CERTIFICATION OF APPROVAL

Visual Automation Tool for Server Validation Process

by

Shazreen Hanna bt. Che Nordin

A project dissertation submitted to the
Information Communication Technology Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirements for
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION COMMUNICATION TECHNOLOGY)

Approved by,

(Mr. Halabi Hasbullah)
PROJECT SUPERVISOR

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
July 2007

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

SHAZREEN HANNA BT. CHE NORDIN

ABSTRACT

The objective of the study is to develop an application that will automate the process of server validation which is observing the status of the server via its LED colors. It is in the form of an executable LabVIEW program that contains the image processing functions and captures the real-time images of the LED and records the necessary output (color results) in a text file format which can be use for future analyzing purpose. Initially, the server validation processes were done manually with the testers' intervention. Testers have to observe the LED color by themselves. There are many available color detection methods available but most of them employ only hardware part and do not support the real-time image grabbing as per required in the server validation process. The project that is going to be developed is enhanced with the aim to solve these problems. The project's methodology can be divided into two: research methodology and design methodology. Research will generally focus on studying the available color detection methods. For the design methodology, a type of Spiral Development Model is adopted in view of the fact that the prototype is enhanced and refined from the previous prototype based on the recursive steps. As a conclusion, this Visual Automation Tool is a practical solution in supporting the manufacturing process by the means of automating it, using technology and future development.

ACKNOWLEDGEMENTS

In the name of God, the most gracious, the most merciful.

First of all, I would like to thank my respective supervisor, Mr. Halabi Hasbullah for assisting me throughout the long period of completing this project, at which I faced trial and tribulation of various events. Without his help and guidance, this project would never been completed successfully.

Special thanks to my internship supervisor from Intel Technology Sdn. Bhd., Ms. Choo Yoon Yan for exposing me to such a challenging project. The experience gained has eventually developed my skills and capability in handling this project on my own.

To the panel of examiners, thank you for your willingness to diligently evaluate my work. I really appreciate your thought and time spent on this project.

Let me extend my gratitude to the coordinator of FYP program, Ms Aliza Sarlan for the unconditional support and her enthusiasm in guiding all of the students through the whole FYP processes, never accepting less than our best.

For my parents, family and friends, I would like to thank all of them for what they have done. They were always there for me and made me believe that this project is going to succeed.

Lastly, I would like to thank all of those who were involved directly or indirectly in this Visual Automation Tool project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1. Background of Study

In the server validation processes, there are two things need to be concerned with which are: (1) detecting the color of LED and, (2) calculating the blink frequency of the LED. The state of the server is represented by its LED color (e.g. when the board is in healthy condition, the LED would be green in color while red for the unhealthy condition), while for some special cases (it happens very rare), the tester has to calculate the blink frequency of the LED (e.g. the hotswapable state of the server board is determined by its blink frequency).

However, these two tasks would be a bit difficult to be conducted manually if the number of server boards per validation process is beyond that tester's capability. The accuracy of the testing results can also be an issue since both tasks require rapid-continuous reflexes from the tester. Based on these reasons, an automated system is needed to replace the manual validation test in order to increase the productivity as well as improving the test result's accuracy.

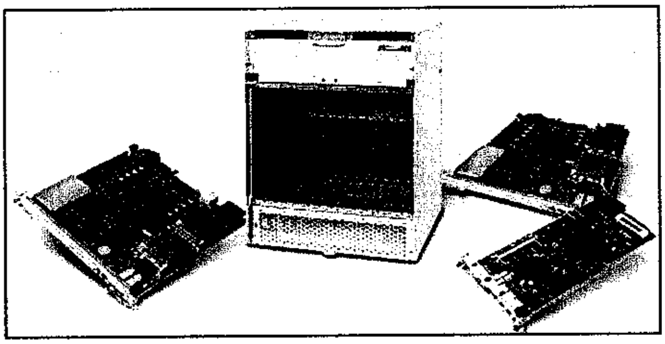


Figure 1.1: Server and the server boards

Server board or also known as single board computer typically refers to the compute blade used in the server. It forms like the real computer with complete components such as RAM, chipsets, processor etc. There are many types of server boards available out there which are varying in the types of processor and chipset. Below is the example of server board named Kennicott (Intel):

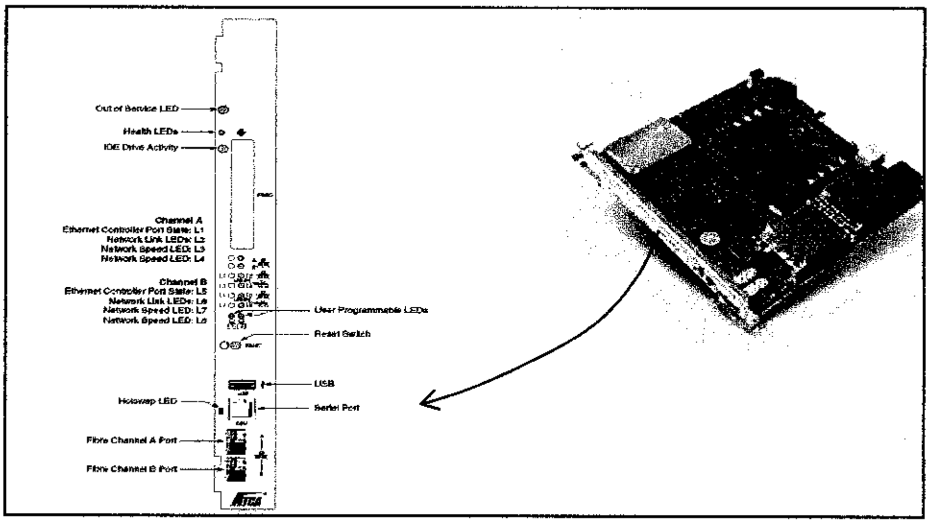


Figure 1.2: Example of the server board

Table 1.1 below shows some of the significant LED types together with the color and status represents by the color:

Table 1.1: LED color and status

LED Name	**Color**	**Condition**
Out of Service (OOS)	Red	Not in service
Health	Red	Critical/non-recoverable error detected
	Green	Healthy
IDE	Green	Disk access (read/write activity)
User	Green	In user defined functions
Network Link	Solid Green	Link established
	Blinking Green	Link with activity
Ethernet Controller	Amber	Active status of user defined functions

1.2. Problem Statement

1.2.1. Problem Identification

Manual test's efficiency is solely depends on the testers. Sometimes when the testers are getting tired, they cannot focus to monitor the color of the LED as well as its blink frequency. This will lower down the testing accuracy as the testers might have slow response on the color changing and blinking time may vary.

Although there are a lot of color automation tools available out there, most of the tools were built focusing on the hardware part only which utilize the technology of electronic components (please refer to appendix no.4) and of course, the price would be a bit costly. Furthermore, these available tools do not have the features to keep the testing result in a text file etc, as per required in server validation processes. In the aspect of cost and efficiency, this visual automation prototype requires only a webcam, with the aid of programming codes, and it is flexible enough to be used in server room or any confined spaces as it can be mounted easily on the server.

1.2.2. Significant of the Project

With the creation of this project, the server validation process can now be automated and the productivity of the team can be increased. The issues of color and blink rate accuracy can subsequently being addressed by improving the accuracy of test results. Moreover, the testers would also feel the convenience of having a recorded output data that can be used again for future assessment. Through some little alterations, this prototype can also be used for other usage e.g. reading signals that employ color as the representation of system's state, helping color-blind people to correctly translate color, etc.

1.3. Objectives and Scopes of Study

1.3.1 Objectives

Visual Automation Tool project is carried out in order to aid the hardware validation team in conducting validation processes for the server board. The main goal is to address the problems of accuracy and resolve the need to execute the tests manually. It automates the visual testing for the server since it is capable of comparing color LED as well as monitoring the LED blink frequency. In addition, this prototype is developed specifically to be used in server validation processes and it fulfill most of the requirements such as portability, efficiency and cost-effective.

1.3.2. Scopes of Project

The scopes of project can be divided into two which are research and supporting materials collection, and project development.

Research and supporting materials collection would be done in the initial phase to gather the useful information pertaining to the project. Literature review will be produced which includes others' paper work, online article and etc.

Project development will include project planning, continued by analysis part, development works and lastly the implementation which involve the image processing functions, written in LabVIEW codes.

CHAPTER 2
LITERATURE REVIEW AND THEORY

2.1. Image Processing

According to Murray (2003)

> Image processing is analyzing and manipulating images with a computer. Image processing generally involves three steps. The first one is importing an image with an optical scanner or directly through digital photography. The second step is manipulating or analyzing the image in some way. This stage can include image enhancement and data compression, or the image may be analyzed to find patterns that aren't visible by the human eye. And the third step is output the result. The result might be the image altered in some way or it might be a report based on analysis of the image.

This idea shares a similar underlying concept with Visual Automation Tool. It is because the tool depends on the web camera to receive the input which is the continuous real-time images. These images are then going to be analyzed; some color processing functions will be implemented on the images to produce the expected output. According to Murray, the third step is output the result. This project is going to have a kind of report (data output files) which holds the results of detected LED colors. Besides this output file, the system also has a GUI (graphical user interface) window that displays the real-time results.

2.2. Color Detection Method

According to Trinh, Schwartz et al. (2003)

> For the color detection theory, we can distinguish unknown colored surfaces by calculating the *Euclidean Distance* against each of our known vectors. The vector producing the shortest *Euclidean Distance* will have the greatest likelihood of being the unknown color.

Euclidean Distance is widely used in determining the color of the unknown color surface. Supposed we have ONE unknown color surface and number of known color surfaces of any colors (e.g. green, blue, red, yellow, purple, brown etc.). These surfaces are then being tested by reflecting 4 colors of LED (RGB colors and white) on the test surface. The values obtained by each color surface are then recorded. The Euclidean values are then calculated. The color that holds the smallest value will be the color of the unknown surface. Please refer to appendix 3 for calculation example.

Most intelligent robots or electronic devices that apply color detection will implement *Euclidean Distance*. It provides the flexibility for each color group where the range of value for each color group is wider; the value of the unknown color surface does not have to be the exact as the value of color that it is supposed to be.

$$D_{Euclidean} = \sqrt{\sum_{n=1}^{m} (X_n - Y_n)^2}$$

D = Euclidean Distance
X = Unknown Vector
Y = Known Vector
M = Number of test points per vector

Figure 2.1: *Euclidean Distance* Formula

RGB (red, green and blue) colors are used because each color is represented by the combination of these colors. Please refer to Table 2.1 for more examples on RGB values for each color.

Table 2.1: RGB Color Values

Color	R value	G value	B value
Black	0	0	0
White	255	255	255
Red	255	0	0
Green	0	255	0
Blue	0	0	255
Yellow	255	255	0
Cyan	0	255	255
Magenta	255	0	255

Figure 2.2 below shows the actual RGB color representation. RGB (red, green and blue) colors are combined in various ways to produce other colors e.g. combination of red, green and blue will produce white color while yellow, magenta and cyan are produced by the combination of two of these RGB colors.

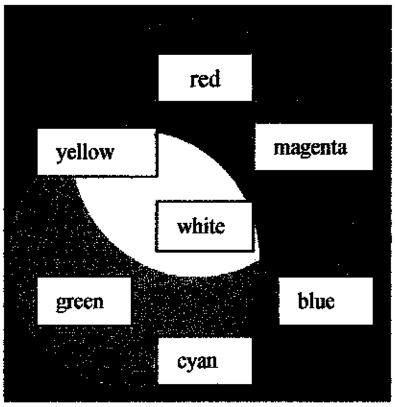


Figure 2.2: RGB color representation

2.3. Color Detection Pitfalls

According to Trinh (2002)

> There are two main pitfalls which hinder the accuracy of any color detection scheme, which are the presence of ambient light and inconsistent positioning of the objects. These two problems are the by far the biggest obstacles one must overcome when attempting color detection. The actual color detection process is very straightforward and easy to reproduce once the previous 2 conditions are met.

The static position of the camera is important because the color is being processed based on its fixed ROI (region of interest). If we positioned the LED out of its region, the tool might not interpret the color correctly or even worst, it might not detect the color at all.

Another issue is the presence of ambient light. To get the best result, the excessive ambient light should be filtered out by covering the LED with something e.g. by using cloth, cover the LED inside the box etc. It is to avoid any disruption on recording the accurate color result since the presence of ambient light will alter the actual color of the LED.

Figure 2.3 on the next page shows the image of the red, green and blue LEDs being captured without the presence of ambient light (top) and captured with the presence of the ambient light (bottom). We can clearly see that the LEDs appear to be whitish, very pale and color-obscured if the ambient light is not filtered, like shown by the images on the top row. This will hinder the accuracy of the test result since the tool might interpret the whitish LED color to be the same although they should be in different color group.

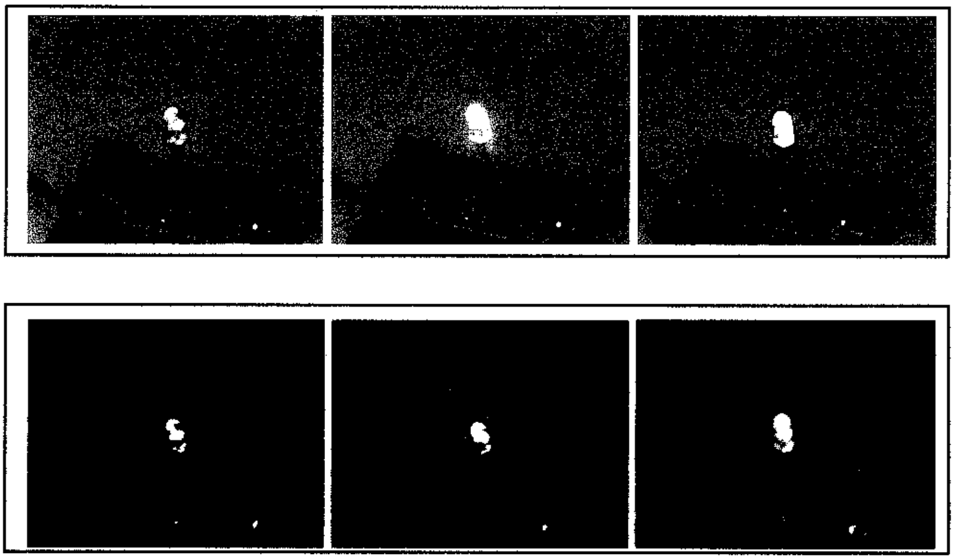


Figure 2.3: The LED color with (top) and without (bottom) the presence of ambient light

CHAPTER 3
METHODOLOGY

3.1. Research Methodology

Reviewing will be implemented as the main of research methodology. The core objective of reviewing is to gather the useful information relevant to the project. It includes reviewing the online report, articles and others projects that are related to this Visual Automation Tool project. In the end, the useful information such as the best method of comparing color is incorporated into the design of the prototype.

3.2. Design Methodology

Spiral Development Model is adopted as a design methodology. This model was originally proposed by Boehm (Boehm, 1988). It represents the software as a spiral sequence of activities with some backtracking from one activity to another. This model is chosen since many phases were involved in creating the prototype for Visual Automation Tool where each of the prototypes is enhanced and refined from the previous prototype based on these recursive steps:

- *Evaluate earlier prototype*

  The first prototype of Visual Automation Tool is developed and it is evaluated based on its strength, weaknesses and risks, focusing on its accurateness of color detecting methods and blink frequency calculation.

- *Define requirements for new prototype*

  Using the information gathered from analyzing the earlier prototype, the new enhanced requirements are defined. There might be some alterations to the color detecting methods in order to produce the most accurate result.
- *Plan and design new prototype*

  The new prototype is planned and designed, considering all the strengths, weaknesses and risks of the previous prototype.
- *Construct and test new prototype*

  The new, enhanced prototype is constructed and tested.

There will be number of prototypes going to be developed. Risk analysis will be done firstly, followed by developing and evaluating the prototype and lastly, requirements for the next phase's prototype are defined. These steps will be repeated recursively.

Spiral model offers a great flexibility to the developers as the risks can be explicitly assessed and problems are resolved throughout the processes. Then the final system is constructed, based on the refined prototypes.

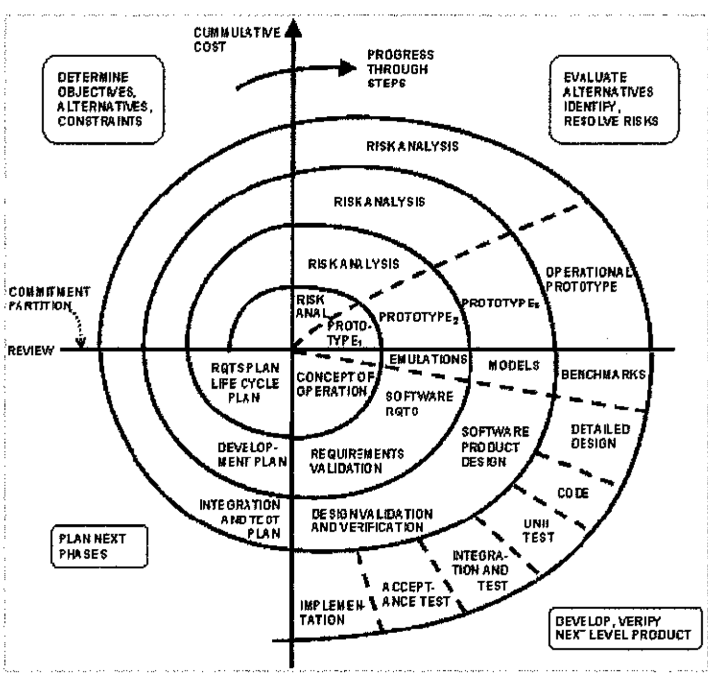


Figure 3.1: Spiral Development Model

3.3. Project Activities

During the first semester of Final Year Project (FYP), two phases of Software Development Lifecycle were carried out, which are Planning and Analysis. Throughout this second semester of FYP, the other two phases which are Design and Implementation phases will be carried out.

3.3.1. Planning

Planning is the initial phase in any software development cycle. This phase has successfully done during the first semester. The tasks that fall under the group of planning include:

- *Defining system to be developed*

  In this case, the project of Visual Automation Tool is identified as the chosen system to be developed according to the problem statements given.
- *Project scope*

  It involves defining high-level system requirements and these data are then put into a project scope.
- *Developing project plan*

  This plan consists of formalizing all details of tasks to be completed and when are they going to be completed.
- *Managing and monitoring project plan*

  This step allows us to stay on track, creating project milestones and feature creeps which let us to add to the initial plan.

3.3.2. Analysis

Analysis is the second phase in software development cycle, executed as a continuation of the planning phase. This phase was also done during the first semester of FYP. The tasks that fall under the group of analysis include:

- *Gathering requirements*

  It involves the task of gathering all the requirements that specify what the system should do (its functions), and its essential and desirable system properties.
- *Analyzing the requirements*

  The requirements gathered are then being analyzed and prioritized according to their importance to the project. It is really essential to ensure these requirements are met successfully during the next phase (design).

3.3.3. Design

Design phase is carried out on the second semester of FYP, together with the Implementation phase. Design tasks are:

- *Designing and developing Replica of Server Board*

  Replica of the server board is built from an electronic circuit that consists of the LEDs, chip, resistor and transistor. It holds number of LEDs that represent each color (blue, green, yellow, amber and red) and one LED (blue) is designed to be blinked since it represents the blinking hotswap LED.
- *Defining system's input and output*

  This task is important since it will signify how the real system is working, focusing on the input and output of the system.
- *Deriving system's Flow Chart*

  The flowchart is really useful to specify the flow of the system's core functions, starting from the initial state until the end of the system's execution.

3.3.4. Implementation

Implementation phase will be the most crucial phase for the whole project works. In this phase, all information gathered during the planning, analysis and design phases will be applied to produce the end product which is the visual automation tool prototype.

3.4. Gantt Chart of the Project Progress

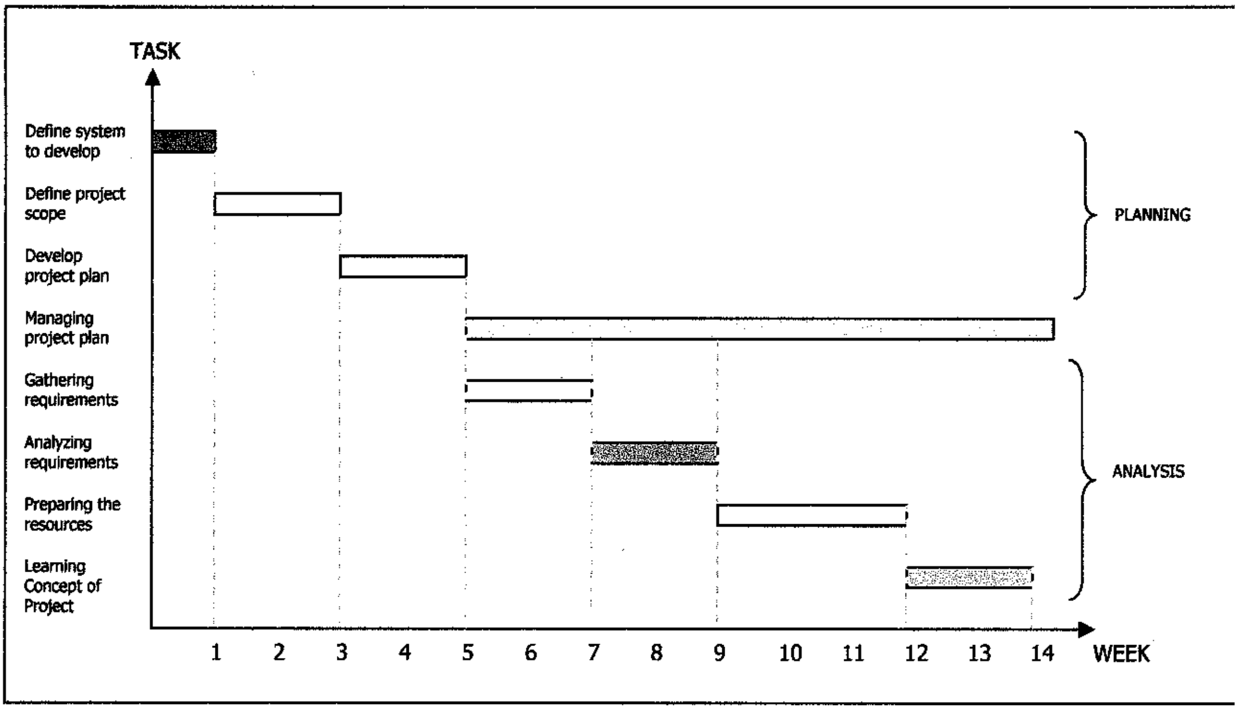


Figure 3.2: Gantt chart of Project Progress (First Semester)

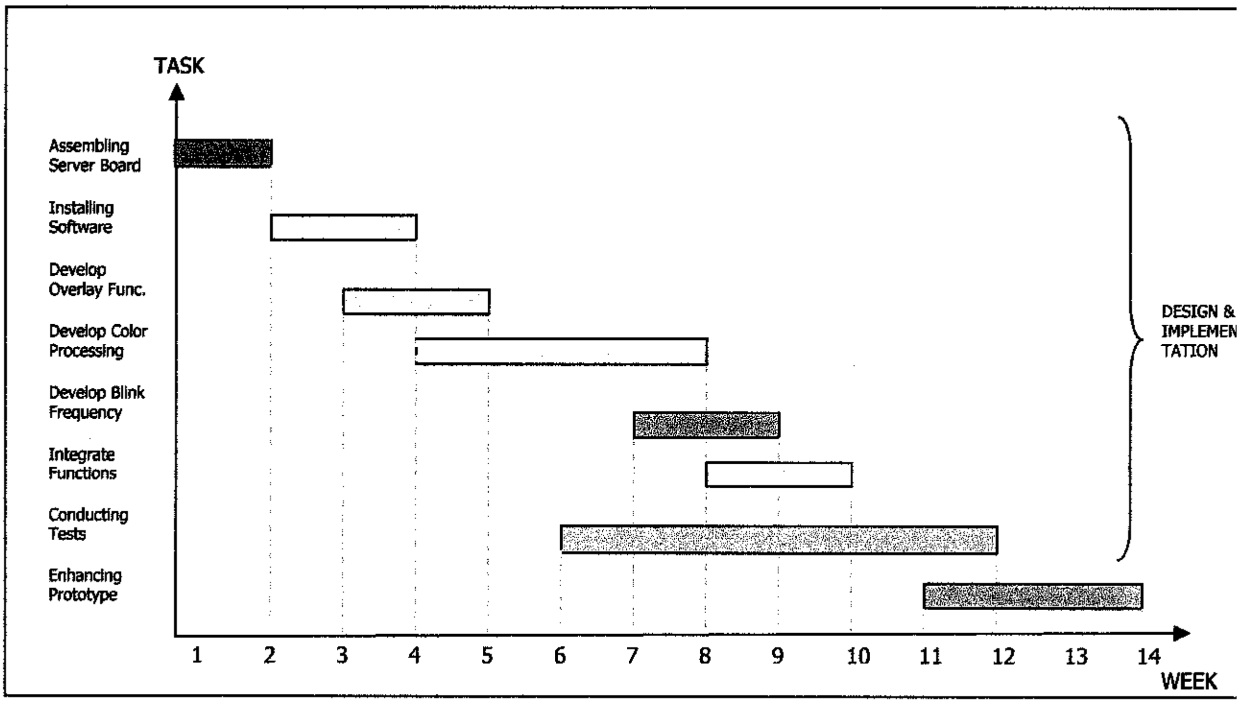
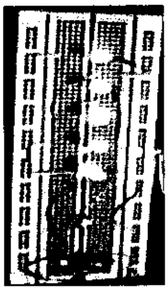


Figure 3.3: Gantt chart of Planned Project Progress (Second Semester)

3.5. Tools Required

Table 3.1: Tools required for Visual Automation Tool's development

Hardware	USB camera (webcam)	
	PC as a controller	
	LED board that replicates the server board	
Software	Machine Vision software	TiNA
	Microsoft Visual C++	Microsoft Visual C++
	NI LabVIEW and Vision Assistant	LabVIEW NI Vision Assistant

CHAPTER 4

RESULT AND DISCUSSION

4.1. Result

4.1.1. The Server Board

An electronic circuit is built to replicate the server board. This electronic circuit consists of the LEDs, chip, resistor, capacitor and transistor. It holds number of LEDs that represent each color (blue, green, yellow, amber and red) and one LED (blue) is designed to be blinked since it represents the blinking hotswap LED. The diagrams below show the steps involved in creating the server board.

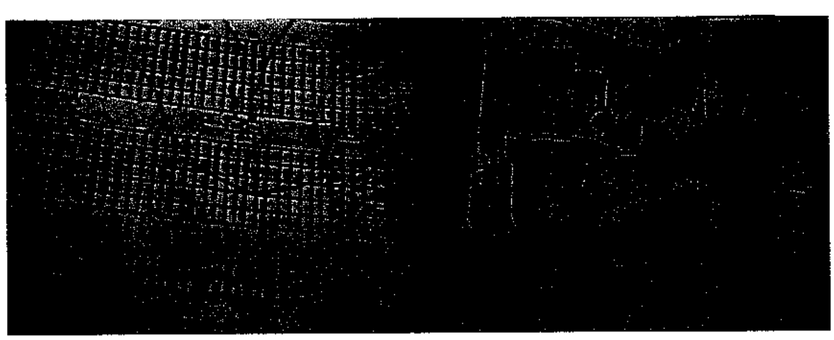


Figure 4.1: Deriving the circuit diagrams of the circuit

The circuit diagram is the first thing needs to be derived since it shows the simplified pictorial representation of the electronic circuit. It depicts the components of the circuit as simplified standard symbols, and the power and signal connections between the devices. This diagram is then used as guidance in assembling the components on the circuit board.

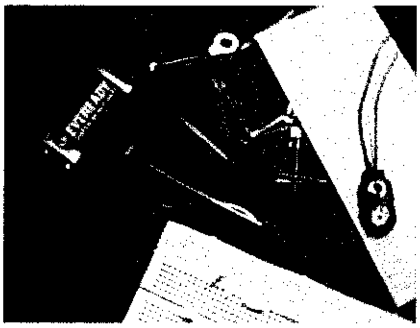


Figure 4.2: Electronic components needed to build the circuit

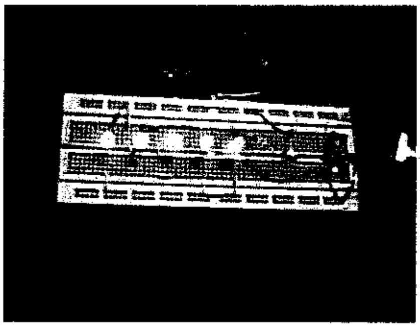


Figure 4.3: Completed 'server board' circuit

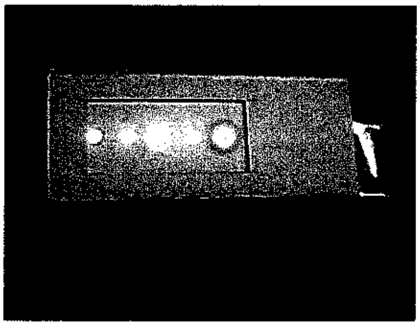


Figure 4.4: Completed 'server board' circuit together with the cover

4.1.2. Prototype Work

The end product of this project would be the prototype of Visual Automation Tool. The tool is constructed in LabVIEW script. Here are the steps involved in creating the prototype:

- *Creating color processing functions in Vision Assistant*

  An image of server board is grabbed using the webcam in Vision Assistant Then, the color processing functions are going to be applied on this static image. After the color processing functions are successfully created, the Vision Assistant script is produced.

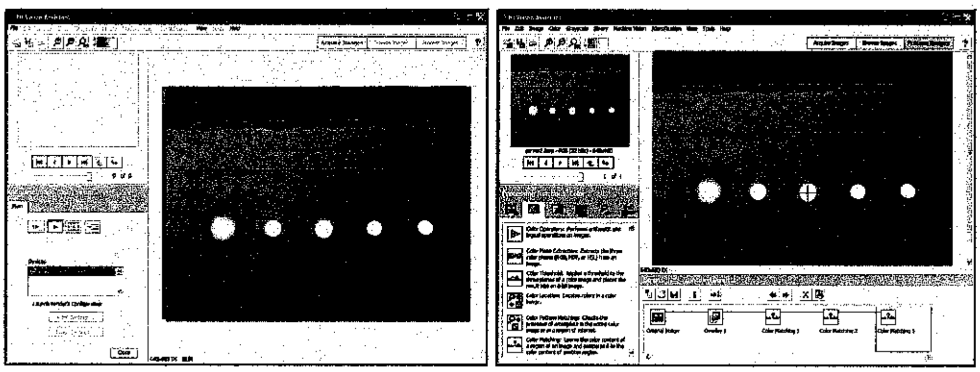


Figure 4.5: Vision Assistant Script

- *Converting the Vision Assistant script into LabVIEW script*

  LabVIEW script is needed to test out the color processing functions that have been created in Vision Assistant script. It is because, Vision Assistant script only applicable to be tested on a single static image while in LabVIEW, the script can be tested on continuous real-time images. Real-time images will be helpful in determining whether the script created is flexible to the changes like lighting, camera position etc. Vision Assistant has a special feature of converting its script into LabVIEW script which is also known as the block diagram (as being shown in figure 4.6). When this block diagram is executed, it will produce the system's front panel like being shown in diagram 4.7.

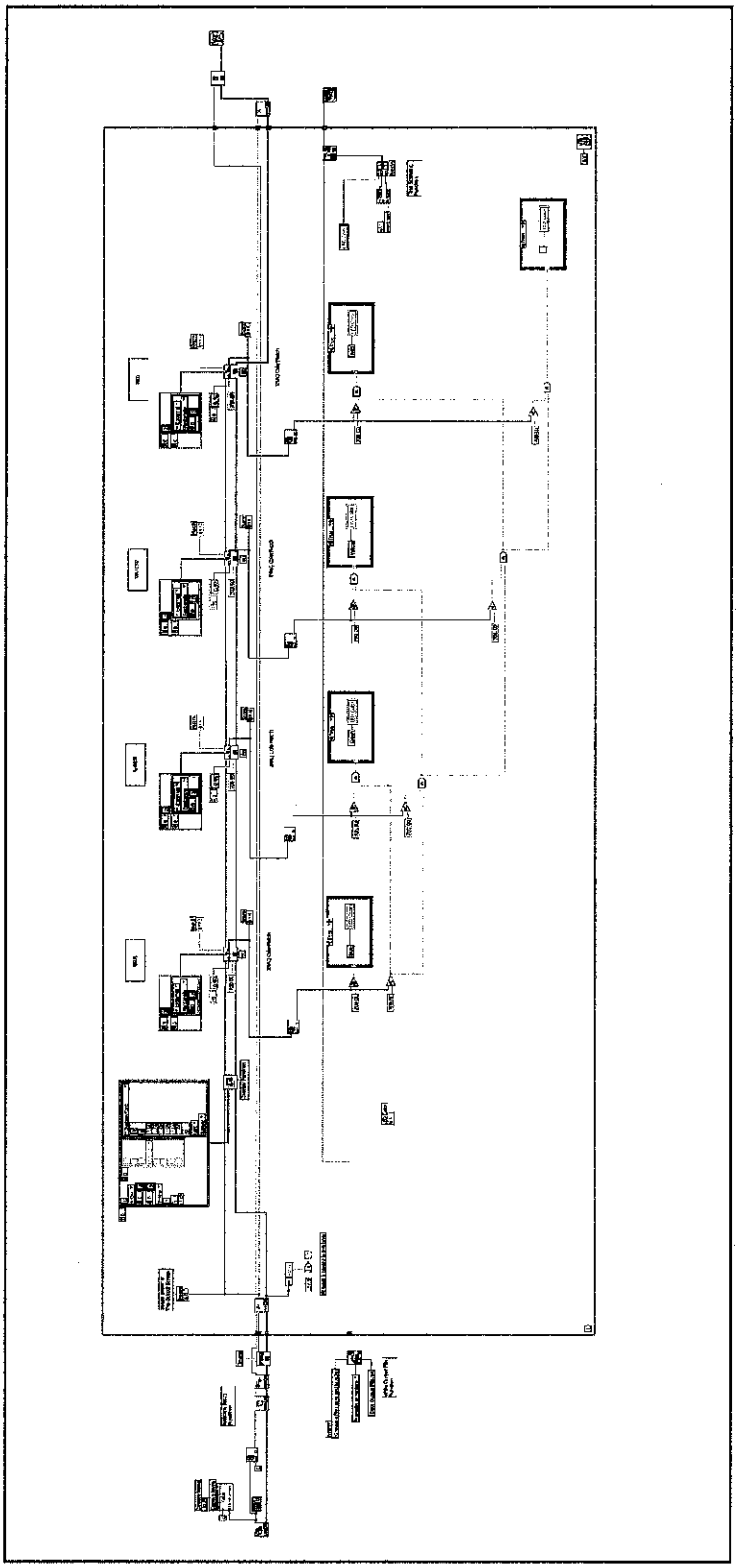


Figure 4.6: LabVIEW Block Diagram

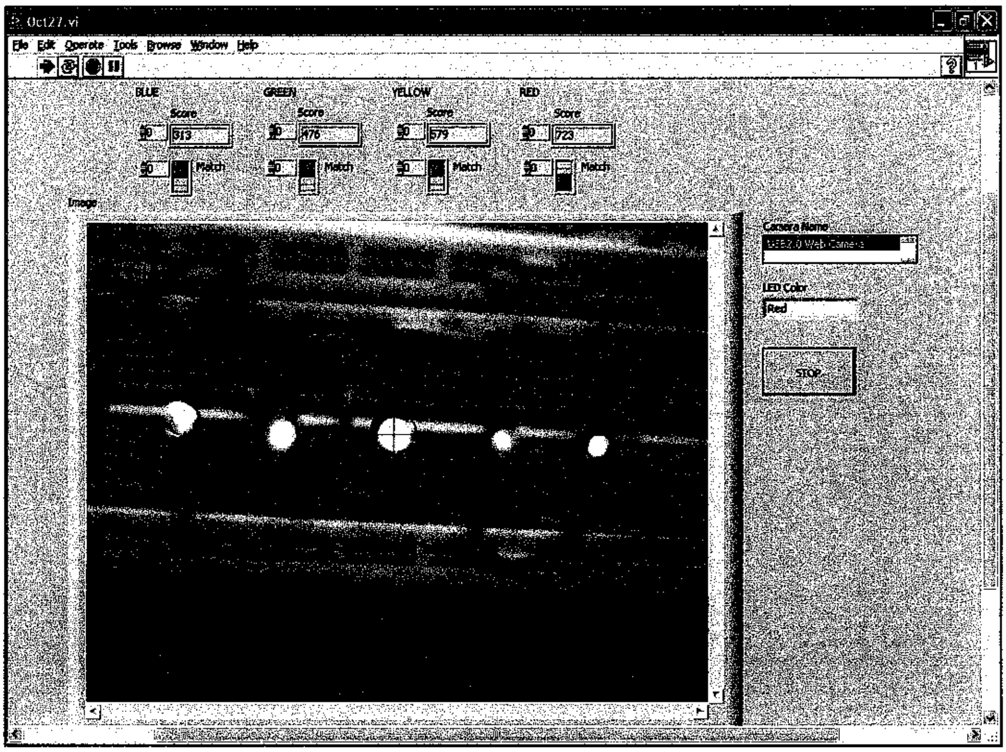


Figure 4.7: The system's front panel

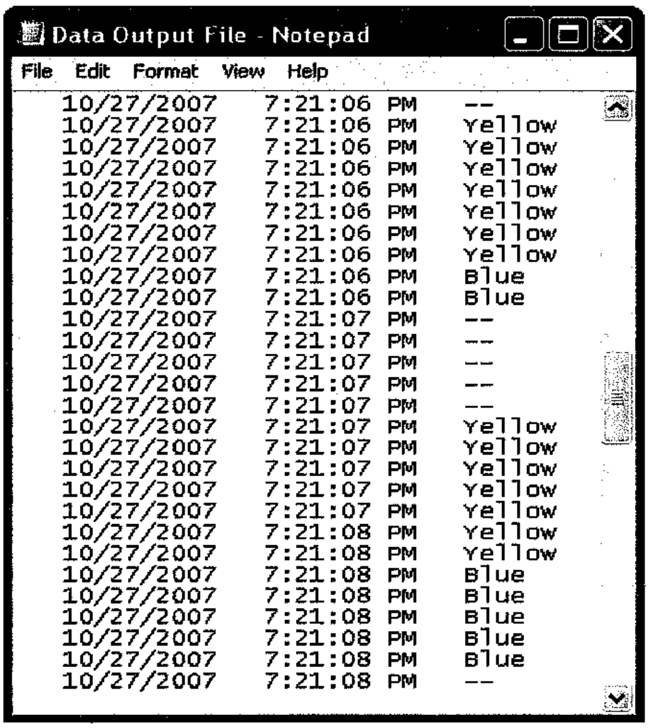

Data Output File - Notepad
File Edit Format View Help
10/27/2007 7:21:06 PM --
10/27/2007 7:21:06 PM Yellow
10/27/2007 7:21:06 PM Yellow
10/27/2007 7:21:06 PM Yellow
10/27/2007 7:21:06 PM Yellow
10/27/2007 7:21:06 PM Yellow
10/27/2007 7:21:06 PM Yellow
10/27/2007 7:21:06 PM Yellow
10/27/2007 7:21:06 PM Blue
10/27/2007 7:21:06 PM Blue
10/27/2007 7:21:07 PM --
10/27/2007 7:21:07 PM --
10/27/2007 7:21:07 PM --
10/27/2007 7:21:07 PM --
10/27/2007 7:21:07 PM --
10/27/2007 7:21:07 PM Yellow
10/27/2007 7:21:07 PM Yellow
10/27/2007 7:21:07 PM Yellow
10/27/2007 7:21:07 PM Yellow
10/27/2007 7:21:07 PM Yellow
10/27/2007 7:21:08 PM Yellow
10/27/2007 7:21:08 PM Yellow
10/27/2007 7:21:08 PM Blue
10/27/2007 7:21:08 PM Blue
10/27/2007 7:21:08 PM Blue
10/27/2007 7:21:08 PM Blue
10/27/2007 7:21:08 PM Blue
10/27/2007 7:21:08 PM --

Figure 4.8: Sample of data output file

The whole process of creating the prototype of Visual Automation Tool can be summarized using the below flow chart:

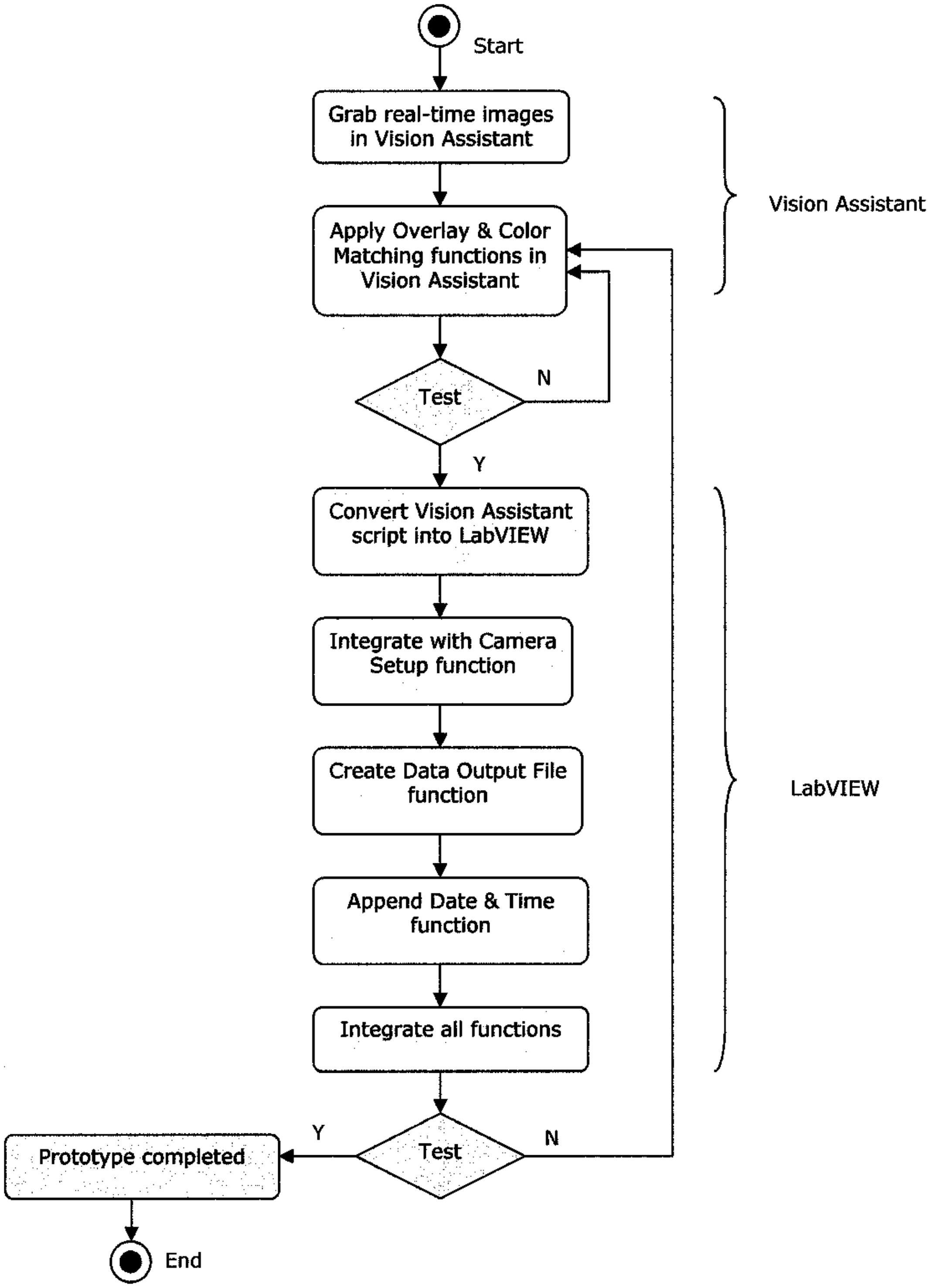


Figure 4.9: Flow Chart of Visual Automation Tool prototype

4.1.3. Image processing functions

The result of this project is going to be an executable form of LabVIEW program that is able to capture the real-time images of the server board, particularly focusing on the LED. The images are then going to be analyzed using some image processing functions such as color detection function and binary image converting function for blink frequency calculation.

Table 4.1 below shows some of the *global color processing functions* that are going to be used in the Visual Automation Tool:

Table 4.1: Global Image Processing Functions

Function Name	Purpose(s)
Image Overlay	Acts as a 'stencil' to guide the user to exactly place the LED inside the ROI boundaries. Each LED will have its own overlay according to its location on the board.
ROI (region of interest)	Specifies the area of the captured images in which needs to be compared with the pre-created LED image templates. The position of ROI is going to be static.

- *Color processing function*

The accuracy of the color detection result is solely depends on the **Color Matching** function (table 4.2). There are many color detecting methods that are suitable to be implied in this project. One of those is using the ***Euclidean Distance*** that helps widen the score range under each particular color group. The principle of *Euclidean Distance* is relevant to be applied because the score does not have to be the exact value as the real color template. With this, the flexibility for the result value is increased. The unknown color will be grouped under the known color that has the closest value to the unknown colors value.

Table 4.2: Color Processing Function

Function Name	Purpose(s)
Color Matching	Compares the LED image templates with the ROI of the captured images. This function will return the matching score of the LED ranges from 0 for non-matching (e.g. blue-red) and 1000 for perfectly matched (e.g. blue-blue).

Another method is by using other assisting application such as *ColorDetector* and *ColorPicker12* tools to get the RGB score of each LED color. *ColorDetector* tool will print out the values for each red, green and blue combination of colors like the example shown in figure 4.10 while in *ColorPicker12*, color values are represented in Delphi, C++ and HTML codes, as shown in figure 4.11.

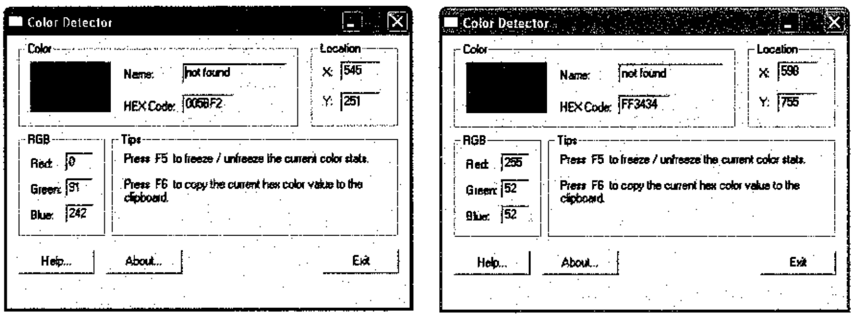


Figure 4.10: ColorDetector Tool with blue (left) and red (right) inputs

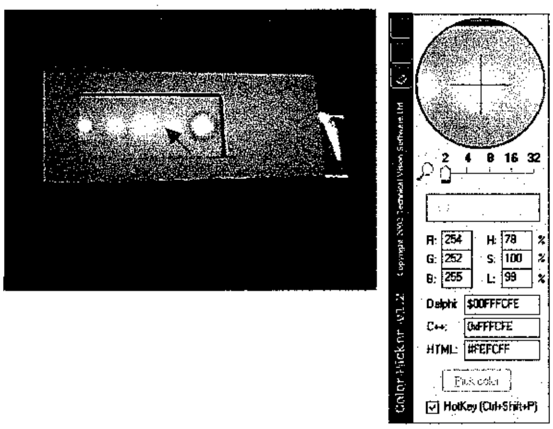


Figure 4.11: ColorPicker12 Tool with input of red LED

- *Blink frequency functions*

For blink frequency function, we need to convert the image to binary first because binary images are images whose pixels have only two possible intensity values. Images are normally displayed as black and white. Binary images are produced by thresholding a grayscale or color image, in order to separate an object in the image from the background. The color of the object (usually white) is referred to as the *foreground color*. The rest (usually black) is referred to as the background color. Figure 4.12 below shows the blue LED images being converted into binary. In this case, the binary image is formatted to be red for the object and black for the background.

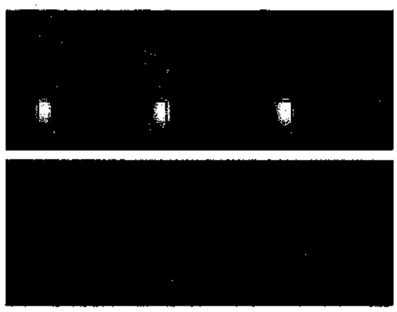


Figure 4.12: Original image (top) and binary image (bottom)

We only need to focus on needed image (LED only) and trace out the background to count the frequency of the blinking LED. If the LED is lights up, score is 1 and 0 when it doesn't light up. The frequency is calculated by counting the time interval between the change from 0 to 1 and from 1 to 0 again. Table 4.3 below shows the blink frequency functions that are going to be implemented in counting the blink frequency of the LED.

Table 4.3: Blink Frequency Functions

Function Name	Purpose (s)
Binary-Image Conversion	The original image is in RGB type. In order to calculate the blink frequency, image needs to be converted into binary format since that function can only support binary images.
Particle Analysis	Filters the unnecessary 'noise' for blink frequency function. It will produce a solid dot (the binary formatted image) and make the frequency calculation more accurate.

4.1.4. Testing the Result's Accuracy

The main concern for this project is all about the accuracy. The series of tests must be conducted to increase the result's accuracy with regards to the inaccuracy factors such as the position of the object and the presence of the ambient light. The test usually is all about observing the accuracy of the results. The methods are: (1) Altering color detection functions repetitively - increase or decrease the score range for each color; (2) Choosing the suitable cover to eliminate ambient light from effecting the color of LED until the consistent results are obtained; and (3) Reposition the camera until the best distance is acquired. These methods will be executed repeatedly until the accuracy of results is achieved. Besides, the function that counts the blink frequency rate also needs to be tested out. The steps are: (1) Ensuring the binary image conversion function has filtered out all unnecessary background; (2) Ensuring LED is correctly placed inside the ROI (region of interest); and (3) The blink rate output from the system is tally with the output obtained manually.

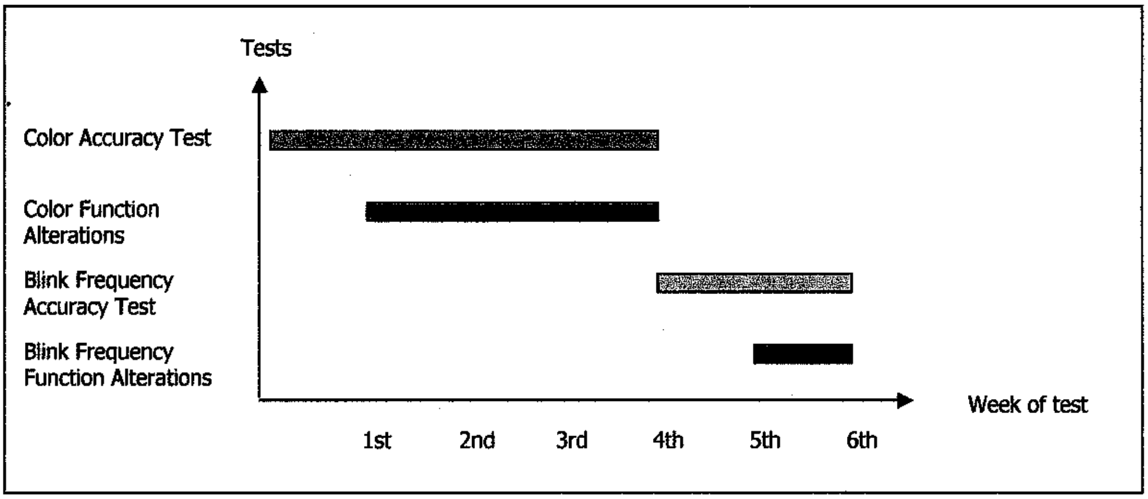


Figure 4.13: The estimation of the 'Accuracy Test' period

4.2. Discussions

4.2.1. Applications of Color Detection & Measurement

Color detection or measurement is widely used in many manufacturing industries to ensure the color consistency for products like paints, metal, glasses, textiles and automobiles. Other than that, in food industries, it can be applied for color detection on powders, granules, liquids etc. All of these products usually will be tested according to significant color features.

Given the diversity of specific applications, there are many types of color measurement instruments available. They are ranging from the simple to a more complex usage. For complex application such as skin color detection etc, the specific-usage and more expensive devices will be used e.g. color sensors (please refer to **appendix no.4**). Whereas for the less demanding applications, instruments built on low-cost RGB detectors are often sufficient e.g. visual automation tool that utilizes a webcam, controlled by LabVIEW program.

4.2.2. Concerns and Limitations

In order to get the best result that is accurate in time-wise, the camera should be able to grab the image at the optimum rate of 30 frames per second (fps). The situations that could possibly hinder this requirement are system's hardware (e.g. low RAM capability) or the web camera itself (e.g. very low resolution and inefficient grabbing speed, max is only 15fps). If the frame rate is failed to be kept at the optimum rate, the result of color reading at any particular time would be incorrect, for example, at the 30^{th} second, it actually reads the color at the 60^{th} second because of the delay.

The other concern is the presence of ambient light (like being discussed under section 2.3-Color Detection Pitfalls). This system is supposed to use IEEE standard firewire camera that has higher resolution and reads color very sharply regardless the presence of ambient lighting or inconsistent positioning. For the prototype purpose, the system is now using a webcam that has less capability compared to this more powerful camera. Excessive or insufficient presence of light will simply alter the color reading of the webcam. Another limitation is: the camera needs to be static while reading the LED color because if there is any glitch on the position, the camera reads incorrectly or even worse, cannot read the color at all.

CHAPTER 5
CONCLUSION & RECOMMENDATION

5.1. Conclusion

Visual Automation Tool project is carried out in order to aid the hardware validation team in conducting validation processes for the server board. The main goal is to address the problems of accuracy and resolve the need to execute the tests manually. To solve these problems, an automated tool is developed to automate the visual testing for the server since it is capable of comparing LED colors and monitoring the blink frequency.

With this solution, the accuracy of the test result is increased since the accuracy does not depend on the tester's performance anymore. The tests can now be controlled automatically and the tester does not have to sit in front of the system to do the testing. Furthermore, remote overnight testing can also be performed and it will bring a lot of benefits in terms of automation, flexibility and shortening the testing time.

Apart from that, the concept of color detection used in Visual Automation Tool can also be used for other purpose for example in the manufacturing field, the quality checking on electronic board can be done by detecting whether the electronic components have been assembled correctly or not based on their colors and the locations.

5.2. Recommendations

For future enhancements, it is recommended if this system could be altered to make it more flexible; not receiving only LED color as an input but capable of receiving any form of input as long as it is related to color comparison and detection.

It is also suggested that the system could read the output file data to calculate the blink frequency, which is believe to be easier compared to applying the method as being discussed in page 25.

Besides, this system would also be better if it can have more interactions with the end users instead of hard-coding elements such as region of interest (ROI), color template etc. It means that, users can select their own ROI according to the area of the sample images that they prefer to test out as well as specify their own template to be compared with the sample images. With these solutions, the system will have more flexibility attributes and the usage of it can be widened.

References

Murray, K., Holladay, S. (2003). Object Color Detection – Intelligent Robot Navigation. PennState University. 1-6.

Trinh, V., Schwartz, E., Arroyo, Antonio., Nechyba.M.(2003). *A Color Detection Method for Introductory Robotics.* 2003 Florida Conference on recent Advances in Robotics. University of Florida, 1-2.

Trinh, V. (2002). EM- The M&M Sorting Robot. [Online]. 2002 Intelligent Machine Design Laboratory Final Paper. University of Florida. 4-6.

Cadmium-Sulfide Color Sensor. (2000). *Robot Room* by David Cook. Retrieved March 20, 2007, from http://www.robotroom.com/ColorSensor.html

Color Sensor. *Philohome* by Philippe Hurbain. Retrieved March 20, 2007, from http://www.philohome.com/sensors/colorsensor.htm

Appendices

Appendix 1: Milestone of the FYP Project Works

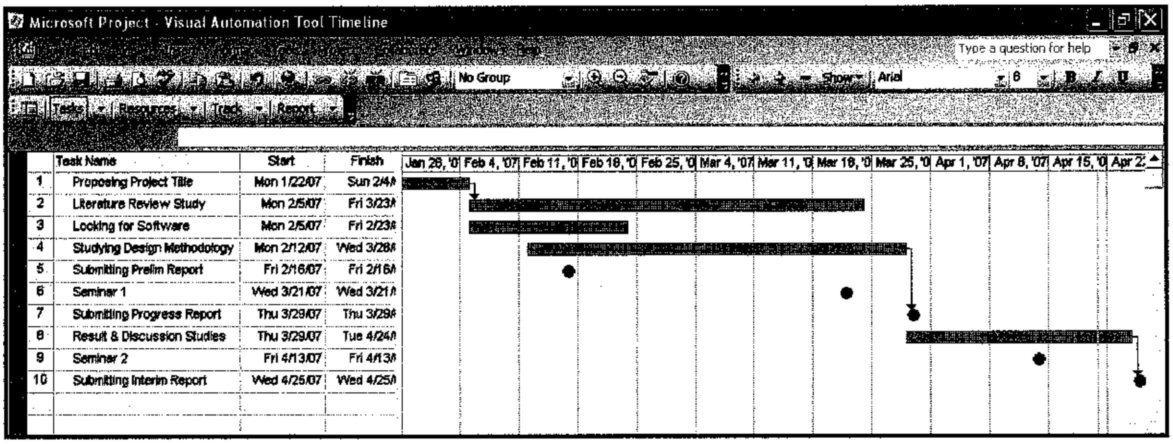


Figure 6.1: Milestone of the FYP project works (first semester)

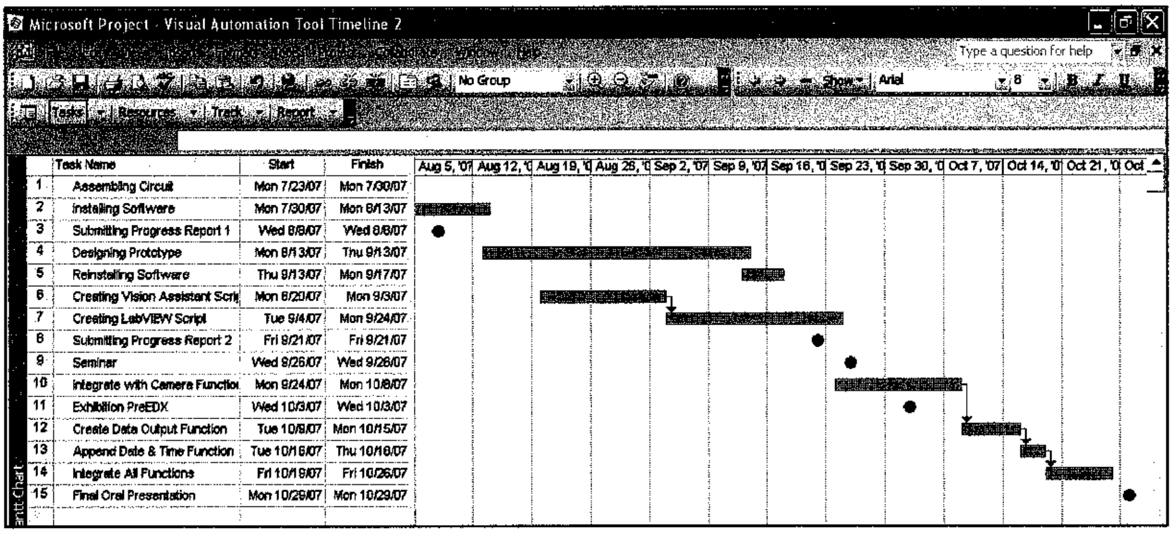


Figure 6.2: Milestone of the FYP project works (second semester)

Appendix 2: Sample illustrations of the system

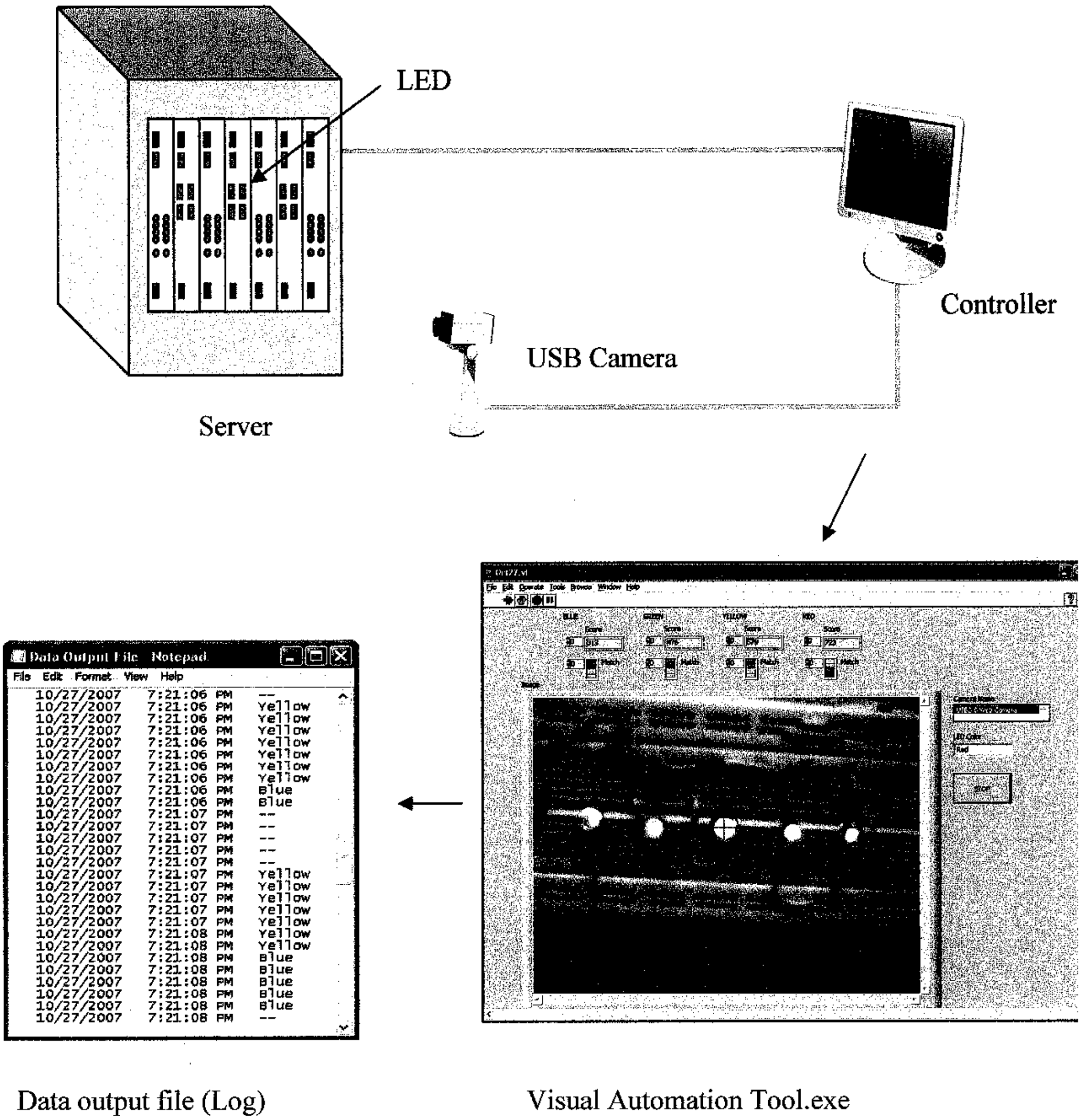


Figure 6.3 Sample of Visual Automation Tool

Appendix 3: *Euclidean Distance* Example Calculation

In this operation, ONE *unknown color* surface is going to be tested. We are using 4 types of LED colors (RGB and white) that are going to be reflected on that color surface. The results of these 4 readings are recorded. The same tasks are performed on the other specified number of *known colors* (for this example, I am using Yellow, Red, Blue, Green and Orange). The readings obtained are then being recorded in the table below.

Table 6.1: Values of known color (0-255)

Color to detect	LED 1 (green)	LED 2 (white)	LED 3 (blue)	LED 4 (red)
Yellow	70	178	121	191
Red	3	80	83	160
Blue	12	90	120	60
Green	80	150	131	100
Orange	8	135	100	186

Unknown Color Reading 1 = 65

Unknown Color Reading 2 = 130

Unknown Color Reading 3 = 150

Unknown Color Reading 4 = 90

Below is the *Euclidean Distance* formula that is going to be used in the calculation:

$$D_{Euclidean} = \sqrt{\sum_{n=1}^{m} (X_n - Y_n)^2}$$

D = Euclidean Distance
X = Unknown Vector
Y = Known Vector
M = Number of test points per vector

Calculations:

$D_{yellow} = \sqrt{[\,(65-70)^2 + (130-178)^2 + (150-121)^2 + (90-191)^2\,]}$

$D_{yellow} = 115.633$

$D_{red} = \sqrt{[\,(65-3)^2 + (130-80)^2 + (150-83)^2 + (90-160)^2\,]}$

$D_{red} = 125.431$

$D_{blue} = \sqrt{[(65-12)^2 + (130-90)^2 + (150-120)^2 + (90-60)^2]}$

$D_{blue} = 78.797$

$D_{green} = \sqrt{[(65-80)^2 + (130-150)^2 + (150-131)^2 + (90-100)^2]}$

$D_{green} = 32.955$

$D_{orange} = \sqrt{[(65-8)^2 + (130-135)^2 + (150-100)^2 + (90-186)^2]}$

$D_{orange} = 122.434$

According to the rules, the color that yields the smallest value is most likely to be the same color as the unknown color surface. In the above calculations, GREEN is the most likely color with a Euclidean distance of 32 (smallest value).

Therefore, we can conclude that the unknown color surface is GREEN.

Appendix 4: More Color Detection Application and Devices

HAMAMATSU PHOTON IS OUR BUSINESS

My Account

Corporate | Events | Support | Contact us | Search...

Divisions | New Products | Product Types | Applications

Applications
Application Notes
Technical Notes
Learning Center
Glossary

Analytical
Capillary electrophoresis
Lab-on-a-chip
Food quality
Atomic absorption
Atomic emission spectroscopy
Color detection
Automotive
Biotechnology
Homeland Security
Industrial/Manufacturing
Medical Diagnostics
Optical Communications

Color detection

Color measurement is widely used in many manufacturing industries to ensure the color consistency of products such as paints, textiles, and automobiles. It is also very important in the food industry, as a number of quality control processes depend on reliable detection of the colors of fresh or processed foods and ingredients.

Given the diversity of specific applications, there are many types of color measurement instruments available. For complex applications that depend on detailed spectral information, spectrometer-type devices are used. One example of this is a Hamamatsu mini-spectrometer, which combines a linear image sensor, quartz transmission grating, and other optical components within a portable package. For less demanding applications, instruments built on low-cost RGB detectors are often sufficient.

An RGB detector is basically a multi-element photodiode coupled to red, green, and blue filters. These filters enable the photodiode to generate separate response curves for the three colors. Due to improvements in silicon processing and other fabrication techniques, it is possible to manufacture RGB sensors with finer pitch and smaller optical packaging. More components can also be integrated on the chip, which helps reduce the effects of interconnection noise, parasitic capacitance and leakage current, and as a result leads to better detector performance.

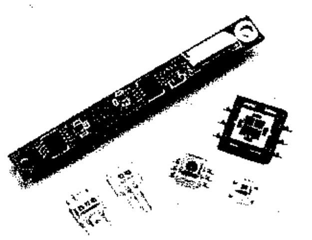
Hamamatsu offers a variety of color detectors, and can also customize other types of components for specific color sensing requirements. For example, CMOS linear sensors can be customized with color filters, and second-order filters can be mounted on photodiode arrays. Mini-spectrometers can also be adjusted to suit certain applications.

Recommended products

- **RGB color sensors and modules:** ideal for backlight monitoring of RGB-LED TFT-LCD screens and other applications.
- **S9706 digital color sensor:** detected RGB signals are serially output as 12-bit digital data.
- **C10083MD:** mini-spectrometer for industrial color measurement.
- **R3896 28 mm side-on PMT:** spectral response from 185 nm to 900 nm. High-sensitivity multialkali photocathode.
- **R928 28 mm side-on PMT:** spectral response from 185 nm to 900 nm. Extended red sensitivity.

Please **contact us** for information on other products which can be used for this application.

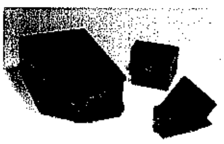
Color Sensors



We offer a variety of sensors and modules for LCD color monitoring and simple color detection. These photosensors are color sensors using a 3-element (or 2-element) photodiode with color sensitivity assembled in one package. Modules for TFT-LCD monitor and evaluation circuits are also provided.

Mini-Spectrometers

Hamamatsu mini-spectrometers cover a wide spectral response range and also provide:



- High-precision measurements
- High throughput
- Ease of use
- Flexibility in measurement setups
- Compact size
- USB connectivity
- Bundled software package (DLL, device driver, and sample software)

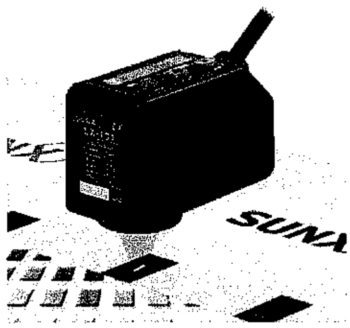
SPEC
The Engineering Search Engine

Your Stuff | Your Profile
Download Engineering Toolbar
GlobalSpec Home

Find: [ ] Search Advanced Search >>

You are here:
GlobalSpec.com > Company Directory > Panasonic Electric Works > LX-100: Digital Mark and Color Detection Sensor

LX-100: Digital Mark and Color Detection Sensor



The LX-100 is the first sensor offering Mark mode and Color mode detection with a 4-digit digital display that enables numerical sensing control and minute settings. The Mark mode automatically selects a single color from the 3 R-G-B LED's to realize an ultra quick 45 micro sec response time. The automatic optimal LED selection function determines the most suitable LED for the largest contrast (S/N ratio) between the mark and base (non-mark area) to ensure optimal sensing. For more stable detection, the sensor makes selections according to the contrast and not according to the reflected light variation between the mark and the base. The Color mode on the LX-100 series utilizes all 3 R-G-B LED's to determine the RGB ratio of the mark sensor. The built in 12-bit A/D converter enables high precision 1/4000-resolution judgements. This function enables effective detection of films with patterns around the area of the mark. The LX-100 comes in a compact design W57 xD24 x H38mm (W2.244 x D0.945 x H1.496in) offering a cable or M12 plug-in type connector. Six indicator lamps using MODE NAVI, a highly praised user interface originally developed for the FX-300 series, represent the sensor's basic operation. The User can check what mode the sensor is presently in with a glance rendering operation simple. With fully automatic, 1 or 2-level calibrations, teaching is as easy as pushing a button. External teachings are also possible by using an operation panel or touch screen for color mark sensors whose position with in the equipment is out of reach. Thus, the LX series is particularly ideal in the pharmaceutical and food packaging industries. It can also be used for various types of packing films such as transparent and aluminum evaporation film.

SUNX is a leading brand of industrial sensors of Matsushita/Panasonic Group. Sold through distribution network of Panasonic Electric Works Corp. of America (formerly Aromat Corporation); a US subsidiary of Matsushita Electric Works (Osaka, Japan) for Americas.

[For additional information] Sensor Tech Support: 1.877.624.7872

Supplier:
Panasonic Electric Works

For More Information

Request a Quote
for this product

Send an E-mail
to the company

See Product Details
on company Web site

Browse Catalog
for this company

See Product Announcements
from this company

PEPPERL+FUCHS
SENSING YOUR NEEDS
Home

Visit Our Global Site
Contact Pepperl+Fuchs: 1.330.425.3555

About Pepperl+Fuchs | Contact Us

Model/ Keyword Search

« Previous Page :: Next Page » You Are Here > Markets & Applications > Printing &

SEARCH

MARKETS & APPLICATIONS

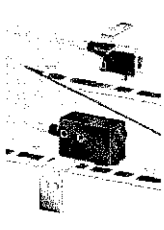
jump to MARKETS & APPLICATIONS GO

jump to PROCESS AUTOMATION GO

jump to FACTORY AUTOMATION GO

Web Printing

Color Detection



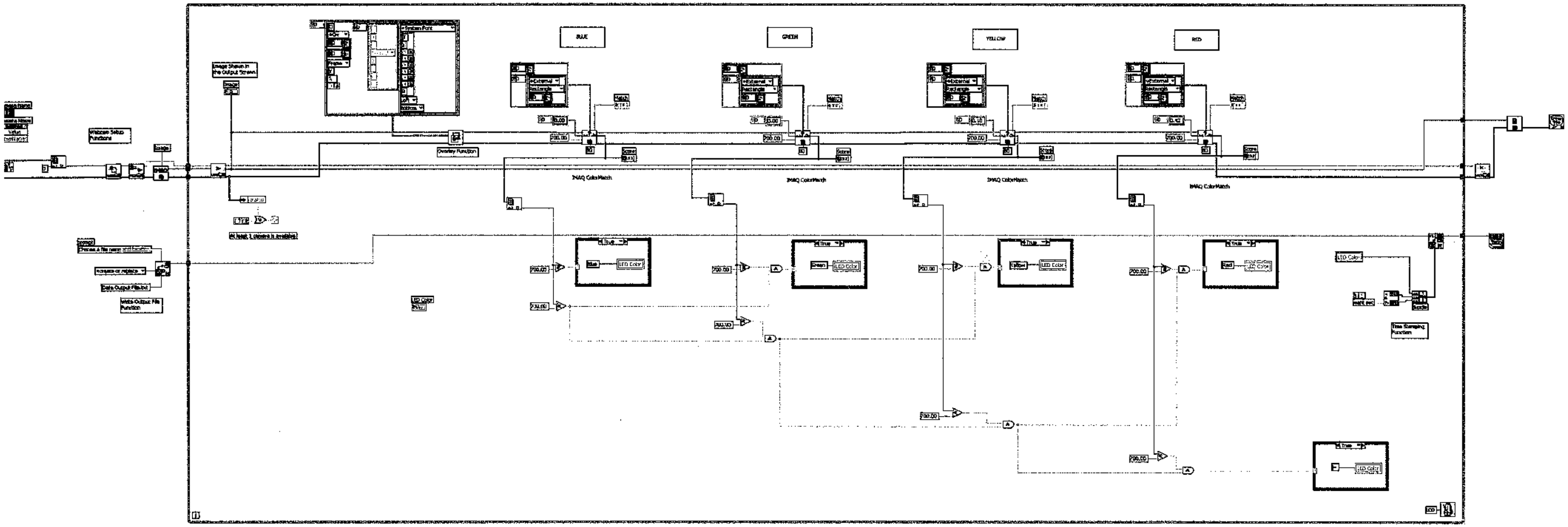
- Identify color variations
- Up to ten independent color channels on one sensor
- Fast push-button programming
- Read smaller print marks with less contrast

Related Products

Photoelectric Sensors Color

Photoelectric Sensors Contrast

Quick Disconnect Cordsets & Receptacles



Appendix 5: LabVIEW Block Diagram