

Write a self-contained 61131-3 structured text program, not a function block, to control a traffic light system. The system shall react to pedestrian push buttons and to the presence of emergency vehicles to give them priority.



iecst

Here's a 61131-3 structured text program that controls a traffic light system. This program has three states: 'NormalOperation', 'Pedestrian' and 'Emergency'. The traffic light will react to pedestrian push buttons and give priority to emergency vehicles:

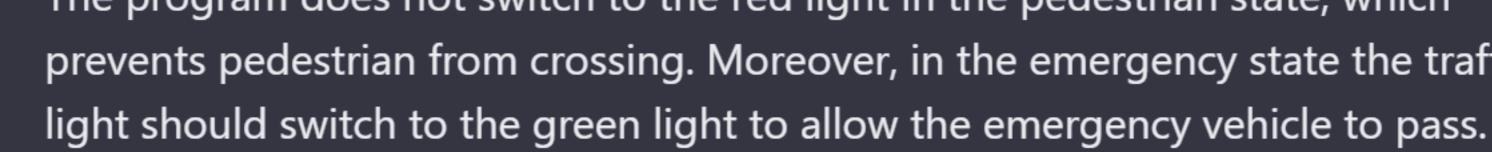
Copy code

```
PROGRAM TrafficLightControl
VAR
   State: (NormalOperation, Pedestrian, Emergency);
   GreenLight, YellowLight, RedLight: BOOL;
    PedestrianRequest, EmergencyVehicle: BOOL;
   NormalOperationTimer, PedestrianTimer, EmergencyTimer: TON;
   NormalOperationTime, PedestrianTime, EmergencyTime: TIME := T#0S;
END_VAR
State := NormalOperation;
NormalOperationTime := T#20S;
PedestrianTime := T#10S;
EmergencyTime := T#5S;
METHOD RunTrafficLightControl: BOOL
    PedestrianRequest := ReadPedestrianPushButton(); // Read pedestrian pus
    EmergencyVehicle := DetectEmergencyVehicle(); // Detect emergency vehicle
   CASE State OF
       NormalOperation:
            GreenLight := TRUE;
            YellowLight := FALSE;
            RedLight := FALSE;
            NormalOperationTimer(IN:= TRUE, PT:= NormalOperationTime);
            IF PedestrianRequest THEN
                State := Pedestrian;
                NormalOperationTimer(IN:= FALSE);
            ELSIF EmergencyVehicle THEN
                State := Emergency;
                NormalOperationTimer(IN:= FALSE);
            END_IF;
        Pedestrian:
            GreenLight := FALSE;
            YellowLight := TRUE;
            RedLight := FALSE;
            PedestrianTimer(IN:= TRUE, PT:= PedestrianTime);
            IF PedestrianTimer.Q THEN
                State := NormalOperation;
                PedestrianTimer(IN:= FALSE);
            END_IF;
        Emergency:
            GreenLight := FALSE;
            YellowLight := FALSE;
            RedLight := TRUE;
            EmergencyTimer(IN:= TRUE, PT:= EmergencyTime);
            IF EmergencyTimer.Q THEN
                State := NormalOperation;
                EmergencyTimer(IN:= FALSE);
            END_IF;
    END_CASE;
   SetTrafficLights(Green:= GreenLight, Yellow:= YellowLight, Red:= RedLig
   RETURN TRUE;
```

implement or integrate these functions into your PLC system.

This program assumes that there are external functions to read the pedestrian push

button, detect emergency vehicles, and set the traffic lights. You will need to



The program does not switch to the red light in the pedestrian state, which prevents pedestrian from crossing. Moreover, in the emergency state the traffic

Please revise the program.

with the appropriate changes:

END_METHOD

END_PROGRAM



Copy code iecst

Apologies for the confusion. Here's the revised 61131-3 structured text program

```
PROGRAM TrafficLightControl
VAR
   State: (NormalOperation, Pedestrian, Emergency);
   GreenLight, YellowLight, RedLight: BOOL;
    PedestrianRequest, EmergencyVehicle: BOOL;
   NormalOperationTimer, PedestrianTimer, EmergencyTimer: TON;
   NormalOperationTime, PedestrianTime, EmergencyTime: TIME := T#0S;
END_VAR
State := NormalOperation;
NormalOperationTime := T#20S;
PedestrianTime := T#10S;
EmergencyTime := T#5S;
METHOD RunTrafficLightControl: BOOL
    PedestrianRequest := ReadPedestrianPushButton(); // Read pedestrian pus
    EmergencyVehicle := DetectEmergencyVehicle(); // Detect emergency vehicle
   CASE State OF
        NormalOperation:
            GreenLight := TRUE;
            YellowLight := FALSE;
            RedLight := FALSE;
            NormalOperationTimer(IN:= TRUE, PT:= NormalOperationTime);
            IF PedestrianRequest THEN
                State := Pedestrian;
                NormalOperationTimer(IN:= FALSE);
            ELSIF EmergencyVehicle THEN
                State := Emergency;
                NormalOperationTimer(IN:= FALSE);
            END_IF;
        Pedestrian:
            GreenLight := FALSE;
            YellowLight := FALSE;
            RedLight := TRUE;
            PedestrianTimer(IN:= TRUE, PT:= PedestrianTime);
            IF PedestrianTimer.Q THEN
                State := NormalOperation;
                PedestrianTimer(IN:= FALSE);
            END_IF;
        Emergency:
            GreenLight := TRUE;
            YellowLight := FALSE;
            RedLight := FALSE;
            EmergencyTimer(IN:= TRUE, PT:= EmergencyTime);
            IF EmergencyTimer.Q THEN
                State := NormalOperation;
                EmergencyTimer(IN:= FALSE);
            END_IF;
   END_CASE;
   SetTrafficLights(Green:= GreenLight, Yellow:= YellowLight, Red:= RedLig
   RETURN TRUE;
```

This updated program switches the traffic light to red in the 'Pedestrian' state, allowing pedestrians to cross safely. In the 'Emergency' state, the traffic light switches to green to let the emergency vehicles pass quickly.





END_METHOD

END_PROGRAM

