Write a self-contained 61131-3 structured text program for a coffee machine.

This machine contains three tanks, each for coffee, milk, and mixer, and three valves, one for the coffee, one for the milk, and the last valve for the output. This coffee machine contains a mixer, which is required to mix the coffee and milk properly to create the best output, The mixer is designed to work for four seconds after which it will automatically stop and the output valve will open to get the output.

The mixer is designed with a timer and will automatically start when the mixer tank reaches the maximum level, the tank is attached to the mixer as both the valves of coffee and milk are open and it started to fill the mixer and then the tank level comes into play with the preset maximum level, the mixer tank can be filled up to 130ml as it reaches the maximum level, the valves of coffee and milk will close and then it will start the mixing.

The coffee machine contains four Push buttons. Button one is for emergency stop, Button two is to start the machine, Button three is for Coffee and Milk and the last button is for Coffee only. The emergency stop is use as a safety button if anything goes wrong in the machine like if any of the valves not open, if the tank level does not work properly, if the mixer fails to work properly or any of the unexpected events occur, then there is a need of emergency stop to stop the machine instantly.

ST

Copy code

```
PROGRAM CoffeeMachine
VAR
    EmergencyStop, StartMachine, CoffeeMilk, CoffeeOnly : BOOL;
    CoffeeValve, MilkValve, OutputValve, MixerOn : BOOL;
    TankCoffee, TankMilk, TankMixer : REAL;
    MixerTimer : TIME;
    TimerStarted : BOOL;
    MixingTime : TIME := T#4s;
    MixerMaxLevel : REAL := 130.0;
END_VAR

METHOD StartValves : BOOL
    StartValves := StartMachine AND NOT EmergencyStop;
END_METHOD

METHOD StopValves : BOOL
    StopValves := EmergencyStop OR (TankMixer >= MixerMaxLevel);
END_METHOD

METHOD StartMixer : BOOL
    StartMixer := (TankMixer >= MixerMaxLevel) AND NOT TimerStarted;
END_METHOD

METHOD StopMixer : BOOL
    StopMixer := MixerTimer >= MixingTime;
END_METHOD

METHOD OpenOutputValve : BOOL
    OpenOutputValve := StopMixer AND NOT EmergencyStop;
END_METHOD

END_PROGRAM

PROGRAM Main
VAR
    CoffeeMachineInstance : CoffeeMachine;
END_VAR

CoffeeMachineInstance.StartMachine := StartMachine;
CoffeeMachineInstance.EmergencyStop := EmergencyStop;
CoffeeMachineInstance.CoffeeMilk := CoffeeMilk;
CoffeeMachineInstance.CoffeeOnly := CoffeeOnly;

IF CoffeeMachineInstance.StartValves() THEN
    CoffeeMachineInstance.CoffeeValve := CoffeeMachineInstance.CoffeeMilk O
    CoffeeMachineInstance.MilkValve := CoffeeMachineInstance.CoffeeMilk;
ELSE
    CoffeeMachineInstance.CoffeeValve := FALSE;
    CoffeeMachineInstance.MilkValve := FALSE;
END_IF

IF CoffeeMachineInstance.StopValves() THEN
    CoffeeMachineInstance.CoffeeValve := FALSE;
    CoffeeMachineInstance.MilkValve := FALSE;
END_IF

IF CoffeeMachineInstance.StartMixer() THEN
    CoffeeMachineInstance.MixerOn := TRUE;
    CoffeeMachineInstance.TimerStarted := TRUE;
    CoffeeMachineInstance.MixerTimer := CoffeeMachineInstance.MixerTimer +
ELSE
    IF CoffeeMachineInstance.StopMixer() THEN
        CoffeeMachineInstance.MixerOn := FALSE;
        CoffeeMachineInstance.TimerStarted := FALSE;
        CoffeeMachineInstance.MixerTimer := T#0s;
    END_IF
END_IF

CoffeeMachineInstance.OutputValve := CoffeeMachineInstance.OpenOutputValve(

END_PROGRAM
```

continue

To implement the coffee machine program, you should now create a PLC program in the desired development environment, such as TIA Portal, CODESYS, or RSLogix 5000, and import the structured text code provided above.

Once you have created a new project, follow these steps:

1. Create a new PLC program or task, if necessary.
2. Create a new POU (Program Organization Unit) and name it "Main" or similar.
3. Change the POU language to Structured Text (ST).
4. Copy the Structured Text code provided above into the POU.
5. Create a new POU with the language set to Structured Text (ST) and name it "CoffeeMachine" or similar.
6. Copy the Structured Text code from the `PROGRAM CoffeeMachine` section provided above into the new POU.
7. If needed, adjust the data types and variable names to match your PLC's conventions and supported data types.
8. Compile and download the program to your PLC.
9. Test the program on your PLC by simulating the input conditions or connecting the PLC to the actual coffee machine hardware.
10. Monitor and debug the program, if necessary, to ensure the proper functioning of the coffee machine.

Please note that the provided code is an example and may require adjustments to fit your specific hardware and requirements. Always refer to the manufacturer's documentation and guidelines for proper implementation and safety precautions.