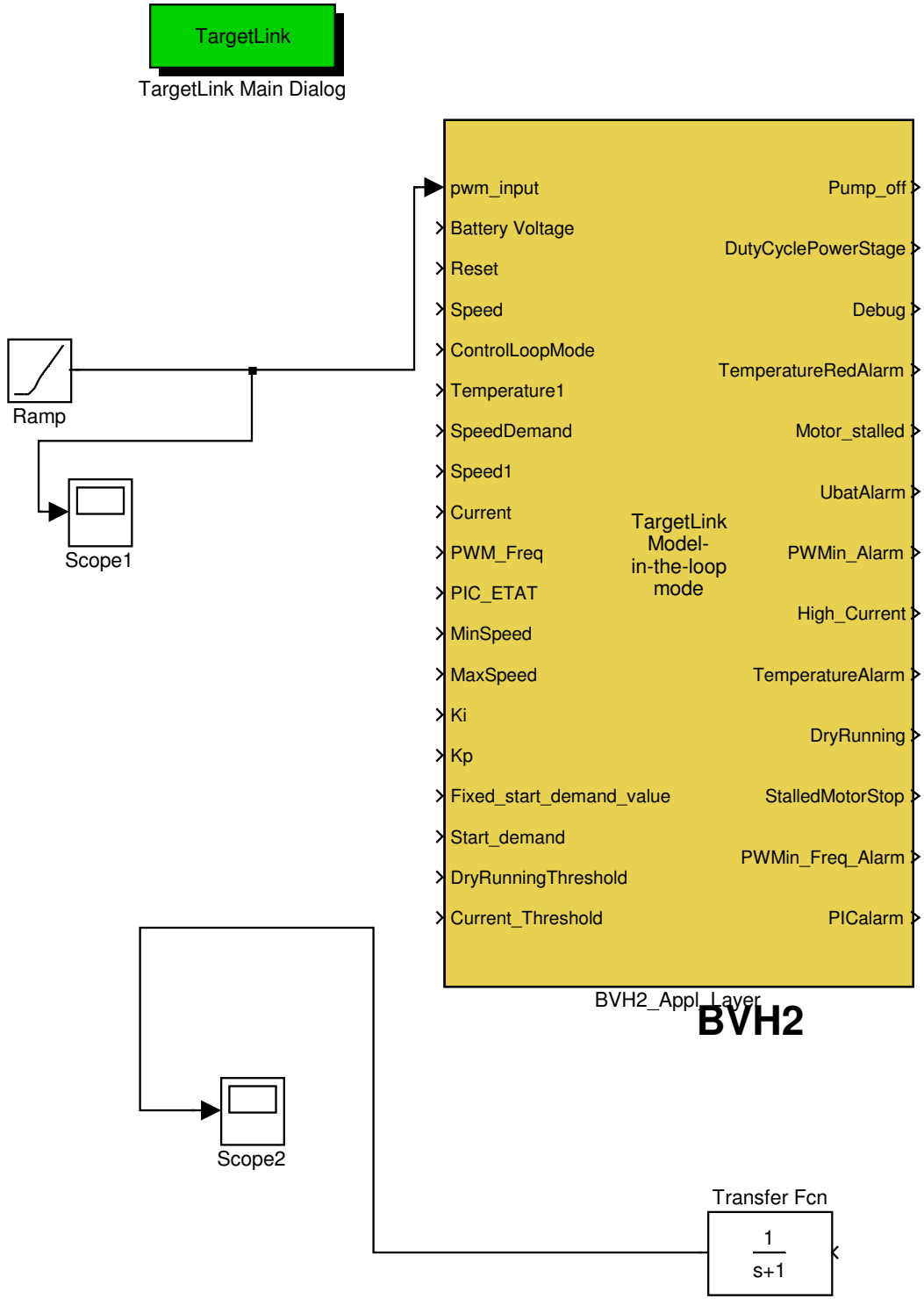
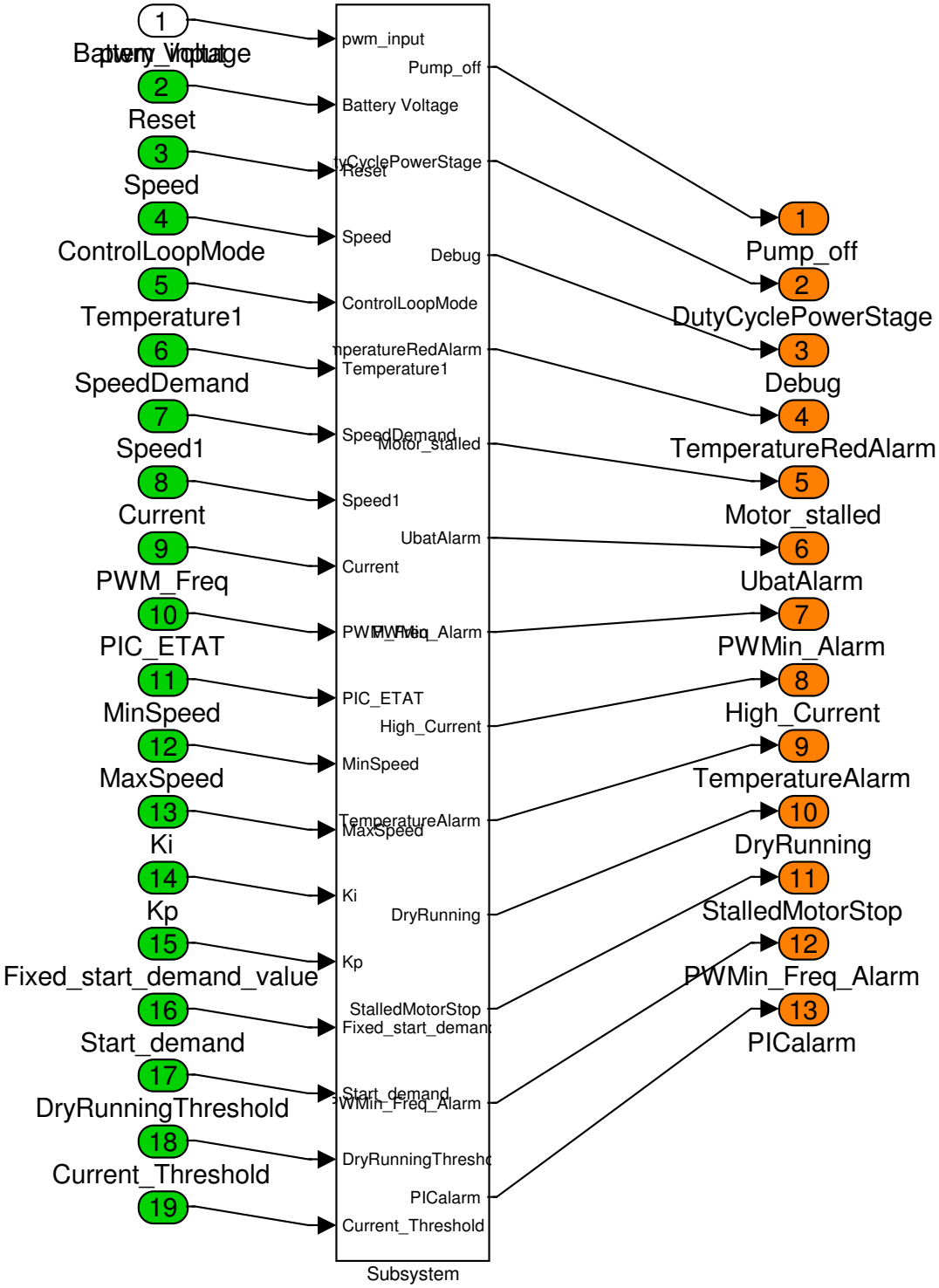
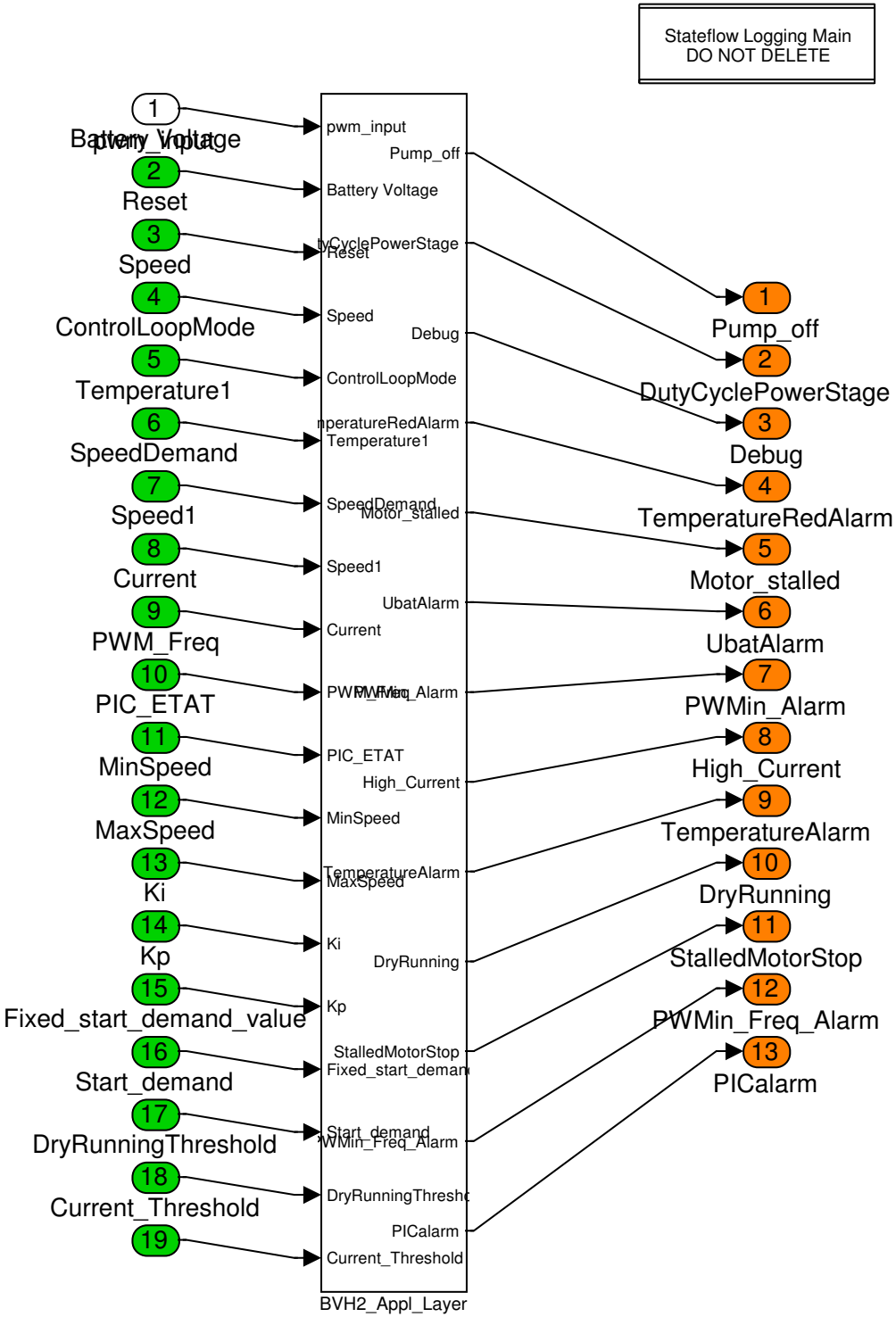
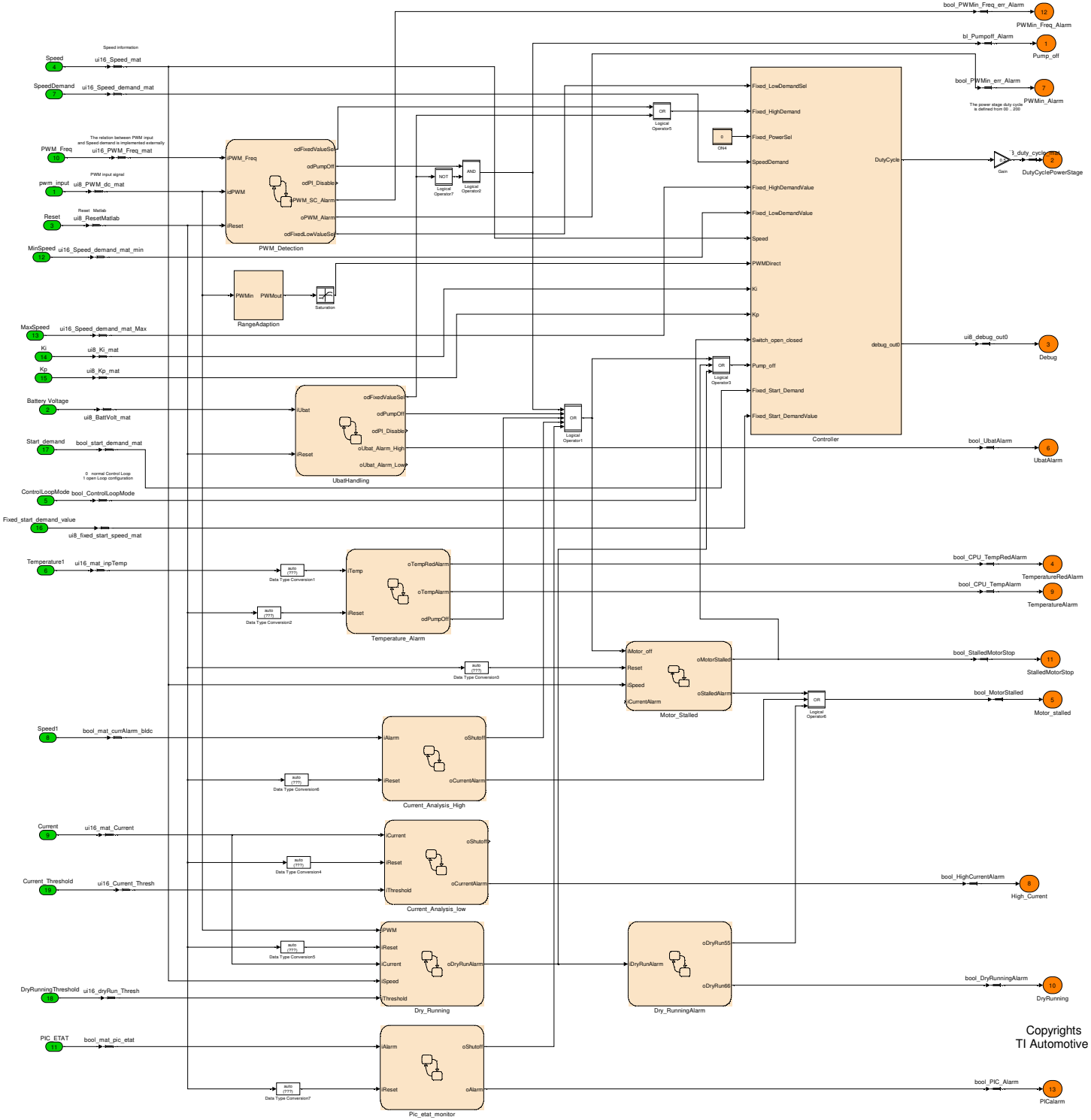
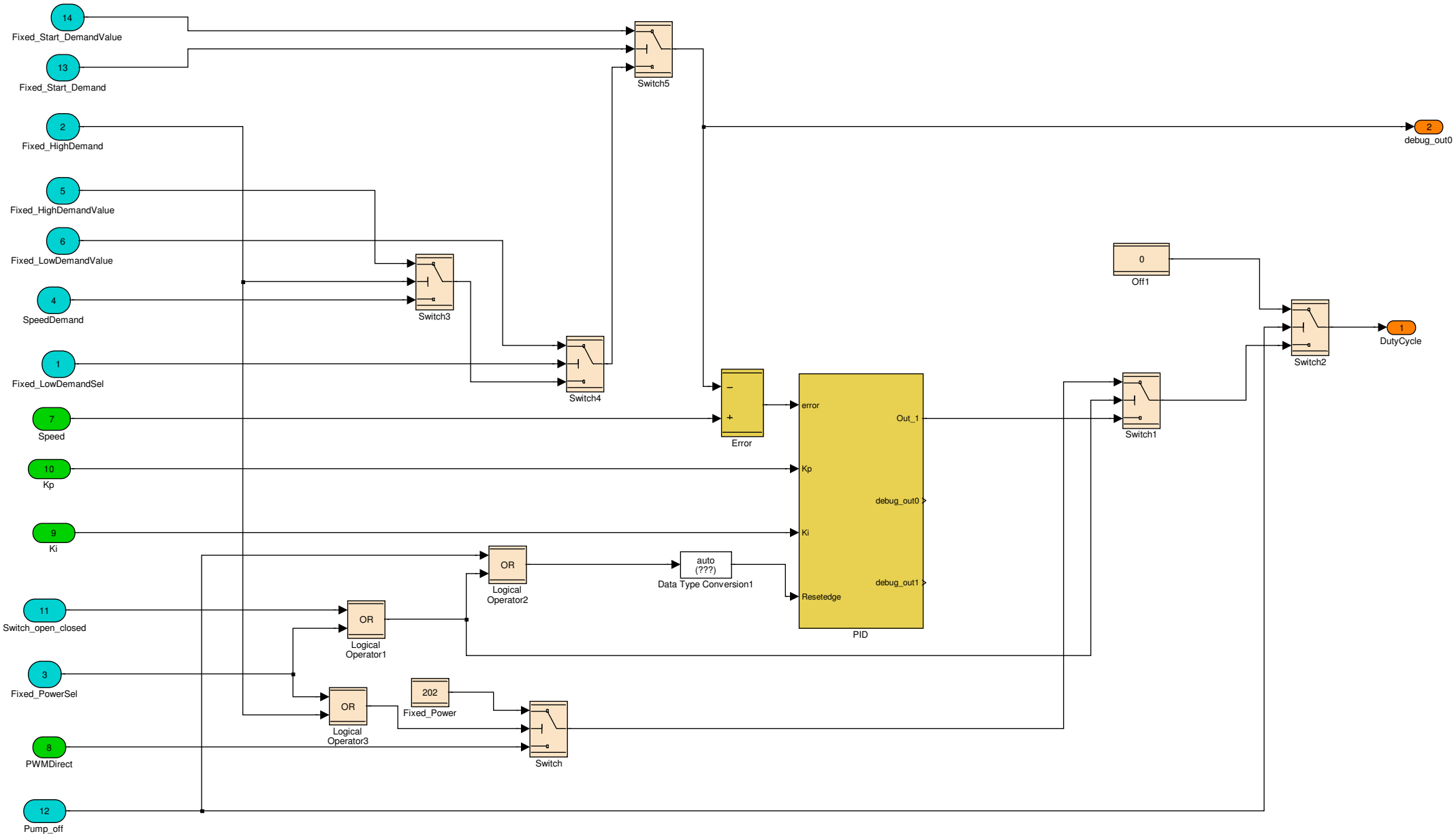
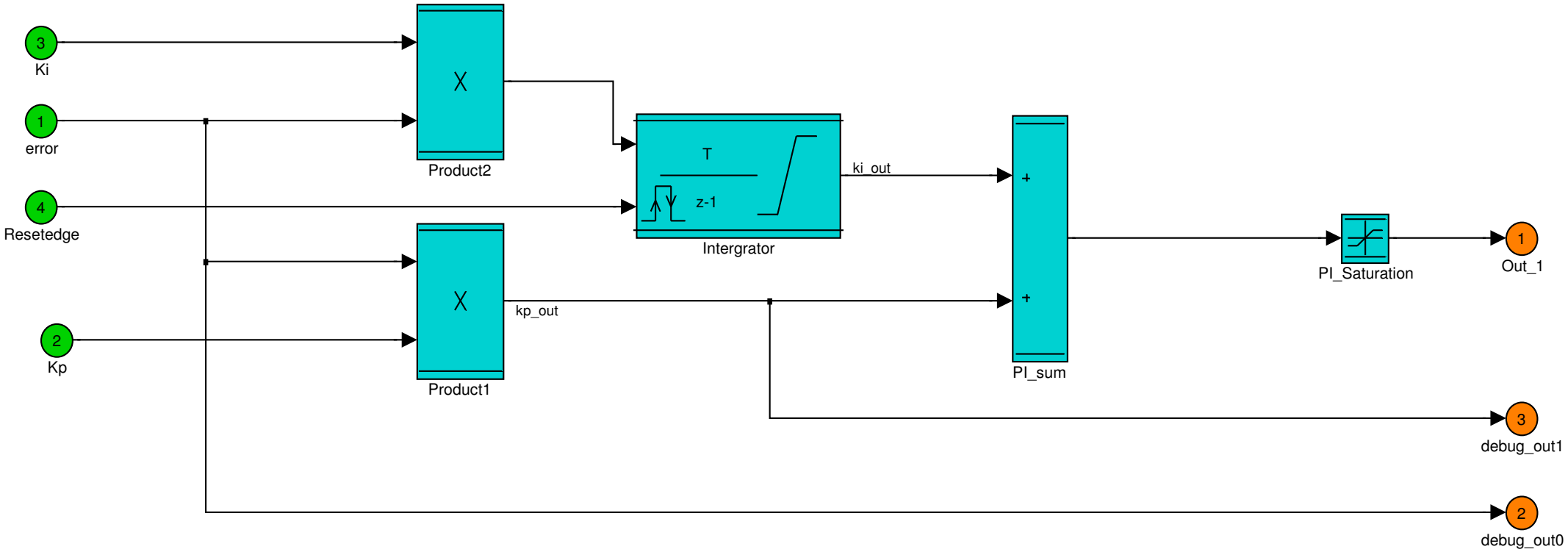

TargetLink
TargetLink Main Dialog
Ramp
Scope1
pwm_input
Battery Voltage
Reset
Speed
ControlLoopMode
Temperature1
SpeedDemand
Speed1
Current
PWM_Freq
PIC_ETAT
MinSpeed
MaxSpeed
Ki
Kp
Fixed_start_demand_value
Start_demand
DryRunningThreshold
Current_Threshold
TargetLink Model-in-the-loop mode
Pump_off
DutyCyclePowerStage
Debug
TemperatureRedAlarm
Motor_stalled
UbatAlarm
PWMin_Alarm
High_Current
TemperatureAlarm
DryRunning
StalledMotorStop
PWMin_Freq_Alarm
PICalarm
BVH2_Appl_Layer
BVH2
Scope2
Transfer Fcn
1
s+1

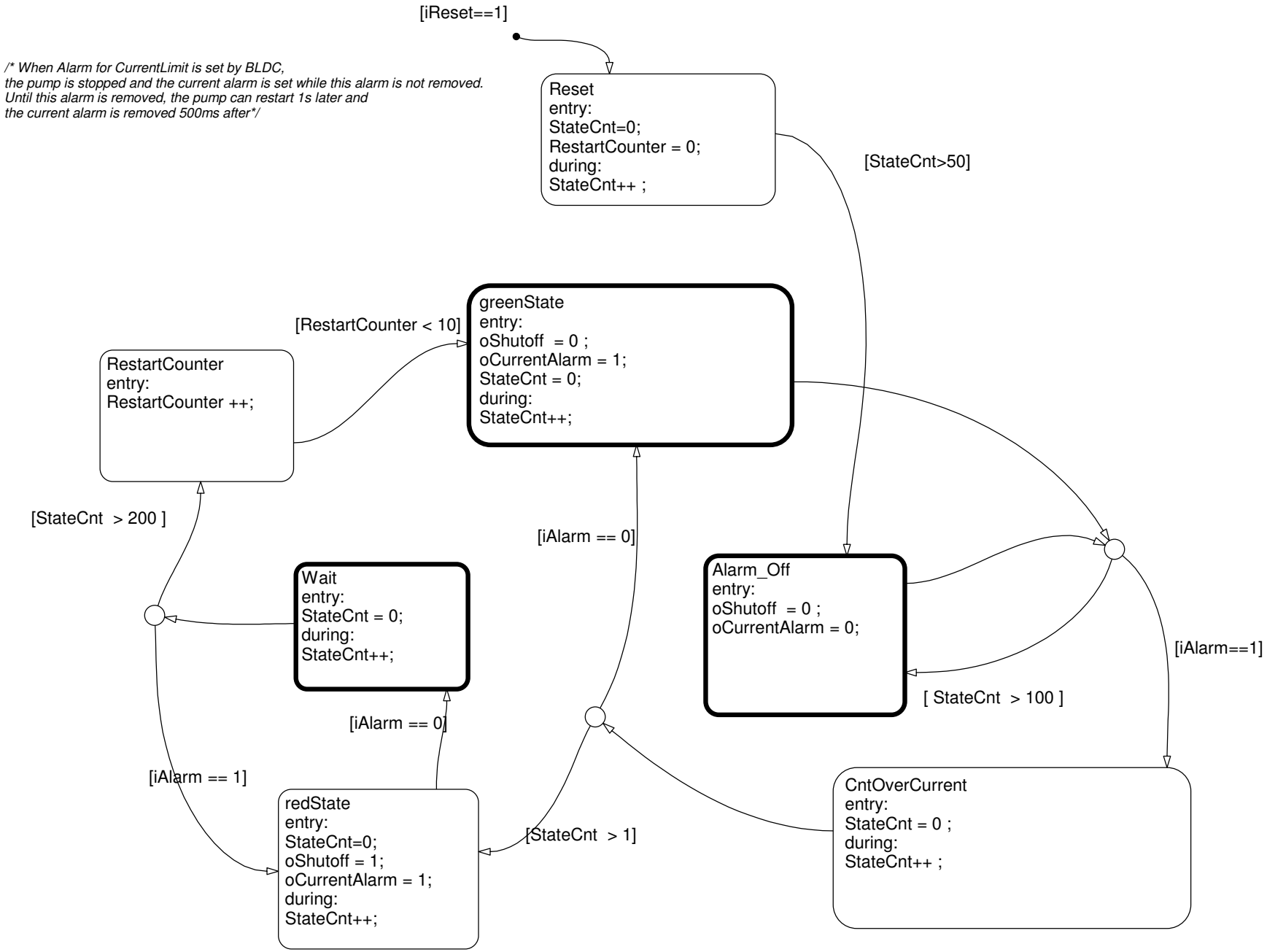

1
Battery Voltage
2
Reset
3
Speed
4
ControlLoopMode
5
Temperature1
6
SpeedDemand
7
Speed1
8
Current
9
PWM_Freq
10
PIC_ETAT
11
MinSpeed
12
MaxSpeed
13
Ki
14
Kp
15
Fixed_start_demand_value
16
Start_demand
17
DryRunningThreshold
18
Current_Threshold
19
pwm_input
Battery Voltage
Reset
Speed
ControlLoopMode
Temperature1
SpeedDemand
Speed1
Current
PIC_ETAT
MinSpeed
MaxSpeed
Ki
Kp
Start_demand
DryRunningThreshold
Current_Threshold
Pump_off
DutyCyclePowerStage
Debug
Motor_stalled
UbatAlarm
High_Current
DryRunning
StalledMotorStop
PICalarm
Subsystem
1
Pump_off
2
DutyCyclePowerStage
3
Debug
4
TemperatureRedAlarm
5
Motor_stalled
6
UbatAlarm
7
PWMin_Alarm
8
High_Current
9
TemperatureAlarm
10
DryRunning
11
StalledMotorStop
12
PWMin_Freq_Alarm
13
PICalarm

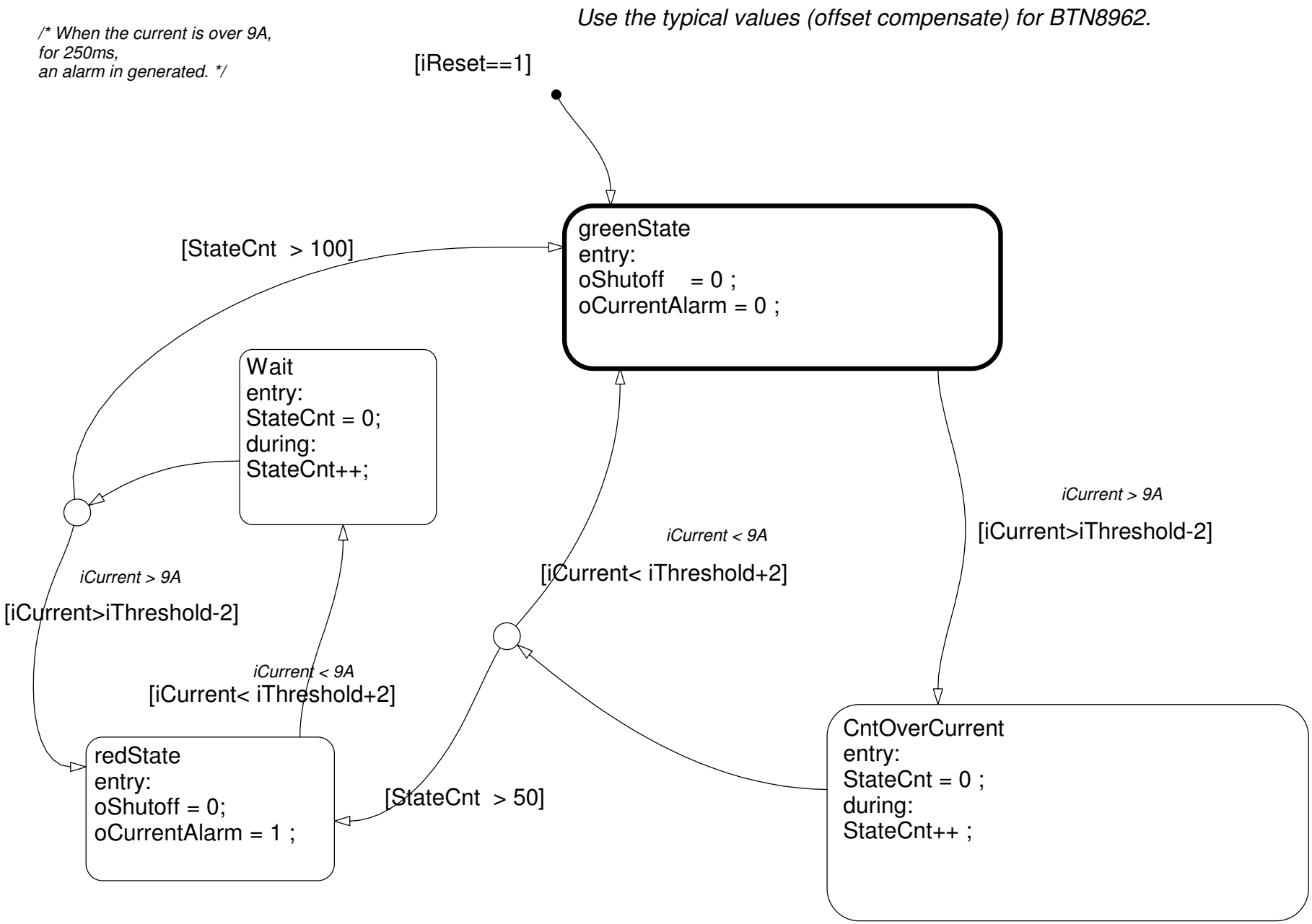

Stateflow Logging Main
DO NOT DELETE
1
Battery Voltage
2
Reset
3
Speed
4
ControlLoopMode
5
Temperature1
6
SpeedDemand
7
Speed1
8
Current
9
PWM_Freq
10
PIC_ETAT
11
MinSpeed
12
MaxSpeed
13
Ki
14
Kp
15
Fixed_start_demand_value
16
Start_demand
17
DryRunningThreshold
18
Current_Threshold
19
pwm_input
Battery Voltage
Reset
Speed
ControlLoopMode
Temperature1
SpeedDemand
Speed1
Current
PIC_ETAT
MinSpeed
MaxSpeed
Ki
Kp
Start_demand
Current_Threshold
Pump_off
DutyCyclePowerStage
Debug
TemperatureRedAlarm
Motor_stalled
UbatAlarm
High_Current
TemperatureAlarm
DryRunning
StalledMotorStop
PICalarm
BVH2_Appl_Layer
1
Pump_off
2
DutyCyclePowerStage
3
Debug
4
TemperatureRedAlarm
5
Motor_stalled
6
UbatAlarm
7
PWMin_Alarm
8
High_Current
9
TemperatureAlarm
10
DryRunning
11
StalledMotorStop
12
PWMin_Freq_Alarm
13
PICalarm

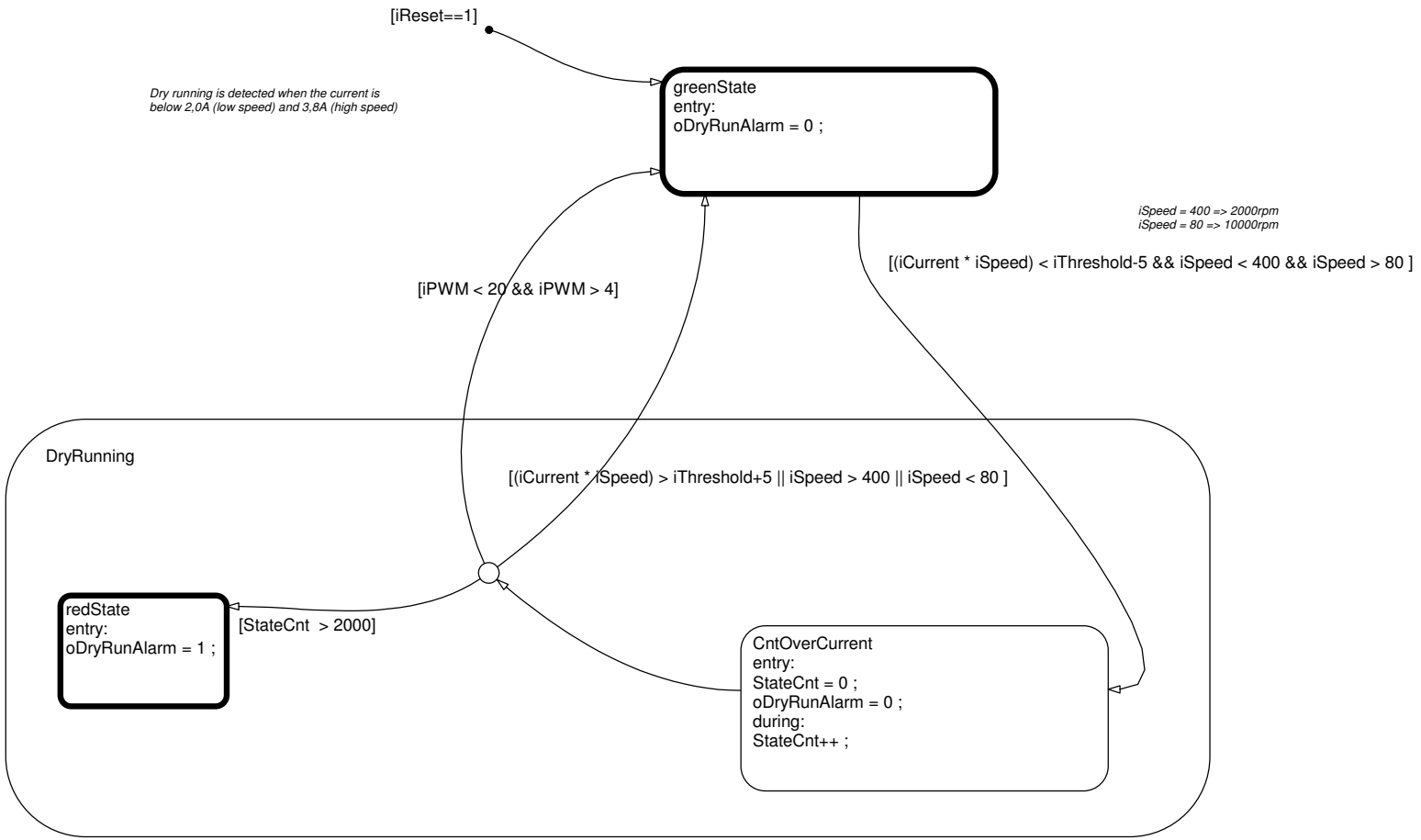

Speed information
Speed
4
ui16_Speed_mat
SpeedDemand
7
ui16_Speed_demand_mat
The relation between PWM input and Speed demand is implemented externally
PWM_Freq
10
ui16_PWM_Freq_mat
PWM input signal
pwm_input
1
ui8_PWM_dc_mat
Reset_Matlab
Reset
3
ui8_ResetMatlab
MinSpeed
12
ui16_Speed_demand_mat_min
MaxSpeed
13
ui16_Speed_demand_mat_Max
Ki
14
ui8_Ki_mat
Kp
15
ui8_Kp_mat
Battery Voltage
2
ui8_BattVolt_mat
Start_demand
17
bool_start_demand_mat
0 : normal Control Loop
1 open Loop configuration
ControlLoopMode
5
bool_ControlLoopMode
Fixed_start_demand_value
16
ui8_fixed_start_speed_mat
Temperature1
6
ui16_mat_inpTemp
Speed1
8
bool_mat_currAlarm_bldc
Current
18
ui16_mat_Current
Current_Threshold
19
ui16_Current_Thresh
DryRunningThreshold
20
ui16_dryRun_Thresh
PIC_ETAT
11
bool_mat_pic_etat
PWM_Detection
iPWM_Freq
idPWM
iReset
odFixedValueSel
odPumpOff
odPI_Disable
oPWM_SC_Alarm
oPWM_Alarm
odFixedLowValueSel
RangeAdaption
PWMin
PWMout
Saturation
UbatHandling
iUbat
iReset
odFixedValueSel
odPumpOff
odPI_Disable
oUbat_Alarm_High
oUbat_Alarm_Low
Temperature_Alarm
iTemp
iReset
oTempRedAlarm
oTempAlarm
odPumpOff
Current_Analysis_High
iAlarm
iReset
oShutoff
oCurrentAlarm
Current_Analysis_low
iCurrent
iReset
iThreshold
oShutoff
oCurrentAlarm
Dry_Running
iPWM
iReset
iCurrent
iSpeed
iThreshold
oDryRunAlarm
Pic_etat_monitor
iAlarm
iReset
oShutoff
oAlarm
Motor_Stalled
iMotor_off
iReset
iSpeed
iCurrentAlarm
oMotorStalled
oStalledAlarm
Dry_RunningAlarm
iDryRunAlarm
oDryRun55
oDryRun66
Controller
Fixed_LowDemandSel
Fixed_HighDemand
Fixed_PowerSel
SpeedDemand
Fixed_HighDemandValue
Fixed_LowDemandValue
Speed
PWMDirect
Ki
Kp
Switch_open_closed
Pump_off
Fixed_Start_Demand
Fixed_Start_DemandValue
DutyCycle
debug_out0
Logical Operator
Logical Operator1
Logical Operator2
Logical Operator3
Logical Operator5
Logical Operator6
Logical Operator7
NOT
AND
OR
ON4
Gain
Data Type Conversion1
Data Type Conversion2
Data Type Conversion3
Data Type Conversion4
Data Type Conversion5
Data Type Conversion6
Data Type Conversion7
auto (???)
bool_PWMin_Freq_err_Alarm
12
PWMin_Freq_Alarm
bl_Pumpoff_Alarm
1
Pump_off
bool_PWMin_err_Alarm
7
PWMin_Alarm
The power stage duty cycle is defined from 00 .. 200
3_duty_cycle_min
2
DutyCyclePowerStage
ui8_debug_out0
3
Debug
bool_UbatAlarm
6
UbatAlarm
bool_CPU_TempRedAlarm
4
TemperatureRedAlarm
bool_CPU_TempAlarm
9
TemperatureAlarm
bool_StalledMotorStop
11
StalledMotorStop
bool_MotorStalled
5
Motor_stalled
bool_HighCurrentAlarm
8
High_Current
bool_DryRunningAlarm
10
DryRunning
bool_PIC_Alarm
13
PICalarm
Copyrights
TI Automotive

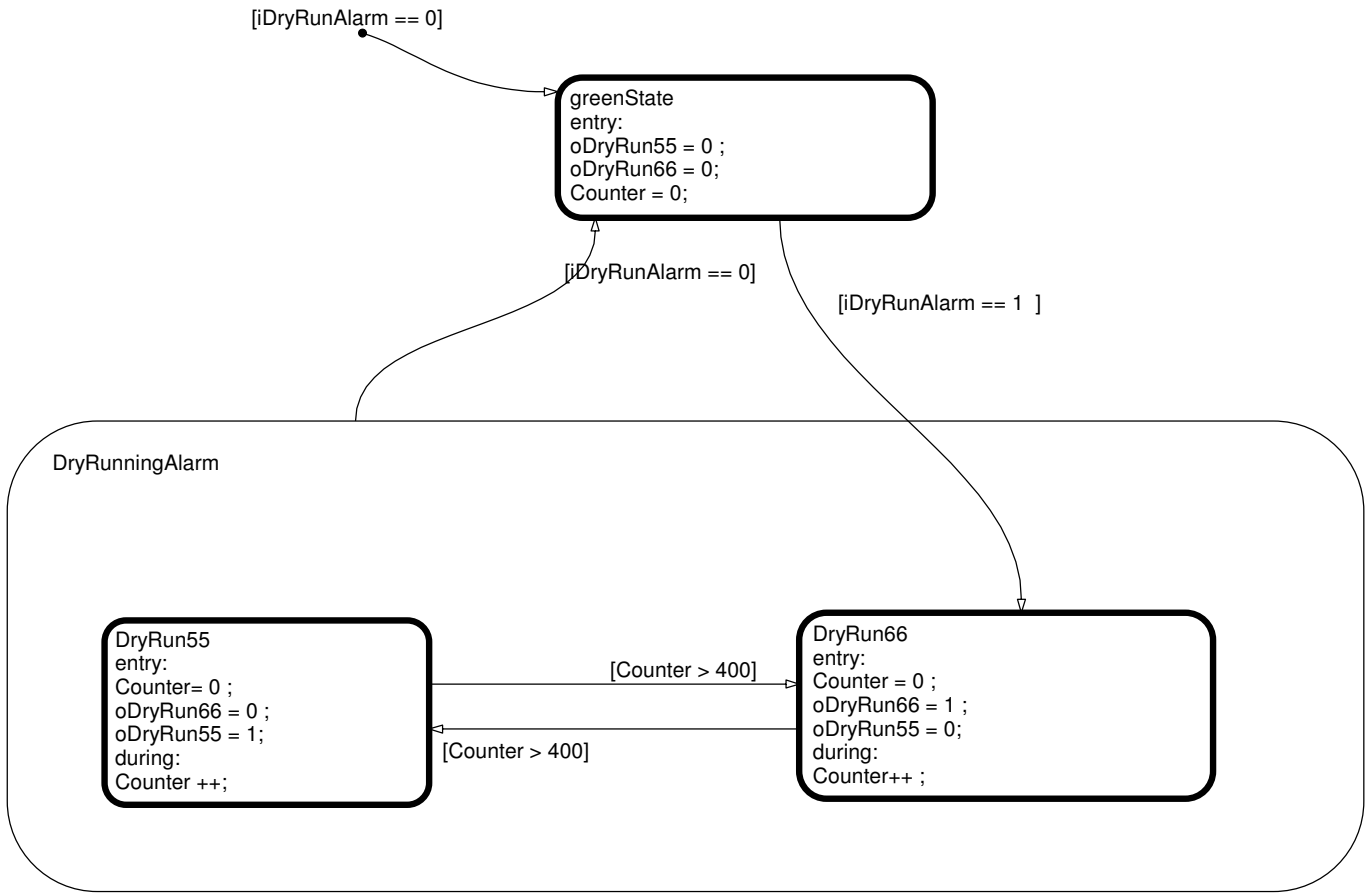

14
Fixed_Start_DemandValue
13
Fixed_Start_Demand
2
Fixed_HighDemand
5
Fixed_HighDemandValue
6
Fixed_LowDemandValue
4
SpeedDemand
1
Fixed_LowDemandSel
7
Speed
10
Kp
9
Ki
11
Switch_open_closed
3
Fixed_PowerSel
8
PWMDirect
12
Pump_off
Switch5
Switch3
Switch4
2
debug_out0
0
Off1
Switch2
1
DutyCycle
Error
error
Out_1
Kp
debug_out0
Ki
debug_out1
Resetedge
PID
Switch1
OR
Logical Operator2
auto (???)
Data Type Conversion1
OR
Logical Operator1
OR
Logical Operator3
202
Fixed_Power
Switch

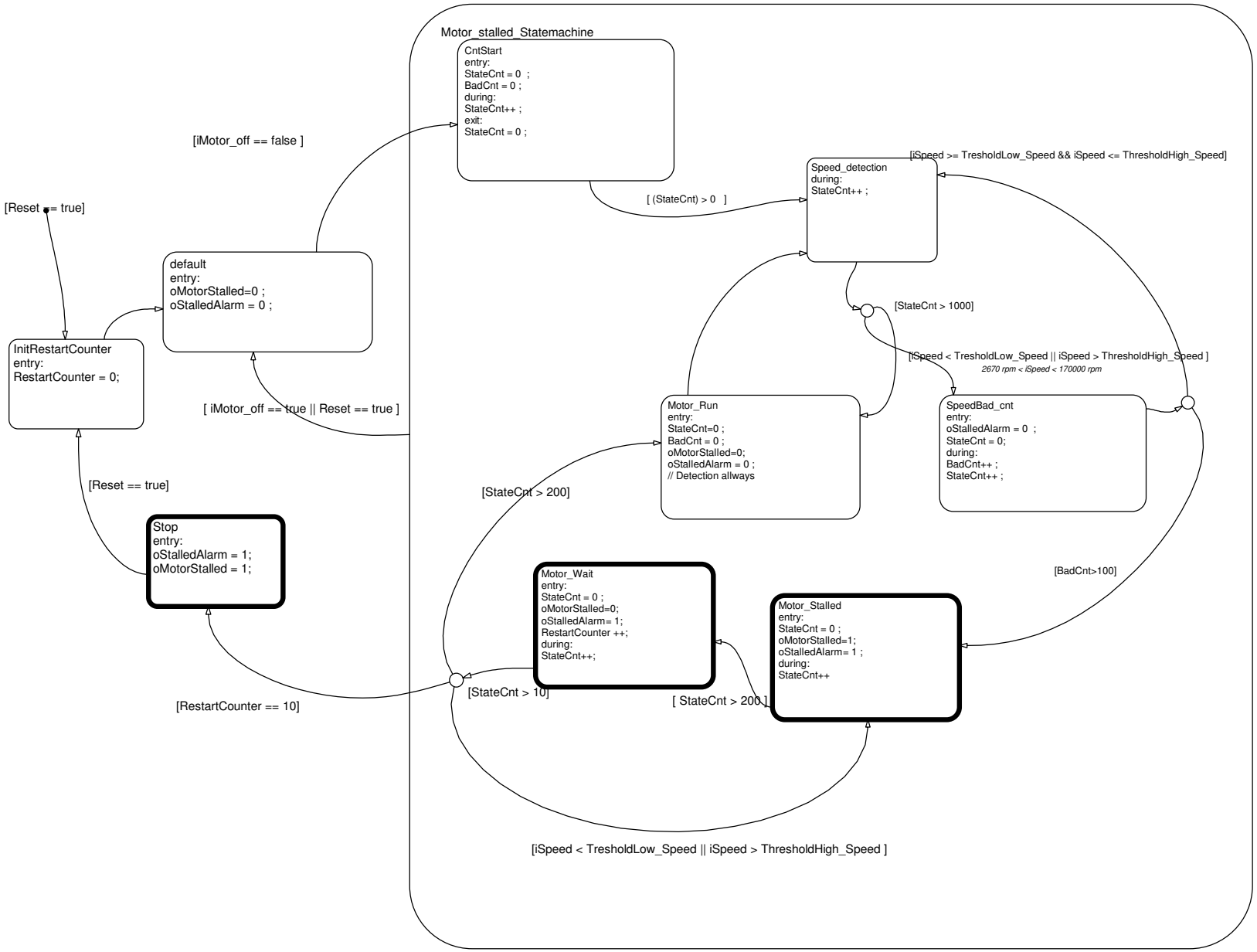

3
Ki
1
error
4
Resetedge
2
Kp
X
Product2
X
Product1
T
z-1
Intergrator
ki_out
kp_out
+
+
PI_sum
PI_Saturation
1
Out_1
3
debug_out1
2
debug_out0


[iReset==1]
/* When Alarm for CurrentLimit is set by BLDC,
the pump is stopped and the current alarm is set while this alarm is not removed.
Until this alarm is removed, the pump can restart 1s later and
the current alarm is removed 500ms after*/
Reset
entry:
StateCnt=0;
RestartCounter = 0;
during:
StateCnt++ ;
[StateCnt>50]
greenState
entry:
oShutoff = 0 ;
oCurrentAlarm = 1;
StateCnt = 0;
during:
StateCnt++;
[RestartCounter < 10]
RestartCounter
entry:
RestartCounter ++;
[StateCnt > 200]
[iAlarm == 0]
Wait
entry:
StateCnt = 0;
during:
StateCnt++;
Alarm_Off
entry:
oShutoff = 0 ;
oCurrentAlarm = 0;
[iAlarm==1]
[StateCnt > 100]
[iAlarm == 0]
[iAlarm == 1]
redState
entry:
StateCnt=0;
oShutoff = 1;
oCurrentAlarm = 1;
during:
StateCnt++;
[StateCnt > 1]
CntOverCurrent
entry:
StateCnt = 0 ;
during:
StateCnt++ ;


/* When the current is over 9A,
for 250ms,
an alarm in generated. */
Use the typical values (offset compensate) for BTN8962.
[iReset==1]
greenState
entry:
oShutoff = 0 ;
oCurrentAlarm = 0 ;
[StateCnt > 100]
Wait
entry:
StateCnt = 0;
during:
StateCnt++;
iCurrent > 9A
[iCurrent>iThreshold-2]
iCurrent < 9A
[iCurrent< iThreshold+2]
iCurrent > 9A
[iCurrent>iThreshold-2]
iCurrent < 9A
[iCurrent< iThreshold+2]
redState
entry:
oShutoff = 0;
oCurrentAlarm = 1 ;
[StateCnt > 50]
CntOverCurrent
entry:
StateCnt = 0 ;
during:
StateCnt++ ;


[iReset==1]
Dry running is detected when the current is
below 2,0A (low speed) and 3,8A (high speed)
greenState
entry:
oDryRunAlarm = 0 ;
iSpeed = 400 => 2000rpm
iSpeed = 80 => 10000rpm
[(iCurrent * iSpeed) < iThreshold-5 && iSpeed < 400 && iSpeed > 80]
[iPWM < 20 && iPWM > 4]
DryRunning
[(iCurrent * iSpeed) > iThreshold+5 || iSpeed > 400 || iSpeed < 80]
redState
entry:
oDryRunAlarm = 1 ;
[StateCnt > 2000]
CntOverCurrent
entry:
StateCnt = 0 ;
oDryRunAlarm = 0 ;
during:
StateCnt++ ;

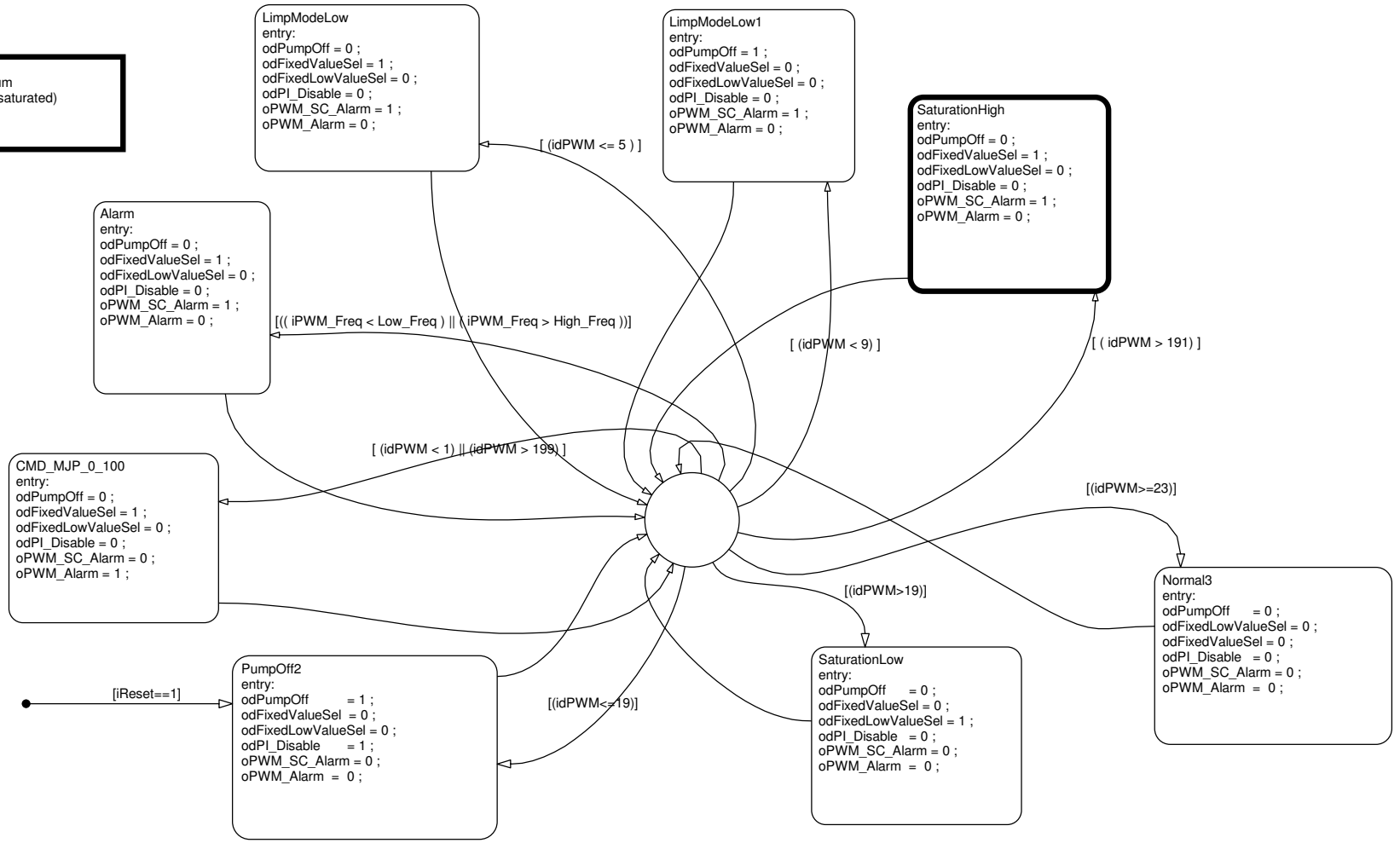

[iDryRunAlarm == 0]
greenState
entry:
oDryRun55 = 0 ;
oDryRun66 = 0;
Counter = 0;
[iDryRunAlarm == 0]
[iDryRunAlarm == 1]
DryRunningAlarm
DryRun55
entry:
Counter= 0 ;
oDryRun66 = 0 ;
oDryRun55 = 1;
during:
Counter ++;
[Counter > 400]
[Counter > 400]
DryRun66
entry:
Counter = 0 ;
oDryRun66 = 1 ;
oDryRun55 = 0;
during:
Counter++ ;

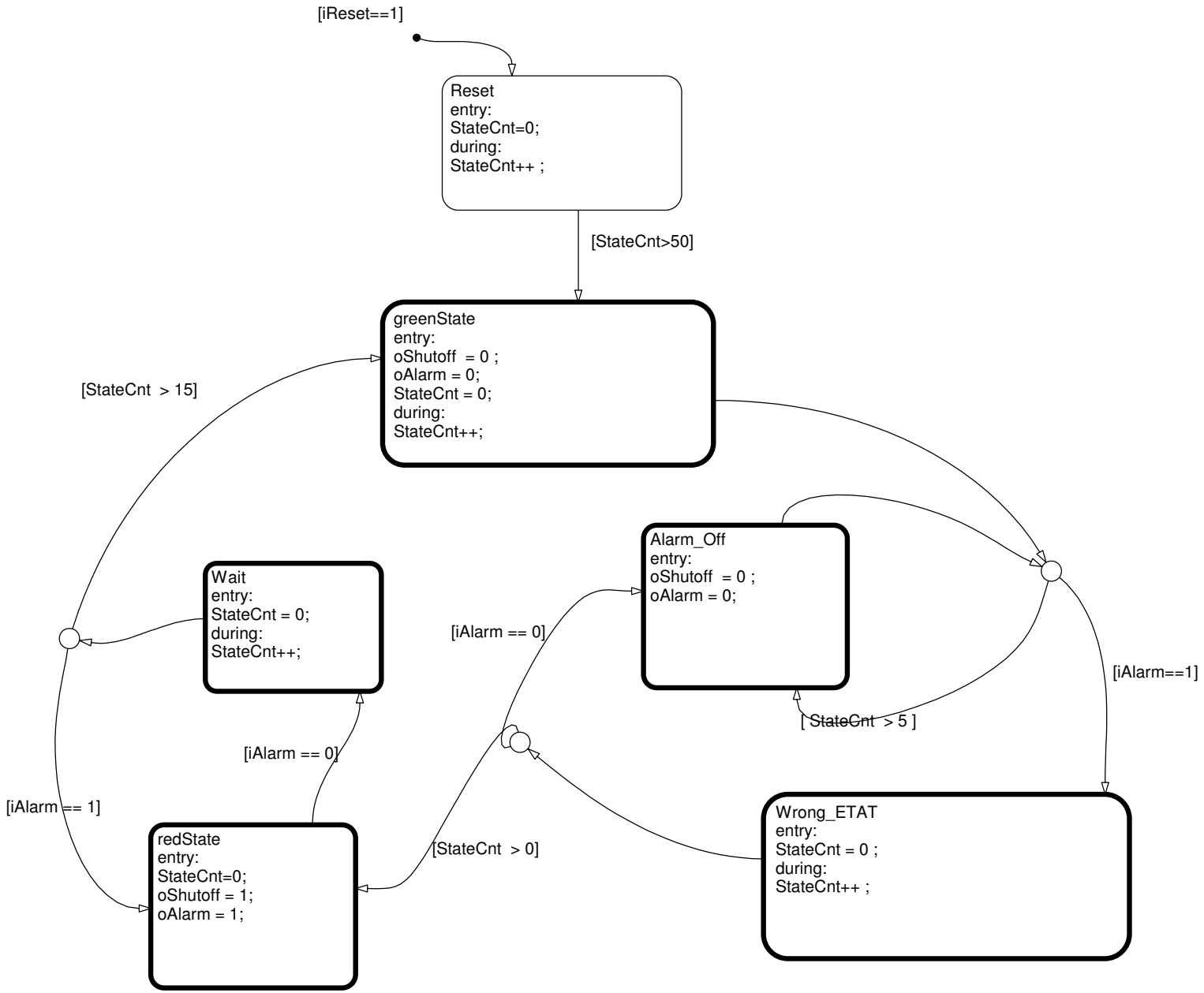

Motor_stalled_Statemachine
CntStart
entry:
StateCnt = 0 ;
BadCnt = 0 ;
during:
StateCnt++ ;
exit:
StateCnt = 0 ;
[iMotor_off == false]
[(StateCnt) > 0]
Speed_detection
during:
StateCnt++ ;
[iSpeed >= TresholdLow_Speed && iSpeed <= ThresholdHigh_Speed]
[Reset == true]
default
entry:
oMotorStalled=0 ;
oStalledAlarm = 0 ;
[StateCnt > 1000]
[iSpeed < TresholdLow_Speed || iSpeed > ThresholdHigh_Speed]
2670 rpm < iSpeed < 170000 rpm
InitRestartCounter
entry:
RestartCounter = 0;
[iMotor_off == true || Reset == true]
Motor_Run
entry:
StateCnt=0 ;
BadCnt = 0 ;
oMotorStalled=0;
oStalledAlarm = 0 ;
// Detection allways
SpeedBad_cnt
entry:
oStalledAlarm = 0 ;
StateCnt = 0;
during:
BadCnt++ ;
StateCnt++ ;
[Reset == true]
[StateCnt > 200]
Stop
entry:
oStalledAlarm = 1;
oMotorStalled = 1;
Motor_Wait
entry:
StateCnt = 0 ;
oMotorStalled=0;
oStalledAlarm= 1;
RestartCounter ++;
during:
StateCnt++;
[BadCnt>100]
Motor_Stalled
entry:
StateCnt = 0 ;
oMotorStalled=1;
oStalledAlarm= 1 ;
during:
StateCnt++
[StateCnt > 10]
[StateCnt > 200]
[RestartCounter == 10]
[iSpeed < TresholdLow_Speed || iSpeed > ThresholdHigh_Speed]

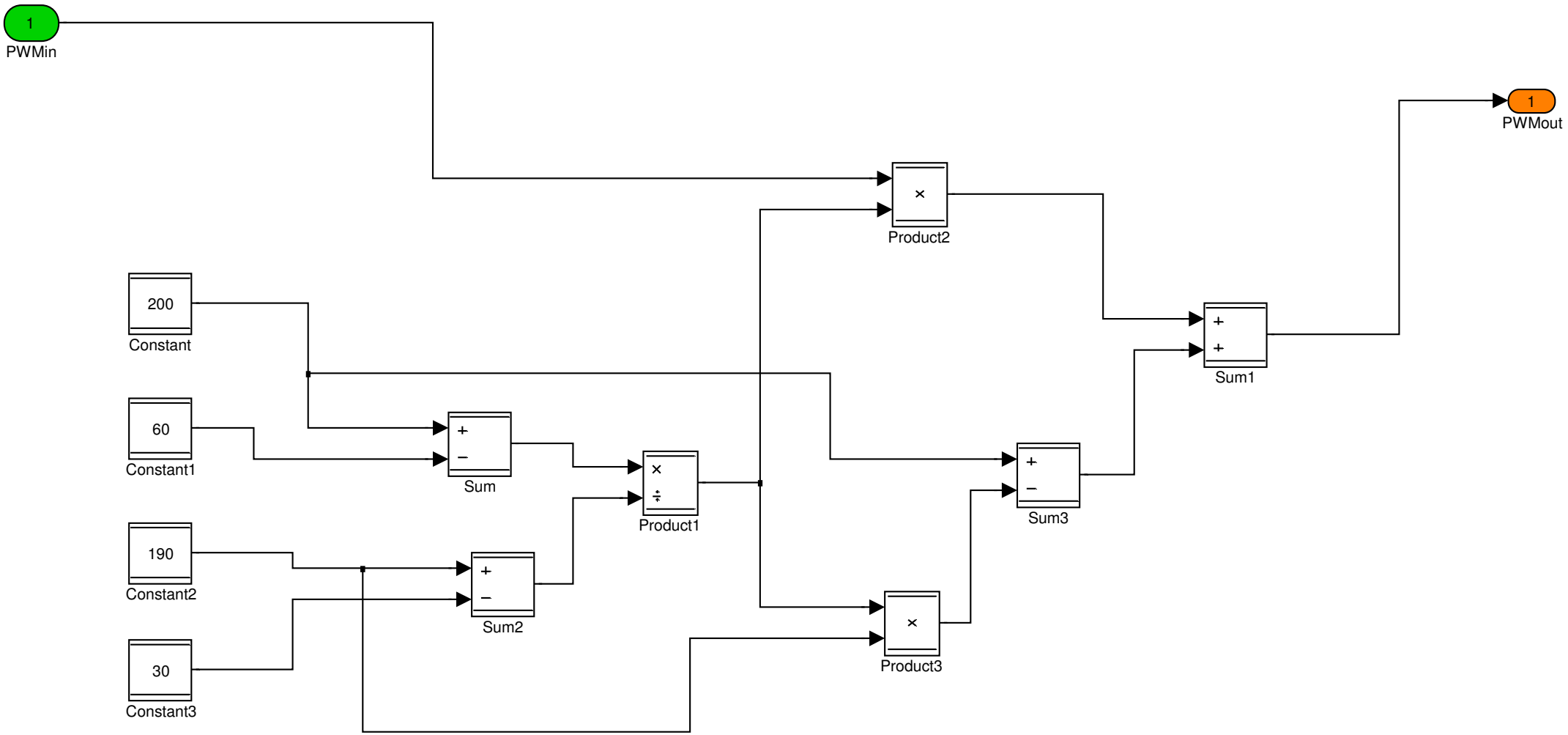
PWMinput_handling

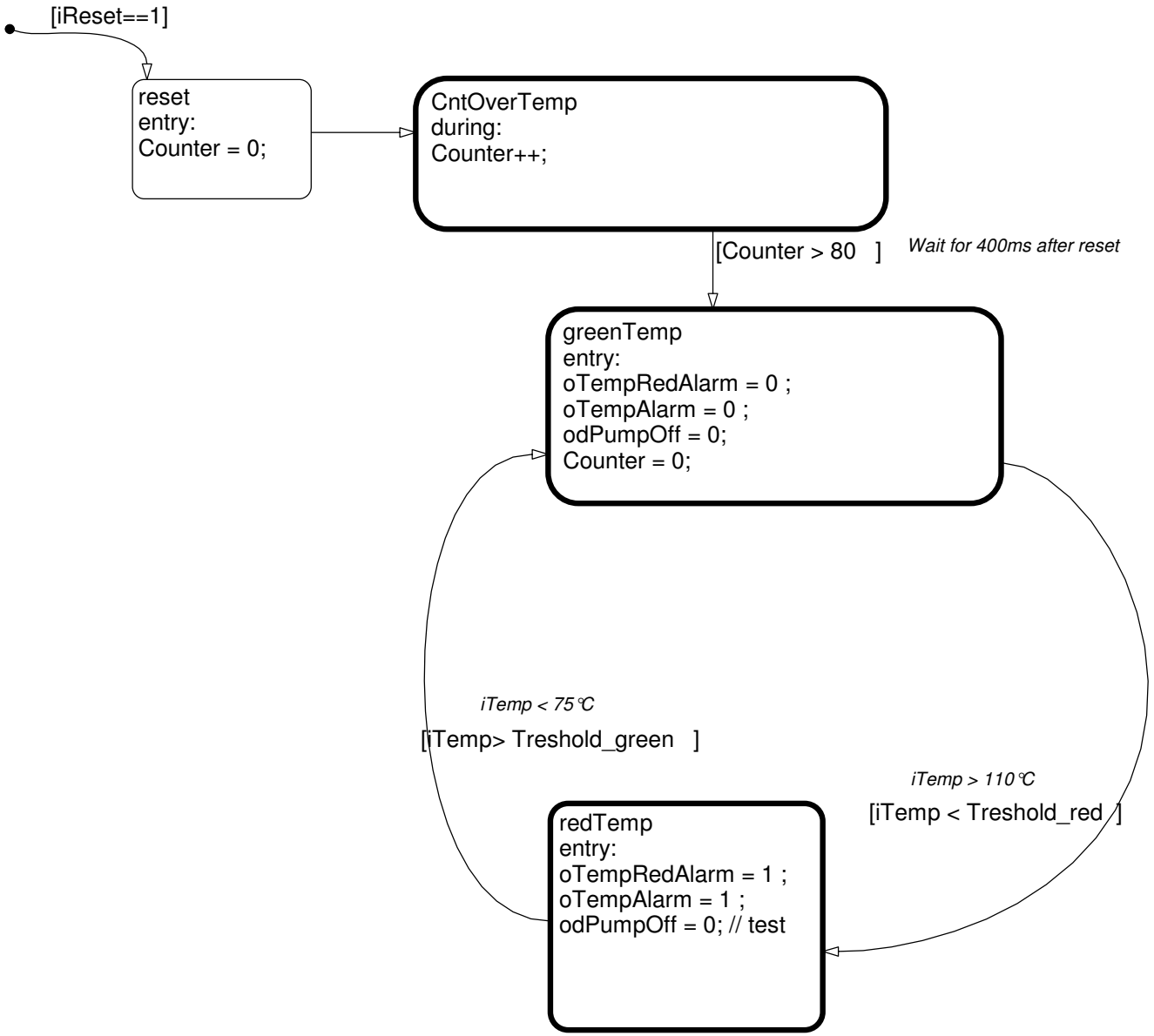
PWMinput_handling

Notice that between 90 -94 %
Input PWM duty cylcle the maximum
output PWM duty cycle is limited (saturated)
outside this statmachine



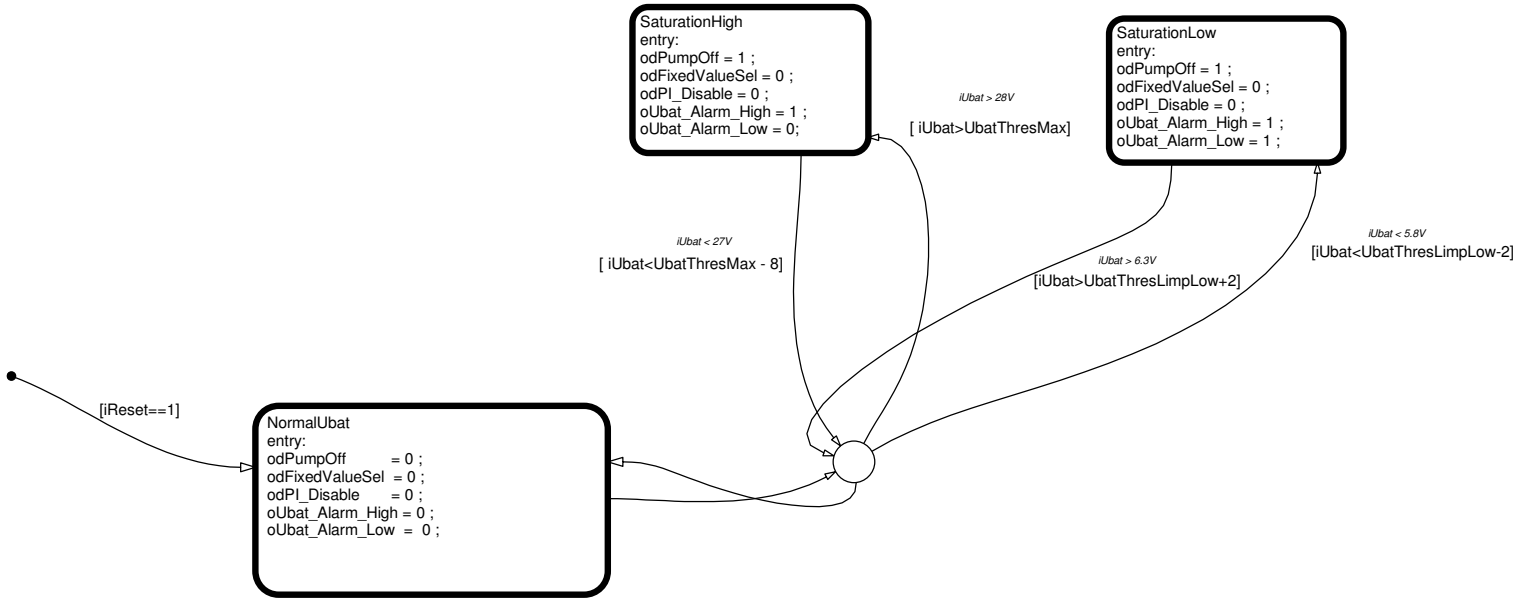

[iReset==1]
Reset
entry:
StateCnt=0;
during:
StateCnt++ ;
[StateCnt>50]
greenState
entry:
oShutoff = 0 ;
oAlarm = 0;
StateCnt = 0;
during:
StateCnt++;
[StateCnt > 15]
Alarm_Off
entry:
oShutoff = 0 ;
oAlarm = 0;
Wait
entry:
StateCnt = 0;
during:
StateCnt++;
[iAlarm == 0]
[iAlarm==1]
[StateCnt > 5]
[iAlarm == 0]
[iAlarm == 1]
redState
entry:
StateCnt=0;
oShutoff = 1;
oAlarm = 1;
[StateCnt > 0]
Wrong_ETAT
entry:
StateCnt = 0 ;
during:
StateCnt++ ;


1
PWMin
1
PWMout
Product2
200
Constant
60
Constant1
Sum
Product1
190
Constant2
Sum2
30
Constant3
Product3
Sum3
Sum1


[iReset==1]
reset
entry:
Counter = 0;
CntOverTemp
during:
Counter++;
[Counter > 80]
Wait for 400ms after reset
greenTemp
entry:
oTempRedAlarm = 0 ;
oTempAlarm = 0 ;
odPumpOff = 0;
Counter = 0;
iTemp < 75℃
[iTemp> Treshold_green]
iTemp > 110℃
[iTemp < Treshold_red]
redTemp
entry:
oTempRedAlarm = 1 ;
oTempAlarm = 1 ;
odPumpOff = 0; // test

Ubat_Handling

Ubat_Handling



Stateflow
Logger