

2221112 2. Factory IO Control System SGEE4004 Assignment

by William Utting

Submission date: 17-Apr-2023 10:56PM (UTC+0100)

Submission ID: 202422509

File name:

18983_William_Utting_2221112__2._Factory_IO_Control_System_SGEE4004_Assignment_181751_1746840805.pdf
(2.63M)

Word count: 3299

Character count: 17376

Course: HNC Electrical and Electronic Engineering
Module: SGEE4004 - Programmable Logic Controllers
Assignment Title: 2. Factory IO Control System
Name: William Utting
University number: 2221112
Date Submitted: 17/04/2023
Word Count: 2527 words

Table of contents

Good Structure

Sections should be numbered according to the brief

Work Plan for Lifelong Learning	3
Research	4
Hardware Design and I/O List	7
Program Design and Explanation	9
System Testing	11
Meeting the Customer Needs	12
Conclusions and Recommendations	13
Team Working	14
Citation List	15
Appendix	
Appendix A: Research	17
Appendix B: Hardware Investigation	20
Appendix C: Program Design and Explanation	21
Appendix D: System Testing	23

excellent work plan

Work Plan for Lifelong Learning

The task description of the assignment must be read thoroughly to gain a full understanding of the objectives. To determine the most appropriate language for the application, research into possible languages and their characteristics must be conducted. A suitable language will then need to be selected. Using Factory IO the appropriate hardware of the tank and valves must be selected, this will ensure that the program functions as intended. A list of the input and outputs can be created, this will assist in the programming of the system. The program will be designed to fit the requirements of the task description. HMI screens will be designed to display as much data as possible in an easily accessible format. The programming will be constructed in a simple and effective structure. To ensure the program functions correctly it will be tested in the Factory IO digital environment. To ensure the final product meets the customers needs the programs of other students will be compared to each other. If any improvements on the programs are found the program will be edited to reflect the result of the discussions. Conclusions and recommendations on the program can be made once the program has been completed and any changes introduced. An explanation of the advantages of the group discussion will be provided. Citations for the research, figures and tables will be included at the end of the paper.

Paragraphs should be fully justified text as required in the brief

Research

IEC Languages In TIA Portal

Within TIA Portal a program can be created using five programming languages. Multiple languages can be used within a single program. The five languages are: instruction list (IL), structured text (ST), function block diagram (FBD), ladder diagrams (LD/ LAD) and sequential function charts (SFC) (Siemens, 2003). The programmable logic controller that is in use at the college is part of the 1200 series of Siemens processors. There are only three out of the five languages available in the 1200 series: LAD, FBD and a ST language called structured control language (SCL) (Programming Guideline for S7-1200/1500 TIA Portal, 2018).
author only

Instruction List

Instruction list is very similar in its structure to an assembly language. The code is read as a list and the instructions are executed in order. Due to the simplicity of the language, IL is best used for mathematical and logical operations (Schneider-electric, 2019). IL is a low level language, this allows operations to be executed at a high rate. Another benefit of its simplicity and low level is the small amount of memory required to run an instruction list. However, as the complexity of the program increases the readability of the IL decreases and it can become hard to understand and fault find the code (Budimir M, 2017). An example of an instruction list can be found in appendix A, figure 1.

Structured Text

Structured text is a high level, text based language that can be used to manipulate variables through logical and mathematical functions. A ST based language can have more methods of control when compared to graphical languages. In larger programs, the logic of a program can be understood easier in ST when compared to other languages. This is due to the simple statements, comments and methods of compartmentalising code. Many ST languages are similar in syntax, therefore, most programming gained in one ST language are transferable to other languages (PLC Academy, 2015). As the language is text based the memory used while executing the program is less than graphical options. Any program written in a text based language can be distributed easily through text files, this allows manufacturers to ship products to customers and transfer the program through email. This also allows the customer to upload the program to as many compatible machines as they need (PLC Programming Courses for Beginners | RealPars, 2021).

author only

Function Block Diagram

FBD languages are graphically based. However, if function blocks are implemented inefficiently the program can become difficult to read. The supplier of the programming tool will provide basic function blocks within the programming environment. This allows the developer to drag and drop functions into the program and set input and output variables easily. Functions can be created by the developer and saved as custom function blocks. Custom function blocks can be used multiple times in one program and also over multiple programs. This will save the developer having to program the same function multiple times in different locations of the program (PLCopen® for efficiency in automation Structuring Program Development with IEC 61131-3 General trends). FBD can be effectively applied when multiple functions need to be connected together and basic logical or mathematical operations need to be conducted on input or outputs (Peter, 2018).

Ladder Diagrams

author could be IEC

Ladder diagrams are a graphical language (IEC 61131-3: a standard programming resource, 2017). The symbols used within the language reflect what an engineer would find in electrical drawings. This allows an experienced engineer to understand and learn the language at a fast pace. The symbol design is also useful for converting electromechanical systems into PLC driven systems (The Automization, 2020). The language reads from left to right and the order of execution is top to bottom. This allows the user to read ladder diagrams like a book (Peter, 2017).

Sequential Function Charts

SFC is a graphical representation of a series of steps and transitions between steps. Each step is an operation, a step can have multiple transitions to other steps. Within a transition conditions can be set so the program will only move onto the next step when the correct conditions are met (Schneider-electric.com, 2020). See appendix A, figure 5 for example.

Application

IL has a narrow range of applications it is useful in. The complexity of the language and it's non-user friendly format makes it difficult to use. Therefore, it will not be used in this application. ST may be useful in this scenario, the format of the language is easy to understand and the flexibility of the language is useful. FBD will most likely be the main language of the program. The option of custom functions will allow the flow and level to be controlled easily. LD will also be used to connect and control the activation of the function blocks. SFC could easily be used to control the program, however, unfamiliarity with the language may cause unwanted difficulty when creating the system. In conclusion FBD will

be used for the more complex control and LD will be used to control the simple operations.
ST may be used within FBD if the programming is too complex.

Your understanding of FBD is incorrect

Hardware Design and I/O List

The specification of the program requires proportional-integral-derivative control (PID). PID control increases or decreases the output value in relation to the ^{error value} setpoint of the system, this requires a variable output. Therefore, the fill and drain valves of the tank must be analogue, this will allow the flow of water through the valves to be controlled. It is also specified that the output flow of the tank and the level of the liquid within the tank needs to be controlled. This could be done by complex mathematical functions, however, it is easier if sensors are installed. A level sensor is installed in the tank and a flow meter is installed at the drain valve. Information and control must be provided to engineers, therefore, a panel with start, stop and reset switches is provided. The setpoint and process variable are shown as well as a potentiometer to change the setpoint is included. If the tank requires cleaning or the fill valve needs servicing a ladder is provided for access. The hardware can be viewed in appendix B figure 1.

Appendix B figure 2 shows the input/output (I/O) list of the tank. The I/O of the tank is as follows:

Name	Input/Output	Address	Analog/Digital
Factory I/O (Paused)	Input		Digital
Factory I/O (Reset)	Input		Digital
Factory I/O (Running)	Input	%I0.3	Digital
Factory I/O (Time Scale)	Input		Analog
Flow meter	Input	%ID34	Analog
Level meter	Input	%ID30	Analog
Reset	Input	%I0.1	Digital
Setpoint	Input	%ID38	Analog
Start	Input	%I0.0	Digital

Stop	Input	%I0.2	Digital
Discharge valve	Output	%QD34	Analog
Factory I/O (Camera Position)	Output		Analog
Factory I/O (Pause)	Output		Digital
Factory I/O (Reset)	Output		Digital
Factory I/O (Run)	Output		Digital
Fill valve	Output	%QD30	Analog
PV	Output	%QD42	Analog
Reset light	Output	%Q0.1	Digital
SP	Output	%QD38	Analog
Start light	Output	%Q0.0	Digital
Stop light	Output	%Q0.2	Digital

Program Design and Explanation

The human machine interface (HMI) of the tank system must show critical information and provide controls to the operator. In total there are four screens that need to be designed: manual controls, semi-automatic control screens and a fully automatic control screen. Each screen has certain information that must be displayed and controls for the tank.

Manual control screen:

- Fill and drain valve fully open/ closed buttons
- Fill and drain valve control
- Manual control button
- Valve positions
- Flow rate
- Level

you should be describing what you made not what you should have made

Semi automatic controls for drain valve

- Semi automatic control button
- Setpoint for flow
- Current flow
- Graphical representation of flow
- Proportional variable
- Integral variable
- Derivative variable
- Error
- Control Variable

Semi automatic control for fill valve

- Semi automatic control button
- Setpoint for level
- Current level
- Graphical representation of the tanks level
- Proportional variable
- Integral variable
- Derivative variable
- Error
- Control Variable

Automatic controls

- Automatic control button
- Setpoint for level
- Setpoint for flow
- Graphical representation of both level and flow

- Current values of level and flow

Each screen can be roughly designed to ensure that the data is displayed in a clear manner (see appendix C, figures 1 to 4). The program behind the HMI must also be designed before the final system is created. The PID controller can be made into its own function block. This will reduce the complexity of the system. The PID function block will have its own local variables, this allows there to be multiple instances of the function. There will be two instances for flow and level semi automatic control. The automatic control will turn both semi automatic controls on at the same time so they are constantly fighting each other.

You don't need to tell your reader that testing is important

You should have produced a testing table instead

System testing

Testing can be conducted throughout programming in addition to a final system test that tests the entirety of the system. By testing regularly, the bug finding and fixing process is easier. All of the bugs are fixed before moving onto the next feature of the program, this means that any new code added to the program is the source of any new errors. When the program has all of the features it requires it should be able to run. The full system test can then be conducted and any oversights in how the program behaves can be corrected.

The regular programming of the system indicated that the manual controls functioned correctly and the semi automatic PID controls functioned for both flow and level. The graphs for flow and level also functioned correctly and were assigned an appropriate scale. The automatic function of the program was constructed, however, it was never tested thoroughly. The software's licence expired before the program could be fully completed. This caused a lack of evidence for testing and the incomplete status of the program (see appendix d, figure 1).

Meeting Customer Needs

The criteria that a customer requires provides the structure of the program. The criteria for the tank control program can be obtained from the assignment brief. The current version of the program satisfies the majority of the requirements. Controls for filling and emptying the tank are provided on the manual control screen. This screen also includes the status of all devices in the system. The screen displays the position of both valves, the level of the tank and the flow rate coming out of the drain valve. The semi automatic screens can change the values of the PID constants to tune the control. Semi automatic screens also include flow-time and level-time graphs, this assists the engineer in tuning the PID constants. A function block contains the PID control. The function has suitable variables for simple use. The automatic screen is not finished. It does contain a graph, however, the graph does not have suitable scales. The automatic code has not been fully tested, therefore, there are likely errors within the code that will cause the automatic mode to not function. The buttons on the automatic page do not work.

In conclusion, the program, in its current state, does not satisfy the customers needs. It is highly likely that a functional program has been made by a peer conducting the same assignment.

Better planning is needed

Conclusions and Recommendations

By consulting the brief of assignment it can be concluded that the objectives of the assignment were not met to their full extent. The program was not fully completed. With no access to the software that the program is created on, no testing of the program can be conducted. This has caused the objective of testing the program to not be met. An absence in a lecture means that this program has not been compared against peers' programs. Due to the technical issues with the software, evidence of the programming cannot be gathered and displayed in the assignment. However, the program has met some of the criteria and any objectives that do not rely on the software functioning have been met to as best of a standard as has been possible. A plan of the project has been produced. Hardware investigation of the tank has been conducted and the I/O of the system has been listed for use in the program. The HMI screens have been designed for a user friendly interface. The basic structure of the code was determined and discussed. The final program has functioning manual and semi automatic controls and the HMI screens reflect the criteria outlined in the brief. Overall, though incomplete, the functioning areas of the program meet the assignment brief.

The largest recommendation for the program is for it to be completed. The code behind the program is optimised well. However, there could be more optimisation if a non-graphical language was used as suggested in the research section. The HMI screens could be colour coded better to indicate the use of some buttons at a glance. There is also a lot of information on the semi automatic screens which may cause an operator to be confused. Perhaps a separate screen with all of the data displayed on it could have been used, or a simpler version of the screen could have been designed as well as the two that display all of the data. The program being created in TIA portal has created a few issues. The main computer used for this assignment cannot run TIA portal for unknown reasons. Siemens support, college IT support and an IT engineer have tried to fix the issue but have not been able to get the software functioning. The IT engineer found a work around using a remote desktop connection to a different computer with the software installed. This functioned for a while before the licence for the software expired. It may have been easier if a more generic cloud-basic programming piece of software was used. Due to the course being part-time the assignment must be available over multiple devices in different locations, a cloud based software would fix this issue.

Your program could have been developed on the college computer

You did not make your tutor aware of these issues

Team Working

If a lecture had not been missed a team could have decided the viability of this program compared to other programs constructed for the same assignment. This would have given an unbiased view of the program and would have shown areas of improvement. The exercise could have also been an opportunity to have multiple people test the program, this would have highlighted any bugs or unintended features that need to be fixed.

To work on the assignment there needed to be communication between different parties. To attempt to get the software functioning correctly on the main computer college IT support had to be contacted. They did not know how to correct the issue, therefore, Siemens support had to be contacted. Siemens support did not reply in time for the assignment deadline. An IT engineer also could not find the error in the software. However, they did provide a work around that functioned for a while. Contacting support requires a detailed explanation of what the error is. This uses IT skills and communication skills to narrow down the issue and eventually resolve the error. This teamwork assisted in developing the program by allowing access to the software required to create the program.

Wide Range of Sources

Correct method of listing.

Citation List

Siemens. (2003). *S/OS*. [online] Available at:

<https://support.industry.siemens.com/cs/document/8790932/iec-61131-3-and-simatic-s7?dti=0&lc=en-GB> [Accessed 3 Apr. 2023].

Programming Guideline for S7-1200/1500 TIA Portal. (2018). [online] Available at:

https://cache.industry.siemens.com/dl/files/040/90885040/att_970576/v1/81318674_Programming_guideline_DOC_v16_en.pdf [Accessed 3 Apr. 2023].

Schneider-electric. (2019). *Instruction List (IL) Language*. [online] Available at:

https://product-help.schneider-electric.com/Machine%20Expert/V1.1/en/SoMProg/SoMProg/FBD_LD_IL_Editor/FBD_LD_IL_Editor-6.htm [Accessed 3 Apr. 2023].

Budimir M. (2017). *What are Instruction Lists (ILs) for PLC programming?* [online]

Motioncontroltips.com. [online] Available at:

<https://www.motioncontroltips.com/instruction-lists-ils-plc-programming/> [Accessed 3 Apr. 2023].

Snayush (2022). *PLC programming languages*. [online] Medium. Available at:

<https://medium.com/@snayush77/plc-programming-languages-6d4ba691e48f> [Accessed 3 Apr. 2023].

PLC Academy. (2015). *Structured Text Tutorial For PLC Programmers*. [online] Available at:

<https://www.plcademy.com/structured-text-tutorial/> [Accessed 4 Apr. 2023].

PLC Programming Courses for Beginners | RealPars. (2021). *Basics of Structured Text (ST) Programming | Examples & Applications*. [online] Available at:

<https://realpars.com/structured-text/> [Accessed 4 Apr. 2023].

PLCopen ® for efficiency in automation Structuring Program Development with IEC 61131-3

General trends. (n.d.). [online] Available at:

<https://www.plcopen.org/sites/default/files/downloads/7steps.pdf> [Accessed 4 Apr. 2023]

Peter (2018). *Function Block Diagram (FBD) Programming Tutorial - PLC Academy*. [online]

PLC Academy. Available at:

<https://www.plcademy.com/function-block-diagram-programming/> [Accessed 4 Apr. 2023].

IEC 61131-3: a standard programming resource. (2017). [online] Available at: https://plcopen.org/sites/default/files/downloads/intro_iec_oct2016.pdf [Accessed 5 Apr. 2023].

The Automization. (2020). *A Simple (But Complete) Guide : PLC Ladder Logic Programming(Basics)*. [online] Available at: <https://theautomization.com/ladder-logic/> [Accessed 5 Apr. 2023].

Peter (2017). *Ladder Logic Tutorial for Beginners - PLC Academy*. [online] PLC Academy. Available at: <https://www.plcacademy.com/ladder-logic-tutorial/> [Accessed 5 Apr. 2023].

Schneider-electric.com. (2020). *Logic Guide - How Sequential Function Charts Work (Sequential Function Charts)*. [online] Available at: <https://tprojects.schneider-electric.com/GeoSCADAHelp/Geo%20SCADA%202020/Content/LogicGuide/HowSequentialFunctionChartsWork.htm> [Accessed 5 Apr. 2023].

Appendix A: Research

Figure 1 - (Snayush, 2022) instruction list example

```
PROGRAM
VAR
    TIMER1 : TON;
    TIMER2 : TON;
    TAG-1 : BOOL;
    TAG-2 : INT;
    TIN : TIME;
    TOUT : TIME;
END_VAR
-----
LD TAG-1
ST TIMER1.IN
GOTO MARK1
CAL TIMER1(
    PT:TIN1,
    ET:>TOUT1)
LD TIMER1.Q
ST TIMER2.IN
-----
MARK1:
LD TAG-2
AND 130
OR 3
```

Figure 2 - (PLC Academy, 2015) structured text example

```
PROGRAM stexample
VAR
    x : BOOL;
END_VAR
x := TRUE;
REPEAT
    x := FALSE;
UNTIL x := FALSE;
END_REPEAT;
END_PROGRAM;
```

Figure 3 - (Peter, 2018) function block example

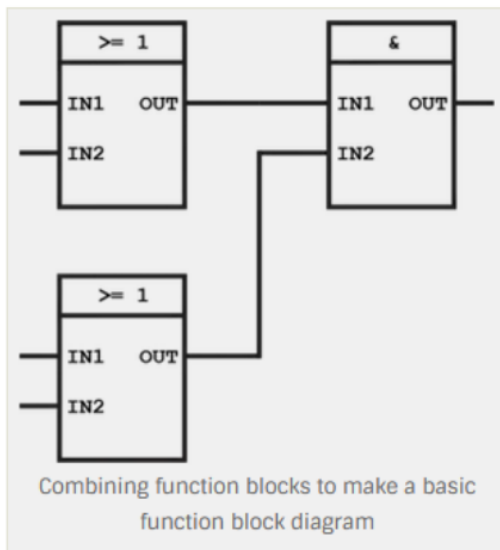


Figure 4 - (The Automization, 2020) ladder diagram example

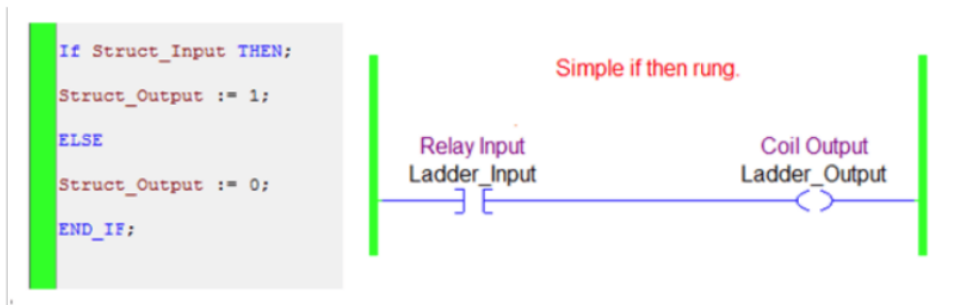
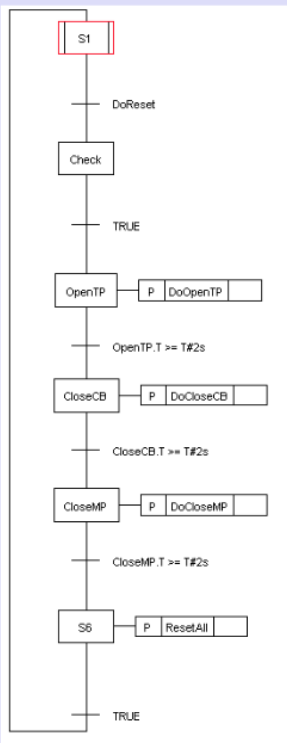


Figure 5 - (Schneider-electric.com, 2020) sequential flow chart example

Example:

The following SFC shows how the sequence can be structured from top to bottom:



1. S1 is step 1. The system begins executing the SFC from this step.
2. If the condition for the DoReset transition is true, the S1 step is deactivated and the Check step is activated.
3. If the TRUE transition has a condition that is true, the Check step is deactivated and the OpenTP step is activated.
4. If the OpenTP step is activated, the DoOpenTP action is performed. The configuration for the action defines what happens when the action is performed—it could write a value or state or could execute a method.
5. If the OpenTPT variable value is greater than 2 seconds, the OpenTP step is deactivated and the Close CB step is activated.
6. If the CloseCB step is activated, the DoCloseCB action is performed.
7. If the CloseCB.T variable value is greater than 2 seconds, the CloseCB step is deactivated and the Close MP step is activated.
8. If the CloseMP step is activated, the DoCloseMP action is performed.
9. If the CloseMPT variable value is greater than 2 seconds, the CloseMP step is deactivated and the S6 step is activated.
10. If the S6 step is activated, the ResetAll action is performed.
11. If the condition for the transition that follows S6 is true, the whole sequence repeats itself, starting from S1.

Appendix B: Hardware Investigation

Figure 1 - Hardware

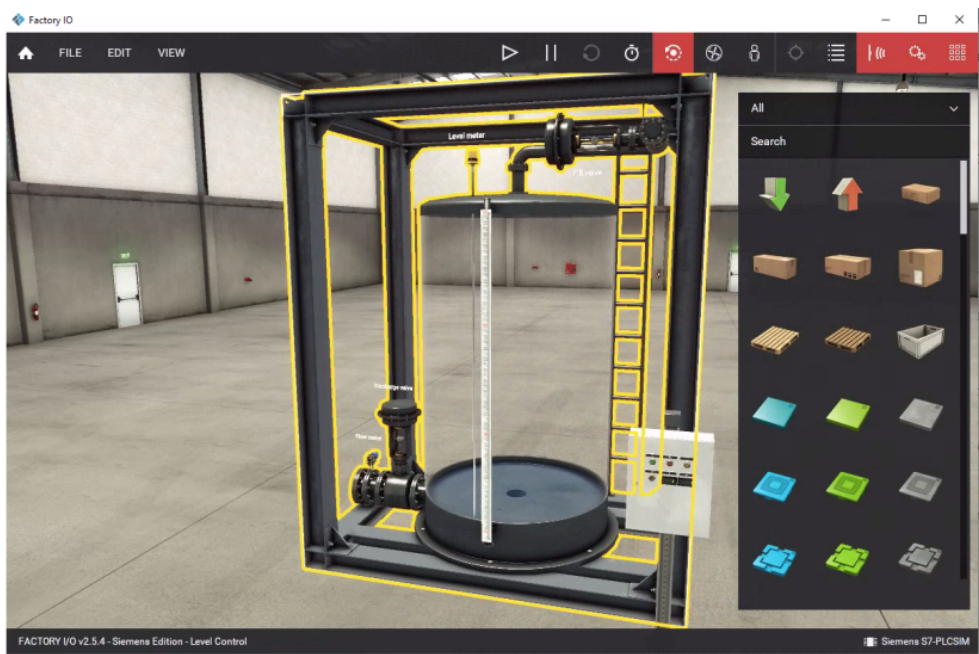
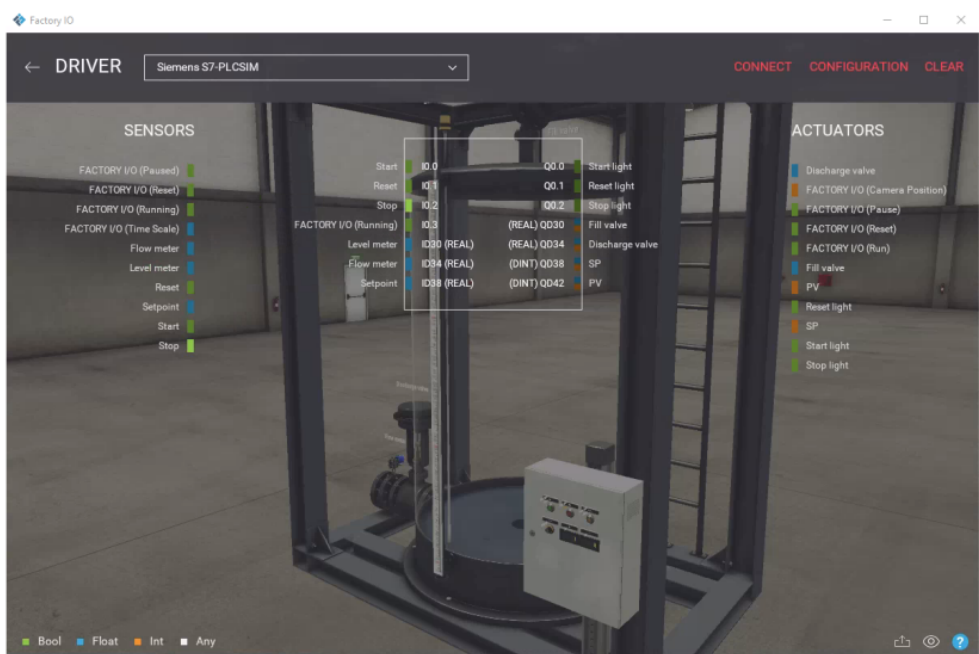


Figure 2 - I/O



Appendix C: Program Design and Explanation

Figure 1

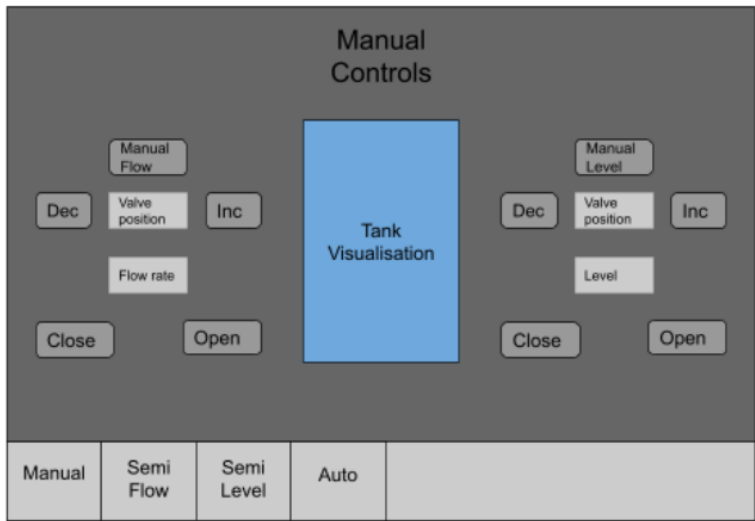


Figure 2

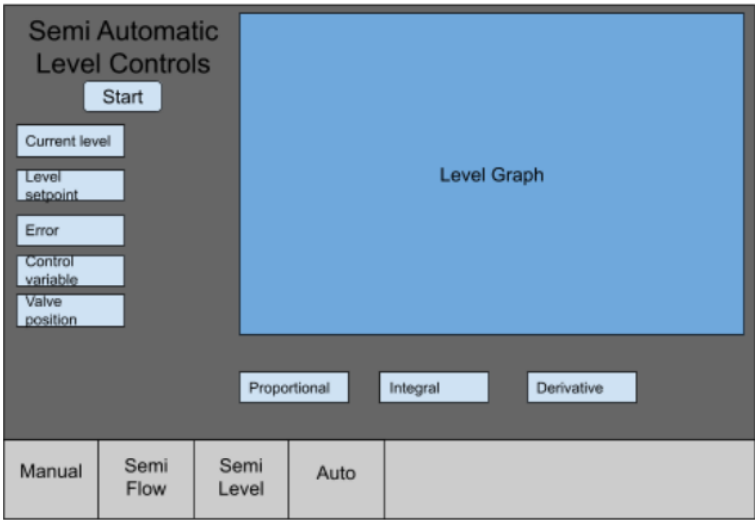


Figure 3

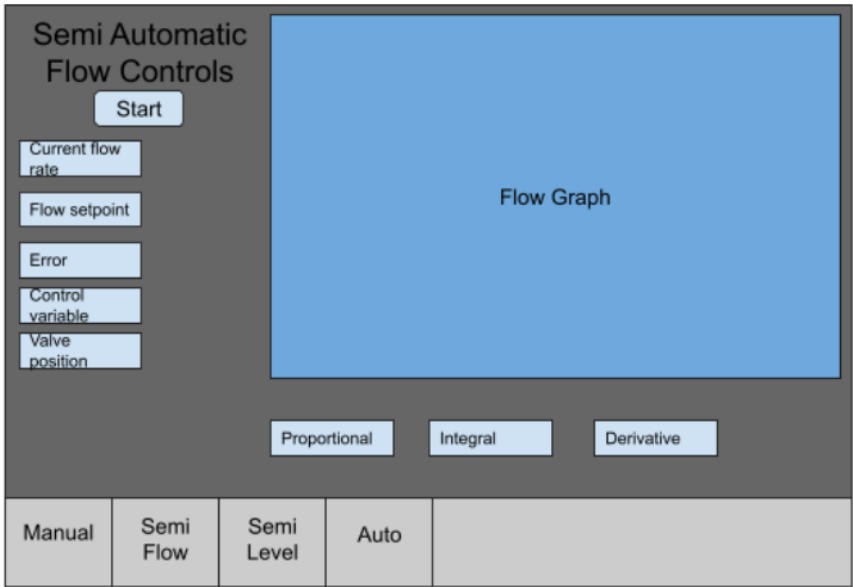
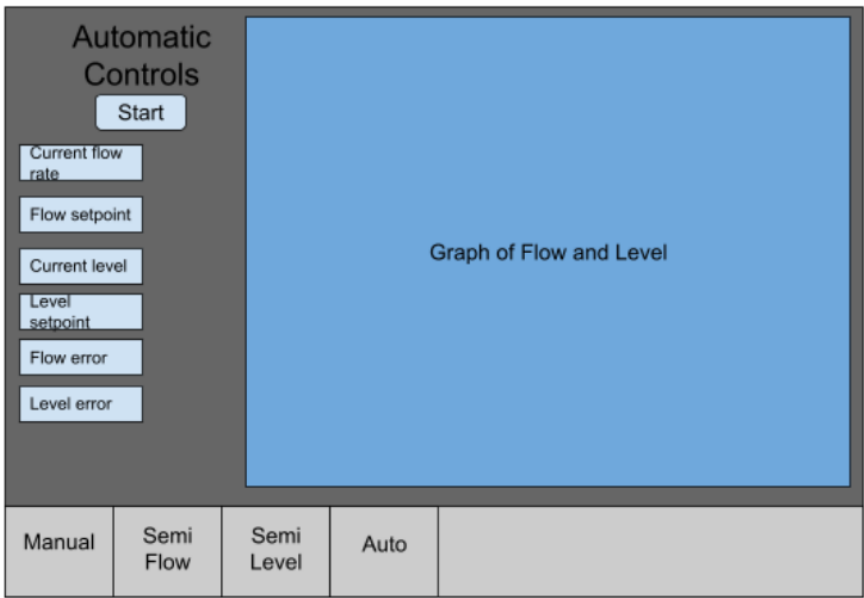


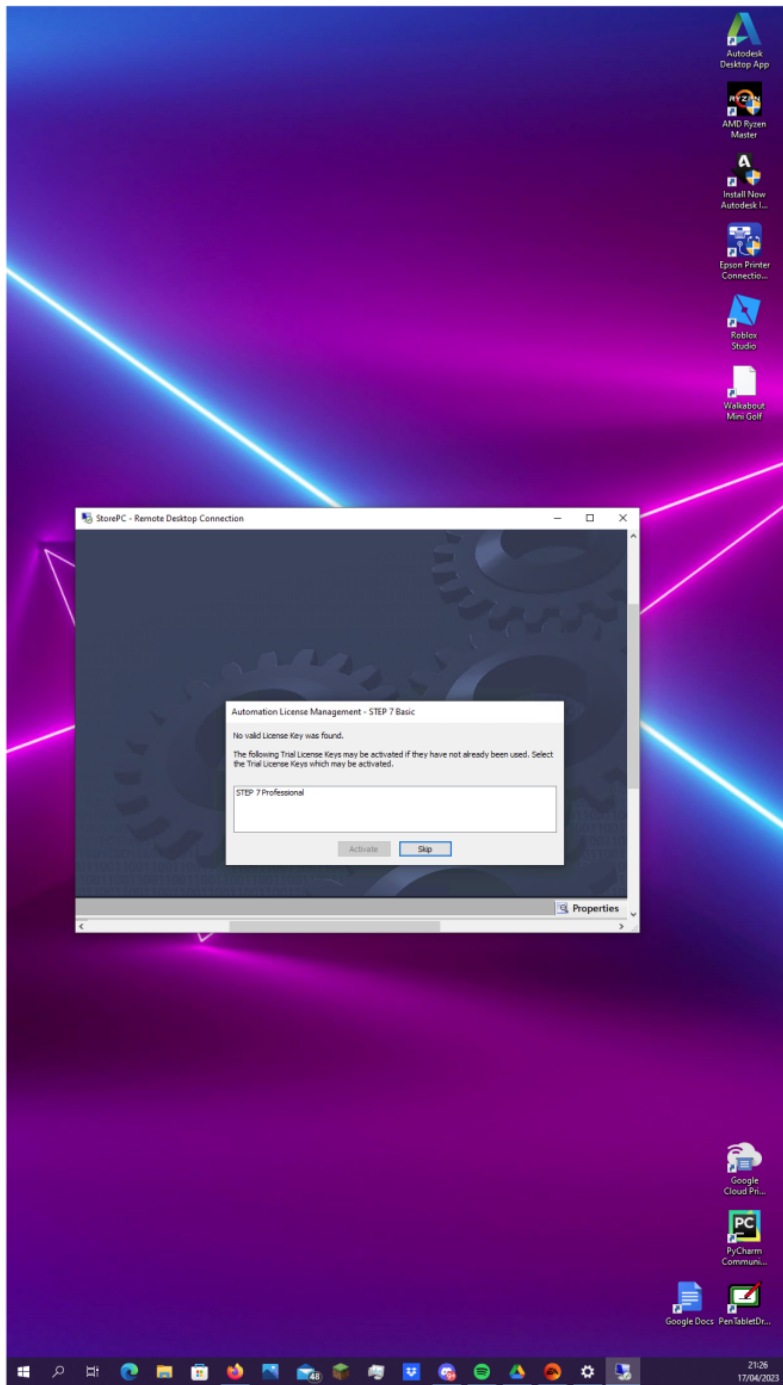
Figure 4



Appendix D: System Testing

Figure 1

You need to plan better to avoid.



2221112 2. Factory IO Control System SGEE4004 Assignment

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

53 /100

PAGE 1

PAGE 2

QM

Good Structure

Report is structured exactly as stated in the assignment brief.

QM

Sections should be numbered according to the brief

PAGE 3

Text Comment. excellent work plan

QM

Paragraphs should be fully justified text as required in the brief

PAGE 4

Text Comment. author only

Text Comment. author only

PAGE 5

Text Comment. author could be IEC

PAGE 6

Text Comment. Your understanding of FBD is incorrect

Text Comment. error value

Text Comment. you should be describing what you made not what you should have made

Text Comment. You don't need to tell your reader that testing is important

Text Comment. You should have produced a testing table instead

Text Comment. Better planning is needed

Text Comment. Your program could have been developed on the college computer

Text Comment. You did not make your tutor aware of these issues

QM

Wide Range of Sources

You have shown good evidence of reading from a wide range of sources.

QM

Correct method of listing.

You have listed your sources correctly according to the Harvard method.

Text Comment. You need to plan better to avoid.

GRADING FORM: NEW CSG-RUBRIC

WILLIAM UTTING

53

1. ENGAGEMENT WITH LITERATURE



Excellent research from a wide range of source
mistly cited correctly and all listed correctly. 16/20

16

2. KNOWLEDGE AND UNDERSTANDING



Very good knowledge and understanding, good
hardware description but no program inlcuded so
comments cannot be viewed. One
misunderstanding in research about FBD. 10/20

10

3. COGNITIVE AND INTELLECTUAL



There is no evidence of testing and no test plan
either. A good conclusion and recommmendations

10

but you must avoid blaming your circumstances.
Better forward planning and communications with
your tutor would have prevented failing at the end.
10/20

4. PRACTICAL APPLICATION



Good HMI designs but not implemented on TIA
Portal software. No program included that would
have demonstrated your practical skills. 5/20

5

5. TRANSFERABLE SKILLS



An excellent workplan, good structure, well written
within the word count. However, format of
paragraphs is not correct, sections should be
numbered and no evidence of team working
provided. 12/20

12