

LEVEL 4 ASSESSMENT SPECIFICATION

Student name:	Jack Collier	Student ID number:	
Programme:	HNC Electrical/Electronic Engineering & HNC Instrumentation Engineering		
Module:	Programmable Logic Controllers		
Module code:	PCEE4005	Contribution to Overall Module Assessment (%):	50%
Lecturer:	Jason Pointer	Internal Verifier:	Lance Rutherford
Assignment Title:	PLC Program design and simulation	Word count (or equivalent):	2000
Submission deadline:	05/05/22	Return date of provisional marks & written feedback:	19/05/22
Submission method:	All written assessments, where practical and possible, must be submitted via Turnitin unless otherwise instructed by the Lecturer. (Please DO NOT put this assessment specification into Turnitin or it will match many similarities with other students' submissions.) Alternative submission method (if applicable): Late submission of the assessment will result in a late penalty mark. Penalties for late submission: Up to one week late, maximum mark of 40%. Over one week late, 0%. Only the Extenuating Circumstances Panel may approve a change to submission dates.		
Academic honesty / referencing:	Academic honesty is required. In the main body of your submission you must give credit to authors on whose research and ideas your work is based. Append to your submission a reference list that indicates the books, articles, etc. that you have used, cited or quoted in order to complete this assessment.		

Module Learning Outcomes (from module syllabus)

1. Demonstrate an understanding of the concepts of process control using an industry standard PLC.
2. Undertake basic PLC system design and implementation processes, comparing theoretical expectations to actual outputs ensuring correct operation and consideration of maintenance and reliability.

STUDENT FEEDBACK FORM

This section details the *extent* to which the assessment criteria are demonstrated by you, which in turn determines your mark. The marks available for each category of skill are shown. Lecturers will use the space provided to comment on the achievement of the task(s), including those areas in which you have performed well and areas that would benefit from development/improvement.

Generic Assessment Criteria	Marks available	Marks awarded
1. Engagement with Literature Skills	15	8.3
Engagement with an appropriate range of literature, including sources retrieved independently. Do need to include some direct references.		
2. Knowledge and Understanding Skills	30	18.5
Task 1 achieved 46%. Task 2 achieved 56% Task 3 achieved 87% Showed some competent knowledge of the basic underlying concepts and principles for assessment but could have been more detail in the explanation of Task 1. Need to include other examples of input devices. Excellent work producing the IL code for Task 3.		
3. Cognitive and Intellectual Skills	10	4.8
Little evaluation of the underlying concepts and principles applied to the construction of the programs for Task 4 and 5.		
4. Practical Application Skills	35	23.3
Task 4 achieved 65%. Task 5 achieved 68%. A very good application of the different software to produce fully functional programs to cover the requirements of Task 4 and 5. Required very little support in the development of these programs and was able to show full operation. The only issue is your presentation of the programs in your report, the screenshots do not show all of the programs.		
5. Transferable Skills for Life and Professional Practice	10	5.8
Mostly coherent, organised and accurate work, in a suitable structure and is for the most part clearly expressed.		

Assessment Mark (Assessment marks are subject to ratification at the Exam Board. These comments and marks are to give feedback on module work and are for guidance only until they are confirmed.)	Late Submission Penalties (tick if appropriate)		61%
	Up to 1 week late (40% Max)		
	Over 1 week late (0%)		

Jack Collier

**Programmable
Logic Controllers**

Temperature Sensors

With respect to temperature, the most common type of device is the RTD (resistive temperature detector). These types of sensors are usually made from platinum but can also be made from things such as nickel or alloys of nickel. RTD's work on the principle that the resistance of a conductor changes depending on the temperature of the conductor, when hotter the resistance increases and when colder the resistance decreases. Platinum has a very linear change in resistance for temperature change which makes it particularly good for this usage. The main issue with using a resistance as a sensing medium is that you run into issues as the cables going from the plc to the sensor will also have a resistance, this will therefore give us inaccurate readings for temperature. To combat this, we use a Wheatstone bridge, as instrumentationtools.com says this works as "Measurements are made by passing current through a sensing resistor and measuring the potential across it. If the current is known, the potential is a measurement of the resistance and hence the temperature. If the current is not known exactly the potential may be compared with the potential across a known resistor; this is the basis of the bridge systems". An example of this is shown in the figure 1 below, it shows a 4 wire system where the red and white cables cancel out the blue cables. ✓

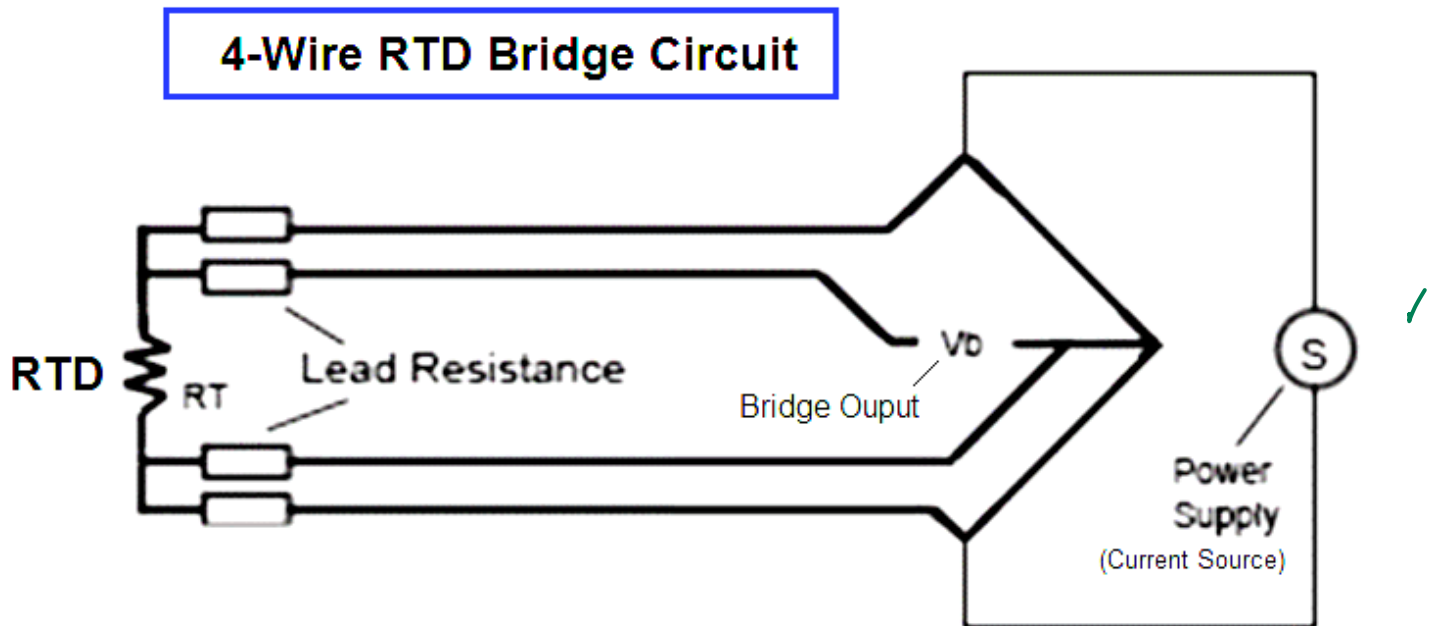


Figure 1 4-wire RTD Wheatstone Bridge Circuit

Pressure Sensors

Another example of a sensor for a plc is a pressure sensor, these can measure the pressure of a system by using a diaphragm. The diaphragm is made out of a flexible material and when there is a pressure difference on either side of the diaphragm it bends, this can then bend against a piezoelectric crystal which will output a voltage. The amount of voltage generated by the piezoelectric crystal is proportional to the pressure difference on either side of the diaphragm and usually generates 0.6mV/kPa. An example of this type of pressure sensor can be seen below in Figure 2. ✓

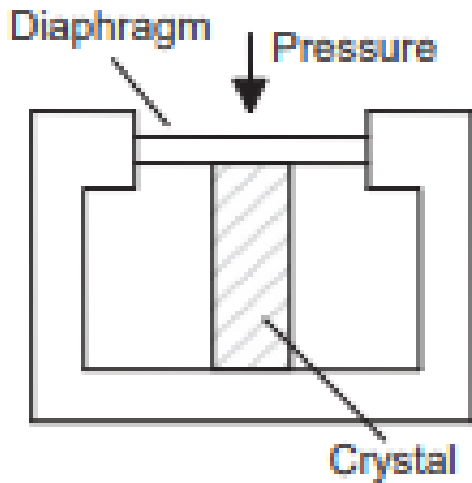
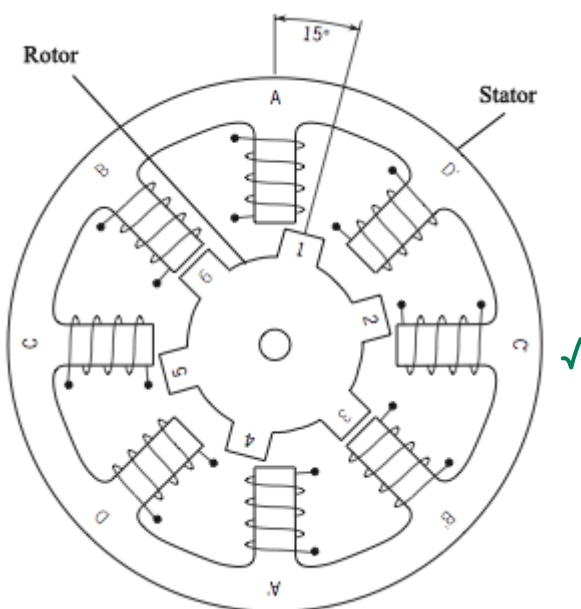


Figure 2 Piezoelectric Pressure Sensor

You should have explained 2 different types for each.

Stepper Motors

A stepper motor is a type of motor that are able to move accurately in increments, they are typically DC. They achieve this by having multiple coils around the outside of the rotor (not on it), these coils are energised in pairs and in sequence which causes the rotor to rotate. This means they can have a very accurate speed control or if you need something to turn a specific amount of degrees. These properties mean they are used in a lot of production equipment like textile machines, printing machines, cnc milling machines and small robotics. ✓



Your explanation should have been more detailed.

Figure 3 DC Stepper Motor Diagram

SCADA Control Systems

SCADA stands for supervisory control and data acquisition, it's the main brain of most large scale controls systems and is responsible for taking in large amounts of information, processing that information and give out the correct outputs to respond to this information. Robert Radvanovsky says in the SCADA Control Systems Handbook "*control systems collect sensor measurements and operational data from the field, process and display this information, and relay control commands to local or remote equipment*". Information can be from various sources such as temperature sensors, pressure sensors, vibrations sensors, speed sensors or any type of signal. This is then put into a RTU (remote telemetry unit) which can take in different types of signals and convert them to signals used by the plc (usually there is more than one) to calculate an output and then control another piece of the process such as what percentage a valve is open, ignitors for starting a boiler or turbine or start signal for a motor or pump. The input signals aren't just used for calculating outputs as the SCADA system usually uses an HMI (human machine interface) so our inputs could be for position of valves or displaying levels in tanks and pressures of vessels and through the HMI, the operator can control different parts of the process from one place. The SCADA system also has a functionality of being able to log data over long periods of time, this means that the operator can trend collected data on a graph, this is useful when there is an issue, or a trip and you can look back in time at different parts of the process allowing you to see what was different from a normal state to when the issue occurred. An example of a SCADA system is the control gear used for the COGEN unit at Valero. This uses a SCADA system developed by GE gas and power called the Mark Vie, this controls all of the running of the gas turbine including controlling all the staging valves (fuel control valves), VBV (variable bleed valve) depending on the mega watt demand which is specified by the operator. This SCADA system (*Figure 4*) also has to communicate with two other SCADA systems as our substations use a SEL derived control system (*Figure 5*) which needs information about the generator speed, voltage and phase angle for synchronising the generator breaker. The other SCADA system that it needs to communicate with is the Delta V SCADA system which controls all the processes across site, the delta v piggy backs off the Mark Vie, this means as well as controlling a gas turbine and all the accessories around it the processor also needs to feed information to two other control systems. ✓

Could have included a diagram showing a SCADA arrangement.



Figure 4 Valero Pembroke MkVIe scada panel for Cogen unit



Figure 5 SEL scada panel for controlling substations

Instructional List:

LD X400
OR Y430
AND M100
AND Y431
OUT Y430

✓

LD X401
OUT T450
K100

✓

LD T450
OUT M100

✓

LD M100
OR Y431
ANI X402
ANI Y430
OUT Y431

✓

LD X403
OR Y432
ANI M101
ANI Y433
OUT Y432

✓

LD X404
OUT T451
K100

✓

LD T451
OUT M101

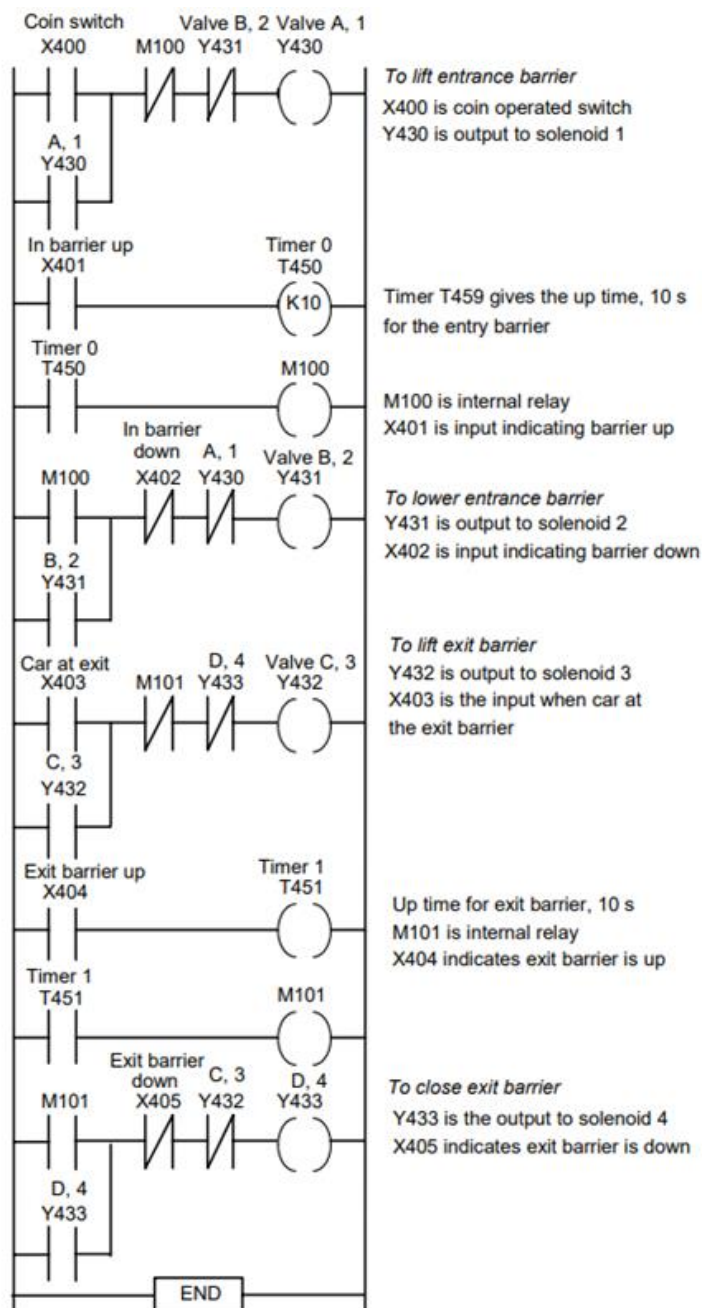
✓

LD M101
OR Y433
ANI X405
ANI X432
OUT Y433

✓

END

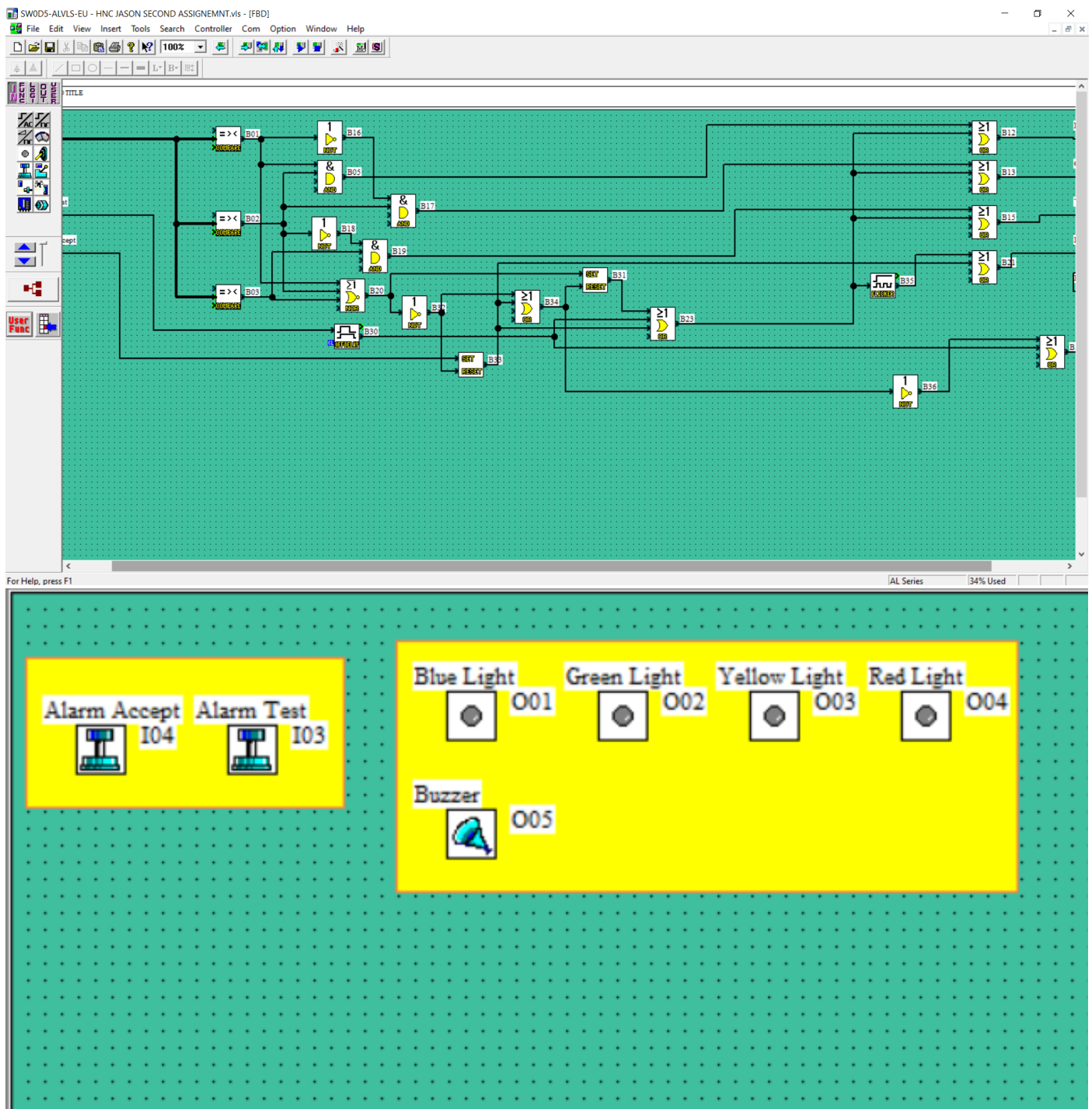
very well done.



Functional Block Program

Overall I am happy with my program as it achieved what it was required, this included the program outputting the correct colour light when in set temperature ranges, an ability to test the function of the lights and buzzer to make sure they are working correctly, I also have to functionality of being able to accept alarms which silences the buzzer and changes the lights from flashing to a steady state. This program also includes a HMI (Human Machine Interface) which allows the user to manual press the buttons and visually see the lights illuminating. If I were to do this program again, I would try and use less logic gates so the program functions faster and takes up less space. ✓

Do not refer to yourself in a report.

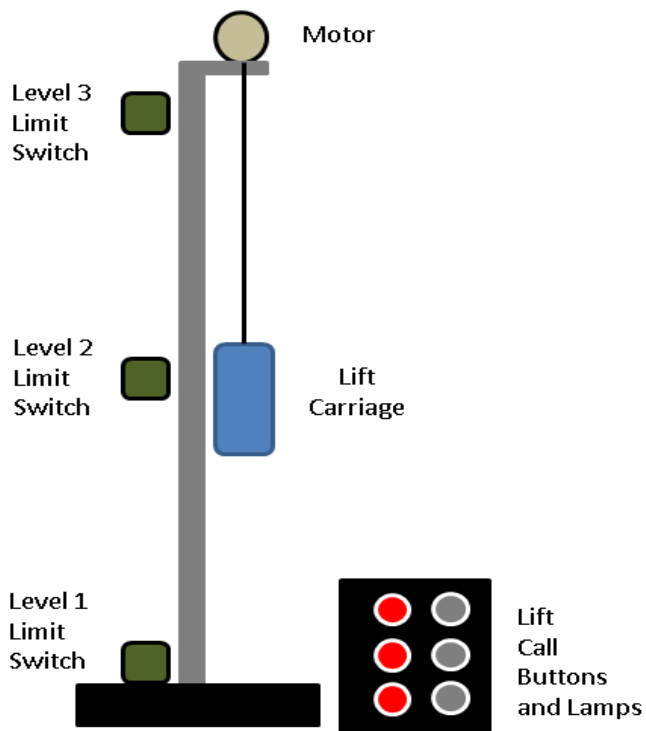


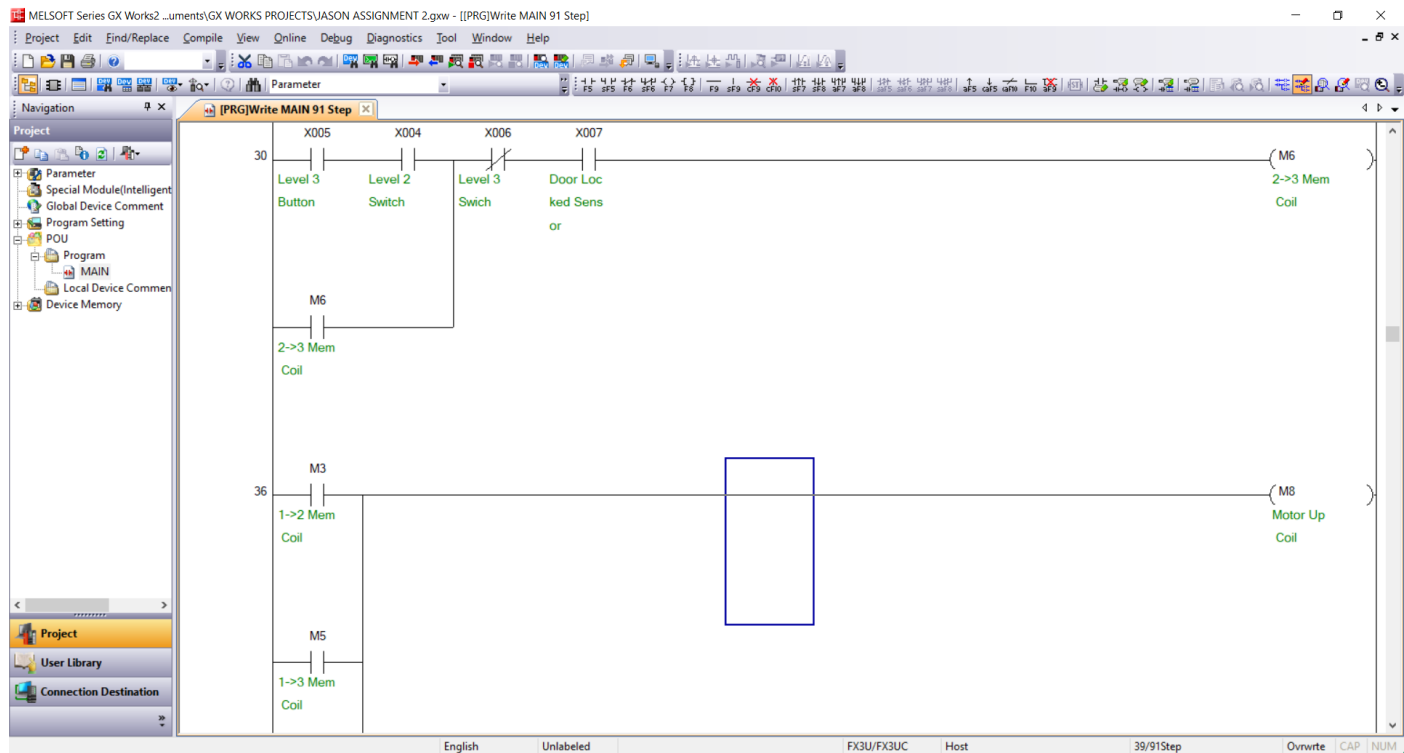
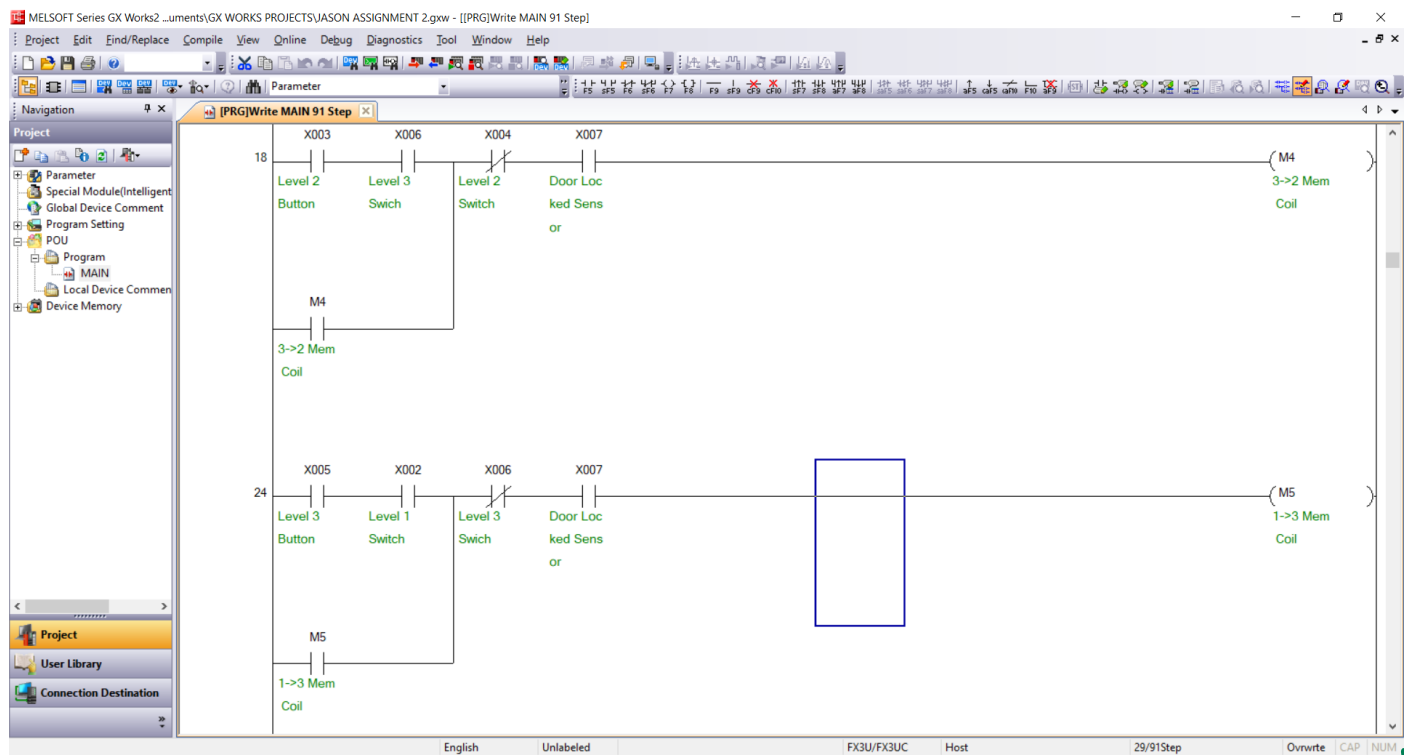
Your screen shot of the program does not show the whole layout. Also you could include some screenshots showing the operation.

I have seen your completed program, layout was correct and the program was fully functional.

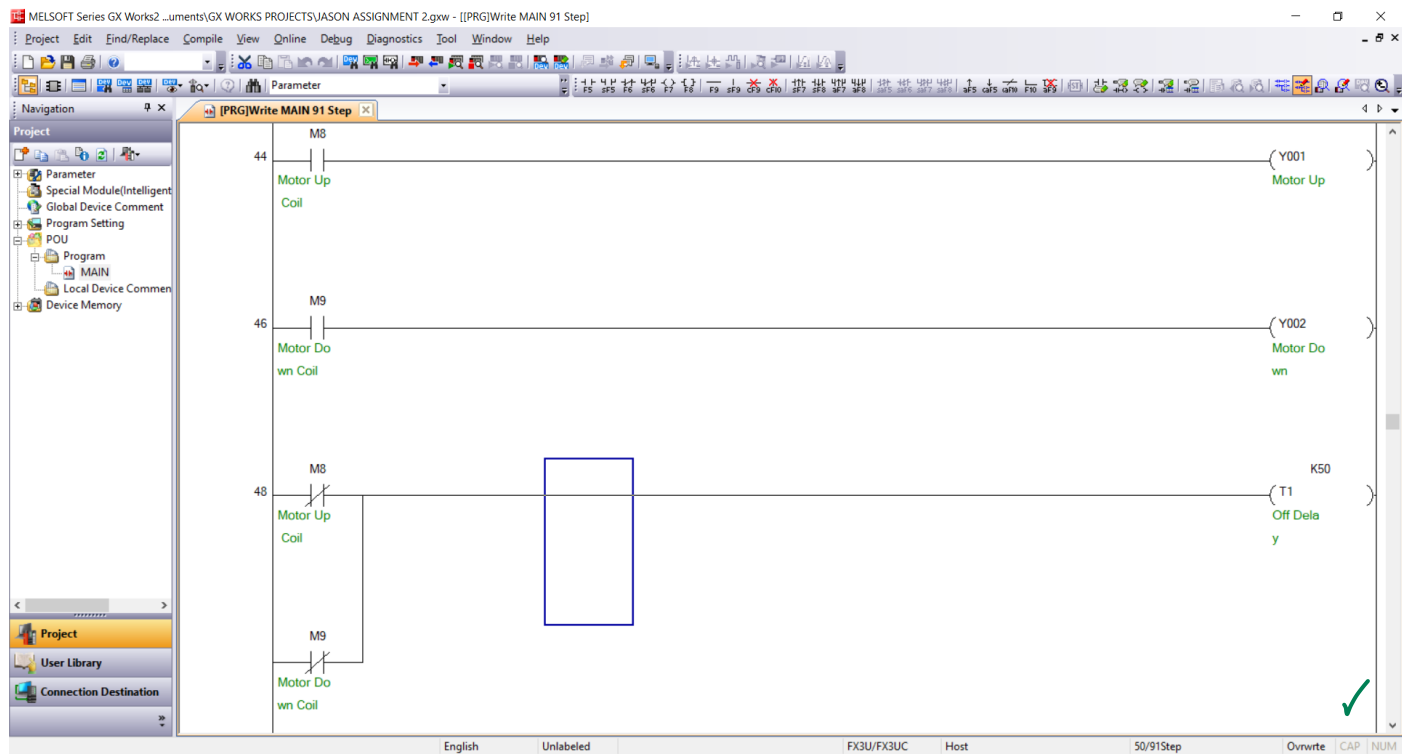
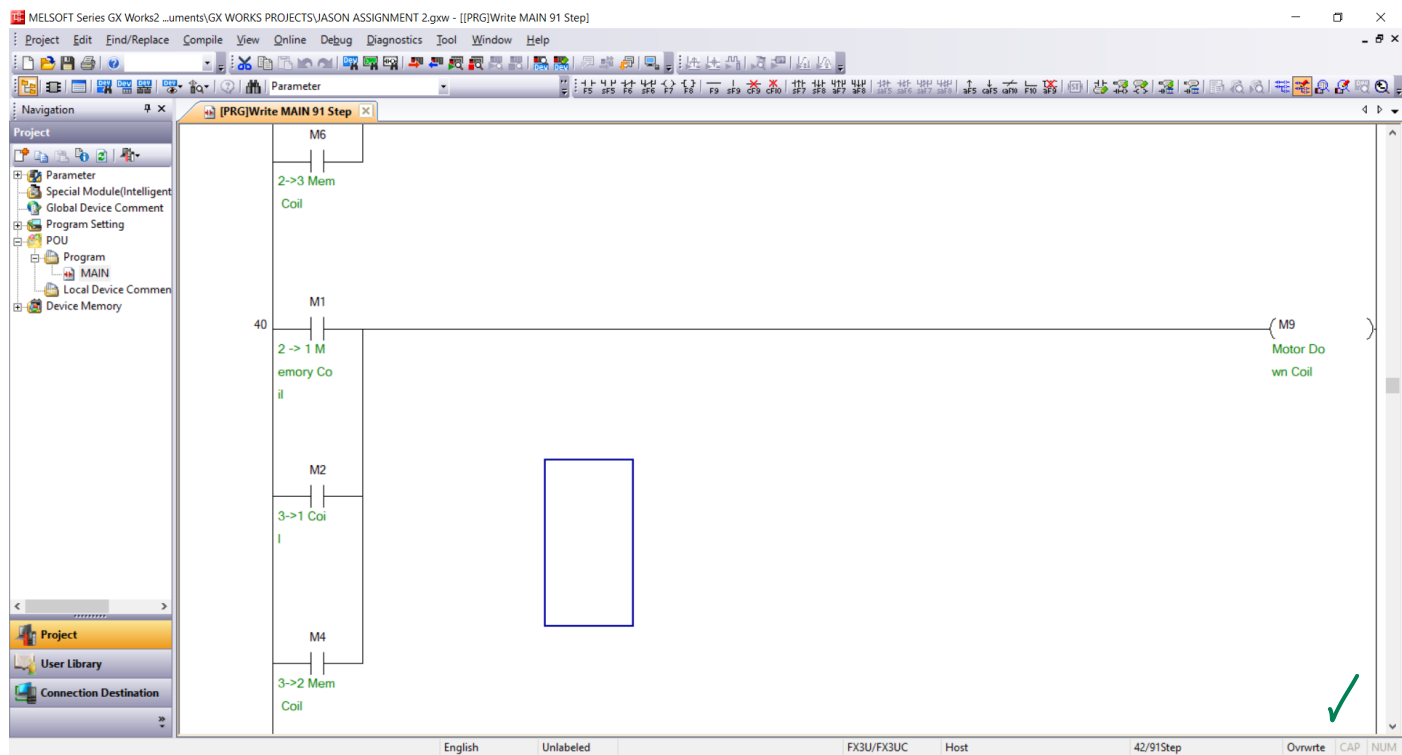
Lift Program

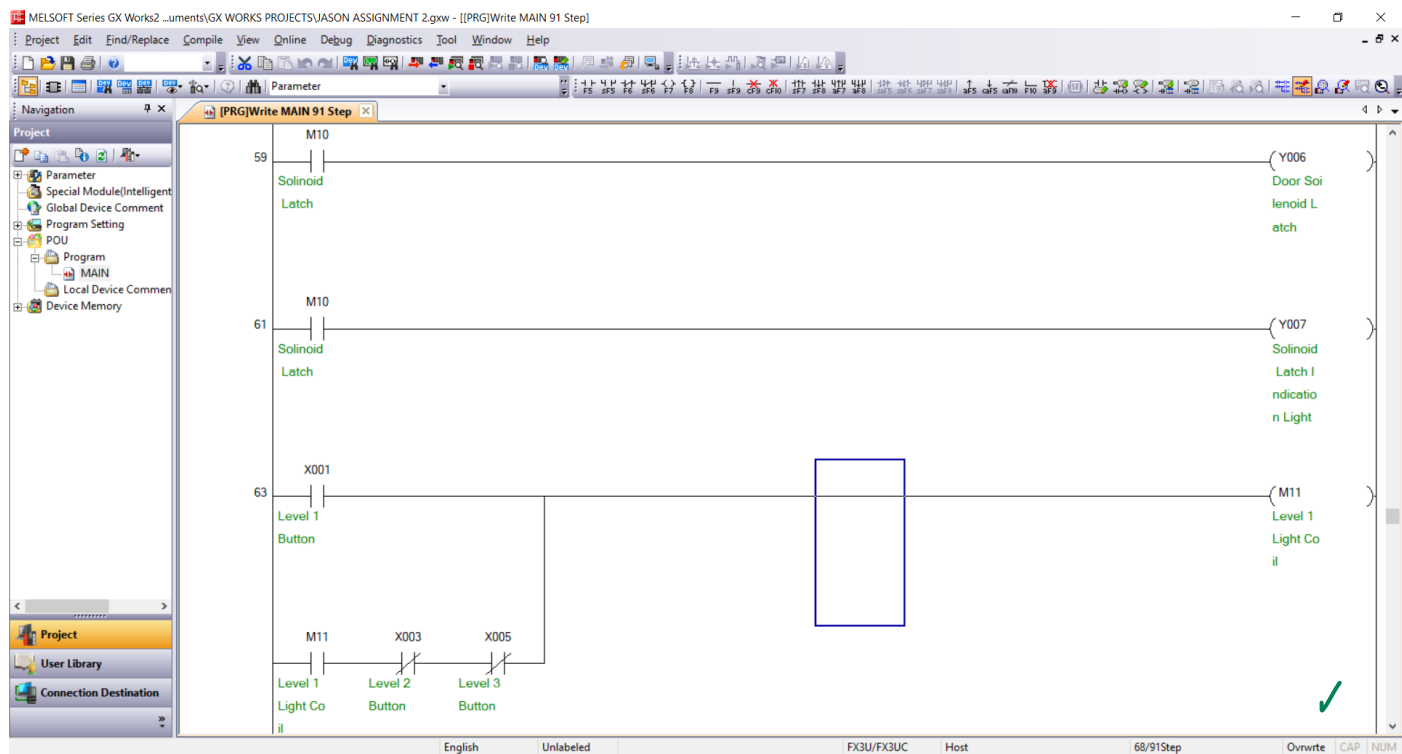
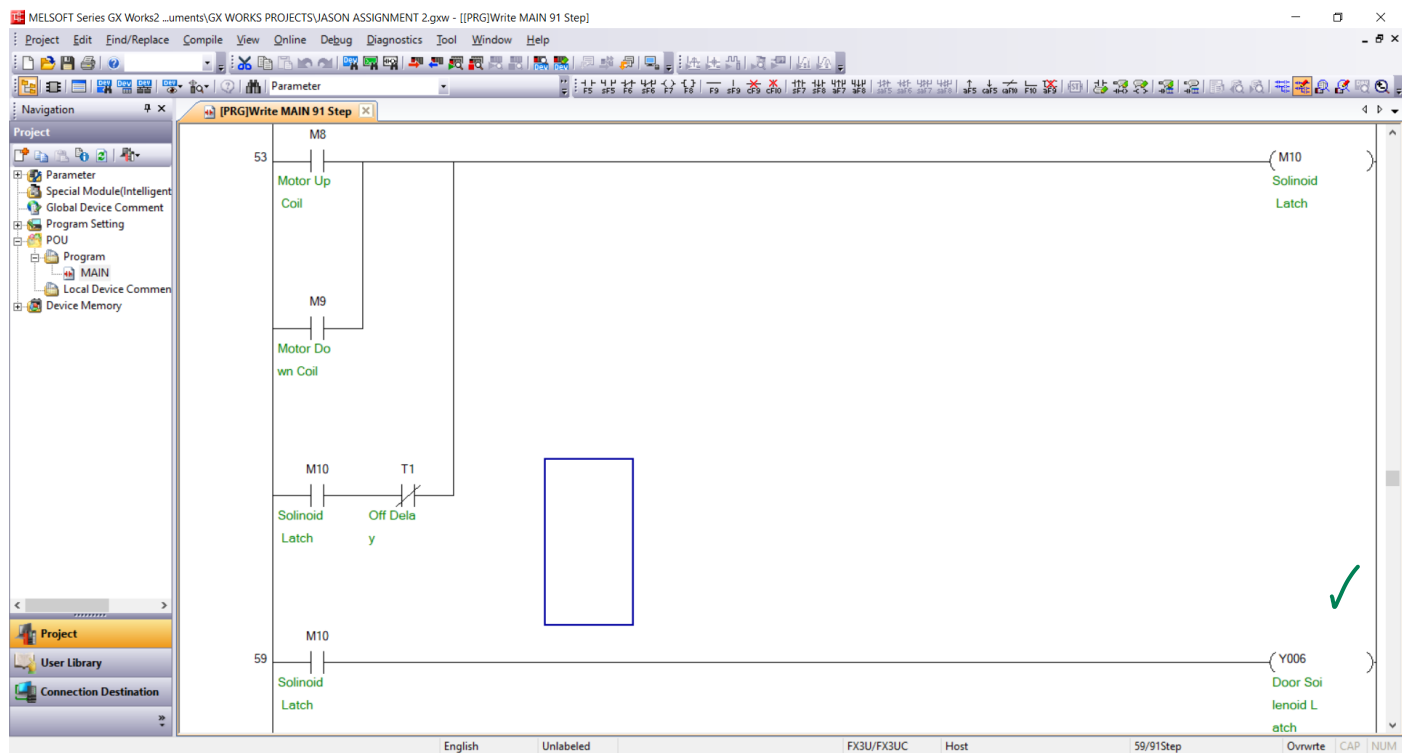
The first requirement for my program was that the lights will need to indicate the position of the elevator when stationary, I achieved this by using the level switches for each level as when one was engaged this would close the contact and illuminate the lamp. By doing it this way it's impossible for two lights to be on at the same time and it also satisfies the second criteria that the light will go off when the elevator leaves a level and turn on the next light. The next requirement for my program is that the door wouldn't open until 5 seconds after the lift had got to its required level, I achieved this by adding a timer to my program which delayed the magnet holding the door closed. The final criteria for my program was that the lift wouldn't operate if the door was open, this was simply achieved by making it so if the door open contact wasn't closed, the start sequence wouldn't start. I am happy with my program as it complies with all the requirements and function correctly but some improvements, I could do are to reduce the amount of rungs used as some of the functions I could have grouped together which would reduce the size of the program which takes up less space and would make it run faster. I could also include some more annotation of the different functions as this would make it easier for someone else to read and understand the program. It would also be a good idea for a mechanical interlock be added to the program so when the lift comes to a level it's not just the motor holding it there. ✓

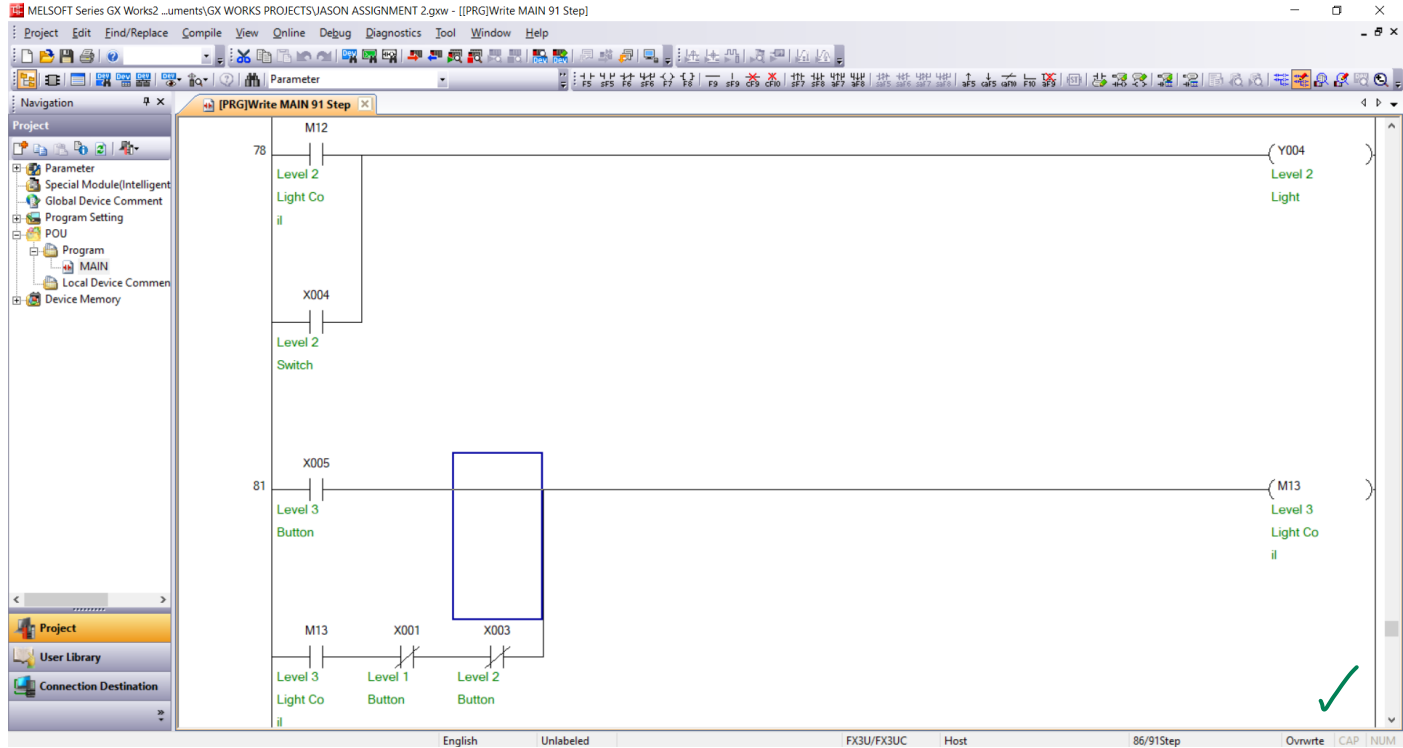
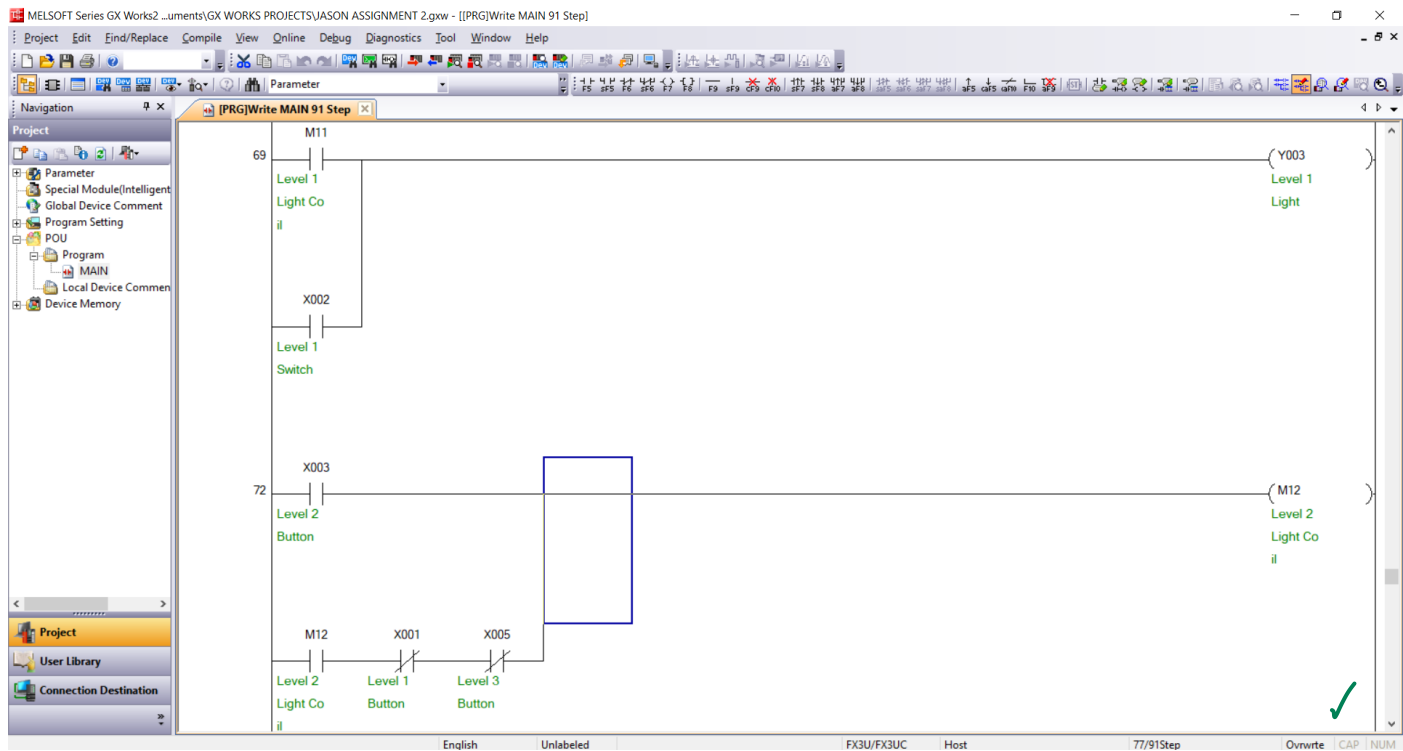


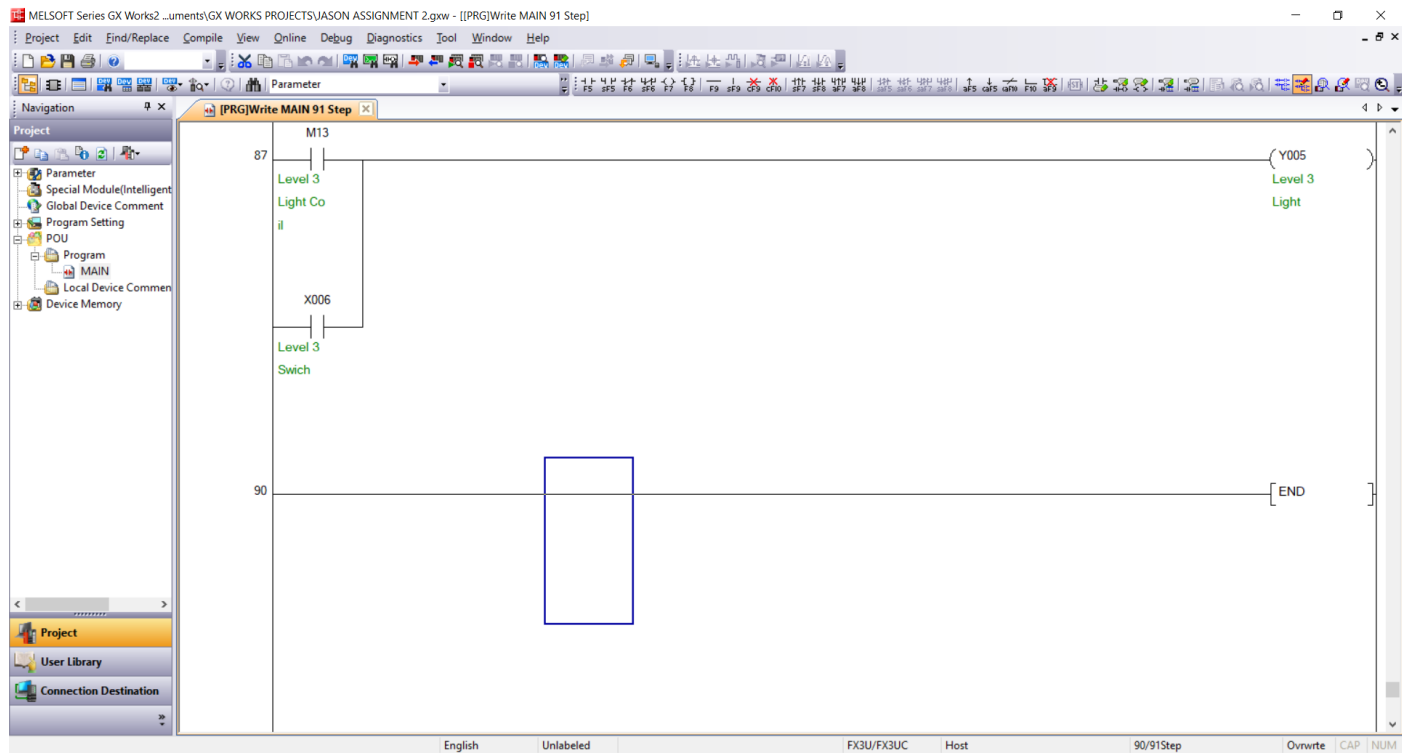


You are missing the beginning of the program.
I can confirm the layout was correct.









If you converted the print of the program to PDF you would have found that easier to copy into your assignment.

Also could include some screen shots of the program in operation.

Bibliography

Robert Radvanovsky (2016). *SCADA/Control Systems Security*. 2nd ed. *: CRC Press. 4-8.

Bill Earl. (2016). *What is a Stepper Motor*. Available: <https://learn.adafruit.com/all-about-stepper-motors>. Last accessed 13/5/2022.

Figure 1

Editorial Staff. (unknown). *Why we use a wheatstone bridge in an RTD?*. Available: <https://instrumentationtools.com/wheatstone-bridge-in-rtd/#:~:text=RTD%20measures%20the%20temperature%20in,is%20called%20the%20excitation%20current.&text=If%20I%20and%20R%20both,voltage%20V%20will%20b>. Last accessed 13/5/2022.

Figure 2

W. Bolton (2015). *Programmable Logic Controllers*. 6th ed. Oxford: Elsevier. 30-63.

Figure 3

Aswinth Raj. (2018). *What is a Stepper Motor and How it Works*. Available: <https://circuitdigest.com/tutorial/what-is-stepper-motor-and-how-it-works>. Last accessed 13/5/2022.

Frank D. Petruzella (2010). *Programmable Logic Controllers*. *: McGraw-Hill Education. *.

Green C (2015). *Learn How to Program and Troubleshoot Ladder Logic*. *: CreateSpace Independent Publishing. *.

Kamel K (2013). *Programable Logic Controllers: Industrial Control*. *: McGraw Hill Education. *.

Mitsubishi Electric (1999). *Programming Manual Simple Application Controller*. Tokyo: Unkno

Please keep everything in the same font.
Some of these are references.