

2224559

by Mohammed Ibrahim

Submission date: 17-Apr-2023 10:51AM (UTC+0100)

Submission ID: 202325592

File name: 19116_Mohammed_Ibrahim_2224559_181751_1169237207.pdf (2.67M)

Word count: 4307

Character count: 28759

Course: HNC Electrical and Electronic Engineering
Module: SGEE4004 Programmable Logic Controllers
Assignment: 2. Factory IO Control System
Name: Mohammed Zeeshan Ibrahim
University number: 2224559
Date submitted: 17/04/2023
Word count: 2360 Words

Table of Contents

Good Structure

List of Figures	4
1 – Work Plan for Lifelong Learning	6
2 - Research	7
Ladder Logic:.....	8
Sequential Function Chart (SFC):.....	9
Structured Text:.....	10
Function Block Diagram (FBD)	11
Instruction List:	12
3 - Hardware Investigation and Input/Output List.....	13
4 – Program Design and Explanation.....	15
Main Program Block	15
Cyclic Interrupts:.....	15
PID Control:.....	15
Sequence	16
HMI Menu:.....	16
5 – System Testing	17
6 – Meeting the Customer Needs.....	18
7 – Conclusions and Recommendations	19
8 – Team Working.....	20
9 - Citation List	21
10 - Appendices	23
Appendix A: – Research.....	23
Ladder Logic:	23
Sequential Function Chart (SFC):	25
Structured Text:	26
Function Block Diagram (FBD):.....	27
Instruction List:.....	28
Appendix B:– Program Design and Explanation	29
Main Program Block:.....	29
Cyclic Interrupts:	34
PID Control:	35
Sequence:.....	37
HMI Menu:	38
PLC Tag List:	40

Appendix C: – System Testing.....	42
Appendix D: - Meeting the Customer Needs.....	47

List of Figures

Figure 1 Factory IO Inputs and Outputs	13
Figure 2 Factory IO Tank with Both Valves and Flow Meter	13
Figure 3 Ladder Diagram Symbols (Edrawsoft, 2023)	23
Figure 4 Ladder Programming Example (Edrawsoft, 2023)	24
Figure 5 SFC With its Corresponding Ladder Diagram (Devasia, 2022)	25
Figure 6 Structured Text Example (Dietrich, 2022)	26
Figure 7 Using Multiple Inputs and Feeding an Output into An Input (Edrawsoft, 2023)	27
Figure 9 'AND' Block	27
Figure 8 Timer Block	27
Figure 10 Instruction List Program Example	28
Figure 11 Networks 1&2	29
Figure 12 Network 3	30
Figure 13 Networks 4, 5, 6 & 7	31
Figure 14 Network 8	32
Figure 15 Network 9	33
Figure 16 Network 10	33
Figure 17 Networks 1&2 for Level Control	34
Figure 18 Networks 1&2 for Flow Control	34
Figure 19 Networks 1,2, 3 & 4	35
Figure 20 Network 5	36
Figure 21 Networks 1,2 & 3	37
Figure 22 Semi – Automatic Level Control Engineer Screen	38
Figure 23 Manual Control Engineer Screen	38
Figure 24 Auto Level and Flow Operator Control Screen	39
Figure 25 Semi - Automatic Flow Control Engineer Screen	39
Figure 26 PLC Tag List 1 - 29	40
Figure 27 PLC Tag List 30 - 53	41
Figure 28 - Test 1 Manual Mode	44
Figure 29 - Test 2 Semi-Auto Level Mode	44
Figure 30 - Test 3 Semi-Auto Flow Mode	45
Figure 31 - Test 4 Auto Mode Setting Level and Flow Rate	45
Figure 32 - Test 4 Draining the Tank	46

Figure 33 Before Altering Resolution.....	47
Figure 34 After Altering Resolution.....	48
Table 1 Simulation Test Table.....	42

1 – Work Plan for Lifelong Learning

This assignment will first be approached by carrying out research on various IEC PLC programming languages available to the programmer. The hardware of the system will then be investigated, and inputs and outputs will be distinguished. The TIA project will then be created with suitable function calls, function blocks and cyclic interrupts. After this, manual controls will be provided for filling and emptying the tank. Semi-automatic control will also be added using PID. A fully automatic mode will also be added along with an HMI screen for the engineer and operator to interpret. The program will then be tested on Factory IO to ensure it functions correctly. Once tested, team working will be done to share ideas and designs to ensure the program meets the correct specifications and customer requirements will be discussed.

Excellent work plan

2 - Research

The International Electrotechnical Commission's IEC 61131 programming standard was first released in 1993. It ensures programming languages operate cost -effectively, efficiently, and accurately and is known to play an important role in open architecture control systems. There are five programming languages that fall under the IEC category, which are Ladder Logic, Sequential Function Chart, Structured text, Function Block Diagram, and Instruction List. (Petrotech, 2017).

The programming languages stated above will be researched and the most appropriate language will be used in this assignment (see next page).

Ladder Logic:

Ladder logic visualizes written program in the form of a ladder, hence the name. These are then based on circuit diagrams, are often easy to interpret and most used in PLCs for industrial applications. Ladder logic is mostly used for sequential manufacturing, where sequential processes are required. The main concept of this language is to allow users without prior programming knowledge to understand the function of the program, which uses common engineering symbols, such as relays and contacts (Techopedia Inc., 2023).

Ladder logic is used when replacing old hardwired systems, such as relay circuits and is used in essential control systems. It can perform functions such as mathematical, arithmetic, linear, measuring etc, in which some of these will be required for use in PID (Edrawsoft, 2023).

The diagram in Appendix A shows the symbols used in ladder diagram along with an example. The example shows a normally open input, normally closed input, and an output latch. It functions as when Input A closes, Output Y closes and latches normally open contact Output Y. When Input B receives power, it will break the circuit, which will turn off Output Y and unlatch it, (see figure 3 and 4 in appendix A, pages 23-24) (Edrawsoft, 2023).

Sequential Function Chart (SFC):

SFC programming offers a visual representation of a program, like a flow chart. SFCs are easier to visualize a scenario where there are multiple operations occurring in a sequence. It allows the user to break down a large process into smaller steps and identify the function they are looking for. States and steps are visualized with boxes, and vertical lines connect the steps. Each step has its own output, connected with a horizontal line. After one step is completed, the program will move onto the next step if the conditions of the transitions are satisfied.

There are two conditions that are to be followed when programming in SFC, in which there should be a step in between two transition conditions, and there should always be a transition between two steps.

SFCs can also be represented with a ladder diagram. The first rung in the ladder diagram represents step 1 in the SFC. Once 'In 1' closes, 'Out 1' will have power, which will satisfy step 1 and move onto the transition. Once 'In 2' closes 'Out 2' will have power, which will satisfy step 2. (See figure 5 in appendix A, page 25).

(Devasia, 2022).

Structured Text:

Structured text is more of a traditional way of PLC programming. It does not require a background of PLC knowledge to be able to program in structured text as it resembles languages such as Java and Python. TIA Portal uses structured text and is called 'SCL' in the software. Structured text also does not take as much memory as ladder and SFC, strictly due to being text based. This means more program functions can be stored in a PLC with less memory (RealPars B.V. , 2023).

Structured text is used in automation, typically robots. Code can be easily represented with structured text programming and inputs and outputs can be mapped to variables in the code. It gives the programmer a larger amount of freedom on developing the program structure and flow (Dietrich, 2022).

See figure 6, page 26 in appendix A for structured text example. This is a basic demonstration of how it would be written in TIA portal.

Function Block Diagram (FBD)

FBD is a widely used programming language that performs functions in a simple way. It implements functions from, timers, logic, PID controllers and in some cases SCADA networks. It is known as a graphically common way to program a system (PLC Academy, 2021).

FBDs breaks down the automation process into smaller units and hence simplifies the process, which makes it easier for the user to understand the function of the program. Each block contains a certain function and is shown as a box with input(s) and output(s). The output of one block can also be connected to another block. This is useful for when mathematical functions are required. Examples of blocks are 'AND' blocks where it gives an output if both the inputs are true. It also features timers and counters, just like ladder logic (Edrawsoft., 2023).

(See figures 7, 8 & 9 in Appendix A, page 27 for FBD functions).

Instruction List:

Like structured text, instruction lists are not a graphical language, hence they also take up less memory and have a better execution speed. Functions such as addition, subtraction, multiplication, and division can also be done with instruction lists. This is not a commonly used language however, due to users commonly preferring visual programming languages (Budimir, 2017).

Instruction lists support programming based on an accumulator. The operators are supported, along with multiple inputs and outputs, sets and resets of outputs and conditional and unconditional jumps. Values are loaded into the accumulator using the ladder diagram operator and the first operation is executed with the first parameter in the accumulator. An example of an instruction list can be seen in figure 10 in appendix A page 28 (Electric, 2019).

The most appropriate language to use in this assignment would be to use ladder logic, due to it being easy to interpret, especially with PID where there are a lot of functions and operations occurring simultaneously. With languages such as structured text or instruction lists, it would be more difficult for an engineer to understand the system and fault find if need be.

3 - Hardware Investigation and Input/Output List

The PLC itself is a Siemens S7-1214C DC/DC/DC. In this assignment, only simulated hardware was used. An HMI screen was developed but was only simulated. The PLC was also simulated and communicated with software called Factory IO, which simulated the tank being filled and emptied, whilst showing the movement of the valves. This also contained simulated inputs and outputs. The inputs and outputs that came as standard are shown below along with the tank:

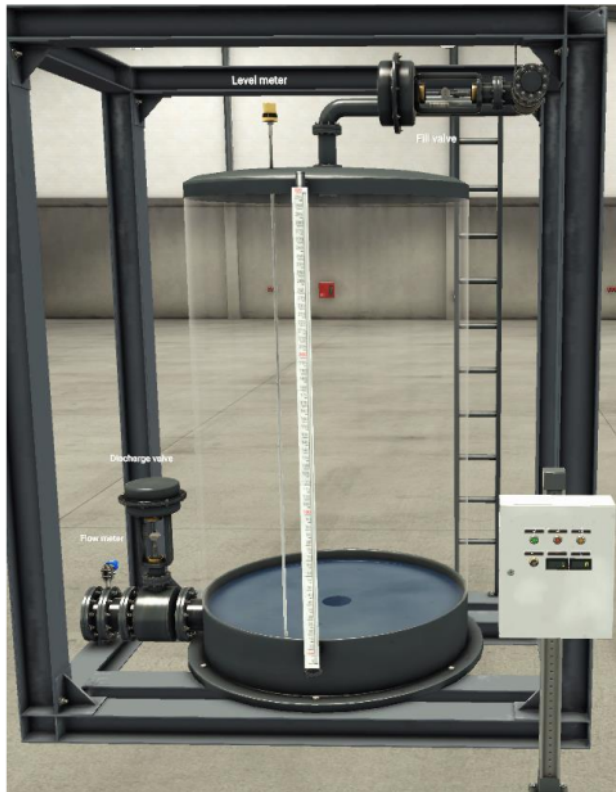


Figure 2 Factory IO Tank with Both Valves and Flow Meter

Start	%I0.0	%Q0.0	Start light
Reset	%I0.1	%Q0.1	Reset light
Stop	%I0.2	%Q0.2	Stop light
CTORY I/O (Running)	%I0.3	(REAL) %QD30	Fill valve
Level meter	%ID30 (REAL)	(REAL) %QD34	Discharge valve
Flow meter	%ID34 (REAL)	(DINT) %QD38	SP
Setpoint	%ID38 (REAL)	(DINT) %QD42	PV

Figure 1 Factory IO Inputs and Outputs

(See next page for I/O lists)

How would you configure the PLC modules to meet these needs?

The inputs and outputs used in the program are as follows:

Digital Input Addresses	Description
%I0.0	Start Button
%I0.1	Reset Button
%I0.2	Stop Button
%I0.3	Factory I/O Running

Digital Output Addresses	Description
%Q0.0	Start Light
%Q0.1	Reset Light
%Q0.2	Stop Light

Analogue Input Addresses	Description
%ID30	Level Meter AI Real
%ID34	Flow Meter AI Real
%ID38	Setpoint Potentiometer

Analogue Output Addresses	Description
%QD30	Fill Valve AQ Real
%QD34	Drain Valve AQ Real
%QD38	Setpoint Panel Meter
%QD42	Process Variable Panel Meter

4 – Program Design and Explanation

Main Program Block:

This contains the Factory I/O PLC driver, allowing the program to be simulated and tested. It also contains a first scan, upon program startup, it will automatically start in manual mode with set tuning parameters. There are start and stop rungs for starting manual control for both of the valves. A counter is also used to manually increment and decrement the fill valve position with a limit block in place to ensure its maximum and minimum value is not exceeded. A counter and limit block is also used for the drain valve in another network. Convert blocks are also used to convert integer values from the valves to real values. The auto sequence function call is also included in the main block (see appendix B, figures 11-16, pages 29-33)

Cyclic Interrupts:

Level – the level cyclic interrupt contains the PID controller, with the setpoint parameter, process value and control value. This is only enabled when auto or semi-auto mode are selected. The scaling is completed in this block for level by multiplying the raw value by 30. It runs when 'Level Control Run' is enabled and resets the controller when its disabled.

Flow – the flow cyclic interrupt also contains a PID controller, with the setpoint parameter, process value and control value. This is also only enabled when auto or semi-auto mode are selected. It runs when 'Flow Control Run' is enabled and resets the controller when its disabled.

(See appendix B, page 34, figures 17 & 18)

PID Control:

This block contains the parameters that are set inside the PID controller stated above where it calculates the error, P part, I part and D part, along with summing them all up to obtain a control signal (see appendix B, pages 35-36, figure 19 & 20).

Sequence:

This sequence was written in ladder code and contains 3 networks. See appendix B, page 37, figure 21 for full explanation of the sequence networks.

HMI Menu:

Manual Control – This enables the engineer to manually test the system. It contains flow and level control, each allowing to manually start, stop and open the valves to their desired position. There is also a level meter, which shows the current level of the tank and a display for the flow rate (see appendix B, page 38, figure 23).

Semi - Auto Control – This allows the engineer to automatically control the drain valve, whilst manually controlling the fill valve. It shows the setpoint, error, process value, control signal, tuning parameters, PID parts and a mimic graph. The tuning parameters can be altered to suit the system and can be monitored with the mimic graph. Another screen identical to this offers the exact same abilities, however, the fill valve is controlled automatically, and the drain valve is controlled manually (see appendix B pages, 38-39, figures 22 & 25).

Automatic Control – This gives fully automatic control for both of the valves for the operator. Setpoints for flow and level are inputted on the HMI screen and the program will automatically add the correct level to the tank and then the set rate of flow. There are stop and start buttons to start the process once the setpoints are set, and a mimic graph shows the setpoints and process values (see appendix B, page 39, figure 24).

A very well explained system

5 – System Testing

From the results in the test table in appendix C, all simulation tests have passed. The tank was simulated on Factory IO with the HMI screen used to apply controls and observe the relevant trends. Actual results were compared with the expected results, which determined if each test passed or failed (see appendix C, Table 1, figures 28-31, pages 42-46).

6 – Meeting the Customer Needs

The program was observed and navigated from a customer point of view and the design aspects were discussed with a group. Good feedback was given on HMI layout, such as clearly labelling interactive features and the ability for the engineer to tune the system. One improvement that was discussed was to add more resolution to the engineer screen number displays, so that the values they are dealing with can be monitored with a better resolution (see appendix D. page 47-48, figures 33 and 34). This may assist in allowing the engineer to better tune the system to their requirements. Maintenance and reliability of the system was also discussed. Adding network comments to each rung should give the engineer adequate assistance when using the program to fault find. These network comments and titles assist the engineer in understanding how the program functions and explains which components interact with each other. The program being written in ladder also allows for easier interpretation, due to being a graphical language. The HMI menu was discussed and from a customer's point of view, it was easy to navigate and allows for easy use when filling and draining the tank. It offered the engineer much flexibility in tuning and monitoring the system and gave the operator ease of use by not making the operator screen too overwhelming.

7 – Conclusions and Recommendations

Overall, the program managed to pass all tests effectively and all the aims were achieved. Manual control was achieved along with showing the status of all the devices. Semi-automatic control for both of the valves were achieved and gave the engineer flexibility for testing the setpoint, monitoring the error, process value, tuning the system etc. It was tuned correctly only with some overshoot on the fill valve. The drain valve reacted quickly, and no steady state error occurred during the tests. The start and stop functions worked correctly for both level control and flow control. The program was easy to navigate and network comments made it easier to understand how the program functions. The HMI screens provided an adequate amount of data along with trend graphs. A lot was learned on the requirements of the customer. An understanding was made on how ease of navigation and program understanding is vital to the user. Making the program as simple as possible whilst providing flexibility gives it reliability. Even for the operator when using the auto control HMI menu. Improvements still can be made to give the program more reliability. One being after a certain number of automatic cycles, to prevent the valves from being operated and add an indication of required maintenance on the valves. Valves can stick after many uses and will not perform properly. This will result in poor PID control. To prevent this, maintenance can be completed on the valves, then the engineer can reset the interlock. Another improvement that can be made is to add an 'Auto-Mode Ready' light. This is to give the operator certainty that the automatic cycle has finished and ready to restart, as currently, if the cycle is started again while still draining, it will start filling up the tank, but provide flow at the same time, when it is only supposed to start when the level has reached the setpoint. Another improvement can be to restrict starting the sequence again until the tank is fully drained.

Excellent work

8 – Team Working

There are various methods of creating a program that satisfies the criteria. Other people will have different ways of creating the program. Discussing these different methods offered improvements on reliability of the system and understanding the program. Discussing different programs allowed for identifying advantages and disadvantages. In engineering practices, teamwork is a requirement and there will always be different approaches to completing a task and working in a team will allow all ideas to be considered equally.

Who did you work with and how did you respect their rights

9 - Citation List

Correct method of listing.

Wide Range of Sources

Budimir, M., 2017. *What are Instruction Lists (ILs) for PLC programming?*. [Online]
Available at: <https://www.motioncontroltips.com/instruction-lists-ils-plc-programming/>
[Accessed 8th March 2023].

Devasia, A., 2022. *An Overview of Sequential Function Chart (SFC) PLC Programming*. [Online]
Available at: <https://control.com/technical-articles/an-overview-of-sequential-function-chart-sfc-programming/>
[Accessed 28th February 2023].

Dietrich, S., 2022. *Comparing Programming Languages: Structured Text vs. Ladder Logic*. [Online]
Available at: [https://control.com/technical-articles/comparing-programming-languages-structured-text-vs-ladder-logic/#:~:text=Structured%20text%20\(ST\)%20is%20a,or%20C%2B%2B%20computer%20programming%20language.](https://control.com/technical-articles/comparing-programming-languages-structured-text-vs-ladder-logic/#:~:text=Structured%20text%20(ST)%20is%20a,or%20C%2B%2B%20computer%20programming%20language.)
[Accessed 7th March 2023].

Edrawsoft., 2023. *What is a Functional Block Diagram?*. [Online]
Available at: <https://www.edrawsoft.com/article/what-is-functional-block-diagram.html>
[Accessed 7th March 2023].

Edrawsoft, 2023. *What Is Ladder Diagram*. [Online]
Available at: <https://www.edrawmax.com/article/what-is-ladder-diagram.html#:~:text=Application%20Scenarios%20of%20Ladder%20Diagrams,-Every%20system%20in&text=Ladder%20logic%20is%20used%20to,circuits%2C%20Ladder%20logic%20is%20useful.>
[Accessed 28th February 2023].

Electric, S., 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 8th March 2023].

Petrotech, 2017. *Why Is the IEC 61131 the Standard Programming Language in Open Architecture Control Systems?*. [Online]

Available at: <https://petrotechinc.com/why-is-the-iec-61131-the-standard-programming-language-in-open-architecture-control-systems/#:~:text=The%20Development%20of%20IEC%2061131%20and%20Its%20Programming%20Languages&text=The%20IEC%2061131's%20five%20programming,a%20cont>
[Accessed 28th February 2023].

PLC Academy, 2021. *Function Block Diagram (FBD) PLC Programming Tutorial for Beginners*. [Online]

Available at: <https://www.plcacademy.com/function-block-diagram-programming/>
[Accessed 7th March 2023].

RealPars B.V. , 2023. *Basics of Structured Text (ST) Programming*. [Online]

Available at: <https://realpars.com/structured-text/#:~:text=As%20one%20of%20the%20IEC,Text%20is%20just%20that%2C%20text!>
[Accessed 7th March 2023].

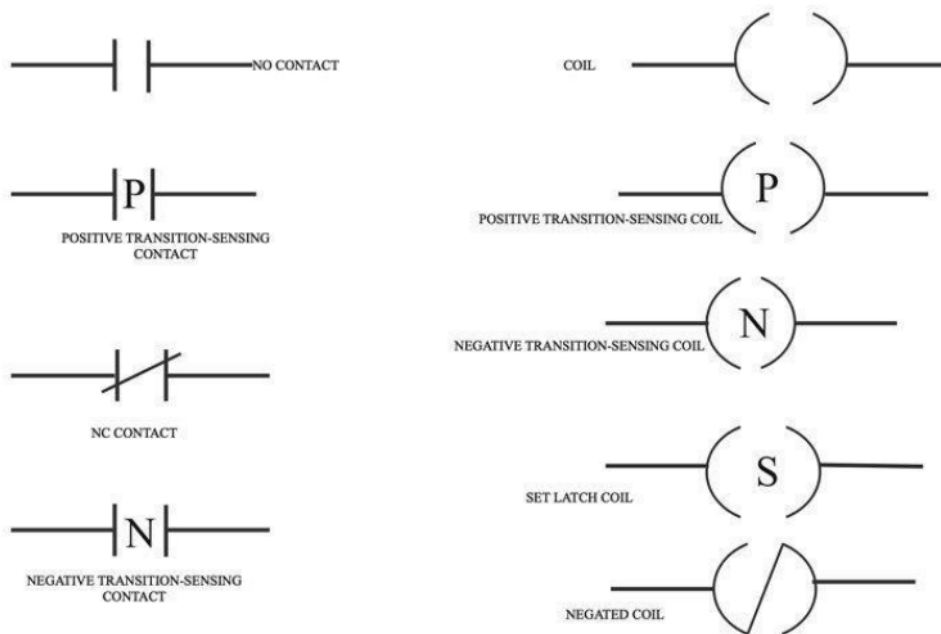
Techopedia Inc., 2023. *Ladder Logic*. [Online]

Available at: <https://www.techopedia.com/definition/20292/ladder-logic>
[Accessed 28th February 2023].

10 - Appendices

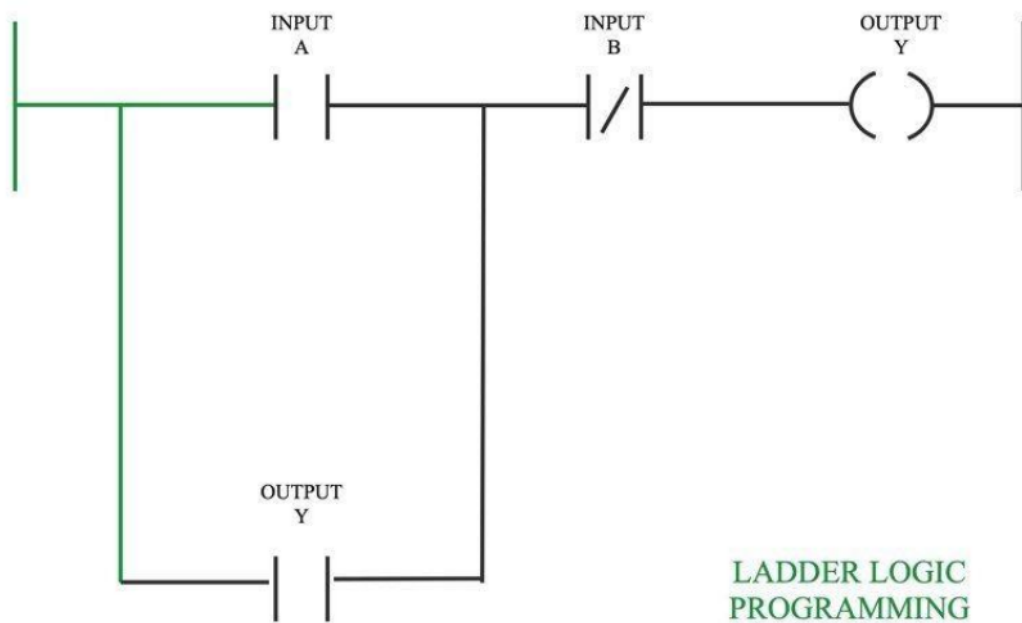
Appendix A: – Research

Ladder Logic:



LADDER DIAGRAM SYMBOL

Figure 3 Ladder Diagram Symbols (Edrawsoft, 2023)



LADDER LOGIC
PROGRAMMING
EXAMPLE

Figure 4 Ladder Programming Example (Edrawsoft, 2023)

Sequential Function Chart (SFC):

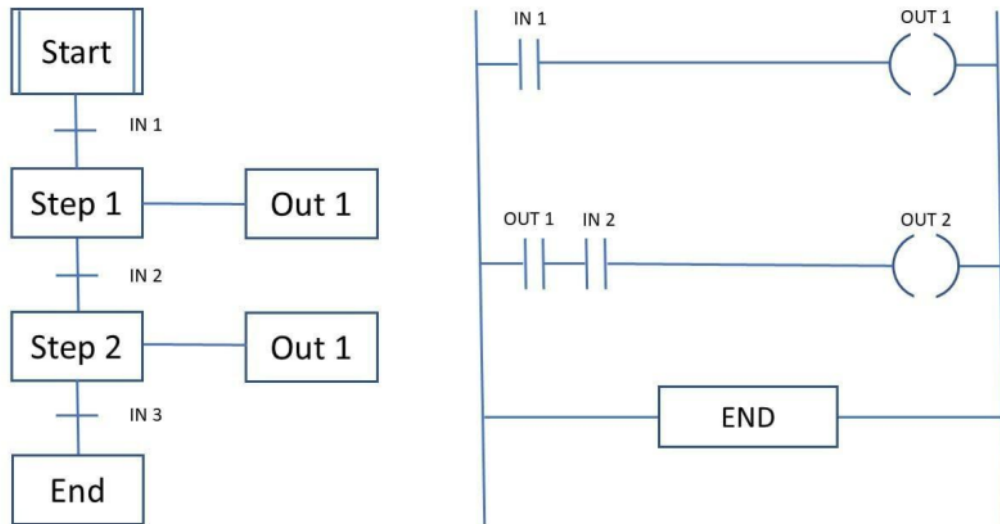


Figure 5 SFC With its Corresponding Ladder Diagram (Devasia, 2022)

Structured Text:

```
PROGRAM_CYCLIC  
  
    IF pbStart THEN  
  
        MotorON := TRUE;  
  
    ELSIF pbStop THEN  
  
        MotorON := FALSE;  
  
    END_IF
```

Figure 6 Structured Text Example (Dietrich, 2022)

Function Block Diagram (FBD):

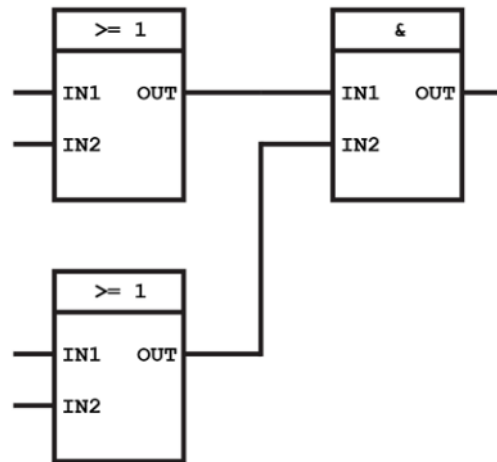


Figure 7 Using Multiple Inputs and Feeding an Output into An Input (Edrawsoft, 2023)

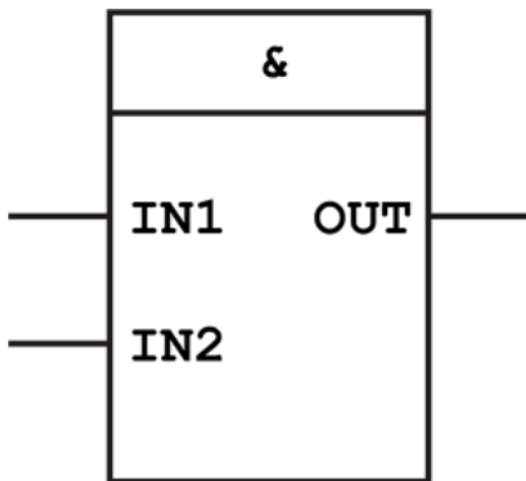


Figure 9 'AND' Block

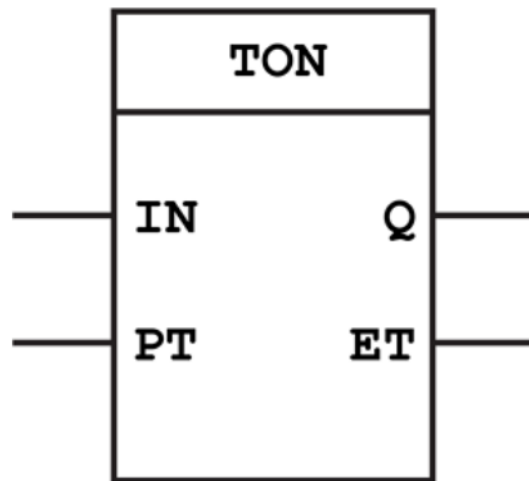


Figure 8 Timer Block

Instruction List:

```
LD          BVar1
ST          tonInst1.IN
CAL         tonInst1(
            PT:=t1,
            ET=>tOut2)
LD          toninst1.Q
JMPC        mark1
ST          tonInst2.IN
```

```
mark1:
LD          iVar2
ADD         230
```

Figure 10 Instruction List Program Example

Appendix B:– Program Design and Explanation

Main Program Block:

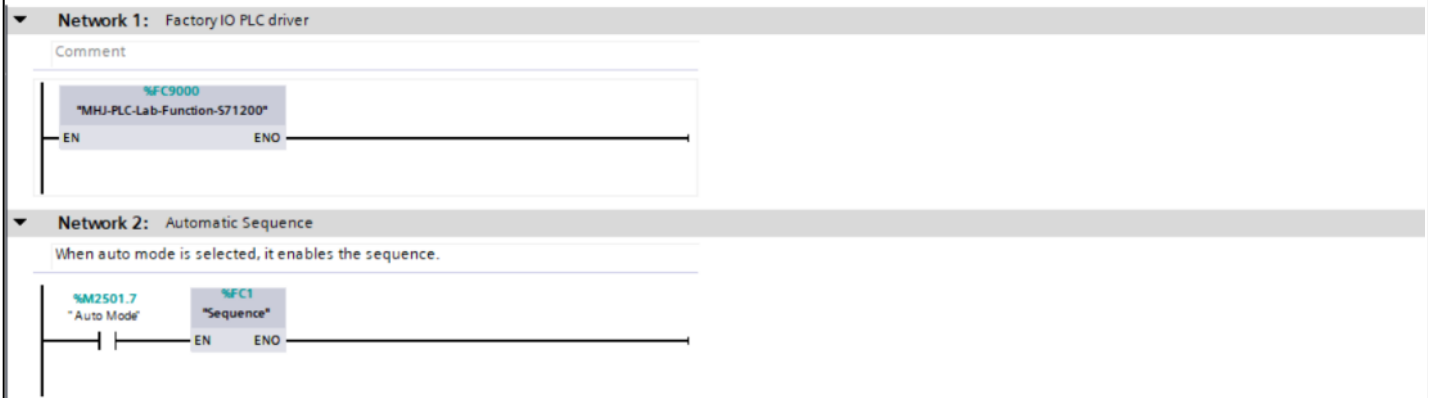


Figure 11 Networks 1&2

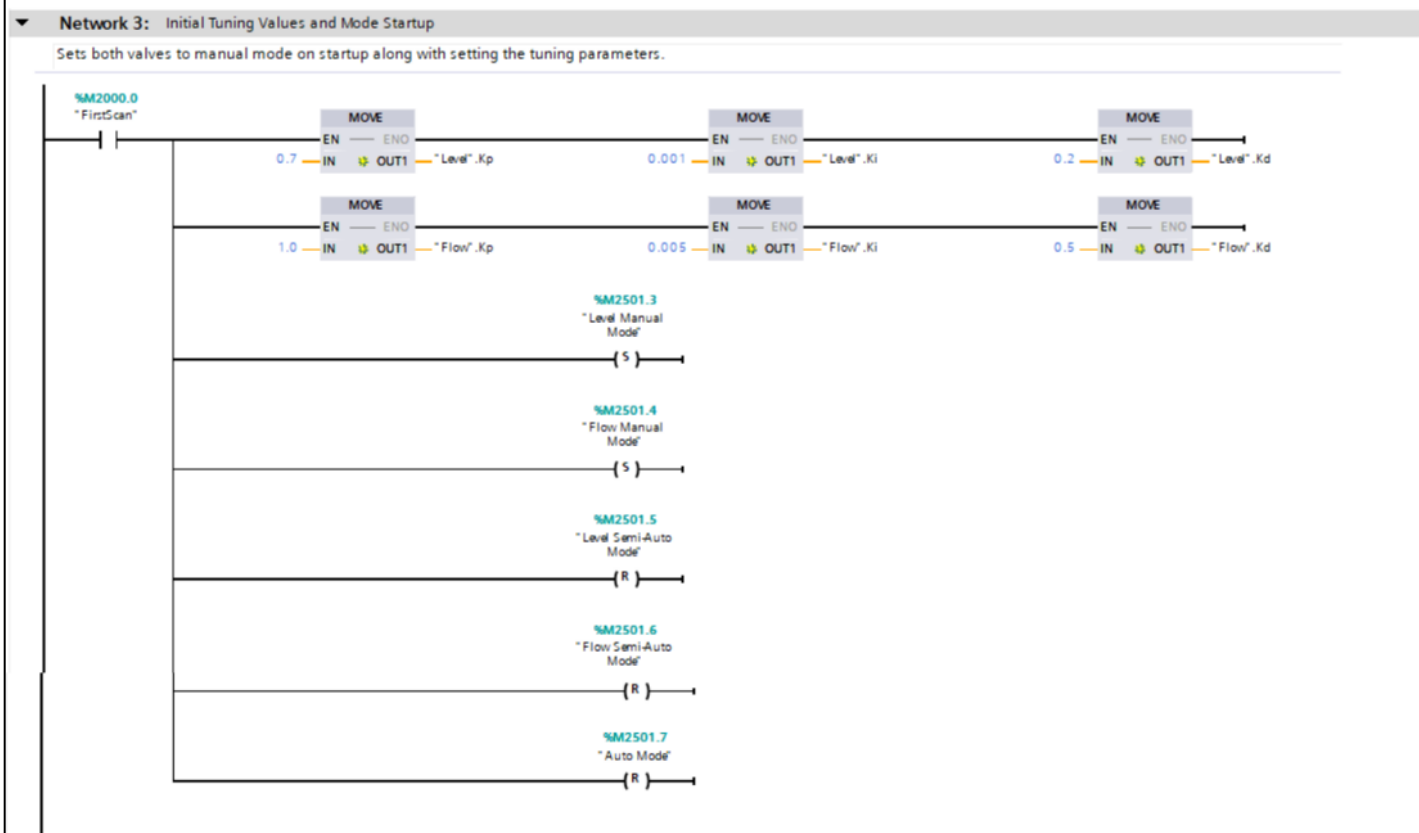


Figure 12 Network 3

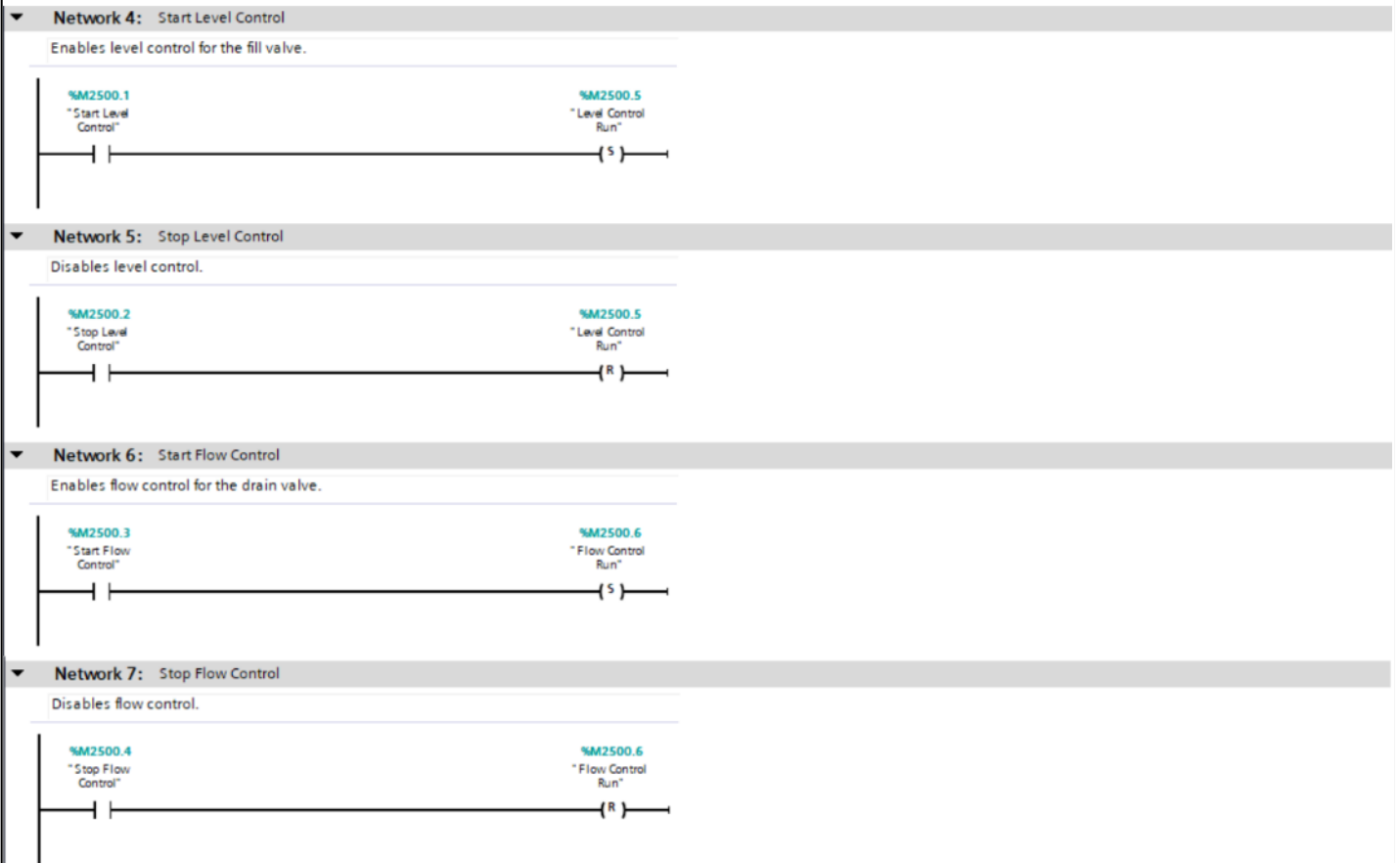


Figure 13 Networks 4, 5, 6 & 7

Network 8: Fill Valve Increment and Decrement

- The counter is used to increment and decrement the position of the fill valve in manual mode and 'Level Control Run' are enabled. This resets once 'Stop Level Control is Pressed'. The LIMIT block prevents the CV from going to high or too low.

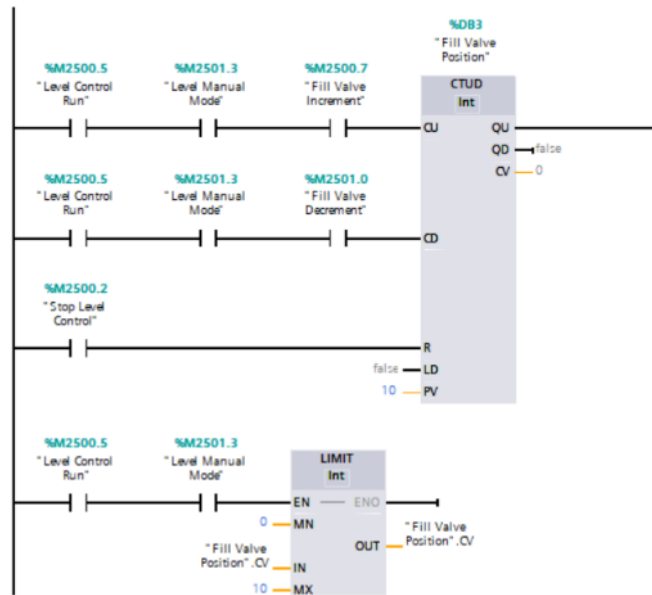


Figure 14 Network 8

Network 9: Drain Valve Increment and Decrement

- The counter is used to increment and decrement the position of the drain valve in manual mode and when 'Flow Control Run' is enabled. This resets once 'Stop Flow Control is Pressed'. The LIMIT block prevents the CV from going to high or too low.

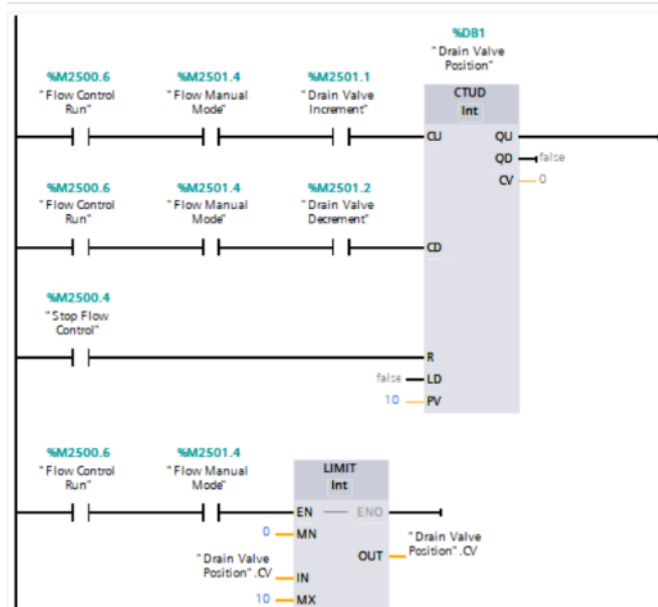


Figure 15 Network 9

Network 10: Convert Int to Real

- Converts the integer values from both the fill valve and drain valve into real values when level and flow control run are enabled and both manual modes are enabled.

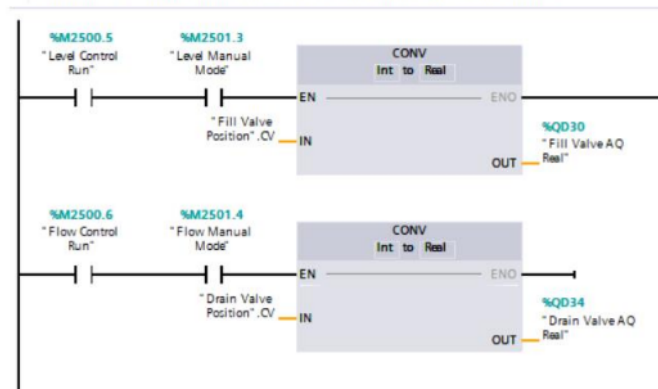
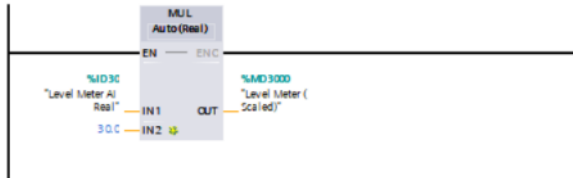


Figure 16 Network 10

Cyclic Interrupts:

Network 1: Level Meter Scaling

Scales the level meter by multiplying ID30 by 30.



Network 2: PID Control

PID control is selected once auto mode or semi-automatic mode is selected. The PID controller resets when 'Level Control Run' is disabled.

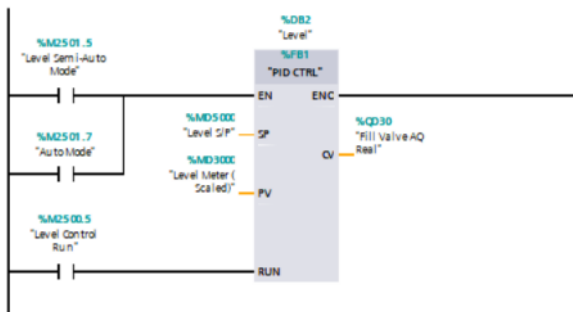


Figure 17 Networks 1&2 for Level Control

Network 1: PID Control

PID control is selected once auto mode or semi-automatic mode is selected. The PID controller resets when 'Flow Control Run' is disabled.

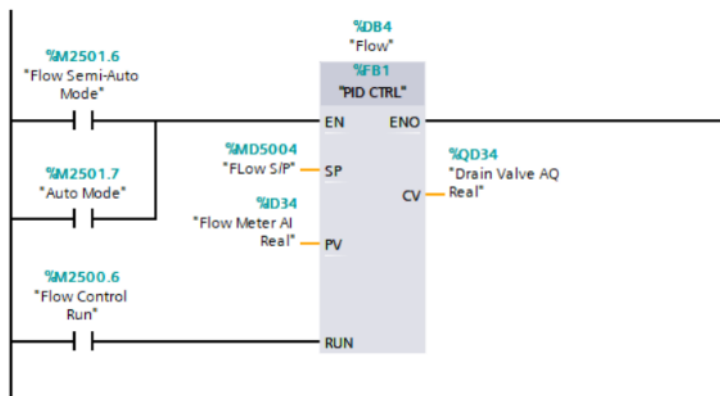
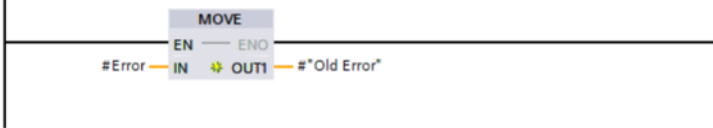


Figure 18 Networks 1&2 for Flow Control

PID Control:

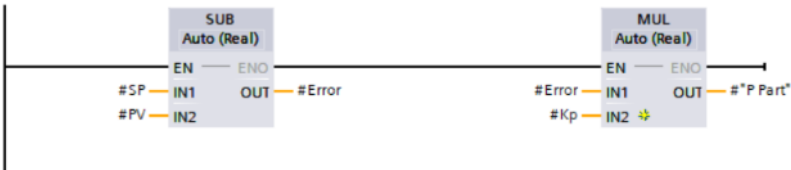
Network 1: Store old error

Comment



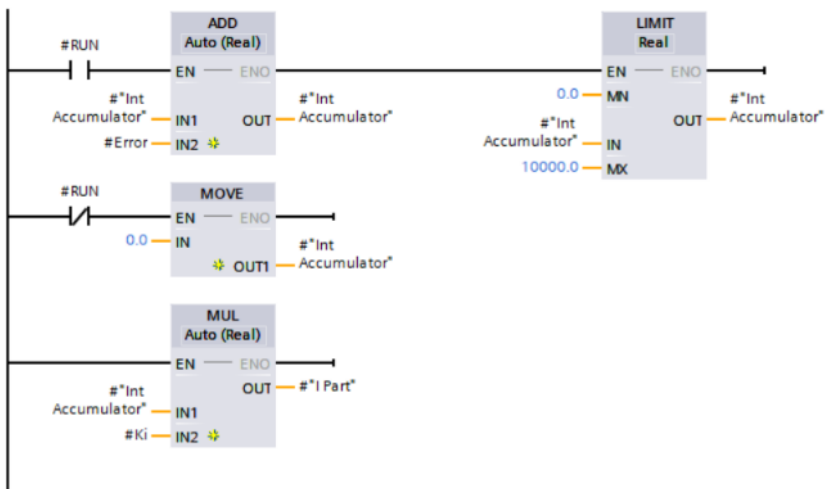
Network 2: Calculate error and P Part

Comment



Network 3: Add error to integral accumulator, limit it and multiply it to Ki to obtain the I part.

Integral accumulator is disabled when RUN is off.



Network 4: Calculate the difference between the old error and new error and multiply to get D part

Comment



Figure 19 Networks 1,2, 3 & 4

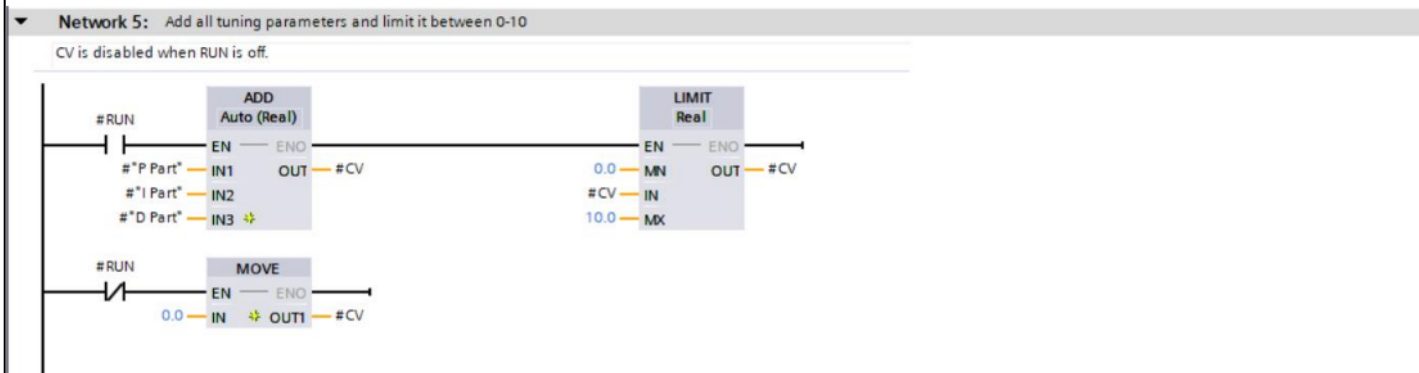


Figure 20 Network 5

Sequence:

Network 1: Start Filling The Tank

- 'Level Control Run' enables once the start button in the operator screen is pressed. This enables the PID controller in the Level Cyclic Interrupt [OB30] to run, which starts filling the tank to the setpoint level.



Network 2: Start Flow

- Once the level becomes greater than or equal to the setpoint, flow control will enable, which enables 'Flow Control Run' on the PID controller in the Flow Cyclic Interrupt [OB31]. The tank drain valve will now open to whichever state is required to produce the setpoint flow rate.



Network 3: Drain The Tank

- When the stop button on the operator screen is pressed, it resets the start button and level control stops. This resets the level PID controller and disables the integral accumulator and CV. It also overwrites the setpoint to 10, using a MOVE block to fully drain the tank. Once the level reaches less than 0.15, it will reset the flow control PID controller and disable the integral accumulator and CV. This also then stops the MOVE block from overwriting the setpoint to 10, so that the operator can set the setpoint on the next cycle.

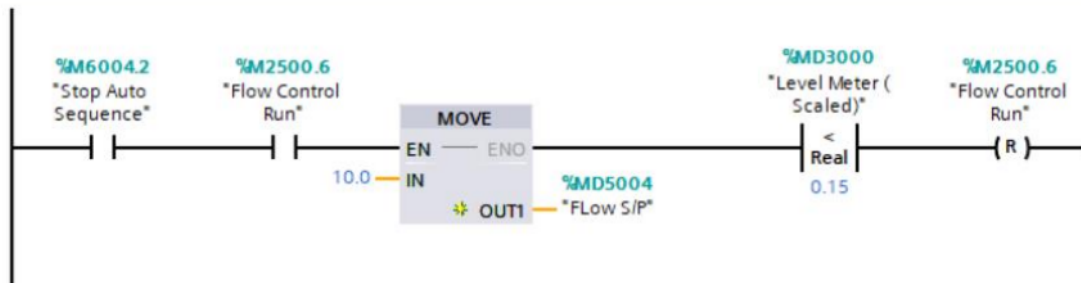


Figure 21 Networks 1,2 & 3

HMI Menu:

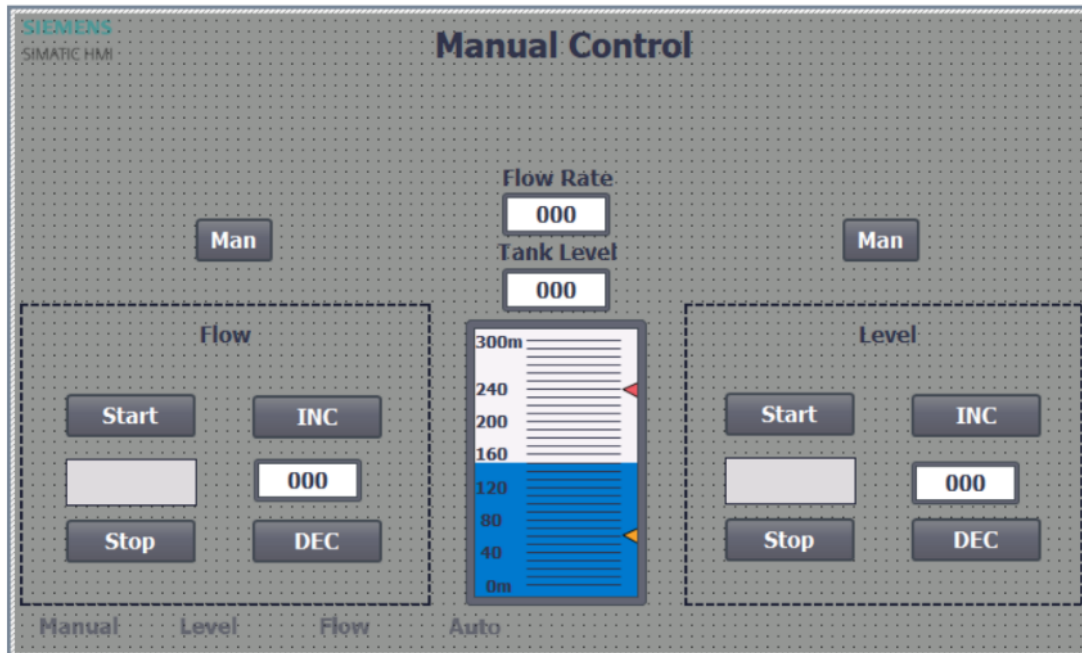


Figure 23 Manual Control Engineer Screen

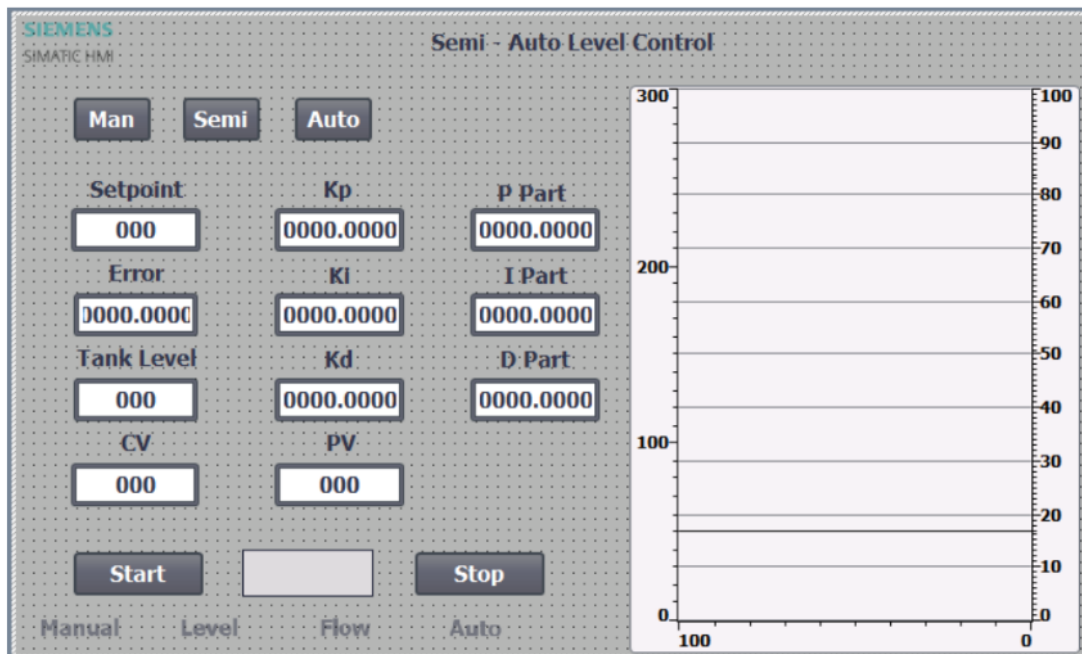


Figure 22 Semi – Automatic Level Control Engineer Screen

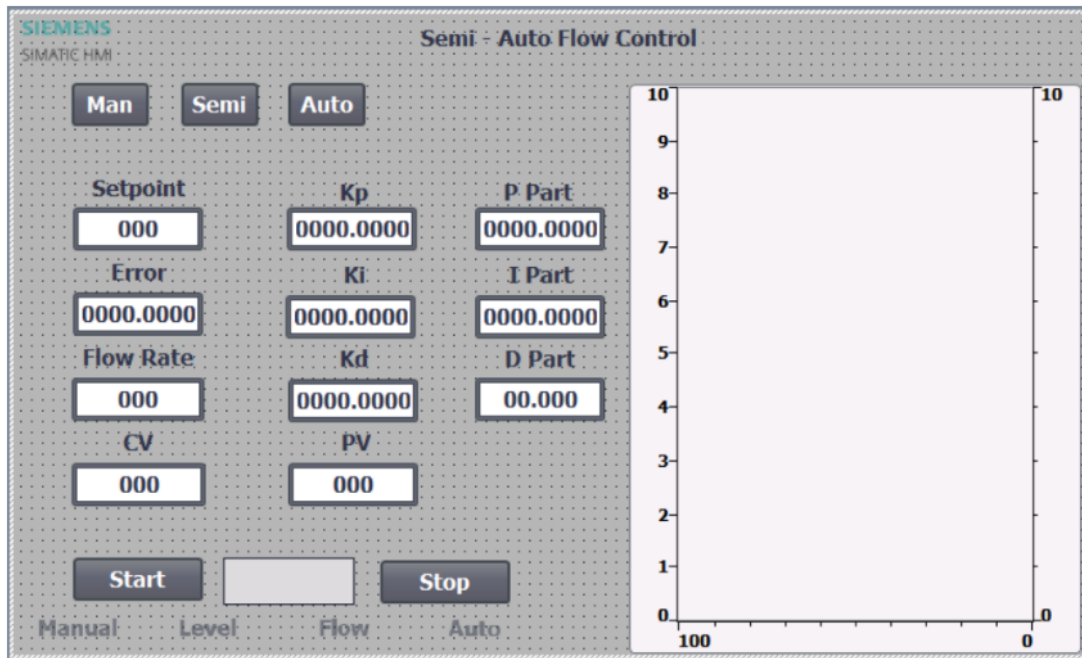


Figure 25 Semi - Automatic Flow Control Engineer Screen

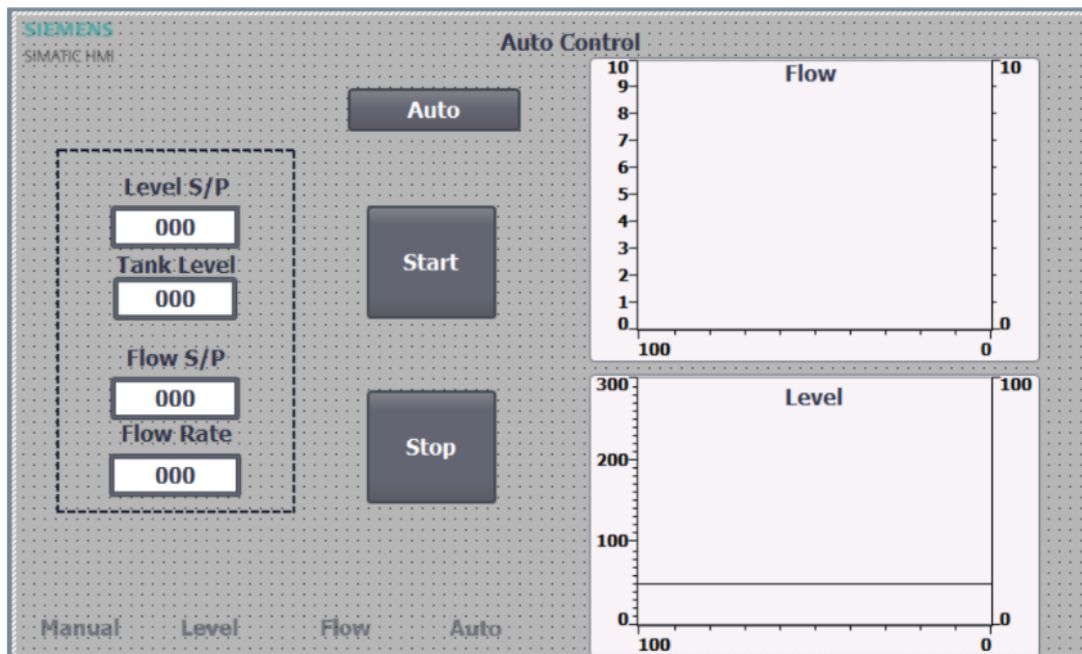


Figure 24 Auto Level and Flow Operator Control Screen

PLC Tag List:

PLC tags								
	Name	Tag table	Data type	Address ▲	Retain	Acces...	Writa...	Visibl...
1	Start Button	Factory IO Tags	Bool	%I0.0	☐	☒	☒	☒
2	Reset Button	Factory IO Tags	Bool	%I0.1	☐	☒	☒	☒
3	Stop Button	Factory IO Tags	Bool	%I0.2	☐	☒	☒	☒
4	Factory IO Running DI	Factory IO Tags	Bool	%I0.3	☐	☒	☒	☒
5	Level Meter AI Real	Factory IO Tags	Real	%ID30	☐	☒	☒	☒
6	Flow Meter AI Real	Factory IO Tags	Real	%ID34	☐	☒	☒	☒
7	Setpoint Potentiometer	Factory IO Tags	Real	%ID38	☐	☒	☒	☒
8	Start Light	Factory IO Tags	Bool	%Q0.0	☐	☒	☒	☒
9	Reset Light	Factory IO Tags	Bool	%Q0.1	☐	☒	☒	☒
10	Stop Light	Factory IO Tags	Bool	%Q0.2	☐	☒	☒	☒
11	Fill Valve AQ Real	Factory IO Tags	Real	%QD30	☐	☒	☒	☒
12	Drain Valve AQ Real	Factory IO Tags	Real	%QD34	☐	☒	☒	☒
13	Setpoint Panel Meter	Factory IO Tags	Dint	%QD38	☐	☒	☒	☒
14	Process Variable Panel Meter	Factory IO Tags	Dint	%QD42	☐	☒	☒	☒
15	System_Byte	System Tags	Byte	%MB2000	☐	☒	☒	☒
16	FirstScan	System Tags	Bool	%M2000.0	☐	☒	☒	☒
17	DiagStatusUpdate	System Tags	Bool	%M2000.1	☐	☒	☒	☒
18	AlwaysTRUE	System Tags	Bool	%M2000.2	☐	☒	☒	☒
19	AlwaysFALSE	System Tags	Bool	%M2000.3	☐	☒	☒	☒
20	Clock_Byte	System Tags	Byte	%MB2001	☐	☒	☒	☒
21	Clock_10Hz	System Tags	Bool	%M2001.0	☐	☒	☒	☒
22	Clock_5Hz	System Tags	Bool	%M2001.1	☐	☒	☒	☒
23	Clock_2.5Hz	System Tags	Bool	%M2001.2	☐	☒	☒	☒
24	Clock_2Hz	System Tags	Bool	%M2001.3	☐	☒	☒	☒
25	Clock_1.25Hz	System Tags	Bool	%M2001.4	☐	☒	☒	☒
26	Clock_1Hz	System Tags	Bool	%M2001.5	☐	☒	☒	☒
27	Clock_0.625Hz	System Tags	Bool	%M2001.6	☐	☒	☒	☒
28	Clock_0.5Hz	System Tags	Bool	%M2001.7	☐	☒	☒	☒
29	Start Level Control	My Tags	Bool	%M2500.1	☐	☒	☒	☒

Figure 26 PLC Tag List 1 - 29

30		Stop Level Control	My Tags	Bool	%M2500.2				
31		Start Flow Control	My Tags	Bool	%M2500.3				
32		Stop Flow Control	My Tags	Bool	%M2500.4				
33		Level Control Run	My Tags	Bool	%M2500.5				
34		Flow Control Run	My Tags	Bool	%M2500.6				
35		Fill Valve Increment	My Tags	Bool	%M2500.7				
36		Fill Valve Decrement	My Tags	Bool	%M2501.0				
37		Drain Valve Increment	My Tags	Bool	%M2501.1				
38		Drain Valve Decrement	My Tags	Bool	%M2501.2				
39		Level Manual Mode	My Tags	Bool	%M2501.3				
40		Flow Manual Mode	My Tags	Bool	%M2501.4				
41		Level Semi-Auto Mode	My Tags	Bool	%M2501.5				
42		Flow Semi-Auto Mode	My Tags	Bool	%M2501.6				
43		Auto Mode	My Tags	Bool	%M2501.7				
44		Level Meter (Scaled)	Factory IO Tags	Real	%MD3000				
45		Flow Meter (Scaled)	Factory IO Tags	Real	%MD3004				
46		Level S/P	My Tags	Real	%MD5000				
47		Flow S/P	My Tags	Real	%MD5004				
48		Start/Stop Semi-Auto Level Con...	My Tags	Bool	%M6000.0				
49		Start/Stop Semi-Auto Flow Con...	My Tags	Bool	%M6004.0				
50		Start Auto Sequence	My Tags	Bool	%M6004.1				
51		Stop Auto Sequence	My Tags	Bool	%M6004.2				
52		Auto Level Control Run	My Tags	Bool	%M6004.3				
53		Auto Flow Control Run	My Tags	Bool	%M6004.4				

Figure 27 PLC Tag List 30 - 53

Appendix C: – System Testing

Table 1 Simulation Test Table

Test Description	Expected Result	Actual Result	Pass or fail	Reference
1 - When in manual mode, all functions must be operable. Incrementing and decrementing both the valve positions must only be able to be done when manual mode is selected. When adjusting these, changes should be seen on the level meter and the tank level and flow rate should be correctly displayed.	The flow rate and tank level should show the correct values. Both valves should be able to alter the valve positions using the increment and decrement buttons. Both counters for the valve positions should not increment or decrement when not in manual mode.	Both valve positions were able to be adjusted using the increment and decrement buttons when in manual mode and when level control run, and flow control run were started. Once the stop button was pressed on each control the counter reset to 0 and the valves closed. The level meter and flow rate on the HMI were also correctly displayed.	Pass	Figure 28
2 – When in semi-auto level mode, the tank level should reach the setpoint level, given that the tuning values are adequate. The tank should also not fill if not in semi-auto level mode.	The mimic graph should display the setpoint and the process value. The process value should meet with the setpoint with minimal steady state error.	Semi-auto level mode was selected and also the setpoint was selected. When the start button was pressed, the fill valve opened, and the tank started to fill to the setpoint. There was a little overshoot, which concludes that better tuning parameters could have been introduced. This was tested with the drain valve slightly opened in manual mode in order to test if the level reached a steady state, which it did.	Pass	Figure 29
3 – When in semi-auto flow mode, the flow rate should be made equal to the setpoint, and the valve should not open if started and not in semi-auto flow mode.	The mimic graph should display the flow setpoint and process value. The process value should meet with the setpoint with minimal steady state error.	Semi-auto flow mode was selected and also the setpoint was selected. When the start button was pressed, the drain valve opened, and the tank started to drain at the set flow rate. There was no overshoot at all and no steady state error.	Pass	Figure 30

4 – When in auto mode, the tank should start filling to the desired setpoint after pressing the start button. Once the tank level reaches the setpoint, flow control should activate and flow at the setpoint flow rate. The tank should then fully drain and reset for the cycle to repeat once the stop button is pressed.	The level mimic graph should display the setpoint and process value. Once the process value meets the setpoint, flow control should start, and the drain valve should open and provide the set flow rate. The flow mimic graph should display the setpoint and the process value should meet the setpoint with minimal steady state error. The tank should then be running at a constant flow rate and level.	The level and flow setpoints were set before starting the system. Once the start button was pressed. The tank started filling until it reached the setpoint. There was some overshoot and eventually there was no error. At the same time, the flow valve opened, and the flow rate eventually met the setpoint without any overshoot or steady state error. When the stop button was pressed, the flow setpoint was set to 10 and the tank drained itself. Once finished, the setpoints were ready to be set and the tank was ready to be filled again.	Pass	Figure 31&32
---	--	---	-------------	-------------------------

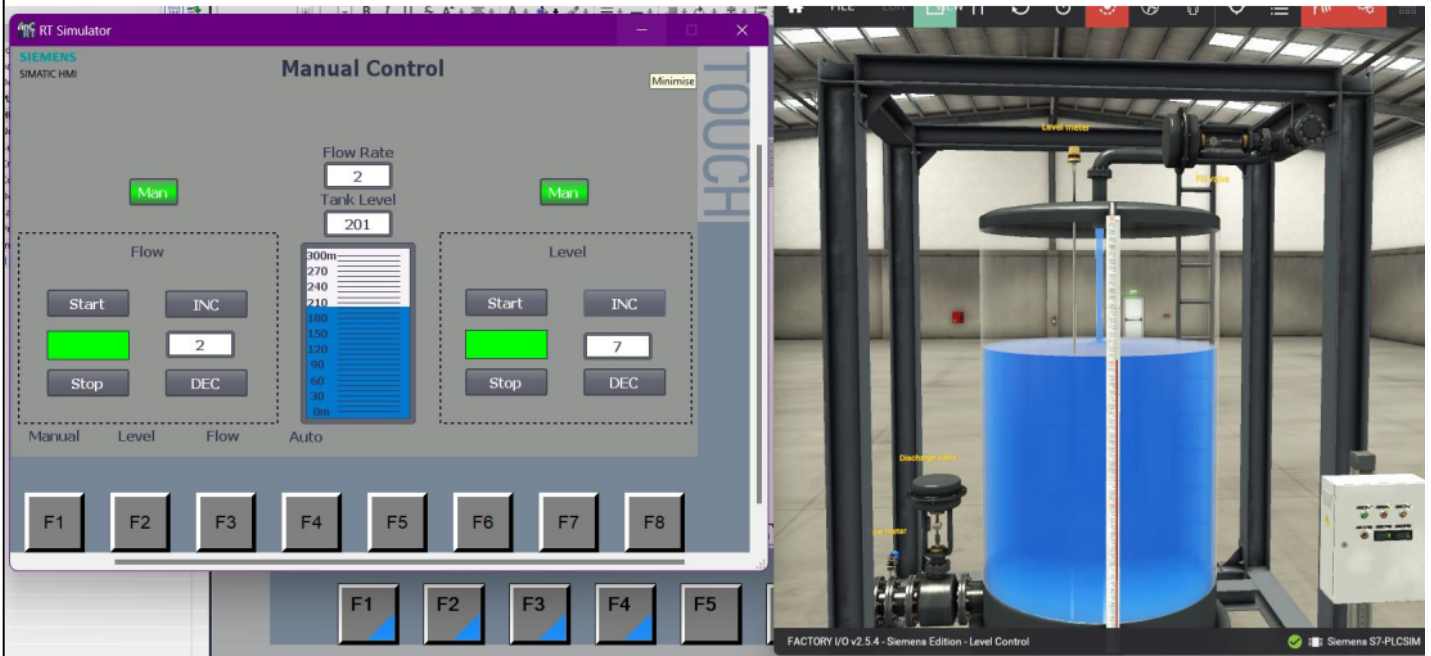


Figure 28 - Test 1 Manual Mode

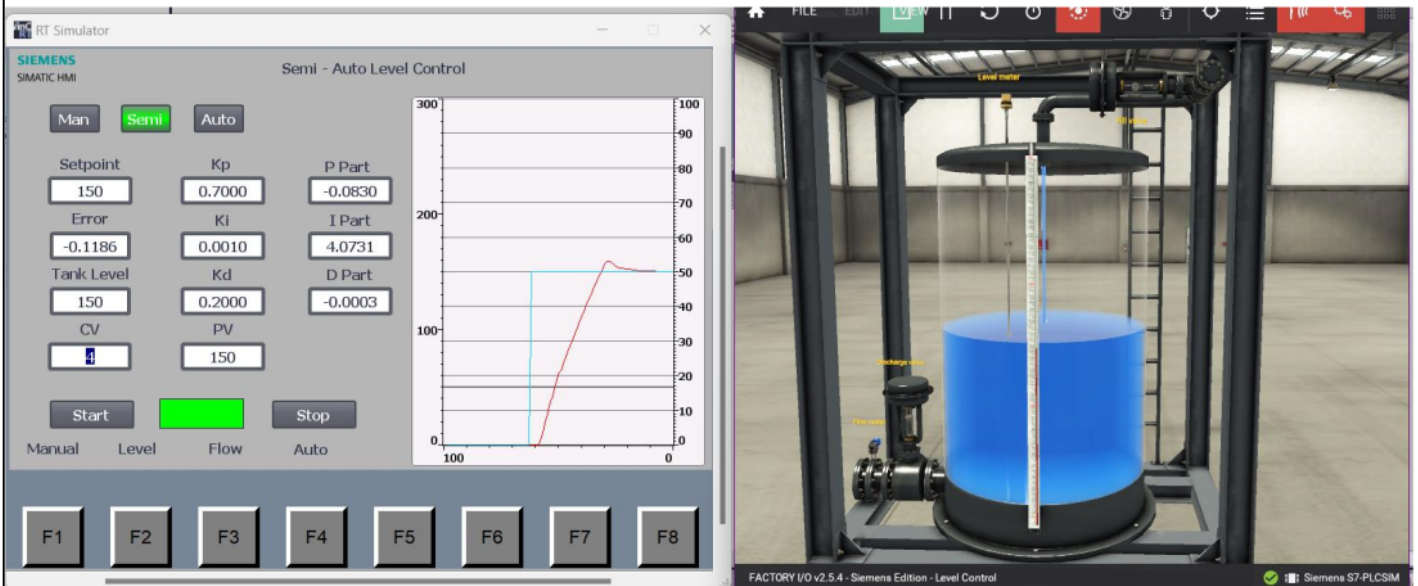


Figure 29 - Test 2 Semi-Auto Level Mode

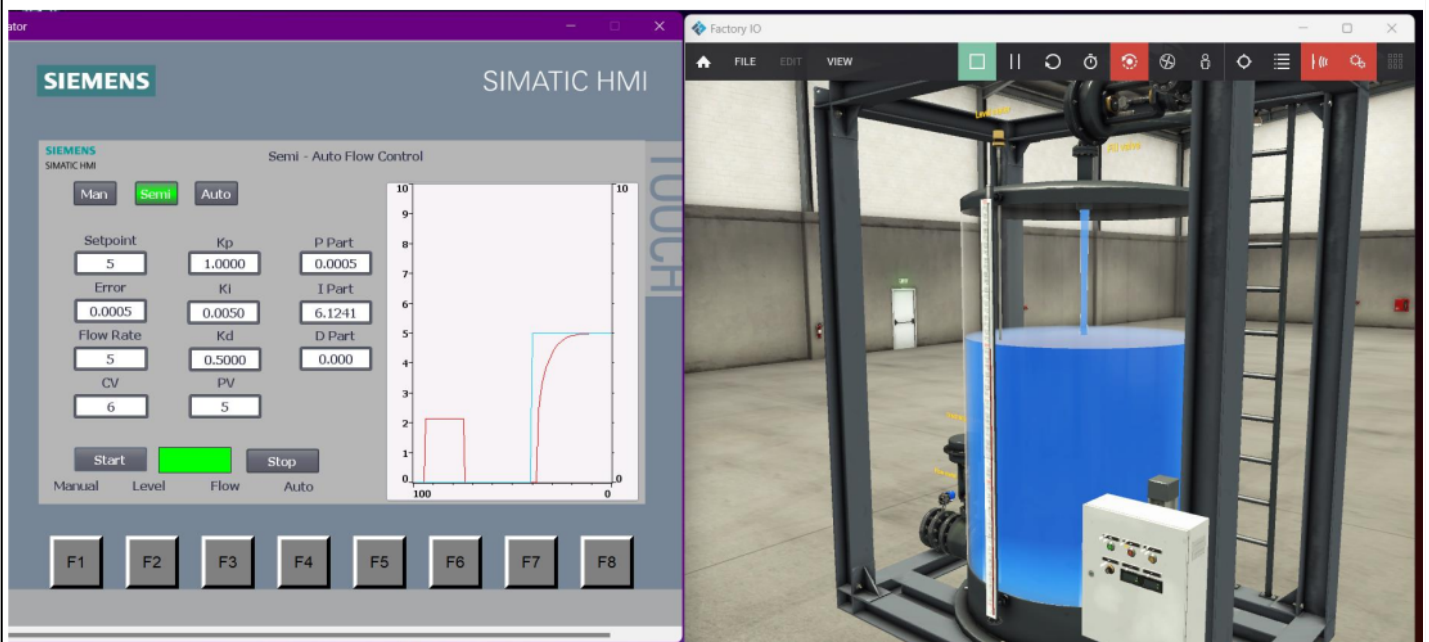


Figure 30 - Test 3 Semi-Auto Flow Mode

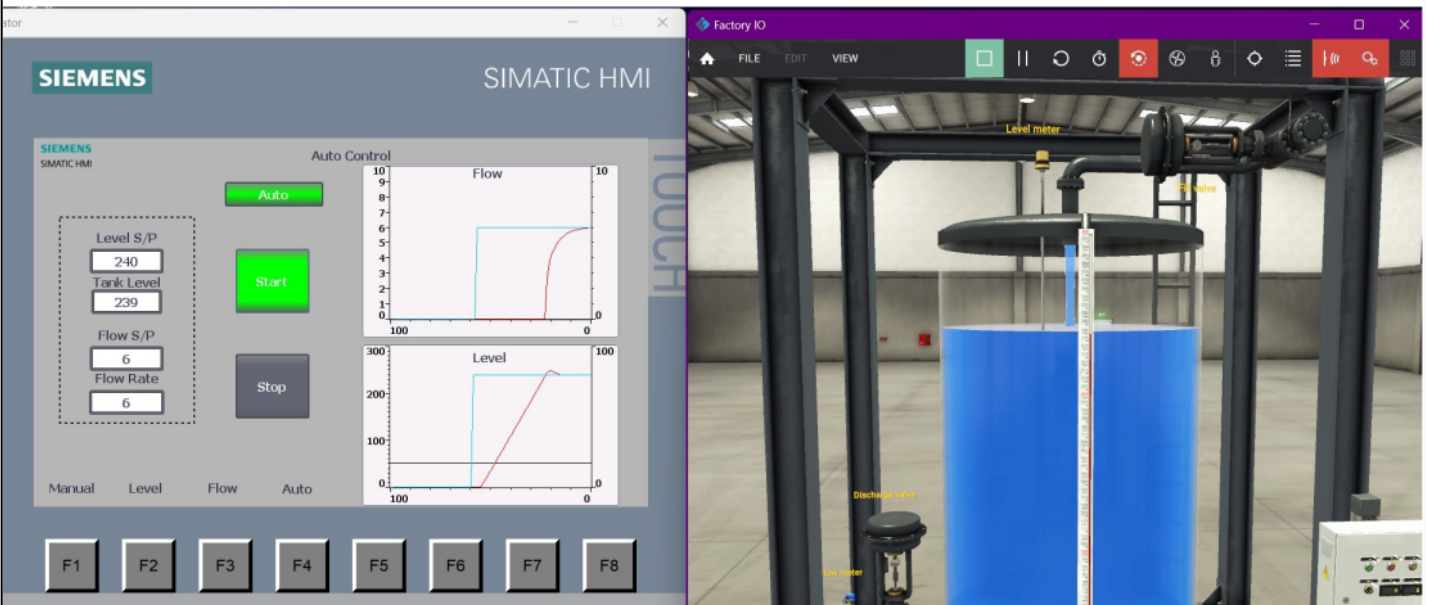


Figure 31 - Test 4 Auto Mode Setting Level and Flow Rate

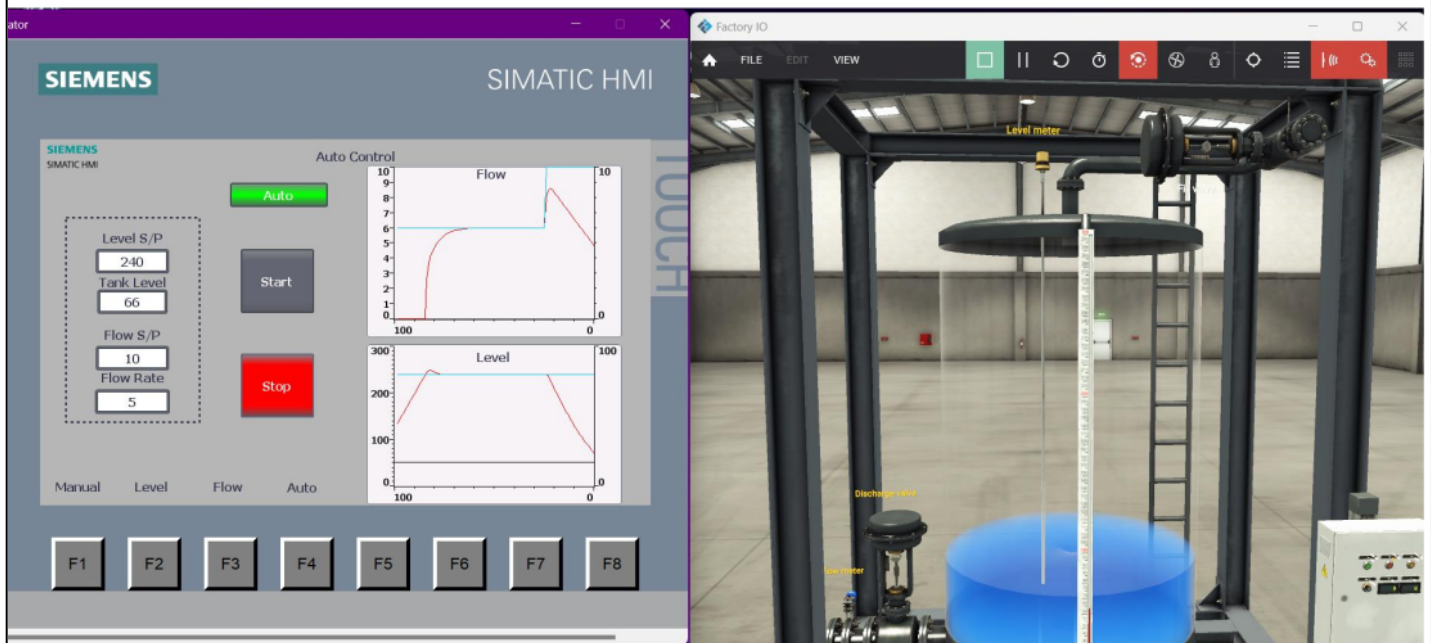


Figure 32 - Test 4 Draining the Tank

Appendix D: - Meeting the Customer Needs

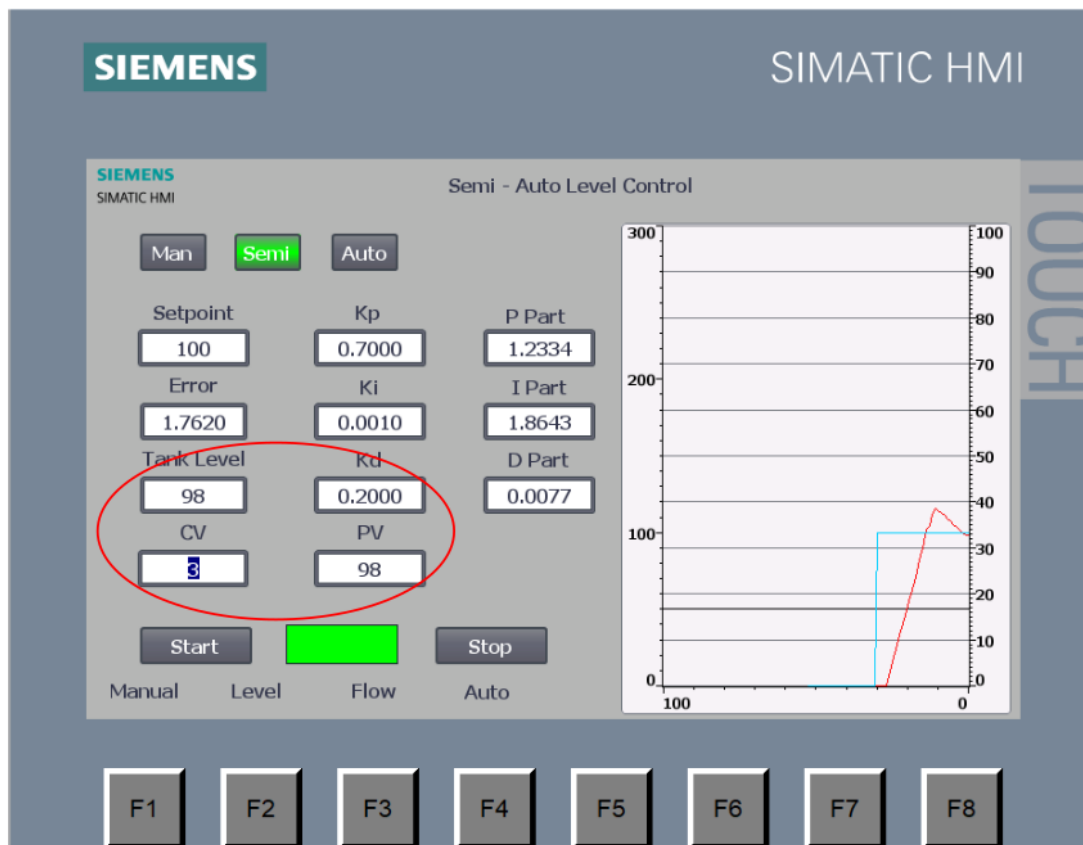


Figure 33 Before Altering Resolution

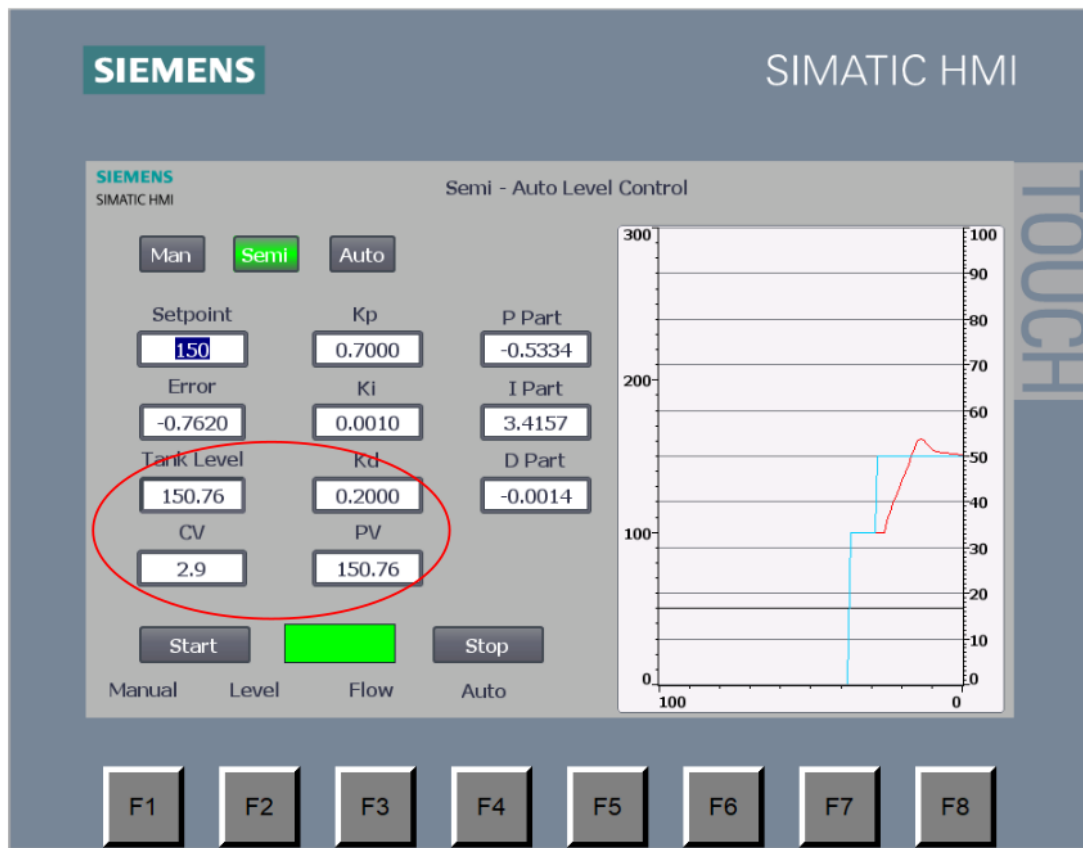


Figure 34 After Altering Resolution

FINAL GRADE

GENERAL COMMENTS

Instructor

84/100

PAGE 1

PAGE 2



Good Structure

Report is structured exactly as stated in the assignment brief.

PAGE 3

PAGE 4

PAGE 5

PAGE 6

Text Comment. Excellent work plan

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

Text Comment. How would you configure the PLC modules to meet these needs?

PAGE 14

PAGE 15

PAGE 16

Text Comment. A very well explained system

PAGE 17

PAGE 18

PAGE 19

Text Comment. Excellent work

PAGE 20

Text Comment. Who did you work with and how did you respect their rights

PAGE 21

QM

Correct method of listing.

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

QM

Wide Range of Sources

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

PAGE 22

PAGE 23

PAGE 24

PAGE 25

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

PAGE 48

GRADING FORM: NEW CSG-RUBRIC

MOHAMMED IBRAHIM

84

1. ENGAGEMENT WITH LITERATURE



An excellent research section with evidence of reading from a very wide range of sources cited and listed correctly using the Harvard method. Correct choice of language. 18/20

18

2. KNOWLEDGE AND UNDERSTANDING



Excellent knowledge and understanding demonstrated in your, IO list and program design and explanations. You could have improved your hardware design by showing the configuration of the PLC. 17/20

17

3. COGNITIVE AND INTELLECTUAL



Excellent simulation test results analysis, meeting customer needs, conclusions and recommendations. However, your testing table should have more entries. 15/20

15

4. PRACTICAL APPLICATION



Excellent program design and simulation testing methods with good presentation of results and meeting customer needs. 18/20

18

5. TRANSFERABLE SKILLS



An excellent work plan . Report is well structured, written and presented with clear diagrams. Word count not exceeded. Explanation of team working was not so good. 16/20

16