

2118459

by Cerys Joseph

MARK OF 58% MARKS


Submission date: 23-May-2023 05:06PM (UTC+0100)

Submission ID: 207358375

File name: 17372_Cerys_Joseph_2118459_181809_714230686.pdf (17.83M)

Word count: 4181

Character count: 22652

Course: HNC Electrical and Electronic Engineering

Module: HNC Electrical and Electronic Engineering Project

Assignment: Final Report and Logbook

Name: Cerys Joseph

University Number: 2118459

Date Submitted: 23/05/2023

Word: 3867

Content Page	
Content Page	2
1. Introduction and Background Research	3
2. Specification.....	4
3. Planning	5
4. Solutions Considered.....	6
5. Final Solution Design and Implementation.....	7
6. Testing and Results.....	23
7. Discussion, Conclusions and Recommendations.....	24
8. Reference List and Bibliography.....	25
9. Appendix.....	26

Good table of contents

Figure 1 Gantt Chart.....	5
Figure 2 Cognex Camera.....	6
Figure 3 G Code.....	8
Figure 4 Zoomed in G Code	8
Figure 5 Zoomed in G Code	9
Figure 6 Zoomed in G Code	9
Figure 7 Zoomed in G Code	10
Figure 8 Vision Assistant.....	11
Figure 9 Setting up USB camera.....	11
Figure 10 Selecting the Acquisition.....	12
Figure 11 Setting up display screen	12
Figure 12 Flat Sequence Structure.....	13
Figure 13 Creating Vision Assistant.....	13
Figure 14 Opening Image Sequence.....	17
Figure 15 First Straight Edge	17
Figure 16 Second Straight Edge	18
Figure 17 Caliper	18
Figure 18 Set Coordinate System	19
Figure 19 Pattern Matching	19
Figure 20 Final Script Implementation	20
Figure 21 Test Image 1	20
Figure 22 Test Image 2	21
Figure 23 Test Image 3	26
Figure 24 Test Image 4	26
Figure 25 Test Image 5	27
Figure 26 Test Image 6	27

Good extensive list of figures

1. Introduction and Background Research

Spelling and grammar errors

The reason for creating this project is to have a system that can scan a board on a quality check line which can identify the components that are on it giving them a pass or fail induction. The camera will be able to identify the components on the board by using the designed template which has to be programmed into it. The problem that is being overcome is identifying the components on a board using a software where the camera has been connected to.

Vision sensor systems will perform inspections at quick speed and can link the factory information straight into the production line, this will help in stock control and inspection details. The sensors are adaptable and is flexible, which makes them ideal to use in a wide range of features. The three big elements of vision sensors are an easy to use interface, powerful inspection and a modular vision system. (ivs, 2023) In the automotive industries use the vision sensor throughout the process, to ensure the quality is kept at the highest standard throughout. (ivs, 2023) In the medical and pharmacies industries the products must be kept to the FDA and GAMP standards, this is done by the vision sensors checking the products throughout. (ivs, 2023)

Relevant sources used

The intro includes some significant points

2. Specification

The project is a detector which can scan the amount of components on the board and if they are in the correct positions. The software will also be able to detect the components even after the board has moved positions. The project will need to be finished by the middle of May, the problems that might arise is having the components delivered to the college. Also a budgetary constraint might be downloading the software as a computer is needed and paying for the licence. *Significant point*

The vision sensor will assist the customer in quality and fault finding as the project is designed to find any faults that might have happened during the building stage. A fault could be the wrong size component on the board or the incorrect colour components on the board, when a fault is found the indicators on the main screen will turn red to symbolise that the board has fail the test. If no fault has been found by the software then a signal will be sent to the inductors and they will turn green. The maintenance should be easy on the project as the software is stored and ran from a laptop which once programmed should run without problem and the camera is a reliable hardware type. Recycling all the packaging in the correct ways, such as blue bags or general waste.

A good descriptive spec

3. Planning

The project will have a time scale of 25 weeks until completion, the first line of the Gantt chart below is the ideal time it will take to complete it and the second line of the section will be the actual time it takes to complete it. The deciding project ideas and the research took longer than was expected which had a knock on effect on everything else. The research into project ideas took less time than was expected and the research into alternative methods was also completed quicker than expected. The next section which is the produce and present presentation was completed on time. The project design is allocated the largest amount of time as the coding and building is part of this stage. The project testing took as long as expected on the Gantt chart, even with learning how to use the program. The report is allocated the time it has due to all the information already being in documents, it just needs to be sorted and written up properly.

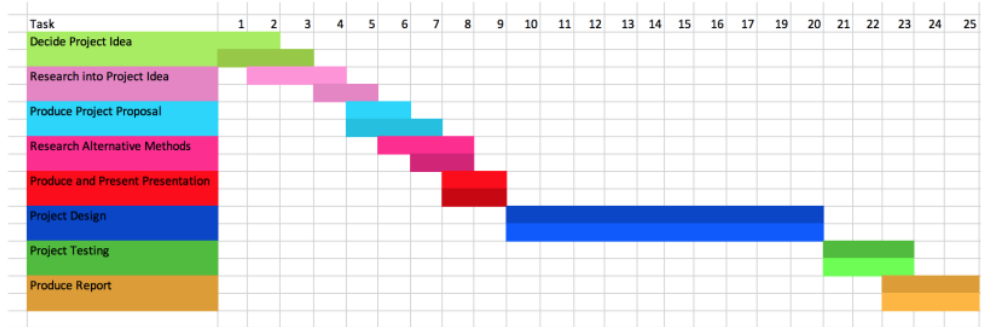


Figure 1 Gantt Chart

Good Gantt chart with comparison timelines

The testing was completed on time because it only needed 12 tests, tests 1 – 8 were testing for the components being covered and uncovered and test 9 – 12 were moving the board around underneath the camera to make sure the script can still work. The software passed the tests four of the tests were when the components were covered and the other four tests were when the components were not covered. Then in two of the test from 9 to 12 the components were covered and the board was moved underneath the camera, it failed on these two tests. Then in the other two tests the components were covered but the board was moved, it passed on these two tests.

Explanation accounts for the timeline

4. Solutions Considered

The two solutions that were considered to solve the problem. The first solution was a PLC with a Cognex camera and the second solution was using LabVIEW to pattern match. These are shown and explain below.

This should have a heading!

The first solution is Cognex Corporation is the world leader in machine vision technology. It is used in all types of industries, from pharmaceuticals to food and beverages. The In-Sight vision system has an advanced vision tools that has a high speed acquisition and quick processing time, the system is used for inspecting, identifying and also guiding the parts. The system can be used in a large variety of models of the Cognex family. The DataMan is handheld high performance, fix-mount reader which is designed to read 1D barcode reader and 2D code reader. It allows for easy recalls and helps to avoid mislabelling the projects. (DirectINDUSTRY, N/A)[Relevant source](#)

The In-Sight system is a compact vision that doesn't require an external processor or a separate camera, the system is self-contained. This is the industrial grade system that is standard to that system, it has a die-cast aluminium and a stainless steel case which withstand any vibrations that would happen. The four cameras in this range are the In-Sight 7000, In-Sight Micro, In-Sight 5000 and In-Sight 500. The In-Sight 7000 is a small and tough camera which has a powerful vision tools, integrated lighting and optics, and many more. The In-Sight Micro is a small and smart, it can meet a variety of demands of the machine vision. The In-Sight 5000 is the rugged, reliable and a powerful camera, the camera has colour, line scanning, ID and many more. The In-Sight 500 is used for inspecting parts on an ultra-high speed processing. (Bibus, N/A)



Figure 2 Cognex Camera

Good explanation with application examples explained

This should have a heading!

The second solution is LabVIEW pattern matching, there are three application areas for the pattern matching software, there is alignment, gauging and inspection. The alignment area is used to find the position and the orientation of the object and the features of this is used to find the reference points. The gauging area is used for measuring the lengths, angles, dimensions. The product will be rejected if the measurements are outside of the tolerance levels. The inspection area is where any flaws are detected and if the flaws are found then its failed, however if it doesn't have any then the product will pass. (NI, 2017)

For the vision assistant to work it must access a powerful tool for the testing the processing applications. A template needs to be made so that the software knows what colour or shape the product is for it to scan. (NI, 2017)

[Relevant sources](#)

5. Final Solution Design and Implementation

The budget for this project is £2650. Below is a list of where the money will be allocated,

Camera £50

Software £1000

Daq Board £1500

Misc £100

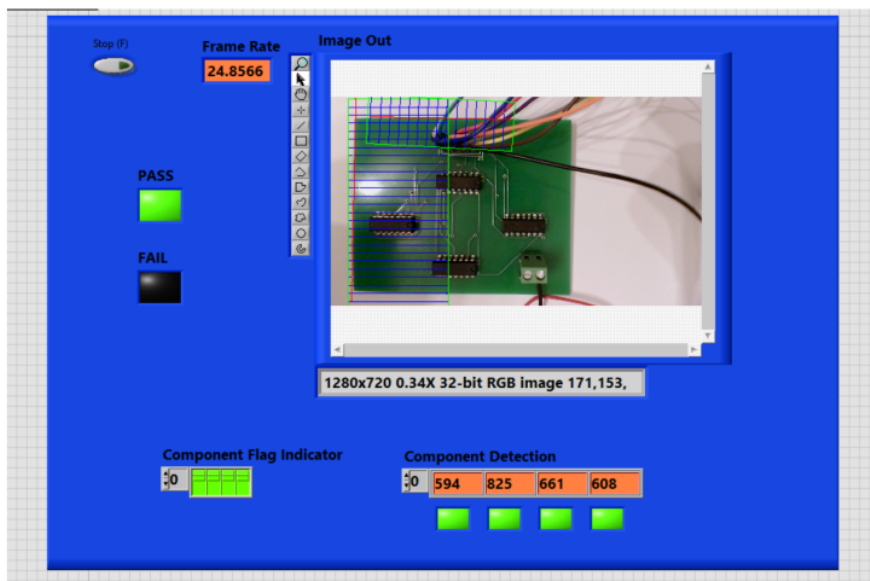
Costs are identified

Prototypes are not included!

This should be headed 'Implementation'

Evidence of the Main Screen and G Code

The front screen of the project is shown below, there are two squares on the side which indicate a pass or fail. The component flag indicator is whether the image has passed or fail and the component detection is the value of each of the components, the higher the value the better. Finally the image out is where the image is shown so the script can run through it. For example when the board is scanned and three components pass and one fails. then the fail will light up, the flag induction will have three green and one red box and the match value of match is shown in the detection.



Good front end design

Figure 3 Main Screen

The figure 4 below is the code that was designed for the LabVIEW software to scan the board for the components that it is looking for, each of the boxes which are used for it to run an explanation is further down in the report.

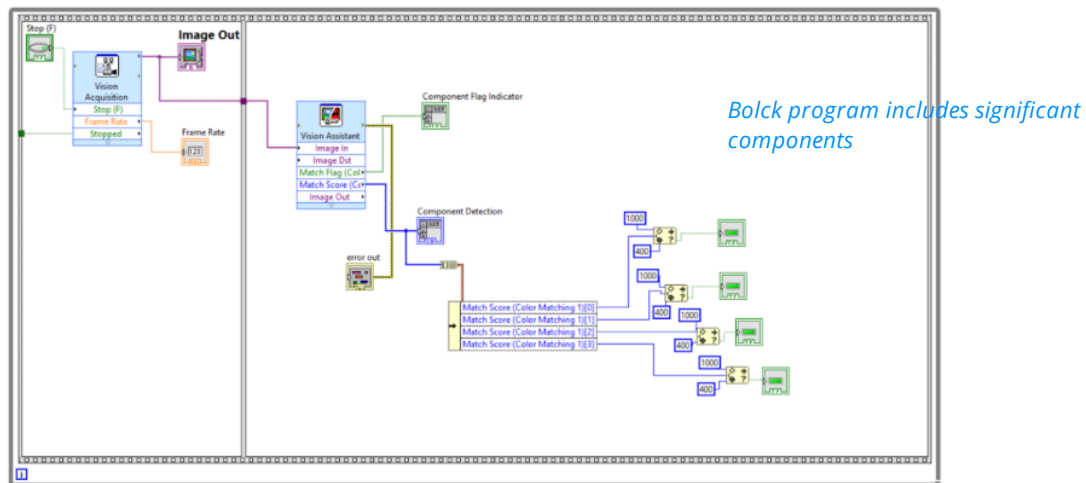


Figure 4 G Code

Evidence of Zoomed in G Code

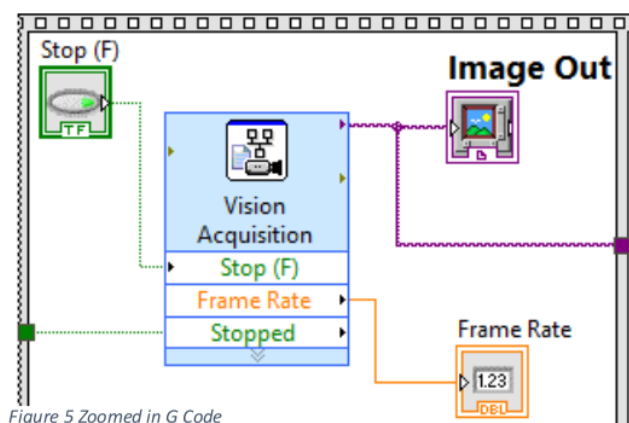


Figure 5 Zoomed in G Code

referenced in this block. The second block connected to Vision Acquisition block is the frame rate, is where the frame rate for the circuit is stored.

The first block that can be seen in the code is the Vision Acquisition block, this is where selecting the acquisition source and configuring it using NI-MAQdx or NI-IMAQ. Once the acquisition is configured, the controls and the indicators are programmed into the LabVIEW software. The first block connected to the Vision Acquisition is the Image Out block this is shown in the purple line The block captures the image underneath the camera and is

Main blocks explained

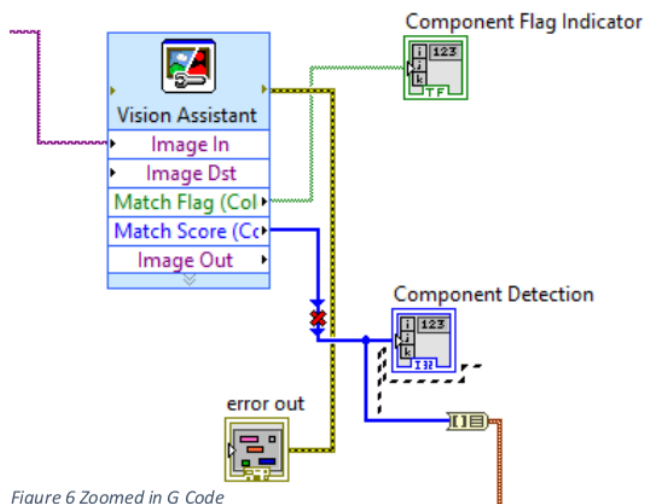


Figure 6 Zoomed in G Code

by the VI which can be used by other VIs. The error cluster has three elements which are Status which is a Boolean code error, Code which is a 32-bit integer error and Source that is a string error. The Component Flag Indicator block is a array that contains the Boolean information and whether it's a match between the input colour information and the specified region. The outputs of the Match Flag block is either TRUE or FALSE. The Component Detection block is where the match score is stored for each of the regions of the image scanned. The Match score has a 32-bit integer. The Array to Cluster converts the 1D array into a cluster of elements which have the similar type to the array element.

Block is explained

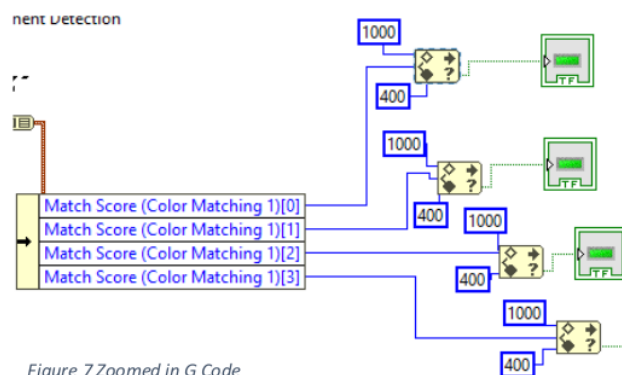


Figure 7 Zoomed in G Code

400. There are four Unbundle By Name blocks because there are four components on the board which need to be scanned for. The coercion only happens in the Compare Elements mode, it accepts time stamp values if all of the inputs have time values. The last block of the code is whether or not the component has passed or failed, Boolean is used to indicate a TRUE or FALSE.

Logic is explained

The Array to Cluster block is connected to the Unbundle By Name, this will return the cluster to the elements which match the name. This block will split into 4 In Range and Coerce blocks, the X value is the scanned score then the upper and lower limits blocks are connected. The upper limit block will be 1000 and the lower limit block will be

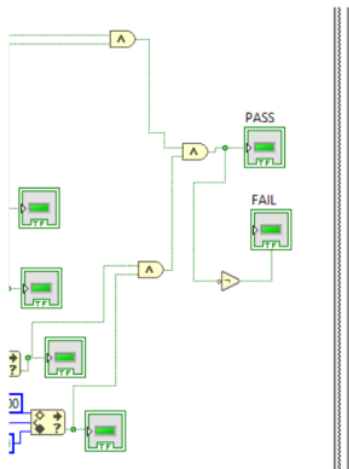


Figure 8 Zoomed in G Code

In the screenshot the PASS and FAIL is connected to the blocks so that the inductors can work properly. If the match score is below 400 then the inductors will turn red and it will be deemed as a fail however if the match score is above 400 then it will be classed as a pass.

Pass fail inverse logic is implemented and explained

Creating Vision Block

The first step is to find the vision block in the program menu, the vision block is in the 'Vision and Motion' section then click on the bottom corner and select 'Vision Acquisition' block. The vision acquisition block is used to add the template photos to get the script for the software to scan correct and the template of what the software is looking for.

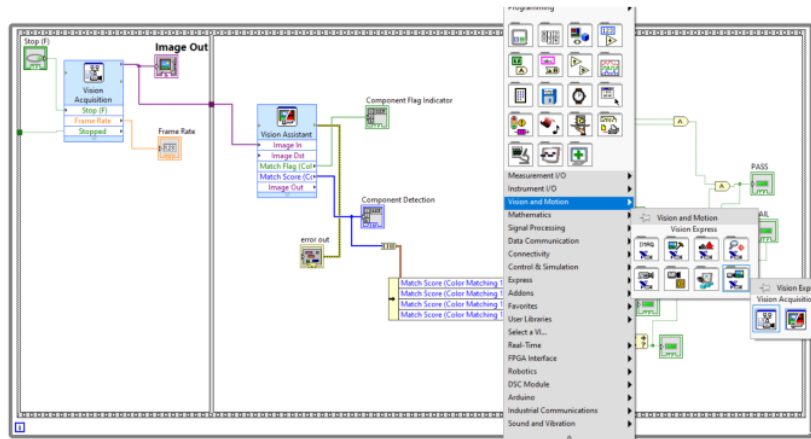


Figure 9 Vision Assistant

Implementation of vision assistant is evident

Setting up the USB camera

To set up the camera, first open up the Vision Acquisition and select the camera which is required, the one used for the project is a HUE camera. When selected the camera should display the image of what is underneath the camera.

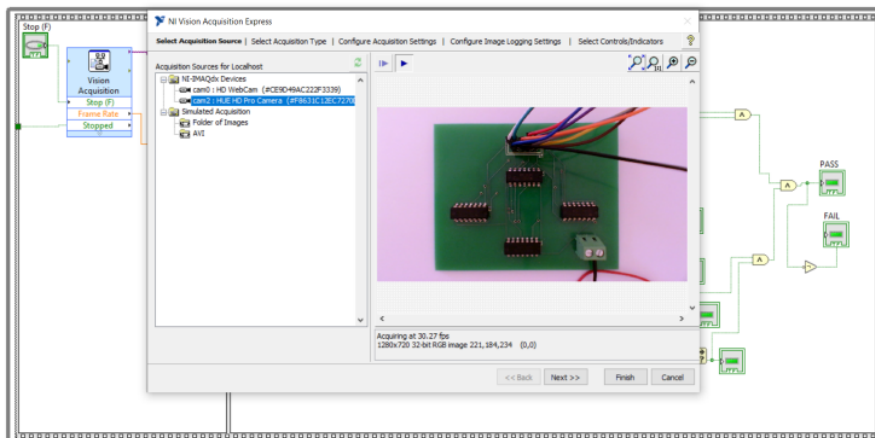


Figure 10 Setting up USB camera

Configuration is evident

The next step is selecting the acquisition type, the one for the project is a continuous acquisition with inline processing. This type is used for a continuous images, so the images are in actual time and can be changed if a fail happens.

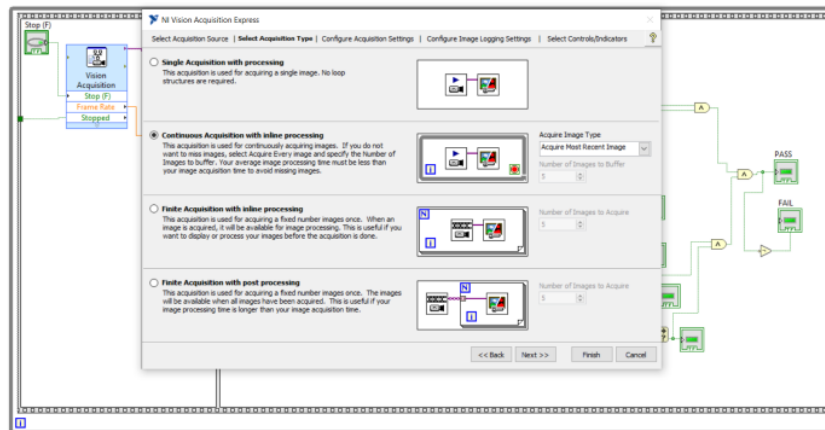


Figure 11 Selecting the Acquisition

The screen needs to be pasted onto the main screen so the images can be view on the screen with the indicators for pass or fail.

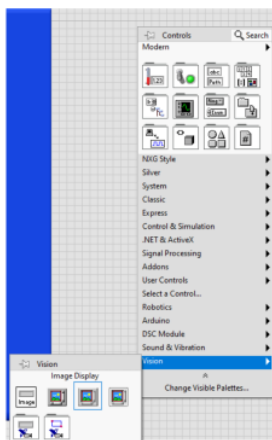


Figure 12 Setting up display screen

Flat Sequence Structure

To start the flat sequence structure first open up programming and select the third box down which is called 'Flat Sequence Structure'.

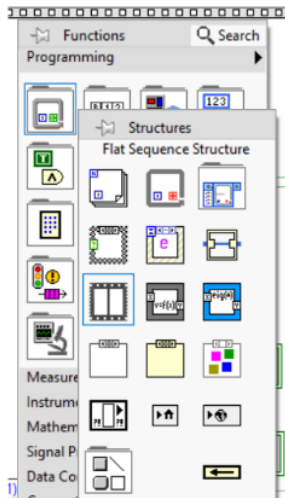
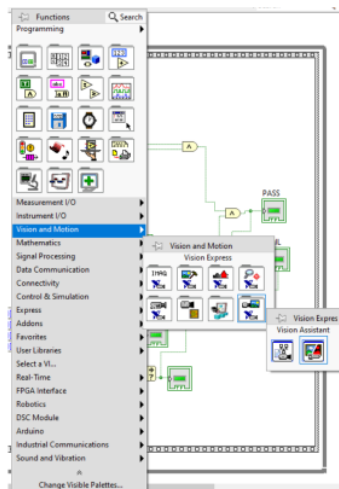


Figure 13 Flat Sequence Structure

Next is to open the Vision and Motion section, and then select the last box in the section, Vision Assistant. When that box opens up, select the Vision Assistant.



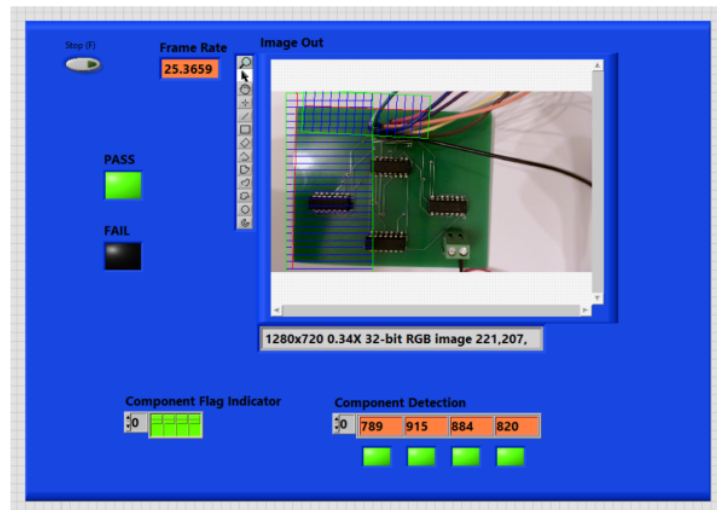
Evidence of configuration is included

Figure 14 Creating Vision Assistant

This should have a heading 'Test and analysis'

Pass and Fail Conditions

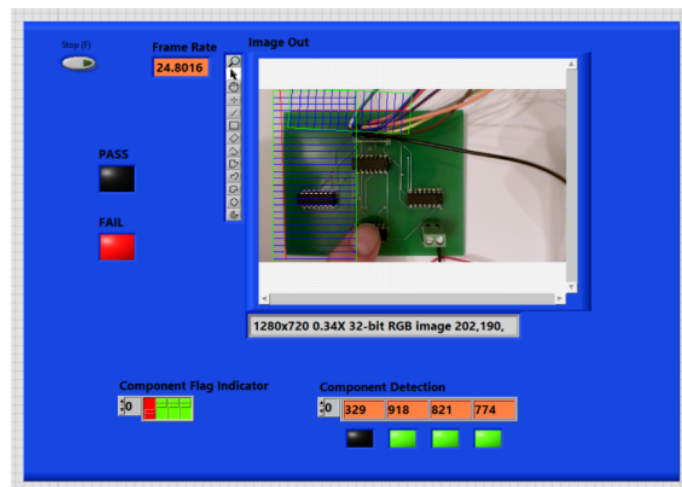
The scores on the component detection for this image is high, all of the lights are green which shows that the image has passed. This image is a pass because the detection component has a value above 400.



Good test evidence for pass condition

Figure 15 Pass Condition

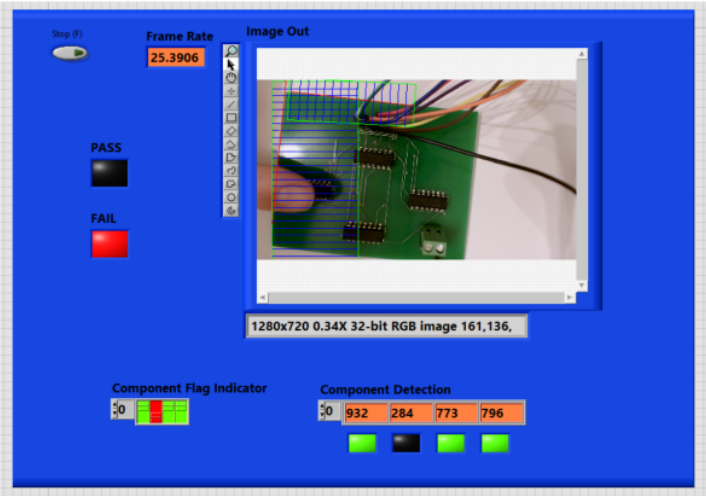
The first component is covered and the board will fail. The first indicator will turn red and the other three indicators will light up green, the first components value is also under 400 which the light wont light up.



Test evidence for fail condition

Figure 16 Fail Condition

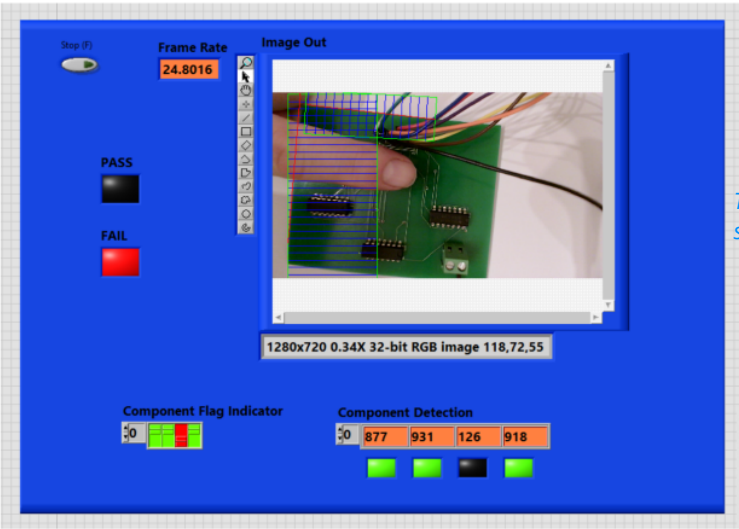
The second component is covered and the second indicator turns red. The second detector doesn't light up as the other three does.



Second component fail test is successful

Figure 17 Fail Condition

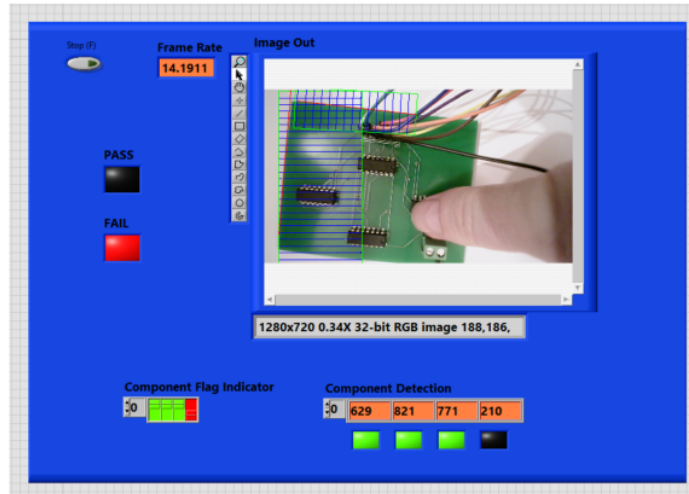
The screenshot shown below is also a failed test, the third component is covered and the detection value is below 400. The third flag indicator is red which is a fail for the board whether as the first, third and fourth are green.



Third component test is successful

Figure 18 Fail Condition

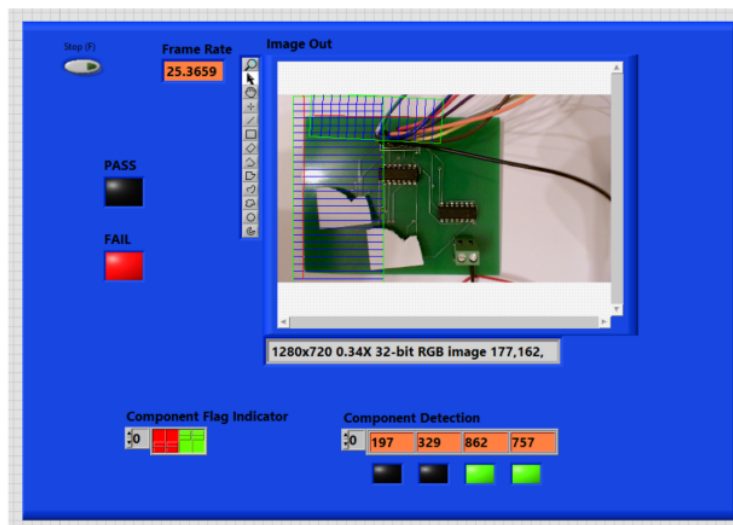
The image below is a failed test because one of the components has been covered. This is shown by having a value of below 400 as can be seen by the black square and the red indicator.



Fourth component test is successful

Figure 19 Fail Condition

Two components were covered and it was a failed test. The flag indicator is red for the first and second one and the third and fourth indicators turn green. Also the detection also shows a fail for the first and second lights are not on and the third and fourth are light up green.



Fail test is successful

Figure 20 Fail Condition

Implementation of script

The Script

The script is needed for the camera to run through all the stages needed to tell if the board will pass or fail. The figures and explanation below show how the script work.

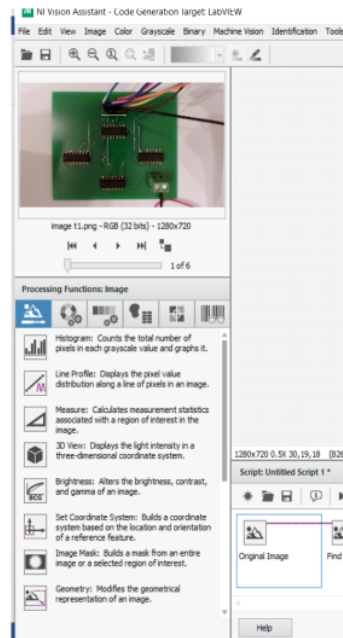


Figure 21 Opening Image Sequence

The first part of the script is the 'Open Image Sequence' this is where the images are storage ready for the next stages. The images were placed in here and checked that they were clear enough to read and use as a template, shown in figure 4 .

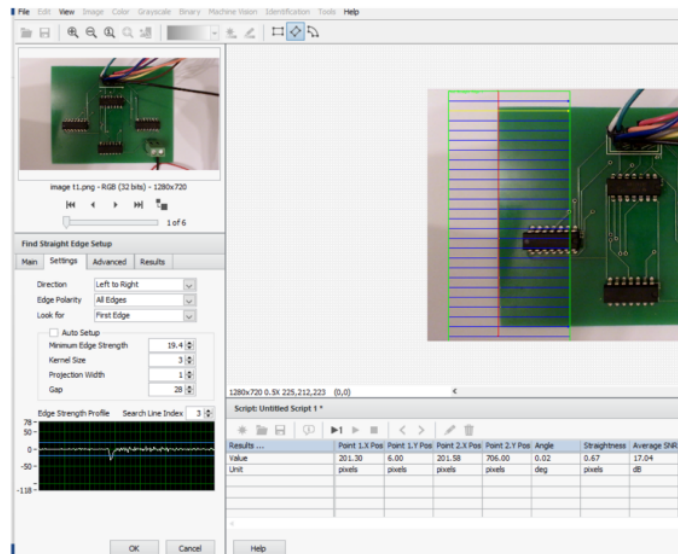


Figure 22 First Straight Edge

The second stage was 'First Straight Edge' in this stage an edge of the board was found and set that it could read wherever it was placed underneath the camera. This is shown in figure 5 the green box is surrounding it.

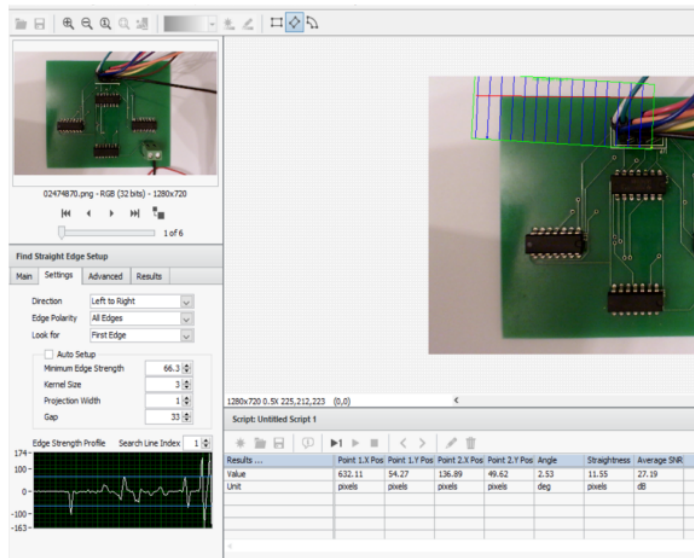


Figure 23 Second Straight Edge

The third stage was 'Second Straight Edge' this stage is the same as stage three, the second straight edge is the edge connecting with the edge from the previous stage. This is because a point has to be created, the straight edge is shown on figure 6 the green box is placed round it.

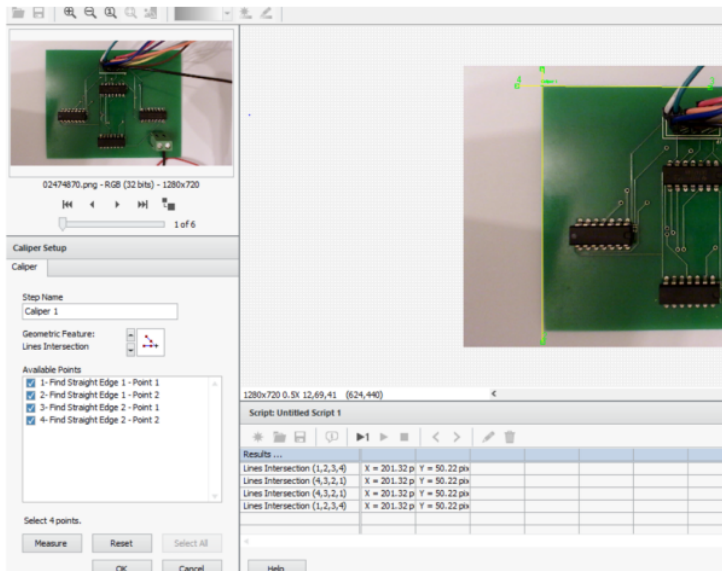


Figure 24 Caliper

The fourth stage is the 'Caliper' this stage is where the two straight edges meet from the previous two stages, this point is where the coordinators will be taken from to read where the components are on the board. This stage is shown in figure 7 in the appendix.

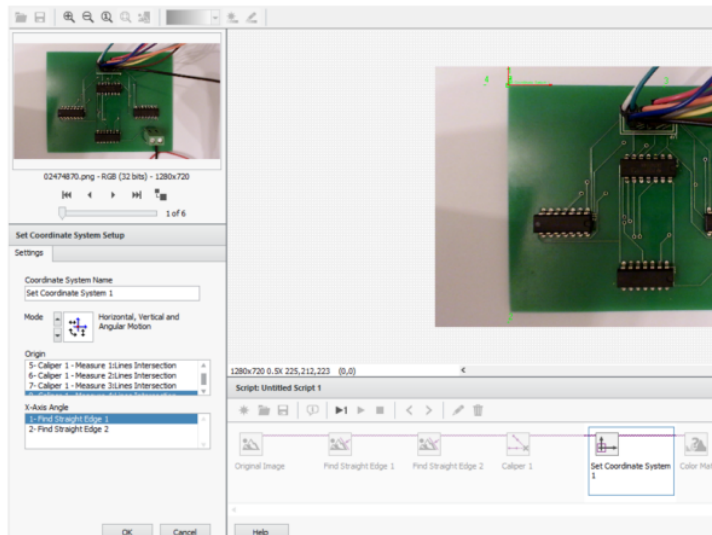


Figure 25 Set Coordinate System

The fifth stage is the 'Set Coordinate System' this stage the coordinators will be taken from the two straight edges. The coordinators will be set so that the components will be in the same distance away from the edge, this is shown in figure 8 in the appendix.

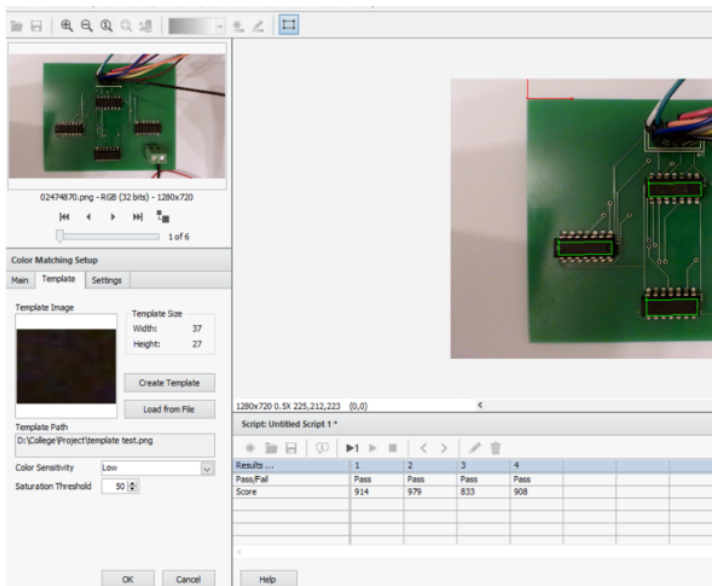


Figure 26 Pattern Matching

The sixth stage is 'Pattern Matching' in this stage the template is made for the software to register if the components are the correct one. The template is shown in the black square on the side of the screenshot. Also, the template is the green rectangles on the board the score of each of the rectangles is shown in the bottom of the figure 9.

The figure 10 below shows the final implementation of the script using DAQ assistant. This is the where the photos are placed and template for the components are created. The steps for this script are written in the section above and the testing process is carried out and documented in the figures shown below

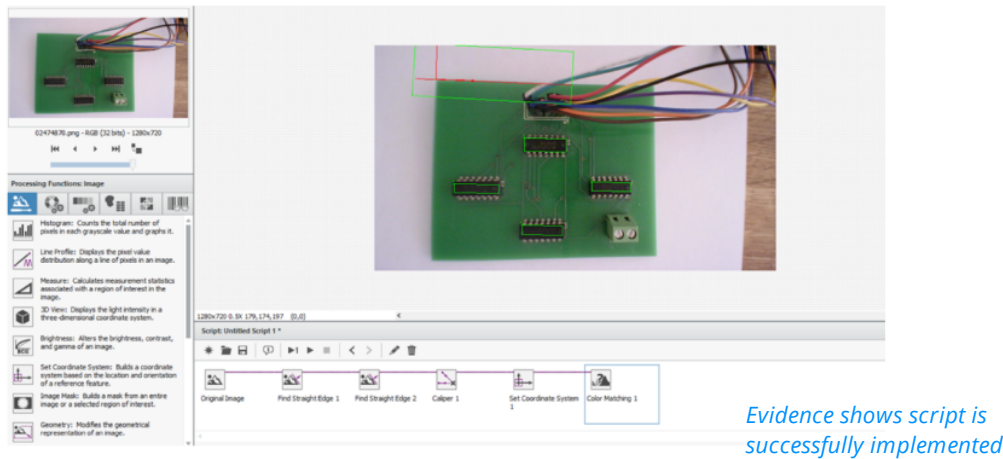


Figure 27 Final Script Implementation

After the script had been filled out and was loaded up, the board was placed underneath the camera and the four components were scanned by the machine. The scanned components can be seen in figure 11, the four green rectangles show that the software has registered them. The red lines on the edge of the board shows that the second and third stages worked correctly as it registered them as well as the green which is around the red lines.

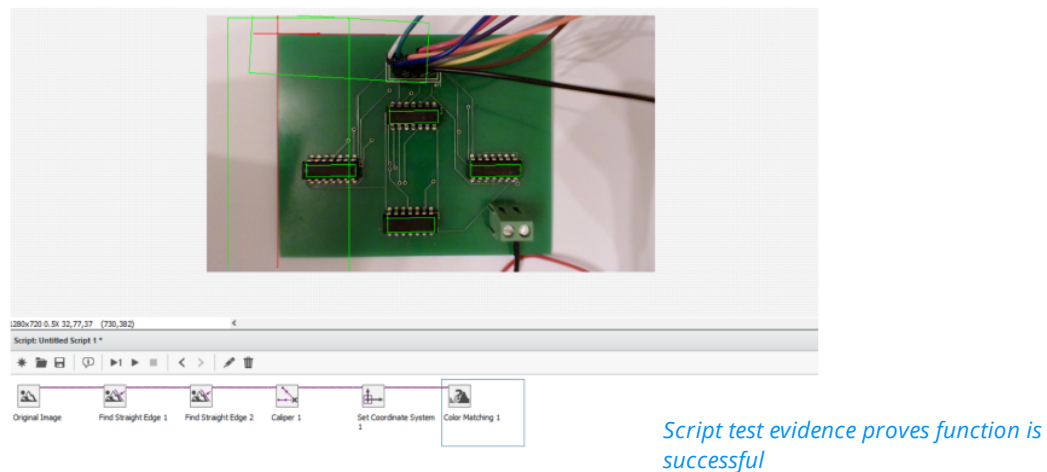


Figure 28 Test Image 1

Once the stages had gone through and been completed, the board was slightly moved and the image was processed again. The reason was to ensure that the board can be moved around the script will still work. As seen in figure 12 below the process still worked with the two straight lines showing and creating a caliper. The software also shows that the components have been scanned as the four green rectangles showing on them.

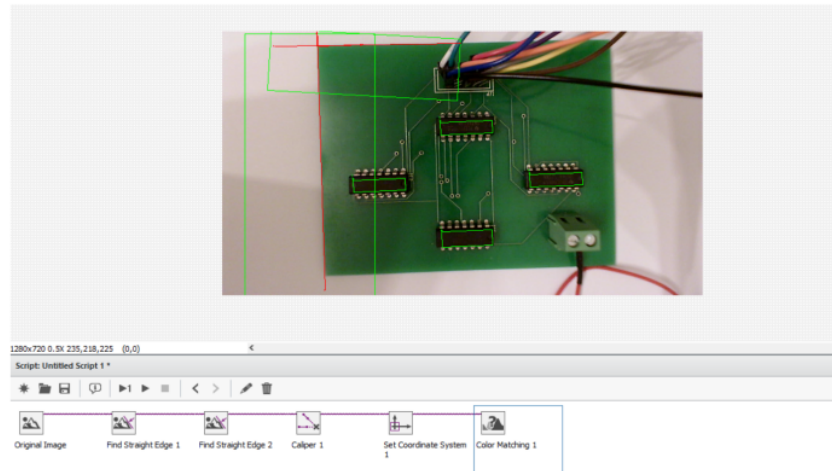


Figure 29 Test Image 2

The other figures, shown in the appendix, the board is again moved to check that it can still read the edges and components after it is moved. The red lines all shown up on the figures and the green boxes are around them. The four rectangles which are placed on the components also show up on the figure as well proving that they have been read by the software. In figure 12 the caliper can be seen, this point is from the two red lines and is where the coordinators came from.

Setting the control for LabVIEW

The controls shown below are what is needed for the main screen to present the relevant information on the screen. For the inductors to work the Match Flag and Match Score are selected so when the Match Score reaches above 400 then the match will happen.

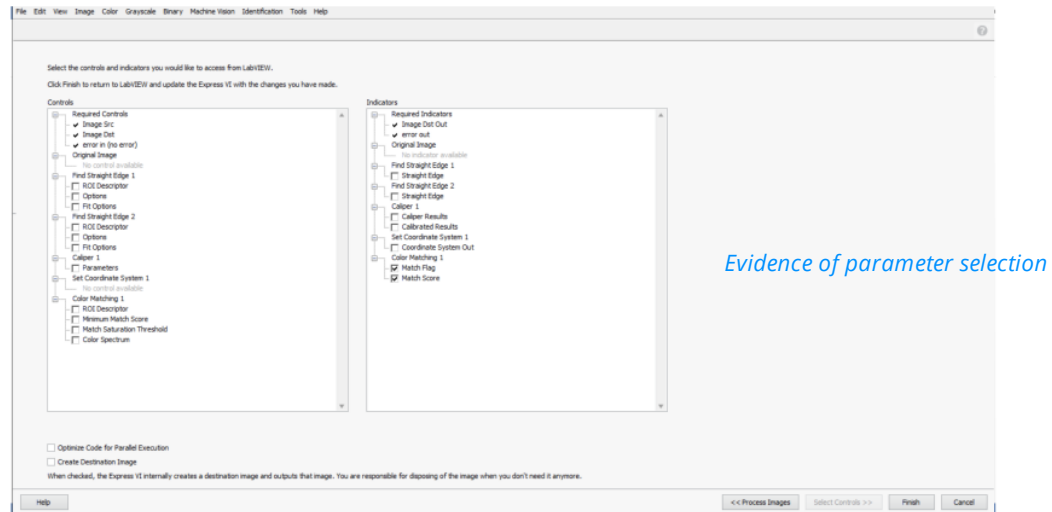


Figure 30 LabVIEW Control

6. Testing and Results

In the table below is the data for the machine vision detection fault test. In one test the component was covered and it failed the test, then the next one the component was uncovered and it passed the test. This is shown in tests 1 and 2 of the table, test one was where one of the components was left uncovered and it showed all green markers in the indication, which means there was no fault detected and as soon in test one the chip was covered the fault was detected and signalled a failure by one of the green markers turning red. This test was continued until test 8 was complete. Then for test 9 to 12 the board was moved to make sure that it could be moved into different positions in the cameras view, to check if the script could still go through the sequence.

Test No.	Original Image	No of IC's Identified	Error	Output Fault detected Y/N	Test P/F
1	IC's x 4	1 – 4	IC1	N	P
2	IC's x 4	1 – 4	IC1	Y	F
3	IC's x 4	1 – 4	IC2	Y	F
4	IC's x 4	1 – 4	IC2	N	P
5	IC's x 4	1 – 4	IC3	N	P
6	IC's x 4	1 – 4	IC3	Y	F
7	IC's x 4	1 – 4	IC4	N	P
8	IC's x 4	1 – 4	IC4	Y	F
9	IC's x 4	1 – 4	Angle Left	Y	F
10	IC's x 4	1 – 4	Angle Right	Y	F
11	IC's x 4	1 – 4	Angle Left	N	P
12	IC's x 4	1 – 4	Angle Right	N	P

A very good set of test results

7. Discussion, Conclusions and Recommendations

The results for the project are what was expected to happen, so test 1 and 2 are testing for the same components. Test 1 is when the component is left uncovered and test 2 is when the first component has been covered over. Test 3 is the component is covered and it fails, and test 4 are when the second component is left uncovered which it then passes the test. The tests 5 and 6 are testing for the third component, for test 5 the component wasn't covered and it passed the test as it was over the matching threshold for matching the template which was created. In test 6 component three was covered and that component then failed the test. Test 7 and test 8 were testing the fourth component, test 7 was a pass as the fourth component wasn't covered and test 8 was a fail because that component was covered. Tests 9 to 12 are when the board is moved around the screen and see if the camera then picks up the lines and starts the script. Test 9 is where the board is moved left without any of the components covered and test 10 is where the board is moved right without the components being covered. Test 11 is where the board is moved left but one or more of the components have been covered and test 12 is where the board is moved right but one or more of the components are covered these tests are tested because if there is more than one mistake happens on the board then it will still pick up the mistake.

The project is fit for purpose as the software runs correct and goes through all the necessary steps for the script. The maintenance of the project is easy as the camera would be easy to access and all the software is on the laptop which doesn't need to be connected.

Conclusion includes significant points. There are no recommendations

8. Reference List and Bibliography

Bibus, N/A. *In-Sight Vision Systems Product Guide*. [Online]

Available at: https://www.bibus.ro/fileadmin/product_data/Cognex/In-Sight_Product_Guide.pdf

[Accessed 13 05 2023].

DirectINDUSTRY, N/A. *Product Recall Prevention with Vision Solutions from Cognex*. [Online]

Available at: <https://trends.directindustry.com/cognex-corporation/project-23274-147784.html>

[Accessed 14 05 2023].

NI, 2017. *Pattern Matching*. [Online]

Available at: <https://forums.ni.com/t5/Example-Code/Pattern-Matching/ta-p/3734017>

[Accessed 13 05 2023].

NI, 2017. *Pattern Matching*. [Online]

Available at: <https://forums.ni.com/t5/Example-Code/Pattern-Matching/ta-p/3734017>

[Accessed 12 05 2023].

Limited number of sources

9. Appendix
Evidence for Task 6 Testing

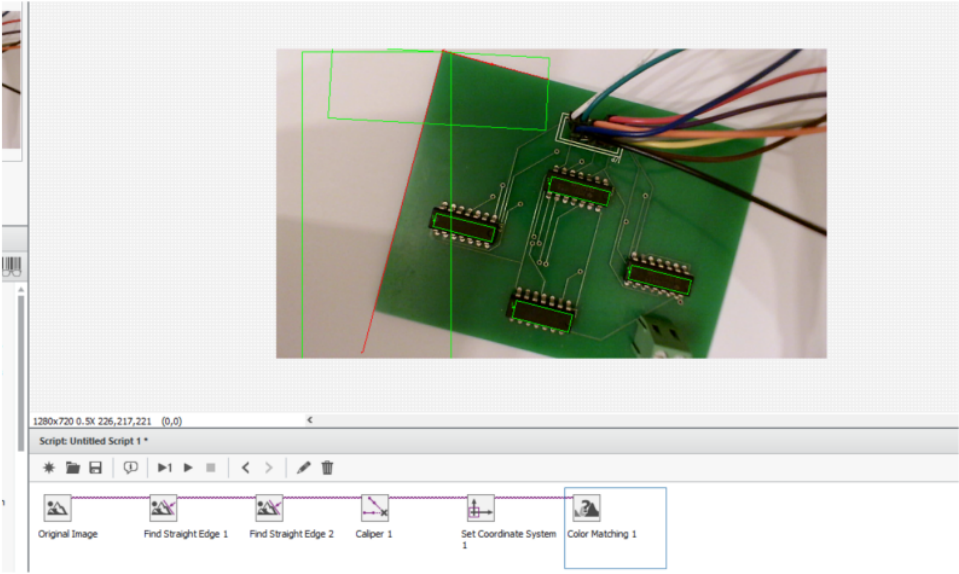


Figure 31 Test Image 3

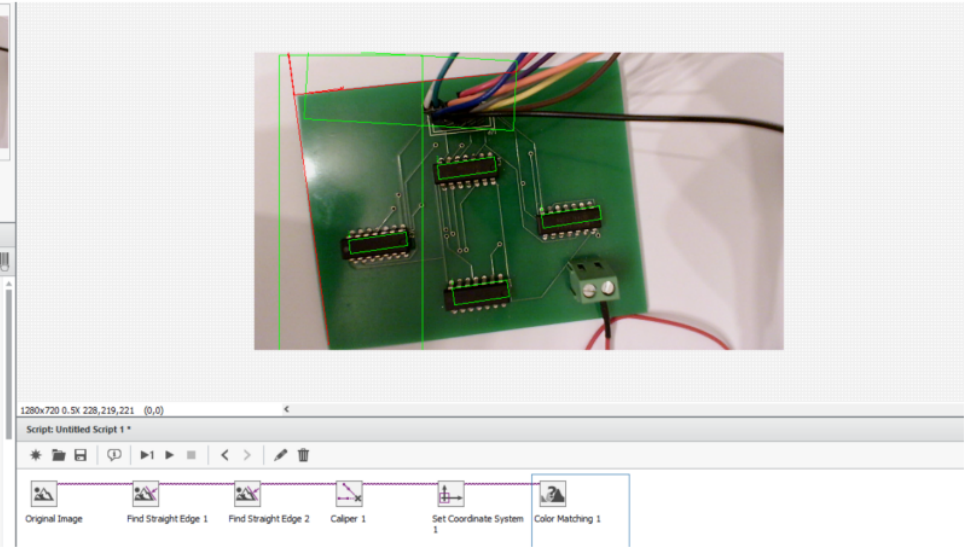


Figure 32 Test Image 4

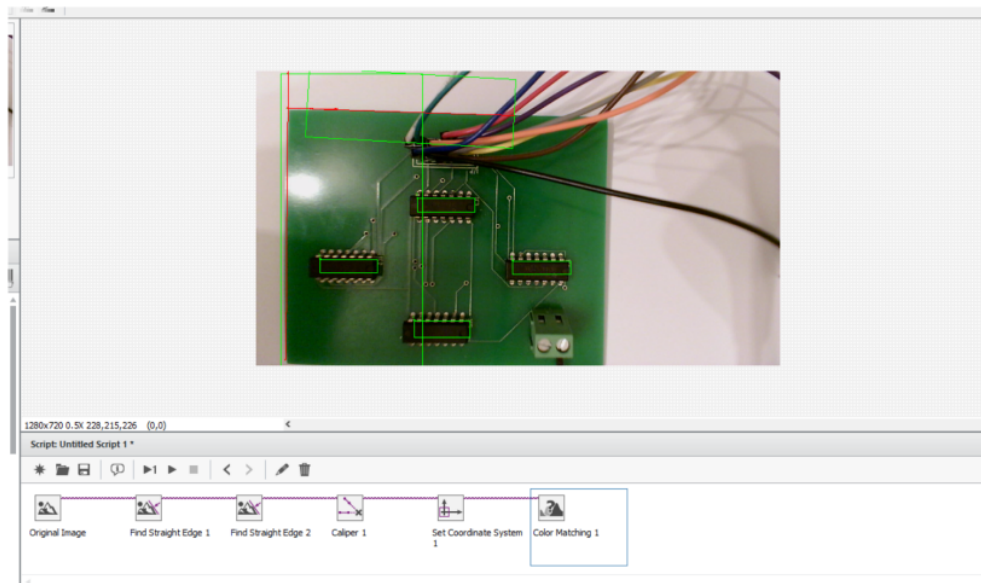
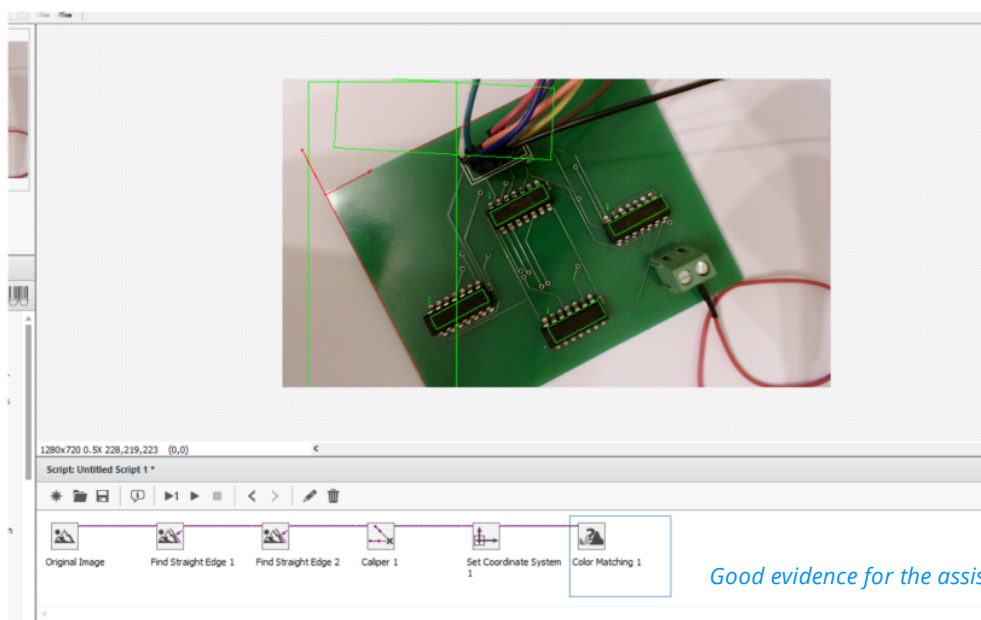


Figure 33 Test Image 5



Good evidence for the assistant script test

Figure 34 Test Image 6

FINAL GRADE

GENERAL COMMENTS

Instructor

58/100

PAGE 1

PAGE 2

Text Comment. Good table of contents

Text Comment. Good extensive list of figures

PAGE 3

Text Comment. Spelling and grammar errors

Text Comment. Relevant sources used

Text Comment. The intro includes some significant points

PAGE 4

Text Comment. Significant point

Text Comment. A good descriptive spec

PAGE 5

Text Comment. Good Gantt chart with comparison timelines

Text Comment. Explanation accounts for the timeline

Text Comment. This should have a heading!

Text Comment. Relevant source

Text Comment. Good explanation with application examples explained

Text Comment. This should have a heading!

Text Comment. Relevant sources

Text Comment. Costs are identified

Text Comment. Prototypes are not included!

Text Comment. This should be headed 'Implementation'

Text Comment. Good front end design

Text Comment. Bolck program includes significant components

Text Comment. Main blocks explained

Text Comment. Block is explained

Text Comment. Logic is explained

Text Comment. Pass fail inverse logic is implemented and explained

Text Comment. Implementation of vision assistant is evident

Text Comment. Configuration is evident

PAGE 12

PAGE 13

Text Comment. Evidence of configuration is included

PAGE 14

Text Comment. This should have a heading 'Test and analysis'

Text Comment. Good test evidence for pass condition

Text Comment. Test evidence for fail condition

PAGE 15

Text Comment. Second component fail test is successful

Text Comment. Third component test is successful

PAGE 16

Text Comment. Fourth component test is successful

Text Comment. Fail test is successful

PAGE 17

Text Comment. Implementation of script

PAGE 18

PAGE 19

PAGE 20

Text Comment. Evidence shows script is successfully implemented

Text Comment. Script test evidence proves function is successful

PAGE 21

Text Comment. Evidence of parameter selection

Text Comment. A very good set of test results

Text Comment. Conclusion includes significant points. There are no recommendations

Text Comment. Limited number of sources

Text Comment. Good evidence for the assistant script test

GRADING FORM: CSG-RUBIC

CERYS JOSEPH

0

RESEARCH-INFORMED LITERATURE



Background research in limited.
Solution research is more extensive and include
relevant images and sources.
The number of sources are limited

11/20

KNOWLEDGE AND UNDERSTANDING



Spec is descriptive and requirements are identified. 6.5/10

Solutions considered are explained in good detail with supporting ref sources and images.

There is a good understanding of requirements. 6.5/10

13/20

ANALYSIS



Prototypes and tests are missing

There is extensive testing carried out for final implementation. 6.5/10

Conclusion includes some good significant points, but there are no recommendations. 4.5/10

11/20

PRACTICAL APPLICATION AND DEP.



Planning timeline and comparison timeline is evident and explained. There is no test plan included. 3/5

Solutions considered are valid and explained. 6/10

Prototypes are missing from the implementation and testing.

There is good evidence of the final implementation. 6/10

14/25

SKILLS FOR PROFESSIONAL PRAC.



Report has some implementation sections in the incorrect locations.

There are headings missing and some grammatical errors

9/15

