PROJECT REPORT
AUTOMATION OF TWO CHEMICAL MIXING TANK USING PLC - APPLYING OT SECURITY

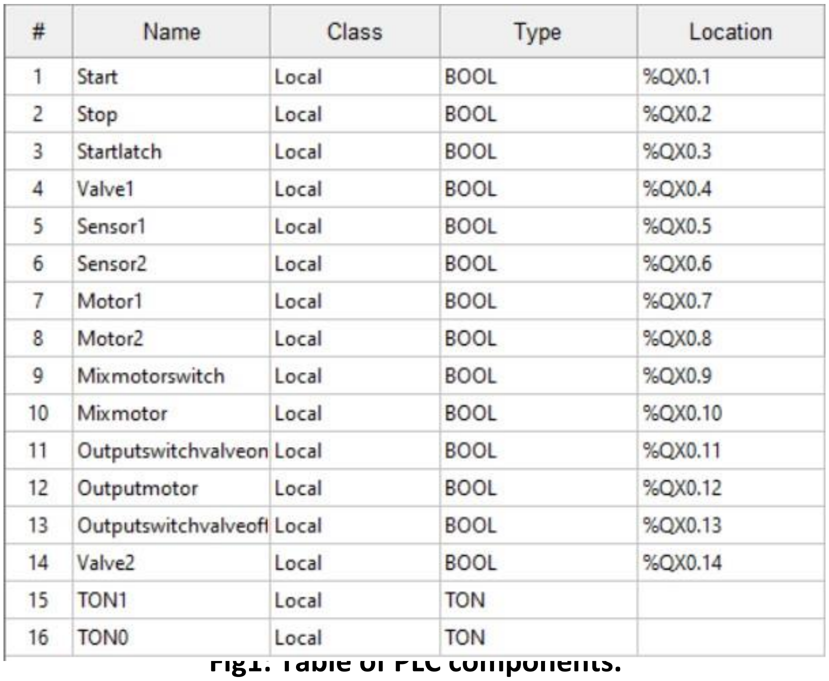
Problem Statement:

In this project, we have to mix two chemicals automatically in a tank where there should be two valve controlling the flow of two chemicals. After two chemicals are brought into the mixing tank the stirring motor should be turned on automatically for few minutes and open the output valve. We need to control the two valves in a tank automatically and stirring motor and the output valve using PLC programming with ladder logic. The whole process should be automated and no human interaction should be in need to control the process. The control system should also have emergency, and control measures by which the system can be managed by humans manually.

Solution:

In this project, I have automate and control the mixing of two different chemicals in a tank. Initially, the tank is empty, the two valve for two different chemicals will be open, when sensor1 detects the chemical on certain level it will close valve1 which will stop the flow of chemical1. Later, when sensor 2 sense the chemical on its level it will close valve2 which will stop the flow of chemical2. After this process, the stirring motor will be turned on automatically for 5 minutes and will stop and it will open output valve automatically and the mixed chemical will flow out for the use.

PLC Components (Input/Output):



#	Name	Class	Type	Location
1	Start	Local	BOOL	%QX0.1
2	Stop	Local	BOOL	%QX0.2
3	Startlatch	Local	BOOL	%QX0.3
4	Valve1	Local	BOOL	%QX0.4
5	Sensor1	Local	BOOL	%QX0.5
6	Sensor2	Local	BOOL	%QX0.6
7	Motor1	Local	BOOL	%QX0.7
8	Motor2	Local	BOOL	%QX0.8
9	Mixmotorswitch	Local	BOOL	%QX0.9
10	Mixmotor	Local	BOOL	%QX0.10
11	Outputswitchvalveon	Local	BOOL	%QX0.11
12	Outputmotor	Local	BOOL	%QX0.12
13	Outputswitchvalveof	Local	BOOL	%QX0.13
14	Valve2	Local	BOOL	%QX0.14
15	TON1	Local	TON	
16	TON0	Local	TON	

Fig1: Table of PLC components.

List of Inputs:

I1: Start
I2: Stop
I3: Startlatch
I4: Sensor1
I5: Sensor2
I6: Outputswitchvalve
I7: Mixmotorswitch

List of Output:

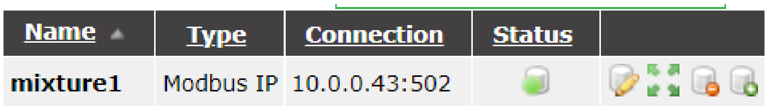
O1: Valve1
O2: Valve2
O3: Motor1
O4: Motor2
O5: Outputmotor
O6: Mixmotor

List of Timer:

TON0: Timer1
TON1: Timer2

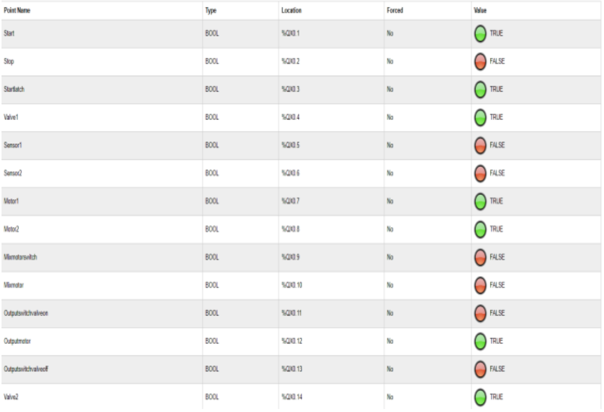
DATA SOURCES:

Initially, I have created a Data source with MODBUS IP configuration named mixture1. In this I have created several points according to the input/output components and then I enabled them all to sync with the HMI (Human-Machine Interface)

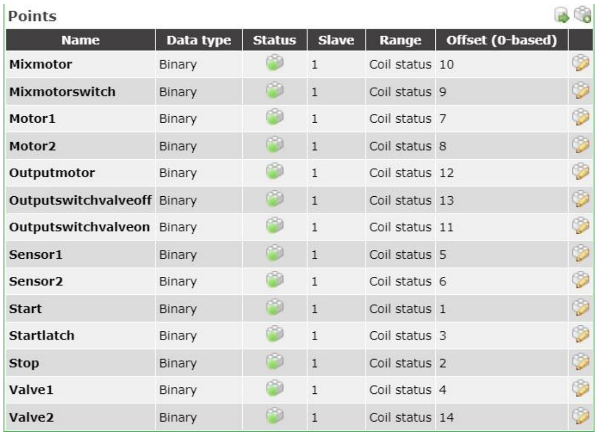


Name ▲	Type	Connection	Status	
mixture1	Modbus IP	10.0.0.43:502	●	

Fig: Data Source for mixture1 on SCADABR



Point Name	Type	Location	Forced	Value
Start	BOOL	%QX0.1	No	🟢 TRUE
Stop	BOOL	%QX0.2	No	🔴 FALSE
Startlatch	BOOL	%QX0.3	No	🟢 TRUE
Valve1	BOOL	%QX0.4	No	🟢 TRUE
Sensor1	BOOL	%QX0.5	No	🔴 FALSE
Sensor2	BOOL	%QX0.6	No	🔴 FALSE
Motor1	BOOL	%QX0.7	No	🟢 TRUE
Motor2	BOOL	%QX0.8	No	🟢 TRUE
Mixmotorswitch	BOOL	%QX0.9	No	🔴 FALSE
Mixmotor	BOOL	%QX0.10	No	🔴 FALSE
Outputswitchvalveon	BOOL	%QX0.11	No	🔴 FALSE
Outputmotor	BOOL	%QX0.12	No	🟢 TRUE
Outputswitchvalveoff	BOOL	%QX0.13	No	🔴 FALSE
Valve2	BOOL	%QX0.14	No	🟢 TRUE



Points

Name	Data type	Status	Slave	Range	Offset (0-based)
Mixmotor	Binary	●	1	Coil status	10
Mixmotorswitch	Binary	●	1	Coil status	9
Motor1	Binary	●	1	Coil status	7
Motor2	Binary	●	1	Coil status	8
Outputmotor	Binary	●	1	Coil status	12
Outputswitchvalveoff	Binary	●	1	Coil status	13
Outputswitchvalveon	Binary	●	1	Coil status	11
Sensor1	Binary	●	1	Coil status	5
Sensor2	Binary	●	1	Coil status	6
Start	Binary	●	1	Coil status	1
Startlatch	Binary	●	1	Coil status	3
Stop	Binary	●	1	Coil status	2
Valve1	Binary	●	1	Coil status	4
Valve2	Binary	●	1	Coil status	14

Fig: Real-Time DataSource for mixture1 on SCADABR

PLC LADDER LOGIC:

To design the Ladder Logic for Automated two chemical mixer, I have created 8 Rungs in total for the system. Each Rung has its own functionality and roles.

Rung 1: In this Rung, I have used normal logic for start and stop, when the Start is true the startlatch will be turn on and the cycle will be on. And when the Negated Stop is False the system will be turned off completely.

Rung 2: In this Rung, When the Valve 1 is True, it will check if the Sensor1 and Sensor2 is not activated, if not then it will turn on the Motor1.

Rung 3: In this Rung, if Sensor2 is activated, it will close the valve1 and Motor1.

Rung 4: In this Rung, When the Valve 2 is True, it will check if the Motor1, and Sensor2 is not activated, if not then it will turn on the Motor2.

Rung 5: In this Rung, if Sensor1 is activated, it will close the valve2 and Motor2.

Rung 6: If the Mixmotor is True, then Mixmotor will be turned on.

Rung 7: When the Outputsitchvalve is true, the outputmotor will be turned on.

Rung 8: When the outputswitchalve is true the Mixmotor will be turned off.

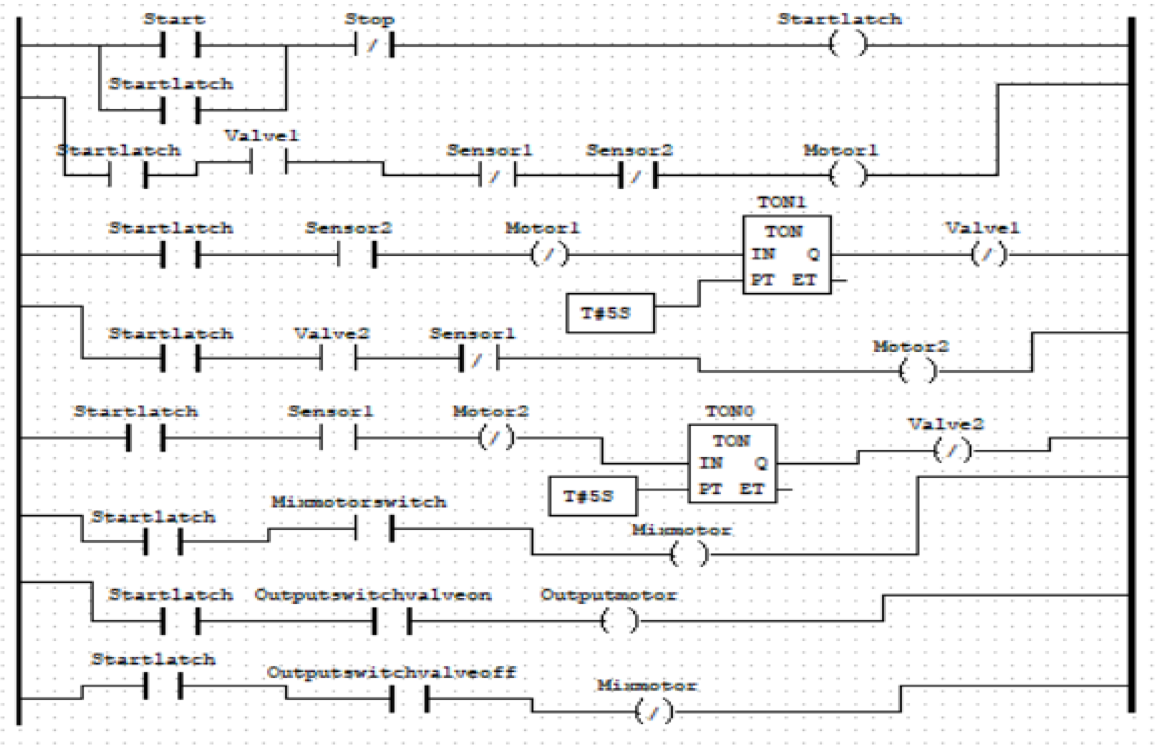
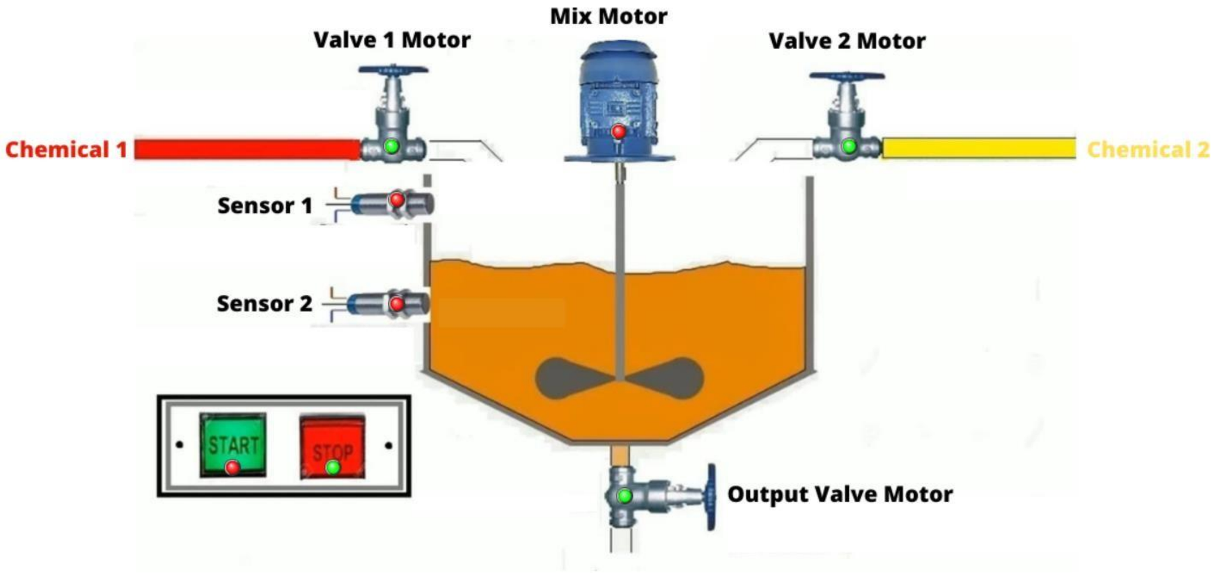


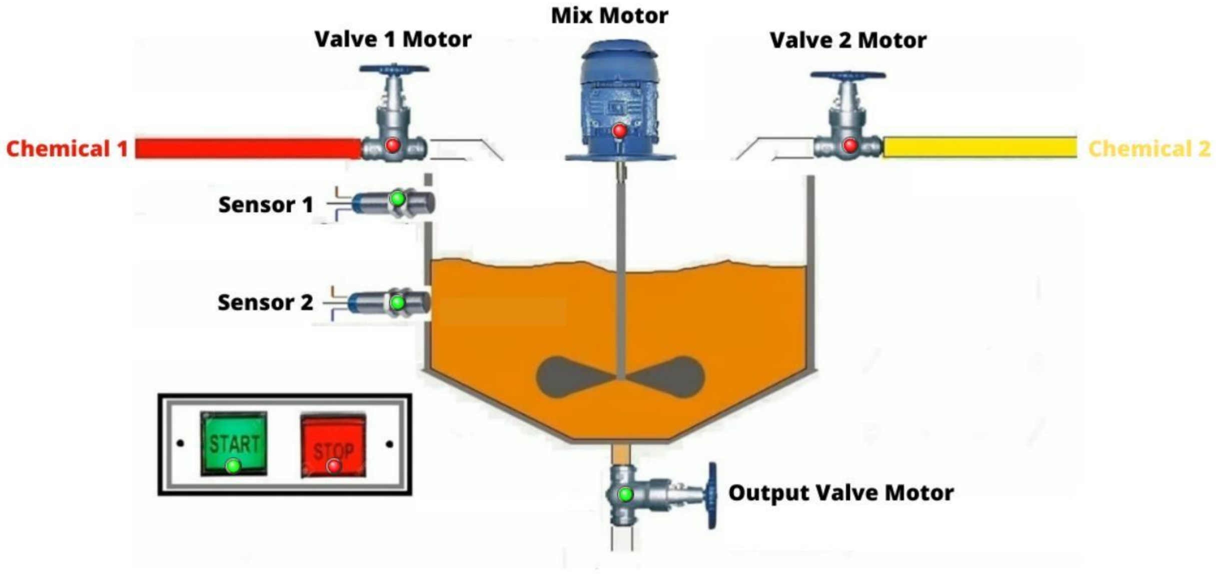
Fig2: Ladder Logic Diagram

Final SCADABR HMI:

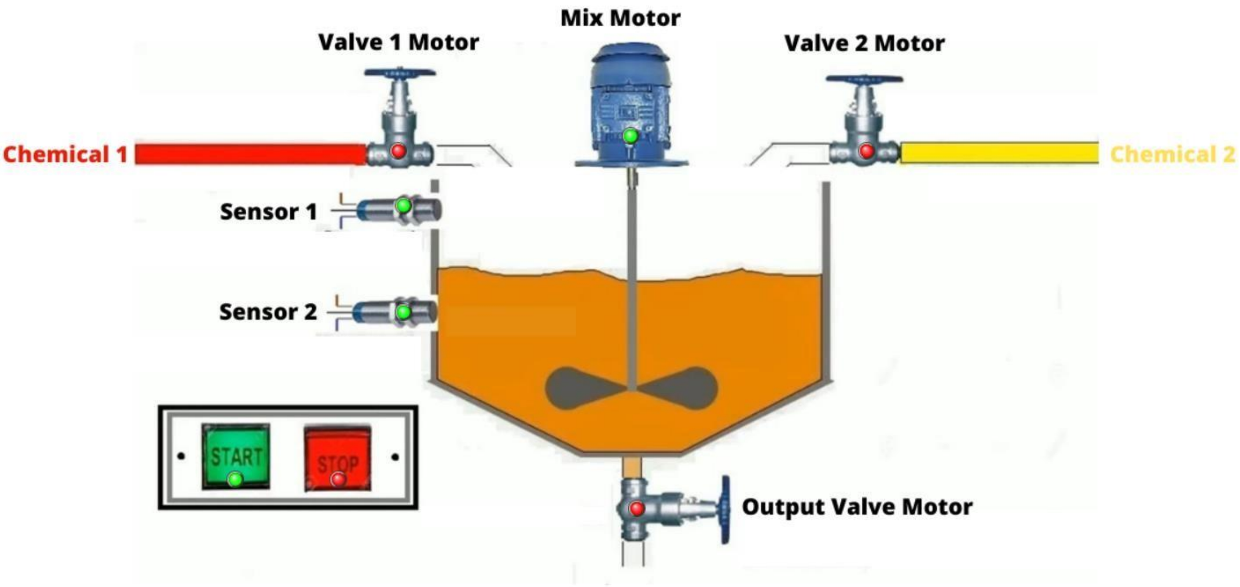
State 1: The system is On, and Valve1 and Valve2 is also turned on and two separate chemical from different pipe is beign filled into the mixing tank.



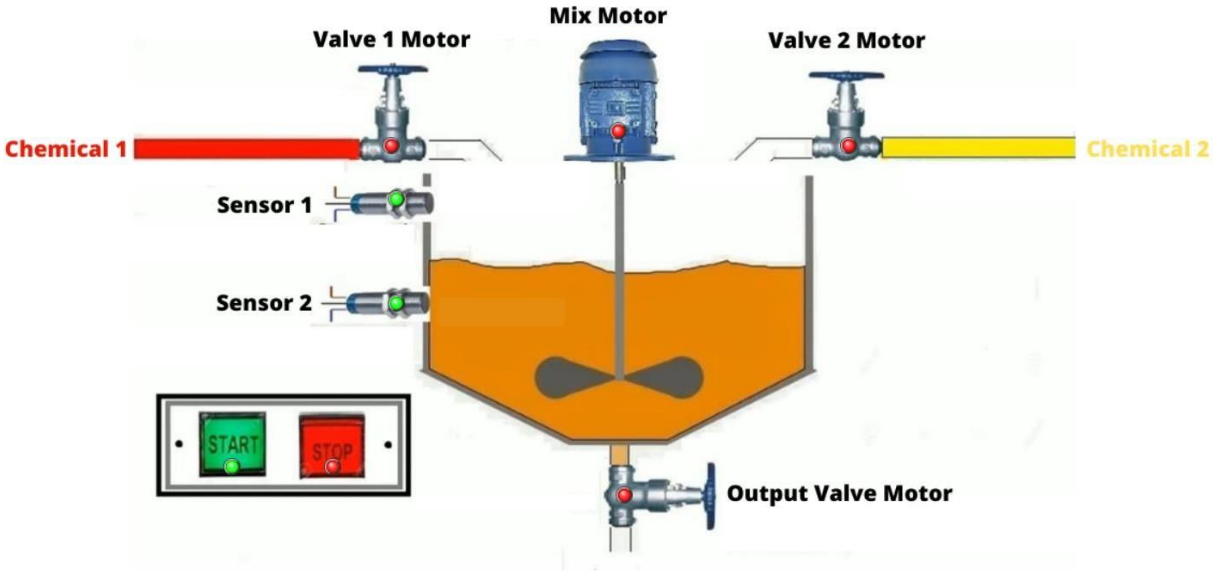
State 2: The tank is filled with two different chemicals, Sensor1 and Sensor2 is activated by which Valve1 and Valve2 is turned off automatically.



State 3: Mixing Motor is turned on and two chemical are mixing in the tank.

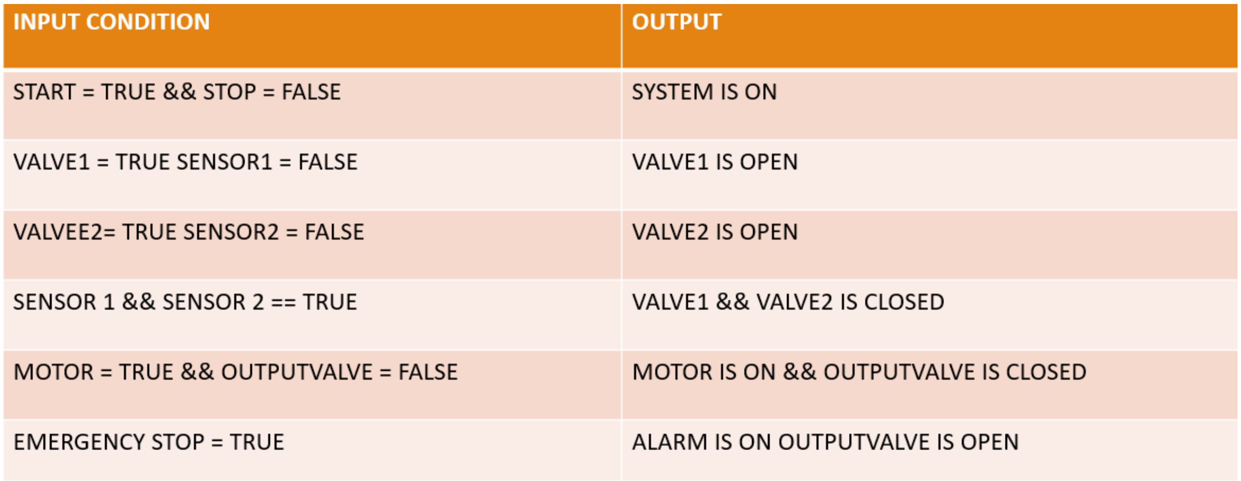


State 4: Mixing Motor is turned of automatically after 10 min.



TEST CASES:

All the given Test Cases are implemented successfully in the control system.



INPUT CONDITION	OUTPUT
START = TRUE && STOP = FALSE	SYSTEM IS ON
VALVE1 = TRUE SENSOR1 = FALSE	VALVE1 IS OPEN
VALVEE2= TRUE SENSOR2 = FALSE	VALVE2 IS OPEN
SENSOR 1 && SENSOR 2 == TRUE	VALVE1 && VALVE2 IS CLOSED
MOTOR = TRUE && OUTPUTVALVE = FALSE	MOTOR IS ON && OUTPUTVALVE IS CLOSED
EMERGENCY STOP = TRUE	ALARM IS ON OUTPUTVALVE IS OPEN

PLATFORM USED:

- OPEN PLC EDITOR
- OPEN PLC RUNTIME
- SCADABR

CONCLUSION:

The above explained Chemical Mixing process is for example only. It may vary from real time. I have used this example program to understand the working of timers, memory bit and interlocking concept function in PLC. Also, I have successfully understood how to create and design ladder logic for any given problem statement.