**NIRMA UNIVERSITY**

**INSTITUE OF TECHNOLOGY**

**MECHANICAL ENGINEERING**

**Programmable logic controller (PLC)**

**LAB – 2**

**19BME134**

**Shrey Shah**

**Aim**

Programming of TwinCAT PLC using relay based ladder language.

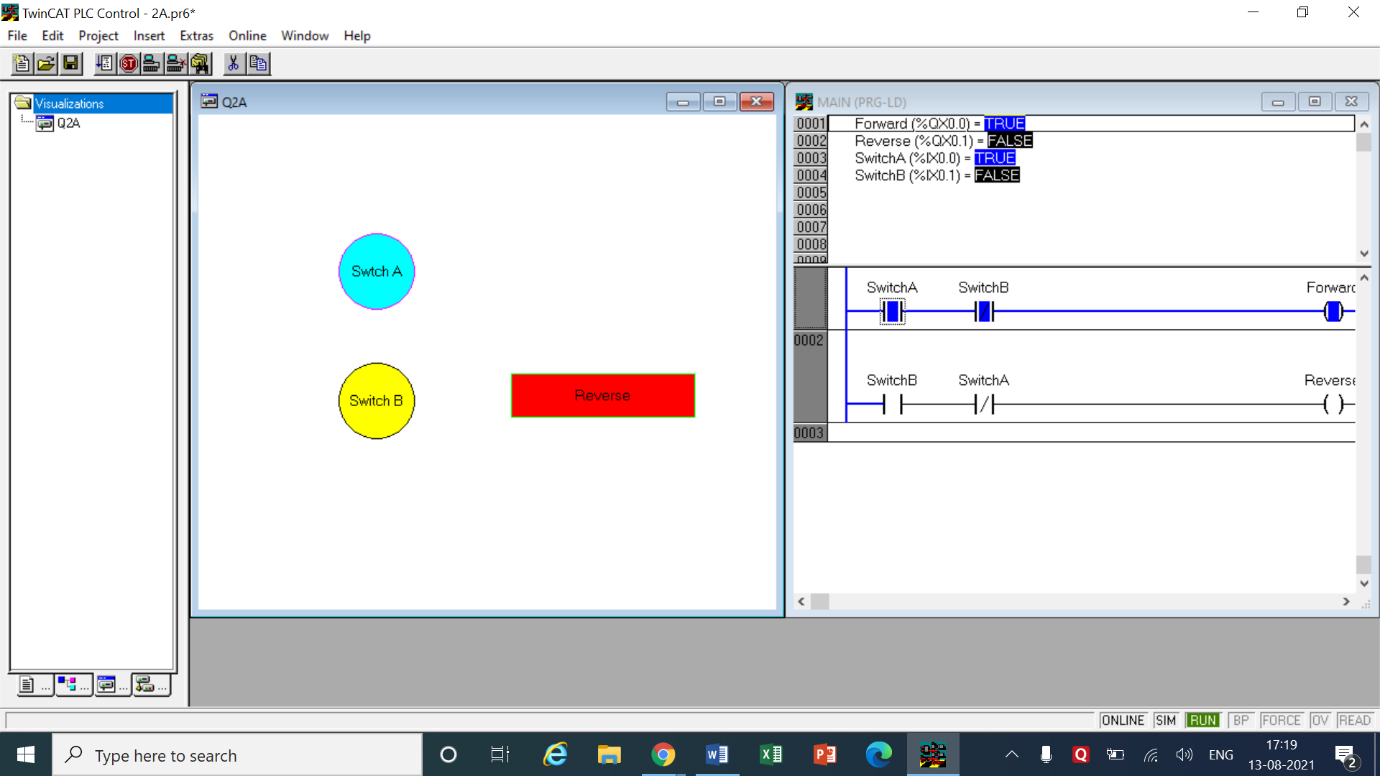
**Theory**

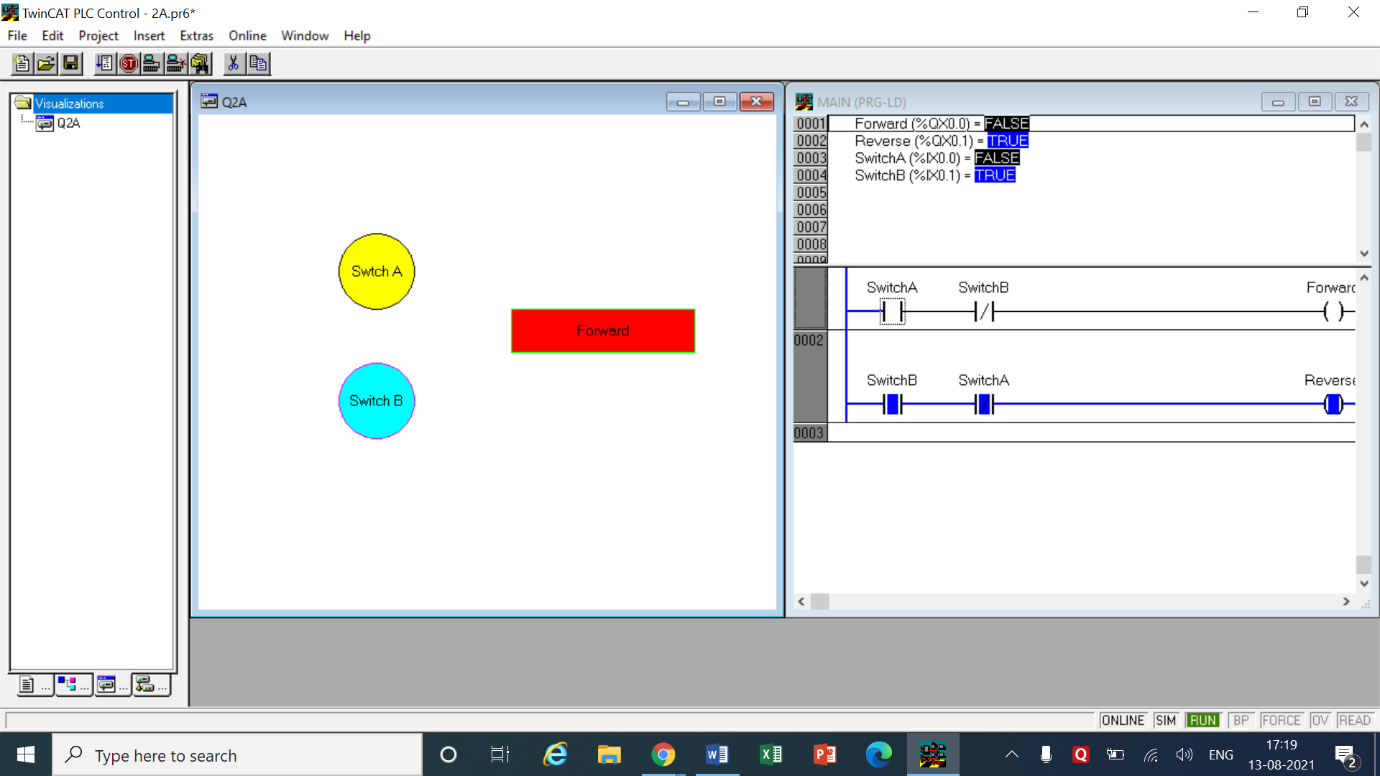
The programmable logic controller (PLC) is a solid state electronic device designed to replace electromechanical relays, timers, counters and sequences, by using a programmable memory for the internal storage of user oriented instructions for implementing specific functions such as logic sequencing timing, 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**

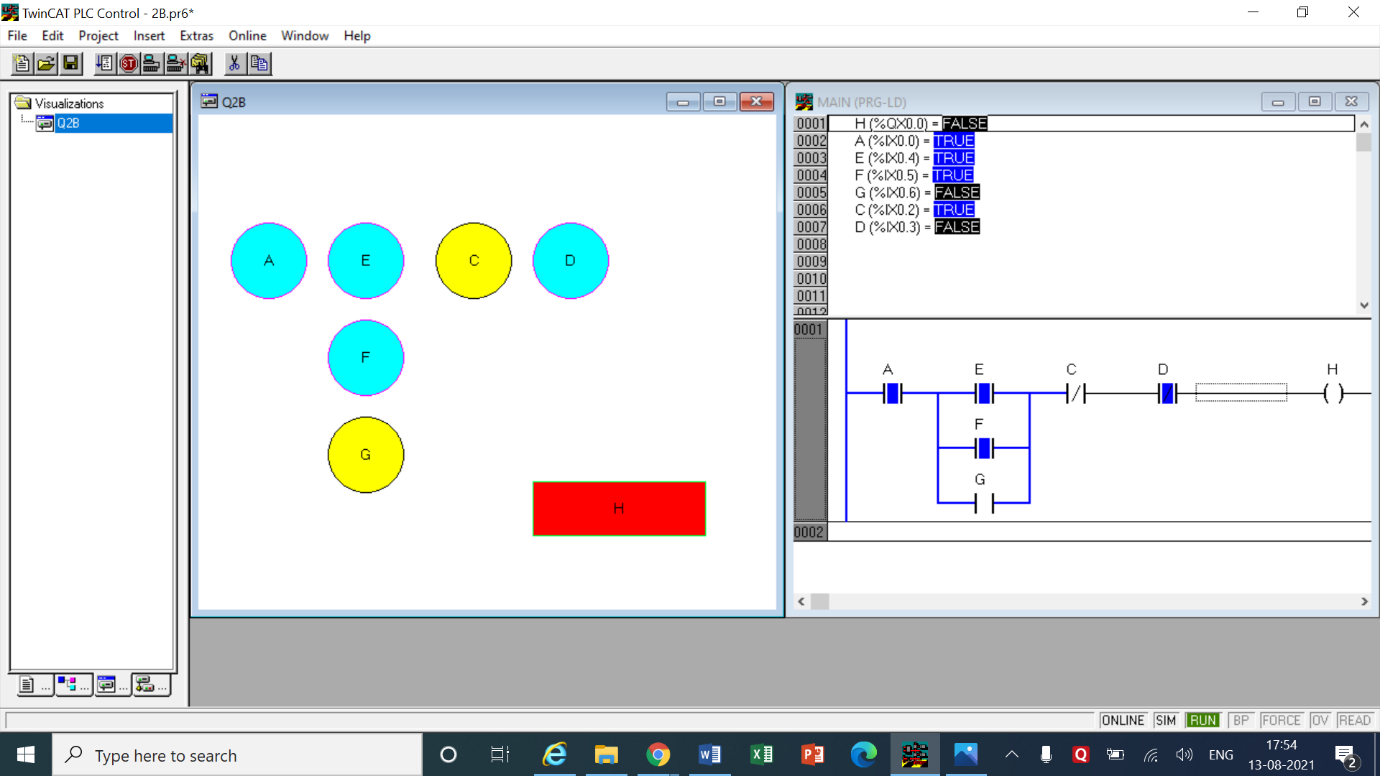
1. Construct a ladder diagram for an instant Forward-Reverse control system for a motor. The motor should run in forward direction with one switch but should be reversed with another switch.

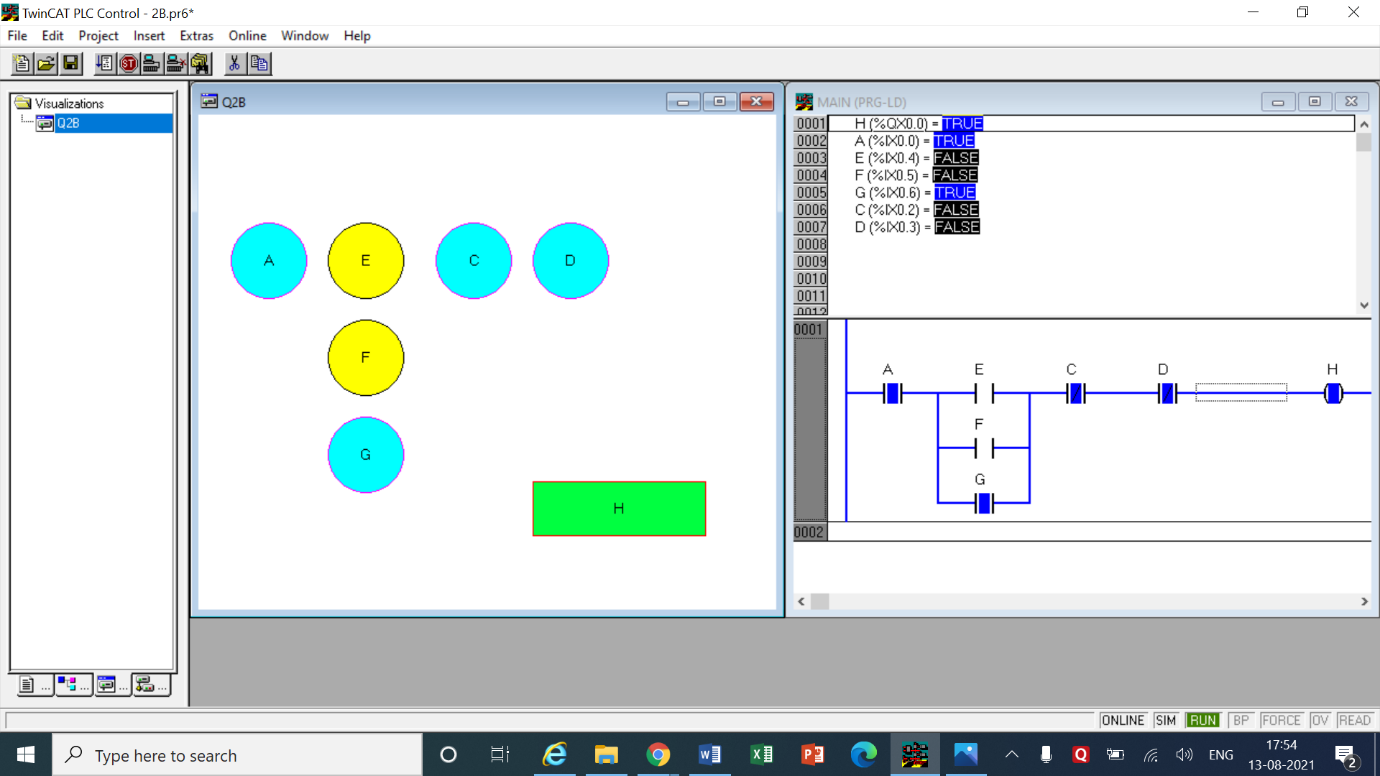


LOGIC -

* In this question it was asked to turn the motor in forward direction with one switch so a NO switch A was added in the first network and at the end the forward connection was attached.
* The task also stated that the motor should be reversed with another switch so a NO switch B was added in second network and the reverse connection was connected at the end.
* The switches in the question do not state what happens when both the switches are on so we add a NC switch B in network 1 and a NC switch A in network 2.
* This new set-up will ensure that when both the switches are on the motor will not run at all.

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Switch A | %ix0.0 | Forward | %qx0.0 |
| Switch B | %ix0.1 | Reverse | %qx0.1 |
| Switch A and B | %ix0.0  %ix0.1 | Nil | - |

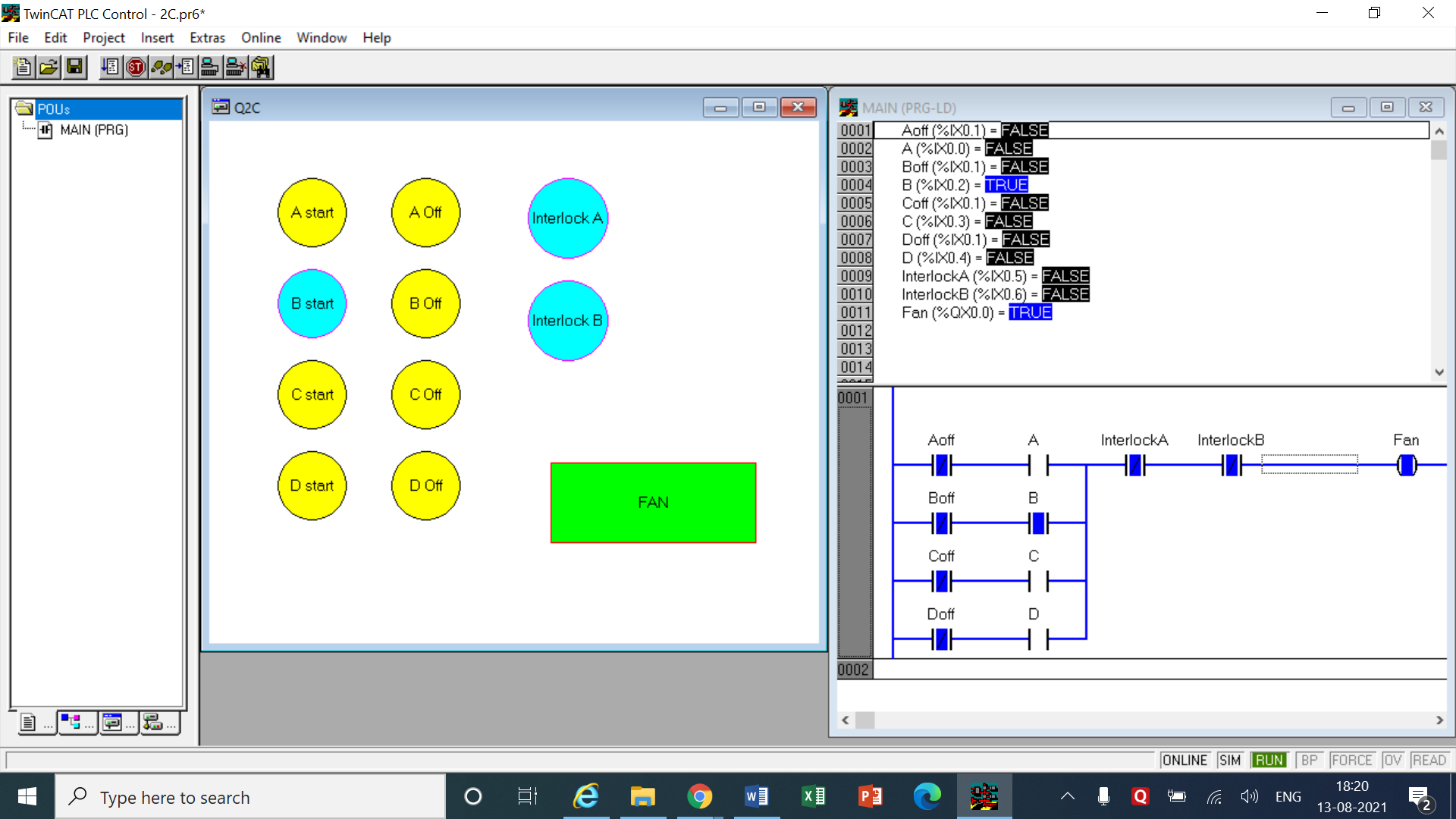
1. For output H to be on, A must be on and both input C and D must be off. In addition one or more of inputs E, F and G must be on.

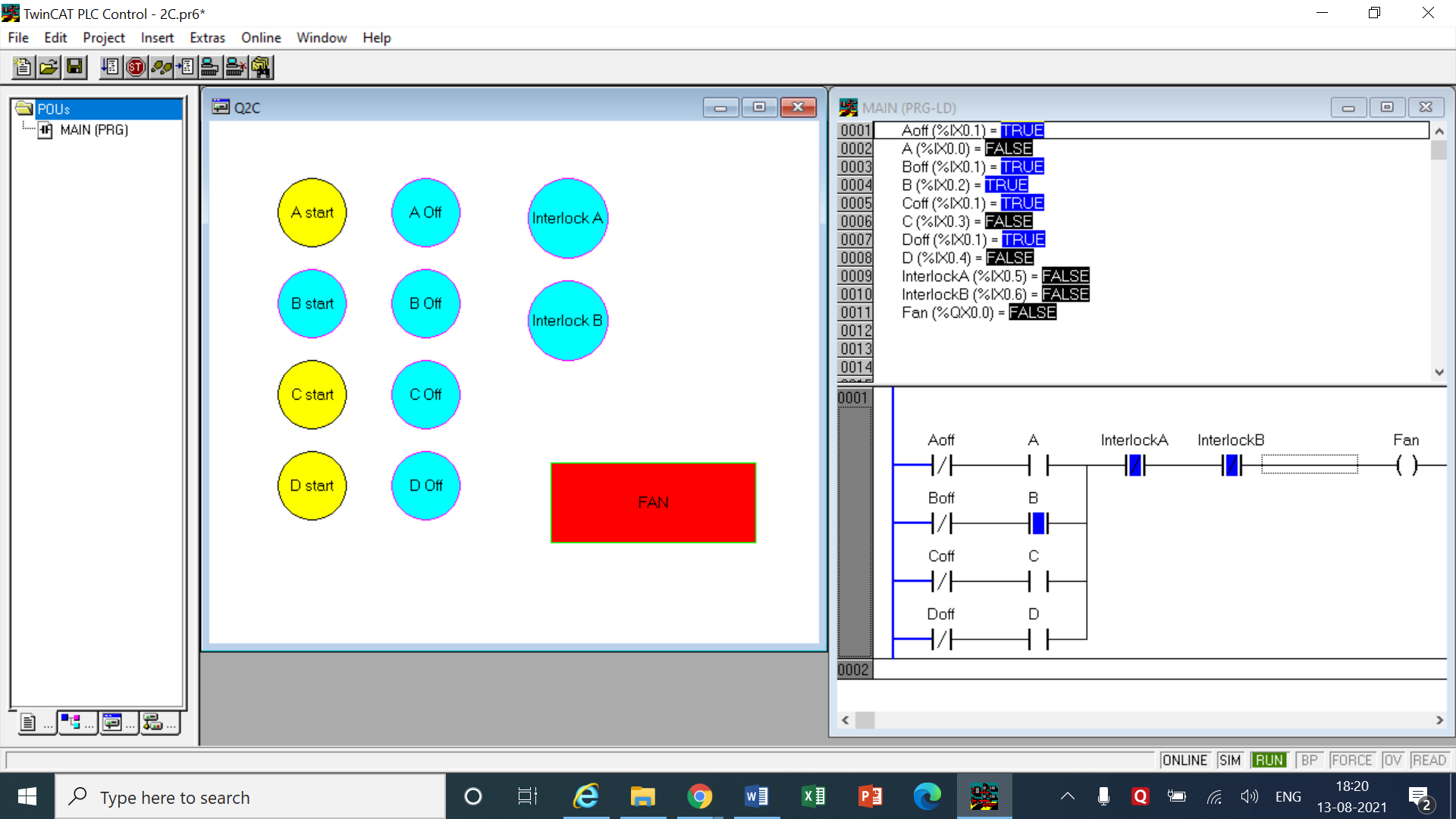


LOGIC

* Switch E, F and G are connected in parallel so whenever one of them turns on the circuit will move forward.
* Switches C and D are taken as NC switches and kept in series because H will only be on when these both switches are kept off so when the inputs are turn on the circuit will break.
* Finally Switch A is kept in series with the entire circuit.

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Switch A | %ix0.0 | H | %qx0.0 |
| Switch E | %ix0.4 | - | - |
| Switch F | %ix0.5 | - | - |
| Switch G | %ix0.6 | - | - |
| Switch C | %ix0.2 | - | - |
| Switch D | %ix0.3 | - | - |

1. Four pushbutton stations control a fan. Each stations has a start and stop button. Two door interlocks must be closed before the fan may run. Pushing any button will make the fan run, and the fan is sealed on when the start button is released. Pushing any stop button turns the fan off and also prevents the fan from starting or running.

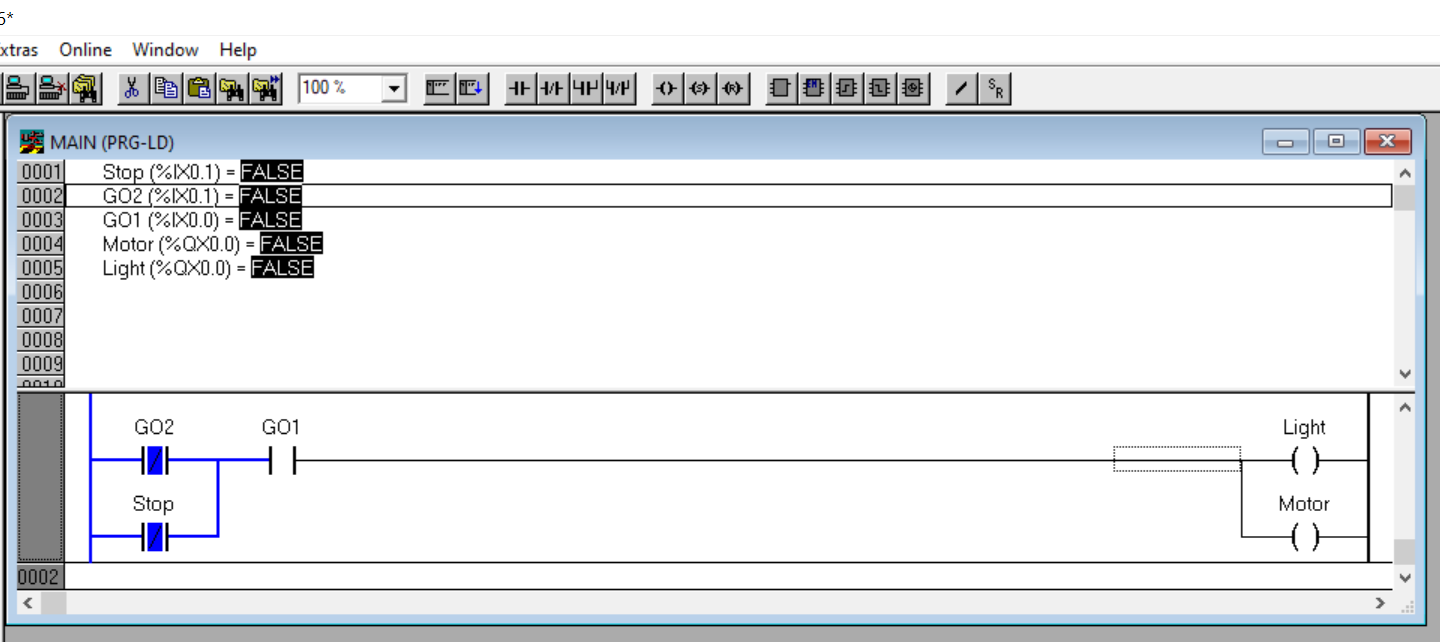


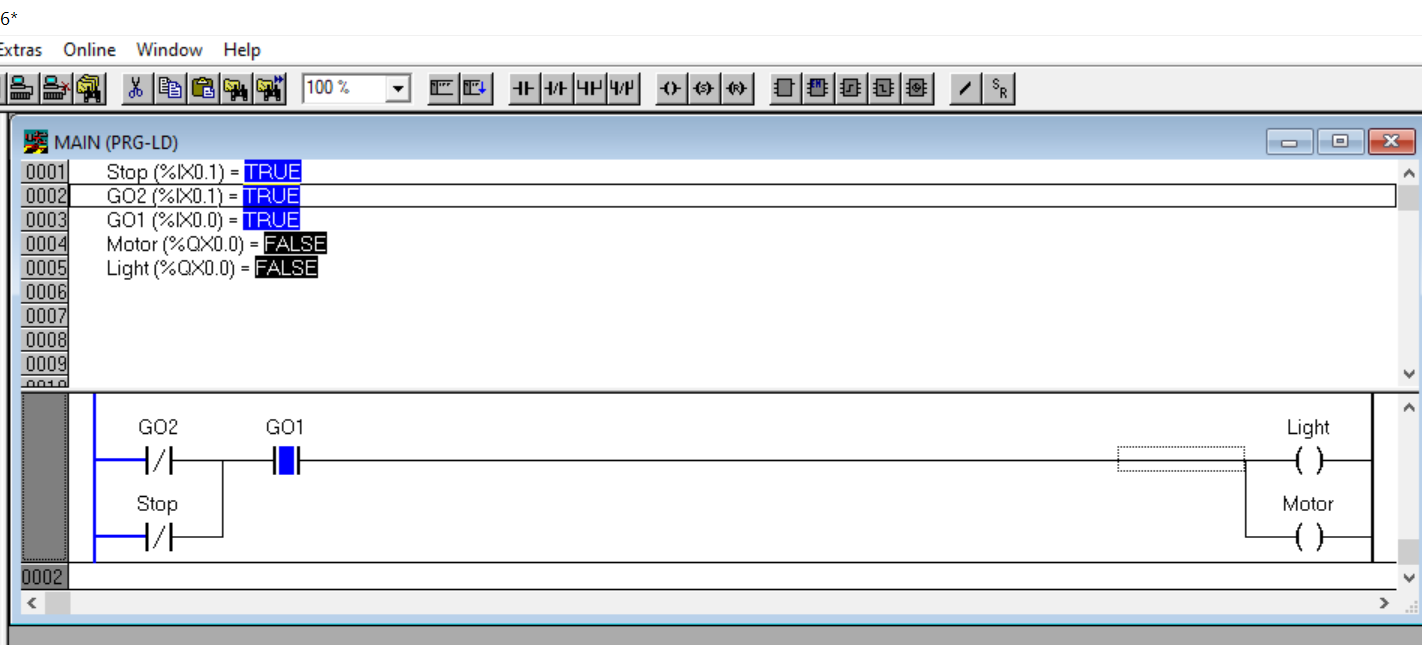
LOGIC

* There are 4 pushbuttons in this network attached parallely in this network: A, B, C and D.
* Each pushbutton has its own off button which is a normally closed switch and by default it is kept on.
* Interlocks A and B are NC switches connected in series with the circuit. So when the interlocks are off the circuit is complete and when the interlocks are turned on the circuit is broken
* When any of the stop buttons is pushed then then the fan should turn off regardless of the button which is pushed
* So the all the buttons have same inputs but different names, when any is pushed, it means actually only that one off switch is pushed

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Switch A | %ix0.0 | FAN | %qx0.0 |
| Switch B | %ix0.2 | - | - |
| Switch C | %ix0.3 | - | - |
| Switch D | %ix0.4 | - | - |
| Switch A off  Switch B off  Switch C off  Switch D off | %ix0.1 | - | - |
| Interlock A | %ix0.5 | - | - |
| Interlock B | %ix0.6 | - | - |

1. A motor will be controlled by two switches. The Go switch will start the motor and the Stop switch will stop it. If the Stop switch was used to stop the motor, the Go switch must be thrown twice to start the motor. When the motor is active a light should be turned on. The Stop switch will be wired as normally closed. Design a ladder for this problem.

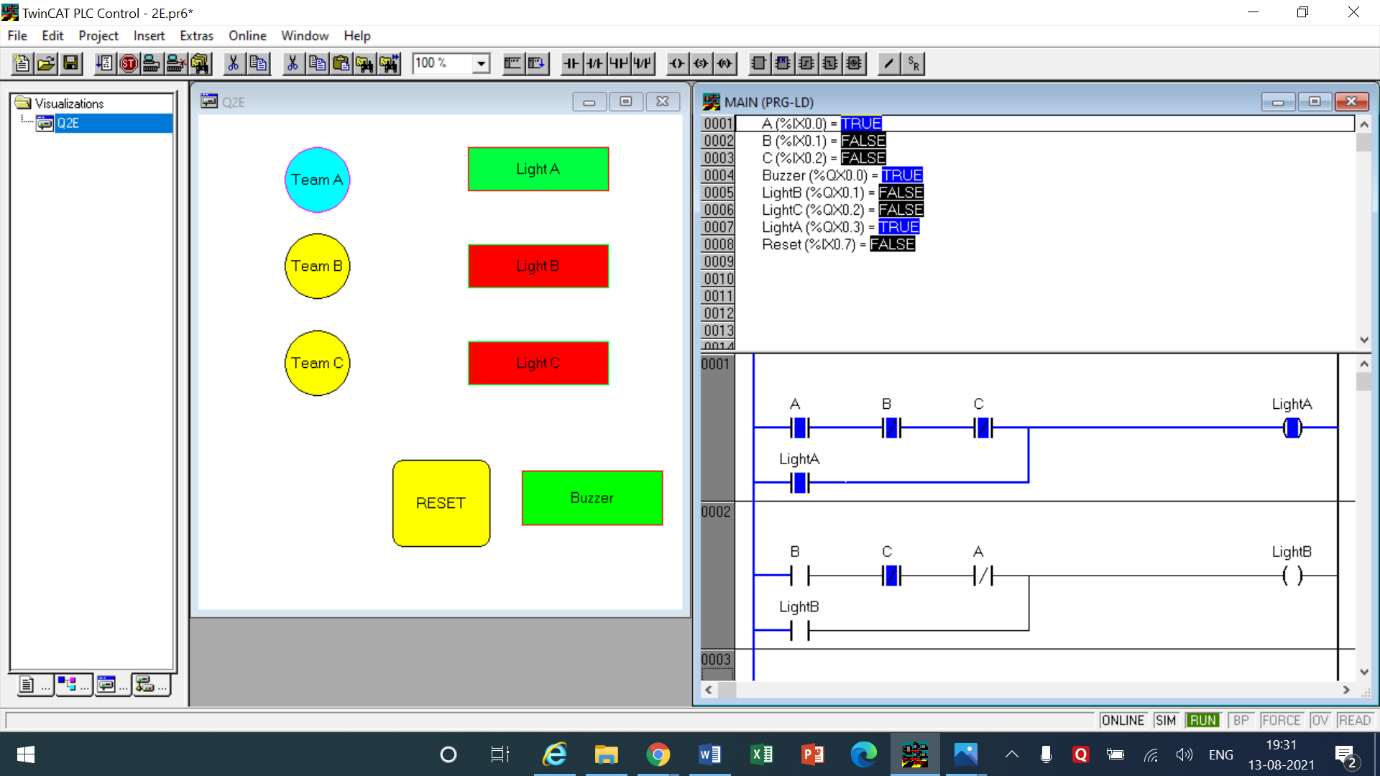


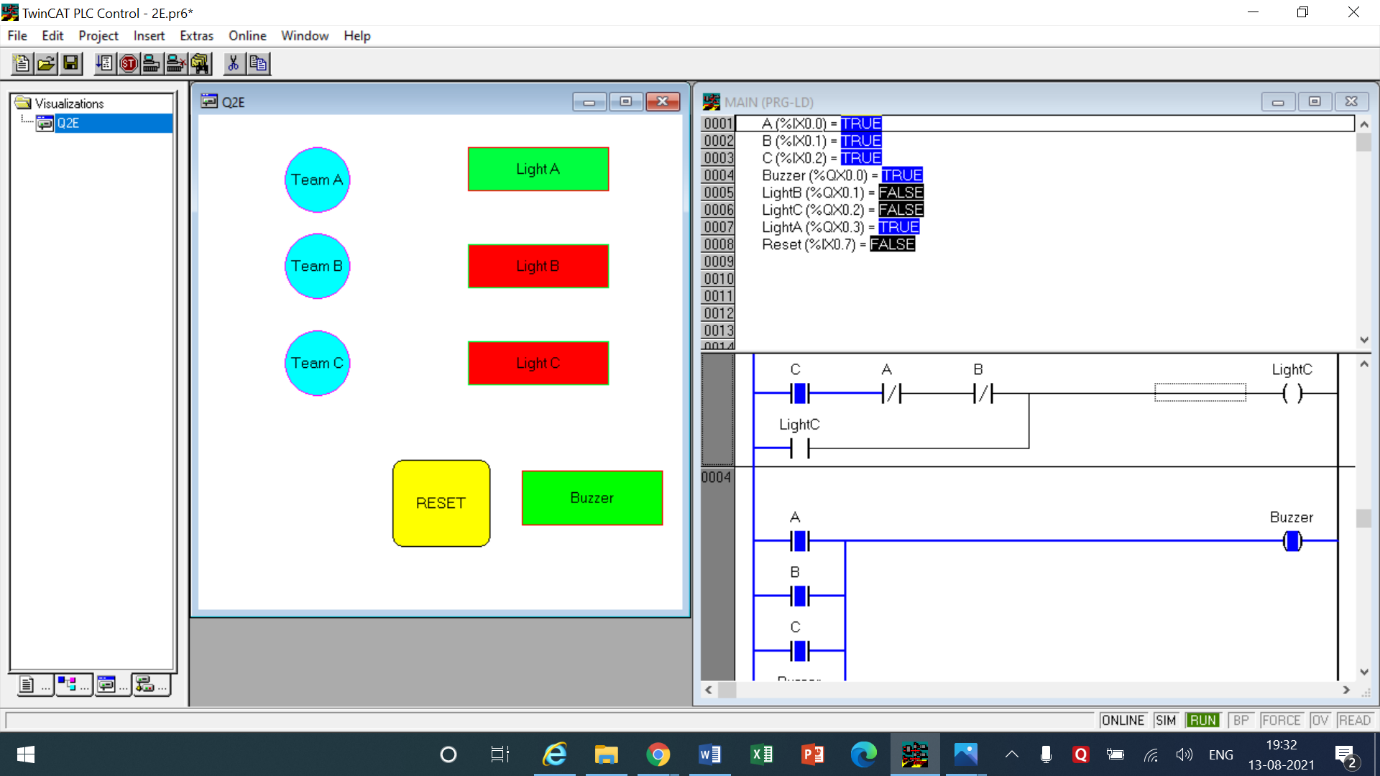


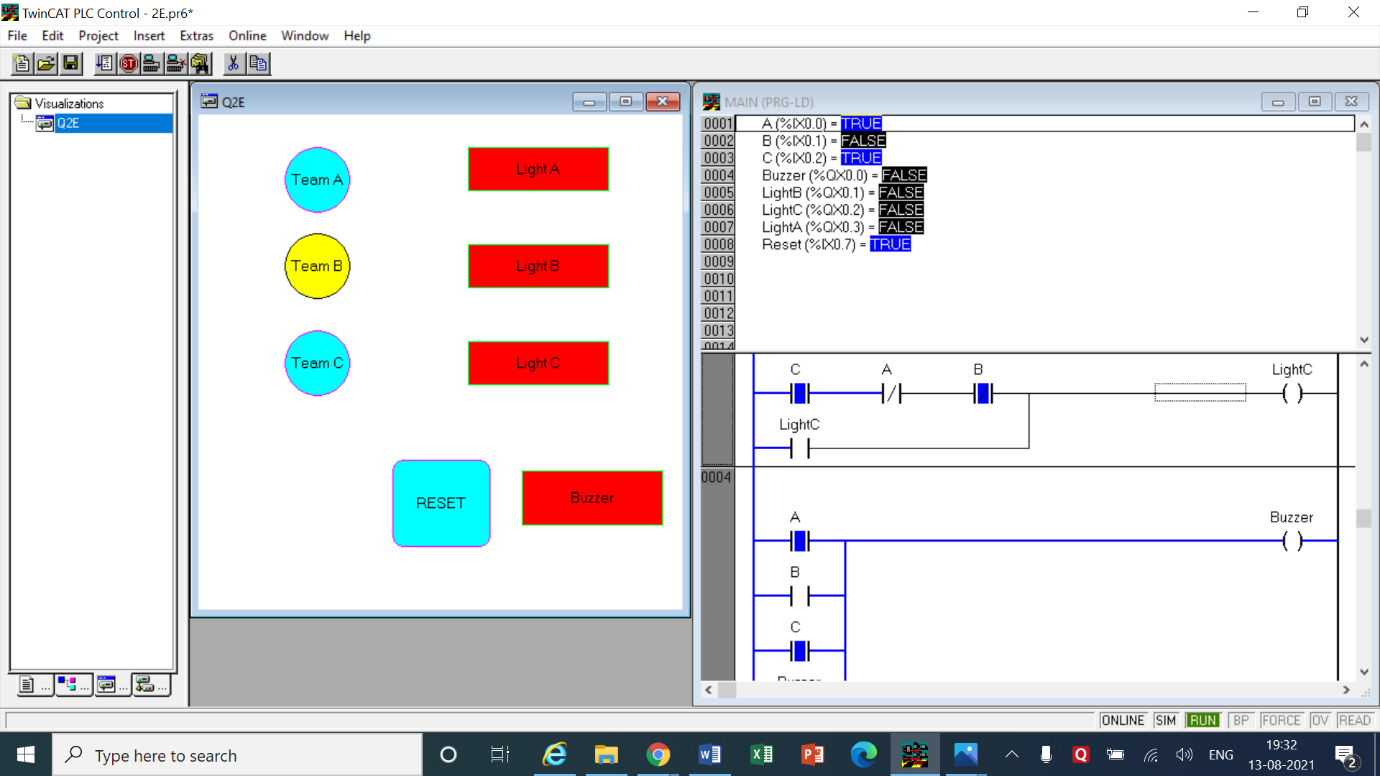
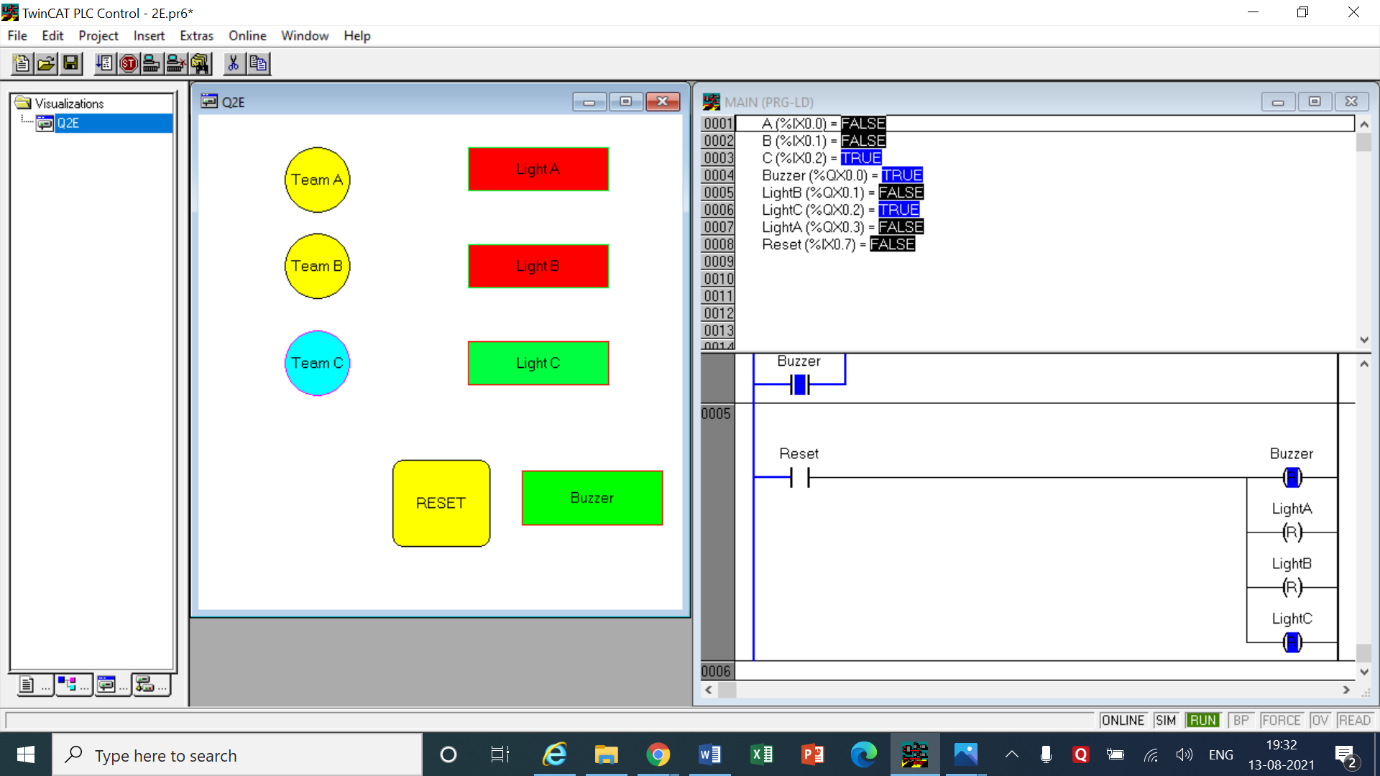
LOGIC

* This Logic was forcefully simplified because the go switch is supposed to act like a toggle switch but in TwinCAT there is only normal switch.
* So here we use 3 different inputs GO switches 1 and 2 along with the stop switch.
* The GO2 and stop switch are normally closed are have the same input connection and are parallel to indicate the differences (no other reason, hence can keep in series too)
* When the program is run GO2 is NC so we have to press GO on the motor only once then light and motor turns on.
* When we press the stop switch the GO2 also gets turned off so on the surface of motor we have to now press GO again twice (Assuming Go is a toggle switch).

1. Priority determination: Three teams will press a switch in front of them to fight to be the first to answer a question. The buzzer will sound after any one of the team touches the switch. Light indicator in front of each team will light up and only the host switch resets the buzzer.

****

****

****

LOGIC

* In this question, there are 5 networks set up.
* The first three networks are for teams A, B and C where the switch of any 1 team is normally open switch and the other 2 are normally closed switches.
* This arrangement will ensure that when any one team presses their switch, their own respective circuit will be completed but the other 2 networks will be broken off, this means that when A is closed the NC A opens breaking the circuit in other 2.
* So because of the above arrangement even if other teams press their switches their circuit won’t be completed and light won’t be turned on because A pressed first breaking their network.
* Now the lights of the respective teams are again looped back into the circuit using a switch connecting it parallel to the existing circuit which ensures that even if the winning team releases their switch the light will stay on regardless.
* The 4th network is made for the buzzer where all teams are connected in parallel to the buzzer and the buzzer is looped backed into the network so it stays on like before
* The 5th network is made for the reset button which the host can press before the next round begins. This uses the reset coil which resets the buzzer and all the team’s lights.

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Team A | %ix0.0 | Light A | %qx0.3 |
| Team B | %ix0.1 | Light A | %qx0.1 |
| Team C | %ix0.2 | Light A | %qx0.2 |
| Reset | %ix0.7 | Buzzer | %qx0.0 |

COMMENTS –

* In the questions above all the input visualizations are yellow in color when they are in off state and they turn sky blue in color when they are put in the on state
* The outputs are different, when they are in the off state they have red color but they are in their on state they are green in color like normal convention.
* In 2.A the on state is depicted as ‘visible’ and the off state is depicted as ‘invisible’. (It was a mistake on my part)

**CONCLUSION**

* In this experiment we thoroughly practiced how to use the Normally Closed and Normally Open switches to create different real life scenarios which can be easily solved by PLC program. Along the way we also learned how to use the Set and Reset coils for the program. We visualized and simulated the output with different colors. In the last question we also saw how we can lop the coil as a switch back into the circuit so the circuit remains on.