Acknowledgements

I would like to thank my supervisor Professor Paul F. Whelan for his guidance, enthusiasm and commitment to this project. I would like to thank Ciaran Lane and the Mechanical Engineering team in Alps for their understanding, support and manufacture of the mechanical element of this project. I would also like to thank my fellow peers, family and friends for their support during my college years. I would like to express my deep appreciation to Mr James Sugrue, Senior Engineering Manager of Alps, and Mr Dominic Raftery, Senior Test Engineer for their continuing support of this project. Finally I would like to thank Alps as a company for affording me the opportunity to take on this challenge, and placing trust in my ability to develop this project.

Declaration

I hereby declare that, except where otherwise indicated, this document is entirely my own work and has not been submitted in whole or in part to any other university.

Signed: .... Date: ....

Abstract

As the complexity of automotive products increases, the requirement for higher caliber testing grows. This requirement is driven internally within Alps from management, the quality and engineering departments, and also externally from the customer. Using machine vision systems to inspect products visually is the growing standard in manufacturing processes. It will improve detection of cosmetic/ graphic defects beyond the capability of operator inspection.

The report describes the design and implementation of a machine vision system for the Volvo Y20 climate control module (CCM). This module is used to control the climate and infotainment in the Volvo v70, xc60 and s80 vehicles. The ongoing idea is to use this system as a benchmark to inspect many more of Alps products.

The system comprises the features of a machine vision system. From a software viewpoint, a fully functional graphical user interface (GUI) is designed, along with the necessary algorithms required to inspect for the many potential defects. Other software features include full inspection logging and inspection setup capability. The full hardware solution is also designed and implemented, in conjunction with Alps mechanical engineering division.

This system will guarantee correct assembly and visual appearance of the Y20 CCM module prior to shipping to customer. Not only does this improve the overall testing of the module, it will also further enhance Alps quality reputation with the customer.

List of Figures

List of Tables

Chapter 1: Introduction

The first chapter gives an introduction to the dissertation. It starts by defining the problem that is to be solved, and also outlines the exact aim of the project. Finally, a description of the content of the thesis is presented.

1.1: Problem Definition

In the ever changing world of automotive electronics, Alps Ireland is always looking to improve their testing ability. With technological advances and increased consumer requirement, more complex products are required. One of those products, the Y20 Climate Control Module (CCM), as shown in figure 1.1.1 below, is currently in mass production in the Alps facility in Millstreet, Co. Cork.

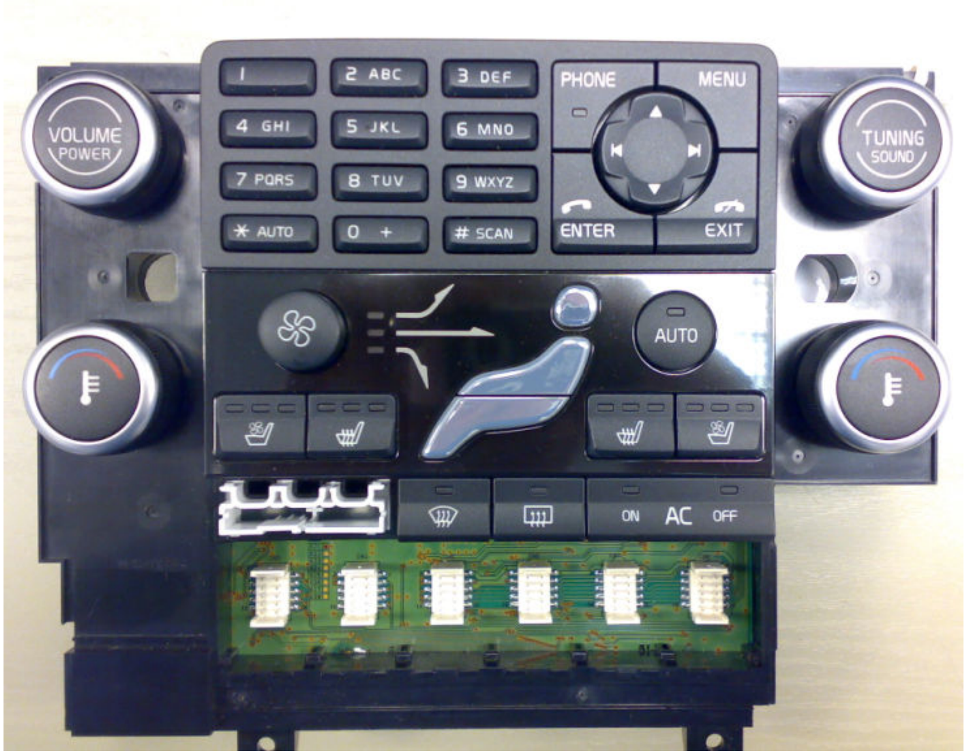


Figure 1.1.1: Y20 Climate Control Module

Mis-assembly of buttons and laser etching mistakes can potentially cost > €100,000 each year and this would affect Alps quality reputation with the customer. This is mainly due to lack of poka-yoke in the design process. A poka-yoke is any mechanism in a Lean manufacturing process that helps an equipment operator avoid (yokeru) mistakes (poka). Its purpose is to eliminate product defects by preventing, correcting, or drawing attention to human errors as they occur [2]. The production operator is relied upon to visually inspect this part. It is thought that the installation of a machine vision system will improve quality levels and improve the process. Industry standards show that the cost of a customer return to any organisation is quite significant. It is necessary to optimise this inspection to prevent claims from the customer.

1.2: Project Aim

The aim of the project is to develop a complete Machine Vision solution system primarily to inspect for defects on the Y20 CCM module. It is also an aim to expand the use of this system across other products in Alps. The design of this system includes full development of the hardware and software solutions, including the graphical user interface.

1.3: Thesis Content

This report is broken down into the following chapters.

Chapter 1: The first chapter provides an introduction to the dissertation. It starts by defining the problem that this system aims to solve. It then outlines the aim of the project, followed by a breakdown of the thesis content. It finally concludes the first chapter.

Chapter 2: The second chapter outlines some background information on the project. It starts with a general introduction to the chapter. It then informs us of the specification of the product, such as cosmetic layout, potential failure analysis and the electrical specification. It then defines the project requirement from Alps as a company. This is broken down into the three main areas of requirement – Engineering, Quality and Production. Finally the project preparation is outlined, covering topics such as the budget allocated to the project, and the knowledge that is needed to lead to a more successful implementation.

Chapter 3: The third chapter describes the system design of the system. It starts by defining the hardware design covering typical elements such the camera choice, optics, lighting and jig design. It then discusses the software design, informing which software is used and why, the graphical user interface (GUI) design and other software elements. Finally the seven algorithms that are used for the analysis are explained in detail.

Chapter 4: The fourth chapter describes the system implementation of the project. It starts by describing the implementation stage of the hardware, showing the final system diagram and jig implementation. It then outlines the problems encountered. Finally the software implementation is presented. This discusses the key elements of the software such as using the camera, the final GUI, inspection setup along with other software elements.

Chapter 5: The fifth chapter outlines the extensive testing and analysis of the system. It starts by describing the detection capability verification, where we ascertain what sort of failure detection we have with the system. The reliability testing of the system is then discussed, where we determine how reliable the system is in the production environment.

Chapter 6: The sixth chapter concludes the project by giving the conclusions of the project and describing the future work that is proposed.

1.4: Conclusion

The first chapter gives an introduction to the dissertation. It starts by defining the problem, which is that potential mix-up of assembly of button or other errors could potentially cost Alps as a company thousands of Euro in quality claims. It is proposed to develop a machine vision system capable to inspecting for these potential failures. The aim of the project is established. Finally, the content of the dissertation is outlined on a chapter by chapter basis.

The next chapter gives some background information on the project. It starts by describing of the specification of the product, such as cosmetic layout, potential failure analysis and the electrical specification. It then defines the project requirement from Alps as a company. Finally the project preparation is discussed, covering topics such as the budget allocated to the project, and the knowledge that is needed to lead to a more successful implementation.

Chapter 2: Background

2.1: Introduction

The second chapter outlines some background information on the project. It starts by describing of the specification of the product, such as cosmetic layout, potential failure analysis and the electrical specification. It then defines the project requirement from Alps as a company. Finally the project preparation is discussed, covering topics such as the budget allocated to the project, and the knowledge that will be attained to lead to a more successful implementation.

2.2: Product Specification

2.2.1: Cosmetic Layout

Figure 2.2.1 below shows the cosmetic layout of the module. This is an example of one model type, and there are fourteen model types in total, with each having its own set of button combinations. These various models are outlined in appendix G. A model matrix is used to control which inspections are implemented for each model.

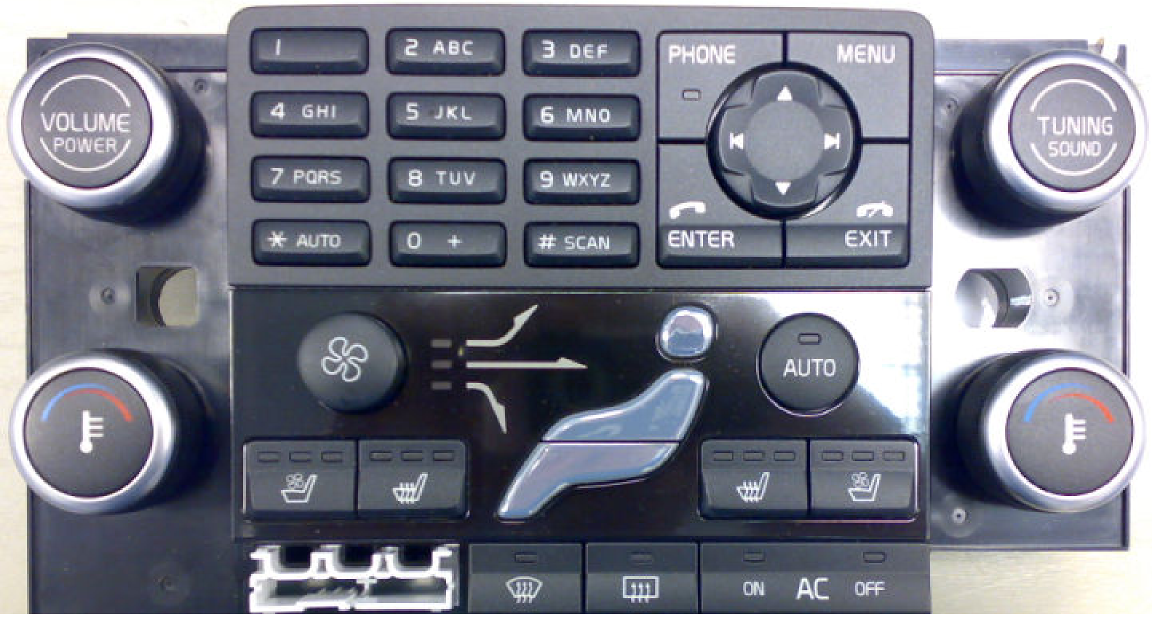


Figure 2.2.1: Volvo Y20 CCM Cosmetic Layout

2.2.2: Potential Failure Analysis

Lack of Poka-Yoke

The two major potential failures are due to lack of poka-yoke on two key button areas. The first, the keypad is shown in figure 2.2.2 below. All of the number buttons can be interchanged, which leads to a massive problem from a quality viewpoint. This means there is potentially a failure count of 144 in this region alone. Also in figure 2.2.3, these two buttons can be switched. Due to the similar nature of the graphics, this is seen as a priority also.

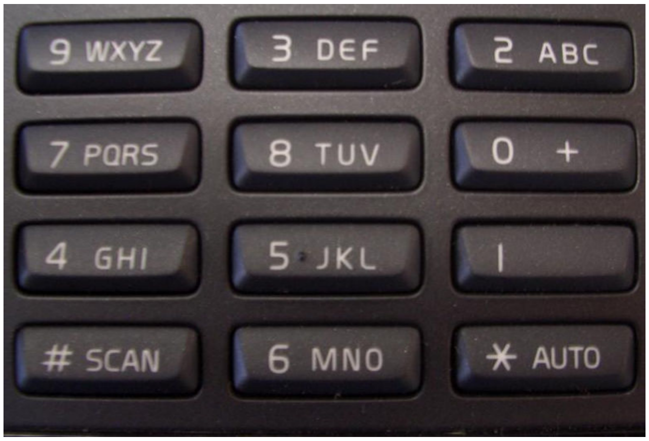


Figure 2.2.2: Mixed up buttons on Keypad

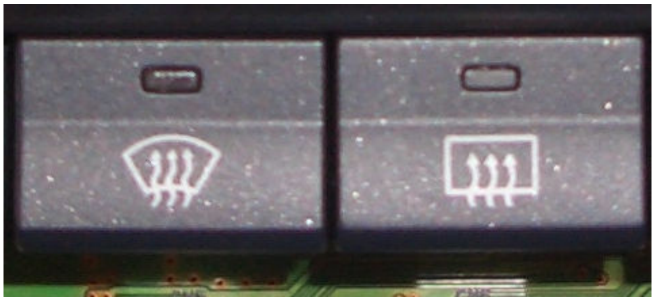


Figure 2.2.3: De-Fog buttons that can be interchanged

Different Model Buttons

Depending on which one of the fourteen varieties of this module is being tested, there are buttons that vary not in shape or poka-yoke, but in appearance. The only difference is the graphic laser etched on the surface. Figure 2.2.4 below shows two such examples. It is necessary to develop methods to inspect for these assembly errors.

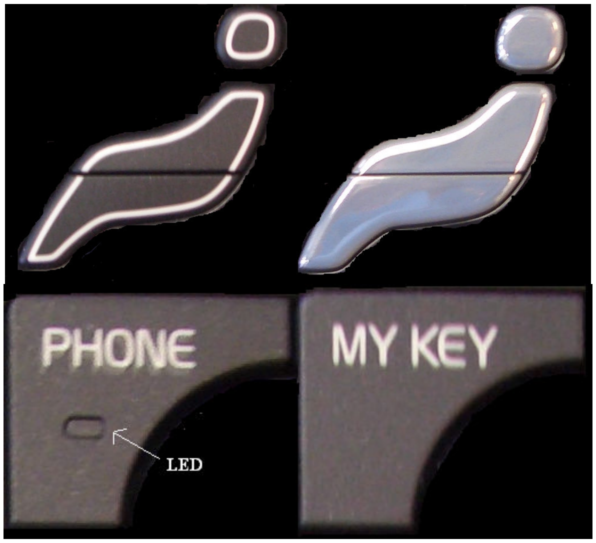


Figure 2.2.4: Example of different model buttons

Incorrect Laser Etching

All buttons shown on this product are manufactured on site in Alps production facility. They are molded first in the molding department, then painted and the graphics laser etched in the paint and laser division (PAL). There are a number of fail-safe's in place in this division to ensure correct laser etching, but there is still a lack of a machine vision inspection system in PAL. Although unlikely, it is still possible for incorrect graphics to be laser etched on the buttons. This is another failure that this system should be able to detect. The system should be tolerable to a level of movement in the position of the graphics, but only within the quality specification. The quality specification for this product is outlined in appendix J.

LED Colour

In the printed circuit board (PCB) manufacturing stage for this product, the LEDs are mounted using surface mount technology (SMT) methods. This is a very reliable process, with numerous fail safe systems in place. However, we have still seen incorrect LED's being mounted. There is a possibility during PCB repair that a wrong LED can be mounted. There are two possibilities of LED colour, green and amber. It is not necessary to inspect for quality of colour, but detecting very dull LED's is a quality requirement.

2.2.3: Electrical Specification

The product is rated at 13 volts, and draws a current of 200 mAmps. It is controlled using the Controller Area Network topology (CAN). CAN is a vehicle bus standard designed to allow microcontrollers and devices to communicate with each other within a vehicle without a host computer. It is designed specifically for automotive applications [3]. This CAN topology is how the button backlighting is controlled.

2.3: Project Requirement from Alps

The requirements of the system need to be determined, from a company viewpoint. To establish this, a number of discussions need to take place with various members of key departments directly affected by the system. These are Engineering, Quality and Production.

2.3.1: Engineering Requirement

The engineering requirement is to ensure that:-

- developed on time, and within the budget,
- the system offers a viable solution for future similar product applications,
- the system shows high detection rates for failures,
- the system records very little downtime i.e. high reliablilty,
- the system is easy to use for Engineers/Technicians.

2.3.2: Quality Requirement

The quality requirement is to ensure that:-

- the system has high visibility, i.e. all units can be traced to have completed the test, and what measurements were logged,
- the systems follows the quality specification for the product,
- the system is a guaranteed process i.e. all units are guaranteed to complete the test,
- the system is foolproof, that inspections cannot be set up incorrectly,
- if a bad unit does get through the system, this exposure is limited to as brief a time as possible. The aim is to have an exposure time of 1 day,
- the system has very good detection capability.

2.3.3: Production Requirement

The production requirement is to ensure that:-

- the system causes very little downtime i.e. high reliablilty,
- the system offers a viable solution for future similar product applications,
- the system shows high detection rates for failures,
- the system is easy to use for operators,
- the system does not interfere with current production cell, and that it does not become the gating element of the final assembly production cell. The gating item is the slowest stage of the production. Currently the gating item in the production cell is the end of line (EOL) electrical test, which has a cycle time of seventy seconds. The cycle time of this vision system must be less than seventy seconds.

2.4: Initial Attempt

When in the discussion stage of the project, it is proposed to keep the system as low cost as possible. A four camera system is initially proposed as shown in figure 2.4.1 below. This system uses a single framegrabber and four low resolution PCB board cameras. Each of these cameras costs a mere €200. The video signal is switched between the four cameras to the video grabber in sequence. One of these cameras alone did not have sufficient resolution to cover the entire product field of view. Thus four cameras are used.

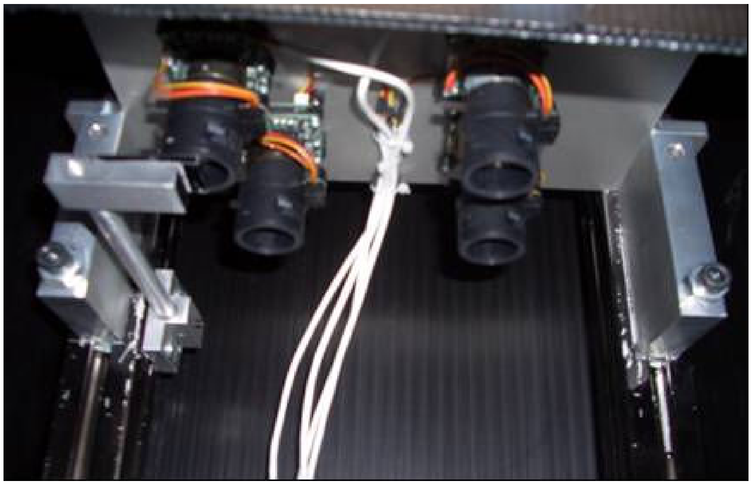


Figure 2.4.1: Initial Four Camera System

Figure 2.4.2 below outlines the field of view (FOV) for each of the four cameras.

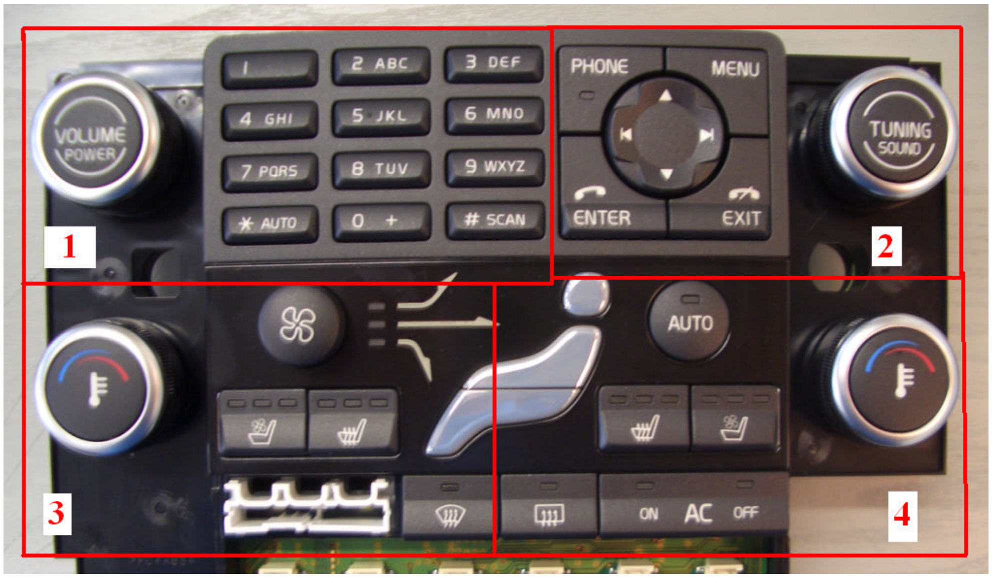


Figure 2.4.2: FOV's of Initial Four Camera System

The four camera system is a low cost alternative but it has a number of issues that become apparent quickly. These are:-

Resolution
Even by using four cameras, the resolution is not adequate. This is especially the case for inspecting the digits, as the complete system has a resolution of 2.4 pixels per millimeter.

Sensor/Lens Quality
The image quality attained from this type of camera is not sufficient. The images are noisy which is a not a desirable trait. The sensor on the camera is sensitive to slight light changes, which is not ideal. Appendix H shows the field of view images for each of the four cameras. Image noise is very visible in these images.

Camera Calibration
The camera used in the four camera system is equipped with a zoom lens. If a camera stops functioning, replacing the camera involves re-calibrating the camera and setting up all the inspections again. This, from a production downtime perspective is not tolerable long term.

The issues outlined above emphasise the importance of camera choice in a machine vision application. Significant gains can be made by selecting a higher resolution industrial machine vision camera. The remaining elements of this dissertation concentrate on a single camera system.

2.5: Project Preparation

2.5.1: Project Budget

Table 2.5.1 below shows the budget breakdown of the project. An overall budget of €5800 is allocated of which €5152 is used. The budget is slightly exceeded on the optical element, due to lack of knowledge on prices of lens. On all other aspects, budget is either on par or below the allocated budget.

Project Element	**Budget Allocated**	**Budget Used**
Camera	€2000	€1769
PC	€800	€800
Interface to PC	€400	€178
Optics	€200	€205
Jig Design	€1000	€800
Load Box	€1000	€1000
Software Cost	€400	€400
Total	**€5800**	**€5152**

Table 2.5.1: Project Budgeting

2.5.2: Knowledge Base Required [1]

In order to develop such a system, it is necessary to have a good solid foundation of knowledge in the area of machine vision. A very good knowledge of computer programming in C++ is also required. It is also necessary to understand the systems engineering and designing of a machine vision application.

2.6: Conclusion

The second chapter outlined some background information on the project. It starts by describing the specification of the product, such as cosmetic layout, potential failure analysis and the electrical specification. It then defines the project requirement from Alps as a company. Finally the project preparation is discussed, covering topics such as the budget allocated to the project, and the knowledge that will be attained to lead to a more successful implementation.

The next chapter describes the system design. It starts by defining the hardware design covering typical elements such the camera choice, optics, lighting and jig design. It then discusses the software design, informing which software is used and why, the graphical user interface (GUI) design and other software elements. Finally the seven algorithms that are used for the analysis are explained in detail.

Chapter 3: System Design

3.1: Introduction

The third chapter describes the system design. It starts by defining the hardware design covering typical elements such the camera choice, optics, lighting and jig design. It then discusses the software design, informing which software is used and why, the graphical user interface (GUI) design and other software elements. Finally the seven algorithms that are used for the analysis are explained in detail.

3.2: Hardware Design

3.2.1: Camera

A key element of the final system is the camera solution that is to be chosen. The camera chosen for this application is the Allied Vision Technologies (AVT) Stingray 201-C. This camera is discussed under the following headings.

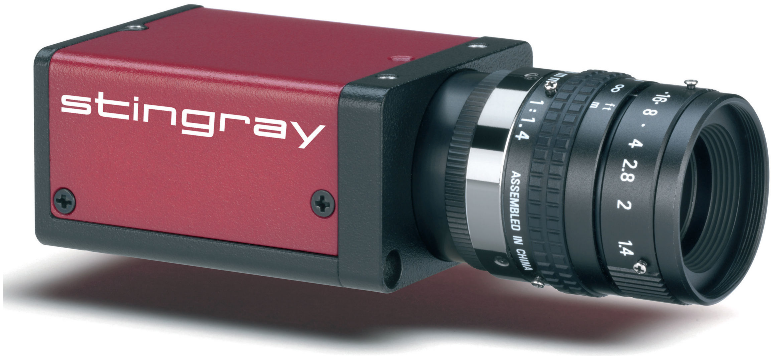


Figure 3.2.1: AVT Stingray 201C Camera [4]

Resolution: The product has an inspection area of 220mm x 120mm. Given that the digits on the number pad need to be inspected, it is necessary to get as many pixels per millimeter as possible. This camera has a resolution of 1634 (Horizontal) by 1224 (Vertical). This results in 7.42 pixels per millimeter, which is sufficient.

Interface: The Stingray interfaces with the PC by using a firewire connection. This removes the need for a framegrabber. Firewire is chosen as it is a good industrial design, with high reliability.

Software Compatibility: The manual for the AVT Stingray shows several programming examples for using this camera in C++. This type of software compatibility is important in case it is necessary to load the settings file directly to the camera from the C drive of the PC using our main program.

Release Date of Camera: The camera is an upgrade of the AVT Marlin, and it is a relatively newly released camera. This is important, as it is the aim to use this system to inspect Alps products other than the Y20 CCM module. It is the intention to standardise the camera for products of similar physical dimensions.

3.2.2: Optics

Some optic parameters are presented in table 3.2.1 below. The distance to product of 320 millimeters is chosen because there are constraints on the height of the final solution (< 1.5 meters from the ground). The camera cannot sit too close to the product either as a certain amount of clearance is needed to load and unload the product.

Parameter	Value (millimeters)
Object Height (Vertical)	220
Object Width (Height)	120
Object to Lens Distance	320
Camera Sensor Size	1/1.8"

Table 3.2.1: Optic Parameters

An essential step is to select a lens size that is bigger than the sensor size of the chosen camera. In this application, the sensor size of the Stingray is 1/1.8 ". To ensure high quality, in-focus images, a 2/3" lens is chosen, which is slightly bigger than the camera sensor size.

Key:

a = Lens-Object Distance
Y' = Image Height
Y = Object Height

$$f' = \frac{a\,Y'}{Y}$$

Figure 3.2.2: Focal Length Formula

Using the formula outlined in figure 3.2.2 above gives a focal length of 12.8 millimeters. The lens chosen has the following parameters.

Parameter	Value
Type	Hi Resolution
Focal Length	12mm
Size	2/3"
F Stop	1.4

Table 3.2.2: Lens Parameters

3.2.3: Lighting

It is often the case to use a lighting solution such as a line bar LED light source or dome LED light source to optimise the stability of your acquired image, especially in areas of varying light. However, introducing lighting into a system must be done only if necessary, as it adds to the complication of the entire system.

All buttons on the Y20 CCM module have button backlighting that can be controlled using the CAN topology. It is proposed to use this button backlighting to illuminate the graphics. Therefore it was deemed that external system lighting is not a suitable design element for this system. Figure 3.2.3 below shows a snapshot of this button backlighting, set to 70% of the maximum intensity of the LED's.

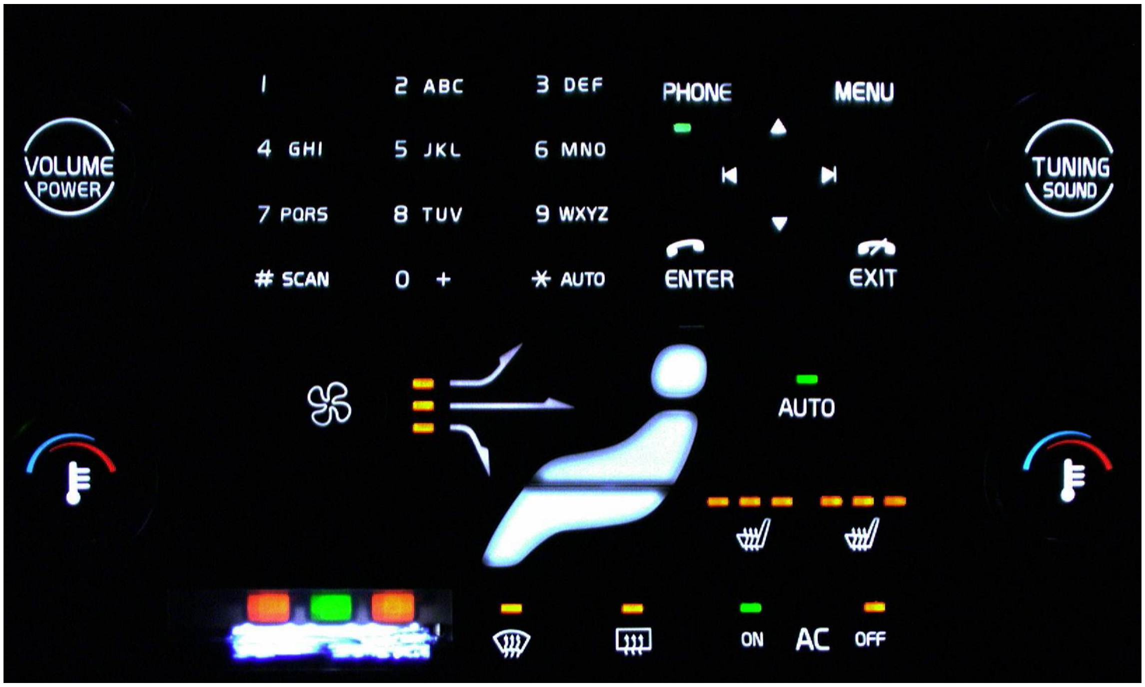


Figure 3.2.3: Full Module Backlighting

3.2.4: Mechanical Jig

Holding the Product: In this application, movement of the product being inspected is ill advised, as it can wreak havoc with the inspection algorithms. The product is held in place by four product ribs, two of which are visible in figure 3.2.4 below. These are also used to keep the product in place rigidly when it is in the vehicle. The metal dowels are machined exactly to fit these holes. The use of a guide plate facilitates the sliding of the product lug holes on top of the locating dowels.

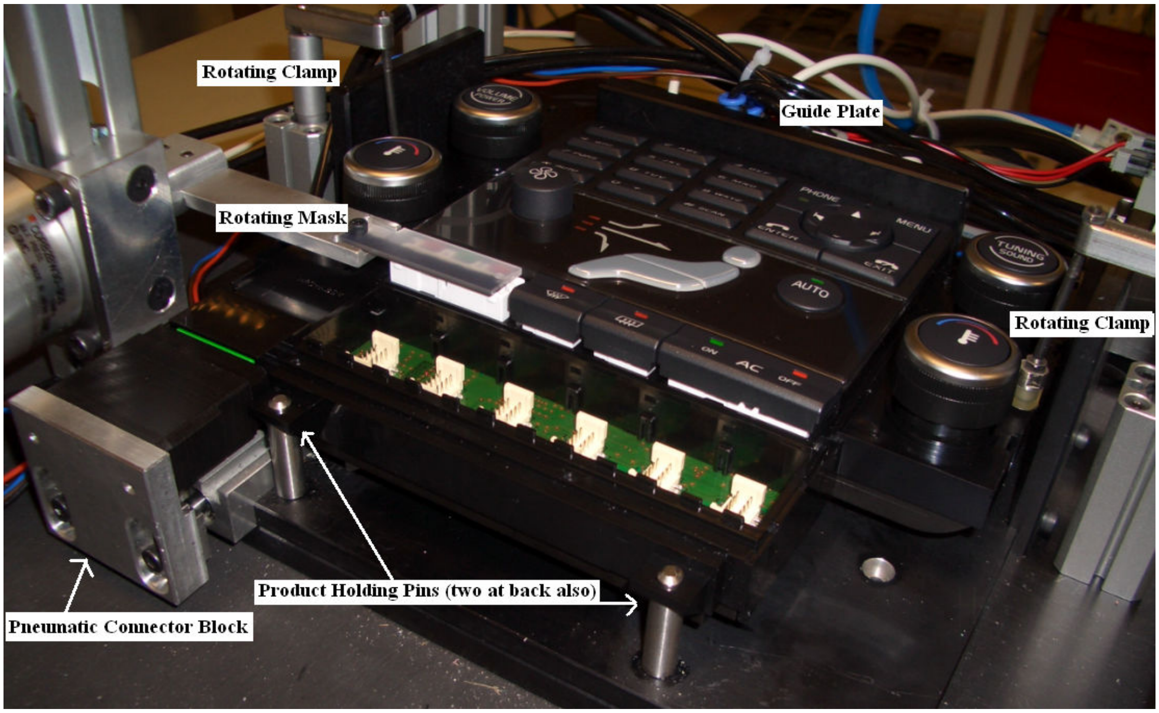

Rotating Clamp
Guide Plate
Rotating Mask
Rotating Clamp
Product Holding Pins (two at back also)
Pneumatic Connector Block

Figure 3.2.4: Mechanical Holding Jig

As an added guarantee that the product is in exactly the same location at all times, two pneumatically driven swing clamps are used to press down on the product as shown in figure 3.2.4 above.

Powering the Module: The module is automatically powered using a pneumatically driven connector block. Figure 3.2.4 above shows the connector block in its driven position to make connection with the product.

Masking out Bright LED's: In section 3.1.3, it is outlined that the LED intensities can be controlled using the CAN network. There are five LEDs that are exposed to the camera, with no button covering them. These are shown in figure 3.2.5 below.

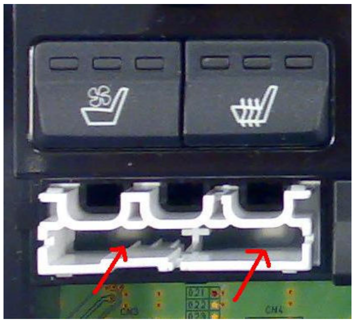


Figure 3.2.5: Location of bright LED's

The intensities of these LED's cannot be controlled separately, and they glare the Stingray camera with their brightness and they project lines of noise up through the area as can be seen in figure 3.2.6 below.

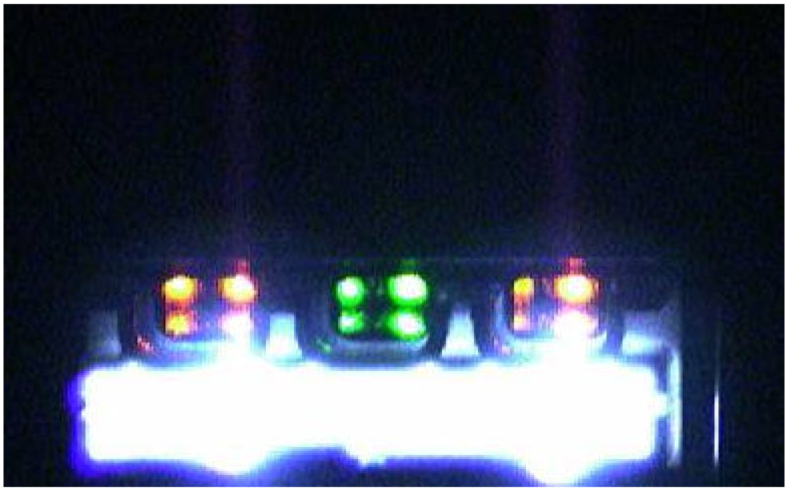


Figure 3.2.6: Noise from bright LED's

In order to solve this problem, it is necessary to apply a mask over this region of the product. This is to diffuse this glare light. However it is part of the quality specification to inspect for the colour of the 3 coloured LED's, and also that the two bright LED's are lighting. A single rectangular piece of diffuser is used, with half of it coloured in black marker. As can be seen from figure 3.2.7 below, the black marker allows a certain amount of light through so that the brightness of the two LED's can be inspected.

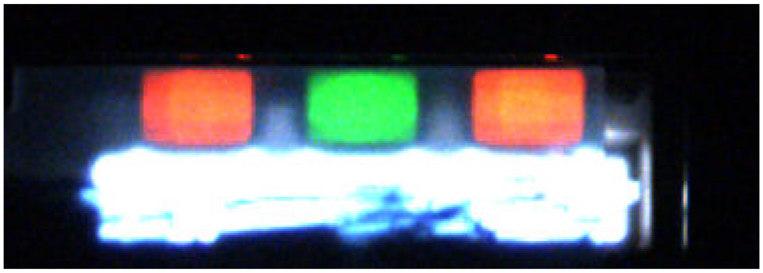


Figure 3.2.7: Final Area Masked

From a production viewpoint, it is not feasible to manually place this mask on this area. A rotary cylinder is employed to pneumatically place this mask over the area.

3.2.5: Jig Control

A method for controlling the jig, and also communicating with the product itself over the CAN network is necessary. This is done by using what Alps refer to as a Load Box. The Load Box for this project is to contain, but not limited to:-

1: 12 Volt and 24 Volt Power Supplies: These are to supply power to the Y20 CCM product, and also to supply power to the various pneumatics on the Jig.

2: Alps USB AD Input/Output Board: This is a module that drives all the pneumatics as outputs and also can take inputs from the various push buttons. The module has a complete library of functions for handling this, all developed within Borland C++ Builder. This makes the Alps USB AD Input/Output Board easy to use from a programming viewpoint.

3: Vector CanCaseXL: This is used to handle all the CAN communication between the PC and product itself. Again, the CanCaseXL comes with a complete library that is compatible with Borland C++ Builder.

4: Main PCB: This PCB contains a series of relays and also a breadboard to interface the stated components of the Load Box to the Jig.

3.2.6: PC and Operating System

In any image processing application, the specification of the PC is important. It is necessary to ensure that the system does not become the slowest element (gating item) of the final assembly cell. The specification of the PC is shown below in table 3.2.2.

Element	**Specification**
Processor	Intel Pentium 4 Processor 3.00 GHz
Memory	2 GB of RAM
Hard Drive	160 GB
Monitor	17 inch LCD
PCI Card	IEEE 1394b Firewire Card

Table 3.2.2: PC Specification

3.3: Software Design

3.3.1: Software Selection

The software language used to implement such an application has a direct impact on many different aspects such as speed of operation, GUI appearance and features. The programming language chosen to develop the application in is Borland C++ Builder 6.

Reasons for choosing Borland C++ Builder

1: Borland C++ Builder 6 is the standard programming environment for Alps worldwide.
2: Alps have programming libraries in Borland for controlling the Alps USB AD I/O and Vector CanCaseXL. This will reduce development time.
3: Borland C++ Builder 6 is very friendly in regard to creating a Graphical User Interface (GUI). This is an essential element of any machine vision application.

3.3.2: Common Vision Blox – Image Manager [8]

There is no known method to acquire an image from the AVT Stingray camera into a Borland project. It is necessary to use an additional piece of software to handle this task. The Image Manager component of the Common Vision Blox image processing SDK is a low cost method of performing this task. It allows the programmer fundamental functions such as acquiring images and also the necessary hardware drivers. If further specialised processing is need, then proprietary algorithms must be purchased.

After carrying out installation instructions for Borland, an extra palette appears in the Borland component palette, and two components are used, shown in figure 3.3.1.

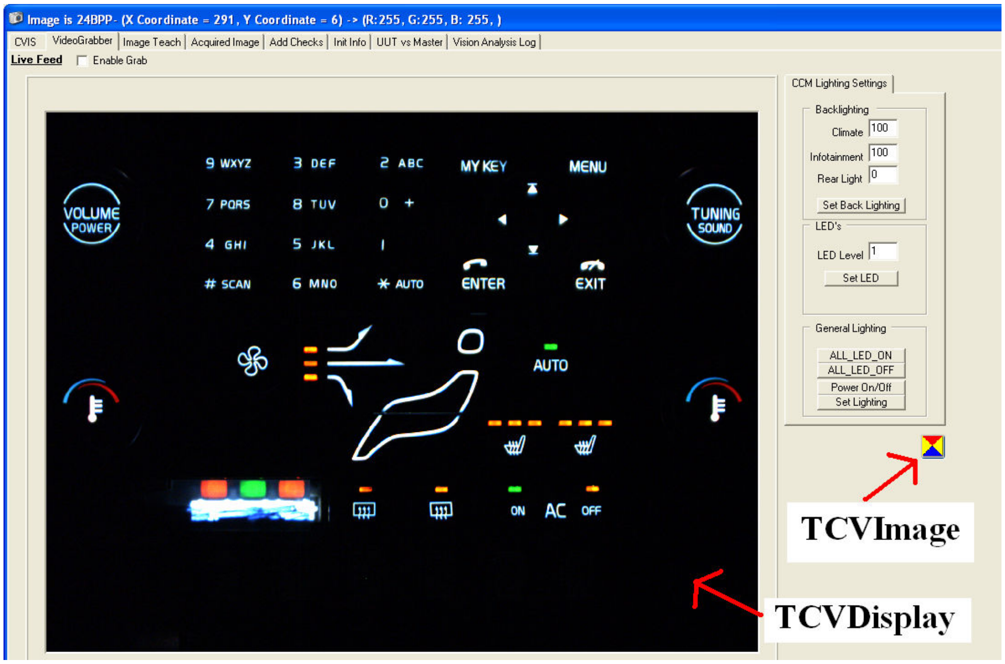


Image is 24BPP- (X Coordinate = 291, Y Coordinate = 6) -> (R:255, G:255, B: 255,)

Climate 100
Infotainment 100
Rear Light 0
LED Level 1

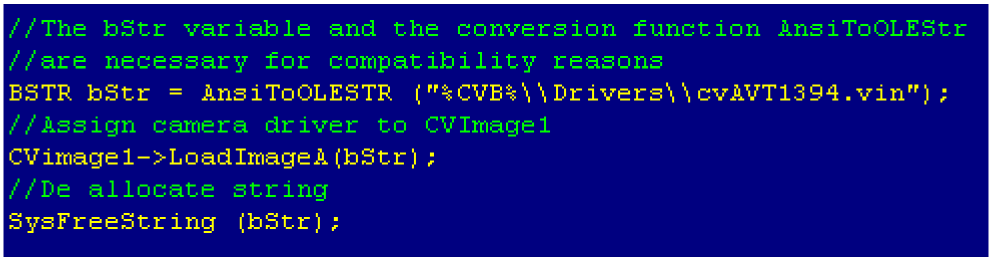
Figure 3.3.1: Common Vision Blox Components

The two components are:

The 'TCVImage' icon represents the Common Vision Image Control.

The 'TCVDisplay' icon represents the Common Vision Display Control.

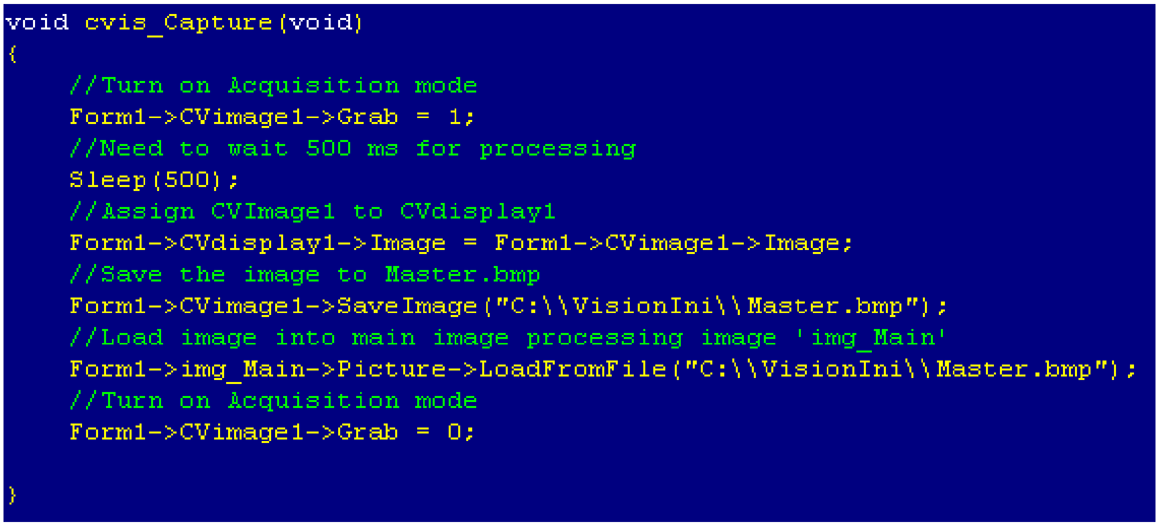
There is a requirement to add the display of live images (provided by the image acquisition device) to the application. At program startup, the camera driver for the AVT camera must be assigned to the TCVImage component. Figure 3.3.2 shows the code to perform this task.



```
//The bStr variable and the conversion function AnsiToOLEStr
//are necessary for compatibility reasons
BSTR bStr = AnsiToOLESTR ("%CVB%\\Drivers\\cvAVT1394.vin");
//Assign camera driver to CVImage1
CVimage1->LoadImageA(bStr);
//De allocate string
SysFreeString (bStr);
```

Figure 3.3.2: Code for assigning AVT driver

The function shown in figure 3.3.3 below, called 'cvis_Capture' is used to acquire an image from the AVT Stingray camera.



```
void cvis_Capture(void)
{
    //Turn on Acquisition mode
    Form1->CVimage1->Grab = 1;
    //Need to wait 500 ms for processing
    Sleep(500);
    //Assign CVImage1 to CVdisplay1
    Form1->CVdisplay1->Image = Form1->CVimage1->Image;
    //Save the image to Master.bmp
    Form1->CVimage1->SaveImage("C:\\VisionIni\\Master.bmp");
    //Load image into main image processing image 'img_Main'
    Form1->img_Main->Picture->LoadFromFile("C:\\VisionIni\\Master.bmp");
    //Turn on Acquisition mode
    Form1->CVimage1->Grab = 0;

}
```

Figure 3.3.3: Code for acquiring an image

3.3.3: Software GUI

The graphical user interface (GUI) is used to convey vital information to the user, be it the operator using the system or the engineer/ technician performing inspection setup. After studying various existing machine vision system GUI's currently on the market, it is determined that the GUI for this application should have the following characteristics:-

Ease of Use for Production Operator: The GUI should be easy to use for the operator. The GUI must be designed from the operator's perspective and not the developers. The application should run seamlessly, only requiring very occasional operator input when instructed to do so. The inspection test should start manually using a hardware startup, such as using double dead-man switches.

Ease of Use for Engineer/Technician: The GUI should be easy to use for the technical person using it. It should not be necessary to have a separate program to re teach inspections. It is thought that the front end GUI will have a password control that will allow the technical person to access other elements of the GUI, made visible upon correct password entry. This secondary GUI is called ImageTeach.

Front End GUI should be Clear: The front end GUI should be clear, not confusing or over cluttered. It should contain all the relevant information that the operator will require at run time. These include:-

- image area currently being inspected,
- operator instruction information,
- tests passed/failed,
- inspection palette,
- password field for technical use,
- inspection status pass/fail for each stage of inspection.

ImageTeach GUI should be application specific: The GUI element that is used by the engineer or technician should contain all the tools necessary for them to re-teach inspections, and to take clear and decisive actions. The GUI should not introduce confusion and should aid the technical person as much as possible.

Bug Free: The GUI as a whole should be free of any software bugs, should start up and close without any Windows errors.

3.3.4: Execution Method

The test is to be executed within a C++ timer. The timer is started at run time, and the program execution follows by using a number of while loops, with appropriate exit conditions. Figure 3.3.4 below outlines the program flow chart.

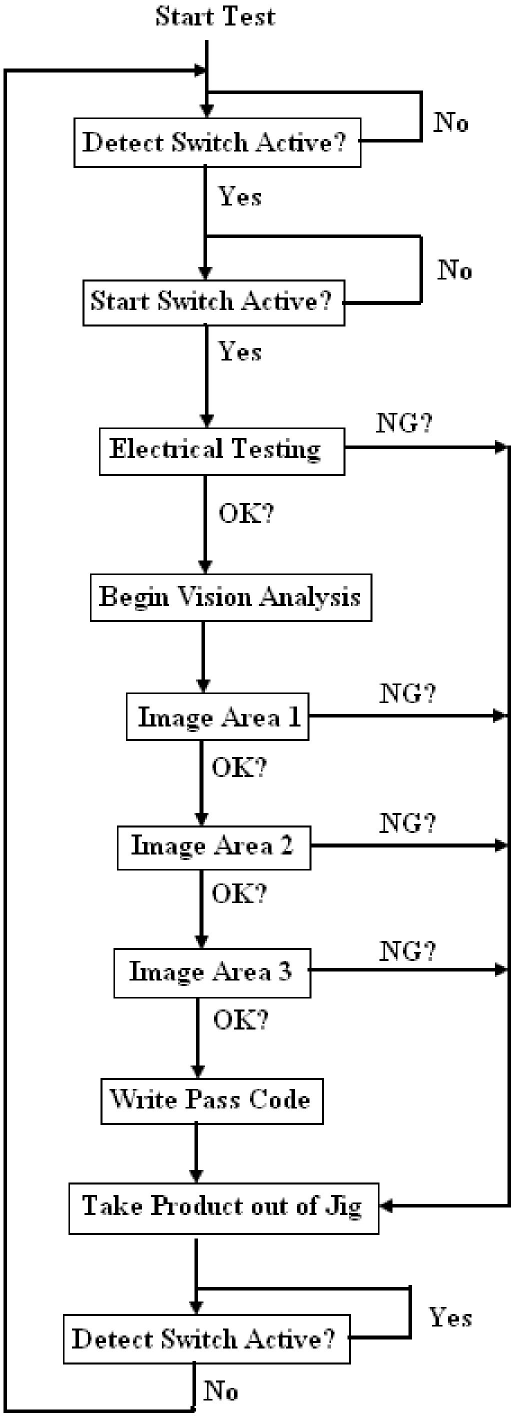
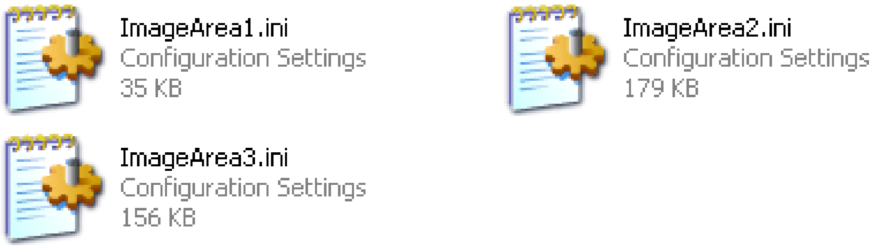


Figure 3.3.4: Program Execution Flow Chart

3.3.5: Storing Inspection Data

The data is to be stored in ini format files. The inspections are split into three ini files, one for each of the image areas that are processed. They are named as shown in figure 3.3.5 below.



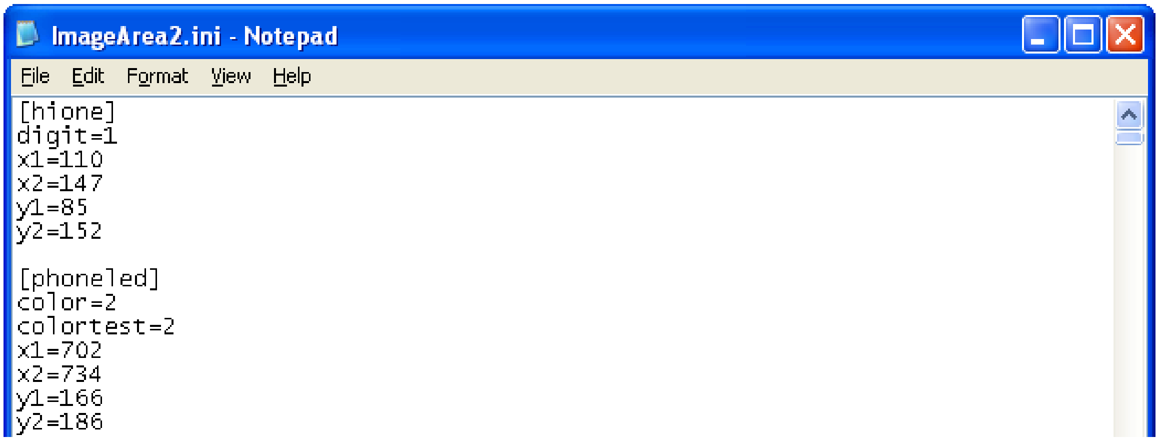
ImageArea1.ini
Configuration Settings
35 KB

ImageArea2.ini
Configuration Settings
179 KB

ImageArea3.ini
Configuration Settings
156 KB

Figure 3.3.5: Ini Inspection File Names

Typical format of the ini files is shown in figure 3.3.6 below. Two inspections are shown for illustration. At run time, when 'ImageArea2' is being processed, all the names of the inspections, along with the configuration parameters are loaded into string lists. The tests that are executed depend on what parameters are set. For example, in figure 3.3.5 below, the test entitled 'hione' is a digit check, and the region of interest (ROI) of the image that is to be inspected is specified by the parameters (x1, y1) to (x2, y2). The inspection entitled 'phoneled' is a colour inspection, inspecting a green LED, specified by the colour parameter set to 2 (see section 3.4.5). This file type is convenient for use with Borland C++ Builder 6.



```
[hione]
digit=1
x1=110
x2=147
y1=85
y2=152

[phoneled]
color=2
colortest=2
x1=702
x2=734
y1=166
y2=186
```

Figure 3.3.6: Ini Inspection File Content

3.3.6: Data Logging

Visibility is important in regard to what is passing through the system. The pass / fail data is to be logged using a combination of Microsoft excel and access. Due to the sheer number of parameters to be logged, it is not feasible to log all measurements to a single log file. Also, it is thought that by using a structure of allotting a log file to each branch of tests, and using a hierarchical approach having a single parent log file will provide an effective solution. It is the intention to have a parent log file on Alps overall equipment efficiency (OEE) system, so that the vision system can be monitored. The product serial number is logged to the parent log file. In every log file that is accessed throughout a single inspection, the serial number is logged, so all the data can be linked together and to the product. The elements are as follows:-

1: Main Log file
This is the parent log file, from which all other log files are referenced. It takes the form of a csv file. The following parameters are logged to it:-

Parameter	**Description**
Software Version	The software version number of the application. When changes are made to the program, the version number is incremented. This is in compliance with Alps test change procedure.
Part Number	The part number of the product being inspected.
Test Number	Every test that is performed is allotted a test number, which is incremented every time a test is completed.
Serial Number	A serial number that is stored to the product at the beginning of the test.
Cell	The identification number of the production cell that the system is on.
Cycle Time	The time it takes for the product to finish the test.
Date & Time	The date and time of inspection.
Test Result	The result of the test. This is either ‘0’ for a failed inspection or ‘1’ for a passed inspection.

Table 3.2.3: Main Log File Parameters

2: Inspection Log Files
Each inspection type, such as a colour check or pattern match, has its own log file. For example, every time a colour inspection is processed, a record of the inspection is taken, along with relevant test results, such as serial number, result, red green and blue values, and what colour was being inspected.

3: Inspection Change Log File
In order to monitor any changes that are made to the system or the inspection data, each time someone makes any change, a record is kept of this, including the name of the person that made the modification, date, time and what they modified. The identity of the person is got from the password that is entered. Each authorised person has a unique password which is checked when the password is entered.

3.4: Algorithm Descriptions [1]

3.4.1: Pre-Processing Step

Prior to any of the following algorithms (apart from the colour inspection algorithm) a pre processing stage is needed, whereby the area is transformed to a strictly binary image, consisting of only two levels, RGB (0,0,0), black and RGB (255,255,255) white. This is done using a thresholding operation.

The threshold level is determined dynamically to account for slight variations in lighting intensity in the LED's. The AVT stingray is a single chip CCD type camera, and the green channel of the input image is used for all threshold calculations, and for thresholding the input image into a binary image.

The dynamic threshold value is determined by the following steps:-

Step 1: For a particular region of interest (ROI), determine the frequency of every green colour from 0 to 255 occuring.

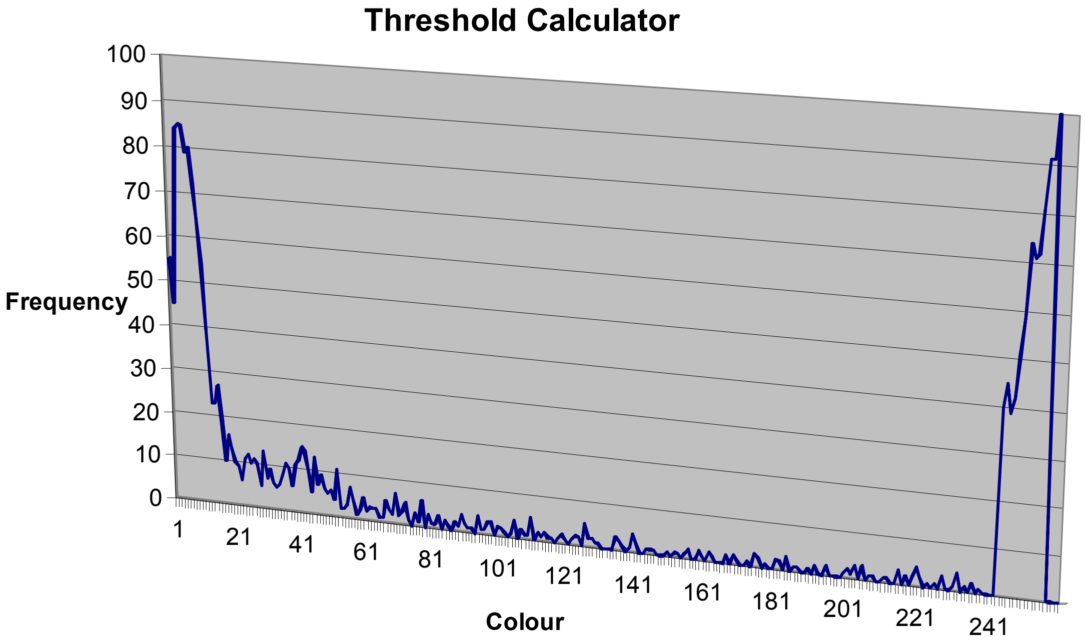


Figure 3.4.1: Colour Frequency Chart

Step 2: Perform a rolling average of every five values on the data to get a smoother curve.

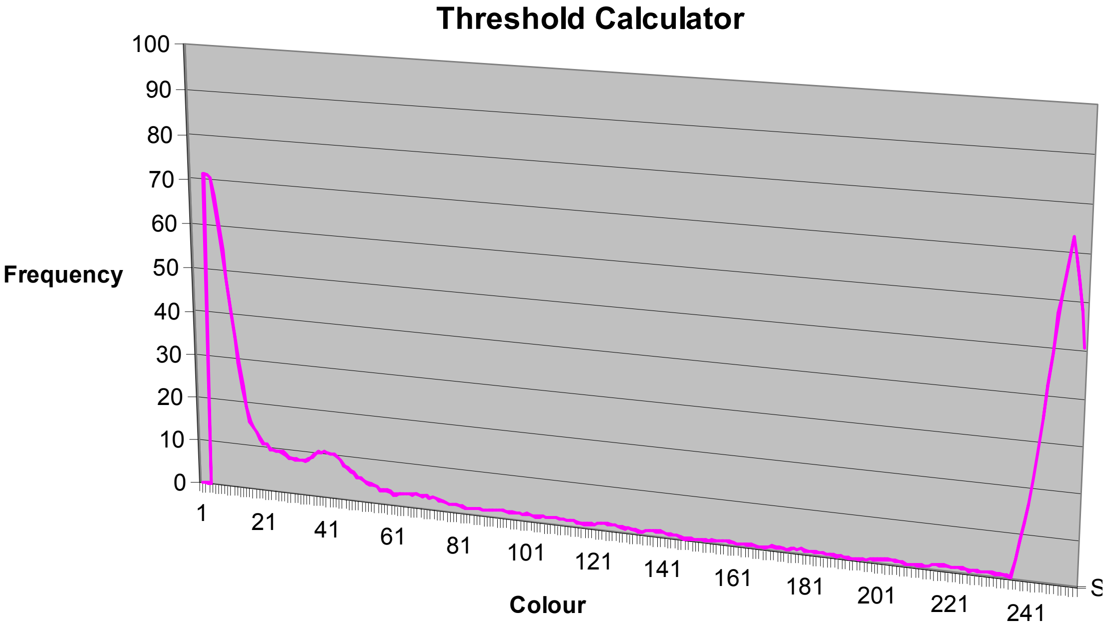


Figure 3.4.2: Colour Frequency Rolling Average Chart

Step 3: Determine the two peak values, and the associated colours. The midpoint of these two colours is the threshold value.

3.4.2: Classic Pattern Match

This operation performs a standard pattern match. It is has two outcomes that are averaged to result in a percentage match.

Outcome 1: Match Percentage

Compare all the pixels in the bounding box of the graphic. When comparing two pixels, there are 4 possible outcomes. Table 3.4.1 below illustrates this.

Outcome	Master Pixel	Test Pixel	Match?
1	0	0	Yes
2	0	1	No
3	1	0	No
4	1	1	Yes

Table 3.4.1: Pattern Match Outcomes

Outcome 2: White Count Percentage

Perform a count of all the white pixels in the test graphic. Compare this to saved white count of master graphic.

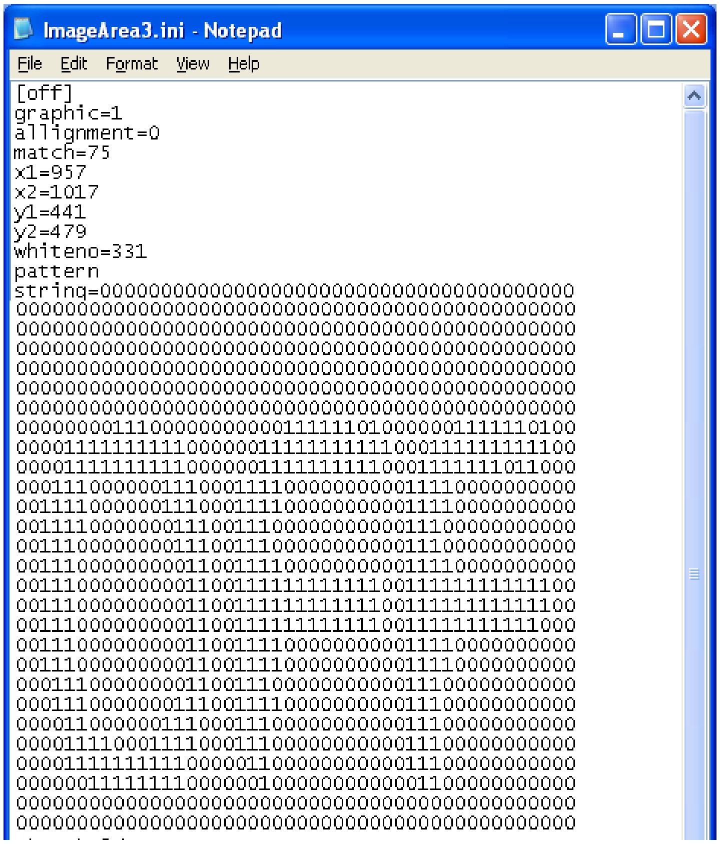
These two percentages are averaged to give a percentage match or fit for the graphic.

Algorithm:

Step 1: Get ROI (x1, y1) to (x2, y2) from ini file. The saved white count and pattern string is loaded from ini file.

Aside:

The pattern string is formed in inspection setup. The boundary box of the setup graphic is scanned left to right, top to bottom. For each pixel in the boundary box, if the pixel under test is above the threshold, set it to 1. If it is below the threshold, set it to 0. A string is saved containing all these individual bits. This is the master graphic pattern string. The white count is also calculated and saved to the ini file. Figure 3.4.3 below shows an example of this for the 'OFF' graphic. This graphic is clearly visible in the pattern string.



```
[off]
graphic=1
allignment=0
match=75
x1=957
x2=1017
y1=441
y2=479
whiteno=331
pattern
string=000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000
0000000011100000000000111111010000001111110100
0000111111111100000011111111111000111111111100
0000111111111100000011111111110001111111011000
0001110000001110001111000000000011110000000000
0011110000001110001111000000000011110000000000
0011110000000111001110000000000011100000000000
0011100000000111001110000000000011100000000000
0011100000000011001111000000000011110000000000
0011100000000011001111111111100111111111111100
0011100000000011001111111111100111111111111100
0011100000000011001111111111100111111111111000
0011100000000011001111000000000011110000000000
0011100000000011001111000000000011110000000000
0001110000000011001110000000000011100000000000
0001110000000111001111000000000011100000000000
0000110000001110001110000000000011100000000000
0000111100011110001110000000000011100000000000
0000111111111100000110000000000011100000000000
0000001111111100000010000000000001100000000000
0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000
```

Figure 3.4.3: Ini Inspection File Content

Step 2: Crop graphic to boundary box around test graphic. For each pixel in the boundary box, if the pixel under test is above the threshold, set it to 1. If it is below the threshold, set it to 0. A string is saved containing all these individual bits. This is the test graphic pattern string.

Step 3: Table 3.4.1 above is now used to compare the master pattern string to the test pattern string. If outcome 1 or 4 from the table occurs, the match percentage is incremented, with respect to the total pixels in the bounding box. An overall match percentage is then found. The white count from the test graphic is found, and a white count match percentage is determined with reference to the master white count.

Step 4: If the average value of the white count and pattern percentages is above the match parameter that is saved in the ini file, then the graphic is a pass. Otherwise it is a fail.

3.4.3: Mahalanobis Taguchi (MT) Method for Pattern Match [6]

Overview of MT Methods

The MT method is named after P.C Mahalanobis, an Indian statistician who invented the distance of Mahalanobis, and Genichi Taguchi, the man who invented the Quality Engineer. It is used to classify a sample as being a good product or a defective one.

A good product always indicates a similar pattern, but a defective product will not display this pattern. The MT Method performs a comparison between the sample that is being tested and a group (group is usually greater than one hundred, but has no upper bound) of known good samples. A distinction can be made between a defective product and a good product by measuring the distance from the defective product to the standard deviation of the group of good products.

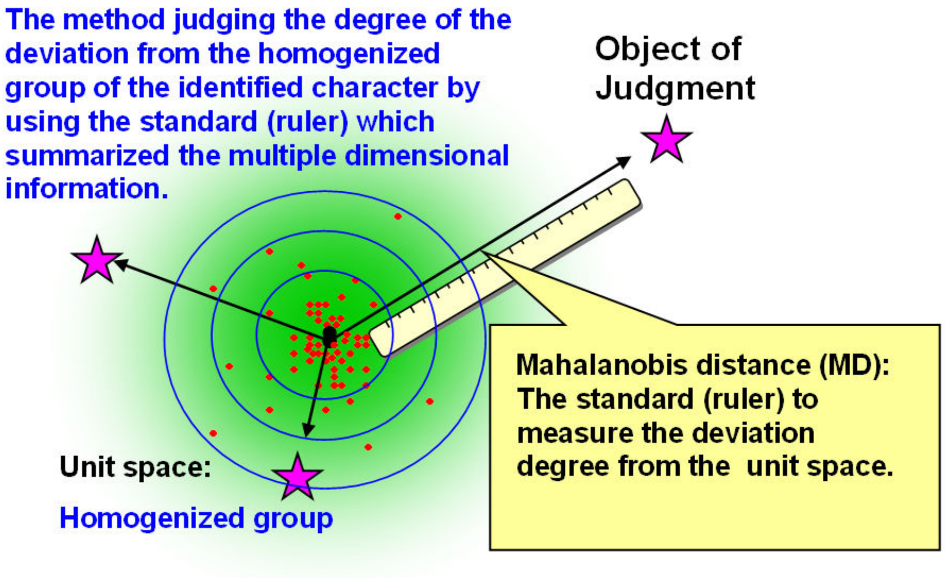

The method judging the degree of the deviation from the homogenized group of the identified character by using the standard (ruler) which summarized the multiple dimensional information.
Object of Judgment
Mahalanobis distance (MD): The standard (ruler) to measure the deviation degree from the unit space.
Unit space:
Homogenized group

Figure 3.4.4: MT Method Overview Diagram [6]

A unit space is built up from a quantity of good samples. This quantity is greater than 100, but the more populated the unit space, the better the classification. The Mahalanobis distance (MD) is the measure of the deviation of a sample from the unit space.

Define the Purpose:
The first step of using MT method is to determine the purpose of using the methods. The purpose is to determine that a particular test graphic pattern is of good fit with the master graphic pattern unit space.

Determine the features of distinction:
The second step of using the MT method is to determine what features of the object of inspection will have a big influence if a failure is present. It is necessary to check to see if they lead to an obvious difference in the judgment.

Consider the following example of comparing two graphics, ‘abc’ and ‘def’, as shown in figure 3.4.5.

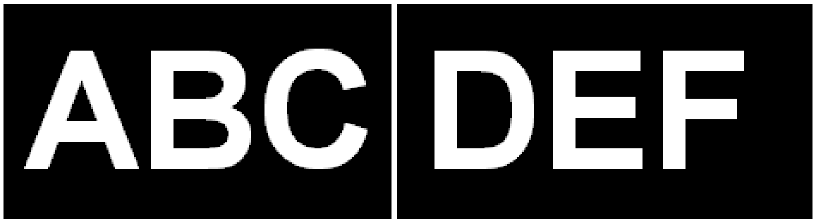


Figure 3.4.5: Ini Inspection File Content

The width or height of the graphic is not a good feature of distinction, as they are similar for both graphics. The chosen feature of distinction is a regionalised white pixel count, whereby the graphic is broken into nine areas, as shown in figure 3.4.6 below.

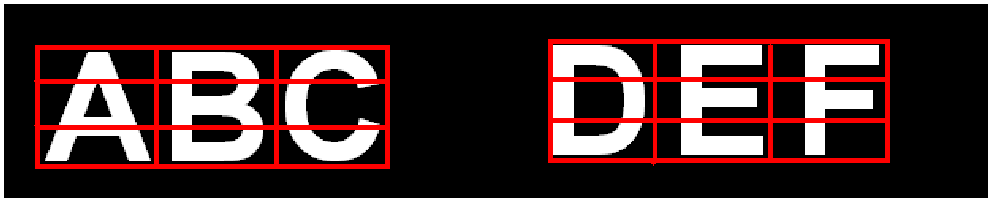


Figure 3.4.6: Ini Inspection File Content

The results of the regionalised white count from these two graphic is shown in table 3.4.2 below.

Sample	Area1	Area2	Area3	Area4	Area5	Area6	Area7	Area8	Area9
abc s1	52	78	59	73	75	29	32	63	50
abc s2	52	78	58	73	75	29	33	63	52
abc s3	54	77	57	73	75	29	33	61	48
abc s4	53	76	58	73	76	30	34	63	51
abc s5	53	76	57	73	76	30	33	63	52
def s1	72	60	63	62	63	76	72	54	27
def s2	74	62	63	61	64	69	67	56	21
def s3	71	59	62	59	65	76	69	60	21
def s4	74	59	62	58	65	67	62	51	21
def s5	72	61	64	60	61	77	69	59	22

Table 3.4.2: Ini Inspection File Content

From studying the table it is determined that there are significant differences in white counts in seven out of the nine areas. This is enough to give a significant difference when a large unit space is formed.

Collect the Data:

In order to form the unit space, a data set is collected, from a group of two hundred different samples. This is done in order to form the unit space. The unit space is represented as a single entity by using a combination of the average and standard deviation of each of the sections along with an inverse matrix representation of this unit space.

Determining the Mahalanobis Data [7]

Normalisation: [7]

It is essential to normalise all of the data in the collected unit space. This is done using the following formula.

$$\text{Normalised Value} = \frac{\text{Original Value - Average}}{\text{Standard Deviation}}$$

Figure 3.4.7: Normalisation Formula

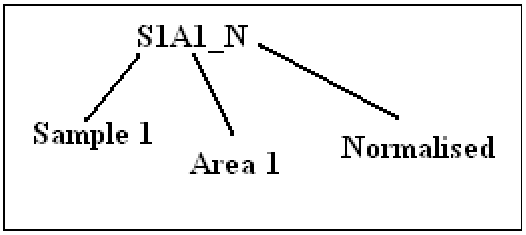


Figure 3.4.8: Table 3.4.3 Mnemonic

Sample	Area1	Area2	Area3	Area4	Area5	Area6	Area7	Area8	Area9
abc s1	S1A1_N	S1A2_N	S1A3_N	S1A4_N	S1A5_N	S1A6_N	S1A7_N	S1A8_N	S1A9_N
abc s2	S2A1_N	S2A2_N	S2A3_N	S2A4_N	S2A5_N	S2A6_N	S2A7_N	S2A8_N	S2A9_N
abc s3	S3A1_N	S3A2_N	S3A3_N	S3A4_N	S3A5_N	S3A6_N	S3A7_N	S3A8_N	S3A9_N
abc s4	S4A1_N	S4A2_N	S4A3_N	S4A4_N	S4A5_N	S4A6_N	S4A7_N	S4A8_N	S4A9_N
abc s5	S5A1_N	S5A2_N	S5A3_N	S5A4_N	S5A5_N	S5A6_N	S5A7_N	S5A8_N	S5A9_N

Table 3.4.3: Normalised Data

Calculating the Correlation Coefficient Matrix: [7]

The coefficient of correlation between each of the nine sections in the regionalised graphic must be determined to calculate the correlation coefficient matrix. The first step here is to represent the data from table 3.4.3 above in a slightly different form for the correlation matrix formula. This is shown in table 3.4.4 below.

Sample	Area1	Area2	Area3	Area4	Area5	Area6	Area7	Area8	Area9
abc s1	Y_{11}	Y_{21}	Y_{31}	Y_{41}	Y_{51}	Y_{61}	Y_{71}	Y_{81}	Y_{91}
abc s2	Y_{12}	Y_{22}	Y_{32}	Y_{42}	Y_{52}	Y_{62}	Y_{72}	Y_{82}	Y_{92}
abc s3	Y_{13}	Y_{23}	Y_{33}	Y_{43}	Y_{53}	Y_{63}	Y_{73}	Y_{83}	Y_{93}
abc s4	Y_{14}	Y_{24}	Y_{34}	Y_{44}	Y_{54}	Y_{64}	Y_{74}	Y_{84}	Y_{94}
abc s5	Y_{15}	Y_{25}	Y_{35}	Y_{45}	Y_{55}	Y_{65}	Y_{75}	Y_{85}	Y_{95}

Table 3.4.4: Renamed Normalised Data

Each coefficient of correlation in the correlation matrix is calculated as shown in figure 3.4.9.

$$r_{ij} \quad = \quad r_{ji} \quad = \quad \Sigma\, (\, y_{ij} \text{ x } y_{ji} \,) \,/\, n$$

Figure 3.4.9: Correlation Matrix Coefficient Formula

The completed matrix is shown in table 3.4.5 below. The diagonal elements are equal to one as the associated areas are the same graphic.

	Area1	Area2	Area3	Area4	Area5	Area6	Area7	Area8	Area9
Area1	1	r_{12}	r_{13}	r_{14}	r_{15}	r_{16}	r_{17}	r_{18}	r_{19}
Area2	r_{21}	1	r_{23}	r_{24}	r_{25}	r_{26}	r_{27}	r_{28}	r_{29}
Area3	r_{31}	r_{32}	1	r_{34}	r_{35}	r_{36}	r_{37}	r_{38}	r_{39}
Area4	r_{41}	r_{42}	r_{43}	1	r_{45}	r_{46}	r_{47}	r_{48}	r_{49}
Area5	r_{51}	r_{52}	r_{53}	r_{54}	1	r_{56}	r_{57}	r_{58}	r_{59}
Area6	r_{61}	r_{62}	r_{63}	r_{64}	r_{65}	1	r_{67}	r_{68}	r_{69}
Area7	r_{71}	r_{72}	r_{73}	r_{74}	r_{75}	r_{76}	1	r_{78}	r_{79}
Area8	r_{81}	r_{82}	r_{83}	r_{84}	r_{85}	r_{86}	r_{87}	1	r_{89}
Area9	r_{91}	r_{92}	r_{93}	r_{94}	r_{95}	r_{96}	r_{97}	r_{98}	1

Table 3.4.5: Correlation Matrix

Calculating the Inverse Matrix: [7]

Due to the nature of the correlation matrix i.e. that rij = rji, using the standard method for inversing this matrix would not suffice, as this would simply transpose the same coefficient of correlation. The inverse matrix is calculated as shown in figure 3.4.10 below, with each term A (i, j) calculated as shown in figure 3.4.11. To make the determinant of the matrix to the 8th degree, we multiply each term by $(-1)^{i+j}$.

$$R^{-1} = \frac{1}{R} = \begin{pmatrix} A(1,1) & A(1,2) & A(1,3) & A(1,4) & A(1,5) & A(1,6) & A(1,7) & A(1,8) & A(1,9) \\ A(2,1) & A(2,2) & A(2,3) & A(2,4) & A(2,5) & A(2,6) & A(2,7) & A(2,8) & A(2,9) \\ A(3,1) & A(3,2) & A(3,3) & A(3,4) & A(3,5) & A(3,6) & A(3,7) & A(3,8) & A(3,9) \\ A(4,1) & A(4,2) & A(4,3) & A(4,4) & A(4,5) & A(4,6) & A(4,7) & A(4,8) & A(4,9) \\ A(5,1) & A(5,2) & A(5,3) & A(5,4) & A(5,5) & A(5,6) & A(5,7) & A(5,8) & A(5,9) \\ A(6,1) & A(6,2) & A(6,3) & A(6,4) & A(6,5) & A(6,6) & A(6,7) & A(6,8) & A(6,9) \\ A(7,1) & A(7,2) & A(7,3) & A(7,4) & A(7,5) & A(7,6) & A(7,7) & A(7,8) & A(7,9) \\ A(8,1) & A(8,2) & A(8,3) & A(8,4) & A(8,5) & A(8,6) & A(8,7) & A(8,8) & A(8,9) \\ A(9,1) & A(9,2) & A(9,3) & A(9,4) & A(9,5) & A(9,6) & A(9,7) & A(9,8) & A(9,9) \end{pmatrix}^{T}$$

Figure 3.4.10: Inverse Matrix Calculation

$$A(1,1) = (-1)^{i+j} \begin{pmatrix} 1 & r_{23} & r_{24} & r_{25} & r_{26} & r_{27} & r_{28} & r_{29} \\ r_{32} & 1 & r_{34} & r_{35} & r_{36} & r_{37} & r_{38} & r_{39} \\ r_{42} & r_{43} & 1 & r_{45} & r_{46} & r_{47} & r_{48} & r_{49} \\ r_{52} & r_{53} & r_{54} & 1 & r_{56} & r_{57} & r_{58} & r_{59} \\ r_{62} & r_{63} & r_{64} & r_{65} & 1 & r_{67} & r_{68} & r_{69} \\ r_{72} & r_{73} & r_{74} & r_{75} & r_{76} & 1 & r_{78} & r_{79} \\ r_{82} & r_{83} & r_{84} & r_{85} & r_{86} & r_{87} & 1 & r_{89} \\ r_{92} & r_{93} & r_{94} & r_{95} & r_{96} & r_{97} & r_{98} & 1 \end{pmatrix}$$

Figure 3.4.11: Inverse Matrix Term Calculation

Calculating the Mahalanobis Distance: [7]

The run time test graphic is broken down into nine regions and the white count is determined for each area. These nine values are normalized using the methods suggested in section entitled ‘Normalisation’ above, and a matrix N is formed.

$$N \quad = \quad \begin{array}{ccc} norm\_y_{11} & norm\_y_{12} & norm\_y_{13} \\ norm\_y_{11} & norm\_y_{22} & norm\_y_{23} \\ norm\_y_{31} & norm\_y_{32} & norm\_y_{33} \end{array}$$

Figure 3.4.12: Normalised Run Time Graphic White Count Matrix

The Mahalanobis Distance (MD) value is calculated using the following formula.

$$MD = \frac{1}{k}(N)R^{-1}(N)^{T}$$

Figure 3.4.13: Mahalanobis Distance Formula

Where

N = Normalised test data
R^{-1} = Inverse matrix representing good product unit space
K = Order of the matrix R^{-1}, in this case 9.

A specification threshold is formed, and if the total MD value is above this threshold, the test data is belonging to a defective product with reference to the unit space.

Algorithm:

Part 1: Set Up of MT data

Step 1: Get ROI (x1, y1) to (x2, y2) from ini file

Step 2: Crop to bounding box, and split into nine regions as shown in figure 3.4.6 above.

Step 3: Perform white pixel count in each of the nine regions. These nine values are stored to an excel file. This operation is repeated for two hundred different samples

Step 4: This information is passed to Alps MT data calculator. This data calculator determines the average, standard deviation associated with each of the nine regions and also the inverse matrix. (See Appendix E for example of this data).

Part 2: Run time determination of Mahalanobis Distance

Step 1: Get ROI (x1, y1) to (x2, y2) from ini file

Step 2: Crop to bounding box, and split into nine regions as shown in figure 3.4.6 above.

Step 3: Perform white pixel count in each of the nine regions.

Step 4: Pass these values and information from part one to the MT algorithm. If Mahalanobis distance is greater than spec, it's a fail.

3.4.4: Digit Inspection

The digit inspection function inspects the region of interest to determine if it contains the correct digit. The digits that can be inspected for are 0-9. The character * and # are also inspected by this algorithm. The algorithm applies an exclusive OR operation between a master image of a digit and an image of the digit being inspected. The resultant output image is searched using a 3 x 4 mask of 1's. If there are one or more occurrences of this mask, an incorrect digit is present.

Algorithm:

Step 1: Get ROI (x1, y1) to (x2, y2) from ini file.

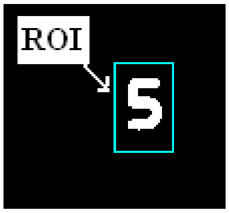


Figure 3.4.14: Digit ROI

Step 2: Crop to bounding box. Leave a three pixels wide black border around bounding box. This is to facilitate the mask in step seven below.

Step 3: Set bounding box plus border to image under test (IUT) image.

Step 4: Load master image from hard drive.

Step 5: XOR test image and master image.

Step 6: Store XOR'ed image as output image.

Step 7: Pass 3 (H) by 4 (V) mask over the image. This mask is shown in figure 3.4.15 below. A '1' constitutes a white pixel. If there is the presence of this mask in the resultant XOR image, digit is incorrect. Otherwise digit is correct.

1	1	1
1	1	1
1	1	1
1	1	1

Figure 3.4.15: 3 (H) x 4 (V) Mask

A correct digit illustration is shown in figure 3.4.16 below called 'Ideal Output Image'. This is a perfect result, with no white pixels present in the output image. An imperfect result is shown also, which would be an actual run time resultant image. There is a certain level of white pixels present, but never in the form of a 3 (H) x 4(V) block.

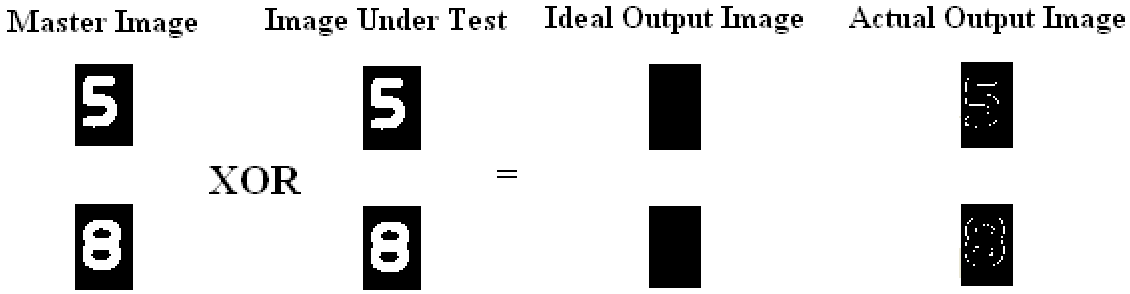


Figure 3.4.16: Correct Digit Illustration

An incorrect digit illustration is shown in figure 3.4.17 below. As can be seen, there is a large degree of white pixels, with multiple occurrences of the 3 (H) x 4(V) block of white pixels.

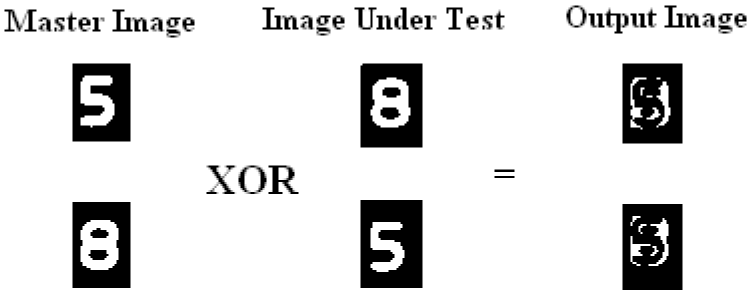


Figure 3.4.17: Incorrect Digit Illustration

3.4.5: Colour Inspection

The colour inspection function searches the region of interest (ROI) for the presence of a particular colour. The only colours on the Y20 CCM product are the three primary colours, red green and blue. In the below description, the colour discussed is red. The same approach is taken exactly for the other two primary colours green and blue.

It is not a quality requirement to measure the exact intensity of the colour, but it is necessary to ensure that the colour is not too dull. This is done by only considering red colours greater than red=160. An example of dullness is shown in figure 3.4.18 below, There is paint back spray on the white under area of the button, thus the material is not allowing enough LED light propagate through the material Depending on which colour is being inspected for, the ROI is searched and the accumulated colour totals for the red green and blue planes are determined. If the expected major colour within the ROI is red, then the total red count should be greater than that of the green and blue counts.

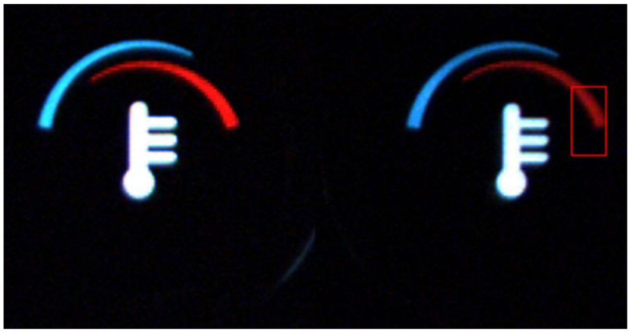


Figure 3.4.18: Right hand temp red colour too dull

Step 1: Get ROI (x1, y1) to (x2, y2) from ini file

Step 2: Get 'colour' parameter from ini file

If colour = 0 do Red colour inspection
If colour = 1 do Blue colour inspection
If colour = 2 do Green colour inspection

Step 3: Perform search on ROI, only considering pixels with colour of inspection greater than 160 (This is lower quality specification limit).

Then for current pixel:
Add current red colour to overall red count, red_total.
Add current green colour to overall green count, green_total.
Add current blue colour to overall blue count, blue_total.

Step 4: Judge the three totals for the colour. If red is the dominant colour, red_total will be greater than green_total and blue_total.

3.4.6: Presence Check

The presence check operation checks a region of interest (ROI) for the presence of white pixels.

Step 1: Get ROI: (x1, y1) to (x2, y2) from ini file.

Step 2: Check ROI for presence of white pixels. If there are white pixels present, return OK, else return NG.

3.4.7: Blank Check

The blank check operation checks a region of interest (ROI) to ensure there are no white pixels present.

Step 1: Get ROI: (x1, y1) to (x2, y2) from ini file.

Step 2: Check ROI for presence of white pixels. If there are white pixels present, return NG, else return OK.

3.4.8: Distance Measurement

The distance measurement operation calculates the width of a graphic, along a specific y coordinate. It scans along a particular y value line, finding the two edges. It then finds the distance, in pixels between these edges.

Step 1: Get ROI: (x1, y1) to (x2, y2) from ini file.

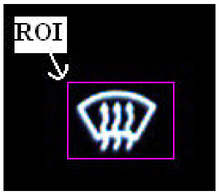


Figure 3.4.19: ROI of Distance Measurement Algorithm

Step 2: Crop image to bounding box enclosing graphic. Bounding box coordinates are (x1_crop, y1_crop) to (x2_crop, y2_crop).

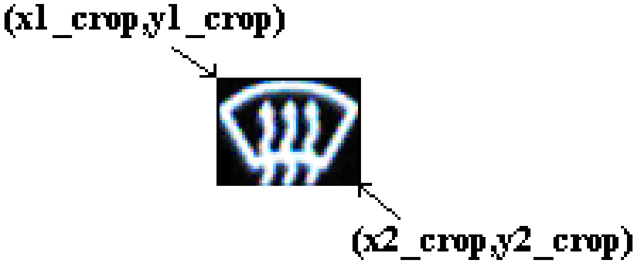


Figure 3.4.20: Cropped ROI of Distance Measurement Algorithm

Step 3: In y, drop y_drop pixels from top of graphic.

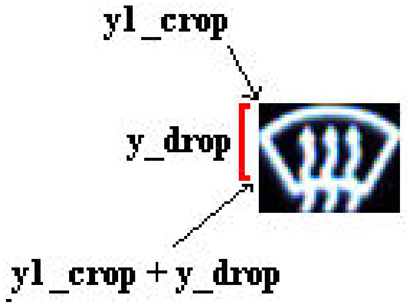


Figure 3.4.21: Drop stage of Distance Measurement

Step 4: Start from x1_crop and scan along single y value (y1_crop+y_drop) until a white value edge pixel is found. This x value is set to x_edge1. Repeat step 4 for x2 to x1 until you find the second edge white pixel. This x value is set to x_edge2.

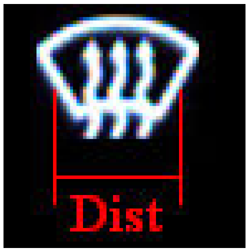


Figure 3.4.22: Distance across Graphic Result

The distance across the graphic is calculated using the formula in figure 3.4.23 below.

$$\text{Dist} = \text{x\_edge2} - \text{x\_edge1}$$

Figure 3.4.23: Distance Formula

3.5: Conclusion

The third chapter described the system design. It starts by defining the hardware design covering typical elements such the camera choice, optics, lighting and jig design. It then discusses the software design, informing which software is used and why, the graphical user interface (GUI) design and other software elements. Finally the seven algorithms that are used for the analysis are explained in detail.

The next chapter describes the system implementation of the project. It starts by describing the implementation stage of the hardware, showing the final system diagram, and the jig implementation. It then outlines the problems encountered. Finally the software implementation is presented. This discusses the key elements of the software such as using the camera, the final GUI, inspection setup along with other software elements.

Chapter 4: System Implementation

4.1: Introduction

The fourth chapter describes the system implementation of the project. It starts by describing the implementation stage of the hardware, showing the final system diagram, and the jig implementation. It then outlines the problems encountered. Finally the software implementation is presented. The software implementation discusses the key elements of the software such as using the camera, the final GUI, inspection setup along with other software elements.

4.2: Hardware Implementation

4.2.1: Final Position of System in Production Process

Figure 4.2.1 below shows the layout of the final assembly cell for this product. The Vision System is installed between the End of Line (EOL) electrical testing and the panel press jig. The product is fully assembled after the panel press jig stage, and this is the first time the unit can be inspected.

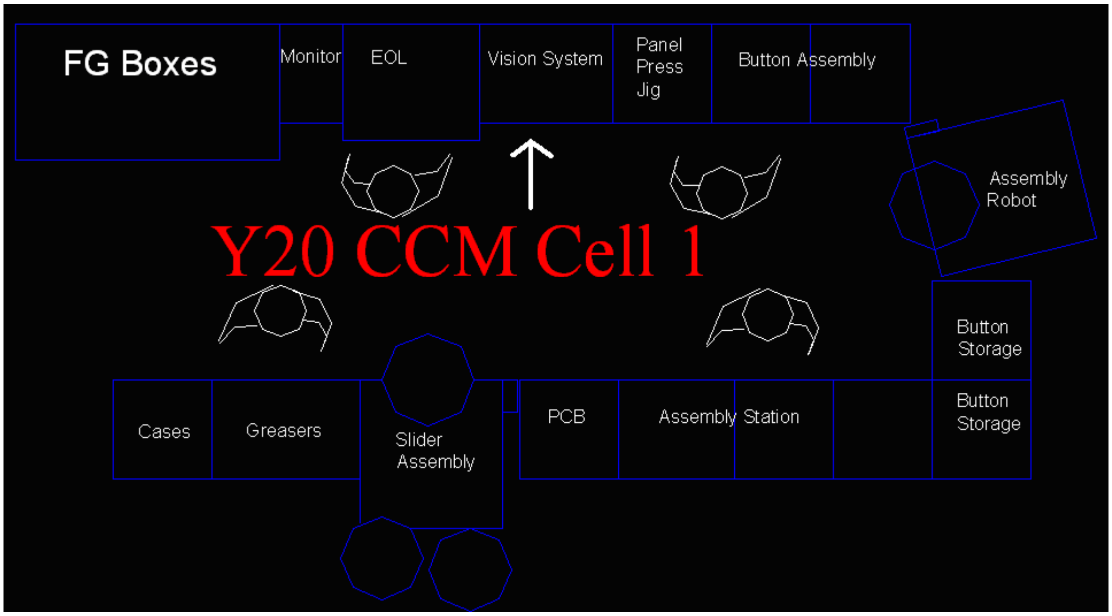


Figure 4.2.1: Y20 Production Cell Layout

Figure 4.2.2 below shows the Vision System on the production line. In following with the diagram in figure 4.2.1 above, it is placed to the right of the EOL electrical testing stage.

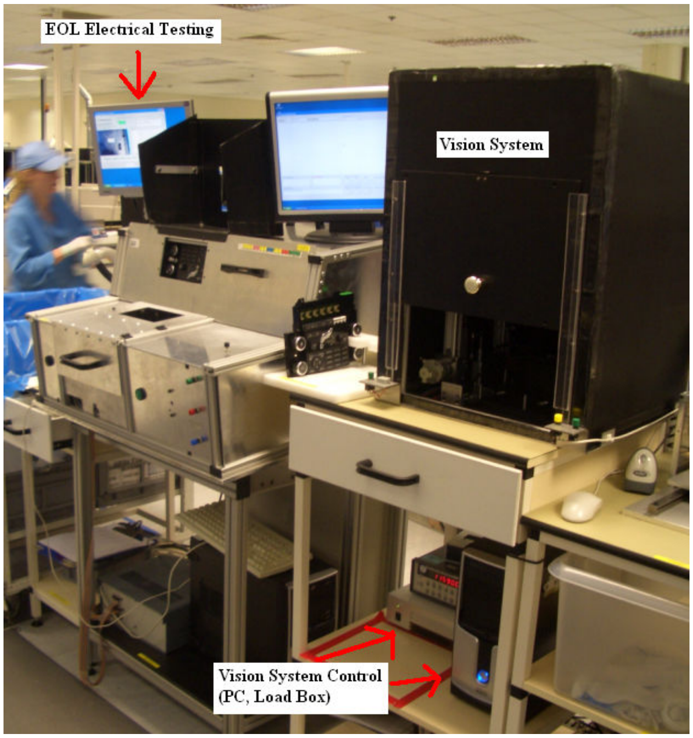


Figure 4.2.2: Y20 Vision System Positioning

4.2.2: Final System Schematic

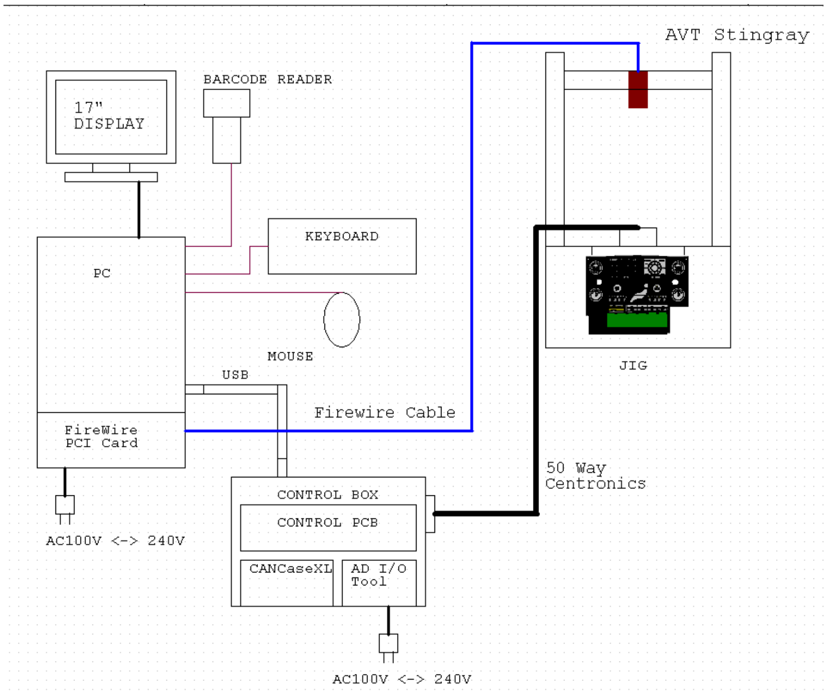


Figure 4.2.3: Final System Schematic

4.2.3: Final Jig Implementation

The final jig implementation is shown in figure 4.2.4 below. The dark hood is removed for illustration. The jig sits on a baseplate which has two functional panels at the front. They contain two deadman start switches to enable the operator to start the test, and a reset switch. A buzzer is included to sound when a test fails. The AVT Stingray camera is mounted on ¾" item profile. The jig is on a standard production area workbench. There is no vibration in the environment that may interfere with the test. All pneumatics sit at the back of the holding jig. The electrical load box that is used to interact with this jig and the PC is shown in appendix F.

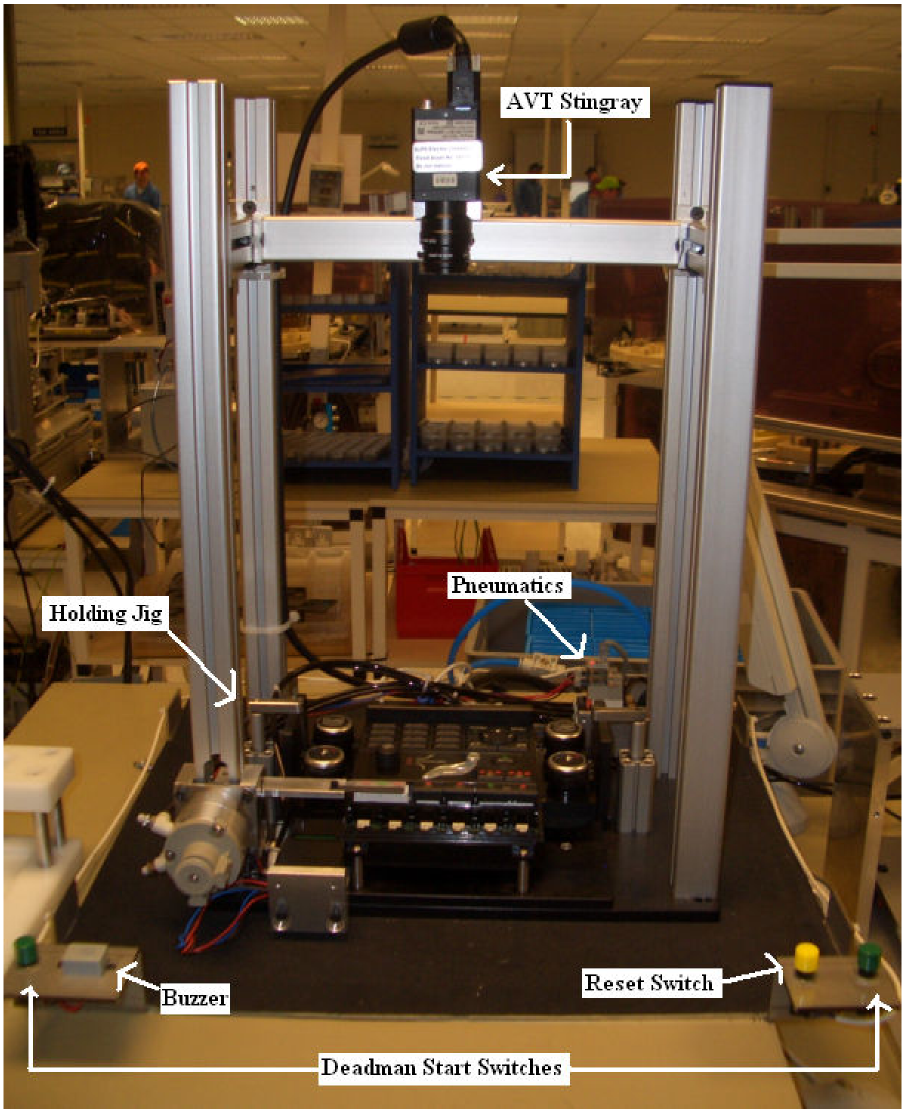


Figure 4.2.4: Final Jig Implementation

4.2.4: Hardware Problems Encountered

Vector CanCaseXL: The module used to control the electrical and lighting properties of the module gave some trouble. If plugged directly into the PC, even via a hub, it delayed the start up time of the PC by two minutes. The cause for this was never determined, but predicted to be a PC issue, as these modules are used on many Alps products. Instead of replacing the PC and all the installation and set up that would have to be performed, a simpler solution is sought. It is noticed that once the PC starts up, there is no issue. The CanCaseXL is plugged into a hub which is powered through a relay. This relay is enabled at program start up.

Product Loading: The product has four lugs that are used to hold the product in place. An issue that arises straight away is how to guide the unit onto these lugs. The issue is solved by using guiding plates at the back and sides of the product. An enhanced solution that is to be looked at is to load the product outside the system and pneumatically drive it underneath the camera.

4.3: Software Implementation

4.3.1: Using the AVT Stingray Camera

The AVT Stingray camera has a package called AVT Smartview. The starting up GUI for this application is shown in figure 4.3.1 below.

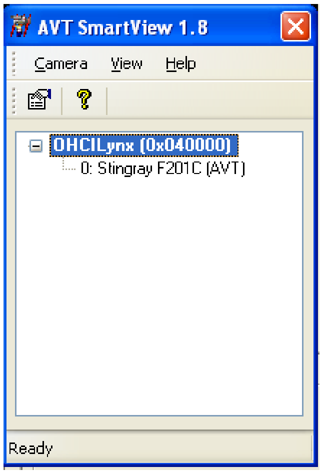
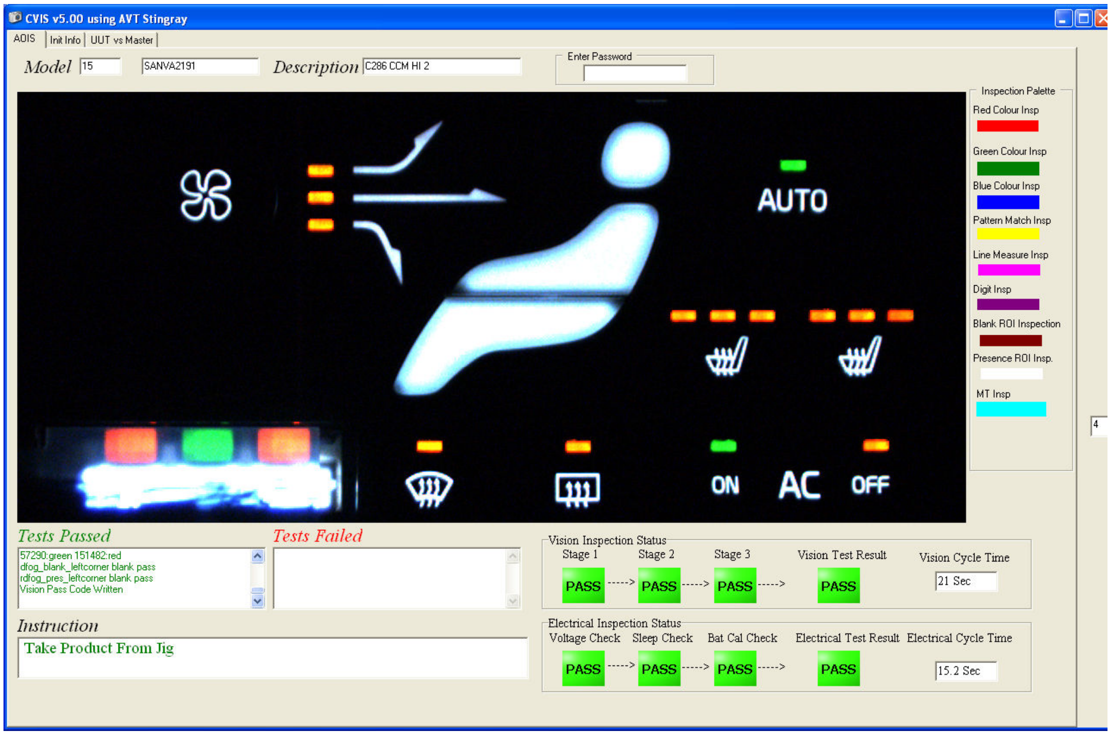


Figure 4.3.1: AVT Smartview startup GUI

The initial set up of the camera is done using Smartview. Once Smartview is loaded, it searches for any compliant hardware on the system. As can been seen in figure 4.3.1 above, the Stingray is found. The live view from the camera is then automatically displayed, along with the settings dialog. Various settings are changed, and these settings are shown in Appendix B. The settings are saved using the flash banks on the camera itself. These settings are loaded automatically when the camera starts up, thus removing the need to load them in using code.

4.3.2: Software GUI

The final front end GUI is shown in figure 4.3.2 below. The area of the product being currently analysed is displayed clearly to the user. The rest of the GUI is split into four areas, explained currently.



Model 15 SANVA2191 Description C286 CCM HI 2

Tests Passed

57290:green 151482:red
dfog_blank_leftcorner blank pass
rdfog_pres_leftcorner blank pass
Vision Pass Code Written

Tests Failed

Instruction

Take Product From Jig

Vision Inspection Status

Stage 1 PASS -----> Stage 2 PASS -----> Stage 3 PASS -----> Vision Test Result PASS

Vision Cycle Time 21 Sec

Electrical Inspection Status

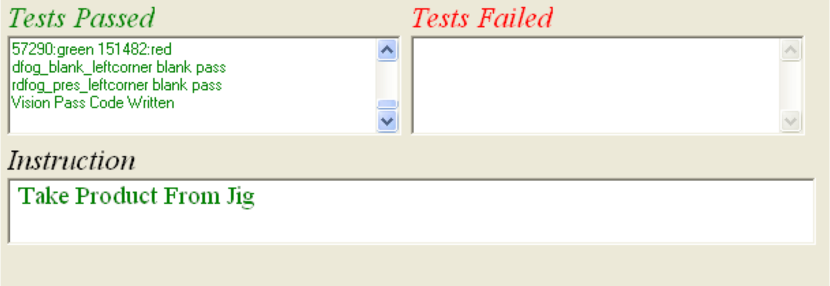
Voltage Check PASS -----> Sleep Check PASS -----> Bat Cal Check PASS -----> Electrical Test Result PASS

Electrical Cycle Time 15.2 Sec

Figure 4.3.2: Final front end GUI

1: Test Information and Instruction Palette

The test information and instruction palette is shown in figure 4.3.3 below. This conveys instructions to the operator and also displays all the passed and failed tests to the operator or technician/engineer.



Tests Passed

57290:green 151482:red
dfog_blank_leftcorner blank pass
rdfog_pres_leftcorner blank pass
Vision Pass Code Written

Tests Failed

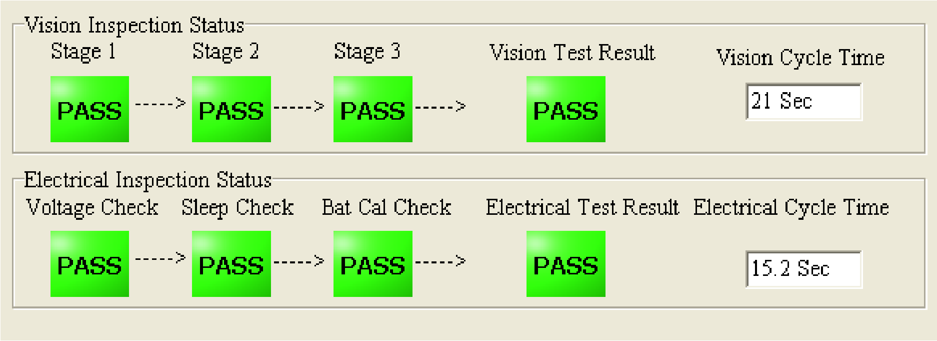
Instruction

Take Product From Jig

Figure 4.3.3: Test Information and Instruction Palette

2: Inspection Status Palette

Figure 4.3.4 below shows the inspection status palette. This informs the operator at what stage the test is at, and the result of that stage of the test. The cycle time is also displayed.



Vision Inspection Status

Stage 1: PASS -----> Stage 2: PASS -----> Stage 3: PASS -----> Vision Test Result: PASS

Vision Cycle Time: 21 Sec

Electrical Inspection Status

Voltage Check: PASS -----> Sleep Check: PASS -----> Bat Cal Check: PASS -----> Electrical Test Result: PASS

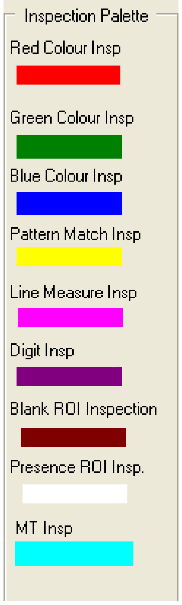
Electrical Cycle Time: 15.2 Sec

Figure 4.3.4: Inspection Status Palette

There is an electrical element included in the software application. This is due to a cycle time analysis performed on the production test. The cycle time of the EOL electrical test is seventy five seconds. This is the gating element (stage with longest cycle time) of the production cell. The cycle time of the Vision Test is twenty one seconds. The Vision Analysis test has the capability to perform some basic electrical testing, so some of the electrical tests are moved to the vision analysis stage. This reduces the gating item by fifteen seconds, and this cell balancing results in a significant cost saving for the company.

3: Inspection Type Palette

If a failure appears on the current image analysis area being processed, a coloured box is drawn around that failure. The inspection palette in figure 4.3.5 allows the user to understand the failure mode.



Inspection Palette

Red Colour Insp 🟥

Green Colour Insp 🟩

Blue Colour Insp 🟦

Pattern Match Insp 🟨

Line Measure Insp 🟪

Digit Insp 🟪

Blank ROI Inspection 🟫

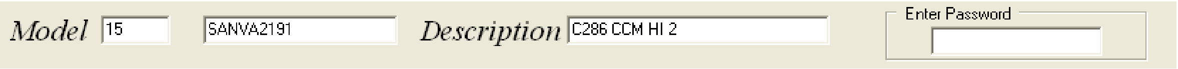
Presence ROI Insp. ⬜

MT Insp 🟦

Figure 4.3.5: Inspection Type Palette

4: Model Information Palette

The model information is also displayed to the operator. This includes the model number, the model part number and a description of the product. The password field for technical use is also displayed on the front GUI.



Model 15 SANVA2191 Description C286 CCM HI 2

Enter Password

Figure 4.3.6: Model Information Palette

4.3.3: Image Acquisition

The image is acquired using Common Vision Blox (CVB) methods from a CVB TCVDisplay component to a Borland image component. This is explained in section 3.3.2. This image is extremely large, and it is not feasible to process it all at once, as the area of inspection would not be visible to the operator. Due to this, the processing is split into 3 image areas.

4.3.4: Execution Sequence

The first area analysed takes 4 regions of interest, namely the four rotary knobs on the product and combined them into a single image for analysis. This is shown in figure 4.3.7.

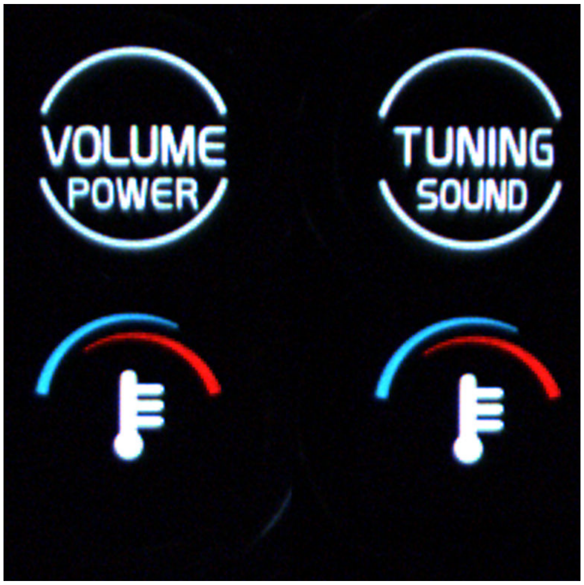


Figure 4.3.7: Image Analysis Area 1

The second area of inspection is shown in figure 4.3.8 below, and covers the digit keypad and the phone keypad.

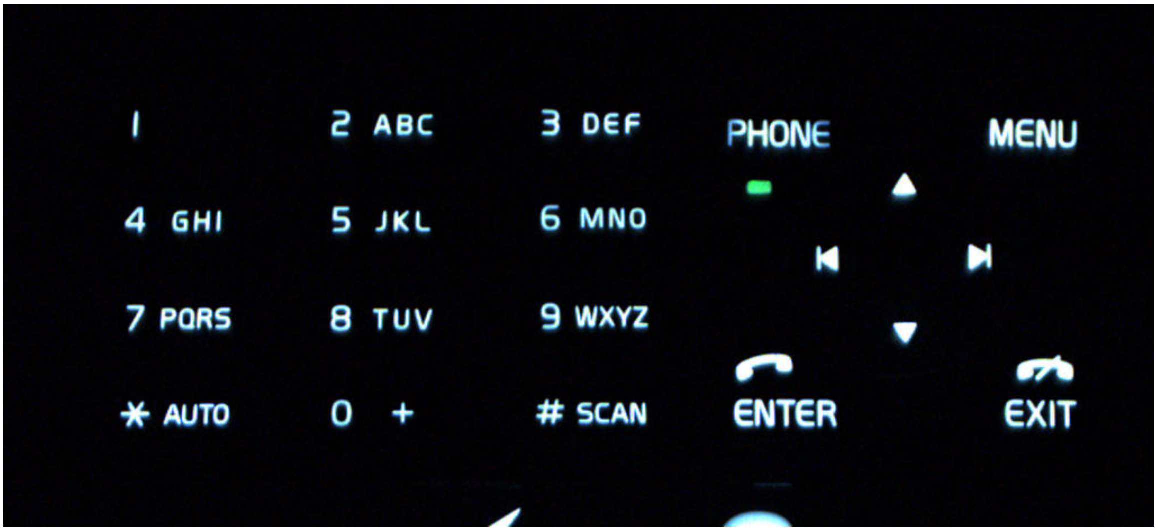


Figure 4.3.8: Image Analysis Area 2

The third area of inspection is shown in figure 4.3.9 below, and covers all the climatic elements of the module.

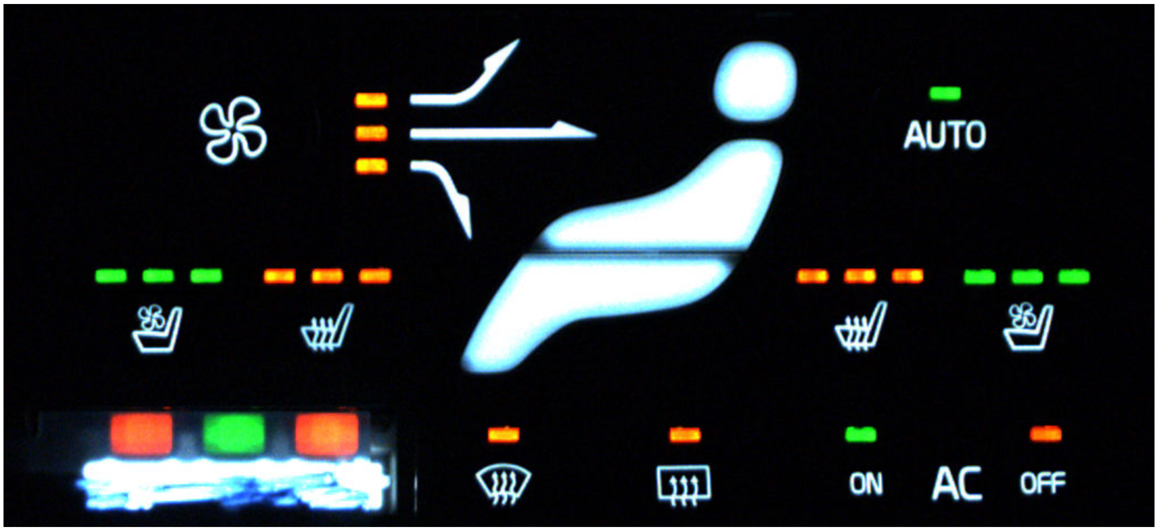


Figure 4.3.9: Image Analysis Area 3

4.3.5: Inspection Setup GUI

Figure 4.3.10 shows the final inspection setup GUI. This GUI is used by the technical person, be it an engineer or technician. It is designed to be simple to use and not to add any confusion for the user.

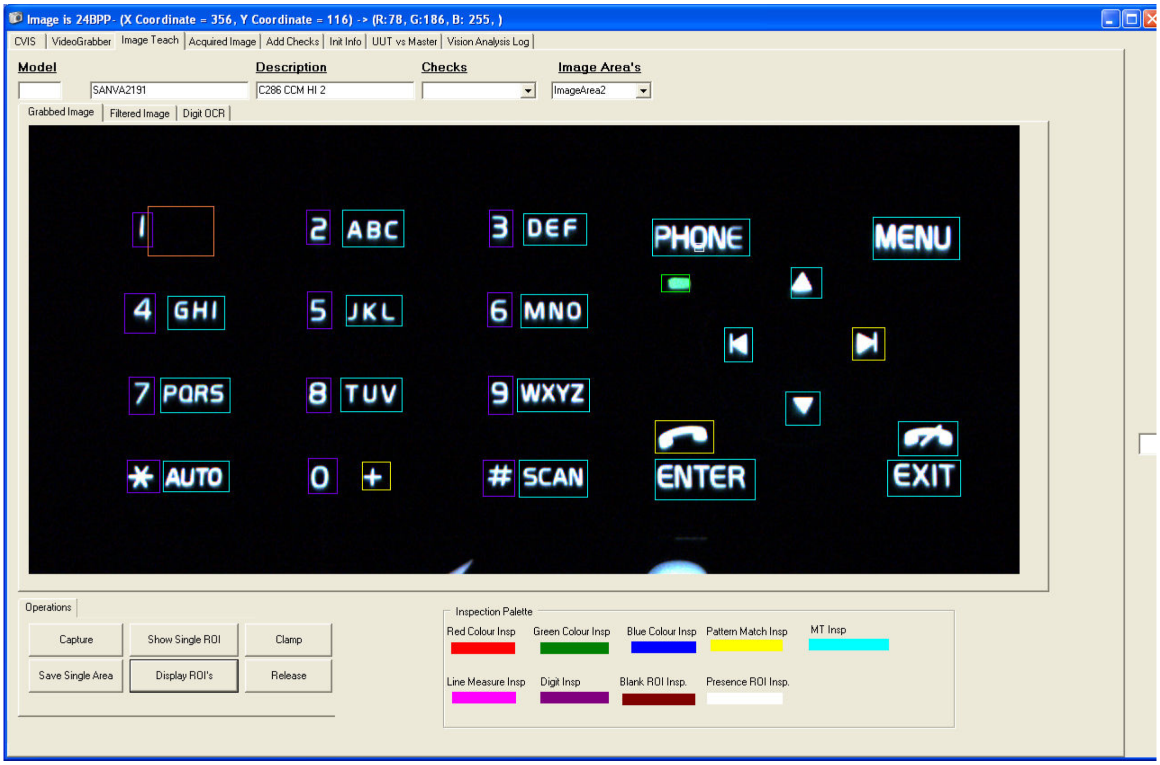


Image is 24BPP- (X Coordinate = 356, Y Coordinate = 116) -> (R:78, G:186, B: 255,)

Model: SANVA2191

Description: C286 CCM HI 2

Checks:

Image Area's: ImageArea2

Figure 4.3.10: Inspection Setup GUI

Once a correct password has been entered on the front end GUI, the 'ImageTeach' tab is displayed. Initially the grabbed image visible in figure 4.3.10 is blank. The user will have the information about the particular failure and what image area the failure has occurred within.

1: Image Information Display

The relevant information that the user needs to know about the image is displayed in the form caption, as shown in figure 4.3.11. The type of image, x and y coordinates of current pixel that the mouse is pointed at, and the RGB value of that pixel is displayed. This is done using a mouse event handler of the displayed image. Every time the cursor moves within the image, the main form caption is updated.

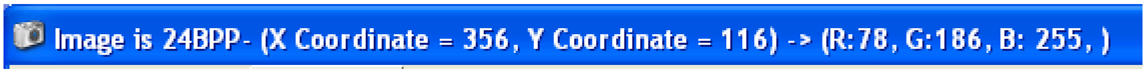


Image is 24BPP- (X Coordinate = 356, Y Coordinate = 116) -> (R:78, G:186, B: 255,)

Figure 4.3.11: Main Form Caption Bar

2: Model and Checks Description Palette (Change the image)

This part of the ImageTeach GUI displays model information and also has two interactive drop down boxes. The first is the 'Image Area's' which contains the three image area names. Depending on which element is selected, the relevant image area is loaded into the image area on the GUI and the checks relevant to that area are loaded into the 'Checks' drop down box. The check to be modified or set up then can be selected from this dropdown box.

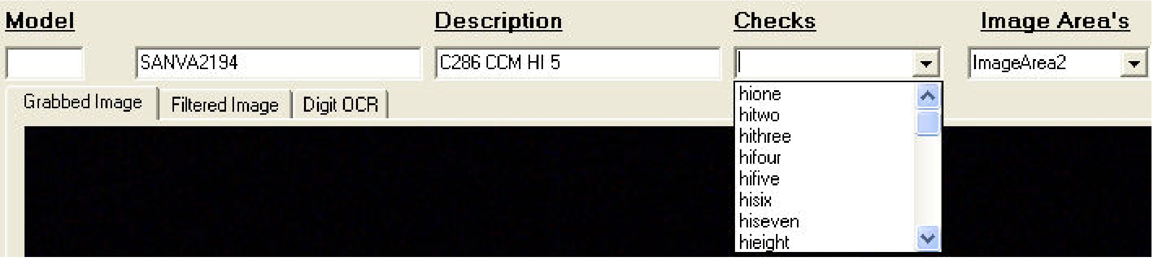

Model: SANVA2194
Description: C286 CCM HI 5
Checks:
Image Area's: ImageArea2

Figure 4.3.12: Model and Checks Description Palette

3: Operational Button Palette

In order for the user to perform tasks such as capture an image or save an inspection, a button palette shown in figure 4.3.13 below is used. Six operations are shown, and do exactly as named. The capture button will capture an image from the camera and load in the relevant image area into the GUI. The 'Save Single Area' button will save the focused check in the 'Checks' drop down menu to the ini file. The 'Show Single ROI' button will show the saved region of interest of the check focused in the 'Checks' drop down box on the image area displayed on the GUI. The 'Display ROIs' will display all the region of interest inspection areas on the image area on the GUI, as shown in figure 4.3.10 above.

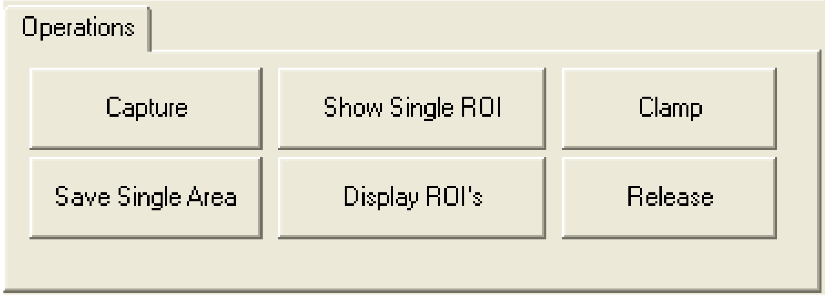


Figure 4.3.13: Operational Button Palette

4.3.6: Controlling Exposure at Customer

If in the event that an inspection is set up incorrectly, and a bad unit gets to the customer, it is a requirement to control the exposure that the customer has. Exposure in this context means to limit the stock that will need to be screened in the event of a customer return.

An inspection setup saving technique is put in place to counteract this. The folder used for keeping all the inspection data is called 'VisionIni'. This is where the initial master setup data is kept. There is also a mirror folder called 'TempVisionIni'. If a change is made to any of the inspections, the inspection is saved to this 'TempVisionIni' folder. This folder is now used for the remainder of that day of production. Once the system is started the next day, the system reverts back to use the master 'VisionIni' file.

The exposure then for any incorrect setup of an inspection is limited to one day. A list of affected serial numbers can be given to the customer, limited to a maximum of 444 units (a day's production). Without this method, all stock since the change happened would need to be screened.

4.3.7: Software Problems Encountered

1: Loading camera settings file: The settings file for the camera is originally loaded at run time in Borland. This was causing problems as the camera was not found when the software was switched from a laptop using a PCIMCIA card to a desktop with a PCI card. It was after looking at the AVT Stingray datasheet that it became apparent that the settings file can be saved to the camera flash memory.

2: Using a Thread for execution method: Initially a thread was used. This caused a lot of problems due to the amount of image processing that was taking place in the thread. There seemed to be a lot of memory allocation problems. A significant amount of time was allocated to fixing this problem, but there was no success. So, the application is now running correctly within a timer.

3: Selecting Model: A significant burden is that during the initial design of the system, the model selection process is done by the operator using a menu system. This is not ideal, as there is opportunity for error. This is no longer the case. The EOL electrical program has an automatic start up whereby the model is selected automatically using a manufacturing order scanning system. As the vision system is placed next to this, the systems are linked via a labjack I/O card. A labjack is a USB/Ethernet based measurement and automation device that provides analog and digital inputs and outputs. It serves as an inexpensive and easy to use interface between computers and the physical world [9]. The labjack is controlled from the EOL, and depending on which model is being run at the EOL test, a binary sequence of five outputs are set. These five outputs are then monitored using the Alps USB I/O in the vision system. Once the model is changed in the EOL, the vision system is automatically set up correctly, thus eliminating operator error in regard to model selection.

4.4: Conclusions

The fourth chapter described the system implementation of the project. It started by describing the implementation stage of the hardware, showing the final system diagram, and the jig implementation. The problems encountered are then discussed. Finally the software implementation is presented. This discusses the key elements of the software such as using the camera, the final GUI, inspection setup along with other software elements.

The next chapter outlines the extensive testing and analysis of the system. It starts by describing the detection capability testing, where we ascertain what sort of failure detection we have with the system. The reliability testing of the system is then discussed, where we determine how reliable the system is in the production environment.

Chapter 5: Testing and Analysis

5.1: Introduction

The fifth chapter outlines the testing and analysis of the system. It starts by describing the detection capability testing, where we ascertain what sort of failure detection we have with the system. The reliability testing of the system is then discussed, where we determine how reliable the system is in the production environment.

5.2 Detection Capability Testing

5.2.1: Overview of Detection Capability Testing

Prior to releasing the system to the production floor, an essential step is to determine the detection capability of the system. Detection capability testing is separated into 8 areas:-

- MT Check on Character Strings
- LED/Cosmetic Colour Inspection
- Mixed Knobs
- Incorrect Knobs
- Incorrect Laser Etching
- LED Electrical Function
- Digit Check
- Poor Graphic Position

5.2.2: MT Pattern Match Check

The MT pattern match check is performed on the character graphics on the unit. The first test set presented here is that of the characters on the keypad as shown in figure 5.2.1 below. An MT pattern match check is performed on the letters of the 12 buttons, only on the characters.

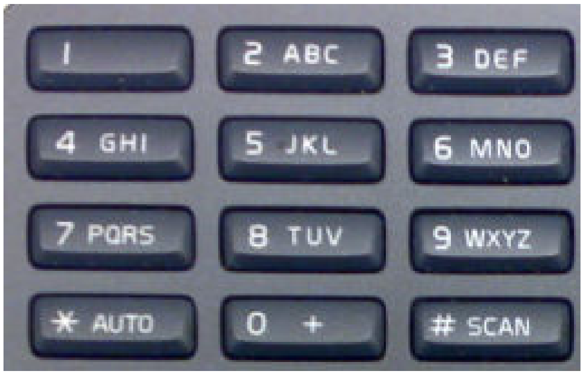


Figure 5.2.1: Number Pad

Table 5.2.1 below shows the results of the detection capability testing. For the diagonal, where the tests pass, the value in brackets is the average pass value. All measurements are rounded to the nearest integer.

Key:

P = Passed Test
F = Failed Test
Insp = Inspection
Spec = Specification Limit. Test will fail above this value.

Insp (spec)	**abc**	**def**	**ghi**	**jkl**	**mno**	**pqrs**	**tuv**	**wxyz**	**auto**	**scan**	+
abc (12)	P (3)	F (77)	F (86)	F (95)	F (86)	F (180)	F (102)	F (210)	F (192)	F (184)	F (633)
def (12)	F (80)	P (2)	F (62)	F (84)	F (71)	F (164)	F (91)	F (182)	F (174)	F (165)	F (590)
ghi (12)	F (72)	F (106)	P (4)	F (81)	F (99)	F (171)	F (88)	F (164)	F (201)	F (189)	F (712)
jkl (12)	F (102)	F (91)	F (88)	P (5)	F (104)	F (168)	F (115)	F (216)	F (222)	F (241)	F (665)
mno (12)	F (79)	F (84)	F (91)	F (64)	P (2)	F (154)	F (69)	F (189)	F (192)	F (179)	F (515)
pqrs (12)	F (165)	F (154)	F (145)	F (187)	F (192)	P (2)	F (84)	F (67)	F (75)	F (95)	F (806)
tuv (12)	F (69)	F (84)	F (57)	F (84)	F (74)	F (117)	P (4)	F (187)	F (167)	F (189)	F (492)
wxyz (12)	F (172)	F (189)	F (165)	F (168)	F (164)	F (89)	F (154)	P (3)	F (91)	F (87)	F (854)
auto (12)	F (165)	F (151)	F (184)	F (175)	F (161)	F (84)	F (171)	F (71)	P (4)	F (84)	F (745)
scan (12)	F (181)	F (169)	F (148)	F (175)	F (150)	F (75)	F (136)	F (117)	F (98)	P (2)	F (684)
+ **(12)**	F (741)	F (657)	F (625)	F (578)	F (764)	F (902)	F (632)	F (954)	F (871)	F (936)	P (3)

Table 5.2.1: Number Pad MT Pattern Match Results

The second test set is associated with the phone keypad, as shown in figure 5.2.2 below. After discussions with the paint and laser division of Alps, it is deemed not possible for the ‘up down skip left or skip right’ buttons to be laser etched on the positions of the four large knobs, phone, menu enter and exit, and visa versa. This is because the centre circular button is one pattern laser etched at once, and the area of the graphic is larger than the area of the four outside buttons. The detection testing is separated for this phone pad. Tables 5.2.2 and 5.2.3 below show the results of the MT detection capability testing for the phone pad.

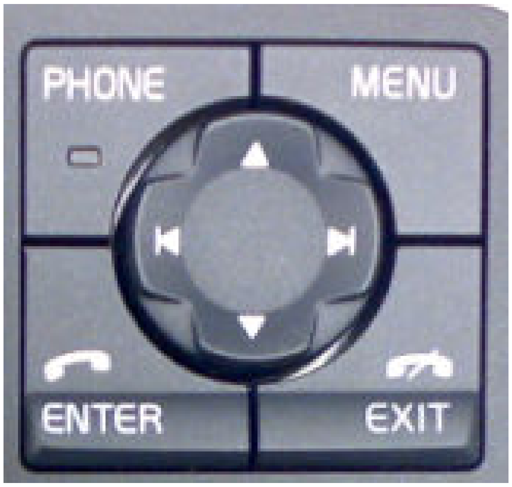


Figure 5.2.2: Phone Pad

Key:

Spec = Specification Limit. Test will fail above this value.

Inspection (Spec)	**Phone**	**Enter**	**Exit**	**Menu**	**MyKey**
Phone (12)	Pass (3)	Fail (77)	Fail (86)	Fail (95)	Fail (141)
Enter (12)	Fail (80)	Pass (1)	Fail (62)	Fail (84)	Fail (122)
Exit (12)	Fail (72)	Fail (106)	Pass (2)	Fail (81)	Fail (85)
Menu (12)	Fail (102)	Fail (91)	Fail (88)	Pass (5)	Fail (102)
MyKey (12)	Fail (79)	Fail (84)	Fail (91)	Fail (64)	Pass (3)

Table 5.2.2: Phone Menu MT Pattern Match Results

Inspection (Spec)	**Up**	**Down**	**Left**	**Right**
Up (12)	Pass (3)	Fail (84)	Fail (91)	Fail (64)
Down (12)	Fail (165)	Pass (4)	Fail (145)	Fail (187)
Left (12)	Fail (69)	Fail (84)	Pass (4)	Fail (84)
Right (12)	Fail (172)	Fail (189)	Fail (165)	Pass (2)

Table 5.2.3: Direction MT Pattern Match Results

5.2.3: LED/Cosmetic Colour Inspection

There are two colours of LED's on this module, amber and green. In order to verify that these are detected, it is necessary to incorrectly mount the wrong colour LED on the PCB, and check for incorrect colour. This is shown in figure 5.2.3 below. All other LED's fail in a similar fashion.

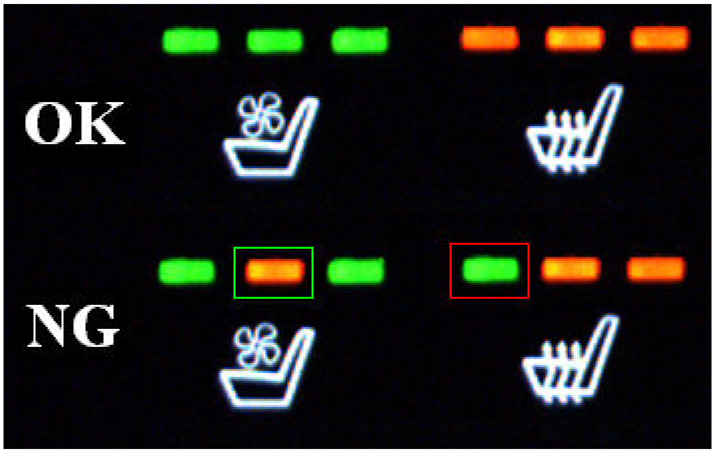


Figure 5.2.3: LED Failure Detection

5.2.4: Mixed Knobs

The detection testing of mixed knobs on the keypad is discussed in section 5.2.2, and will not be considered here. The only other possibilities for lack of poke yoke are discussed presently. The line distance measurement algorithm is used to detect these failures.

DFog Knobs: One possibility for mixed knobs is on the front and rear dfog buttons as shown below in figure 5.2.4. The failure results are shown in table 5.2.4.

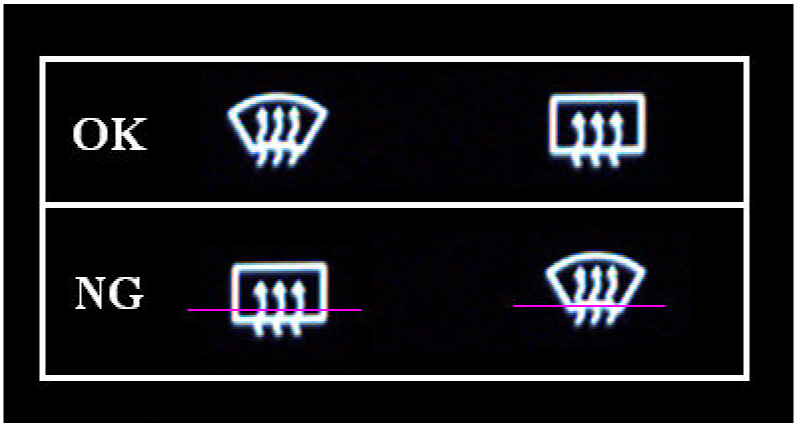


Figure 5.2.4: DFog and Rear DFog Failure Detection

Inspection (Specification)	DFog	Rear DFog
DFog (Distance = 37)	Pass	Fail (Distance = 50)
Rear DFog (Distance = 50)	Fail (Distance = 37)	Pass

Table 5.2.4: Mixed Dfog Knobs Distance Results

Seat Knobs: Another possibility for mixed knobs is on the seat heat and seat fan buttons as shown in figure 5.2.5 below. The failure results are shown in table 5.2.5.

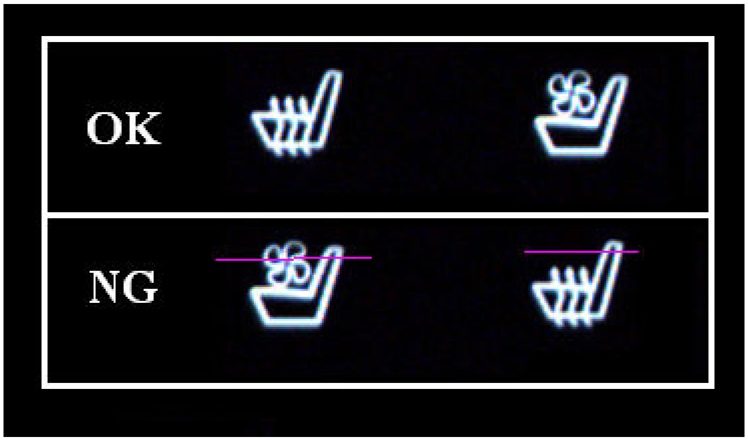


Figure 5.2.5: Seat Heat and Seat Fan Failure Detection

Inspection (Specification)	Seat Heat	Seat Fan
Seat Heat (Distance = 8)	Pass	Fail (Distance = 19)
Seat Fan (Distance = 19)	Fail (Distance = 8)	Pass

Table 5.2.5: Mixed Heat Knobs Distance Results

5.2.5: Incorrect Knobs

The Y20 CCM module had fourteen different model varieties, and the combination of buttons on each is different. There are possibilities for knobs that are poka-yoke, but that there are two buttons of similar size and shape, but of different cosmetic appearance. The two possibilities for this are shown in figure 5.2.6 below.

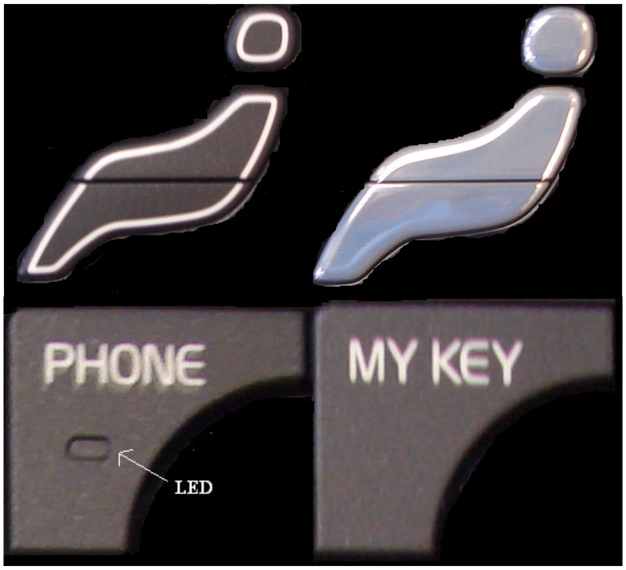


Figure 5.2.6: Man, Phone and MyKey Knobs

Inspection (Specification)	**Man Outline**	**Man Full**
Man Outline (Threshold = 80)	Pass	Fail (Match= 6%)
Man Full (Threshold = 80)	Fail (Match= 11%)	Pass

Table 5.2.6: Man Pattern Match Results

Inspection (Specification)	**Phone**	**My Key**
Phone (Threshold = 80)	Pass	Fail (Match= 6%)
MyKey (Threshold = 80)	Fail (Match= 11%)	Pass

Table 5.2.7: Phone and MyKey Pattern Match Results

5.2.6: Incorrect Laser Etching

Incorrect laser etching is anything that deviates from what is expected. This type of failure is inherent in the other detection capabilities discussed presently. For example, a ‘5’ number keypad laser etched as an ‘8’ pattern is covered by the digit check, and is the same as mis-assembly of the knobs.

5.2.7: LED Electrical Function

As discussed previously in section “3.1.3 Lighting”, the backlighting provides the illumination for the buttons. Apart from providing nice even illumination, it has an inherent testing function. If any of the LED’s are not lighting, perhaps due to lack of solder or a smashed LED bulb, the button above this LED will not function, and thus fail any subsequent test. This is extremely advantageous, as on the schematic of the button, there are a number of LED’s controlled by a single line. If one of these LED’s is not functioning, it is difficult to pick up electrically.

5.2.1: Digit Check

All digit keys can be interchanged, giving a failure set of 144. Table 5.2.8 shows a complete detection capability test set.

Insp.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**0**	*	#
1	Pass	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail
2	Fail	Pass	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail
3	Fail	Fail	Pass	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail
4	Fail	Fail	Fail	Pass	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail
5	Fail	Fail	Fail	Fail	Pass	Fail	Fail	Fail	Fail	Fail	Fail	Fail
6	Fail	Fail	Fail	Fail	Fail	Pass	Fail	Fail	Fail	Fail	Fail	Fail
7	Fail	Fail	Fail	Fail	Fail	Fail	Pass	Fail	Fail	Fail	Fail	Fail
8	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Pass	Fail	Fail	Fail	Fail
9	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Pass	Fail	Fail	Fail
0	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Pass	Fail	Fail
*	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Pass	Fail
#	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Pass

Table 5.2.8: Digit Check Detection Results

5.2.8: Poor Graphic Position

There is a tolerance for graphic movement on the button itself. This is defined by the ROI area that is specified at system setup in accordance with quality specification. This specification varies for the button type and size. Two instances of poor graphic position are shown in figure 5.2.7 below.

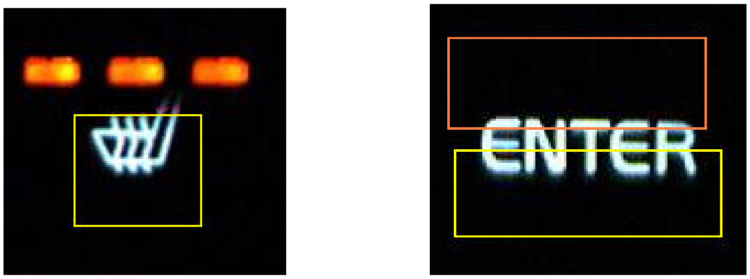


Figure 5.2.7: Poor Graphic Position Detection

5.2.9: Detection Capability Results

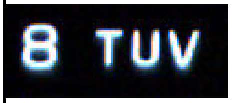
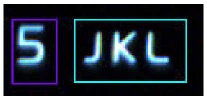
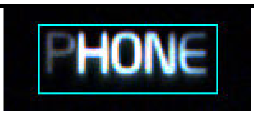
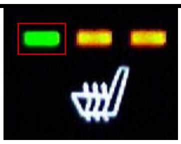
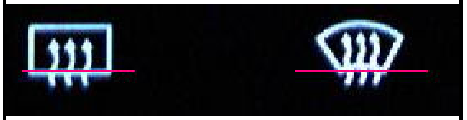
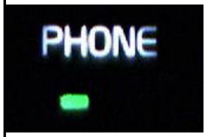
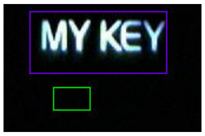
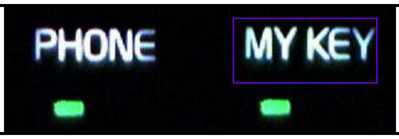
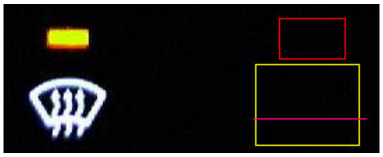
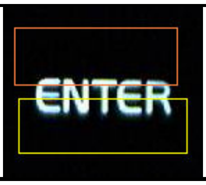
Inspection Type	Example	Detection
Digit Check	 	100%
MT Check		95%
LED/Cosmetic Colour Insp.		100%
Mixed Knobs		100%
Incorrect Knobs	 	100%
Incorrect Laser Etching		100% *
LED functioning OK		100%
Poor Graphic Position		100% *

Table 5.2.9: Detection Capability Results 1

*** = of samples tried**

5.2.10: Discussion of Results

The system shows excellent detection capability, with an overall detection capability of ~100%. Two elements, incorrect laser etching and poor graphic position show 100% detection of the samples tried. It is not practical to try every conceivable laser etching mistake, as one would need to not only consider the graphics on this unit, but also all graphics in the paint and laser division, which stands at well over one thousand. This would mean over one million combinations.

The MT Pattern Match method attained a detection capability of 95%. This method did not reach 100% of what was expected of the algorithm. The method had difficulty distinguishing some of the numbers, such as the'5' and the '8' digits. These digits are inspected already using the digit check algorithm. Some of the other graphics caused spurious results in regard to the Mahalanobis Distance (MD) value attained. These graphics are inspected using one of the other algorithms described earlier. More time is necessary to re-visit this algorithm and try to optimise the product inspection coverage it can handle. One idea is to increase the unit space from 200 to a larger number and monitor the results. It is important to note that of the MT inspections that are performed, as outlined in section 5.2.2, a detection rate of 100% is achieved.

5.3: Reliability Testing

5.3.1: Overview of Reliability Testing

Reliability testing is carried out prior to the system being released onto the production floor. The reason for reliability testing in this case is two fold:-

1: To ensure the system performs reliably.
2: To iron out any potential false negatives or false positives that occur.

It is imperative that the system is released into the production environment in an extremely reliable state. Not only does lack of reliability cause unnecessary downtime, it also leads to lack of confidence in the system with the user. This stage of testing also allows the developer to see any false negatives or false positives that occur, and to correct the system accordingly.

5.3.2: Receiver Operator Curve (ROC) Testing Method [6]

Recevier Operator Curve (ROC) methods are used to determine how reliable the system performs at discriminating good parts from bad. The ROC curve is represented by plotting the true positive rate or sensitivity, versus the false positive rate or (1-specitivity). This system is a two class prediction problem, with one class called a positive (p) and one class called a negative (n).

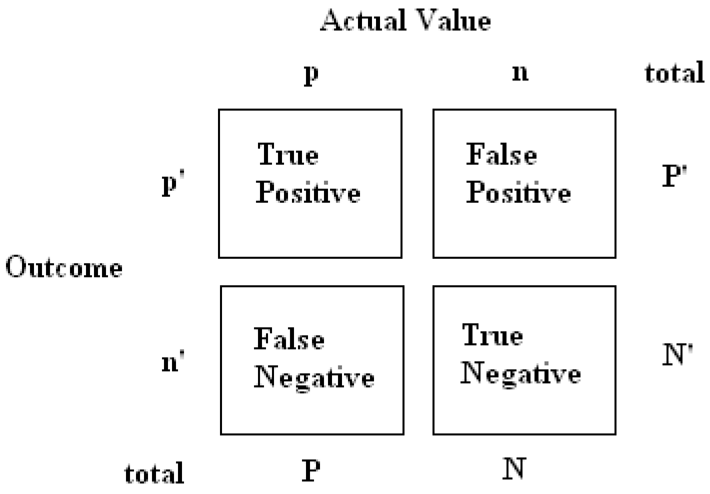
At the outset, there are four parameters:-

●**True Positive:** If the outcome of the test is positive (p) and the actual result is positive (p)

●**False Positive:** If the outcome of the test is positive (p) and the actual result is negative (n).

●**True Negative:** If the outcome of the test is negative (n) and the actual result is negative (n).

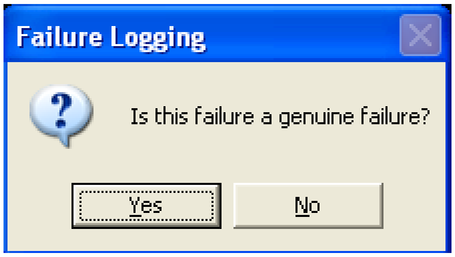
●**False Negative:** If the outcome of the test is negative (n) and the actual result is positive (p).



		Actual Value		
		p	n	total
Outcome	p'	True Positive	False Positive	P'
	n'	False Negative	True Negative	N'
total		P	N	

Figure 5.3.1: Receiver Operator Curve Matrix [7]

During program run time, the quantities of the above four parameters are updated as we test. Any time an inspection passes, the true positive is incremented. The false positives are flagged by production, and a record of them is formed. Customer returns for units that pass this system are also recorded. The operator distinguishes true negatives from false negatives. Any time a unit fails, the program presents a message box to the operator asking them is it a genuine failure. This message box is shown in figure 5.3.2 below.

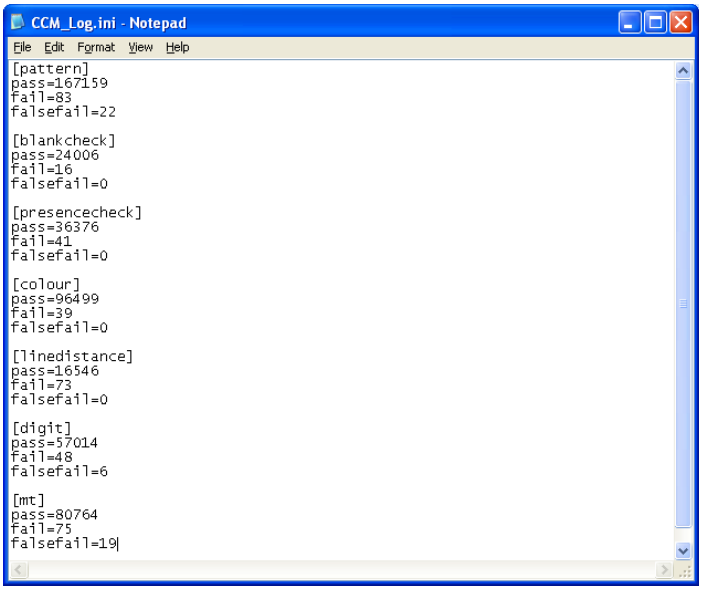


Failure Logging

Is this failure a genuine failure?

Figure 5.3.2: Failure Logging Message Box

If the operator clicks on ‘Yes’ then the number of true negatives is incremented. If ‘No’ is clicked, then the number of false negatives is incremented. The three parameters are automatically stored in an ini file called ‘CCM_Log.ini’. Figure 5.3.3 below shows the contents of the ini file.



```
[pattern]
pass=167159
fail=83
falsefail=22

[blankcheck]
pass=24006
fail=16
falsefail=0

[presencecheck]
pass=36376
fail=41
falsefail=0

[colour]
pass=96499
fail=39
falsefail=0

[linedistance]
pass=16546
fail=73
falsefail=0

[digit]
pass=57014
fail=48
falsefail=6

[mt]
pass=80764
fail=75
falsefail=19
```

Figure 5.3.3: Failure Logging ini file

5.3.3: Reliability Results

The ROC analysis is performed twice, once for the pre production trial stage and then a second time whereby the results are based on a 2 week production spell. The results are outlined below.

Formulae:

TP = True Positive
FP = False Positive
TN = True Negative
FN = False Negative

True Positive Rate = TP / (TP + FN)
False Positive Rate $= FP / (FP + TN)$
Accuracy = (TP + TN) / (P + N)
Specificity = 1 − FPR
Positive Predictive Value = TP / $(TP + FP)$
Negative Predictive Value = TN / $(TN + FN)$
False Discovery Rate *FDR* $= FP / (FP + TP)$

Testing Phase:

Inspection	Insp. Quantity	TP	FP	TN	FN
Pattern	23056	22945	5	84	22
Blank	3095	3074	0	16	5
Presence	4447	4431	0	11	5
Colour	13112	13100	0	10	2
Line Distance	2795	2784	0	8	3
Digit	8412	8386	0	20	6
MT	1403	1379	0	18	6

Table 5.3.1: Testing Phase ROC Data 1

Inspection	TPR	FPR	Accuracy	Specificity	PPV	NPV	FDR
Pattern	0.9992	0.0561	0.9988	0.9438	0.9991	0.7924	0.0002
Blank	0.9983	0.0588	0.9981	0.9411	0.9994	0.7615	0.0003
Presence	0.9991	0.0831	0.9986	0.9166	0.9996	0.6875	0.0002
Colour	0.9994	0.0909	0.9997	0.9090	0.9992	0.8333	0.0000
Line Distance	0.9985	0.0001	0.9989	1.0000	1.0000	0.7272	0.0001
Digit	0.9991	0.0049	0.9992	1.0000	1.0000	0.7692	0.0001
MT	0.9956	0.0575	0.9950	0.9473	0.9994	0.7500	0.0007

Table 5.3.2: Testing Phase ROC Data 2

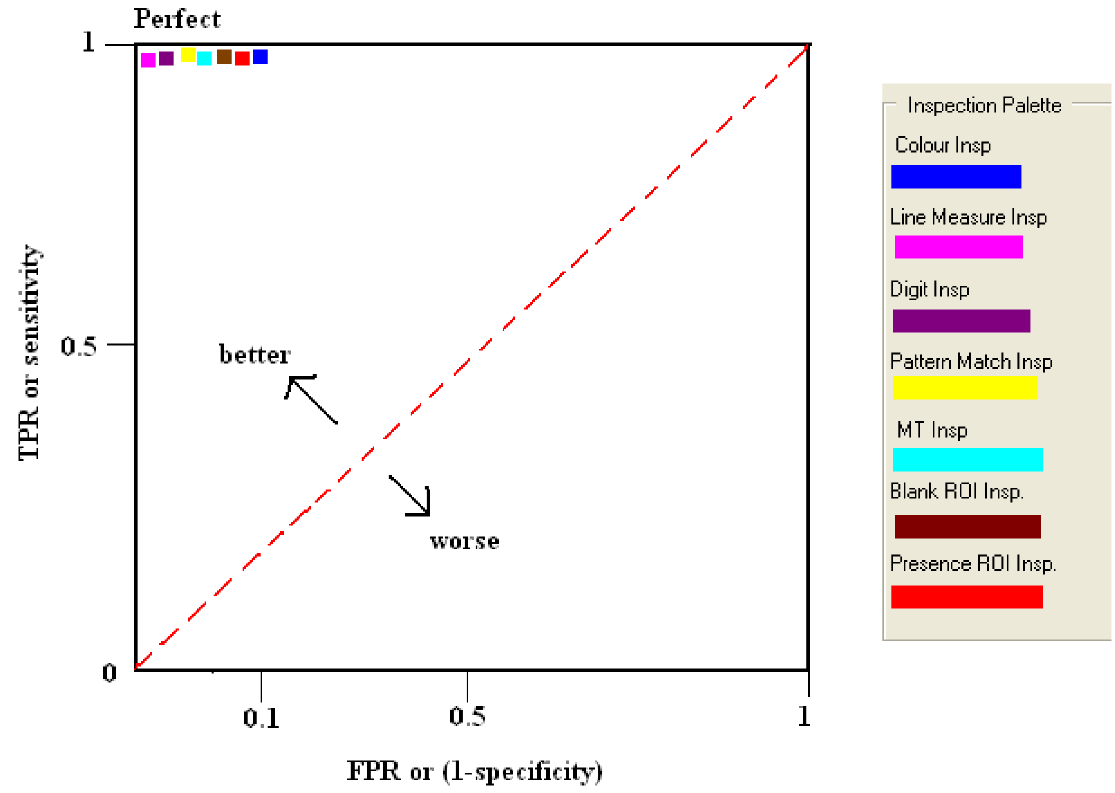


Figure 5.3.4: Testing Phase ROC Curves

Production Phase:

Inspection	Insp. Quantity	TP	FP	TN	FN
Pattern	167266	167159	3	83	22
Blank	24022	24006	0	16	0
Presence	36417	36376	0	41	0
Colour	96538	96499	0	39	0
Line Distance	16619	16546	0	73	0
Digit	57068	57014	0	48	6
MT	80858	80764	0	75	19

Table 5.3.3: Production Phase ROC Data 1

Inspection	TPR	FPR	Accuracy	Specificity	PPV	NPV	FDR
Pattern	0.999	0.013	0.999	0.988	0.999	0.790	0.0000058
Blank	0.999	0.064	0.999	0.941	0.999	0.888	0.000041
Presence	0.999	0.026	0.999	0.976	0.999	0.995	0.000027
Colour	0.999	0.027	0.999	0.975	0.999	0.975	0.00001
Line Distance	0.999	0.014	0.999	0.986	0.999	0.973	0.000060
Digit	0.999	0.022	0.999	0.975	0.999	0.761	0.000017
MT	0.999	0.014	0.999	0.986	0.999	0.797	0.000012

Table 5.3.4: Production Phase ROC Data 2

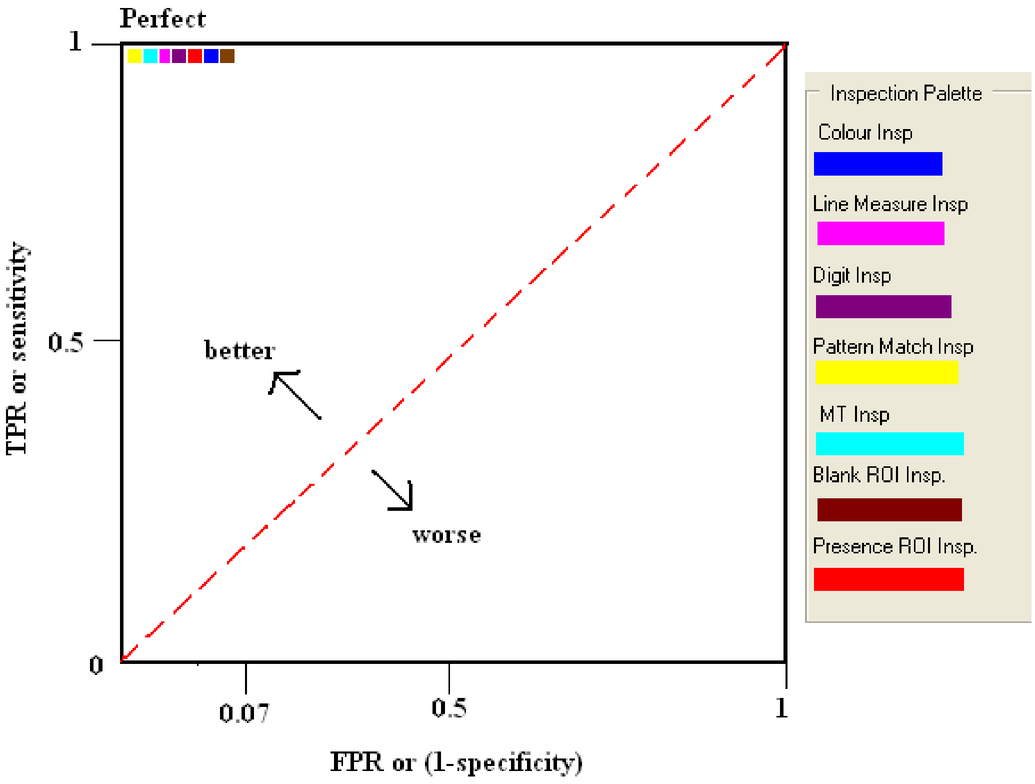


Figure 5.3.5: 2 Week Production Phase ROC Curves

5.3.4: Discussion of Results

The system shows excellent reliability through both the testing stage and for the first two weeks of using the system in the production cell. An average reliability of 99.99% is attained for each of the inspection sets. A slight increase in reliability is found in the 2 week production phase. This is because it is based on the finished production system, with a larger test set.

The inspection quantities vary for each inspection type as there is a varying number of each inspection per unit. For example, there are more colour inspections per module than line distance measurements. The system shows to have 99.99% accuracy for all inspections.

False positives occurred in both the testing and production phases for the pattern match algorithm. This is due to similar white counts of different graphics, leading to higher than expected percentage matches. This was resolved by removing the white count element from these particular inspections, thus only performing pixel matching element of classical pattern match algorithm.

5.3.5: Downtime Analysis

Extensive testing of the system is performed to try to eliminate even initial downtime of the system. From a company viewpoint, if the system is down, it is costing the company money. The company is devoting a large portion of resources in reducing the downtime of the production cells. Any contributor to the overall downtime is flagged immediately to management level.

The system performed very well in the first 3 weeks of production, with only 20 minutes downtime from a production time of 180 hours, a percentage downtime of 0.001%. This is a very satisfying result.

5.4: Conclusions

The fifth chapter outlined the extensive testing and analysis of the system. It started by describing the detection capability testing, where we ascertain what sort of failure detection we have with the system. The reliability testing of the system is then discussed, where we determine how reliable the system is in the production environment.

The next chapter describes the final conclusions drawn from the system and also discusses proposed future work.

Chapter 6: Conclusions and Future Work

6.1: Final Conclusions

The aim of this project was to develop a fully functional machine vision system for the Alps Y20 CCM product. This aim was achieved. A complete hardware and software solution was designed and implemented. This system is running on the production cell for the past four weeks (19th of June 2009 – 17th of July 2009).

This system represents a significant advancement in Alps outgoing inspection capability. It has enhanced Alps quality reputation by meeting the customer's requirement for quality. The idea going forward is to use this technology to strengthen our detection capability on other products in Alps, and to interact with other Alps plants worldwide in the hope of forming an Alps worldwide vision system group.

Appendix H shows some questionnaire results taken from four people, two operators, a technician and an engineer. The system received positive feedback in all four, and this is important for any future implementations. However, there were some very good suggestions to enhance the system. These are currently being considered.

There is room for improvement in the system, as outlined in the 'Future work' section. The idea is to use this system as a benchmark within Alps, and to build on it with the addition of newly developed algorithms, tighter control and improved data logging.

In Summary:

Detection Capability
The system shows excellent detection capability, with an overall detection capability of ~100%.

Reliability
The system shows excellent reliability through both the testing stage and for the first two weeks of using the system in the production cell. An average reliability of 99.99% is attained.

Downtime
The system performed very well in the first three weeks of production, with only twenty minutes downtime from a production time of 180 hours, a percentage downtime of 0.001%. This is a very satisfying result.

Cycle Time
The cycle time of the finished solution is twenty one seconds for the vision test, with an extra thirteen seconds for an electrical test. This is well within the gating item cycle time of seventy seconds.

6.2: Future Work

1: Future Installation

● Install the same system into the second Y20 CCM cell. The target for completion is 28th of August.

● Install a similar system into the Y283 CCM cell. The target for completion is 7th of September.

● This system is custom to the dimension of the Y20 CCM module. It could be easily modified to suit other Alps products. This will result in greater levels of quality within the organisation.

2: Image Acquisition Method:

Section entitled '3.2.2: Common Vision Blox – Image Manager' discusses how the image is acquired from the AVT Stingray camera. The image is acquired in full, and the relevant image areas are cropped out of this image and processed. This is not a fast operation, taking up 10 seconds of the cycle time of 21 seconds. Cycle time is not a factor, as the allotted cycle time is 40 seconds, and the test is performing well within this limit. But the intended method is to directly access the raw image data from the camera, thus eliminating the need to acquire the image to memory and then select various regions.

3: Future Algorithm Development:

During production testing of the system, a false positive occurred, which is shown in table 6.2.1 below. The tuning knob's graphic is slightly twisted, and would be deemed a failure. This is a type of failure not in the original quality specification, but it is essential to pick up this type of failure. None of the current algorithms of the system will detect this failure, and a new algorithm is needed. This is in line with a suggestion put forward by the senior product engineer in Appendix G.

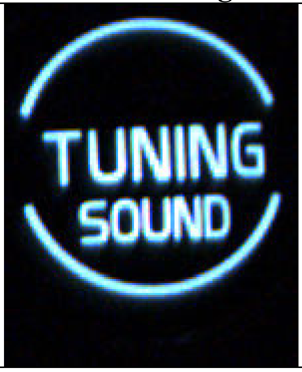
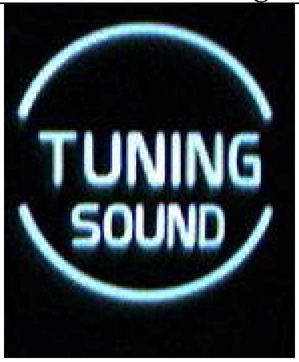
Failure: Tuning Knob Graphic Twisted	
Root Cause of Failure: Improper jigging set up in paint and laser process	
Failure Image	**Correct Pass Image**
	

Table 6.2.1: Twisted Graphic on Tuning Knob

References

[1] Machine Vision Algorithms in Java: Techniques and Implementation: Paul F Whelan, Derek Molloy.

Accessed: 11 June 2009

[2] Wikipedia description of poka-yoke.

http://en.wikipedia.org/wiki/Poka-yoke

Accessed: 01 July 2009

[3] Wikipedia description of CAN topology.

http://en.wikipedia.org/wiki/CAN-bus

Accessed: 30 June 2009

[4] AVT Stingray Datasheet.

http://www.kamery.sk/fileadmin/datasheets/Stingray_DataSheet_F201BC_fiber_V2.1.1_en.pdf

Accessed: 03 July 2009

[5] Wikipedia description of ROC Curves.

http://en.wikipedia.org/wiki/Receiver_operating_characteristic

Accessed: 03 July 2009

[6] Alps Document on MT Methods.

Protected by Copyright. MT system (outline) .ppt

Accessed: 11 June 2009

[7] Alps Document on MT Calculation.

Protected by Copyright. MT calculation example (English).doc

Accessed: 11 June 2009

[8] How to create a Common Vision Blox application with C++ Builder.

PC based application supplied with Common Vision Blox

Accessed: 17 May 2009

[9] USB Labjack U12

http://www.labjack.com/labjack_u12.php?prodId=27

Accessed: 20 July 2009

Appendix

Note: Code is available on disk at the discretion of Alps Electric Irl Ltd. There are over fourteen thousand lines of code. It is not practical to include this code in the appendix.

Appendix A: SolidWorks Drawing of the final Jig

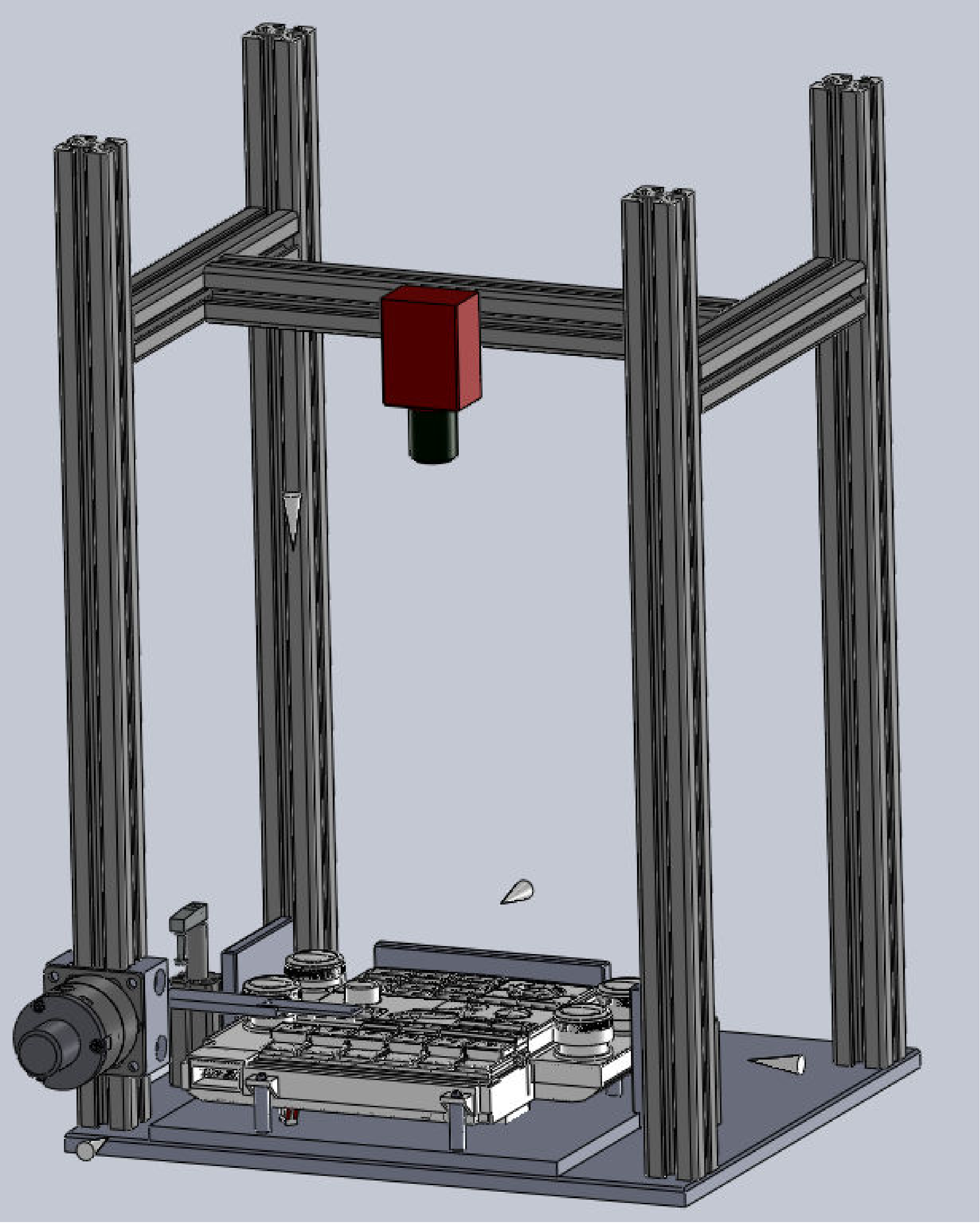
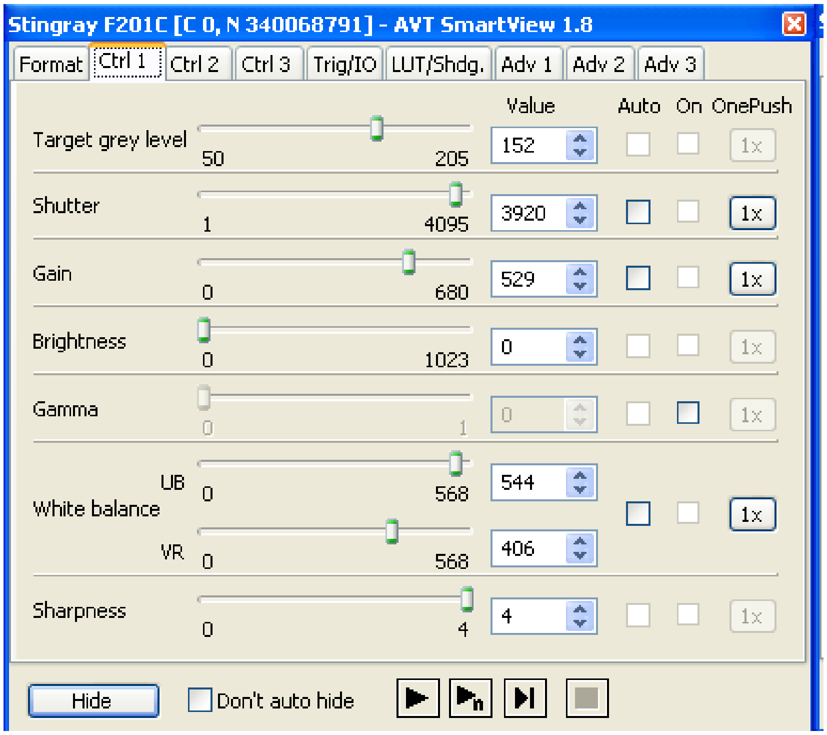


Figure A.1: SolidWorks Drawing of final Jig

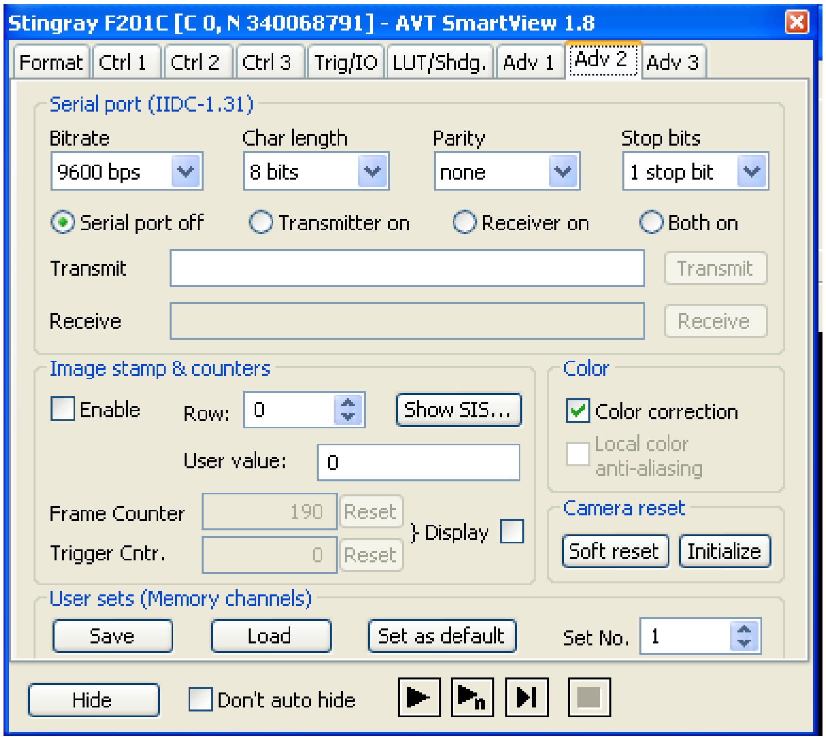
Appendix B: AVT Stingray Settings



Stingray F201C [C 0, N 340068791] - AVT SmartView 1.8

			Value	Auto	On	OnePush
Target grey level	50	205	152	□	□	1x
Shutter	1	4095	3920	□	□	1x
Gain	0	680	529	□	□	1x
Brightness	0	1023	0	□	□	1x
Gamma	0	1	0	□	□	1x
White balance UB	0	568	544	□	□	1x
White balance VR	0	568	406			
Sharpness	0	4	4	□	□	1x

Figure B.1: Stingray Settings 1



Stingray F201C [C 0, N 340068791] - AVT SmartView 1.8

Serial port (IIDC-1.31)

Bitrate	Char length	Parity	Stop bits
9600 bps	8 bits	none	1 stop bit

● Serial port off ○ Transmitter on ○ Receiver on ○ Both on

Transmit

Receive

Image stamp & counters

□ Enable Row: 0

User value: 0

Frame Counter 190

Trigger Cntr. 0

} Display □

Color

☑ Color correction

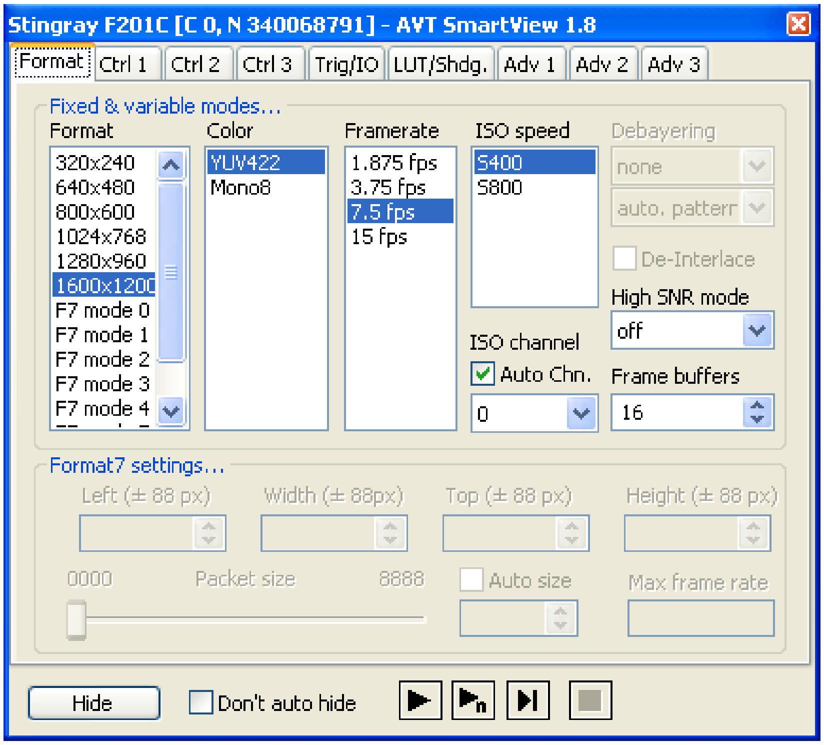
□ Local color anti-aliasing

Camera reset

User sets (Memory channels)

Set No. 1

Figure B.2: Stingray Settings 2



Stingray F201C [C 0, N 340068791] - AVT SmartView 1.8

Format | Ctrl 1 | Ctrl 2 | Ctrl 3 | Trig/IO | LUT/Shdg. | Adv 1 | Adv 2 | Adv 3

Fixed & variable modes...

Format: 320x240, 640x480, 800x600, 1024x768, 1280x960, **1600x1200**, F7 mode 0, F7 mode 1, F7 mode 2, F7 mode 3, F7 mode 4

Color: **YUV422**, Mono8

Framerate: 1.875 fps, 3.75 fps, **7.5 fps**, 15 fps

ISO speed: **S400**, S800

Debayering: none; auto. patterr

□ De-Interlace

High SNR mode: off

ISO channel: ☑ Auto Chn.; 0

Frame buffers: 16

Format7 settings...

Left (± 88 px) | Width (± 88px) | Top (± 88 px) | Height (± 88 px)

0000 Packet size 8888

□ Auto size

Max frame rate

Hide   □ Don't auto hide

Figure B.3: Stingray Settings 3

Appendix C: Sample failures on first 3 weeks production

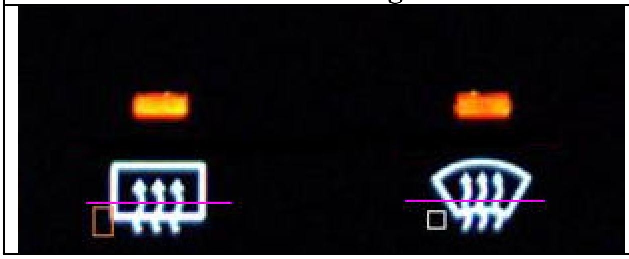
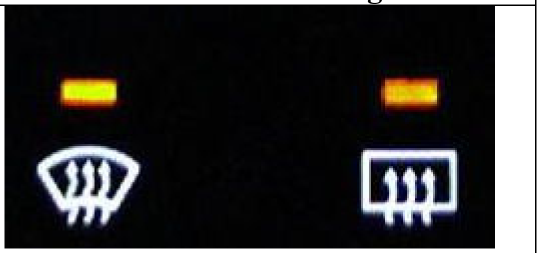
Failure: Front de-fog and rear de-fog swapped	
Root Cause of Failure: Operator Error in Assembly	
Inspection Method: Line measurement, Blank and Presence Check	
Failure Image	**Correct Pass Image**
	

Table C.1: Production Sample Failure 1

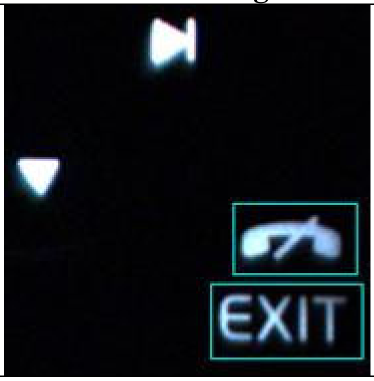
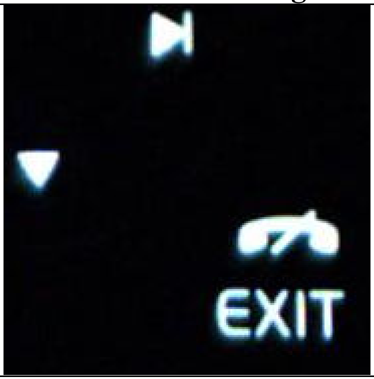
Failure: Exit Knob too Dull	
Root Cause of Failure: Paint back spray on underside of button	
Inspection Method: MT Inspection	
Failure Image	**Correct Pass Image**
	

Table C.2: Production Sample Failure 2

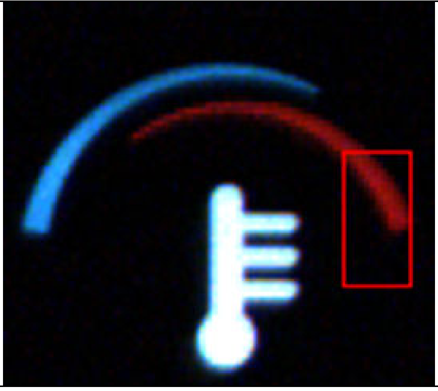
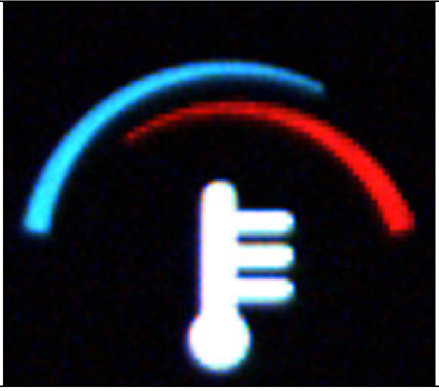
Failure: Red Colour on Right Hand Temp knob too dull	
Root Cause of Failure: Paint back spray on underside of button	
Inspection Method: Red Colour Inspection	
Failure Image	**Correct Pass Image**
	

Table C.3: Production Sample Failure 3

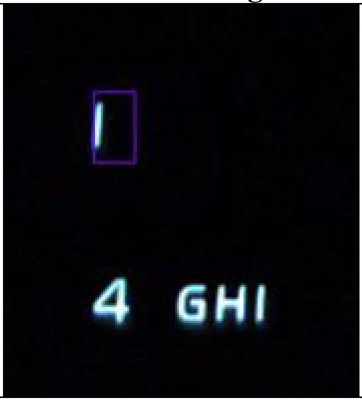
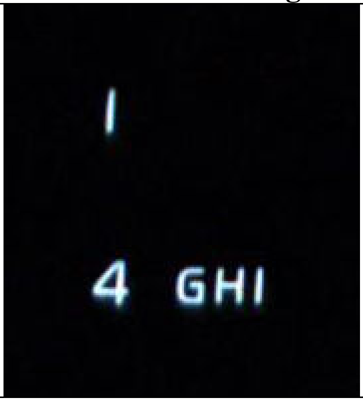
Failure: Digit Failure on '1'	
Root Cause of Failure: Poor Jigging in Paint and Laser Etching Process	
Inspection Method: Digit Inspection	
Failure Image	**Correct Pass Image**
	

Table C.4: Production Sample Failure 4

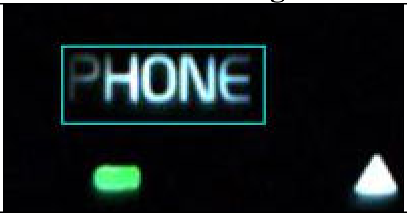
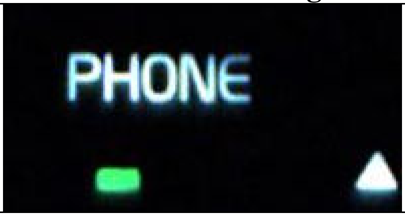
Failure: Phone Graphic 'P' Dull	
Root Cause of Failure: Paint back spray on underside of button	
Inspection Method: MT Inspection	
Failure Image	**Correct Pass Image**
	

Table C.5: Production Sample Failure 5

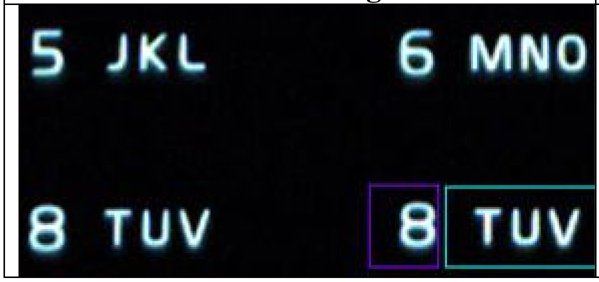
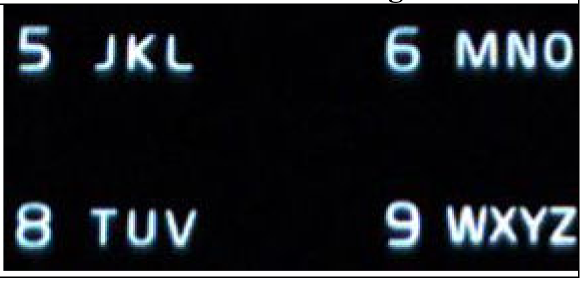
Failure: Digit Failure on '9'	
Root Cause of Failure: Rework of parts on cell, 8 placed instead of nine	
Inspection Method: MT Inspection, Digit Inspection	
Failure Image	**Correct Pass Image**
	

Table C.6: Production Sample Failure 6

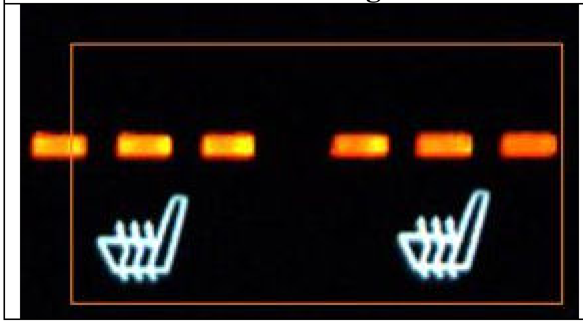
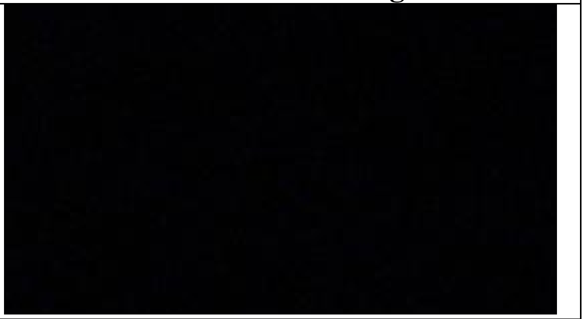
<table>
<tr><td colspan="2"><b>Failure:</b> Blank Failure on Right Seat Section</td></tr>
<tr><td colspan="2"><b>Root Cause of Failure:</b> Wrong Model Built, SANRA2191 built, should have been SANRA2892.</td></tr>
<tr><td colspan="2"><b>Inspection Method:</b> Blank Region Inspection</td></tr>
<tr><th>Failure Image</th><th>Correct Pass Image</th></tr>
<tr><td></td><td></td></tr>
</table>

Table C.7: Production Sample Failure 7

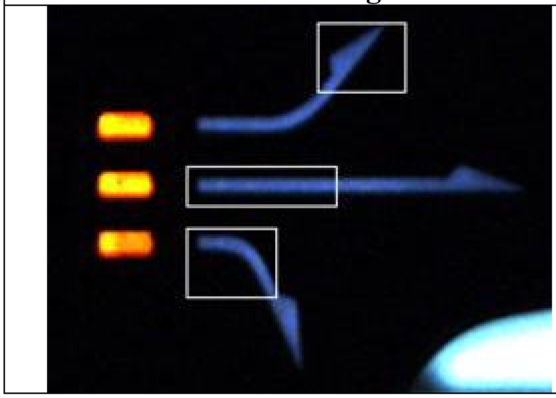
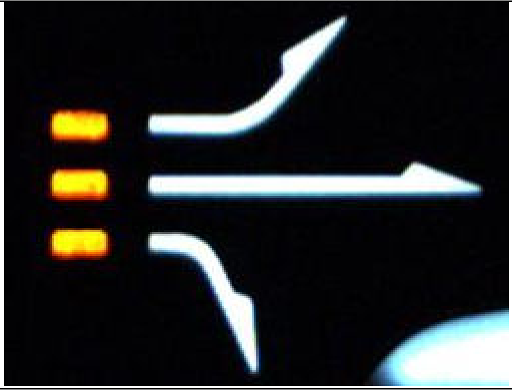
<table>
<tr><td colspan="2"><b>Failure:</b> Blank Failure on Air Direction Arrows</td></tr>
<tr><td colspan="2"><b>Root Cause of Failure:</b> Plastic blue adhesive sheet left on back of the panel</td></tr>
<tr><td colspan="2"><b>Inspection Method:</b> White Presence Region Inspection</td></tr>
<tr><th>Failure Image</th><th>Correct Pass Image</th></tr>
<tr><td></td><td></td></tr>
</table>

Table C.8: Production Sample Failure 8

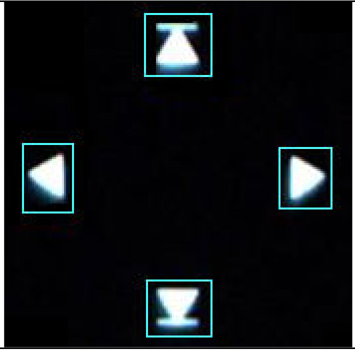
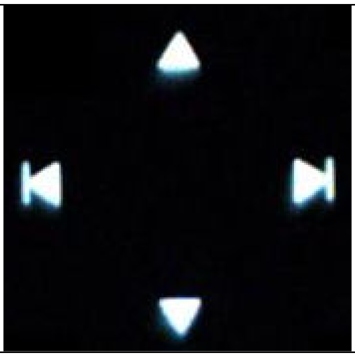
<table>
<tr><td colspan="2"><b>Failure:</b> MT failure on four way directional</td></tr>
<tr><td colspan="2"><b>Root Cause of Failure:</b> Knob Lasered incorrectly (90 degrees out of spec)</td></tr>
<tr><td colspan="2"><b>Inspection Method:</b> MT Inspection</td></tr>
<tr><th>Failure Image</th><th>Correct Pass Image</th></tr>
<tr><td></td><td></td></tr>
</table>

Table C.9: Production Sample Failure 9

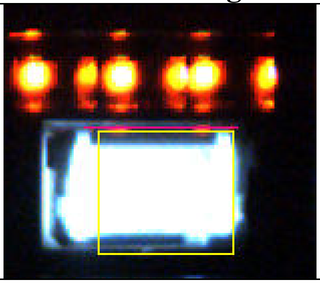
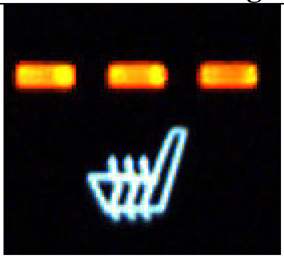
Failure: Seat Heat Pattern Match Fail	
Root Cause of Failure: Knob not assembled	
Inspection Method: Pattern Match, Line measurement	
Failure Image	**Correct Pass Image**
	

Table C.10: Production Sample Failure 10

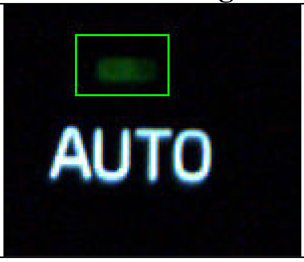
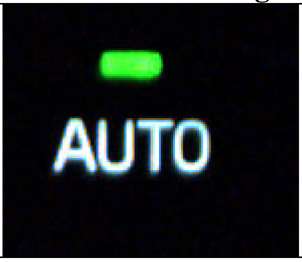
Failure: Auto green LED colour fail	
Root Cause of Failure: LED lightpipe is cloudy, poor light propagation	
Inspection Method: Green Colour Inspection	
Failure Image	**Correct Pass Image**
	

Table C.11: Production Sample Failure 11

Appendix D: Electrical Load Box Schematics

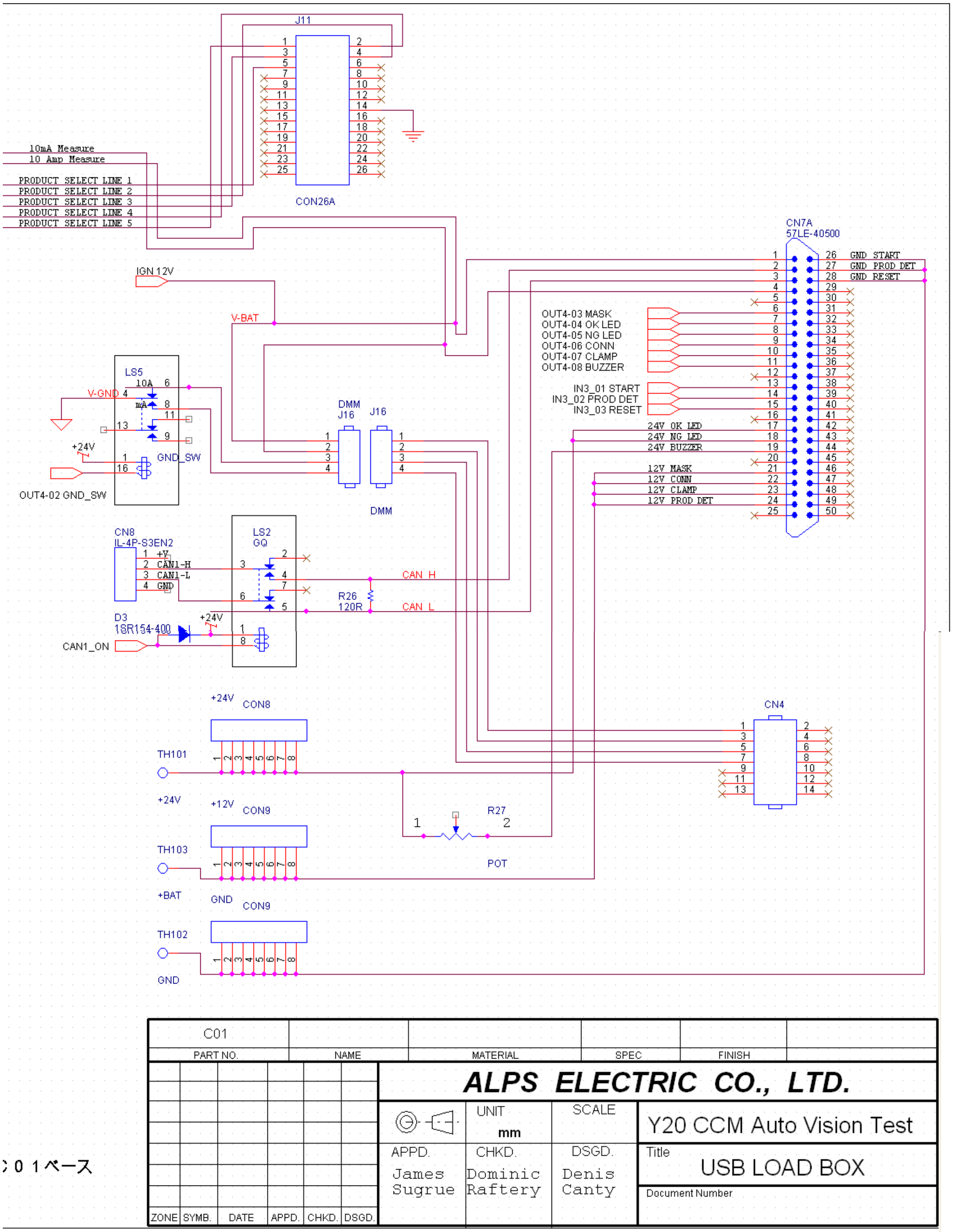


Figure D.1: Load Box Schematics 1

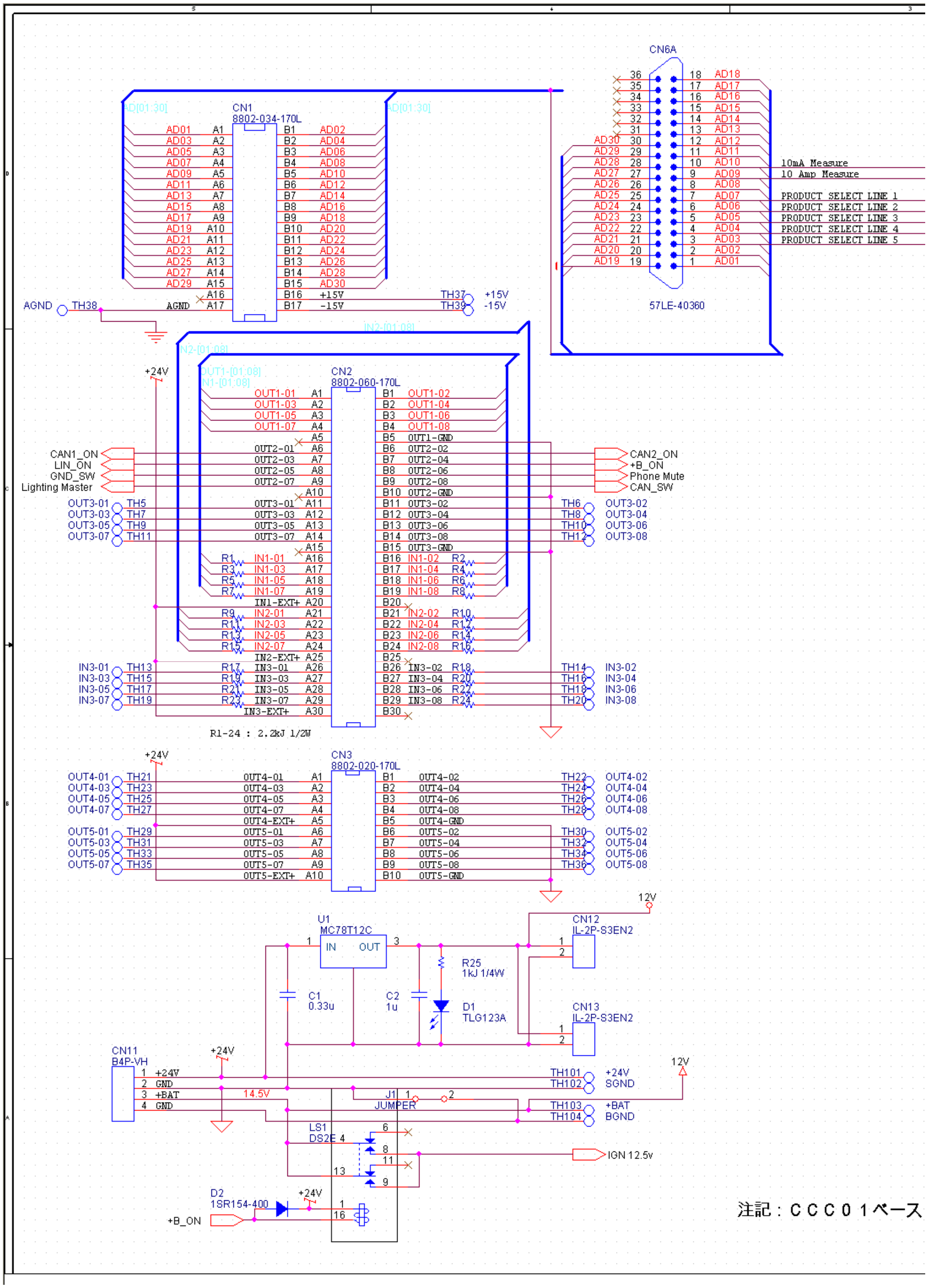
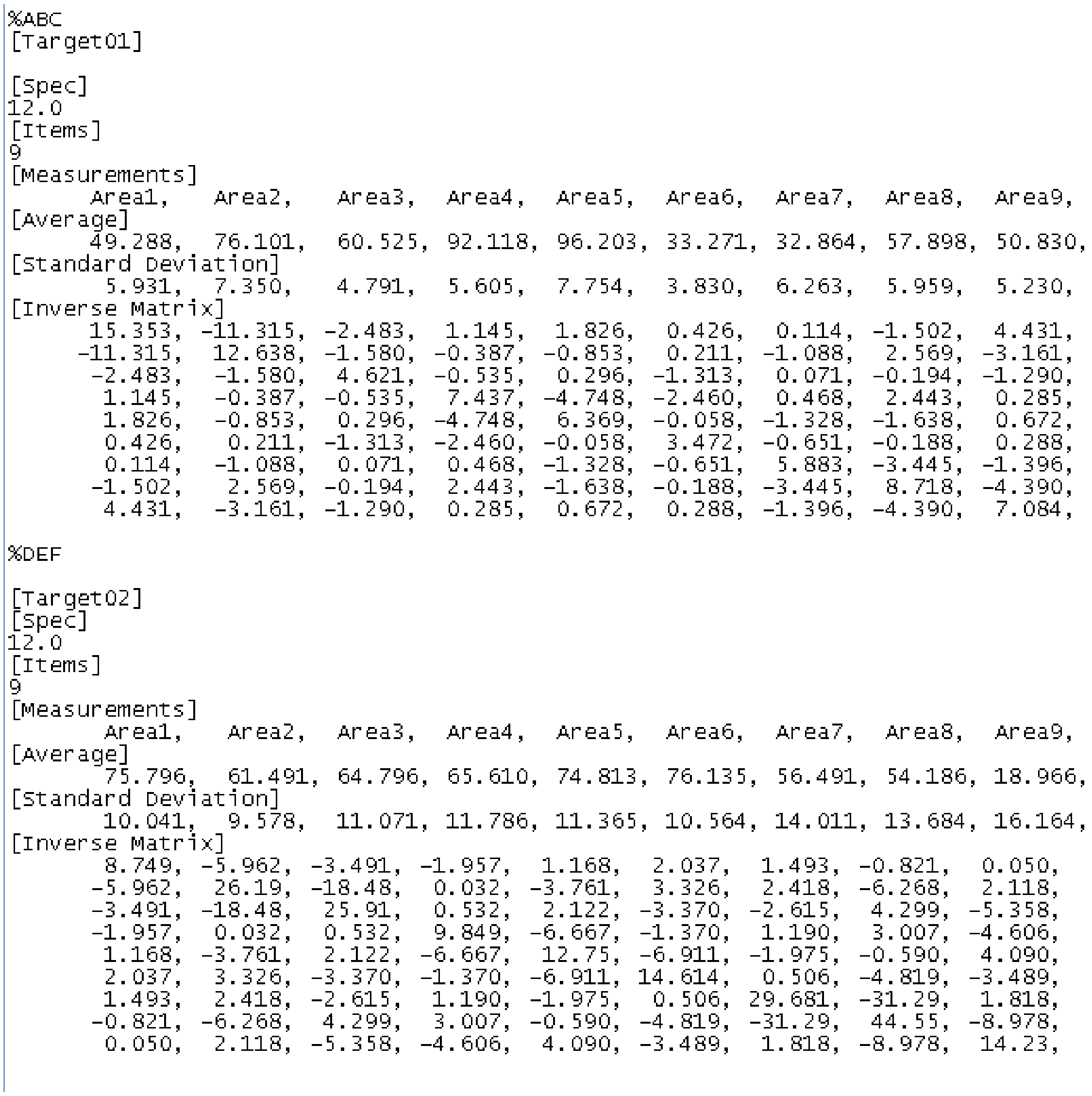


Figure D.2: Load Box Schematics 2

Appendix E: MT Calculation Data for ‘abc’ and ‘def’ graphics



```
%ABC
[Target01]

[Spec]
12.0
[Items]
9
[Measurements]
      Area1,   Area2,   Area3,  Area4,  Area5,  Area6,  Area7,  Area8,  Area9,
[Average]
      49.288,  76.101,  60.525, 92.118, 96.203, 33.271, 32.864, 57.898, 50.830,
[Standard Deviation]
       5.931,  7.350,   4.791,  5.605,  7.754,  3.830,  6.263,  5.959,  5.230,
[Inverse Matrix]
      15.353, -11.315, -2.483,  1.145,  1.826,  0.426,  0.114, -1.502,  4.431,
     -11.315,  12.638, -1.580, -0.387, -0.853,  0.211, -1.088,  2.569, -3.161,
      -2.483,  -1.580,  4.621, -0.535,  0.296, -1.313,  0.071, -0.194, -1.290,
       1.145,  -0.387, -0.535,  7.437, -4.748, -2.460,  0.468,  2.443,  0.285,
       1.826,  -0.853,  0.296, -4.748,  6.369, -0.058, -1.328, -1.638,  0.672,
       0.426,   0.211, -1.313, -2.460, -0.058,  3.472, -0.651, -0.188,  0.288,
       0.114,  -1.088,  0.071,  0.468, -1.328, -0.651,  5.883, -3.445, -1.396,
      -1.502,   2.569, -0.194,  2.443, -1.638, -0.188, -3.445,  8.718, -4.390,
       4.431,  -3.161, -1.290,  0.285,  0.672,  0.288, -1.396, -4.390,  7.084,

%DEF

[Target02]
[Spec]
12.0
[Items]
9
[Measurements]
       Area1,   Area2,  Area3,  Area4,  Area5,  Area6,  Area7,  Area8,  Area9,
[Average]
       75.796,  61.491, 64.796, 65.610, 74.813, 76.135, 56.491, 54.186, 18.966,
[Standard Deviation]
       10.041,  9.578,  11.071, 11.786, 11.365, 10.564, 14.011, 13.684, 16.164,
[Inverse Matrix]
       8.749, -5.962, -3.491, -1.957,  1.168,  2.037,  1.493, -0.821,  0.050,
      -5.962,  26.19, -18.48,  0.032, -3.761,  3.326,  2.418, -6.268,  2.118,
      -3.491, -18.48,  25.91,  0.532,  2.122, -3.370, -2.615,  4.299, -5.358,
      -1.957,  0.032,  0.532,  9.849, -6.667, -1.370,  1.190,  3.007, -4.606,
       1.168, -3.761,  2.122, -6.667,  12.75, -6.911, -1.975, -0.590,  4.090,
       2.037,  3.326, -3.370, -1.370, -6.911, 14.614,  0.506, -4.819, -3.489,
       1.493,  2.418, -2.615,  1.190, -1.975,  0.506, 29.681, -31.29,  1.818,
      -0.821, -6.268,  4.299,  3.007, -0.590, -4.819, -31.29,  44.55, -8.978,
       0.050,  2.118, -5.358, -4.606,  4.090, -3.489,  1.818, -8.978,  14.23,
```

Figure E.1: MT Calculation Data for abc and def graphics

Appendix F: Image of Electrical Load Box

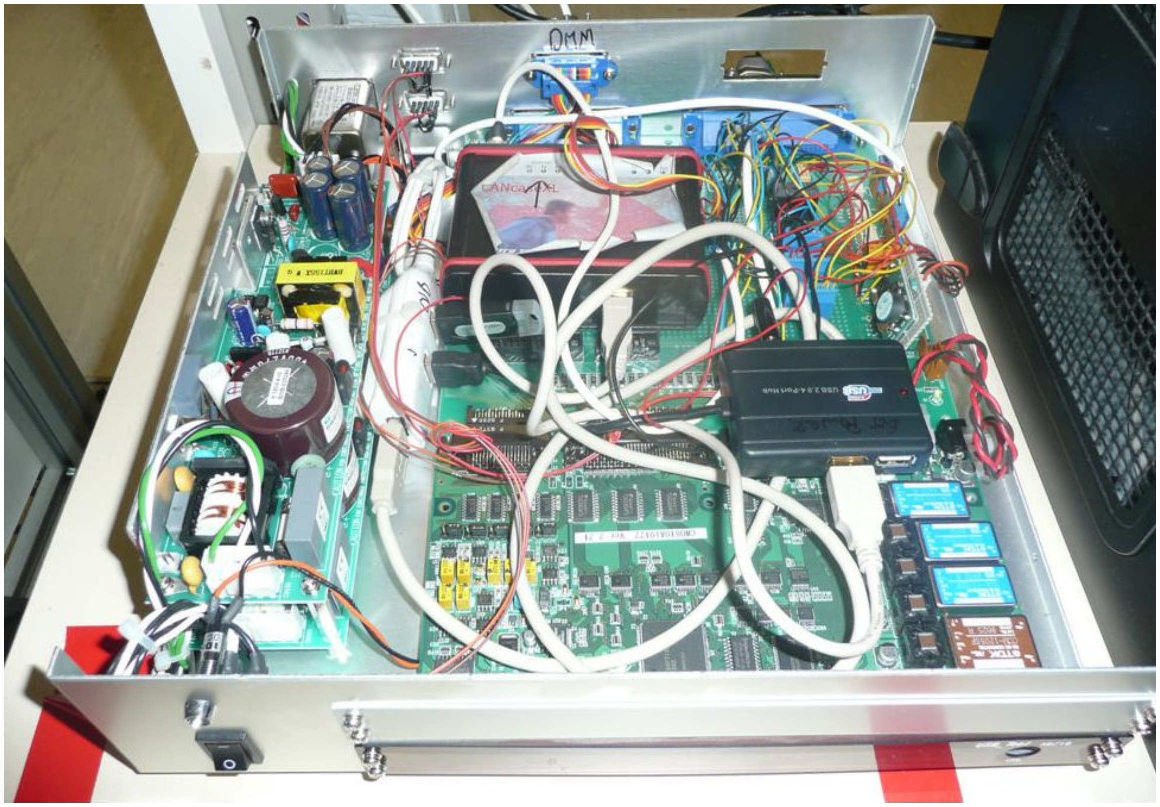


Figure F.1: Load Box

Appendix G: Model Types

SANVA2190-95

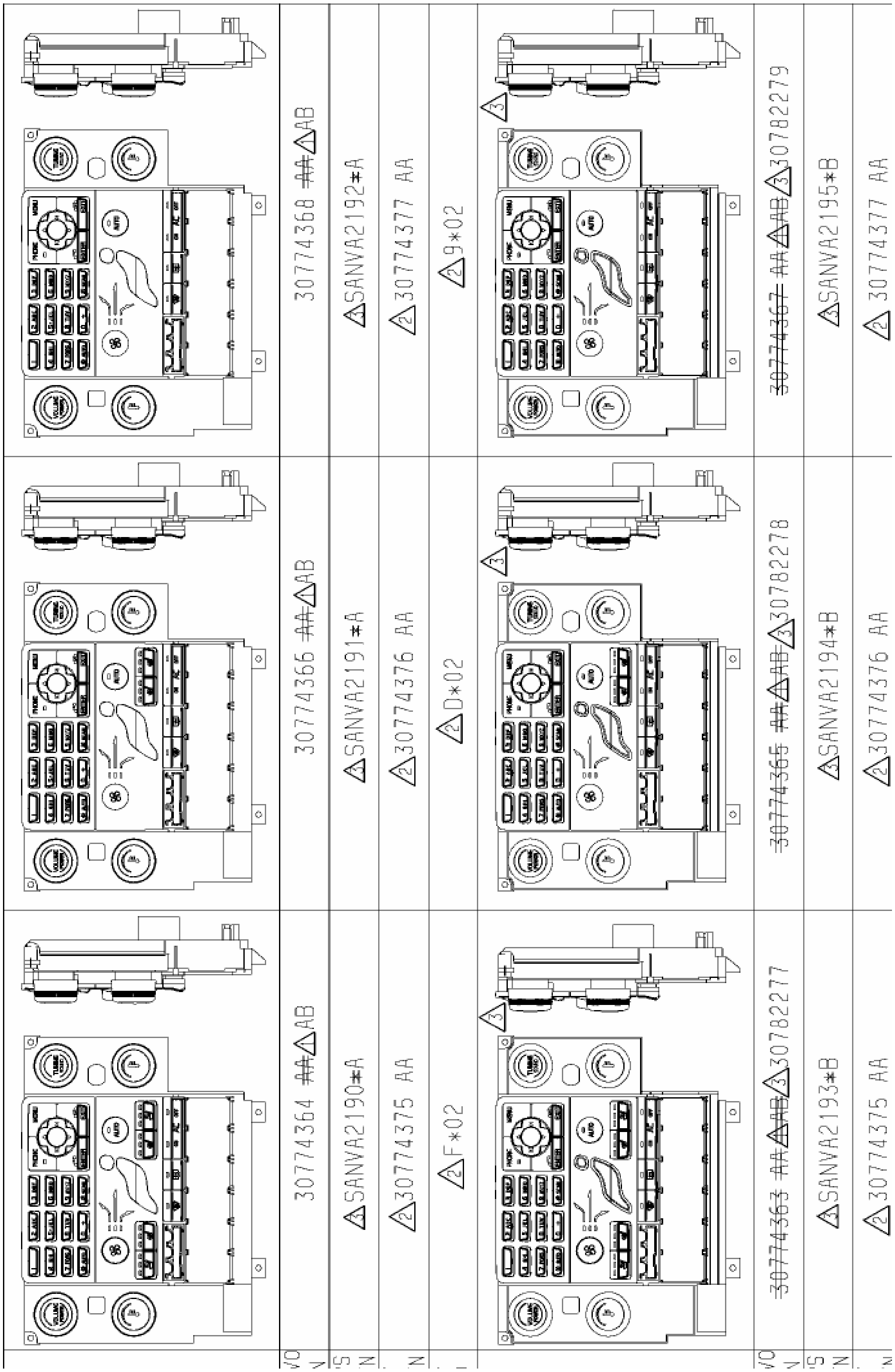


Figure G.1: SANVA2190-95 Model Types

SANVA2200-05

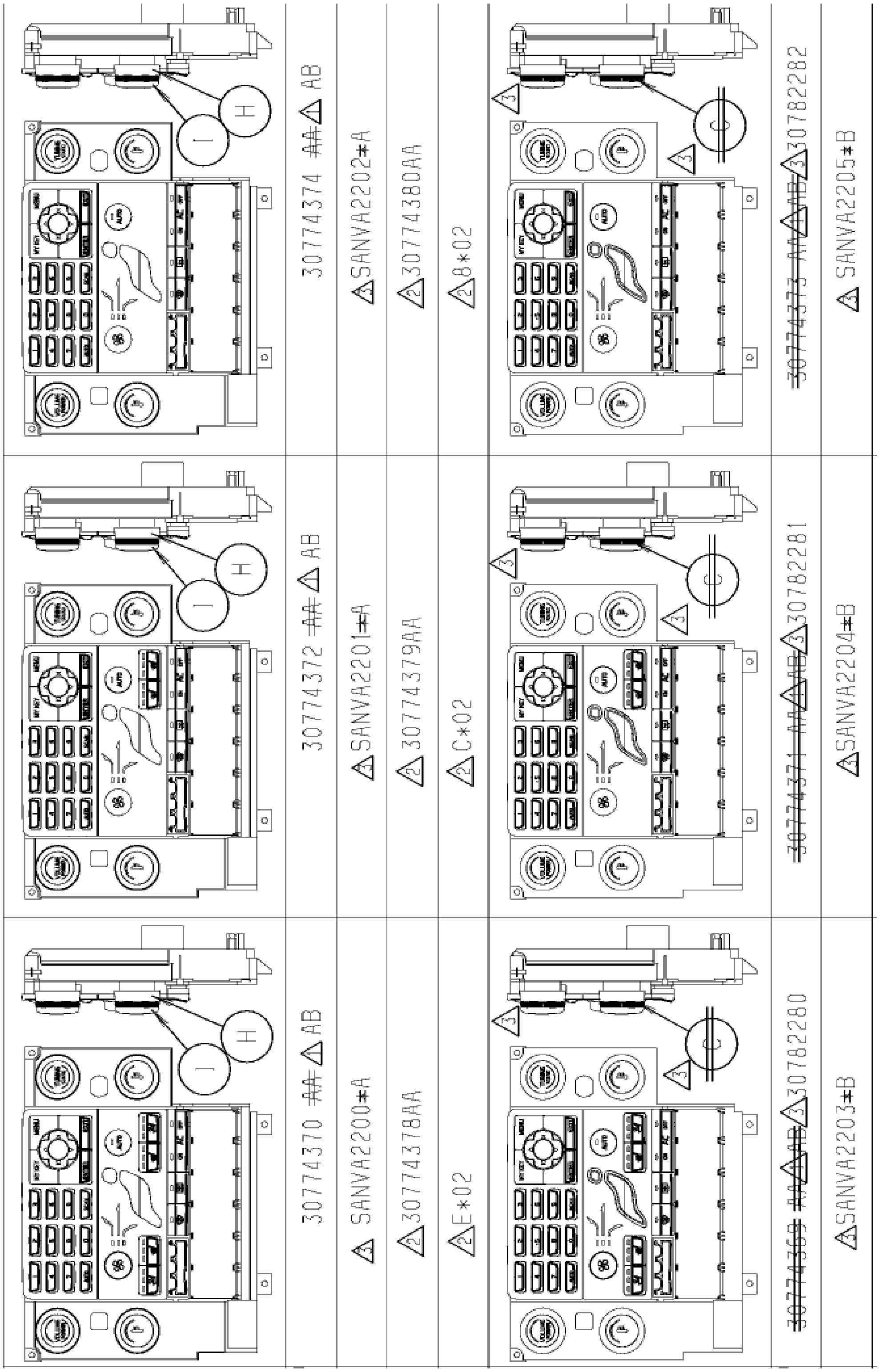


Figure G.2: SANVA2200-05 Model Types

SANVA2180-81

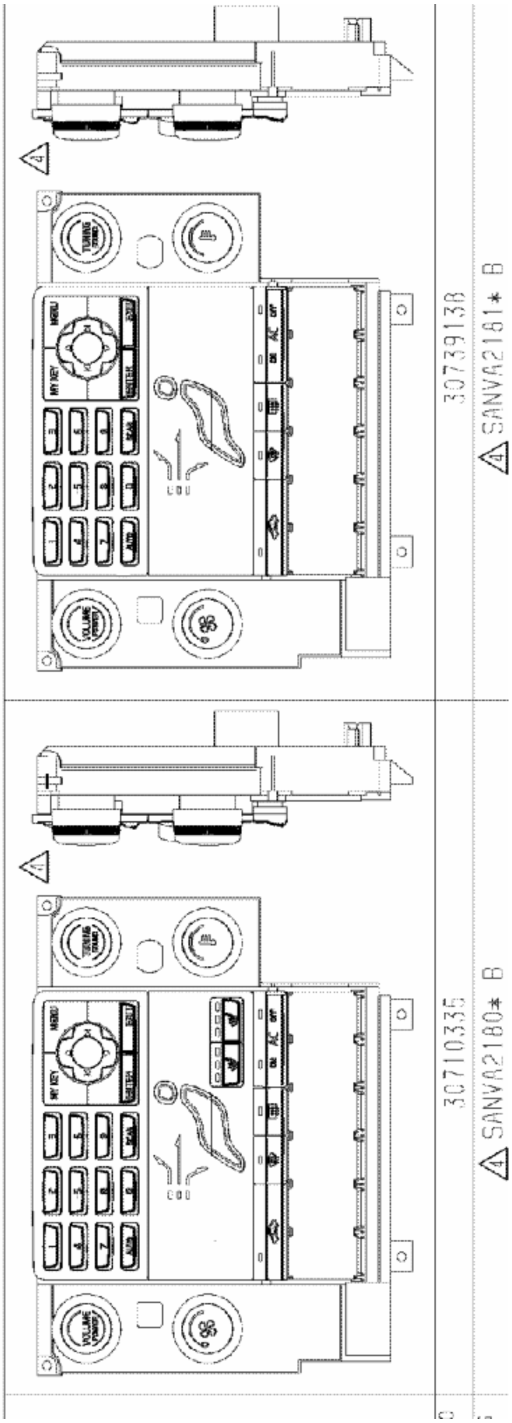
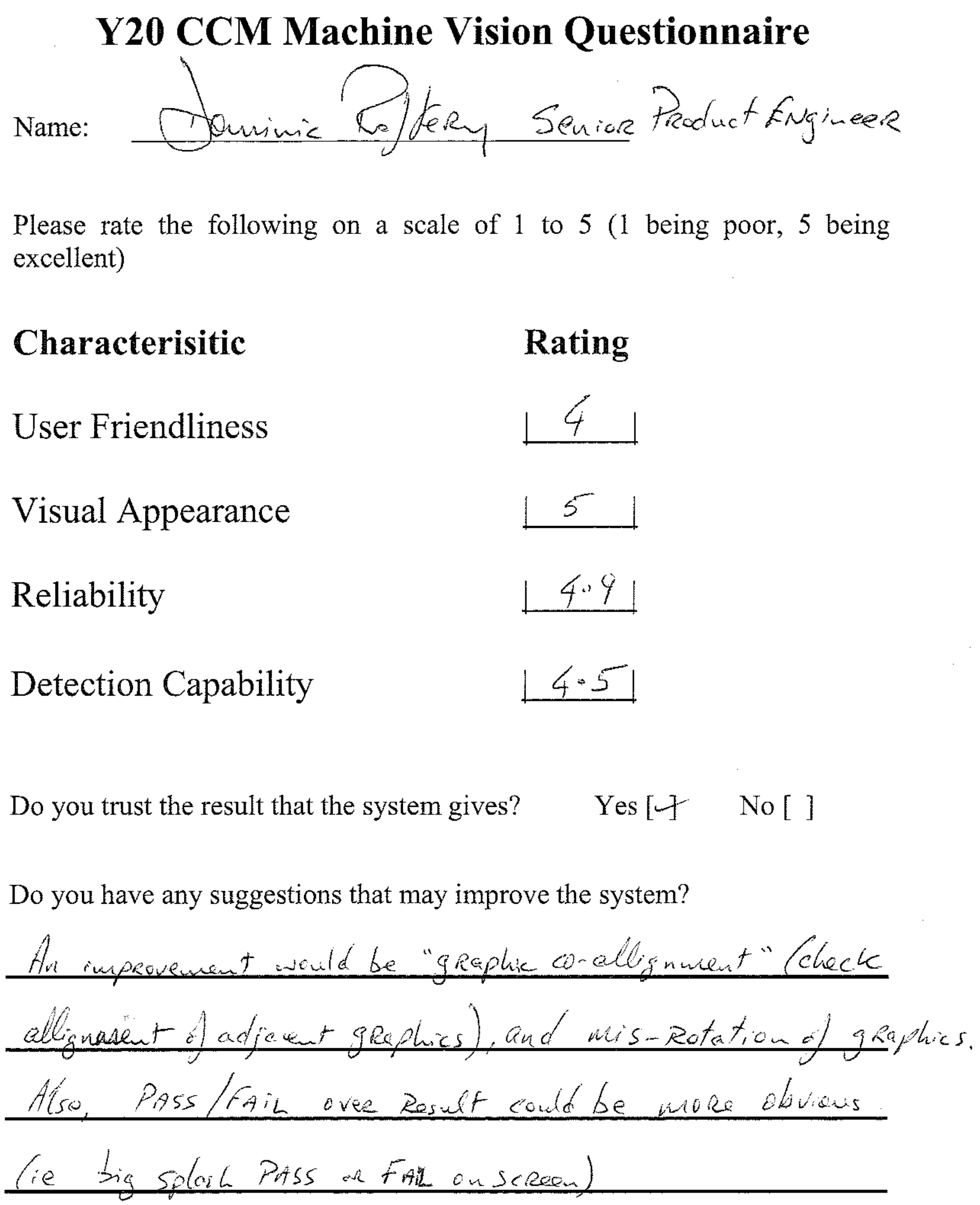


Figure G.3: SANVA2180-81 Model Types

Appendix H: Questionnaire Results



Y20 CCM Machine Vision Questionnaire

Name: Dominic Raftery Senior Product Engineer

Please rate the following on a scale of 1 to 5 (1 being poor, 5 being excellent)

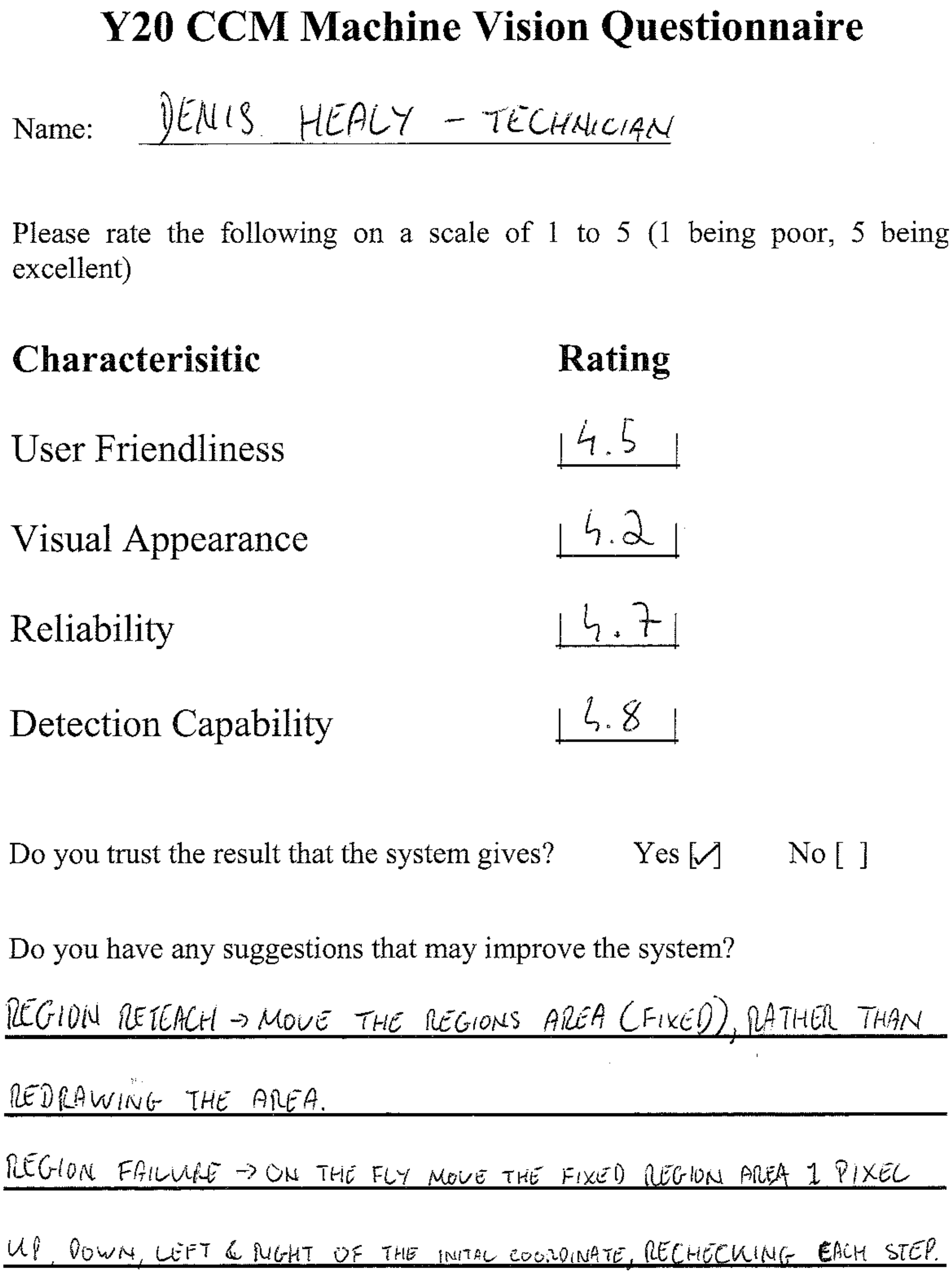
Characterisitic	Rating
User Friendliness	4
Visual Appearance	5
Reliability	4.9
Detection Capability	4.5

Do you trust the result that the system gives? Yes [✓] No [ ]

Do you have any suggestions that may improve the system?

An improvement would be "graphic co-allignment" (check allignment of adjacent graphics), and mis-Rotation of graphics. Also, PASS/FAIL over Result could be more obvious (ie big splash PASS or FAIL on screen)

Figure H.1: Dominic Raftery Questionnaire Result



Y20 CCM Machine Vision Questionnaire

Name: DENIS HEALY - TECHNICIAN

Please rate the following on a scale of 1 to 5 (1 being poor, 5 being excellent)

Characterisitic	Rating
User Friendliness	4.5
Visual Appearance	4.2
Reliability	4.7
Detection Capability	4.8

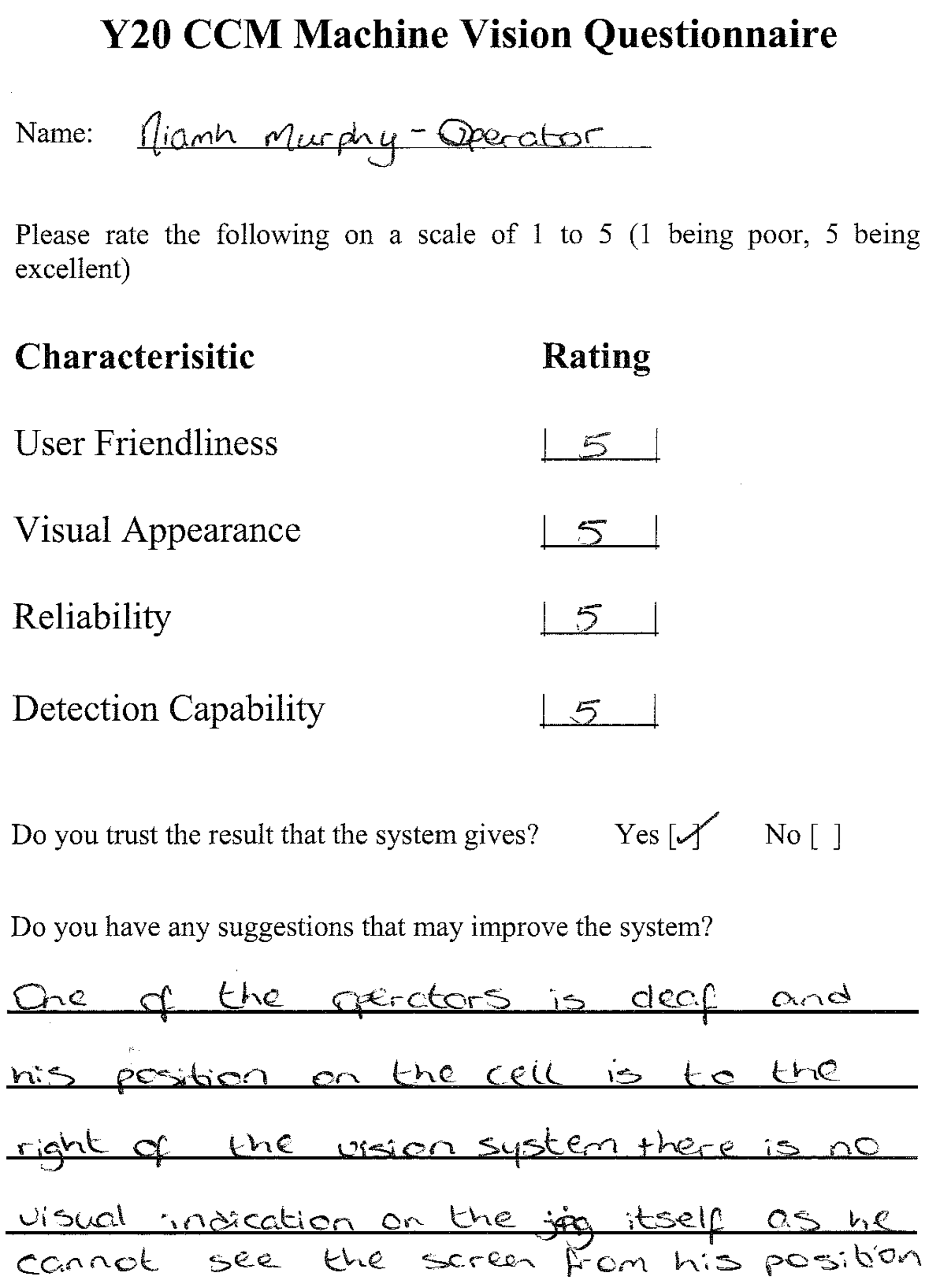
Do you trust the result that the system gives? Yes [✓] No [ ]

Do you have any suggestions that may improve the system?

REGION RETEACH -> MOVE THE REGIONS AREA (FIXED), RATHER THAN REDRAWING THE AREA.

REGION FAILURE -> ON THE FLY MOVE THE FIXED REGION AREA 1 PIXEL UP, DOWN, LEFT & RIGHT OF THE INITAL COORDINATE, RECHECKING EACH STEP.

Figure H.2: Denis Healy Questionnaire Result



Y20 CCM Machine Vision Questionnaire

Name: Niamh Murphy - Operator

Please rate the following on a scale of 1 to 5 (1 being poor, 5 being excellent)

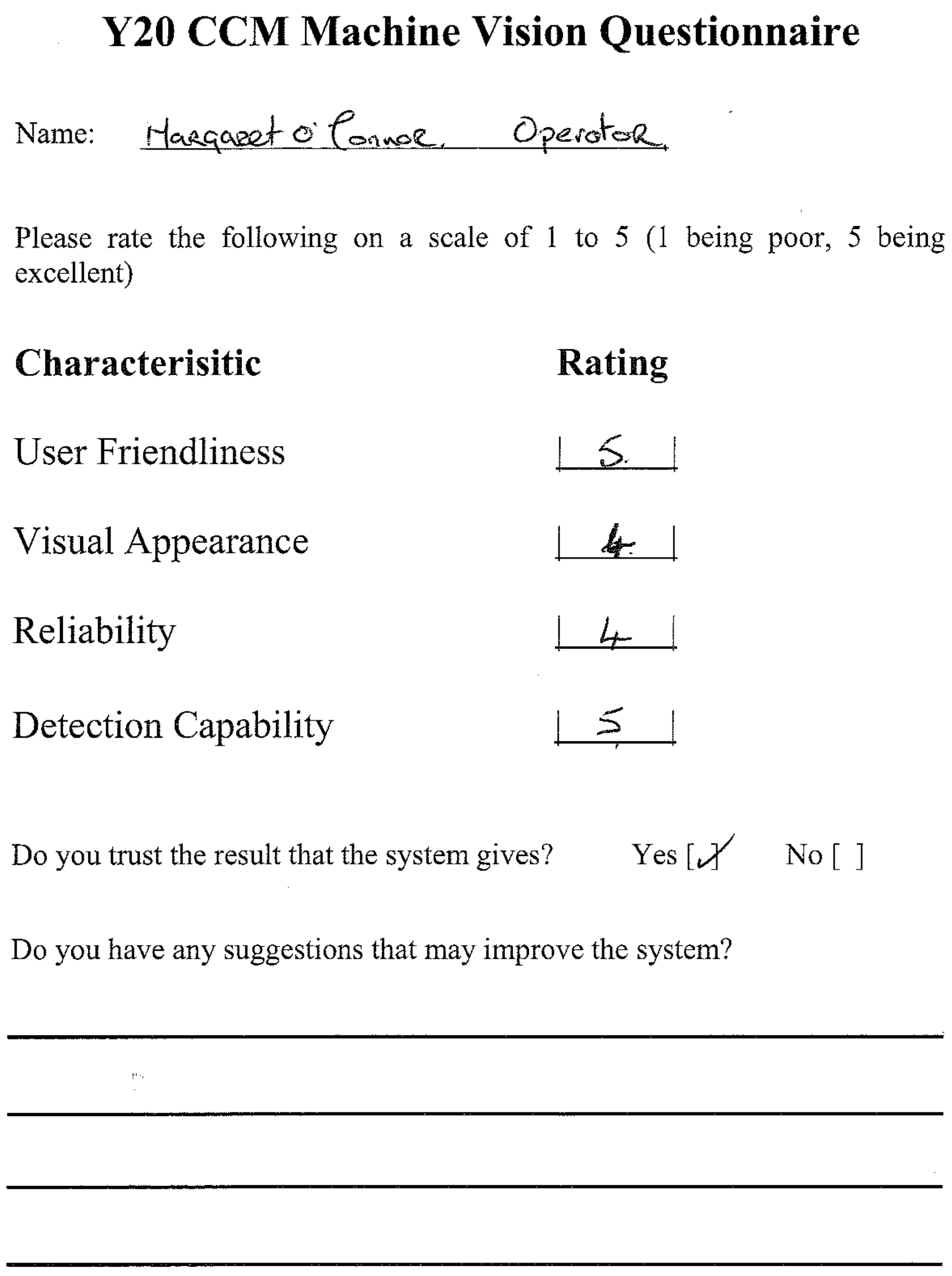
Characterisitic	Rating
User Friendliness	5
Visual Appearance	5
Reliability	5
Detection Capability	5

Do you trust the result that the system gives? Yes [✓] No [ ]

Do you have any suggestions that may improve the system?

One of the operators is deaf and his position on the cell is to the right of the vision system there is no visual indication on the jig itself as he cannot see the screen from his position

Figure H.3: Niamh Murphy Questionnaire Result



Y20 CCM Machine Vision Questionnaire

Name: Margaret O' Connor. Operator

Please rate the following on a scale of 1 to 5 (1 being poor, 5 being excellent)

Characterisitic	Rating
User Friendliness	5.
Visual Appearance	4
Reliability	4
Detection Capability	5

Do you trust the result that the system gives? Yes [✓] No [ ]

Do you have any suggestions that may improve the system?

Figure H.4: Margaret O Connor Questionnaire Result

Appendix I: Field of view images from original 4 camera system

FOV Image Area 1:

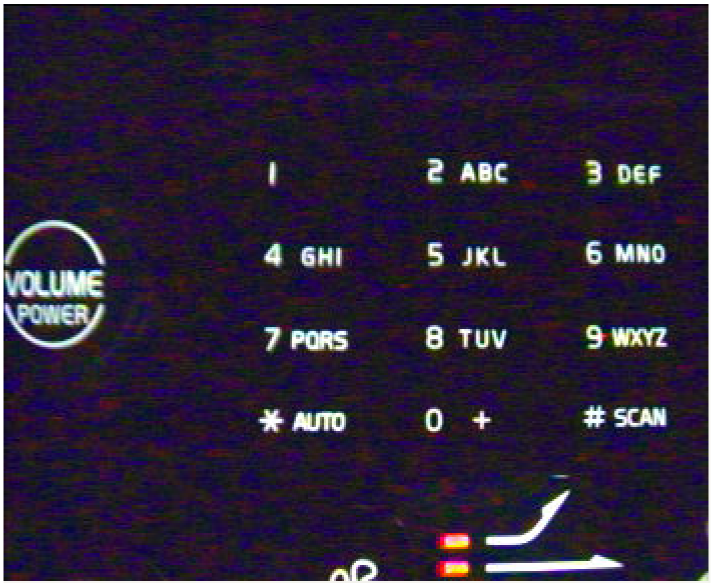


Figure I.1: Four Camera System FOV 1

FOV Image Area 2:

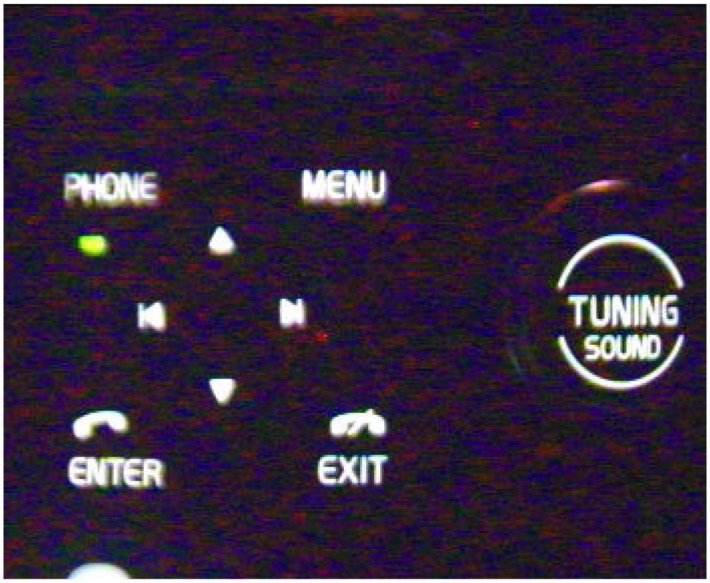


Figure I.2: Four Camera System FOV 2

FOV Image Area 3:

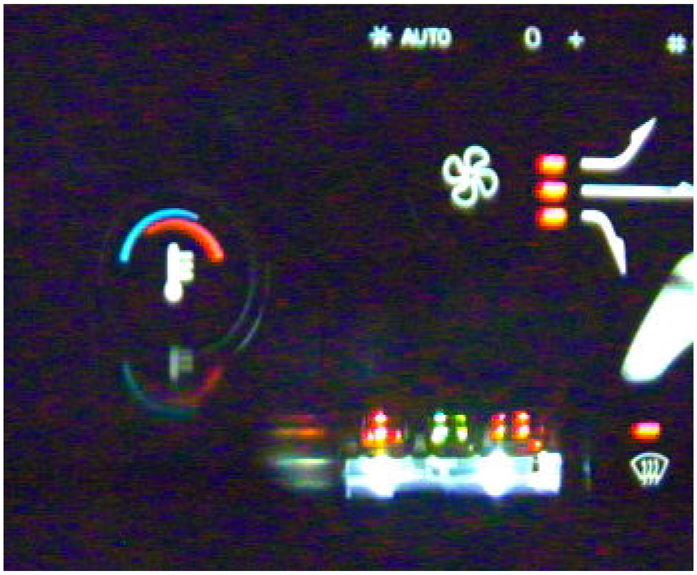


Figure I.3: Four Camera System FOV 3

FOV Image Area 4:

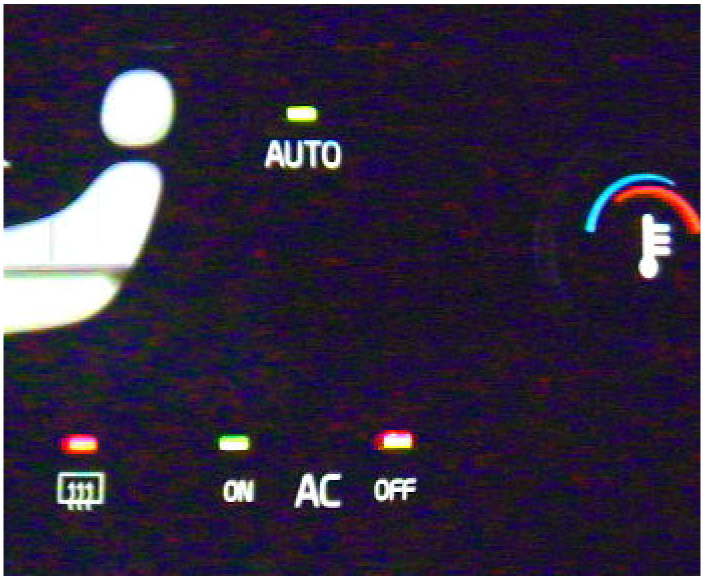


Figure I.4: Four Camera System FOV 4

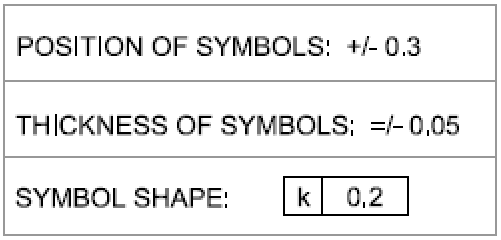
Appendix J: Quality Specification

LED's and Cosmetic Colour

1: All assembled LED's are of the correct colour.
2: All cosmetics are laser etched the correct colour.
3: All colour levels are above respective RGB level 160.

Graphic Symbols

Note: All measurements in millimetres.



POSITION OF SYMBOLS: +/- 0.3		
THICKNESS OF SYMBOLS: =/- 0,05		
SYMBOL SHAPE:	k	0,2

Table J.1: Symbol Outline Specification

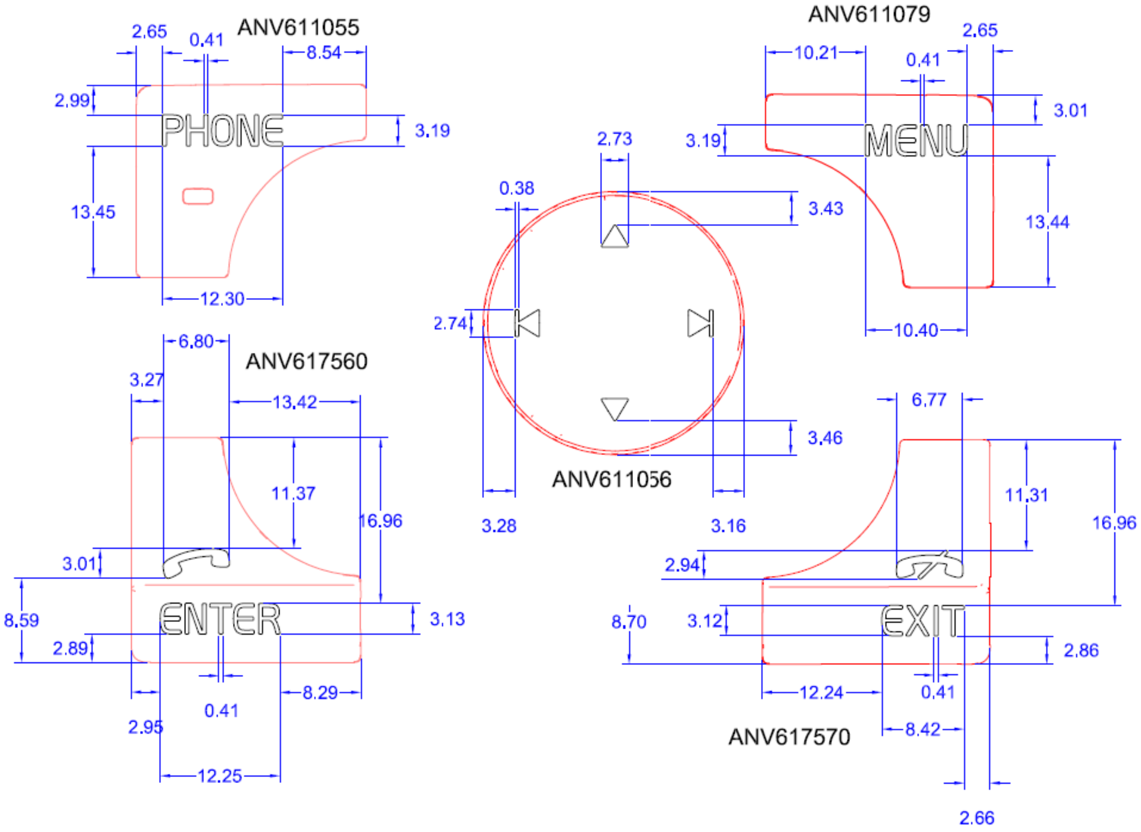


Figure J.1: Phone Keypad Specification

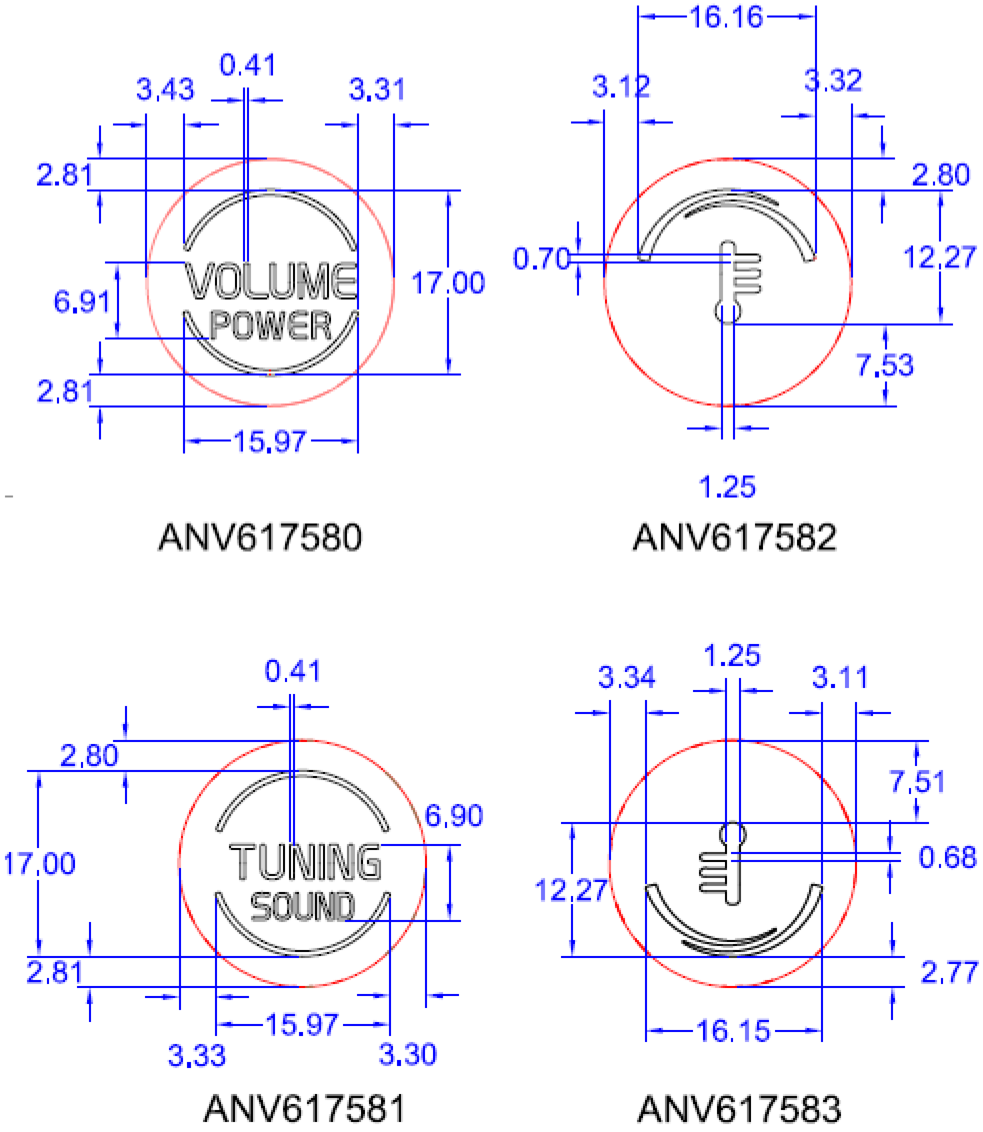


Figure J.2: Rotary Knob Specification

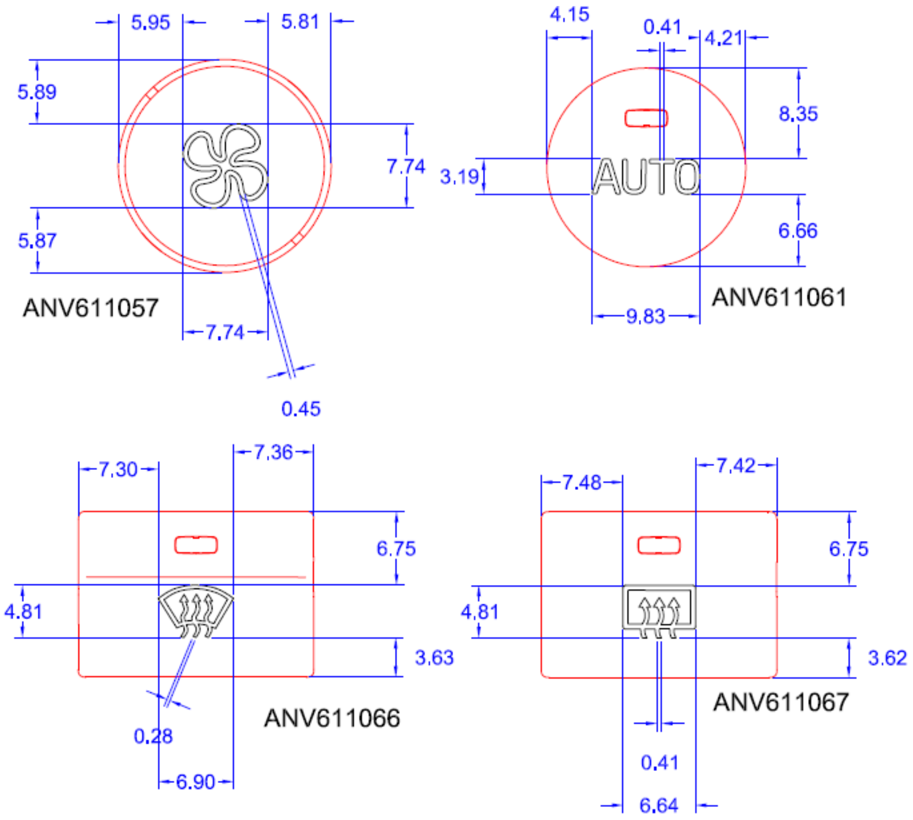


Figure J.3: Dfog, Fan and Auto Specification

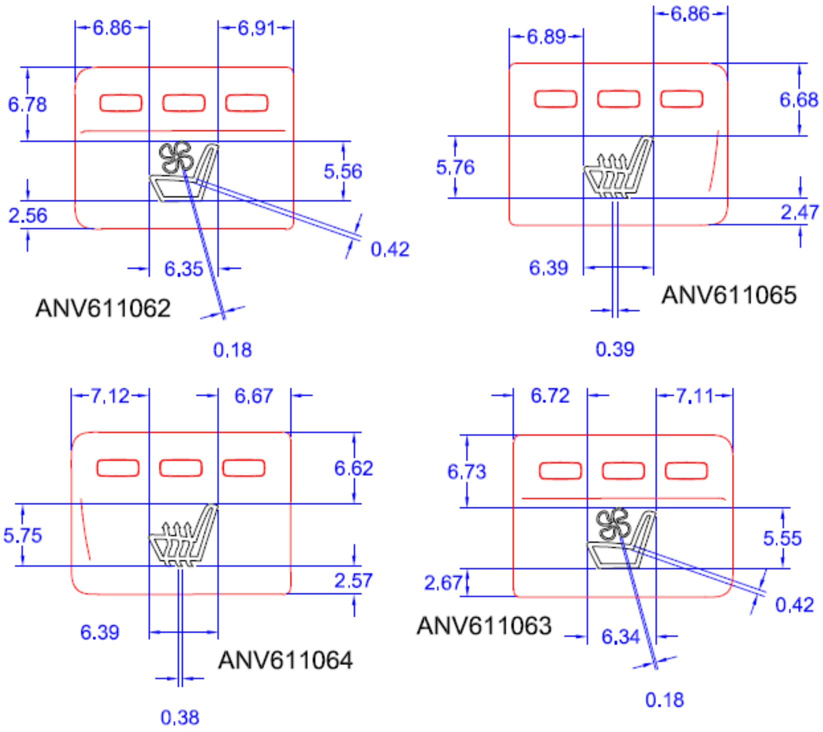


Figure J.4: Seat Heat and Seat Fan Specification

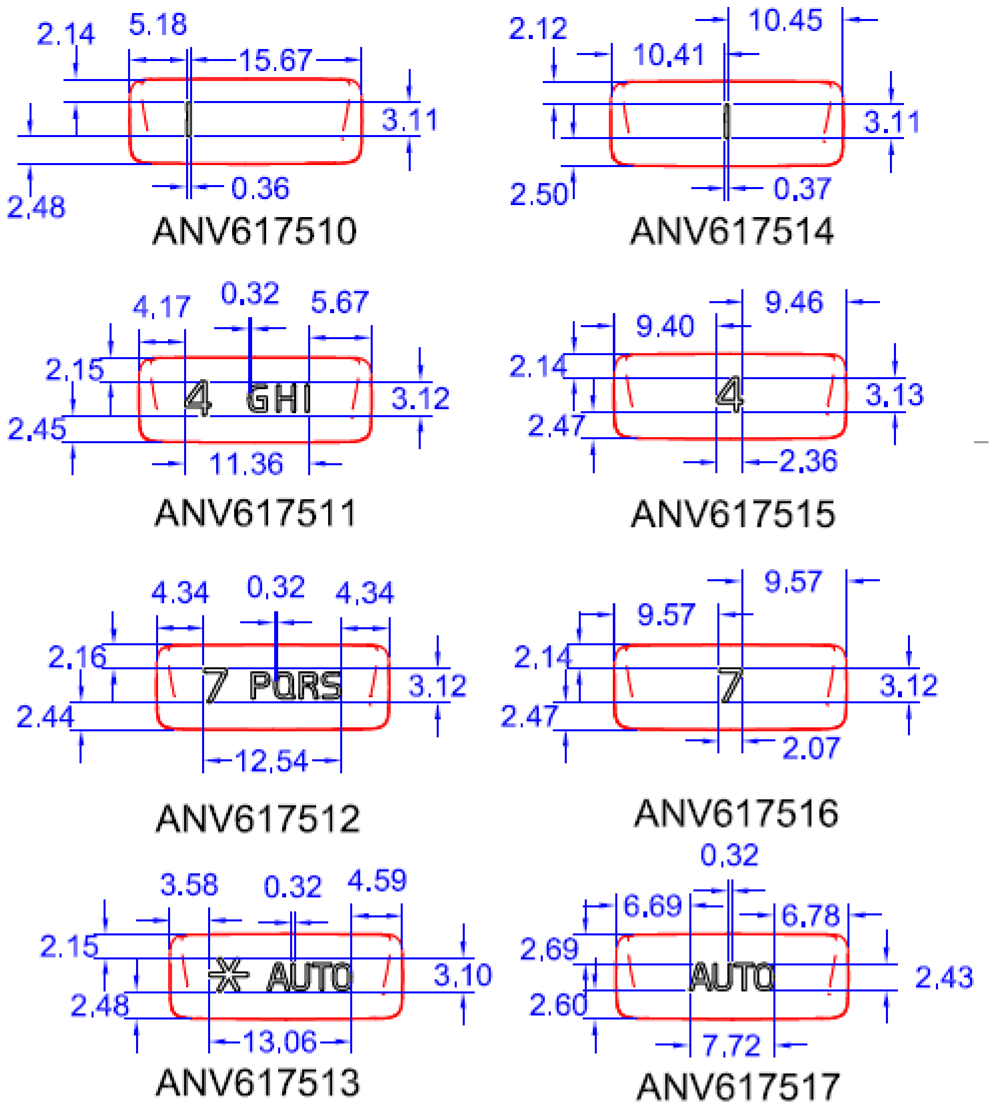


Figure J.5: 1, 4, 7, Auto Specification

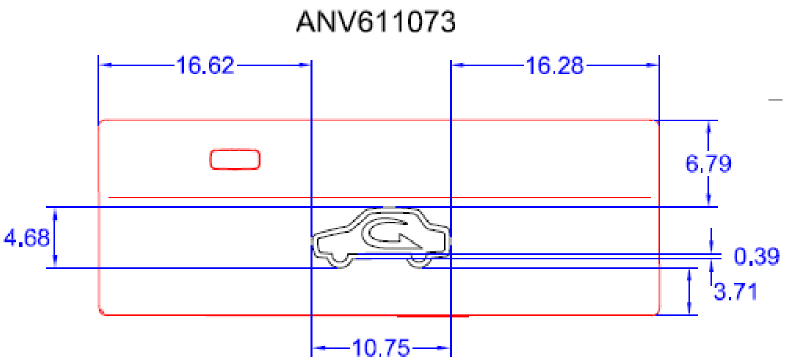


Figure J.6: Internal Recirculation Specification

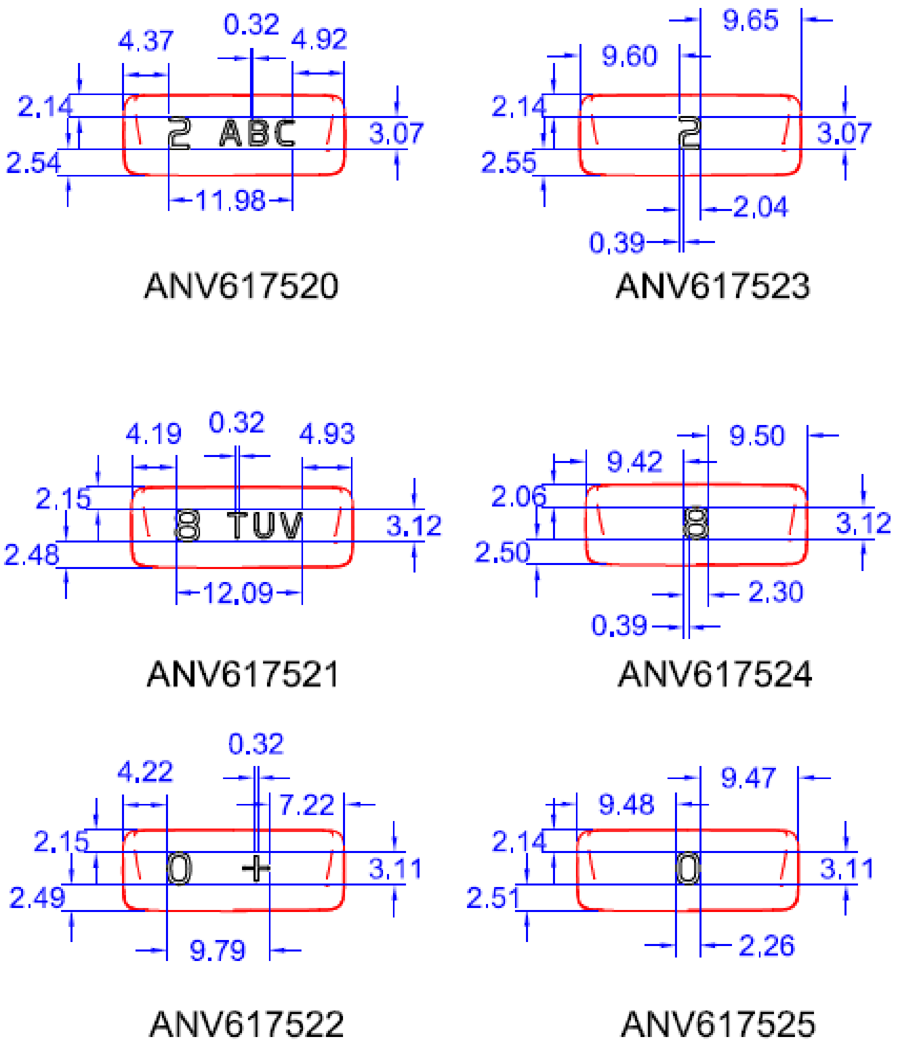


Figure J.7: 2, 8, and 0 Specification

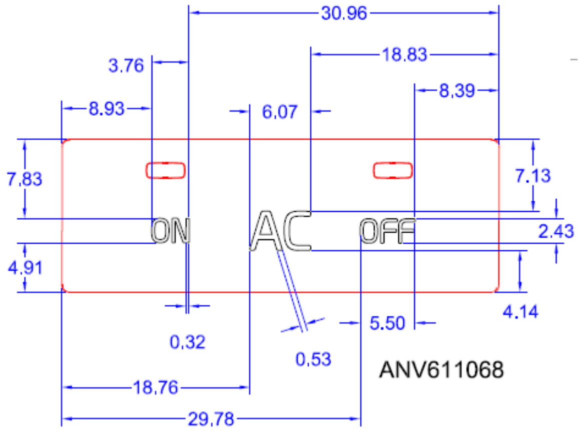


Figure J.8: AC, On and Off Specification

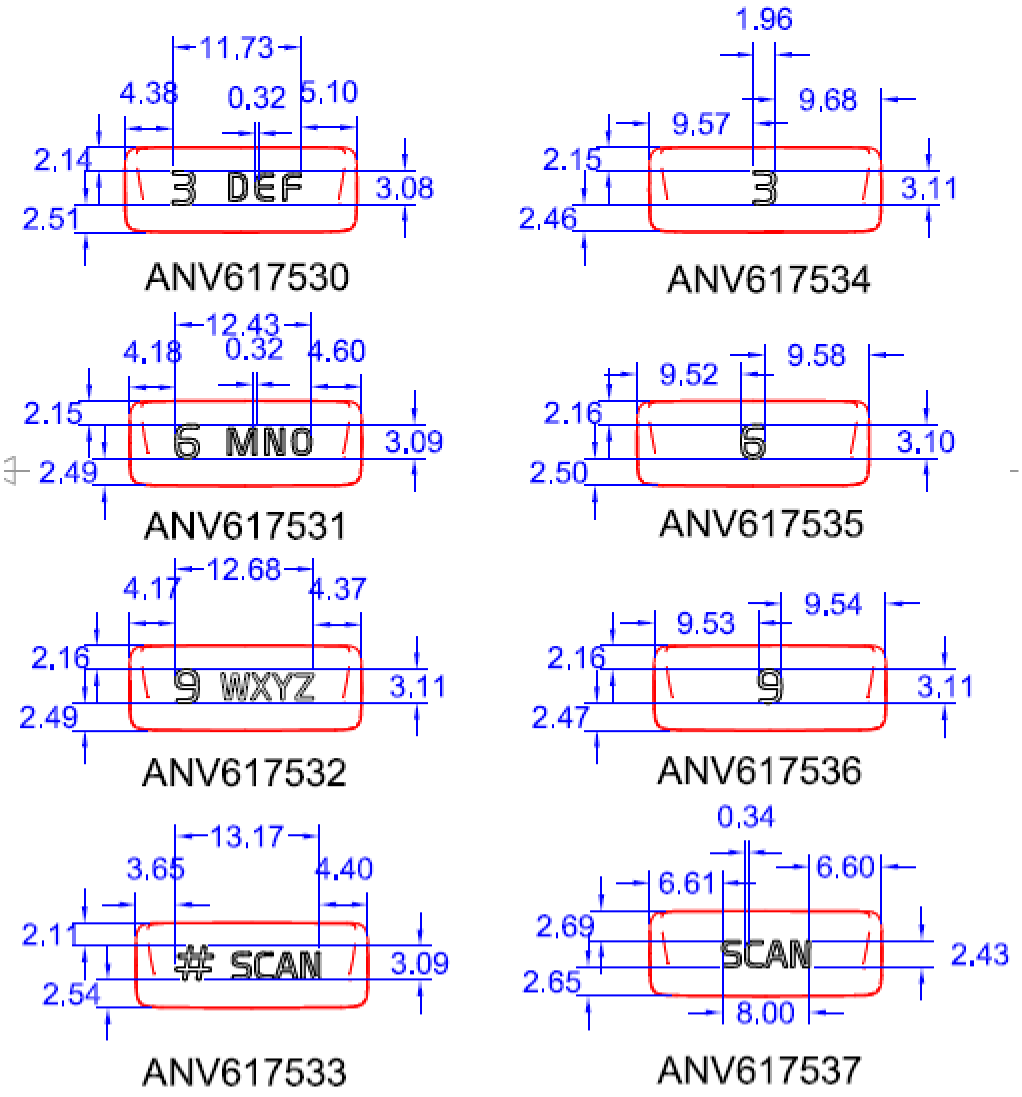


Figure J.9: 3, 6, 9 and Scan Specification

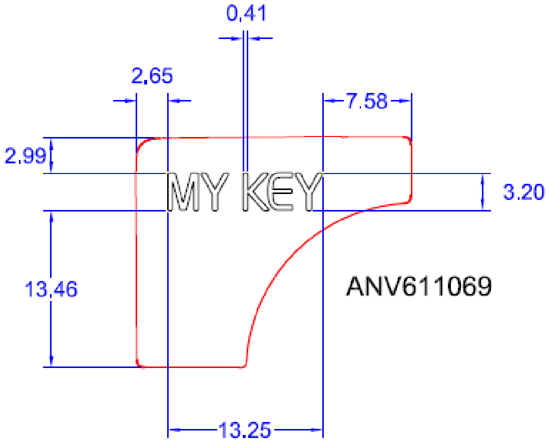


Figure J.10: MyKey Specification