikipedia, 2013. Depth of Field.)

2.4.5 Image resolution

Image resolution indicates how clearly an image is visible. In other words, it is the minimum distance between two lines in an image such that the image is visibly resolved. Image resolution can be defined in different ways such as pixel resolution and spatial resolution.

Pixel resolution is indicated in the form M x N (M by N). M represents the number of pixel-columns (width) whereas N represents the number of pixel rows (height). It is also sometimes indicated by the total number of pixels in the image or the region of interest in the image. The more the pixel resolution, the better is the image visibility or quality. (Wikipedia, 2013. Image resolution.) Figure 13 shows the visibility with different image resolutions for the same image.

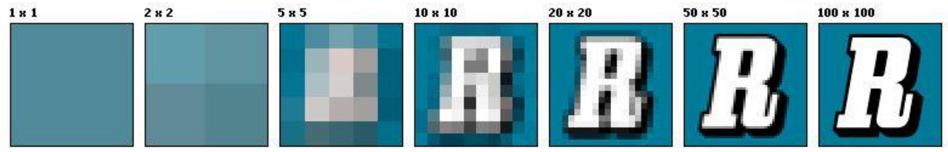


Figure 13 Showing visibility of an image with different pixel resolutions (Wikipedia. Image resolution).

Spatial resolution is the measure of how closely lines can be resolved in an image. It is determined by the pixel spacing, lens aberrations, diffraction and the depth of field. But in most machine vision applications, pixel spacing and depth of field are the deterministic features of spatial resolution. It suggests that images with high pixels are not necessarily sharp. (Wikipedia, 2013. Image resolution.)

2.5 Machine vision tools

Machine vision tools refer to the software algorithms used for image processing and analysis. These are the building blocks for developing machine vision applications. These tools are supplied by different manufacturers but all have the same working principles. All the machine vision tools are developed in a way that they analyse a particular set or number of pixels within a defined region of interest of the object, show the graphics of the object being analysed, and provide data information about the image to make a decision for the controller and the output devices. The vision tools can be categorized into different types as described in the following sub-sections.

2.5.1 Image processing and filtering tools

Before an object is analysed, the image filters (or image pre-processing filters) can be applied to the image of the object to sharpen the image pixels, increase the edge contrasts and remove noise from the image or even the reverse of these. Lighting variations and insufficiencies brings quality degradation in the image. Thus the filters enhance the image. Figure 14 shows images before and after filtering; the original image is blurred using an image filter.

Figure 14 Images before filtering (left) and after filtering (right) (Lode's Computer Graphics Tutorial)

2.5.2 Positioning (or Locating), counting and measurement tools

Machine vision tools are widely applied in industries for object positioning, object counting and measuring dimensions of the objects. Basically, all these processes are interconnected. So, all these can be achieved with the same adjustment or utilization of the vision tools. For example, object positioning requires finding distances between two objects or matching the pattern where the object needs to be located.

2.5.3 Application specific tools

Different application specific machine vision tasks such as bar code reading, defect detection, color recognition, etc. can be accessed using machine vision tools. Figure 15 alongside shows an example of application specific tools included in the NI vision development module of LabVIEW to read the barcodes.

Figure 15 Machine vision application reading barcodes (National Instruments. Barcode Example)

2.6 Device communications

When the image of the object being analysed is acquired, and processed, the system should have the ability to trigger output(s) to the output device(s) and (or) receive input signal from the input device(s). This involves the communications among the machine vision tools, image acquisition device, input devices such as proximity sensors, and output devices such as valves. Figure 16 shows the schematic representation of device communications among the different components of a machine vision system.

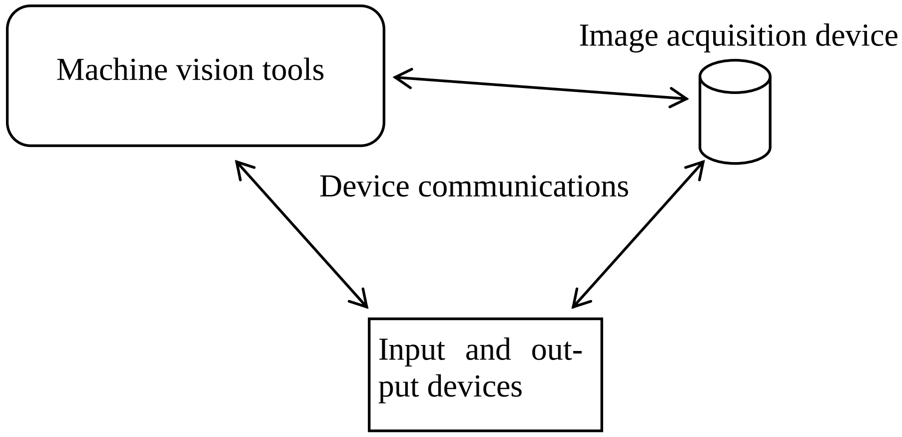


Figure 16 Schematic representations of Device communications in a machine vision system (Pokharel, 2013)

3 MACHINE VISION PROCESS

Generally, the processes involved in machine vision applications can be categorized into three areas as:

3.1 Image Acquisition
3.2 Image Processing
3.3 Output

3.1 Image acquisition

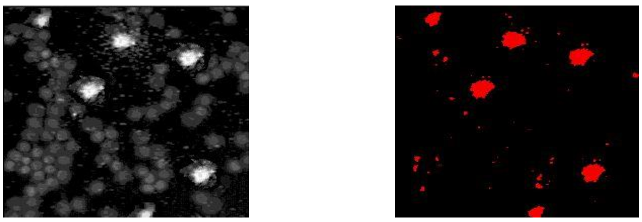
Image acquisition is the process of acquiring an image with an image acquiring device, a camera. While the image-acquiring device captures an image, the image would lose much information. This loss might be due to bad light exposure, bad angle adjustment, noise in the surroundings, and so on. So, in order to extract and regain the lost information, the images needs to be further processed using image processing and machine vision tools for a better output.

3.2 Image processing

The acquired image needs to be further processed by using image-processing tools such that the information lost during acquisition could be regained. Some of the generally used image-processing tools for machine vision applications are as follows:

i. Pixel counting: Pixel counting is one of the most common image processing methods. It involves counting of the light or dark pixels that an image is formed of. It can be analysed by histogram that shows the grayscale distribution in an image.

ii. Thresholding: It is the simplest process of dividing the image into segments. It is used for creating a binary image from a grayscale image so that the region of interest is separated from the background. It requires that the region of interest and the background have enough contrast. Figure 17 shows image before (a) and after (b) thresholding.



(a) (b)

Figure 17 Stained biological cells (a) after thresholding (b) (National Instruments)

iii. Image filtering: Different image filters can be applied for the image processing. Ready-made filters are available and also user-customized filters can be created. Linear filters, spatial filters, FFT (Fast-Fourier Transform filters), etc. are some of the filters used in image processing. The algorithms of the image processing filters are not included in this thesis. But, usually the software that is being used for machine vision has most of the common filters inbuilt. Figure 18 alongside shows the binary image after using a filter to reduce the noise present in Figure 17(b) above.

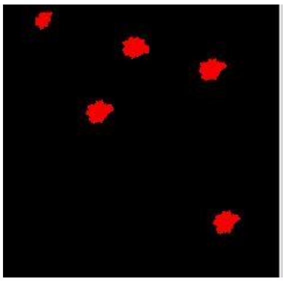


Figure 18 Filtered image(National Instruments)

3.3 Output

Output is of immense importance in any process. Something is done to get something in return. Machine vision applications also have a final output. Generally the outputs of machine vision applications are categorized as Pass/Fail. But additional attributes for Pass/Fail outputs can also be defined, for instance number of passed/failed items, setting an alarm if the items are failed, and so on. It depends on the objective of the application. But it is always significant to define the attributes for further R&D tasks. Figure 19 illustrates an easy-to-understand representation of output starting from image acquisition in a machine vision application.

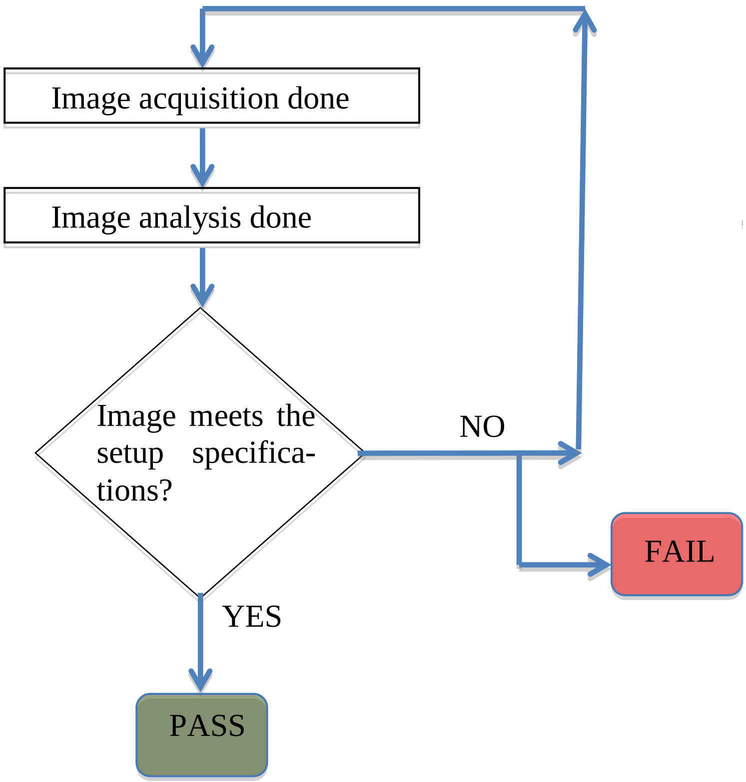


Figure 19 A simple schematic representation of machine vision process (Pokharel, 2013)

4 MACHINE VISION AND OBJECT SORTING

Object sorting is one of the most important processes in industrial production lines. During the production process, many damaged or undesirable products are also produced. These kinds of products are unfit for use and deploying into the market and thereby need to be sorted out.

To sort those unfit products manually is time consuming, expensive, inaccurate and tiresome. The application of machine vision enables this task to be implemented with higher degree of accuracy in comparatively less time, negligible labor, and low cost. However, technical problems might arise sometimes.

Today, several machine vision software and cameras are available in the market. The software available is so effective that ready-made applications for automated inspection already exist. The user needs to give only the inputs to the software. But the users can develop their own application. One significance of developing own application is the reduction in the expenditure because, the ready-made application demands extra budget. A machine vision application is similar in both ways- theoretically as well as practically. Furthermore, the more a user is experienced, the more significant application can be developed. As per demands, different methods may be applied to develop machine vision applications. But, only few methods for object sorting have been dealt in this document.

4.1 Machine vision methods for object sorting

Object sorting has become a common procedure in industries. The application of machine vision in object sorting can be significant as it increases accuracy and reduces redundancies in the products. Different vision inspection methods can be applied for object sorting depending upon the products and standards to be met. Different algorithms made ready for executing inspection through machine vision by different hard-working programmers and software companies are available. The customers are free to choose from the ready-made algorithms or create their own customized algorithms using the tools provided by the software. Some of the methods that can be applied for object sorting are described in the following subsections.

4.1.1 Pattern matching

Patten matching is the process where a template image is defined and stored in the memory. This template image is also referred to as reference image. When the real-world images; for example, image of products in a factory production line are acquired using camera, the software searches for instances of the template images stored in the memory and confirms if the product matches the pattern defined in the template image or not. A rectangular box around the image often indicates pattern matching. Pattern matching is a significant machine vision tool as it is independent of lighting variation, blur, and noise.

Pattern matching technique can be used for gauging (measuring lengths, diameters, angles, etc.), product inspection (detecting flaws such as missing parts or components from a product), and product alignment (determining the position and (or) orientation of a known object by locating fiducials. (NI Vision for LabVIEW User Manual.) Hence, pattern matching can be applied for quality control of the products.

In this thesis project also, pattern-matching tool was applied for sorting of objects. The program is described in Section 8.

4.1.2 Edge detection

Edges are of importance while determining the structure of the object. Typically edges occur between the boundary separating two regions in an object. So edges are the significant changes to analyse the image. These local changes are seen in intensity of different regions in an image. The change or discontinuity in the intensity might give rise to the formation of step edges, line edges, ramp edges or roof edges. (Jain et al. 1995, 140-141.) Figure 20 shows the original image (left) and the edge-detected image (right).

Figure 20 Images before and after edge formation (National Instruments)

Step edges occur when the intensity abruptly changes from one value on one side of discontinuity to the opposite value on the other side. Figure 21 shows the step edge formation behaviour of the intensity in an image.

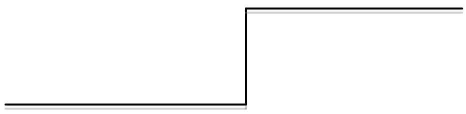


Figure 21 Step edge formation behavior of intensity in an image (Pokharel, 2013)

Line edges occur when the intensity abruptly changes its value but regains its original value within a short distance. The line edge formation behaviour is shown in Figure 22 below.

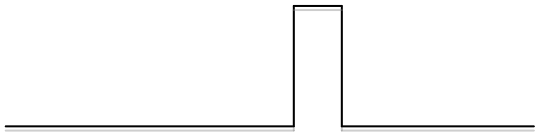


Figure 22 Line edges formation behavior of intensity in an image (Pokharel, 2013)

However, step and line edges are rare in the real world images. The behaviours of step edges changing to ramp edges, and line edges changing to roof edges are respectively shown in Figure 23(a), and Figure 23(b).

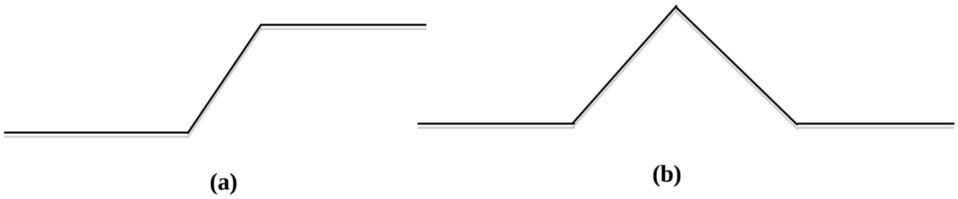


Figure 23 Ramp edges formation behavior (a), and Roof edges formation behavior (b) (Pokharel, 2013)

The edge detection algorithms are typically used in three different areas of machine vision applications namely gauging, alignment, and geometric transformations of objects (NI Vision for LabVIEW User Manual).

4.1.3 Geometric matching

”Geometric matching locates regions in a grayscale image that match a model, or template, of a reference pattern” (NI Vision for LabVIEW User Manual. 71).

Geometric matching and pattern matching tools resemble closely. It is also independent of lighting variations, blur, noise, and geometric transformations. And, the steps involved are also quite similar.

Like pattern matching, a template image is created that works as a reference image for which the geometric matching tool searched from the acquired images. Figure 24 below shows geometric matching and the template image in LabVIEW.

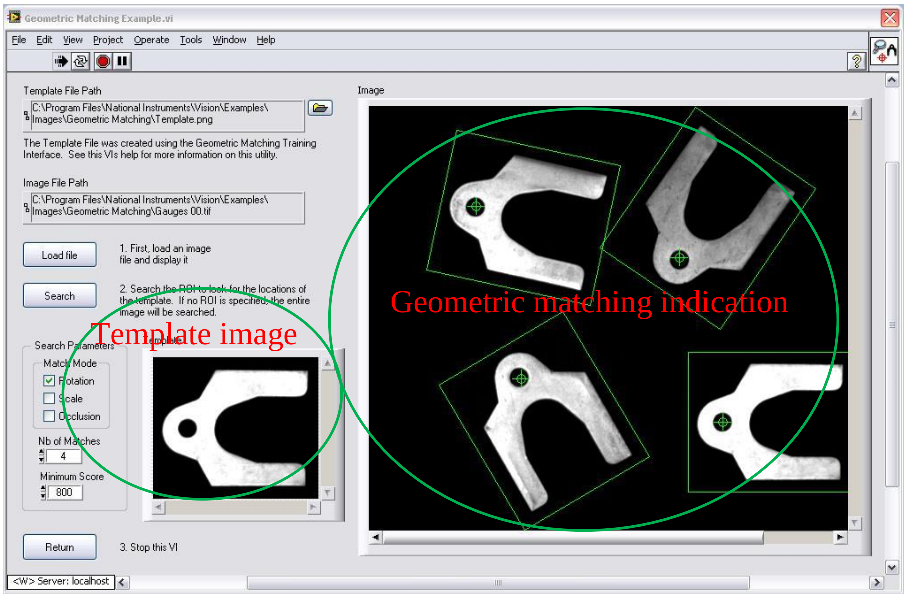


Figure 24 Geometric matching illustration with the reference of a template image (National Instruments. Geometric matching example)

5 SORTING STATION

The sorting station is shown in Figure 25.

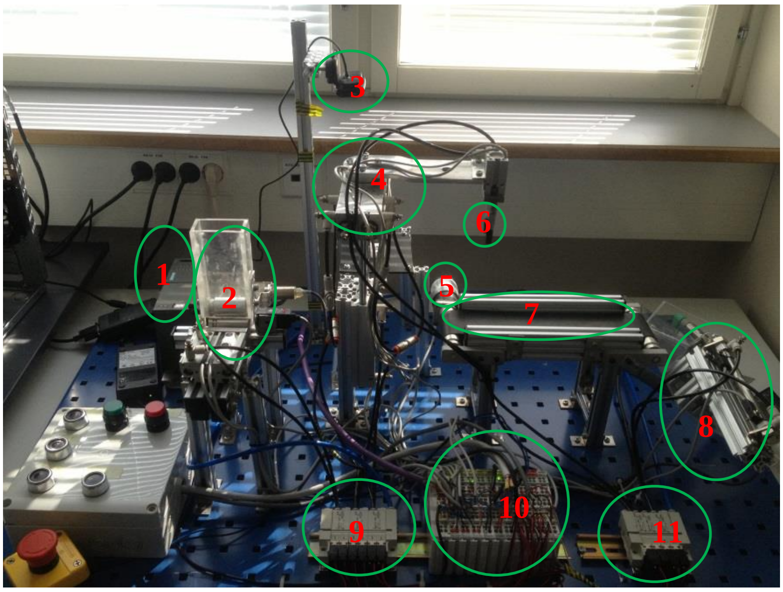


Figure 25 showing Sorting station showing major components. The station was constructed for the object sorting purpose. It was constructed in such a manner that it accepts the products that pass the Machine vision test while the products that could not pass the test are rejected by the system. (Pokharel, 2013)

1-PLC, 2-Object Feeder, 3-Camera, 4-Rotating table (Manipulator), 5-Motor, 6-Gripper, 7-Conveyor, 8-Sorter, 9&11-Valves, 10-Distributed module

A simple logical sequence program was coded using STEP 7 software for the automation of the system. The program was downloaded to the PLC so that the station can run as soon as the electrical supply is made. For the inspection of the product, a VI (virtual instrument/ LabVIEW program) was made. The image is acquired into LabVIEW using a webcam. The software analyses the image and triggers the corresponding valve for passed product. In this system, only one trigger is sent to the valve from LabVIEW. The valve reset was done inside STEP 7 codes.

The object is pushed out from the feeder by a piston. The rotating table then picks the object and places it under the camera. The camera acquires image into LabVIEW and thus the VI created analyses the image. If the image matches the pattern defined in the program, it triggers a signal to the valve so that the sorter (cylinder piston) moves out; otherwise no action is seen in the sorter. Then the rotating table picks that object and places on the conveyor. The object then is conveyed to the corresponding sides. Thus the sequence continues as long as the power is ON.

5.1 Construction of the station

The system consists of cylinders, sensors, a motor, and belt and conveyor as the physical components. As the PLC used in the station requires additional signal modules for the I/O (inputs and outputs) devices, signal modules from Beckhoff (www.beckhoff.com) was used. The data-sheet of the signal modules can be found in Appendices 7 and 8. These modules were networked to the PLC using Bus-coupler LC3100 (also from Beckhoff) via PROFIBUS cable.

Table 2 provides basic information about the devices that have been used in the sorting station.

Table 2 Basic information on the devices used in the station

S.N	Device Name	Device Type	Manufacturer
1	Simatic-300, CPU315-2DP	PLC	Siemens
2	LC3100	Bus Coupler	Beckhoff
3	KL1114	Input signal module	Beckhoff
4	KL2114	Output signal module	Beckhoff
5	3-position rotating table	Pneumatic actuator	SCM
6	Valves	Pneumatic	
7	S7 MPI	Interface drive	Siemens
8	Webcam	Image acquisition	Logitech
9	Power supply	AC/DC adapter	
10	Motor	Motor	
11	Actuator	Pneumatic	SCM
12	Sensors	various	

5.2 Vision section of the station

Figure 26 shows the diagrammatic section of the station that is responsible for providing information to the sorter if the image of the object meets the required specifications or not.

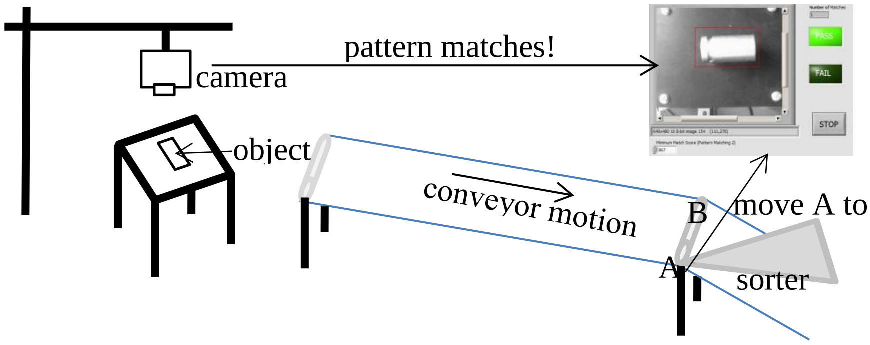


Figure 26 Vision section of the system determining the move direction of the sorter (Pokharel, 2013)

If the machine vision application matches the object the sorter moves from position A to B (marked in Figure 26 above). By default, the sorter is always at position A; thus the sorter moves only if the object matches. The sorter also has to return back to position A, when it has moved to position B; this was done in the automation codes. The codes for resetting the sorter can be found in Appendix 6.

In Figure 26 above, it has been indicated that the pattern matches. Therefore, in this condition, the sorter should move from position A to B as indicated by the arrow (move A to B).

5.3 System communication

The PLC was interfaced to the computer using S7 MPI adapter. The bus coupler, LC3100 was networked with the PLC using PROFIBUS cable. Also OPC communication was done using the same S7 MPI adapter. A diagrammatic representation of communication among the system components is shown in Figure 27.

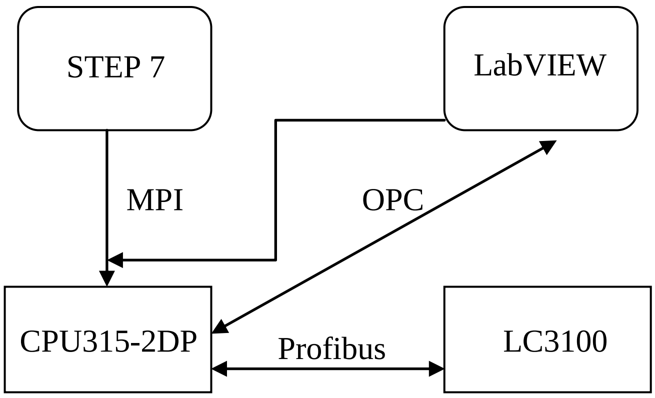


Figure 27 Communications among STEP 7, LabVIEW and CPU315-2DP and LC3100 (Pokharel, 2013)

6 AUTOMATION OF THE STATION

6.1 STEP 7 professional

STEP 7 PROFESSIONAL (STEP 7 in short) is automation software from Siemens Industry (www.siemens.com). It is used for programming simatic PLC stations. Figure 28 shows the system manger window of simatic STEP 7, where a created project with some blocks is also shown.

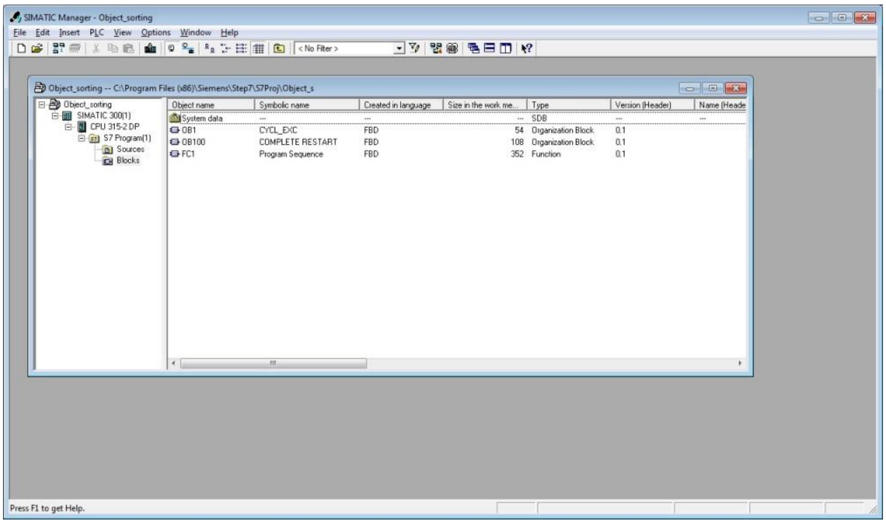


Figure 28 A screenshot of simatic manager (STEP 7) (Pokharel, 2013)

STEP7 provides provision to code the program using seven different languages. Some of them are FBD (Function Block Diagram), LD (Ladder Diagram), and SCL (Statement List), etc. The user can freely choose the language. An STEP7 automation program may contain functions (FCs), function blocks (FBs), organization blocks (OBs), sequence functional charts (SFCs), etc. But, every STEP7 program must have OB1, because it is the main function. Details about the STEP7 programs are not covered in this thesis.

The language used for the automation of the system described in this thesis was FBD. A screenshot of the codes is shown in Figure 29 alongside. But, the complete program codes can be found as Appendix 1 to Appendix 6.

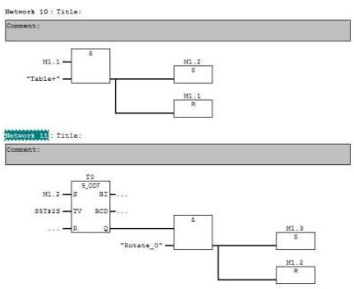


Figure 29 FBD codes in STEP 7 (Pokharel, 2013)

‘

6.2 CPU315-2DP

CPU315-2DP belongs to Simatic300 group of PLC controllers. It consists of different indicators, a program manipulation key, and a memory card slot. It has 2 serial-connection ports; one of them is for interfacing between STEP7 and the PLC whereas the other is for connecting to a distributed module. Figure 30 alongside shows CPU315-2DP.

Figure 30 CPU315-2DP (Pokharel, 2013)

6.3 Hardware configuration

Hardware configuration (HW) needs to be done before any program can be downloaded into the PLC. In the hardware configuration, the type of power supply, the CPU model (such as CPU315-2DP), the signal modules and (or) other distributed modules being used need to be specified so that the software and the hardware can inter-connect. Figure 31 shows the hardware configuration done for this project.

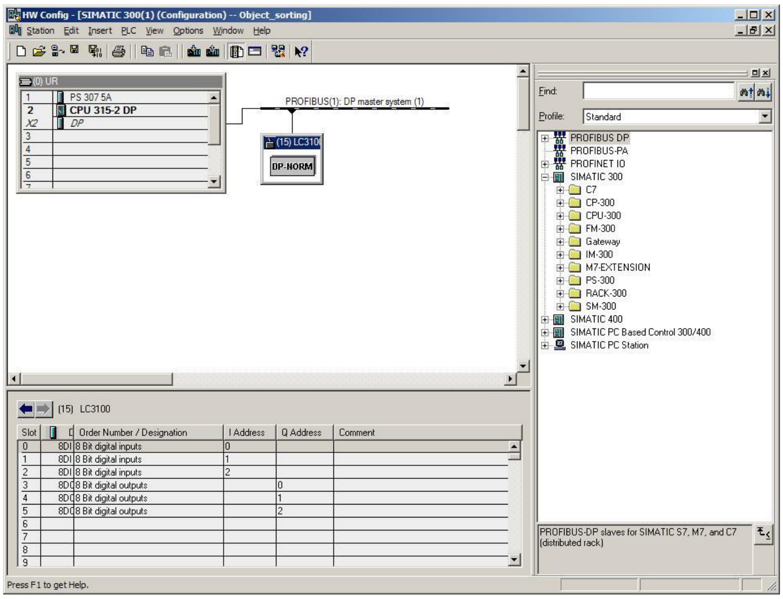


Figure 31 An example Hardware configuration for simatic PLC (Pokharel, 2013)

7 NI LABVIEW

NI LabVIEW is a graphical program (G-programming) development environment from National Instruments (www.ni.com). Programs written in LabVIEW are called VIs (virtual instruments). It is a powerful tool used by engineers and scientists for different kind of measurements, process controls and R&D (Research and Development). It can easily be integrated with most of the hardware, provided the driver of the hardware is installed into the computer.

LabVIEW can also be integrated with other software tools such as Matlab and Simulink. It can be connected to different PLCs (Programmable Logical Controllers) via different industrial communication protocols, for example through OPC, Ethernet, Profibus, ProfiNET protocols. It can be implemented into process controls ranging from small scale to large scales.

One advantage of programming in LabVIEW is that we don't have the overhead to write huge codes. Just the function of the different blocks needs to be known. Also, another advantage is that it works on 'Dataflow Programming' principle, i.e. the output is only obtained when all the inputs get their input data.

7.1 LabVIEW basics

The installation procedure can be found over the internet or once the installation requirements (installation CD/DVD, operating systems, etc.) are available, LabVIEW can easily be installed following the onscreen setup wizard.

As soon as LabVIEW is installed into the computer, it can be accessed in the same way as other software is accessed. If the operating system is Windows, it is common that it can be accessed from the Start menu or through a shortcut on the Desktop. Figure 32 shows the start-up window screenshot of LabVIEW 2012.

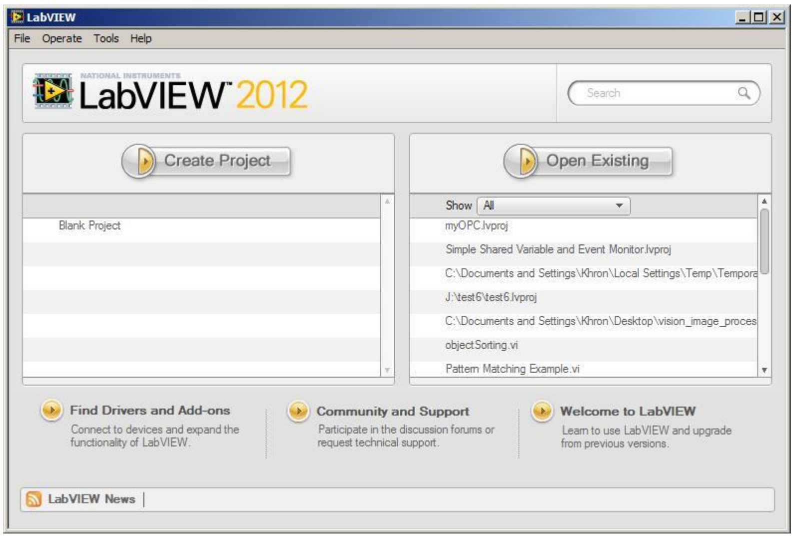


Figure 32 LabVIEW 2012 start-up window (Pokharel, 2013)

LabVIEW consists of two windows called the Front Panel and the Block Diagram. The two windows are explained with example in the following sub-sections.

7.1.1 LabVIEW front panel

The LabVIEW front panel is the window where, different controls (such as switch, knobs, numeric inputs, etc.) and indicators (such as LEDs, graphs, numeric outputs, etc.) can be viewed; it can also be called as Lab-VIEW HMI (Human Machine Interface). Figure 33 shows the Front panel with some controls and indicators.

Also, the controls menu can be seen, which are categorized into different groups. The menu can be accessed by right-clicking inside an empty region of the Front panel. The tools palette is also shown, where the different types of mouse manipulators are shown. This palette can be set to be in automatic mode or manual mode.

A waveform chart, a stop button, two LEDs and two numeric inputs are shown in the panel. Also, one of the LEDs is lit 'ON' on the condition of Number 1 is being less that Number 2. The conditions are made using wirings in the block diagram that will be explained in the LabVIEW Block Diagram immediately in the next sub-section.

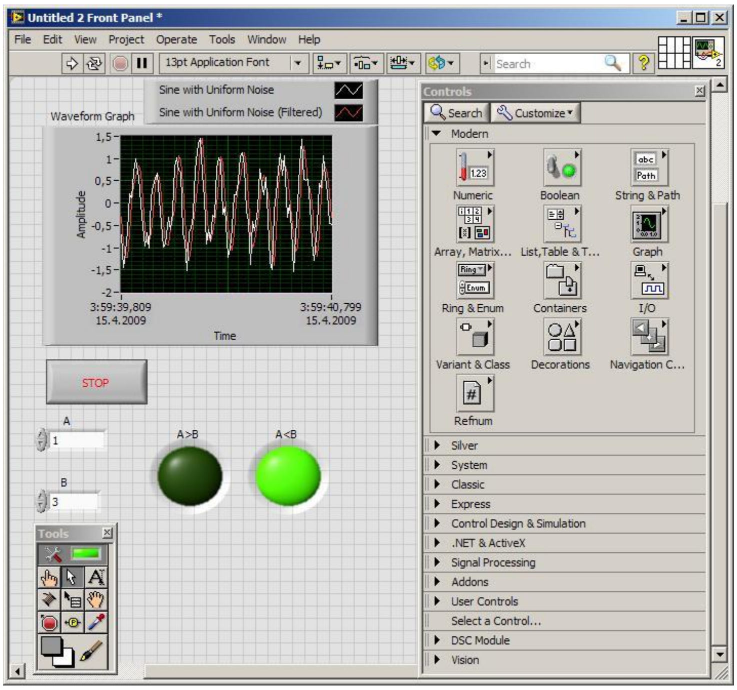


Figure 33 NI LabVIEW Front Panel (Pokharel, 2013)

7.1.2 LabVIEW block diagram

LabVIEW block diagram serves as the brain of the program. All inputs and outputs are wired in the block diagram. The block diagram basically consists of different functions such as mathematical functions, Boolean functions, programming loops, etc. Figure 34 shows the LabVIEW block diagram window, with the necessary wirings for the Front panel discussed in sub-section 7.1.1.

An express VI, named Simulate signal was used to simulate a sine signal. Also, a uniform noise was defined in it. A filtering express VI was used to filter the noisy sine signal. In the Front panel window shown in Figure 33 of sub-section 7.1.1 above, the noisy and filtered sine signals can be viewed. The noisy signal is the white signal whereas the filtered sine signal is the red one.

Also, two numeric controls are connected to logical operators ‘>’ and ‘<’. Each logical operation has its output shown with a LED.

The whole block diagram is placed inside a ‘WHILE’ loop. The loop has a time lag of 100 milliseconds and a stop control to stop the loop operation. The controls are done in the Front panel.

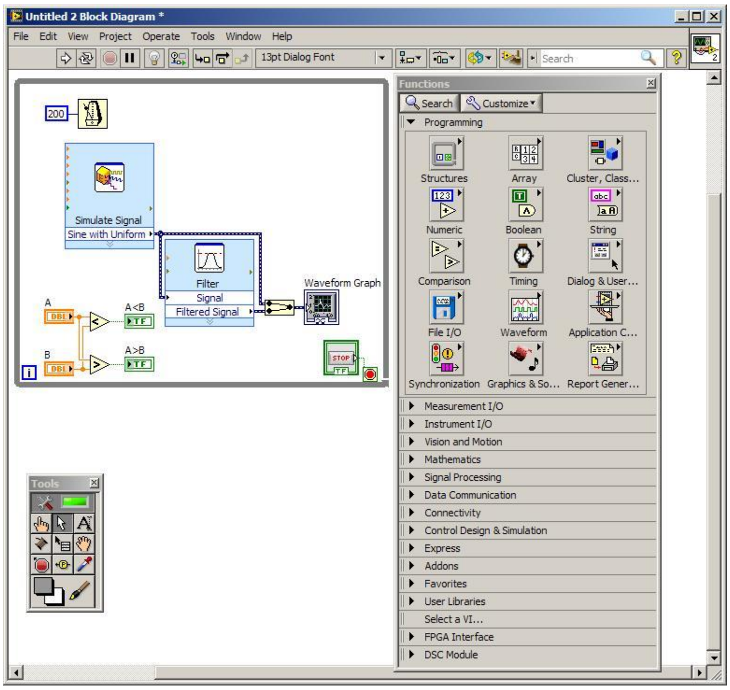


Figure 34 NI LabVIEW Block Diagram (Pokharel, 2013)

7.2 NI vision

NI vision is a library included in the NI Vision Development Module for LabVIEW. NI Vision is meant for the development of machine vision and scientific imaging applications.

The steps that are involved in creating a machine vision application in LabVIEW are discussed in the following sub-sections.

7.2.1 Setting up the imaging system

Setting up the imaging system is the most important part for developing a vision application. Before an image is acquired, the imaging environment should be favorable for the image analysis method going to be used. The imaging environment should produce image with quality high enough to extract the information needed.

The important aspects to be fulfilled are the type of camera being used, the lens of the camera, its resolution, and the surrounding lightings. Lighting is a vital aspect for image acquisition as poor or vivid light accounts for poor image and thus a lot of information from the image is lost. For more about types of lighting, section 2.1 of this document should be referred.

The camera should always be positioned in a way that it is perpendicular (90° angle) to the object(s) being analyzed as shown in Figure 35. It is alright if some errors in perpendicularity occur as the software is capable to compensating such errors but it is recommended that the camera be placed perpendicular to the object as precisely as possible. A clamp and stand could be used to install the camera perpendicular to the object.

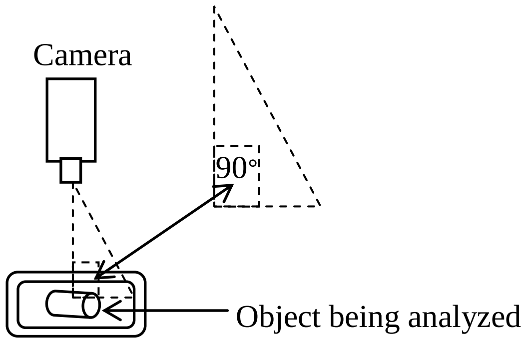


Figure 35 showing camera installation for machine vision purpose (Pokharel, 2013)

The selection of the camera also accounts for a good vision application. NI Vision supports cameras ranging from simple web cameras to complex GigE Ethernet cameras.

7.2.2 Calibrating the imaging system

After the imaging system is set up properly, the next step would be calibrating the imaging system. Calibration of the imaging system is critical because, all the machine vision tasks will be based on the calibration made. The better the calibration, the better would be the image analysis and machine vision tasks.

Calibration involves assigning the real-world coordinate system to the pixel-based coordinate system. It also assists in compensating perspective and non-linear errors that might be present in the imaging system. Perspective errors arise due to the camera not being perpendicular to the object under inspection while non-linear errors arise due to lens aberration of the camera. Simply to understand, lens aberration can be defined as the fault in the lens that prevents the lens from converging different rays of lights to a single focus point.

7.2.3 Creating image

NI Vision has a block called IMAQ Create (Vision Utilities>>Image Management) to create a reference image. While using IMAQ Create to create a reference image, the data type of the image should be specified. The image types are not described in this document but the different image types are shown in Table 3.

Table 3 Image and image data types (National Instruments: Machine Vision User Manual)

S.N	Image type	Image data type
1	Grayscale (U8)	8-bit unsigned
2	Grayscale (I16)	16-bit signed
3	Grayscale (SGL)	complex
4	RGB (U32)	32-bit RGB
5	HSL (U32)	32-bit HSL
6	RGB (U64)	64-bit RGB

The purpose of creating image is to allow NI Vision to create an internal image structure to hold the different image properties such as the name of the image, and the border size of the image. However, memory is not yet allocated for image pixels by creating the image. The image thus created is passed over to every other subsequent NI Vision functions as input.

NI vision has different IMAQ blocks as shown in Figure 36.

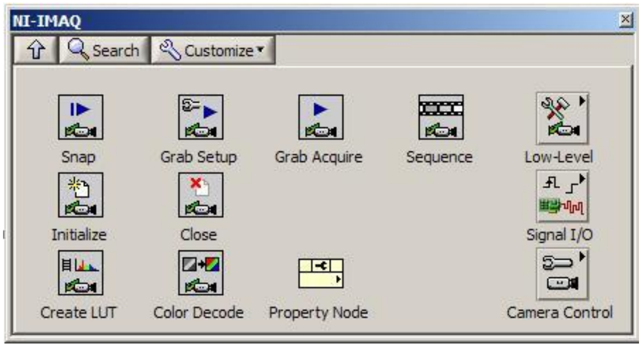


Figure 36 NI IMAQ blocks (Pokharel, 2013)

Note: At the end of every vision application, the created image must be disposed using IMAQ Dispose (Vision Utilities>>Image Management).

7.2.4 Image analysis

The created image has to be further analyzed. As described before, different image analysis techniques can be applied for this purpose. Figure 37 below shows the different image analysis functions available in NI vision module.

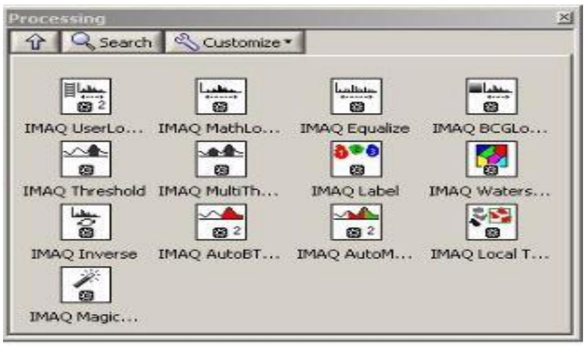


Figure 37 Different image analysis VIs in NI Vision module (Pokharel, 2013)

7.2.5 Acquiring or reading an image

Finally an image is acquired. The acquiring or reading of the image is possible from an image acquisition device such as the camera or from an image stored in a directory in the computer or converting a 2D array into an image.

Typically separate image acquisition devices are available for industries. Such devices have several inbuilt capabilities for industrial machine vision applications. But, for this thesis project, a simple webcam was used as the image acquisition device. It is shown in Figure 38.

Figure 38 Logitech HD Webcam C615 (Logitech)

Figure 39 below shows the express image acquisition functions available in NI vision module.

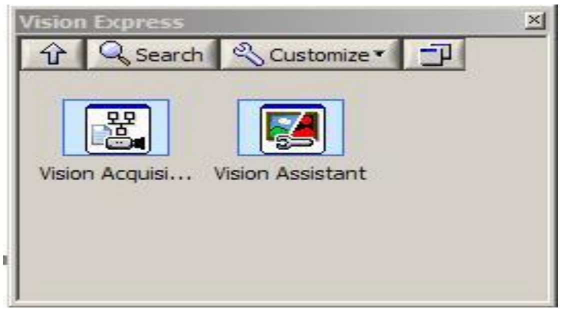


Figure 39 Express functions for vision applications in NI Vision Development module (Pokharel, 2013)

7.2.6 Displaying the image

The image should be displayed on the screen after the image acquisition is done. NI Vision provides a facility to display image internally inside the Front panel or externally in a different window. Figure 40 shows the different types of image display and indicator blocks available in NI vision module.

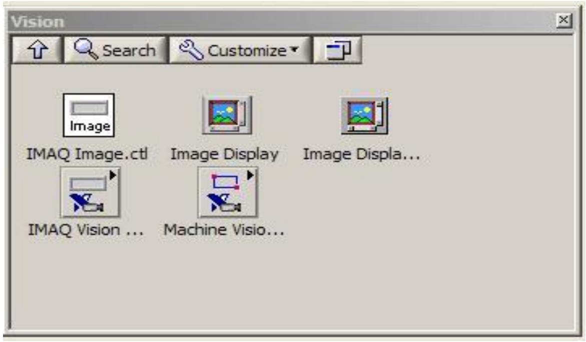


Figure 40 Vision Development module blocks for Front panel (Pokharel, 2013)

7.3 NI LabVIEW OPC servers

OPC stands for OLE (Object Linking and Embedding) for Process Control. It is a ‘standard defined’ interface to link devices in industries; it is used for connecting with different databases, laboratory equipment and test systems. Most of the industrial data acquisition devices and control devices such as PLCs are compatible with the OPC standard. The idea for the interconnection among device(s) and (or) database(s) is to share the data, and allow making changes in data such that a change in one device(s) or database(s) will make the same change in the other device(s) or database(s).

NI OPC servers can be launched as soon as it has been installed into the computer. To interconnect the simatic PLC and LabVIEW 2012, ‘NI OPC servers’ was used to create a simatic-300 based CPU315-2DP as an OPC server. Different tags corresponding to the signals for the I/O devices used in the CPU was created, and thus created tags could be used in VI to view and control the system statuses and outputs.

The next sub-sections on OPC servers are discussed in instructional ways. The instructions are based on the thesis project.

7.3.1 Configuration of an OPC server

Configuration of an OPC Server in NI OPC Servers is a simple process. It involves creating a new channel for a new device and adding the tags that represent the inputs and outputs for the I/O devices. The steps for creating an OPC server for a CPU315-2DP are described as follows:

i. In the configuration window (NI OPC Servers-Runtime), right-click on the left section of the window and select New Channel as shown in Figure 41. This follows a wizard similar to any other Windows-based wizards. The wizard is meant for selecting different data such as device type, device driver, connection time, etc. for OPC server configuration.

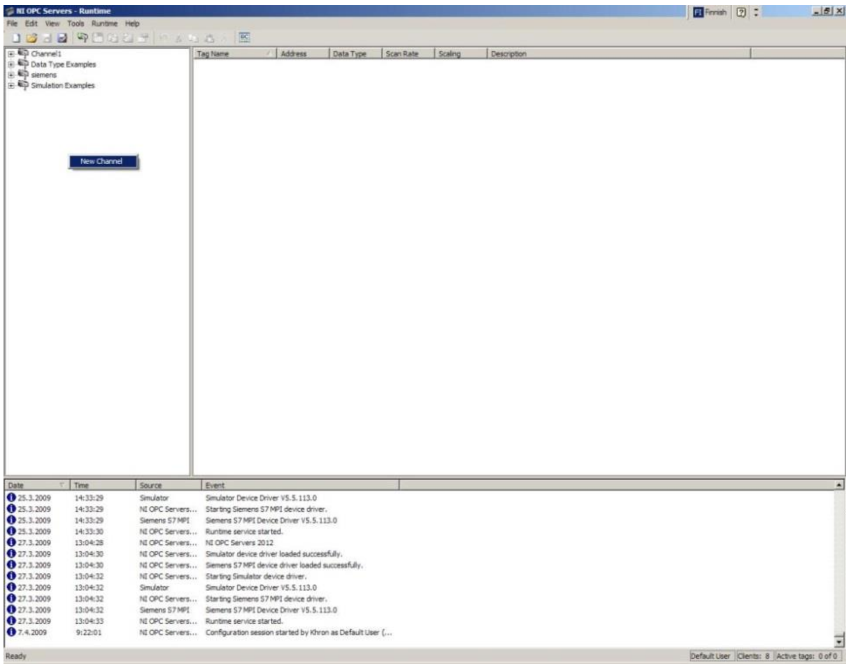


Figure 41 NI OPC Servers-Runtime window showing New Channel Creation (Pokharel, 2013)

ii. Provide a suitable channel name, for example Siemens, and click Next button.

iii. In the next window of the wizard, the device driver being used needs to be defined. For example, in this thesis project Siemens S7 MPI driver was selected; click on Next> button.

iv. Now the necessary communication parameters are defined in the next window of the wizard. The communication parameters defined for this thesis project is shown in Figure 42 below. Note that the parameters should be correct for the communication. For example, the baud rate for the PLC used was 19200.

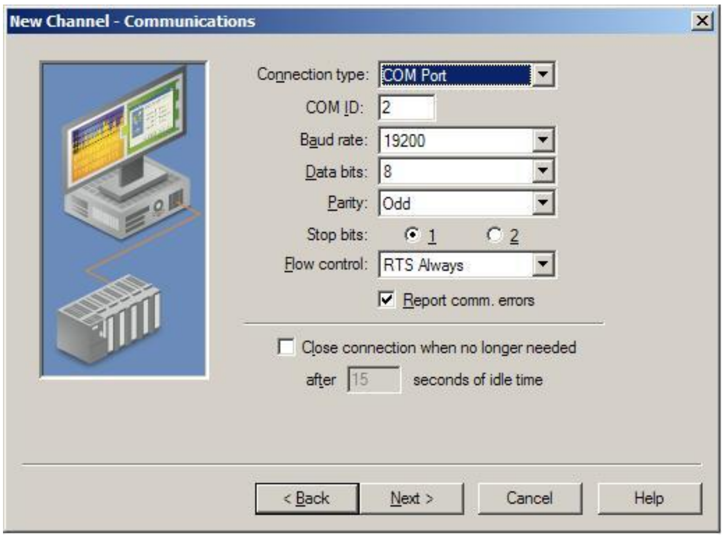


Figure 42 Communication Parameters setting for OPC Server (Pokharel, 2013)

v. Continue with the wizard providing the parameters necessary until the final window of the wizard appears where a summary of the configuration can be seen as shown in Figure 43. To change any configuration parameter <Back button can be clicked.

Figure 43 Summary of the parameter configuration for OPC server (Pokharel, 2013)

Hence a new channel for OPC communication was created. The newly created and configured channel can be visible on the left section of the runtime window shown in Figure 41 above (p. 32). Next, a device needs to be created and configured. The steps involved to create a new device are as follows:

i. Click on the 'click to add a new devices' that is visible under the created channel as shown in Figure 41 above (p. 32). If it is not visible, click the plus sign (+) on the left side of the channel name. After this, the new device wizard appears.

ii. Provide a suitable device name, for example CPU315-2DP, and click Next> button.

iii. Follow the wizard until the summary window of the wizard appears as shown in Figure 44. Finally click Finish button.

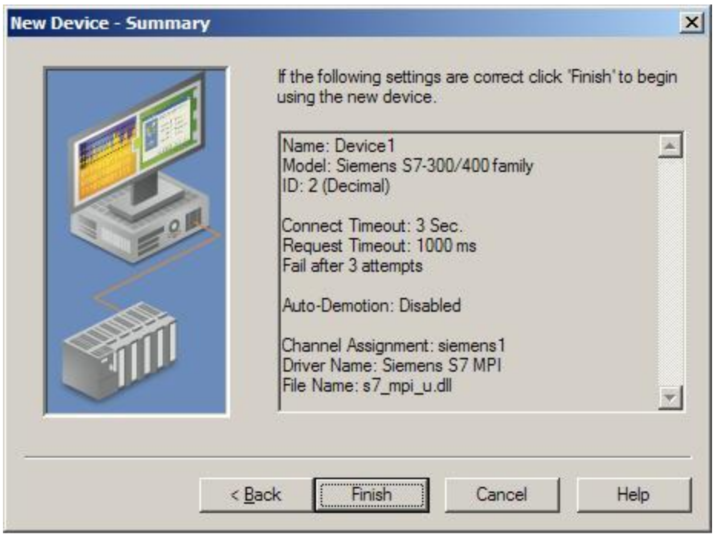


Figure 44 Summary of the New OPC server Device configuration (Pokharel, 2013)

Thus the device has also been created. Next, the tags are to be created. To create tags, click on the 'Click to add static tags…' that is visible on the right section of the runtime window shown in Figure 41 above (p. 32); it is visible only after the device has been created. A tag-creating window appears where the tag can be configured. An example tag configuration is shown in Figure 45.

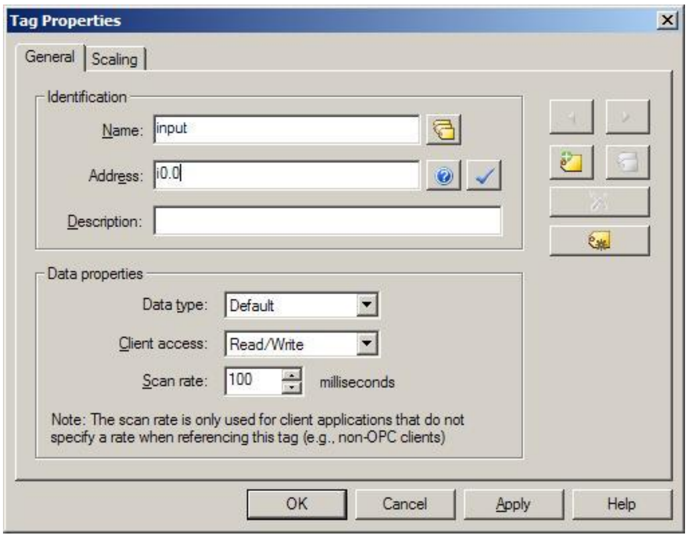


Figure 45 Configuring tag for OPC server (Pokharel, 2013)

7.3.2 OPC quick client

After the OPC server was configured with all the necessary tags, the states of different I/O devices can be viewed in the OPC quick client. Clicking the 'QC' tool on the toolbar of the runtime window can launch the OPC quick client. To view the states of different devices, the created device needs to be selected in the quick client window. Note that the device name appears in the format ‘channelName.deviceName’. An example OPC quick client window is shown in Figure 46.

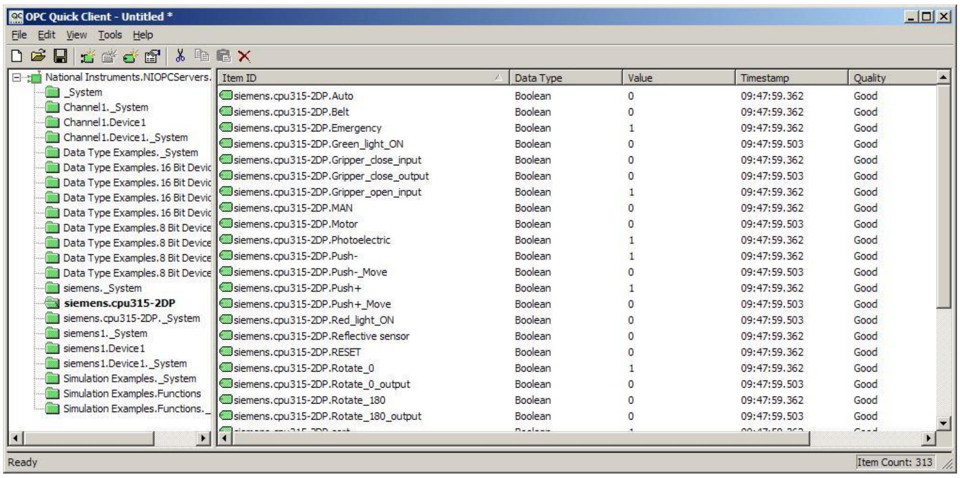


Figure 46 OPC Quick Client showing data type, value, timestamps, and quality of different Items or tags (Pokharel, 2013)

7.3.3 NI DSC module

NI DSC (Datalogging and Supervisory Control) module extends the Lab-VIEW graphical development environment by providing additional functionality to connect to distributed measurement, control and high-channel-count monitoring applications (National Instruments, 2012. NI LabVIEW Datalogging and Supervisory Control (DSC) Module).

The DSC module enables connection through shared variables and provides a method for interacting with touch panel devices. The details relating to it is out of the scope in this thesis.

7.3.4 OPC implementation in LabVIEW

LabVIEW can be interconnected with other devices or software to share data as well as read and write data using OPC protocol. LabVIEW can be connected to both OPC servers and OPC clients to share data.

The primary component that allows LabVIEW to implement OPC is the SVE (Shared Variable Engine) that is installed at the time of LabVIEW installation. The SVE manages the shared variable using a proprietary technology called the NI Publish-Subscribe Protocol (NI-PSP). The SVE runs as a separate process on the computer as soon as the shared variables are deployed to it. No LabVIEW or VI needs to be running for the shared items to be available. Even after the computer restarts the SVE will be running as long as it is not stopped. Hence, the shared variables can be shared with any other OPC servers or OPC clients. Also, the SVE can be a server or a client. (National Instruments, 2012. Introduction to OPC.)

SVE acts as a mediator among NI-PSP and other application(s). The I/O servers are configured as OPC clients by the help of NI DSC Module.

Similarly the SVE can be configured as OPC server to publish the NI-PSP data over the network for other OPC clients to interact.

Figure 47 shows a schematic diagram for OPC implementation in LabVIEW.

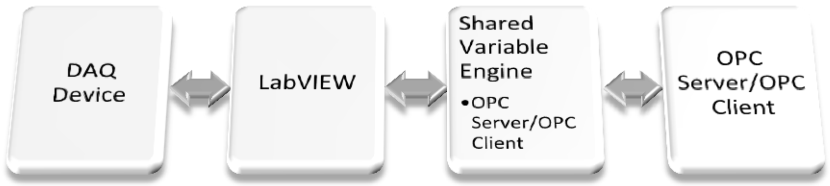


Figure 47 Shared Variable Engine can be either an OPC server or client (Pokharel, 2013)

7.3.5 LabVIEW as OPC client

LabVIEW can operate as both OPC server and OPC client. But, the necessity of this thesis project targets it to act as OPC Client. This section explains only the method of creating LabVIEW OPC client; the steps are as follows:

i. When the LabVIEW is launched, select ‘Blank Project’. Note that the method might vary depending upon the version of LabVIEW. The steps mentioned here are based on LabVIEW 2012. But, whatever the version is, this can be done from the File menu (File>>New Project).
ii. The New Project window appears. Right-click My Computer and from the context menu choose New>>I/O Server as shown in Figure 48 below.

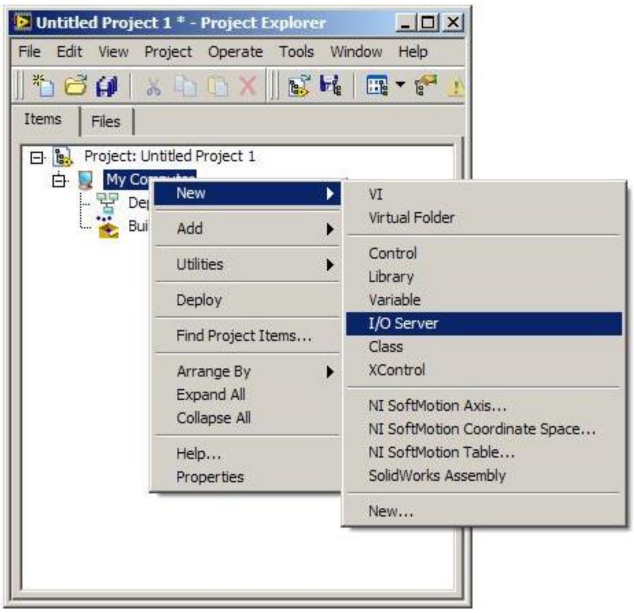


Figure 48 Creating a New I/O Server (Pokharel, 2013)

iii. Then the Create New I/O server window displays; select OPC client as shown in Figure 49, and click 'Continue'.

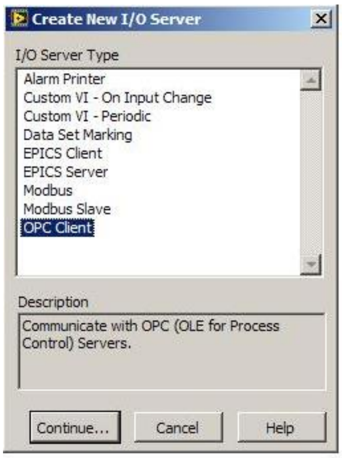


Figure 49 Create New I/O Server window (Pokharel, 2013)

iv. Next step is configuring the OPC client. Select National Instruments.NIOPCServers.V5 as shown in Figure 50 below. Change the Update rate (ms) to 100 (default value is 1000), and click OK button. A prompt window titled DCOM configuration Recommendation appears; accept the prompt clicking OK button.

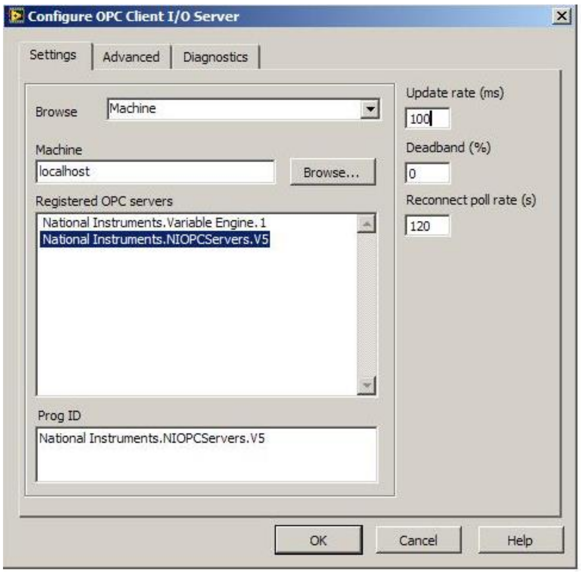


Figure 50 Configuring OPC Client I/O Server (Pokharel, 2013)

v. Then a library is created (Untitled Library 1). The name of the library can be altered as desired. On expanding the library name clicking the + sign a label OPC Client (the label can be altered) can be seen. Right-click OPC Client and select Create Bound Variables as shown in Figure 51.

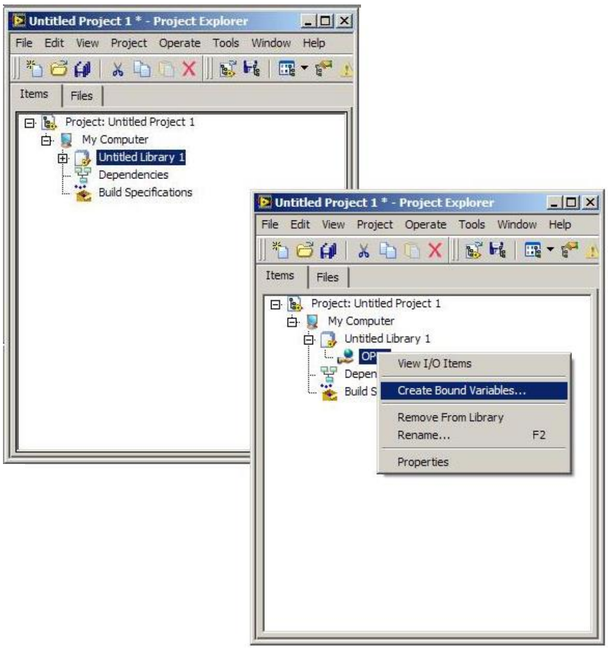


Figure 51 Creating Bound Variables (Pokharel, 2013)

vi. Then the Create Bound Variables Window appears as shown in Figure 52 below. From the left of the window, expand the device that was created using NI OPC Server, and select the variables (tags) to be implemented in LabVIEW client. Multiple tags can be selected, and added by clicking Add>> button, and then click OK.

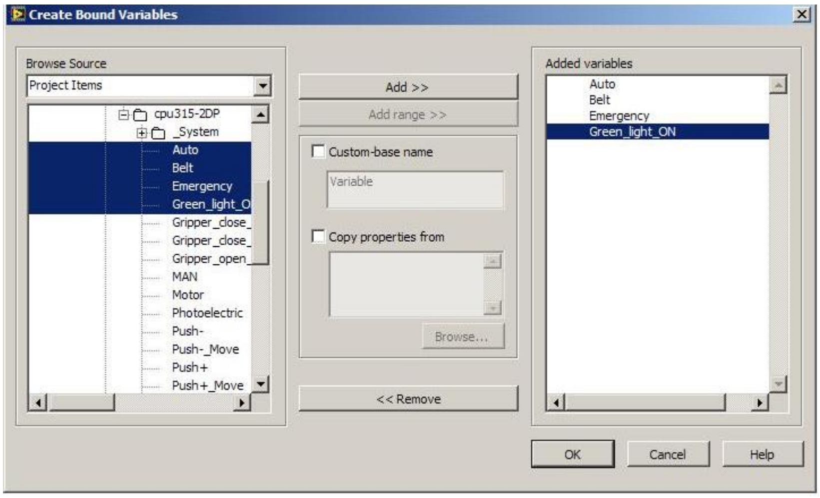


Figure 52 Create Bound Variables Window (Pokharel, 2013)

vii. Then Multiple Variable Editor window appears as shown in Figure 53; click Done button.

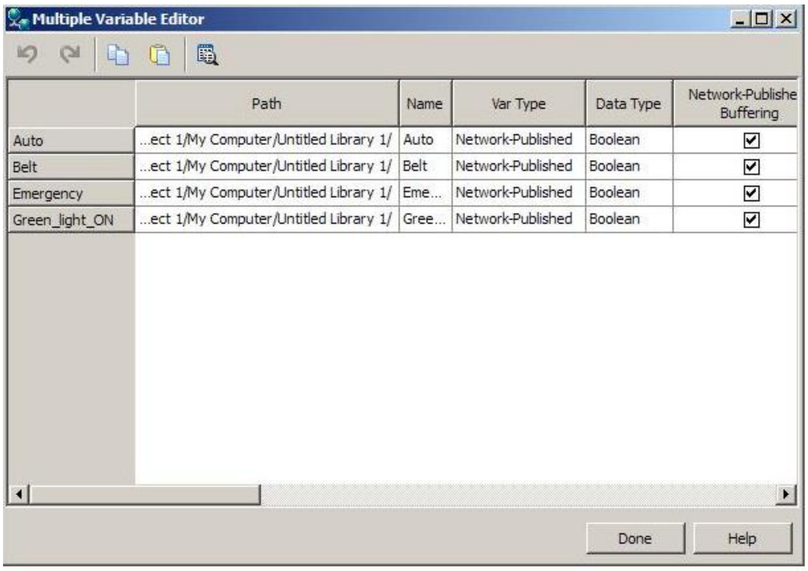


Figure 53 Multiple Variable Editor Window (Pokharel, 2013)

viii. The bound variables are then visible in the project window as shown in Figure 54 below. Right-click on the created Library (here Untitled Library 1) or My Computer and select ‘Deploy’ such that these shared variables can be available to all the networked components. The LabVIEW VI can now access all the variables of PLC using the created tags in OPC Server; read from PLC as well as write data to PLC.

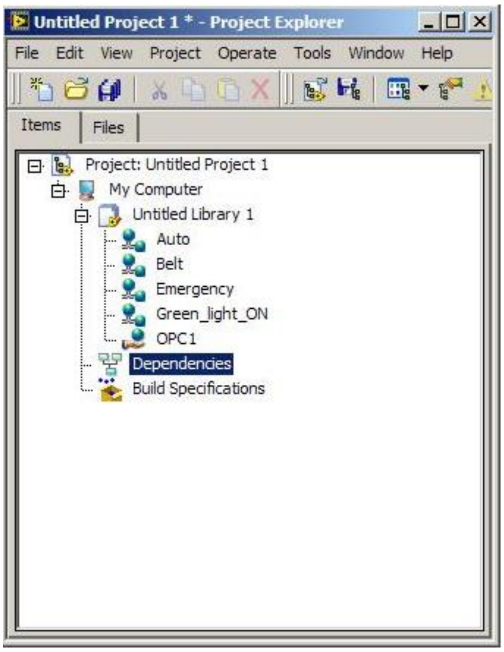


Figure 54 LabVIEW Project Explorer window showing the added shared variables (Pokharel, 2013)

Now, a VI can be added to the project using the following steps:

i. Right-click My computer on the project window and select 'New>>VI' as shown in Figure 55.

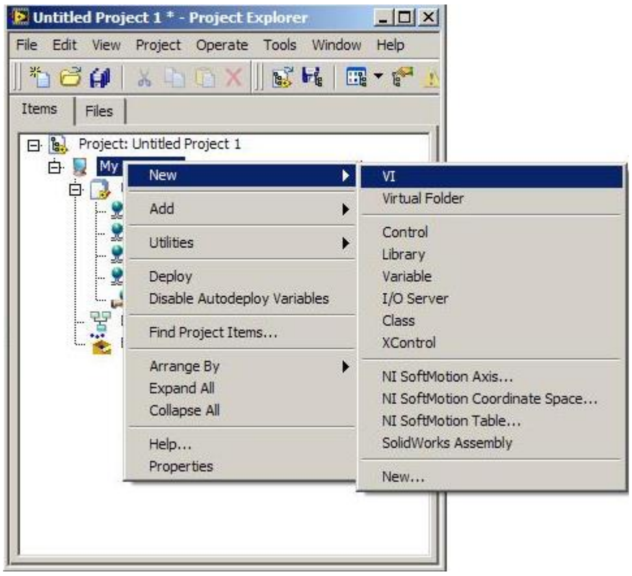


Figure 55 Adding a VI to the LabVIEW project (Pokharel, 2013)

ii. Then the required kind of program can be coded in the VI. The shared variables can be added easily by drag-and-drop method.

8 PATTERN MATCHING USING NI VISION

Figure 56 shows the graphical program or the human-machine-interface (HMI) created in LabVIEW for object sorting.

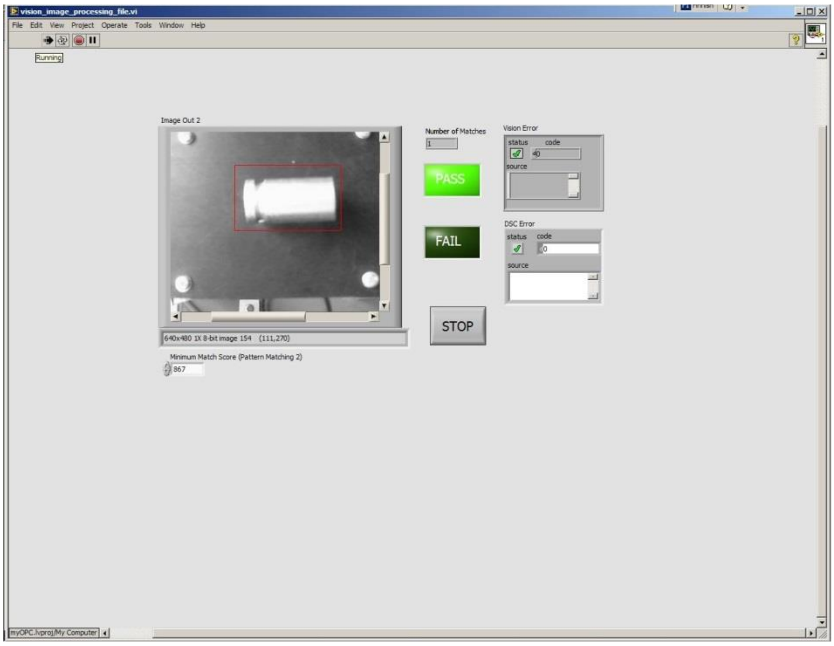


Figure 56 Pattern matching shown by a rectangular box around the object (Pokharel, 2013)

Express machine vision VIs were used for this purpose with some additional LabVIEW blocks and functions. Using express VI is easier in comparison to other separate machine vision blocks. It also reduces time and mistakes. Pattern matching machine vision tool was used to reach the goal of object sorting.

The program follows simple steps. Figure 57 shows the block diagram created for object sorting in LabVIEW.

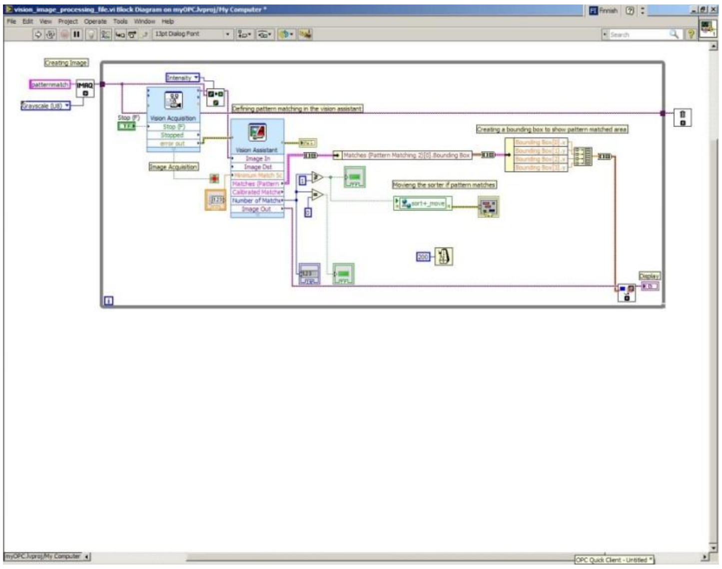


Figure 57 Block diagram created for object sorting, using pattern matching machine vision tool (Pokharel, 2013)

The steps that were used for developing the application shown in Figures 56, and 57 above are as follows:

1. First of all the camera was adjusted in a proper position such that it was perpendicular to the object scene.
2. Then image acquisition express VI was launched to check if the camera was working or not; a window as shown in Figure 58 appears. This is the place, where the acquisition device, acquisition type, acquisition settings, and controls and indicators can be selected.

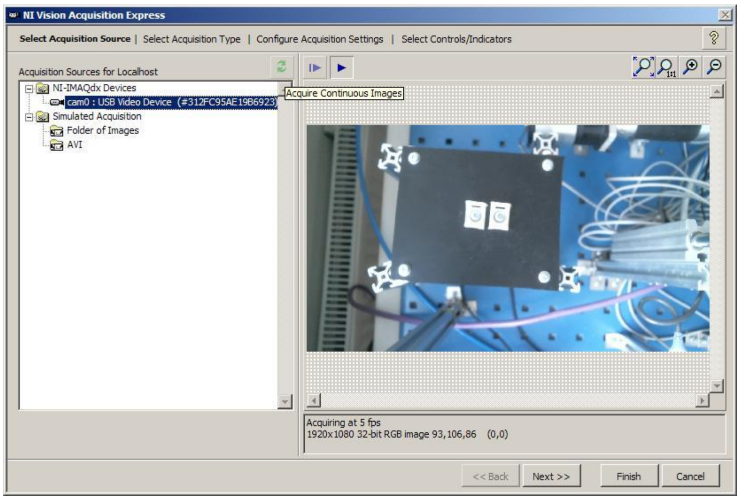


Figure 58 NI Vision Acquisition Express window, where different acquisition settings can be done (Pokharel, 2013)

The acquisition was set to continuous as shown in Figure 59.

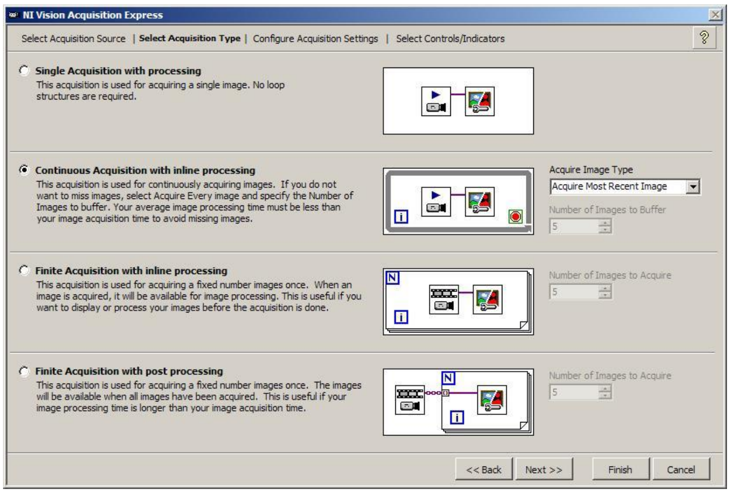


Figure 59 Selecting the acquisition type to Continuous (Pokharel, 2013)

After the settings were complete, the window was closed using ‘Finish’ button.

3. The created VI was allowed to ‘Run’ once, which was then stopped after an image was acquired.

4. Then an image was created using ‘IMAQ Create’ VI. Image name and the image type were also defined. This can be clearly seen in Figure 57 (p. 41).
5. The ‘New Image’ output terminal of the IMAQ Create VI was wired to the ‘Image Dst’ input terminal of ‘IMAQ ExtractSingleColorPlane’ VI (Vision Utilities>> Color Utilities). This block enables to extract one color plane only from the acquired color image. ‘Intensity’ was selected for the color plane. Also, the ‘Image Out’ output terminal of the vision acquisition block was wired to the ‘Image Src’ output terminal of the IMAQ ExtractSingleColorPlane VI. Pattern matching tool does not support RGB (32-bit) image and supports only 8-bit image. So choosing only one plane enables pattern matching to be used.
6. Then the ‘Vision Assistant’ express VI was launched and immediately navigated to Machine Vision tab; pattern matching was selected under this tab. This is shown in Figure 60.

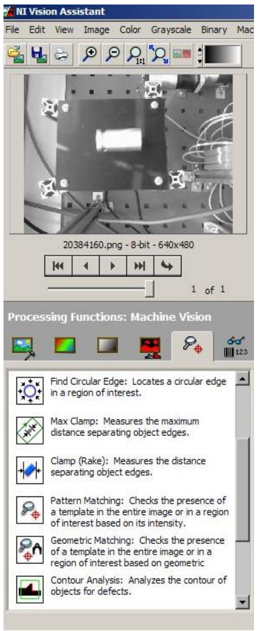


Figure 60 A section of Vision Assistant showing pattern matching tool in the machine vision tab (Pokharel, 2013)

7. After that, New Template button was clicked; it is visible on the left lower section of the pattern matching setup screen of the vision assistant. Then a template editor window appears where a region of interest (ROI) was created as shown in Figure 61 below using rectangular select tool that can be found on the right upper side of the window.

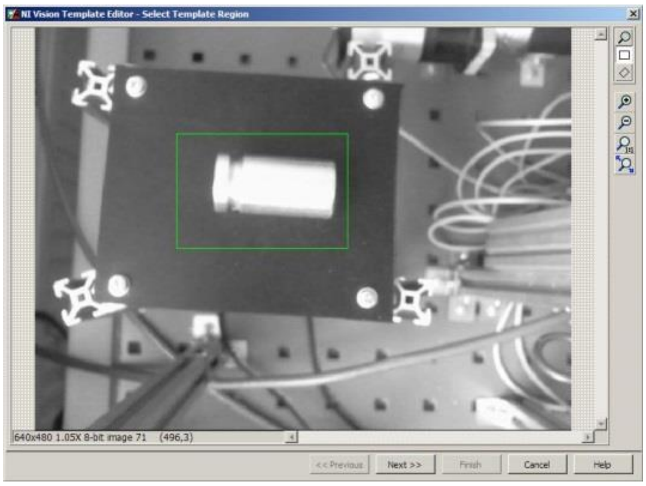


Figure 61 Selecting a region of interest (ROI) for template creation; ROI shown by green rectangle (Pokharel, 2013)

8. The selected area can be further tuned after clicking Next>> button that can be visible in the template editor window shown in Figure 61 above. And finally the template was created clicking the Finish button.
9. The x- and y- offsets were also defined but the angle shift was unchecked, because the objects being analyzed were always at the same place with the same orientation. Then the created template was saved into the hard-drive of the computer. Figure 62 shows the created template image for pattern matching. The x-position, y-position, angle and the score for the template image are generated by the vision assistant automatically; this can be noted from the area that is highlighted by blue color as shown in Figure 62.

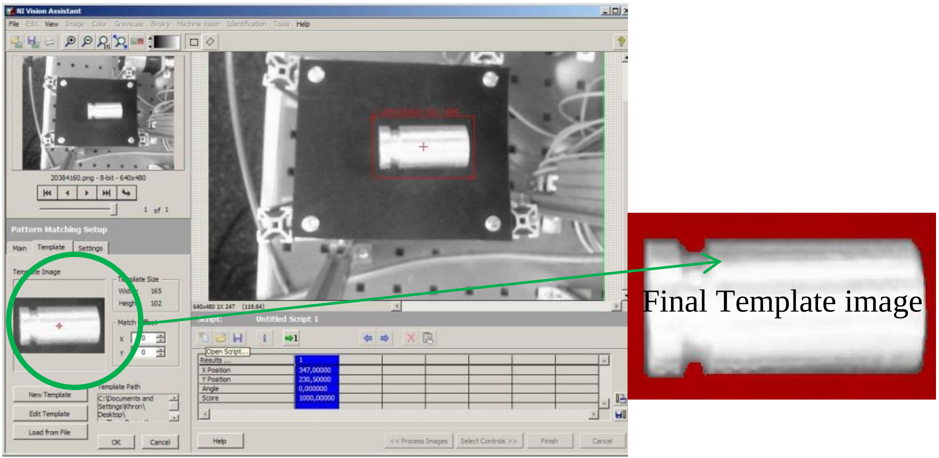


Figure 62 Creating an image template for pattern matching in NI vision assistant (Pokharel, 2013)

10. The required controls and indicators were also selected and finally the vision assistant was closed clicking Finish button.
11. The ‘Image Dst Out’ output terminal of the IMAQ ExtractSingleColorPlane was wired to ‘Image In’ input terminal of the vision assistant.
12. The ‘Matches Pattern’ output terminal from the vision assistant was wired to ‘array-to-cluster’ VI which then was connected to ‘Unbundle by name’ VI. Bounding box was selected by right clicking it and navigating to select item as shown in Figure 63 below.

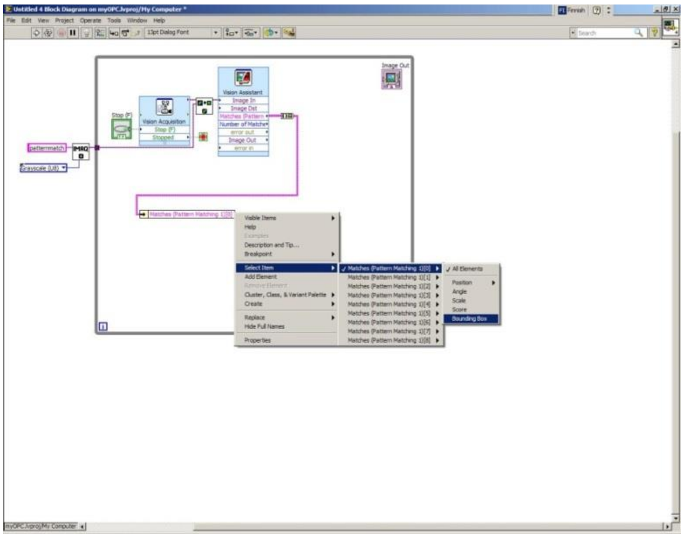


Figure 63 Selecting bounding box to create a bounding box (Pokharel, 2013)

13. The output terminal from the Unbundle by Name VI was once again wired to Array-to-Cluster VI, whose output terminal was also once again wired to another Unbundle by Name VI. This time it was enlarged to get 4 output terminals. The first output terminal was assigned to ‘Bounding Box[0]>x’; the second to ‘Bounding Box[1]>y’; the third to ‘Bounding Box[2]>x’; and the fourth to ‘Bounding Box[3]>y’as shown in Figure 64.

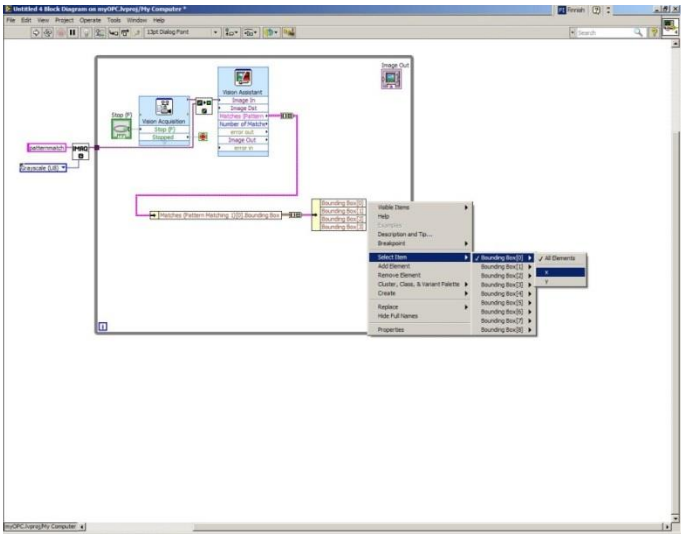


Figure 64 Assigning x- and y- axes to the bounding box (Pokharel, 2013)

14. Each of the outputs was then connected to the input terminals of a ‘Build Array’ VI. The output terminal of this VI was also wired to another Array-to-Cluster VI; one thing to be noted is that all the clusters’ size needs to be changed to 4; it can be done by right-clicking the Array-to-Cluster VI and selecting ‘Cluster Size’ from the context menu as shown in Figure 65 below.

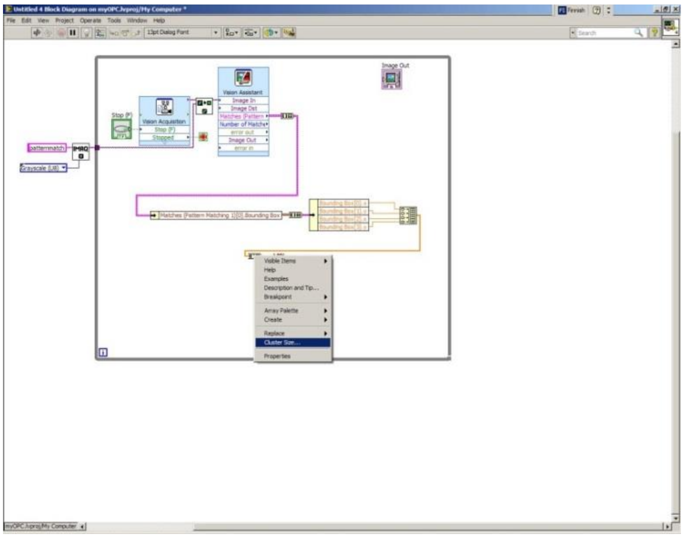


Figure 65 Changing the cluster size (Pokharel, 2013)

15. The output terminal from that final Array-to-Cluster VI was connected to the ‘Rectangle’ input terminal of the ‘IMAQ Overlay Rectangle’ VI. Also, the input terminal ‘Image’ of this VI was wired to the ‘Image out’ output terminal of the vision assistant.
16. Finally, the output terminal from the Overlay Rectangle VI was wired to the input terminal of the ‘Image Display’ VI.

Thus the machine vision application for object sorting was developed.

8.1 Object sorting

The output terminal ‘Number of Matches’ of the vision assistant was wired to a conditional operator “greater or equal to, (≥=)”. The other input terminal of that operator was given a constant value of 1. The output of this conditional operation was connected to the tag named “Sort+_Move” that was created in the OPC server. The part of LabVIEW block diagram that executes for sorting purpose is shown in Figure 66.

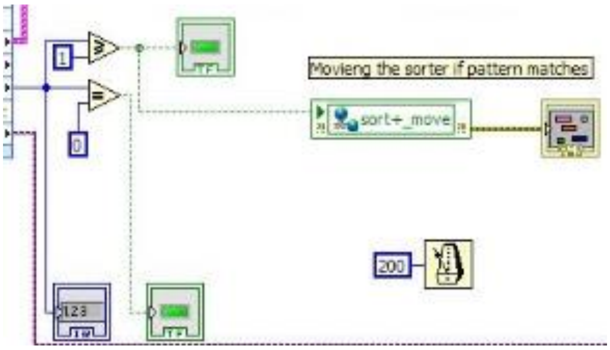


Figure 66 Object sorting section of the LabVIEW block diagram (Pokharel, 2013)

Every time the image matched the template image, or number of matches was equal to or greater than 1, the sorter moved and thus sorted the object. One tag was only used to make the program in LabVIEW simpler. As mentioned before in section6, the sorter reset was done in the automation codes in STEP 7.

Hence, the developed machine vision application was tested for object sorting.

9 CONCLUSIONS

From the project performed, it was obvious that machine vision technology and NI Vision tools together can be used for sorting objects in a factory production line. It was confirmed that out of many machine vision tools, the pattern-matching algorithm could be applied for the object sorting purpose. Also, it was clear that a simple webcam could be used for performing machine visions tasks. Because, a webcam was used for pattern matching, it is not for sure that it can be used for different machine vision applications. But, if the application is independent of different image processing tasks, it might be possible to use webcams instead of other industrial cameras.

The pattern matching was done at different times of the day; every time the score generated by the vision assistant needed adjustment for the correct pattern matching. This shows that pattern matching is based on direct correlation matrix algorithm. The score generated for the image template was 900. If the score was increased over 900, none of the objects matched the pattern; below 900 pattern matched, but if the score was below certain score (for example 850), the objects which tend to be of the same pattern also matched. In that case, the aim of pattern matching failed.

Although pattern matching is not affected by lighting, it was seen that shadow formation of some other objects over the analyzed object hindered to match the pattern stored in the pattern template. So to say that pattern matching is independent of lighting might be somewhat unrealistic. It might be said that pattern matching not affected by lighting directly, but there might be some other indirect cause relating to light that affects pattern matching.

It was also confirmed that PLC could be integrated with LabVIEW using OPC communication protocol; communication was enabled using NI OPC Servers in this thesis. The integration of simatic PLC was illustrated in this thesis but almost all the PLCs can be integrated with LabVIEW using NI OPC servers; provided there is an OPC driver for the PLC.

In this thesis document it was possible to present only a simple pattern recognition technique for the object sorting. Although the objective of sorting object using machine vision tool was met, it was realized that geometric matching tool would be more robust for sorting objects. The objects used in the project were all similar; defining only one parameter could be enough in such condition to get the result. But the case would not be the same always. The objects may have different shapes and sizes; in such cases, it is not enough to match only the pattern of the object. Measuring distances between two points, measuring diameters, etc. could be added to the application to make it more powerful and error-free.

SOURCES

ALLIED Vision Technologies 2006. Application Note: Choosing a Machine Vision Camera. Accessed 15th March 2013.
http://www.alliedvisiontec.com/fileadmin/content/PDF/Support/Application_Notes/ALL_ApplNote_Choosing_A_Machine_Vision_Camera.pdf

Basic Photography Tutorials 2001. An Introduction To Exposure. Accessed 16th March 2013.
http://www.silverlight.co.uk/tutorials/compose_expose/exposure.html

Davies, E. 2012. Computer and Machine Vision. 4th edition. Waltham: Academic Press.

EURESYS 2005. ADR Technology. Accessed 18th March 2013.
http://www.euresys.com/Products/grablink/ADRTechnology.asp

Gonzalez, R., Woods, R. & Eddins, S. 2009. Digital Image Processing using Matlab. 2nd edition. Gatesmark Publishing.

Logitech 2010. Logitech HD WebCam C615. Accessed 15th March 2013.
http://www.logitech.com/en-us/product/hd-webcam-c615

Lode's Computer Graphics Tutorial 2004. Image Filtering. Accessed 30th April 2013.
http://lodev.org/cgtutor/filtering.html

Machine Vision, pdf-file (ebook). Accessed 15th January-23rd April 2013.
http://www.cse.usf.edu/~r1k/MachineVisionBook/MachineVision.pdf

Movimed Custom Imaging Solutions 2007. Illumination. Accessed 16th April 2013.
http://www.movimed.com/Illumination.htm

National Instruments 2013. Connect LabVIEW to any industrial Network and PLC. Accessed 13th January 2013. http://www.ni.com/white-paper/5407/en

National Instruments 2012. Introduction to OPC. Accessed 7th March 2013.
http://www.ni.com/white-paper/7451/en

National Instruments 2012. LabVIEW 2012 Datalogging and Supervisory Control Module Help. Accessed 19th March 2013.
http://digital.ni.com/manuals.nsf/websearch/51AA757C42C65F0D86257A0C0029ABD0

National Instruments 2008. Robotics Fundamental Series: Feature Extraction. Accessed 18th April 2013. http://www.ni.com/white-paper/8222/en

National Instruments 2005. NI Vision for LabVIEW User Manual. Accessed 21st February 2013. http://www.ni.com/pdf/manuals/371007b.pdf

National Instruments 2006. Counting Particles or Cells Using IMAQ. Accessed 19th March 2013. http://www.ni.com/white-paper/3169/en

National Instruments 2011. Geometric Matching Example. Accessed 22nd February 2013. http://zone.ni.com/devzone/cda/epd/p/id/5555

National Instruments 2011. Bar Code Example. Accessed 23rd March 2013. http://zone.ni.com/devzone/cda/epd/p/id/5533

Pokharel, B. 2013. HAMK University of Applied Sciences. Degree Programme in Automation Engineering (2009-2013).

Quality Digest 2001. The Future of Machine Vision. Site Editor John E. Agapakis. Accessed 7th January 2013. http://www.qualitydigest.com/oct98/html/machfutr.html

Sudcamp 2012. A Technology Hub. Accessed 23rd March 2013. http://www.sudcamp.com/ccd-vs-cmos-image-sensor-in-digital-camera/

Vision Systems Oy. 2012. Robola project. Education Materials.

Wikimedia Commons 2012. File: Ccd-sensor.jpg. Accessed 18th March 2013. http://commons.wikimedia.org/wiki/File:Ccd-sensor.jpg

Wikipedia 2013. Camera resectioning. Accessed 18th February 2013. http://en.wikipedia.org/wiki/Camera_resectioning

Wikipedia 2013. Depth of Field. Accessed 17th March 2013. http://en.wikipedia.org/wiki/Depth_of_field

Wikipedia 2013. Image Resolutions. Accessed 17th March 2013. http://en.wikipedia.org/wiki/Image_resolution

Wikipedia 2013. Machine Vision. Accessed 15th December 2012. http://en.wikipedia.org/wiki/Machine_vision

WiseGeek 2011. What is Machine Vision Image Processing?. Site Editor Andrew Jones. Accessed 20th January 2013. http://www.wisegeek.com/what-is-machine-vision-image-processing.htm#did-you-know

Appendix 1

STEP 7 OB1-Functions Call

SIMATIC Object_sorting\ 04/24/2013 01:58:17 PM
SIMATIC 300(1)\CPU 315-2 DP\...\OB1 - <offline>

OB1 - <offline>

"CYCL_EXC"

Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 04/24/2013 01:57:17 PM
Interface: 02/15/1996 04:51:12 PM
Lengths (block/logic/data): 00168 00050 00022

Name	Data Type	Address	Comment
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

Block: OB1 "Main Program Sweep (Cycle)"

Network: 1

FC1
"Program Sequence"
EN ENO

Network: 2

FC2
"Reset_Sorter"
EN ENO

Network: 3

FC4
"Outputs"
EN ENO

Page 1 of 2

Appendix 2

STEP 7 Symbol Table

SIMATIC Object_sorting\ 04/24/2013 01:58:46 PM
SIMATIC 300(1)\CPU 315-2 DP\S7 Program(1)\Symbols

Properties of symbol table

Name:	Symbols
Author:	
Comment:	
Created on	02/18/2013 01:33:20 PM
Last modified on:	04/24/2013 01:58:38 PM
Last filter criterion:	All Symbols
Number of symbols:	39/39
Last Sorting:	Symbol Ascending

Status	Symbol	Address	Data type	Comment
	AUTO	I 2.1	BOOL	
	Belt Sensor	I 1.7	BOOL	
	COMPLETE RESTART	OB 100	OB 100	
	CYCL_EXC	OB 1	OB 1	
	EMERGENCY	I 2.4	BOOL	
	Green_light_ON	Q 0.5	BOOL	
	Gripper_close_input	I 0.5	BOOL	
	Gripper_close_output	Q 1.1	BOOL	
	Gripper_open_input	I 0.4	BOOL	
	MAN	I 2.2	BOOL	
	Motor	Q 1.3	BOOL	
	Outputs	FC 4	FC 4	
	Photoelectric	I 0.1	BOOL	
	Program Sequence	FC 1	FC 1	
	Push-	I 0.0	BOOL	
	Push-_Move	Q 0.0	BOOL	
	Push+	I 0.2	BOOL	
	Push+_Move	Q 0.2	BOOL	
	Red_light_ON	Q 0.7	BOOL	
	Reflective Sensor	I 1.5	BOOL	
	RESET	I 2.3	BOOL	
	Reset_Sorter	FC 2	FC 2	
	Rotate_0	I 0.6	BOOL	
	Rotate_0_output	Q 0.6	BOOL	
	Rotate_180	I 0.7	BOOL	
	Rotate_180_output	Q 0.4	BOOL	
	Sort-	I 1.3	BOOL	
	Sort-_Move	Q 1.0	BOOL	
	Sort+	I 1.1	BOOL	
	Sort+_Move	Q 1.2	BOOL	
	START	I 2.0	BOOL	
	Stop_Cycle	FC 3	FC 3	
	Table-	I 1.0	BOOL	
	Table-_Move	Q 0.1	BOOL	
	Table+	I 1.2	BOOL	
	Table+_Move	Q 0.3	BOOL	
	Tcenter	I 0.3	BOOL	
	VAT_1	VAT 1		
	Wait	M 0.0	BOOL	

Page 1 of 1

Appendix 3

STEP 7 OB100- Complete Restart Organization Block

SIMATIC Object_sorting\ 04/24/2013 01:59:03 PM
SIMATIC 300(1)\CPU 315-2 DP\...\OB100 - <offline>

OB100 - <offline>

"COMPLETE RESTART"

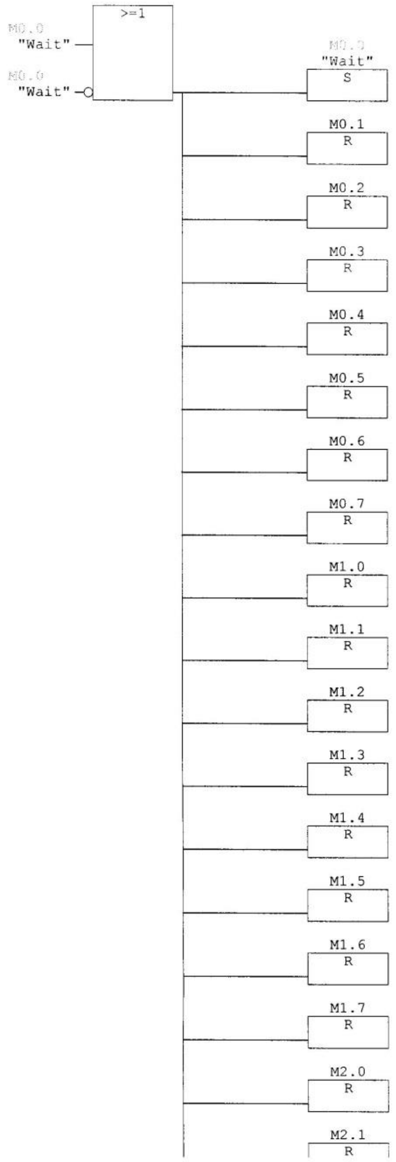
Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 04/21/2009 01:06:37 AM
Interface: 02/15/1996 04:51:10 PM
Lengths (block/logic/data): 00180 00070 00020

Name	Data Type	Address	Comment
TEMP		0.0	
OB100_EV_CLASS	Byte	0.0	16#13, Event class 1, Entering event state, Event logged in diagnostic buffer
OB100_STRTUP	Byte	1.0	16#81/82/83/84 Method of startup
OB100_PRIORITY	Byte	2.0	Priority of OB Execution
OB100_OB_NUMBR	Byte	3.0	100 (Organization block 100, OB100)
OB100_RESERVED_1	Byte	4.0	Reserved for system
OB100_RESERVED_2	Byte	5.0	Reserved for system
OB100_STOP	Word	6.0	Event that caused CPU to stop (16#4xxx)
OB100_STRT_INFO	DWord	8.0	Information on how system started
OB100_DATE_TIME	Date_And_Time	12.0	Date and time OB100 started

Block: OB100 "Complete Restart"

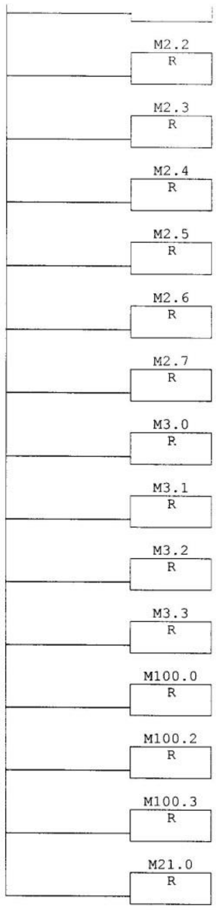
SIMATIC Object_sorting\ 04/24/2013 01:59:03 PM
SIMATIC 300(1)\CPU 315-2 DP\...\OB100 - <offline>

Network: 1



Page 2 of 3

SIMATIC Object_sorting\ 04/24/2013 01:59:03 PM
SIMATIC 300(1)\CPU 315-2 DP\...\OB100 - <offline>



Page 3 of 3

Appendix 4

STEP 7 FC1- Function for a Complete automation sequence

SIMATIC Object_sorting\ 04/24/2013 01:59:17 PM
SIMATIC 300(1)\CPU 315-2 DP\...\FC1 - <offline>

FC1 - <offline>

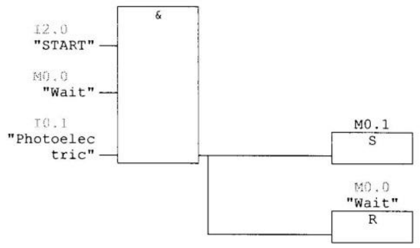
"Program Sequence"

Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 04/24/2013 01:46:09 PM
Interface: 02/18/2013 12:35:26 PM
Lengths (block/logic/data): 00388 00240 00000

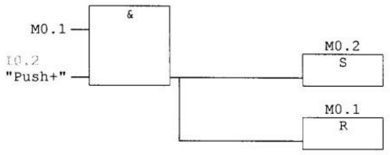
Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC1 Sequence flow of the program. Use of Set and Reset functions.

Network: 1

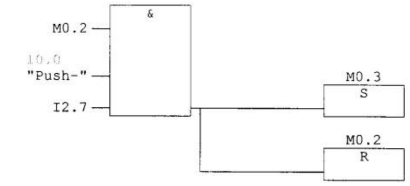


Network: 2

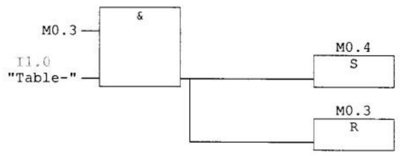


SIMATIC Object_sorting\ 04/24/2013 01:59:18 PM
SIMATIC 300(1)\CPU 315-2 DP\...\FC1 - <offline>

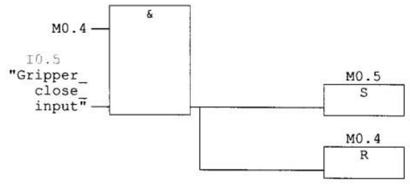
Network: 3



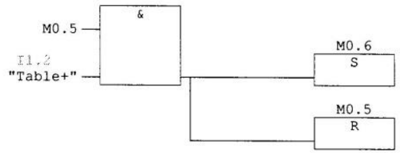
Network: 4



Network: 5



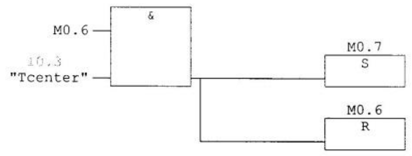
Network: 6



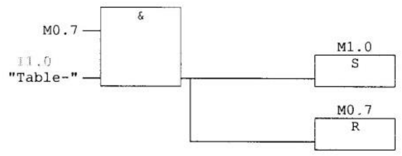
Page 2 of 6

SIMATIC Object_sorting\ 04/24/2013 01:59:18 PM
SIMATIC 300(1)\CPU 315-2 DP\...\FC1 - <offline>

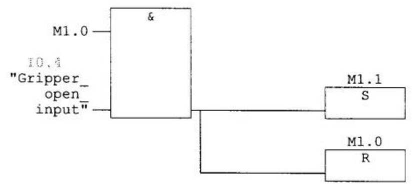
Network: 7



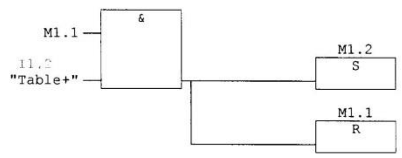
Network: 8



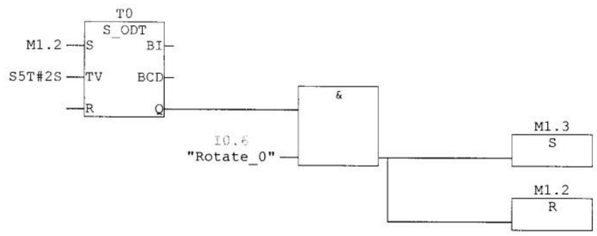
Network: 9



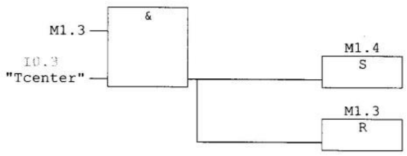
Network: 10



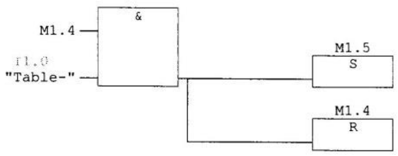
Network: 11



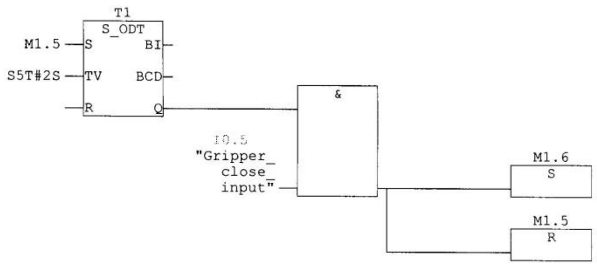
Network: 12



Network: 13

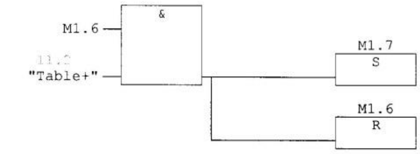


Network: 14

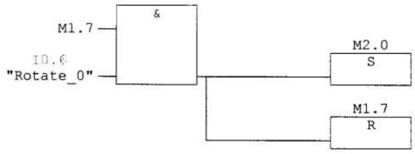


SIMATIC Object_sorting\ 04/24/2013 01:59:18 PM
SIMATIC 300(1)\CPU 315-2 DP\...\FC1 - <offline>

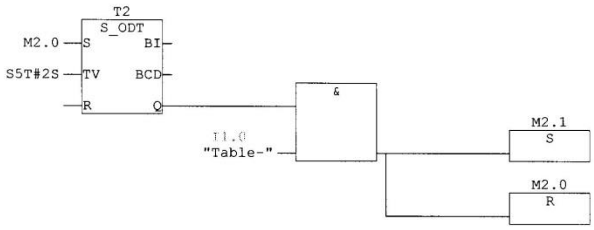
Network: 15



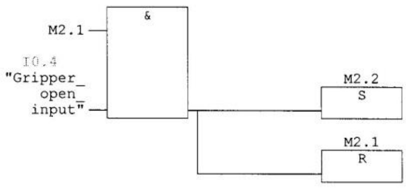
Network: 16



Network: 17

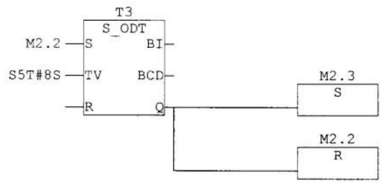


Network: 18

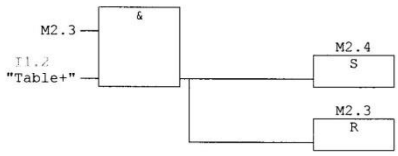


SIMATIC Object_sorting\ 04/24/2013 01:59:18 PM
SIMATIC 300(1)\CPU 315-2 DP\...\FC1 - <offline>

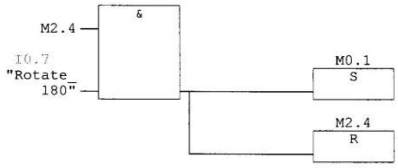
Network: 19



Network: 20



Network: 21



Appendix 5

STEP 7 FC4- Outputs Defined Function

SIMATIC Object_sorting\ 04/24/2013 02:43:54 PM
SIMATIC 300(1)\CPU 315-2 DP\...\FC4 - <offline>

FC4 - <offline>

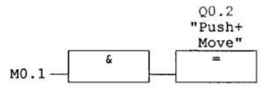
"Outputs"

Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 04/24/2013 01:56:50 PM
Interface: 04/24/2013 01:46:23 PM
Lengths (block/logic/data): 00178 00072 00000

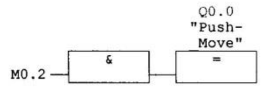
Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC4 Outputs
This function has all the outputs defined.

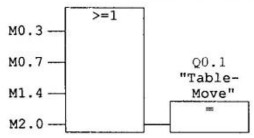
Network: 1 Push+Motion
When memory M0.1 is set, the piston moves out of the cylinder pushing the object out of the feeder.



Network: 2 Push-Motion
When memory M0.2 is set, the piston moves into the cylinder so that it is reseted for the next object push.

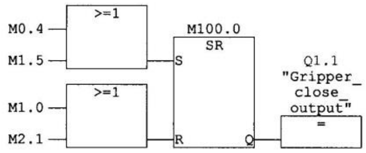


Network: 3 Rotary table Down
When at least one of the following mememories is set, the rotary table moves down.



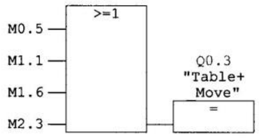
Network: 4 Gripper set/reset

The gripper is set to close or open using the SR function.



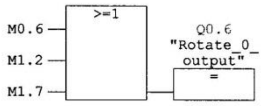
Network: 5 Rotary Table Up

When at least one of the following memeories is set, the rotary table moves up.



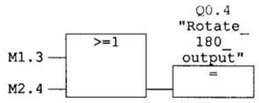
Network: 6 Rotation Clockwise

When at least one of the following memories is set, the table rotates in clockwise direction. The rotation may be 180 degrees or 90 degress depending upon the memories.



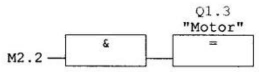
Network: 7 Rotation counterclockwise

When at least one of the following memories is set, the table rotates in counter-clockwise direction. The rotation may be 180 degrees or 90 degress depending upon the memories.



Network: 8 Motor

If memory M2.2 is set, the motor starts in clockwise direction.



Appendix 6

STEP 7 FC2- Sorter Reset Function

SIMATIC Object_sorting\ 04/24/2013 01:59:29 PM
SIMATIC 300(1)\CPU 315-2 DP\...\FC2 - <offline>

FC2 - <offline>

"Reset_Sorter"

Name: **Family:**

Author: **Version:** 0.1

Block version: 2

Time stamp Code: 04/24/2013 01:55:51 PM

Interface: 04/15/2009 02:56:36 AM

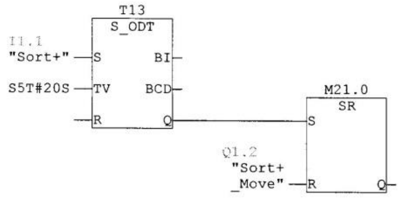
Lengths (block/logic/data): 00130 00034 00000

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC2 Sorter reset function

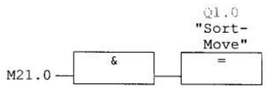
The reset function is active when the sorter is set through LabVIEW.

Network: 1



Network: 2

Whenever M21.0 is set the sorter piston will move into the cylinder.

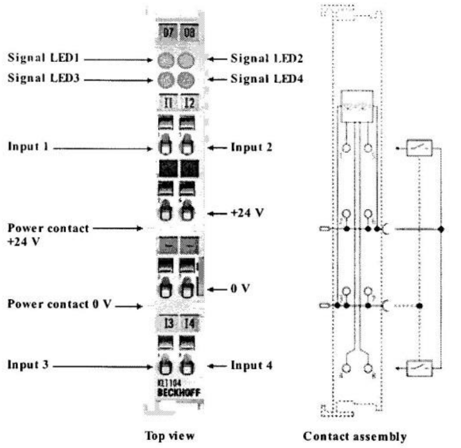


Appendix 7

Signal Module KL1114 Datasheet

Digital Input

KL1104, KL1114



KL1104, KL1114 | 4-channel digital input terminals 24 V DC

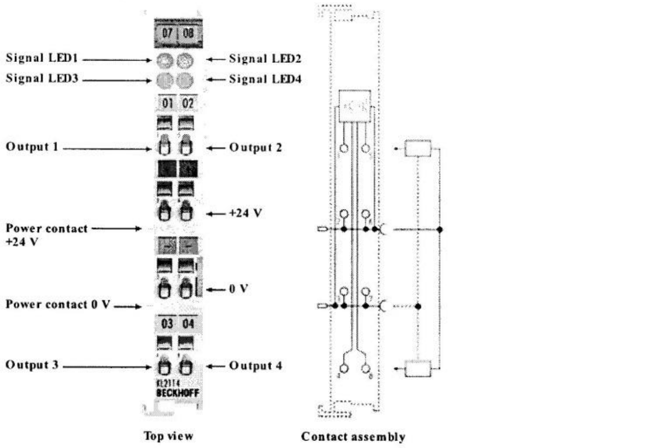
The KL1104 and KL1114 digital input terminals acquire the binary control signals from the process level and transmit them, in an electrically isolated form, to the higher-level automation unit. The KL1104 and KL1114 versions have input filters of different speeds. The Bus Terminals contain four channels that indicate their signal state by means of light emitting diodes. The KL1104 and KL1114 are particularly useful for space-saving use in control cabinets.

Technical data	KL1104 | KS1104	KL1114 | KS1114
Connection technology	2-/3-wire	
Number of inputs	4	
Nominal voltage	24 V DC (-15 %/+20 %)	
"0" signal voltage	-3…+5 V	
"1" signal voltage	15…30 V	
Input filter	typ. 3.0 ms	typ. 0.2 ms
Input current	typ. 5 mA	
Current consumpt. K-bus	typ. 5 mA	
Electrical isolation	500 V (K-bus/field potential)	
Bit width in the process image	4 inputs	
Configuration	no address or configuration setting	
Weight	approx. 55 g	
Operating/storage temperature	-25…+60 °C/-40…+85 °C	
Relative humidity	95 %, no condensation	
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27	
EMC immunity/emission	conforms to EN 61000-6-2/EN 61000-6-4	
Protect. class/installation pos.	IP 20/variable	
Pluggable wiring	for all KSxxxx Bus Terminals	
Approvals	CE, UL, Ex, GL	

Appendix 8

Signal Module KL2114 Datasheet

Digital Output KL2114, KL2134



KL2114, KL2134 | 4-channel digital output terminals 24 V DC

The KL2114 and KL2134 digital output terminals connect the binary control signals from the automation unit on to the actuators at the process level with electrical isolation. The load current output of the KL2114 version is protected against overload and short-circuit. The KL2134 is protected against reverse polarity connection. The Bus Terminals contain four channels that indicate their signal state by means of light emitting diodes.

Technical data	KL2114 | KS2114	KL2134 | KS2134
Connection technology	2-/3-wire	
Number of outputs	4	
Rated load voltage	24 V DC (-15 %/+20 %)	
Load type	ohmic, inductive, lamp load	
Max. output current	0.5 A (short-circuit-proof) per channel	
Short circuit current	< 2 A	
Breaking energy	< 150 mJ/channel	
Reverse voltage protection	–	yes
Electrical isolation	500 V (K-bus/field potential)	
Current consumption power contacts	typ. 30 mA + load	
Current consumpt. K-bus	typ. 9 mA	
Bit width in the process image	4 outputs	
Configuration	no address or configuration setting	
Weight	approx. 70 g	
Operating/storage temperature	0…+55 °C/-25…+85 °C	-25…+60 °C/-40…+85 °C
Relative humidity	95 %, no condensation	
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27	
EMC immunity/emission	conforms to EN 61000-6-2/EN 61000-6-4	
Protect. class/installation pos.	IP 20/variable	
Pluggable wiring	for all KSxxxx Bus Terminals	
Approvals	CE, UL, Ex	CE, UL, Ex, GL