**NIRMA UNIVERSITY**

**INSTITUE OF TECHNOLOGY**

**MECHANICAL ENGINEERING**

**Programmable logic controller (PLC)**

**LAB – 8**

**19BME134**

**Shrey Shah**

**Aim**

PLC Programming for Automation and Industrial Control Applications – II

**Objectives**

To implement different industrial process with PLC programming

**Theory**

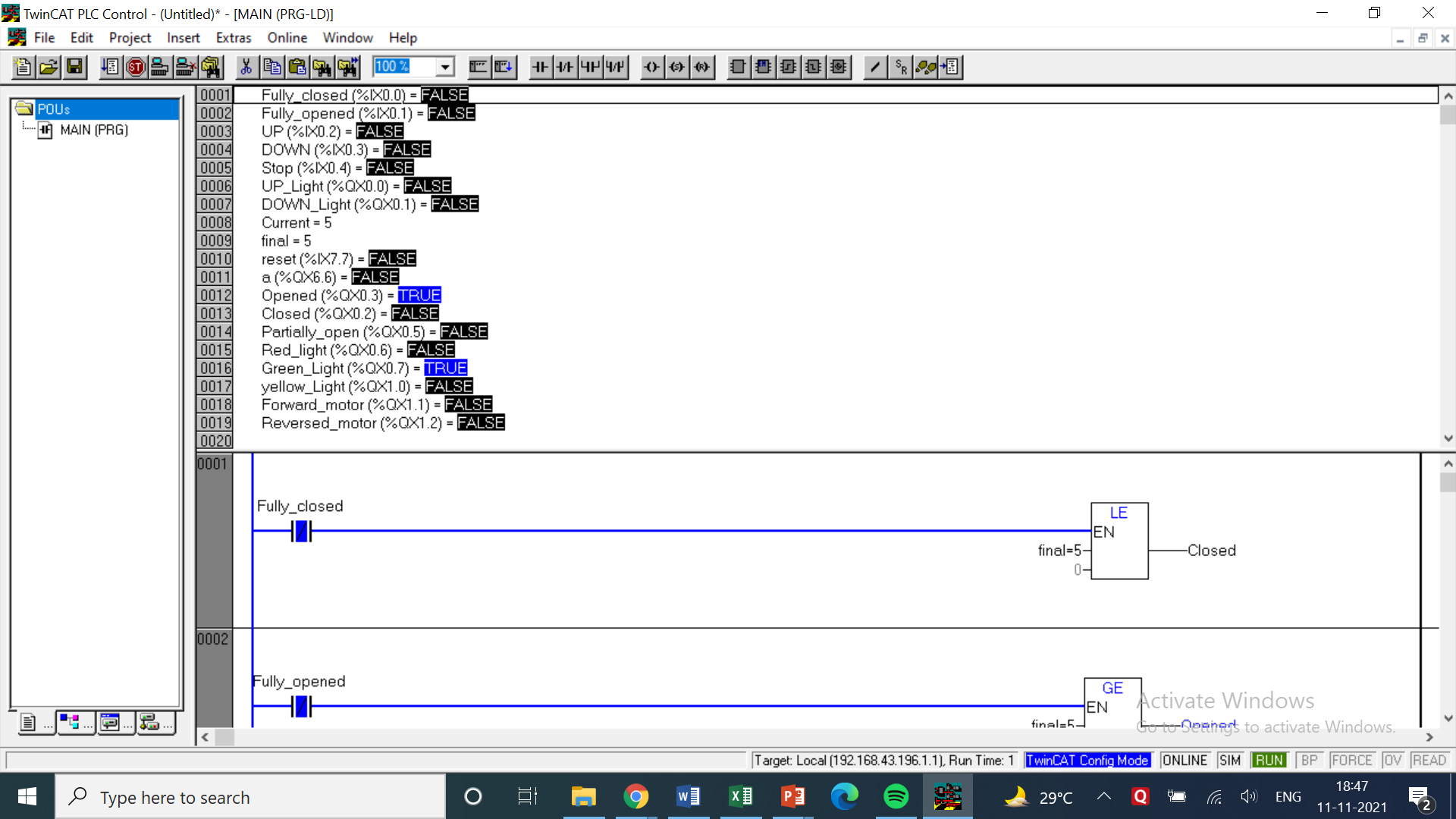
The action of the control response can be summarized as follows:

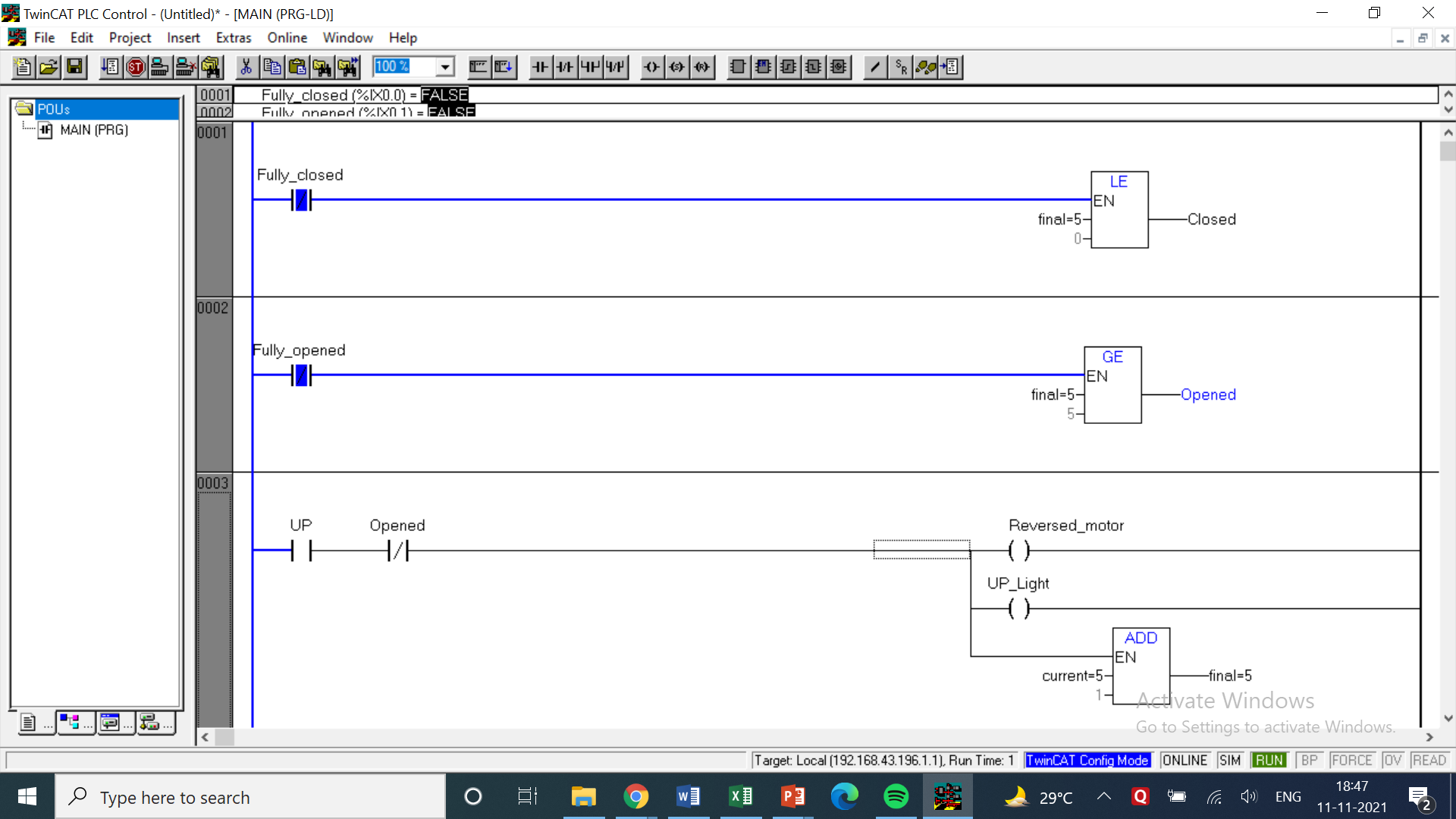
* Proportional-integral-derivative (PID) control is the most sophisticated and widely used type of process control.
* PID operations are more complex and are mathematically based. PID controllers produce outputs that depend on the magnitude, duration, and rate of change of the system error signal.
* Sudden system disturbances are met with an aggressive attempt to correct the condition. A PID controller can reduce the system error to 0 faster than any other controller.
* Programmable controllers are either equipped with PID I/O modules that produce PID control or have sufficient mathematical functions of their own to allow PID control to be carried out.

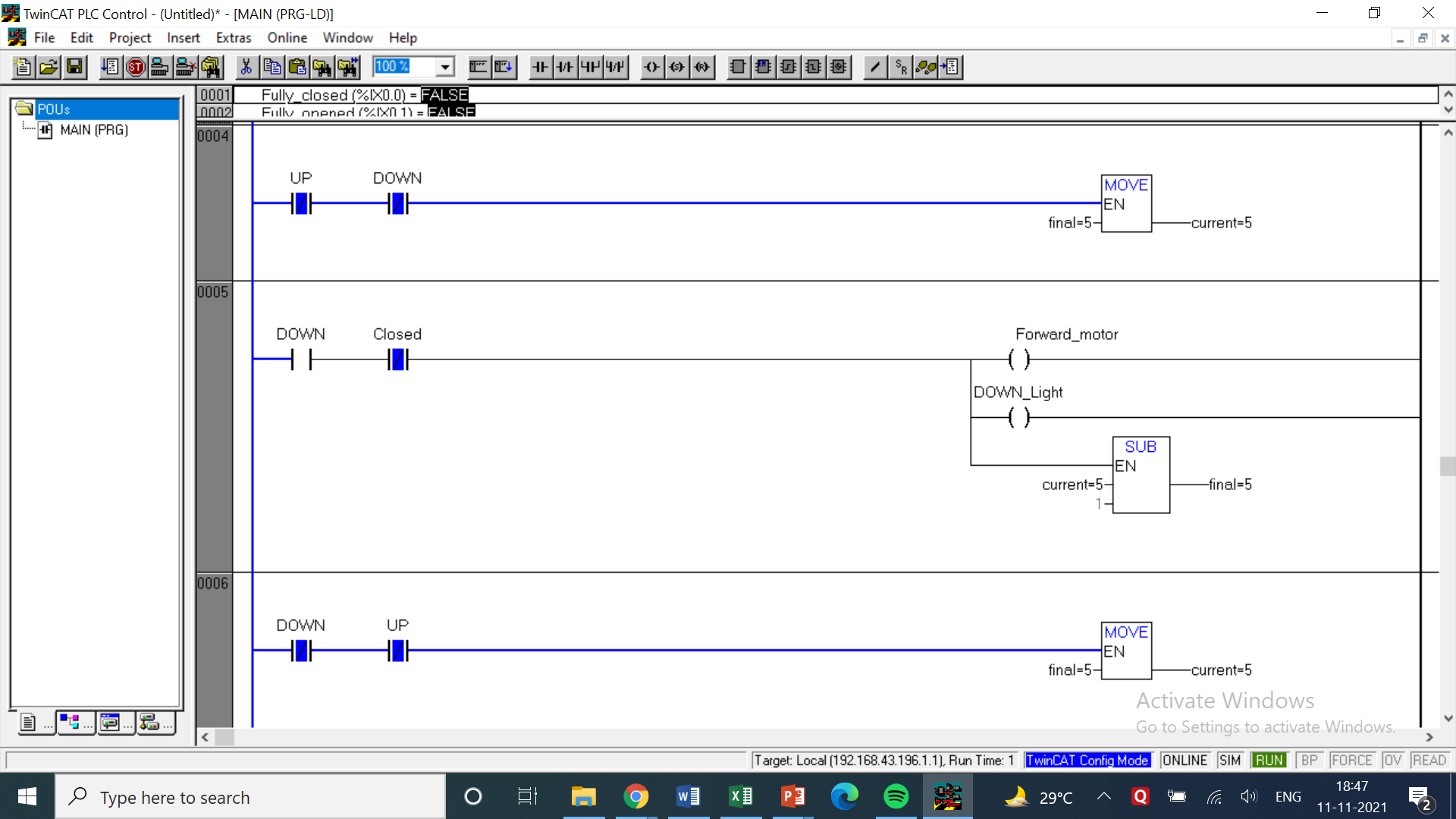
**Question 1**

1. Design a PLC ladder logic for the process mentioned below. A motorized overhead garage door is to be operated automatically to preset open and closed positions. The field devices include one of each of the following:

* Reversing motor contactor for the up and down directions.
* Normally closed down limit switch to sense when the door is fully closed. Normally closed up limit switch to sense when the door is fully opened. Normally open door up button for the up direction.
* Normally open door down button for the down direction.
* Normally closed door stop button for stopping the door.
* Red door ajar light to signal when the door is partially open.
* Green door open light to signal when the door is fully open.
* Yellow door closed light to signal when the door is fully closed.







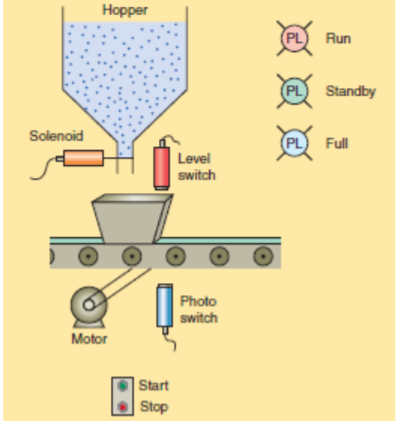
LOGIC -

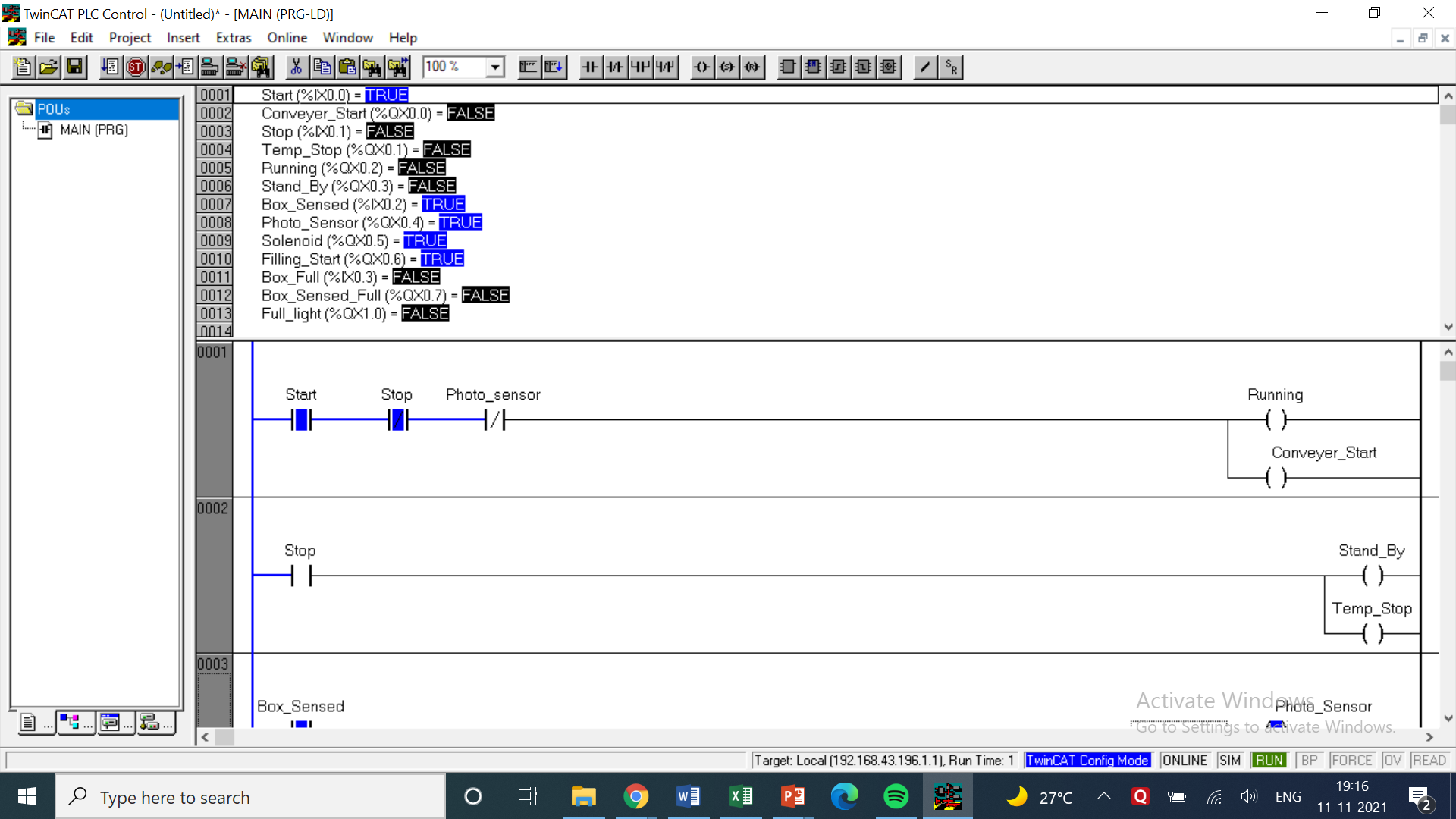
* The Logic here is simple but implementing took some thought. When the Up switch button is pressed the door will start opening and when the down button is pressed the door will start closing.
* To show how the process will be done, in the twincat ladder logic I added a counter to count upto 5 representing 5 levels of door.
* This means then when the up button will be pressed the door value will increase by 1 and when the down button will be pressed the door current value will be decreased by 1.
* If the current value of door reaches 0 then the fully closed door light will be turned on along with yellow light.
* If the current value of door reaches 5 then the fully opened door light will be turned on along with green light.
* When either 0 or 5 is reached the down and up buttons will stop function respectively and the value won’t change unless the opposite button is pushed.
* If the current value of door is anywhere between 0 and 5 then the partially open light will be turned on along with red light.
* This demonstrates how the control action will be performed. It can also be automated with some modifications.

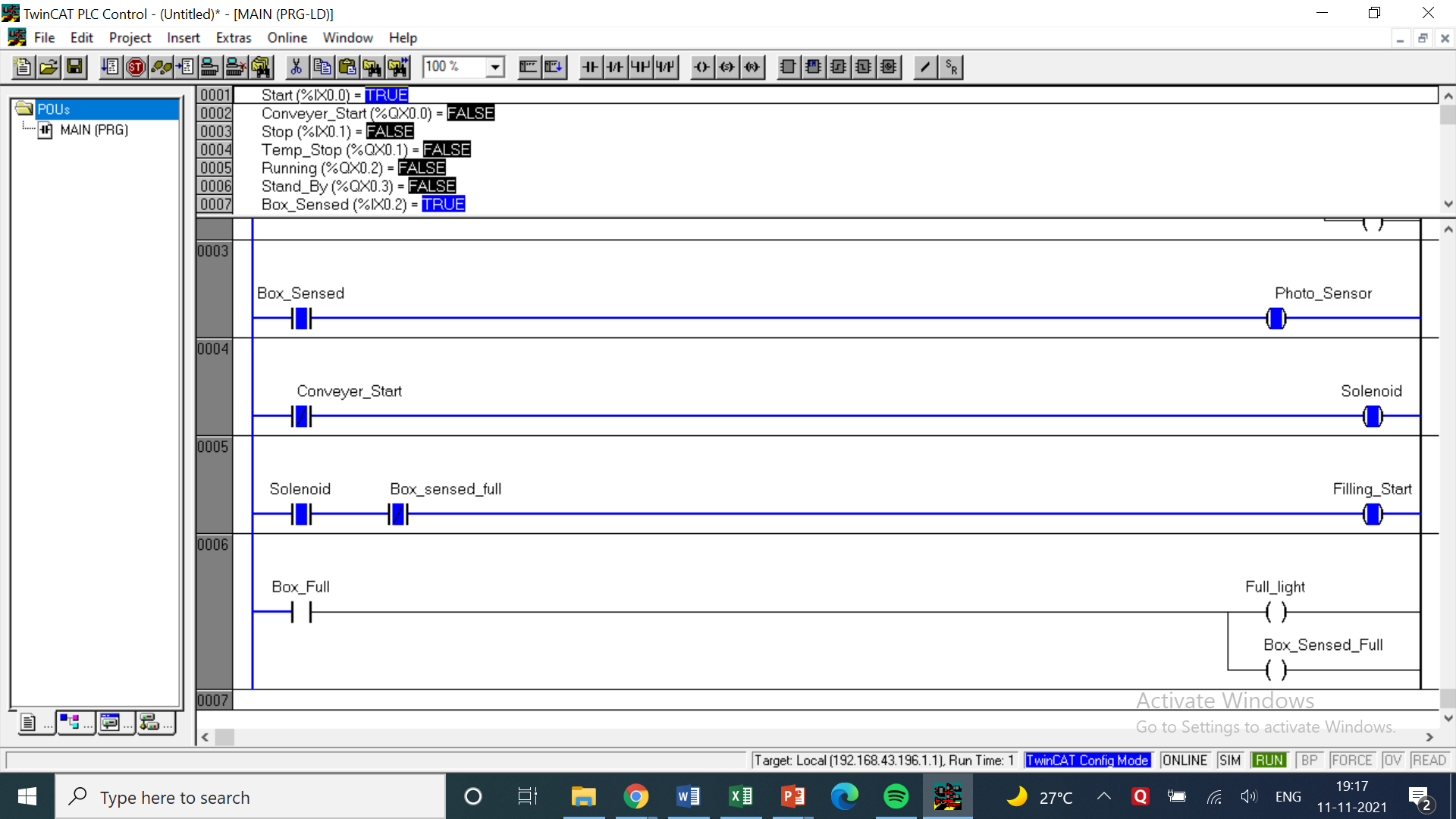
|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Fully closed | %ix0.0 | UP Light | %qx0.0 |
| Fully opened | %ix0.1 | Down Light | %qx0.1 |
| UP | %ix0.2 | Closed | %qx0.2 |
| DOWN | %ix0.3 | Opened | %qx0.3 |
| Reset | %ix7.7 | Partially Open | %qx0.5 |
| Stop | %ix0.4 | Red Light | %qx0.6 |
| Current (Int) | Initial Value = 0 | Green Light | %qx0.7 |
| Final (Int) | Initial Value = NA | Yellow Light | %qx1.0 |
| - | - | Forward Motor | %qx1.1 |
| - | - | Reversed Motor | %qx1.2 |

1. Design a PLC ladder logic for the continuous filling process as shown in figure 3. The process requires that boxes moving on a conveyor be automatically positioned and filled. The sequence of operation for the continuous filling operation is as follows:

* Start the conveyor when the start button is momentarily pressed.
* Stop the conveyor when the stop button is momentarily pressed.
* Energize the run status light when the process is operating.
* Energize the standby status light when the process is stopped.
* Stop the conveyor when the right edge of the box is first sensed by the photo sensor.
* With the box in position and the conveyor stopped, open the solenoid valve and allow the box to fill. Filling should stop when the level sensor goes true.
* Energize the full light when the box is full. The full light should remain energized until the box is moved clear of the photo sensor.







LOGIC

* The logic in this question is easier to understand as it directly follows the steps mentioned in the question.
* Start and stop buttons are placed in series with normally open and normally closed contacts respectively to complete and break the circuit.
* These switches are connected to the running light and will start the conveyer belt.
* When the stop switch is pressed the stand-by light and the temporary stop light will be turned on.
* It should be done by the sensor in real time but for simplicity a switch is placed which will turn on when a box is sensed and thus turning on the sensed light.
* When it is sensed it is kept as normally closed contact before the conveyer belt so the conveyer belt will stop moving. After being sensed the solenoid valve will also open and a light will be turned on indicating that.
* Similarly another switch is kept which will turn on if the box is sensed full and light will turn on. That light is kept as normally closed contact in the above solenoid rung which will break the circuit and close the solenoid.
* The full sensor will reset the conveyer belt and it will again start moving.

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Start | %ix0.0 | Conveyer Start | %qx0.0 |
| Stop | %ix0.1 | Temp Stop | %qx0.1 |
| Box Sensed | %ix0.2 | Running | %qx0.2 |
| Box Full | %ix0.3 | Stand By | %qx0.3 |
| - | - | Photo sensor | %qx0.4 |
| - | - | Solenoid | %qx0.5 |
| - | - | Filling starts | %qx0.6 |
| - | - | Box sensed full | %qx0.7 |

COMMENTS –

* The counters in most of the questions are denoted by the letter ‘C’ followed by the number of the counter or the letter.
* In case of UP and DOWN timers their names have been specified while mentioning the use of counters. 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.