**NIRMA UNIVERSITY**

**INSTITUE OF TECHNOLOGY**

**MECHANICAL ENGINEERING**

**Programmable logic controller (PLC)**

**LAB – 5**

**19BME134**

**Shrey Shah**

**Aim**

Programming of TwinCAT PLC using relay based ladder language.

**Theory**

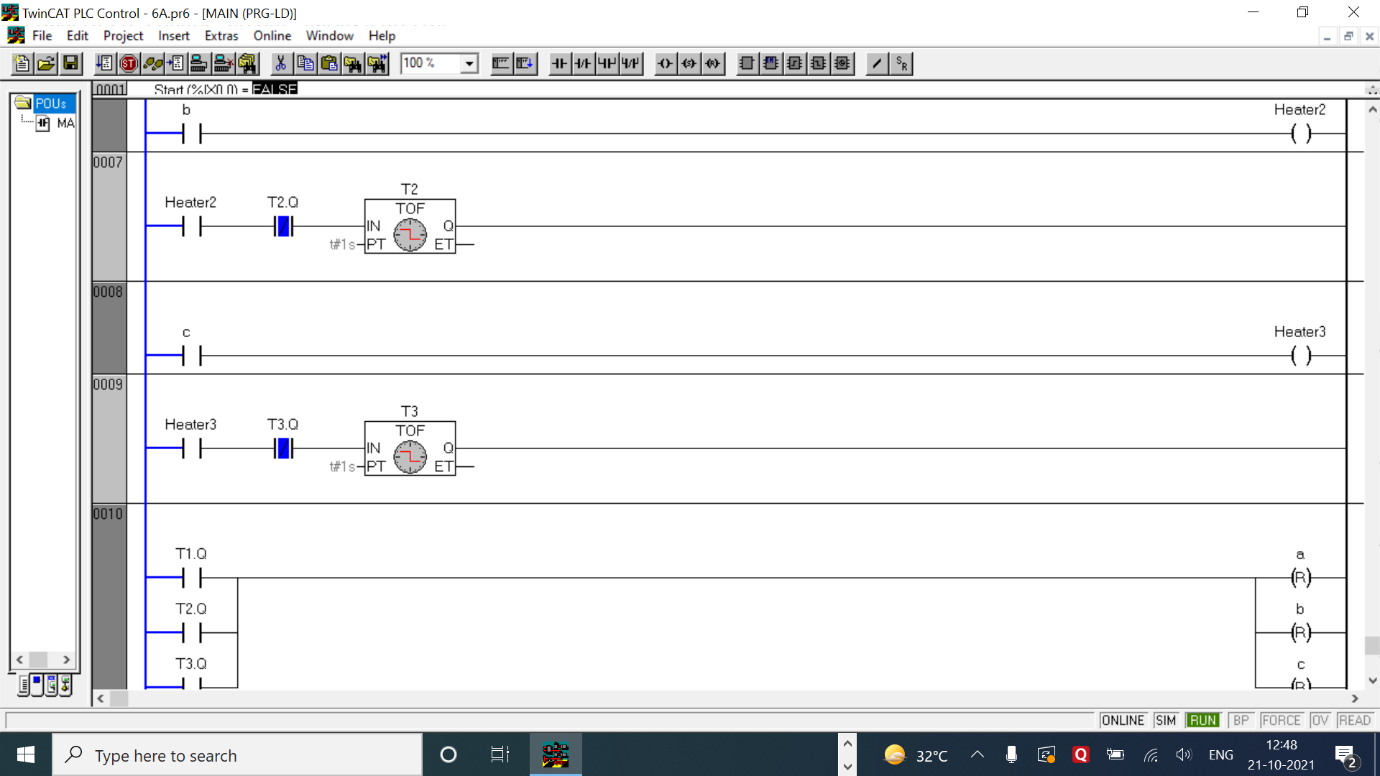
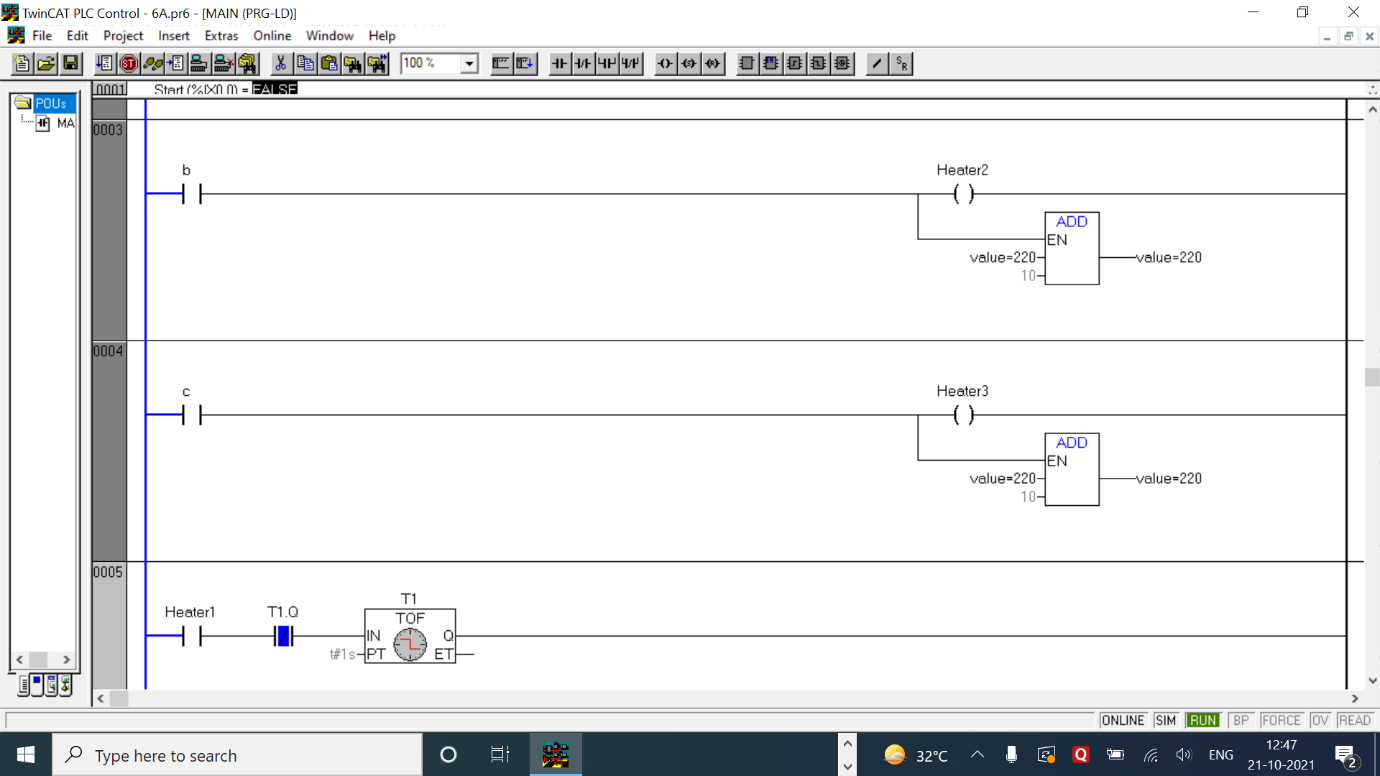
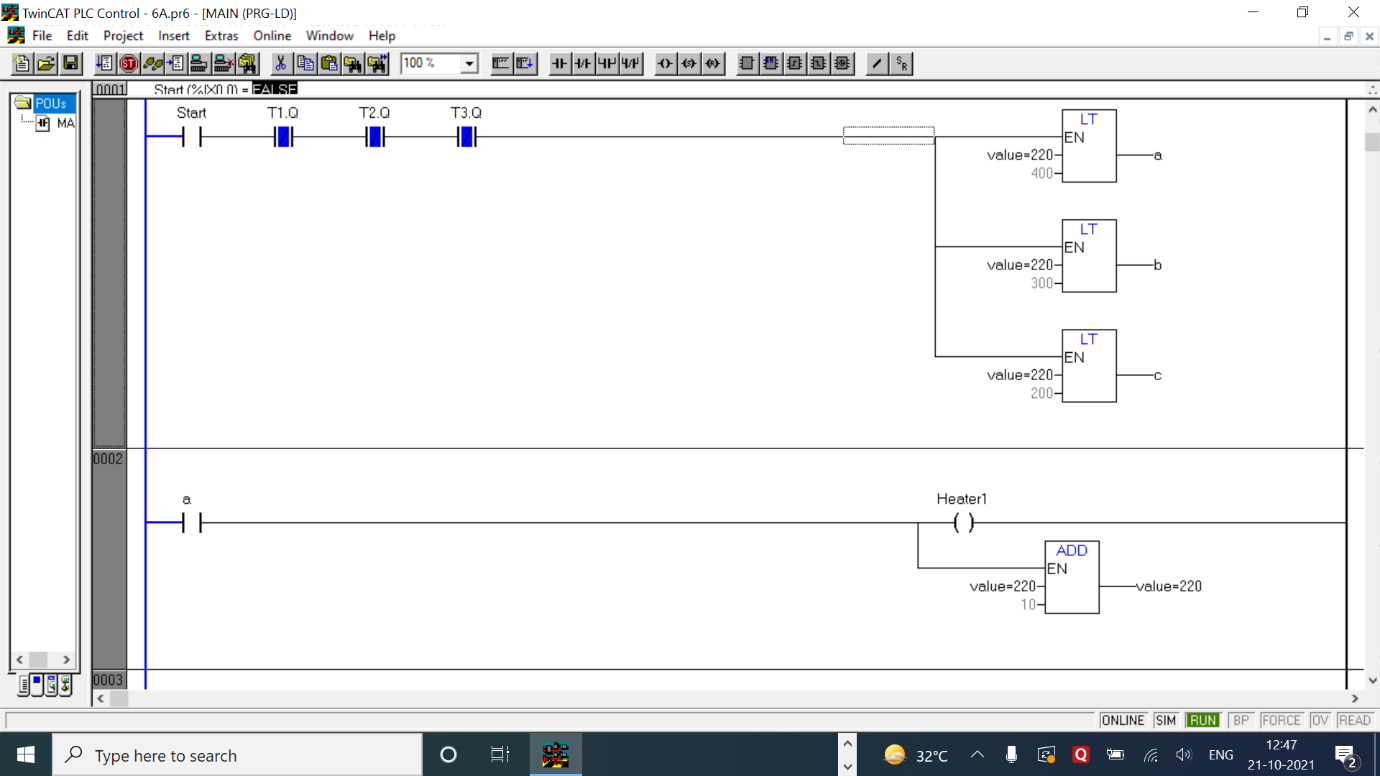
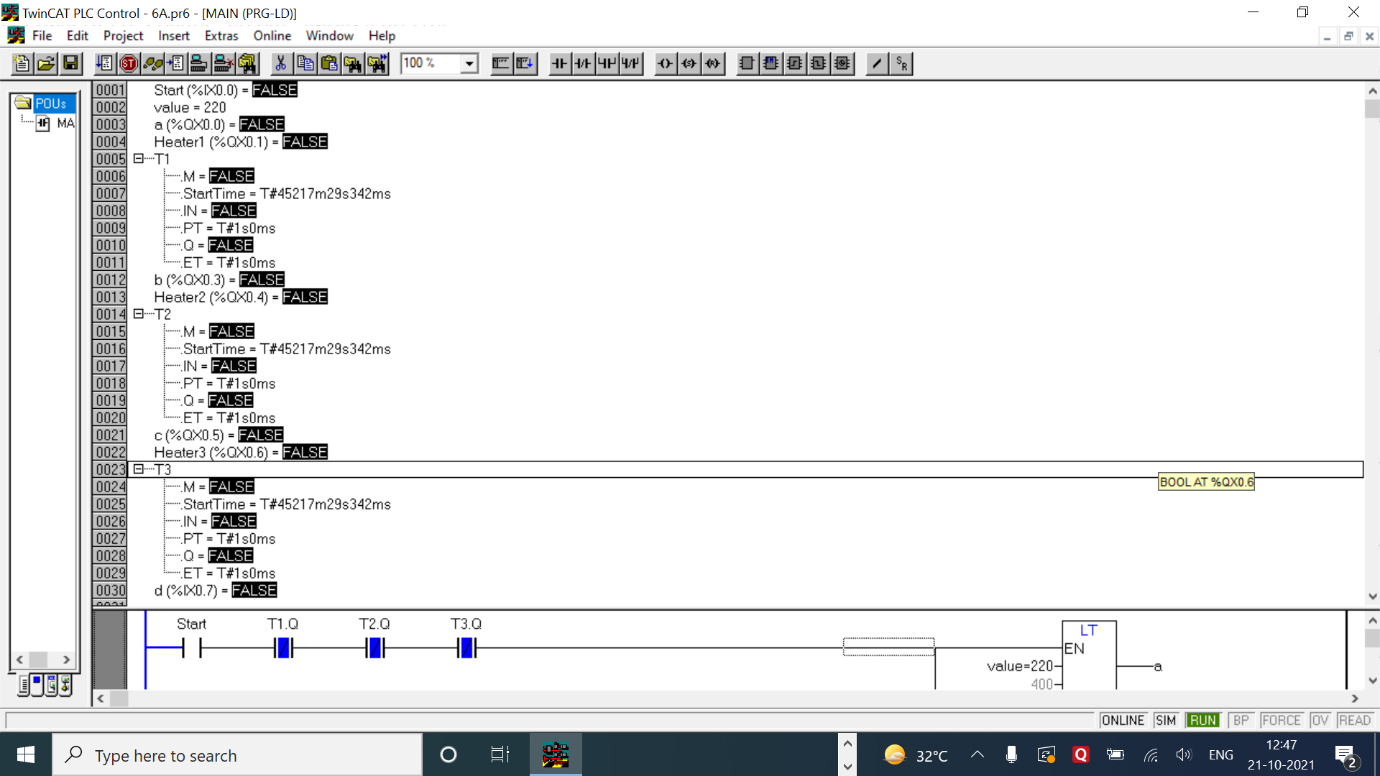
The programmable logic controller (PLC) is a solid state electronic device designed to replace electromechanical relays, timers, counters and sequences, by using a programmable memory for the internal storage of user oriented instructions for implementing specific functions such as logic sequencing timing, country and arithmetic control through digital or analog inputs and outputs, various types of machines or processes.

Major advantages of using PLC are as follows:

1. The PLC is a hardened industrial computer designed to withstand the harsh factory environment.
2. PLCs are reusable they contain a changeable program that eliminators extensive and component changes and that makes them cost effective
3. PLCs offer easy troubleshooting
4. PLCs feature easy installation and small size.
5. Increase productivity.
6. Ease of programming.
7. Ability to communicate with computer.

**Question 1**

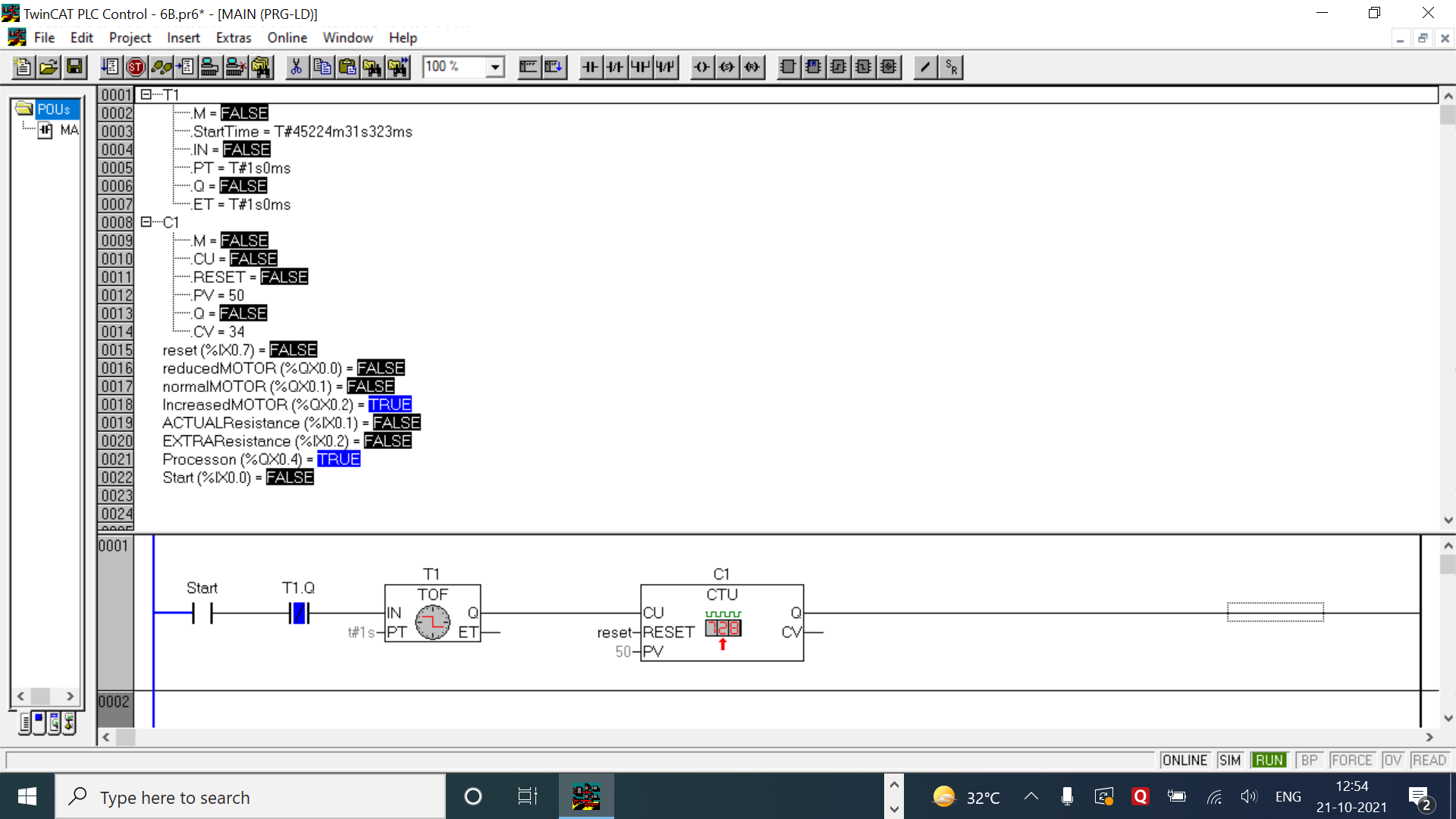
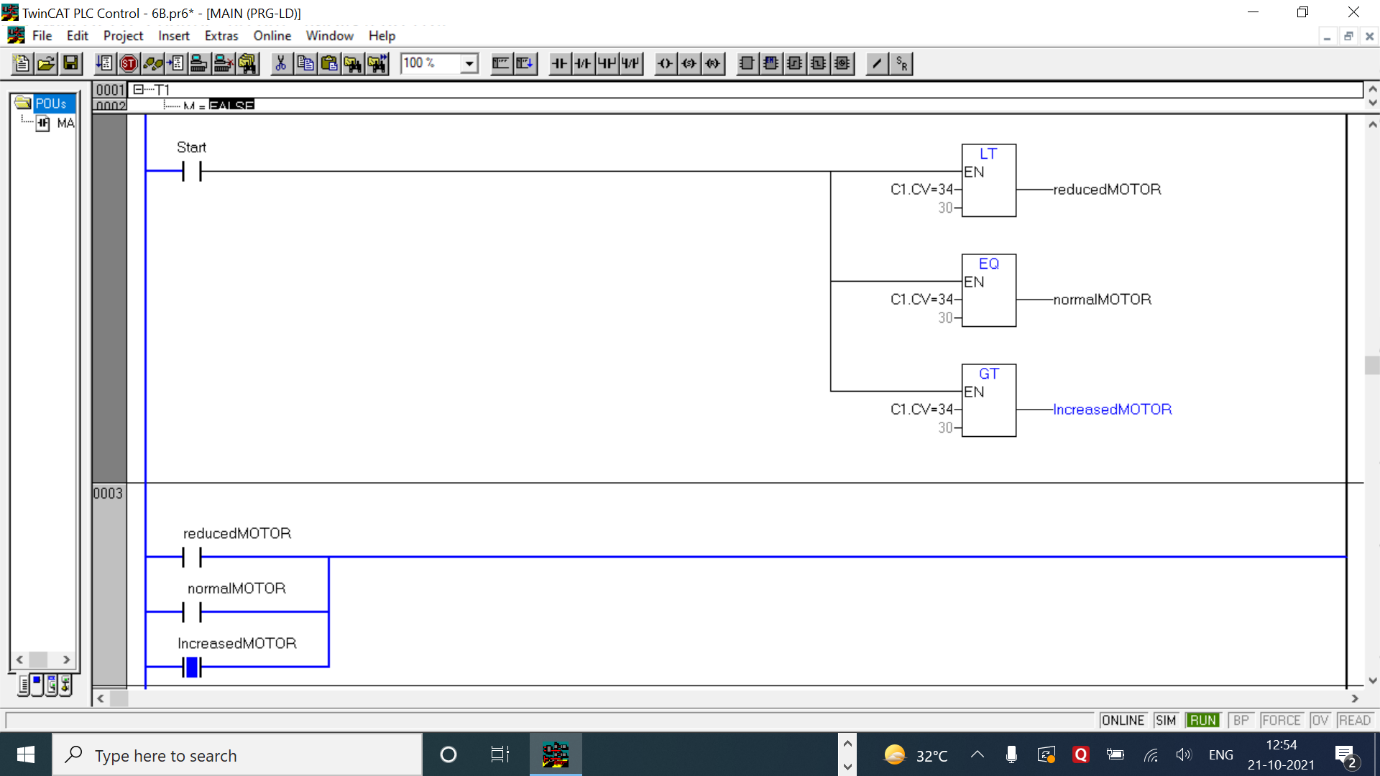
1. Design a ladder program to control three heater elements in a process in a process fluid storage tank used to heat the fluid to 400 C. Assume that the heater contactors are energized at 1- second intervals to produce a smoother heat curve and better energy conversion. Use ON-OFF control to maintain the temperature at the set point.

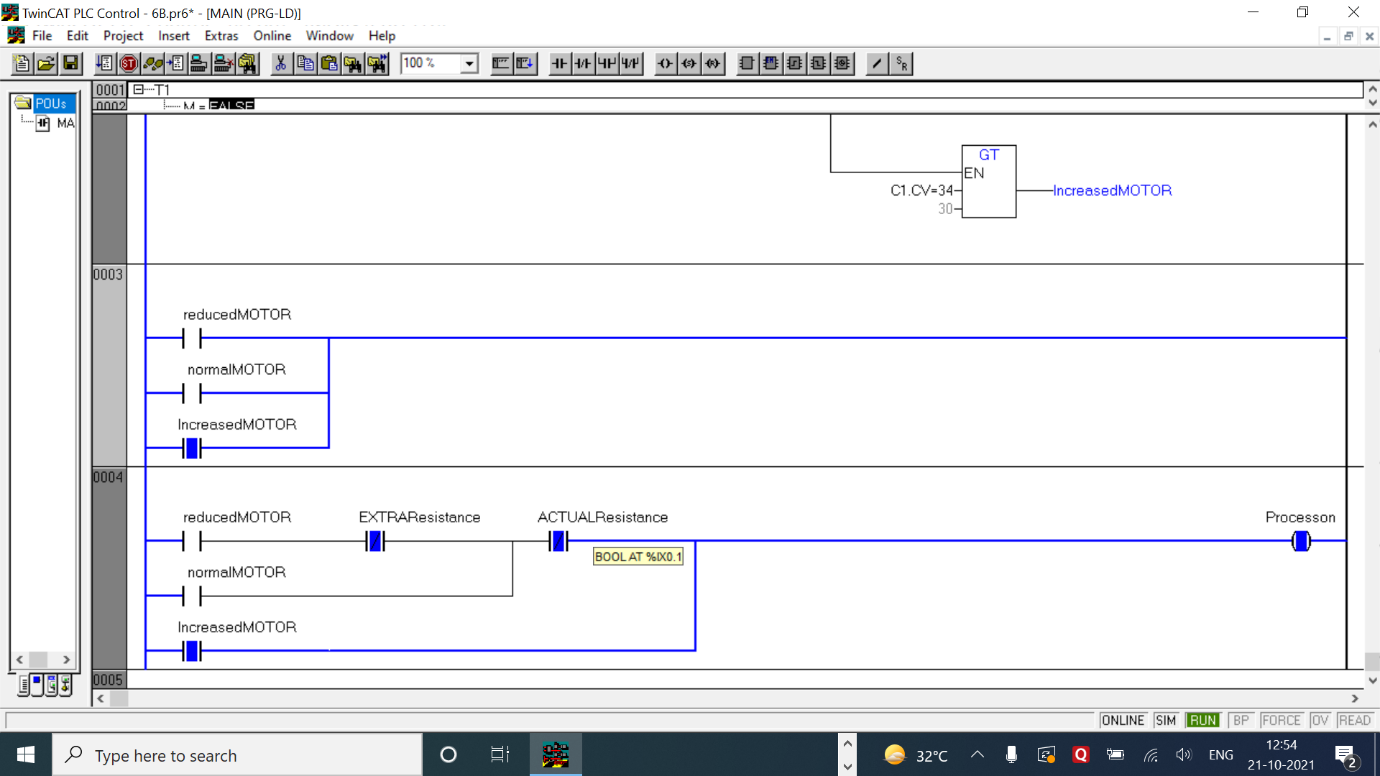


LOGIC -

* In this questions there are 3 heaters which control the heating of the fluid. Each heater has its own heating temperature, i.e, when the fluid reaches the set temperature of the respective heater the heater will turn off.
* In the following ladder logic there are 3 heaters which has the set point of 200 °C. The second heater has the set point of 300 °C. The last heater has the set point of 400 °C.
* Each heater increases the temperature of the fluid by 10 °C which means that below 200 °C, 30 °C are increased at once. Below 300 °C, 20 °C are increased at once. This helps in smoothening the curve and steady heating.
* After pressing Start button 1st rung checks the temperature of the fluid and decides which heaters to be turned on. After the needed heaters are turned on. Each heater has corresponding timer attached to it which regularly updates the settings.
* In the rungs after the heater simple add temperature function block is used to add 10 °C to the current value by each heater.
* After reaching the final value all heaters are turned off

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Start | %ix0.0 | A | %qx0.0 |
| Off Tier t1 | Delay time = 1 sec | Heater1 | %qx0.1 |
| Off Tier t2 | Delay time = 1 sec | Heater2 | %qx0.4 |
| Off Timer t3 | Delay time = 1 sec | Heater3 | %qx0.6 |
| D | %ix0.7 | B | %qx0.3 |
| Value | 100 (int) | T1.Q | Output of t1 |
| ADD function block | Value + 10 | T2.Q | Output of t2 |
| (LT function block) x 3 | Value < 200, 300, 400 | T3.Q | Output of t3 |
| - | - | C | %qx0.5 |
|  | - | Reset coils for A, B, C | A, B, C |

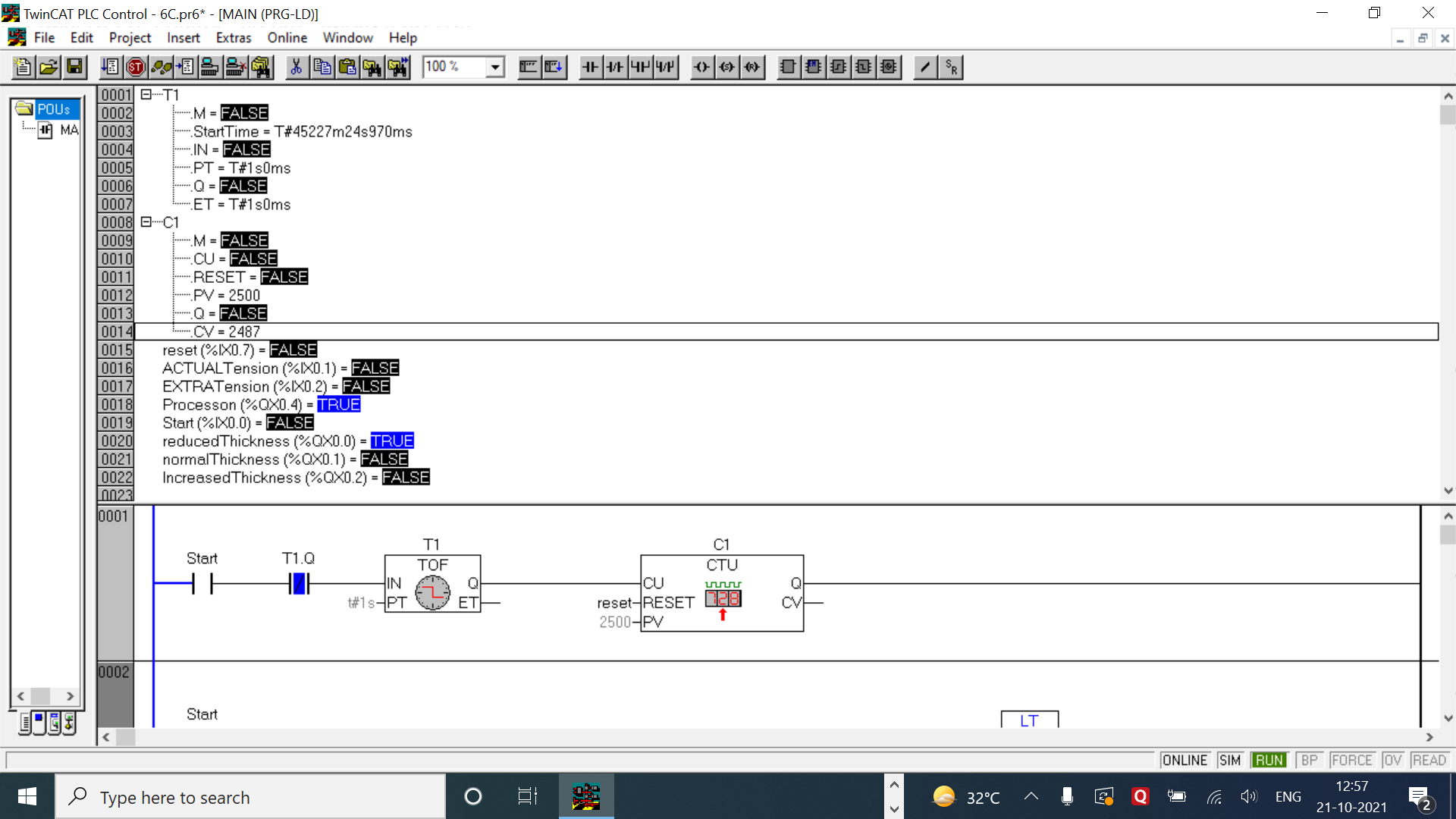
1. A conveyor system has a weight detector and three speed motor. The system is designed to move the conveyor belt at a certain speed when a specific value of weight is on the conveyor. If the weight exceed the preset value, the conveyor speed increases to move the material off the belt at a faster rate. If the weight flow levels below the process value, the conveyor speed is reduce. When output 01000 is energized the motor run at normal speed. When 01000 is ON, resistance is added to DC motor winding to slow the conveyor down. When 01002 is energized, resistance is removed from the winding to speed up the motor. Write a ladder logic program to automatically change the speed of the motor.

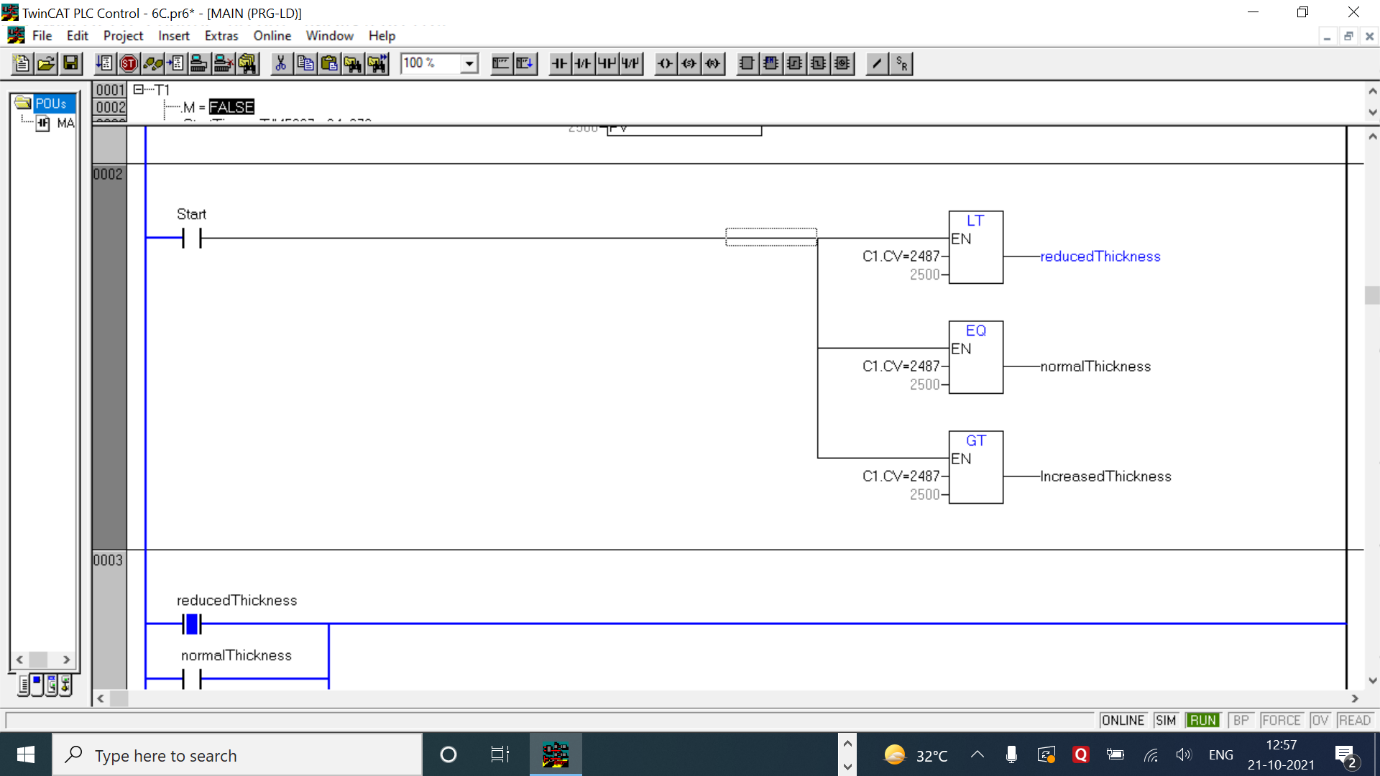
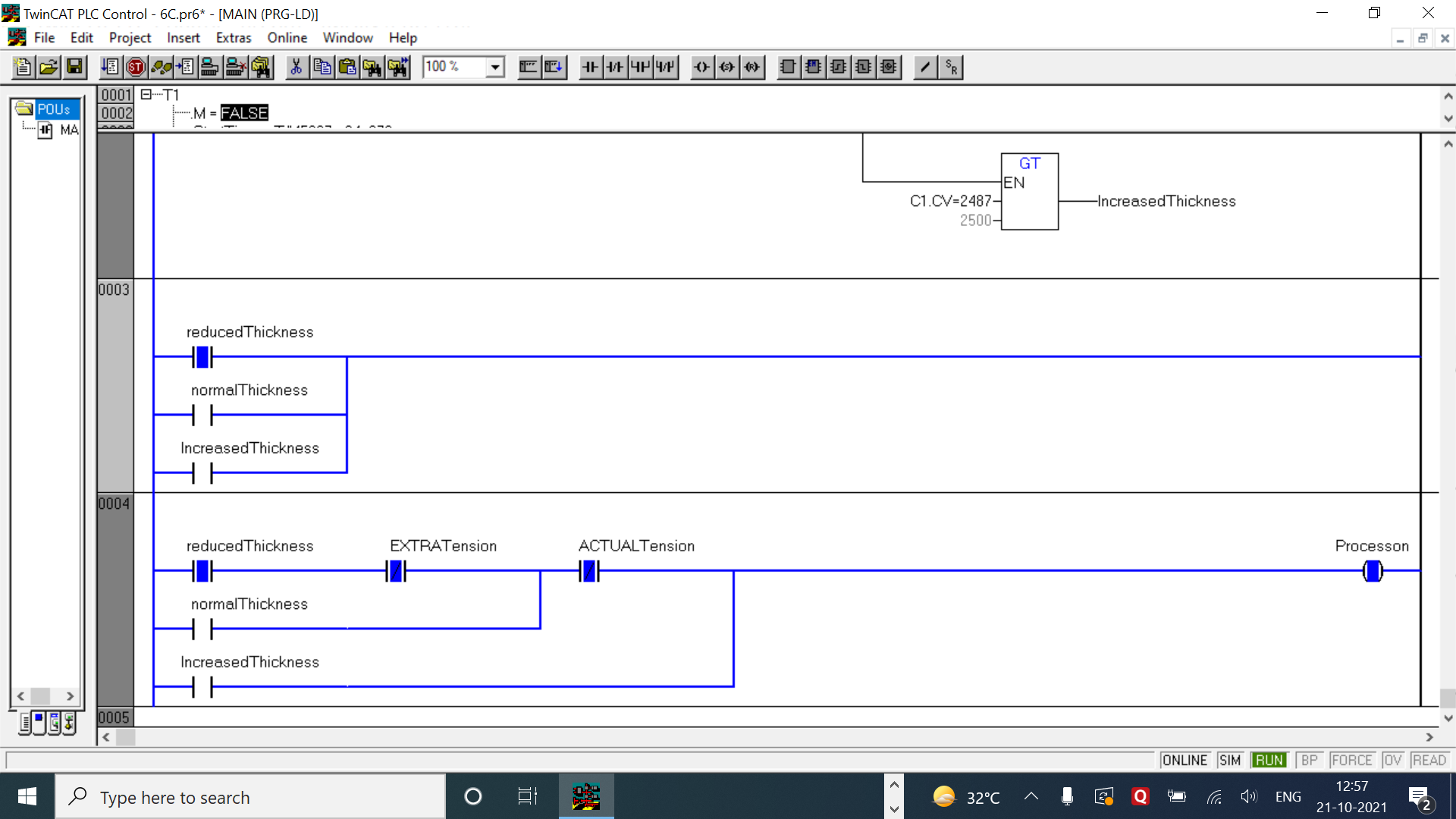


LOGIC

* Here the logic is very simple. 3 conditions are given to control the speed of the motor for reduced speed, normal speed and increased speed.
* After starting the process the function blocks check whether the weight is less than, more than or equal to 30kg. After determining the state it proceeds to use the corresponding motor.
* To use the same motor, resistance can be added or removed to increase or decrease speed. The last rung shows that by adding a resistance we can decrease the speed and by removing the actual resistance along with the extra resistance speed can be increased. The process ladder logic is demonstrated below.
* For the sake of this experiment I have included an automated counter with a timer which sends the weight of the object increasingly.

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Start | %ix0.0 | Reduced Motor | %qx0.0 |
| Actual Resistance | %ix0.1 | Normal Motor | %qx0.1 |
| Extra Resistance | %ix0.2 | Increased Motor | %qx0.2 |
| Reset | %ix0.7 | Process On | %qx0.4 |
| Up Counter C1 | Preset value = 50  RESET = Reset | C1.CV | Current value of C1 |
| Off Timer t1 | Delay time = 1 sec | T1.Q | Output of t1 |
| LT function block | C1.CV, 30 | - | - |
| EQ function block | C1.CV, 30 | - | - |
| GT function blocl | C1.CV, 30 | - | - |

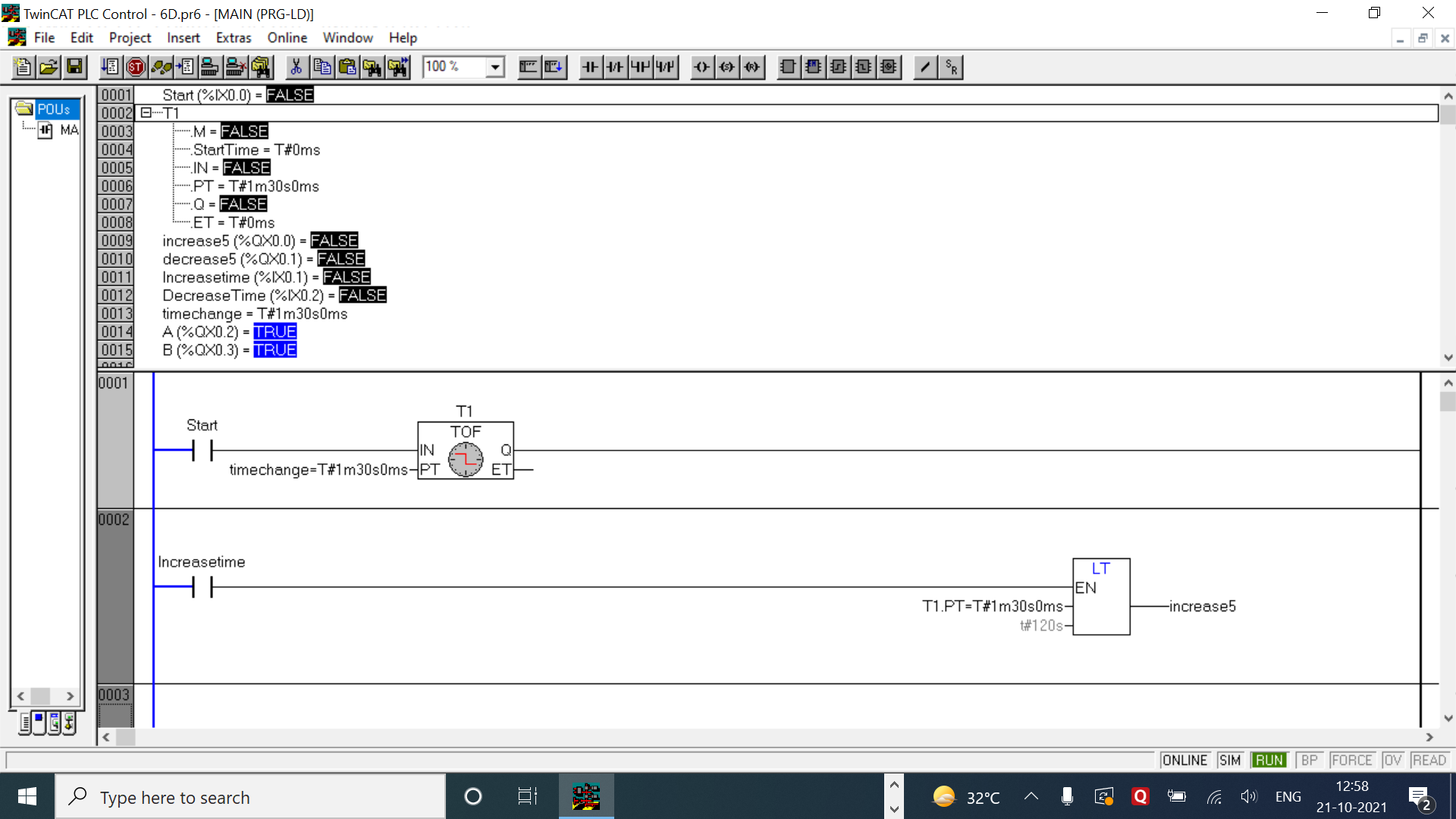
1. A part thickness controlling system uses PLC to compare the thickness of the wet fibre to constant stored in memory. If the obtain by the measurement transducer is greater than the set point, output 01001 is energized and the tension between the movable roller and large fixed roller is increased. If the input data is less than the set point, output 01002 is energized and the tension on the conveyor is reduced. Assume 2500 is the set point for data comparison. Use to hold the set point. Write a ladder diagram for this.

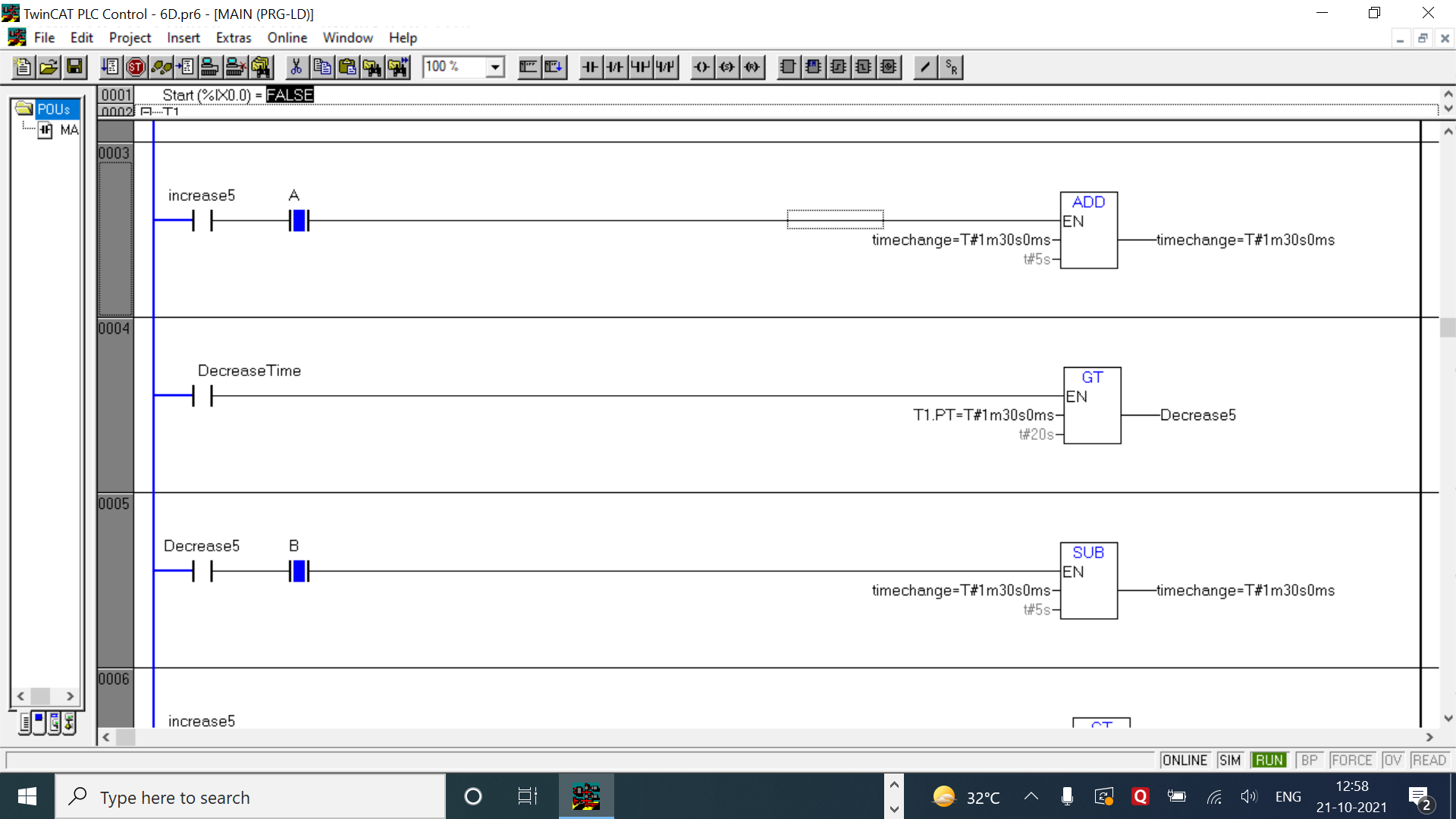


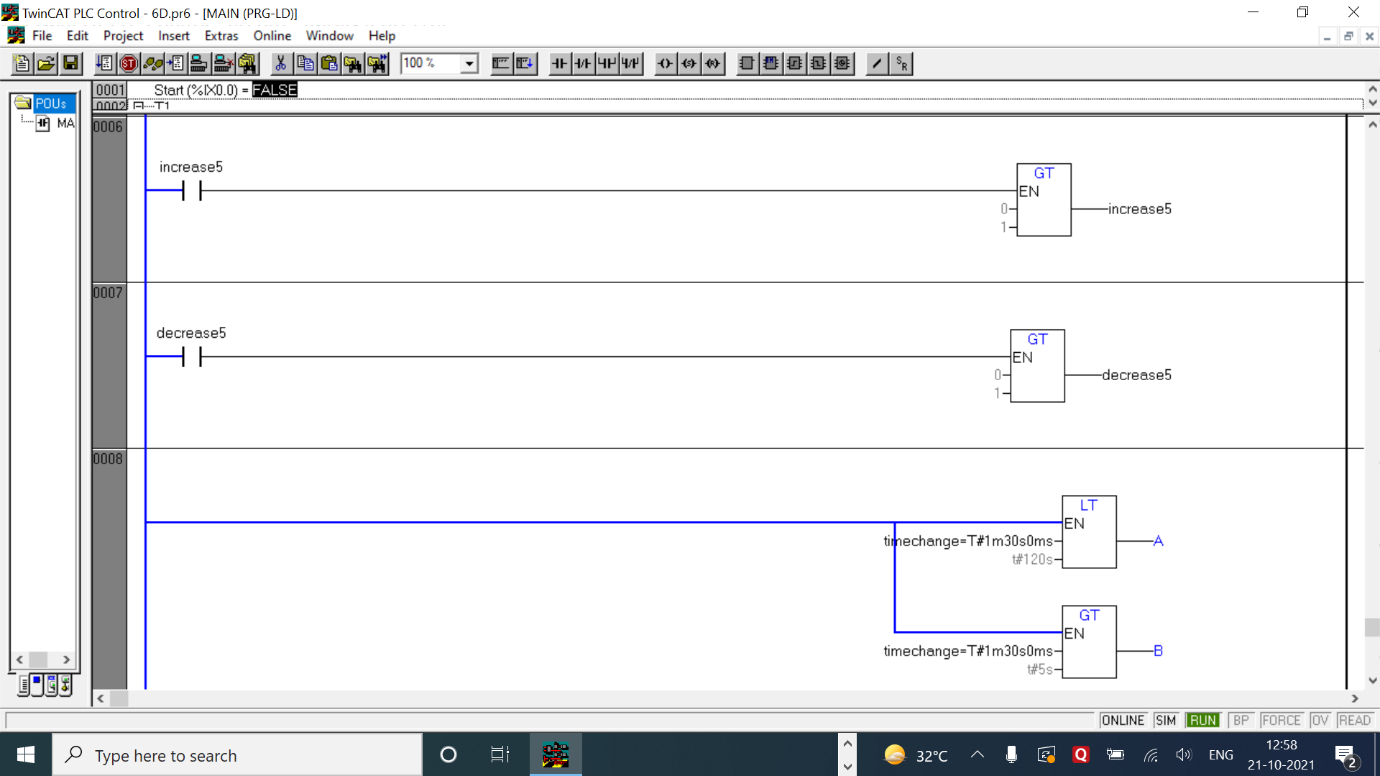
LOGIC

* The logic in this question is extremely similar to the previous question, so much that same ladder logic can be used to solve this question.
* The difference in this question is that instead of weight, we are currently measuring the thickness of the fibre stored in memory so if the fibre thickness is less the tension is reduced and if fibbed thickness is more the tension is increased thus stretching the fibre.
* In this way the control of the thickness of the fibre can be maintained.

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Start | %ix0.0 | Reduced Thickness | %qx0.0 |
| Actual Tension | %ix0.1 | Normal Thickness | %qx0.1 |
| Extra Resistance | %ix0.2 | Increased Thickness | %qx0.2 |
| Reset | %ix0.7 | Process On | %qx0.4 |
| Up Counter C1 | Preset value = 3000  RESET = Reset | C1.CV | Current value of C1 |
| Off Timer t1 | Delay time = 1 sec | T1.Q | Output of t1 |
| LT function block | C1.CV, 2500 | - | - |
| EQ function block | C1.CV, 2500 | - | - |
| GT function blocl | C1.CV, 2500 | - | - |

1. D Write a program for adjustable timer whenever increase Push Button is pressed timer value is increased by 5 sec it is not more than 120 sec. whenever decrease push button is pressed timer value is decreased by 5 Sec provided timer value is not less than 20 sec.



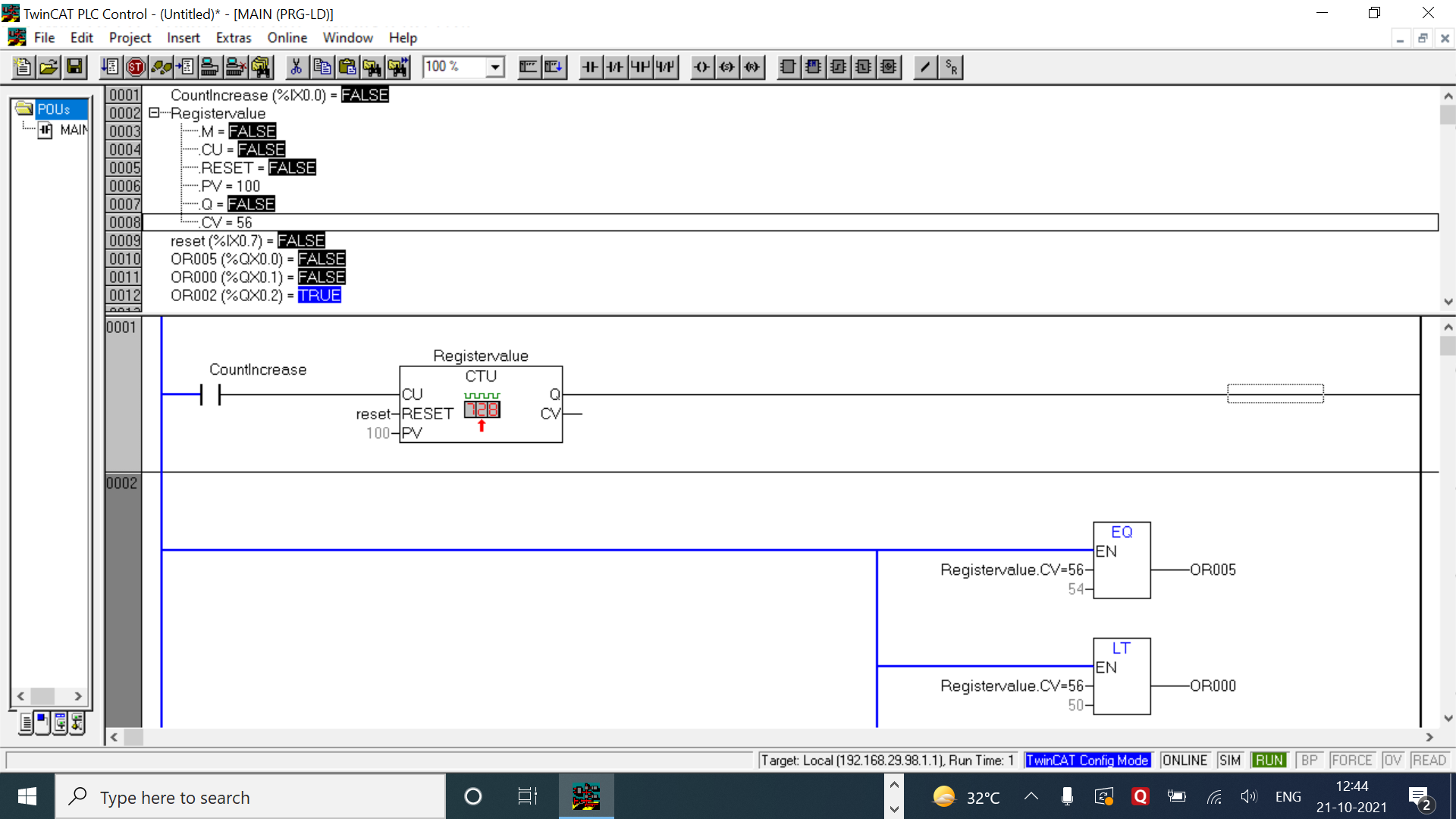


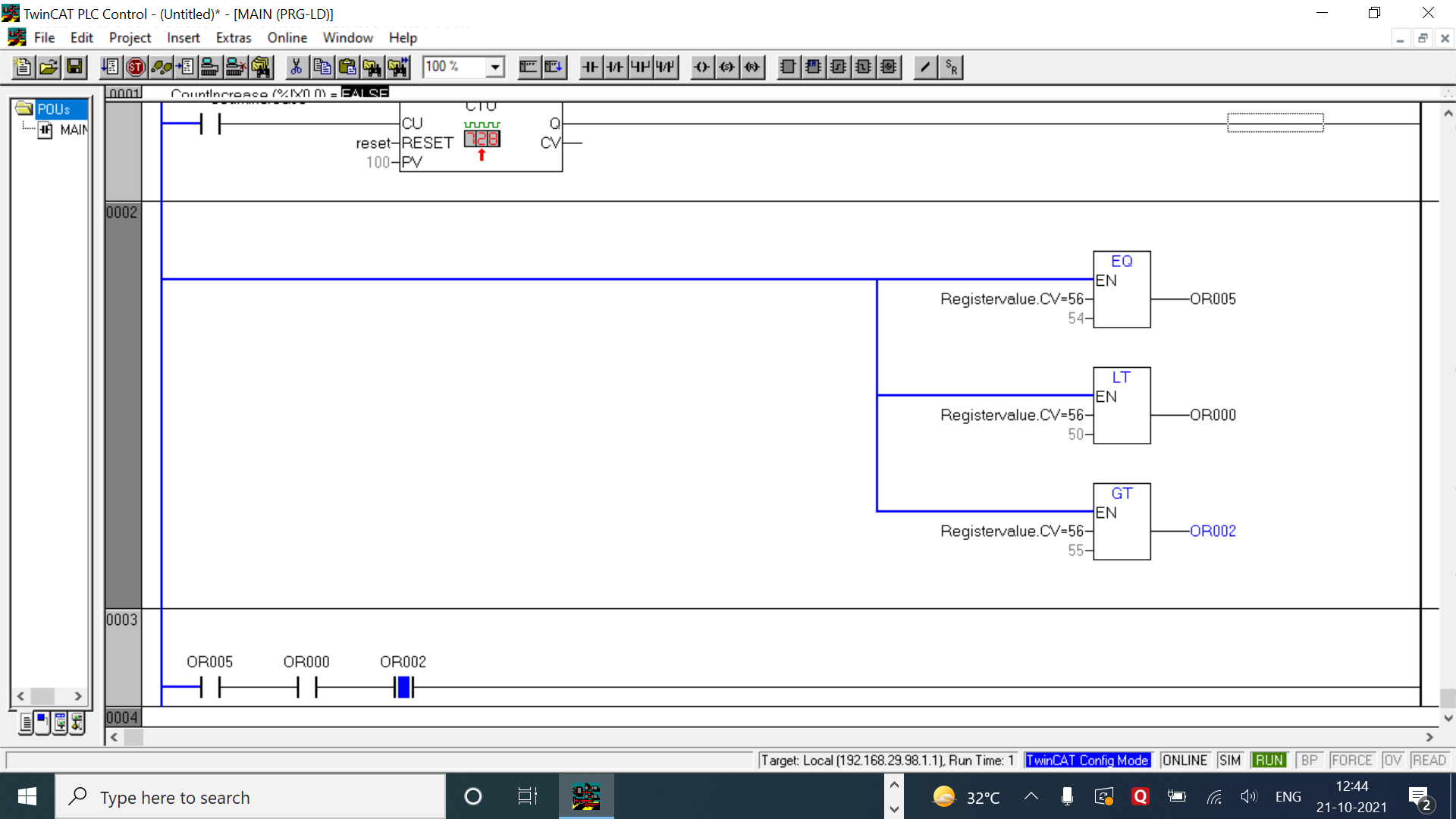
LOGIC

* In this question I have added 2 types of switches for both increasing and decreasing the tine of the timer - The first case will automatically adjust the time of the timer to 120s or 20s whichever button is pressed while in the other case, upon pressing the pushbutton the timer will only increase or decrease 5 seconds at a time.
* If Increasetime or Decreasetime buttons are pressed then a function block is first kept which checks if the time is smaller than 120s or greater than 20s.
* After checking if the output is true then increase5 or decrease5 will be enabled which connects to a function block which continuously adds 5 seconds until increase5/decrease5 is turned off.
* The above set up is used for automatic increase or decrease.
* For manual increase decrease of 5 seconds at a time, the last rung only has function blocks which again checks if the value of time falls in the range or not.
* The outputs of the above mentioned comparison is added after the increase5 and decrease5 switch which stays on until the condition is true.
* As we know the fault of comparison operators which is that they will remain true until checked again or turned off manually. So to avoid this 2 rungs have been used as a method to turn off the on true status of increase and decrease outputs.
* So after each execution, it first becomes true then proceeds to increase 5 seconds but at the same time in the above mentioned extra rungs it does not fulfill the condition and hence again becomes false hence breaking the input acting like a push button.
* NOTE – this weird and time consuming set up is required because if the comparison condition stays on then the add function will indefinitely increase the value of time until reaches 120s or whatever is the limit.

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Start | %ix0.0 | Increase 5 | %qx0.0 |
| Increase Time | %ix0.1 | Decrease 5 | %qx0.1 |
| Decrease Time | %ix0.2 | A | %qx0.2 |
| Time change | T#5s (time variable) | B | %qx0.3 |
| Off timer T1 | Delay time =  Time change | C2.CV | Current value of C3 |
| LT function block | Time change, t#120s | (GT function block) x 2 | 0, 1 |
| GT function block | Time change, t#20s | SUB function block | Time change, t#5s |
| ADD function block | Time change, t#5s | - | - |

1. The OR 005 is set ON when the holding register value is 54. If the value in the register is less than 50 then OR 000 is ON. If the value in the register is greater than 55 then OR 002 is ON.

****

****

LOGIC

* This is a very simple and basic program where a counter is used to count the value of register. In the 2nd rung 3 function blocks are used for the 3 conditions given in the question for the register count to be less than 50, equal to 54 and greater than 55.
* Since theses function blocks have no input in their rung, it means that they check the conditions indefinitely and precisely.
* In the third and last rung the outputs of the comparison function blocks are used as input switches as a indicator to show which one of the conditions is being fulfilled now.

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Count Increase | %ix0.0 | OR005 | %qx0.0 |
| Register Value | Preset value = 100  RESET = Reset | OR000 | %qx0.1 |
| Reset | %ix0.7 | OR002 | %qx0.2 |
| LT function block | RegisterValue.CV, 50 | RegisterValue.CV | Current count of Register |
| GT function block | RegisterValue.CV, 55 | - | - |
| EQ function block | RegisterValue.CV, 54 | - | - |

COMMENTS –

* The counters in most of the questions are denoted by the letter ‘C’ followed by the number of the counter or the letter.
* Cn.q represents the output of those counters which may be taken as NO or NC switches.
* Each question has a table of inputs and outputs which specifies which I/Os have been taken along with its addresses.
* All the timers are generally denoted by the symbol ‘tn’ where n represents the number of the timer.
* tn.q represents the output of the timer tn which can be both normally open or normally closed depending on how it is used based on the. Question’s requirements.
* The functions of the function blocks can be interpreted using the short abbreviation at the top of the block.
* Both mathematical and comparison operators are used in this experiment as function blocks with EN.

**CONCLUSION**

* In this experiment we learnt about different types of mathematical operations and comparisons that could be performed in TwinCAT using ‘Box with EN’ function block. Some of the types of operations include – Greater than, less than, equal to, greater than equal to, less than equal to and many more. We learned the importance of using these operation in real life as counts and remaining products are needed to be kept in check. Comparators are very important in plc as they can also be used/ interpreted as if/else statements if used properly and wisely.