

1707739

by Liam Phillips

FILE	11714_LIAM_PHILLIPS_1707739_179196_1239641181.DOCX (7.82M)		
TIME SUBMITTED	24-MAY-2021 11:35PM (UTC+0100)	WORD COUNT	3034
SUBMISSION ID	154683004	CHARACTER COUNT	16462

Course: HNC Electrical and Electronic Engineering
Module: SGEE4004 Programmable Logic Controllers
Assignment: 2. Level and Flow Control System
Name: Liam Phillips
University number: 1707739
Date submitted: 24/05/21
Word count: 2364

Contents Good Structure

1. Research	4
1.1. Function block diagram (FBD)	4
1.2. Sequential function chart (SFC)	5
1.3. Structured text (ST)	5
1.4. Ladder diagrams (LAD)	6
1.5. Instruction list (IL)	7
2. Hardware Design	8
3. Program Design and Explanation	9
3.1. PLC Tags	9
3.2. Program blocks	9
3.3. HMI Screens	9
4. System Testing	10
5. Meeting the Customer Needs and Team Working	12
6. Conclusions and Recommendations	13
7. Works Cited	14
8. Appendix	15
8.1. Example of FBD	15
8.2. Example of SFC	15
8.3. Example of ST	16
8.4. Example of LAD	16
8.5. Example of IL	16
8.6.1. PLC drawing	17
8.6.2. Analogue Input Module drawing	18
8.7. Factory I/O addresses	19
8.8.1. Factory I/O Tags	20
8.8.2. System Operating Mode tags	20
8.8.3. Buttons for Fill Valve Level Control HMI screen tags	20
8.8.4. Displays for Fill Valve Level Control HMI screen tags	21
8.8.5. Buttons for Drain Valve Flow Control HMI screen tags	21
8.8.6. Displays for Drain Valve Flow Control HMI screen tags	21
8.9.1. Main Control Blocks	22
8.9.2. Level Control Blocks	24
8.9.3. Flow Control Blocks	32
8.10.1. HMI screen for fill valve level control screen	38

8.10.2.	HMI screen for drain valve flow control screen.....	38
8.11.	Filling in semi auto PID by tuning potentiometer	39
8.12.	In ON-OFF mode and choosing a deadband of 20	40
8.13.	Manually opening fill valve	41
8.14.	Manually closing fill valve	42
8.15.	Manually opening drain valve	43
8.16.	Manually closing drain valve	44

1. Research

PLCs (programmable logic controllers) have been the most widely used implementation medium for control algorithms in industrial applications for many decades. After their introduction, multiple companies developed PLCs with a various run-time environments, operating systems, and programming languages; thus, to minimise complexity for the users of PLCs, the International Electrotechnical Commission (IEC) implemented the standard IEC 61131.

This standard specifies five PLC programming languages. These are;

- Function block diagram (FBD)
- Sequential function chart (SFC)
- Structured text (ST)
- Ladder diagrams (LAD)
- Instruction list (IL)

(Palensky, 2009)

FBD, ST, and LAD are available on TIA portal, the software being used for the project. These languages will be used in the designing stage.

1.1. Function block diagram (FBD)

FBD is a language that is often used for process control systems as they allow for the implementation of continuous control loops. The language consists of a collection of blocks with input and output ports that form a network. Every block performs a function on the input data and outputs the results. Data, which can be of any rudimentary data type, is passed from one block to another via connections from output to input ports and can also be exchanged with sensors and actuators. (Gruner, 2012)

In principle, the blocks' functions are to perform continuously, allowing for the application of different equations, for example. In practice, the function blocks are executed sequentially at a given time rate.

An example of FBD is shown in appendices 8.1.

Advantages	Disadvantages
Being that it is graphical, it isn't necessary for system designers to have extensive training to understand the language.	FBDs are typically unable to integrate natively with IT systems, whereas text-based languages have that capability.
Debugging is easy as system designers can follow the wire connections between functions and function blocks.	Aren't ideal for solving sequencing problems due to their left to right continuous behaviour when executing.

Table 1

(Veeramani, 2007)

1.2. Sequential function chart (SFC)

Sequential function chart is a graphical language that displays the sequential behaviour of a program. One of the most significant features of SFC is that it displays the system's key states, as well as all potential state changes and the reasons for those specific changes.

Improper Citation

(Bruggink, 1999) States that a SFC consists of:

- Steps – These are depicted by rectangular boxes and each of these represents a particular state of the system being controlled. The beginning step is depicted by a rectangular box with vertical bars on each side.
- Transitions – These are the lines between the steps, forming the connection. A condition is assigned to each transition. When a condition following an active step becomes true, the next step is activated and the active step becomes deactivated.
- Actions – Each step is associated with some action, performed while the step is active. Actions can be described by any of the five languages defined by IEC.

An example of SFC is shown in appendices 8.2.

Advantages	Disadvantages
Allows for code to be divided into small parts, enabling users to create large, complex systems.	Doesn't always fit every application.
Programs can be monitored online, allowing the engineer to progress through the live system when troubleshooting.	Simple tasks can require a complex sequence.

Table 2

(Nilsson, 2001) (Dixon, 2018)

1.3. Structured text (ST)

Structured text is a programming language that is very similar to the Pascal programming language. Programs are made up of statements that are divided by semicolons. The statements use predefined statements and subroutines, which are specified values, internally stored values, or inputs and outputs. (Bolton, 2015)

Complex statements and nested instructions are supported:

- Iteration loops (REPEAT-UNTIL; WHILE-DO)
- Conditional execution (IF-THEN-ELSE; CASE)
- Functions (SQRT(), SIN())

(Roos, 2007) states that the elementary data types provided by the standard are the following:

- Integers
- Floating point numbers
- Data types that represent time
- Strings
- Bit strings

An example of ST is shown in appendices 8.3.

Advantages	Disadvantages
Similar to English vocabulary so it is simple to understand and learn.	High level languages have to be translated into machine language by translator which costs time.
It is a high-level language; therefore, it can be run on any computer.	When changes occur in data types, the corresponding change must be made to every location that acts on those data types within the program.

Table 3

(Shukla, 2017)

1.4. Ladder diagrams (LAD)

Ladder is a graphics-based programming language that resembles electrical schematic drawings. The diagrams consist of vertical lines named “rails”, and between them, are horizontal lines containing control logic called “rungs”. The logic itself closely mimics relay logic and that is a reason why they can be favoured over text-based programming languages. Complete diagrams closely resemble actual ladders, hence the name of the language. (Bolton, 2006)

An example of LAD is shown in appendices 8.4.

Advantages	Disadvantages
The rungs allow the system to be organised and easy to follow.	Being that it is so similar to relay logic, it is limited to programming more complex systems.
Having a background knowledge in relay logic limits the amount of training needed.	Building multi-rate applications is difficult due to the language operating in a left-to-right and top-to-bottom fashion; this costs time depending on how the PLC scans and operates.

Table 4

(Dixon, 2018) (Walter, 2007)

1.5. Instruction list (IL)

Instruction list is a low-level text-based programming language. They mostly resemble assembly language programming as the program is a series of instructions. Typical operations are mathematical such as adding, subtracting, multiplying, and dividing values. Other operations include jumping to a program label as well as returning from separate functions. (Budimir, 2017)

An example of IL is shown in appendices 8.5.

Advantages	Disadvantages
Valuable for applications that need code that is compact and time critical.	It has weak semantics and no code structure. This means if someone were to write code in an untidy manner, it would be difficult for others to read.
Low in memory compared to other languages.	Due to its lack of visual structure, it can be difficult to tell what the program does.

Table 5

(Visaya, 2019) (Dixon, 2018) (Budimir, 2017)

2. Hardware Design

The hardware of the machine consists of the tank, a control panel, and a modular PLC with two analogue modules. The control panel has a green light start button, a yellow light reset button, a red light stop button, a potentiometer, a setpoint meter, and a process variable meter.

I/O involving the tank includes a level meter, flow meter, a fill valve, discharge valve and a setpoint.

To simplify, this information is populated in tables below;

Digital inputs;

Address	Tag Name	Device Comment
%I0.0	Start	Pressing this will begin the sequence.
%I0.1	Reset	Pressing this will reset the sequence.
%I0.2	Stop	Pressing this will stop the sequence.
%I0.3	Factory I/O (Running)	This tells the PLC that the machine is on.

Table 6

Digital outputs;

Address	Tag Name	Device Comment
%Q0.0	Start light	When the start button is pressed, the start light turns on.
%Q0.1	Reset light	When the reset button is pressed, the reset light turns on.
%Q0.2	Stop light	When the stop button is pressed, the stop button turns on.

Table 7

Analogue inputs;

Address	Tag Name	Device Comment
%ID30	Level meter	This reads the level of the tank.
%ID34	Flow meter	This reads the flow of the tank.
%ID38	Setpoint	This is the potentiometer that when adjusted, controls the level of the tank.

Table 8

Analogue outputs;

Address	Tag Name	Device Comment
%QD30	Fill valve	This fills the tank at any given rate.
%QD34	Discharge valve	This drains the tank at any given rate.
%QD38	SP	This displays the setpoint value.
%QD42	PV	This displays the process variable value.

Table 9

The I/O of the PLC is put into the PLC tags section of the program, from there, they can be referenced within the program blocks. Check appendices 8.6.1. and 8.6.2. for electrical drawing of PLC. The factory I/O addresses can be found in appendices 8.7.

3. Program Design and Explanation

3.1. PLC Tags

Factory IO Tags are presented in appendices 8.8.1.

System Operating Mode tags are presented in appendices 8.8.2.

Buttons for Fill Valve Level Control HMI screen tags are presented in appendices 8.8.3.

Displays for Fill Valve Level Control HMI screen tags are presented in appendices 8.8.4.

Buttons for Drain Valve Flow Control HMI screen tags are presented in appendices 8.8.5.

Displays for Drain Valve Flow Control HMI screen tags are presented in 8.8.6.

3.2. Program blocks

Refer to appendices 8.9.1. for main control blocks.

Refer to appendices 8.9.2. for level control blocks.

Refer to appendices 8.9.3. for flow control blocks.

3.3. HMI Screens

The buttons of the HMI screens have the following operations:

- Auto – Start automatic sequence.
- Semi auto – Gives operator method controls i.e., PID and ON-OFF.
- Manual – Give operator manual controls i.e., open and close valves.
- Start – Commences filling or draining of tank depending on the screen.
- Stop – Stops filling or draining of tank depending on the screen.
- PID – Enables the operator to tune values when filling tank.
- ON-OFF – Allows the operator to choose a dead-band when filling tank.
- Open – Opens valve.
- Close – Closes valve.

The buttons located on the bottom corner of the screen allow the user to switch between both fill and drain control screens.

The graph shows a live line graph of the setpoint and process variable values; the setpoint being the tuned value, and the process variable being the tank level. The error display is the difference between the two. Both HMI screens can be found in appendices' 8.10.1. and 8.10.2.

4. System Testing

Table of results below;

Description of Test	Expected Results	Actual Results	Pass or Fail Decision	Reference to evidence in appendix
In PID control, the engineer can choose tuning values to fill tank.	By going into PID control, and adjusting the setpoint to 155 by using the potentiometer, the process variable should reach 155.	The setpoint displayed on the HMI read 154.706 and the process variable reached 155.619. This left an error of - 0.914.	Pass	Appendices 8.11.
In ON-OFF control, the engineer can choose dead-band.	In ON-OFF control, with the setpoint remaining at 154.706, the dead-band is set to a value of +20. The tank should fill to around 175.	The setpoint stayed the same and the process variable increased to 179.482. This was also visualised on the graph.	Pass	Appendices 8.12.
In semi auto, the engineer can choose tuning values to drain tank.	Pressing semi auto and adjusting the potentiometer to a lower value and then pressing start should cause the tank to drain to that level.	The setpoint could be adjusted and the semi auto control could be enabled, but on pressing start, the tank did not drain.	Fail	N/A
The engineer can manually open the fill valve.	When on the fill valve level control screen and going into manual mode, pressing the open	Pressing the open button did open the valve and the tank began to fill at a steady rate. This was	Pass	Appendices 8.13.

	button should open the valve to fill tank.	displayed on the graph.		
The engineer can manually close the fill valve.	When on the fill valve level control screen and going into manual mode, pressing the close button should close the valve to stop filling the tank.	Pressing the close button did close the fill valve thus stopping the water filling the tank. This was displayed on the graph, showing the line at a constant.	Pass	Appendices 8.14.
The engineer can manually open the drain valve.	When on the drain valve flow control screen and going into manual mode, pressing the open button should open the valve to drain tank.	Pressing the open button did open the drain valve and the tank began to drain at a steady rate. This was displayed on the graph.	Pass	Appendices 8.15.
The engineer can manually close the drain valve.	When on the drain valve flow control screen and going into manual mode, pressing the close button should close the valve to stop the draining of the tank.	Pressing the close button did close the drain valve thus stopping the water from draining. This was displayed on the graph, showing the line at a constant.	Pass	Appendices 8.16.

Table 10

5. Meeting the Customer Needs and Team Working

A program was presented to a group of individuals including myself to ensure that the project meets the customer needs. One suggestion that was highlighted, was to add labels to the axis of the graph, as well as legends that tell the user what line is what. This was valid point because the graph itself looked bare.

Another suggestion was to organise the screen in a more user-friendly setting. As it was, the displays were considerably close together and the buttons varied in size; this was quite an unappealing look.

Other than those two aesthetic based recommendations, the program worked swiftly in manual and semi-auto mode. However, it did lack the automatic running function.

6. Conclusions and Recommendations

The testing was performed using the simulation software 'Factory I/O' as an alternative to it being tested on a physical level and flow control system. The testing showed that the program met the criteria on some of the customer needs. Each test was performed several times to ensure that it was working correctly, eliminating any anomalies. One failure occurred when testing of the draining of the tank in semi-auto mode. Although this was something that wasn't resolved, a lot was still overcome through learning and researching into the PLC programming languages.

Ladder diagrams were used heavily throughout the project, therefore, this language of the five developed an understanding the most. There was an opportunity to use the language structured text for programming the automatic sequence, but this proved to be too difficult to do. The use of ladder diagrams wouldn't suffice for an automatic sequence as the language doesn't really operate sequentially.

Therefore, a future recommendation would be to implement a program that can run automatically in both filling and draining the tank by using structured text. Another recommendation could be to have a single screen controlling both the filling and draining of the tank, eliminating the process of switching between screens to control either or; this suggestion would be more efficient and less time consuming.

Correct method of listing.

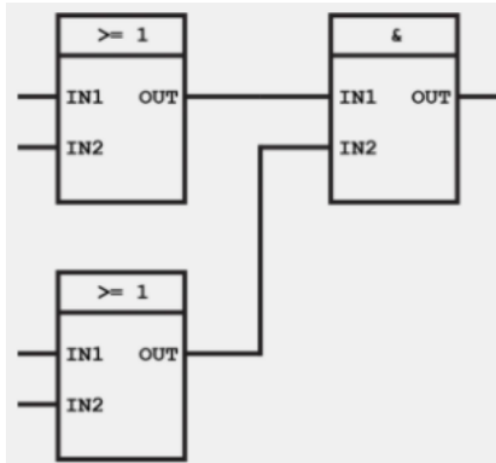
Good range of sources

7. Works Cited

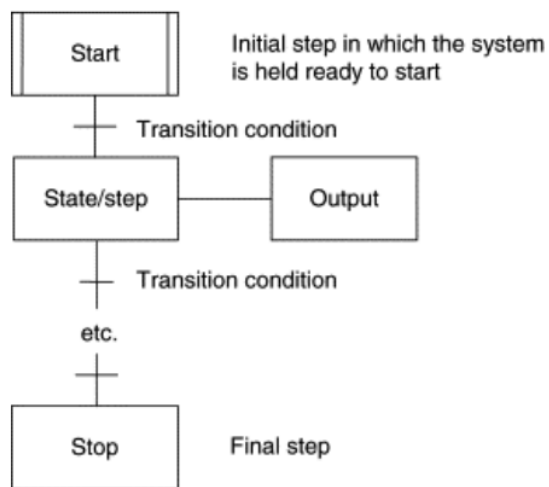
- Bolton, W., 2006. *Programmable Logic Controllers*. 4th ed. Burlington: Newnes.
- Bolton, W., 2015. IL, SFC, and ST Programming Methods. In: *Programmable Logic Controllers (Sixth Edition)*. s.l.:Newnes, p. 151.
- Bruggink, M., 1999. Programming PLCs using Sequential Function Chart. In: *Programming PLCs using Sequential Function Chart*. Nijmegen: University of Nijmegen, pp. 5-6.
- Budimir, M., 2017. *What are Instruction Lists (ILs) for PLC programming?*. [Online]
Available at: <https://www.motioncontroltips.com/instruction-lists-ils-plc-programming/#:~:text=Like%20any%20PLC%20programming%20language,to%20take%20up%20less%20memory.>
[Accessed 19 May 2021].
- Dixon, M., 2018. *What are the most popular PLC programming languages?*. [Online]
Available at: <https://realpars.com/plc-programming-languages/>
[Accessed 19 May 2021].
- Gruner, S., 2012. A Model-Based Implementation of Function Block Diagram. In: *Model-Based Development of Embedded Systems*. Munich: Tagungsband, p. 81.
- Nilsson, M., 2001. Sequential Function Chart. In: *Sequential Function Chart Interfacing*. Lund: Lund University, p. 15.
- Palensky, P., 2009. IEEE Industrial Electronics Magazine (Volume 3, Issue 4). In: *IEEE Industrial Electronics Magazine (Volume 3, Issue 4)*. s.l.:IEEE, pp. 27-31.
- Roos, N., 2007. *Programming PLCs using Structured Text*, Nijmegen: University of Nijmegen.
- Shukla, A., 2017. *Structured programming*. [Online]
Available at: <https://www.includehelp.com/data-structure-tutorial/structured-programming-its-advantages-and-disadvantages.aspx>
[Accessed 19 May 2021].
- Veeramani, A., 2007. *Function block diagrams*. [Online]
Available at: <https://www.controleng.com/articles/function-block-diagrams/>
[Accessed 19 May 2021].
- Visaya, 2019. *PLC programming languages - Function Block Diagram, Instruction List, and Ladder Diagram*. [Online]
Available at: <https://visaya.solutions/en/video/plc-programming-languages>
- Walter, T., 2007. *Control Engineering*. [Online]
Available at: <https://www.controleng.com/articles/ladder-logic-strengths-weaknesses/>
[Accessed 19 May 2021].
- World, L. L., 2021. *Ladder Logic Basics*. [Online]
Available at: <https://ladderlogicworld.com/ladder-logic-basics/>
[Accessed 15 May 2021].

8. Appendix

8.1. Example of FBD



8.2. Example of SFC



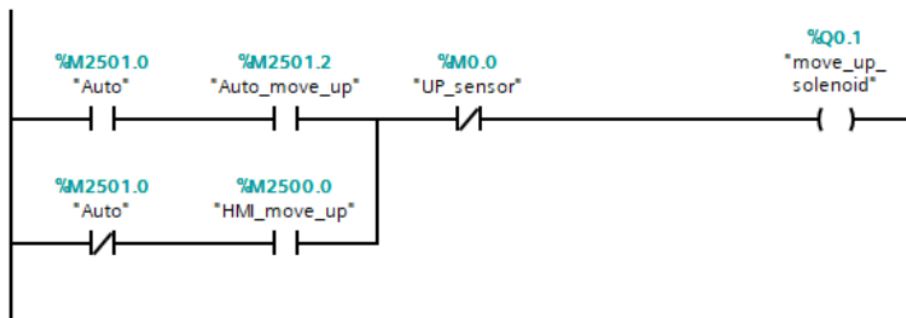
8.3. Example of ST

```

12 CASE "STEP" OF
13   0: //move up
14     "Auto_move_up" := 1;
15   IF "UP_sensor" = 1 THEN
16     "Auto_move_up" := 0;
17     "next_STEP" := 1;
18   END_IF;
19

```

8.4. Example of LAD



8.5. Example of IL

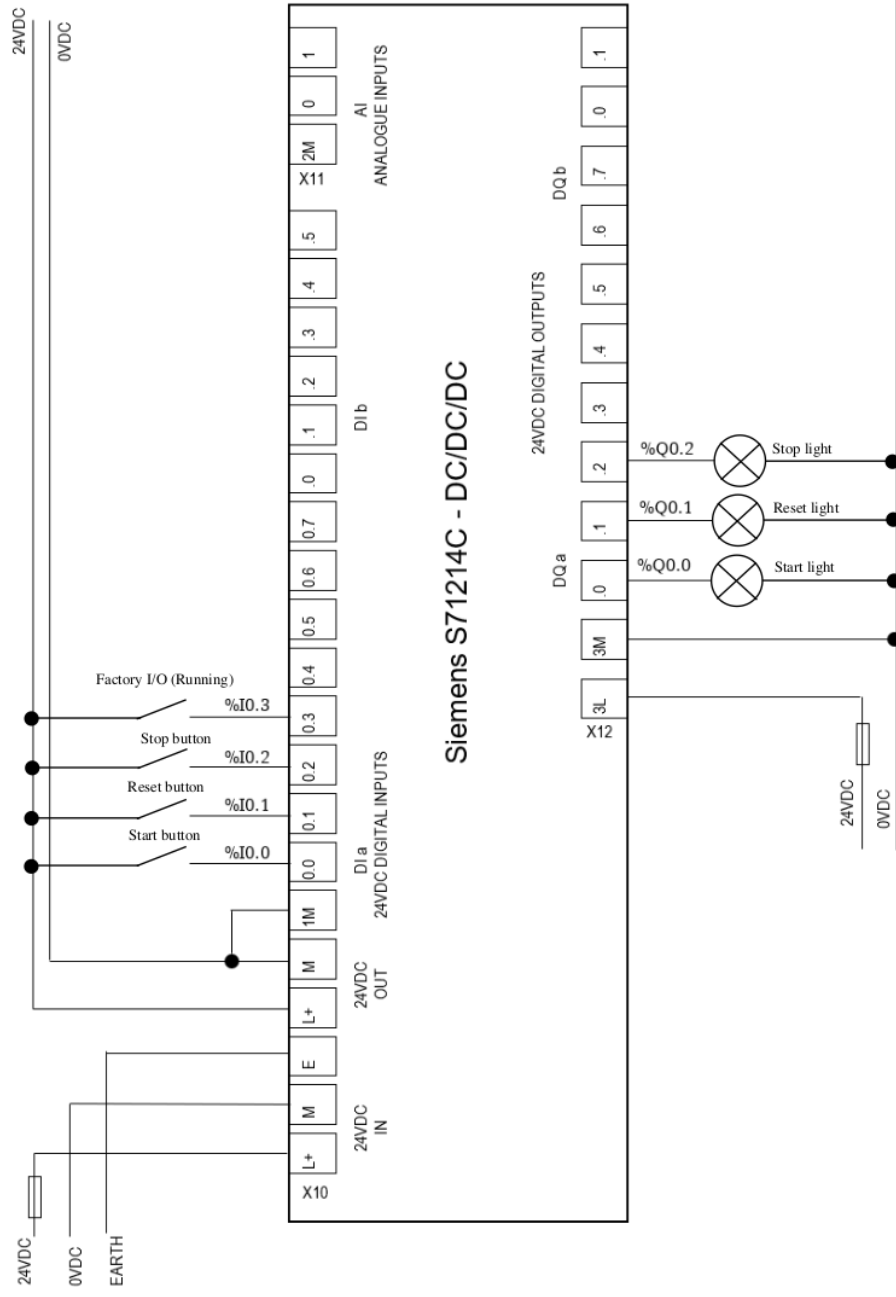
```

1.  LD  Start           ;Bit Instructions
2.  O   Flag
3.  A   Stop
4.  A   BLK
5.  =   Flag
6.  L   Time            ;Byte Instruction
7.  SD Timer          ;Byte-Bit Instr.
8.  LD Timer
9.  =   Start Up        ;Bit Instruction

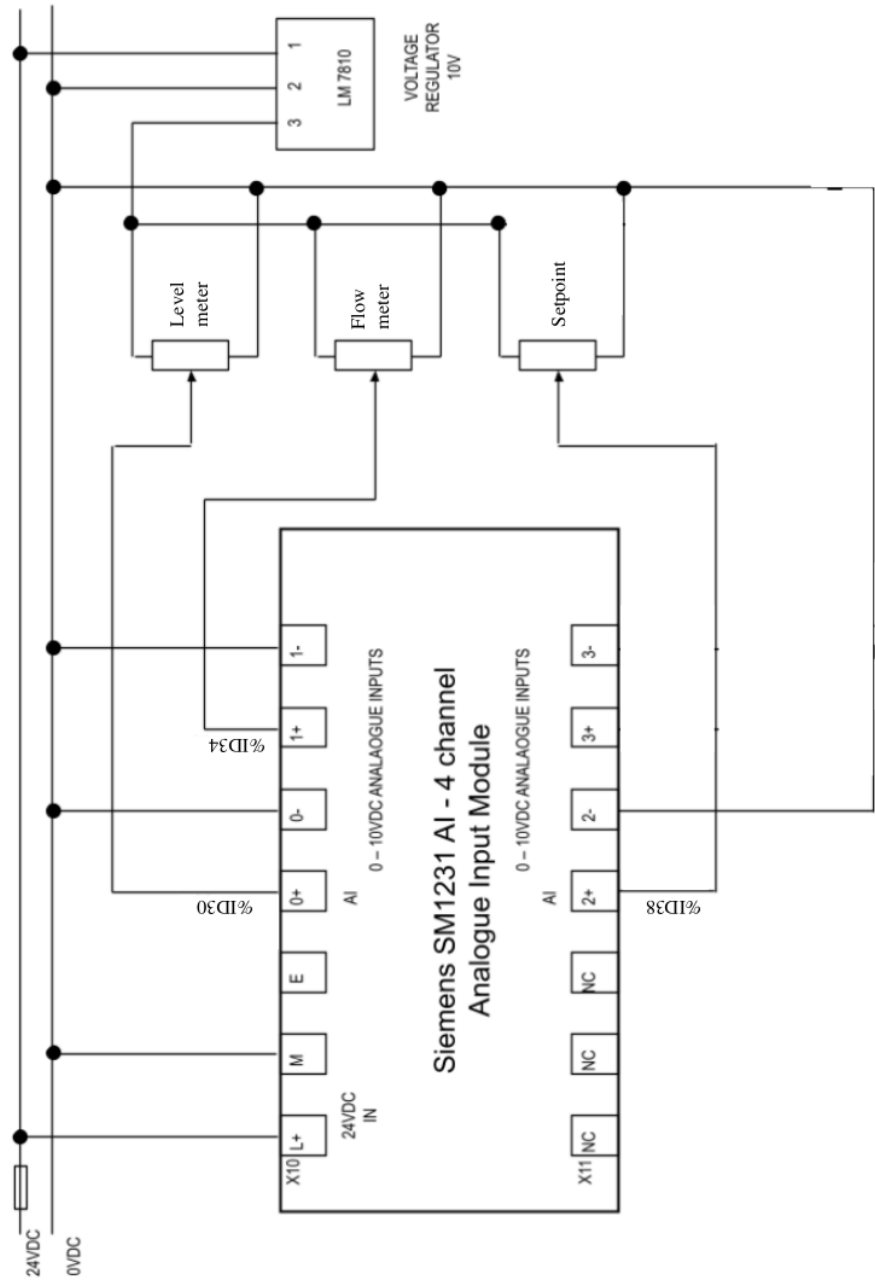
10. LD  Start Up        ;Bit Instruction
11. L   SP_cur          ;Byte Instructions
12. L   SP_nom
13. A>=I              ;Byte-Bit Instr.
14. =   Out             ;Bit Instructions

```

8.6.1.1. PLC drawing



8.6.2. Analogue Input Module drawing



analogue output module
is missing

8.7. Factory I/O addresses

Start	%I0.0	%Q0.0	Start light
Reset	%I0.1	%Q0.1	Reset light
Stop	%I0.2	%Q0.2	Stop light
FACTORY 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

8.8.1. Factory I/O Tags

		Name	Data type	Address
1		Start_Button	Bool	%I0.0
2		Reset_Button	Bool	%I0.1
3		Stop_Button	Bool	%I0.2
4		Factory_IO_Running_DI	Bool	%I0.3
5		Level_Meter_AI_Real	Real	%ID30
6		Flow_Meter_AI_Real	Real	%ID34
7		Fill_Valve_AQ_Real	Real	%QD30
8		Drain_Valve_AQ_Real	Real	%QD34
9		Setpoint_Potentiometer	Real	%ID38
10		Start_Light	Bool	%Q0.0
11		Reset_Light	Bool	%Q0.1
12		Stop_Light	Bool	%Q0.2
13		Setpoint_Panel_Meter	DInt	%QD38
14		Process_Variable_Panel_Meter	DInt	%QD42

8.8.2. System Operating Mode tags

15		Auto_Mode	Bool	%M2550.0
16		Semi_Auto_Mode	Bool	%M2550.1
17		Manual_Mode	Bool	%M2550.2

8.8.3. Buttons for Fill Valve Level Control HMI screen tags

9		HMI_Start_Level_Control	Bool	%M2500.4
10		HMI_Stop_Level_Control	Bool	%M2500.5
11		HMI_Manual_Open_Fill_Valve	Bool	%M2500.6
12		HMI_Manual_Close_Fill_Valve	Bool	%M2500.7
13		HMI_Manual_Open_Fill_Valve_...	Bool	%M2501.0
14		HMI_Manual_Close_Fill_Valve_...	Bool	%M2501.1

8.8.4. Displays for Fill Valve Level Control HMI screen tags

1		Level_Control_RUN	Bool	%M2500.0
2		Level_ON-OFF_Control	Bool	%M2500.2
3		Level_Setpoint	Real	%MD3000
4		Level_Process_Variable	Real	%MD3004
5		Level_Error	Real	%MD3008
6		Level_Kp	Real	%MD3012
7		Level_CV	Real	%MD3016
8		Level_PID_Control	Bool	%M2500.1

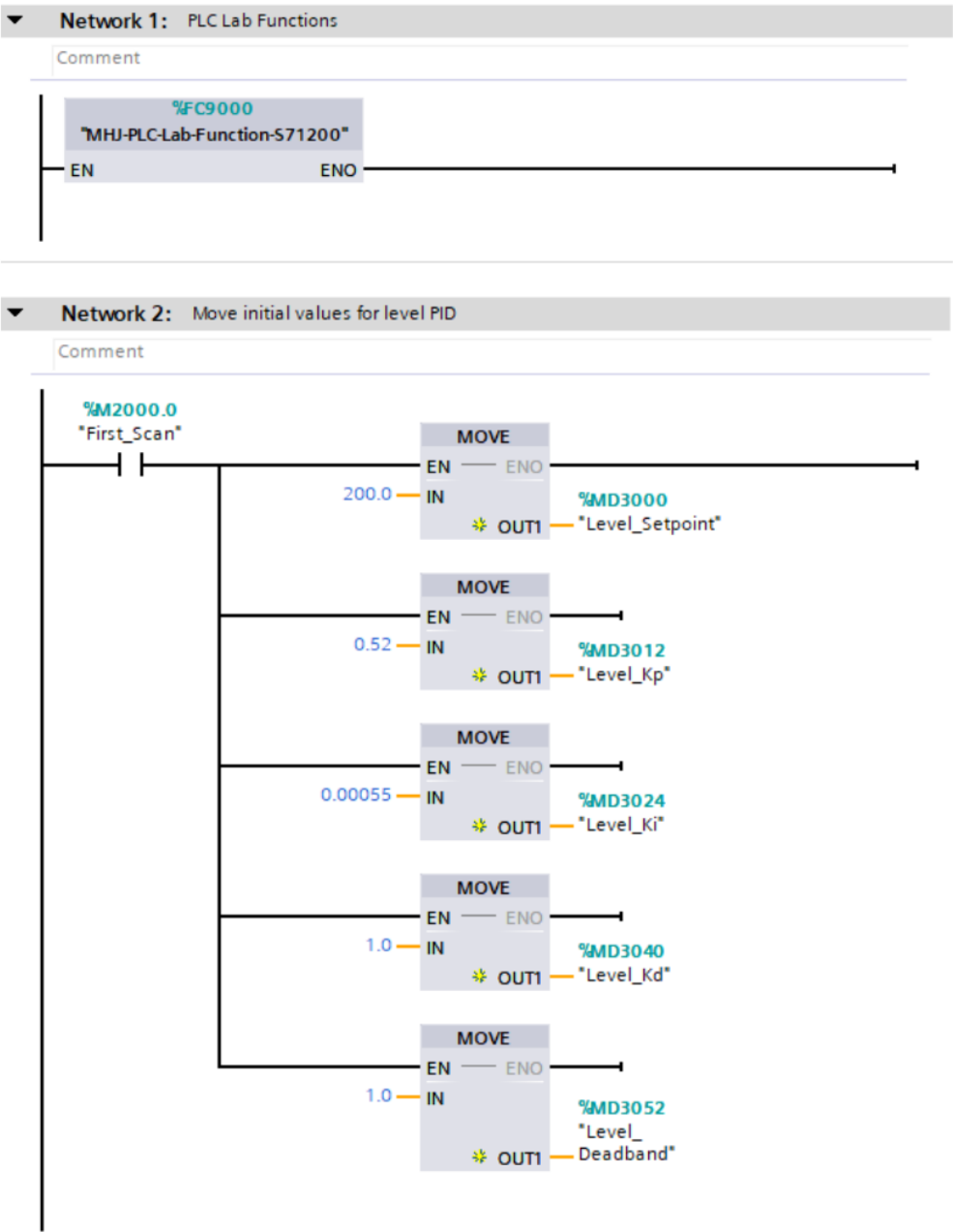
8.8.5. Buttons for Drain Valve Flow Control HMI screen tags

19		HMI_Start_Flow_Control	Bool	%M2600.2
20		HMI_Stop_Flow_Control	Bool	%M2600.3
21		HMI_Manual_Open_Drain_Valve	Bool	%M2600.4
22		HMI_Manual_Close_Drain_Valve	Bool	%M2600.5
23		HMI_Manual_Open_Drain_Valv...	Bool	%M2600.6
24		HMI_Manual_Close_Drain_Valv...	Bool	%M2600.7

8.8.6. Displays for Drain Valve Flow Control HMI screen tags

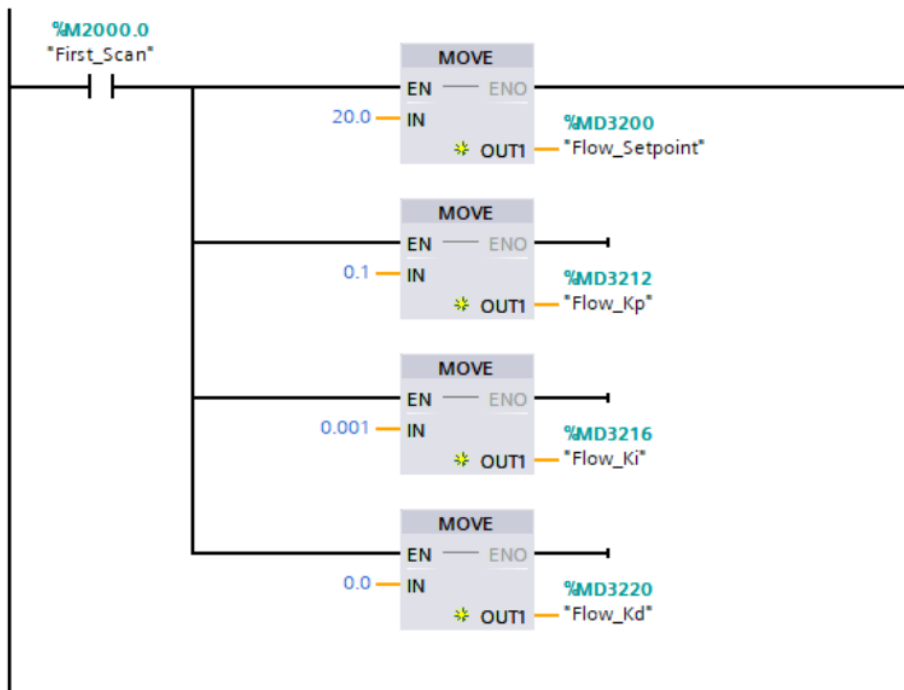
18		Flow_Control_RUN	Bool	%M2600.0
37		Flow_Setpoint	Real	%MD3200
38		Flow_Process_Variable	Real	%MD3204
39		Flow_Error	Real	%MD3208
40		Flow_Kp	Real	%MD3212
41		Flow_Ki	Real	%MD3216
42		Flow_Kd	Real	%MD3220

8.9.1. Main Control Blocks



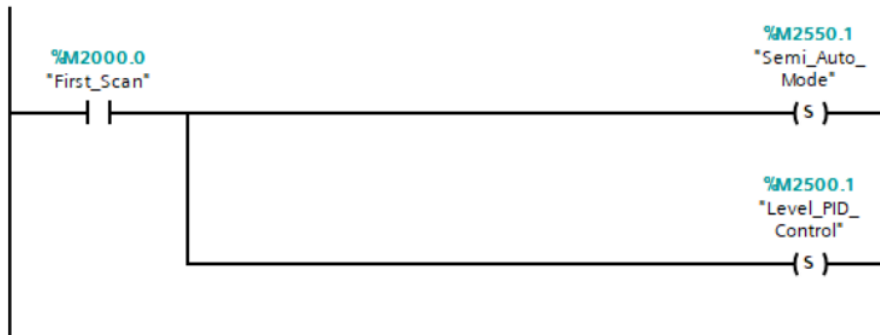
▼ **Network 3:** Move initial values for flow PID

Comment

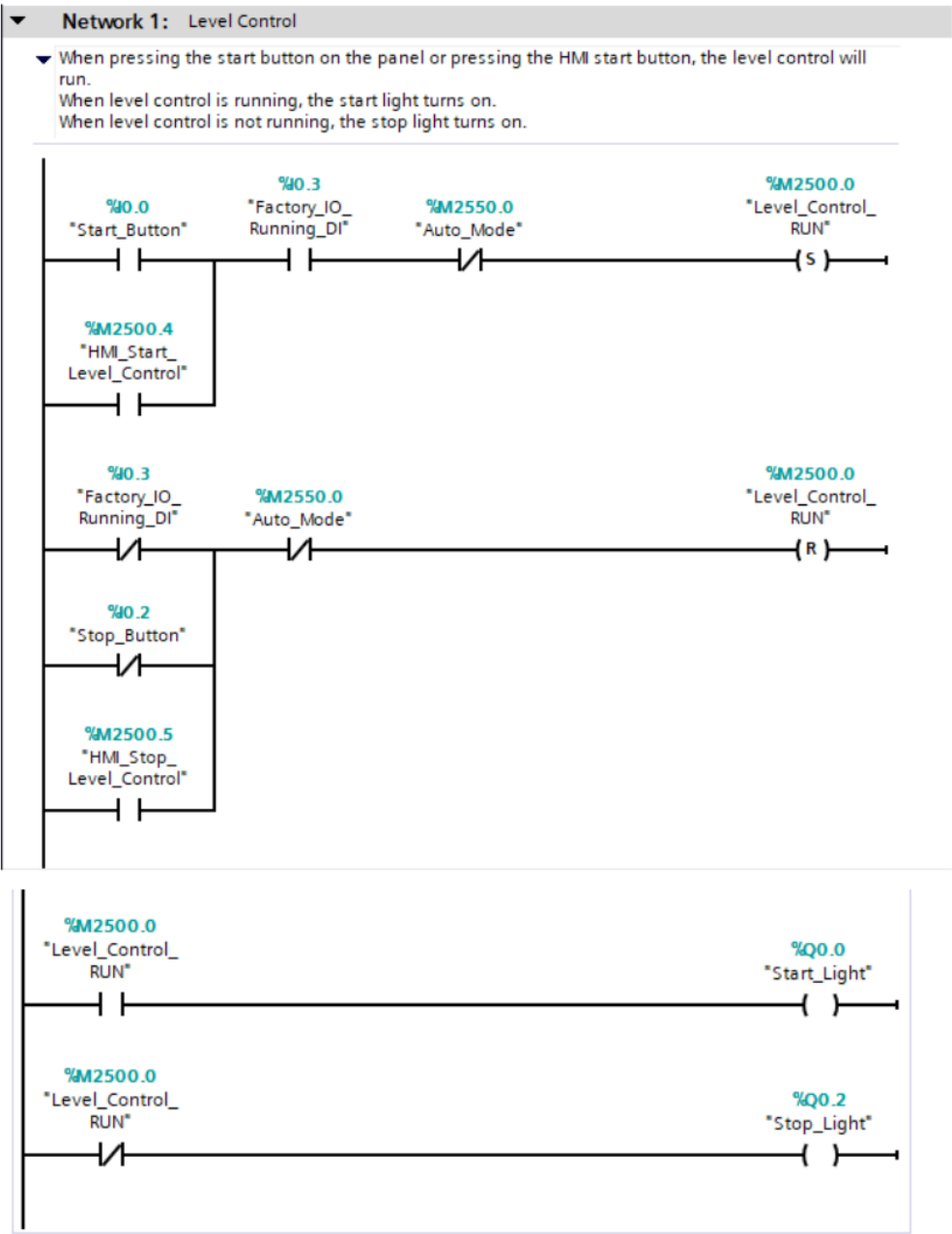


▼ **Network 4:** Initial mode on first boot up

Comment

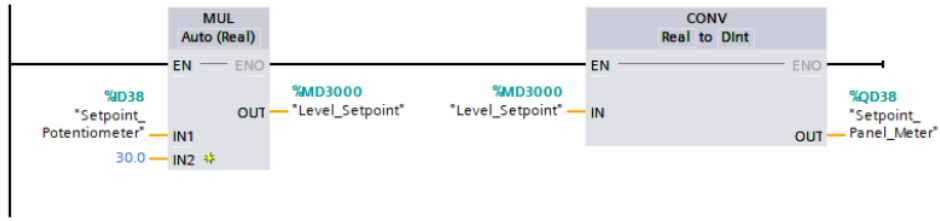


8.9.2. Level Control Blocks



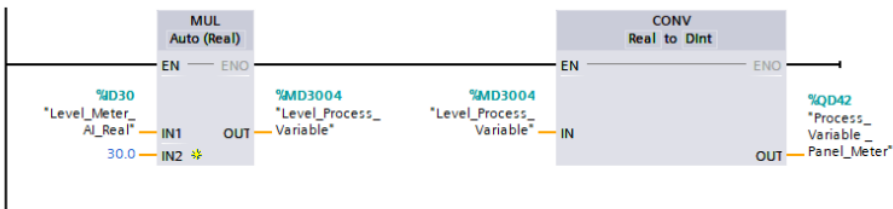
Network 2: Scale Level Setpoint and Display

- ▼ The 10V potentiometer is multiplied by 30 to match the tank level of 300. When the potentiometer is adjusted, the level setpoint adjusts with it. The level is then displayed on the setpoint panel meter by converting it from a real data type to a Dint.



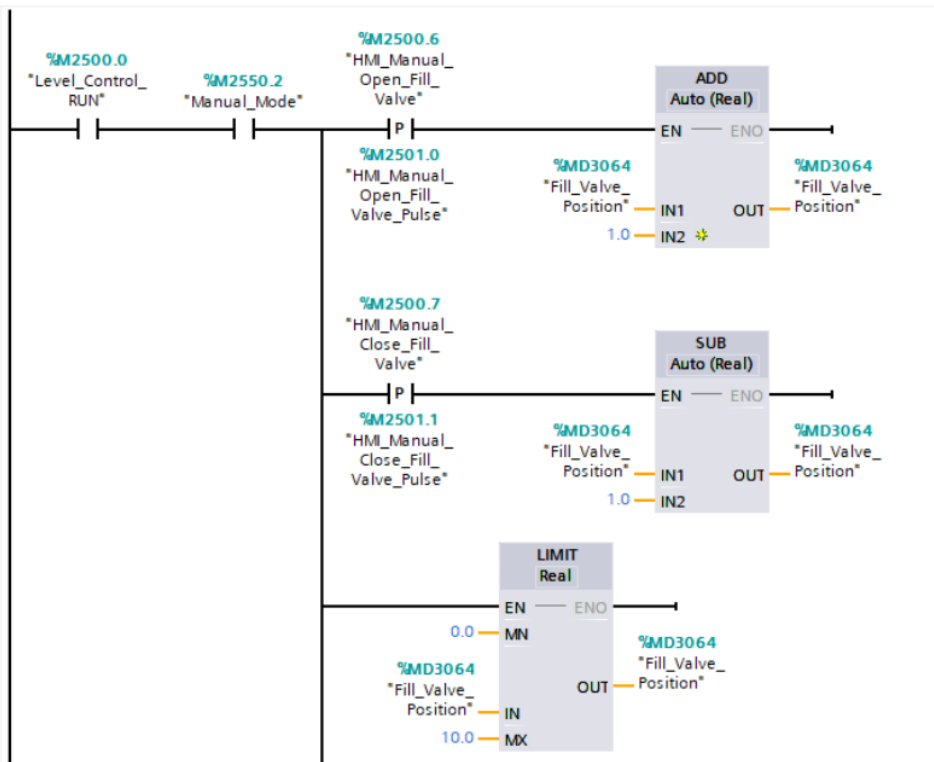
Network 3: Scale Level Process Variable and Display

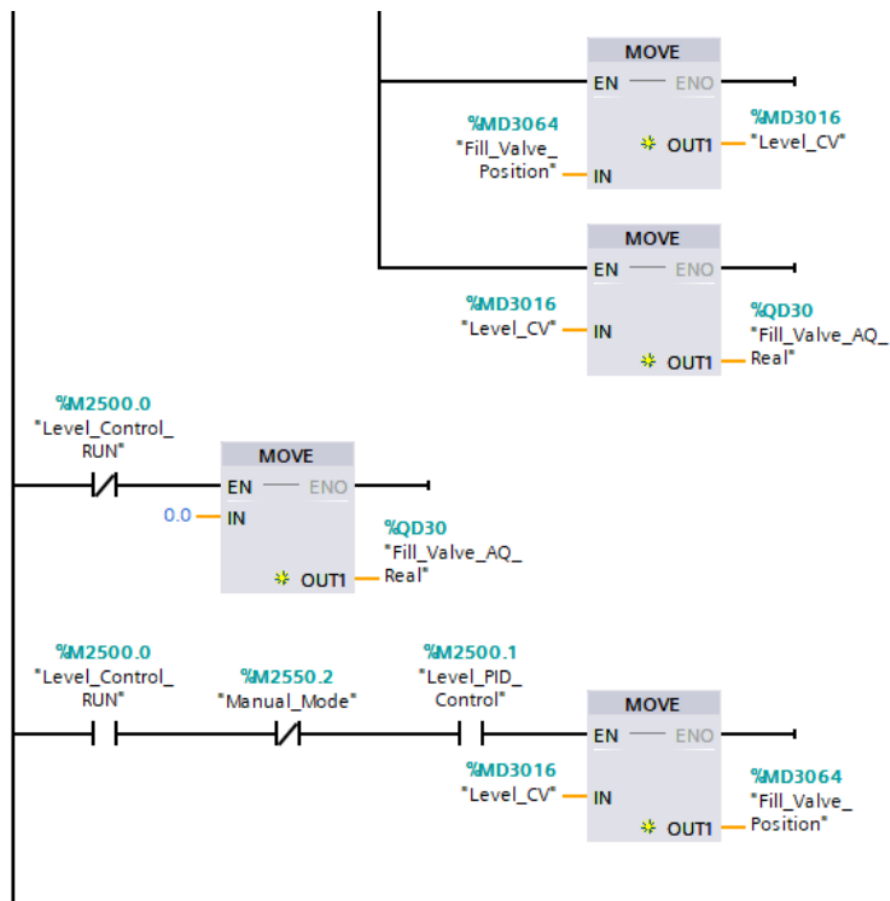
- ▼ The 10V level meter is multiplied by 30 to match the tank level of 300. The level meter detects what level the fluid is at and displays it on the process variable panel meter by converting it from a real data type to a Dint.



Network 4: Manual Fill Valve Control

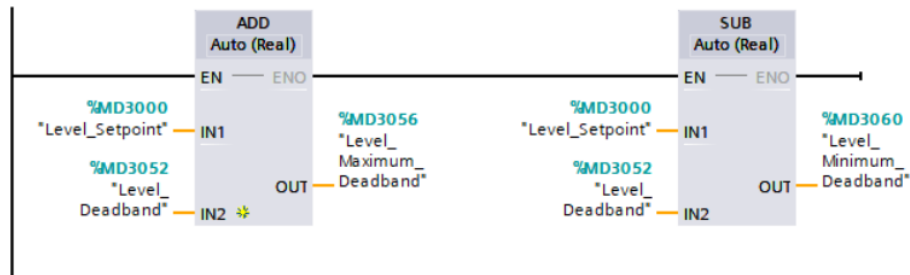
- Pressing the manual button will enable the user to operate manual controls.
- The open/close fill valve has a tolerance of 1 hence them being added and subtracted by 1.
- The fill valve position has a minimum of 0 and maximum of 10.





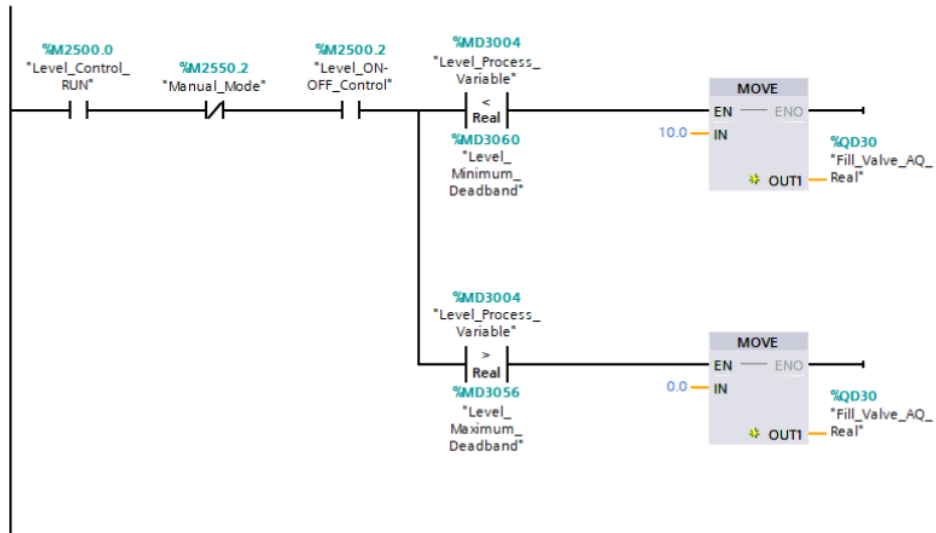
Network 5: ON-OFF Control Deadband

- ▼ To work out the maximum deadband, it adds the setpoint level to the deadband.
To work out the minimum deadband, it subtracts the deadband level from the setpoint.



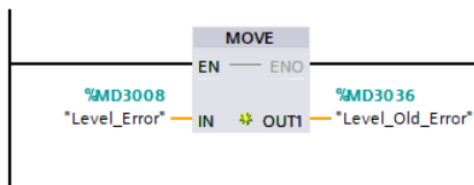
▼ Network 6: Level ON-OFF Control with Deadband

- ▼ The ON-OFF controls can be operated with deadband when manual mode is off.
When the level process variable is less than the minimum deadband, the tank fills by 10.
When the level process variable is greater than the maximum deadband, the tank stops filling.



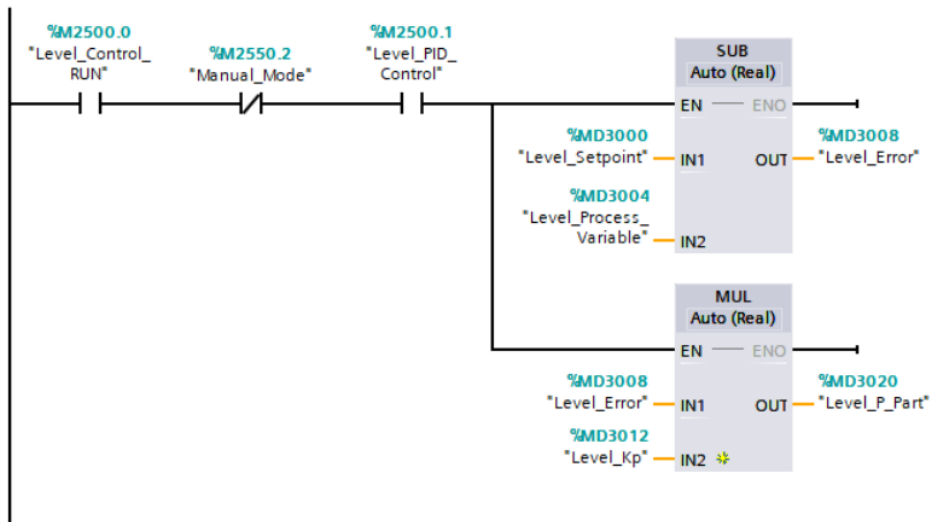
Network 7: Store the Old Error

Stores the error



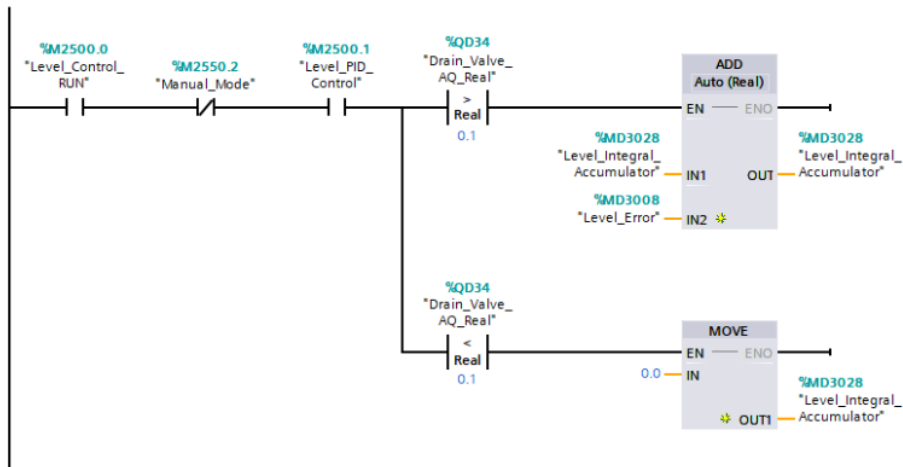
Network 8: Level PID Control - Calculate the Error and the P Part

- The PID control can be operated when out of manual mode.
Level error is calculated by subtracting the level process variable from the level setpoint.
The level P part is worked out by multiplying the level error with the level Kp.



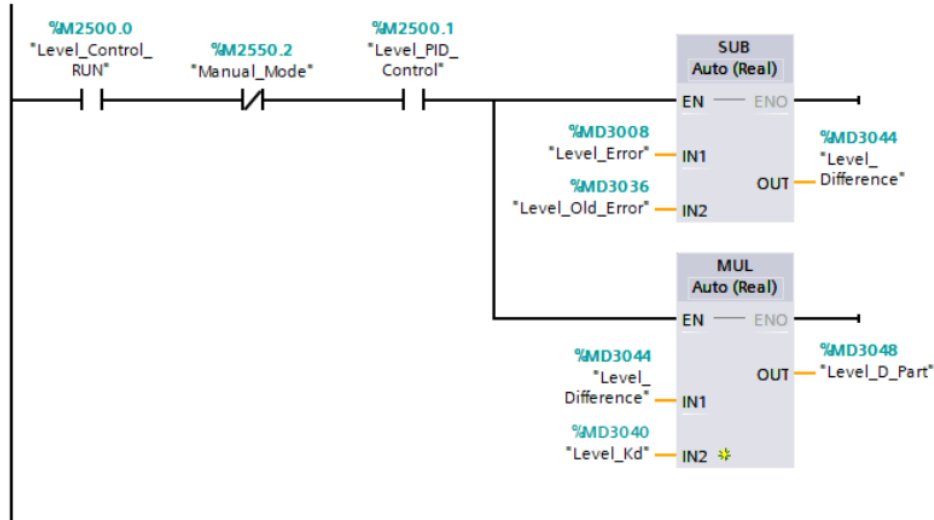
Network 9: Level PID Control - Calculate the I Part

- When the drain valve is greater than 0.1, the level integral accumulator will add to the level error.
When the drain valve is less than 0.1, the level integral accumulator remains the same.



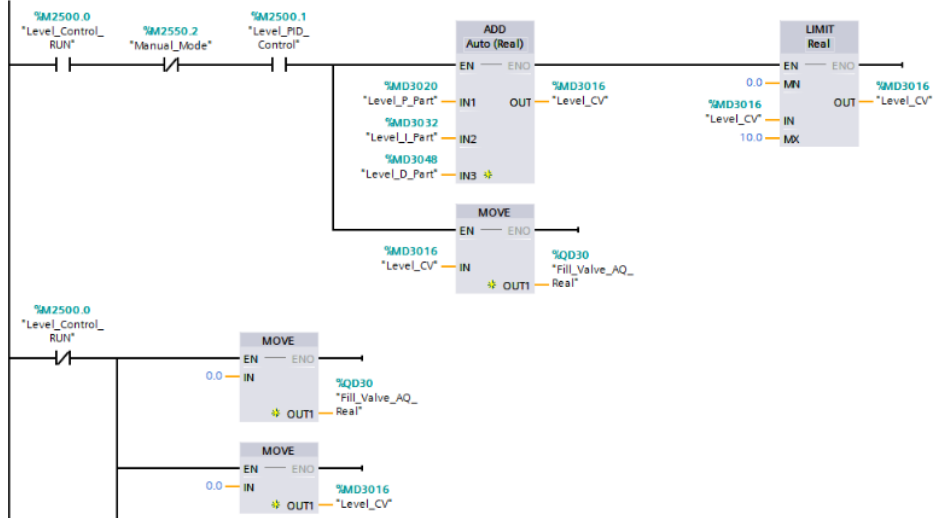
Network 10: Level PID Control - Calculate the D Part

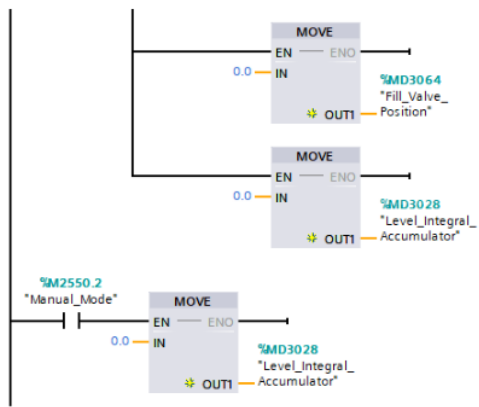
- The level difference is calculated by subtracting the old error level from the level error.
- The D part is calculated by multiplying the level difference by the level Kd.



Network 11: Level PID Control - Calculate the CV

- The CV is calculated by adding the level P part, I part, and D part together.
- The limit of the CV is the CV plus 10.

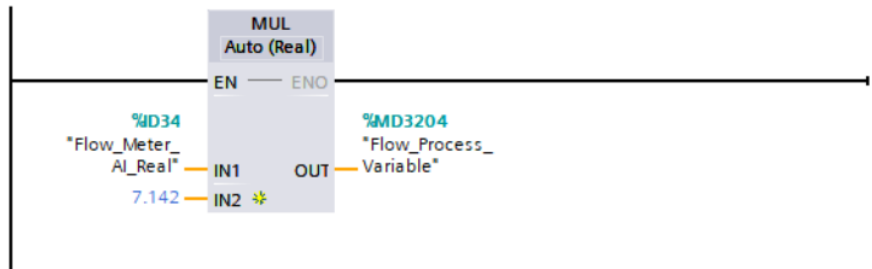




8.9.3. Flow Control Blocks

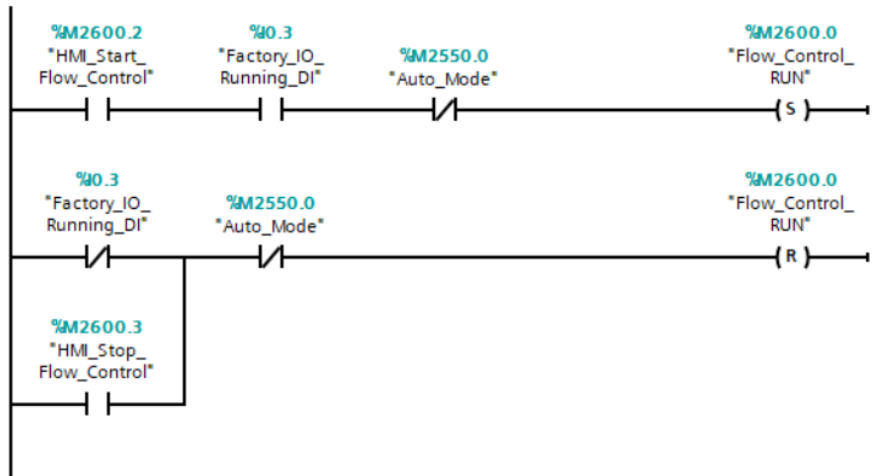
▼ **Network 1:** Read in Analogue Input for Flow

Multiplies the flow meter value by 7.142 and displays the answer on the process variable.



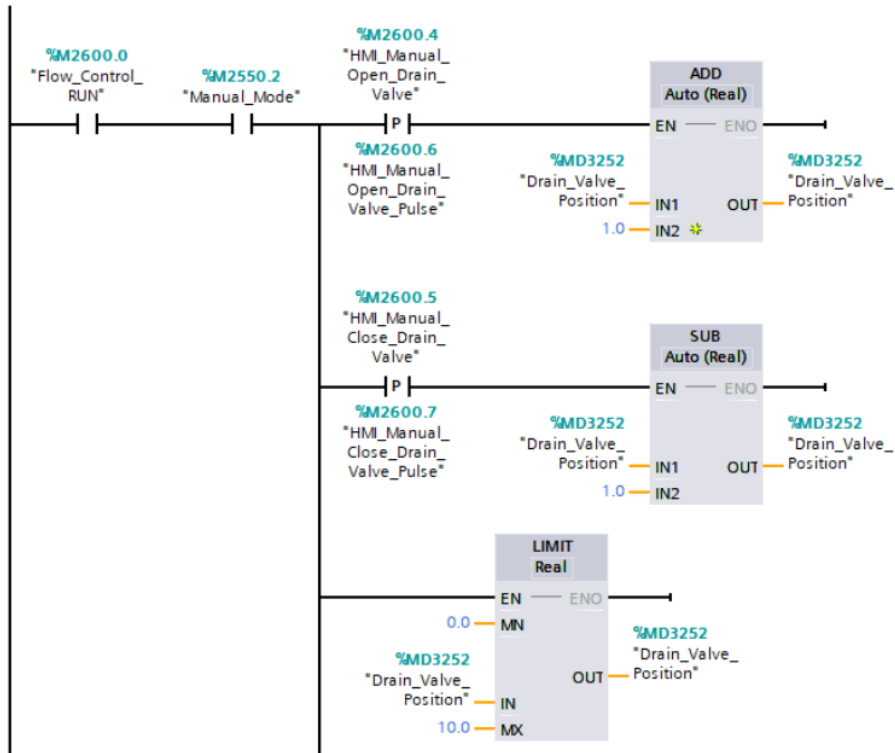
▼ **Network 2:** Flow Control

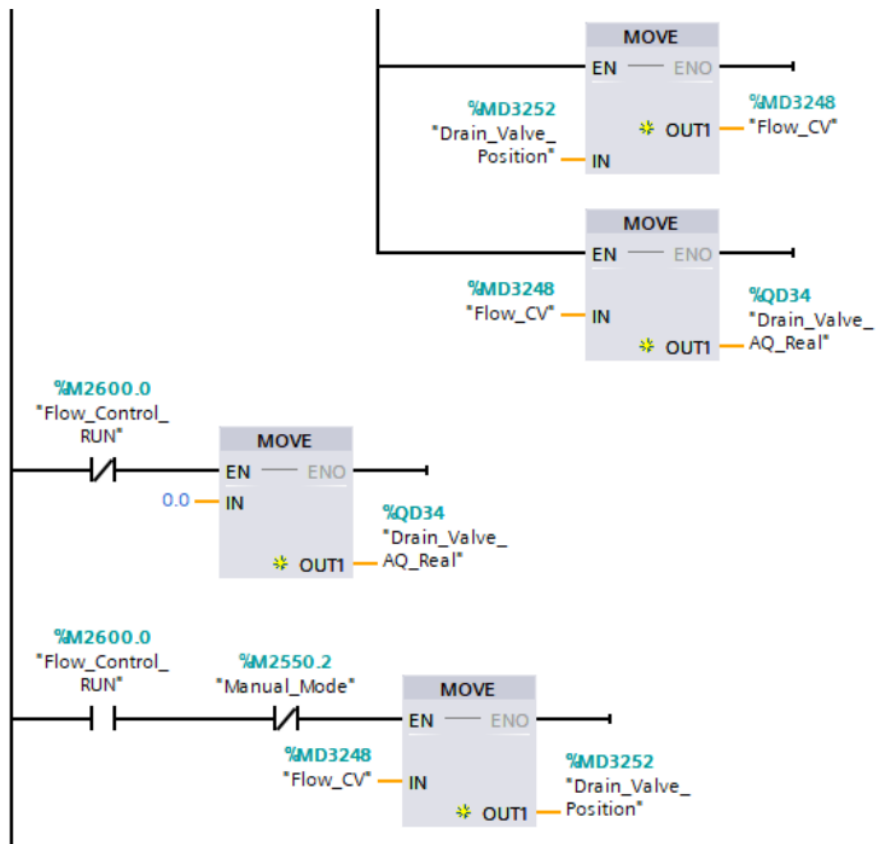
- ▼ Pressing the start button on the flow control HMI screen will begin the flow control.
- Pressing the stop button on the flow control HMI screen will end the flow control.



Network 3: Manual Drain Valve Control

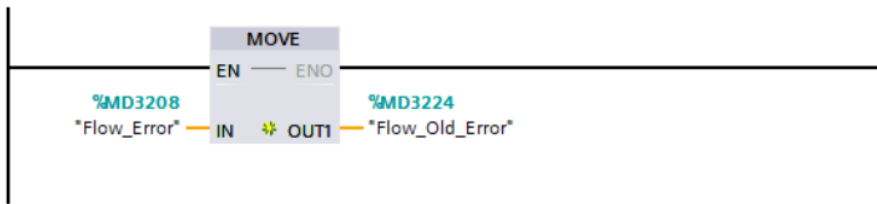
- Pressing the manual button will enable the user to operate manual controls.
The open/close drain valve has a tolerance of 1 hence them being added and subtracted by 1.
The drain valve position has a minimum of 0 and maximum of 10.





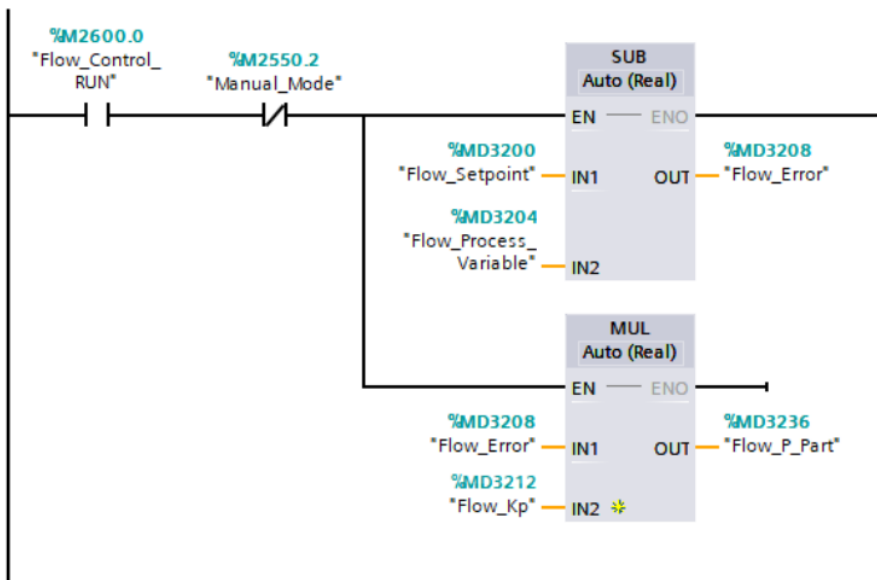
▼ **Network 4:** Store the Old Error

Stores the old error.



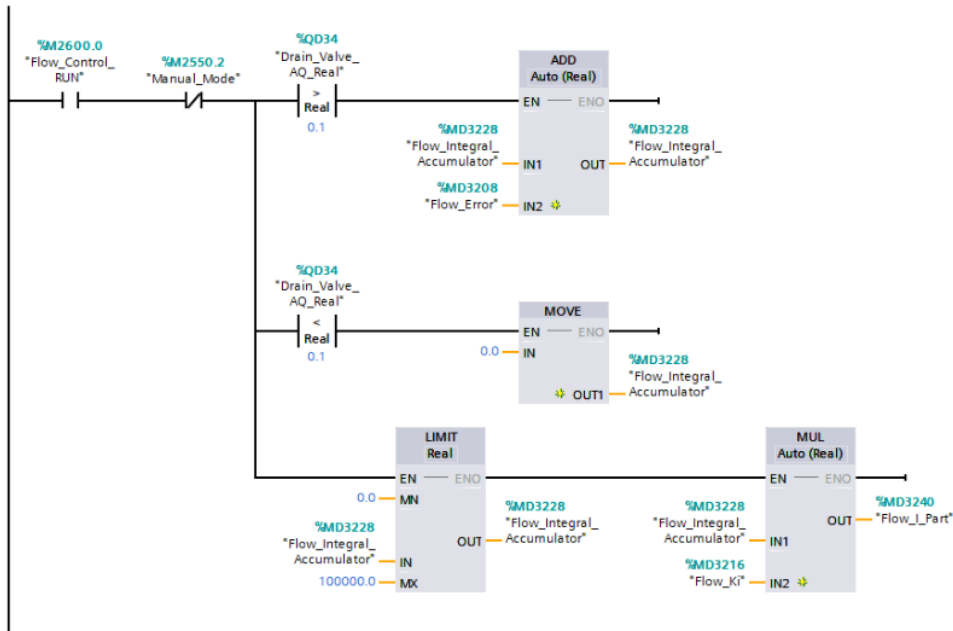
▼ **Network 5:** Flow PID Control - Calculate the Error and the P Part

- ▼ Flow Error is calculated by subtracting the flow process variable from the flow setpoint.
The flow P part is worked out by multiplying flow error with flow Kp.



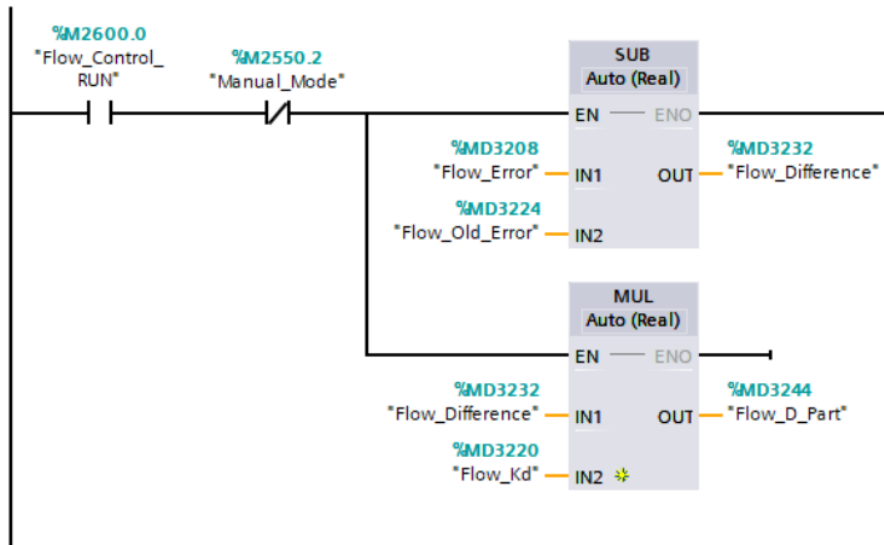
Network 6: Flow PID Control - Calculate the I Part

- When the drain valve is greater than 0.1, the flow integral accumulator will add to the flow error.
- When the drain valve is less than 0.1, the flow integral accumulator remains the same.
- The flow I part is calculated by multiplying the flow integral accumulator with the flow Ki.



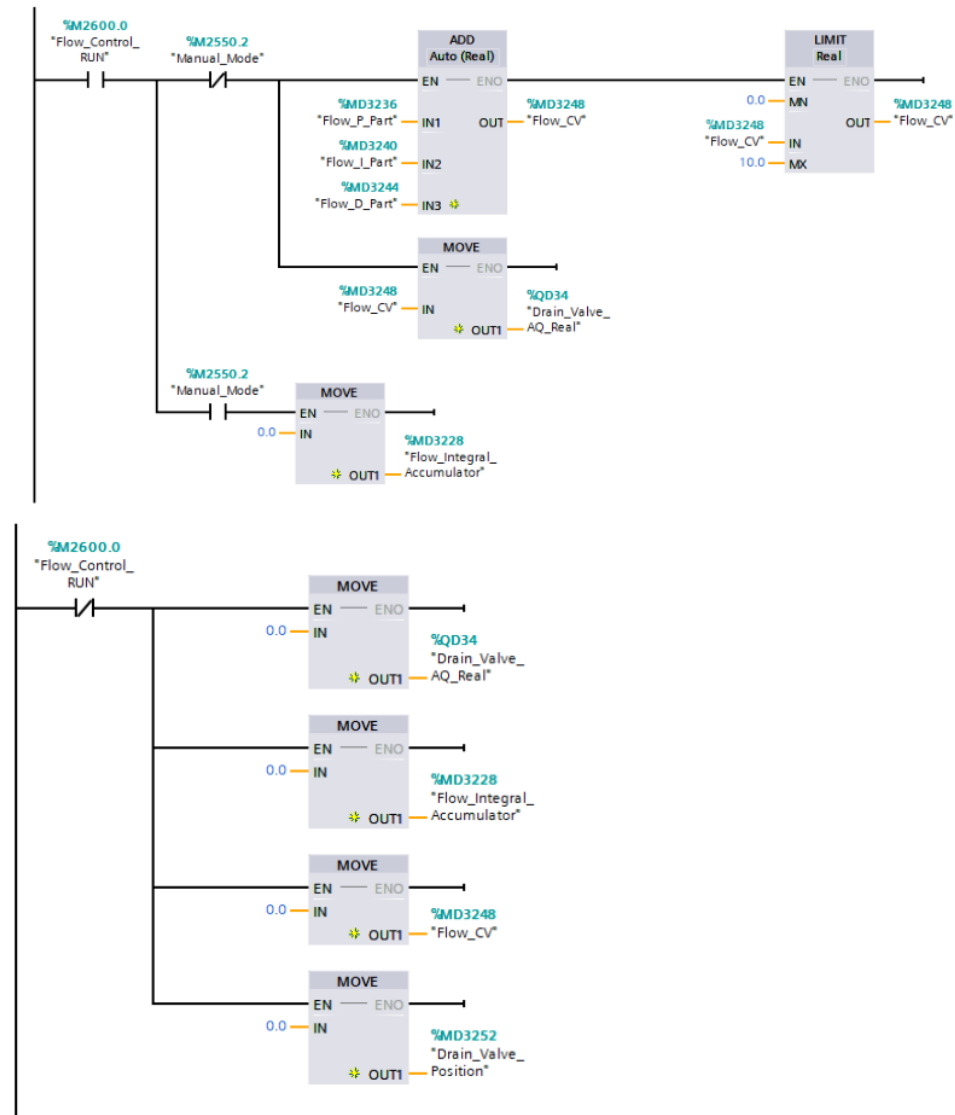
Network 7: Flow PID Control - Calculate the D Part

- Flow difference is calculated by subtracting flow old error from flow error.
- The flow D part is the product of the flow difference and flow Kd.

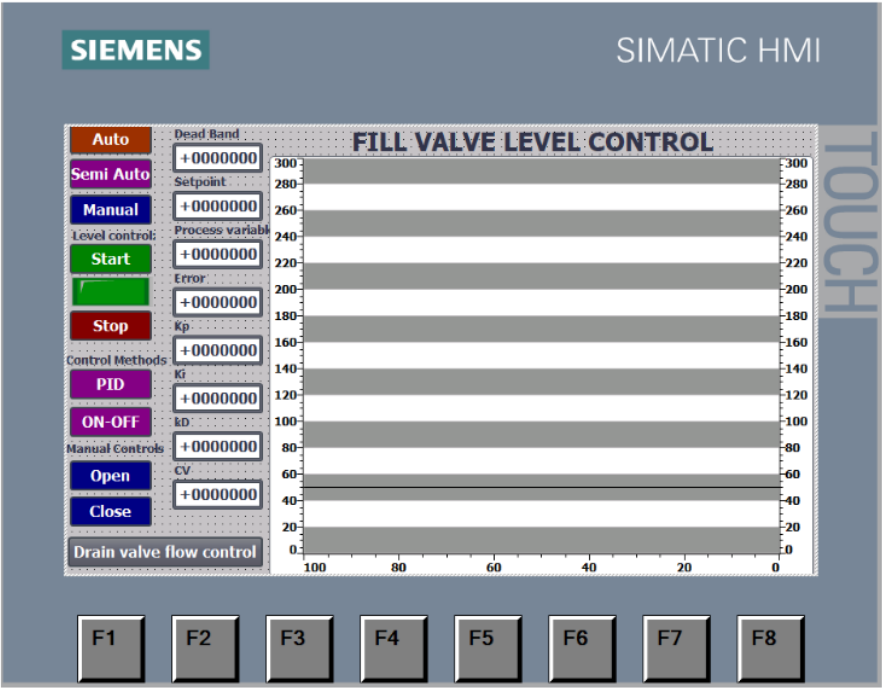


Network 8: Flow PID Control - Calculate the CV

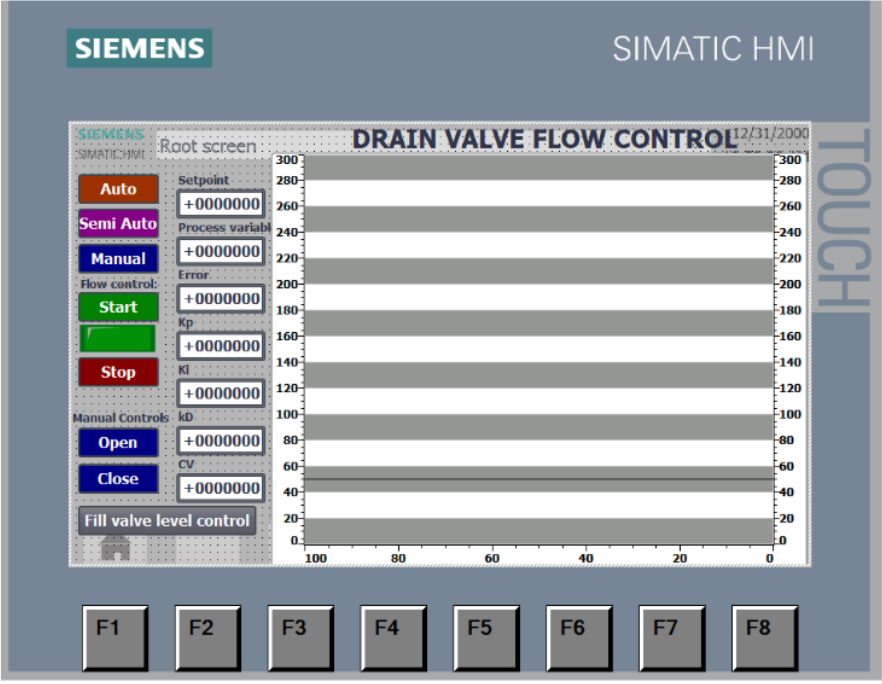
Flow CV is calculated by adding the flow P part, I part, and D part together.
The limit of the CV is the CV plus 10.



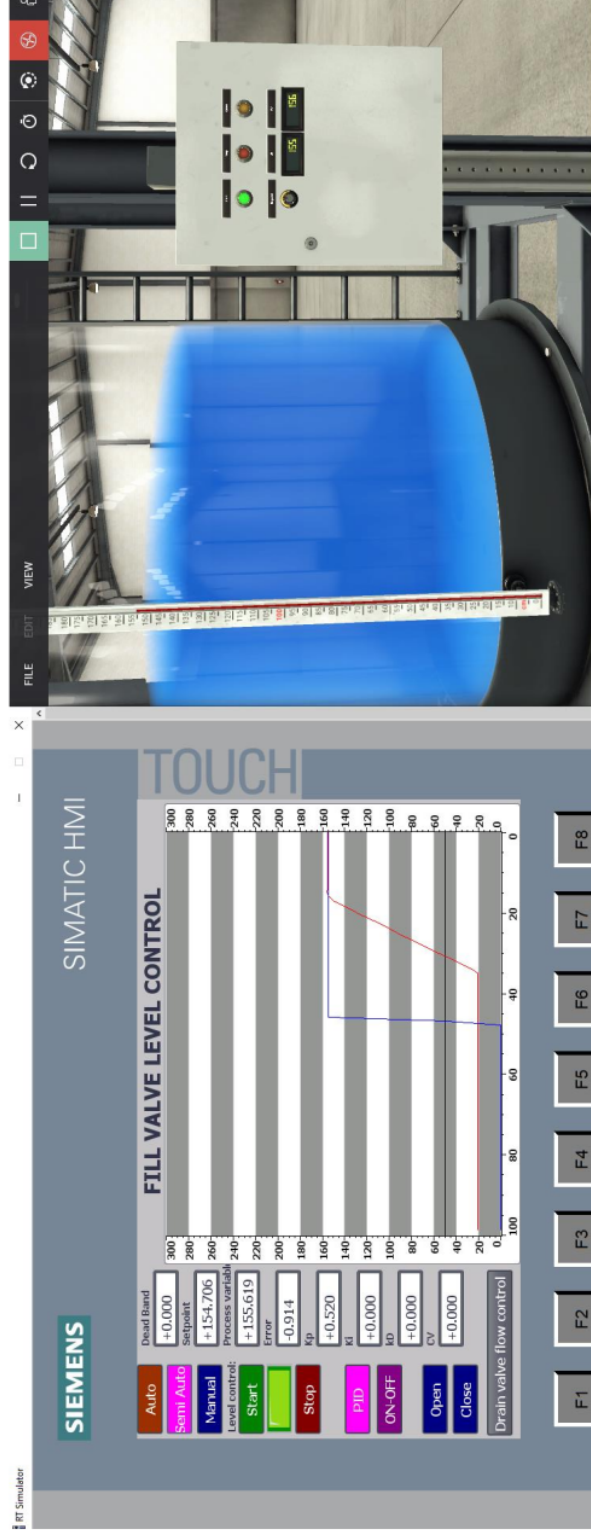
8.10.1.HMI screen for fill valve level control screen



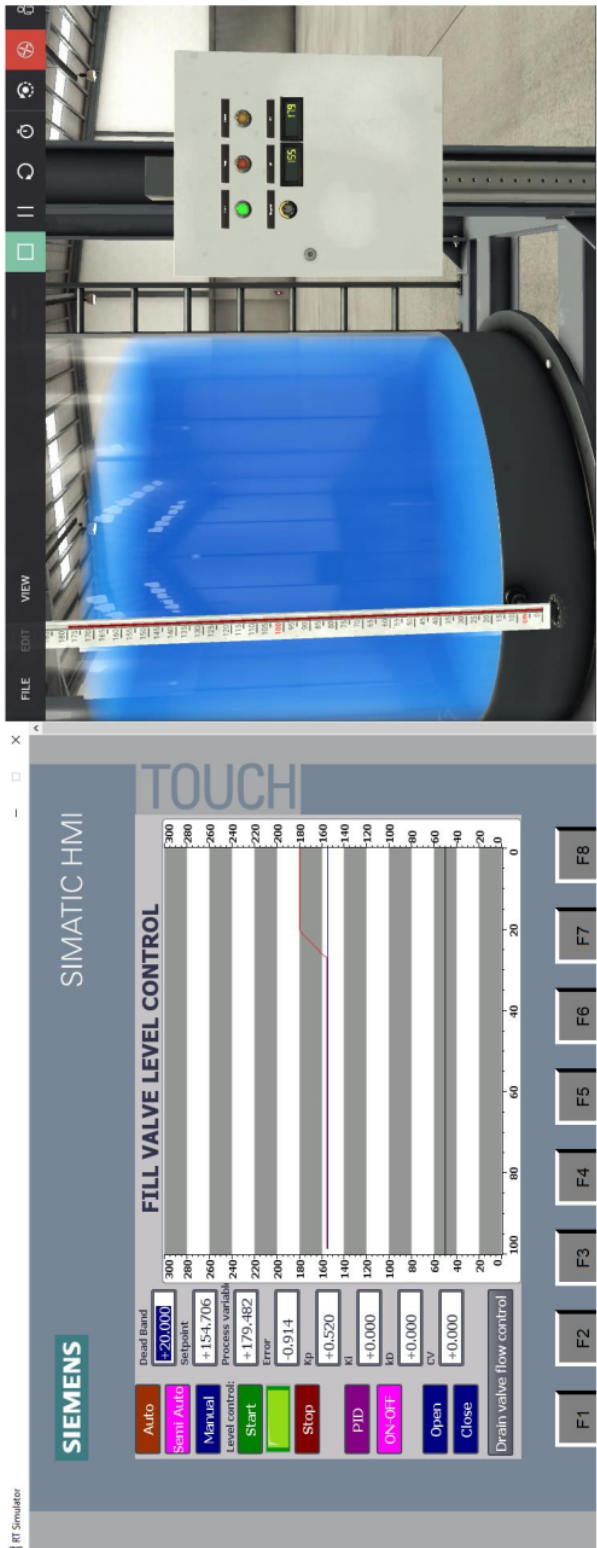
8.10.2.HMI screen for drain valve flow control screen



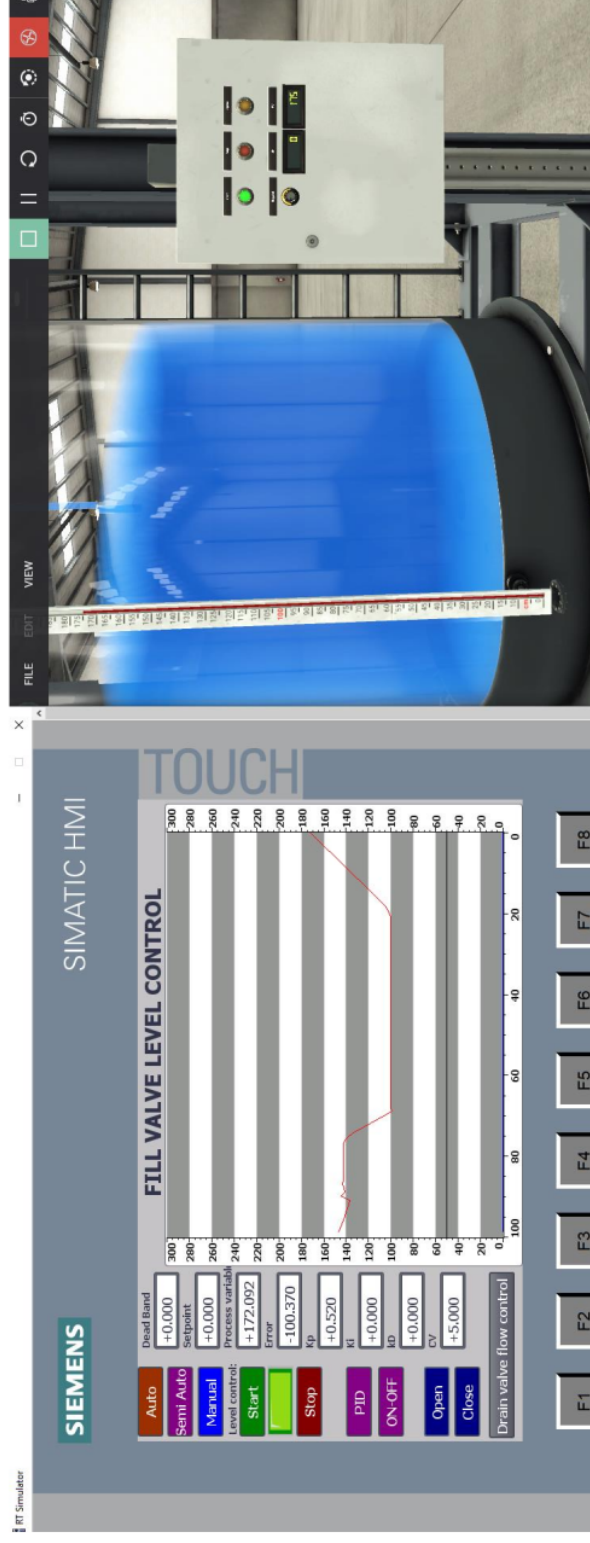
8.11. Filling in semi auto PID by tuning potentiometer



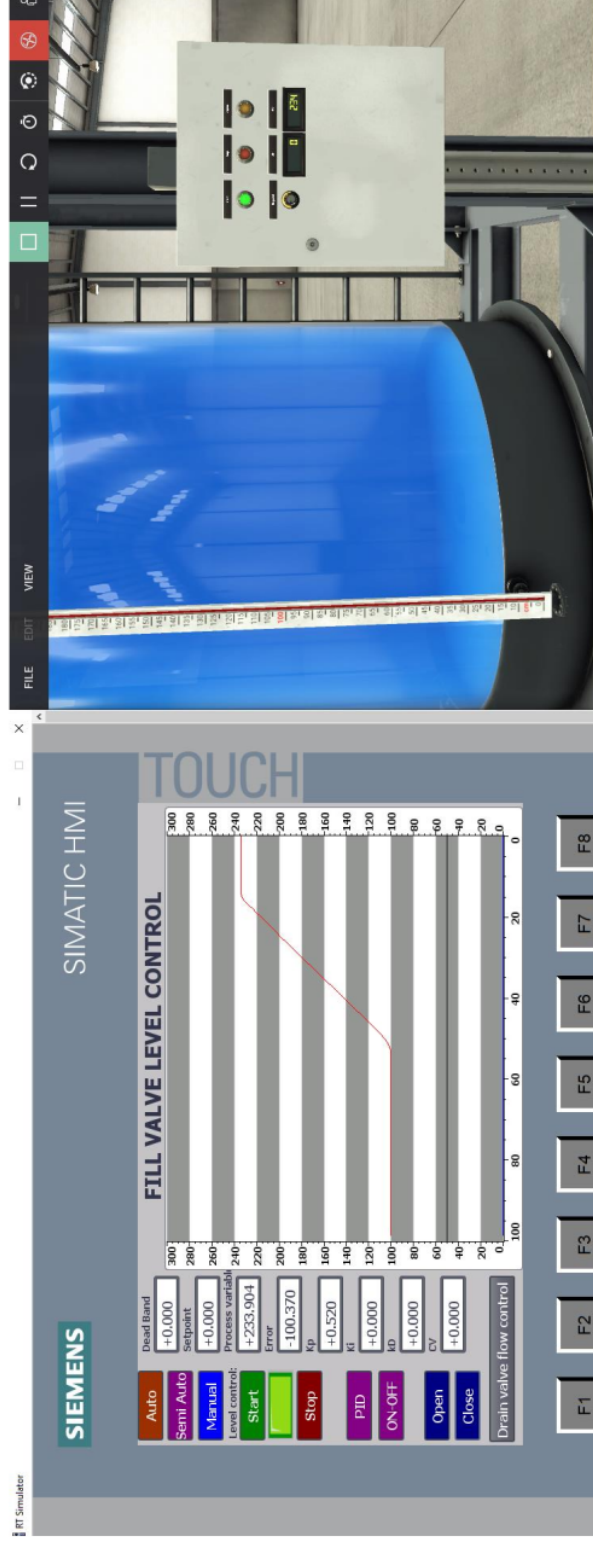
8.12. In ON-OFF mode and choosing a deadband of 20



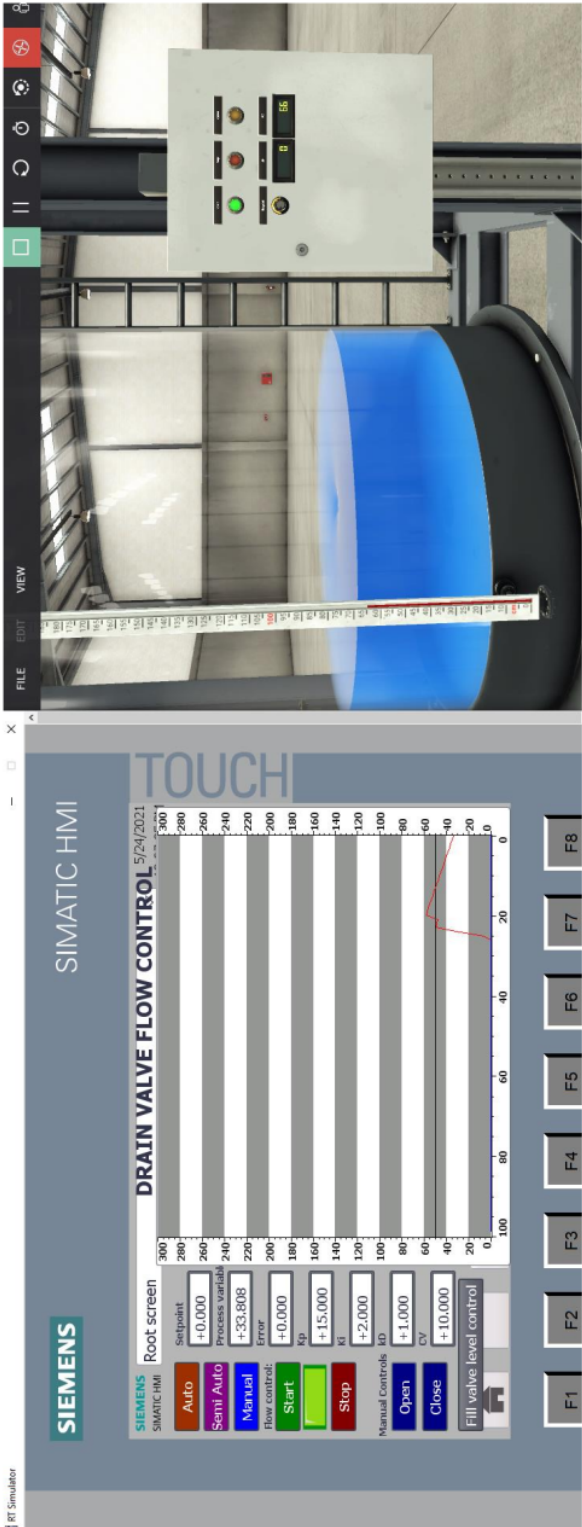
8.13. Manually opening fill valve



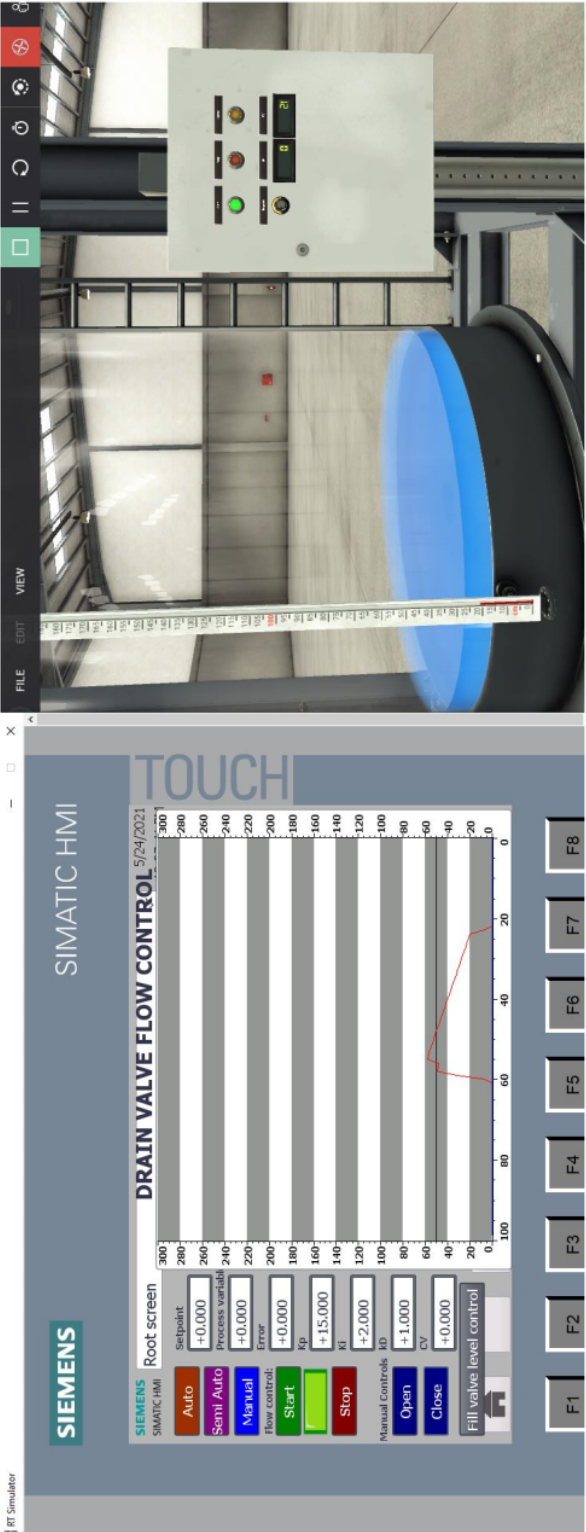
8.14. Manually closing fill valve



8.15. Manually opening drain valve



8.16. Manually closing drain valve



FINAL GRADE

GENERAL COMMENTS

Instructor

68/100

PAGE 1

PAGE 2

QM

Good Structure

Report is structured exactly as stated in the assignment brief.

PAGE 3

PAGE 4

QM

Wrong choice of word/s

You could have chosen a better word to express your thought here

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

PAGE 5

QM

Improper Citation

Improper citation:

Improperly cited material. Please use the link below to find links to information regarding specific citation styles: http://www.plagiarism.org/plag_article_citation_styles.html

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

QM

Correct method of listing.

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

QM

Good range of sources

You have demonstrated reading from a good range of sources shown by your list below

PAGE 15

PAGE 16

PAGE 17

PAGE 18

Text Comment. analogue output module is missing

PAGE 19

PAGE 20

PAGE 21

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

GRADING FORM: NEW CSG-RUBRIC

LIAM PHILLIPS

68

1. ENGAGEMENT WITH LITERATURE



Engagement with an extensive range of relevant and credible literature. Consistently accurate application of referencing. (14/20)

14

2. KNOWLEDGE AND UNDERSTANDING



Competent knowledge of the basic underlying concepts and principles. Exhibits very good understanding. (13/20)

13

3. COGNITIVE AND INTELLECTUAL



Excellent evaluation of the underlying concepts and principles. Can interpret qualitative and quantitative data accurately and with some insight. Excellent ability to devise arguments using evidence and make appropriate and valid judgements. (14/20)

14

4. PRACTICAL APPLICATION



An advanced application of a range of taught, basic methods, materials, tools and/or techniques. The context of the application is well considered, and insightful. Excellent evidence of different approaches to problem-solving in particular contexts and proposes appropriate solutions. (14/20)

14

5. TRANSFERABLE SKILLS



Work is accurate, coherent, fluent, well-structured and organised. Can work effectively independently and/or as part of a team, with very good contribution to group activities. Demonstrates very good skills for employment requiring the exercise of some personal responsibility, with just occasional minor weakness. (13/20)

13