**NIRMA UNIVERSITY**

**INSTITUE OF TECHNOLOGY**

**MECHANICAL ENGINEERING**

**Programmable logic controller (PLC)**

**LAB – 3**

**19BME134**

**Shrey Shah**

**Aim**

Programming of TwinCAT PLC using relay based ladder language.

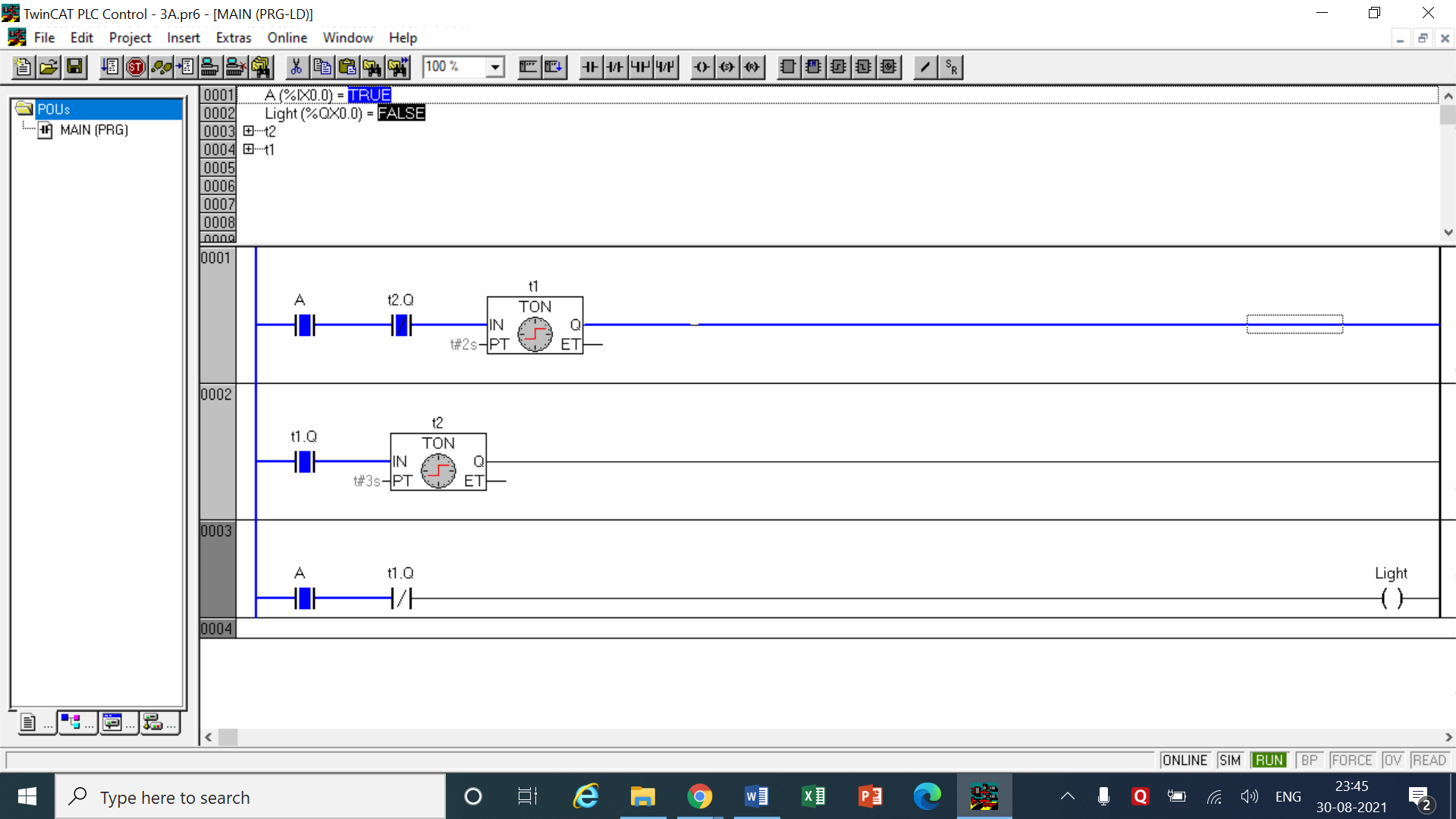
**Theory**

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**

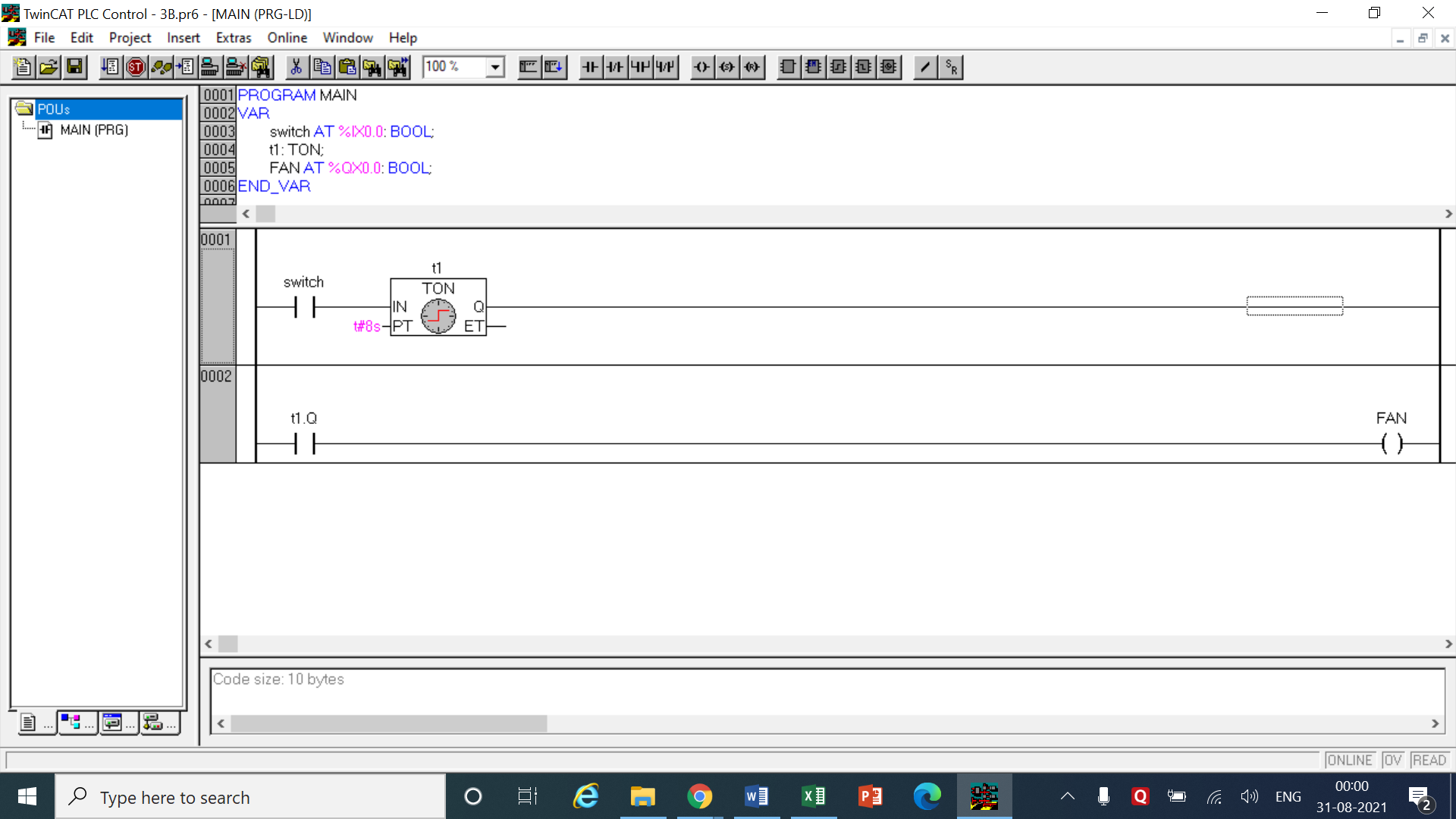
1. Write a program to generate Square wave of 40 % duty cycle with cycle time of 5 seconds.

LOGIC -

* This question required us to generate a duty cycle with on state of 40% that means in a 5 second interval we have to make the output, LED light in this case, turn on for 2 seconds and turn off for 2 seconds.
* The ‘A’ switch is for turning on the entire process. A ON timer is placed of delay time to turn on of 2 second in the first network. The timer is named t1.
* The same switch ‘A’ is also present in the 3rd network along with a normally closed output of the t1 timer. When A is turned on the light immediately turns on too and the timer t1 starts countdown.
* After 2 seconds of on state are over, output of t1 turns one and the normally closed switch opens in network 3 turning the light off.
* A normally open switch of output of t1 is also connected in 2nd network with another ON timer of delay of 3 seconds for the ‘off’ state.
* When t1’s output is turned on the countdown of t2 starts for 3 seconds. We add a normally closed switch of output of t2 in 1st network. After 3 seconds are over, t2’s output is on and the normally closed switch in network 1 is opened breaking the circuit.
* When the circuit is broken t1 is shut off and it becomes normally closed again in 3rd network and completing that circuit.
* This cycle now goes on forever as long as the switch A is on.

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Switch A | %ix0.0 | light | %qx0.0 |
| On Timer t1 | Delay = 2 sec | t1.q | Output of t1 |
| On Timer t2 | Delay = 3 sec | t2.q | Output of t2 |

1. Write a PLC program for a timer which turns on a fan switch 8 seconds after the wall switch is turned on. If the wall switch is turned off during the 8 second time interval, the timer is reset to zero, so that when the wall switch is again turned on, the delay is full 8 seconds.

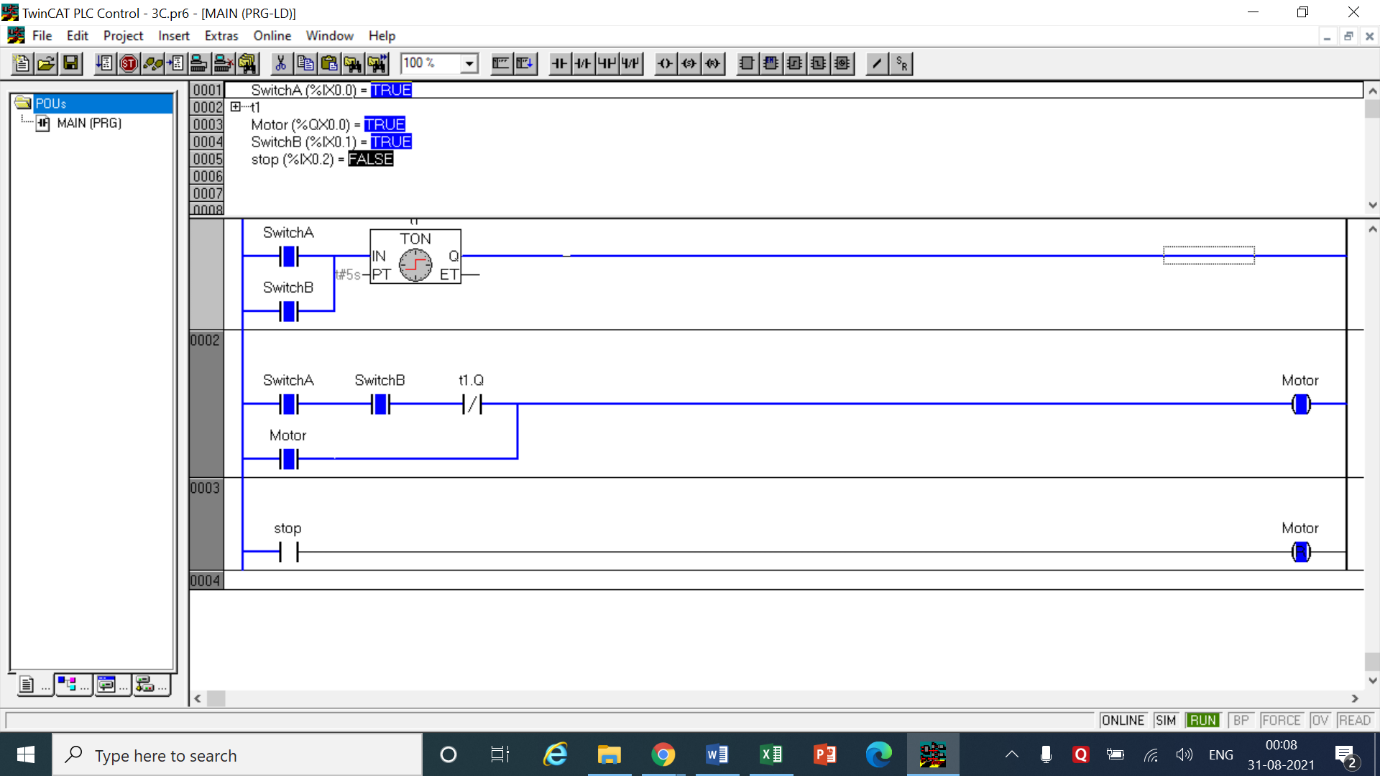


LOGIC

* The logic here is pretty simple. We first add a switch to turn on the fan. A On timer ‘t1’ is connected after the switch with a delay time of 8 seconds.
* The output of the timer is now taken in the 2nd network and connected to the fan.
* This means that when the switch is turned on the fan will switch on after 8 seconds obviously BUT the default setting of an off timer is such that when the power is turn off it automatically resets its countdown which means no external setup is required for setting up in this question

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Switch | %ix0.0 | Fan | %qx0.0 |
| On Timer t1 | Delay = 8 sec | t1.q | Output of t1 |

1. In dangerous process it is common to use two palm buttons that require an operator to use both hands to start a process (this keeps hands out of process, etc.) To develop this there are two inputs that must be turned on within 0.25 seconds of each other before a machine cycle may begin

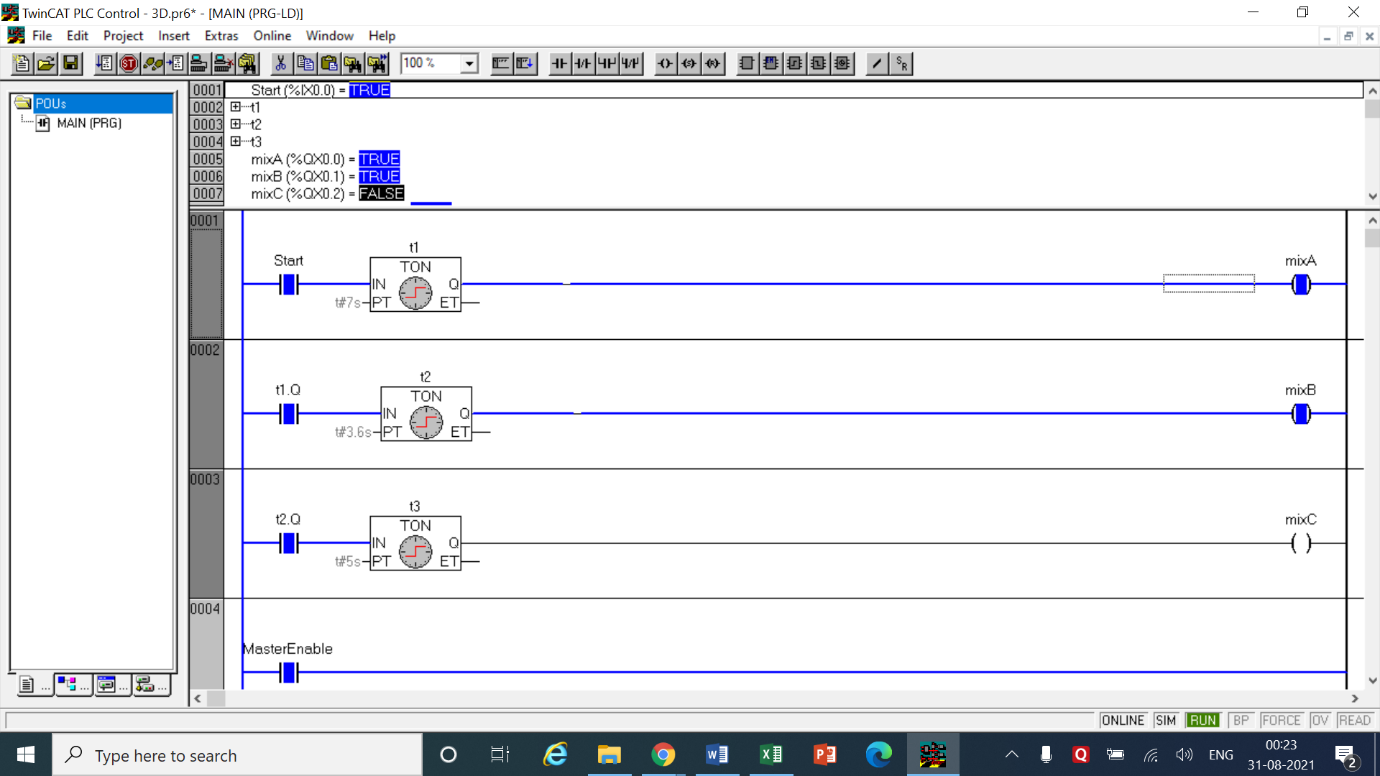


LOGIC

* It is stated that one has to press both of the switches under 0.25 seconds for the motor to successfully turn on. This means that when any one of the switch is turned on, the countdown of 0.25 seconds will be started and the other switch should be turned on within this time.
* For the sake of simplicity I have taken the time as 5 seconds instead of 0.25 seconds.
* Now first we add both the switches in parallel which connects them to the On timer ‘t1’ with a delay time of 5 seconds in the 1st network.
* In the 2nd network these both switches A and B are connected in series which means that the circuit will only continue when both are closed.
* A normally closed connection of the output of ‘t1’ is taken after both A and B so that if the second switch is not turned on within 5 seconds it will break the circuit and the motor will not be turned on.
* NOW, if a person manages to press second switch within 5 seconds then the motor is connected in parallel with the entire circuit in 2nd network as an INPUT.
* Then the motor will remain on even after 5 second normally closed output is opened.
* In the 3rd network a stop switch is used to turn off the motor and it connects to a reset coil which resets the motor breaking the loop.

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Switch A | %ix0.0 | Motor | %qx0.0 |
| Switch B | %ix0.1 | t1.q | Output of t1 |
| On timer t1 | Delay = 5 sec | Reset coil  for motor | %qx0.0 |
| Stop | %ix0.2 | - | - |

1. There are three mixing devices on a processing line: A, B and C. After process begins, mixer A is to start after 7 seconds elapse. Next, mixer B is to start 3.6 seconds after A. Mixer C is to start 5 seconds after B. All then remain on until a master enable switch is turned off

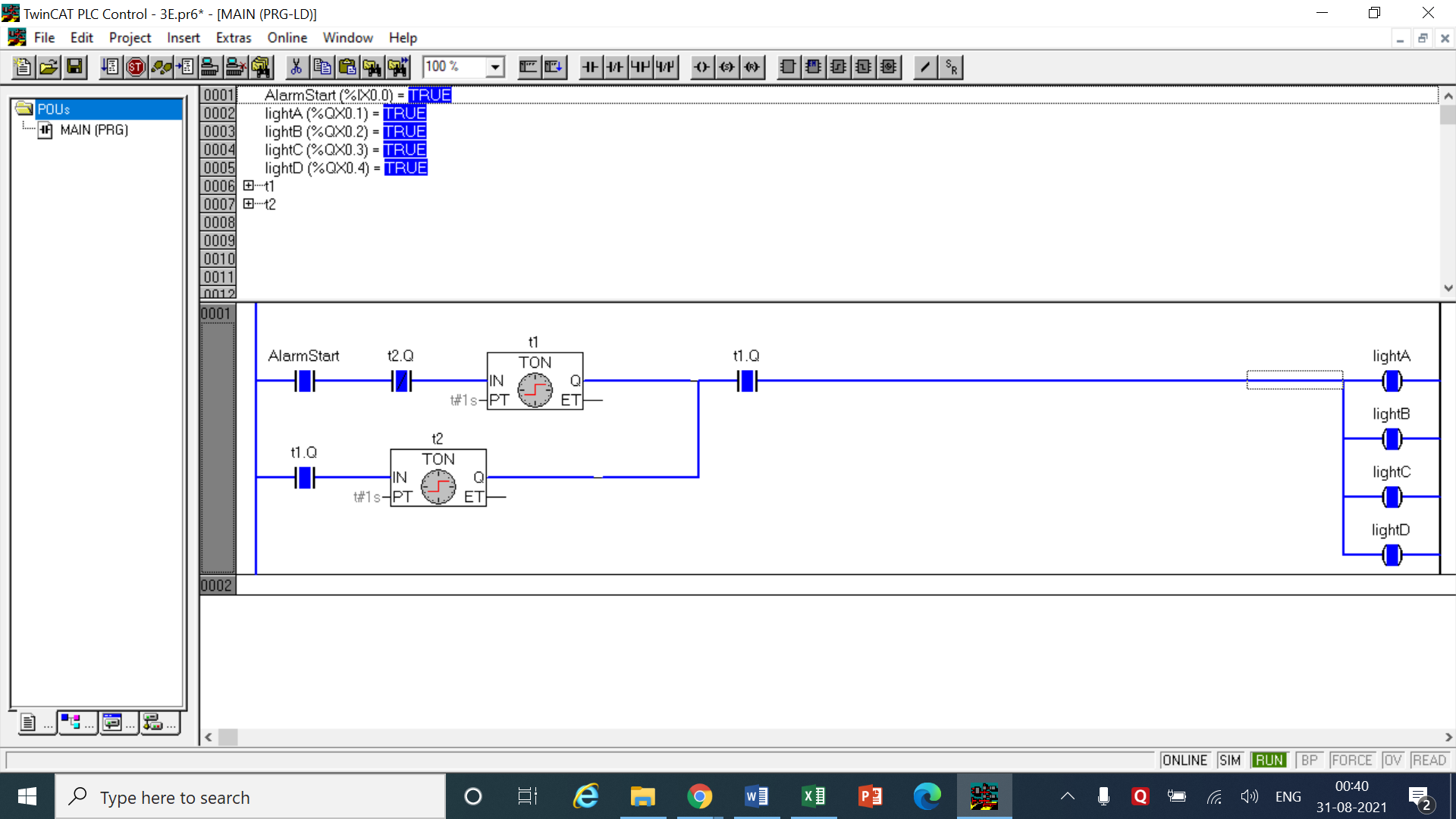


LOGIC

* This logic is also quite simple like the second question. Here we just have to turn each mixer on after specific intervals after the initial start switch is turned on.
* So in the first network a start switch is kept and along with it an On timer of delay time of 7 seconds is kept because the question requires the mixer A to turn on after 7 seconds of start switch.
* When 7 seconds are over the mixer A starts and the output of timer as, ‘t1.q’, is connected in the 2nd network with another On timer t2 with delay of 3.6 seconds to which the output mixer B is connected.
* So now mixer B starts after 3.6 seconds of mixer A. when output of timer t2 is on, it is again connected in a 3rd network with an On timer t3 which has a delay time of 5 seconds.
* The mixer 3 is connected at the end of 3rd network which turns on after 5 seconds of mixer B getting turned on.
* Now all the mixers are on as desired. In the 4th network a master enable switch is attached which has basically the same input as the start switch.
* When Master Enable switch is turned on (off as switch A was already on) switch A is turned off which shuts of the supply to t1 which shuts of its supply to t2 and so on thereby shutting off all the supply and all the mixers along with them.

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Switch A | %ix0.0 | Motor | %qx0.0 |
| On Timer t1 | Delay = 7 sec | t1.q | Output of t1 |
| On Timer t2 | Delay = 3.6 sec | t2.q | Output of t2 |
| On Timer t3 | Delay = 5 sec | t3.q | Output of t3 |
| Master Enable | %ix0.0 | MixA | %qx0.1 |
| - | - | MixB | %qx0.2 |
| - | - | MixC | %qx0.3 |

1. Design a ladder program using timer instruction to flash 4 process alarm lights on a control panel at a rate of 1 second. The alarm light is driven by PC outputs at 01000, 01001, 01002 and 01003. Connect a panel mounted push button to input point 00000 to acknowledge the alarms. Assume the alarm lights will remain on after acknowledge push button is depressed and will turn off only if the alarm input has been cleared.

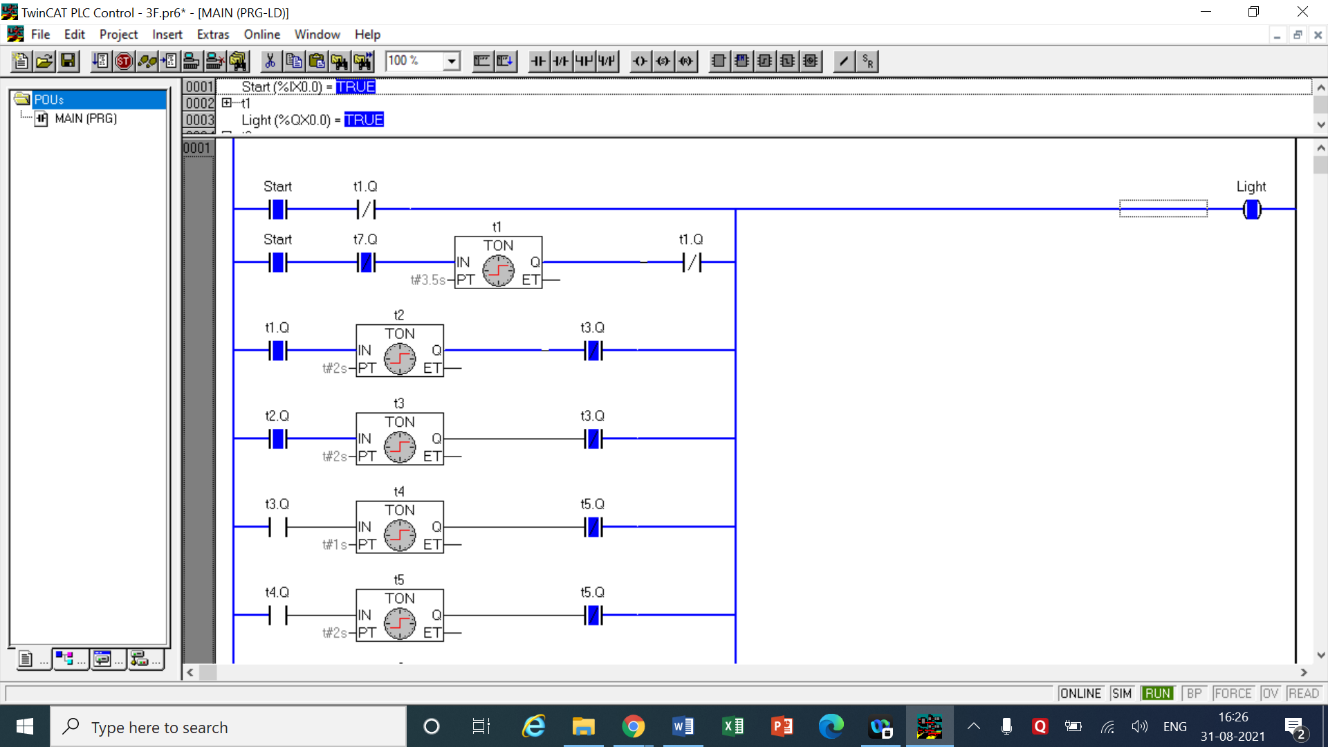
****

LOGIC

* The logic I implemented was based on my understanding of the question.
* Assuming an alarm goes off, for that an Alarm switch is placed at the start of network 1
* Similar to the first question an On timer t1 is added in network with delay time of 1 second.
* When the Alarm is turned on the timer starts and after 1 second the output of first timer is connected with another On timer t2 in parallel with the above connection with delay time of 1 second.
* After 1 second the output of t1 is on and circuit is completed and the lights are turned on simultaneously. And timer 2 starts whose output is connected as a normally closed switch before timer 1.
* When 1 second of timer 2 is over circuit is broken in the first timer and it is shut off henceforth turning the lights off and after 1 second t2.q closed again completing the circuit repeating the process over and over until the alarm is turned off

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Alarm start | %ix0.0 | Light A | %qx0.3 |
| On Timer t1 | Delay = 1 sec | Light B | %qx0.1 |
| On timer t2 | Delay = 1 sec | Light C | %qx0.2 |
| - | - | Light D | %qx0.0 |
| - | - | t1.q | Output of t1 |
| - | - | t2.q | Output of t2 |

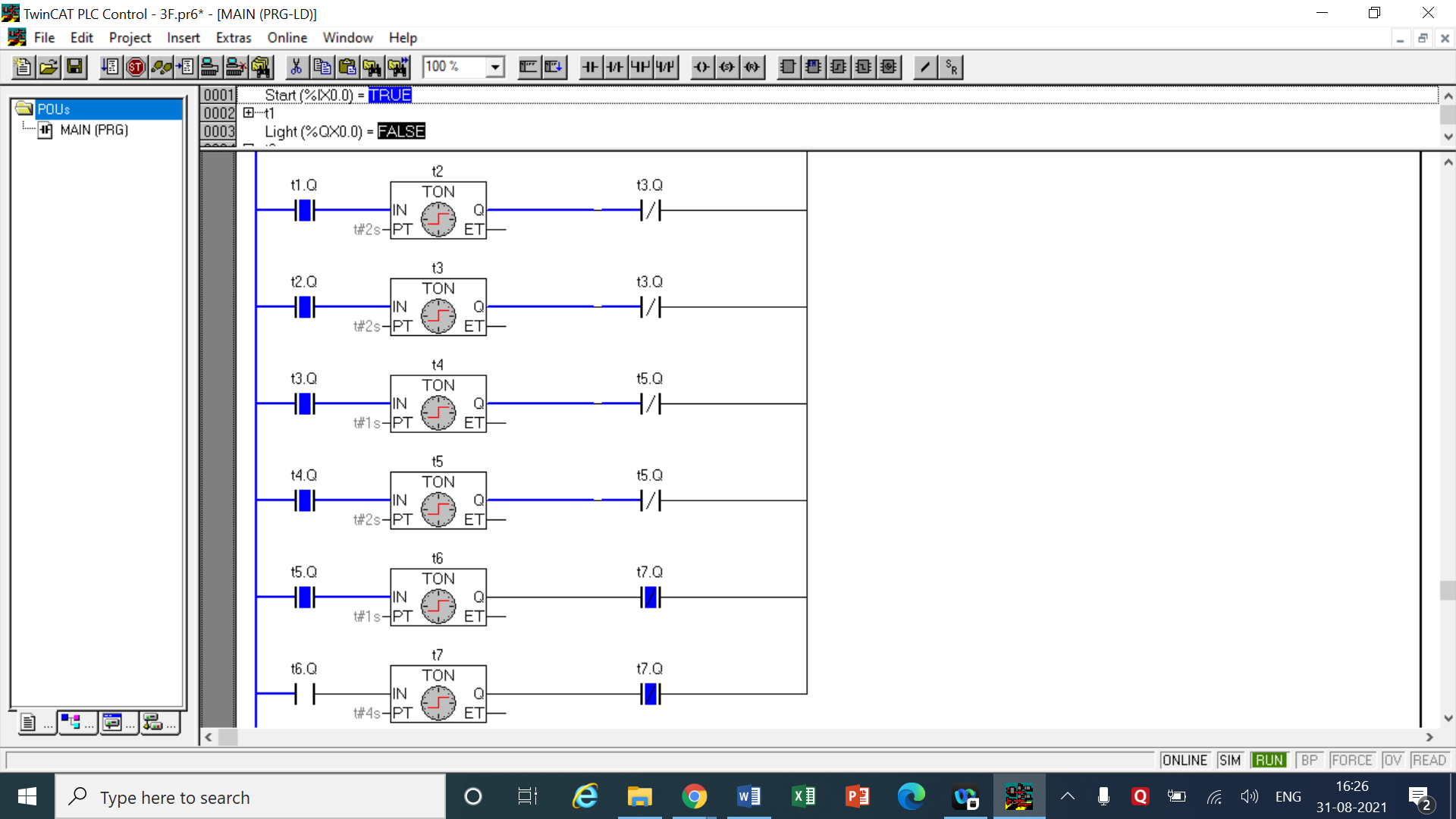
1. Write ladder logic that will give the following timing diagram for B after input A is pushed. After A is pushed any changes in the state of A will be ignored.



LOGIC

* In this question we have to form a waveform of the given graph.
* To start first the waveform remains on for 3.5 seconds hence first we add a start switch we add the output of On timer t1 as a normally closed switch. All the subsequent parallel connections come from connecting to the power input loop.
* Now in the second connection below we add the same start switch to which t1 is connected with delay time of 3.5 seconds. In series with t1 is the output of t1 as a normally closed switch. When 3.5 seconds are over t1.q is turned on breaking the normally closed t.1 hence turning off the light.
* The next step is to turn off the waveform for 2 seconds (3.5 – 5.5 seconds). So we add another timer t2 with delay time of 2 seconds to which as add the output of On timer t3 as a normally closed switch in series.
* Before timer t2 we add normally open switch of output of t1 so we t1 is done timer t2 starts.
* In all the subsequent connections below we follow a commons rule which is –

A normally open switch of output of t(n-1) with timer tn and a normally closed switch of output of tn.

* When we reach the end of wave form to make it in a loop we take the normally closed switch of the last timer tn used and connect it just before timer t1.
* ****This setting will run the waveform indefinitely until the start switch is turned off again.

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Alarm start | %ix0.0 | Light A | %qx0.0 |
| On Timer t1 | Delay = 3.5 sec | t1.q | Output of t1 |
| On timer t2 | Delay = 2 sec | t2.q | Output of t2 |
| On timer t3 | Delay = 2 sec | t3q | Output of t3 |
| On timer t4 | Delay = 1 sec | t4q | Output of t4 |
| On timer t5 | Delay = 2 sec | t5q | Output of t5 |
| On timer t6 | Delay = 1 sec | t6q | Output of t6 |
| On timer t7 | Delay = 4 sec | t7q | Output of t7 |

COMMENTS –

* In the questions above 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.

**CONCLUSION**

* In this experiment we thoroughly practiced how to use the On state timers which have a delay time to turn on when the circuit is on. These timers help in creating circuits which operate on intervals of time may that me on or off, both of the functions can be performed using this timer. Real life applications were realized in this experiment using timers such as alarm of a factory when there is an emergency or a timely operation of several different components of a factory. All of this was made possible due to the use of timers.