

Project Design, Implementation and Evaluation

by Emyr Dyer

FILE	7311_EMYR_DYER_PROJECT_DESIGN__IMPLEMENTATION_AND_EVAL UATION_177909_1804199797.PDF (3M)		
TIME SUBMITTED	15-MAY-2020 07:38PM (UTC+0100)	WORD COUNT	8019
SUBMISSION ID	127831209	CHARACTER COUNT	48796



Course: BSc (Hons) Electrical and Electronic Engineering
Module: SGEE6003 – Major Project
Assignment: 2. Project Design, Implementation and Evaluation
Name: Emyr Wyn Dyer
Student No. 1503008
Date: 15/05/2019
Word Count: 5555

CONTENTS

Acknowledgements.....	1
1 Executive Summary	2
2 Introduction	3
3 Design	4
3.1 User Interface	4
3.2 Network Data Flow	5
3.3 Pseudo Code	8
3.4 Security.....	10
4 Development.....	11
4.1 Budgeting	11
4.2 Network Configuration	12
4.3 PLC Re-Configuration	13
4.4 PLC Program.....	14
4.4.1 Function Gens	14
4.4.2 Input-Output Interfacing	16
4.4.3 SCL Alarm Conditions	18
4.4.4 Motor Speed Control and Webserver Enable	19
4.5 ABA	21
4.5.1 CSS Styling	21
4.5.2 Index	22
4.5.3 ‘Master’ Page	26
4.5.4 ‘Pneumatic’ Page	33
4.5.5 ‘Process’ Page.....	38
5 Implementation	41
5.1 Device Connection	41

5.2	Memory Card	41
5.3	Adding Final ABA to PLC Program	44
5.4	Downloading PLC Program	46
6	Testing and Results.....	49
6.1	Test	49
6.2	Results	51
7	Evaluation and the Wider Issues	55
7.1	Reflecting on FDS	55
7.1.1	Hosting	55
7.1.2	Browser	55
7.1.3	Displaying Data	55
7.1.4	Controllability and Response	55
7.1.5	Virtual LAN	55
7.1.6	Password Protection.....	56
7.1.7	Meeting Budget.....	56
7.2	Health & Safety	56
7.3	Product Safety	56
7.4	Development Risk	56
7.5	Environmental Considerations	57
7.6	Gantt Chart	57
8	Conclusions and Recommendations	58
9	Works Cited	59
10	Glossary of Terms	60

FIGURES

Figure 3.1 User Interface Sketch	4
Figure 3.2 Browser Navigation Buttons	4
Figure 3.3 Page Mock-up	5
Figure 3.4 Network Topology	5
Figure 3.5 Ethernet Frame	6
Figure 3.6 DOM	6
Figure 3.7 CSSOM	7
Figure 4.1 Budget Sheet	11
Figure 4.2 IP Address reserved for PLC	12
Figure 4.3 Warm Restart Setting	13
Figure 4.4 Function Call 1	14
Figure 4.5 Function Call 4	15
Figure 4.6 Triangle Wave	15
Figure 4.7 'webdata' Datablock	16
Figure 4.8 IO Interfacing	16
Figure 4.9 Current Output	17
Figure 4.10 Voltage Percent	17
Figure 4.11 SCL instructions	18
Figure 4.12 Analogue variables	18
Figure 4.13 Motor Control in Main	19
Figure 4.14 Webserver Enable	19
Figure 4.15 Default Tag Table	20
Figure 4.16 Typical Folder Hierarchy	21
Figure 4.17 Colouring in w3.css	22
Figure 4.18 index.html lines 1 to 29	23
Figure 4.19 index.html lines 31 to 54	24
Figure 4.20 index.html lines 56 to 59	24
Figure 4.21 index.html lines 61 to 79	25
Figure 4.22 index.html lines 82 to 86	25
Figure 4.23 plc101master.html lines 1 to 8	26
Figure 4.24 plc101master.html lines 36 to 58	26
Figure 4.25 plc101master.html lines 65 to 83	27
Figure 4.26 plc101master.html lines 85 to 96	27

Figure 4.27 plc101master.html lines 98 to 119.....	28
Figure 4.28 plc101master.html lines 122 to 129.....	28
Figure 4.29 plc101master.js	29
Figure 4.30 CIMcond.json	29
Figure 4.31 plc101chart.js lines 1 to 13.....	29
Figure 4.32 plc101Chart.js lines 14 to 35	30
Figure 4.33 plc101chart.js lines 36 to 54	31
Figure 4.34 plc101chart.js lines 56 to 68	31
Figure 4.35 plc101chart.js lines 70 to 76	32
Figure 4.36 plc101tri.json	32
Figure 4.37 plc102pneumatic.html line 1	33
Figure 4.38 IOWrite.html	33
Figure 4.39 plc102pneumatic.html lines 52 to 57.....	33
Figure 4.40 plc102pneumatic.html lines 76 to 87.....	34
Figure 4.41 plc102pneumatic.html lines 88 to 115.....	34
Figure 4.42 plc102pneumatic.html lines 117 to 147.....	35
Figure 4.43 plc102pneumatic.html lines 149 to 161.....	35
Figure 4.44 plc102pneumatic.html lines 164 to 171.....	36
Figure 4.45 plc102pneumatic.js lines 1 to 16	36
Figure 4.46 plc102pneumatic.js lines 76 to 86	37
Figure 4.47 plc102pneumatic.js lines 88 to 111	37
Figure 4.48 IOresult2.html	37
Figure 4.49 led0.png (green) and led1.png (red)	37
Figure 4.50 plc103process.html lines 53 to 56	38
Figure 4.51 plc103process.html lines 74 to 95	38
Figure 4.52 plc103process lines 97 to 114.....	38
Figure 4.53 plc103process.html lines 117 to 124	39
Figure 4.54 plc103process.js lines 1 to 15	39
Figure 4.55 plc103process.js lines 17 to 41	40
Figure 4.56 IOresult1.html	40
Figure 5.1 DHCP enabled and IP address reserved	41
Figure 5.2 Before Deletion	42
Figure 5.3 Deletion Prompt.....	42
Figure 5.4 Warm Restart	42
Figure 5.5 Set to Program Mode	43

Figure 5.6 Generate Blocks.....	44
Figure 5.7 Block Generation	44
Figure 5.8 ABA Files	45
Figure 5.9 PLC in blank state due to empty memory card	46
Figure 5.10 PLC ready for download	46
Figure 5.11 Program downloaded	48
Figure 5.12 Memory Card Programmed	48
Figure 6.1 index.html	51
Figure 6.2 Master Page	51
Figure 6.3 Pneumatics Page	52
Figure 6.4 Corresponding IO in Watch Table	52
Figure 6.5 Process Control Page	53
Figure 6.6 index.html on Mobile Device	54
Figure 6.7 index.html with Menu open	54
Figure 6.8 Master Page on Mobile Device	54
Figure 6.9 Master Page Scrolled	54

ACKNOWLEDGEMENTS

I would like to give my deepest appreciation to everyone in the Coleg Sir Gâr Engineering Faculty who over the last 6 years helped me develop the knowledge required to develop this Project.

I give special thanks to Course Tutor and Academic Supervisor, Mr. Robert Davies, whose suggestions of developing a project based in Industry 4.0, helped me to consolidate my knowledge of information systems with knowledge of automated systems, to develop this project.

I would also like to give thanks Mr. Robert Jenkins who helped with advice on which direction to take my research and the discussions that were had regarding solutions for the project.

I would also like to thank Mr. Robert Davies and Mr. Robert Jenkins again for permission to use equipment used to develop the project, which have been crucial during this unprecedented time.

And of course, I would like to thank my colleagues Mr. Sam Harries and Mr. Kris Payne who have been invaluable in their support, particularly during the last 2 years.

This should summarise all that was planned and completed in the project, together with a statement on success. You have written it as an introduction to the report.

I EXECUTIVE SUMMARY

Spelling

Within this report, a means of developing an Industry 4.0 solution for the CIM System is described. This solution is called the Automation Browser Application or ABA.

Judgments are made based on typical Web Development design to produce diagnostics pages that can produce the same level of information as a SIMATIC HMI unit.

The process of configuring a Development Network is shown, as well as how to prepare the S7-1200 PLC, to handle a web-based application reliably.

Sections of HTML and JavaScript code that makes up the ABA is included in this report, along with explanation of functionality with PLC Step 7 Programming.

2 INTRODUCTION

What were the aims and objectives of the project?

This report describes the process of Developing a Webpage application called the ABA, to be used in conjunction with a PLC Program running on a Siemens S71214C PLC via its Webserver feature.

The technology and methods used are not entirely typical of traditional PLC programming disciplines but are typical of traditional Web Development. This report covers the process of using both disciplines to develop an Industry 4.0 product.

The report introduces AJAX, CSS, HTML and JavaScript/Query and how they may be used to implement the objective set out in the FDS document for the Project.

You have introduced the report not the project .

3 DESIGN

3.1 USER INTERFACE

Based on the conclusions from the Project Definition document, the selected solution for addressing the requirements for an alternative to the Simatic HMI is the Webpage solution or Automation Browser Application (ABA).

To make an initial parity to the existing solution, pages will need to be made for each station in operation, namely:

- Master Station (PLC101)
- Pneumatic Station (PLC102)
- Process Station (PLC103)
- Hydraulic Station (PLC104)

A Launch Page would be beneficial to both 'open' the ABA and serve as index.html, which should always be the first page to open (Fred Gerantabee, 2007).



Figure 3.1 User Interface Sketch

As per traditional website design, it should be possible to navigate to any page at any time without using the 'back' button on the browser (Sherry Bishop, 2012).

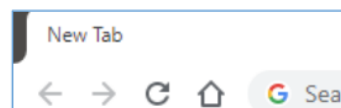


Figure 3.2 Browser Navigation Buttons

To get around this issue, a navigation bar must be made available on all pages of the ABA.

As it is expected that the ABA should be accessible on a touchscreen device i.e. phone or tablet, a means of dynamically re-orienting the content of each page must be implemented.

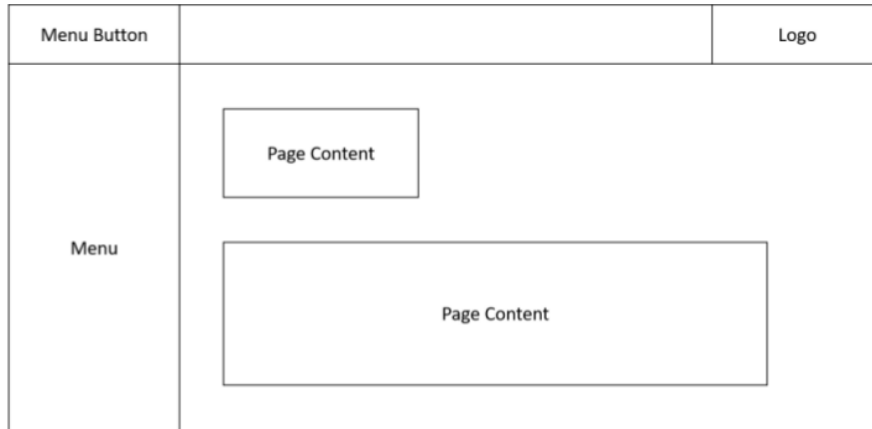


Figure 3.3 Page Mock-up

3.2 NETWORK DATA FLOW

The way data flows around the system is an important consideration, as it determines what hardware and software technologies are implemented to address the requirements.

For development purposes, a typical Wi-Fi router is sufficient for communication between PLC and Wireless Devices. Most domestic Wi-Fi routers have firewall protections built-in, merely requiring minor configuration to get up and running.

The topology for the Network is shown below:

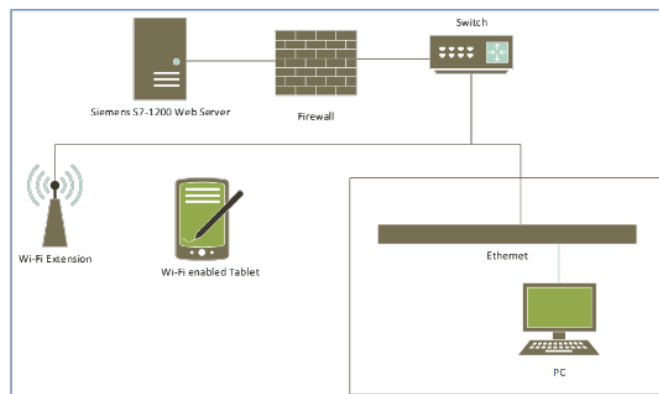


Figure 3.4 Network Topology

Along the network links depicted above, data is transmitted via 'frames' to recipients according to their unique MAC addresses. To transmit data to another LAN, the frame is further encapsulated into a 'packet' which is addressed using an IP address (Charles E. Spurgeon, 2014).

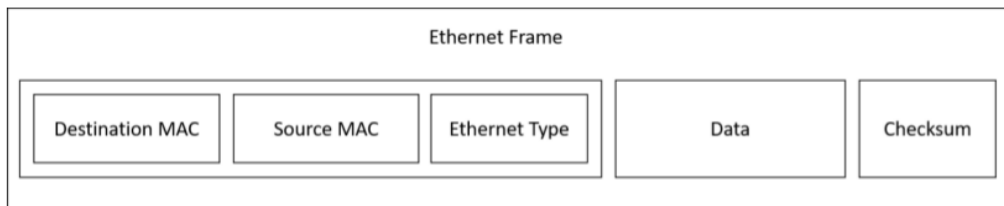


Figure 3.5 Ethernet Frame

In the shown example (Figure 3.4), the PLC fills the role of the webserver, which holds the source webpages and running data for Automation processes carried out by the PLC.

For the browser to load index.html, a request is sent either by clicking a link or entering the address of the page in the address bar. After receiving the request, the webserver sends the HTML file as frames to the requesting device. After the whole HTML file is received by the browser, the file is parsed, to determine if there are other elements required (CSS, JavaScript, etc) to render the webpage, and if there are, then these elements will also be requested from the webserver (Campbell, 2014).

Once all the resources are received by the browser, it begins to build the Document Object Map (DOM)

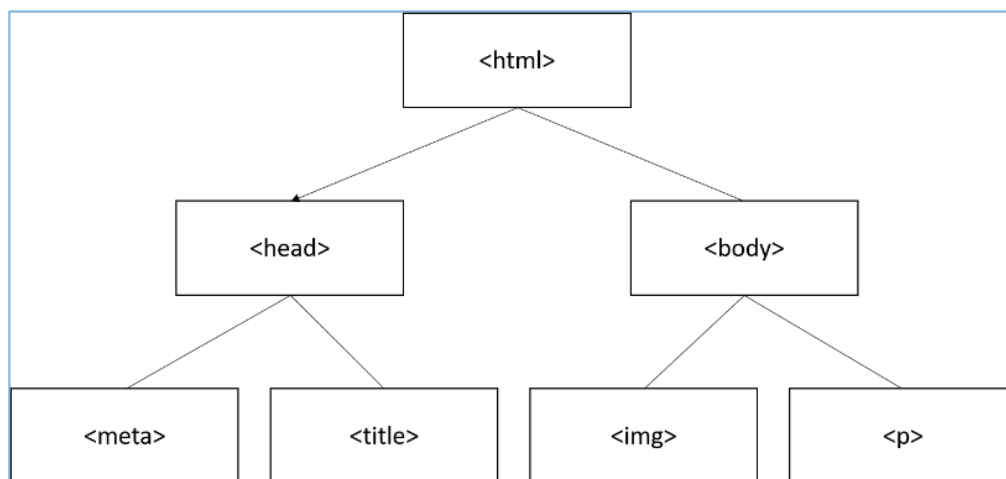


Figure 3.6 DOM

Next, the browser builds the CSS Object Map (CSSOM). This map describes how styling is applied to each HTML element.

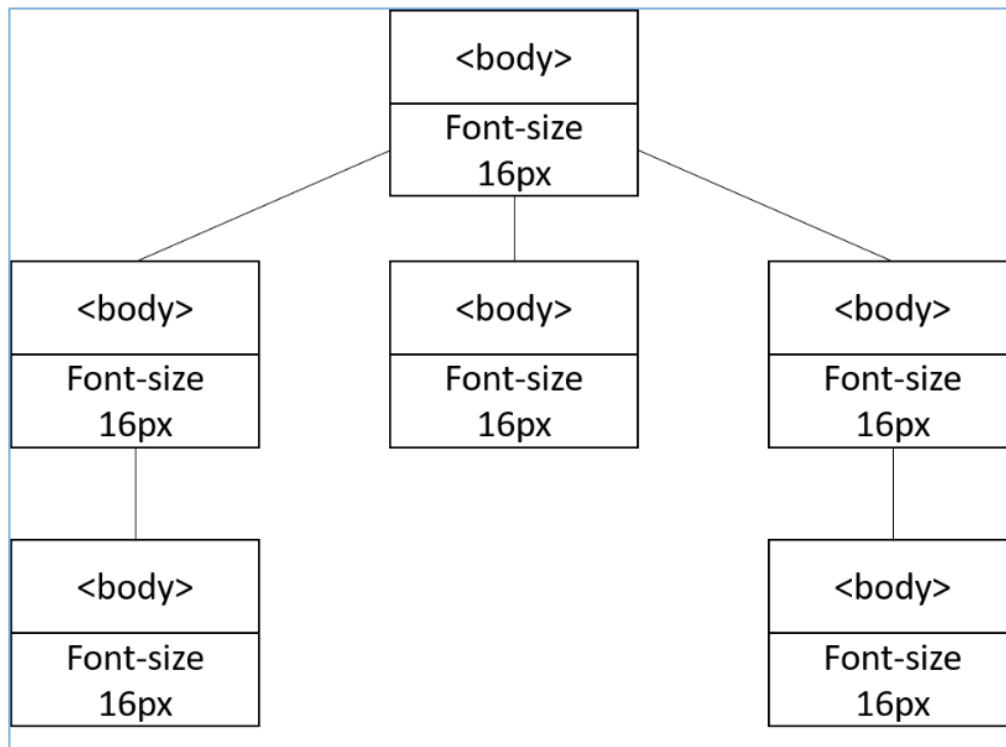


Figure 3.7 CSSOM

The browser then combines both maps and from them, builds the webpage according to the 'recipe' of the DOM and the 'rules' of the CSSOM (Grigorik, 2013).

To load the other pages of the ABA, the whole process repeated, and again if the page is manually 'refreshed' by pressing F5.

3.3 PSEUDO CODE

Below is the typical code structure for each webpage of the ABA. There will be some differences according to functionality of each page. Pseudo code does not necessarily fit the syntax of the final code but helps in laying out how the code will be written (B. B. Agarwal, 2010).

`<html>` (The opening tag for a webpage, everything in-between is the recipe for the webpage)

`<head>`

`<meta>` (the meta tag has coding for initial page scaling)

`<stylesheet reference>` (tag contains location for the CCS document)

`<style>`

(additional styling that is not universal to other pages may be placed between these tags)

`</style>`

`<title>`Page Title (Appears on page tab) `</title>`

`</head>`

(the content between the head tags is not visible to the user, but contains information used by browsers to build the page and is also used by search engines to make the page ‘discoverable’)

`<body class=grey>`

(the class aspect in this tag is tied to CSS, where the rule of a grey background is defined and applied to the page)

`<div class=bar dark grey>`menu button & logo`</div>`

(the bar attribute applied to the div tag as defined in the CSS, will apply a dark grey bar across the top of the page. Between the tags, instructions for positioning of a menu button and logo is written.)

`<nav class=sidebar>` (nav tag declared with CSS rule for a ‘sidebar’ applied)

```
<link>Page1</link>

<link>Page2</link>

</nav>

<div class=main>

    (between the 'main' div, is the canvas for the code unique to the page to
    fulfil the purpose of the page.)

    <header>Page Title (if required) </header>

    <div class=quarter>

        (a box on the page that takes up 25% of the space for the 'main'
        div. Any content put between these tags will not stretch the box
        beyond 25%. A rule called 'third' could limit box size to 33%.)

    </div>

</div>

<script>JavaScriptLibrary.js</script> (Add-on libraries should be called at the
end of the page, to give the browser opportunity to build the webpage, before
applying scripted programming)

<script>pageScript.js</script> (To maintain readability of code, additional and
unique scripting for the webpage should be kept in a separate JavaScript file.
This also makes it easier to apply code to other projects.)

</body>

</html> (Closing of html tag)
```

3.4 SECURITY

During development, the LAN will be kept 'offline' so that no connection of the Internet is possible. It should be noted that all devices should have some form of anti-virus or anti-malware software installed as individual devices will likely at some point be connected to the Internet.

Regarding testing and development of the ABA by downloading code to the CIM System, existing security measures established by the Network Administrators of Coleg Sir Gâr will apply as dictated by the Network Security Policy.

4 DEVELOPMENT

4.1 BUDGETING

Below is a spreadsheet detailing what components have been acquired as per the WBS document. Costings have been avoided where equipment and software as acquired on-loan or through self-donation.

Project Lead:		Emyr Wyn Dyer						
		Materials						
WBS	Task	Units	£/Unit	Fixed Cost	Budget	Actual	Under/Over	Notes
1	Research							
1.1	Hardware Requirements							
1.2	Software Requirements							
2	Specification							
2.1	PLC Peripherals							
2.2	Networking Standards							
2.3	Programming Languages							
3	Design							
3.1	User Interface							
3.2	Network Data Flow							
3.3	Pseudo Code							
3.4	Security Policy							
4	Development				£ 200.00			
4.1	Hardware							
4.1.1	Network Config	1	£ 34.99	£ 34.99	£ -	£ -	£ -	Donated spare router
4.1.2	User Access							
4.1.3	PLC Re-config	1	£ 291.60	£291.60	£ -	£ -	£ -	PLC on-loan for development
4.1.4	Documentation							
4.2	Software							
4.2.1	PLC Program	1	£ 138.24	£ 138.24	£ -	£ -	£ -	Laptop with TIA Software on-loan for development
4.2.2	Jquery/AJAX Coding		£ -	£ -	£ -	£ -	£ -	Open Source, no licence-fees
4.2.3	GUI CSS Styling							
5	Implement				£ 200.00			
5.1	Hardware							
5.1.1	Install S7 Memory Card	1	£ 169.00	£ 169.00	£ 169.00	£ 169.00	£ 31.00	Memory card expansion for s71214C PLC
5.1.2	Connect Network Devices							
5.2	Software							
5.2.1	Download New PLC Program to PLC							
5.2.2	Download ABA to PLC							
6	Test							
6.1	Hardware							
6.1.1	Network Reliability/Response							
6.1.2	System Performance							
6.2	Software							
6.2.1	Fidelity of Shown Data							
6.2.2	Interface Response							
7	Documentation							
7.1	Network Setup Guide							
7.2	Coding Utility Guidance							
7.3	Blank Page Templates for Development							
Total Cost:					£ 169.00			

Figure 4.1 Budget Sheet

4.2 NETWORK CONFIGURATION

NETGEAR SMARTWIZARD settings Wireless-N 150 ADSL2+ Modem Router model DGN1000

LAN Setup

LAN TCP/IP Setup

IP Address: 192.168.0.1
IP Subnet Mask: 255.255.255
RIP Direction: Non
RIP Version: RIP

☐ Access Router Management Interface on additional port 8080 (NAT-disabled mode only)

☒ Use Router as DHCP Server

Starting IP Address: 192.168.0.1
Ending IP Address: 192.168.0.1

Address Reservation

#	IP Address	Device Name	MAC Address
1	192.168.0.190	S71214C-190	28:63:36:A2:83:86

Add Edit Delete

Apply Cancel

Figure 4.2 IP Address reserved for PLC

DHCP has been enabled, where all network devices use an IP address allocated by the router save for the PLC. The PLC has a static address set on the PLC itself. To ensure that there are no addressing conflicts, the router has the PLC's address reserved according to the PLC's MAC Address (Soper, 2014). As access to hardware and software such as Network Switches and Server hardware required for more advanced Network Management is made difficult due to lockdowns issued by the British Government as a result of the COVID-19 Global Pandemic (Rigby, 2020), implementation of User Access protocols have not been possible under the circumstances.

4.3 PLC RE-CONFIGURATION

To ensure no unexpected issues arose during development, only minor settings were changed when setting up the PLC. The IP address was kept the same with accommodations made through IP Addressing with the Router. Start-up was modified as per guidance for installing the S7 Memory Card before any programming was uploaded to the PLC (Siemens, 2014).

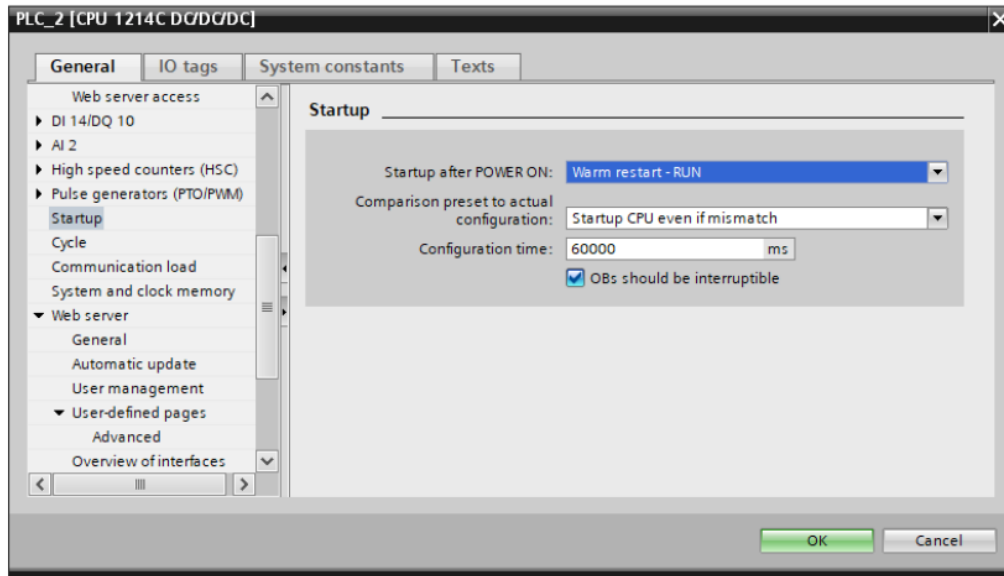


Figure 4.3 Warm Restart Setting

4.4 PLC PROGRAM

Because of issues surrounding the COVID-19 Pandemic of 2020, access to the CIM System located at Coleg Sir Gâr was no longer possible due to lockdown of businesses for the sake of public safety (Coleg Sir Gâr, 2020).

To mitigate the problem, a Developmental PLC Program was written, to provide Inputs and Outputs that development of the ABA could continue.

4.4.1 Function Gens

The first step was to produce suitable outputs for the ABA that would change over time. To achieve this, a Counter was created using Timers that would trigger after a set time of 5 seconds. The result would output to the 'webdata' data block, as shown in Figure 4.4.

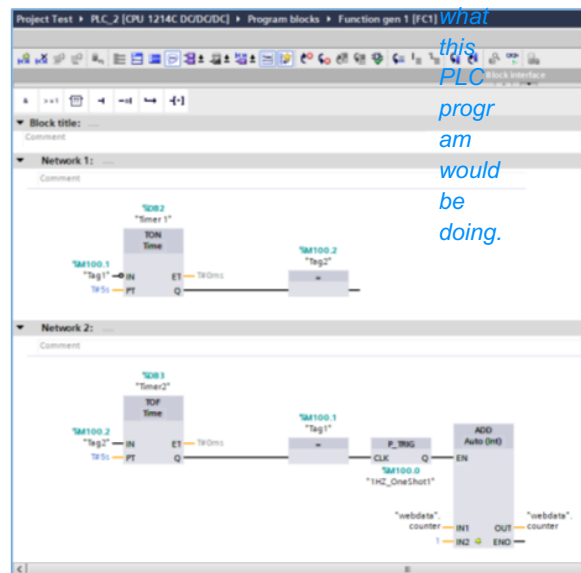


Figure 4.4 Function Call 1

Four other Function Calls would be written similarly to provide similar Counter operations, but with different time delays set (1, 2 and 10 seconds).

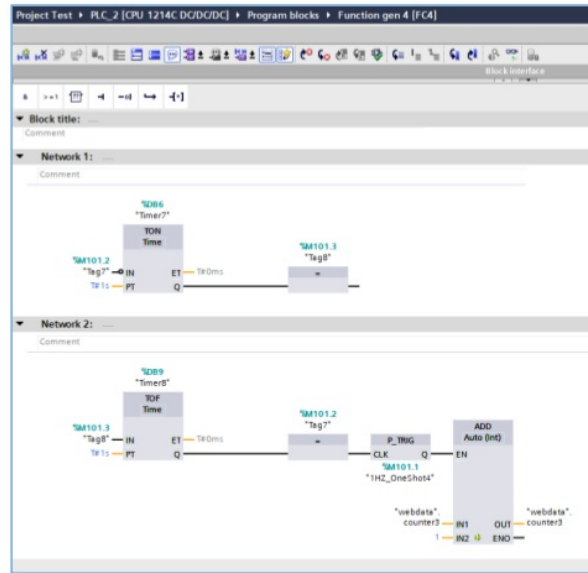


Figure 4.5 Function Call 4

A fifth Function Call was written to produce Chart data in the form of a Triangular Wave. This was achieved by setting maximum count limits of 100 to -100, where once a maximum count was reached, the direction (countup, countdown) would change.

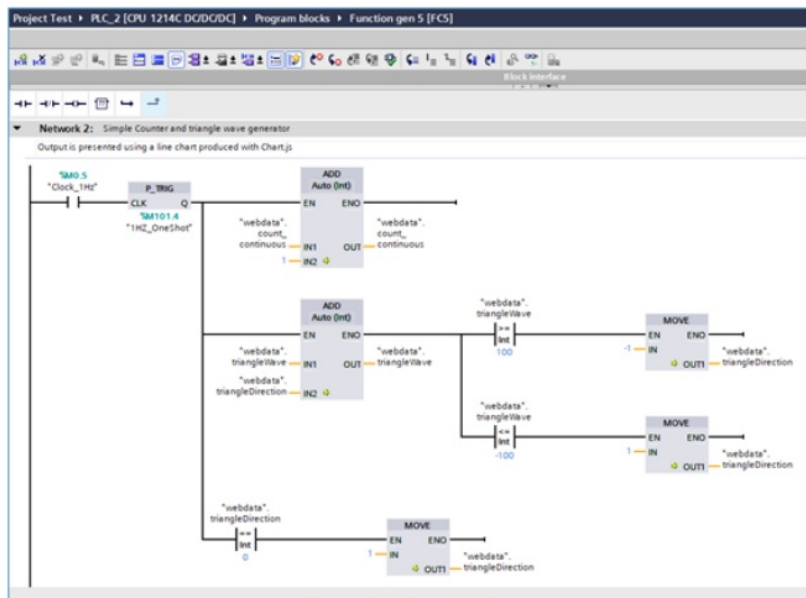


Figure 4.6 Triangle Wave

All the variables stored within 'webdata' were set as integers.

	Name	Data type	Start value	Retain	Accessible f...	Write...	Visible in ...	Setpoint	Com...
1	Static								
2	counter	Int	0		☒	☒	☒		
3	web_ok	Int	0		☒	☒	☒		
4	counter1	Int	0		☒	☒	☒		
5	counter2	Int	0		☒	☒	☒		
6	counter3	Int	0		☒	☒	☒		
7	myInt	Int	0		☒	☒	☒		
8	count_continuous	Int	0		☒	☒	☒		
9	triangleWave	Int	0		☒	☒	☒		
10	triangleDirection	Int	0		☒	☒	☒		
11	Motor_PWM	Int	0		☒	☒	☒		

Figure 4.7 'webdata' Datablock

4.4.2 Input-Output Interfacing

To send Analogue data to the ABA, a Function call called 'Input-Output Interfacing' was written to read in Voltage data that is manually adjusted via a Potentiometer. Using the Analogue readings, a gauge graphic could be animated in the ABA.

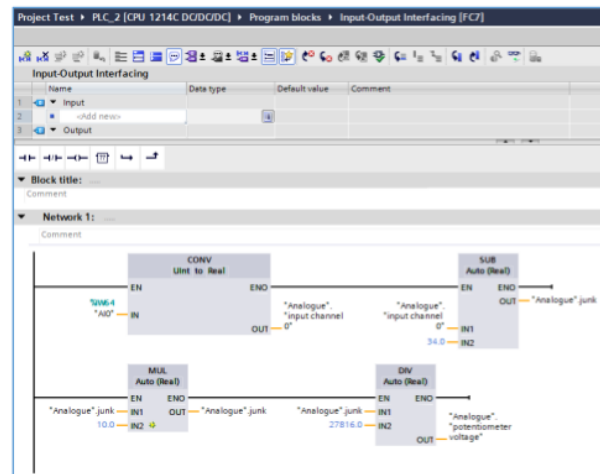


Figure 4.8 IO Interfacing

It is not clear what part the DC motor is playing in this project

By reading the Voltage values produced by the potentiometer, the DC motor current is then calculated using $\frac{V}{R} = I$ where $R = 0.03$ as shown in Figure 4.9.

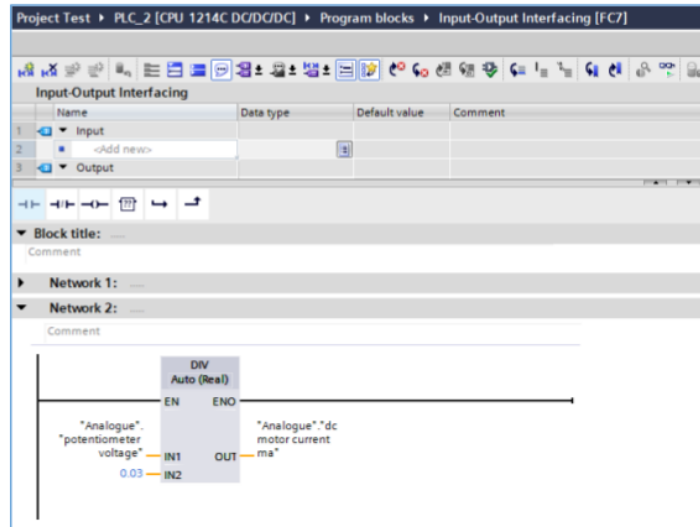


Figure 4.9 Current Output

Volts are also expressed as a percentage by multiplying the Potentiometer Voltage by 10.

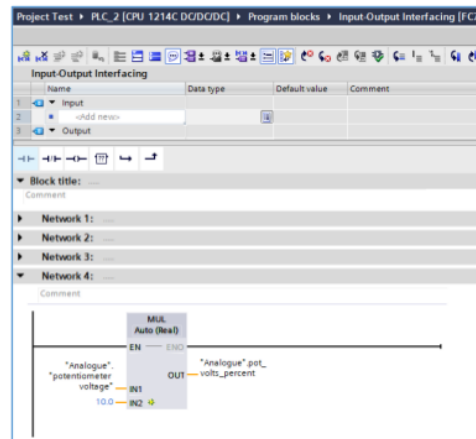


Figure 4.10 Voltage Percent

4.4.3 SCL Alarm Conditions

Although not implemented in the ABA, alarm conditions were added to the PLC Program via SCL, where current levels would trigger conditional statements and set 7 Boolean variables accordingly. This feature could be used to build a segmented gauge in the ABA with segments made using animated PNG indicators.

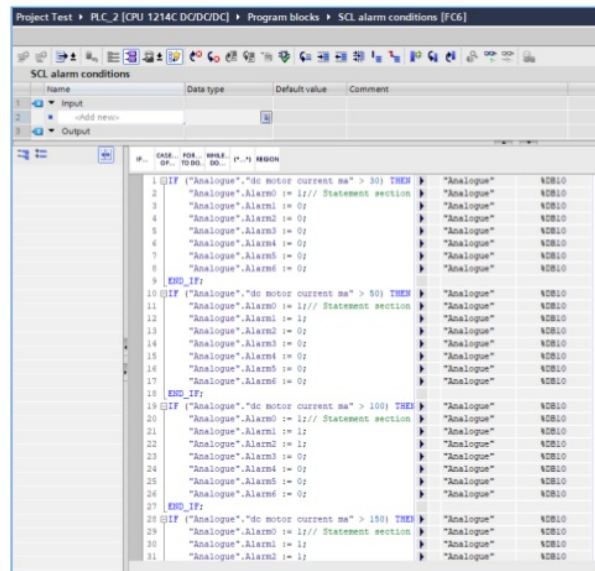


Figure 4.11 SCL instructions

The screenshot shows the 'Analogue' variables table. It has columns for Name, Data type, Start value, Retain, Accessible f..., Writ..., Visible in ..., and Setpoint. The table lists 14 variables, including 'input channel 0' through 'input channel 1', 'junk', 'potentiometer voltage', 'pot_volts_percent', and seven 'Alarm' variables (Alarm0 through Alarm6), and 'dc motor current ma'.

Name	Data type	Start value	Retain	Accessible f...	Writ...	Visible in ...	Setpoint
Static							
input channel 0	Real	0.0		☒	☒	☒	☐
input channel 1	Real	0.0		☒	☒	☒	☐
junk	Real	0.0		☒	☒	☒	☐
potentiometer voltage	Real	0.0		☒	☒	☒	☐
pot_volts_percent	Real	0.0		☒	☒	☒	☐
Alarm1	Bool	false		☒	☒	☒	☐
Alarm2	Bool	false		☒	☒	☒	☐
Alarm3	Bool	false		☒	☒	☒	☐
Alarm4	Bool	false		☒	☒	☒	☐
Alarm5	Bool	false		☒	☒	☒	☐
Alarm6	Bool	false		☒	☒	☒	☐
Alarm0	Bool	false		☒	☒	☒	☐
dc motor current ma	Real	0.0		☒	☒	☒	☐

Figure 4.12 Analogue variables

Again, a DC motor's voltage is being controlled but there is no explanation of the practical setup

4.4.4 Motor Speed Control and Webserver Enable

Along with the Function Calls in the 'Main' Organisation Block, the Motor Speed Control is configured, using the PWM variable stored in the 'webdata' Data Block. The PWM variable will be adjustable via the ABA.

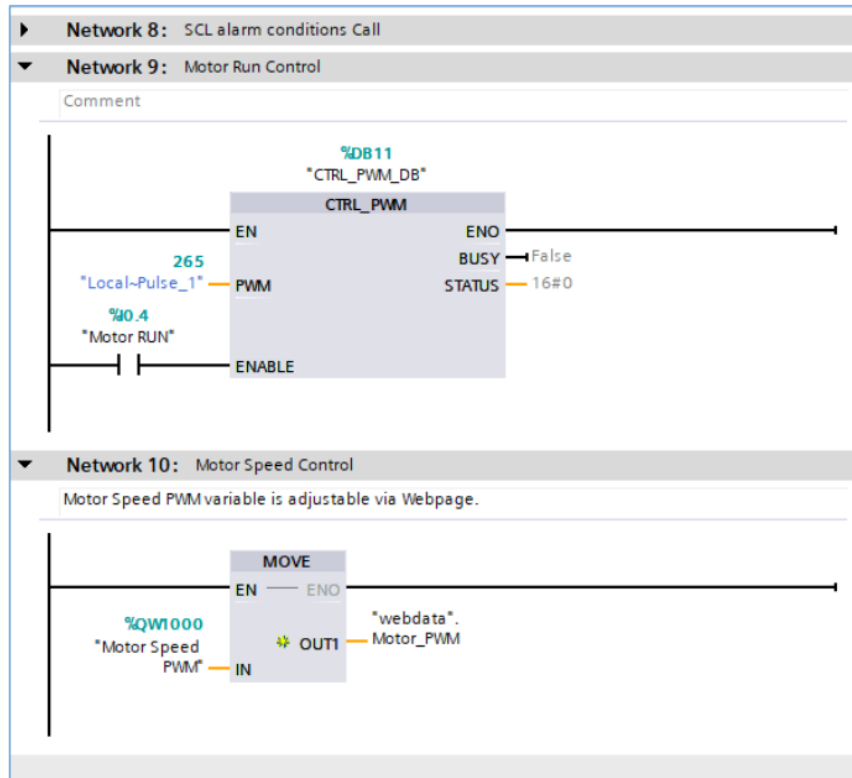


Figure 4.13 Motor Control in Main

The Webserver is also enabled in 'Main' as shown in Figure 4.14.

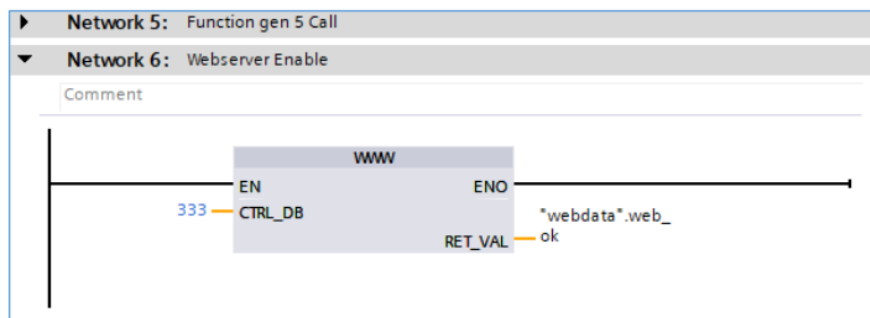


Figure 4.14 Webserver Enable

Default tag table [653]								
PLC tags								
PLC tags								
	Name	Data type	Address	Retain	Access- ible from HMI/O PC UA	Writa- ble from HMI/O PC UA	Visi- ble in HMI engi- neer- ing	Sup
	Tag1	Bool	%M100.1	False	True	True	True	
	Tag2	Bool	%M100.2	False	True	True	True	
	1HZ_OneShot2	Bool	%M100.3	False	True	True	True	
	Tag3	Bool	%M100.4	False	True	True	True	
	Tag4	Bool	%M100.5	False	True	True	True	
	1HZ_OneShot3	Bool	%M100.6	False	True	True	True	
	Tag5	Bool	%M100.7	False	True	True	True	
	Tag6	Bool	%M101.0	False	True	True	True	
	1HZ_OneShot4	Bool	%M101.1	False	True	True	True	
	Tag7	Bool	%M101.2	False	True	True	True	
	Tag8	Bool	%M101.3	False	True	True	True	
	op	Bool	%Q0.0	False	True	True	True	
	Clock_10Hz	Bool	%M0.0	False	True	True	True	
	Clock_5Hz	Bool	%M0.1	False	True	True	True	
	Clock_2.5Hz	Bool	%M0.2	False	True	True	True	
	Clock_2Hz	Bool	%M0.3	False	True	True	True	
	Clock_1.25Hz	Bool	%M0.4	False	True	True	True	
	Clock_1Hz	Bool	%M0.5	False	True	True	True	
	Clock_0.625Hz	Bool	%M0.6	False	True	True	True	
	Clock_0.5Hz	Bool	%M0.7	False	True	True	True	
	1HZ_OneShot1	Bool	%M100.0	False	True	True	True	
	op2	Bool	%Q0.1	False	True	True	True	
	op3	Bool	%Q0.2	False	True	True	True	
	1HZ_OneShot	Bool	%M101.4	False	True	True	True	
	Clock_Byte	Byte	%MB0	False	True	True	True	
	ip0	Bool	%I0.0	False	True	True	True	
	ip1	Bool	%I0.1	False	True	True	True	
	ip2	Bool	%I0.2	False	True	True	True	
	ip3	Bool	%I0.3	False	True	True	True	
	AI0	UInt	%IW64	False	True	True	True	
	AI1	UInt	%IW66	False	True	True	True	
	Motor Speed PWM	Word	%QW1000	False	True	True	True	
	Motor RUN	Bool	%I0.4	False	True	True	True	
	Motor PWM output	Bool	%Q1.1	False	True	True	True	

Figure 4.15 Default Tag Table

4.5 ABA

Following typical website design, pages are broken up into components where sensible and stored in appropriate folders. By following this rule, webpage resources are easier to locate and manage (Sherry Bishop, 2012).

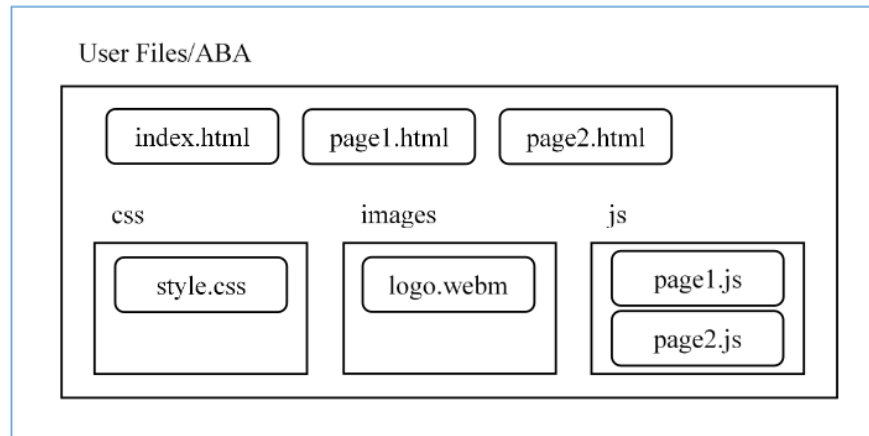


Figure 4.16 Typical Folder Hierarchy

To develop the ABA, the Atom text editor was chosen to build the ABA due to the packages that could be used to aid in development such as 'Emmet' which allows for streamlined coding, and 'atom-live-server' which allows for testing of server side script without needing to upload to a Webserver.

4.5.1 CSS Styling

To write an entire Cascading Stylesheet document takes time, so to ensure that a viable product was delivered in a timely manner, a template that was freely available from w3schools.com was used and modified according to preference. The styling was geared towards a simplified appearance that would not detract from the information presented on screen. This styling is also tied to a JavaScript file (sidebar.js), that produces a side menu for navigating a website.

```

.w3-brown,.w3-hover-brown:hover{color:#fff!important;background-color:#795548!important}
.w3-cyan,.w3-hover-cyan:hover{color:#000!important;background-color:#00bcd4!important}
.w3-blue-grey,.w3-hover-blue-grey:hover,.w3-blue-gray,.w3-hover-blue-gray:hover{color:#fff!important;background-color:#607dbb!important}
.w3-green,.w3-hover-green:hover{color:#fff!important;background-color:#4caf50!important}
.w3-light-green,.w3-hover-light-green:hover{color:#000!important;background-color:#8bc34a!important}
.w3-indigo,.w3-hover-indigo:hover{color:#fff!important;background-color:#3f51b5!important}
.w3-shade,.w3-hover-shade:hover{color:#000!important;background-color:#008080!important}
.w3-lime,.w3-hover-lime:hover{color:#000!important;background-color:#cddc39!important}
.w3-orange,.w3-hover-orange:hover{color:#000!important;background-color:#ff9800!important}
.w3-deep-orange,.w3-hover-deep-orange:hover{color:#fff!important;background-color:#ff5722!important}
.w3-pink,.w3-hover-pink:hover{color:#fff!important;background-color:#e91e63!important}
.w3-purple,.w3-hover-purple:hover{color:#fff!important;background-color:#9c27b0!important}
.w3-deep-purple,.w3-hover-deep-purple:hover{color:#fff!important;background-color:#673ab7!important}
.w3-red,.w3-hover-red:hover{color:#fff!important;background-color:#f44336!important}
.w3-sand,.w3-hover-sand:hover{color:#000!important;background-color:#fdf5e6!important}
.w3-teal,.w3-hover-teal:hover{color:#fff!important;background-color:#009688!important}
.w3-yellow,.w3-hover-yellow:hover{color:#000!important;background-color:#ffeb3b!important}
.w3-white,.w3-hover-white:hover{color:#000!important;background-color:#fff!important}
.w3-black,.w3-hover-black:hover{color:#fff!important;background-color:#000!important}
.w3-grey,.w3-hover-grey:hover,.w3-gray,.w3-hover-gray:hover{color:#000!important;background-color:#9e9e9e!important}
.w3-light-grey,.w3-hover-light-grey:hover,.w3-light-gray,.w3-hover-light-gray:hover{color:#000!important;background-color:#f1f3f3!important}
.w3-dark-grey,.w3-hover-dark-grey:hover,.w3-dark-gray,.w3-hover-dark-gray:hover{color:#fff!important;background-color:#616161!important}
.w3-pale-red,.w3-hover-pale-red:hover{color:#000!important;background-color:#ff8a65!important}
.w3-pale-green,.w3-hover-pale-green:hover{color:#000!important;background-color:#80ffa0!important}
.w3-pale-yellow,.w3-hover-pale-yellow:hover{color:#000!important;background-color:#ffffcc!important}
.w3-pale-blue,.w3-hover-pale-blue:hover{color:#000!important;background-color:#b3ffff!important}
.w3-text-amber,.w3-hover-text-amber:hover{color:#ff1717!important}
.w3-text-aqua,.w3-hover-text-aqua:hover{color:#0000ff!important}
.w3-text-blue,.w3-hover-text-blue:hover{color:#2196f3!important}

```

Figure 4.17 Colouring in w3.css

Additional CSS Styling is included in separate files for the Raleway font selected for the pages, and for the Chart.js JavaScript feature.

4.5.2 Index

The index page serves two purposes for the ABA, which is to function as the launch page for the ABA and to be used as a template to build other pages.

The file starts with an opening <html> tag and <head> tag. Between the head tags, the preliminary elements for building the webpage are placed.

In the <meta> tags, a rule is applied to set the width of the screen or ‘viewport’ to the ‘device-width’ which would vary depending on whether the user was viewing the page on a desktop browser or a browser in a mobile device. Additionally, the initial zoom level of the page is set to 1.

Two CSS files are linked to the page with additional styling placed between <style> tags to apply the Raleway font styling to the body text and headings.

The Body has a Grey colour applied, and between the Body tags the first <div> container of the page to be added is a Dark Grey top bar with a Menu Button on the left and a Coleg Sir Gâr logo on the right.

```

1  <!DOCTYPE html>
2
3  <html lang="en">
4    <head>
5      <!-- Required meta tags -->
6      <meta charset="utf-8">
7      <meta name="viewport" content="width=device-width, initial-scale=1,
8        shrink-to-fit=no">
9
10     <!-- Stylesheets -->
11     <link rel="stylesheet" type="text/css" href="css\w3.css">
12     <link rel="stylesheet" type="text/css" href="css\raleway.css">
13     <style>
14       html,body,h1,h2,h3,h4,h5 {font-family: "Raleway", sans-serif}
15     </style>
16
17     <title>CIM System - Automatic Browser Application</title>
18   </head>
19   <body class="w3-light-grey">
20
21     <!-- Top Container -->
22     <div class="w3-bar w3-top w3-dark-grey w3-large" style="z-index:4">
23       <button class="w3-bar-item w3-button w3-hide-large w3-hover-none
24         w3-hover-text-light-grey" onclick="w3_open();">
25         <i class="fa fa-bars"></i> &nbsp;Menu</button>
26       <span class="w3-bar-item w3-right">
27         </span>
29     </div>

```

Figure 4.18 index.html lines 1 to 29

The next section of code to be added is the <nav> or Sidebar. Within the Sidebar, a dark grey colouring is applied with links for other pages listed. Additionally, when the width of the screen is reduced, the Sidebar is collapsed to accommodate smaller screens, so to pull the list of pages back up, the Menu button on the Top bar must be clicked to bring up the Sidebar. To collapse the Sidebar again, the user clicks the Close button which only appears on a reduced screen size.

```

31 <!-- Sidebar/menu -->
32 <nav class="w3-sidebar w3-collapse w3-white w3-animate-left"
33 style="z-index: 3;width: 300px;" id="mySidebar">
34   <hr>
35   <div class="w3-container">
36     <h5>Stations</h5>
37   </div>
38   <div class="w3-bar-block">
39     <!-- Appears when window size reduced -->
40     <a href="#" class="w3-bar-item w3-button w3-padding-16 w3-hide-large
41 w3-dark-grey w3-hover-black"
42 onclick="w3_close()" title="close menu">
43 <i class="fa fa-remove fa-fw"></i>&nbsp;Close Menu</a>
44 <!-- Links (Active Page has blue styling) -->
45     <a href="plc101master.html"
46 class="w3-bar-item w3-button w3-padding">Master</a>
47     <a href="plc102pneumatic.html"
48 class="w3-bar-item w3-button w3-padding">Pneumatics</a>
49     <a href="plc103process.html"
50 class="w3-bar-item w3-button w3-padding">Process Control</a>
51     <a href="plc104hydraulic.html"
52 class="w3-bar-item w3-button w3-padding">Hydraulics</a>
53   </div>
54 </nav>

```

Figure 4.19 index.html lines 31 to 54

A similar rule is also applied for Mobile Devices or Small Screens.

```

56 <!-- Overlay for Sidebar on small screens -->
57 <div class="w3-overlay w3-hide-large w3-animate-opacity"
58 onclick="w3_close()" style="cursor:pointer" title="close side menu"
59 id="myOverlay"></div>

```

Figure 4.20 index.html lines 56 to 59

The opening and closing div tags shown from lines 62 to 79, is the space where unique content for the ABA pages is placed. In this instance, a header is placed with the text 'Welcome'. Underneath the header a div container is placed that takes up a quarter of the screen space with a blue background and white text. A message is placed in the container using a <p> tag, informing the user of the function of the Sidebar.

```
61 <!-- Page Content! -->
62 <div class="w3-main" style="margin-left: 300px;margin-top: 43px;">
63
64 <!-- Header -->
65 <header class="w3-container" style="padding-top:22px">
66 <h5><b>Welcome</b></h5>
67 </header>
68
69 <!-- System Warnings -->
70 <div class="w3-quarter">
71 <div class="w3-container w3-blue w3-text-white w3-padding-16">
72 <div class="w3-clear"></div>
73 <p>To monitor a Station, select one of the options from the menu
74 on the left. When using a smaller screen, click 'Menu' to show
75 list of Stations.
76 menu.</p>
77 </div>
78 </div>
79 </div>
```

Figure 4.21 index.html lines 61 to 79

At the end of the HTML file, links are added for jQuery and the Sidebar JavaScript code. The reason for adding the JavaScript files last, is so that code is only executed after all the design elements have had an opportunity to load. After the script linking, the body tag is closed out and then the html tag itself.

```
82 <!-- JavaScript files -->
83 <script src="js/jquery-3.5.1.min.js"></script>
84 <script src="js/sidebar.js"></script>
85 </body>
86 </html>
```

Figure 4.22 index.html lines 82 to 86

4.5.3 'Master' Page

At the top of the page, AWP Variables are added. AWP commands allow direct communication with the CPU of the PLC. To ensure reliable operation of the AWP commands, it is recommended that the file size of the HTML files do not exceed 64KB (Siemens, 2012), which is why the recommended file structure of the ABA shown in Figure 4.16 is important. The AWP Variables shown in Figure 4.23 are legacy elements from previous development via The CIM System. Although continuing development via the CIM System ceased, the Variables were retained for future potential development later.

```
1 <!-- AWP_In_Variable Name='\"PLC_102 PUT counter\".CV' -->
2 <!-- AWP_In_Variable Name='\"HMI CIM Start\"' -->
3 <!-- AWP_In_Variable Name='\"CIM Run\"' -->
4
5 <!DOCTYPE html>
6
7 <html lang="en">
8   <head>
```

Figure 4.23 plc101master.html lines 1 to 8

As the page is built on index.html, modifications are required. The Sidebar is modified to highlight the current page by altering the link's background colour to blue and disabling the link to prevent unnecessary reloading of the page.

```
36 <!-- Sidebar/menu -->
37 <nav class="w3-sidebar w3-collapse w3-white w3-animate-left"
38 style="z-index: 3;width: 300px;" id="mySidebar">
39   <hr>
40   <div class="w3-container">
41     <h5>Stations</h5>
42   </div>
43   <div class="w3-bar-block">
44     <!-- Appears when window size reduced -->
45     <a href="#" class="w3-bar-item w3-button w3-padding-16 w3-hide-large
46 w3-dark-grey w3-hover-black" onclick="w3_close()" title="close menu">
47     <i class="fa fa-remove fa-fw"></i>&nbsp; Close Menu</a>
48     <!-- Links (Active Page has blue styling) -->
49     <a href="#"
50 class="w3-bar-item w3-button w3-padding w3-blue">Master</a>
51     <a href="plc102pneumatic.html"
52 class="w3-bar-item w3-button w3-padding">Pneumatics</a>
53     <a href="plc103process.html"
54 class="w3-bar-item w3-button w3-padding">Process Control</a>
55     <a href="plc104hydraulic.html"
56 class="w3-bar-item w3-button w3-padding">Hydraulics</a>
57   </div>
58 </nav>
```

Figure 4.24 plc101master.html lines 36 to 58

The Master PLC page has an appropriate header applied, to indicate the current page loaded, with a heading size of <h5>.

Below the header, a container which takes up 25% percent of the available screen space is created with an orange background. The container serves as a placeholder for an alert box. It is intended that should any system faults be detected by the PLC; a variable could be added to the PLC program to record the number of faults. Where the value 50 is on line 78, the hypothetical 'Fault' PLC tag could be placed instead.

```
65 <!-- Page Content! -->
66 <div class="w3-main" style="margin-left: 300px;margin-top: 43px;">
67
68 <!-- Header -->
69 <header class="w3-container" style="padding-top:22px">
70 <h5><b>Master</b></h5>
71 </header>
72
73 <!-- System Warnings -->
74 <div class="w3-quarter">
75 <div class="w3-container w3-orange w3-text-white w3-padding-16">
76 <div class="w3-left"><i class="#"></i></div>
77 <div class="w3-right">
78 <h3>50</h3>
79 </div>
80 <div class="w3-clear"></div>
81 <h4>Alerts</h4>
82 </div>
83 </div>
```

Figure 4.25 plc101master.html lines 65 to 83

In the Variables container shown in Figure 4.2, two methods of showing a PLC tag on page are included. On line 90, The PLC tag is added in plain text, which would be replaced on-screen by the value held in that variable on the PLC. On line 93, the table cell has an id applied to it with a placeholder value of 2. The id on line 93 is used in plc101master.js to alter the placeholder value of '2' to reflect the value stored in the PLC tag.

```
85 <!-- System Information -->
86 <div class="w3-container">
87 <h5>Variables</h5>
88 <table class="w3-table w3-striped w3-white">
89 <tr>
90 <td>:=HMI CIM Start:</td>
91 </tr>
92 <tr>
93 <td id="cim_start">2</td>
94 </tr>
95 </table>
96 </div>
```

Figure 4.26 plc101master.html lines 85 to 96

The div container in Figure 4.27 is a panel which takes up the entire screen width. Two components are featured within this panel, which are a Chart and Controls. The components are positioned using two div containers, where the Chart takes up two thirds of the panel, and the controls are set to take up a third. The Chart is placed using a <canvas> tag with appropriate id. The controls are two buttons that push either a 1 or a 0 to the 'HMI CIM Start' PLC tag.

```
98      <!-- Chart and Controls -->
99      <div class="w3-panel">
100        <div class="w3-row-padding" style="margin: 0 -16px">
101          <div class="w3-twothird">
102            <h5>Chart</h5>
103            <canvas id="triChart"></canvas>
104          </div>
105          <div class="w3-third">
106            <h5>Controls</h5>
107            <p>:= "CIM Run":</p>
108            <form>
109              <input type="submit" value="CIM Start">
110              <input type="hidden" name="HMI CIM Start" value="1">
111            </form>
112            <form>
113              <input type="submit" value="CIM Stop">
114              <input type="hidden" name="HMI CIM Start" value="0">
115            </form>
116          </div>
117        </div>
118      </div>
119    </div>
```

Figure 4.27 plc101master.html lines 98 to 119

At the end of the html file, links to three additional JavaScript files are included which are Chart.bundle.min.js, plc101master.js and plc101chart.js.

```
122    <!-- JavaScript files -->
123    <script src="js/jquery-3.5.1.min.js"></script>
124    <script src="js/Chart.bundle.min.js"></script>
125    <script src="js/sidebar.js"></script>
126    <script src="js/plc101master.js"></script>
127    <script src="js/plc101chart.js"></script>
128  </body>
129 </html>
```

Figure 4.28 plc101master.html lines 122 to 129

The JavaScript file plc101master.js is written to read the contents of a json file called CIMcond every 200 milliseconds which contains a PLC tag stored as a json variable. The read json variable is then pushed to line 93 in plc101master.html

```
1 $(document).ready(function(){
2   setInterval(function(){
3     $.getJSON("CIMcond.json", function(data)
4     {
5       $('#cim_start').text(data.start_state);
6     });
7   },200);
8 });
```

Figure 4.29 plc101master.js

```
1 {"start_state" : "HMI CIM Start:"}
```

Figure 4.30 CIMcond.json

The script for plc101chart.js starts with the instruction to only execute once the page has loaded completely. On line 5, the updateInterval is set so that the Chart updates every 1 second. On line 6, the number of values on the x-axis is 20. Line 12 applies the contents of the variable triChart to a html canvas asset with the id of #triChart.

```
1 window.onload = function(){
2   console.log("LOADED");
3
4   //Configuration variables
5   var updateInterval = 1000;
6   var numberElements = 20;
7
8   //Globals
9   var updateCount = 0;
10
11  //Chart Objects
12  var triChart = $("#triChart");
```

Figure 4.31 plc101chart.js lines 1 to 13

Lines 15 to 35 of Figure 4.32 show the configuration of the Chart asset presented on plc101master.html, where the time is formatted as shown on line 21, the y-axis starts at 0, and the legend is disabled.

```
14 //Chart instances & config
15 var commonOptions = {
16   scales: {
17     xAxes: [{
18       type: 'time',
19       time: {
20         displayFormats: {
21           millisecond: 'mm:ss:SSS'
22         }
23       }
24     }],
25     yAxes: [{
26       ticks: {
27         beginAtZero: true
28       }
29     }]
30   },
31   legend: {display: false},
32   tooltips: {
33     enabled: false
34   }
35 };
```

Figure 4.32 plc101Chart.js lines 14 to 35

Further configuration is shown in Figure 4.33 which specify that a line chart is used, with additional font styling exclusive to the chart specified.

```
36     var triChartInstance = new Chart(triChart, {
37         type: 'line',
38         data: {
39             datasets: [{
40                 label: "Triangular Wave",
41                 data: 0,
42                 fill: false,
43                 borderColor: '#343e9a',
44                 borderWidth: 1
45             }]
46         },
47         options: Object.assign({}, commonOptions, {
48             title:{
49                 display: true,
50                 text: "Triangle Wave",
51                 fontSize: 18
52             }
53         })
54     });
```

Figure 4.33 plc101chart.js lines 36 to 54

The script shown in Figure 4.34 handles the addition of the current PLC tag value and increments the previous value by shifting.

```
56     function addData(data) {
57         if(data){
58             triChartInstance.data.labels.push(new Date());
59             triChartInstance.data.datasets.forEach((dataset)
60             =>{dataset.data.push(data['tri'])});
61             if(updateCount > numberElements){
62                 triChartInstance.data.labels.shift();
63                 triChartInstance.data.datasets[0].data.shift();
64             }
65             else updateCount++;
66             triChartInstance.update();
67         }
68     };
```

Figure 4.34 plc101chart.js lines 56 to 68

Lines 70 to 76 of plc101chart.js are used to pull the current PLC tag value stored in plc101tri.json to add to the scrolling line chart.

```
70     function updateData(){
71         console.log("Update Data");
72         $.getJSON("plc101tri.json", addData);
73         setTimeout(updateData,updateInterval);
74     }
75     updateData();
76 }
```

Figure 4.35 plc101chart.js lines 70 to 76

```
1  {"tri" : "!="webdata".triangleWave:"}
```

Figure 4.36 plc101tri.json

4.5.4 'Pneumatic' Page

As the pages are stored in a Webserver (PLC), advantage can be taken of server-side 'includes', where entire webpages are added to the current page by adding a line as shown in Figure 4.37

```
1 <!--#include file="IOWrite.html"-->
```

Figure 4.37 plc102pneumatic.html line 1

To change the state of a variable on the PLC, the variables need to be formatted as shown in Figure 4.38. if the variables are used by more than one page, it is more effective to have a repository of variables to call from.

```
1 <!-- Variable declaration: necessary only for the variables to be written -->
2 <!-- AwP_In_Variable Name="webdata".counter' -->
3 <!-- AwP_In_Variable Name="webdata".counter1' -->
4 <!-- AwP_In_Variable Name="webdata".counter2' -->
5 <!-- AwP_In_Variable Name="webdata".counter3' -->
6 <!-- AwP_In_Variable Name="Tag1" -->
7 <!-- AwP_In_Variable Name="Tag3" -->
8 <!-- AwP_In_Variable Name="Tag5" -->
9 <!-- AwP_In_Variable Name="Tag7" -->
10 <!-- AwP_In_Variable Name="ip0" -->
11 <!-- AwP_In_Variable Name="ip1" -->
12 <!-- AwP_In_Variable Name="ip2" -->
13 <!-- AwP_In_Variable Name="ip3" -->
14 <!-- AwP_In_Variable Name="HMI CIM Start" -->
15 <!-- AwP_In_Variable Name="CIM Run" -->
16 <!-- AwP_In_Variable Name="Motor Speed PWM" -->:= "Motor Speed PWM":
```

Figure 4.38 IOWrite.html

As with the 'Master' page, the links in the Sidebar are modified to disable the link to the current page and change the background colour to highlight the current page.

```
52 <a href="plc101master.html"
53 class="w3-bar-item w3-button w3-padding">Master</a>
54 <a href="#"
55 class="w3-bar-item w3-button w3-padding w3-blue">Pneumatics</a>
56 <a href="plc103process.html"
57 class="w3-bar-item w3-button w3-padding">Process Control</a>
```

Figure 4.39 plc102pneumatic.html lines 52 to 57

A div container for counters is added to the webpage. Lines 81 and 86 are table headings.

```

76      <!-- Counters -->
77      <div class="w3-container">
78          <h5>Counters</h5>
79          <table class="w3-table w3-striped w3-white">
80              <tr>
81                  <th><h3>Counter1:</h3></th>
82                  <th><h3>Counter2:</h3></th>
83                  <th><h3>Counter3:</h3></th>
84                  <th><h3>Counter4:</h3></th>
85                  <th>&nbsp;</th>
86                  <th><h3>Counter states:</h3></th>
87              </tr>

```

Figure 4.40 plc102pneumatic.html lines 76 to 87

Lined up underneath the previous table headings, plc tags are listed across the table, with ids linked to facilitate updating via scripting. Additional cells are added for 'LED' lamps which are added through scripting.

```

88          <tr>
89              <td>&nbsp;</td>
90              <td><label id="counter">:="webdata".counter:</label></td>
91              <td>&nbsp;</td>
92              <td>&nbsp;</td>
93              <td><label id="counter1">:="webdata".counter1:</label></td>
94              <td>&nbsp;</td>
95              <td>&nbsp;</td>
96              <td>&nbsp;</td>
97              <td><label id="counter2">:="webdata".counter2:</label></td>
98              <td>&nbsp;</td>
99              <td>&nbsp;</td>
100             <td>&nbsp;</td>
101             <td><label id="counter3">:="webdata".counter3:</label></td>
102             <td>&nbsp;</td>
103             <td>&nbsp;</td>
104             <td>&nbsp;</td>
105             <td><label id="Tag1"></label></td>
106             <td><label id="Tag3"></label></td>
107             <td><label id="Tag5"></label></td>
108             <td><label id="Tag7"></label></td>
109             <td><label id="img"></label></td>
110             <td><label id="img2"></label></td>
111             <td><label id="img3"></label></td>
112             <td><label id="img4"></label></td>
113         </tr>
114     </table>
115 </div>

```

Figure 4.41 plc102pneumatic.html lines 88 to 115

In Figure 4.42, another table is made for displaying the binary states of the input switches attached to the PLC with on-screen LEDs that change depending on the associated input state.

```

117 <!-- Sensor Inputs -->
118 <div class="w3-container">
119   <h5>Sensor Inputs</h5>
120   <table class="w3-table w3-striped w3-white">
121     <tr>
122       <td></td>
123       <th><h3>Input0 state:</h3></th>
124       <td><label id="ip0"></label></td>
125       <td><label id="img5"></label></td>
126     </tr>
127     <tr>
128       <td></td>
129       <th><h3>Input1 state:</h3></th>
130       <td><label id="ip1"></label></td>
131       <td><label id="img6"></label></td>
132     </tr>
133     <tr>
134       <td></td>
135       <th><h3>Input2 state:</h3></th>
136       <td><label id="ip2"></label></td>
137       <td><label id="img7"></label></td>
138     </tr>
139     <tr>
140       <td></td>
141       <th><h3>Input3 state:</h3></th>
142       <td><label id="ip3"></label></td>
143       <td><label id="img8"></label></td>
144     </tr>
145   </table>
146 </div>
147

```

Figure 4.42 plc102pneumatic.html lines 117 to 147

In a div container set to take up 66% of screen width, a gauge is added as identified by the id 'g1'.

```

149 <!-- Gauge -->
150 <div class="w3-panel">
151   <div class="w3-row-padding" style="margin: 0 -16px">
152     <div class="w3-twothird">
153       <table>
154         <tr>
155           <td><div id="g1"></div></td>
156         </tr>
157       </table>
158     </div>
159   </div>
160 </div>
161

```

Figure 4.43 plc102pneumatic.html lines 149 to 161

Three additional scripts are added to the page being, raphael.2.1.0.min.js which is used for gauge animations, justgage.1.0.1.min.js which is a development of the Raphael JavaScript library, and plc102pneumatic.js for page exclusive scripting.

```
164     <!-- JavaScript files -->
165     <script src="js/jquery-3.5.1.min.js"></script>
166     <script src="js/sidebar.js"></script>
167     <script src="js/raphael.2.1.0.min.js"></script>
168     <script src="js/justgage.1.0.1.min.js"></script>
169     <script src="js/plc102pneumatic.js"></script>
170 </body>
171 </html>
```

Figure 4.44 plc102pneumatic.html lines 164 to 171

In the script, the json array IOCounter is polled for values which are stored in the id attributes in lines 5 to 16 and would keep polling every 700ms.

```
1  $(document).ready(function(){
2    $.ajaxSetup({ cache: false });
3    setInterval(function(){
4      $.getJSON("IOCounter.json", function(result){
5        $('#counter').text(result["counter"]);
6        $('#counter1').text(result["counter1"]);
7        $('#counter2').text(result["counter2"]);
8        $('#counter3').text(result["counter3"]);
9        $('#Tag1').text(result["Tag1"]);
10       $('#Tag3').text(result["Tag3"]);
11       $('#Tag5').text(result["Tag5"]);
12       $('#Tag7').text(result["Tag7"]);
13       $('#ip0').text(result["ip0"]);
14       $('#ip1').text(result["ip1"]);
15       $('#ip2').text(result["ip2"]);
16       $('#ip3').text(result["ip3"]);
```

Figure 4.45 plc102pneumatic.js lines 1 to 16

Conditional statements are written for the id attributes shown in Figure 4.45, which tie the Boolean values of the associated PLC tags to the LED icons.

```

76     if(result["ip3"]==0){
77         $('#img8').html('');
78     }
79     else
80     {
81         $('#img8').html('');
82     }
83     });
84 },
85 //Refresh rate in ms.
86 700);

```

Figure 4.46 plc102pneumatic.js lines 76 to 86

The gauge 'g1' is initialised with a random integer value, which would then be replaced with the current value for the potentiometer reading in the PLC as found in the file 'IOresult2.html'.

```

88 //Initialise the gauge dynamic variables. Only g1 is used here, g2, g3 and g4,
89 are leftovers from previous implementation.
90 var g1, g2, g3, g4;
91
92 var g1 = new JustGage({
93     id: "g1",
94     value: getRandomInt(0, 100),
95     min: 0,
96     max: 100,
97     title: "Position",
98     label: " Analogue Value"
99 });
100
101
102 //call the ajax function. This can easily be converted to JSON if necessary
103 $.ajaxSetup({ cache: false });
104 setInterval(function(){
105     $.get("IOresult2.html", function(result){
106         g1.refresh(parseInt(result));
107     });
108 },700);
109 });
110
111 document.ontouchmove = function(e) {e.preventDefault()};

```

Figure 4.47 plc102pneumatic.js lines 88 to 111

```

1 <!-- AWP_In_Variable Name="webdata".counter' -->
2 := "Analogue".pot_volts_percent:

```

Figure 4.48 IOresult2.html

The LED icon assets used are only 3 by 2 pixels in size at 0.1KB. When used, they are magnified in html code.



Figure 4.49 led0.png (green) and led1.png (red)

4.5.5 'Process' Page

As before, the current page is highlighted in the Sidebar.

```
53     class="w3-bar-item w3-button w3-padding">Pneumatics</a>
54     <a href="#"
55     class="w3-bar-item w3-button w3-padding w3-blue">Process Control</a>
56     <a href="plc104hydraulic.html"
```

Figure 4.50 plc103process.html lines 53 to 56

In Figure 4.51, an id attribute for Motor_PWM is placed along with a text box and button for setting a new PWM value.

```
74     <!-- Controls -->
75     <div class="w3-quarter">
76         <h3>Controls</h3>
77         <div class="w3-container w3-orange w3-text-black w3-padding-16">
78             <table>
79                 <tr>
80                     <th><h5>Motor Duty Cycle:</h5></th>
81                     <td><h5><label id="Motor_PWM">0</label></h5></td>
82                     <td><h5>%</h5></td>
83                 </tr>
84                 <tr>
85                     <td>
86                         <input id='setvar' type="text" style="width:150px;height:30px;
87                         background-color:w3-white;color:black;"/>
88                         <button style="background-color:w3-grey;color:black;
89                         width:150px;height:40px;">Set PWM</button>
90                     </td>
91                 </tr>
92             </table>
93         </div>
94     </div>
```

Figure 4.51 plc102process.html lines 74 to 95

In Figure 4.52, a gauge is added, so show the current Motor Speed.

```
97     <!-- Gauge -->
98     <div class="w3-panel">
99         <div class="w3-row-padding" style="margin: 0 -16px">
100             <div class="w3-twothird">
101                 <table>
102                     <tr>
103                         <td><div id="g1"></div></td>
104                     </tr>
105                 </table>
106             </div>
107             <div class="w3-third">
108                 <h5>Status</h5>
109                 <label id="Motor RUN"></label>
110                 <label id="img9"></label>
111             </div>
112         </div>
113     </div>
114 </div>
```

Figure 4.52 plc103process lines 97 to 114

Like the 'Pneumatic' Page, the same JavaScript files are used except for plc103process.js.

```
117     <!-- JavaScript files -->
118     <script src="js/jquery-3.5.1.min.js"></script>
119     <script src="js/sidebar.js"></script>
120     <script src="js/raphael.2.1.0.min.js"></script>
121     <script src="js/justgage.1.0.1.min.js"></script>
122     <script src="js/plc103process.js"></script>
123 </body>
124 </html>
```

Figure 4.53 plc103process.html lines 117 to 124

In the script shown in Figure 4.54, every 1 second, the IOCounter json file is read for two variables which are, Motor_PWM and Motor RUN. There is also a conditional statement stipulating which LED icon is applied to the html asset called #img9, depending on the state of Motor RUN.

```
1  $(document).ready(function(){
2    $.ajaxSetup({cache:false});
3    setInterval(function(){
4      $.getJSON("IOCounter.json",function(result){
5        $('#Motor_PWM').text(result["Motor_PWM"]);
6        $('#Motor RUN').text(result["Motor RUN"]);
7
8        if(result["Motor RUN"]==0){
9          $('#img9').html('');
10       }
11       else{
12         $('#img9').html('');
13       }
14     });
15   },1000);
```

Figure 4.54 plc103process.js lines 1 to 15

The gauge on the 'Process' Page is initialised with a random integer and then refreshed with value held in the PLC tag stored in IOresult.html, every 1 second. Script shown in lines 33 to 40 of Figure 4.55 handle the updating of the Motor Speed PWM tag in IOwrite.html using a setvar method.

```

17     var g1, g2, g3, g4;
18
19     var g1 = new JustGage({
20         id: "g1",
21         value: getRandomInt(0,100),
22         min: 0,
23         max: 100,
24         title: "Duty Cycle %",
25         label: "Motor Power"
26     });
27     $.ajaxSetup({cache:false});
28     setInterval(function(){
29         $.get("IOresult1.html", function(result){
30             g1.refresh(parseInt(result));
31         });
32     },1000);
33     $("button").click(function(){
34         url="IOwrite.html";
35         name='"Motor Speed PWM"';
36         val=$('#input[id=setvar]').val();
37         sdata=escape(name)+'='+val;
38         $.post(url,sdata,function(result){});
39         document.getElementById('setvar').value='';
40     });
41 });

```

Figure 4.55 plc103process.js lines 17 to 41

```

1  <!-- AWP_In_Variable Name=' "webdata".Motor_PWM' -->:"webdata".Motor_PWM:

```

Figure 4.56 IOresult1.html

5 IMPLEMENTATION

5.1 DEVICE CONNECTION

*The IP addresses of the all
CIM stations were
10.13.90.101 onwards*

On initial connect to the router, the PLC requests the IP address of 192.168.0.190. Provided that the address isn't already taken by another device, the connection should establish. With the connection made, the PLC address is manually added to the list of reserved addresses, to ensure that no other device is allocated the same IP address while the PLC is powered down. Reservation is achieved by tying the IP address to the PLC's MAC address.

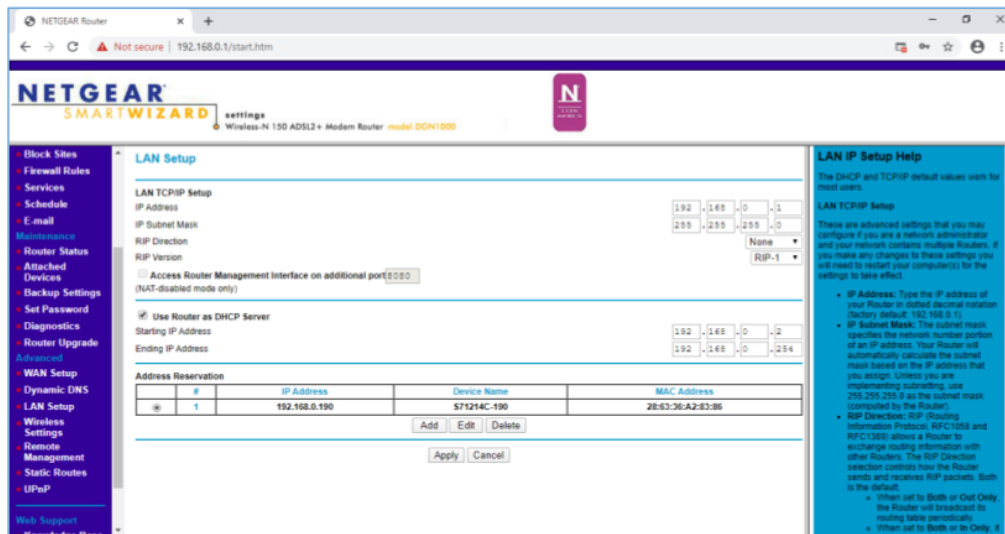


Figure 5.1 DHCP enabled and IP address reserved

As access to hardware and software such as network switches and dedicated server management software required for more advanced network management is made difficult due to the emerging global issues, implementation of User Access policies have not been possible for this implementation at this time (Coleg Sir Gâr, 2020).

5.2 MEMORY CARD

The S7-1200 memory card can be used in 4 different ways depending on firmware versions used on the PLC. The memory card can be used to update PLC firmware in an offline state (firmware mode), it can be used as a shuttle for a new version of the PLC Program if the PLC is not networked (transfer mode), and it can also be used as default program storage on a PLC (program mode) (Siemens, 2014).

To accommodate for the large cumulative size of various JavaScript libraries, the use of the memory card as expanded program storage, avoids detrimental performance in the PLC.

By default, the memory card is configured as a 'program' card, but to ensure reliable operation, the following steps are taken.

1. Delete card contents, which will be replaced by new files by the download later.

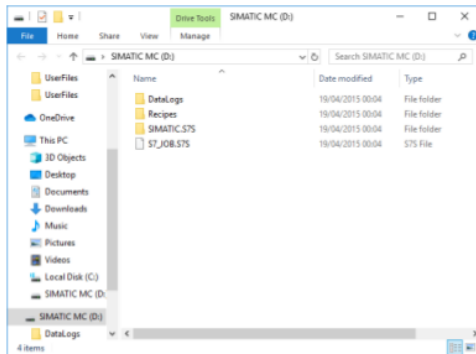


Figure 5.2 Before Deletion

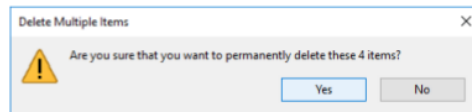


Figure 5.3 Deletion Prompt

2. Change Start-up to "Warm Restart - RUN".

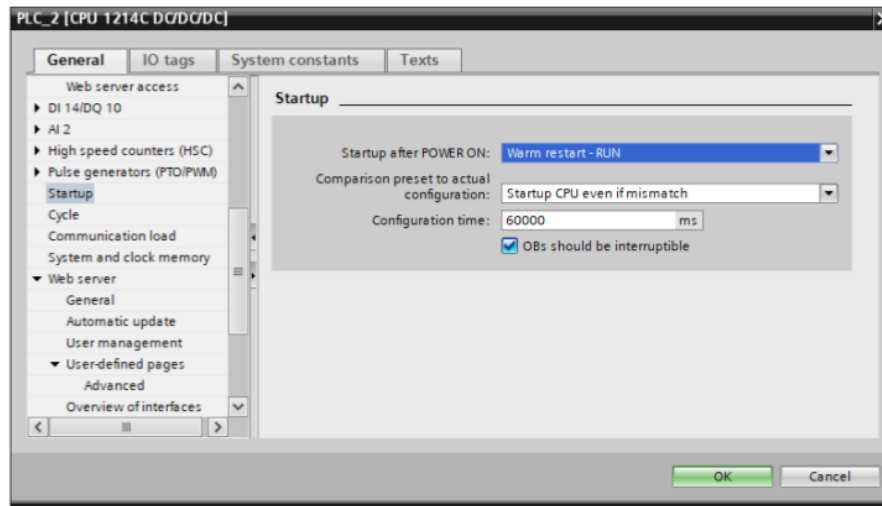


Figure 5.4 Warm Restart

3. Connect Memory Card to TIA Portal and ensure card is set to Program mode

Your report is written as an instruction manual, you are telling your reader what to do. Rather, you should be explaining what has been done to achieve the project outcome.

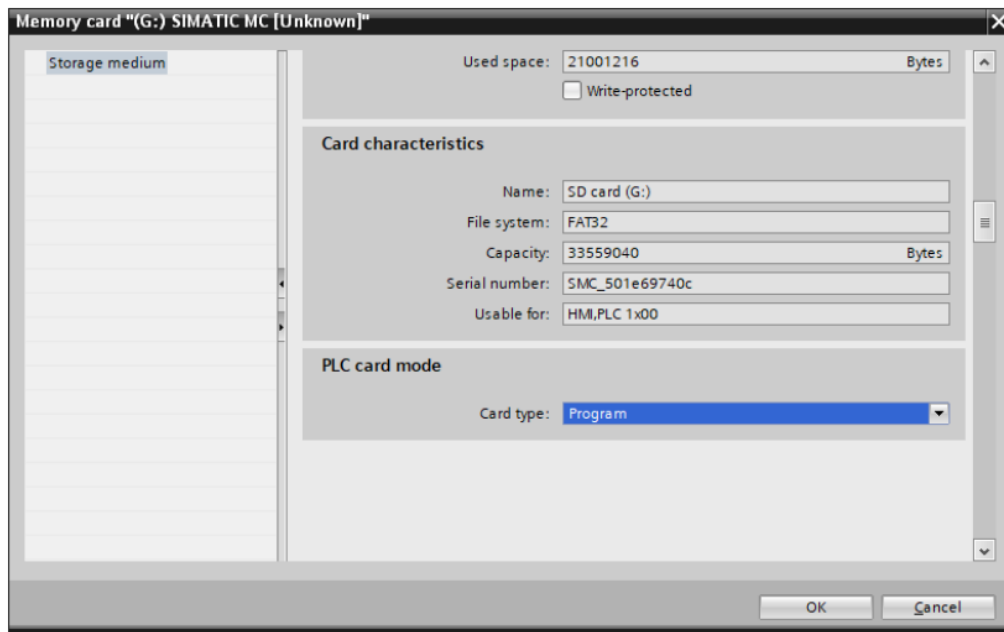


Figure 5.5 Set to Program Mode

4. Power down PLC.
5. Insert memory card into card slot.
6. Power up PLC.
7. Download Project files to PLC.

5.3 ADDING FINAL ABA TO PLC PROGRAM

The finalised ABA files are added to the PLC Program by pointing to the source directory in the PLC Project files, which is usually 'User Files'. The Default HTML page, which is the first to open is also specified, which should be index.html. Once both are selected, the blocks are generated.

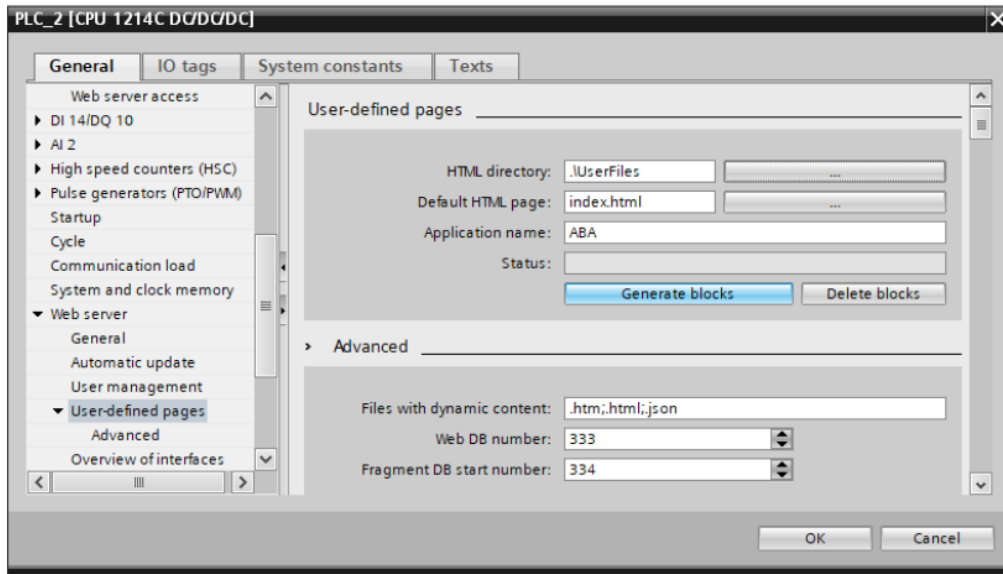


Figure 5.6 Generate Blocks

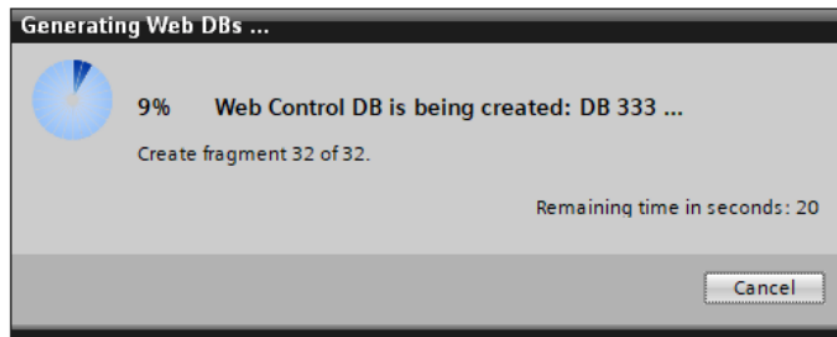


Figure 5.7 Block Generation

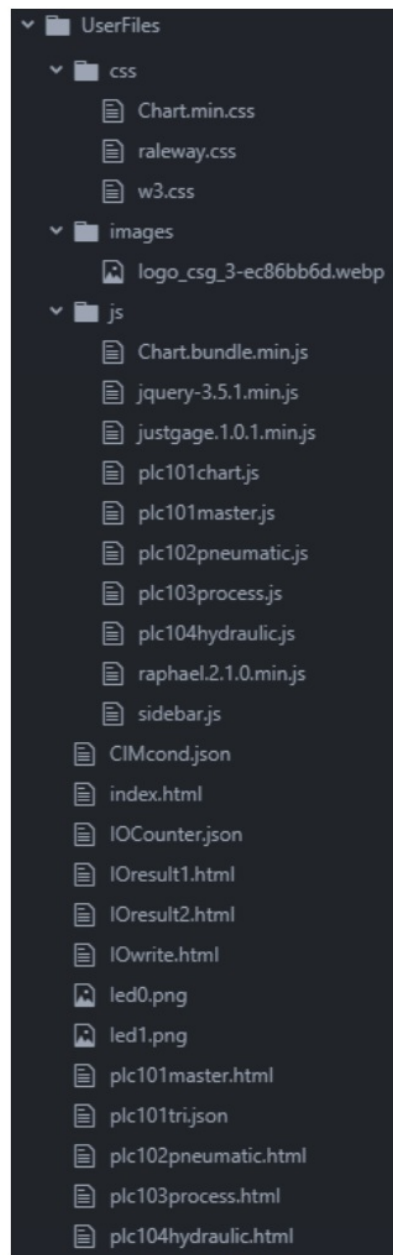


Figure 5.8 ABA Files

5.4 DOWNLOADING PLC PROGRAM

Before downloading to the PLC, a connection is made to ensure a download can proceed. As is shown in Figure 5.9, the PLC was ready for a download.

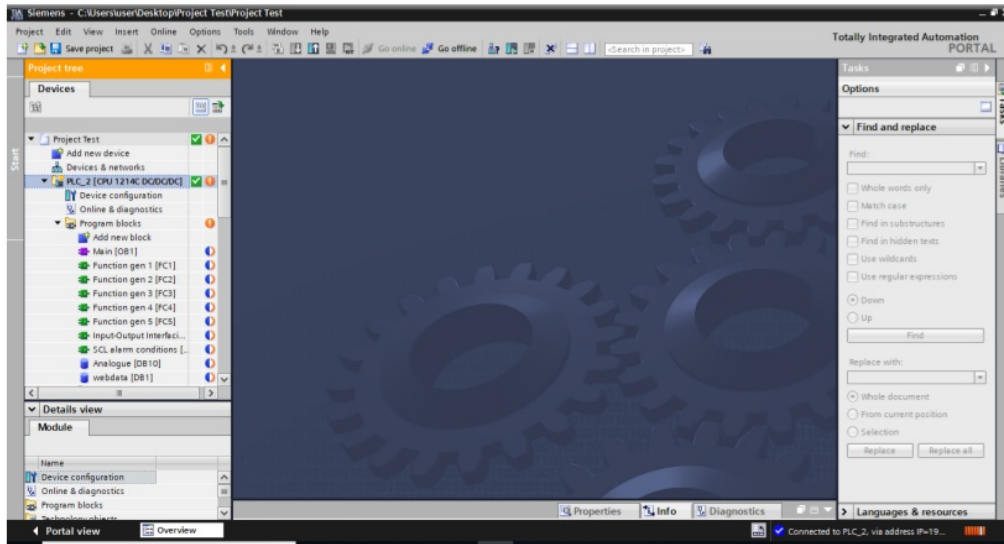


Figure 5.9 PLC in blank state due to empty memory card

No major issues surfaced during upload.

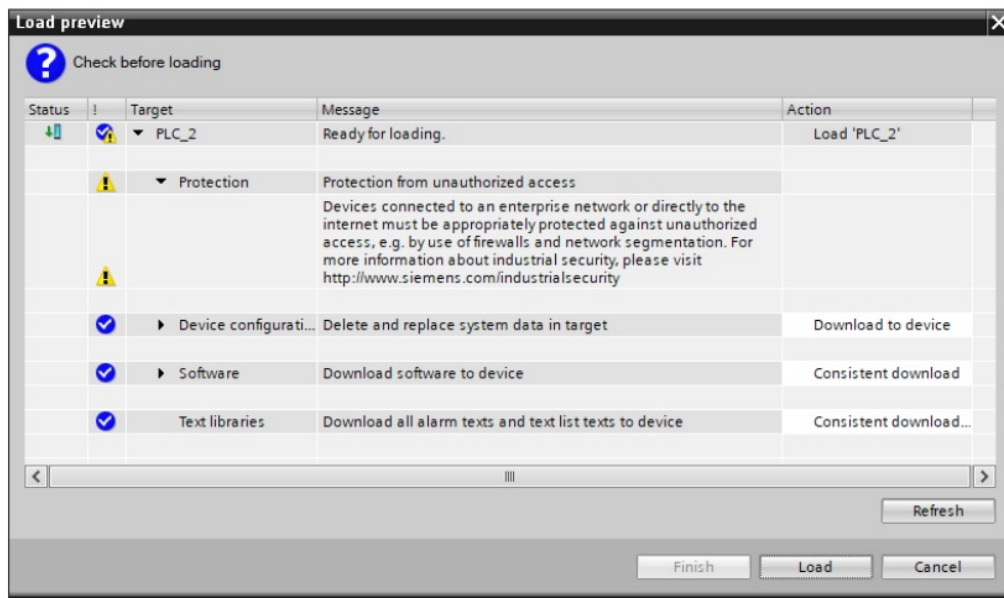


Figure 5.10 PLC ready for download

A successful download was achieved as shown in Figure 5.11.

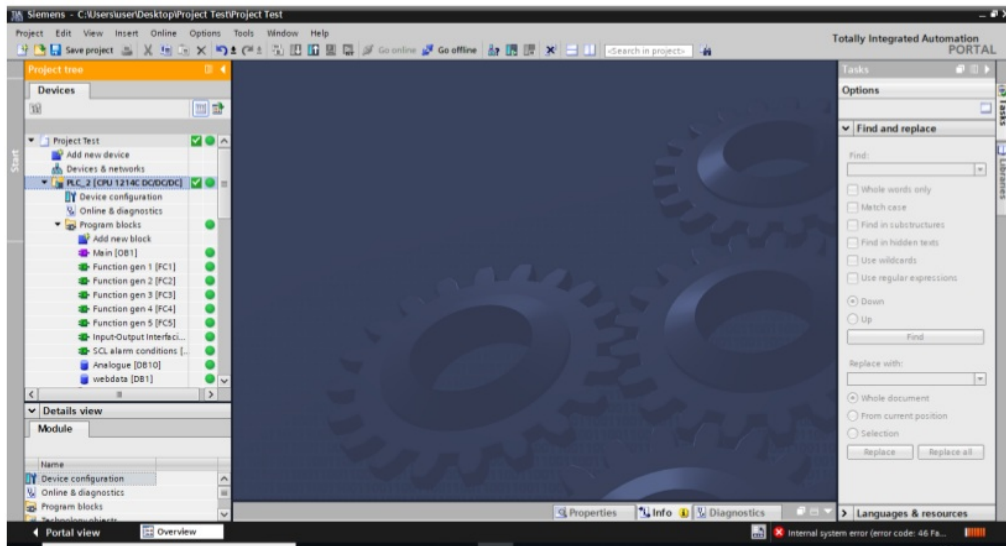


Figure 5.11 Program downloaded

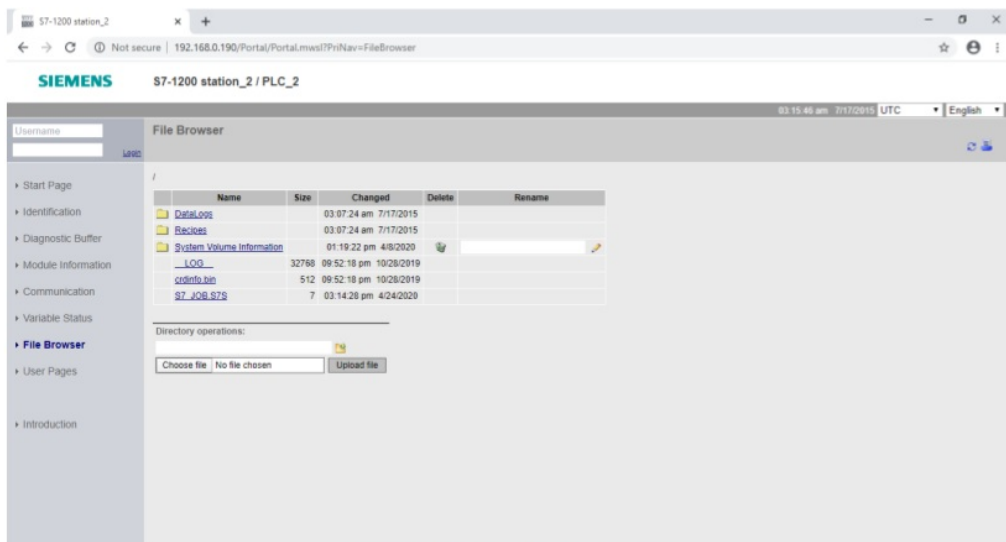


Figure 5.12 Memory Card Programmed

6 TESTING AND RESULTS

6.1 TEST

Test No.:001 Date of Test:24/04/2020				
Test Step	Test Description	Target	Actual	Pass/Fail and Comment
1	Does the ABA start? Tested by opening Address/Shortcut to ABA.	ABA Opens	Figure 6.1	Pass – Index.html opens.
2	Measure response time between the PLC and User Input and Feedback to ABA	= or < 1 second	Figure 4.54	Pass – Features refresh at time specified in code i.e. 1000ms.
3	There should be at least 1 page per CIM System Station. Are all Stations represented in the ABA?	1 Page minimum per Station	Figure 6.2, Figure 6.3, Figure 6.5	Pass – Station Pages created, but content does not necessarily reflect CIM System. Hydraulics Page was duplication of Process Control Page
4	Readings displayed are accurate to values read and stored in the PLC. Achieved by comparing ABA to online connection to PLC through Step7.	Values are within 1%	Figure 6.2, Figure 6.3	Pass – Readings are reflective of PLC input states.

5	Do the Security Measures perform as expected? Security measure rejects login attempts after failure.	Login rejected	N/A	Fail – Feature not implemented.
6	Does the ABA at any point lose connection to the CIM System? Measured by desynchronization of values between PLC and ABA.	Continuous Connection	Figure 6.2	Pass – No connection lost at any time.
7	When Operating are there any failures to proceed. (CIM System stops without warning)	No unexpected stoppages	N/A	Pass – No unexpected stoppages during testing
8	When ABA is running, are there any failures to Start or Stop the CIM System?	Starts and Stops on command	N/A	N/A – Test could not be conducted via the CIM System as no access possible

6.2 RESULTS

The following are screengrabs taken from the CIM System running.

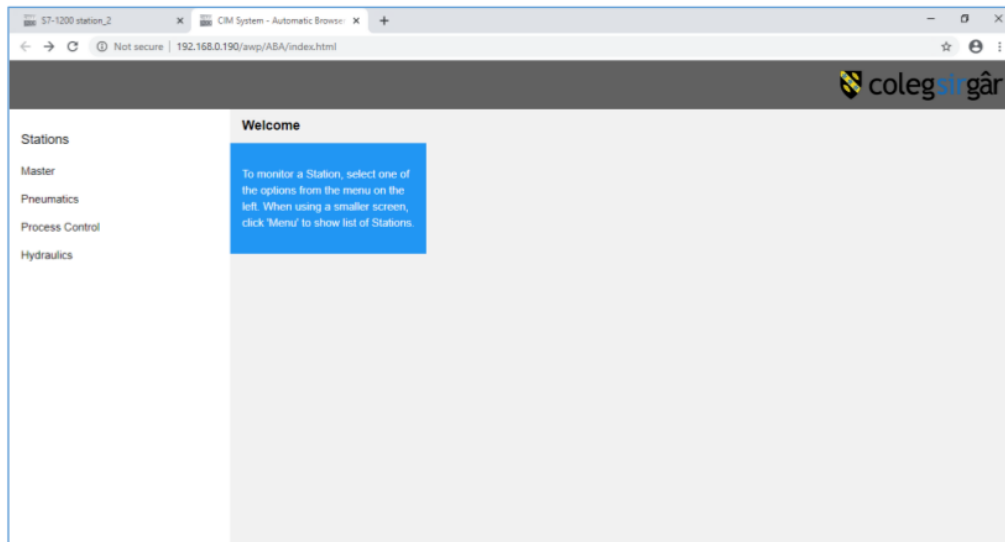


Figure 6.1 index.html

The Chart feature on the Master page would fill to 20 values and shift along to add updated values.

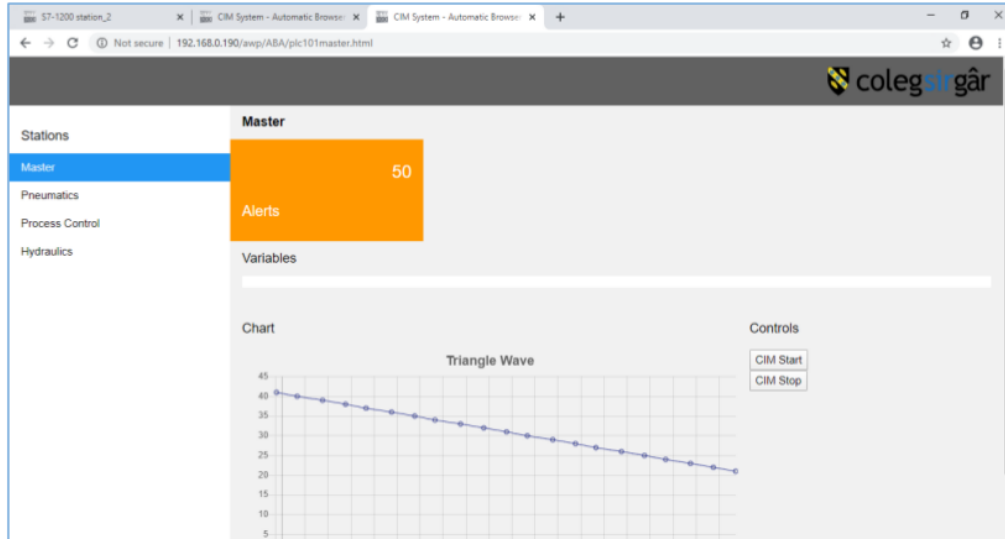


Figure 6.2 Master Page

Some minor issues occurred in lining up the counters with headings, however the inputs and tied LED icons functioned well.

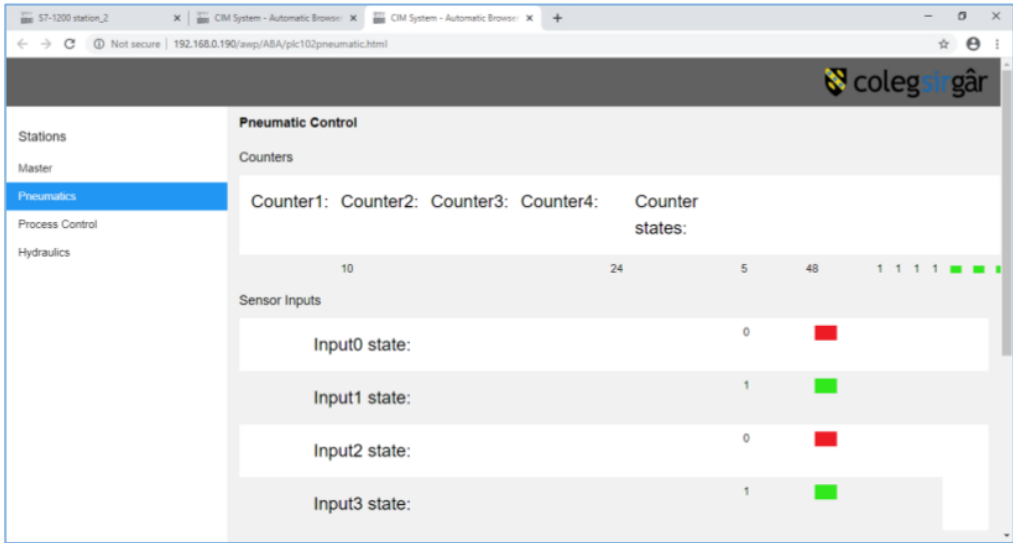


Figure 6.3 Pneumatics Page

Name	Address	Display format	Monitor value	M
Analogue.pot_v...		Floating-point nu...	27.11389	
ip0	%I0.0	Bool	FALSE	
ip1	%I0.1	Bool	TRUE	
ip2	%I0.2	Bool	FALSE	
ip3	%I0.3	Bool	TRUE	
<Add new>				

Figure 6.4 Corresponding IO in Watch Table

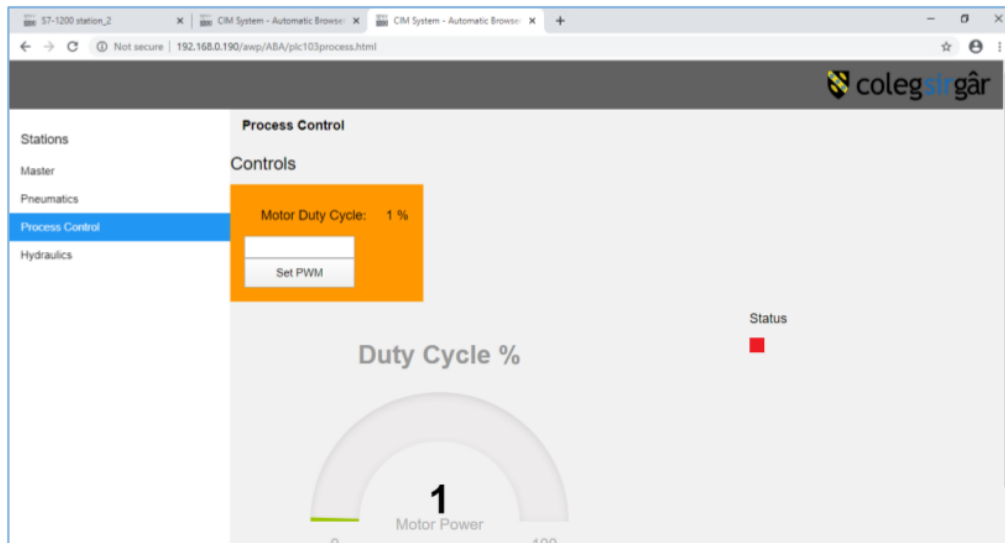


Figure 6.5 Process Control Page

A difficulty arose in testing the Duty Cycle for a DC motor, as a Motor could not be sourced in a reasonable amount of time. A substitution was attempted using LEDs; however, this was unsuccessful, possibly due to insufficient protection for LED devices.

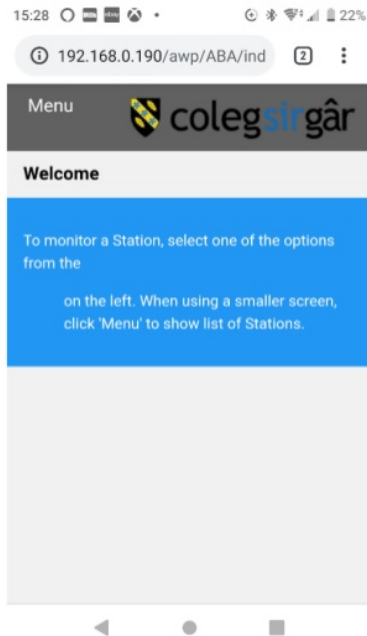


Figure 6.6 index.html on Mobile Device

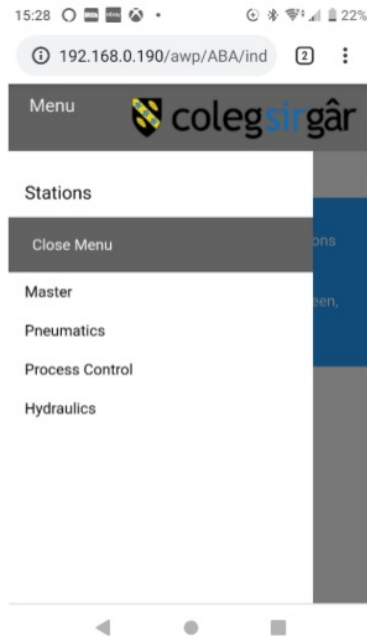


Figure 6.7 index.html with Menu open

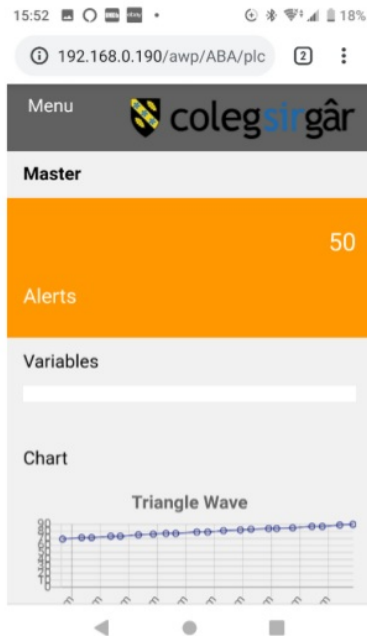


Figure 6.8 Master Page on Mobile Device

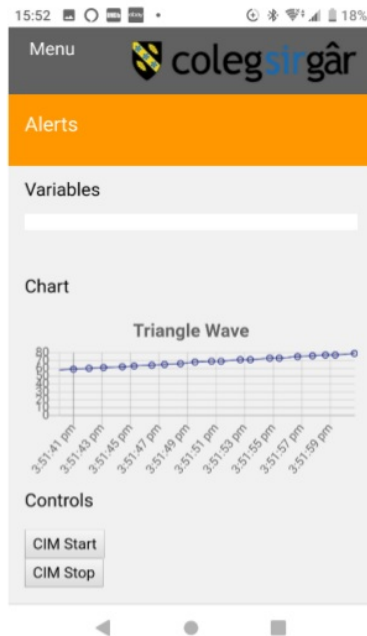


Figure 6.9 Master Page Scrolled

You have demonstrated a smooth running system but the original brief required a mimic of the CIM system conveyor. It is a shame that you didn't build that screen and simulate the data from the PLC. Your use of a completely different PLC program is good but does not relate to the original FDS.

7 EVALUATION AND THE WIDER ISSUES

7.1 REFLECTING ON FDS

To determine the level of success, a review of the FDS and comparison with the final product is carried out.

7.1.1 Hosting

It was specified that the ABA be hosted via the S7-1200 PLC itself with PLC Program and ABA stored on a Memory Card. The final product met this goal by storing the ABA as DataBlocks within the PLC Program itself (Figure 5.6) and implementation of the Memory Card (Figure 5.12).

7.1.2 Browser

It was specified that at minimum, that the Google Chrome browser should work with the ABA. Through testing, it can be confirmed that this goal was achieved on both the Desktop version (Figure 6.1) and the Mobile version (Figure 6.6) of Google Chrome.

7.1.3 Displaying Data

The ABA had to show variables accurately and without observable flicker. The Graph feature on the Master page animated as smoothly as possible without flicker and showed a readable chart. Regarding the Pneumatics page, difficulties were encountered in sourcing appropriate test equipment because of the lockdown imposed by the British Government on businesses. (Rigby, 2020)

7.1.4 Controllability and Response

While it was decided that the ABA could be compared to the HMI Screens of the CIM System, this has not been possible due to lack of physical access to the CIM System. In lieu of the original comparison, a visual observation of changes on the Pneumatics page compared to a Watch Table set up in TIA Portal was carried out instead, and changes were observed to be close to simultaneous.

7.1.5 Virtual LAN

During development, a Separate LAN from the CIM System was established using a Netgear N300 Wireless Router, which permitted communication between a Laptop running TIA Portal and other Mobile Devices. Due to lack of access to more sophisticated networking hardware at the time, a virtual LAN was replaced with a dedicated LAN.

7.1.6 Password Protection

While Password Protection was not implemented into the ABA itself, the Siemens S7-1200 PLC has built-in password protection to protect against unauthorised access.

7.1.7 Meeting Budget

The only purchase made for the Project was a 12MB SIMATIC Memory Card at £169. All other procurements were either on-loan from Coleg Sir Gâr (PLC and Laptop) or self-owned (Wireless Router). As the budget was set at £200, cost has been successfully minimised by coming in £31 underbudget.

7.2 HEALTH & SAFETY

At all points during the development of the ABA, Health & Safety guidance as dictated by legislations such as:

- Health and Safety at Work Act (1974)
- Electricity at Work Regulations (1989)
- Display Screen Equipment Regulations (1992)

Also followed were the BS 7671:2001 standards for electrical installations, which aided in adhering to the previously mentioned legislations.

As a great deal of the development of the ABA took place in a Home environment, it was advantageous as it provided a level of safety in working, and aided in the building of a viable Working from home environment, ideal for self-employed individuals.

7.3 PRODUCT SAFETY

By mitigating perceptible flicker, risk of causing epileptic seizures is only determinant using display screen equipment by the end user, of which fell outside the boundaries of the specification. As the development of the ABA utilised technologies developed by other developers and manufacturers, product safety in general lies mostly with Siemens as there have been no hardware modifications made to the PLCs design and specification.

7.4 DEVELOPMENT RISK

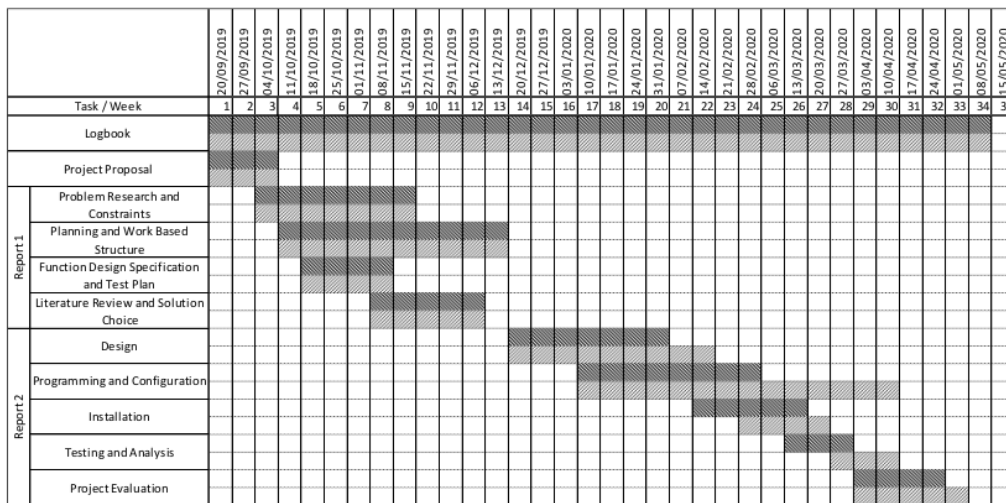
Not perceived as a Development Risk was the event of a Global Pandemic which led to widespread lockdowns (Coleg Sir Gâr, 2020). The result of the lockdown was loss of physical

access to the CIM System to facilitate development. To address the issue, a PLC was procured and utilised to continue development.

7.5 ENVIRONMENTAL CONSIDERATIONS

To minimise unnecessary use of electrical power, use of locally hosted web servers allowed development of the ABA to continue with the PLC in a powered down state until testing was required. Additionally, most hardware used, was sourced from storage, and re-used, avoiding unnecessary purchasing and use of electronics resources.

7.6 GANTT CHART



Deadlines were mostly met, however once events occurred where access to the CIM System became impossible (Coleg Sir Gâr, 2020), timely completion of tasks drifted as solutions to development problems had to be found.

8 CONCLUSIONS AND RECOMMENDATIONS

Despite successes in producing and sourcing assets to be used in the development of the ABA, a solution tailored for the CIM System was not achieved during the Project.

Elements that have been successful include:

- The identification of additional software suitable for developing webpages (Atom Text Editor).
- The overall appearance of the pages, displaying information with reasonable fidelity and no observable running errors.
- Application of Charts (Chart.js) and Gauges (JustGage.js) and updating graphics via AJAX.
- Building a usable webpage template that can be modified to develop further (index.html).
- Configuration of a Network to develop the ABA.
- Reliability via the Google Chrome browser across multiple Internet Enabled Devices

With these successes considered, development of the ABA with the CIM System could be continued via a remote working and communication via Video Conferencing and e-mail with those capable of accessing the CIM System to implement changes.

To develop further, the following can be undertaken:

- Implement Sliders included in jQuery to adjust values.
- Implement Progress Bars as Gauges for fluid or temperature levels.
- An error logging/diagnostics page, where errors can be formatted for download.
- Retention of Chart data for downloading as csv files.
- Buffer clear option to restart Chart trending.

9 WORKS CITED

B. B. Agarwal, S. P. T. M. G., 2010. *Software Engineering and Testing*. 1 ed. Burlington: Jones & Bartlett Learning.

Campbell, J., 2014. *Discovering the Internet: Complete*. 5 ed. Boston: Cengage Learning.

Charles E. Spurgeon, J. Z., 2014. *Ethernet: The Definitive Guide: Designing and Managing Local Area Networks*. 2nd ed. Sebastopol: O'Reilly Media.

Coleg Sir Gâr, 2020. *Coronavirus Update*. [Online]
Available at: <https://www.colegsirgar.ac.uk/index.php/en/18-college-content/682-coronavirus-update>
[Accessed 21 April 2020].

Fred Gerantabee, A. C. T., 2007. *Dynamic Learning: Dreamweaver CS3*. 1st ed. Sebastopol: O'Reilly Media.

Grigorik, I., 2013. *High Performance Browser Networking*. 1 ed. Sebastopol: O'Reilly Media.

Rigby, S., 2020. *COVID-19: What's the UK's exit strategy out of lockdown?*. [Online]
Available at: <https://www.sciencefocus.com/news/covid-19-whats-the-uks-exit-strategy-out-of-lockdown/>
[Accessed 20 April 2020].

Sherry Bishop, J. S. B. M. W., 2012. *The WEB Collection Revealed Standard Edition*. 1st ed. Boston: Cengage Learning.

Siemens, 2014. *Usage of S7-1200 Memory Card*. [Online]
Available at: <https://support.industry.siemens.com/cs/document/87133851/usage-of-s7-1200-memory-card?dti=0&lc=en-WW>
[Accessed 15 03 2020].

Soper, M. E., 2014. *The PC and Gadget Help Desk*. 1 ed. Indianapolis: Que Publishing.

10 GLOSSARY OF TERMS

AJAX – Asynchronous JavaScript and XML. The XML element has been supplanted by JSON.

`<body> </body>` – Used to contain the visible content of a page.

`<br/>` – Used to break up a paragraph.

class – Multiple Classes can be applied to HTML elements, to apply CSS Styles.

Comments – Used to explain HTML code without affecting the rendering of a webpage. In HTML a comment is formatted as `<!--Comment -->`, while in JavaScript a `//` is used.

`<div> </div>` – A container or division of a webpage `href` – links that point to pages via URLs.

`<head> </head>` – Contains metadata and links to stylesheets.

Headings – Just like a Word document, heading styles can be applied to text. There are six heading styles that can be applied.

`href` – Links that point to pages via URLs.

HTML – Hyper Text Markup Language.

`<html> </html>` – As a tag, everything that falls between the opening and closing of the tag, defines the webpage.

Hyperlinks – Another term for links. Can be clicked to navigate to another page.

ID – A HTML can have an id attribute to allow modification through scripting.

`<img src='name.png' />` – Used to embed an image in the page

JavaScript – A scripting language used in conjunction with HTML.

JSON - JavaScript Object Notation. Standard file format for storing data as an array.

`<li> </li>` – Item in a list.

`<link />` – Used to connect other documents such as stylesheets and scripts to the webpage.

`<ol> </ol>` – Ordered Lists. Items added between tags have numberings applied.

`<p> </p>` – A Paragraph.

`<table> </table>` – A Table. Rows are identified with a `<tr>` tag, Cells are identified with a `<td>` tag, table headings with `<thead>` and table body with `<tbody>`.

`<title> </title>` – The name of the webpage should be placed here.

`<ul> </ul>` – Unordered Lists- Items added between tags have bullet points applied.

Project Design, Implementation and Evaluation

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

64/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

Text Comment. This should summarise all that was planned and completed in the project, together with a statement on success. You have written it as an introduction to the report.

QM

Spelling

Spelling error

PAGE 9

Text Comment. What were the aims and objectives of the project?

Text Comment. You have introduced the report not the project.

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

Text Comment. It would have helped here to know what this PLC program would be doing.

PAGE 21

PAGE 22

PAGE 23

Text Comment. It is not clear what part the DC motor is playing in this project

PAGE 24

PAGE 25

Text Comment. Again, a DC motor's voltage is being controlled but there is no explanation of the practical setup

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

Text Comment. The IP addresses of the all CIM stations were 10.13.90.101 onwards

PAGE 48

Text Comment. Your report is written as an instruction manual, you are telling your reader what to do. Rather, you should be explaining what has been done to achieve the project outcome.

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

PAGE 56

PAGE 57

PAGE 58

PAGE 59

PAGE 60

PAGE 61

Text Comment. You have demonstrated a smooth running system but the original brief required a mimic of the CIM system conveyor. It is a shame that you didn't build that screen and simulate the data from the PLC. Your use of a completely different PLC program is good but does not relate to the original FDS.

PAGE 62

PAGE 63

PAGE 64

PAGE 65

PAGE 66

PAGE 67

GRADING FORM: NEW CSG-RUBRIC

EMYR
DYER

64

1. ENGAGEMENT WITH LITERATURE

Judicial use and critical evaluation of technical



literature in the design of the project: There is good evidence of reading from a range of sources with some critical evaluation. (3/5) Extent of reading, range of credible sources and use of Harvard method: The range is not extensive but the sources are credible., in addition the list included wider issues such as the current Covid-19 crisis. (3/5) Total = 6/10

6

2. KNOWLEDGE AND UNDERSTANDING



Advanced knowledge and critical understanding demonstrated in the design of the project: The design is very good in terms of the development of smooth data transfer and display from the webserver to the PC display. An excellent understanding of the principles of programming using HTML and JavaScript was demonstrated throughout the project. (7/10) Total = 7/10

7

3. COGNITIVE AND INTELLECTUAL



Testing and analysis of result: Good evidence of testing and analysis of results within the limitations brought about by the Covid-19 restrictions imposed after lock-down. An 8-stage test procedure was tabulated although results were mainly categorised as pass/fail. Additional screenshots were also included showing some samples of CIM output while the program was running. (6/10) Critical evaluation of the project outcomes: Review given of the requirements of the FDS in some detail though evaluation could have been more judicial in terms of the front end of the delivered solution. The final

18

product does not relate directly to the FDS. A brief evaluation of each aspect of the program was included and some explanation was offered for shortfalls in the expected performance. Wider issues such as H&S and environmental (6/10) Conclusions and recommendations: Generally good, honest set of conclusions with recommendation for completing the requirements of the FDS. (6/10) Total = 18/30

4. PRACTICAL APPLICATION



Formulation of innovative and creative design solution to the project problem: Project solution is innovative and provides a good method of display of internal PLC data for use on the webserver. (7/10) Deployment method, use of material, tools, equipment and techniques in the construction of the project: Professional approach adopted with use of suitable software for a robust and secure solution. Extensive use of software development environment. (3/5) Quality of the final product or system: Smooth running of the ABA shows high quality results but the CIM system mimic was not implemented. Final implementation into the college CIM was not possible due to Covid-19 restrictions. (7/10) Use of test equipment and presentation of results of the project: Good use of testing methods with PLC and software. (3/5) Management of the design process: Basic Linear Gantt chart show some project management, working within the restrictions imposed by Covid-19 rules. (2/5) Total = 22/35

22

5. TRANSFERABLE SKILLS



Executive summary, acknowledgements and introduction: Summary is more of an introduction to the report rather a summary of the project. Good acknowledgements but the introduction should explain the aims and objects of the project. Acknowledgements were included thanking contributors. (2/5) Quality of the report: Generally well written with few errors. High quality figures. The report was well presented and was written in a professional style. (4/5) Word count : Declared as 5555. Word count was within limits, no penalty imposed. (5/5) Total = 11/15