**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**

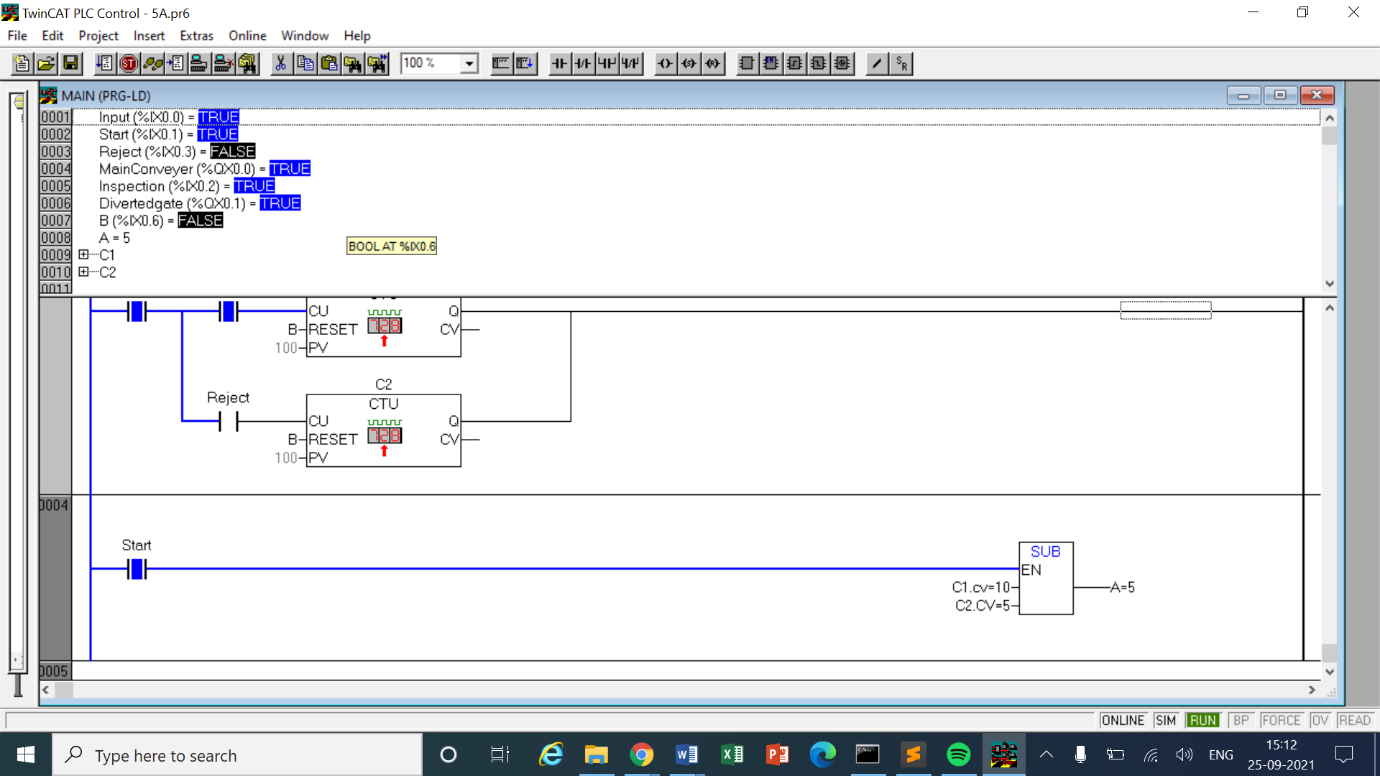
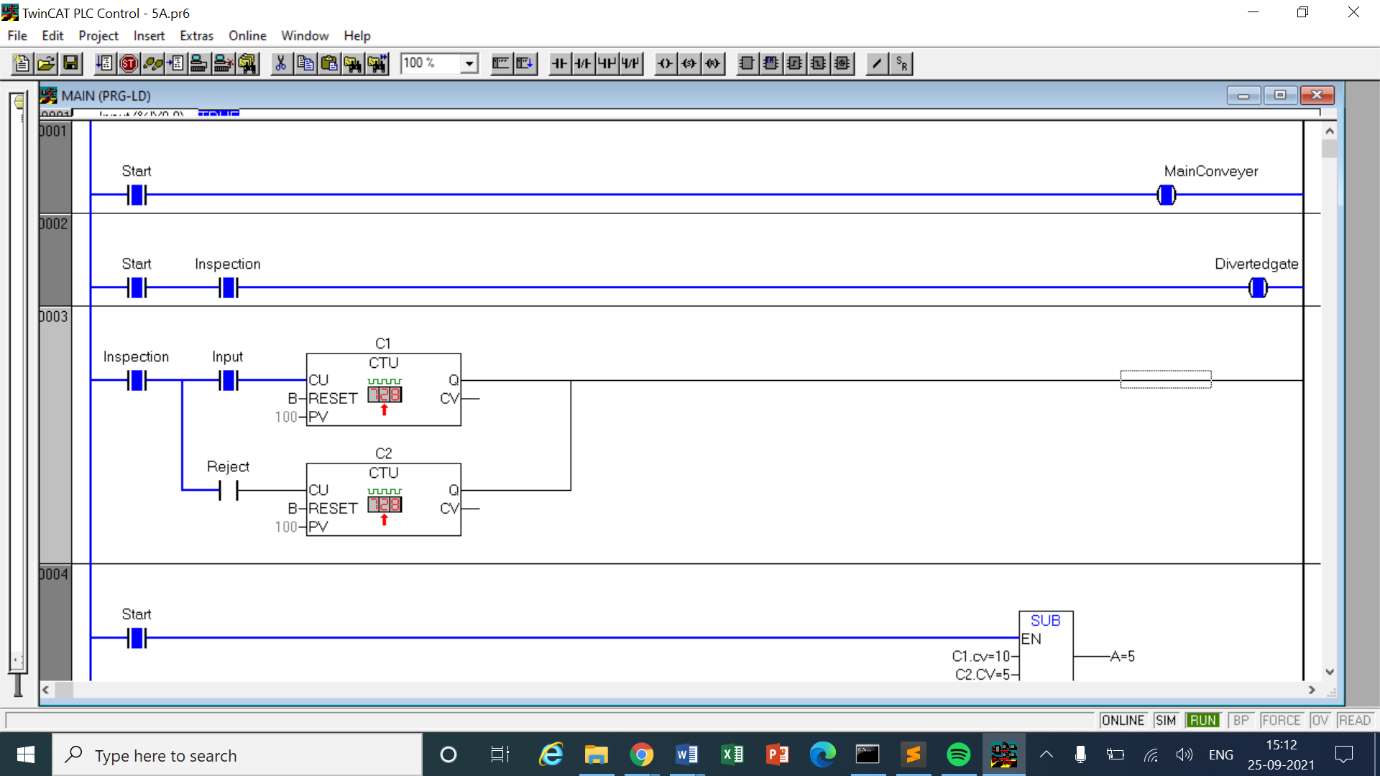
The programmable logic controller (PLC) is a solid state electronic device designed to replace electromechanical relays, timers, counters and sequences, by using a programmable memory for the internal storage of user oriented instructions for implementing specific functions such as logic sequencing timing, 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. The main conveyor with a diverter gate for defective parts to be fed onto a rejection conveyor. If a work piece fails inspection, the diverter gate is energized and the part is routed onto the reject conveyor. One counter is a situated on the main conveyor at the inspection station. The other counter is on the reject conveyor. Write a program using the subtraction function that will determine the number of parts that passed inspection.

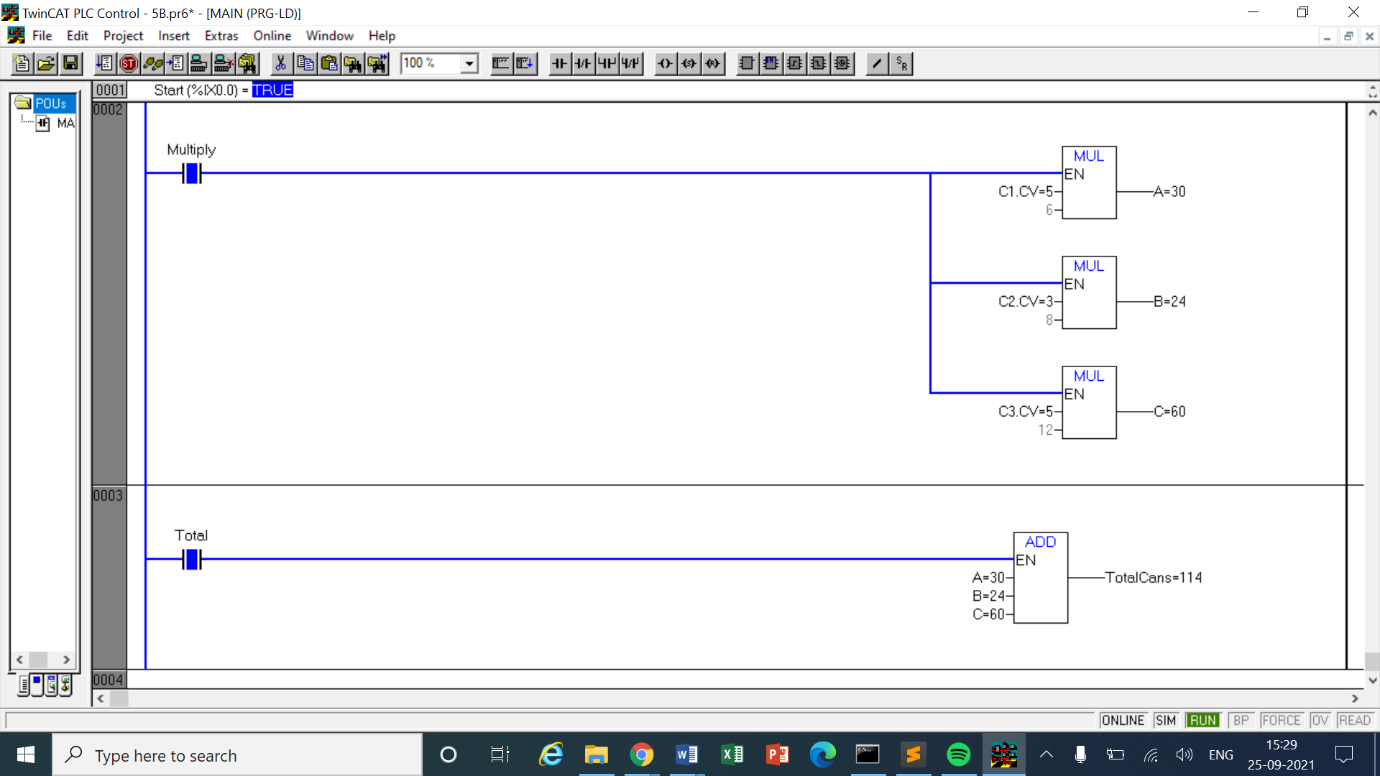
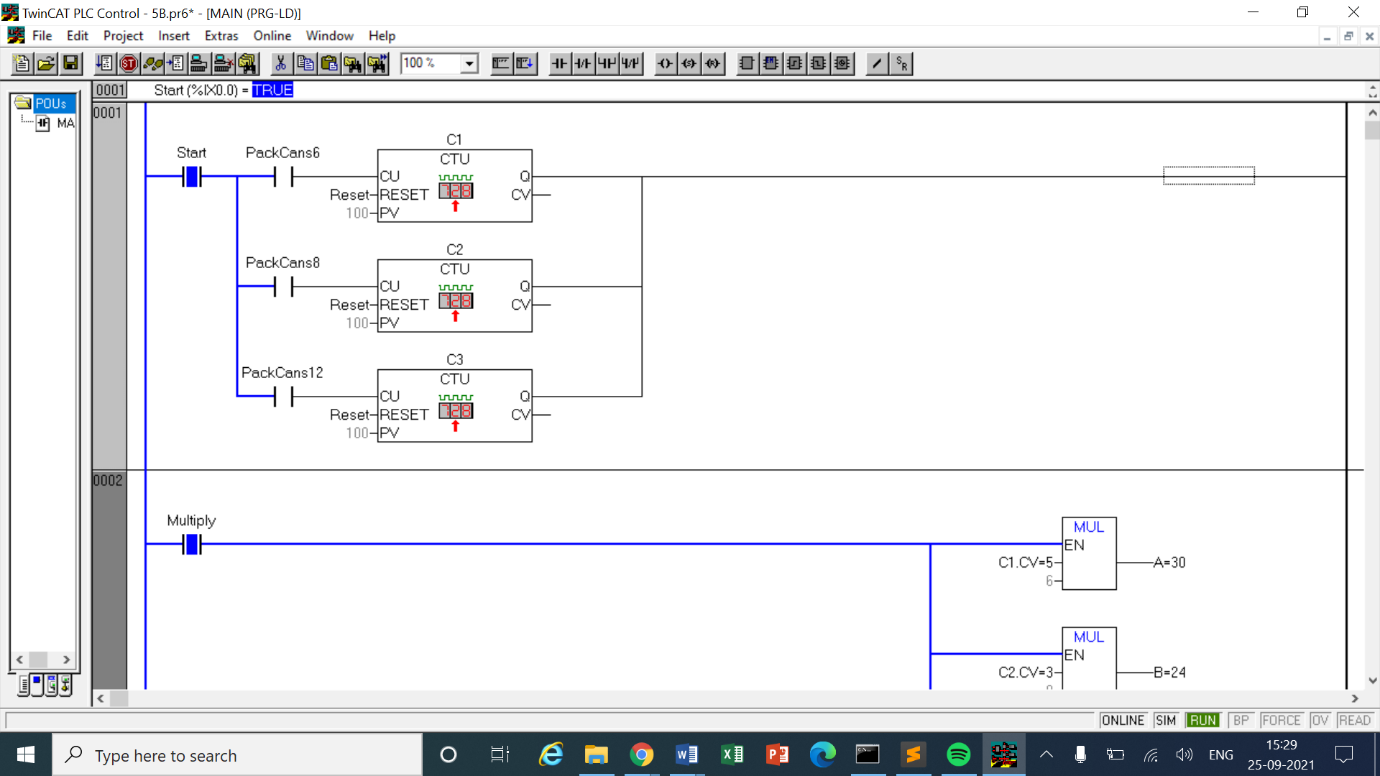
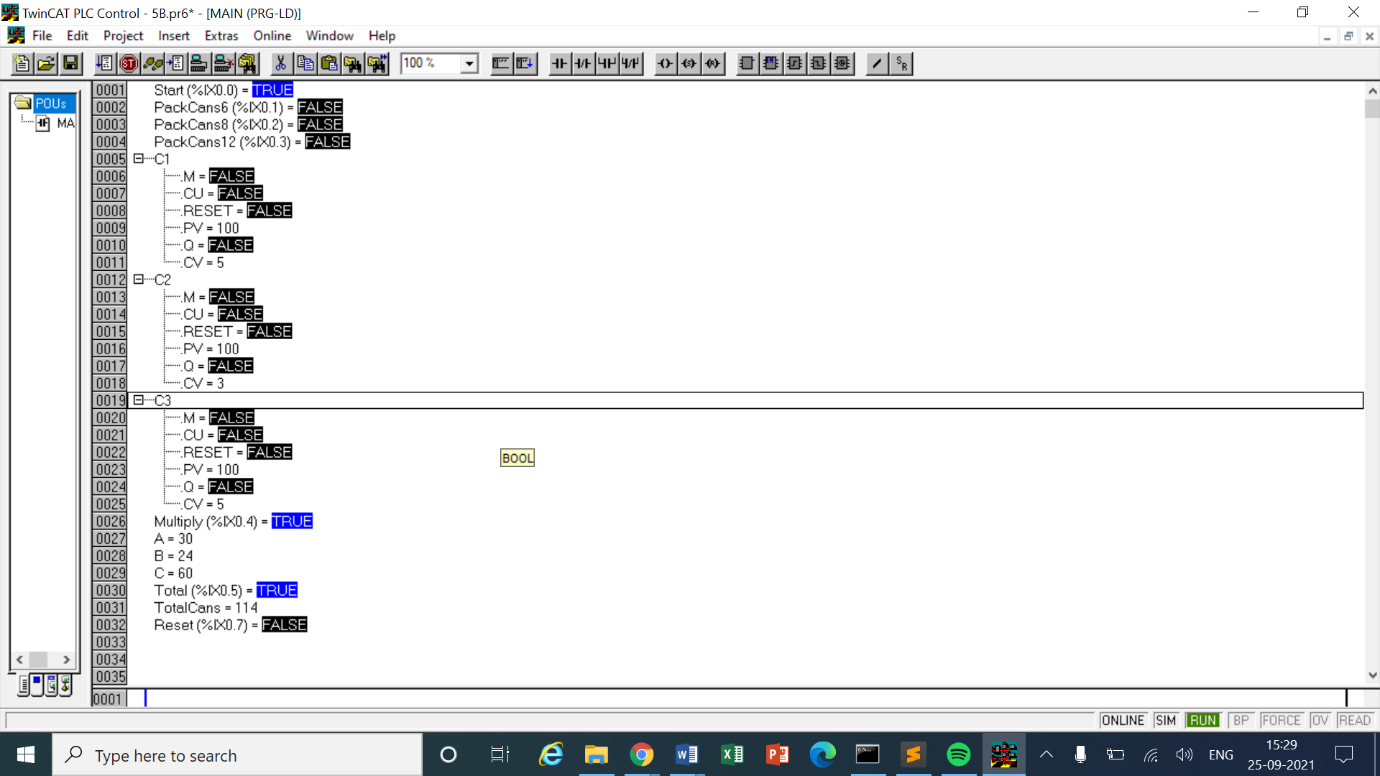


LOGIC -

* The start switch will turn on the conveyer belt which means the belt will start moving. In the 2nd network there is an inspection switch will should turn on the sensing system which further detects if an item is faulty or passes the inspection.
* After the inspection switch is turned on, in the 3rd network there are 2 Up Counters C1 and C2 which will respectively count the total and the defective pieces through the inspection lane.
* The preset value of both the counters are kept very high so we can ignore those.
* We know that each counter also carries a Current Value for itself. We take the current value of both the counters in a Subtraction function block in the 4th network.
* This function block will subtract the rejected ones from the total to get the number of items which passed the inspection test.

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Start | %ix0.1 | MainConveyer | %qx0.0 |
| Inspection | %ix0.2 | Divertedgate | %qx0.1 |
| Input | %ix0.0 | SUB | A  (INT variable) |
| Reject | %ix0.3 | - | - |
| Up Counter C1 | Preset value = 100  RESET = B | C1.CV | Current value of C1 |
| Up Counter C2 | Preset value = 100  RESET = B | C2.CV | Current value of C2 |
| B | %ix0.1 | - | - |

1. A conveyor has 6, 8 and 12 packs of canned soda entering it. Each size of an entering pack has an individual pack quantity counter feeding a PLC register. To know how many total cans are entering the conveyor, set up a program for multiplying and then adding to give a total can count.

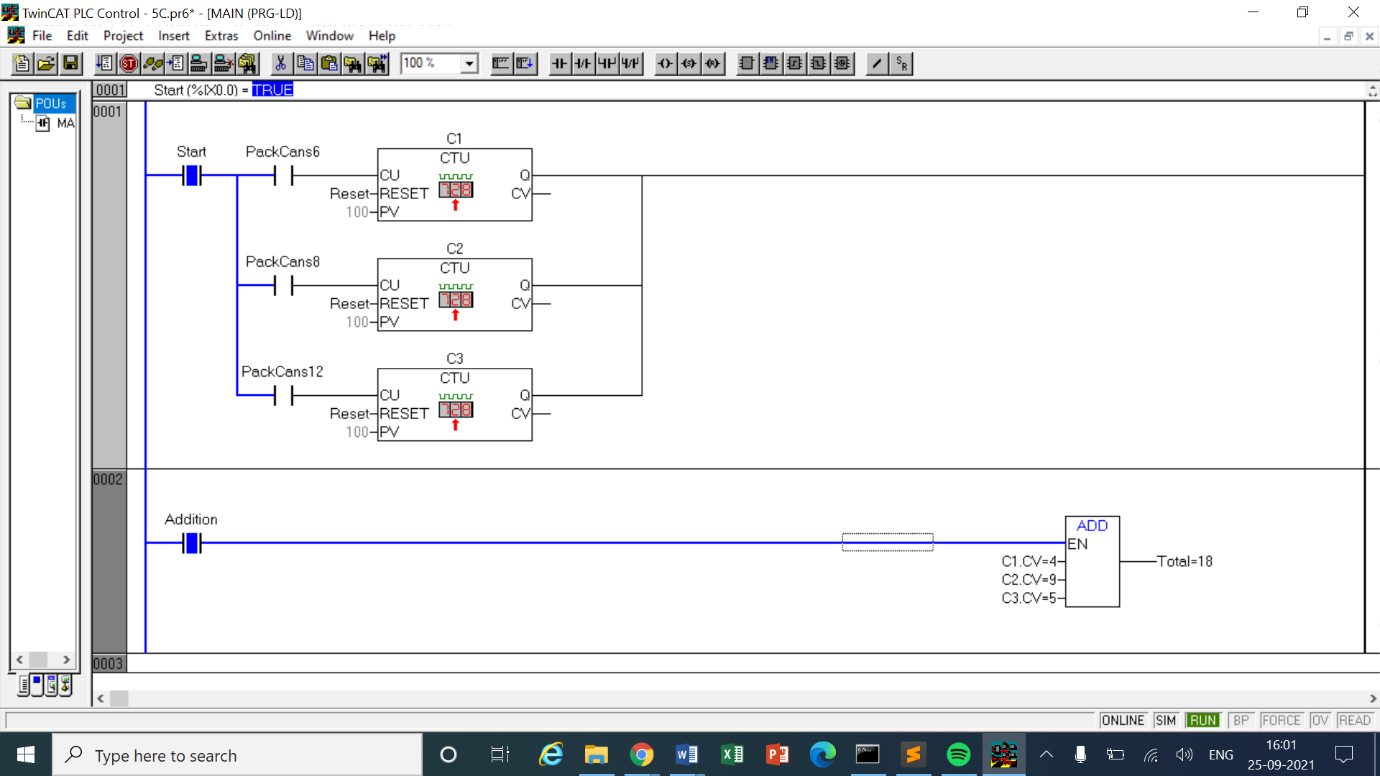
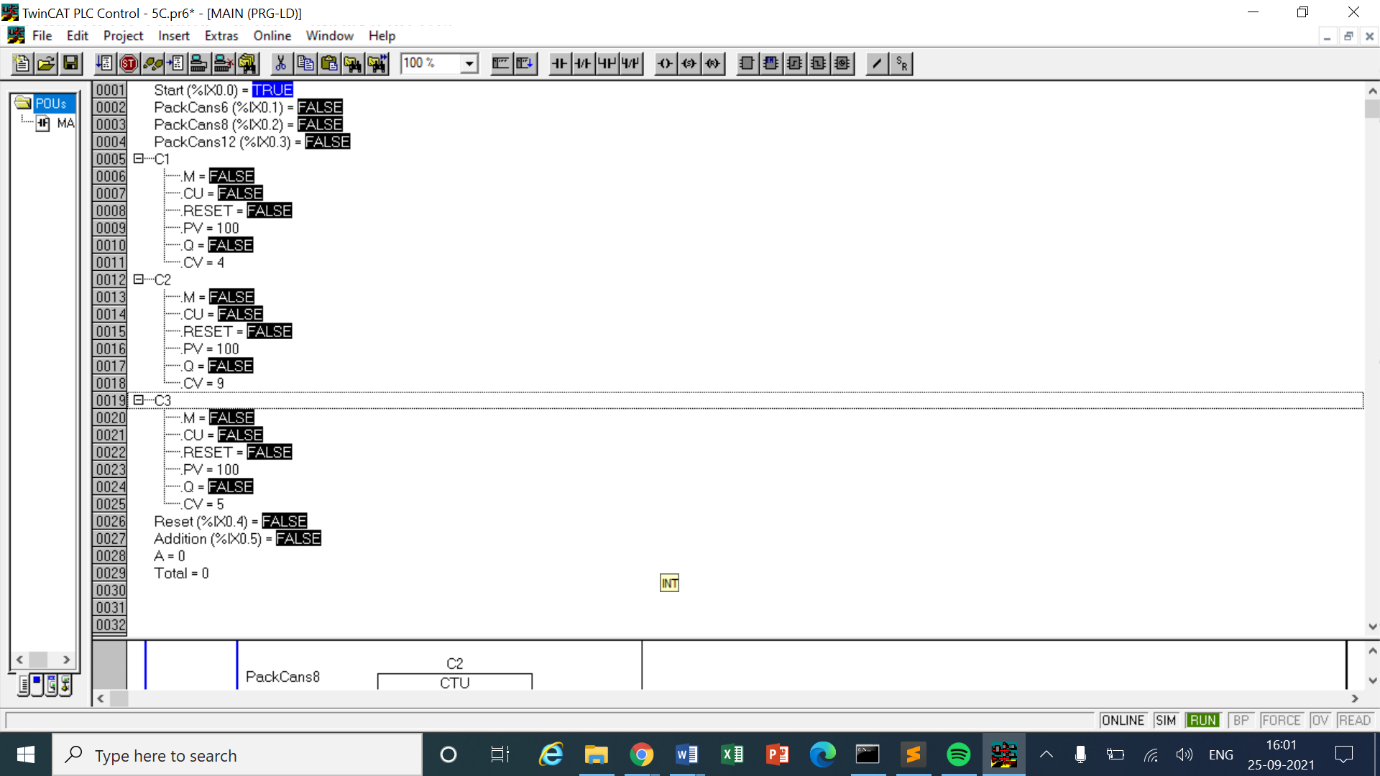


LOGIC

* From the question the given aspect is that there are 3 types of pack of cans entering the conveyer belt with 6,8 and 12 number of cans inside of them respectively.
* In the first network we take 3 Up Counters C1, C2 and C3 parallel with each other and each having respective switch to provide and input. This 1st network is initiated by a start switch.
* We take the current value of each of the Counters and use the multiply function in ‘Box with EN’. 3 multiplication function block will be needed to multiply with 6, 8 and 12 numbers respectively.
* The output of those function are stored in Integer type variables.
* In the last network with a switch ‘Total’ as an initiator we again take a function block but with addition.
* It will take the outputs of multiplication as 3 different inputs and add them and store in a variable.
* This variable will the total number of cans that we have now.

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Start | %ix0.0 | MainConveyer | %qx0.0 |
| PackCans6 | %ix0.1 | MUL | A (Int Varaible) |
| PackCans8 | %ix0.2 | MUL | B (Int Varaible) |
| PackCans12 | %ix0.3 | MUL | C (Int Varaible) |
| Up Counter C1 | Preset value = 100  RESET = Reset | C1.CV | Current value of C1 |
| Up Counter C2 | Preset value = 100  RESET = Reset | C2.CV | Current value of C2 |
| Up Counter C3 | Preset value = 100  RESET = Reset | C2.CV | Current value of C3 |
| Multiply | %ix0.4 | ADD | A + B + C = TotalCans (Int) |
| Total | %ix0.5 | - | - |
| Reset | %ix0.6 |  |  |

1. A main conveyor has three conveyors feeding it. One feeder puts 12 parts on the main conveyor and second puts a 6 parts and third puts a 8 parts. All feeder conveyors have counters that count the numbers of parts leaving them. Design a program to give a total count on the main conveyor.

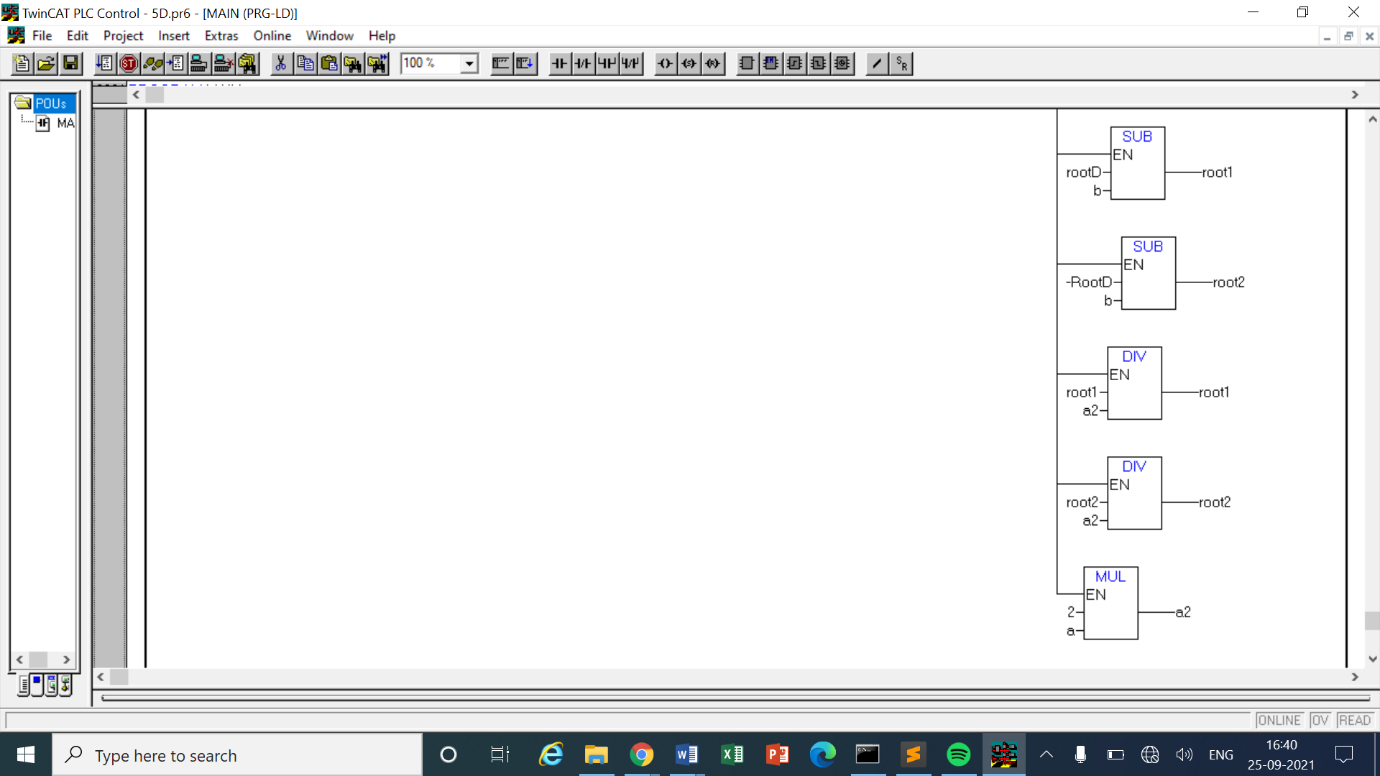
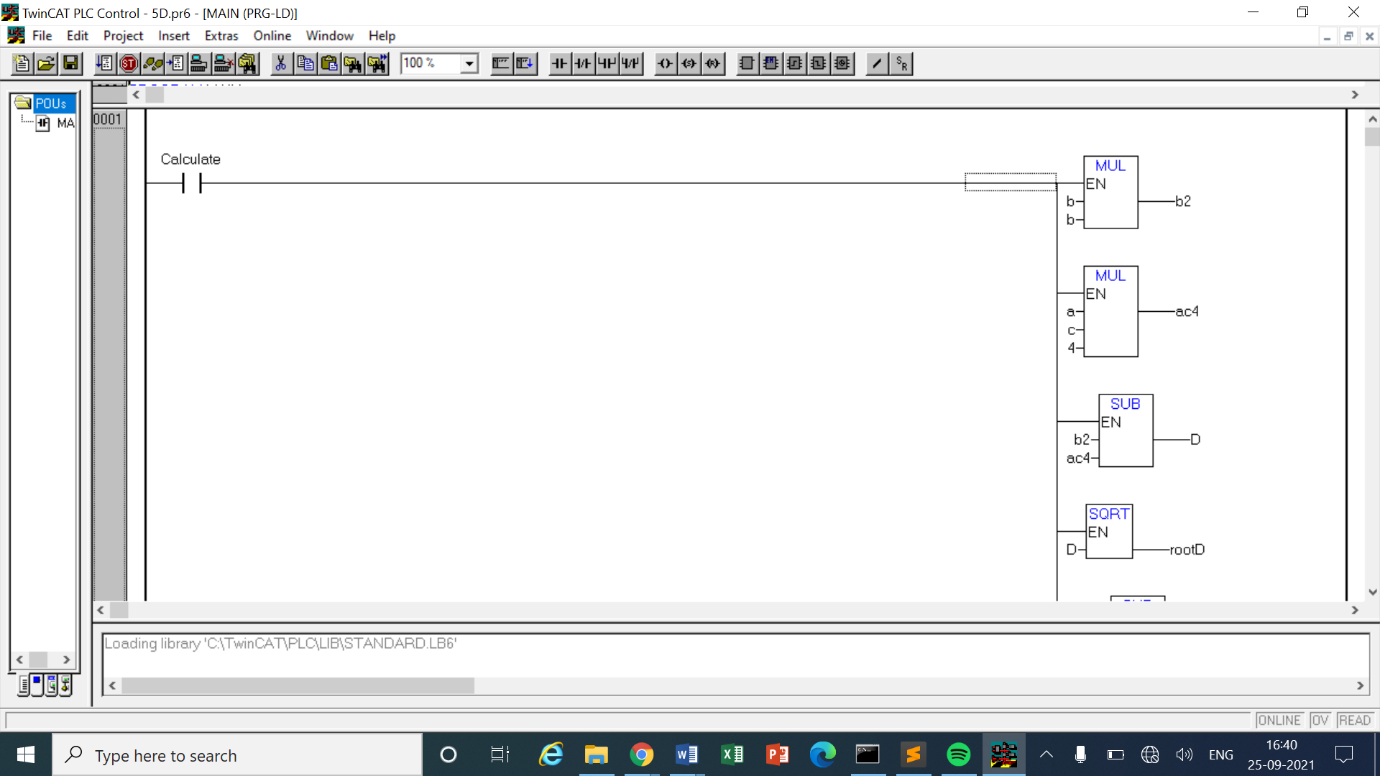


LOGIC

* The logic in this question is similar to previous question but easier. In the last question we calculated the total number of CANS as the output but in this question we only need to calculate the total number of PACKS.
* The first network is entirely similar to that in Q2 – there are 3 types of pack of cans entering the conveyer belt with 6, 8 and 12 number of cans inside of them respectively.
* In the first network we take 3 Up Counters C1, C2 and C3 parallel with each other and each having respective switch to provide and input. This 1st network is initiated by a start switch.
* We take the Current Value of each of these counter and use a ‘Box with EN’ of Addition type to add the outputs. This summation will tell us the total number of Packs that have been received with the total switch in the last network

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Start | %ix0.0 | MainConveyer | %qx0.0 |
| PackCans6 | %ix0.1 | - | - |
| PackCans8 | %ix0.2 | - | - |
| PackCans12 | %ix0.3 | - | - |
| Up Counter C1 | Preset value = 100  RESET = Reset | C1.CV | Current value of C1 |
| Up Counter C2 | Preset value = 100  RESET = Reset | C2.CV | Current value of C2 |
| Up Counter C3 | Preset value = 100  RESET = Reset | C2.CV | Current value of C3 |
| Addition | %ix0.5 | ADD | C1.CV + C2.CV + C3.Cv = Total (Int) |
| Reset | %ix0.4 | - | - |

1. Develop program to implement following math function.

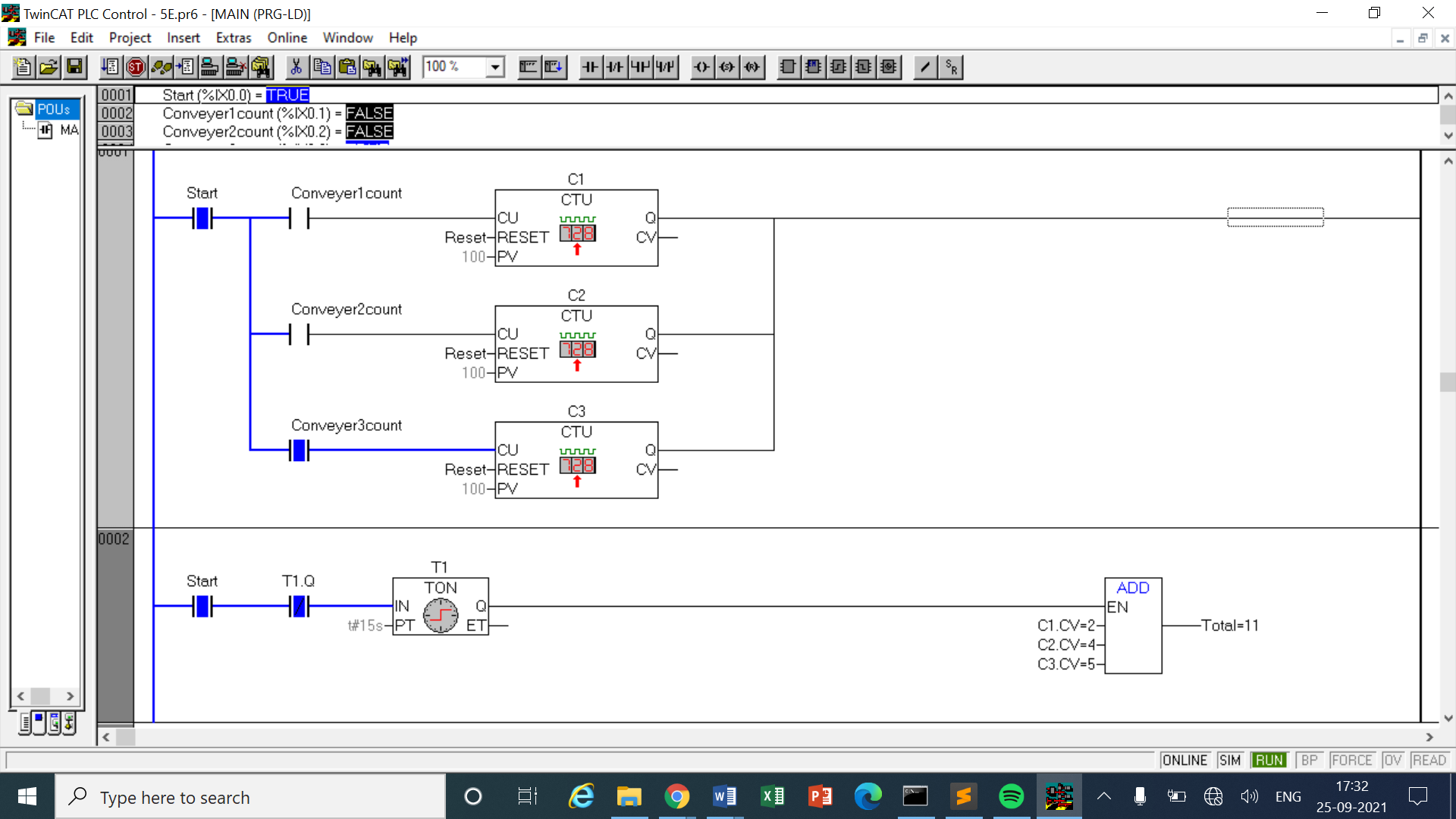


LOGIC

* This mathematical expression can be easily implemented. We will need Addition, Division and multiplication function blocks of ‘Box with EN’
* First user has to enter the value of a, b and c. Then the determinant will be calculated using (b^2 – 4ac) requiring 2 multiplication blocks and 1 subtraction block. After determinant as been calculated, its square root will be taken using yet another block.
* After that we use the negative of b and add and subtract the square root of D to get 2 different values, these values will be further divided by 2a to get the final value of the roots of the equation

|  |  |
| --- | --- |
| INPUT | |
| MUL | b x b | 16 |
| MUL | 4 x a x c | -180 |
| SUB | b^2 – 4ac = D | 196 |
| SQRT | Root (D) | 14 |
| SUB | -b – root (D) = root1 | -18 |
| ADD | -b + root (D) = root2 | 10 |
| MUL | 2 x a | 2 |
| DIV | Root1 / 2a | -9 |
| DIV | Root2 / 2a | 5 |

1. Three conveyors feed a main conveyor. The count from each feeder conveyor is fed into an input register in the PLC. Construct a PLC program to obtain the total count of parts on the main conveyor. Use timer to update total every 15 seconds. If the total is 100 then stop the process and reset the all the counter and input register.

****

LOGIC

* This code is similar to question 3 but the difference lies in the aspect that the total number of counts only updates after 15 seconds of intervals whether any input was given in those 15 seconds or not.
* In the 1st network there are 3 Up counters C1, C2 and C3 with preset values very high such as 100.
* Each of these counters have a count input switch which gives the signal to count.
* The first network is initiated by a start switch at the front.
* In the second network there is a normally closed output of timer t1 after the timer t1 of delay 15 seconds is connected. Which means that the first output will be shown after 15 seconds and after every 15 seconds the circuit will be completed hence updating the output

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Start | %ix0.0 | MainConveyer | %qx0.0 |
| Conveyer1Count | %ix0.1 | - | - |
| Conveyer2Count | %ix0.2 | - | - |
| Conveyer3Count | %ix0.3 | - | - |
| Up Counter C1 | Preset value = 100  RESET = Reset | C1.CV | Current value of C1 |
| Up Counter C2 | Preset value = 100  RESET = Reset | C2.CV | Current value of C2 |
| Up Counter C3 | Preset value = 100  RESET = Reset | C2.CV | Current value of C3 |
| Addition | %ix0.5 | ADD | C1.CV + C2.CV + C3.Cv = Total (Int) |
| On Timer t1 | Delay time = 15s | T1.Q | Output of t1 |
| Reset | %ix0.4 | - | - |

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.

**CONCLUSION**

* In this experiment we learnt about different types of mathematical operations that could be performed in TwinCAT using ‘Box with EN’ function block. Some of the types of operations include – addition, multiplication, division, square root and many more. We learned the importance of usinf these operation in real life as counts and remaining products are needed to be kept in check.