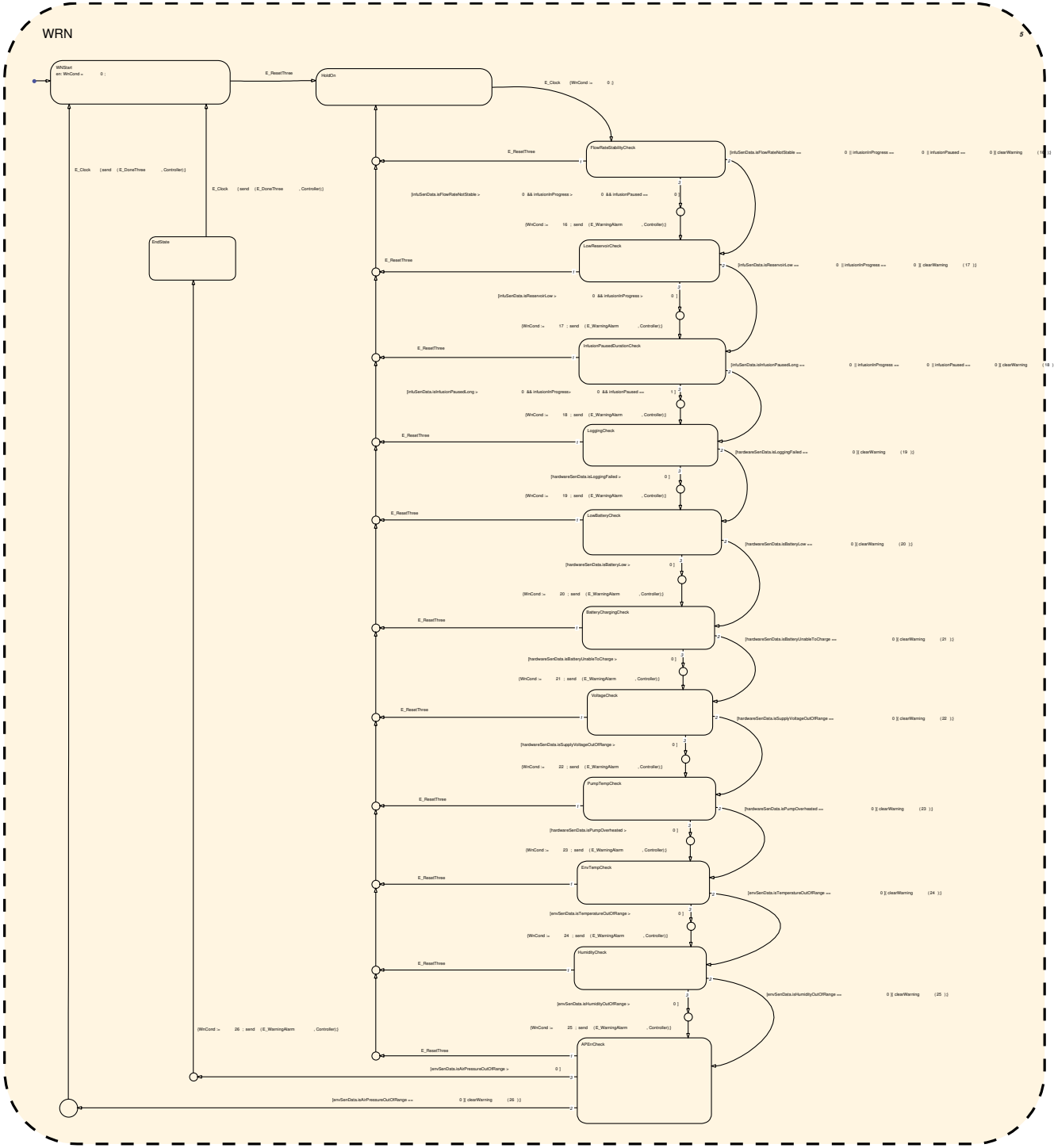
WRN


WRNStart
en: WinCond := 0 ;
E_ResetThree
HoldOn
E_Clock {WinCond := 0 ;}
FlowRateStabilityCheck
[infuSenData.isFlowRateNotStable == 0 || infusionInProgress == 0 || infusionPaused == 0 || clearWarning {16 ;}]
[infuSenData.isFlowRateNotStable > 0 && infusionInProgress > 0 && infusionPaused == 0]
{WinCond := 16 ; send {E_WarningAlarm , Controller;};
LowReservoirCheck
[infuSenData.isReservoirLow == 0 || infusionInProgress == 0 || clearWarning {17 ;}]
[infuSenData.isReservoirLow > 0 && infusionInProgress > 0]
{WinCond := 17 ; send {E_WarningAlarm , Controller;};
InfusionPausedDurationCheck
[infuSenData.isInfusionPausedLong == 0 || infusionInProgress == 0 || infusionPaused == 0 || clearWarning {18 ;}]
[infuSenData.isInfusionPausedLong > 0 && infusionInProgress> 0 && infusionPaused == 1]
{WinCond := 18 ; send {E_WarningAlarm , Controller;};
LoggingCheck
[hardwareSenData.isLoggingFailed == 0 || clearWarning {19 ;}]
[hardwareSenData.isLoggingFailed > 0]
{WinCond := 19 ; send {E_WarningAlarm , Controller;};
LowBatteryCheck
[hardwareSenData.isBatteryLow == 0 || clearWarning {20 ;}]
[hardwareSenData.isBatteryLow > 0]
{WinCond := 20 ; send {E_WarningAlarm , Controller;};
BatteryChargingCheck
[hardwareSenData.isBatteryUnableToCharge == 0 || clearWarning {21 ;}]
[hardwareSenData.isBatteryUnableToCharge > 0]
{WinCond := 21 ; send {E_WarningAlarm , Controller;};
VoltageCheck
[hardwareSenData.isSupplyVoltageOutOfRange == 0 || clearWarning {22 ;}]
[hardwareSenData.isSupplyVoltageOutOfRange > 0]
{WinCond := 22 ; send {E_WarningAlarm , Controller;};
PumpTempCheck
[hardwareSenData.isPumpOverheated == 0 || clearWarning {23 ;}]
[hardwareSenData.isPumpOverheated > 0]
{WinCond := 23 ; send {E_WarningAlarm , Controller;};
EnvTempCheck
[envSenData.isTemperatureOutOfRange == 0 || clearWarning {24 ;}]
[envSenData.isTemperatureOutOfRange > 0]
{WinCond := 24 ; send {E_WarningAlarm , Controller;};
HumidityCheck
[envSenData.isHumidityOutOfRange == 0 || clearWarning {25 ;}]
[envSenData.isHumidityOutOfRange > 0]
{WinCond := 25 ; send {E_WarningAlarm , Controller;}
APErrCheck
[envSenData.isAirPressureOutOfRange > 0]
[envSenData.isAirPressureOutOfRange == 0 || clearWarning {26 ;}]
{WinCond := 26 ; send {E_WarningAlarm , Controller;}
EndState
E_Clock {send {E_DoneThree , Controller;}
E_Clock {send {E_DoneThree , Controller;}