Write a IEC 61131-3 ST program to go through all states of an EtherCAT slave device using the ESM function block. Use the typical statenames instead of numbers and only perform state changes if they are allowed. Add a 5 second timer delay before each state transititon.

The IEC 61131-3 Structured Text (ST) program below uses the EtherCAT State Machine (ESM) function block to go through all states of an EtherCAT slave device. The program uses typical state names instead of numbers and checks if state transitions are allowed. It also includes a 5-second timer delay before each state transition.

iecst Copy code

```
PROGRAM EtherCAT_State_Transition;
VAR
    ESM_Instance: ESM; // EtherCAT State Machine function block
    StateTimer: TON; // Timer for 5-second delay
    TransitionRequest: BOOL := FALSE;
    CurrentState, NextState: STRING;
END_VAR

METHOD AllowedTransition: BOOL;
VAR_INPUT
    CurrentState, NextState: STRING;
END_VAR
CASE CurrentState OF
    'INIT': AllowedTransition := (NextState = 'PREOP');
    'PREOP': AllowedTransition := (NextState = 'SAFEOP') OR (NextState = 'I
    'SAFEOP': AllowedTransition := (NextState = 'OP') OR (NextState = 'PREO
    'OP': AllowedTransition := (NextState = 'SAFEOP');
END_CASE;
END_METHOD

METHOD SetNextState;
CASE CurrentState OF
    'INIT': NextState := 'PREOP';
    'PREOP': NextState := 'SAFEOP';
    'SAFEOP': NextState := 'OP';
    'OP': NextState := 'SAFEOP';
END_CASE;
END_METHOD

StateTimer(IN:=TransitionRequest, PT:=T#5S);
IF NOT StateTimer.Q THEN
    TransitionRequest := FALSE;
    CurrentState := ESM_Instance.CurrentState;
    SetNextState();
    IF AllowedTransition(CurrentState, NextState) THEN
        ESM_Instance.ExecuteTransition(NextState);
        TransitionRequest := TRUE;
    END_IF;
END_IF;
END_PROGRAM
```

This program defines an EtherCAT State Machine (ESM) function block instance, a timer (StateTimer) for a 5-second delay, and variables for the current and next state. The `AllowedTransition` method checks if a state transition is allowed, while the `SetNextState` method sets the next state based on the current state. The main program checks if the timer is not active, and if so, it sets the next state and executes the transition if allowed. It then activates the timer for the next state transition.