



### Target: ECU555-80 (DEV)

Floating Point: single (32 bits)  
Stacks - FGND: 4096 BGND: 2048  
IDLE: 1024 IRQ: 1280

Heap Size: 4096

DLL Filename: BaseEngin\_002

SRZ Filename: BaseEngineController\_LS\_002



Total FLASH:	373552
Total EEPROM:	1254
Total RAM:	26040
App FLASH:	154160
App EEPROM:	1080
App RAM:	7766

MotoHawk (RTW)  
Fault Manager Definition

Storage: FLASH  
X/Y Data Type: uint16  
Read Access: 1  
Write Access: 1  
Clear Access: 1

MotoHawk Function Trigger  
FGND\_RTI\_PERIODIC  
Priority Order: 0

f()

Foreground

Main Power Relay  
On Delay: 100 ms  
Off Delay: 250 ms

Main Power Relay

### MotoHawk Trigger

FGND\_RTI Period: 5 ms  
BGND\_BASE Period: 50 ms

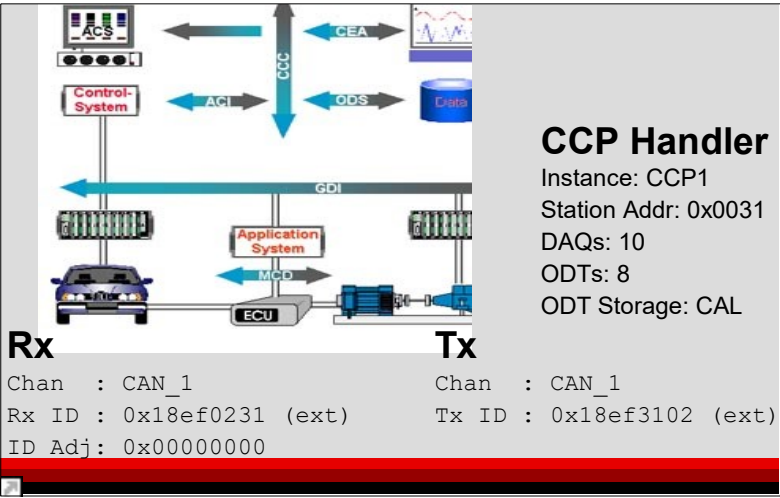
### MotoHawk CAN Definition

**Name:** CAN\_1  
**Bus:** CAN1  
**Bit Timing:** 500 kbaud  
**TX Queue:** 16 messages  
**RX Queue:** 16 messages

MotoTune Protocol Enabled  
**City ID:** 0x0B (PCM-1)

**r Definition**

is  
0 ms (FGND x 10)



CCP CAN Protocol



## MotoHawk CAN Definition

**Name:** CAN\_2

**Bus:** CAN2

**Bit Timing:** 500 kbaud

**TX Queue:** 16 messages

**RX Queue:** 16 messages

MotoTune Protocol Enabled

**City ID:** 0x0B (PCM-1)

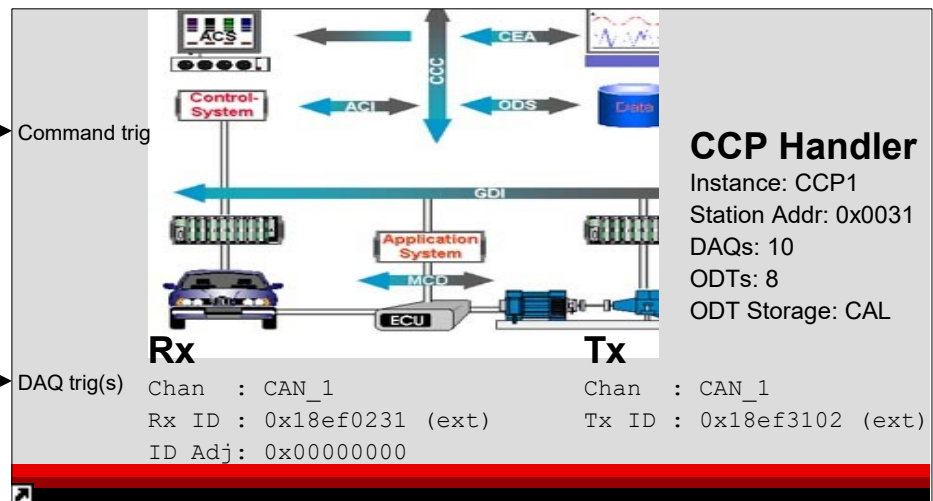




**MotoHawk CCP DAQ Triggers:**

- FGND\_RTI\_PERIODIC
- FGND\_5XRTI\_PERIODIC
- FGND\_MID\_TDC\_EVENT
- FGND\_20XRTI\_PERIODIC
- BGND\_BASE\_PERIODIC

MotoHawk Function Trigger  
IDLE\_EVENT  
Priority Order: 255  
(Used in triggered subsystem)  
(Runs even when App Monitor stops)



CCP CAN Protocol External Trig1

f()

f



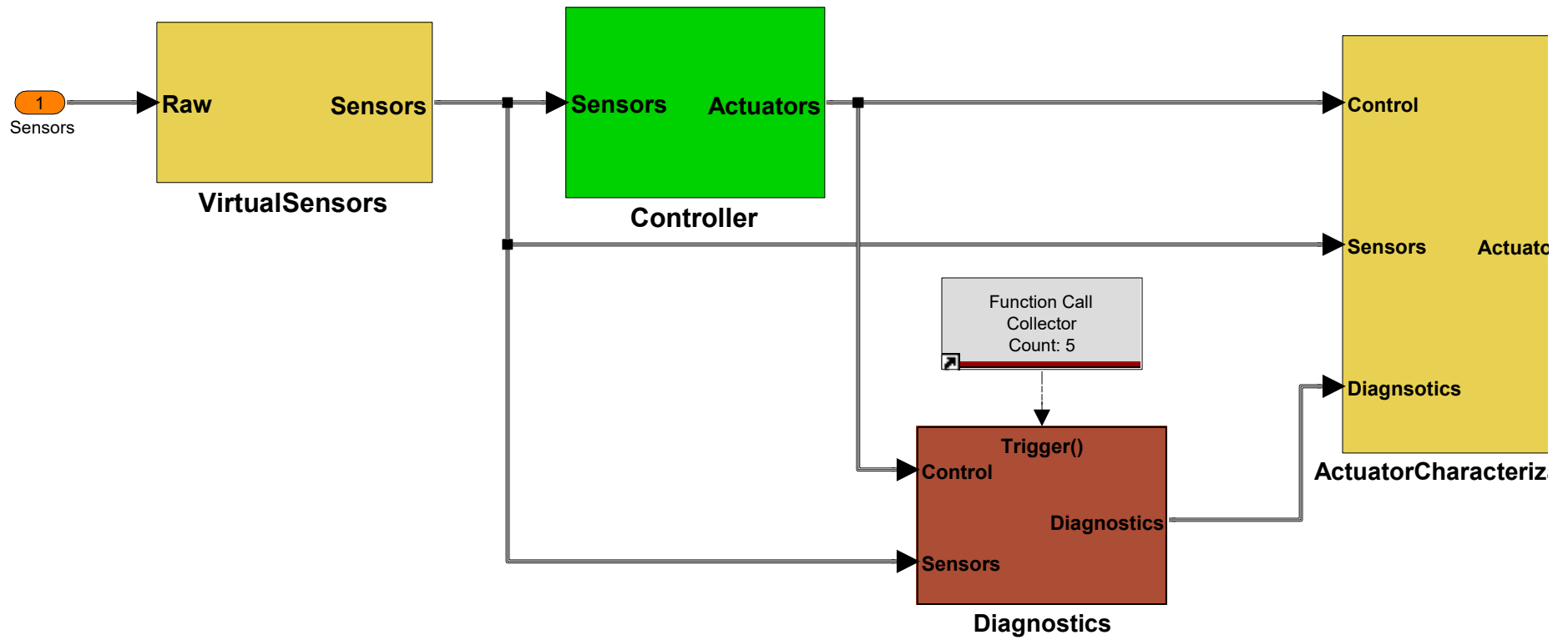
Inputs

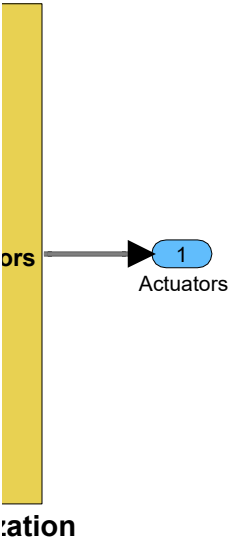


Control



Outputs







3  
Diagnostics

Diagnostics

Controller

Sensors

1  
Control

2  
Sensors

- double
- <SparkAdv>
- double
- <SparkEnergy>
- double
- <FPC>
- double
- <FuelMult>
- double
- BaroPress [kPa]
- <Baro>
- double
- <MAP\_Angular>
- index\_T
- <SysVolt09Idx>
- double
- <RPM>
- double
- <FuelPres>

FPC [mg]

FuelMult

MPW [ms]

BaroPress [kPa]

MAP [kPa]

SysVolt09Idx

RPM

EPWOffset [ms]

FuelPress [kPa]

INJECTOR  
CHRZN.  
(High Z Only)

Injector Characterization

MotoHawk Function Trigger  
FGND\_5XRTI\_PERIODIC  
Priority Order: 1  
(Used in triggered subsystem)

Stop Angle [degCA]

RPM

PulseWidth [ms]

Start Angle [degCA]

Trigger()

Spark Advance to Start Angle

double

InjMPW

double

InjEPWOffset

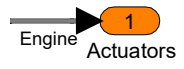
SparkStartAngle

double

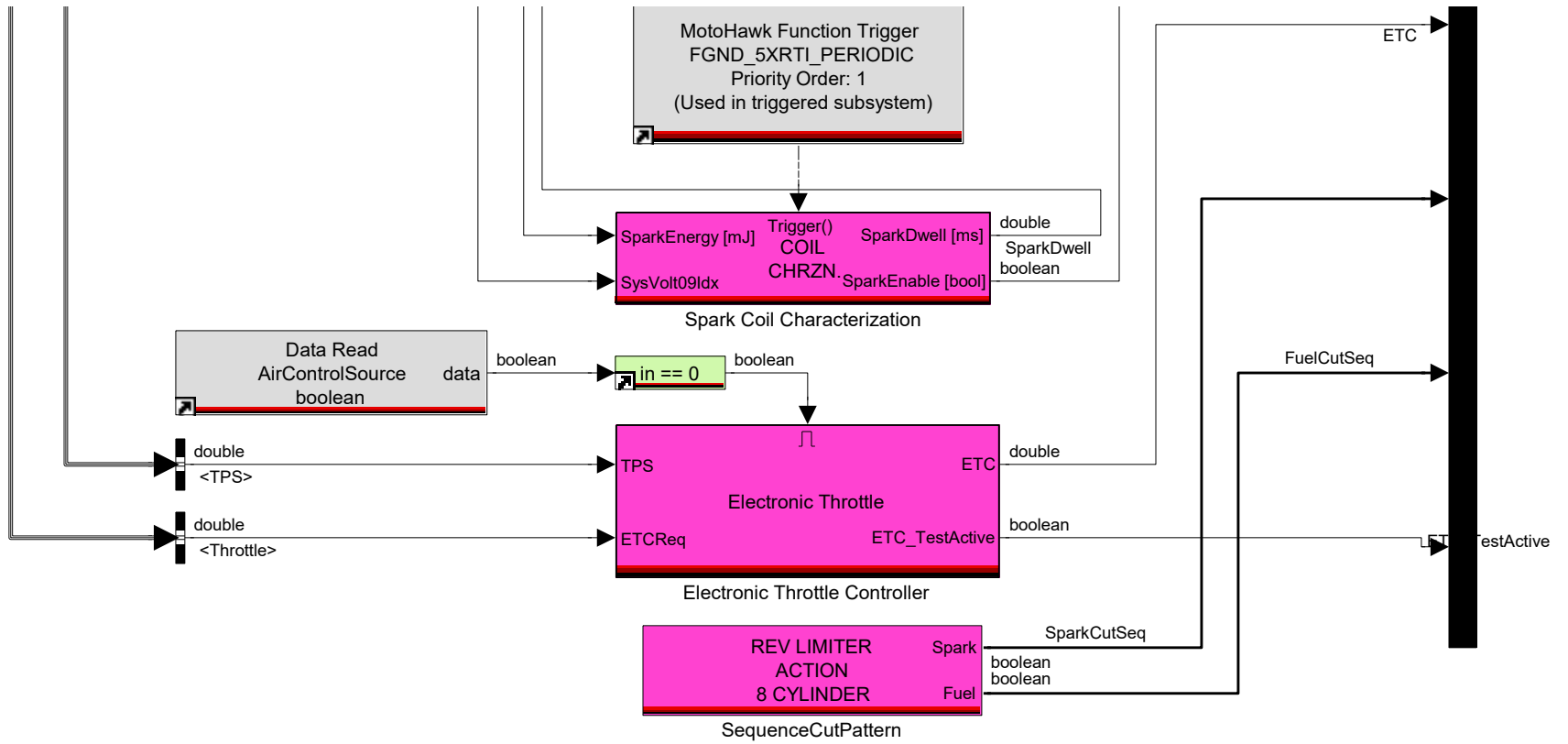
SparkStartAngle

SparkEnable

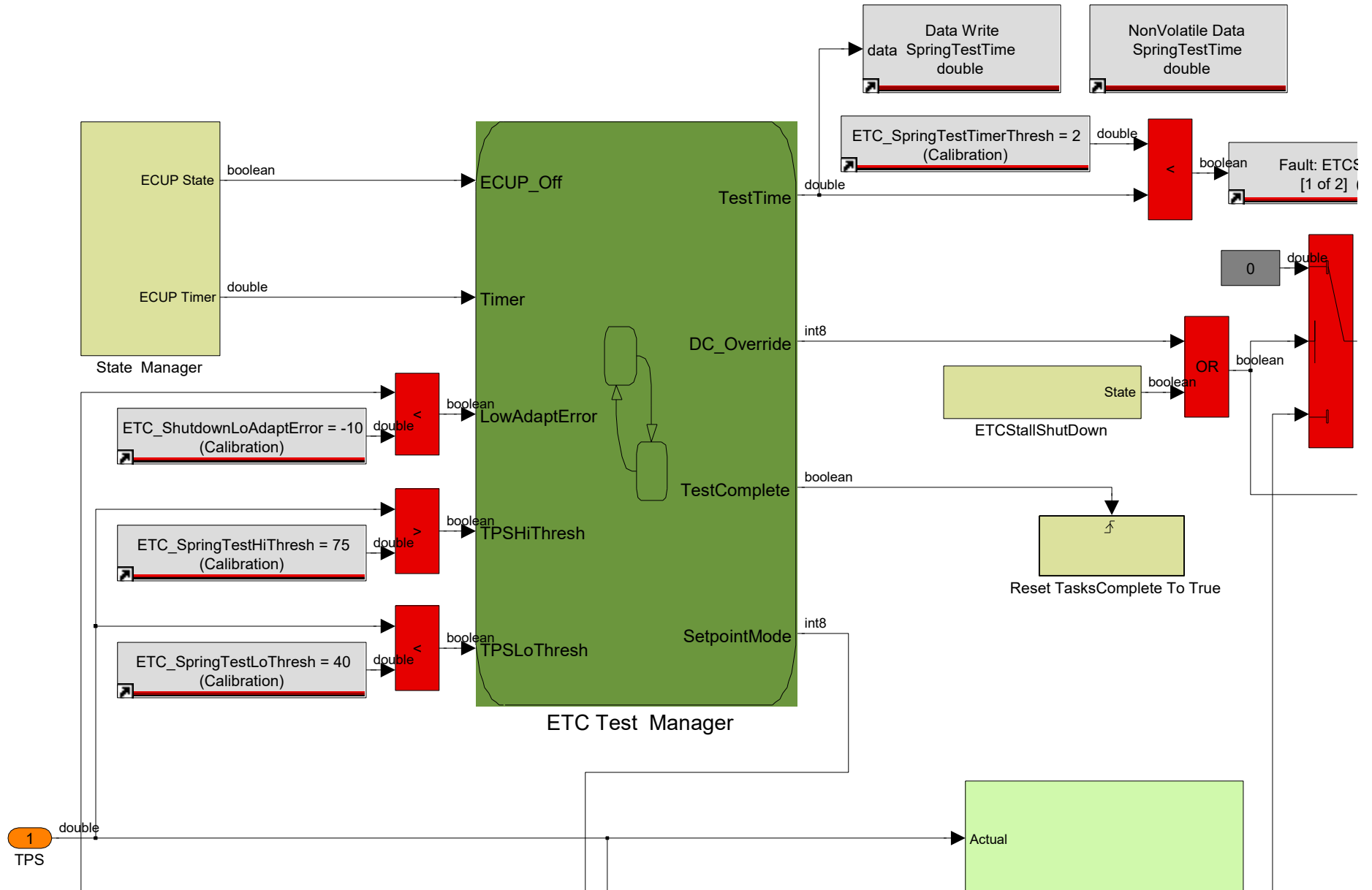
ActuatorChare



acterization



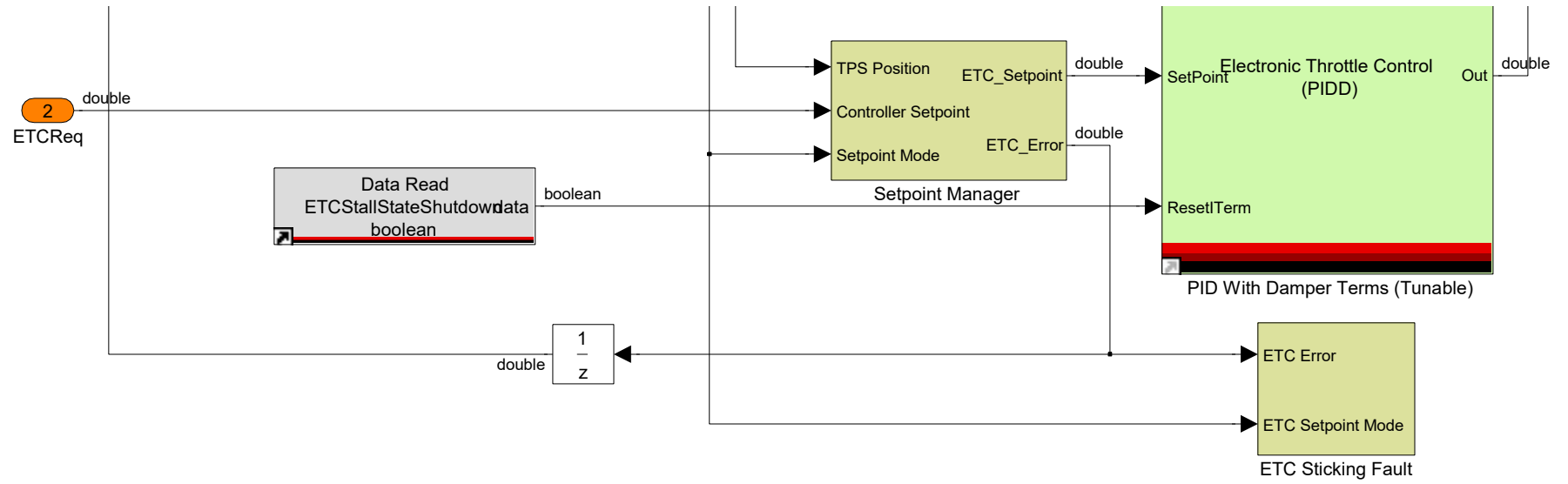




SpringTest  
(Sticky)

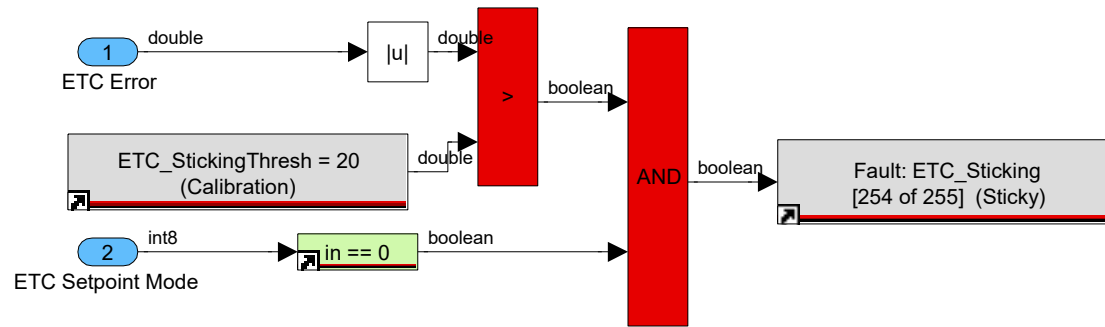
double → 1  
ETC

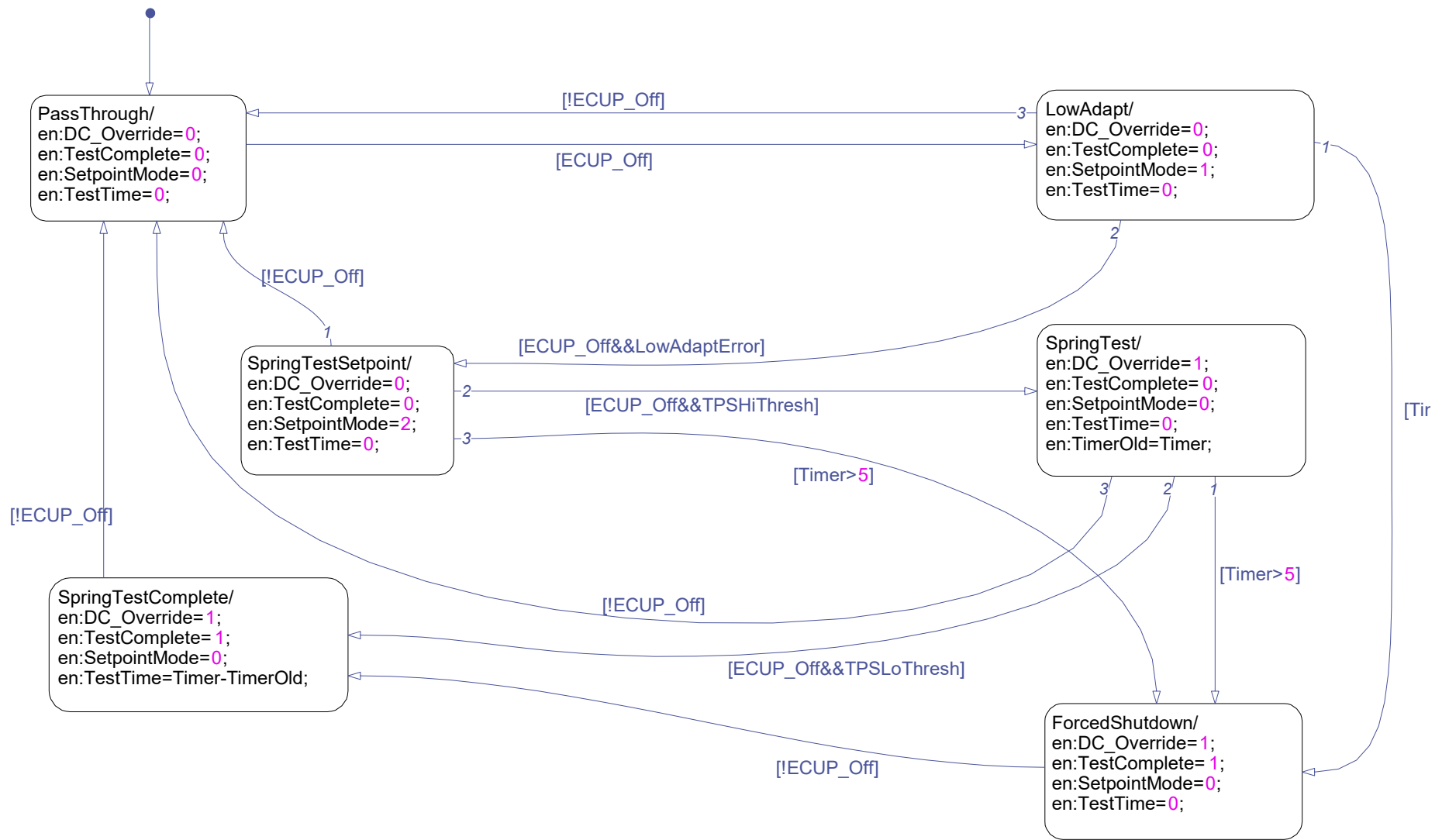
→ 2  
ETC\_TestActive



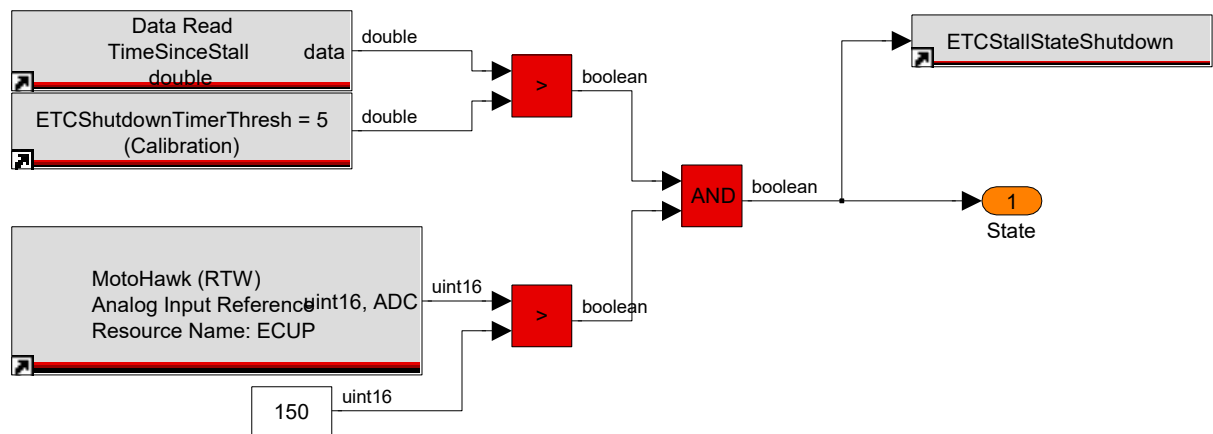


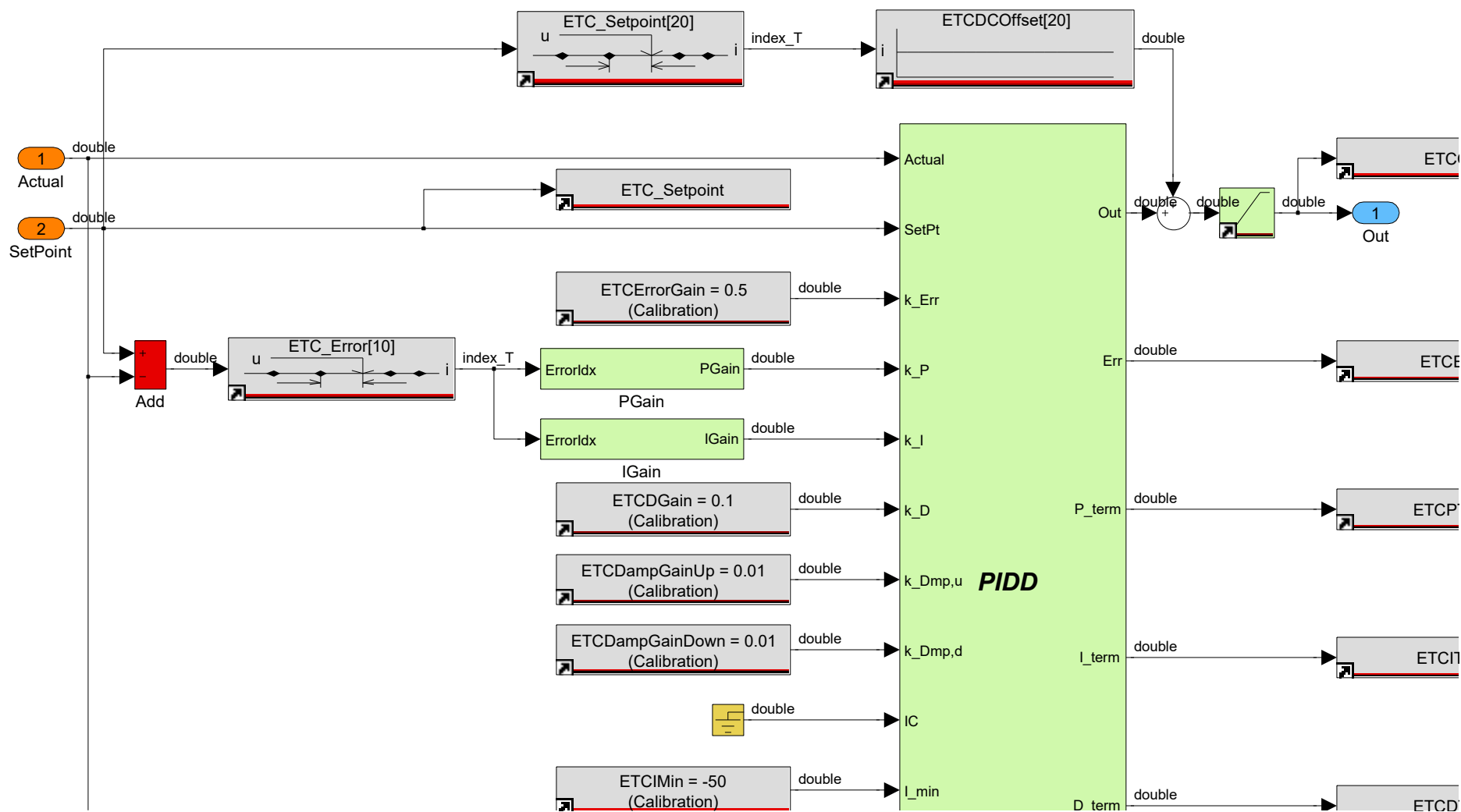






mer>5]





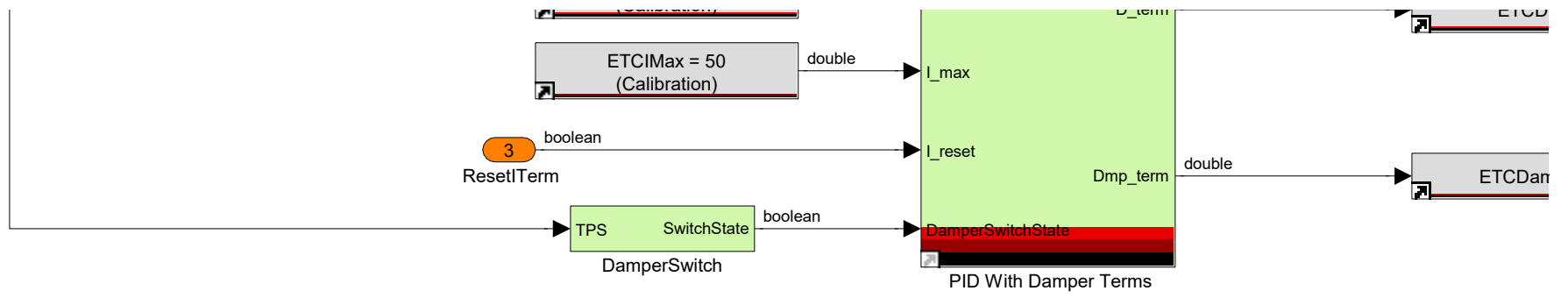
:Out

Error

'Term

Term

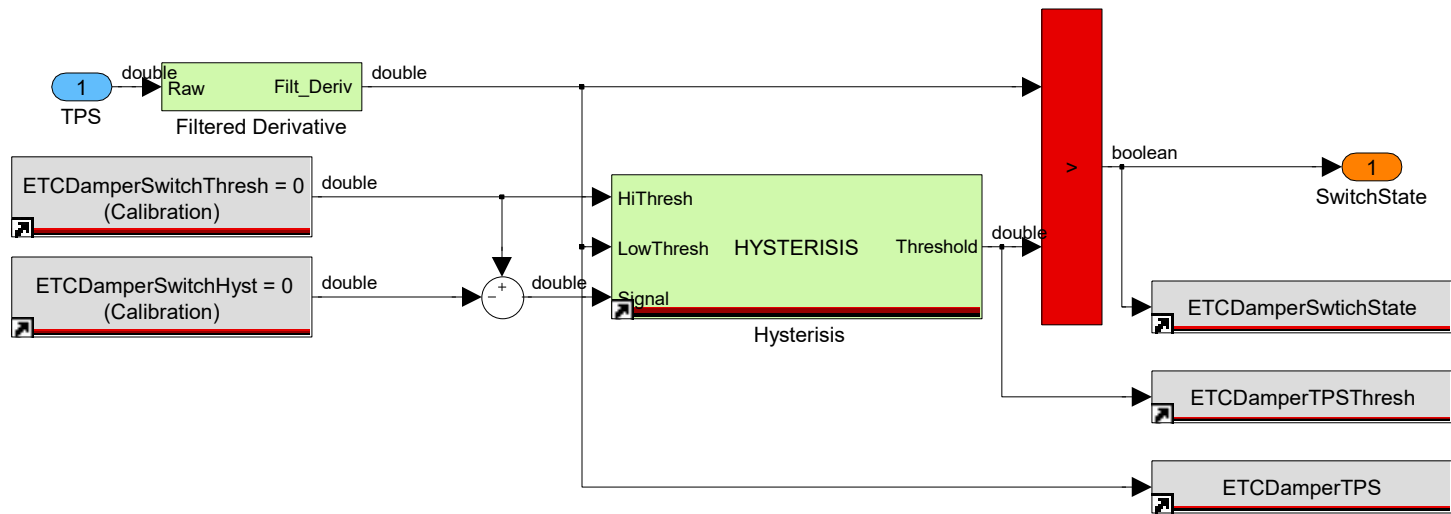
)Term

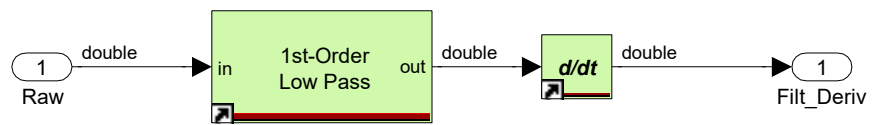


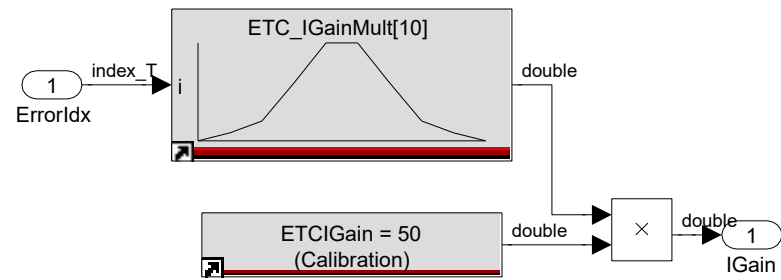
Term

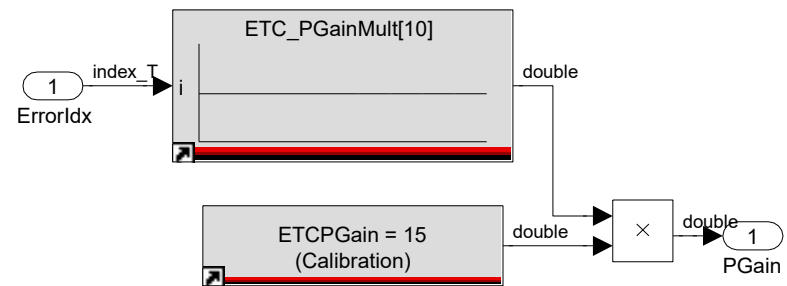
npTerm





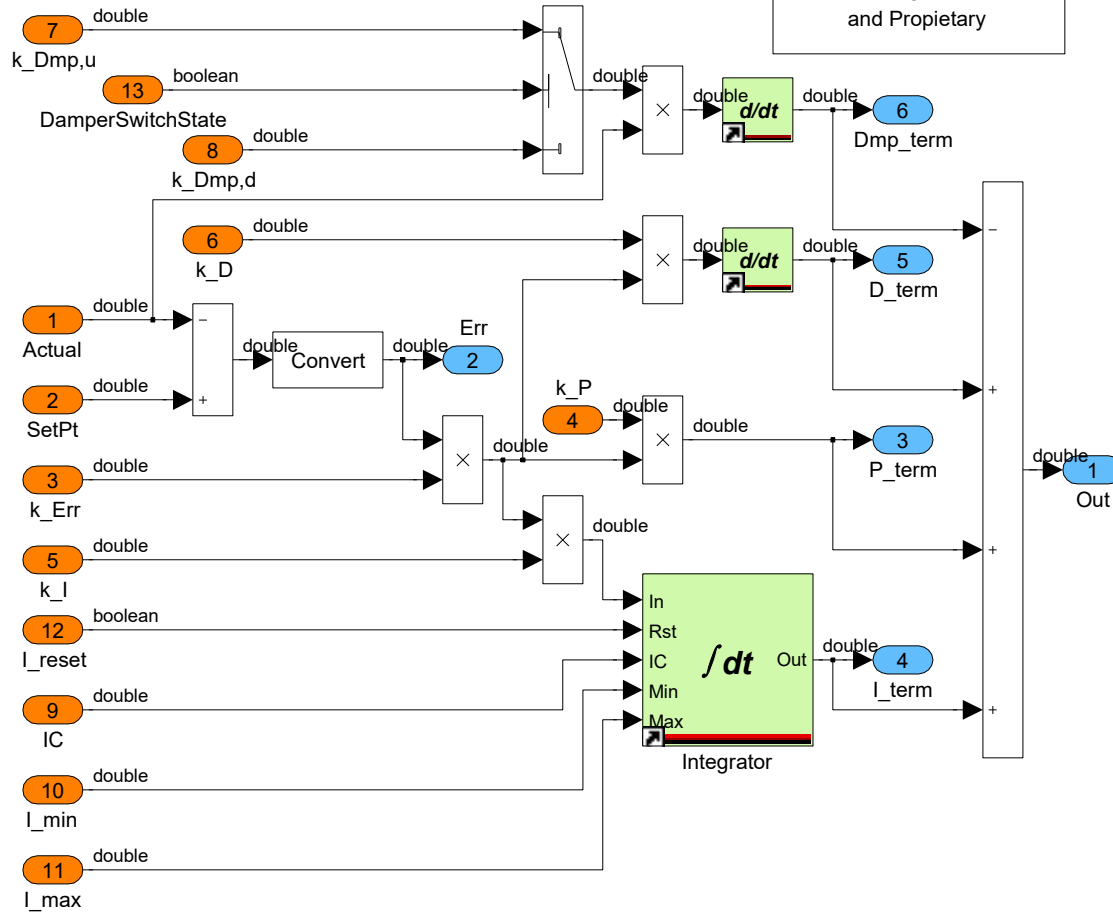






# Universal PID Controller with Dampers

Model Info:  
Author: Suhre, Poorman  
Mon May 16 15:00:08 2005  
Sun Nov 17 04:24:00 2019  
Version: 1.1787  
MotoTron Confidential  
and Proprietary

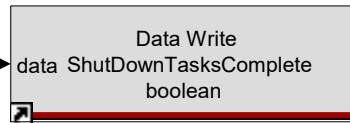


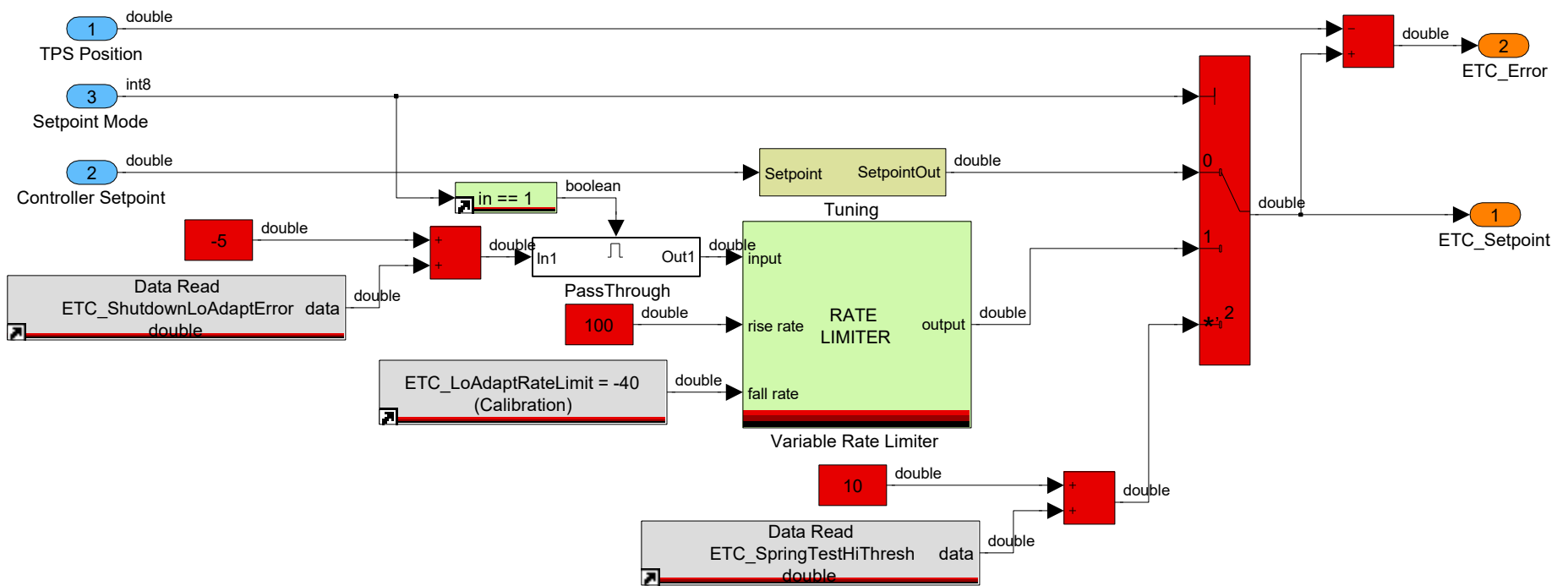


Trigger

true

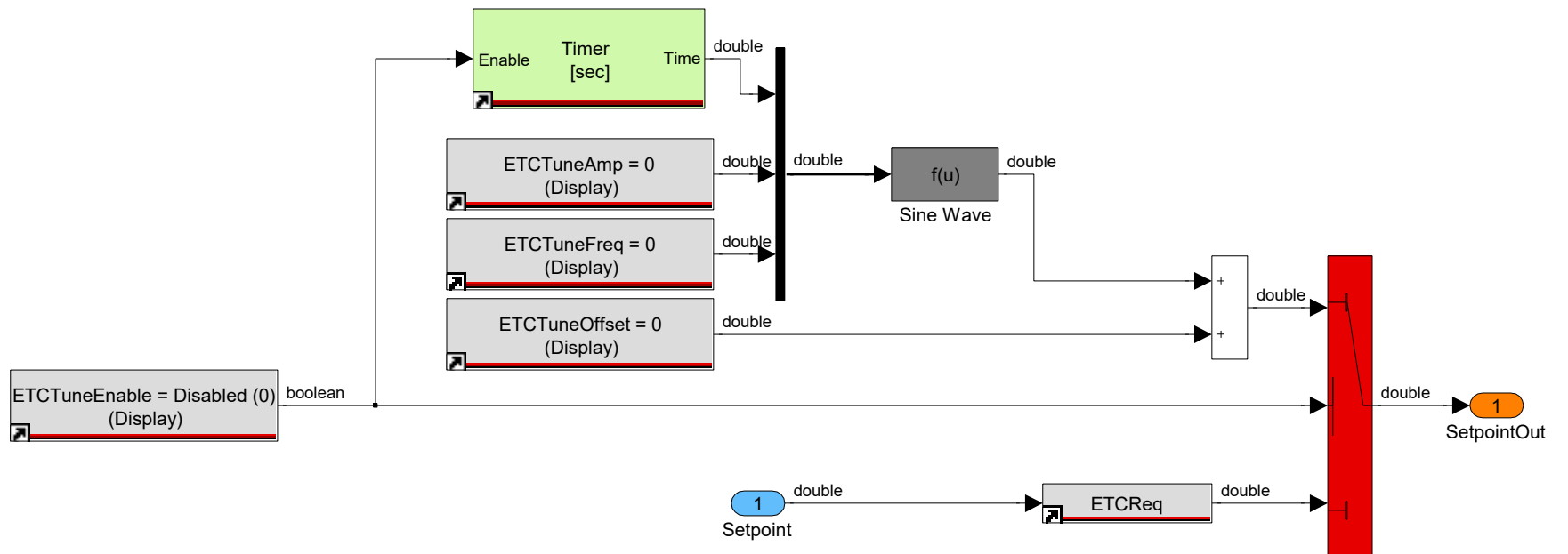
boolean



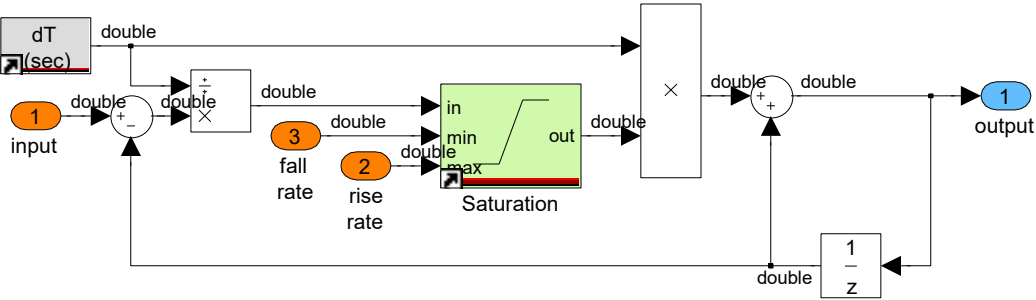


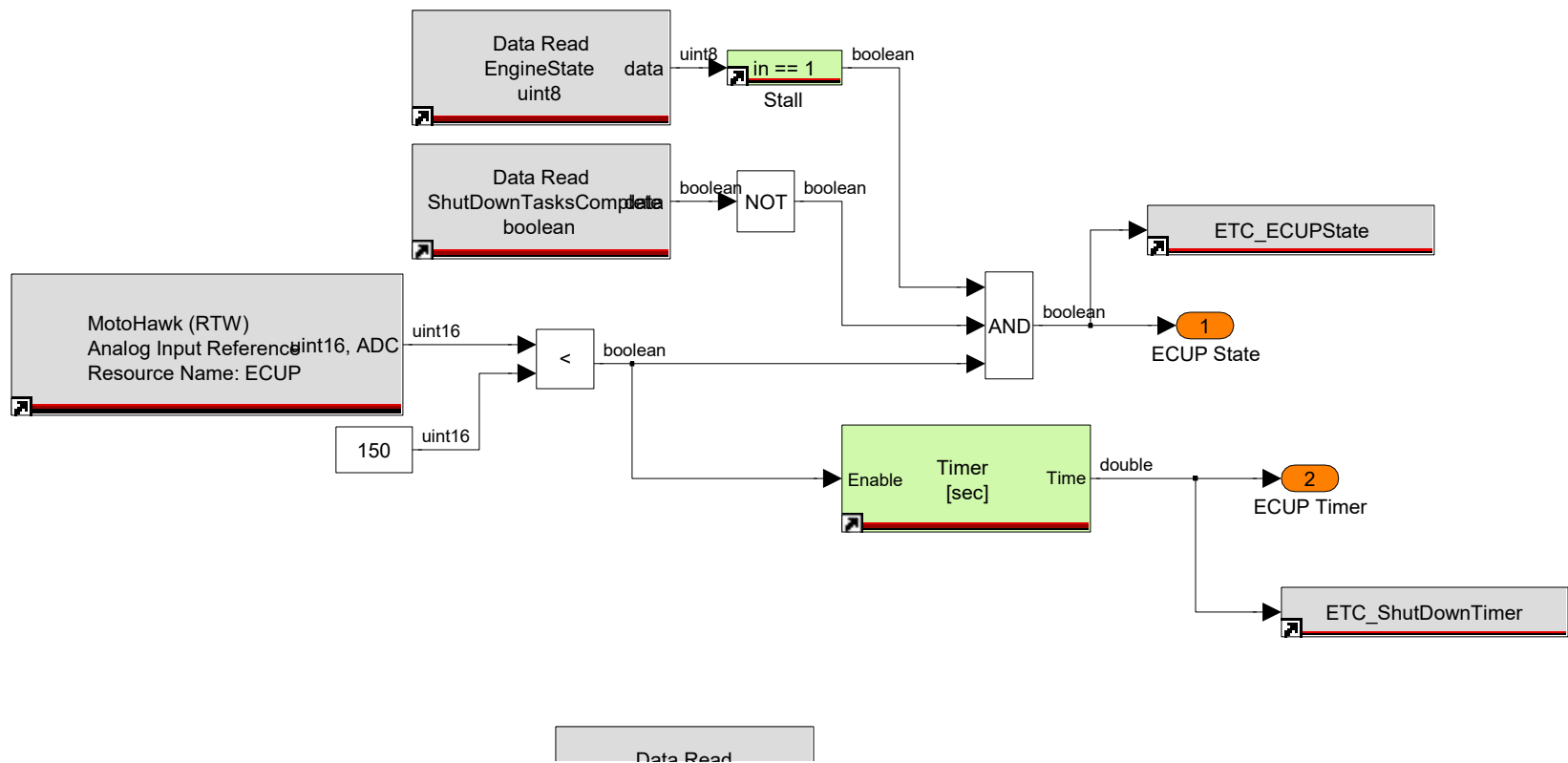


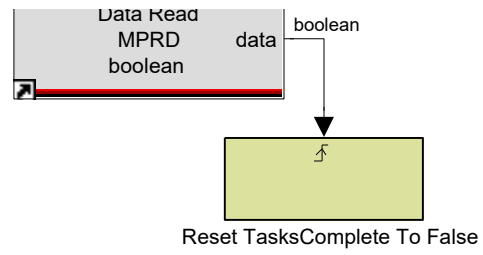




Rate Limiter - Limit allowable change in signal per timestep





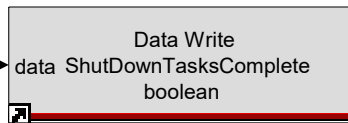


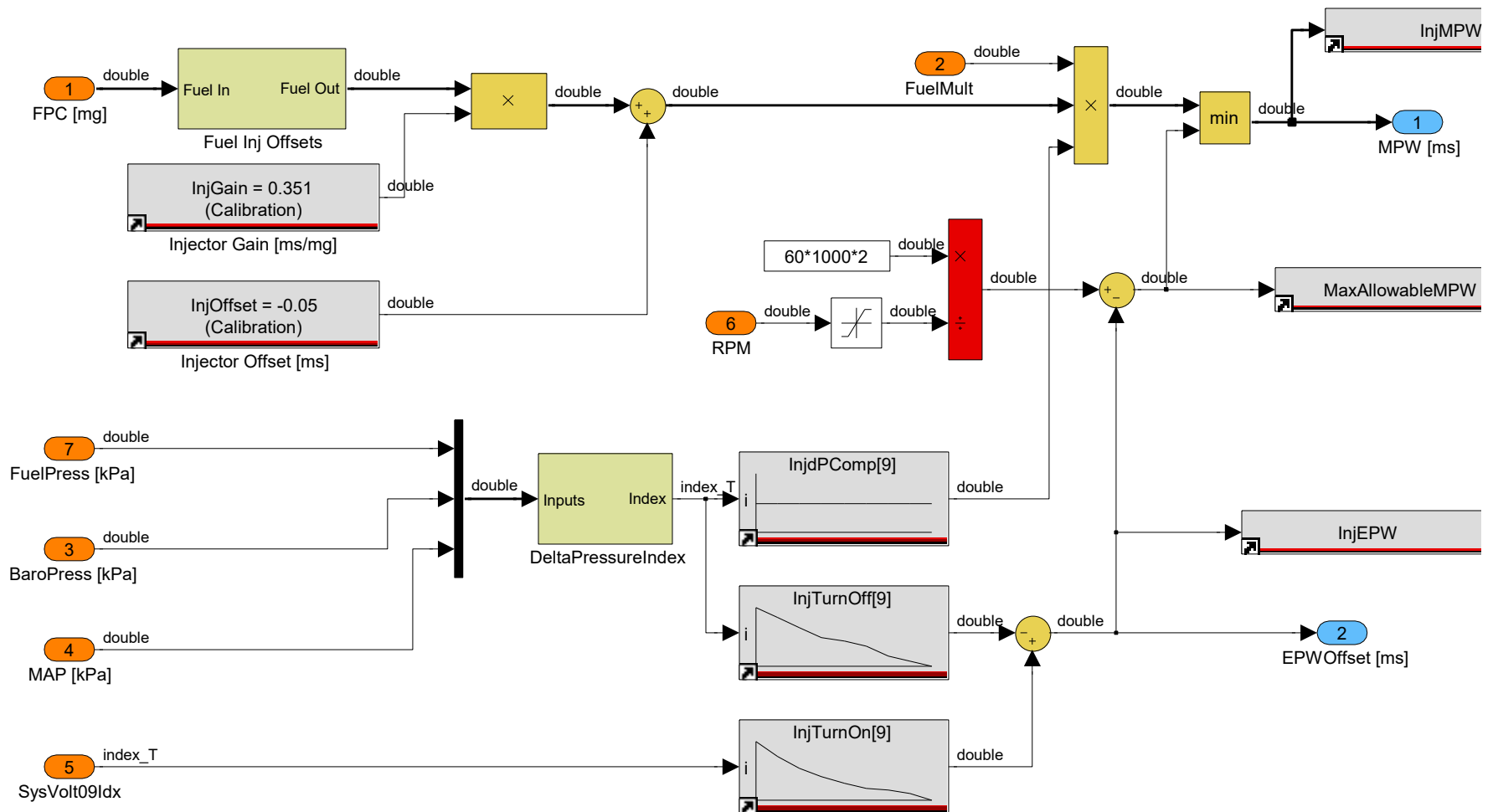


Trigger

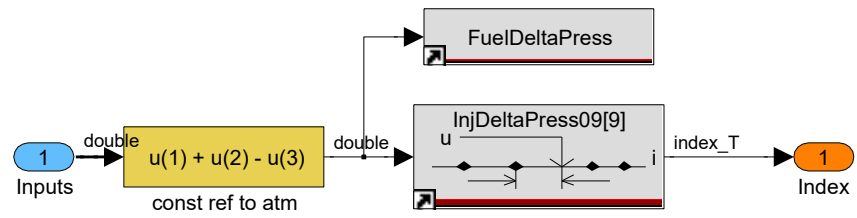
false

boolean

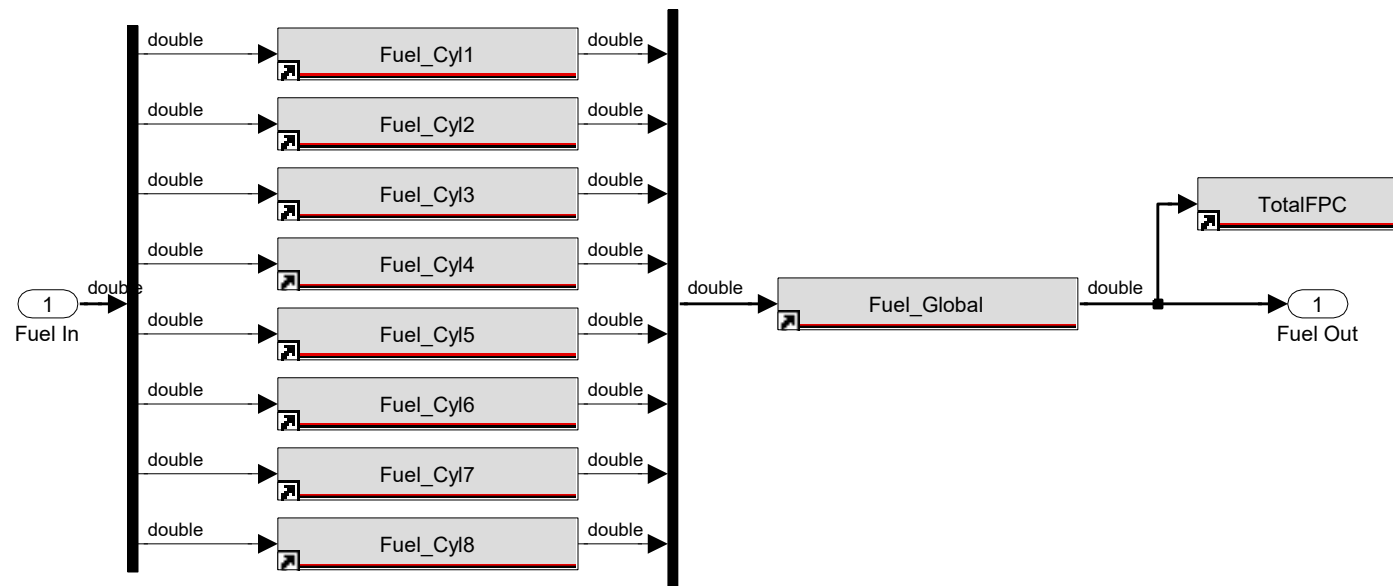


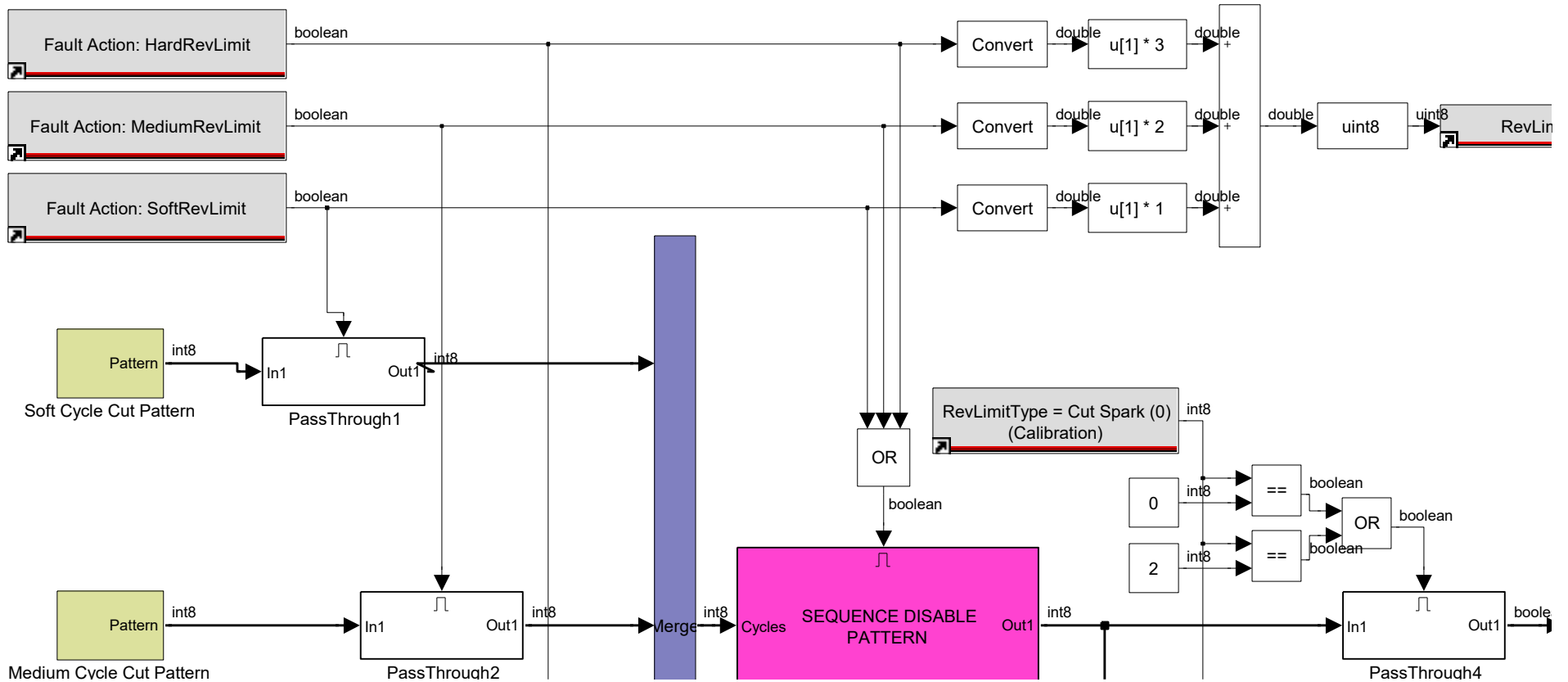




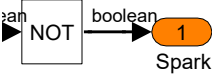






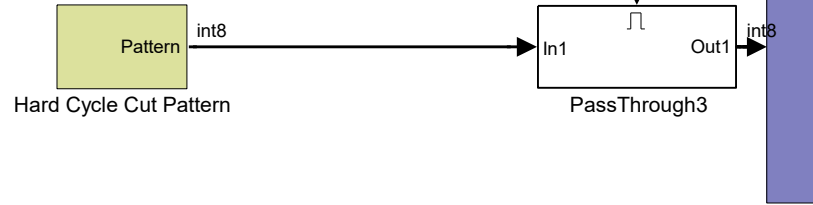


miterState

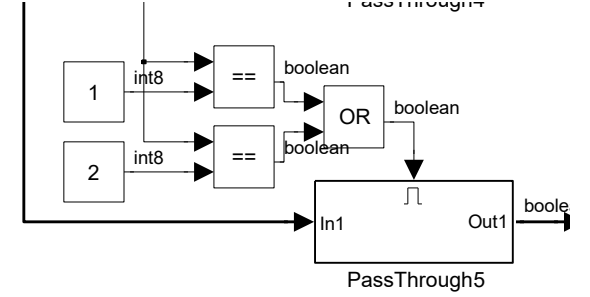


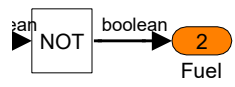
Medium Cycle Cut Pattern

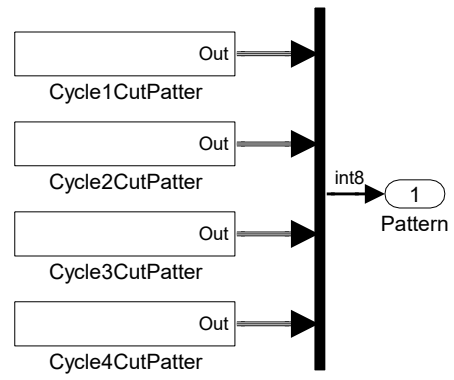
PassThrough2

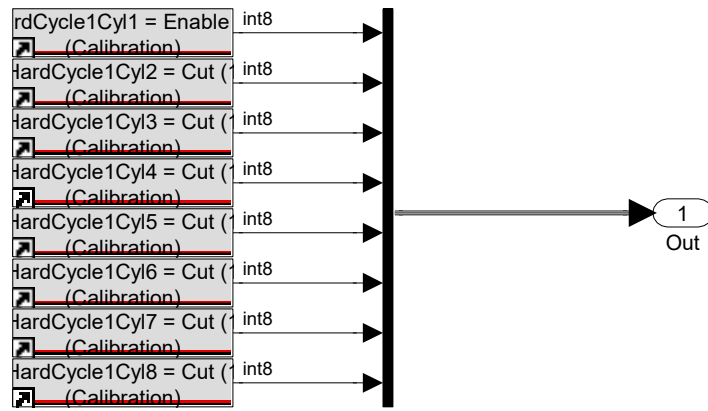


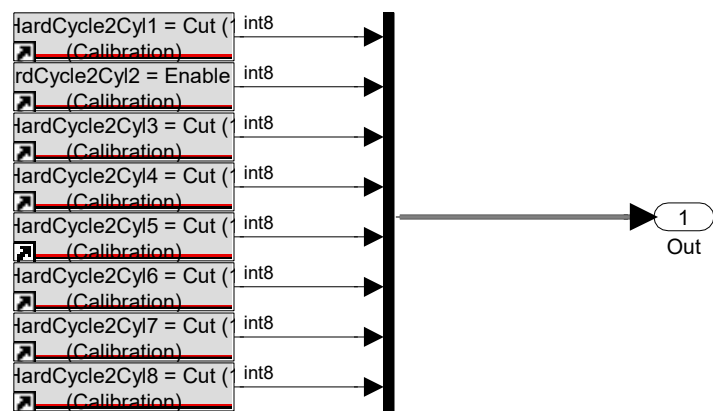
PassThrough4



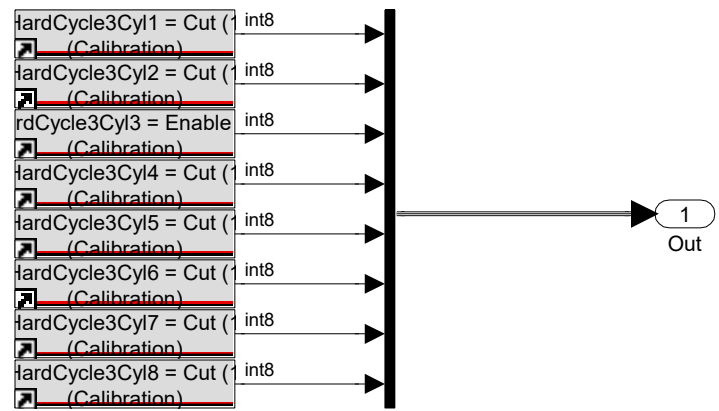


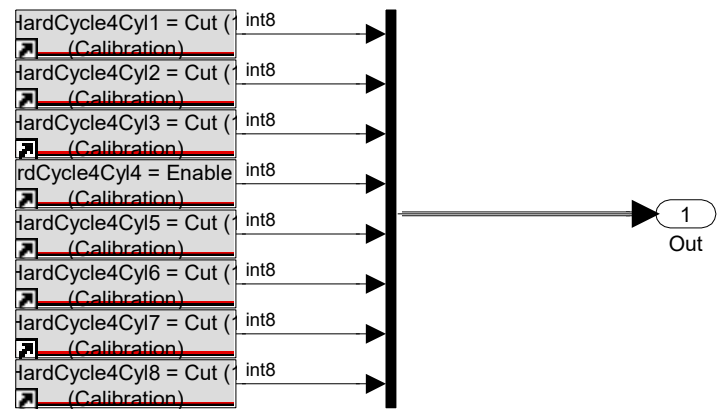


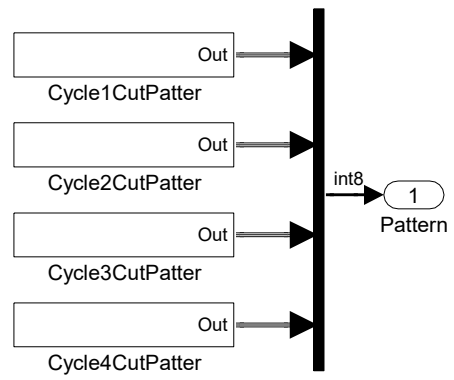


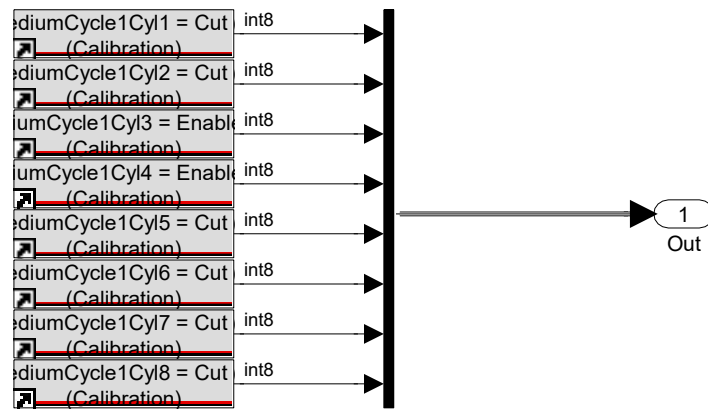


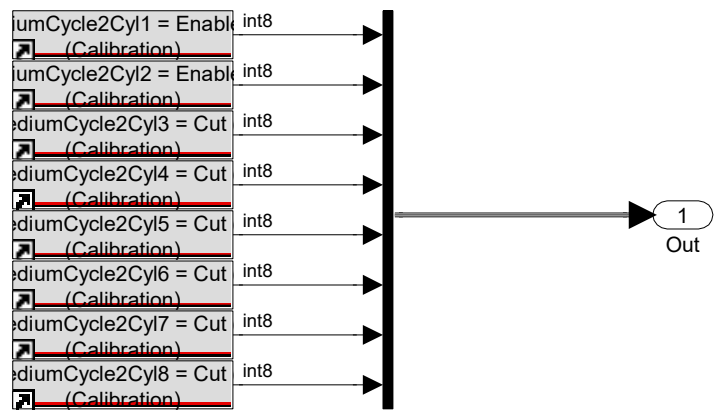


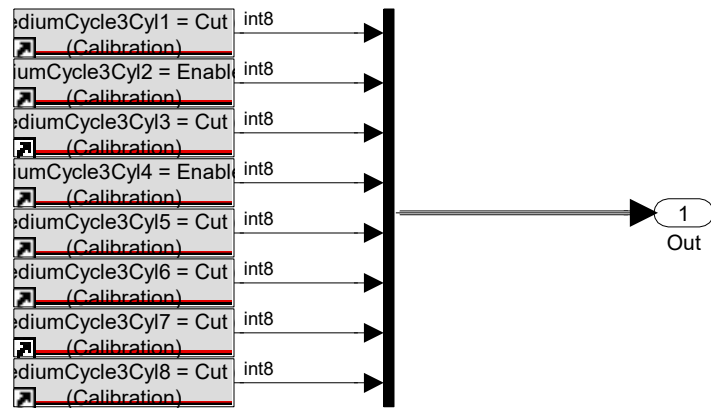


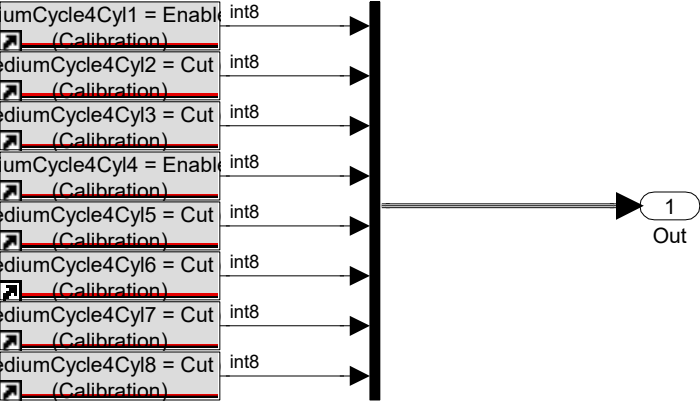


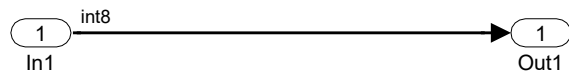




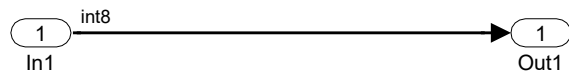


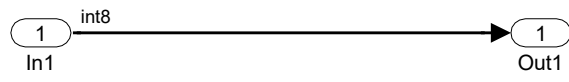


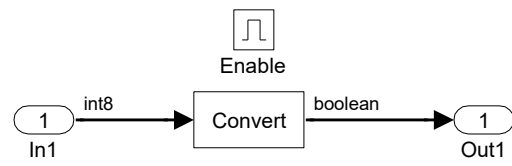


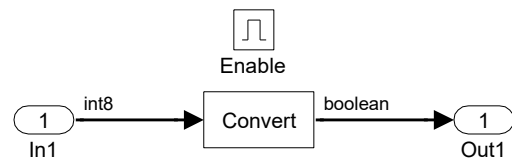


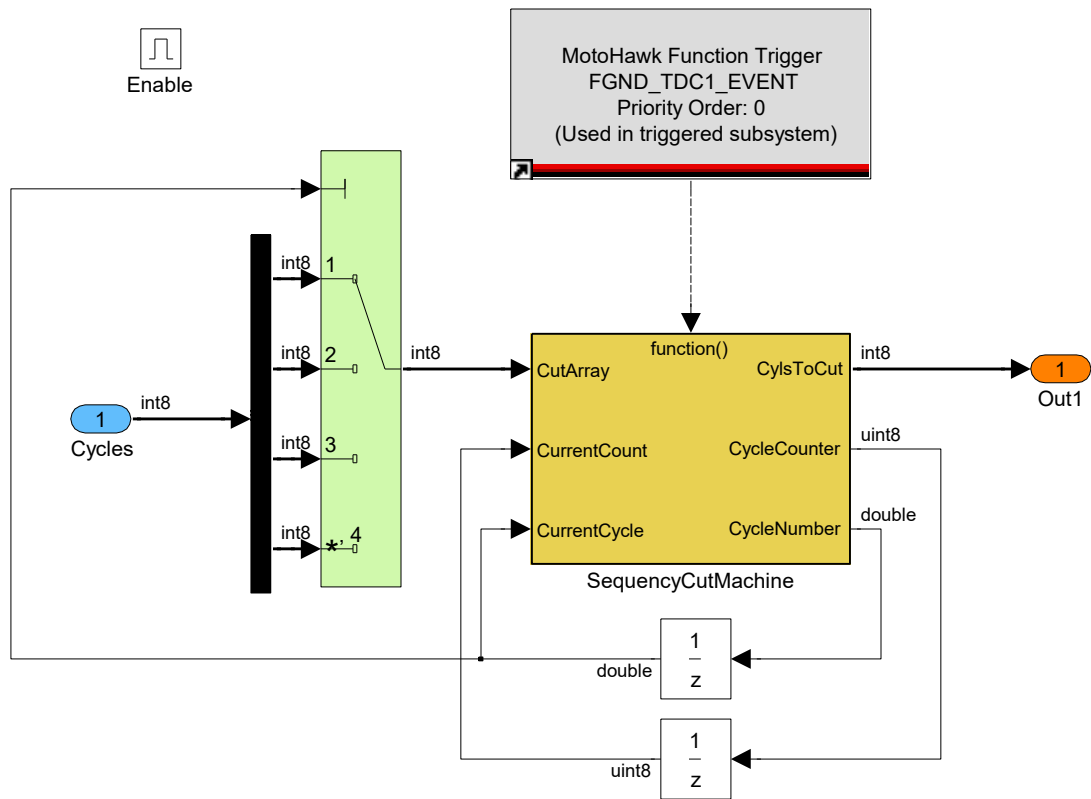


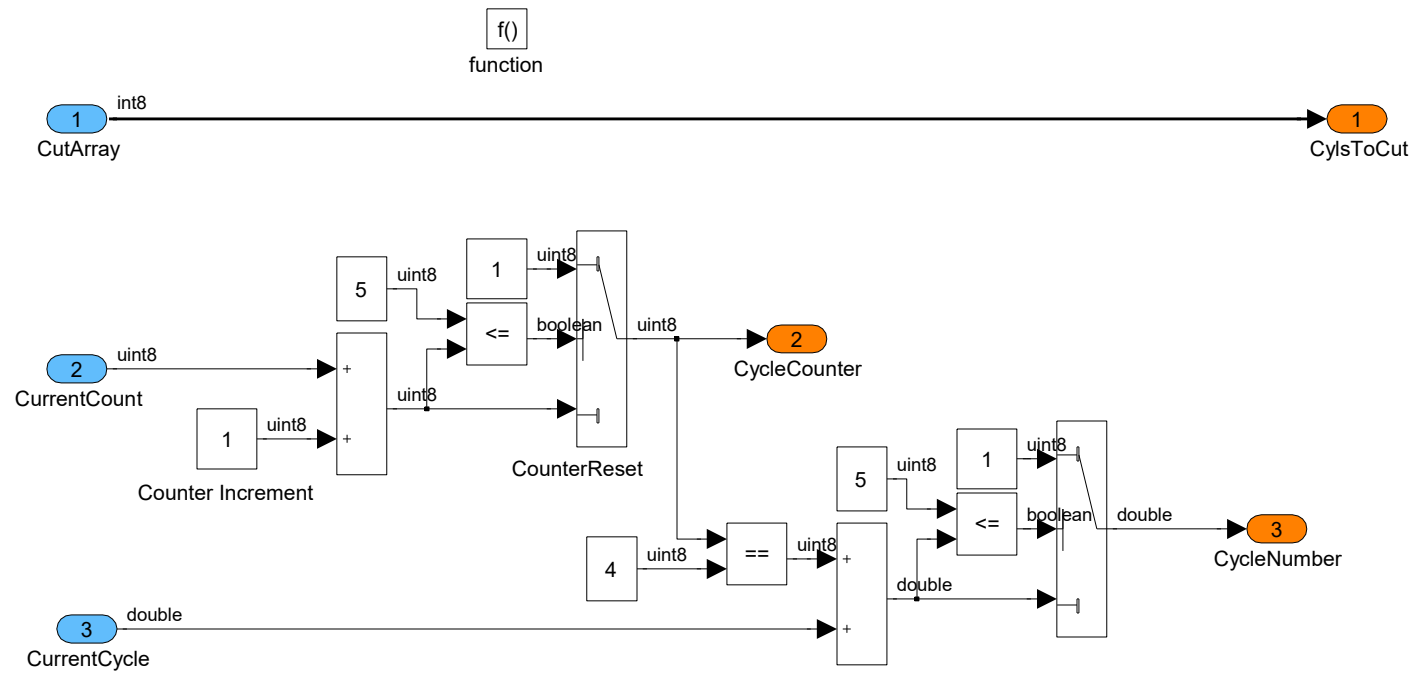


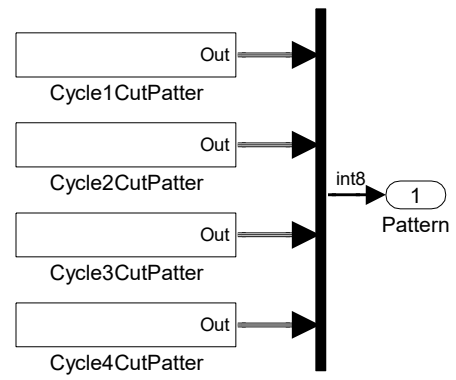


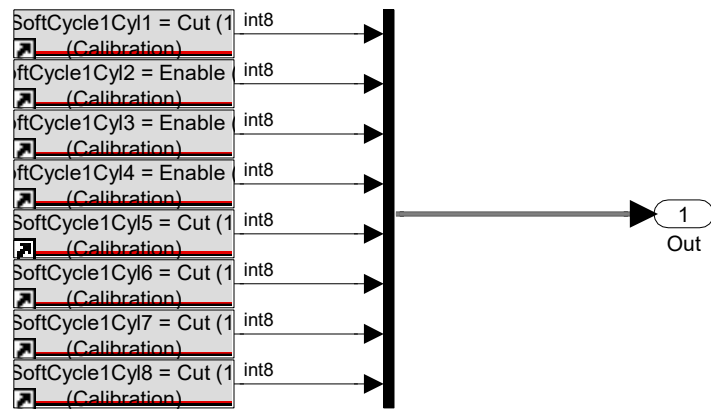




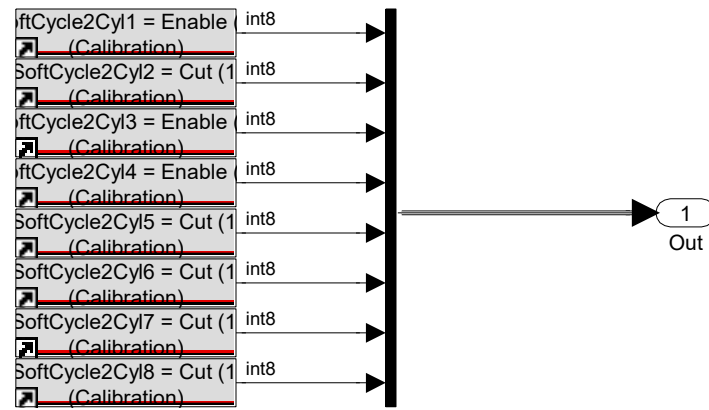


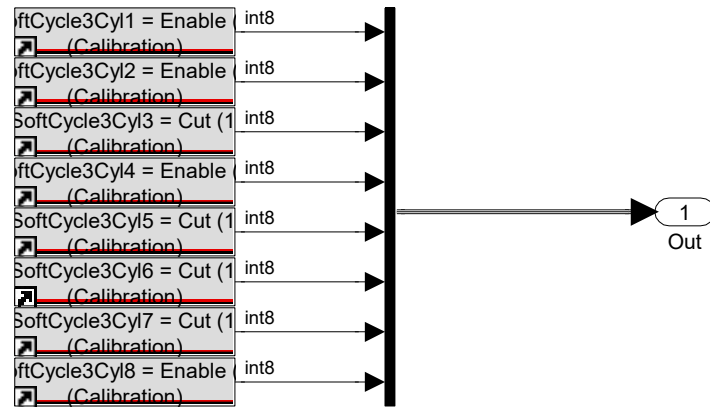


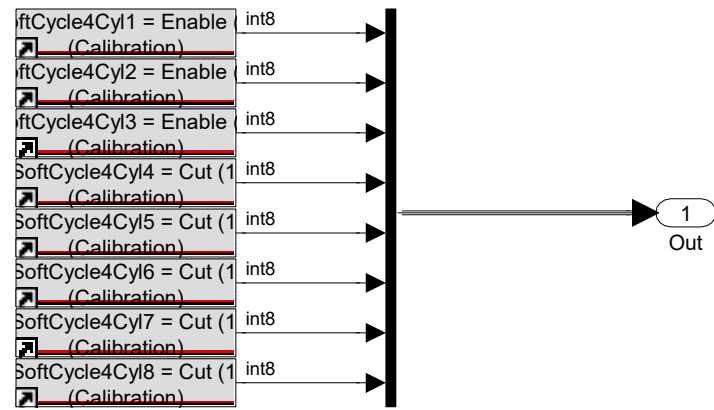




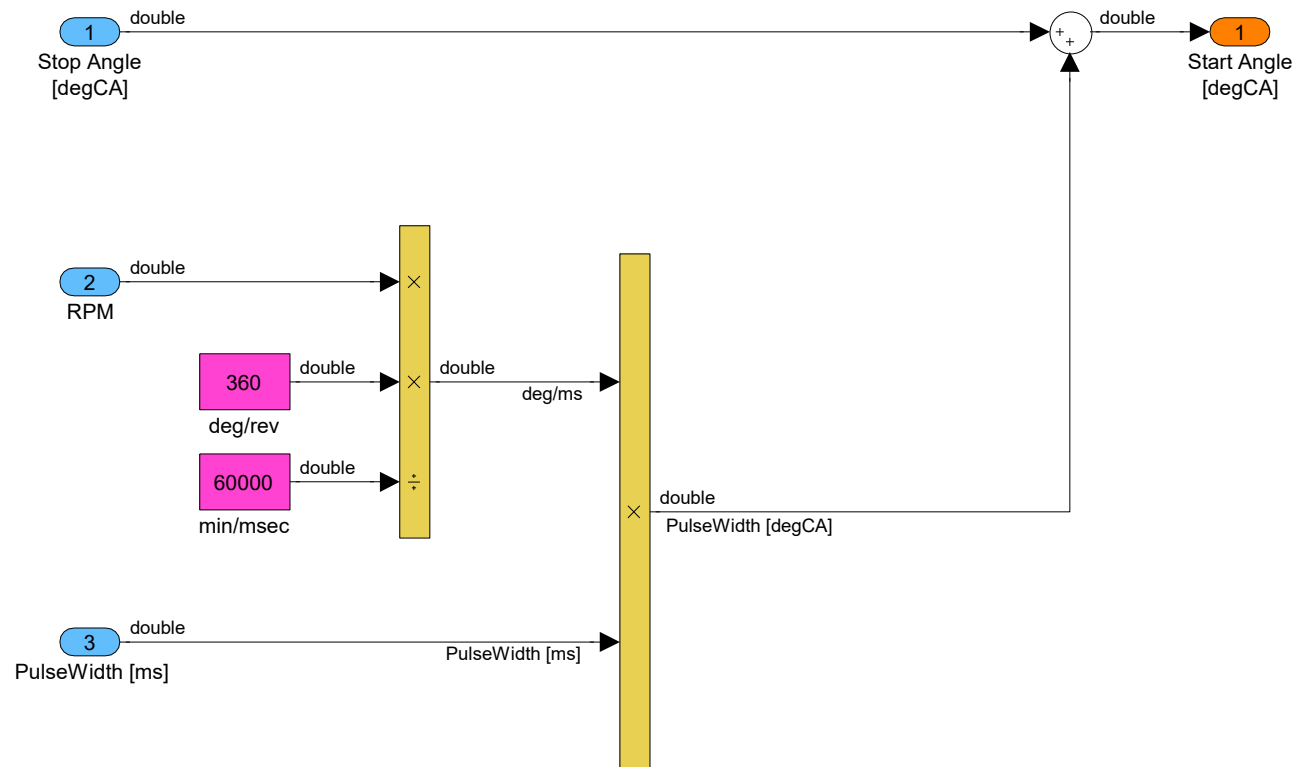




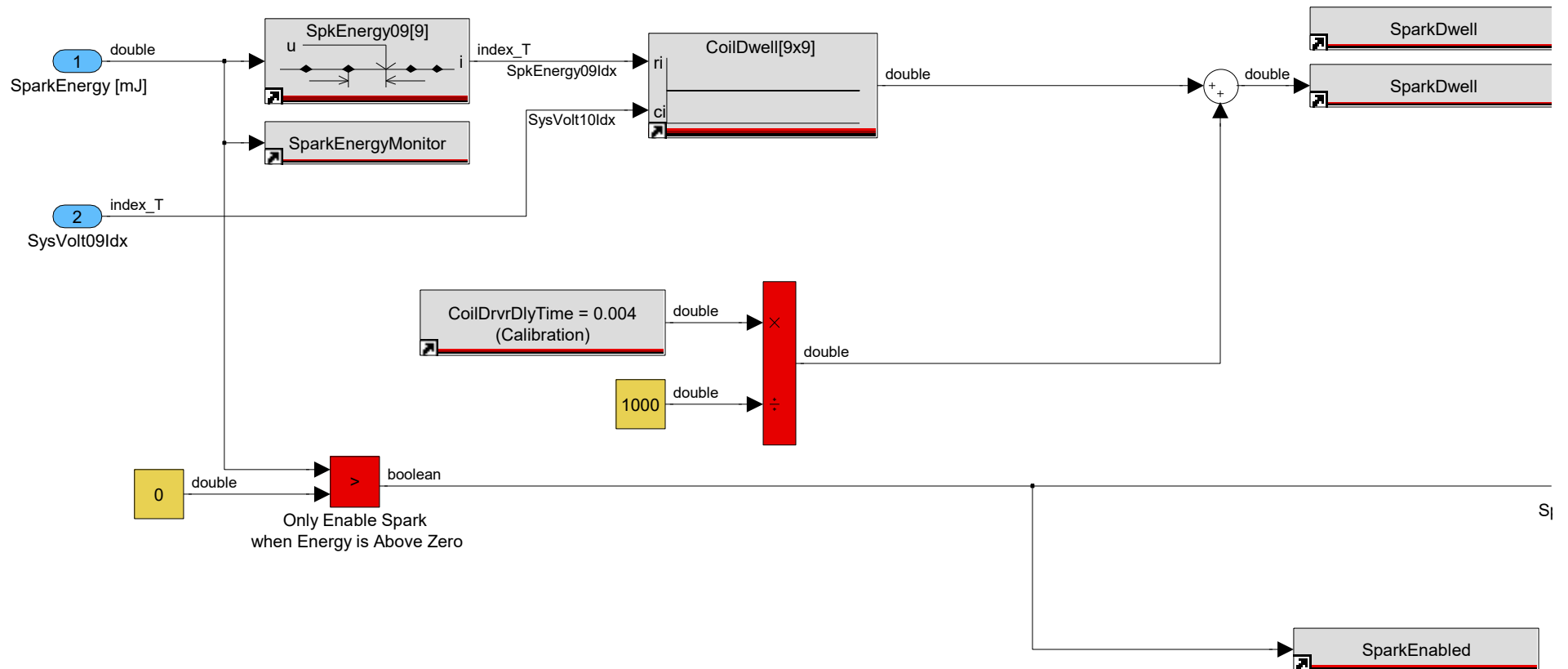


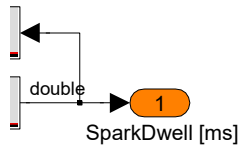


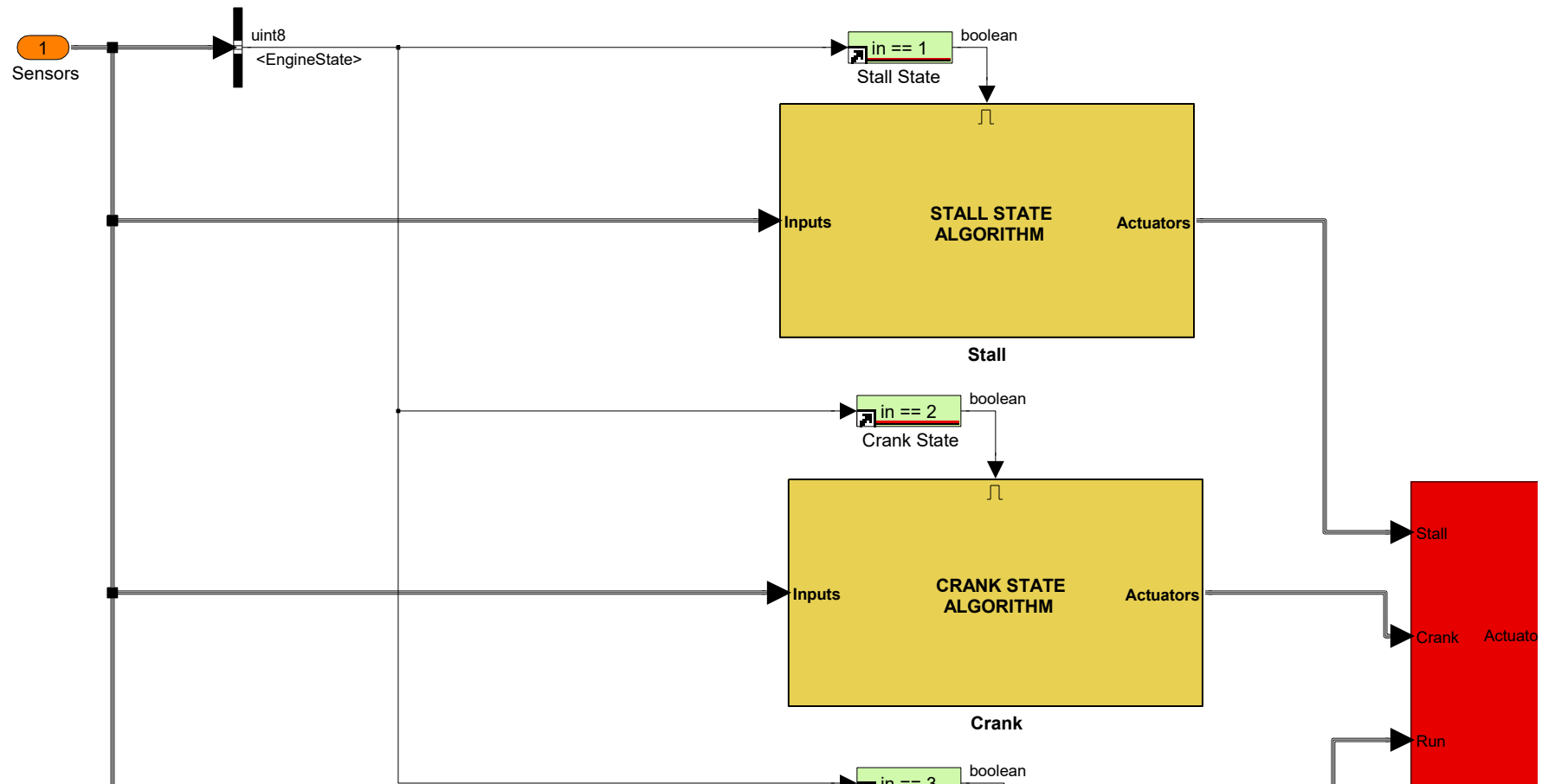
f()

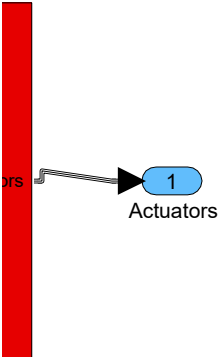


f()

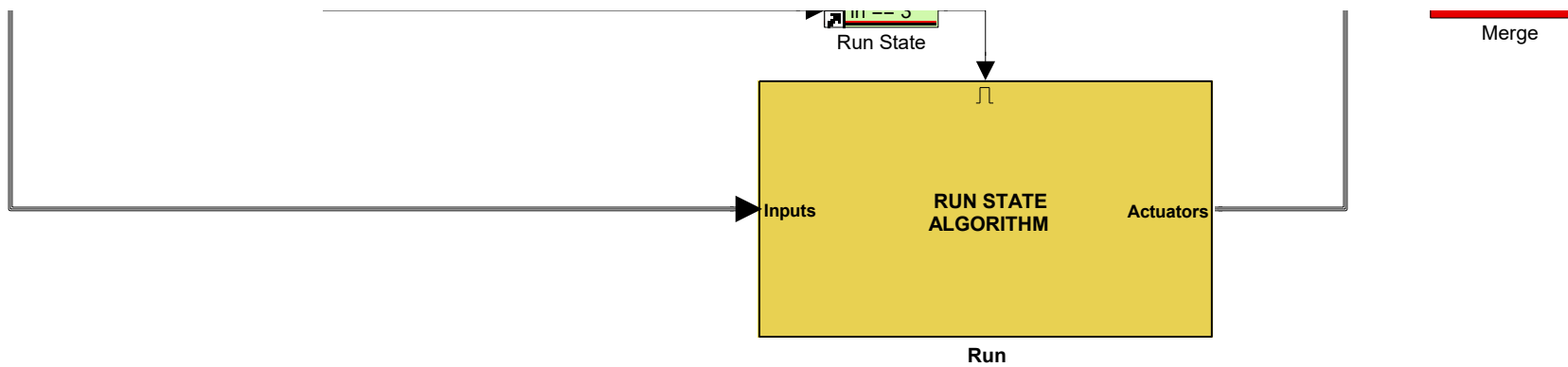




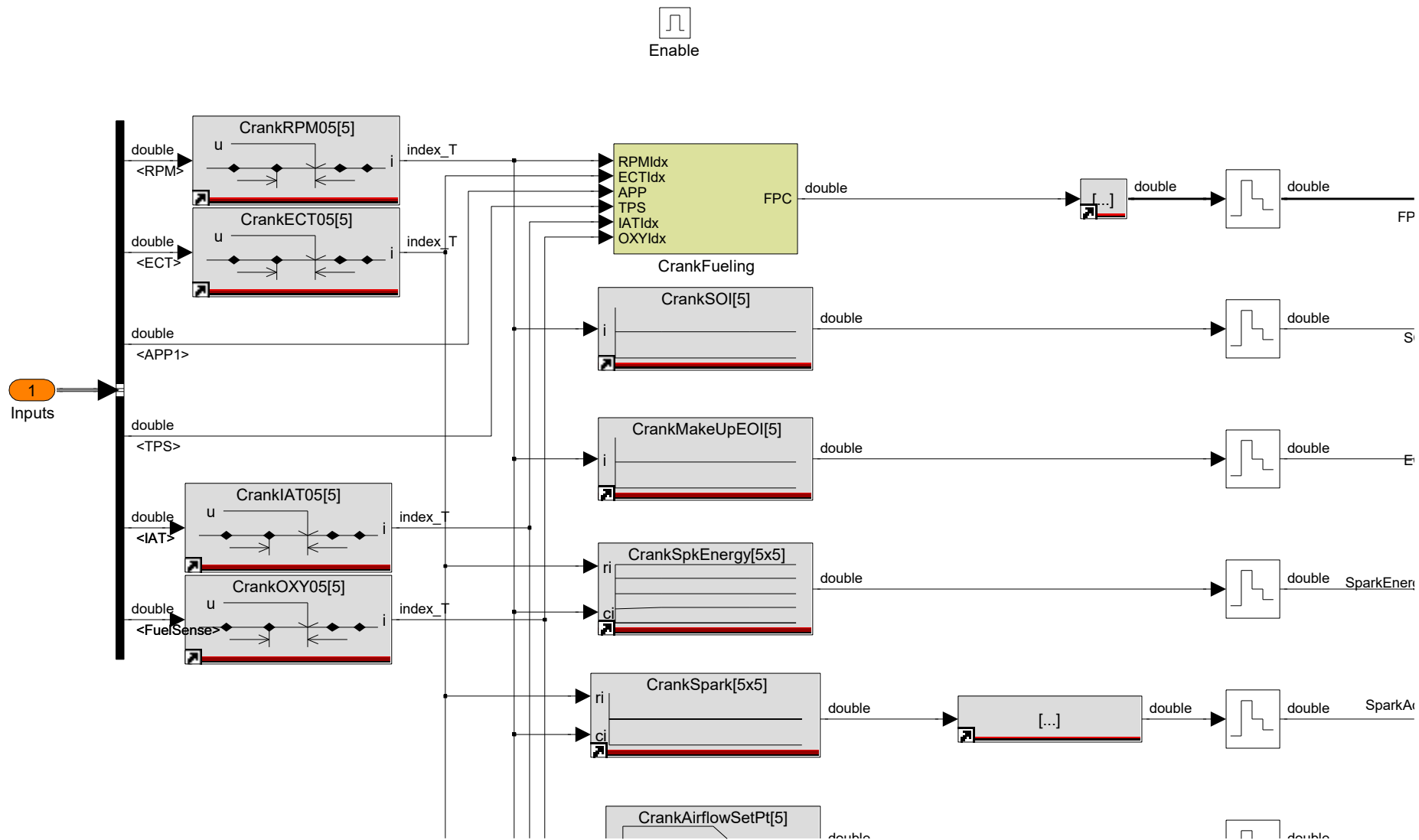


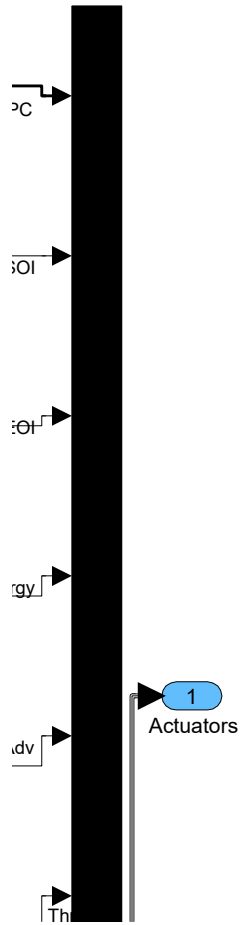


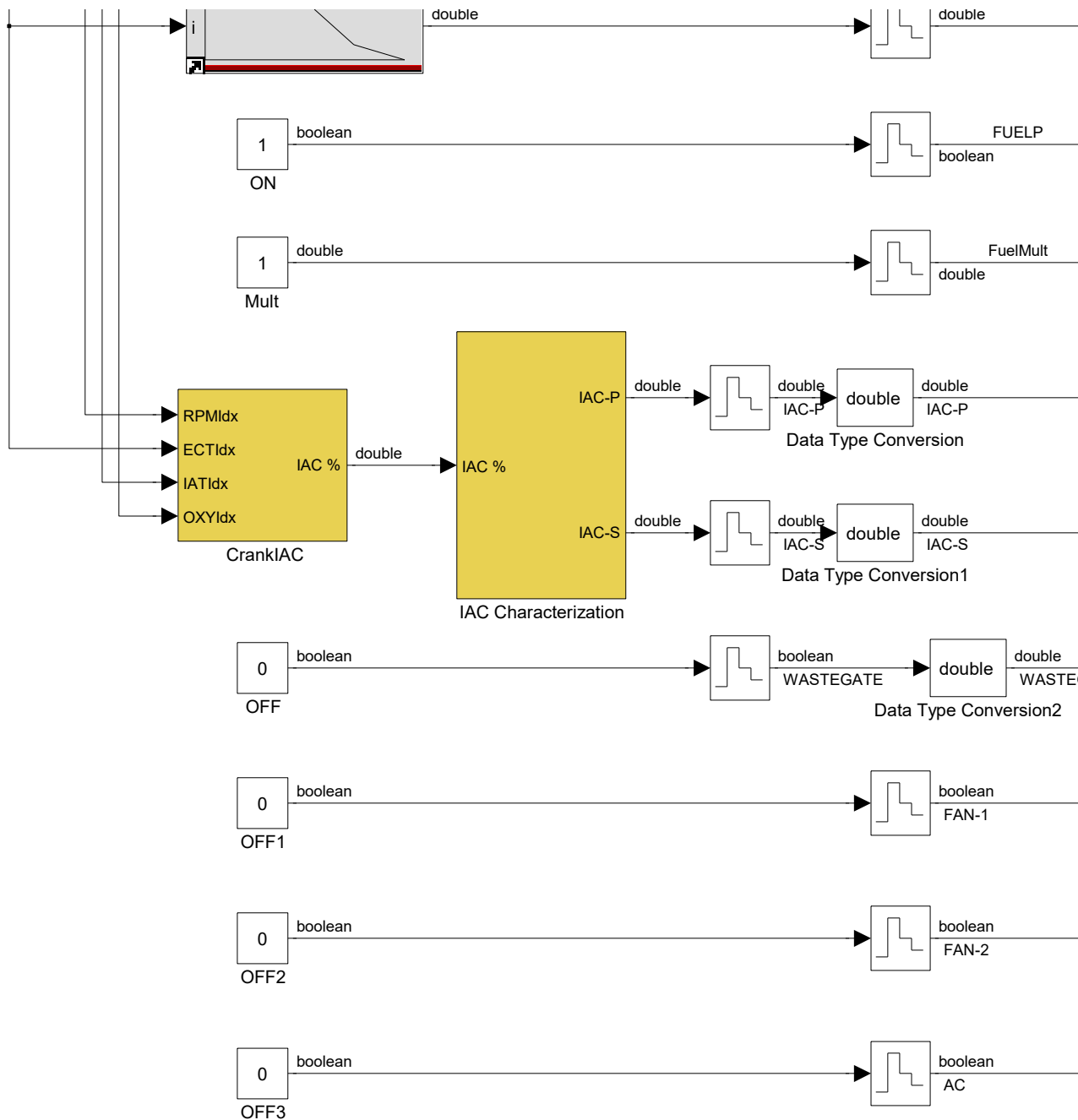


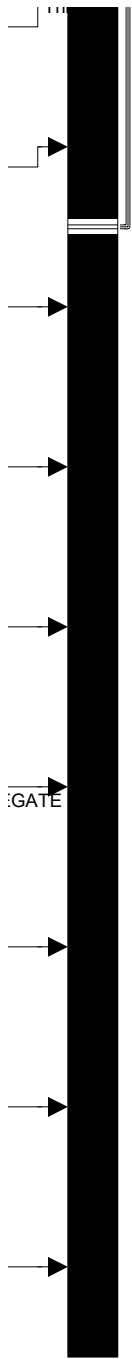


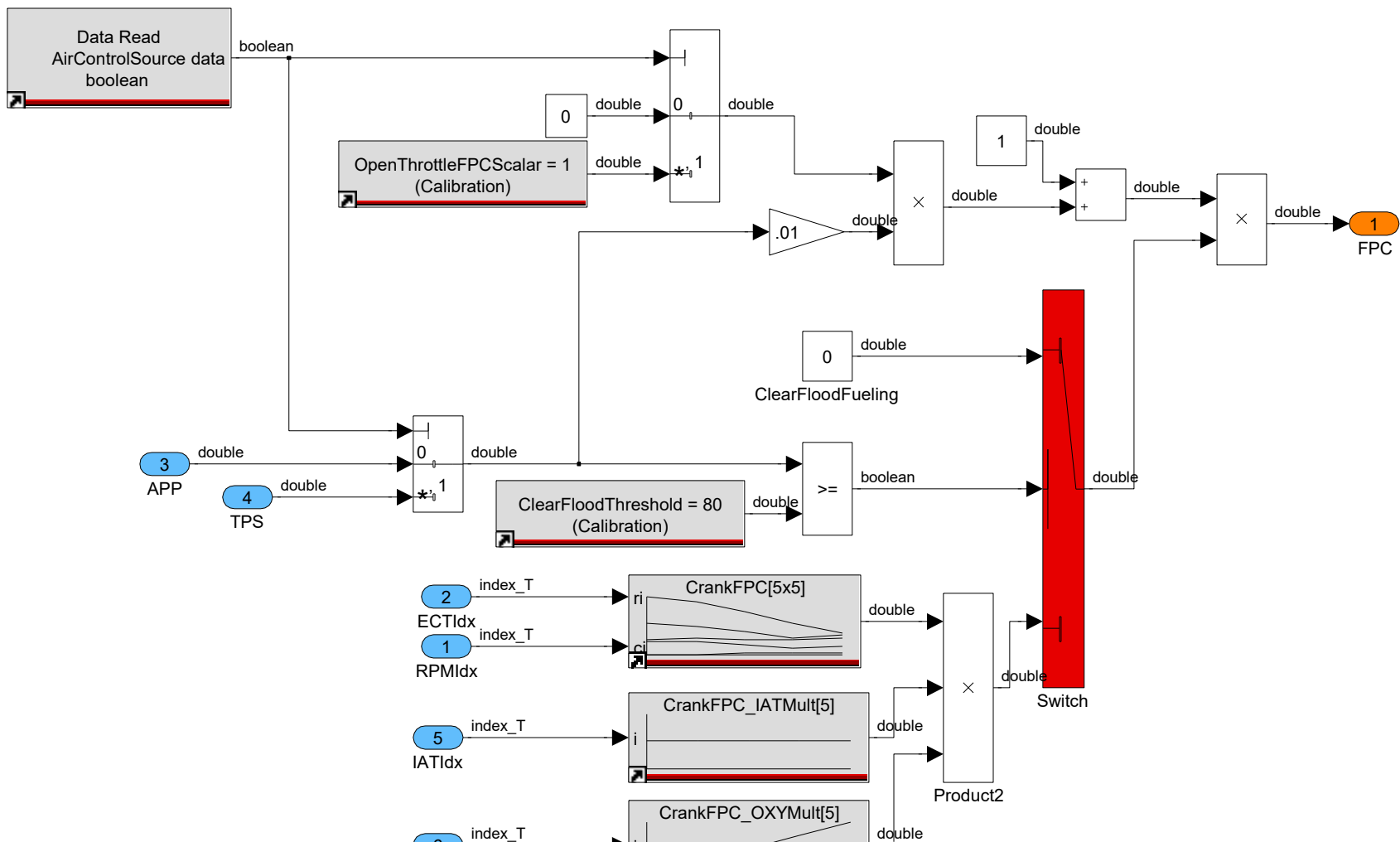






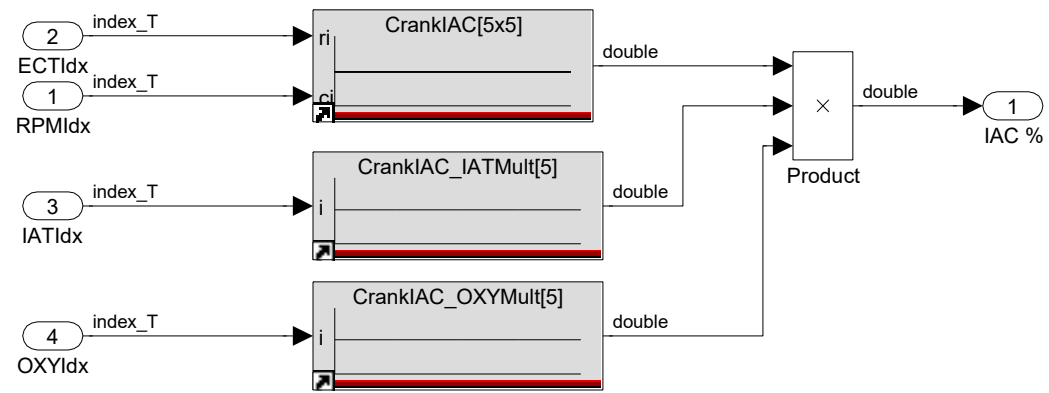


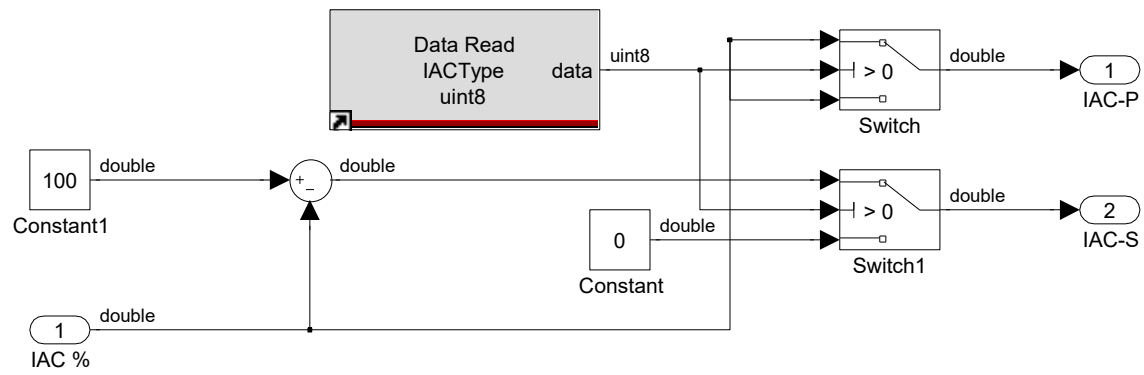


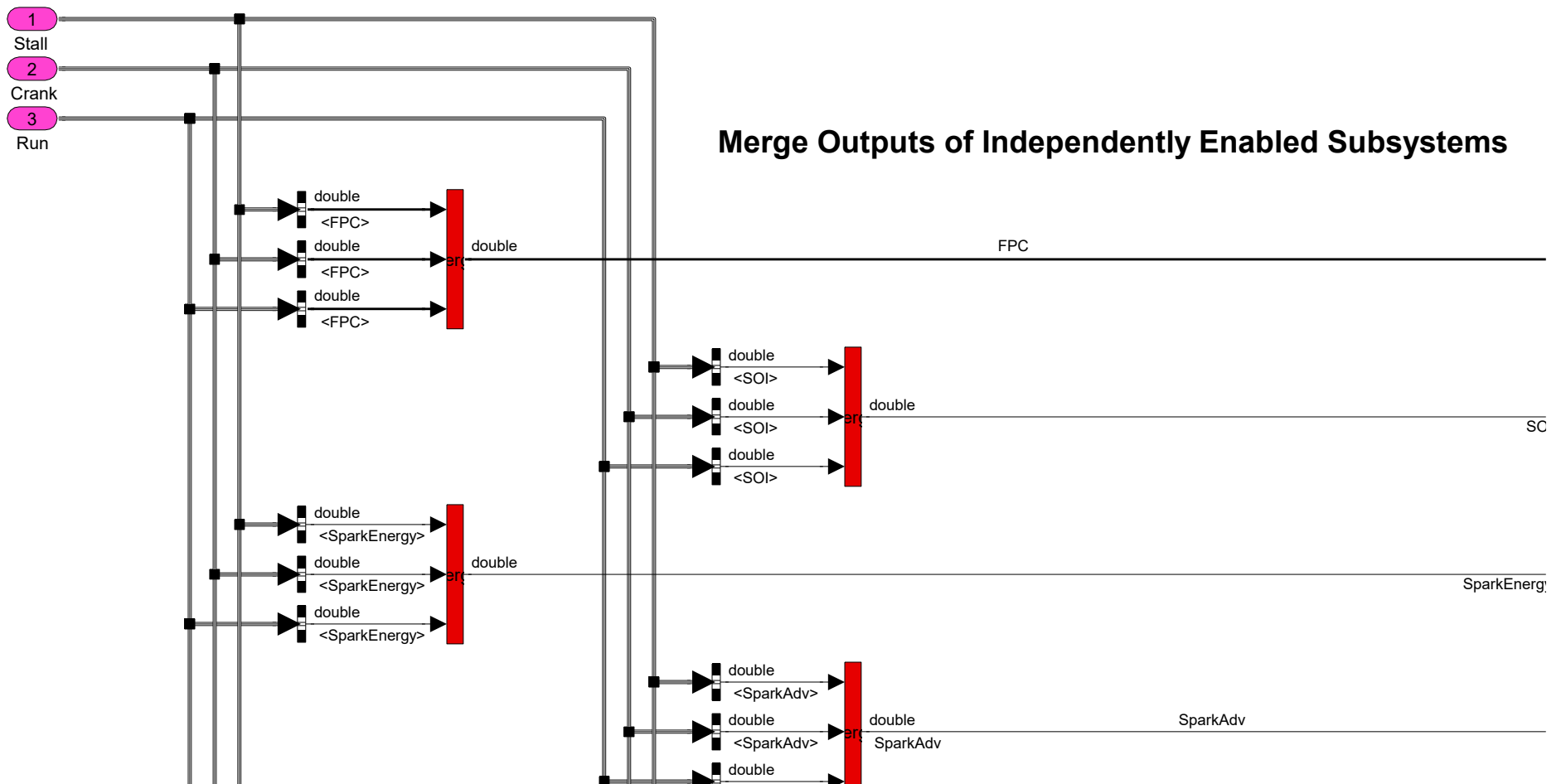




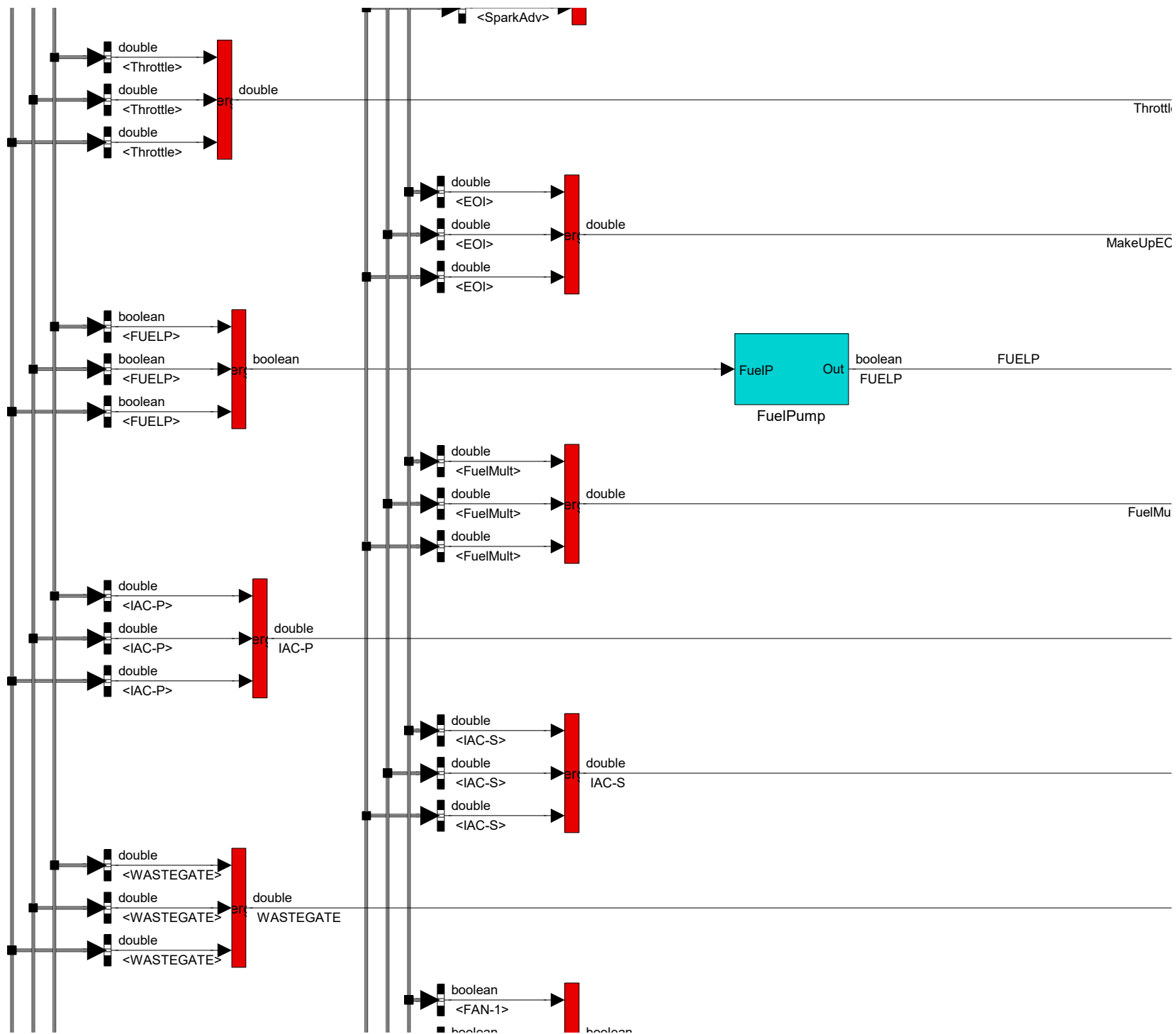


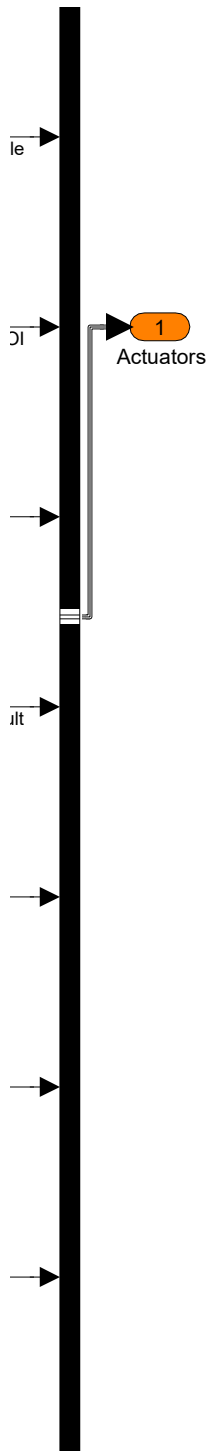


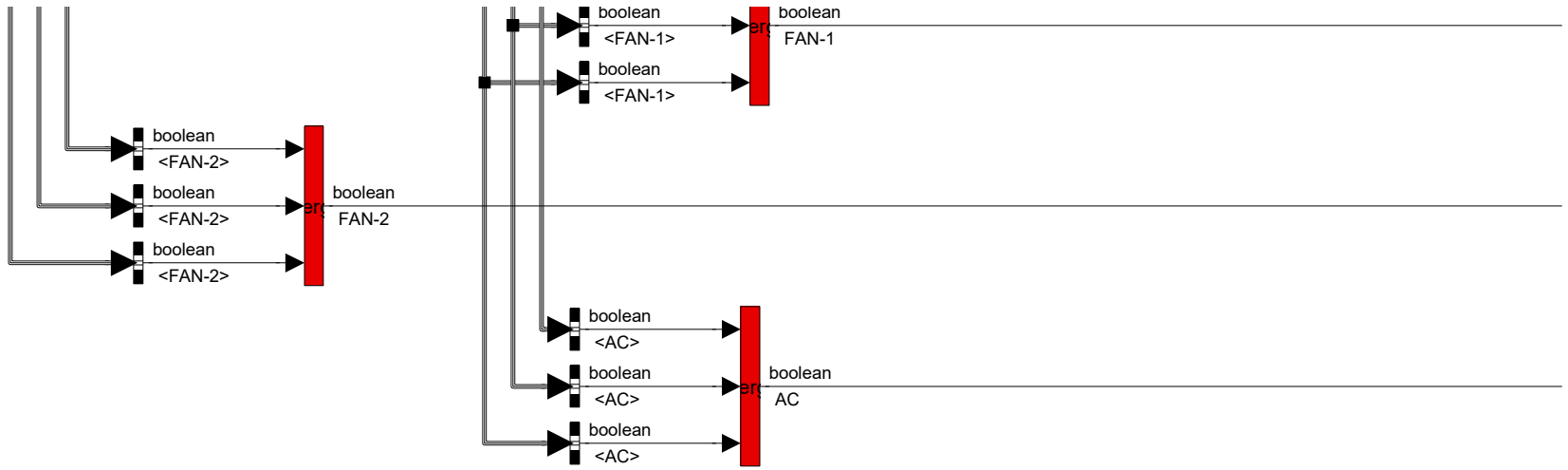


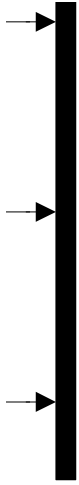




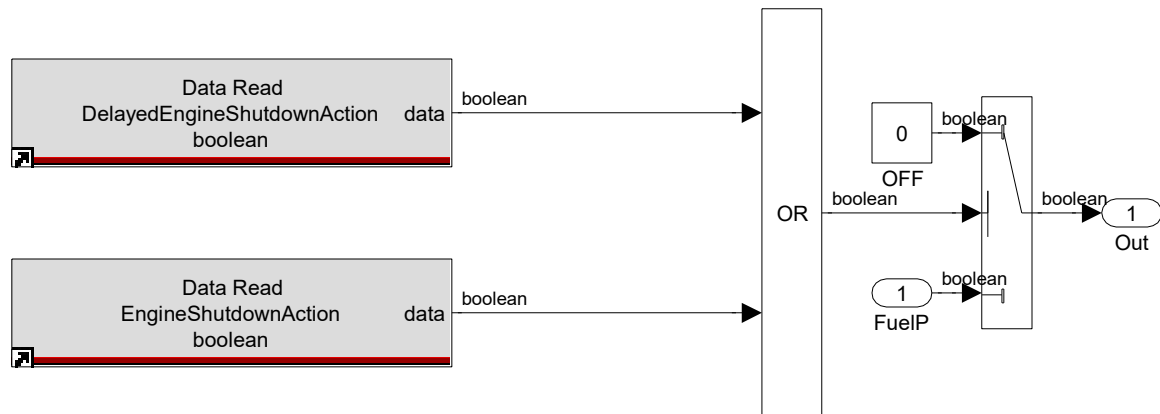












MotoHawk Function Trigger  
FGND\_5XRTI\_PERIODIC  
Priority Order: 1  
(Used in triggered subsystem)

1  
Inputs

boolean  
Enable

Sensors  
EquivRatioInfo  
Base Equiv Ratio

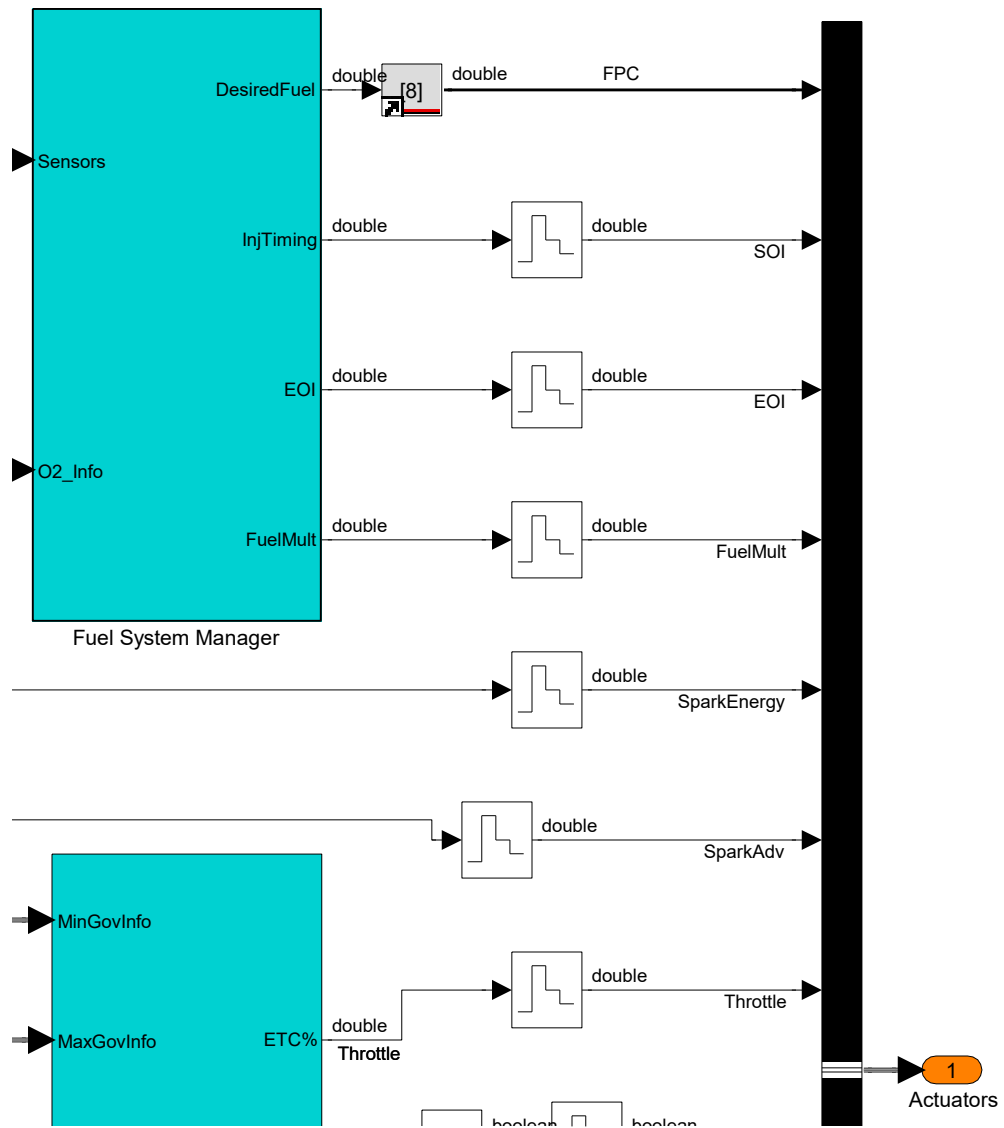
EquivRatioInfo  
DesEquivRatio  
Sensors  
DFCOInfo  
DFCO

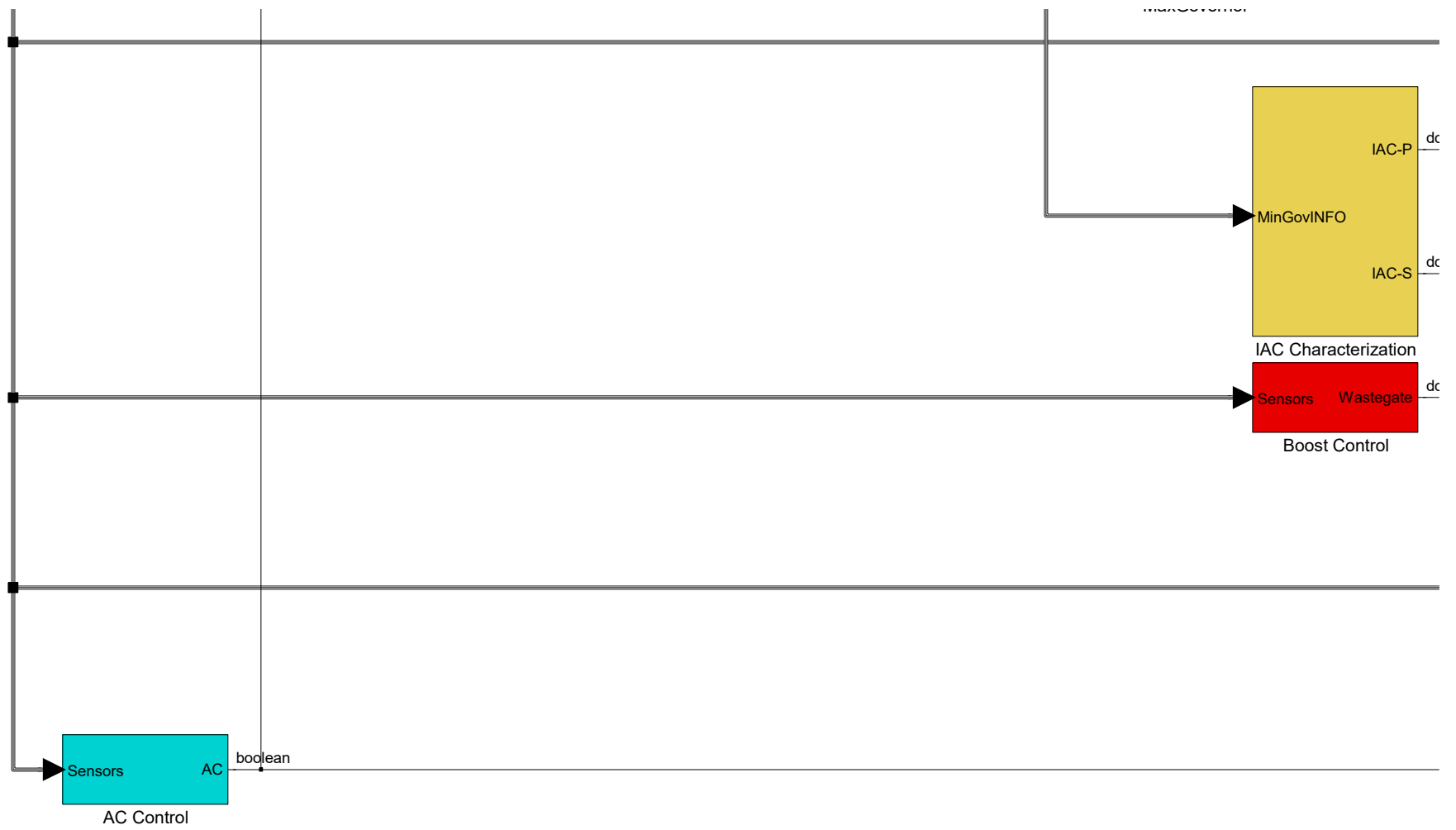
Reset  
Sensors  
O2\_Info  
Desired\_Equiv  
O2 Control

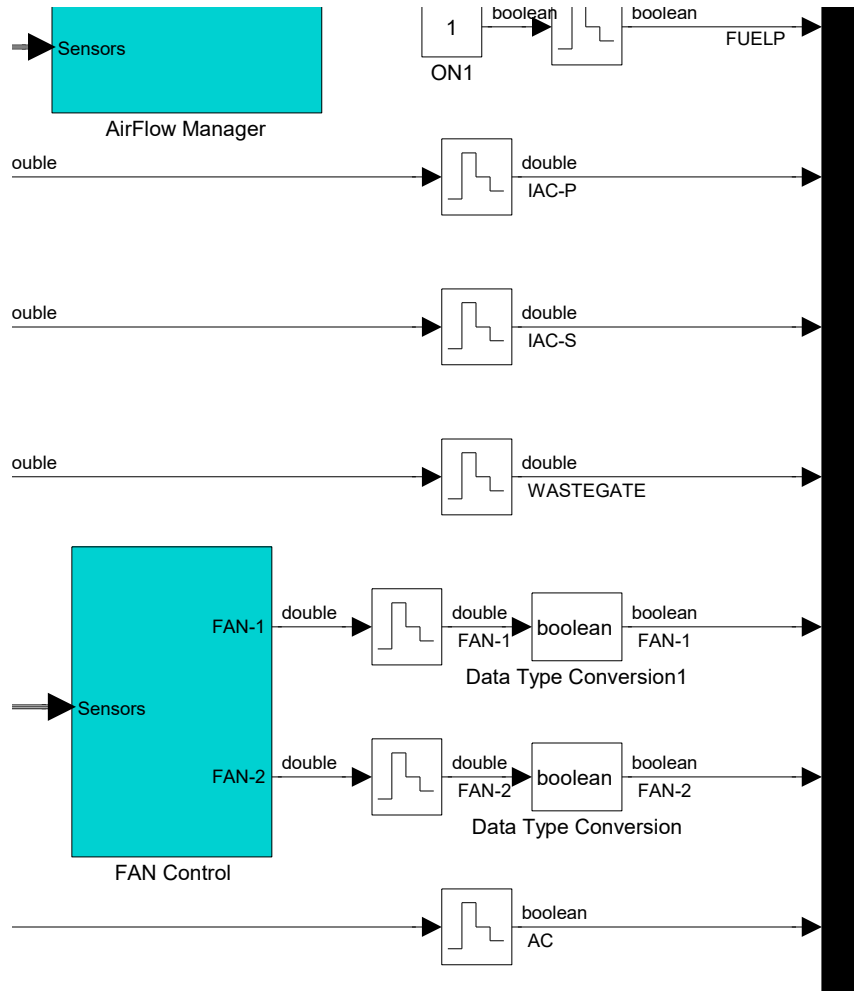
Sensors  
Trigger()  
MinGovInfo  
AC  
MinGovernor

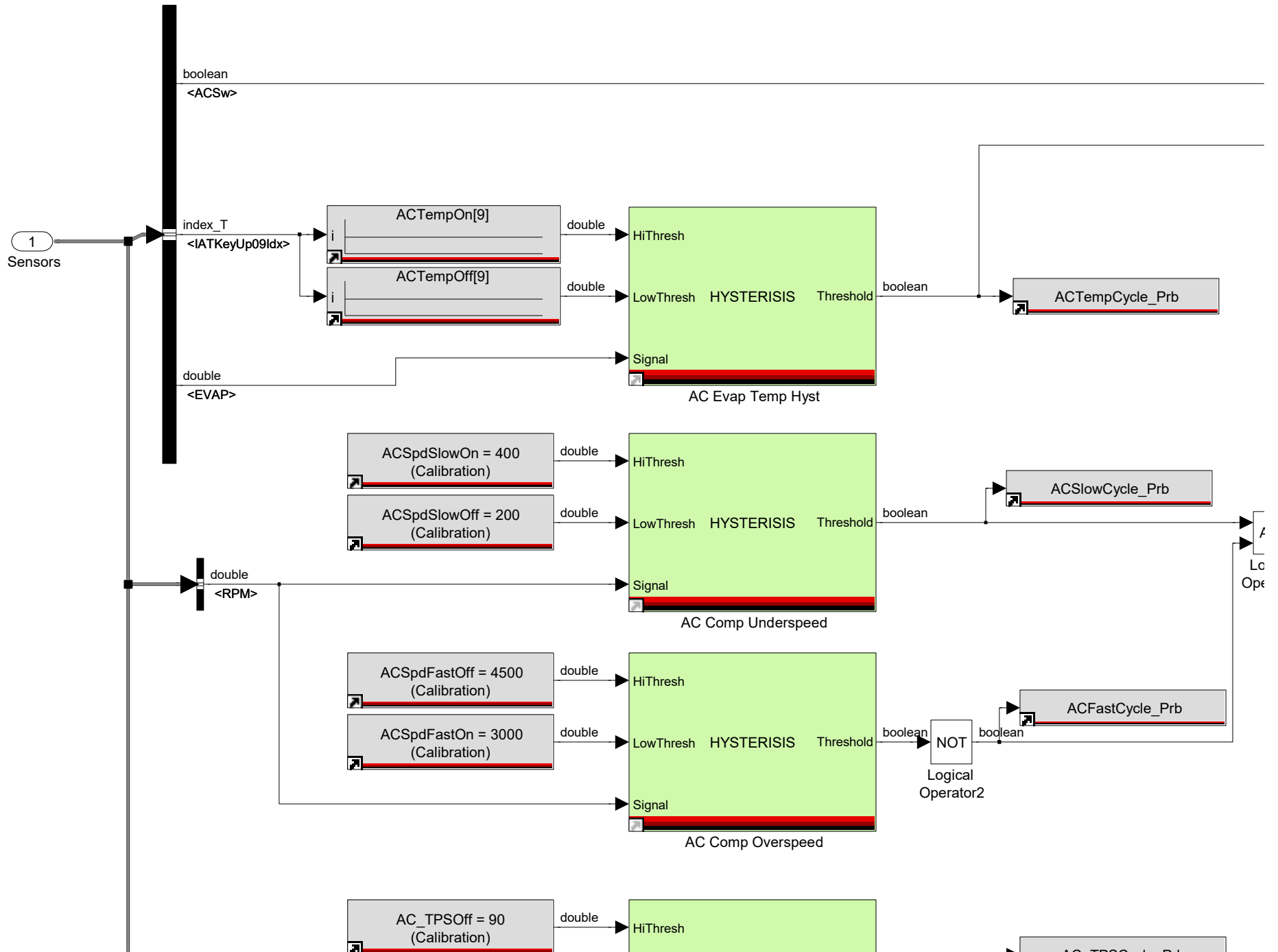
Trigger()  
DesEquivRatio  
Sensors  
SparkEnergy  
RPM\_Ctrl\_Ofst  
SpkAdv  
DFCOInfo  
Spark Manager

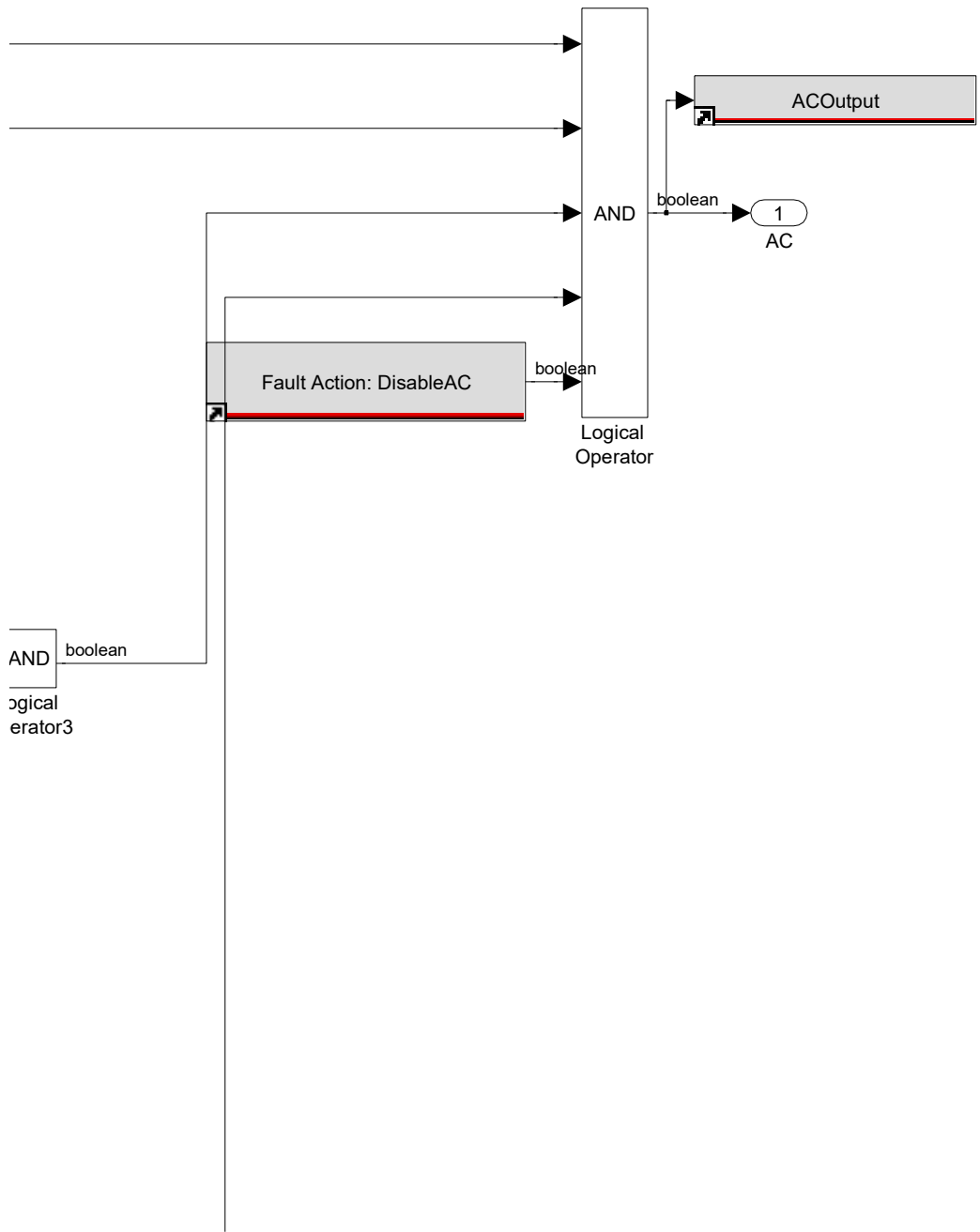
Trigger()  
MinGovInfo  
MaxGovInfo  
Sensors  
MaxGovernor

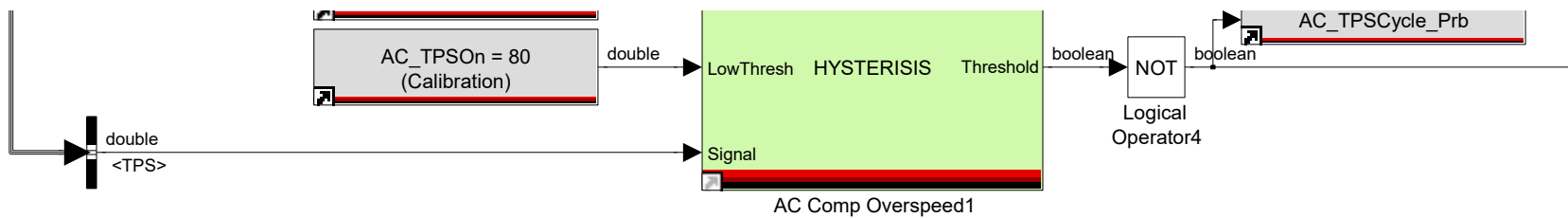






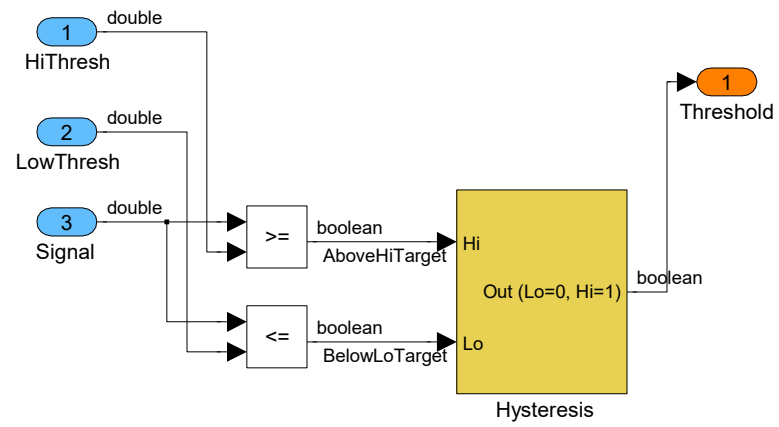




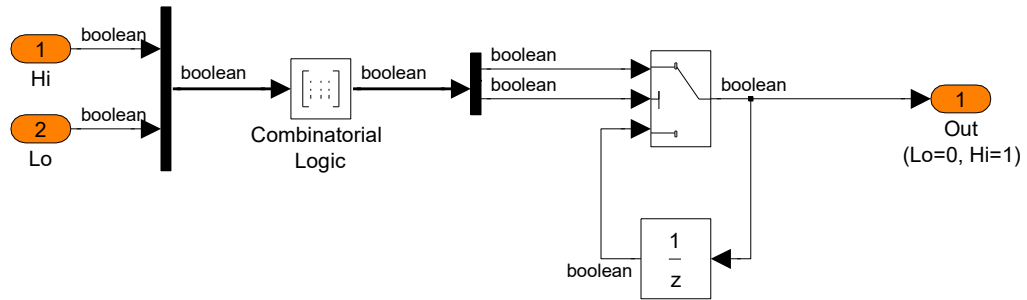


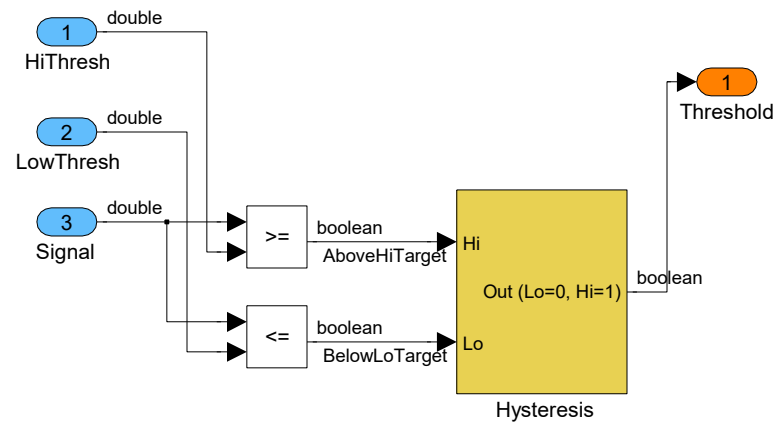




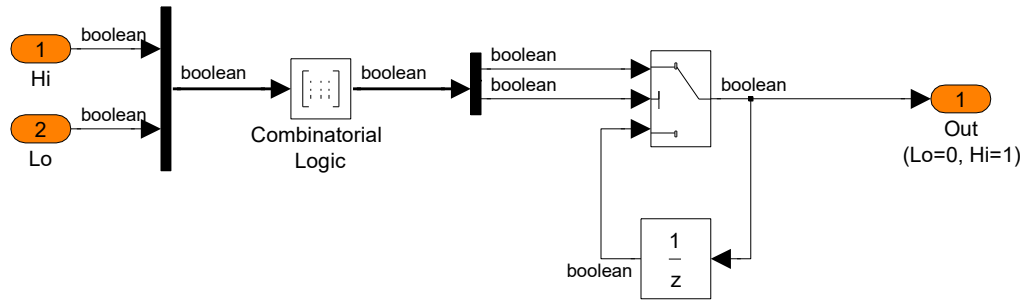


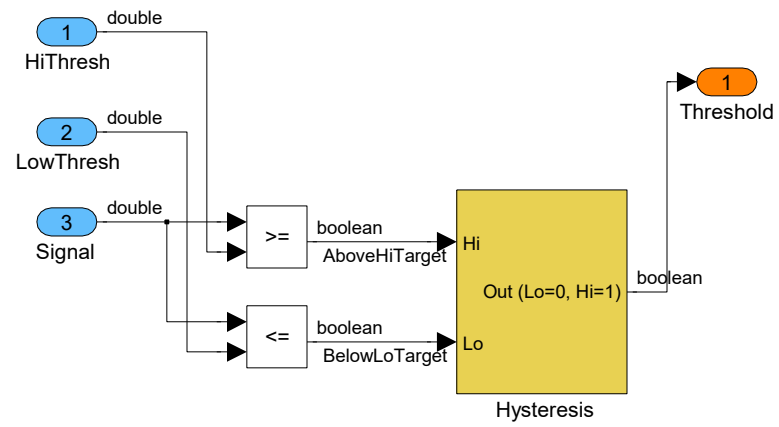
**Hi & !Lo => Hi**  
**Lo & !Hi => Lo**  
**Otherwise, Hold**



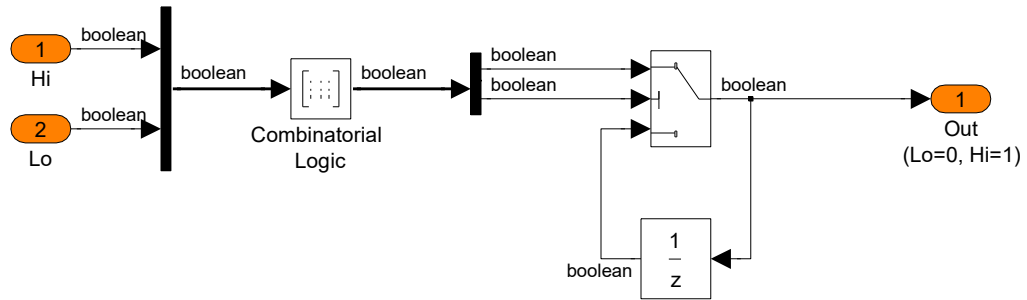


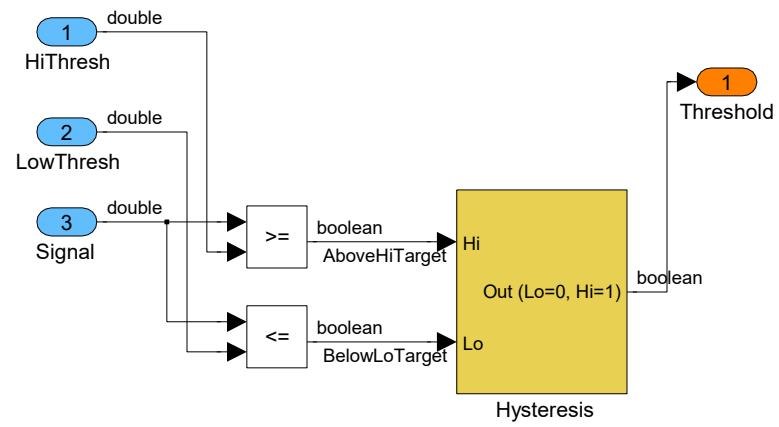
**Hi & !Lo => Hi**  
**Lo & !Hi => Lo**  
**Otherwise, Hold**





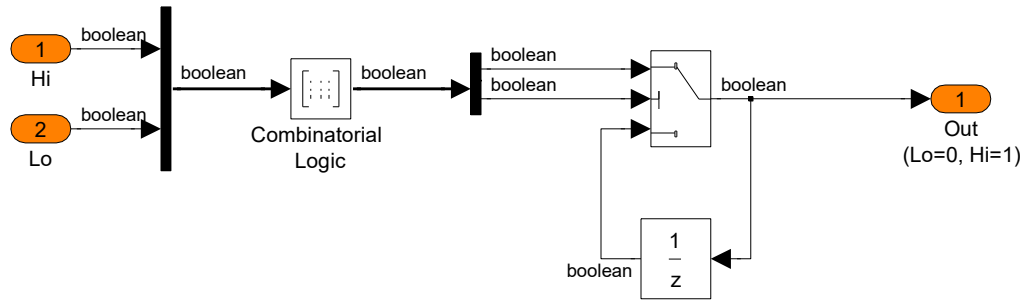
**Hi & !Lo => Hi**  
**Lo & !Hi => Lo**  
**Otherwise, Hold**



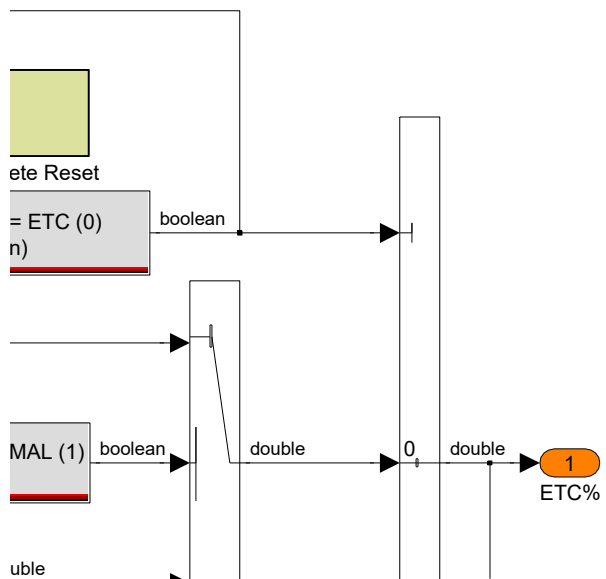




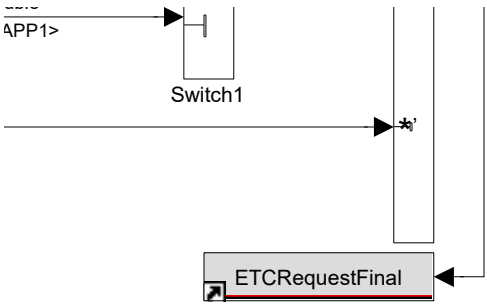
**Hi & !Lo => Hi**  
**Lo & !Hi => Lo**  
**Otherwise, Hold**













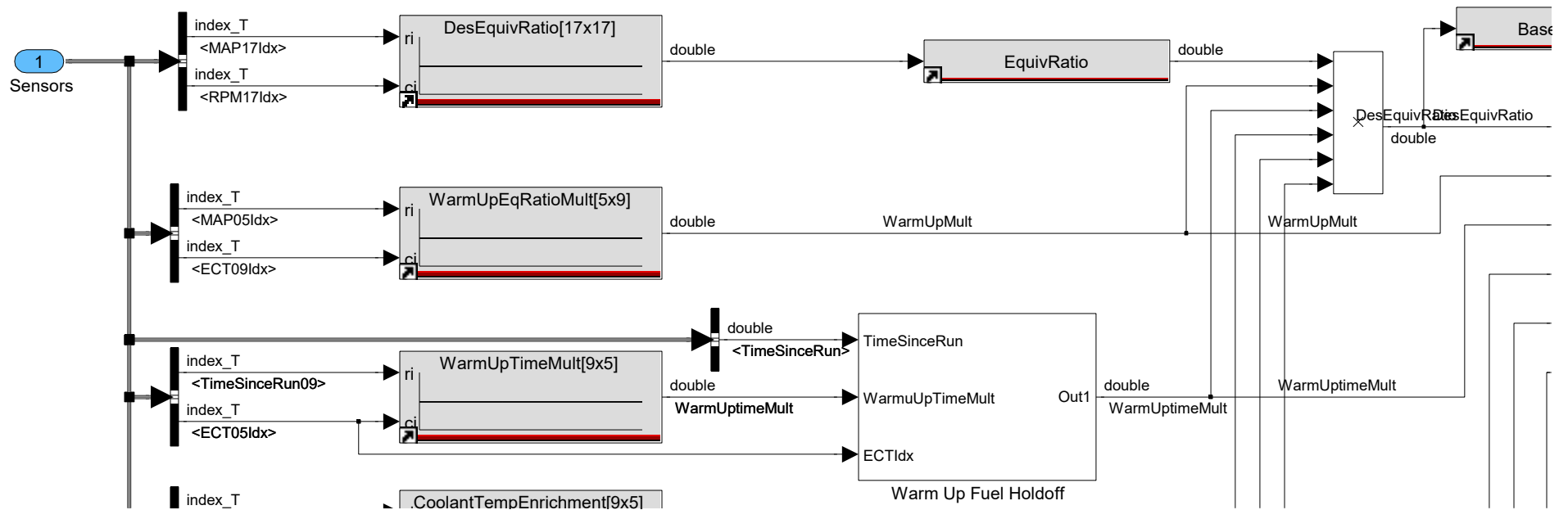
boolean(1)

boolean

data

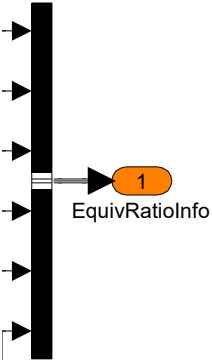
Data W  
ShutDownTask  
boolea

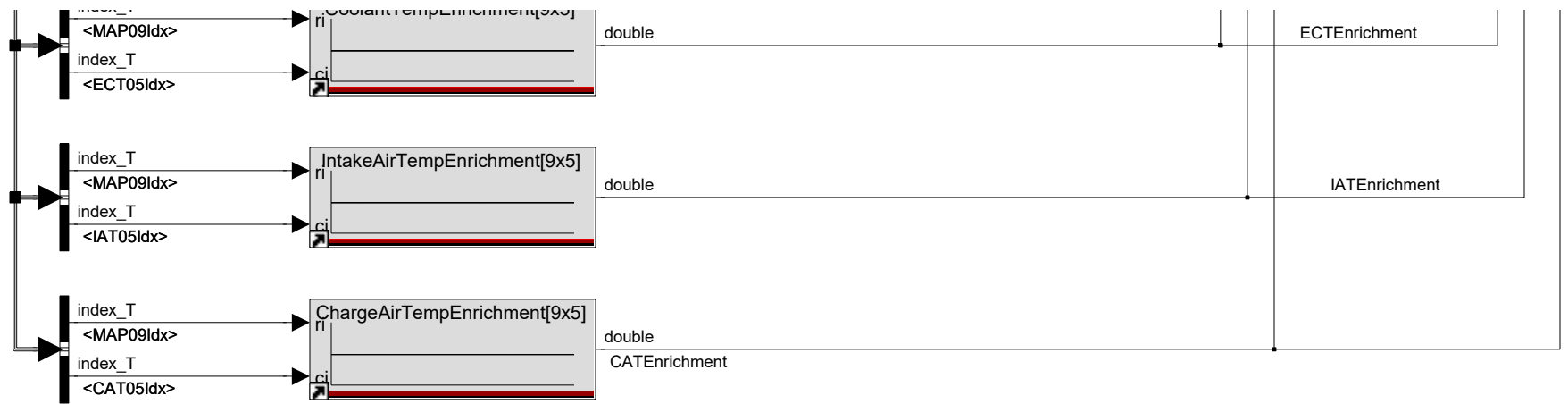
/rite  
ksComplete  
an



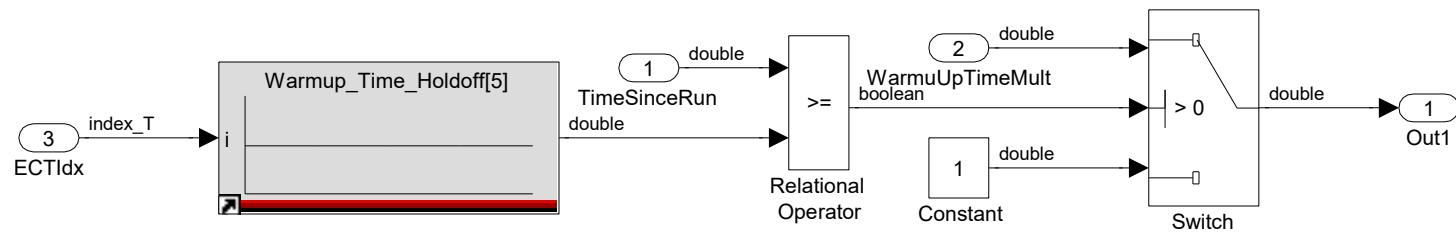


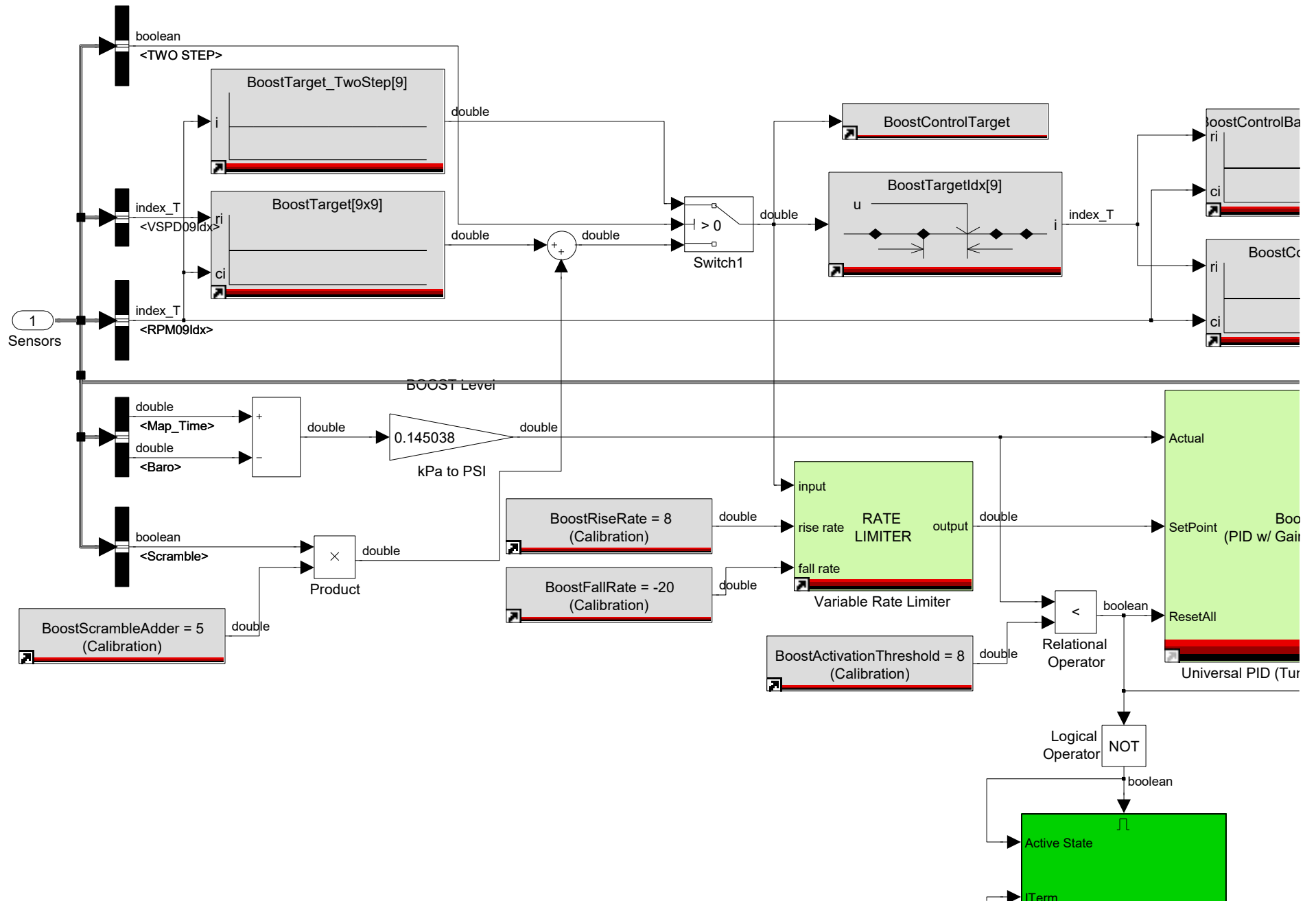
eEquivRatio

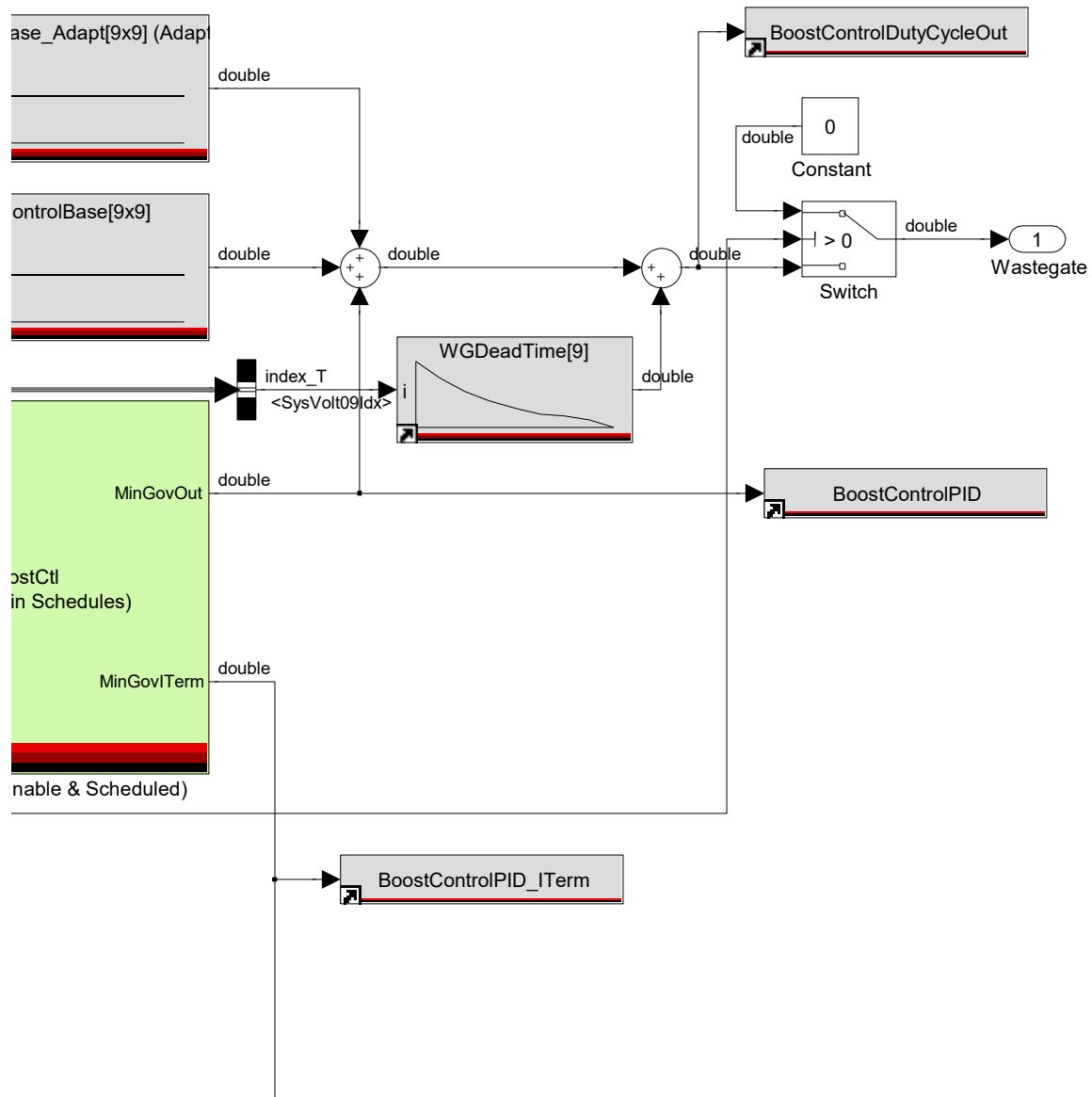


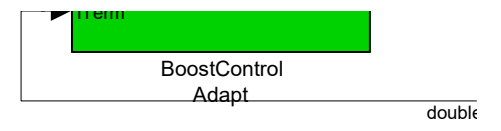


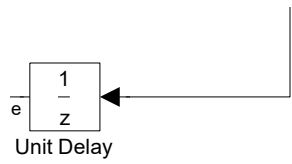




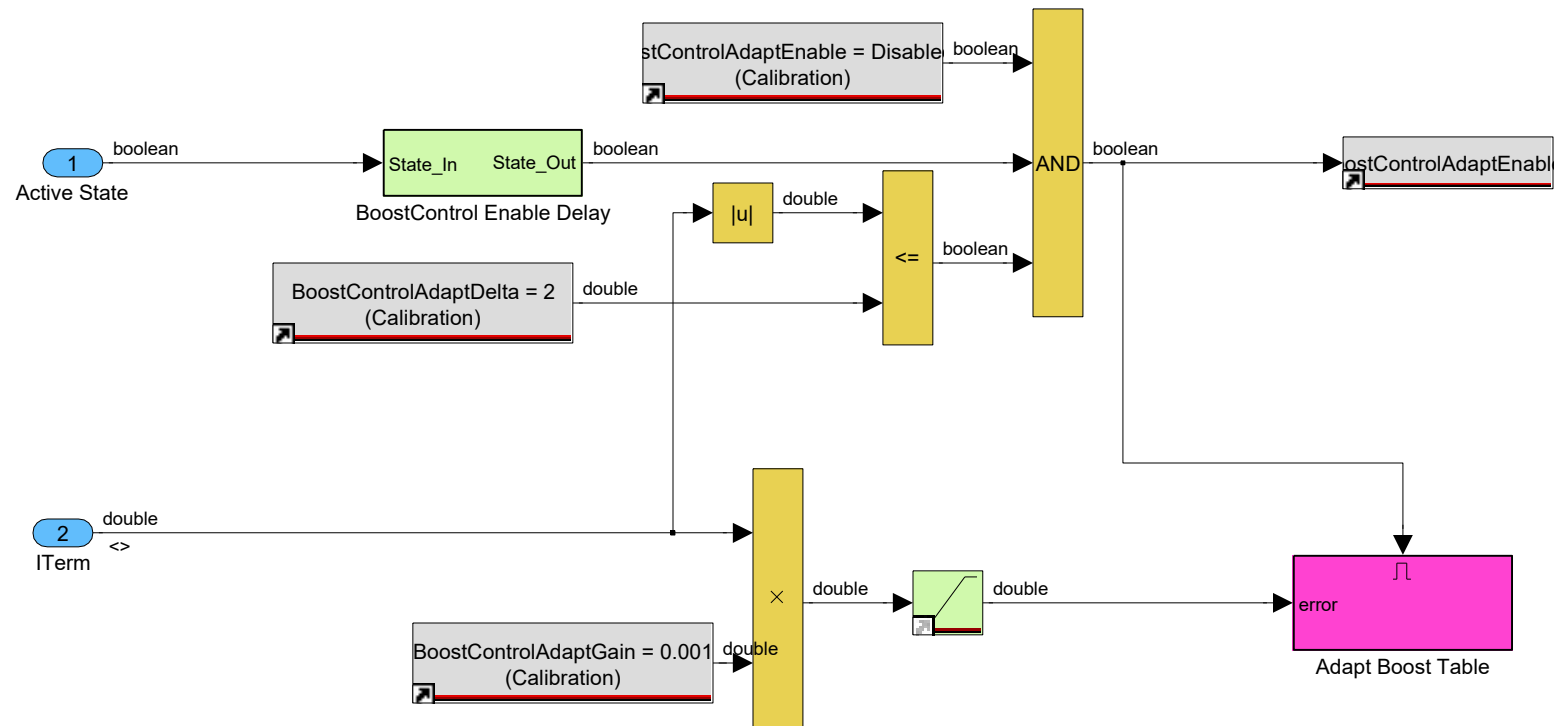


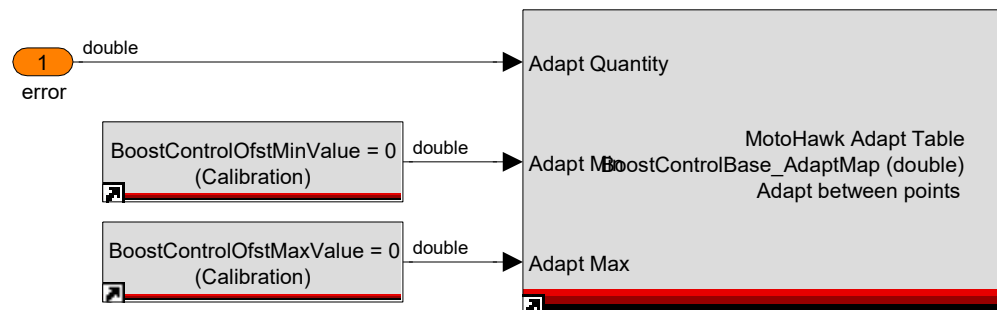














---

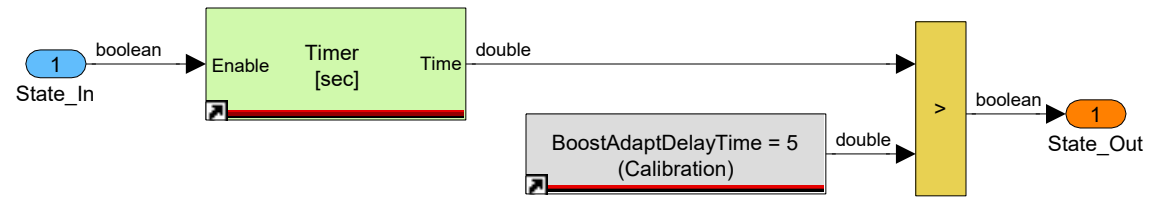
## ***Calibrations***

BaseAirflowOfstMaxValue %	Maximum value that can be adapted into the BaseAirflowOfst table. MotoTune Path:Engine Control   Run   RPM Control   Min Gov   Min Gov Co
BaseAirflowOfstMinValue %	Minimum value that can be adapted into the BaseAirflowOfst table. MotoTune Path:Engine Control   Run   RPM Control   Min Gov   Min Gov Co



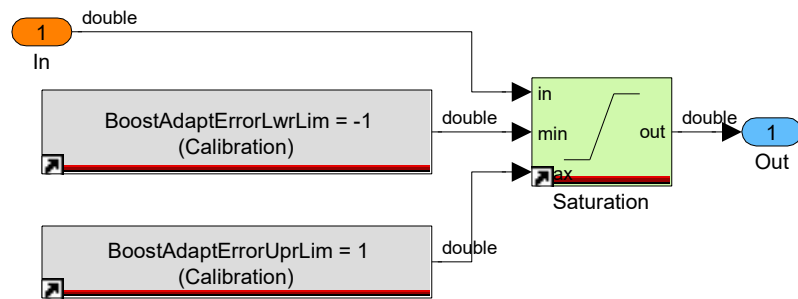


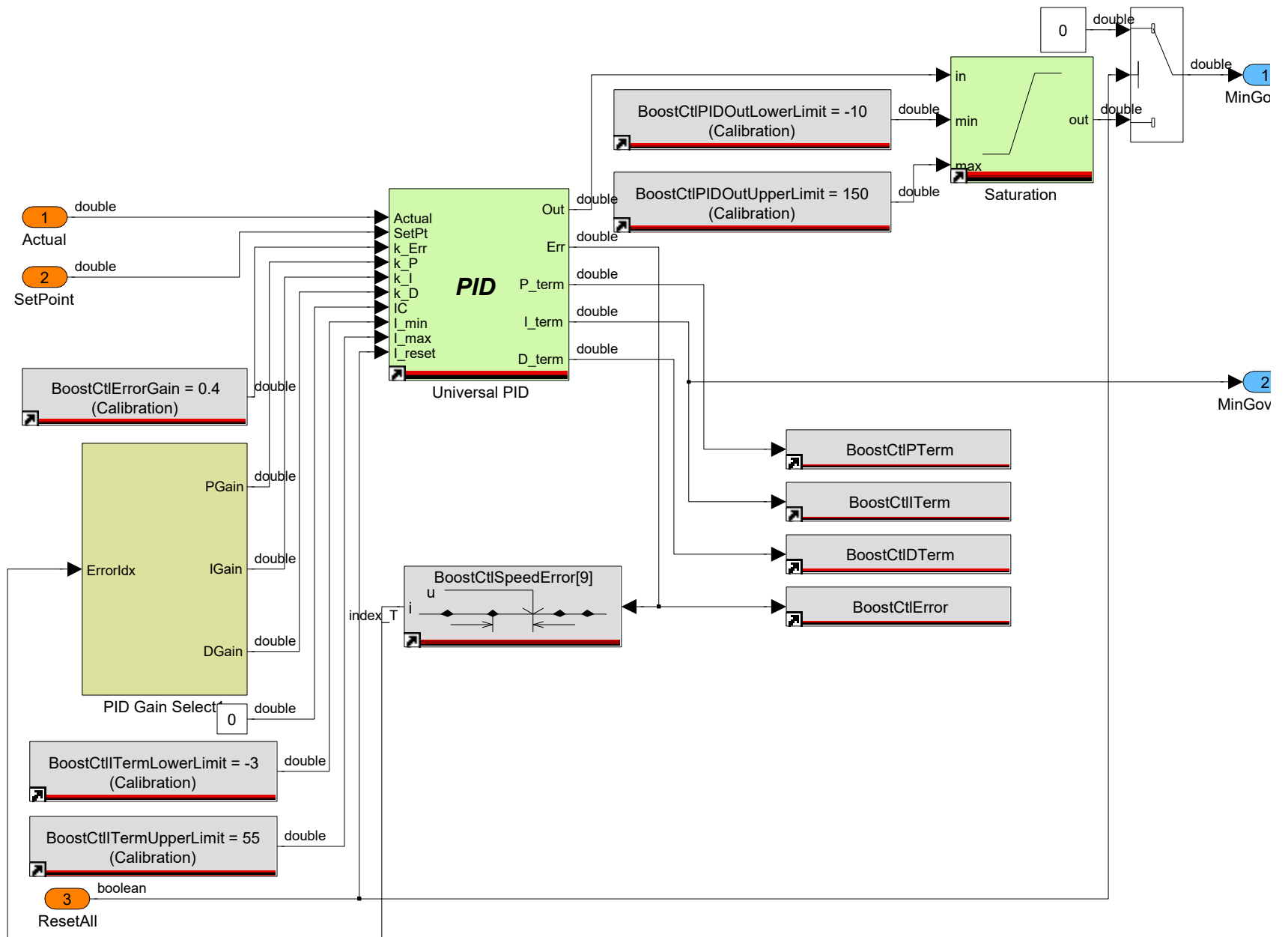






## Calibratable wrapper around the Saturation block

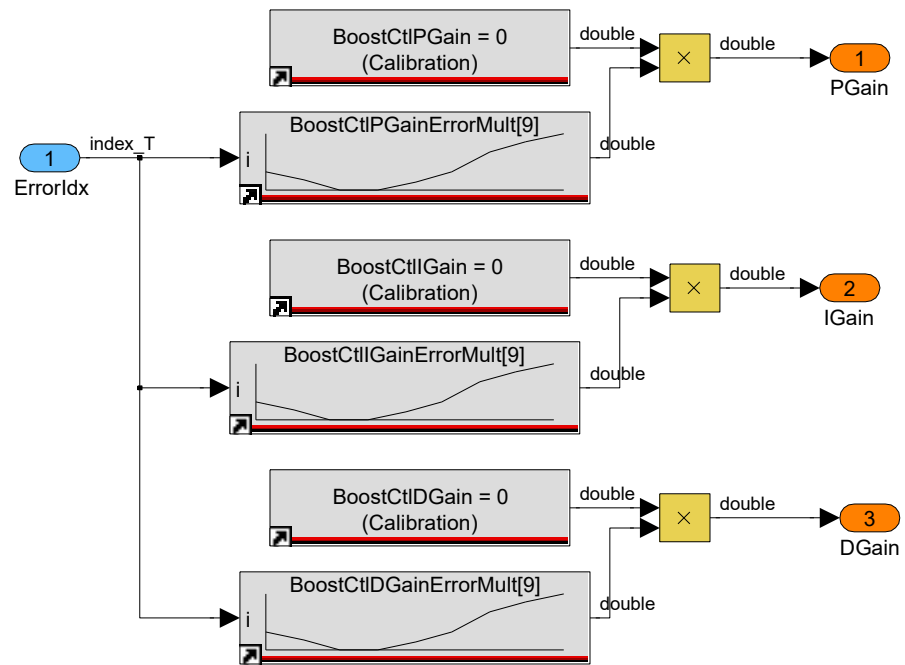


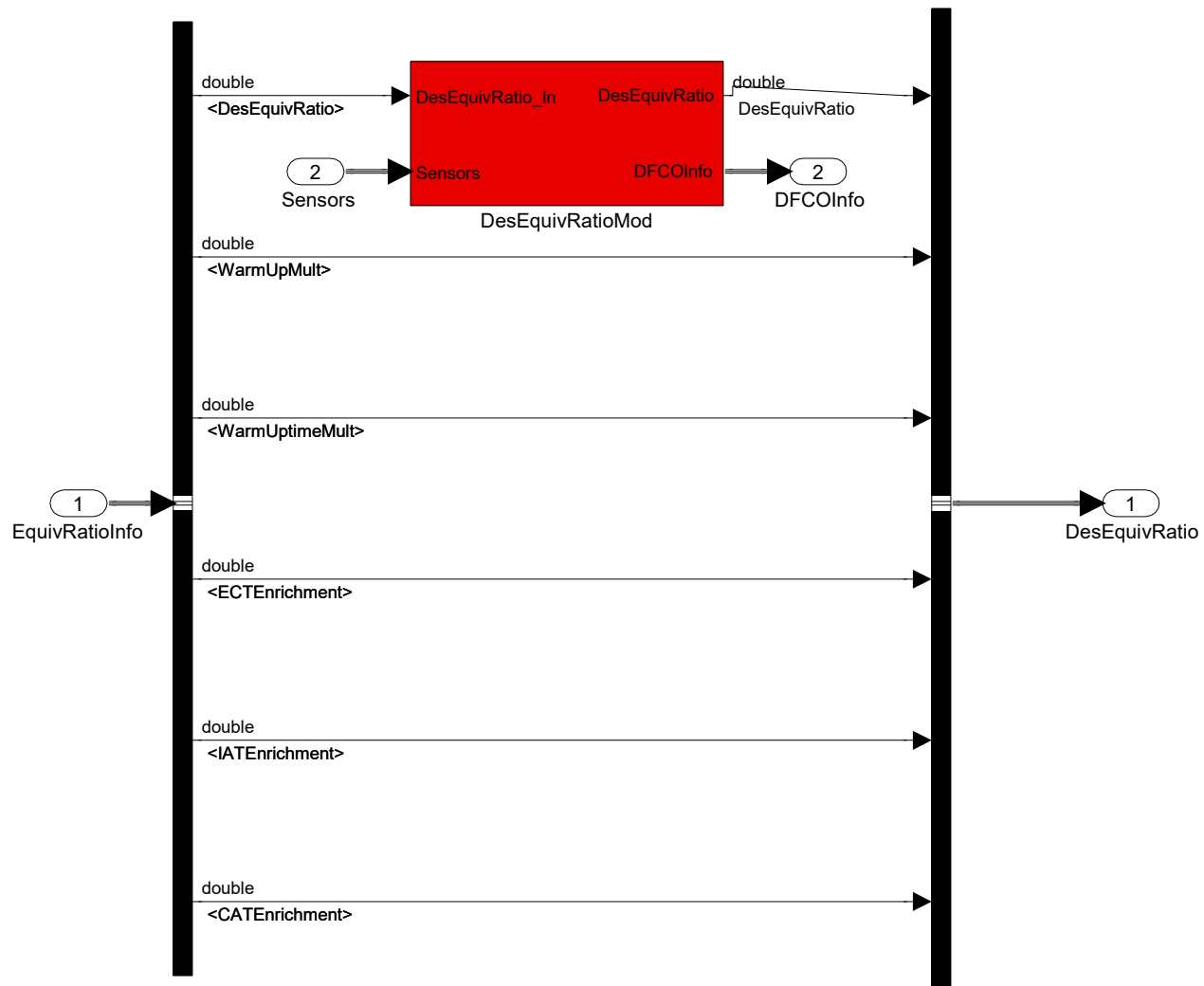


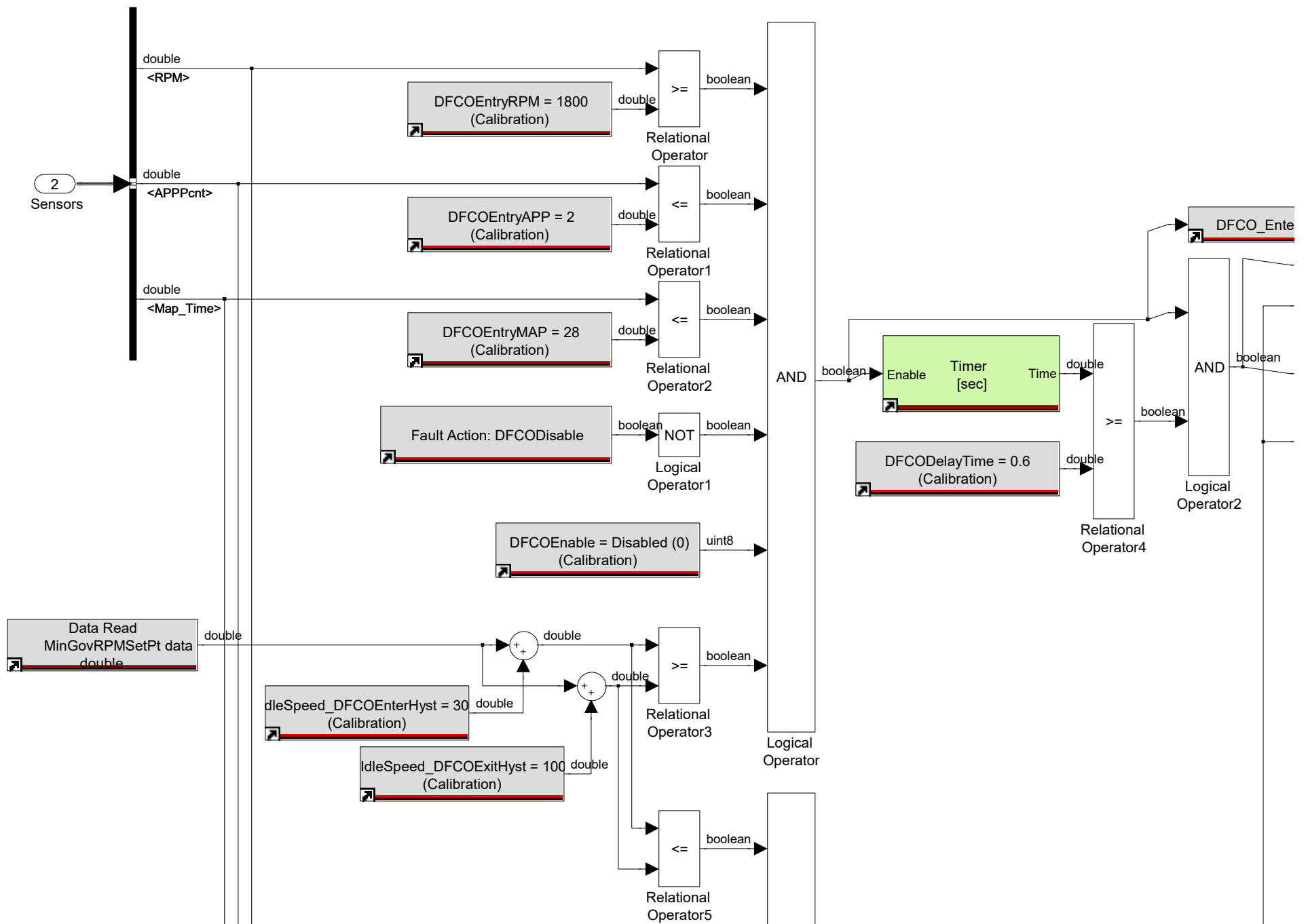




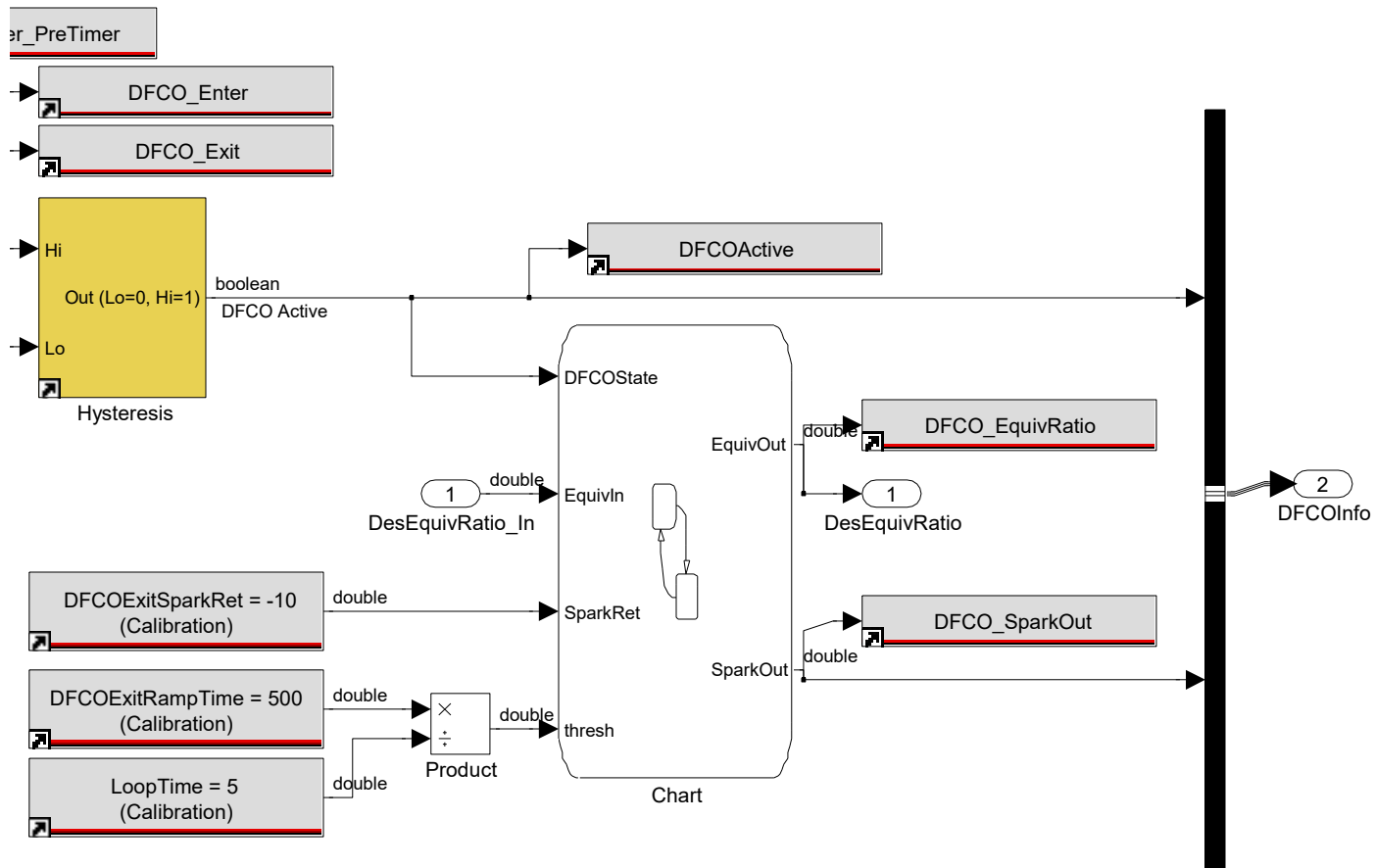


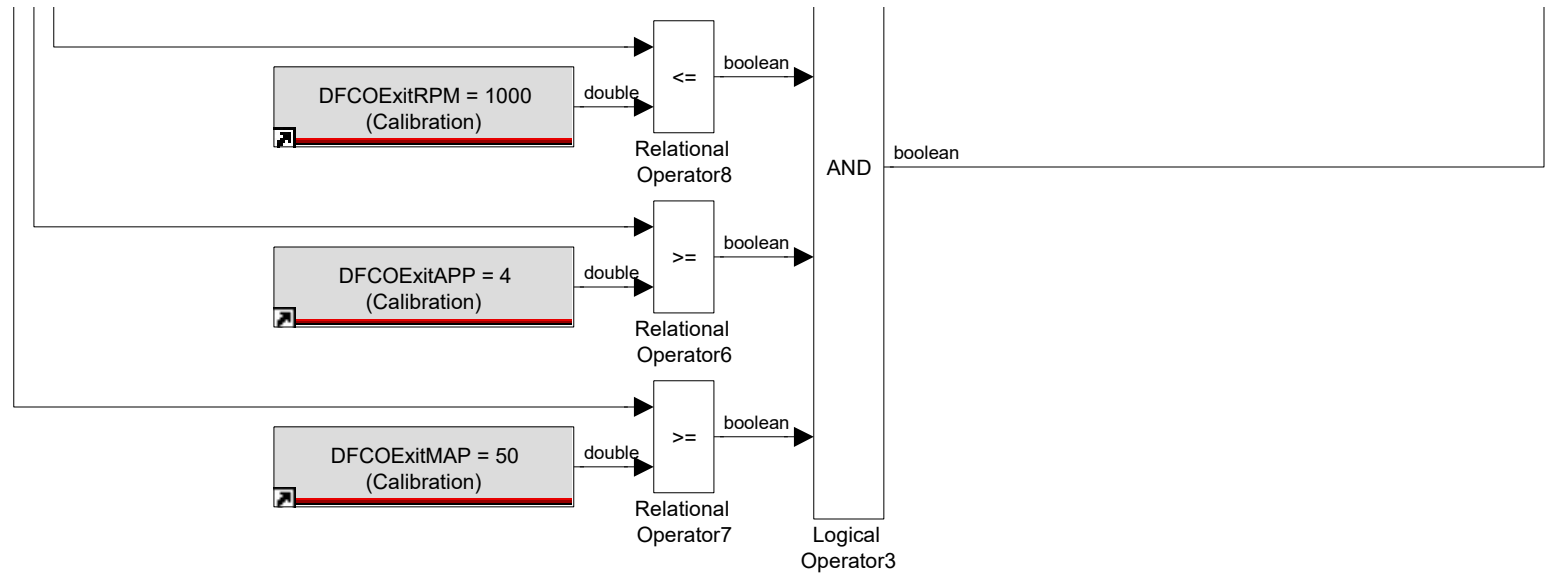




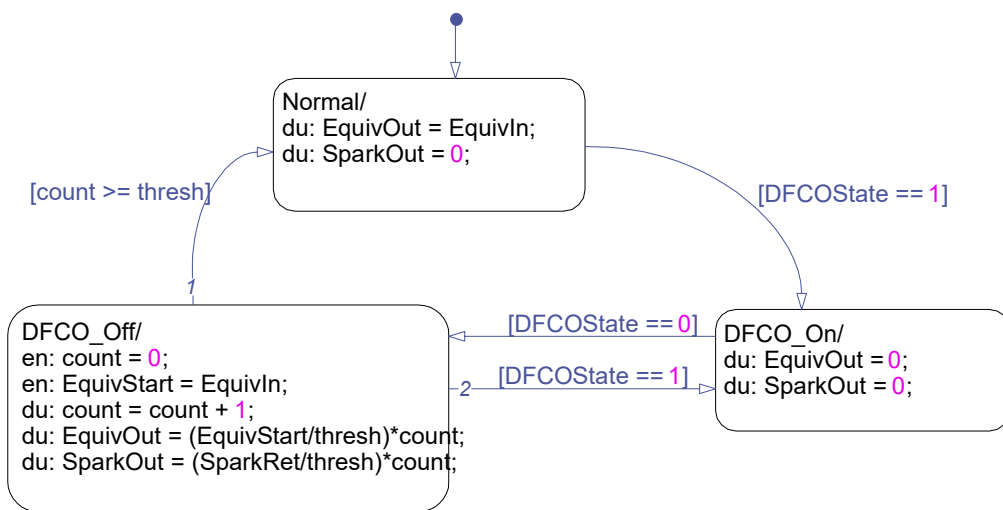


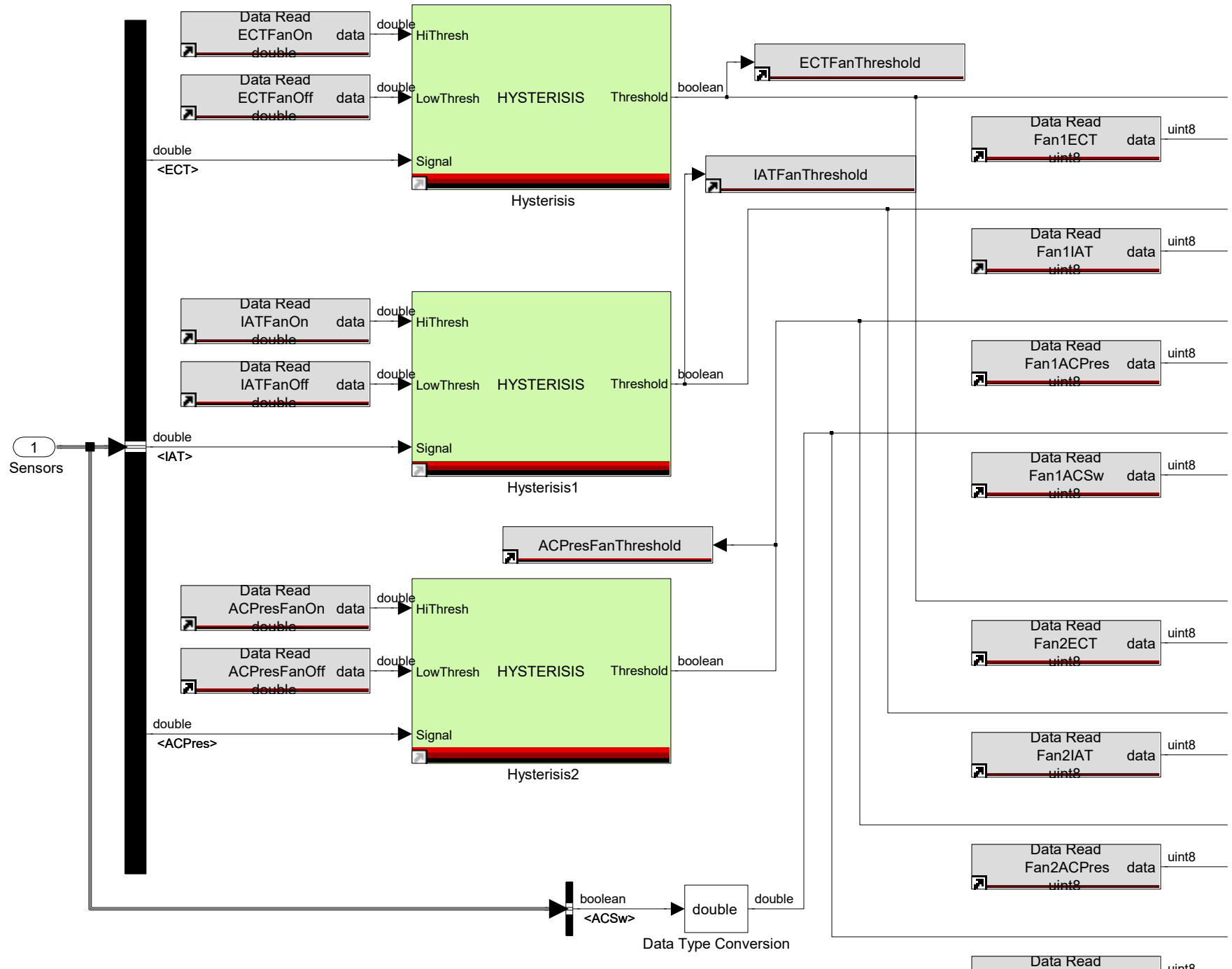


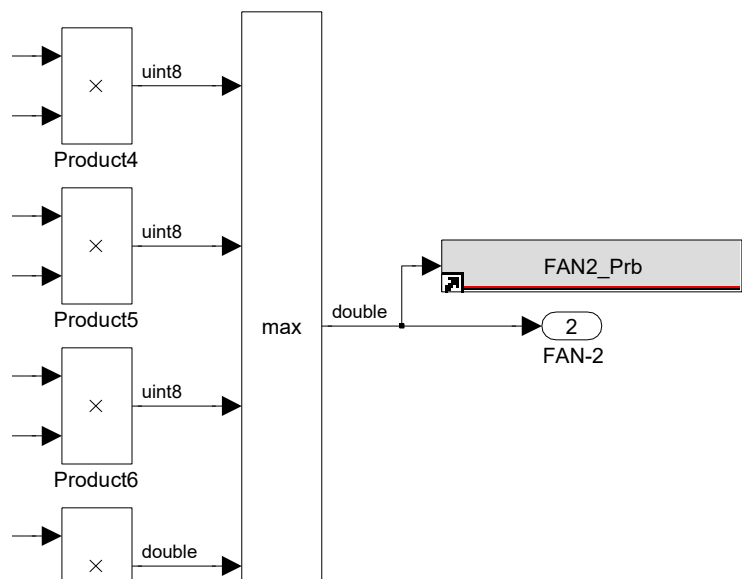
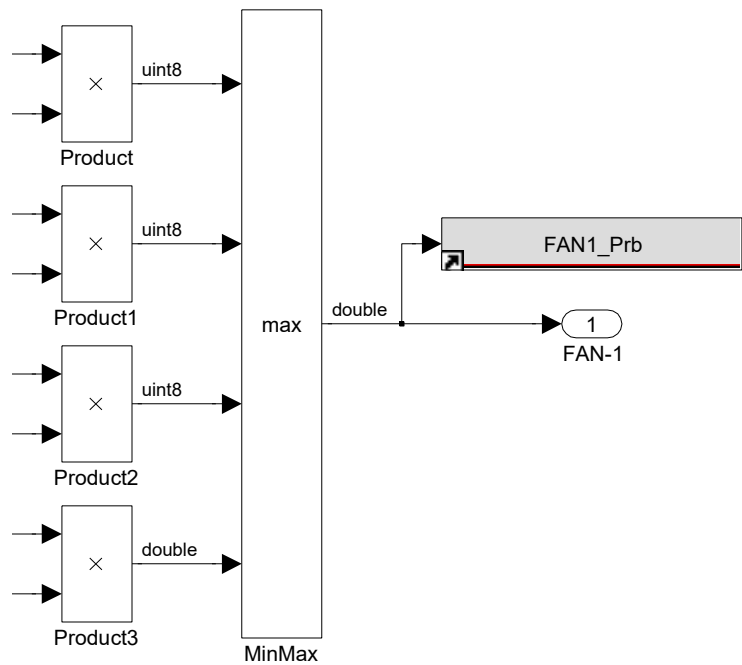


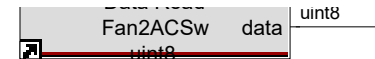


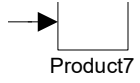




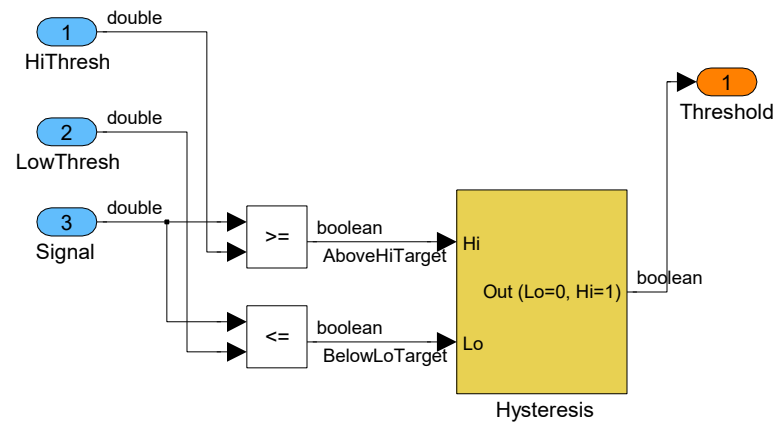




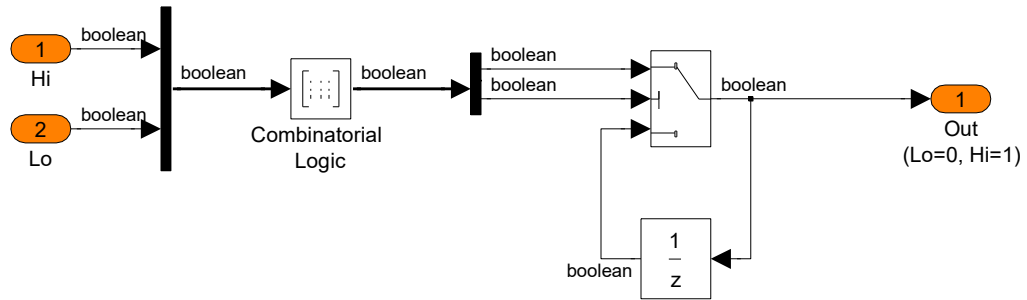


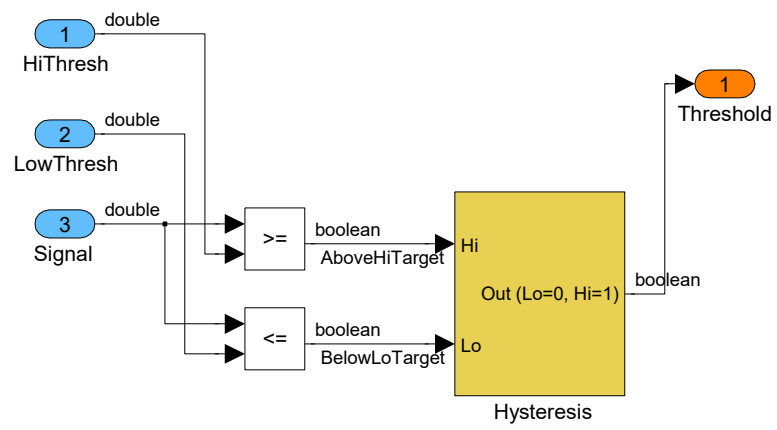




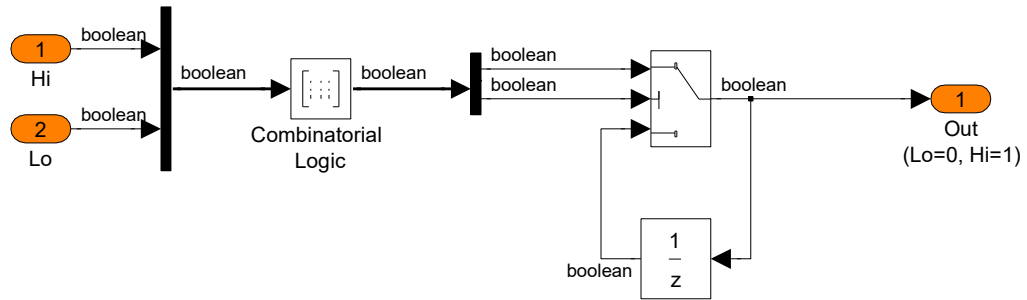


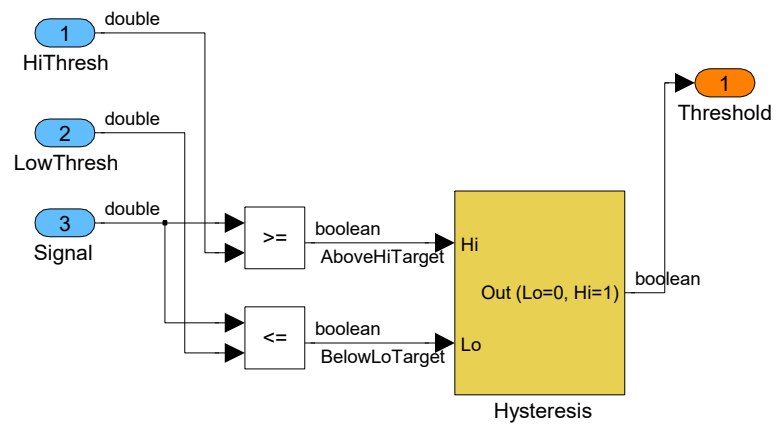
**Hi & !Lo => Hi**  
**Lo & !Hi => Lo**  
**Otherwise, Hold**



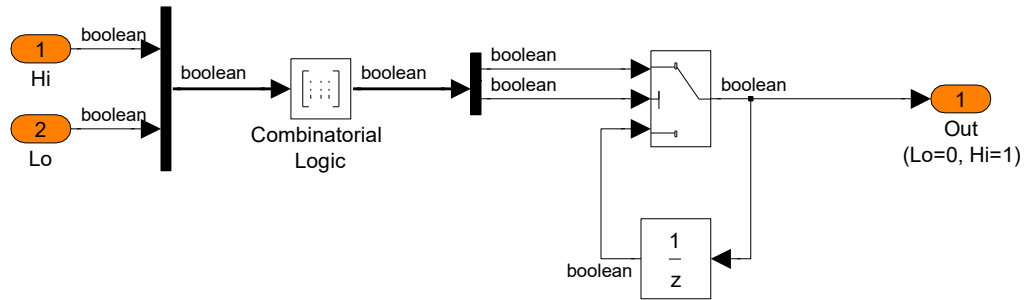


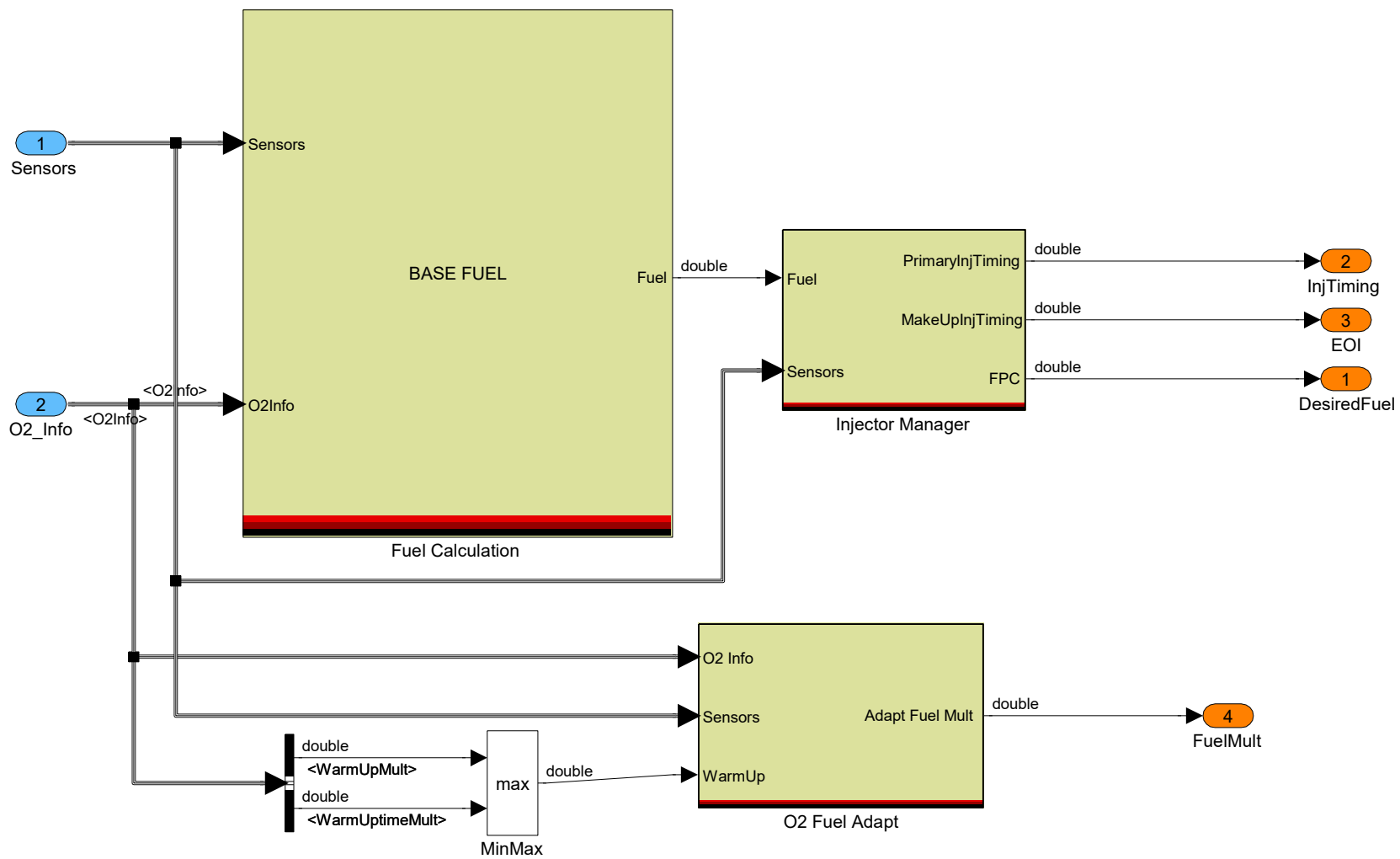
**Hi & !Lo => Hi**  
**Lo & !Hi => Lo**  
**Otherwise, Hold**

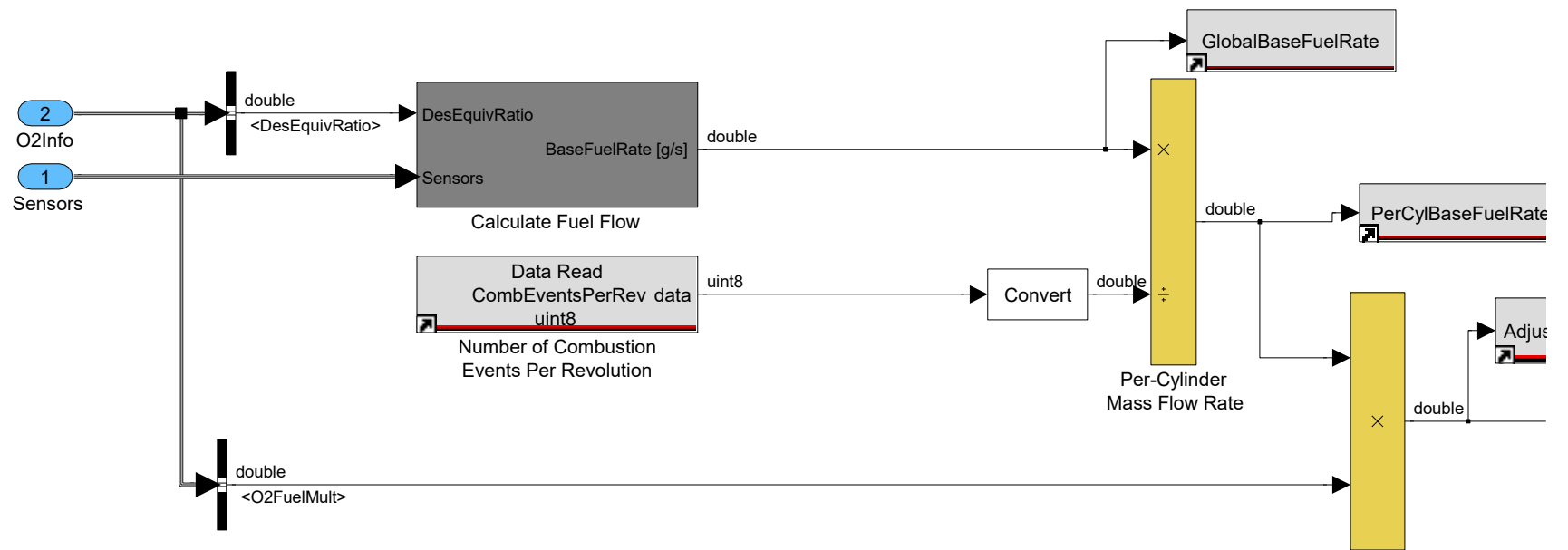




**Hi & !Lo => Hi**  
**Lo & !Hi => Lo**  
**Otherwise, Hold**



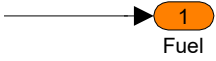


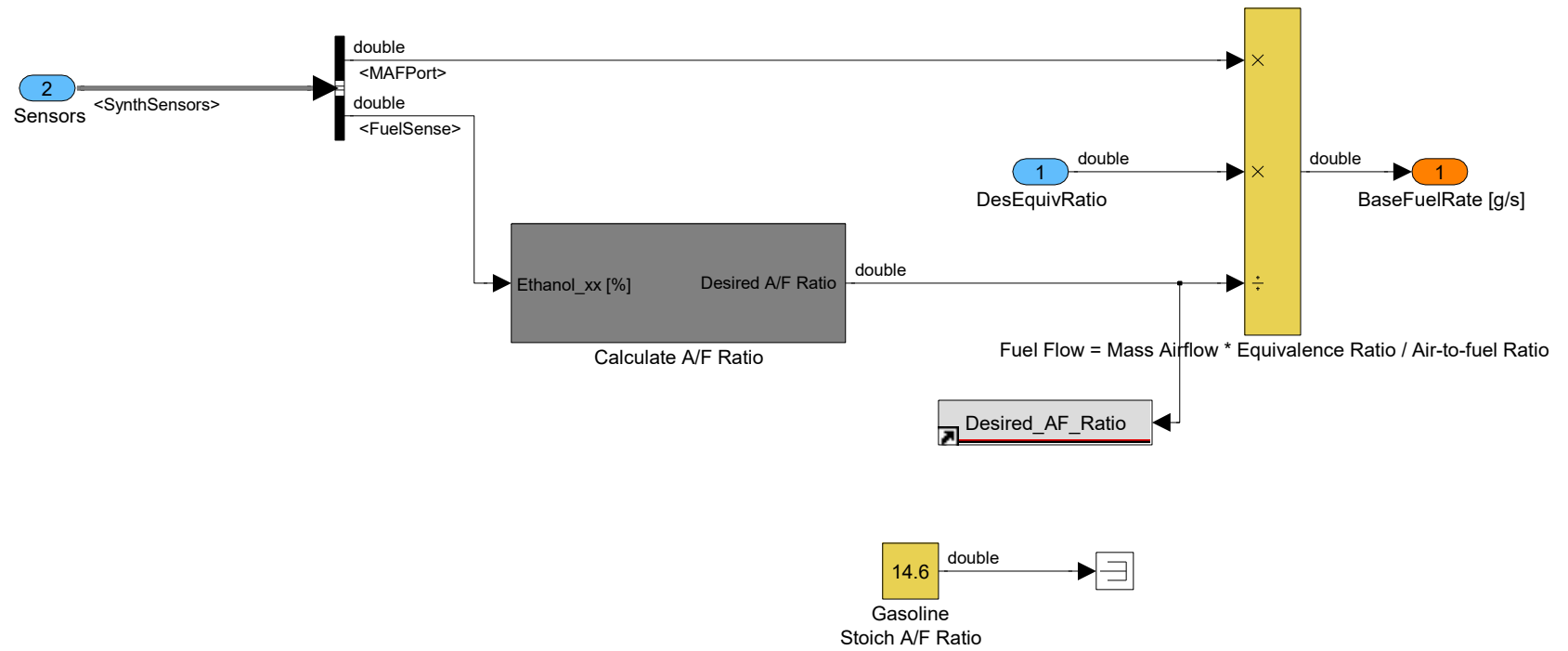


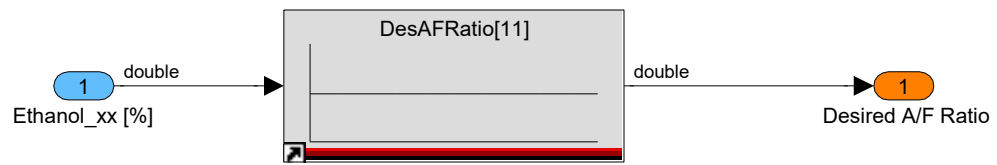


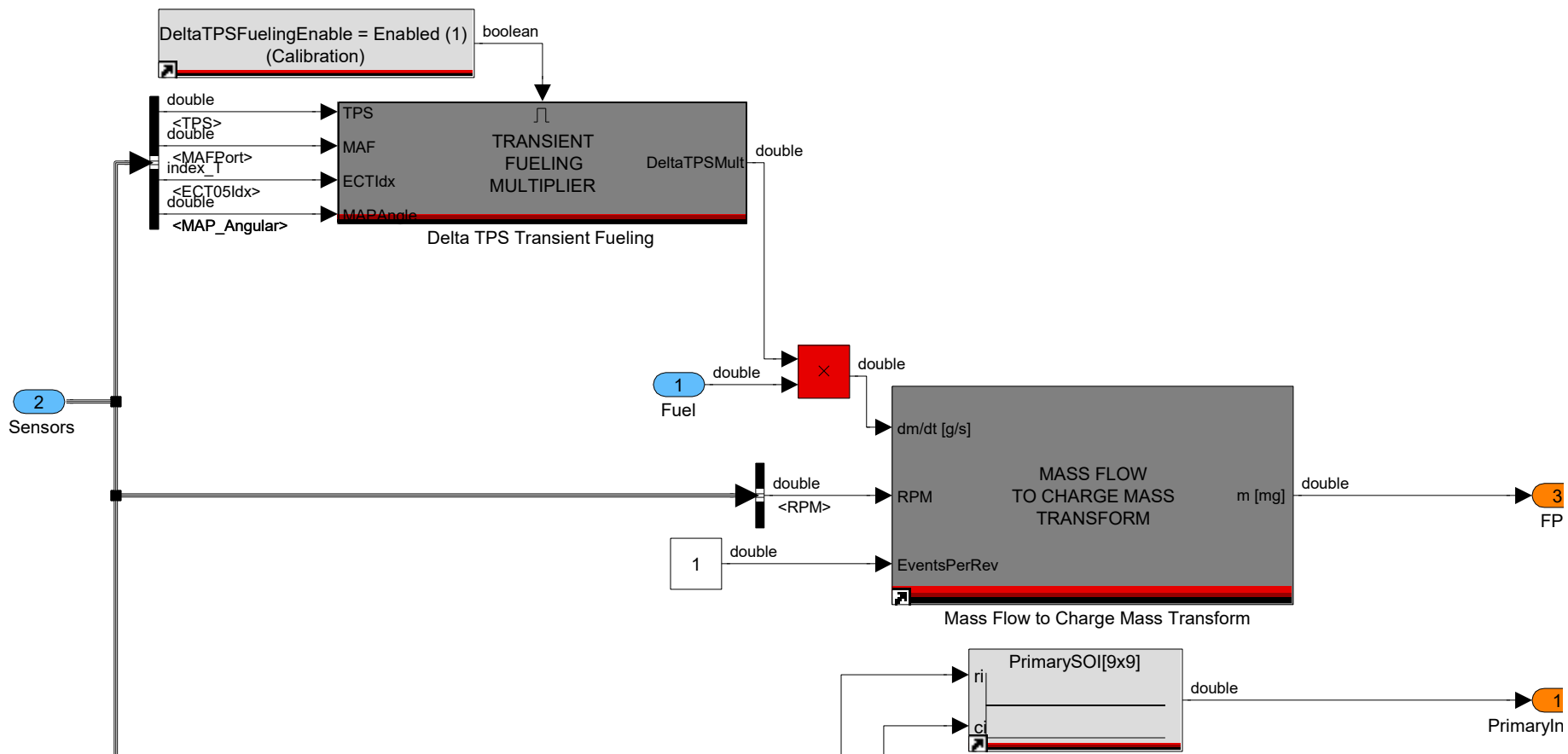
e

stedPerCylFuelRate







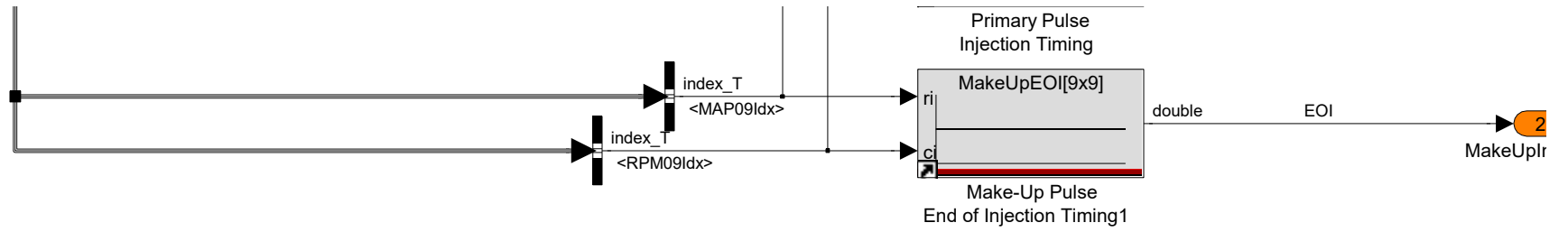




י"ד

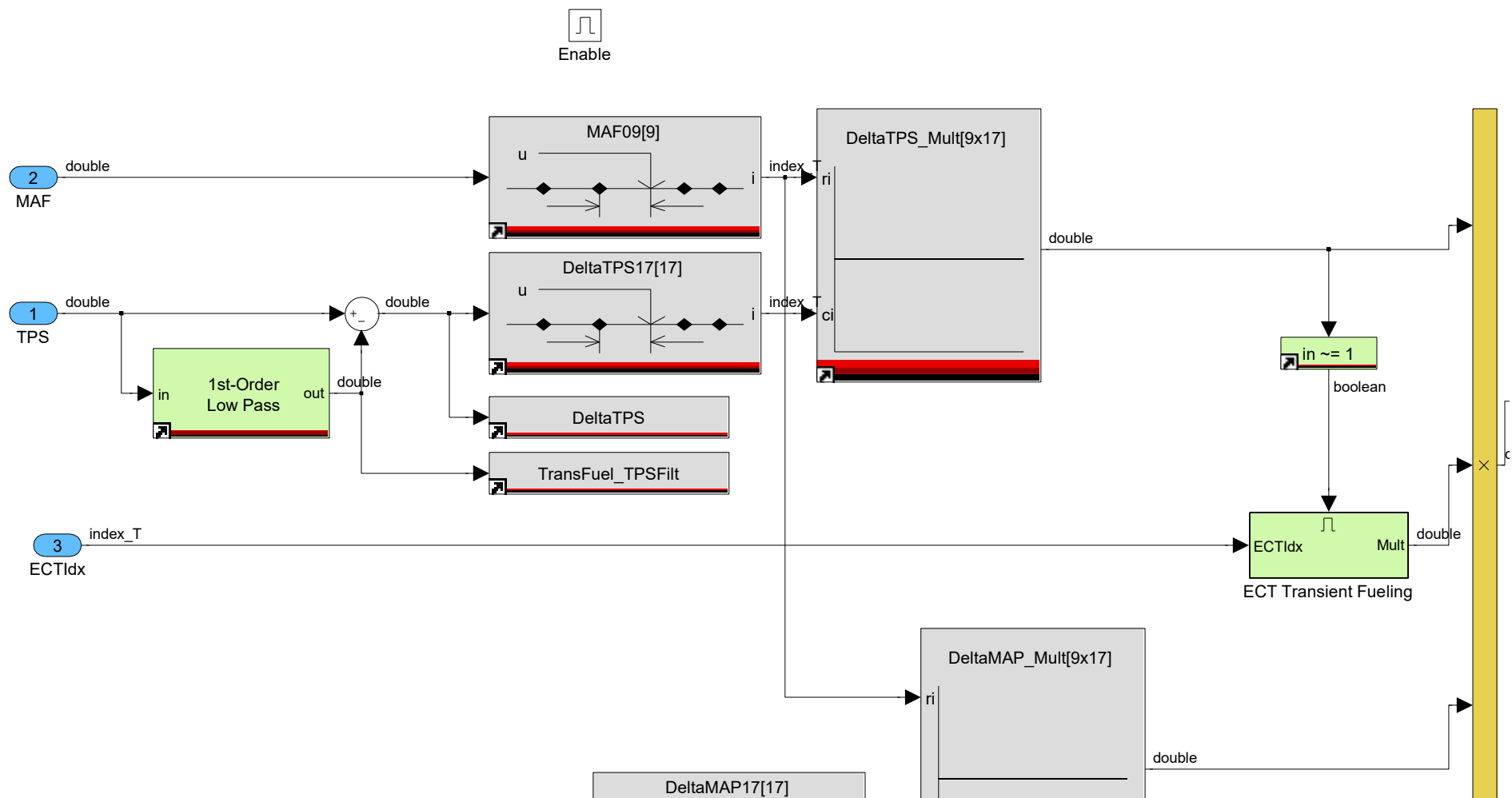


י"דTiming

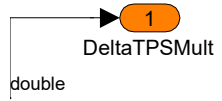


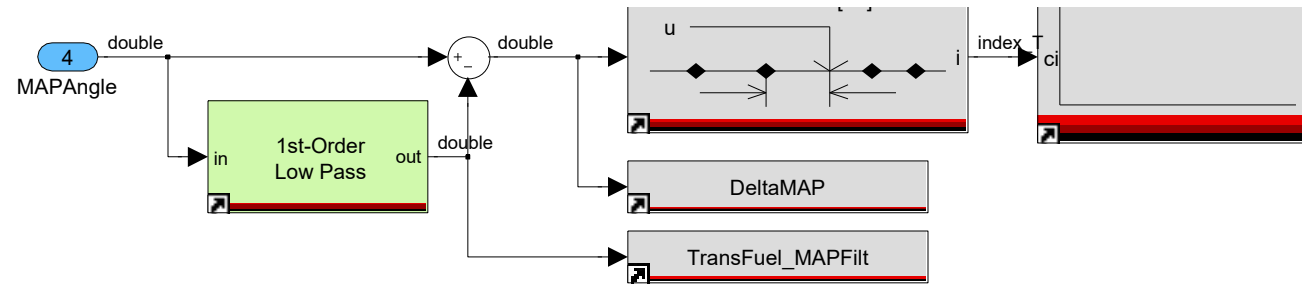


njTiming

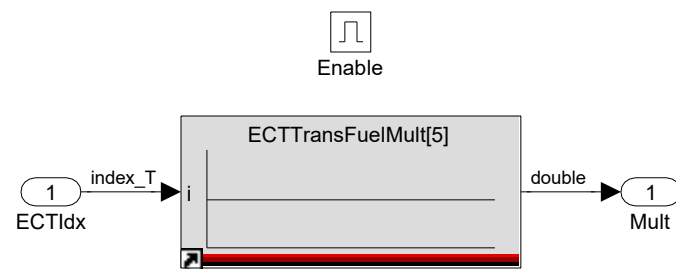


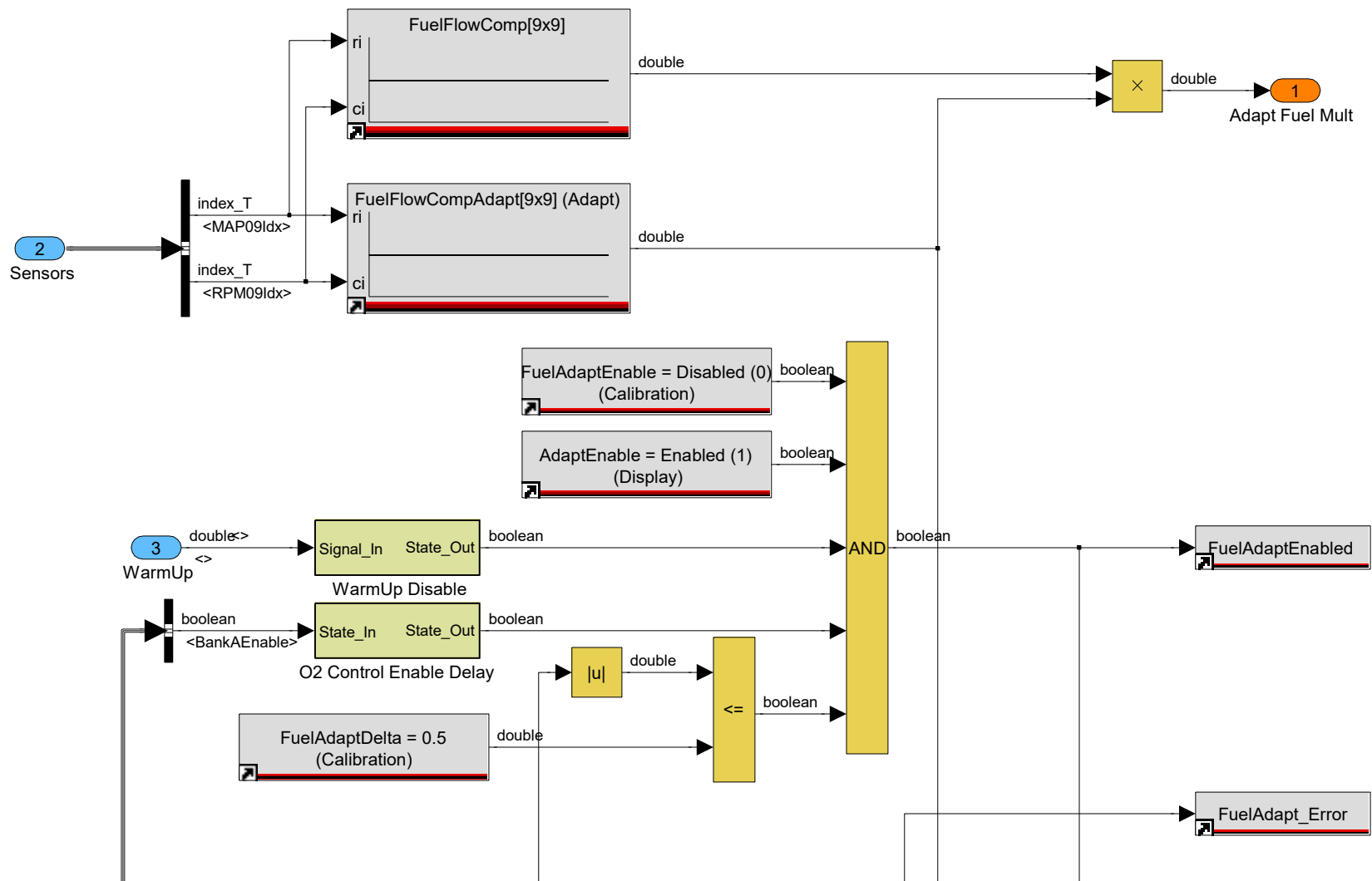


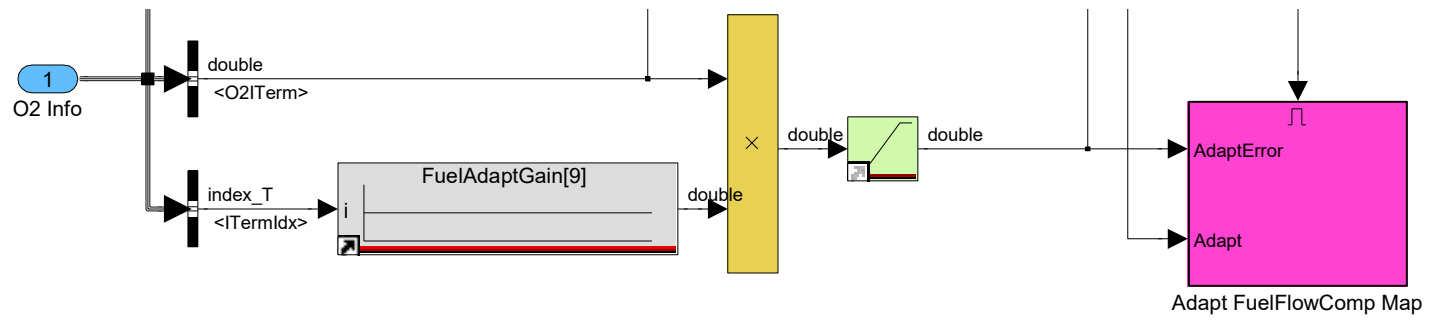


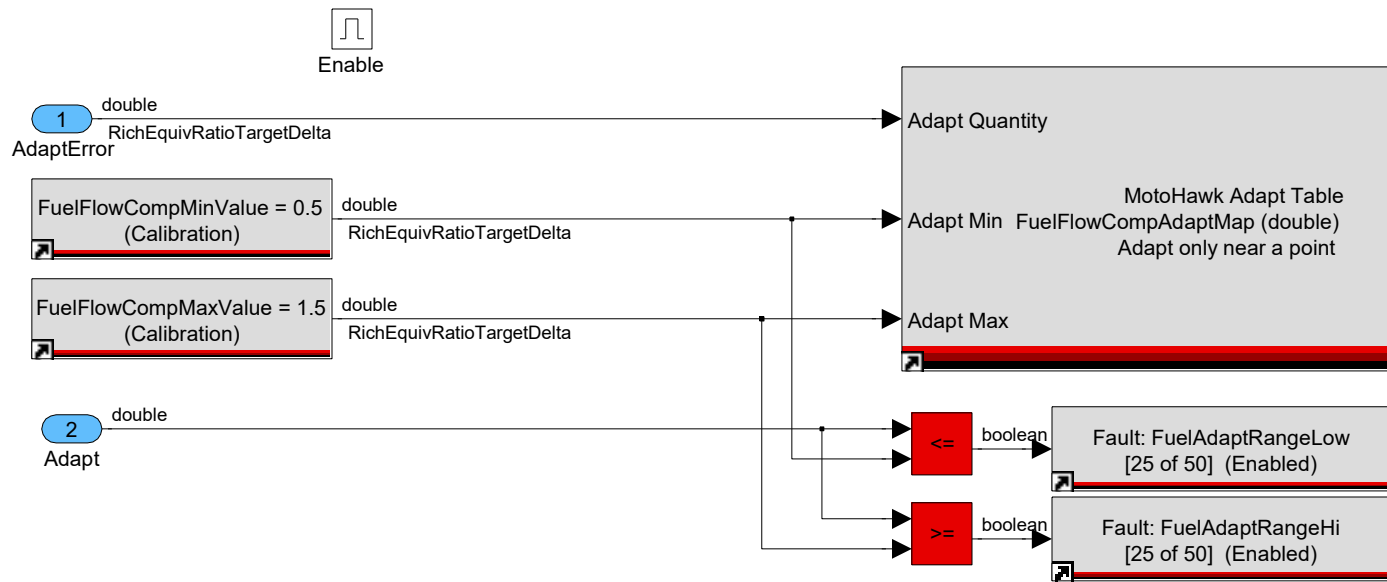


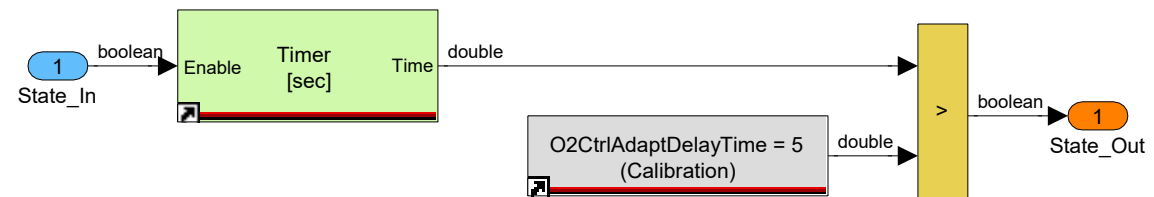






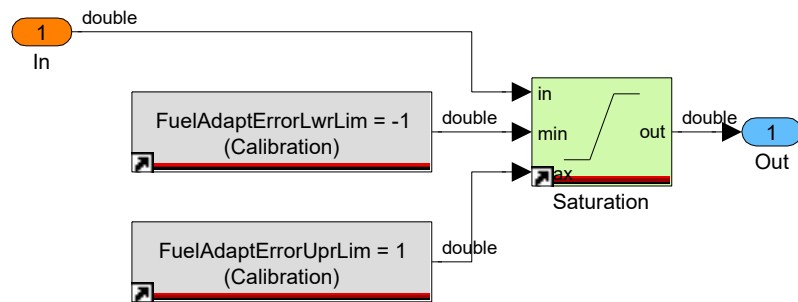


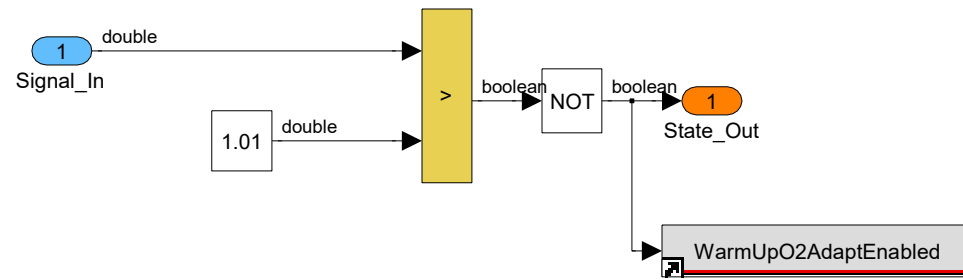


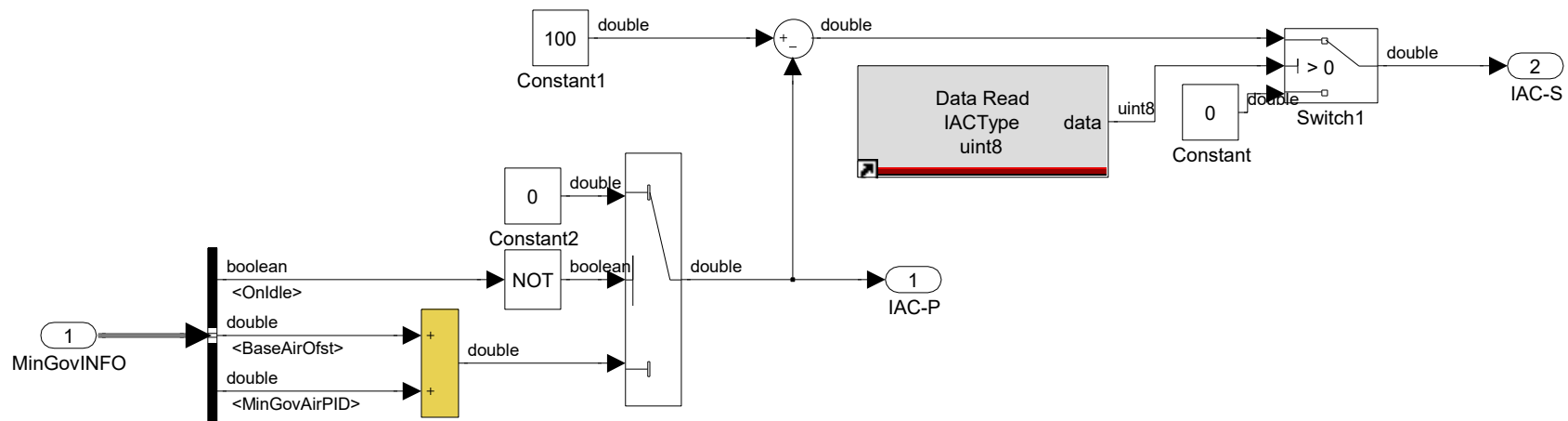




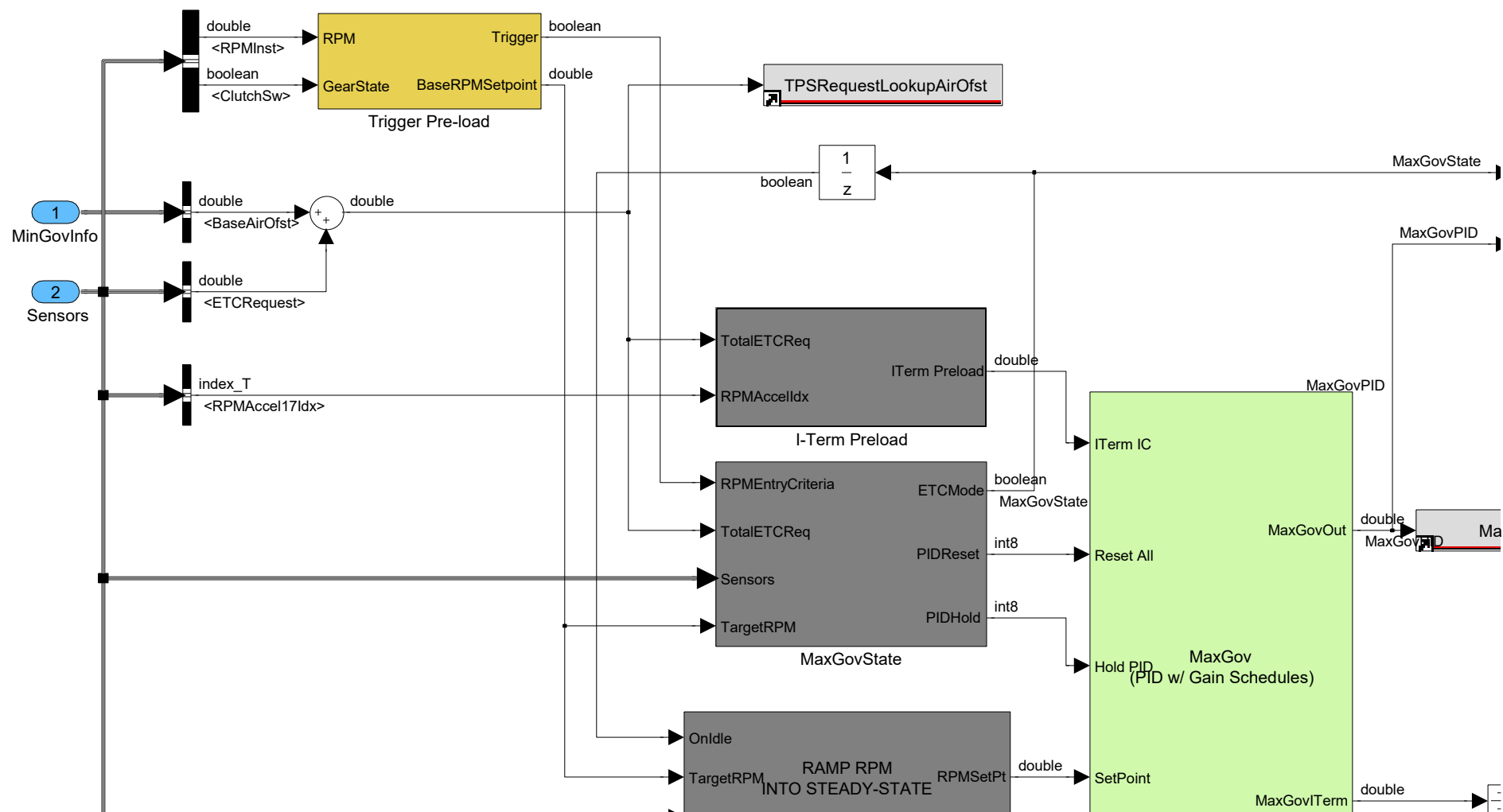
## Calibratable wrapper around the Saturation block

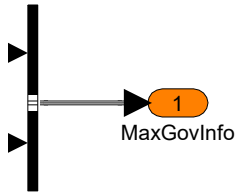






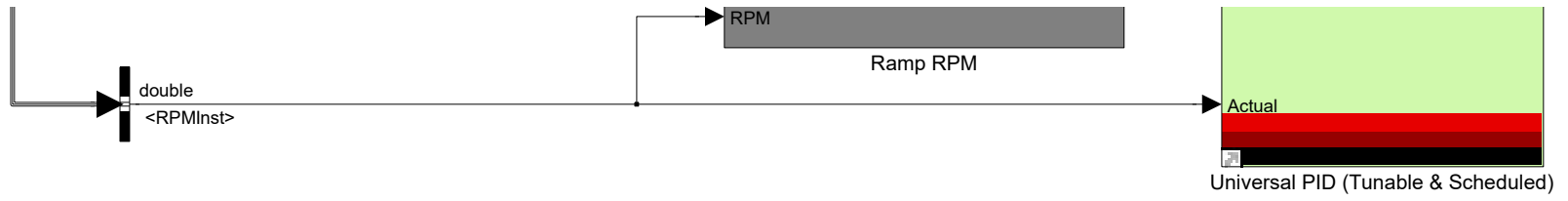
f()



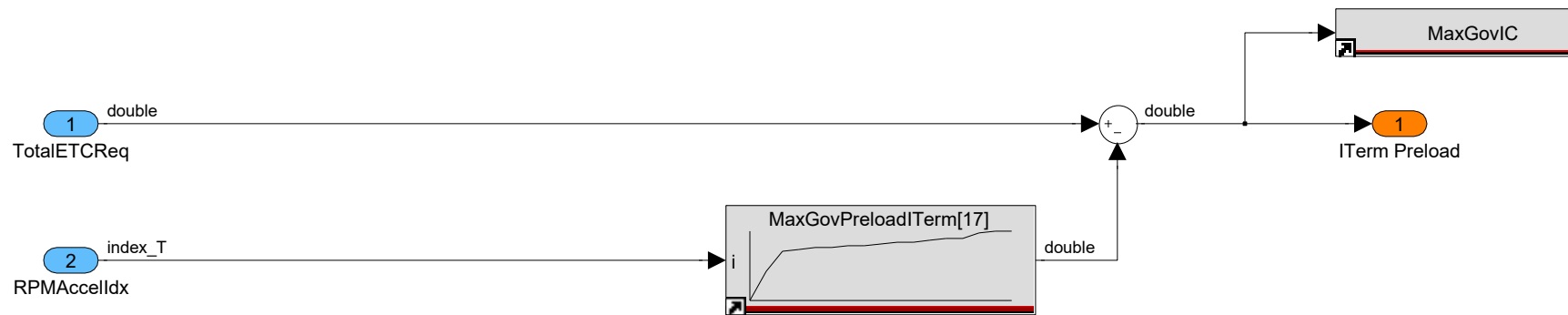


axGovOutput



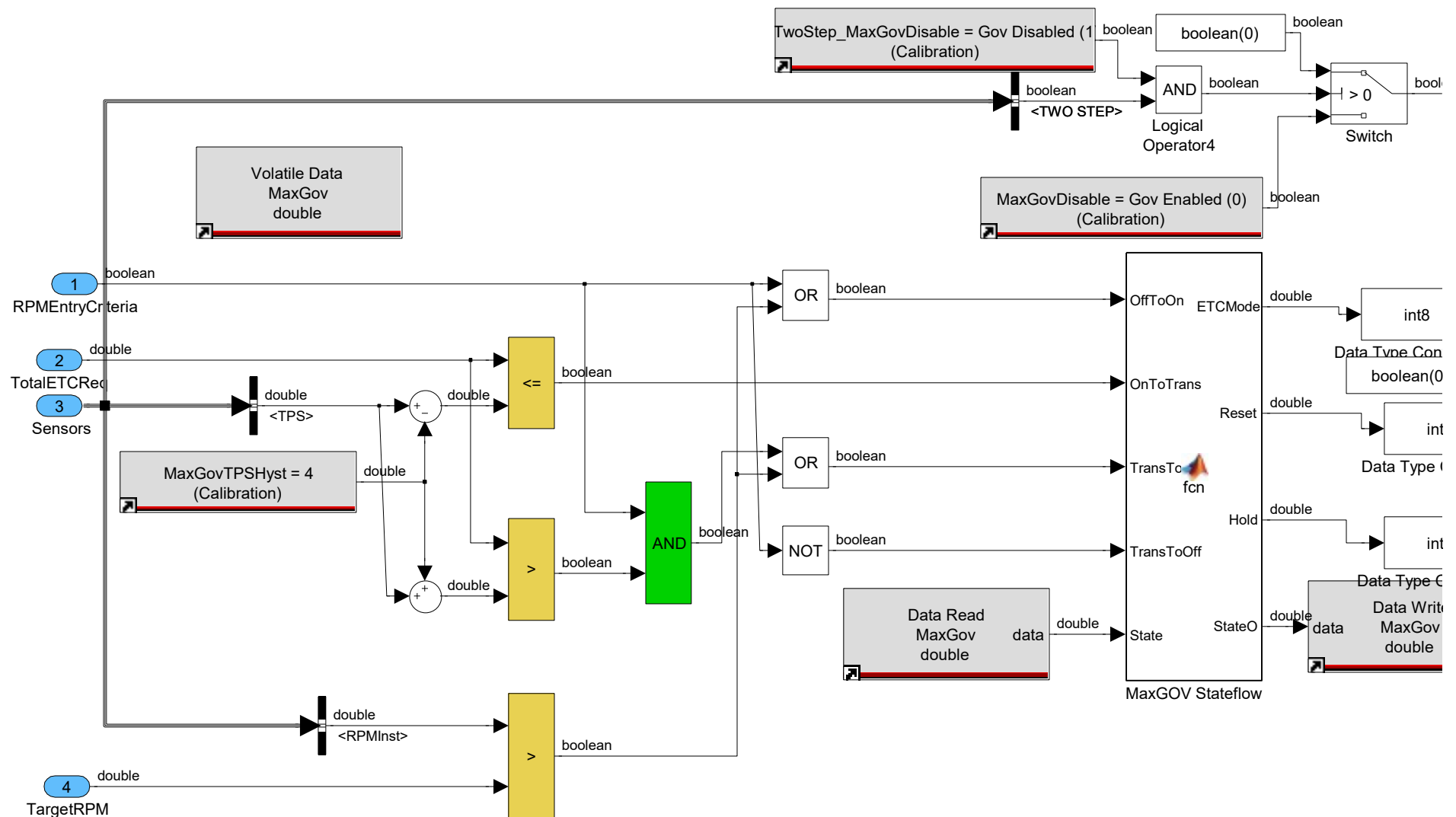












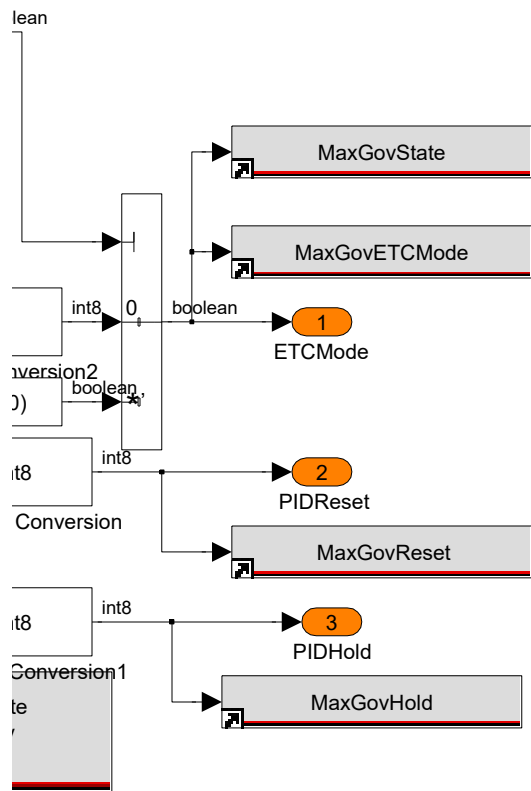
Move into On State if  $RPM > EntryCriteria$  or Target.

Move into Trans State if APP falls below TPS, PID controller is frozen during Trans State, but output is based on APP.

When in AllSpeedGov mode, switch to Trans if AllSpeedGov setpoint is less than MaxGov setpoint minus hyst.

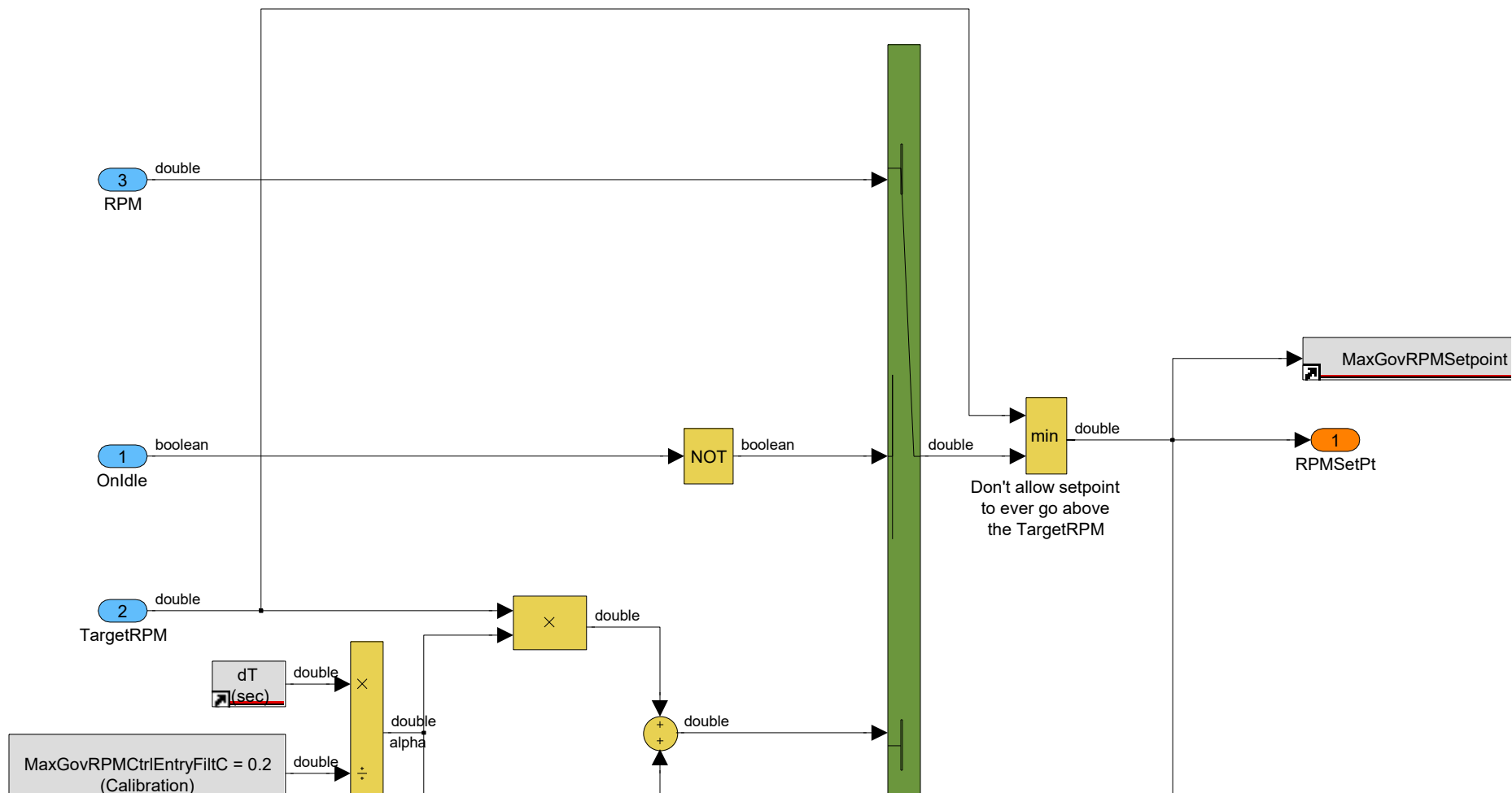
Move from Trans State back to On State if  $RPM > Target$ .

Move from Trans State to Off if  $RPM < EntryCriteria$ .

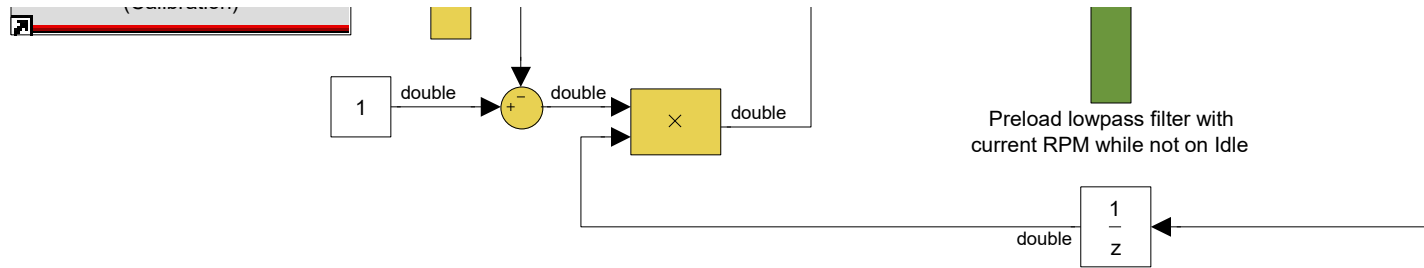


Move from Trans State to Off if  $RPM < \text{Entry Criteria}$ .









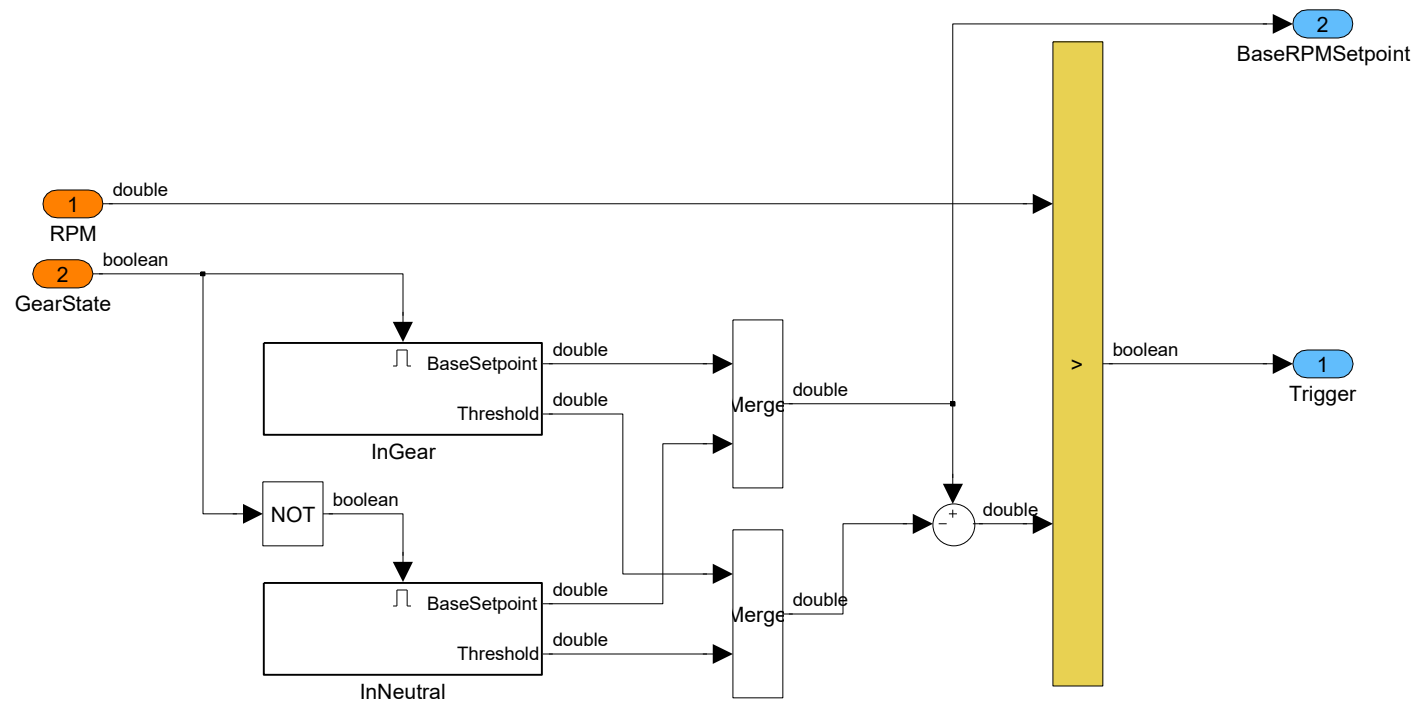
**1st-Order Ramp Up**

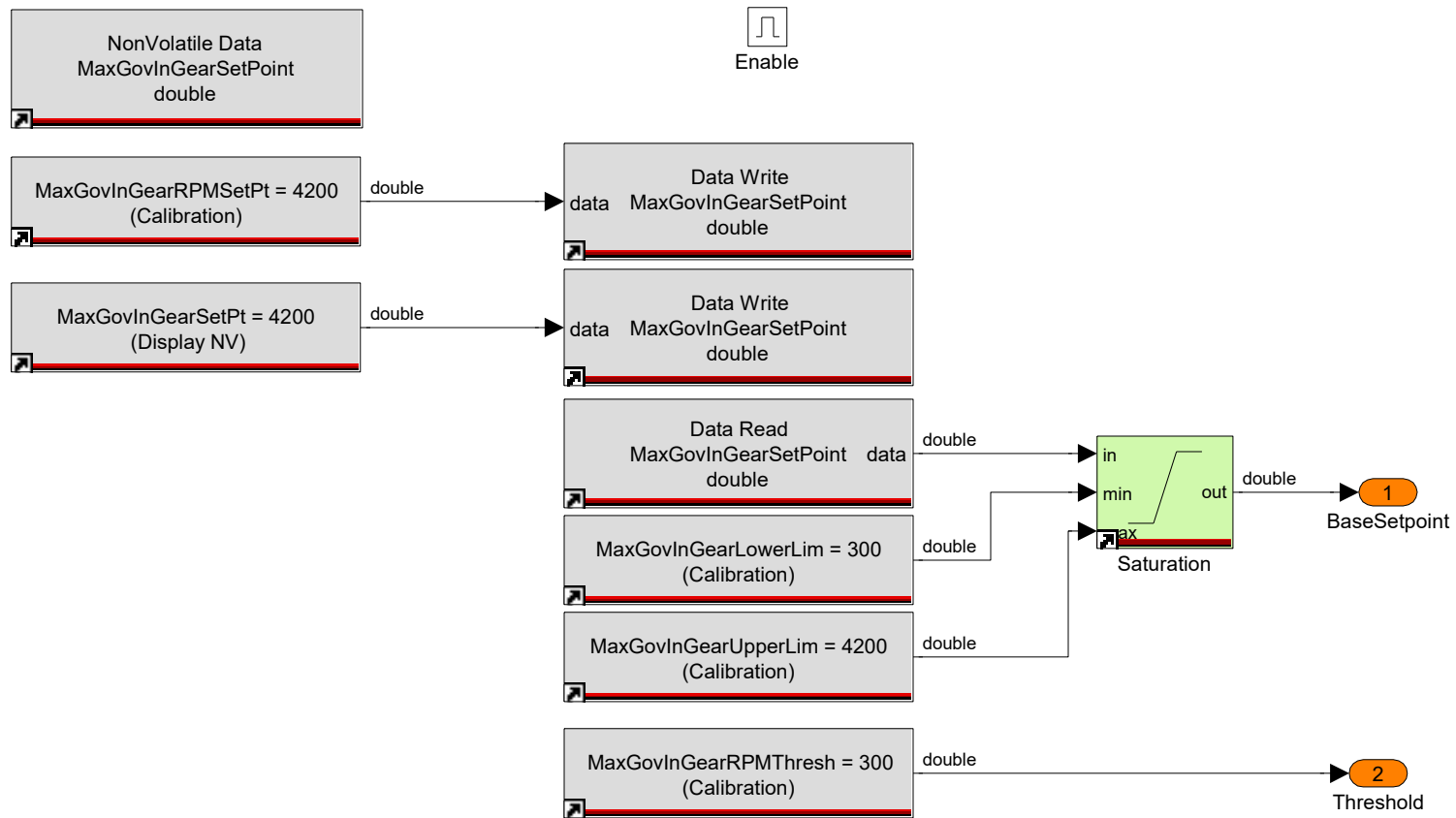
$$y[k] = a \cdot x[k] + (1-a) \cdot y[k-1]$$

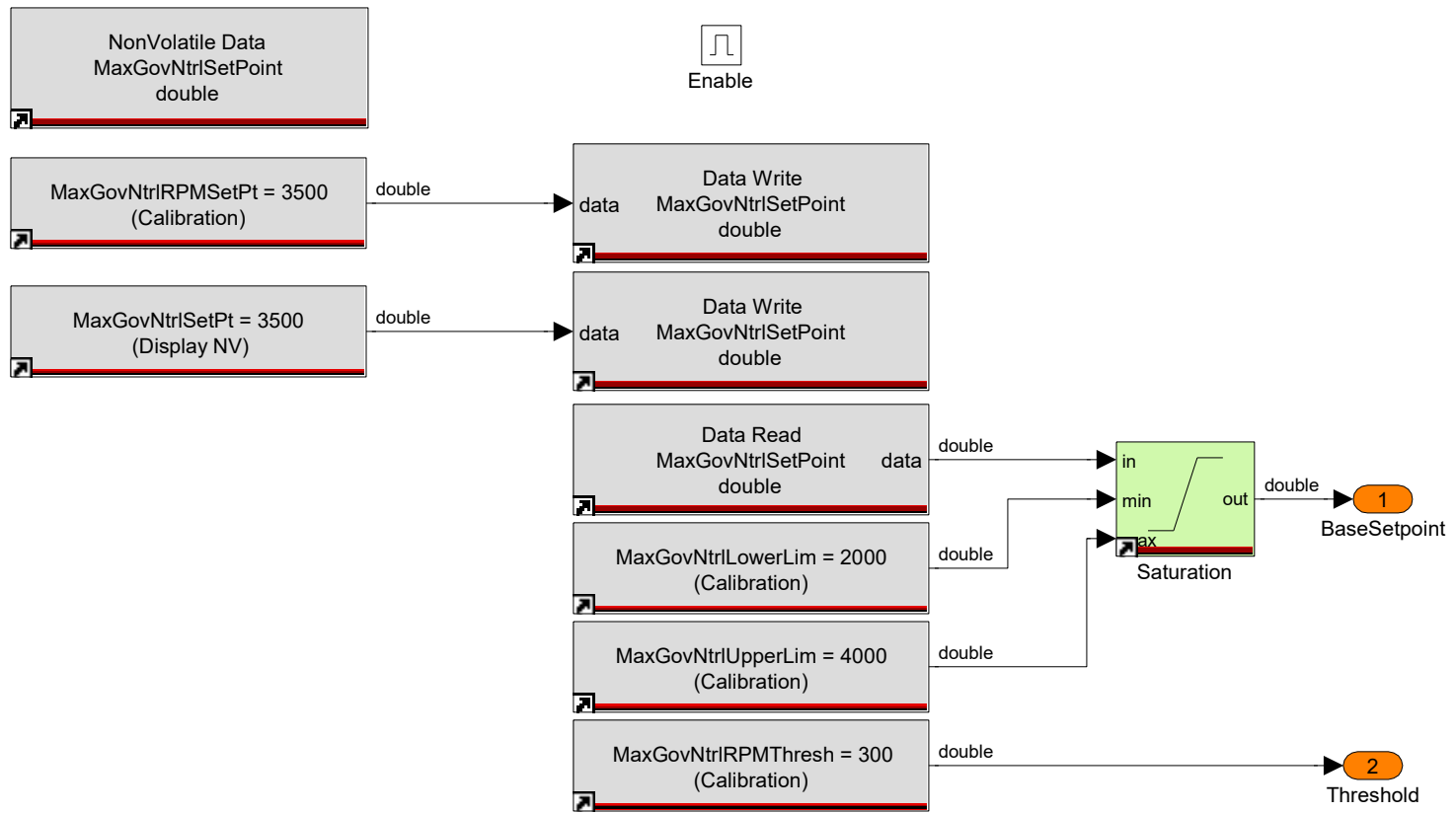
where  $a = t/T$

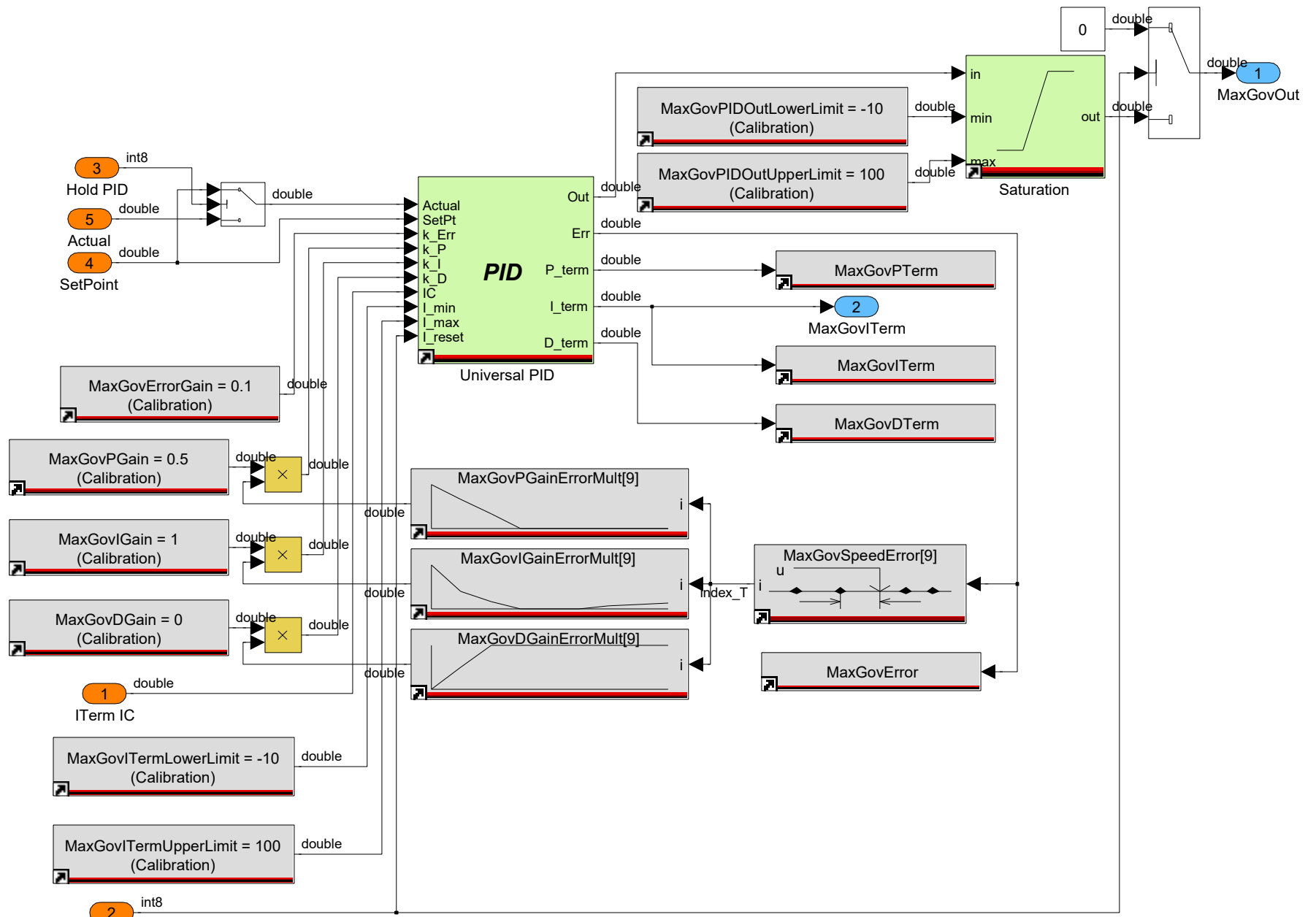






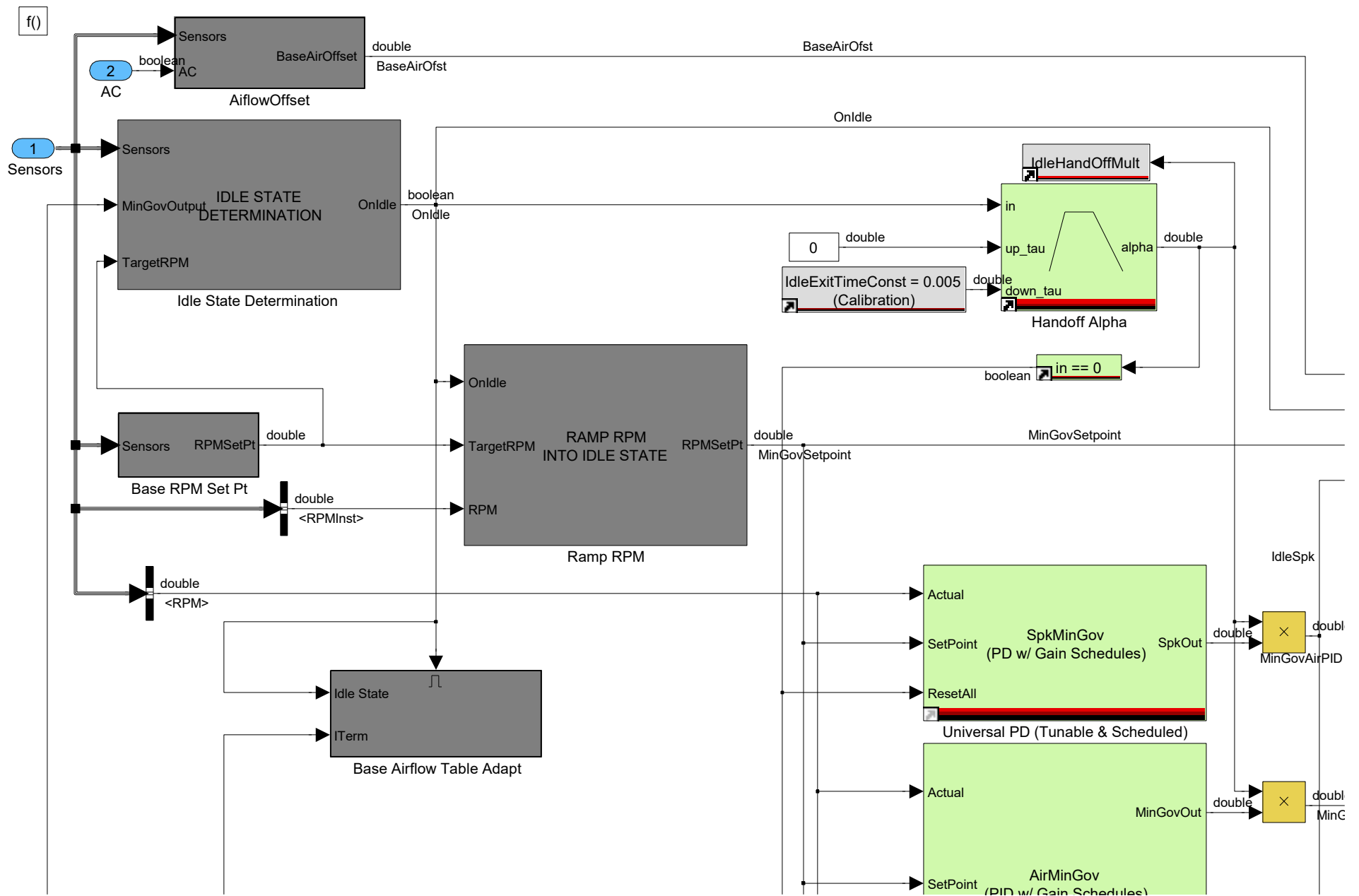


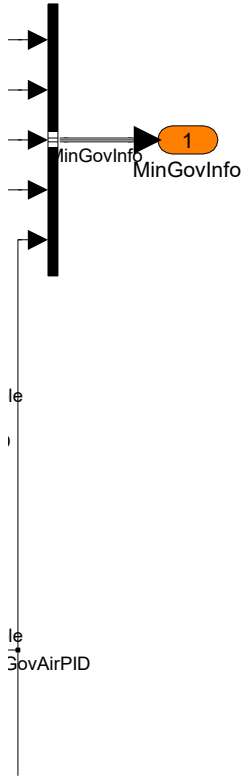




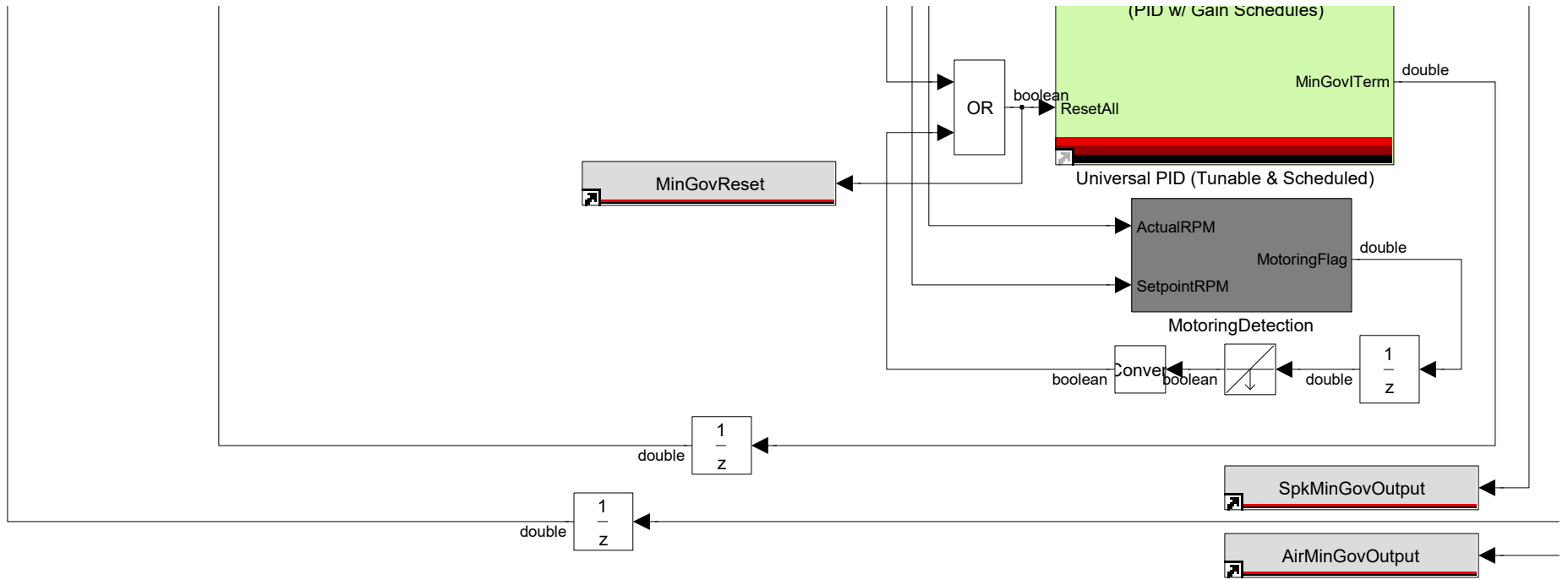
 Reset All

---

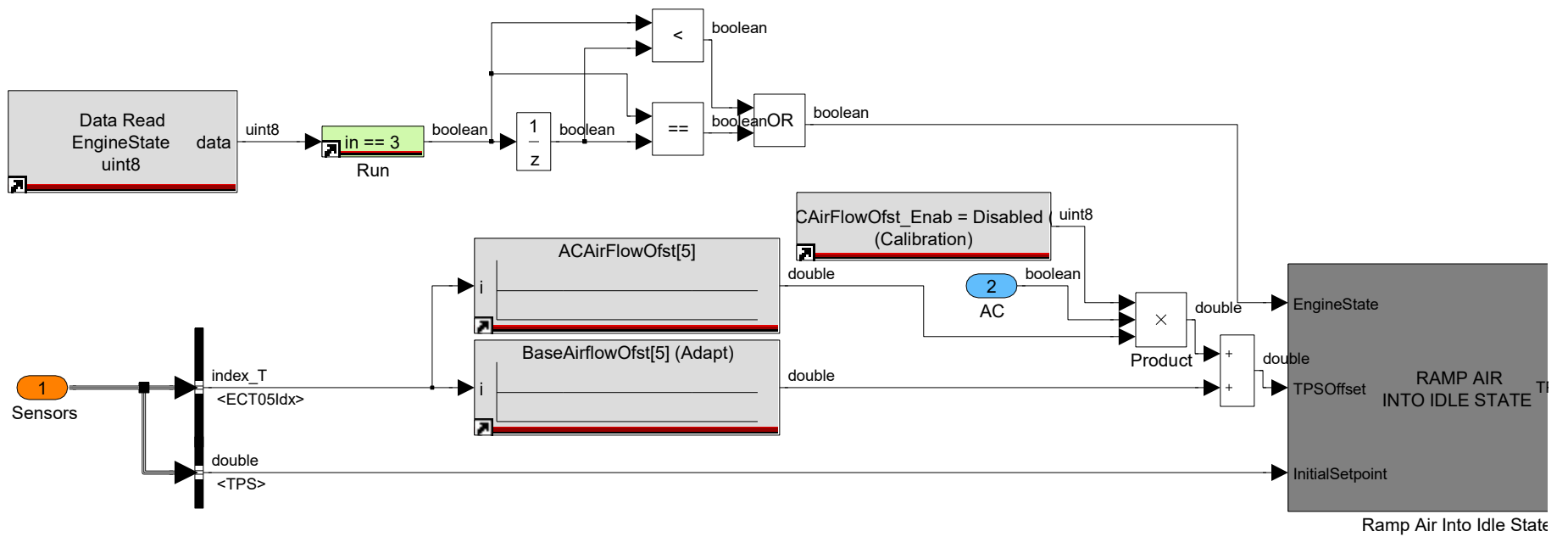


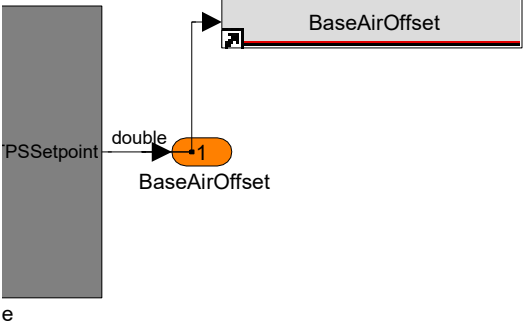


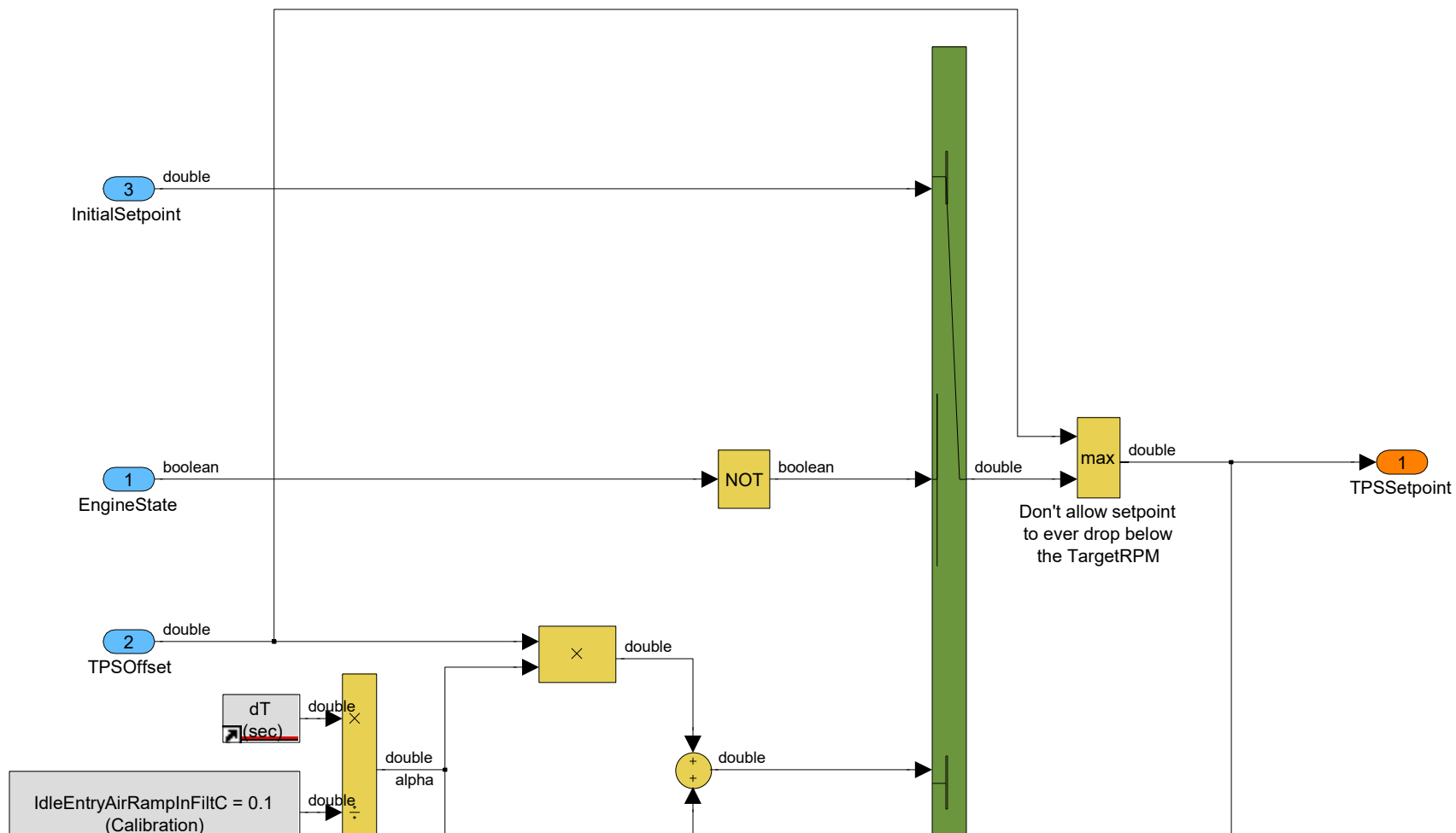


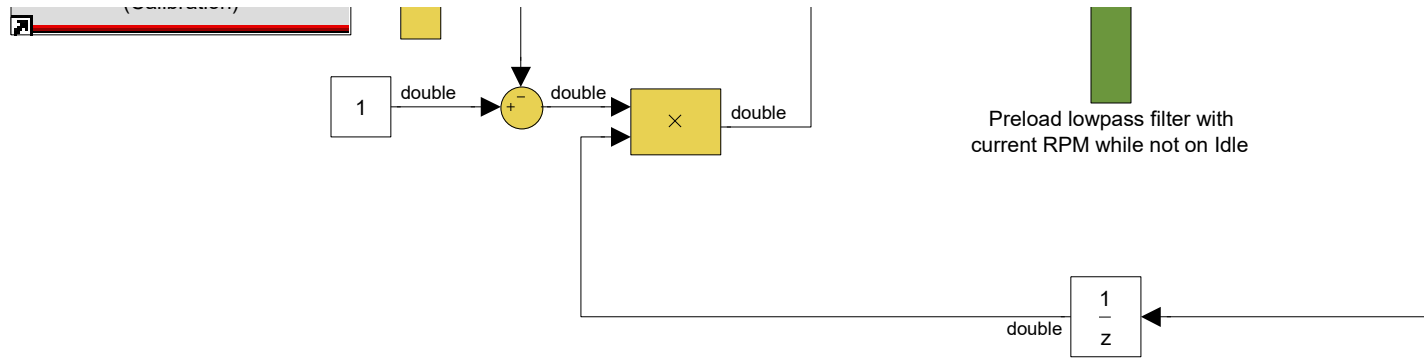




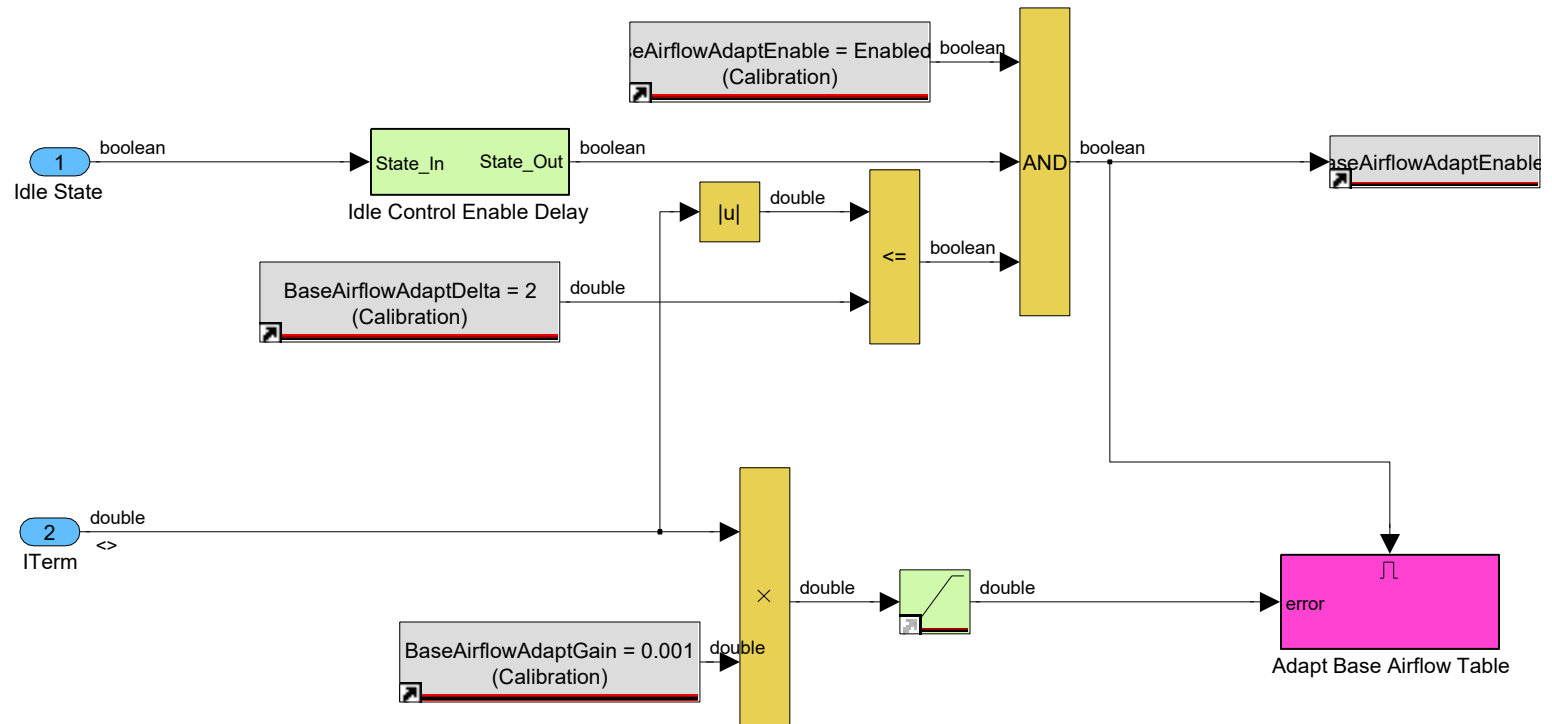


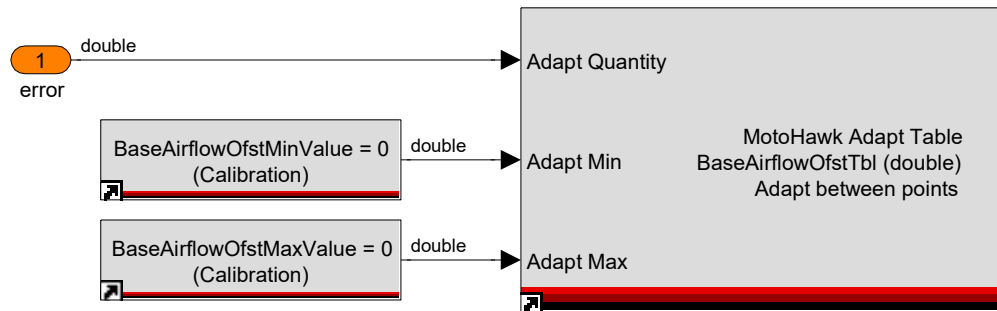




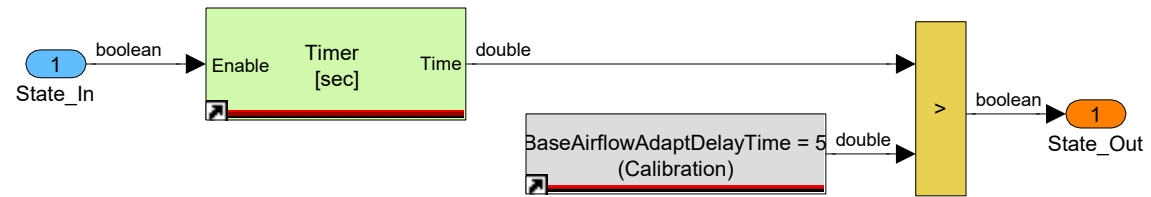


**1st-Order Ramp Down**  
 $y[k] = a \cdot x[k] + (1-a) \cdot y[k-1]$   
 where  $a = t/T$

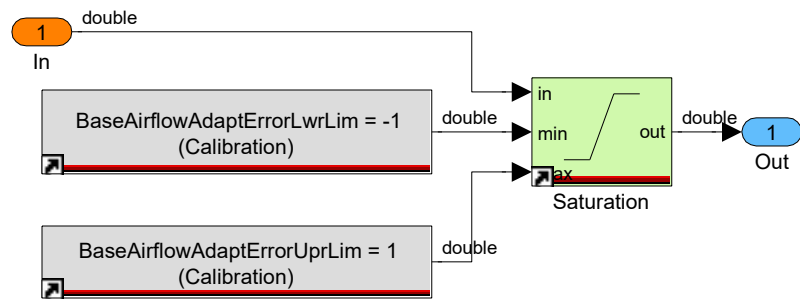


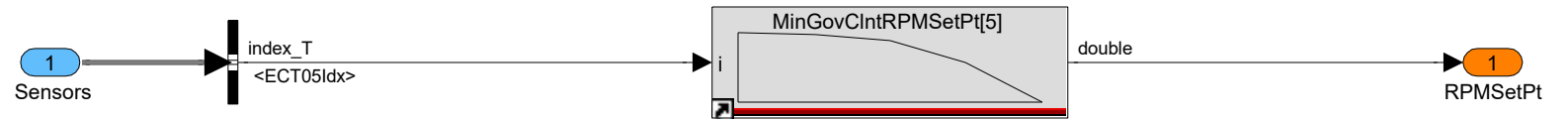


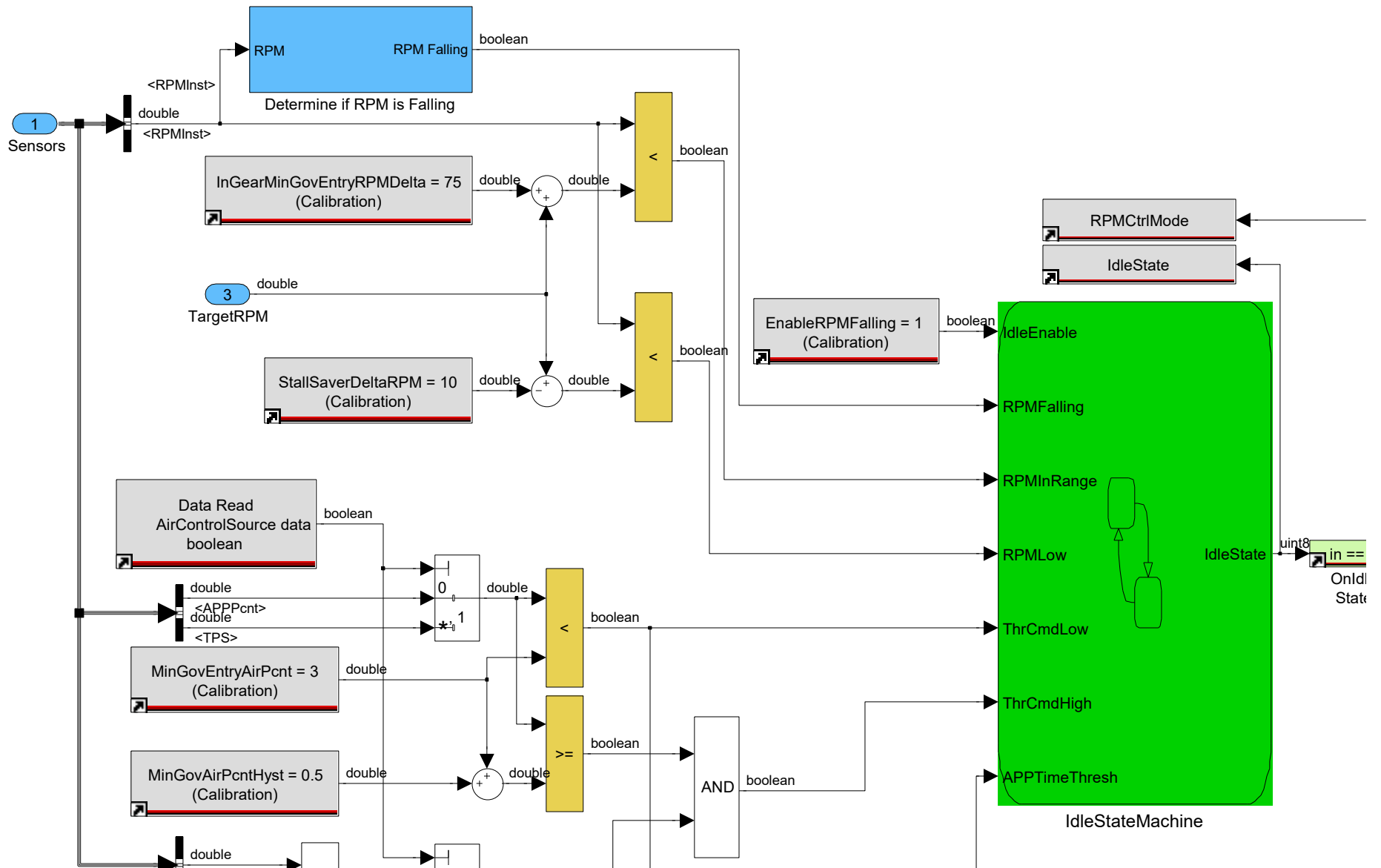


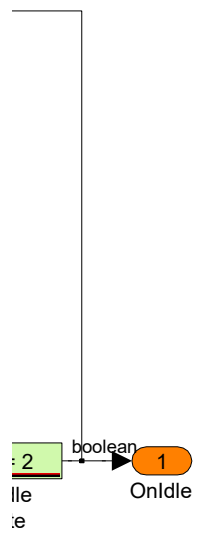


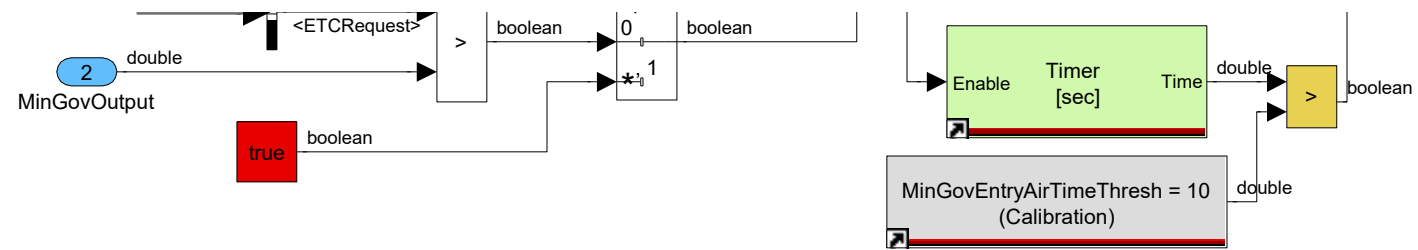
## Calibratable wrapper around the Saturation block





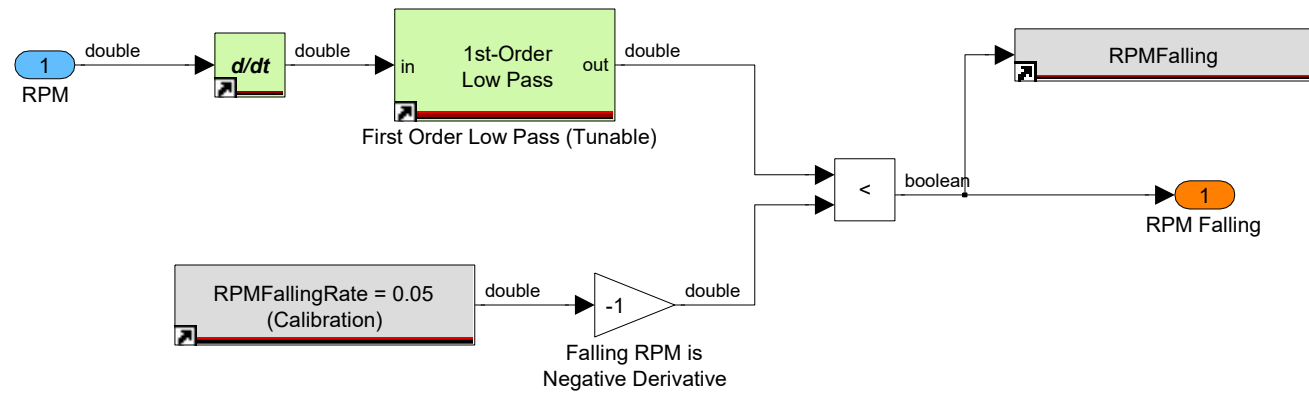




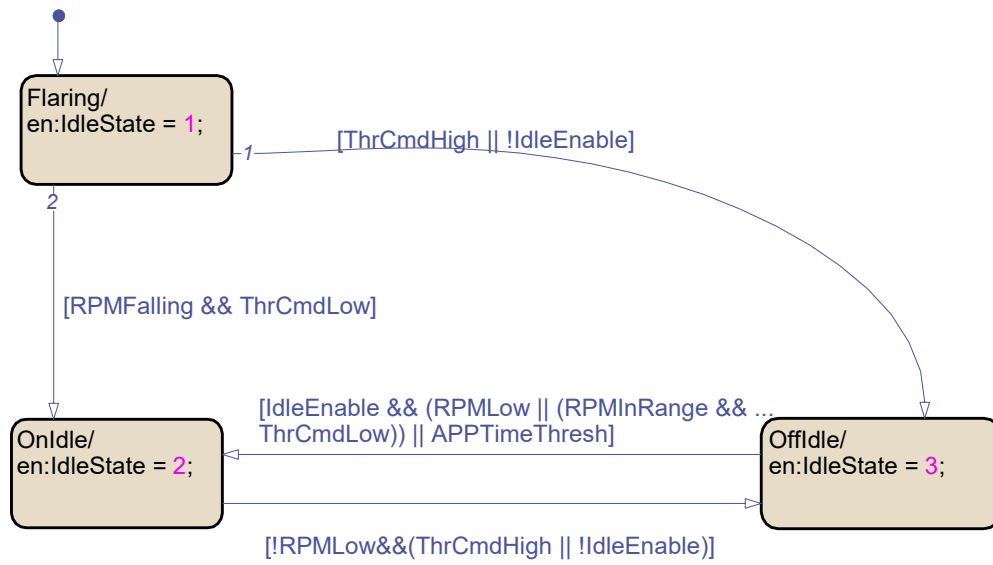


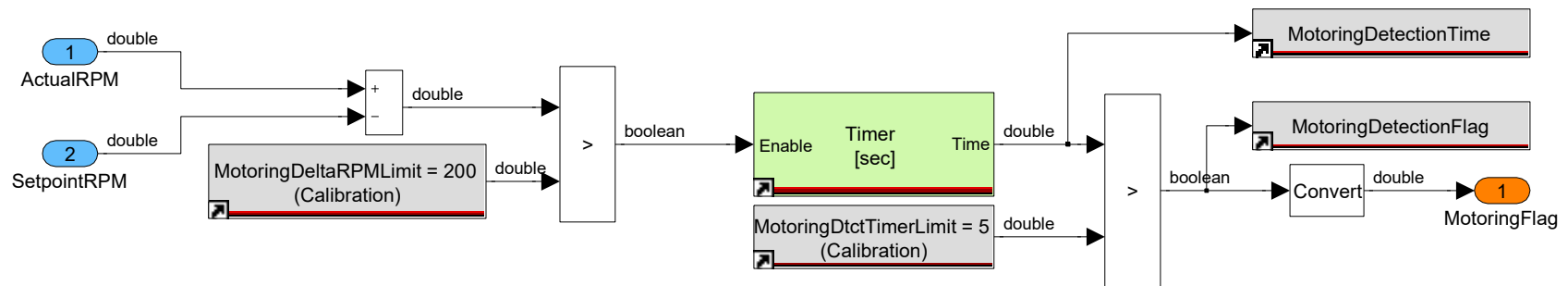


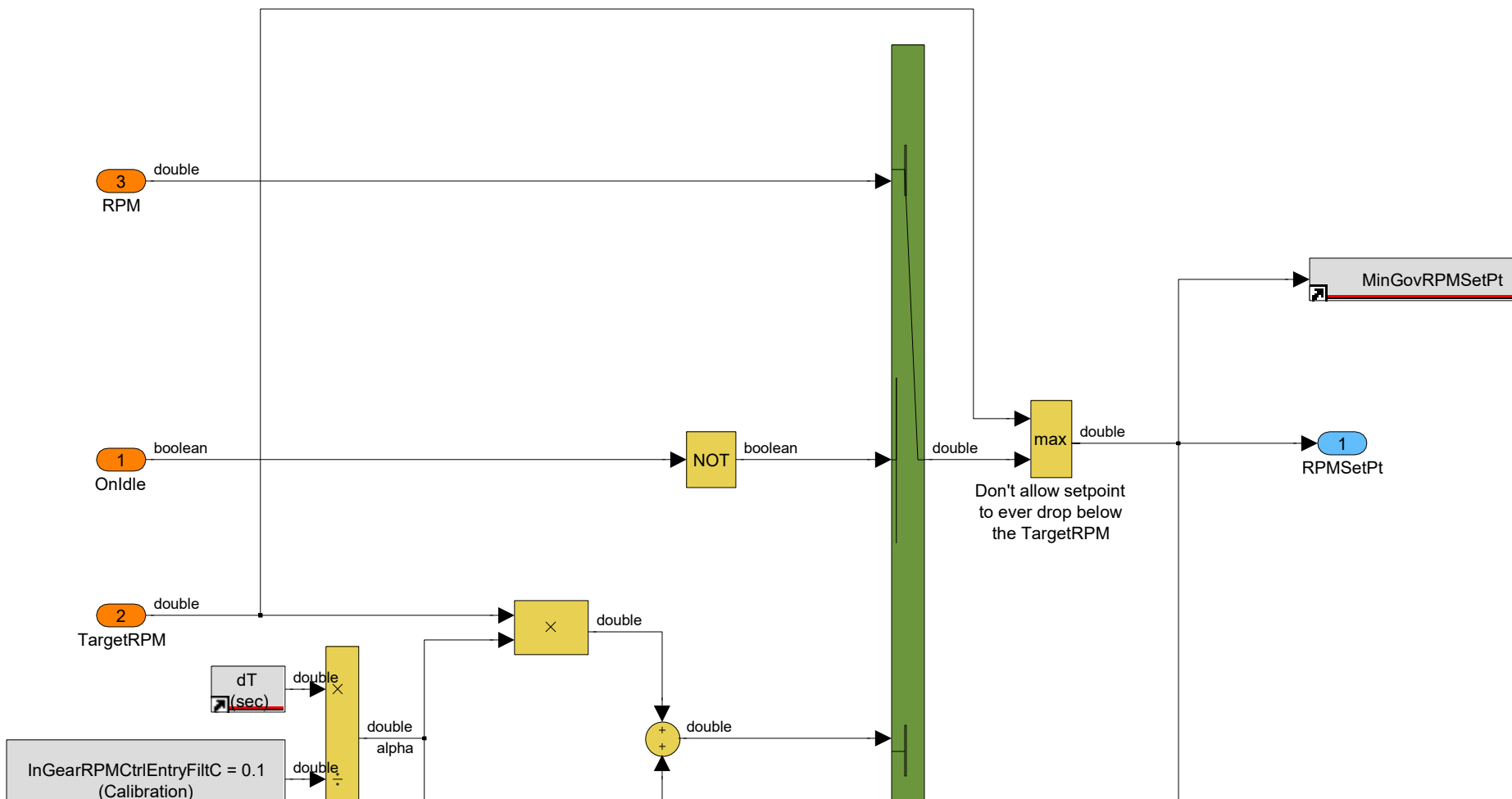
## Determine if RPM is falling



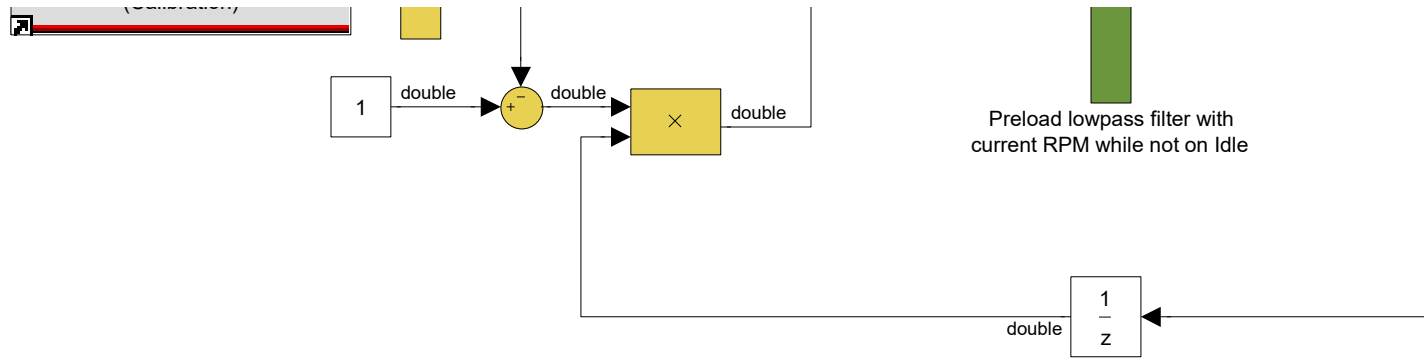






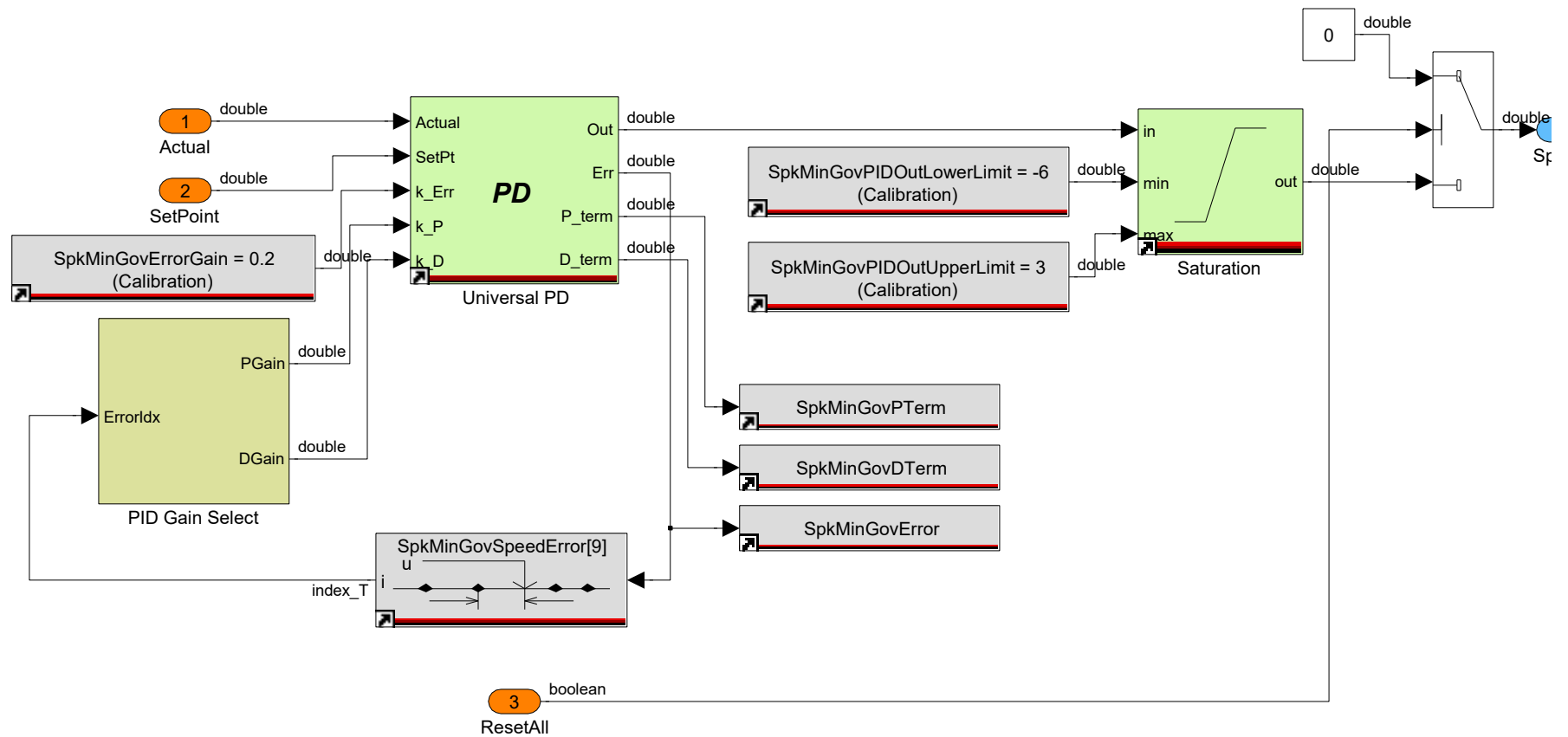






**1st-Order Ramp Down**  
 $y[k] = a \cdot x[k] + (1-a) \cdot y[k-1]$   
 where  $a = t/T$

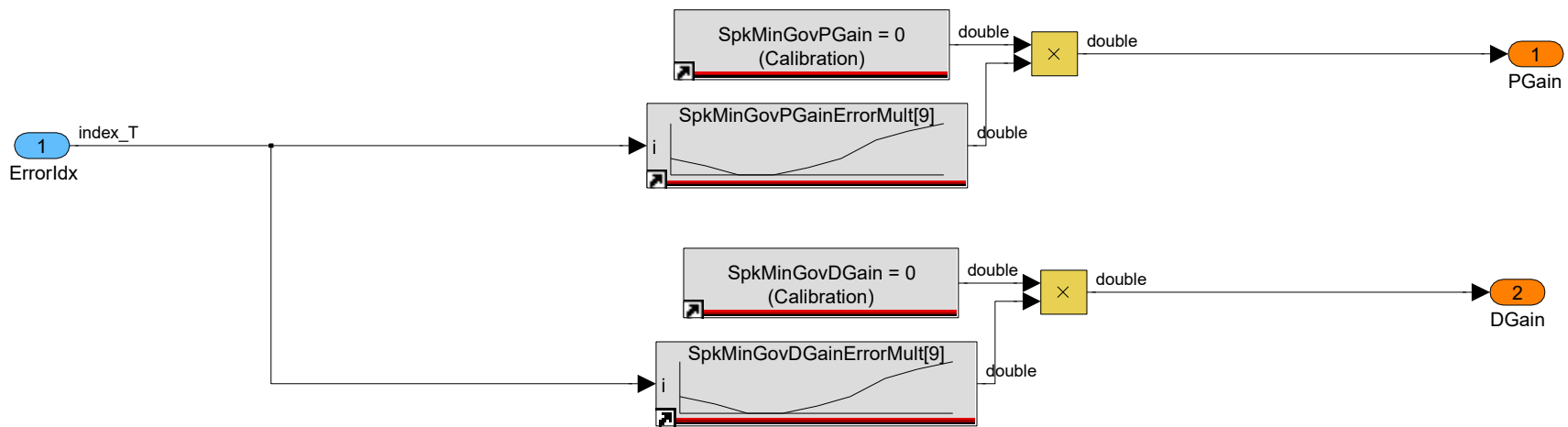


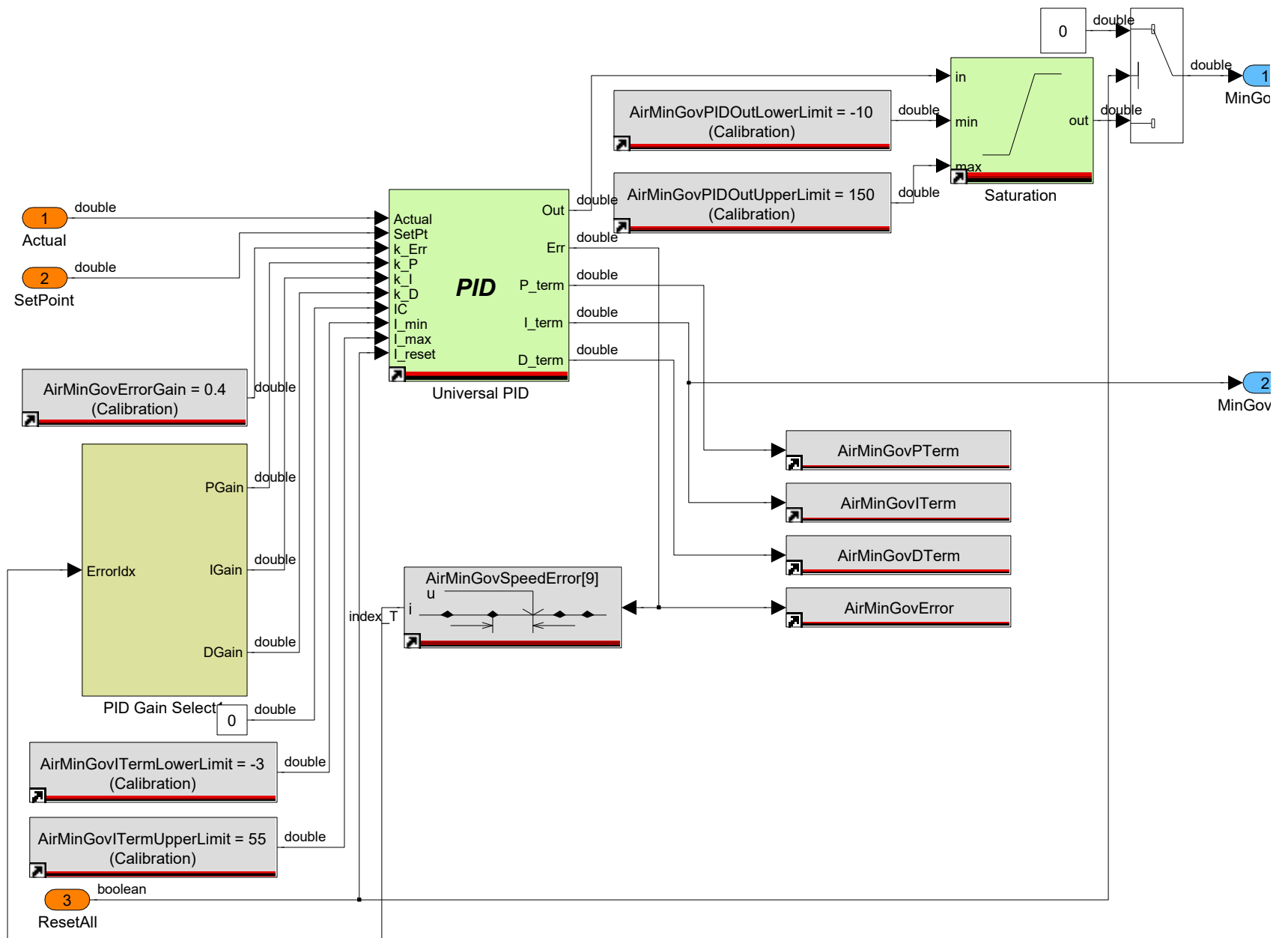


1

pkOut



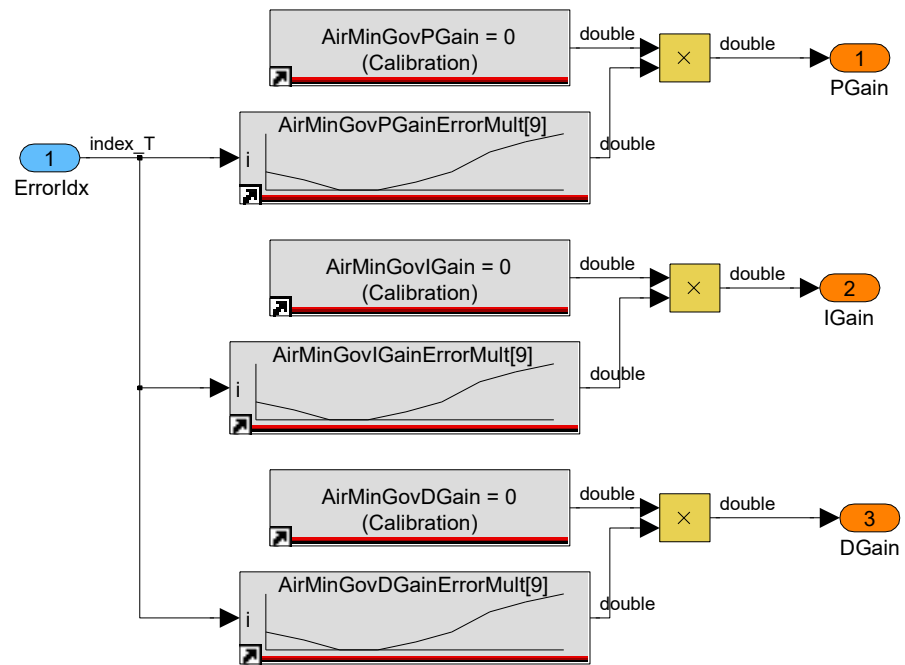


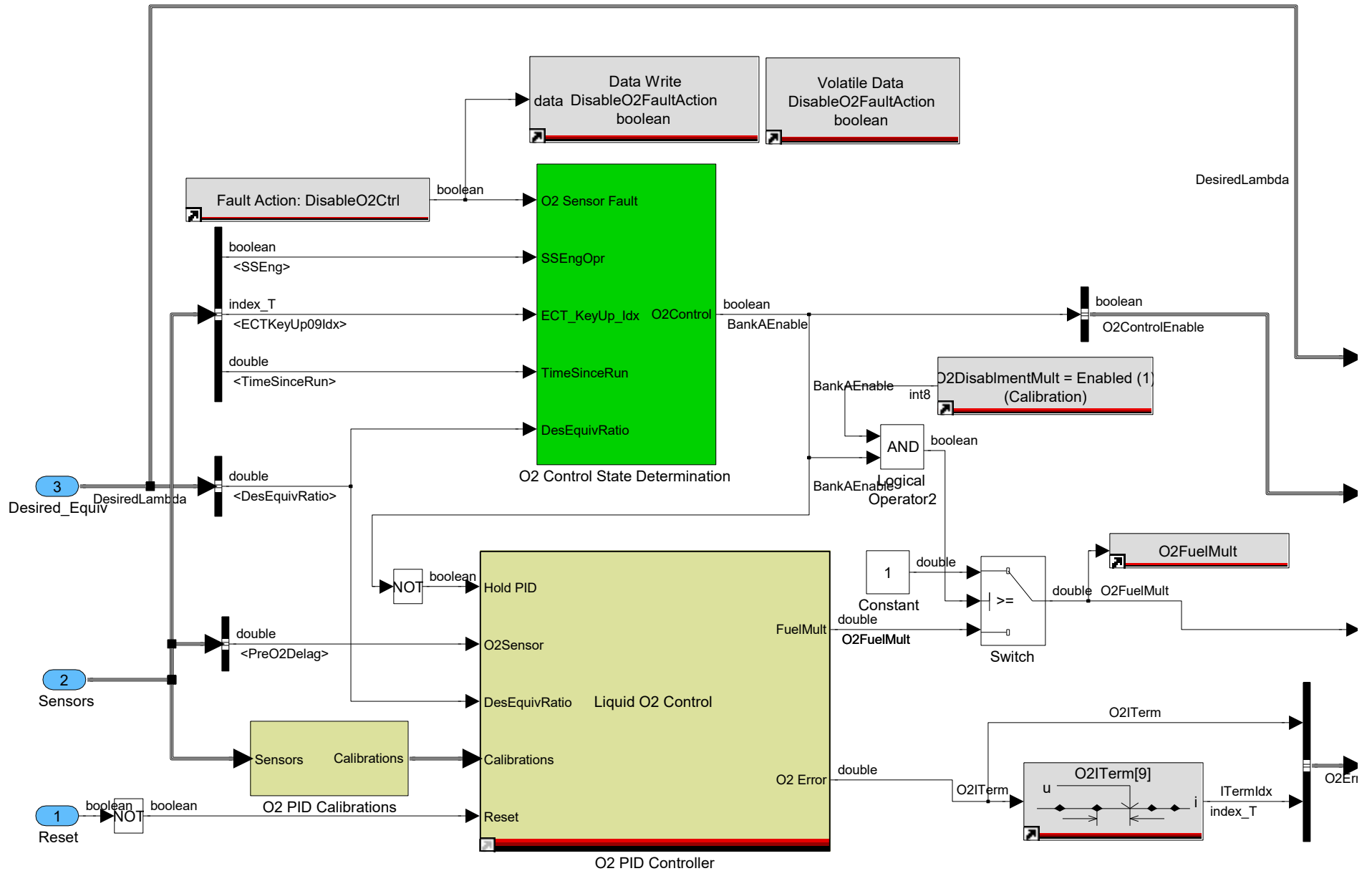


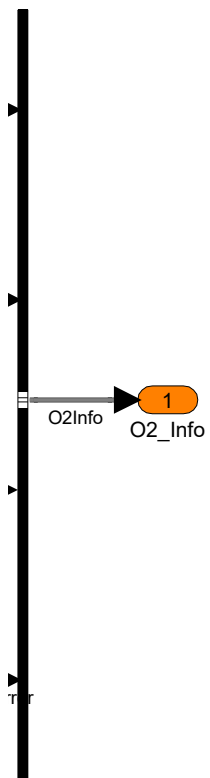




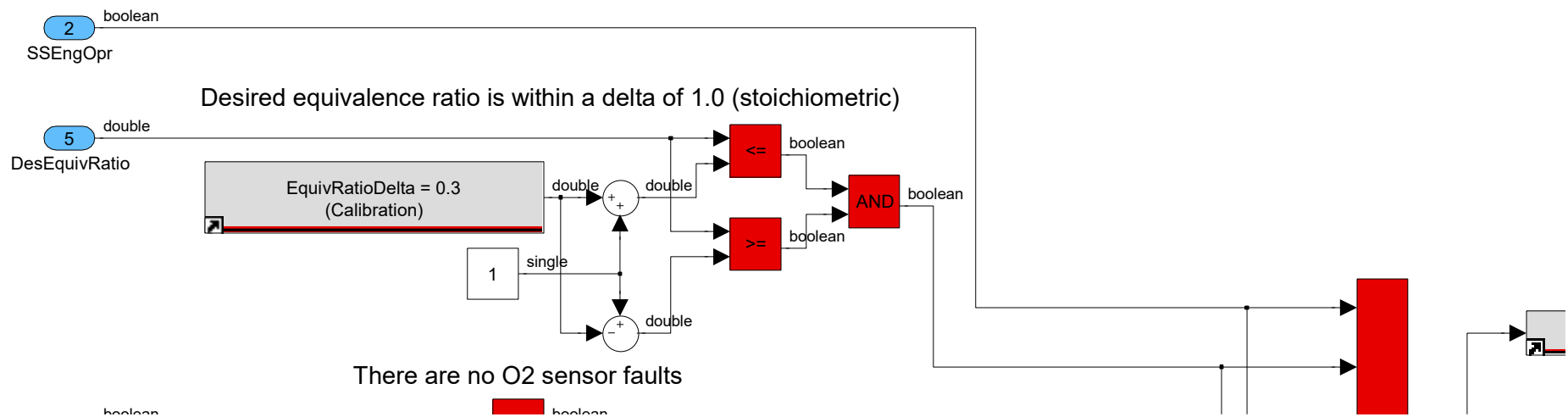




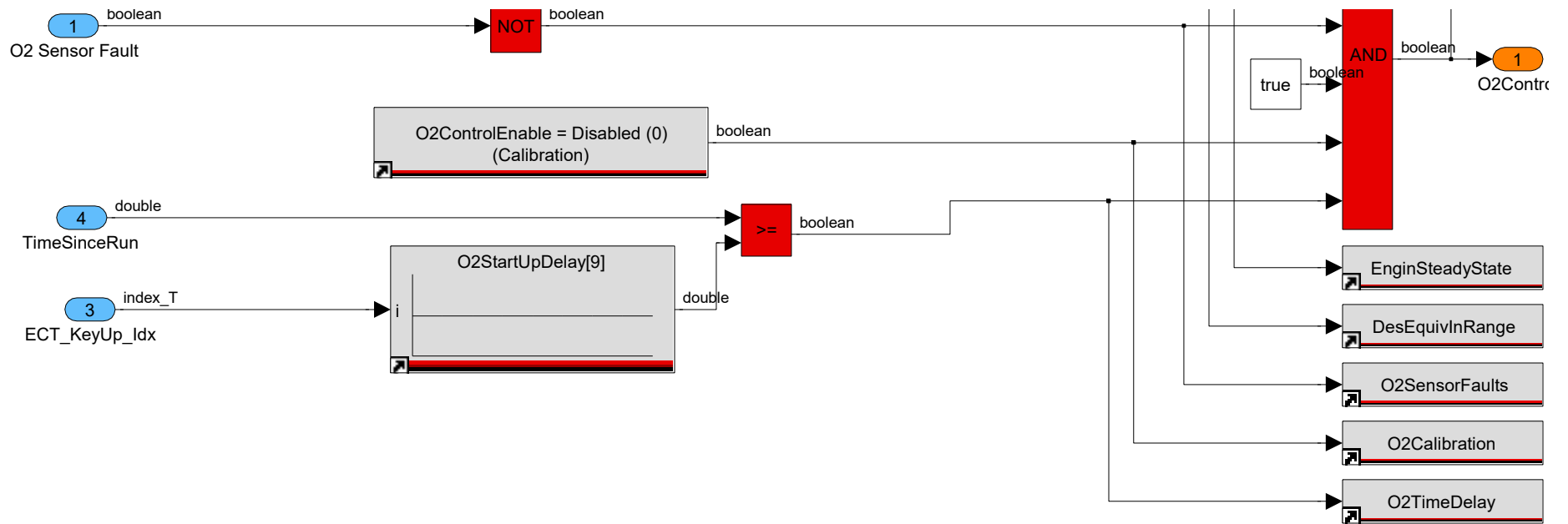




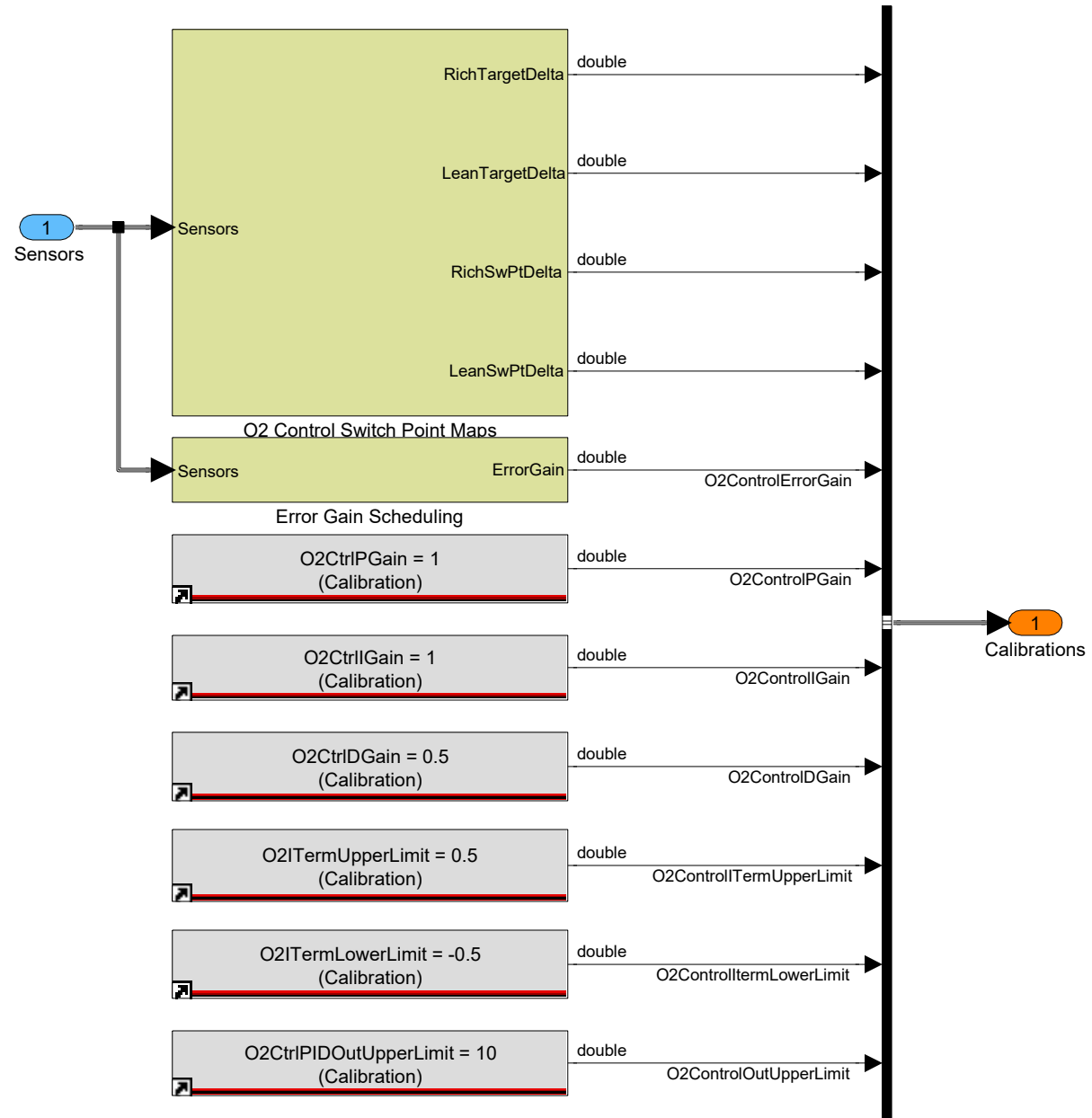


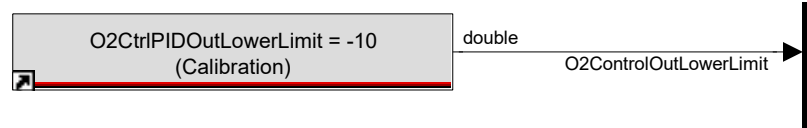


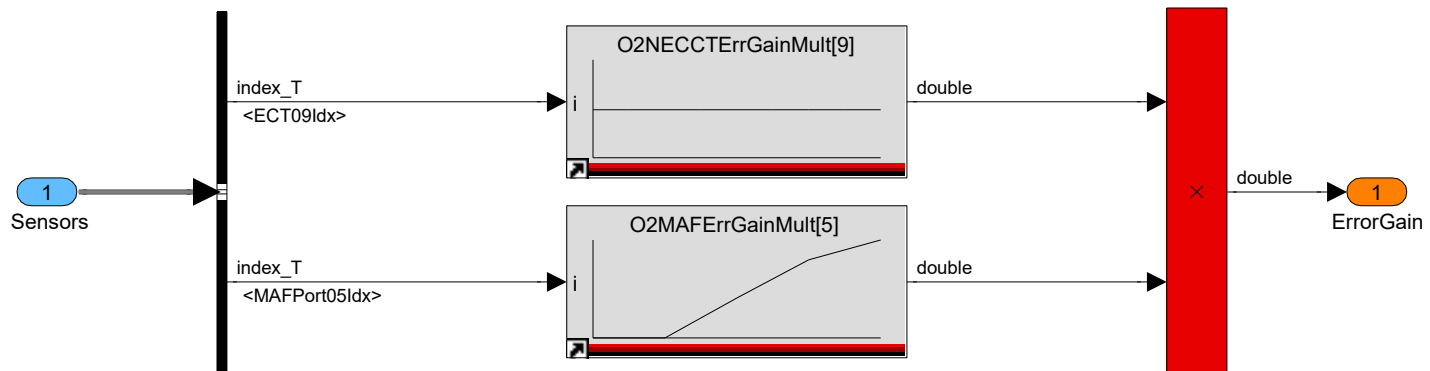
O2CtrlMode

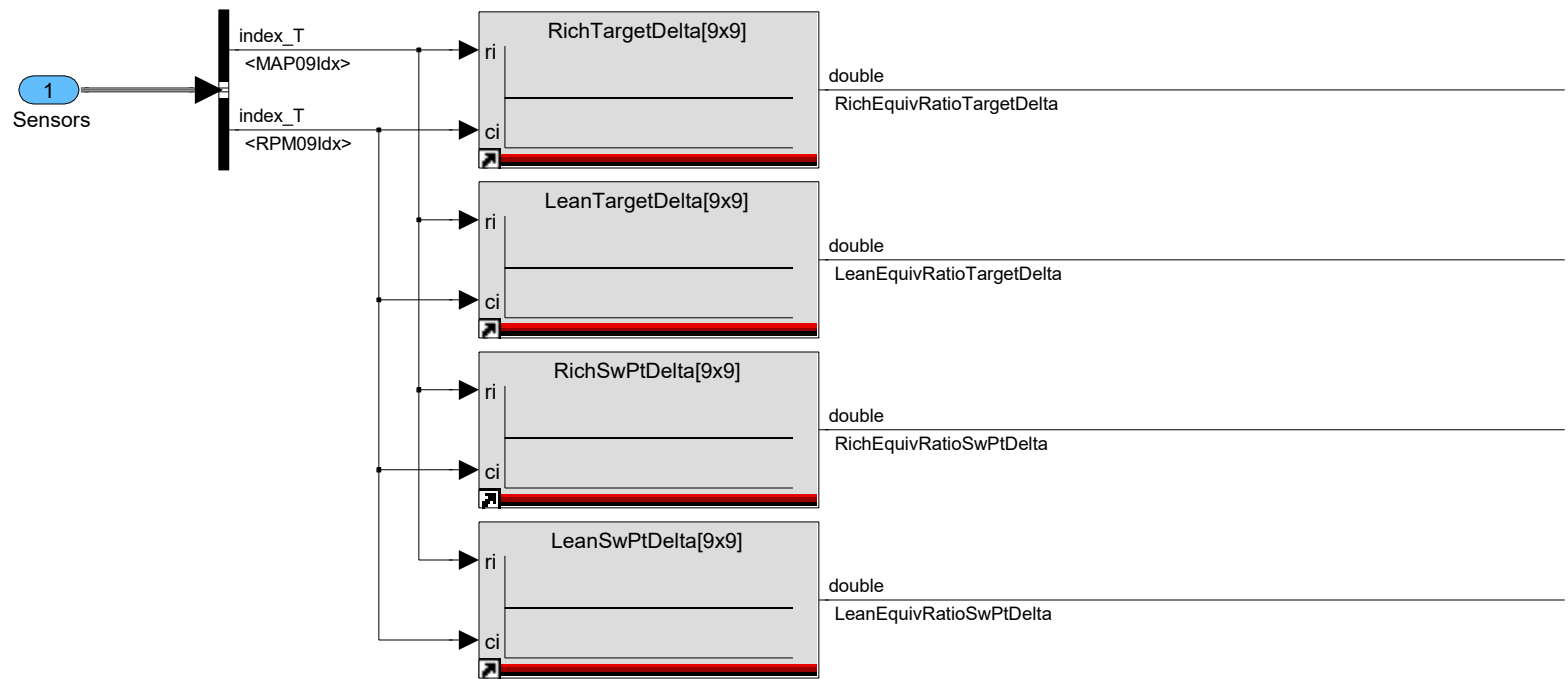


|  
|  
|  
|  
|











→ 1  
RichTargetDelta

→ 2  
LeanTargetDelta

→ 3  
RichSwPtDelta

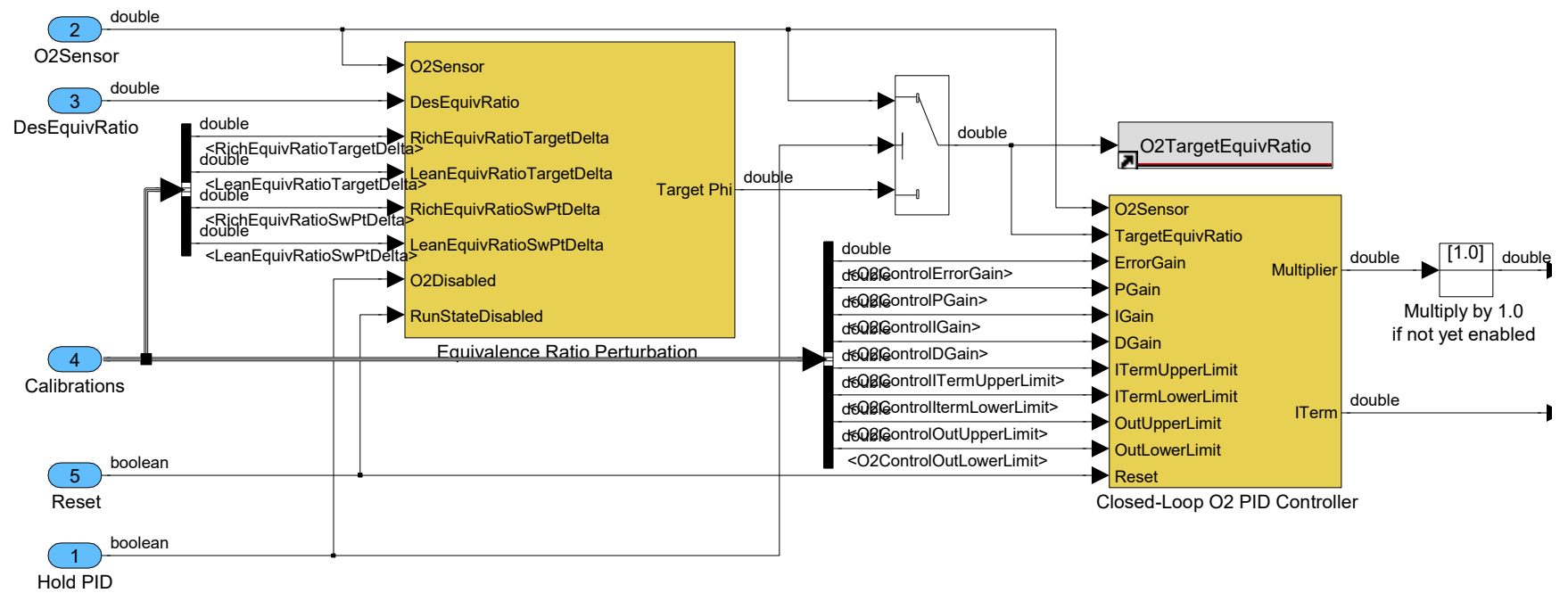
→ 4  
LeanSwPtDelta





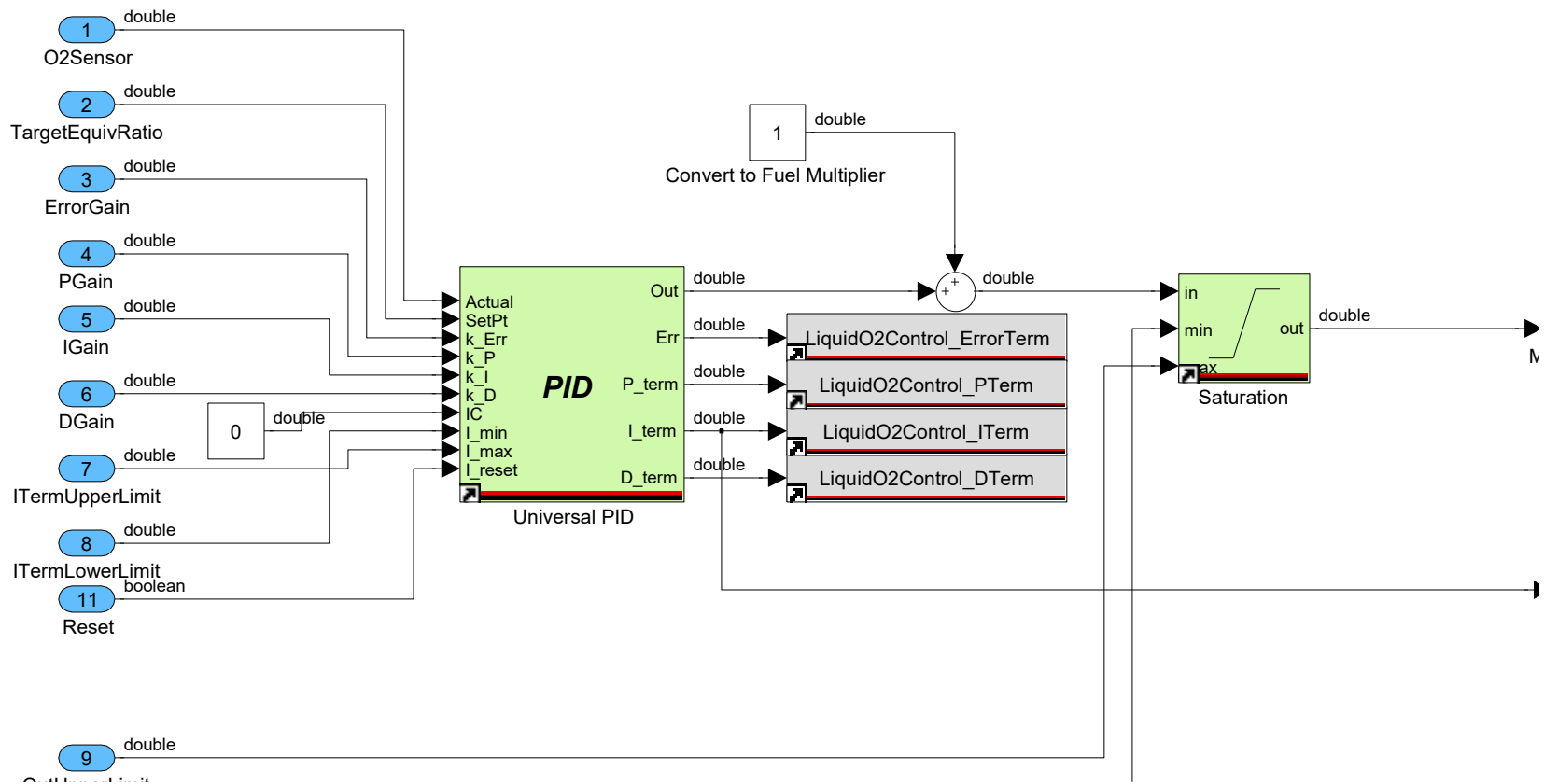






1  
FuelMult

2  
O2 Error



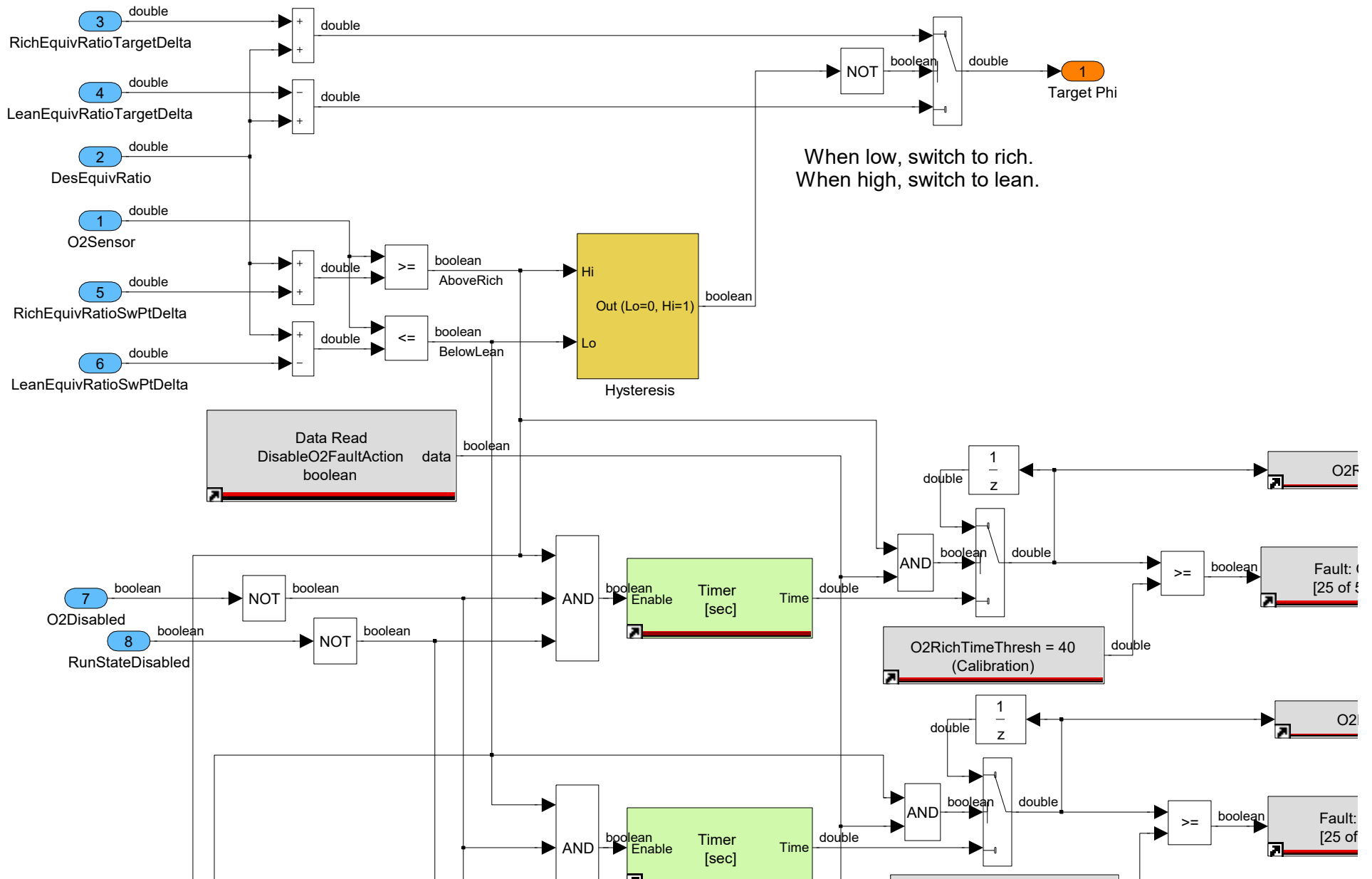


► 1  
Multiplier

► 2  
ITerm

OutUpperLimit  
double  
10  
OutLowerLimit



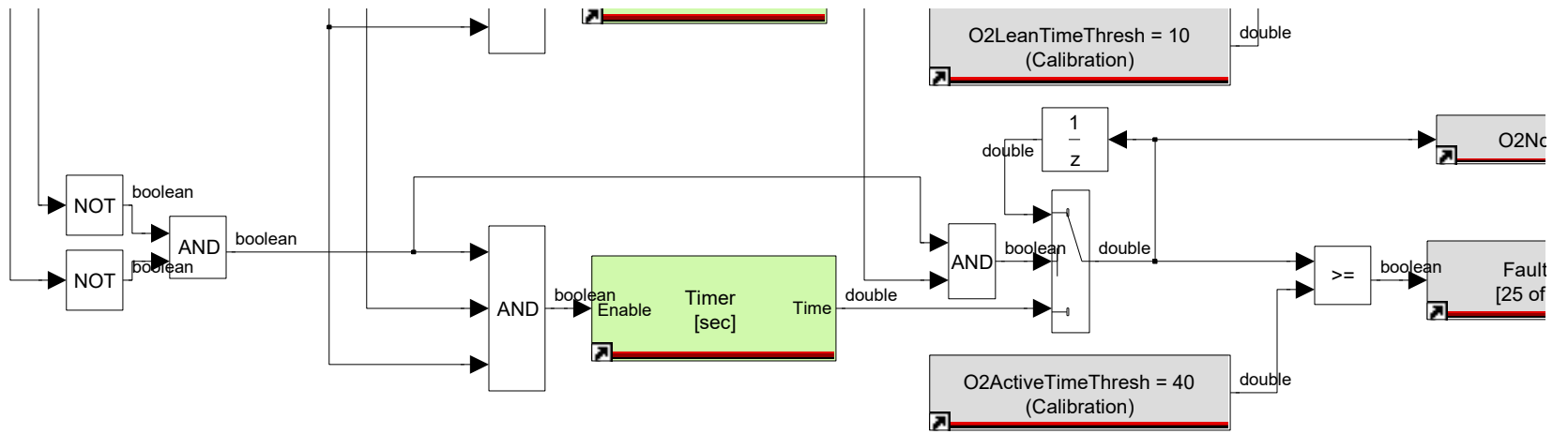


RichTime

O2FailedRich  
50] (Enabled)

?LeanTime

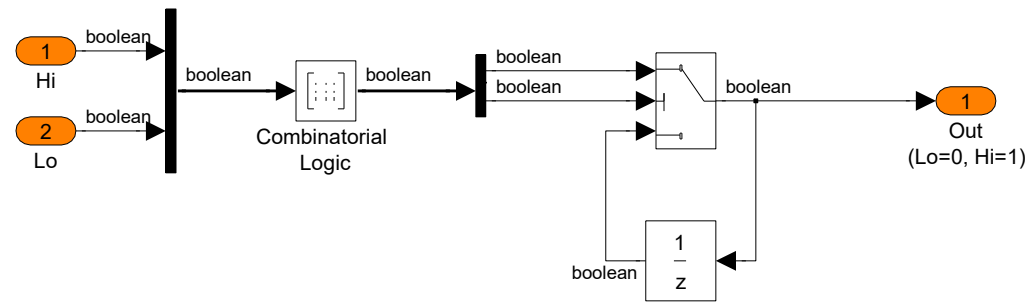
: O2FailedLean  
f 50] (Enabled)



otActiveTime

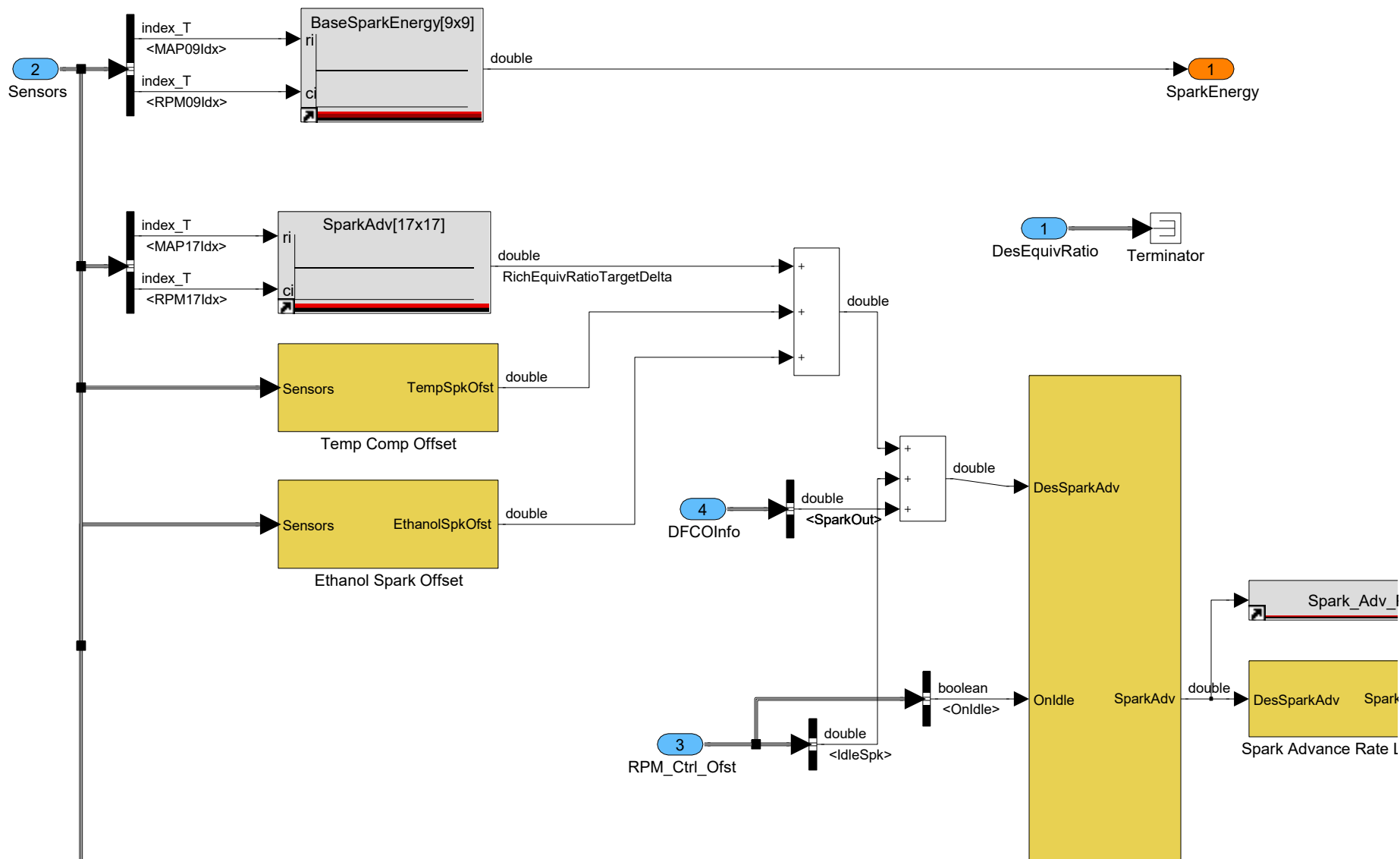
lt: O2NotActive  
f 50] (Enabled)

**Hi & !Lo => Hi**  
**Lo & !Hi => Lo**  
**Otherwise, Hold**

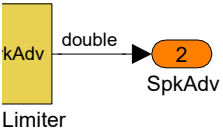




f()

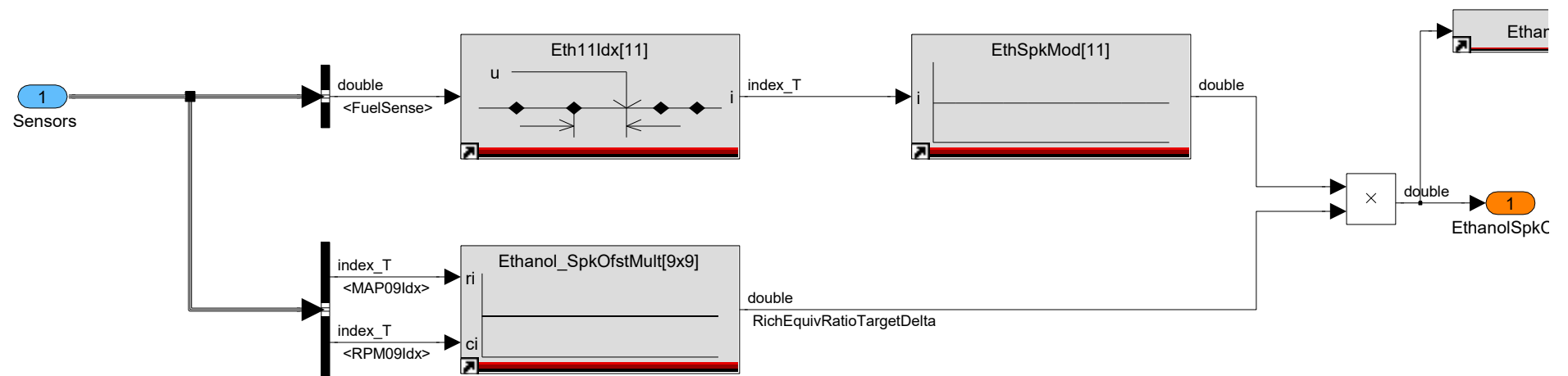


Prb



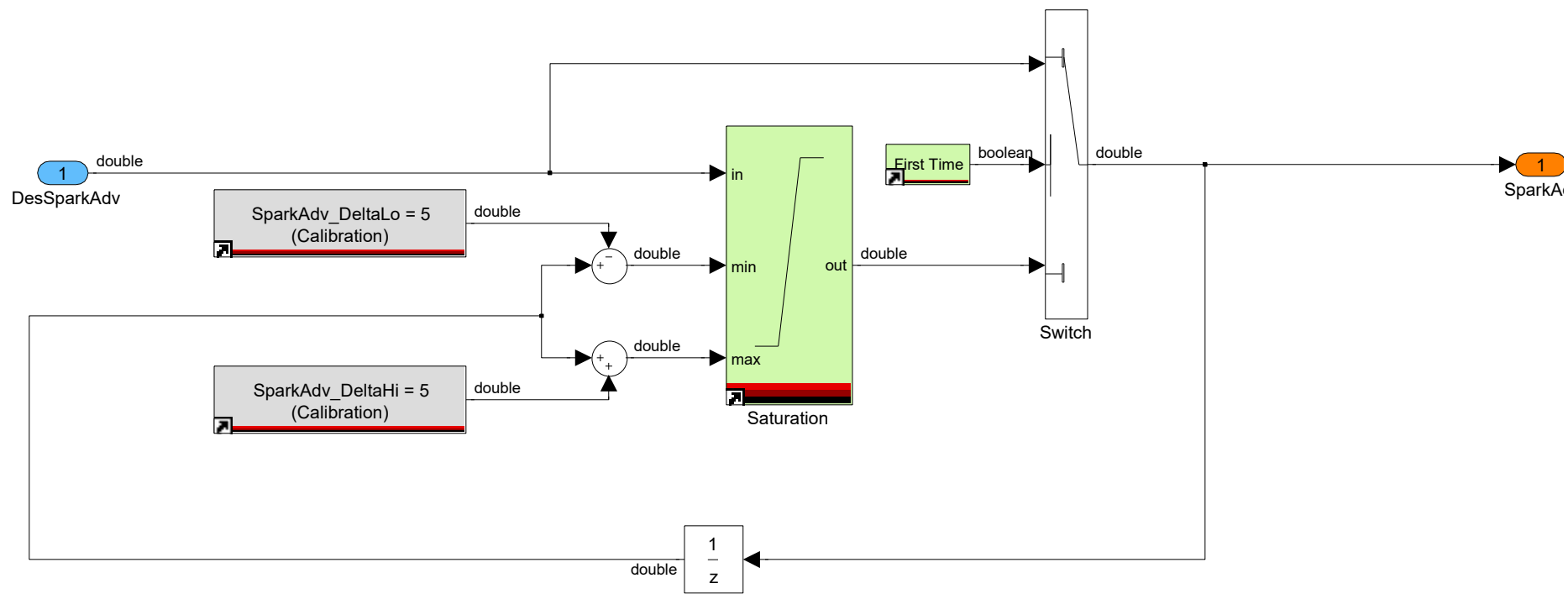






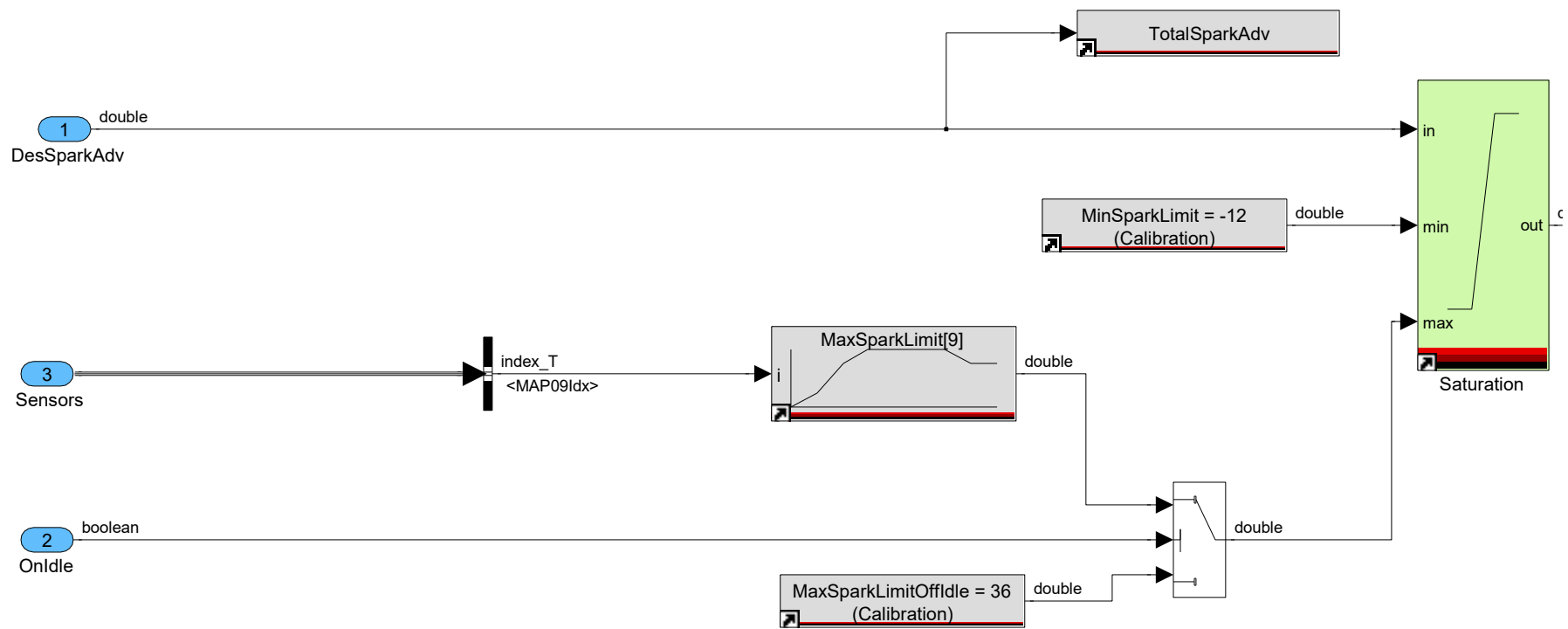
noI\_Spk\_Ofst

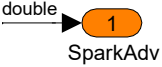
Ofst

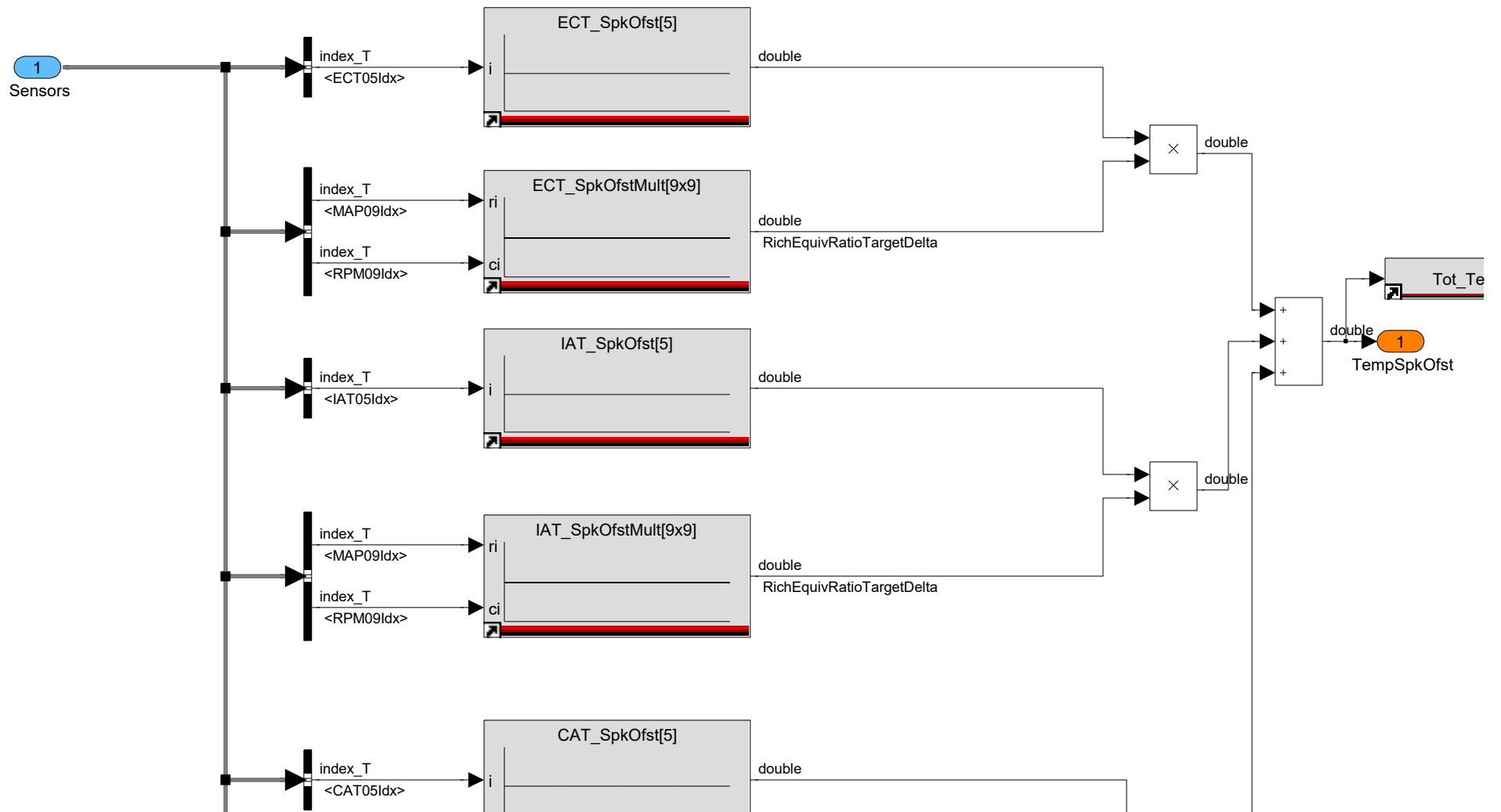




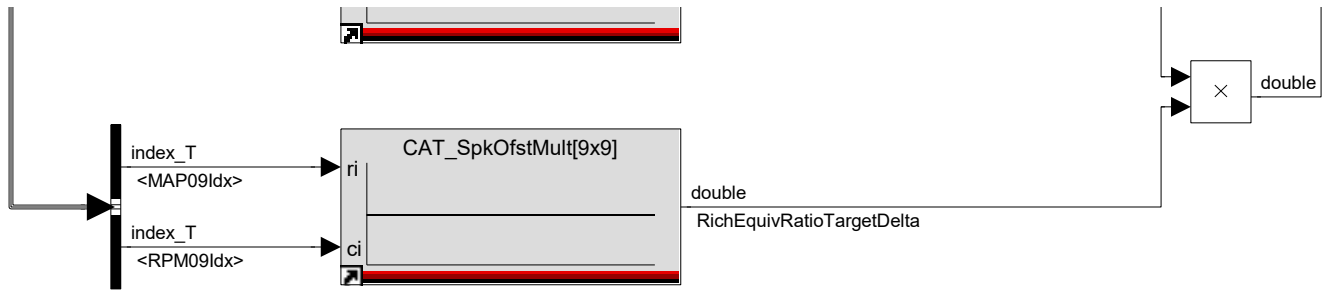






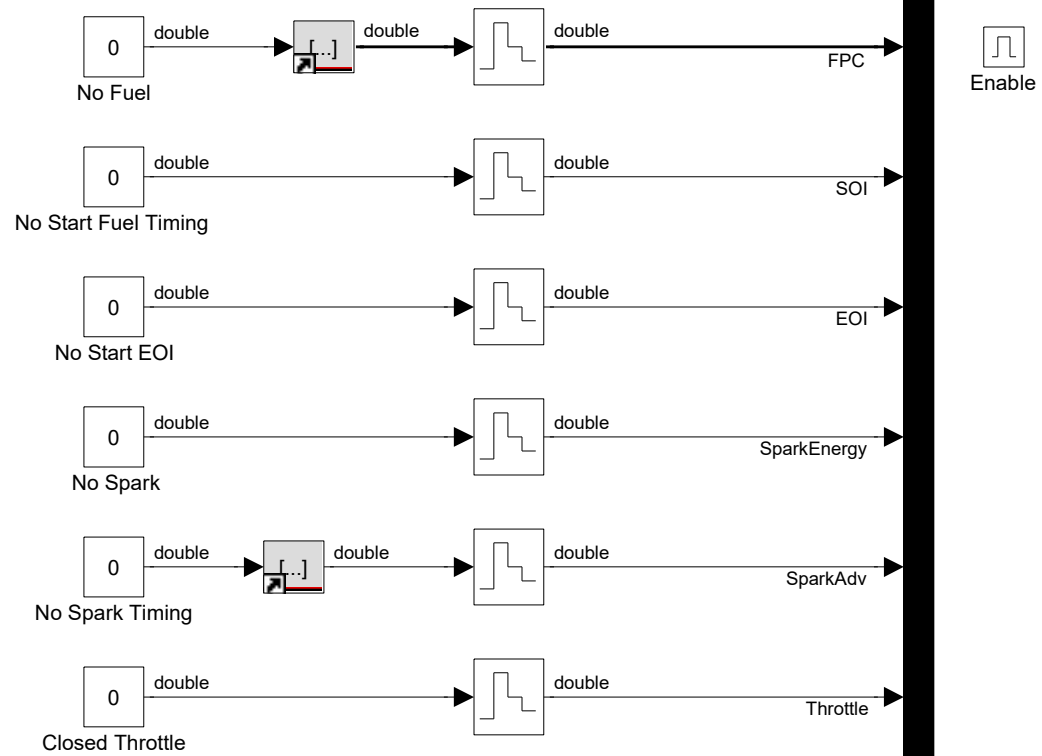


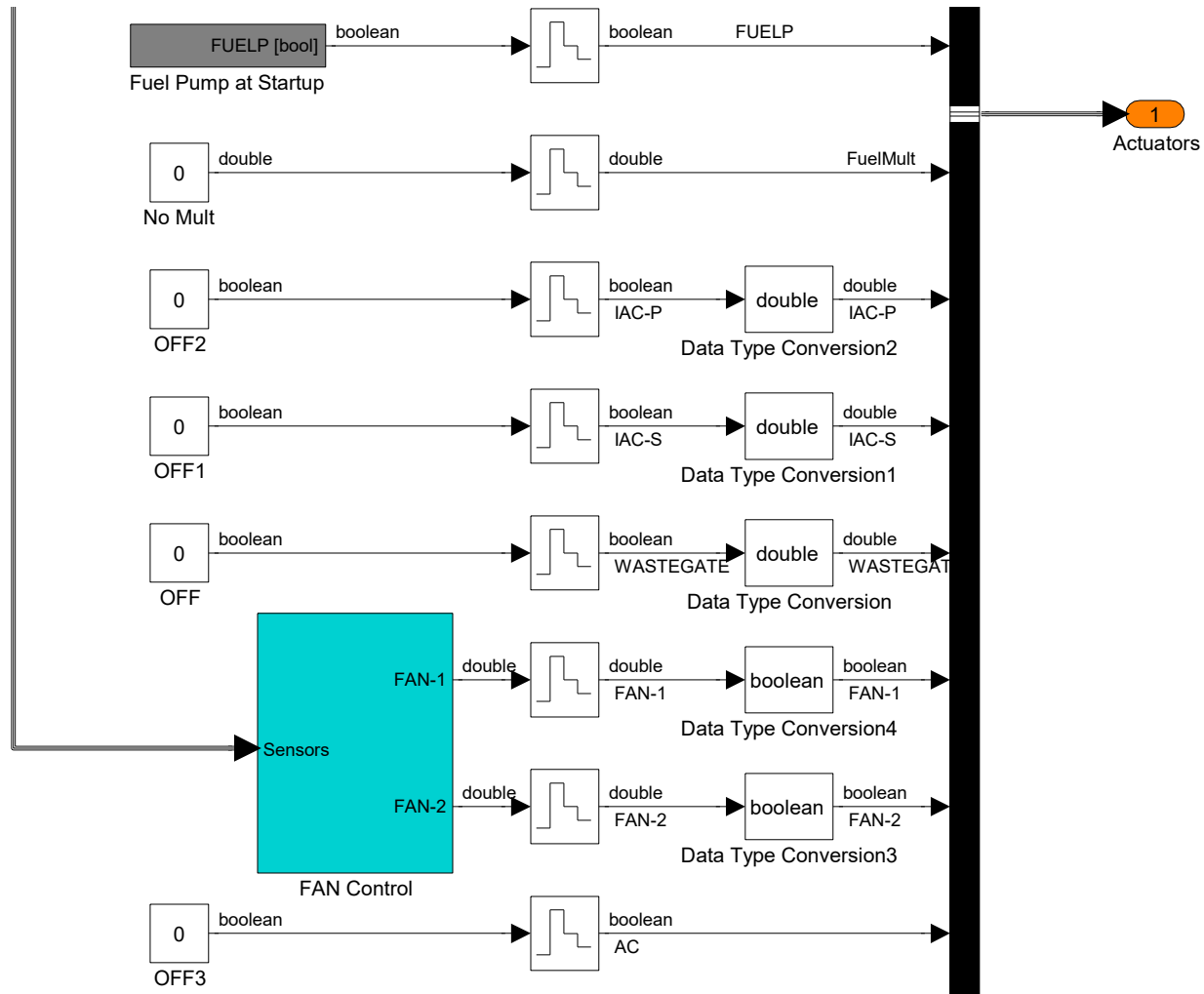
amp\_SpkOfst



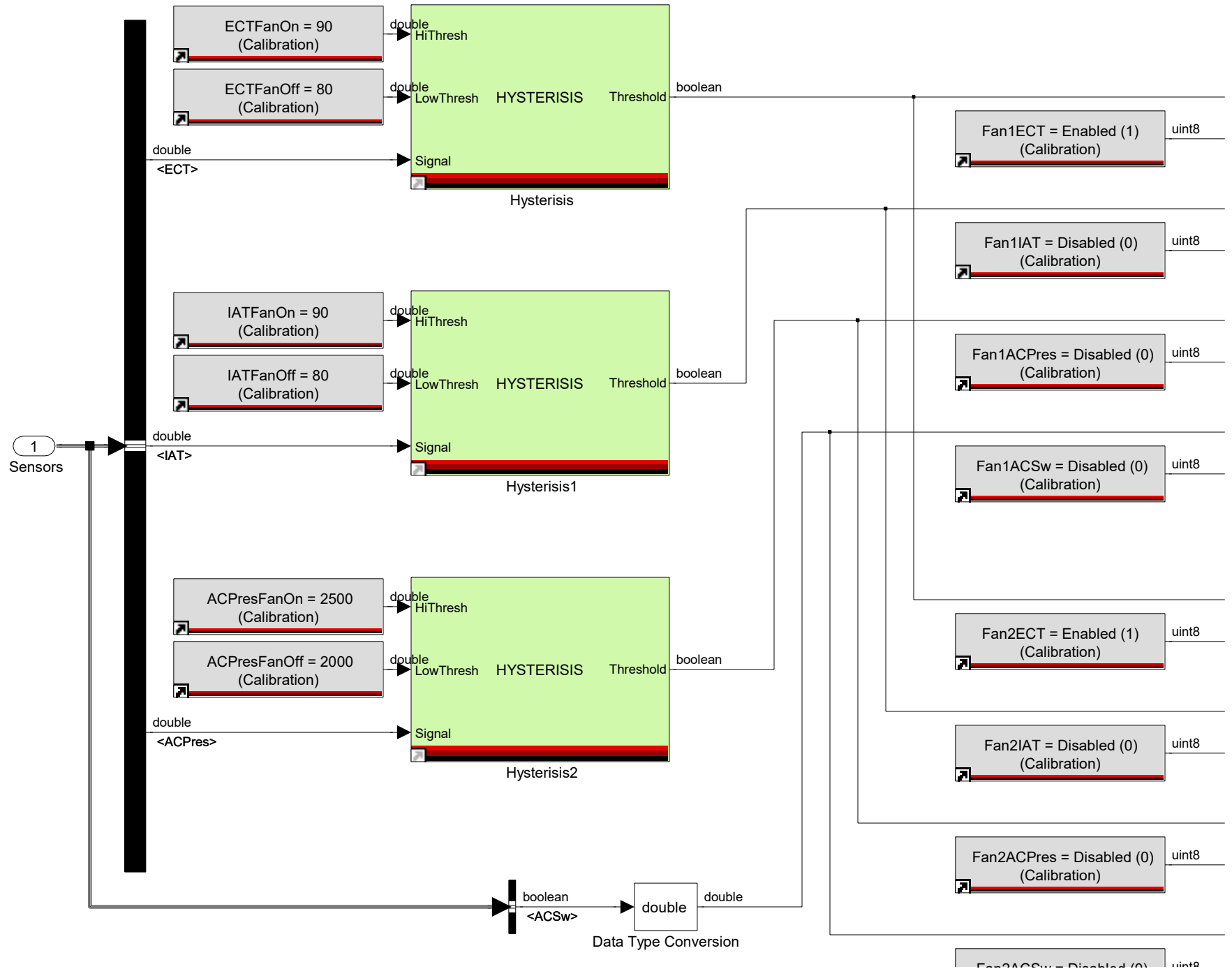


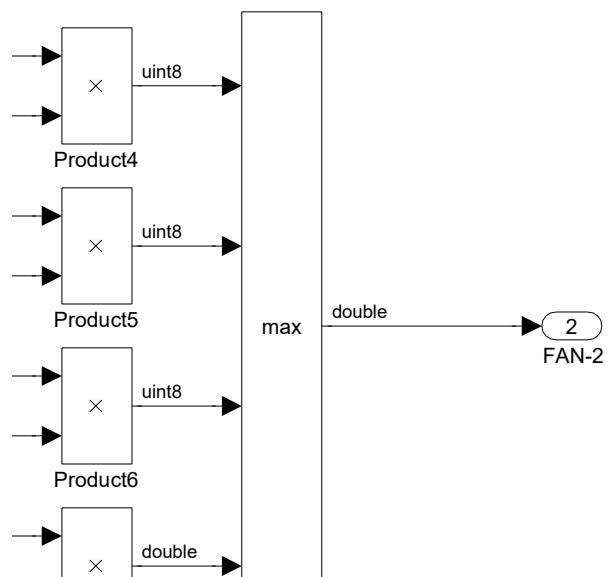
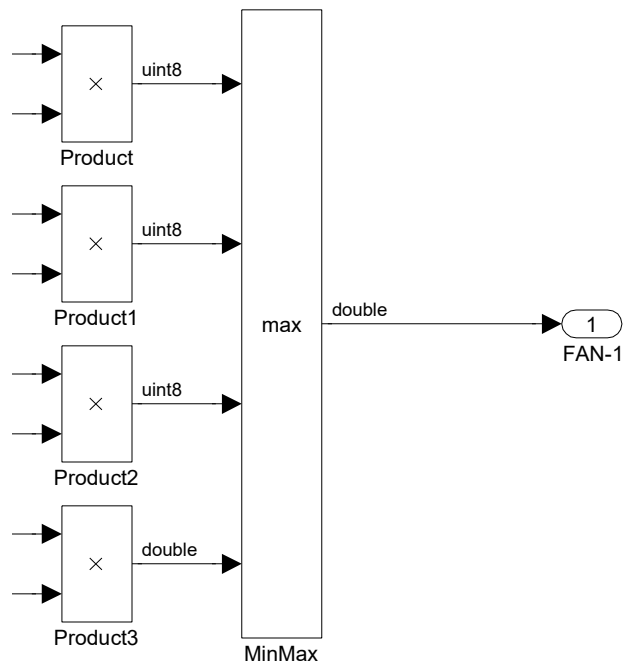
1  
Inputs



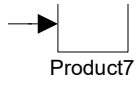


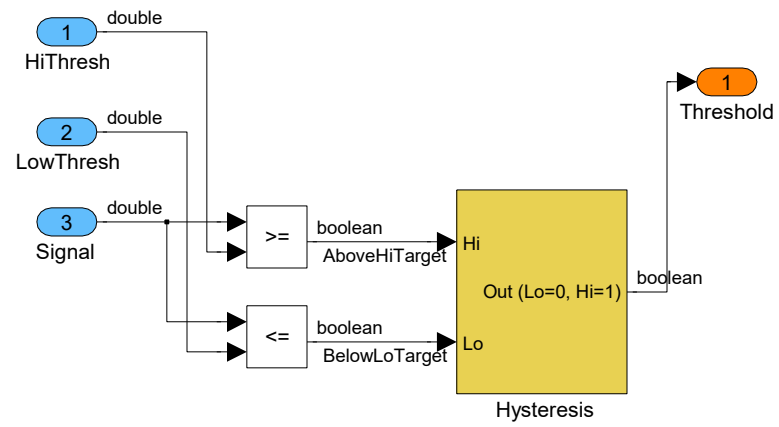




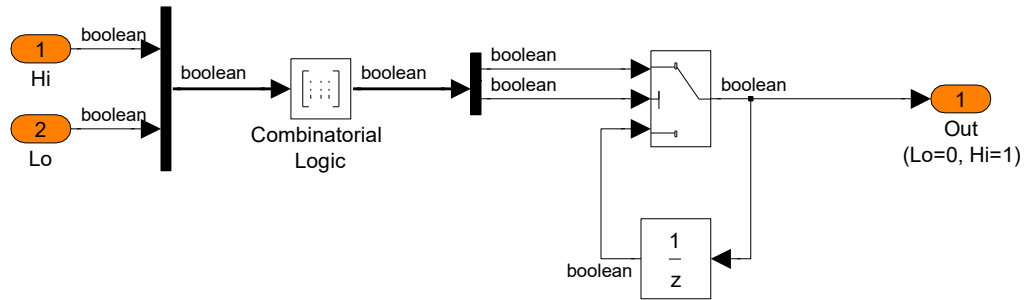


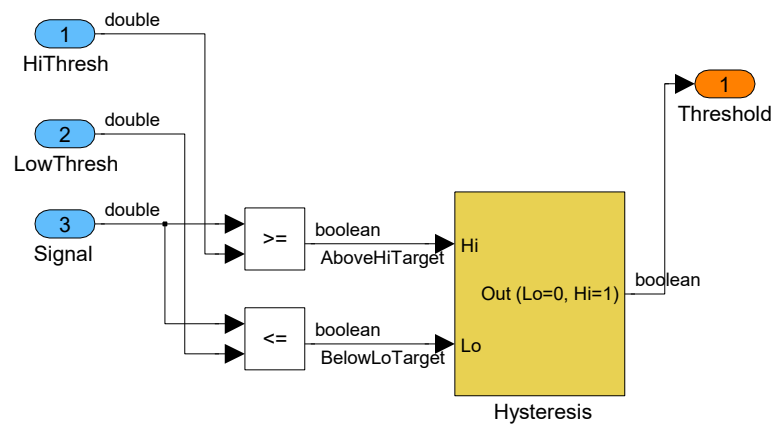
Fan2ACSw = Disabled (0) (Calibration)	uint8
--	-------



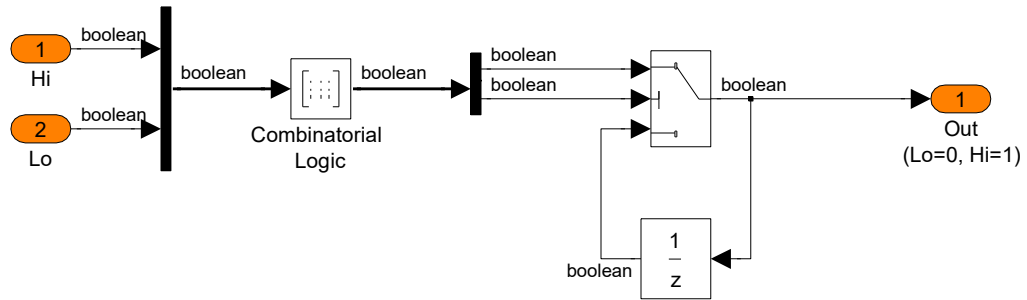


**Hi & !Lo => Hi**  
**Lo & !Hi => Lo**  
**Otherwise, Hold**

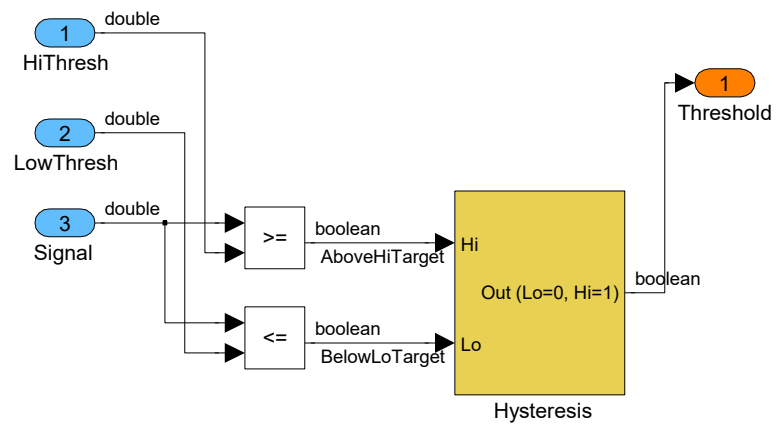




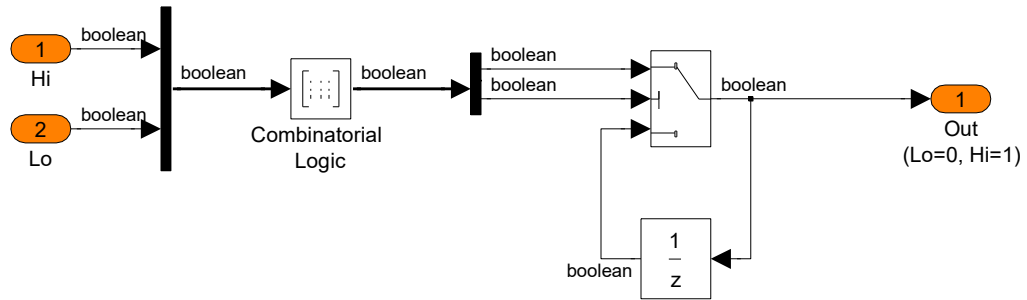
**Hi & !Lo => Hi**  
**Lo & !Hi => Lo**  
**Otherwise, Hold**

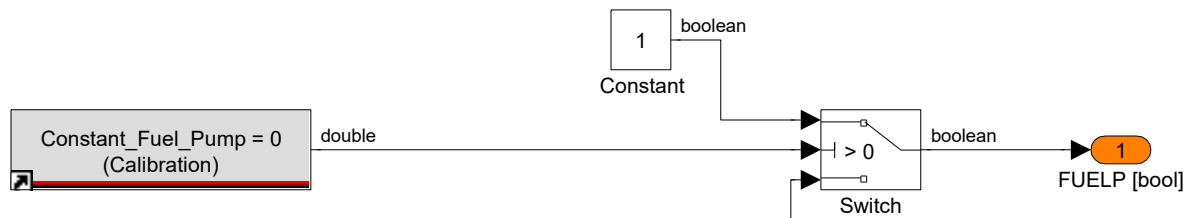




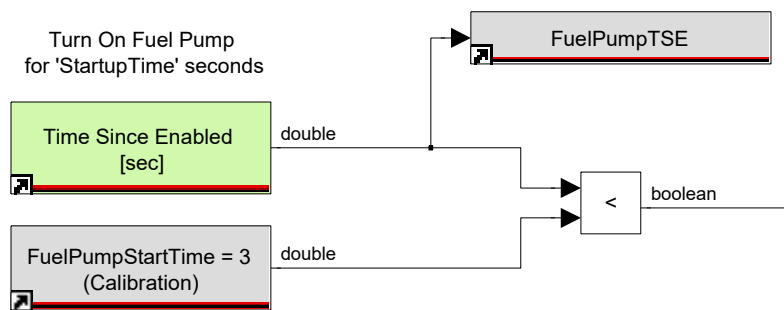


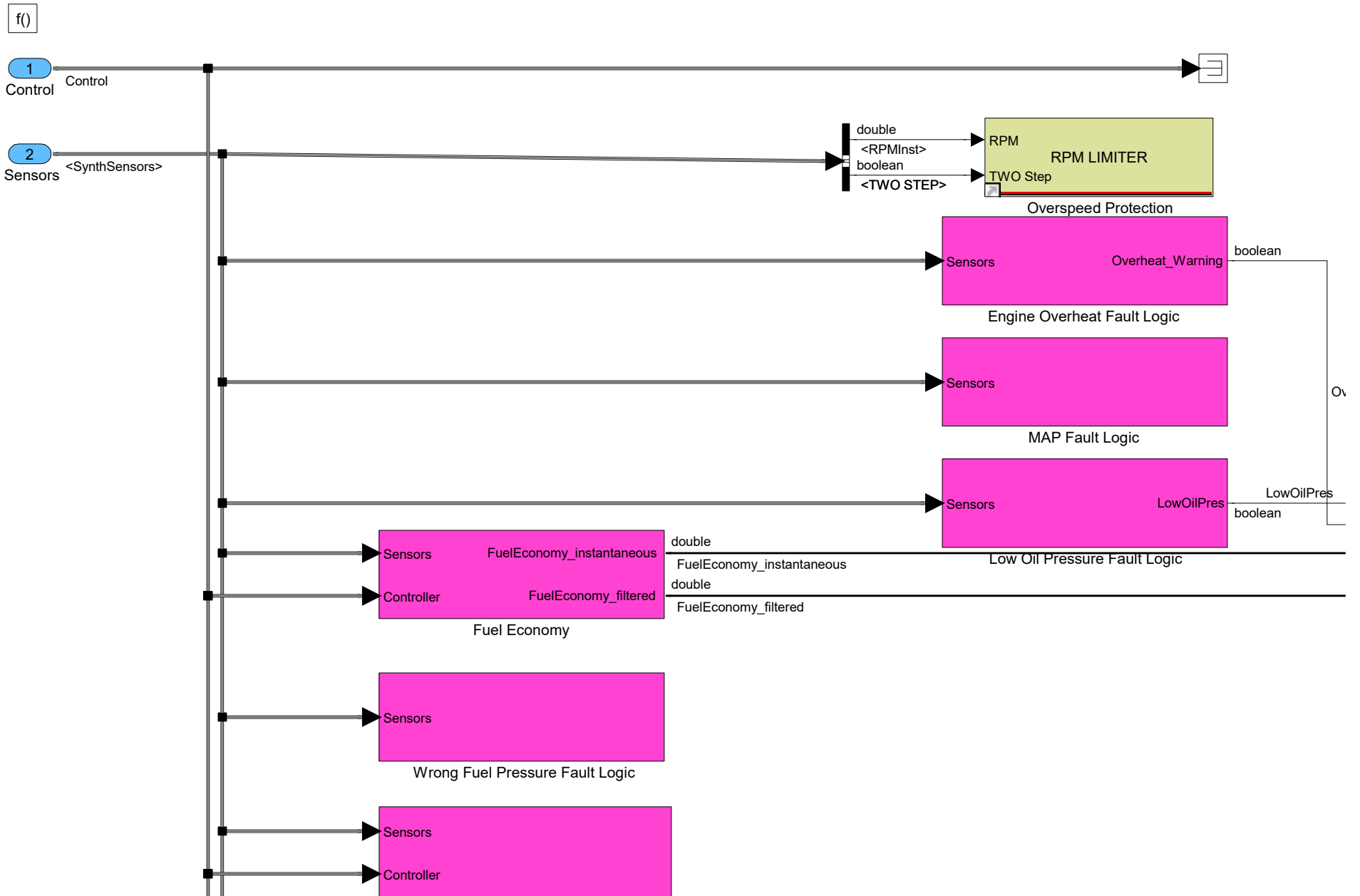
**Hi & !Lo => Hi**  
**Lo & !Hi => Lo**  
**Otherwise, Hold**



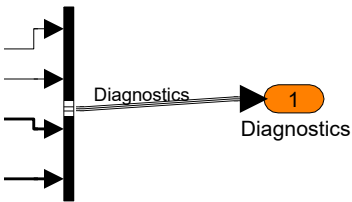


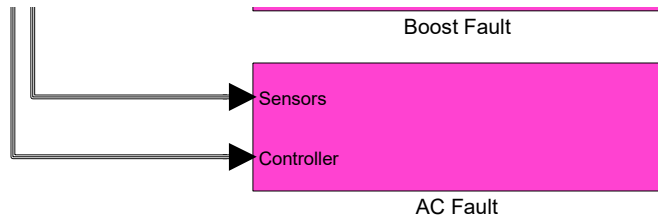
## Turn on Fuel Pump at Startup



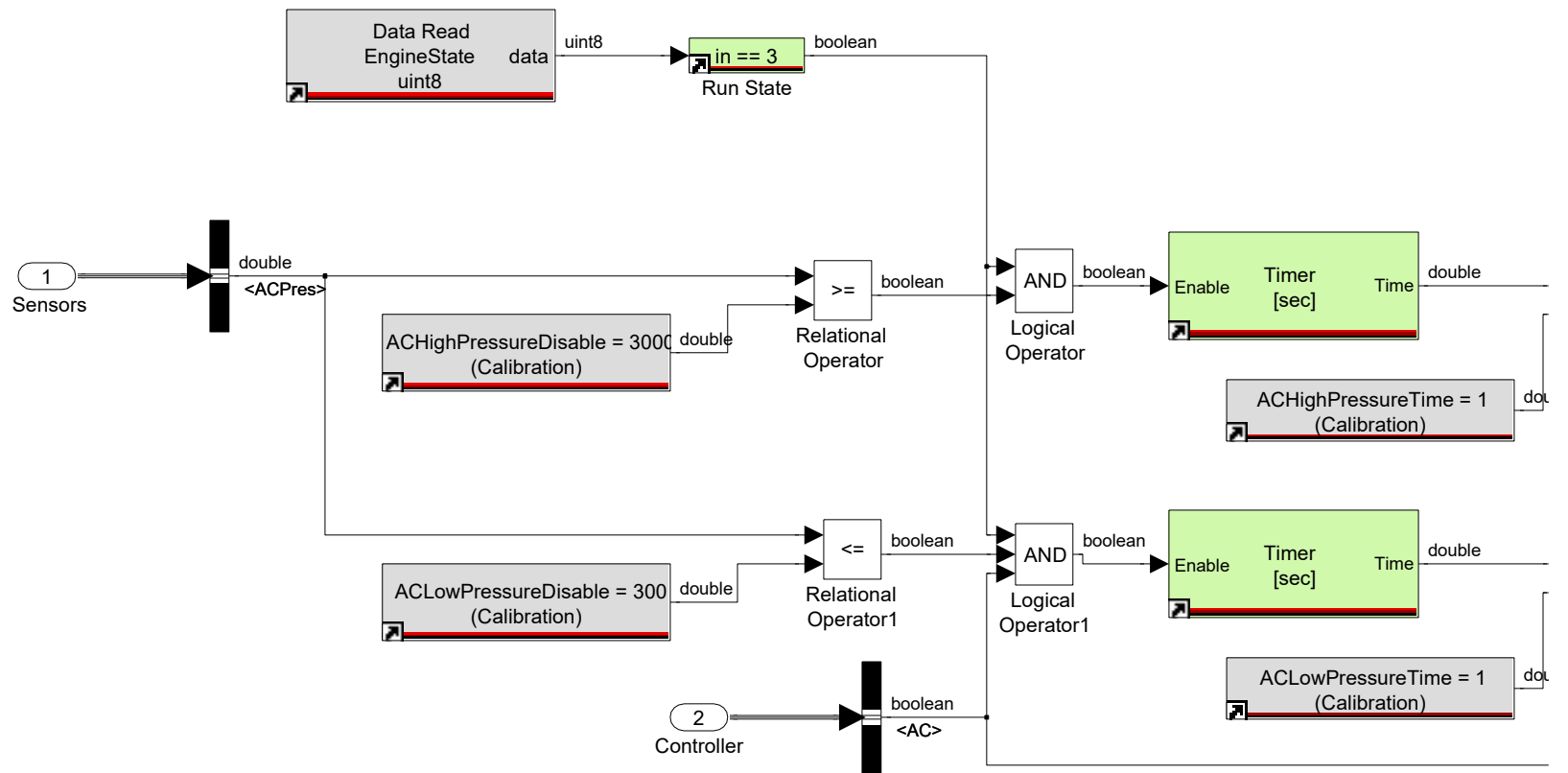


verheat\_Warning

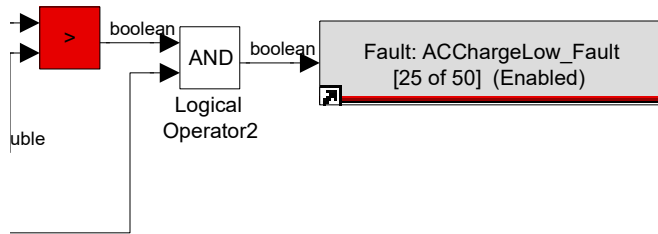
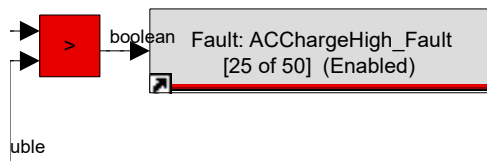


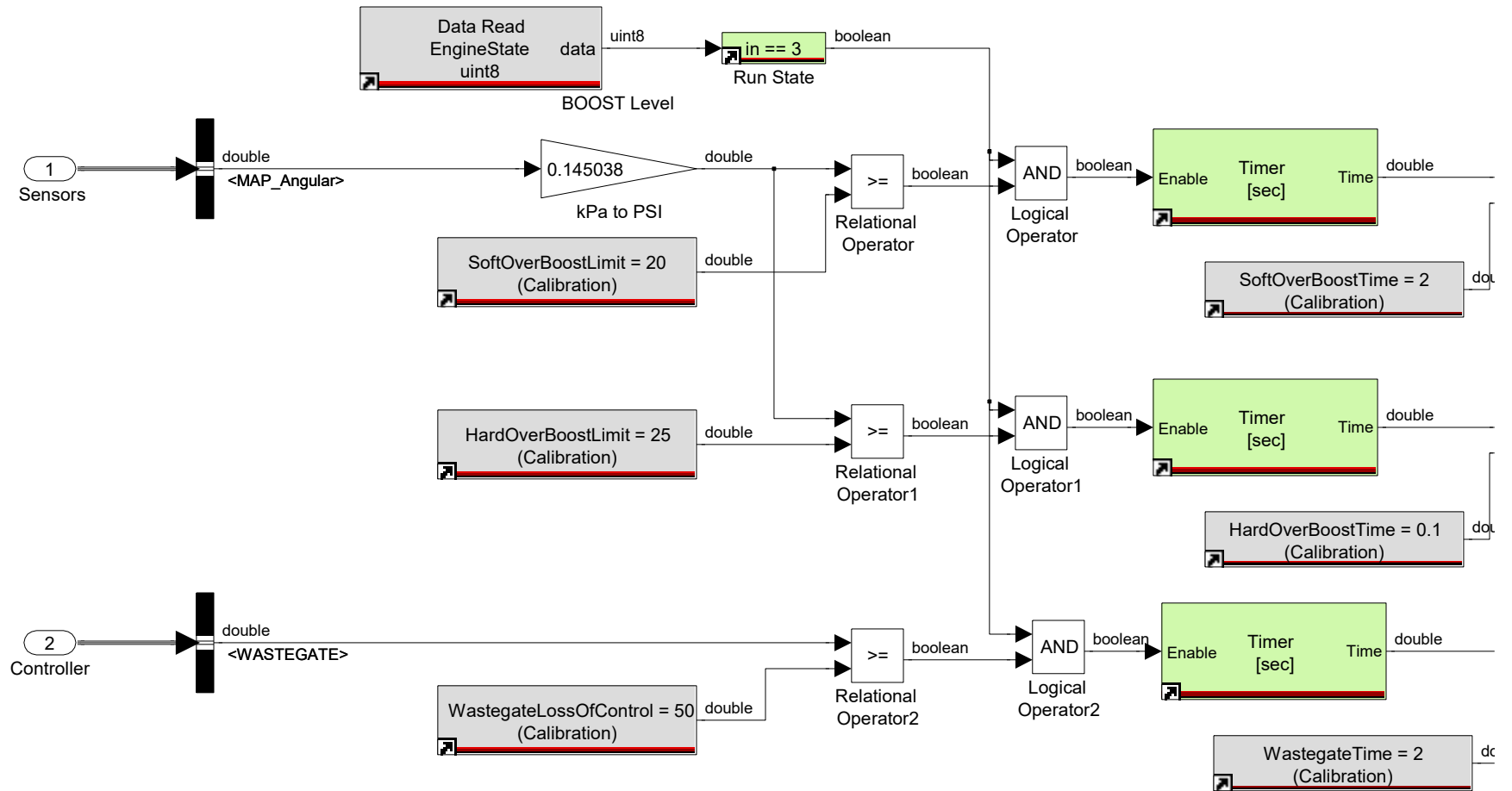


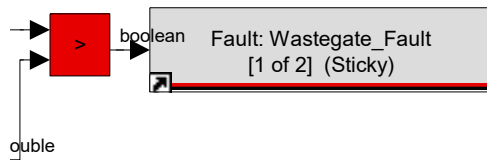
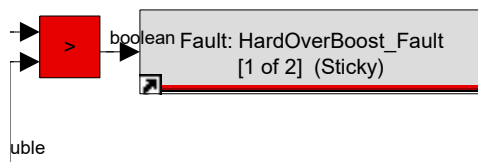
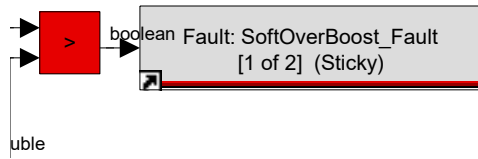


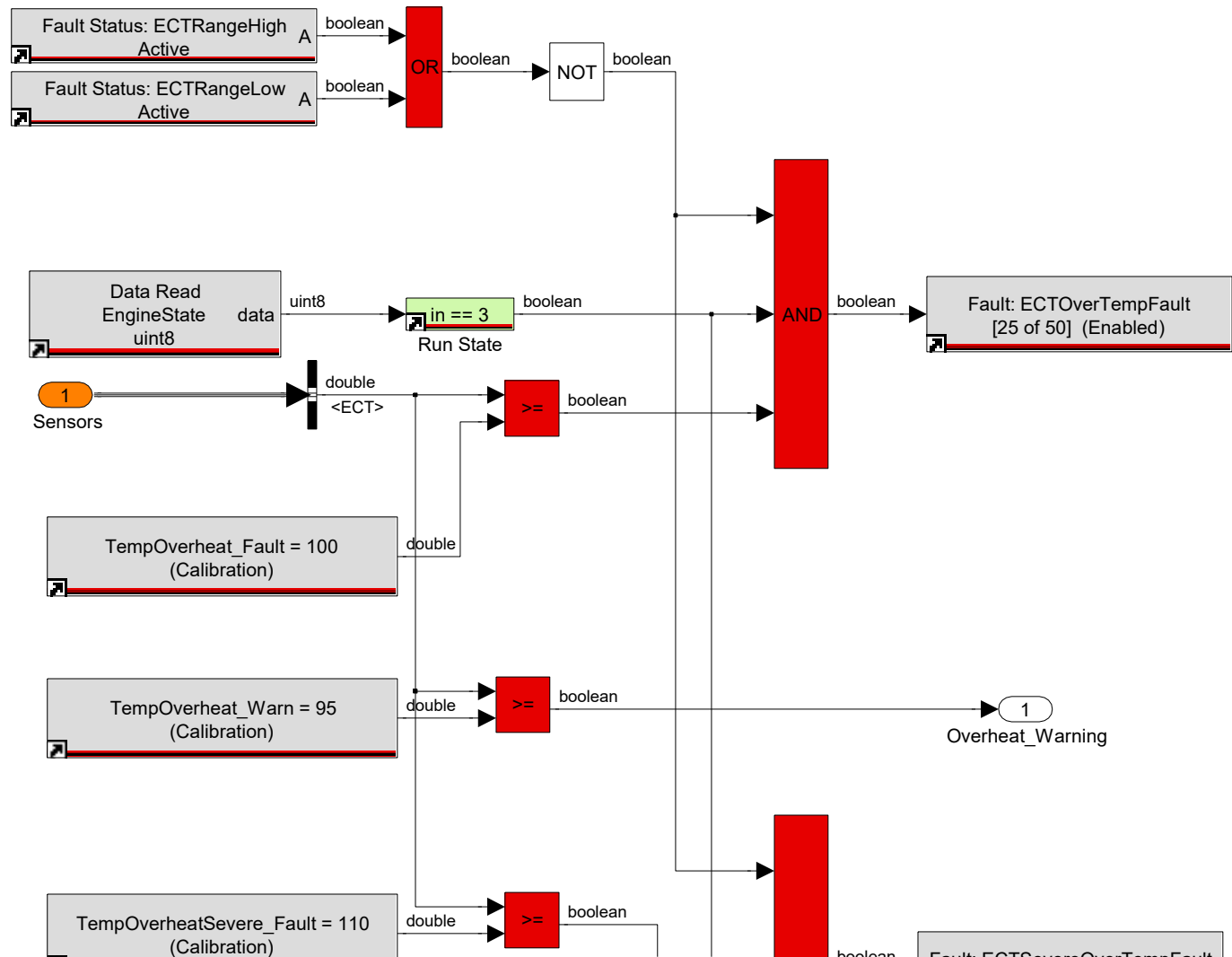


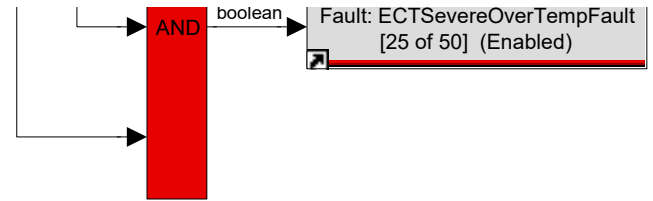


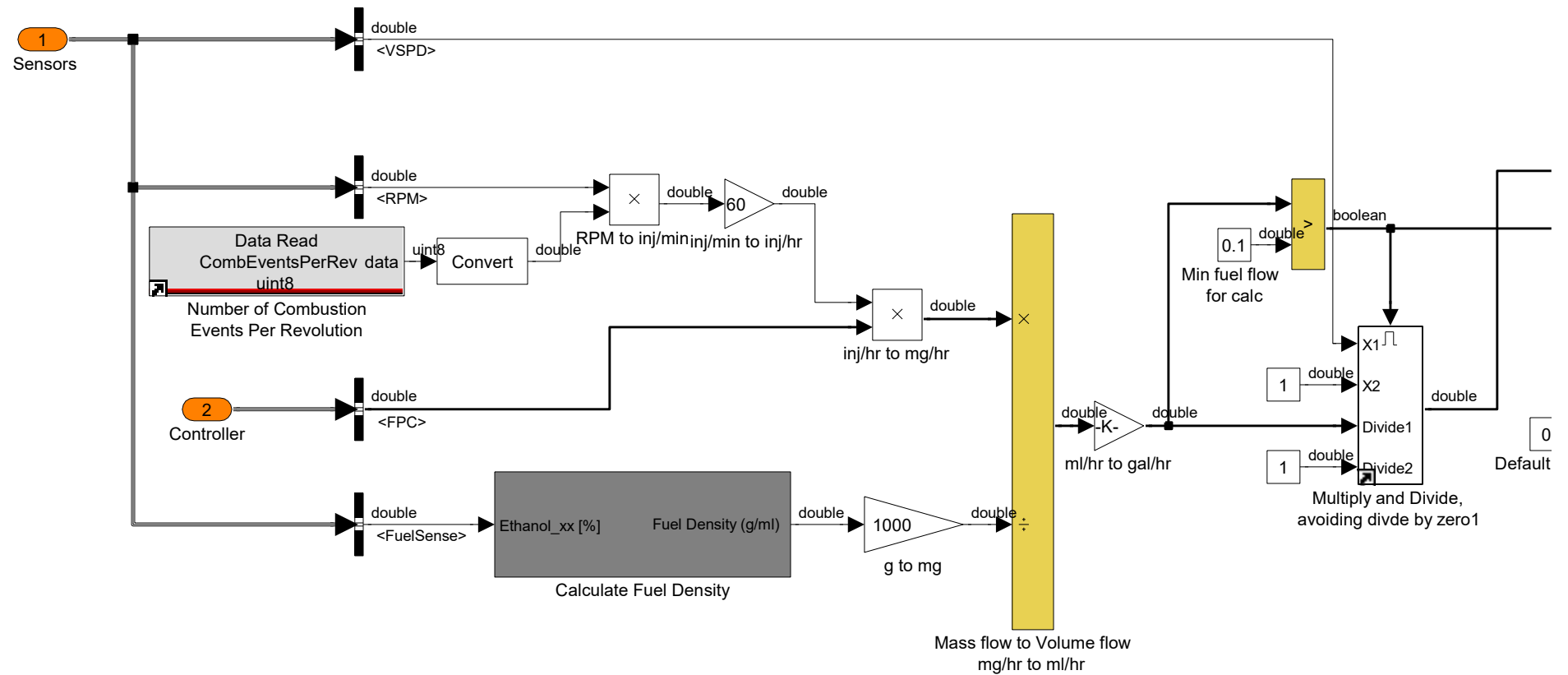


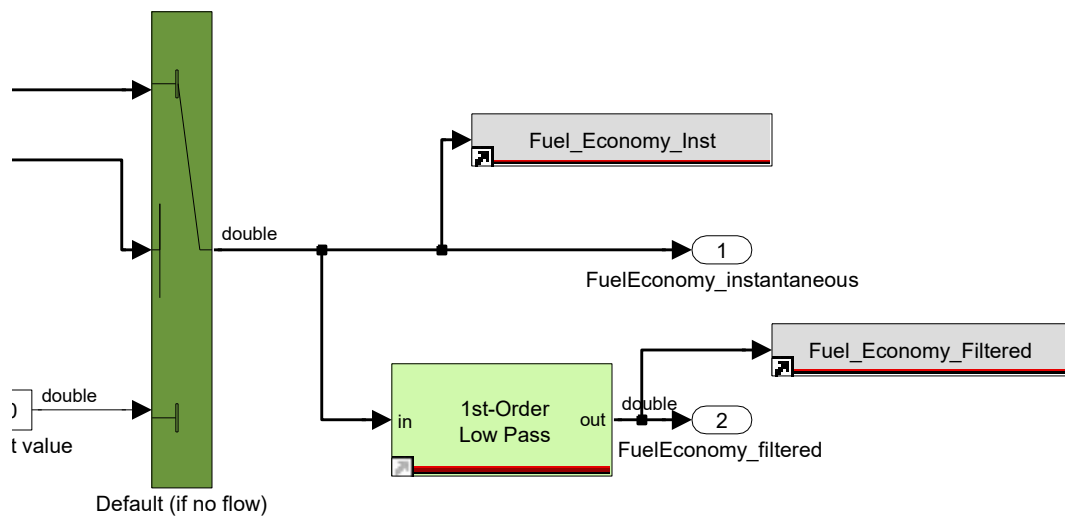


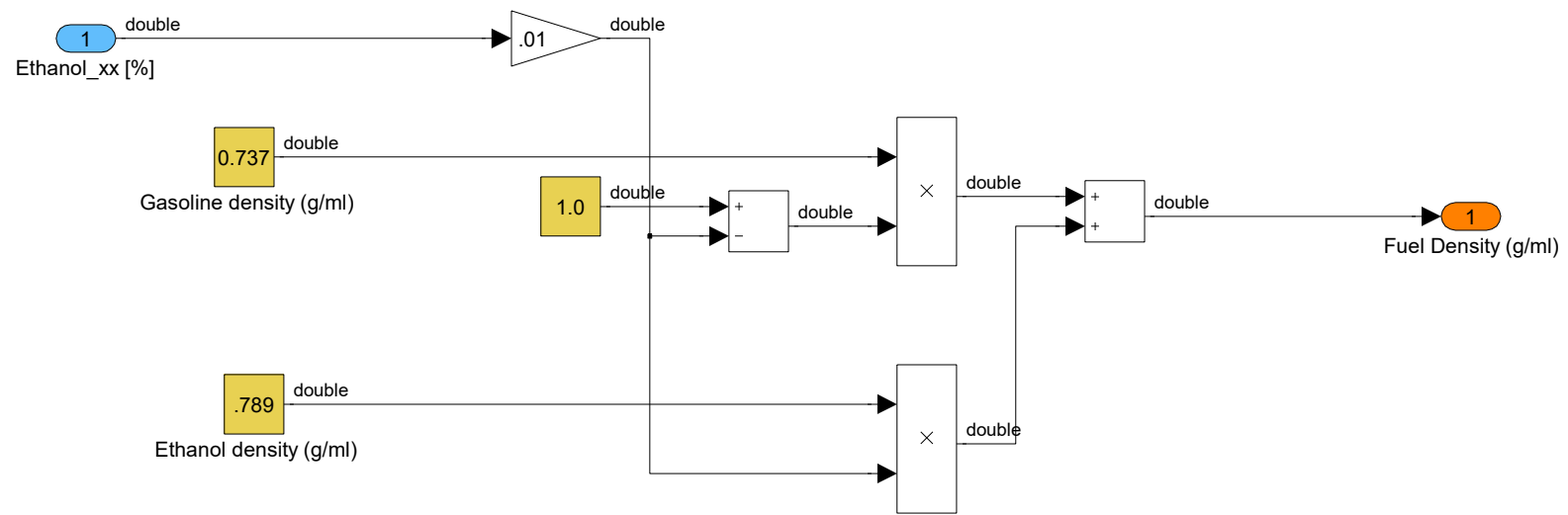






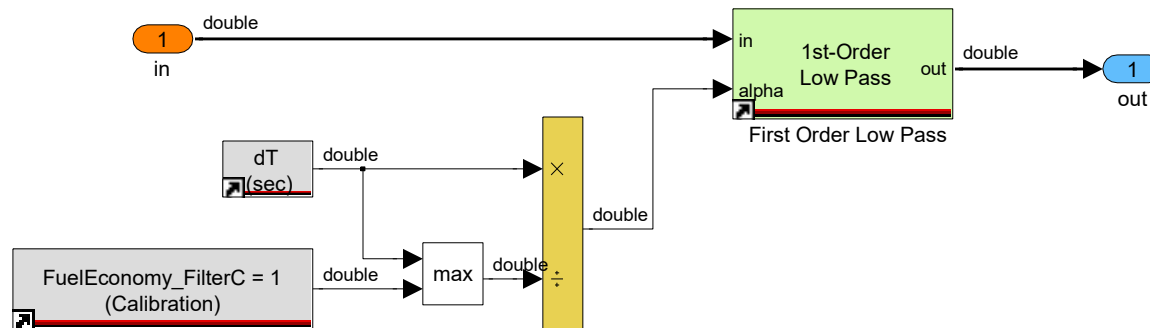


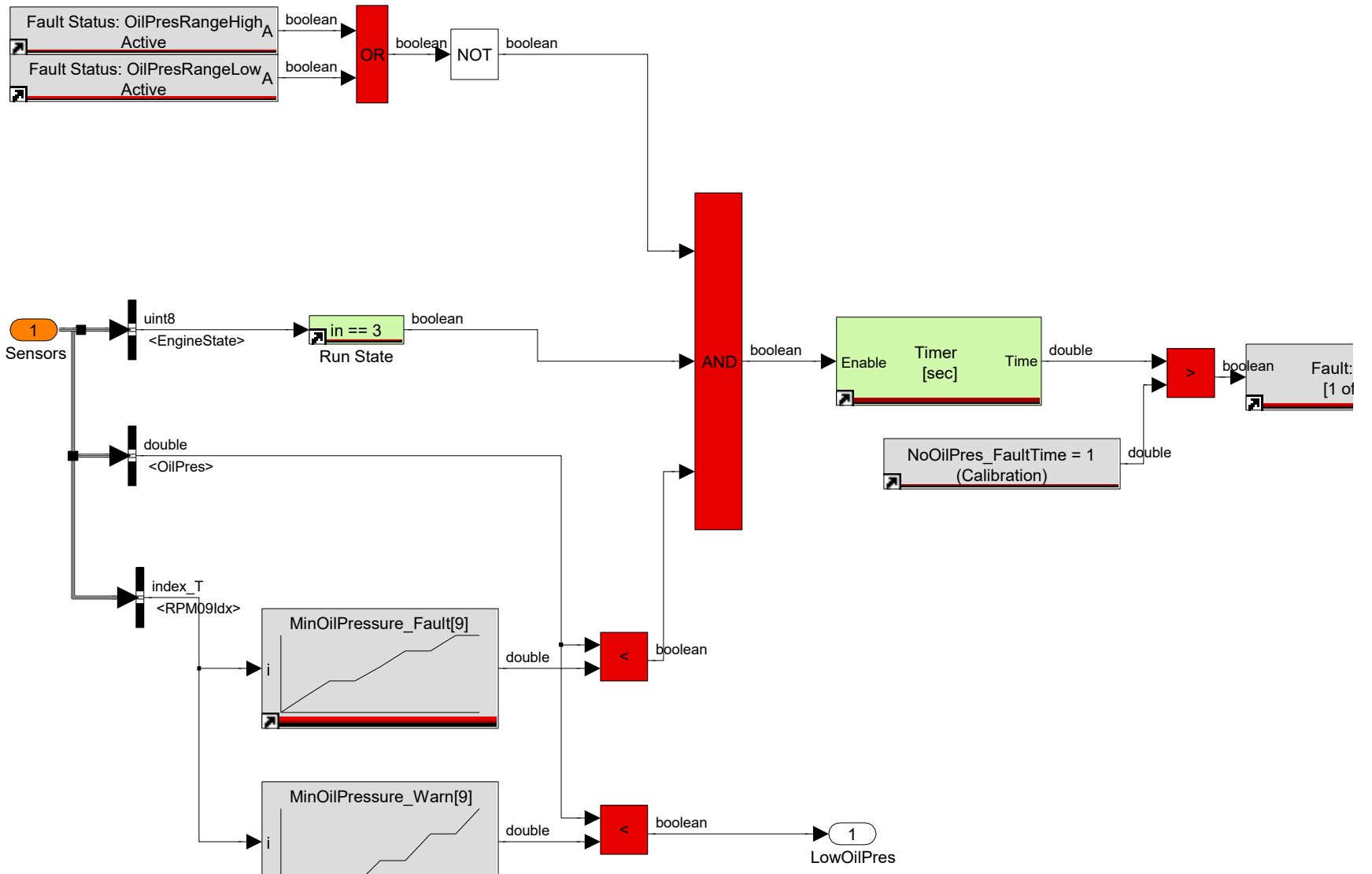






## Tunable 1st-Order Low-Pass Filter

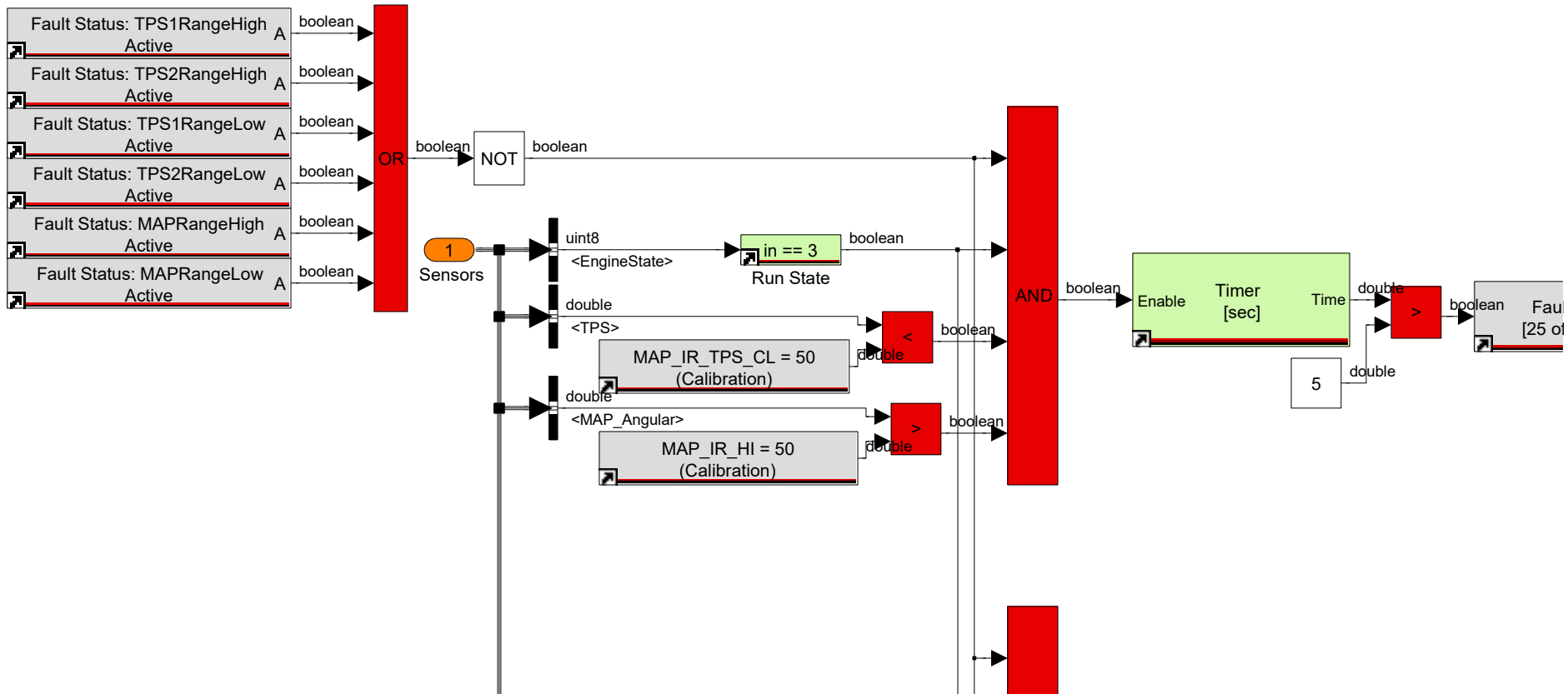




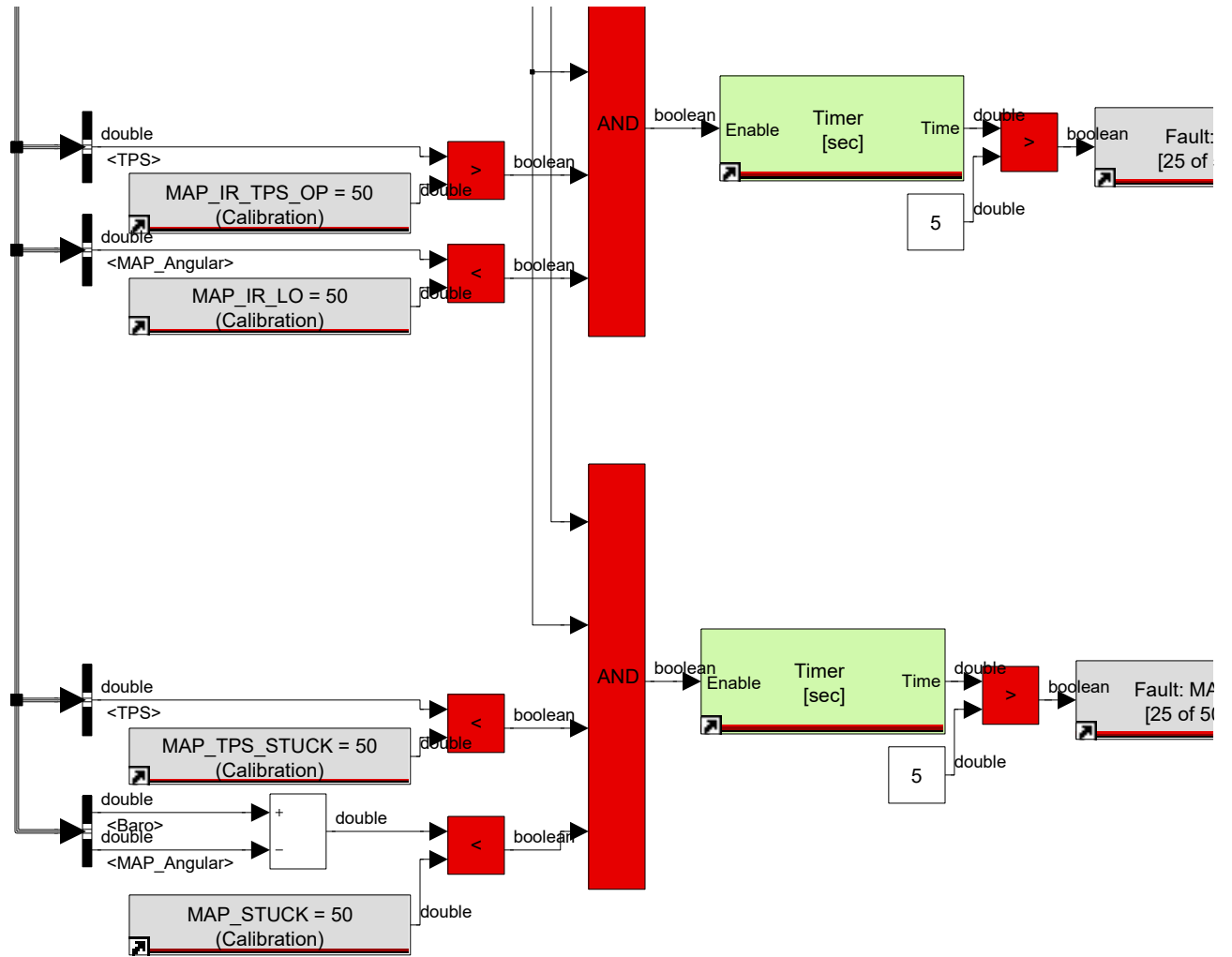
: NoOil\_Fault  
if 2] (Sticky)







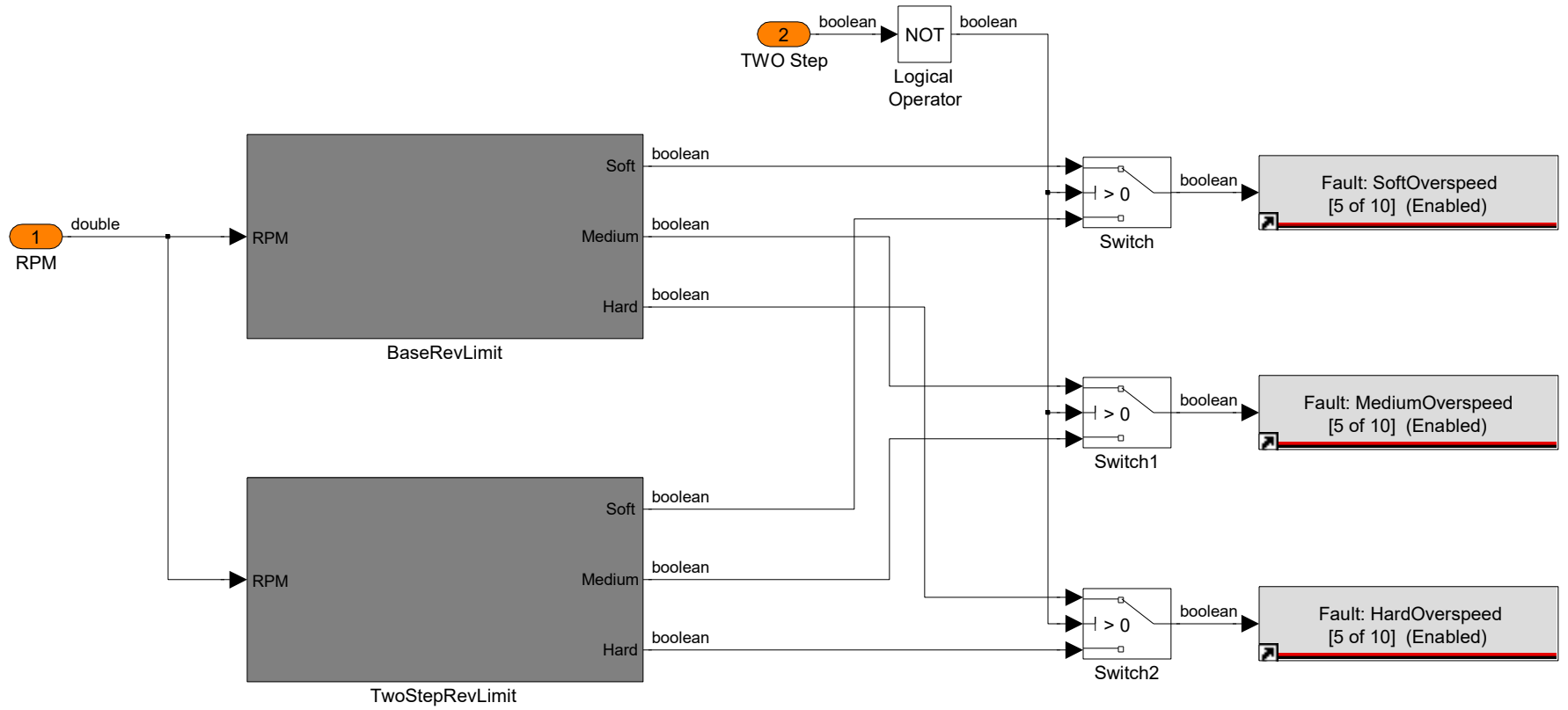


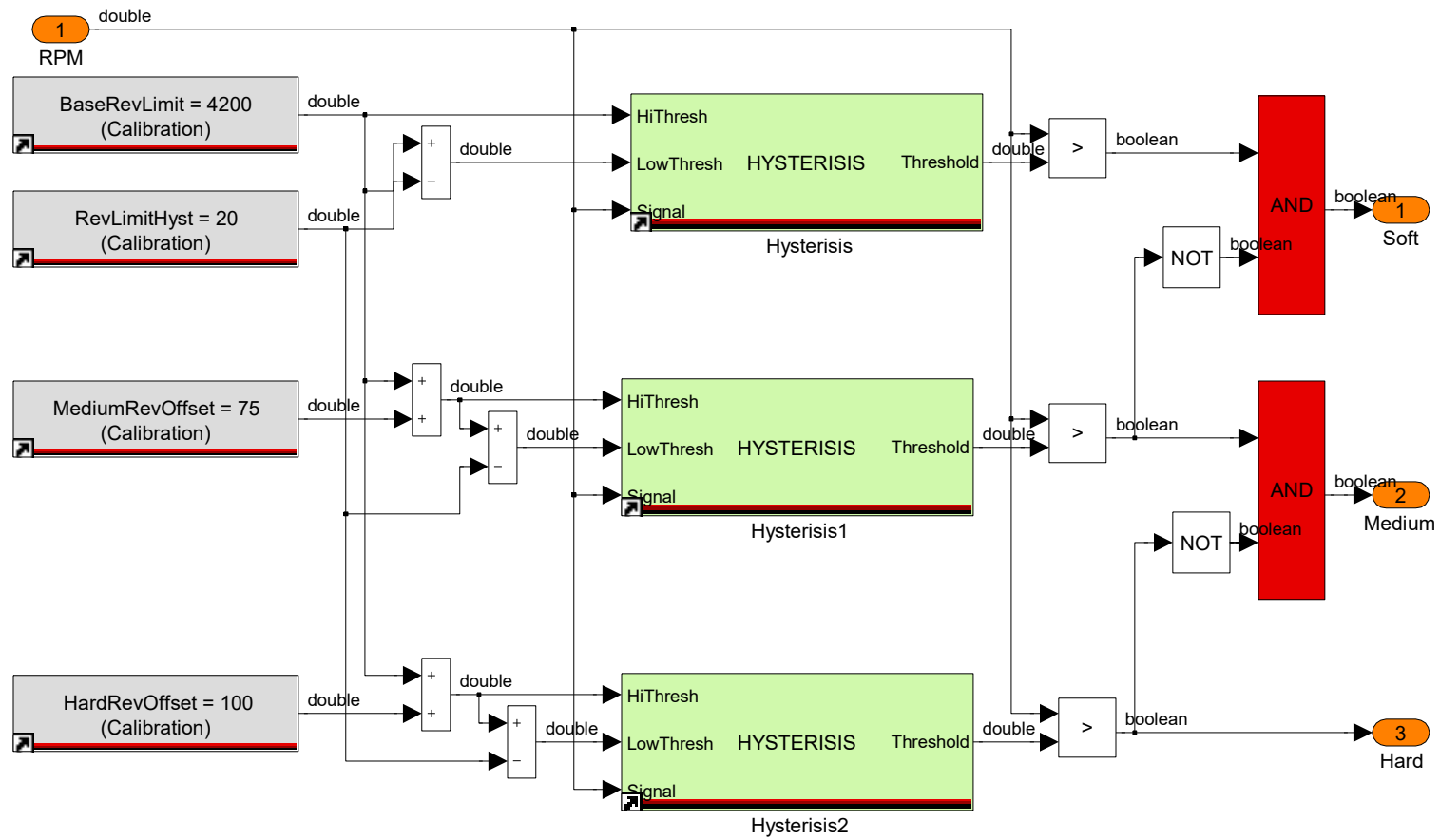


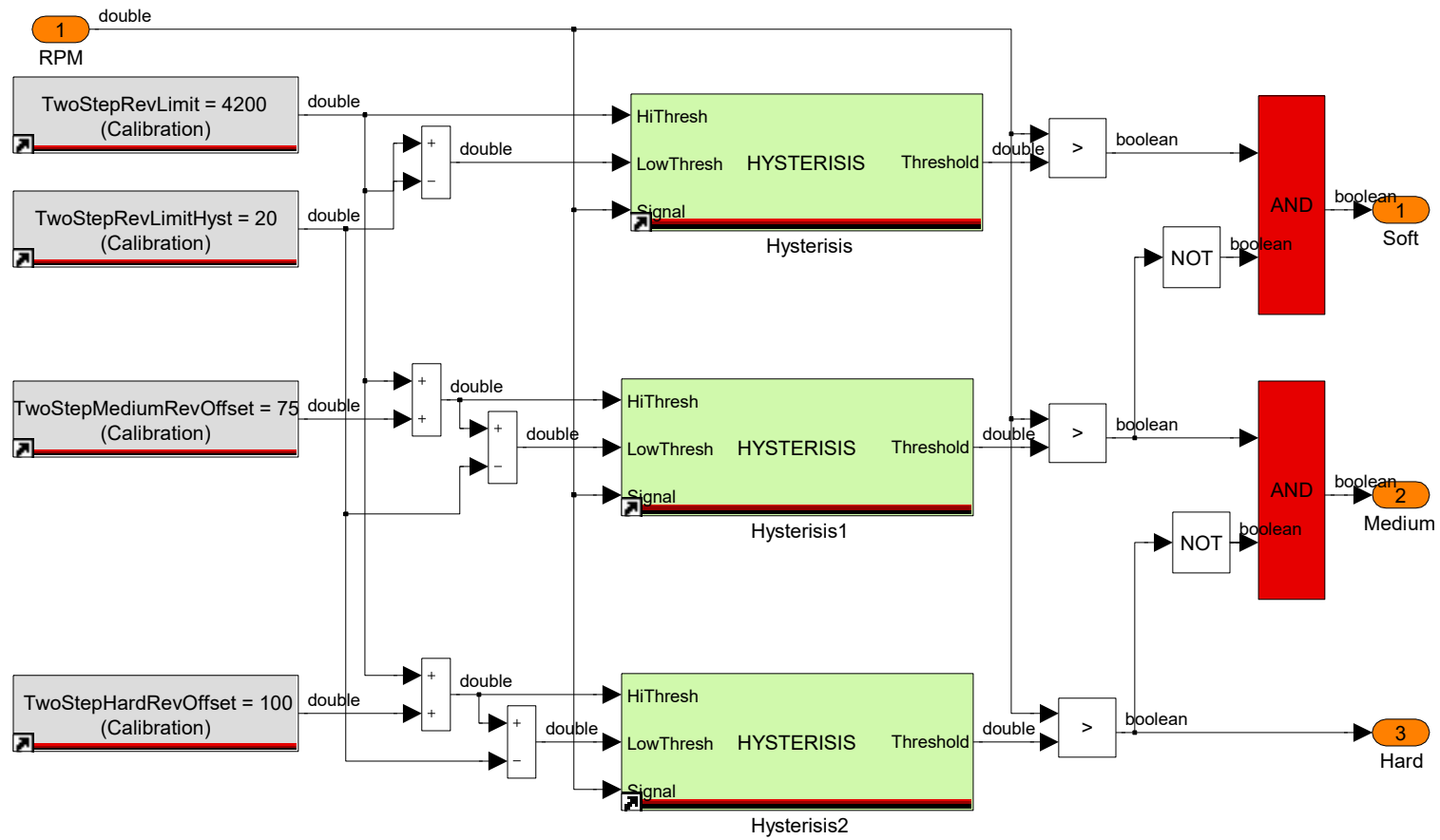


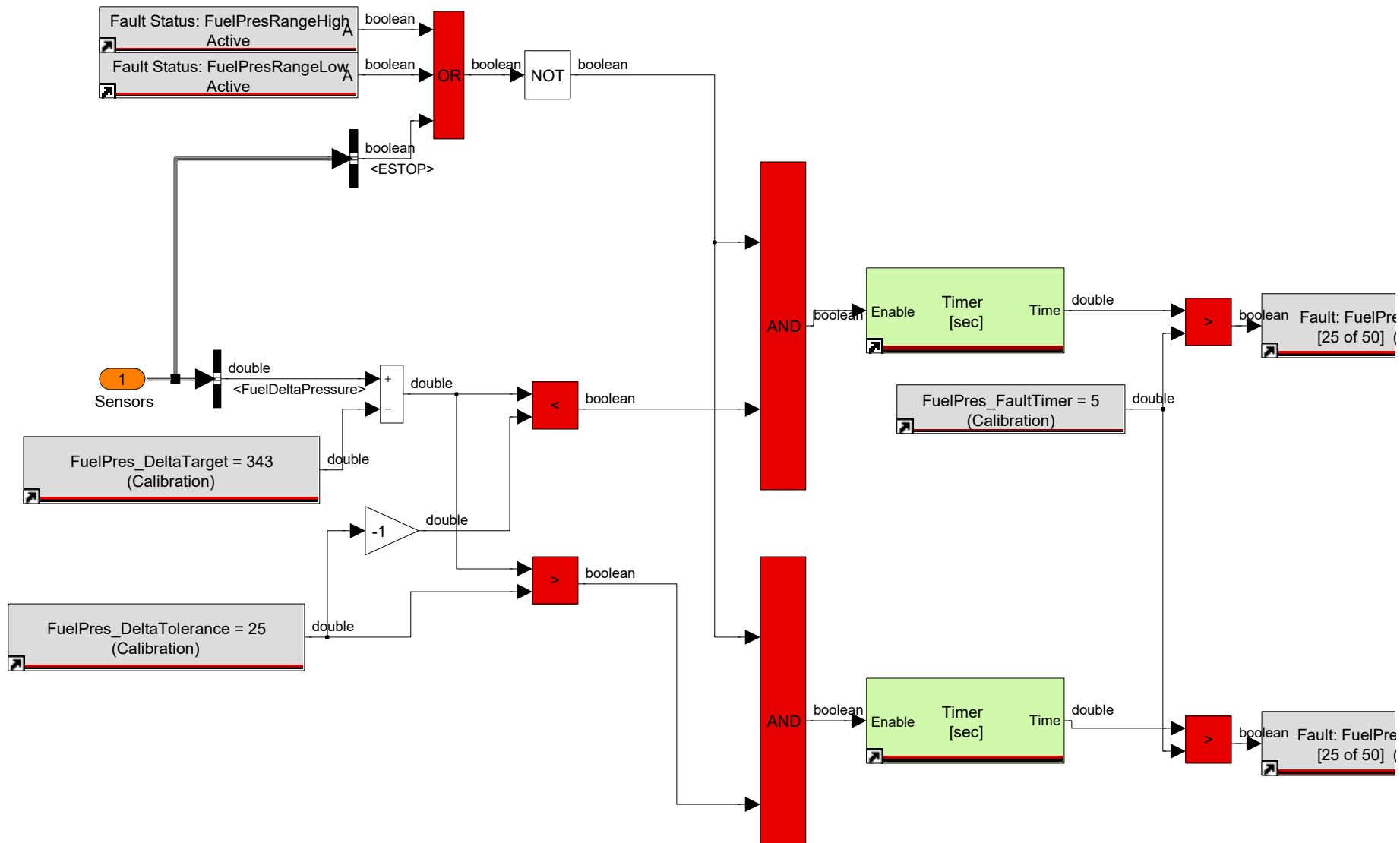
MAP\_IR\_LO  
50] (Enabled)

AP\_STICKING  
0] (Enabled)









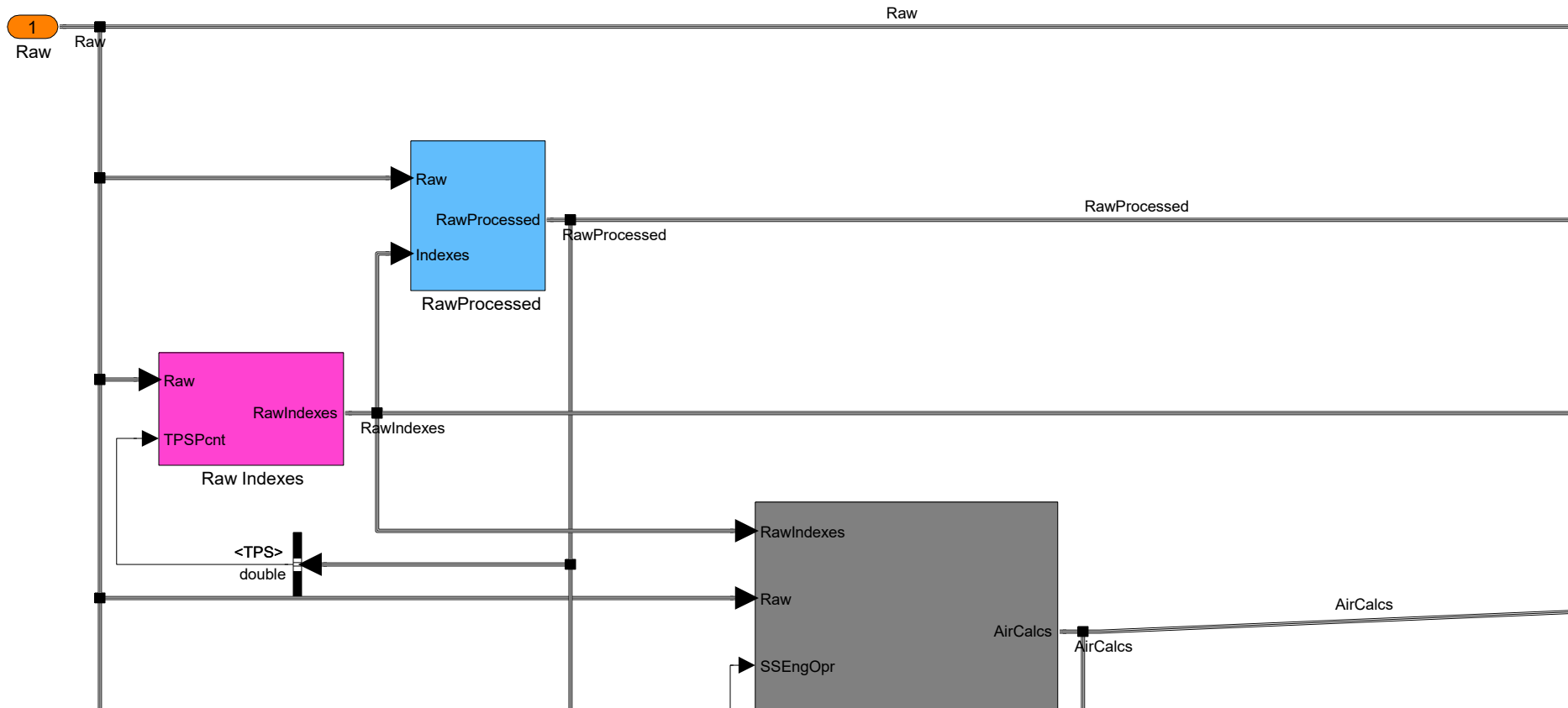
esLow\_Fault  
(Enabled)

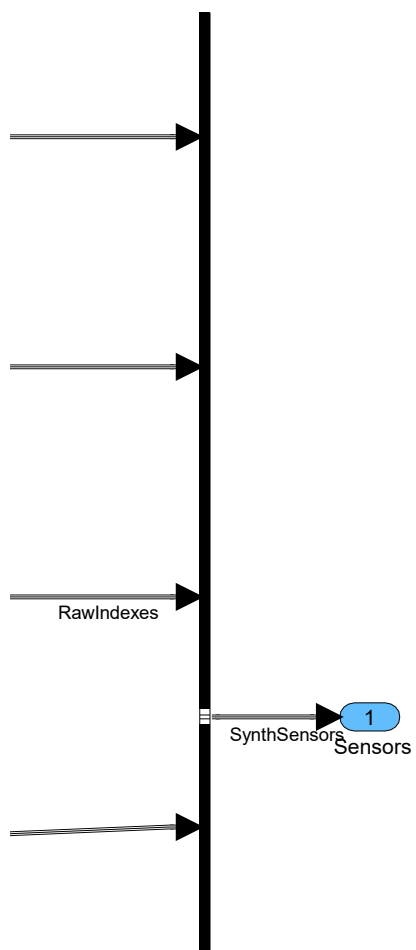
esHigh\_Fault  
(Enabled)

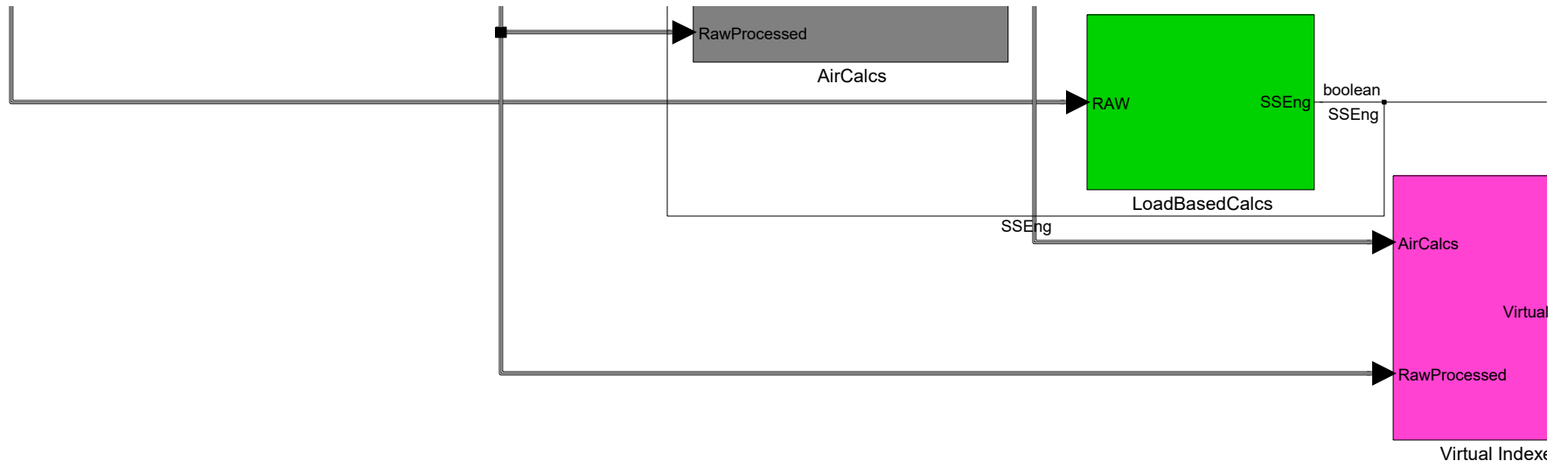


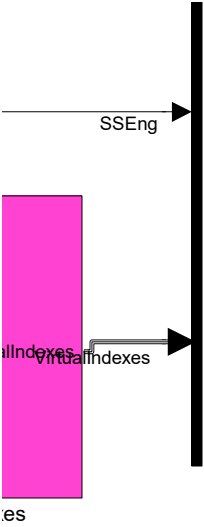


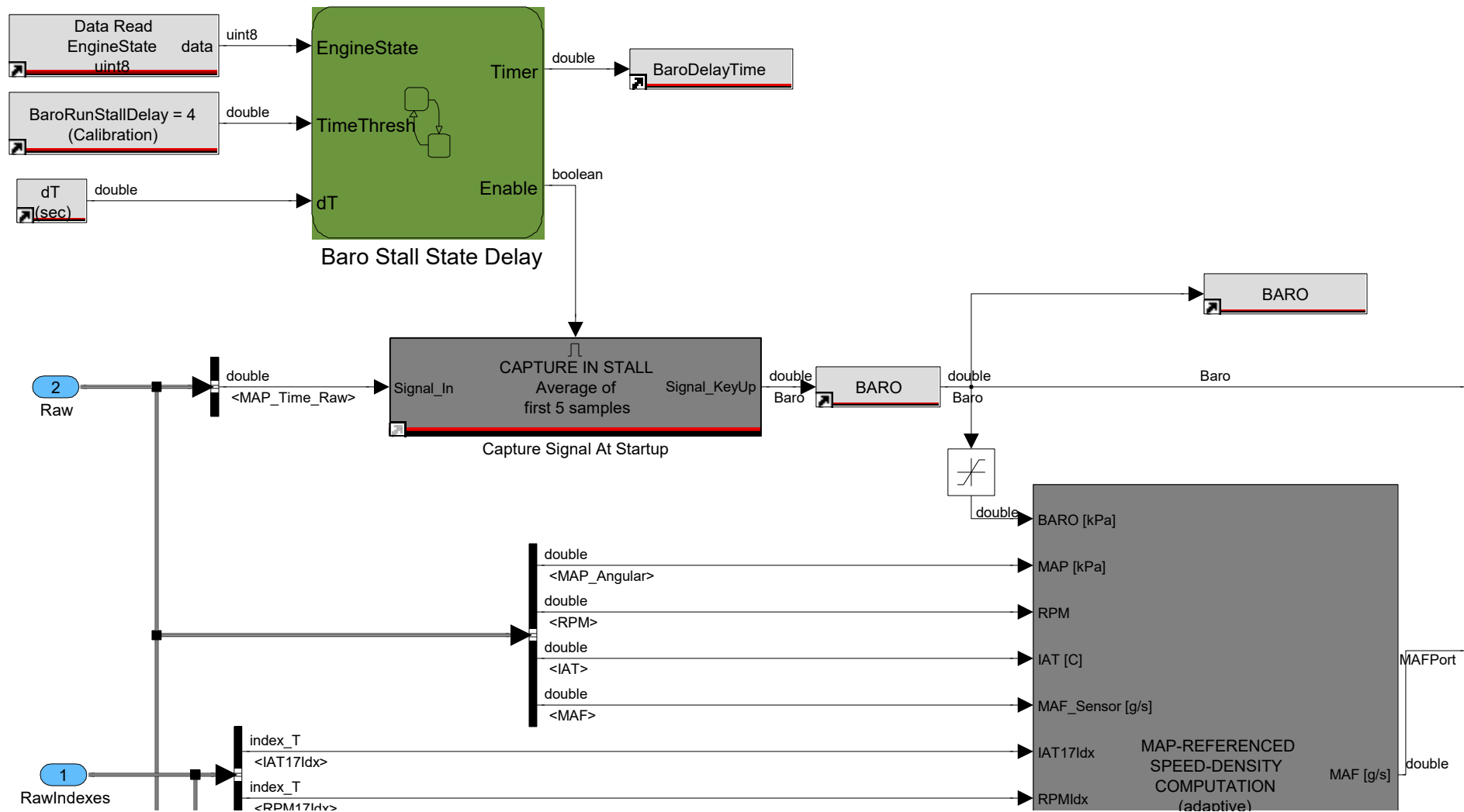


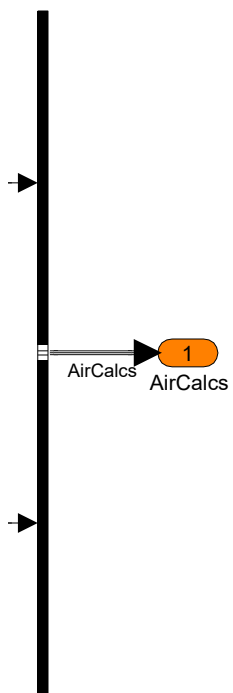


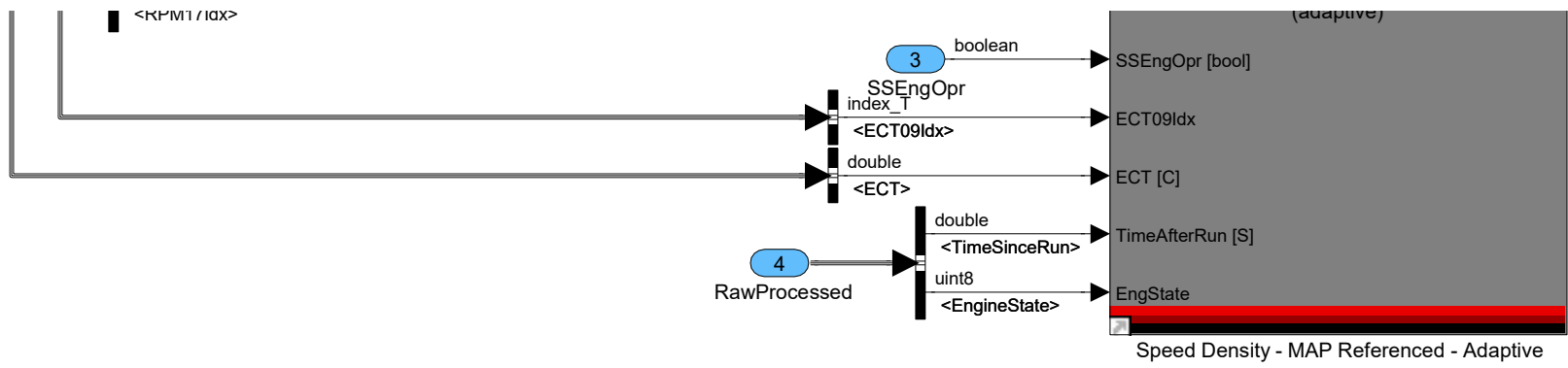






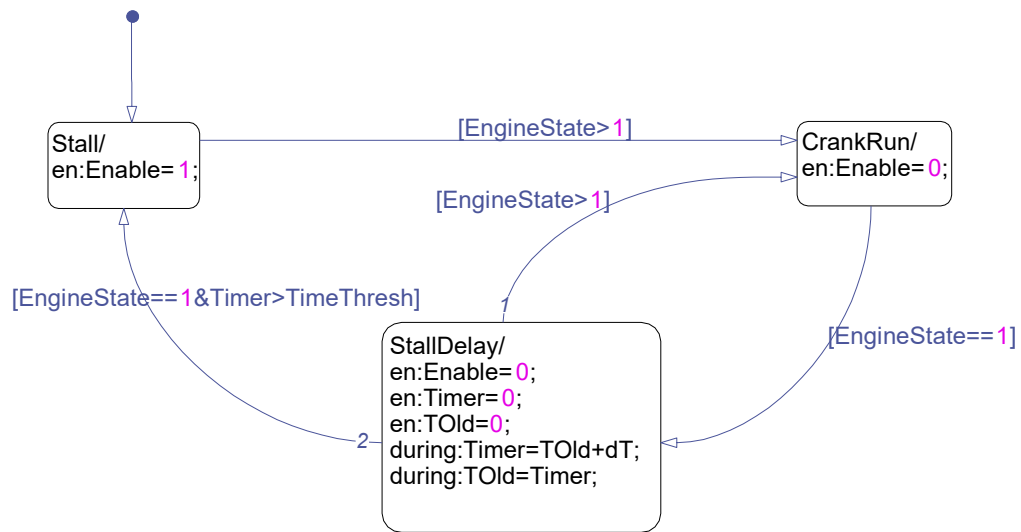




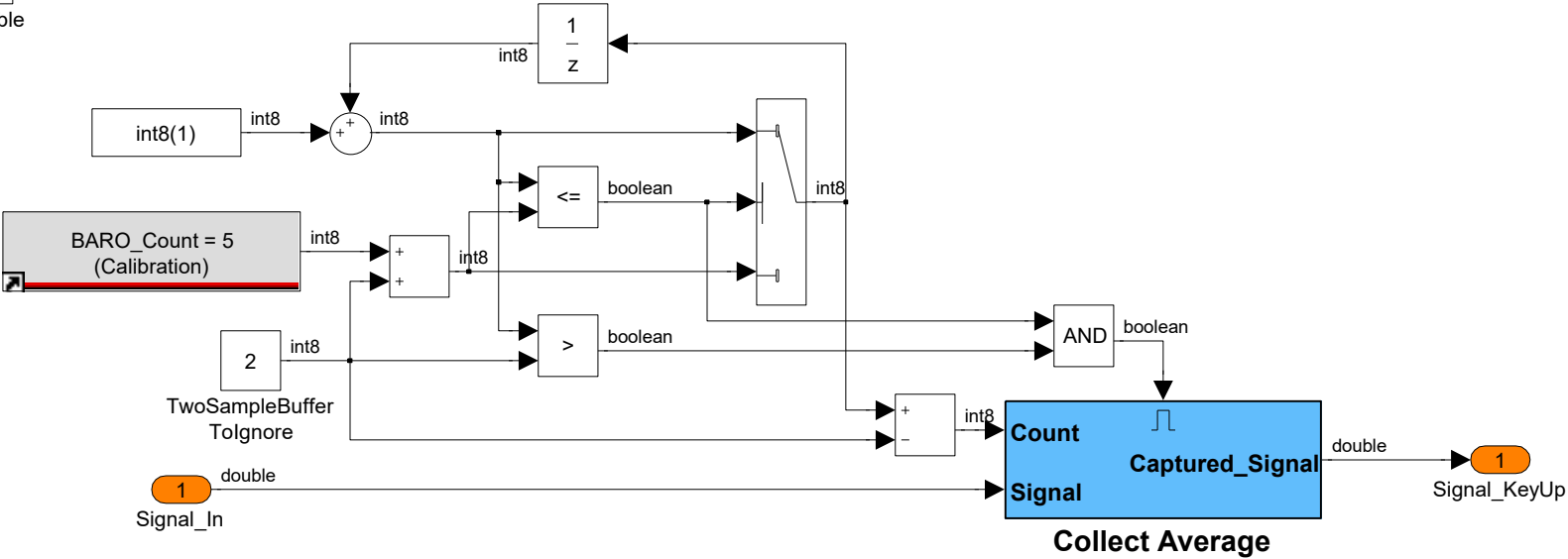




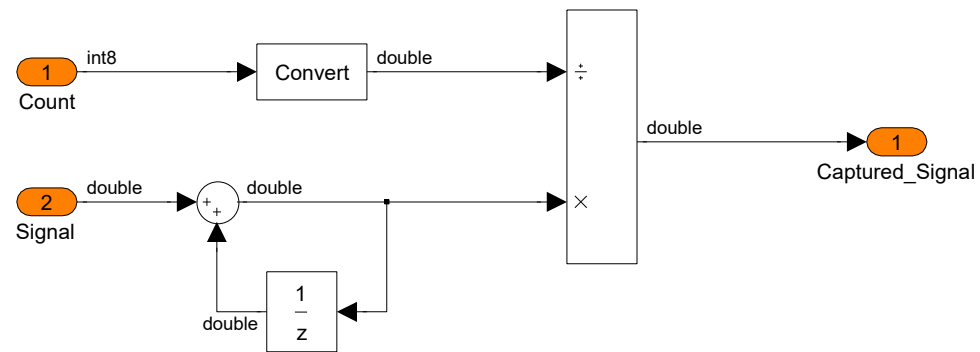


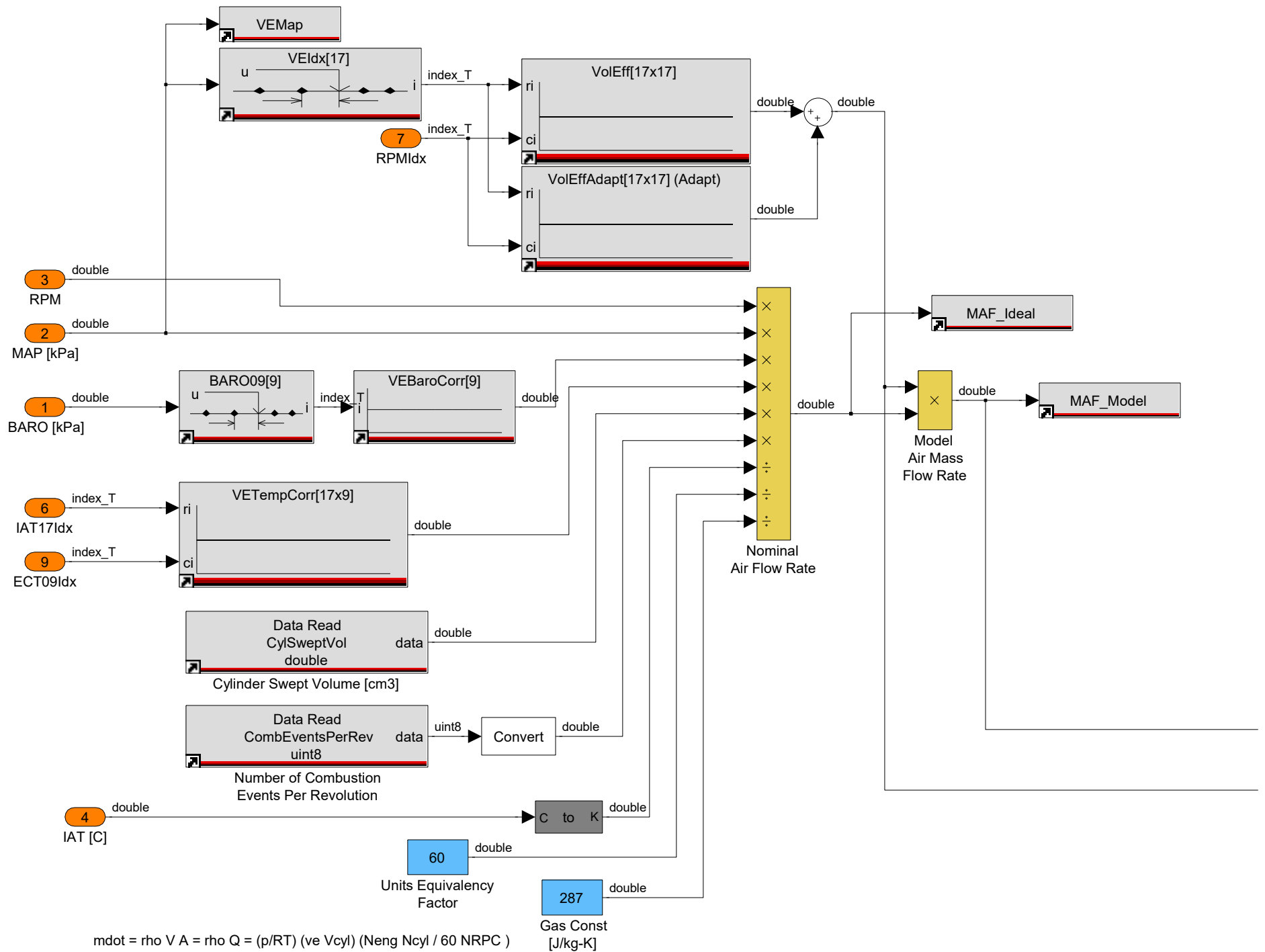


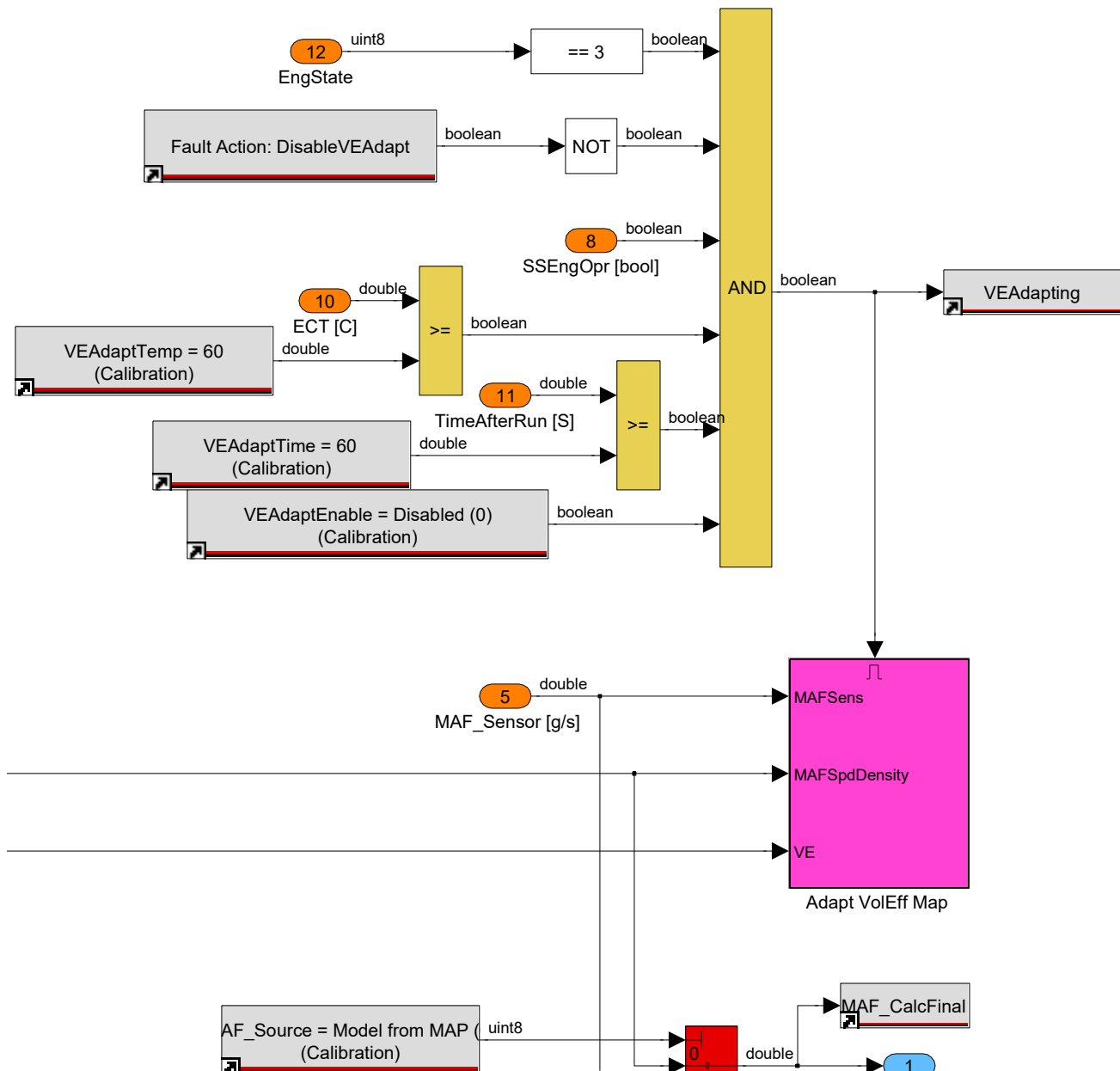
Count up to "BARO\_Count" samples

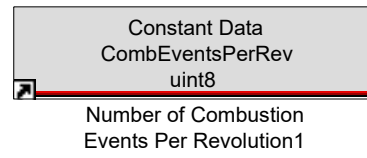
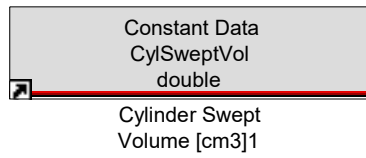


**Collect samples by summing,  
and divide by the total, to obtain average**

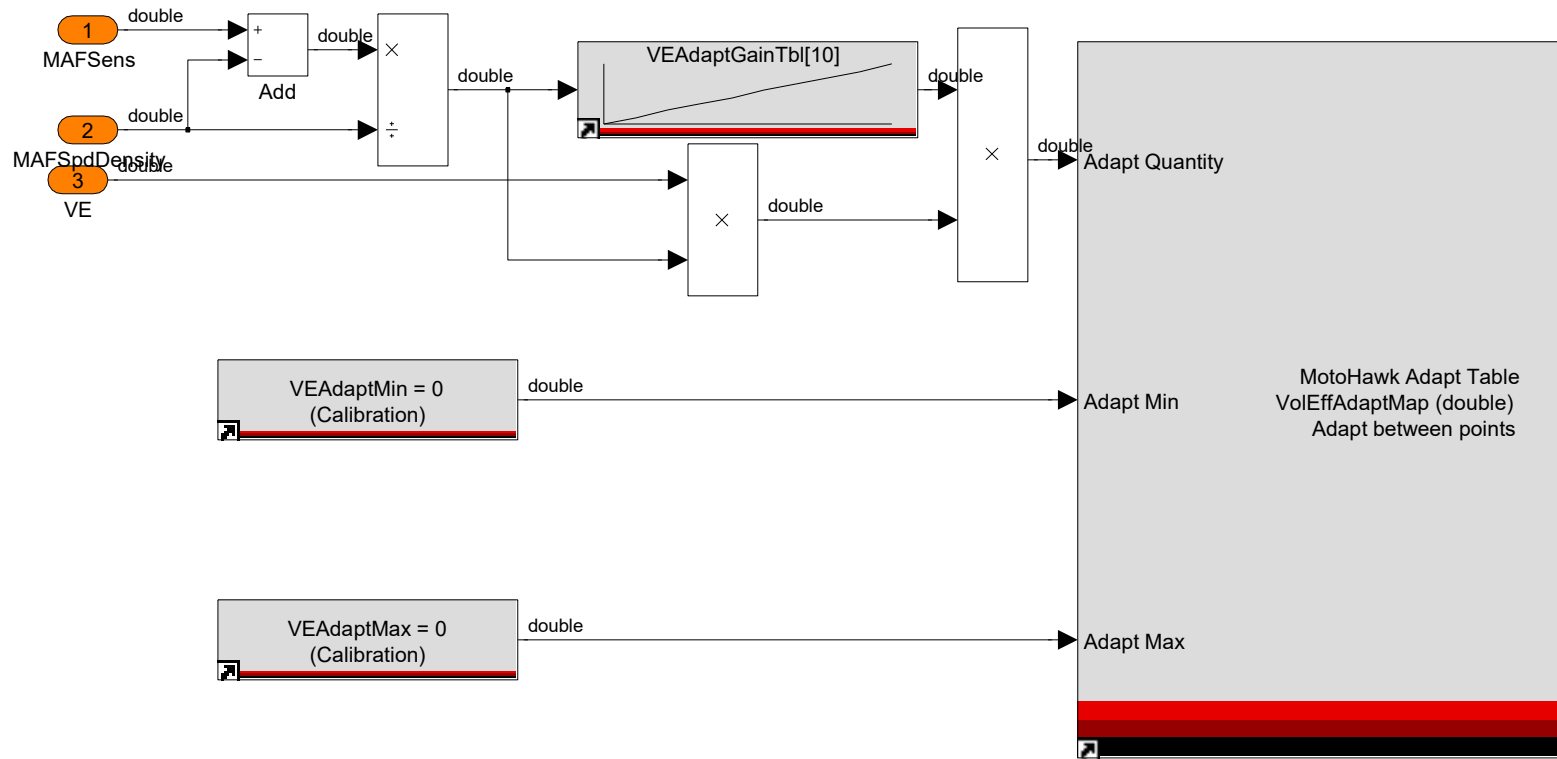




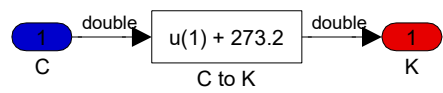


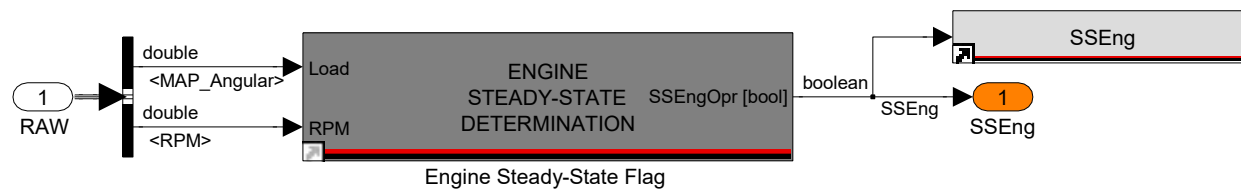


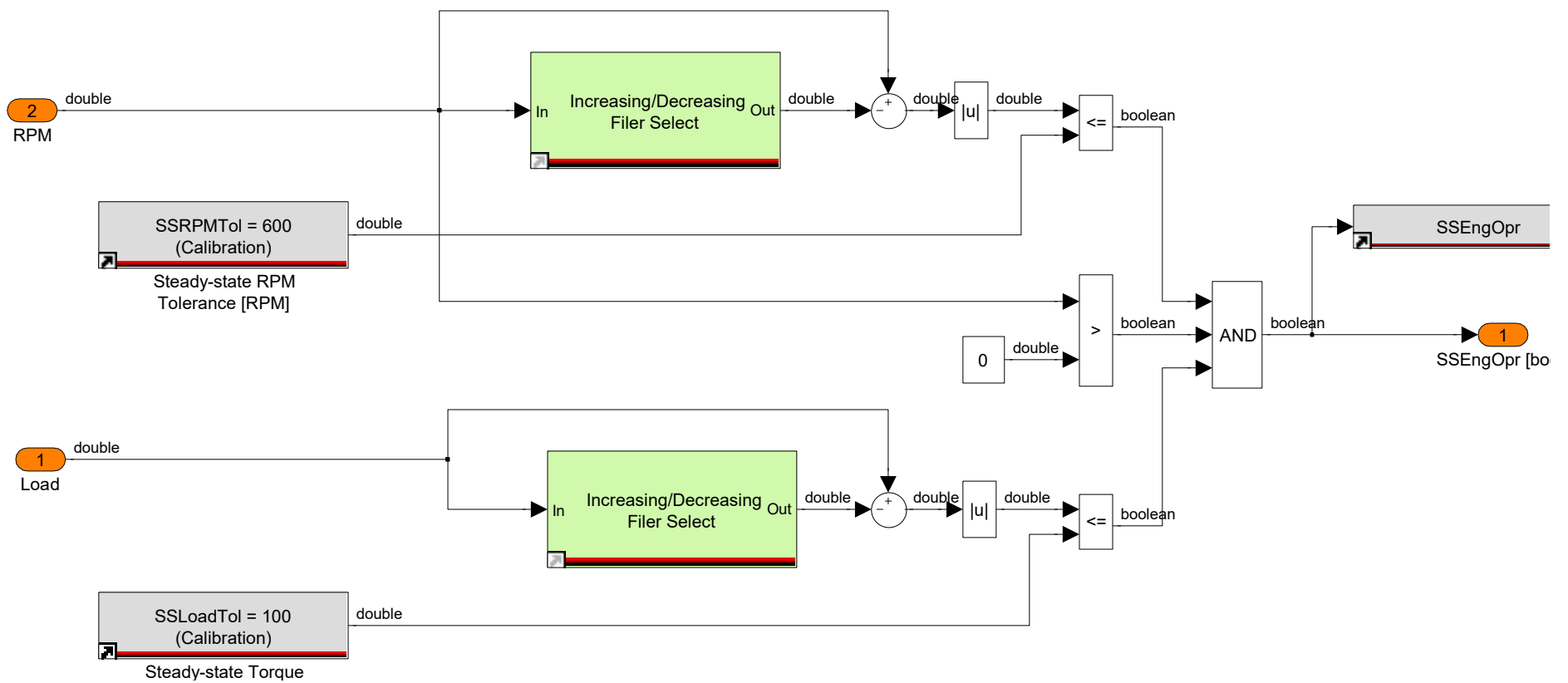










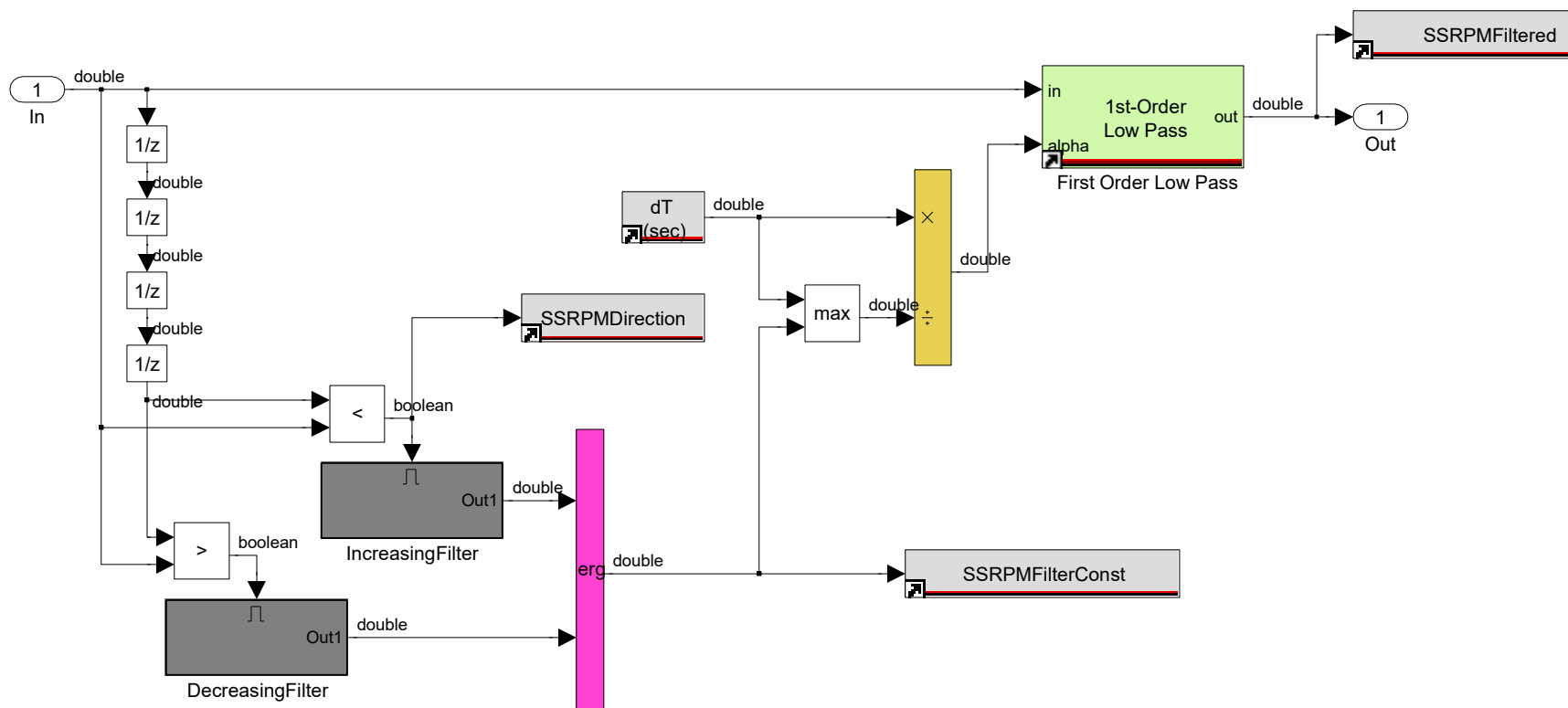




sol]

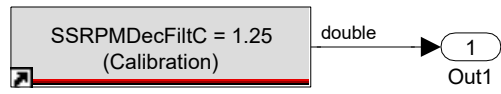
Steady state torque  
Tolerance [N.m]





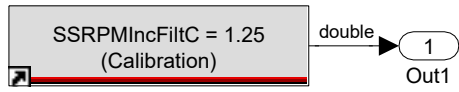


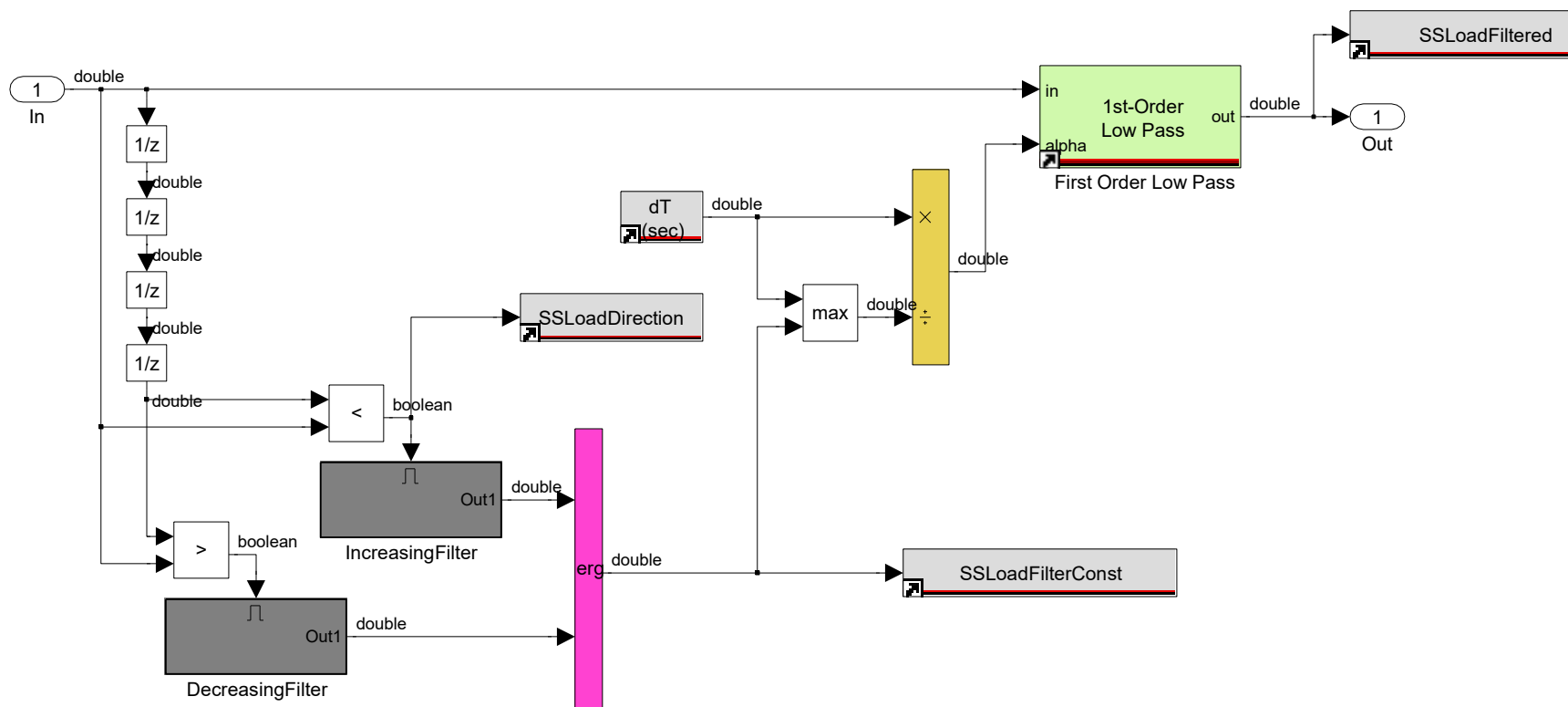




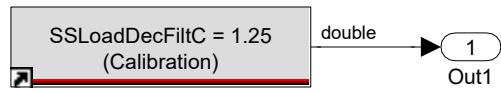


Enable



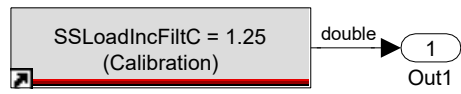






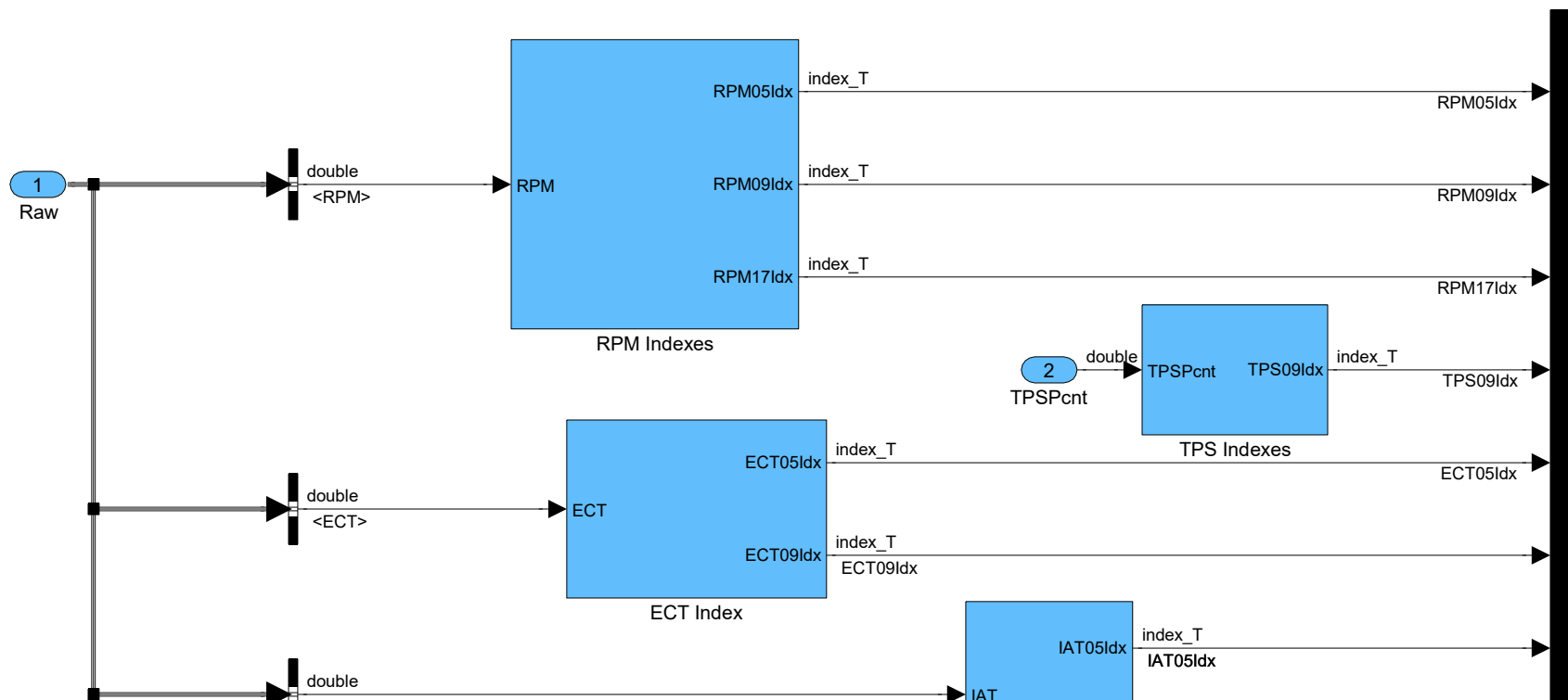
double





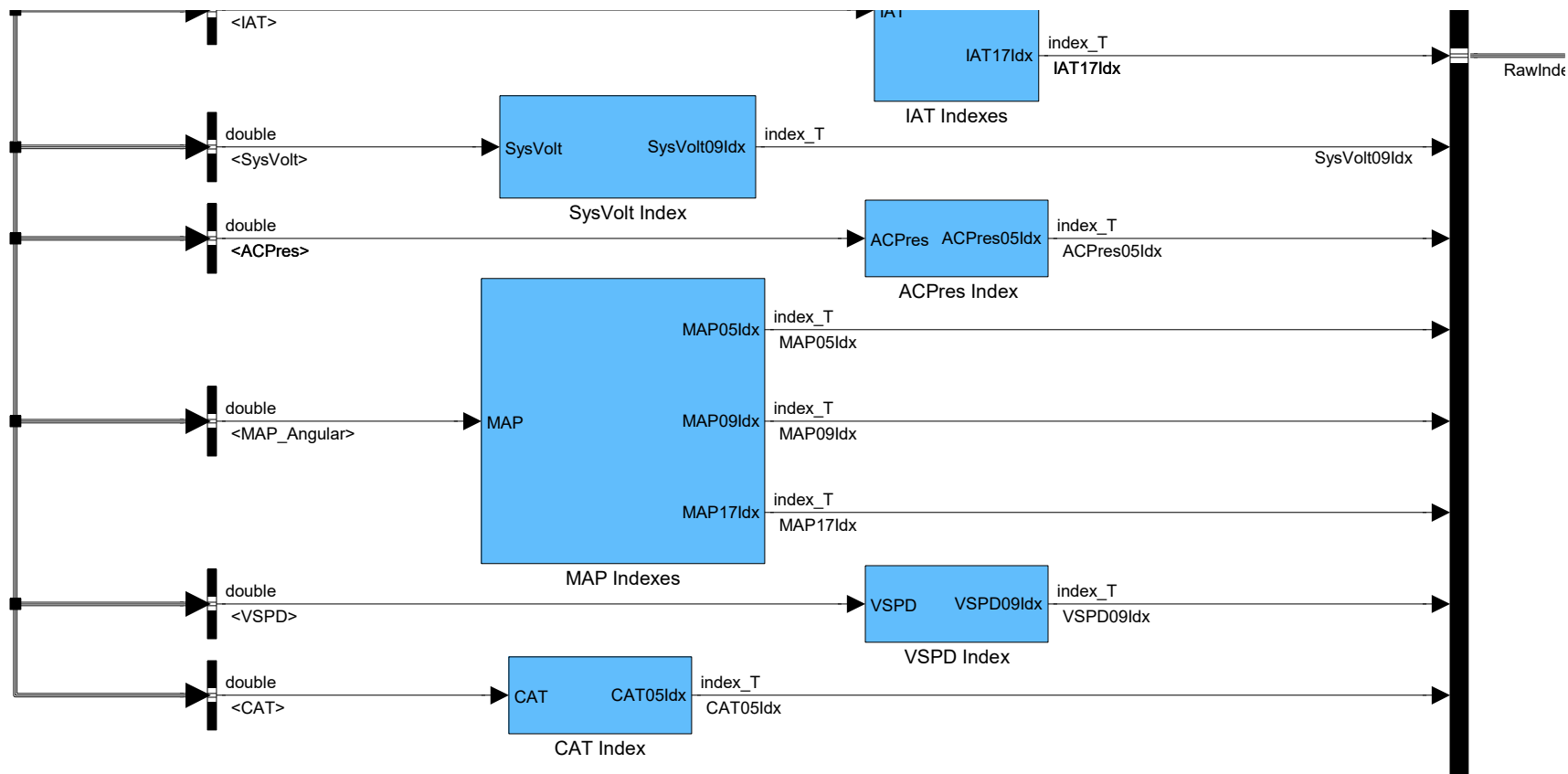
double

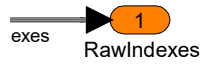


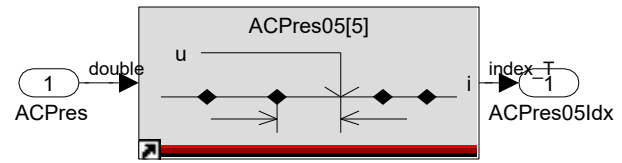


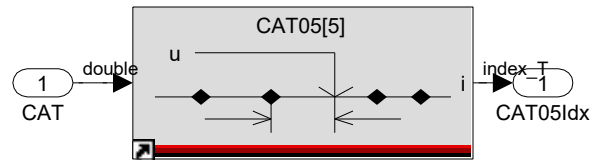


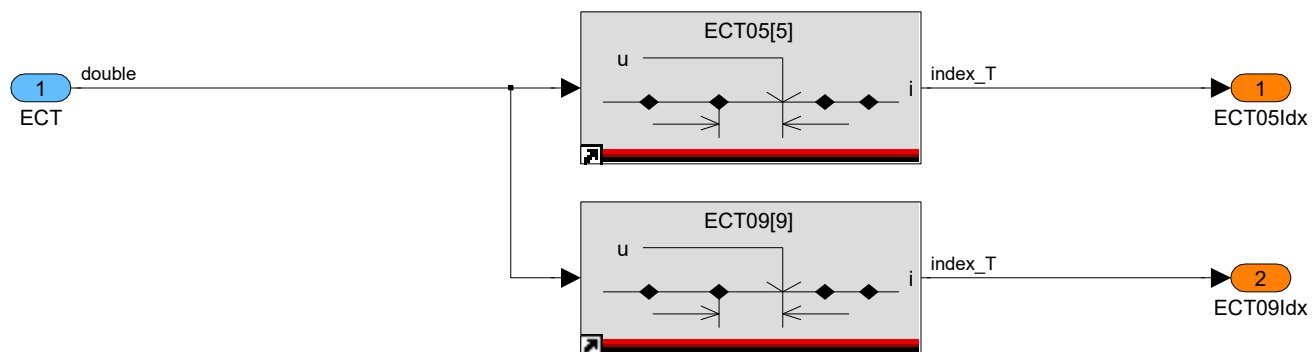


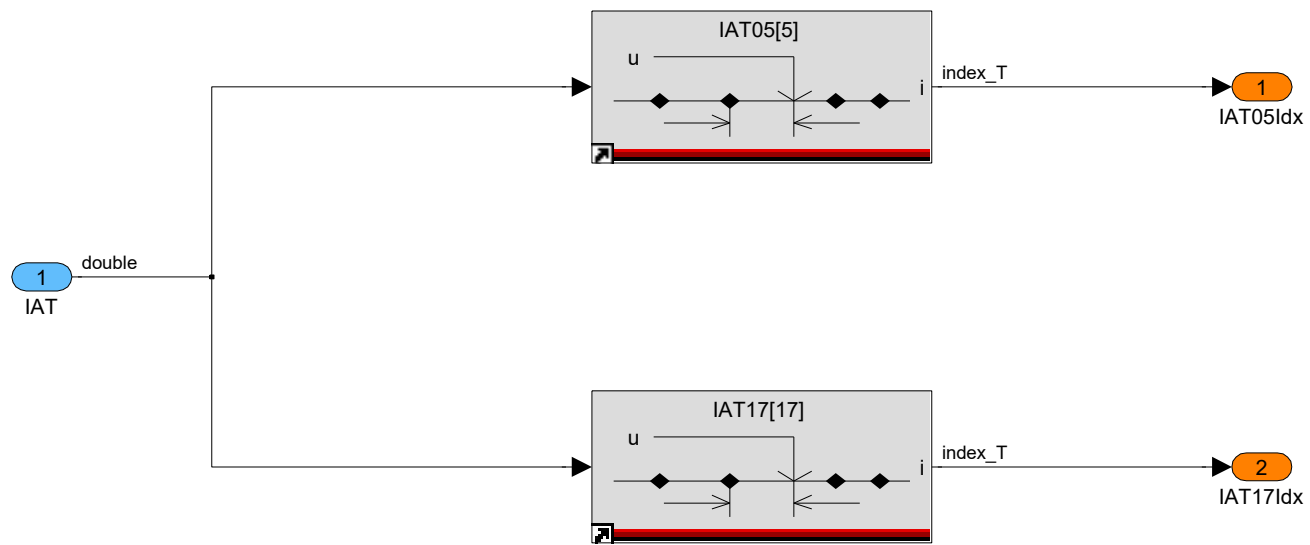


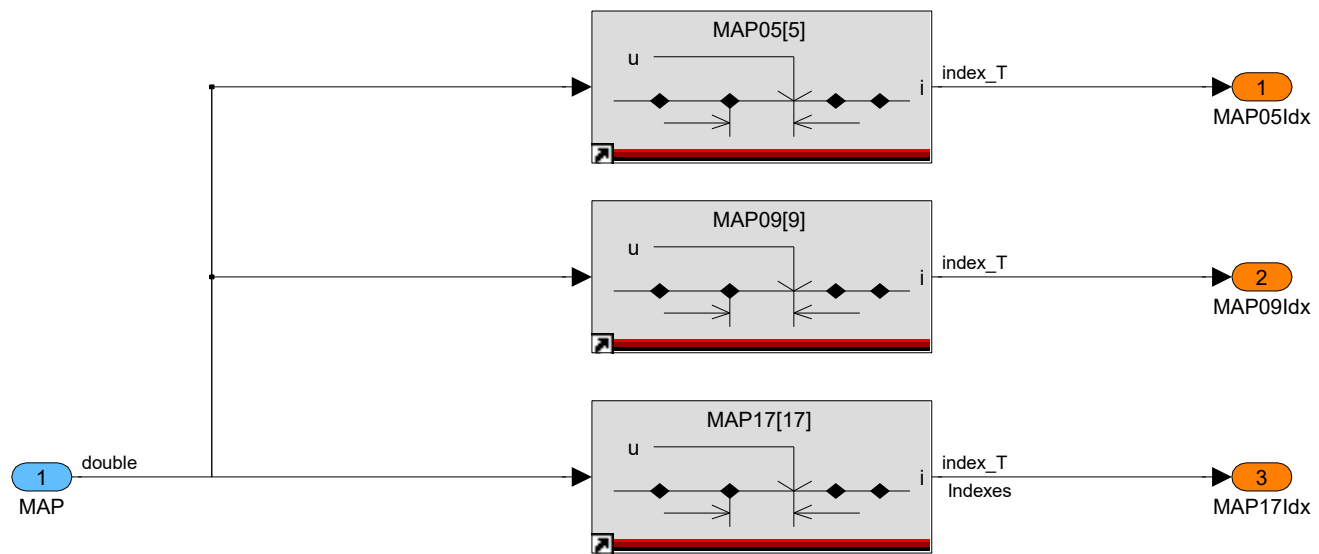


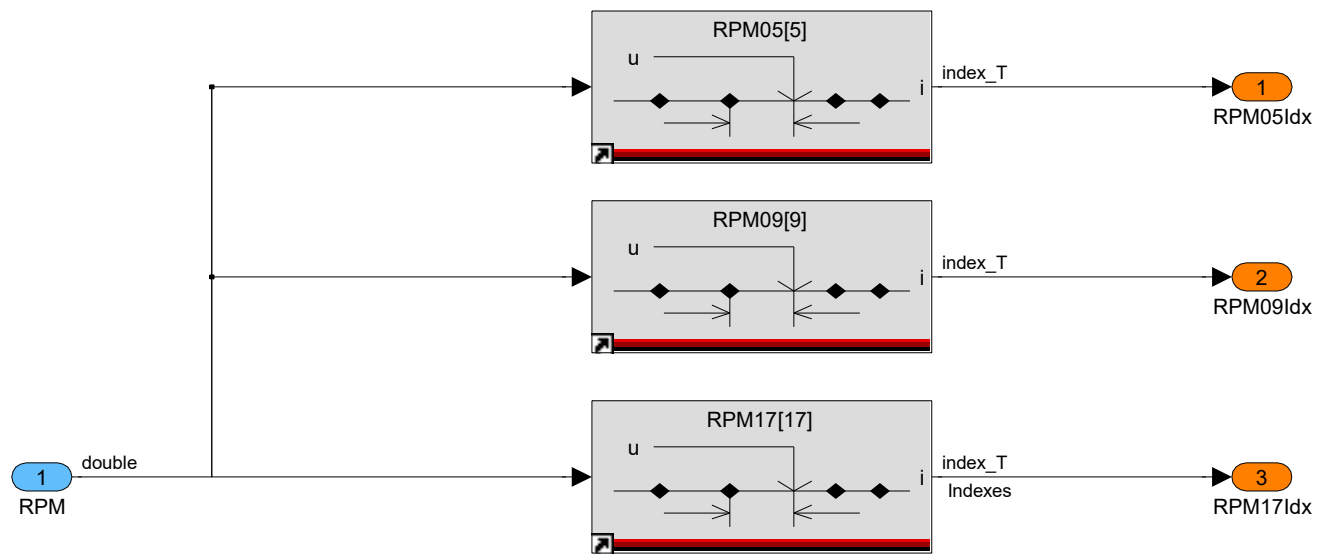




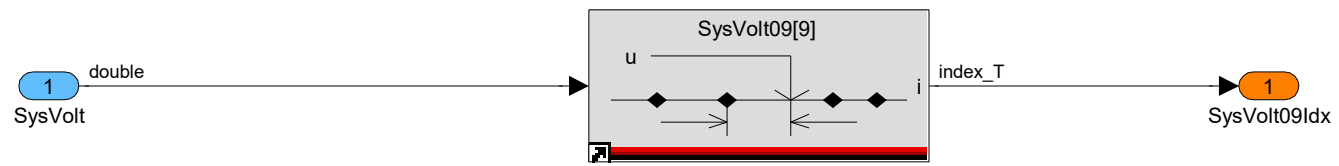


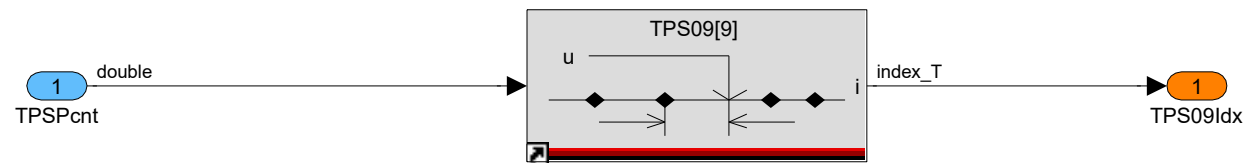


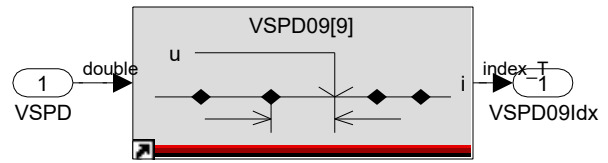


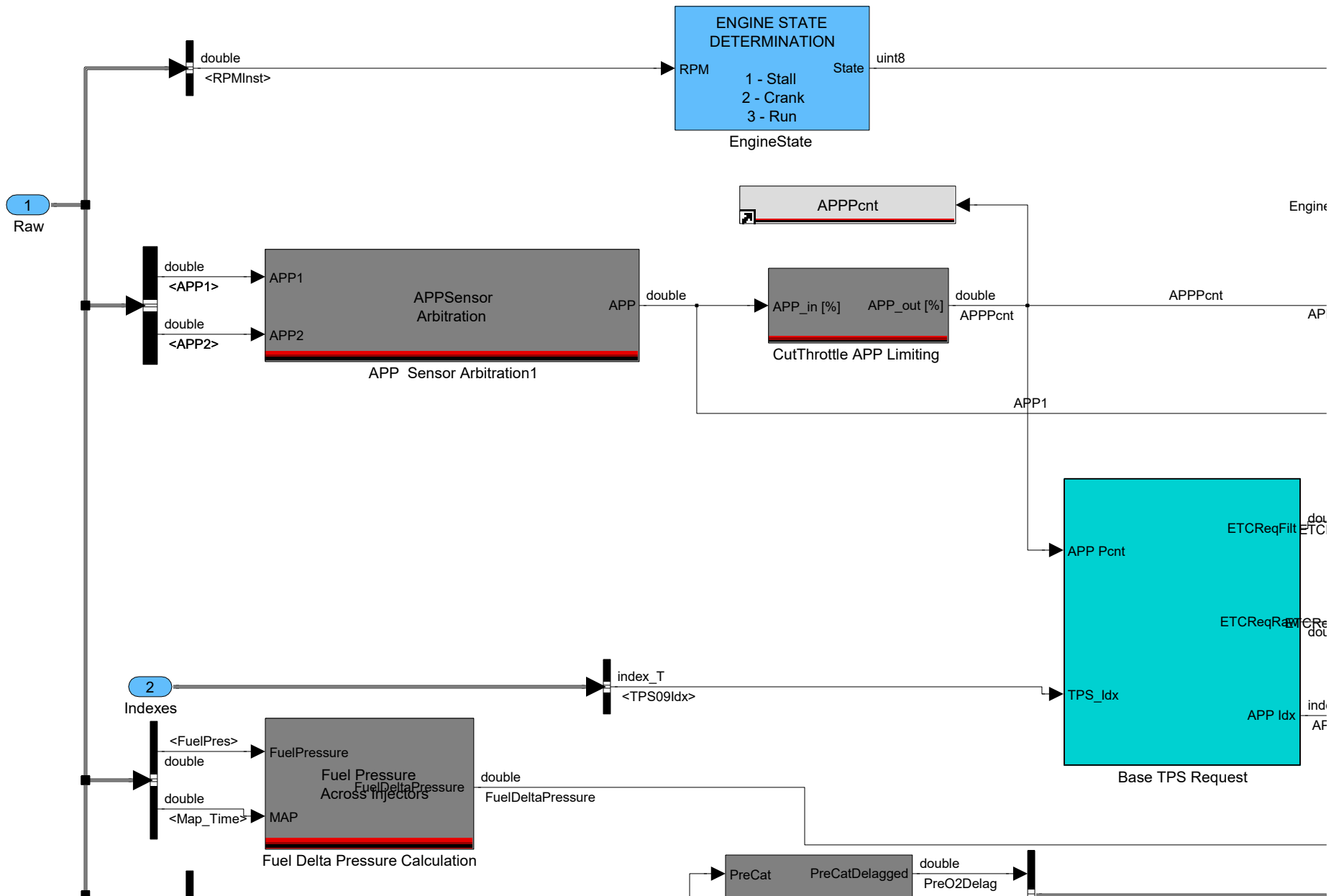


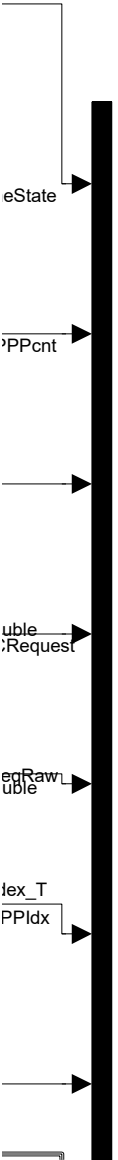


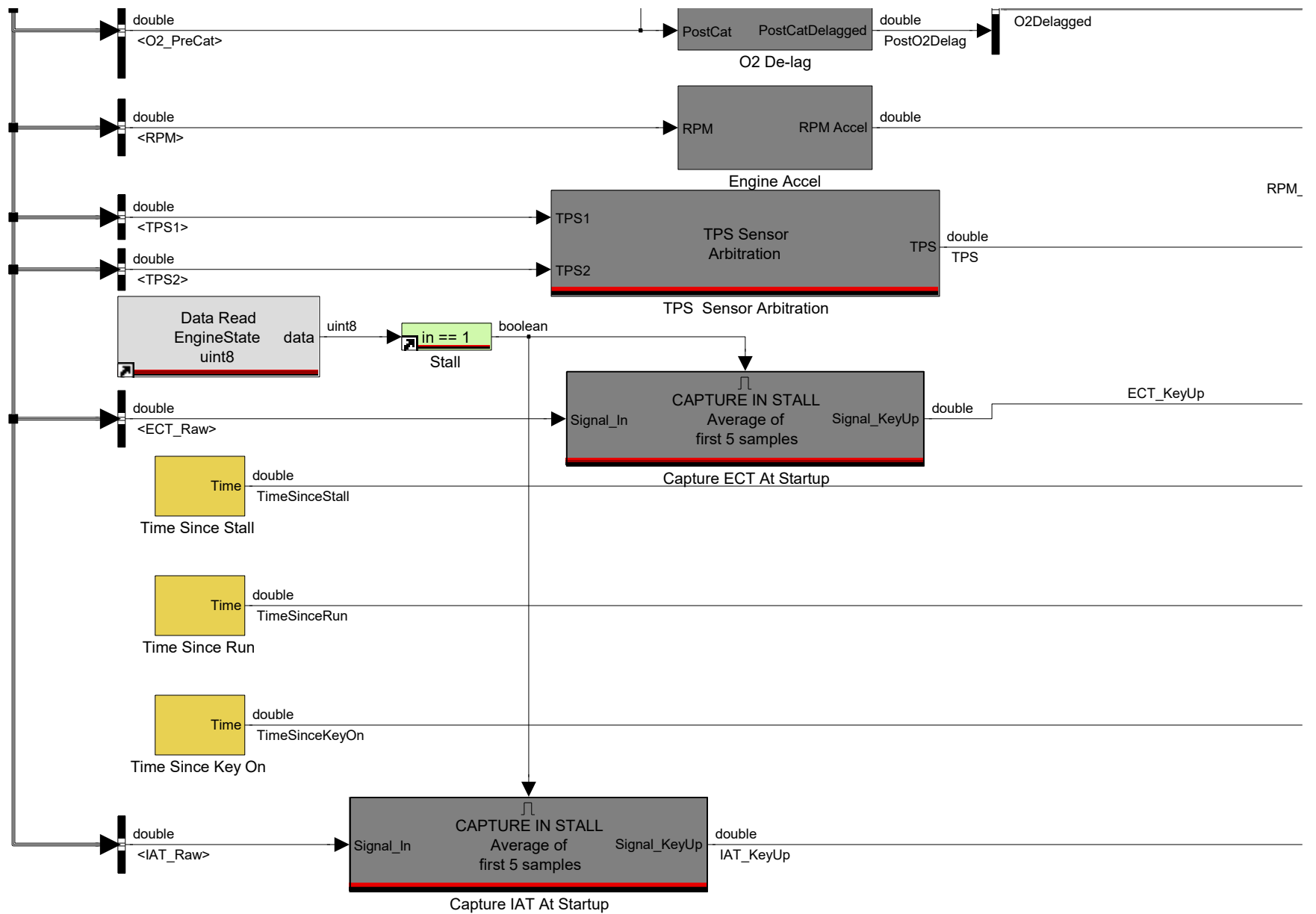


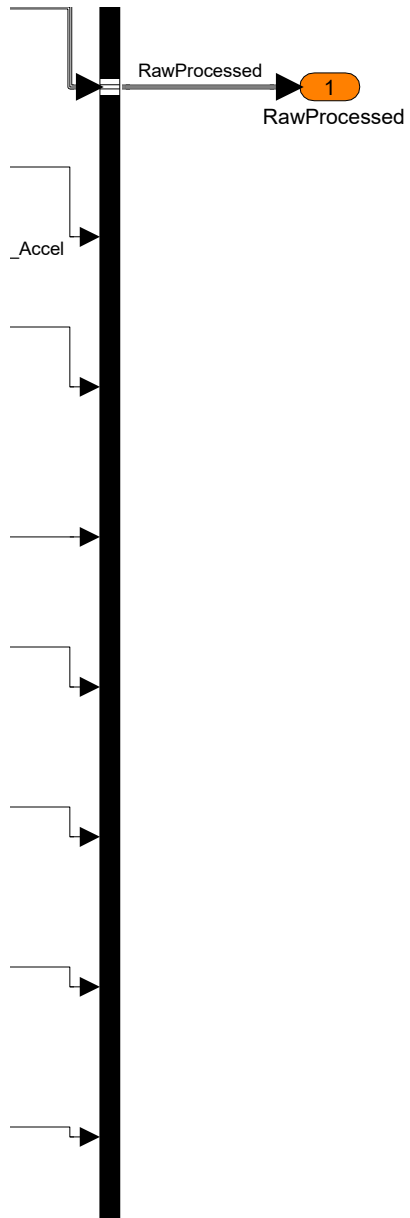


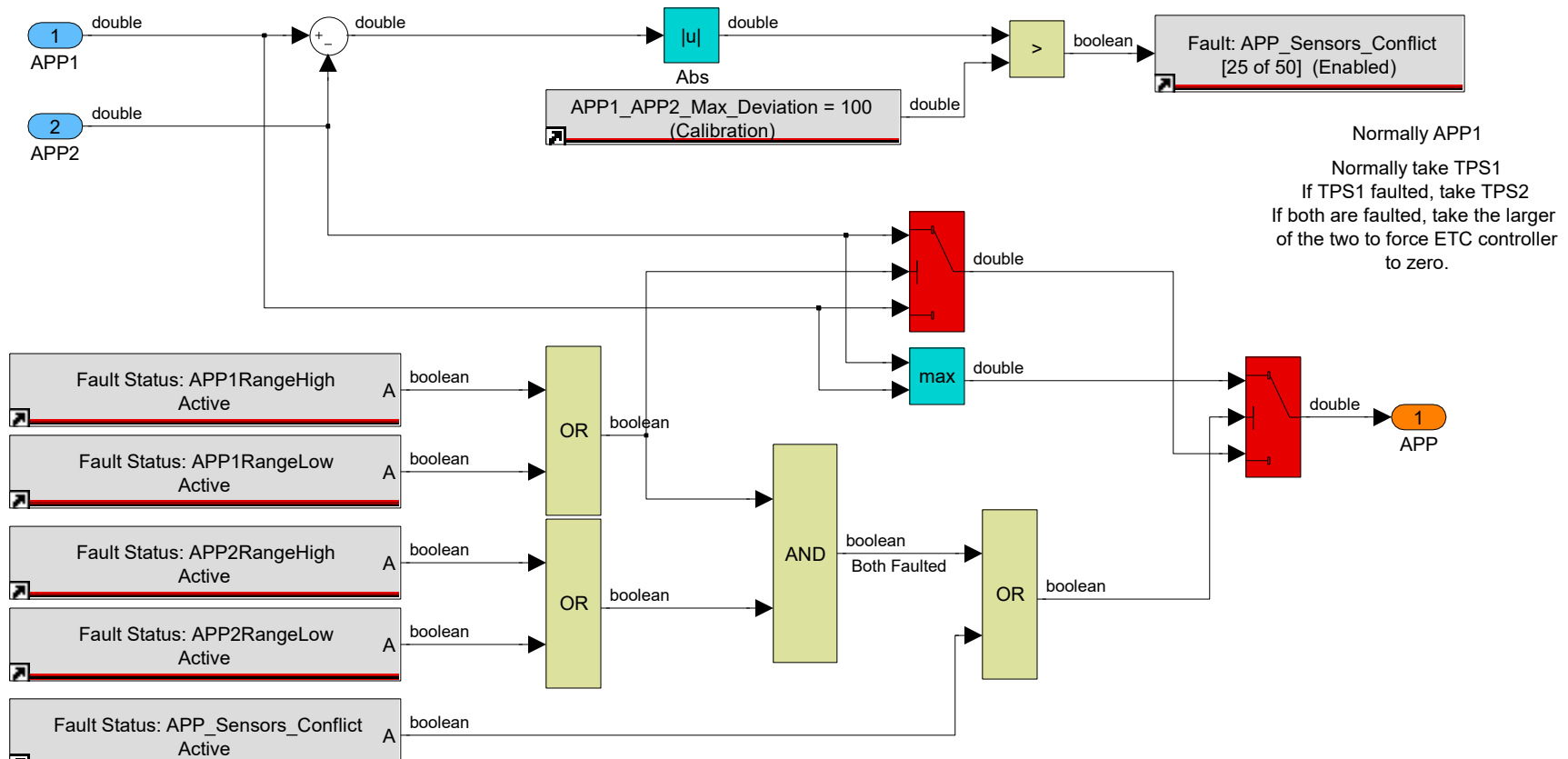






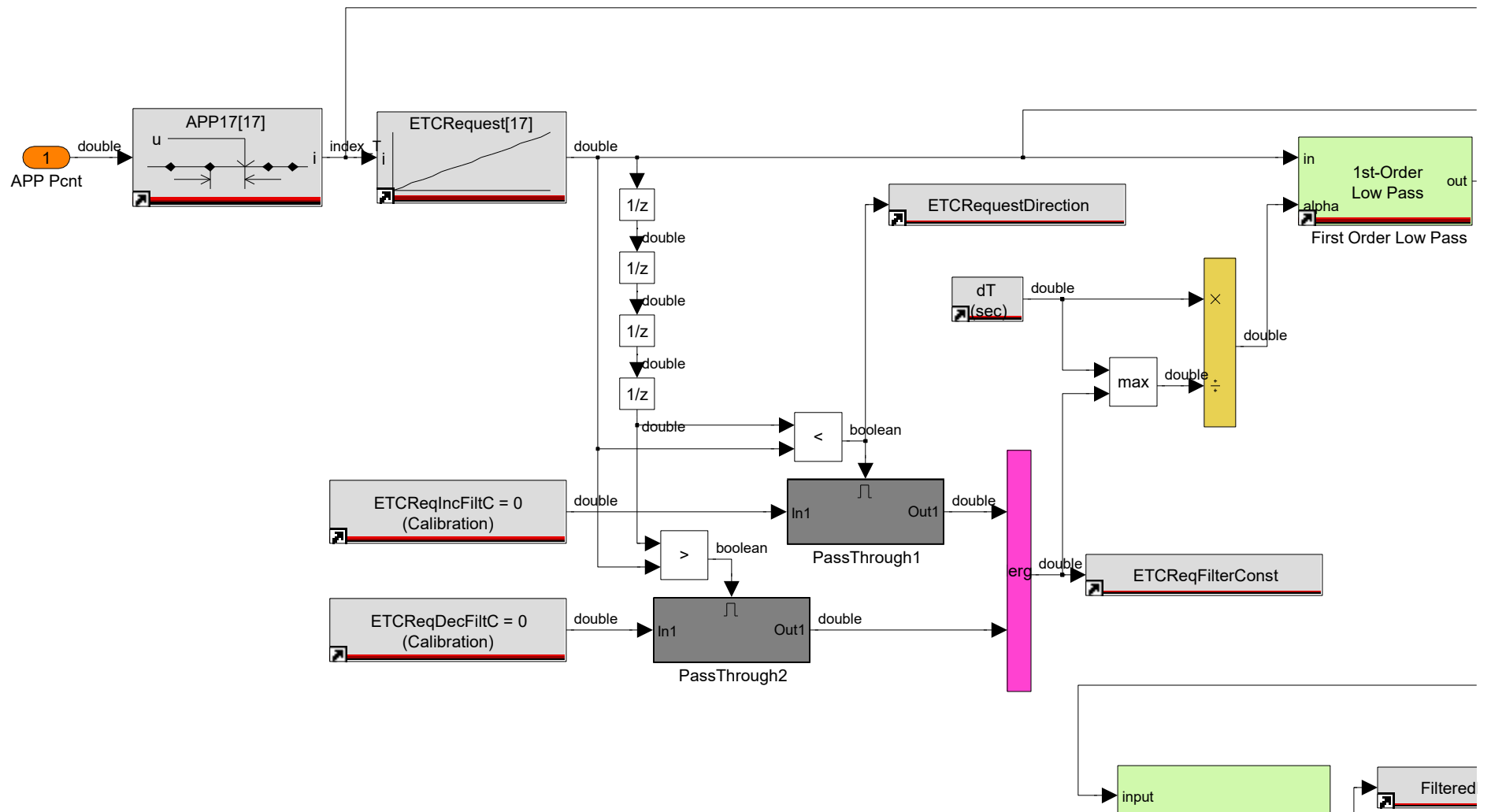




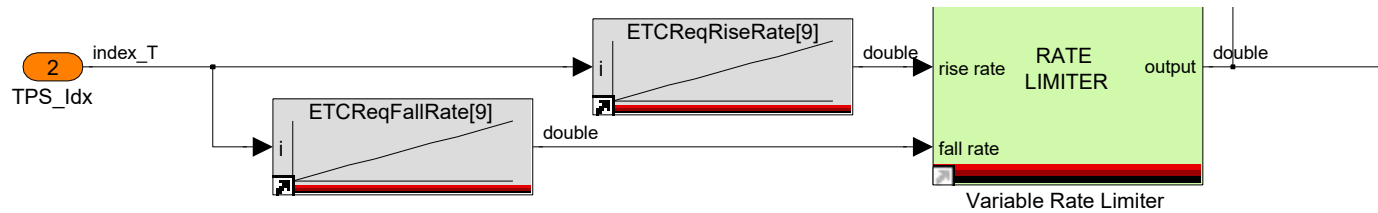




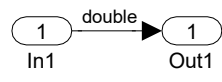






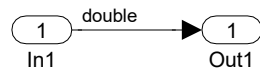




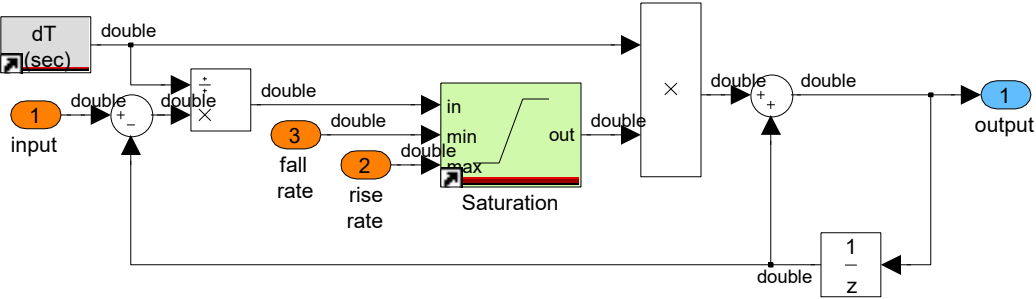




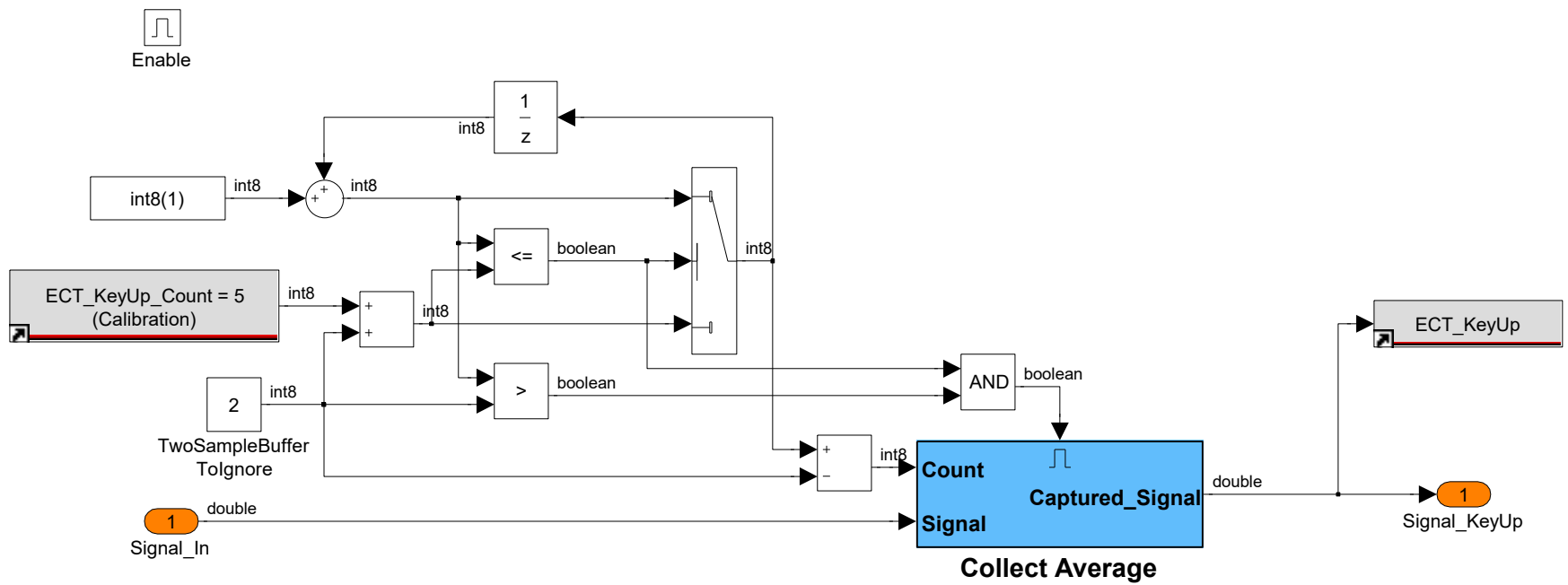
Enable



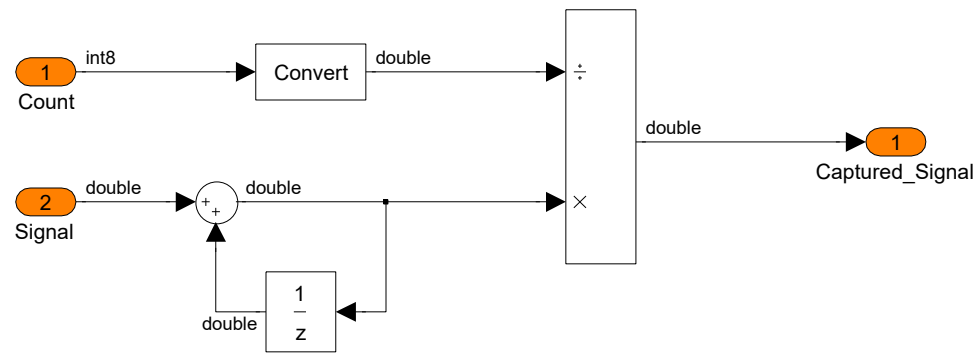
Rate Limiter - Limit allowable change in signal per timestep

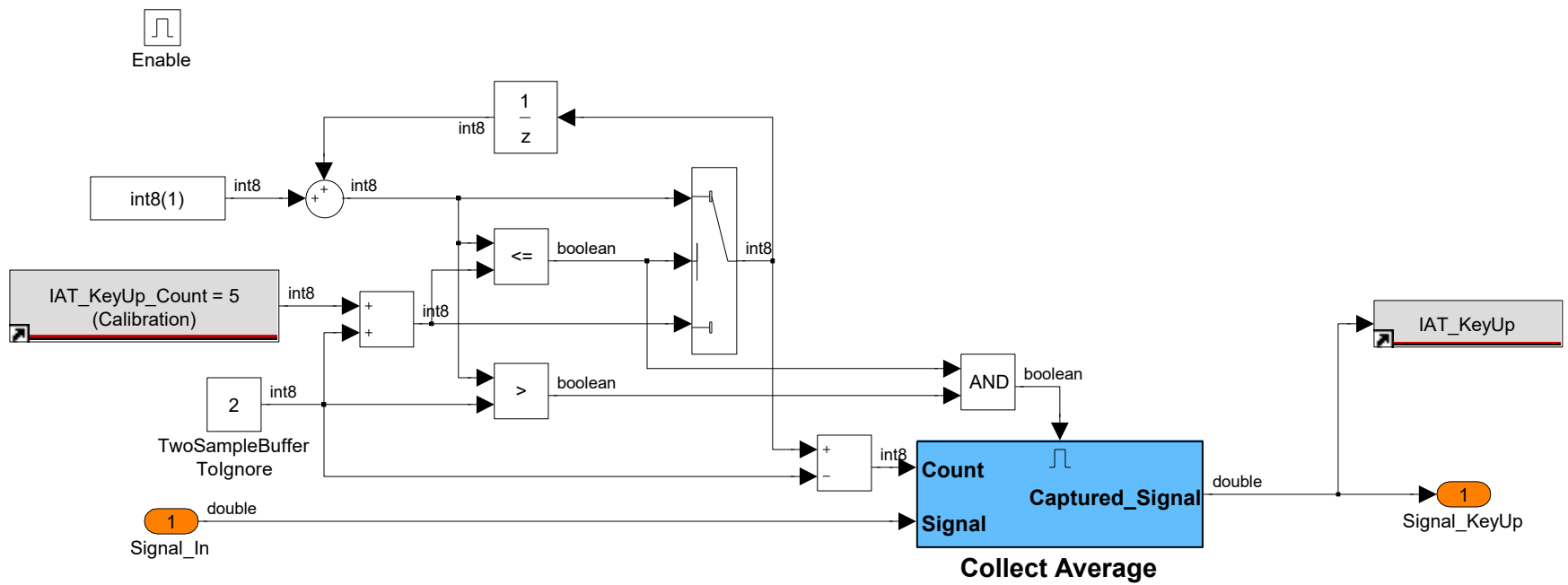




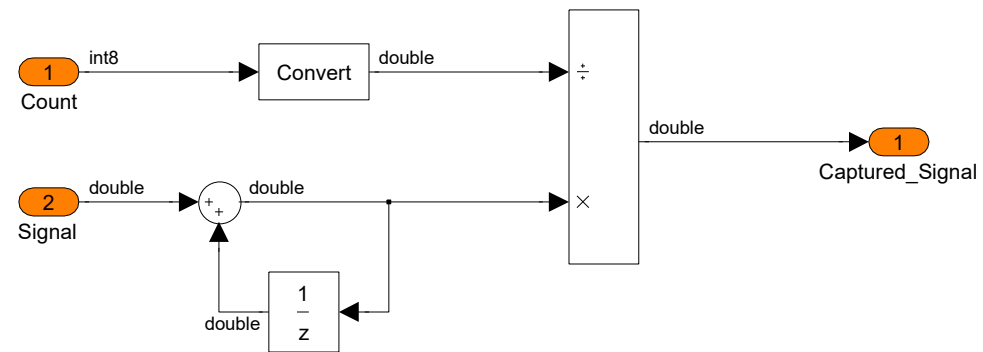


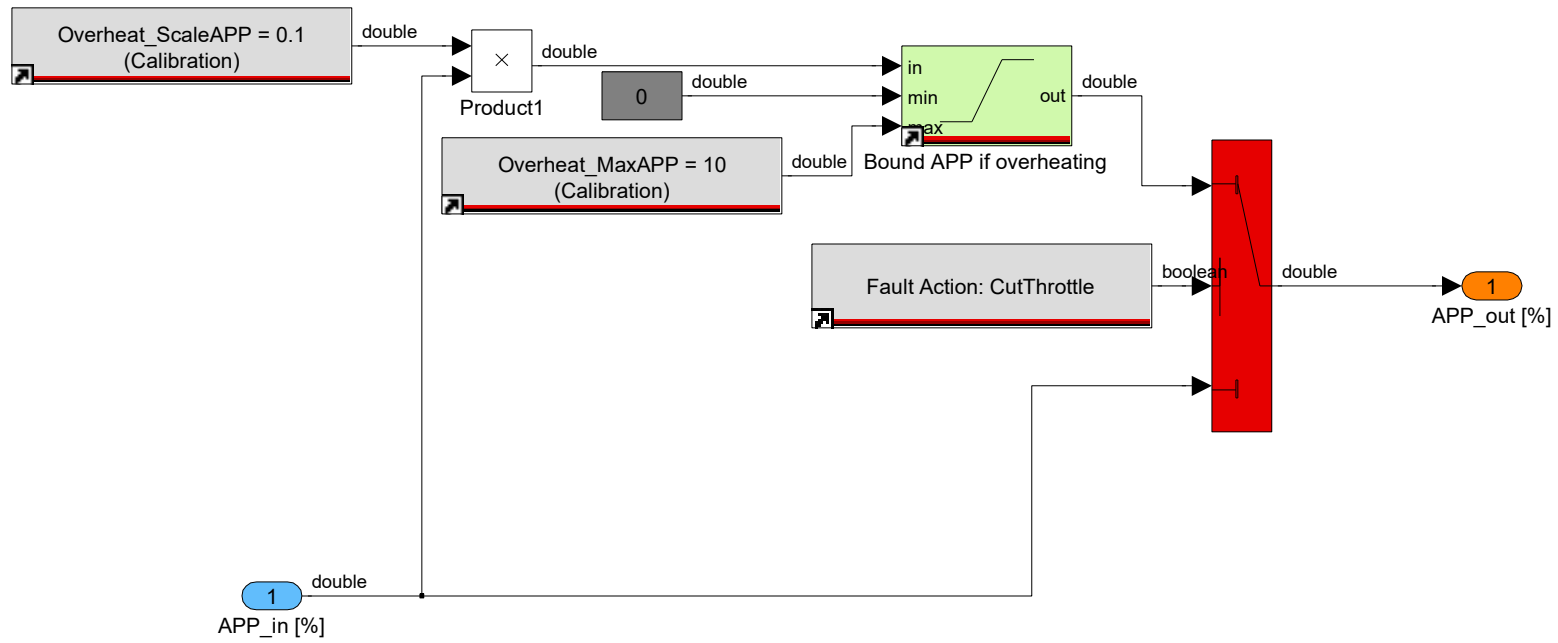
**Collect samples by summing,  
and divide by the total, to obtain average**

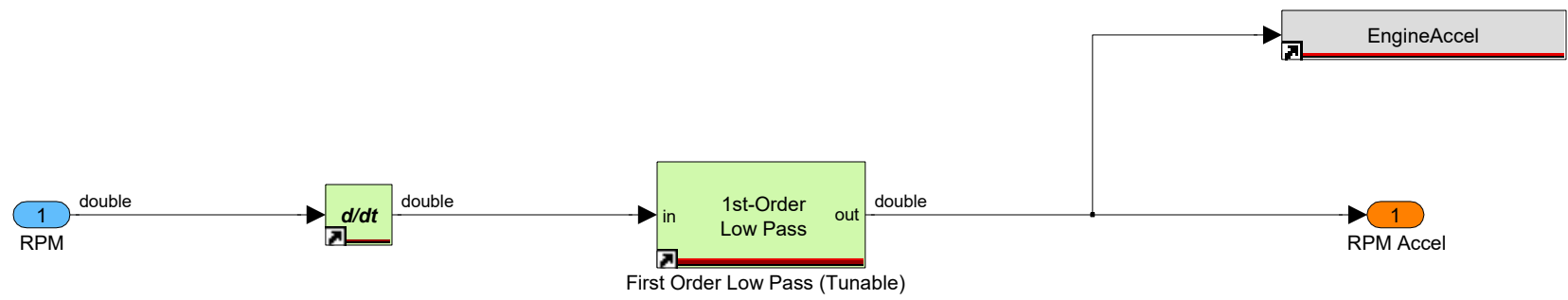


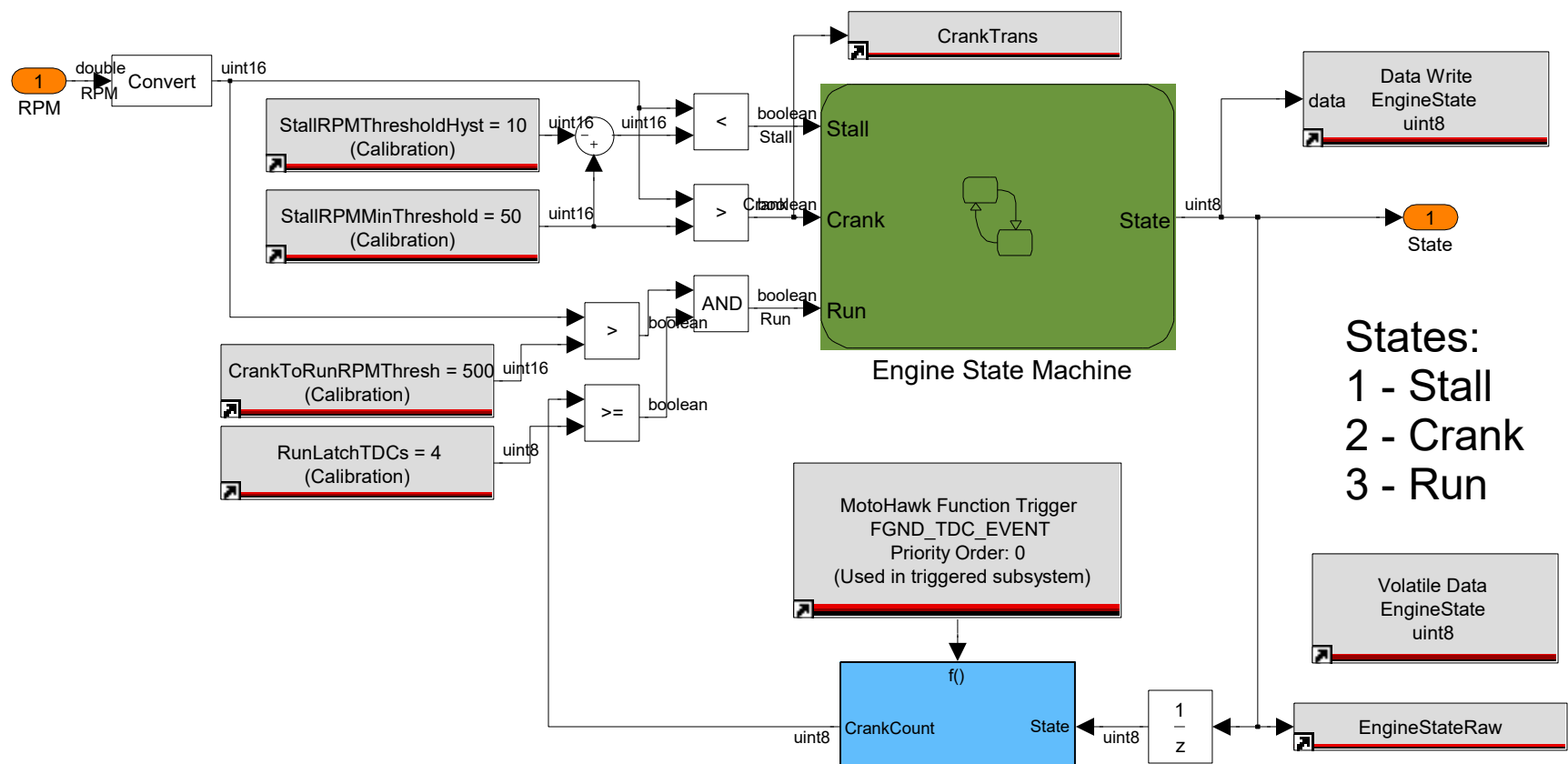


**Collect samples by summing,  
and divide by the total, to obtain average**







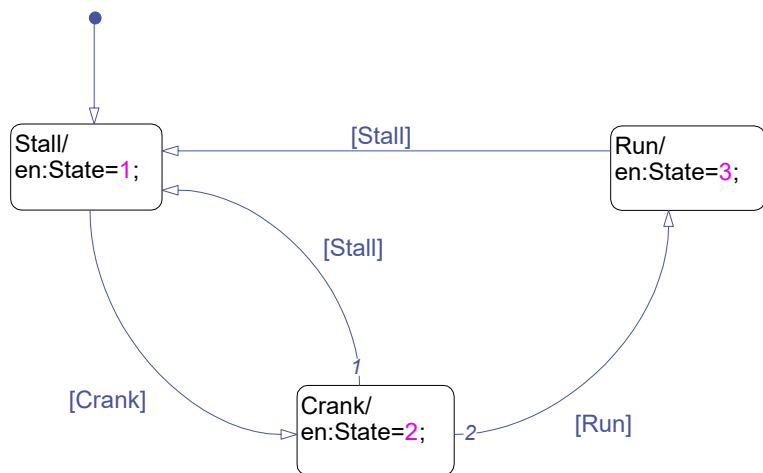


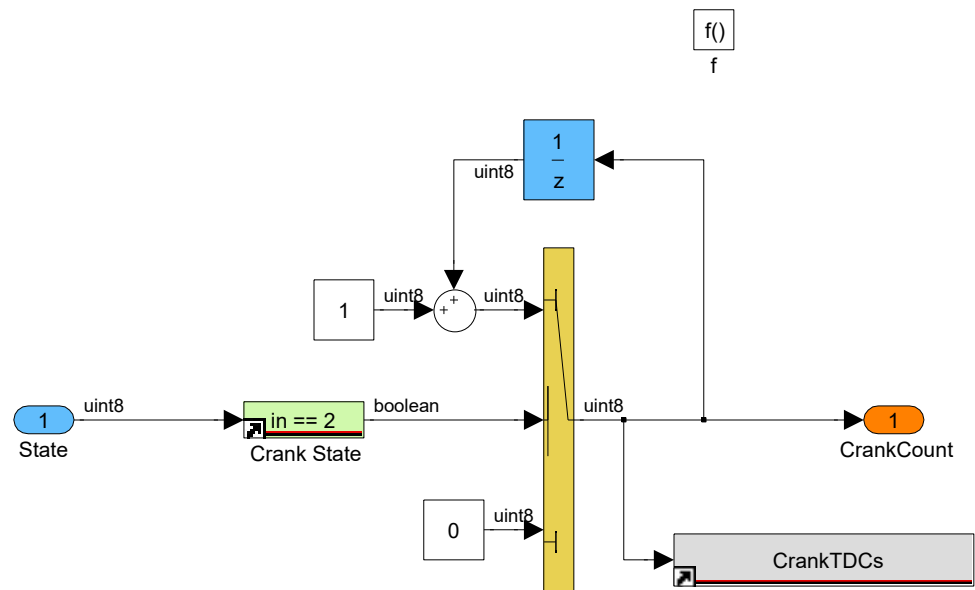


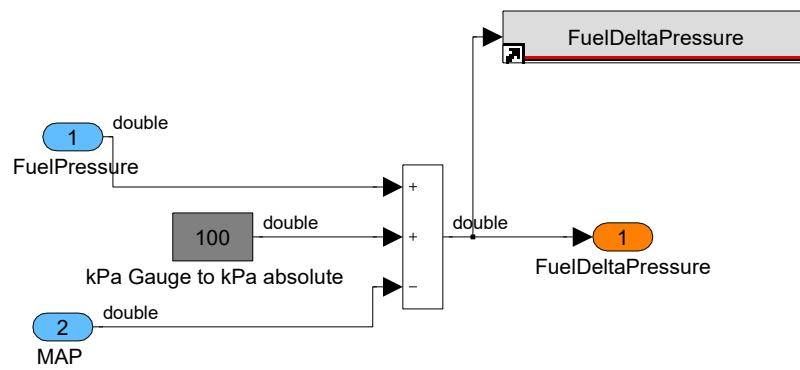
TDC\_Counter

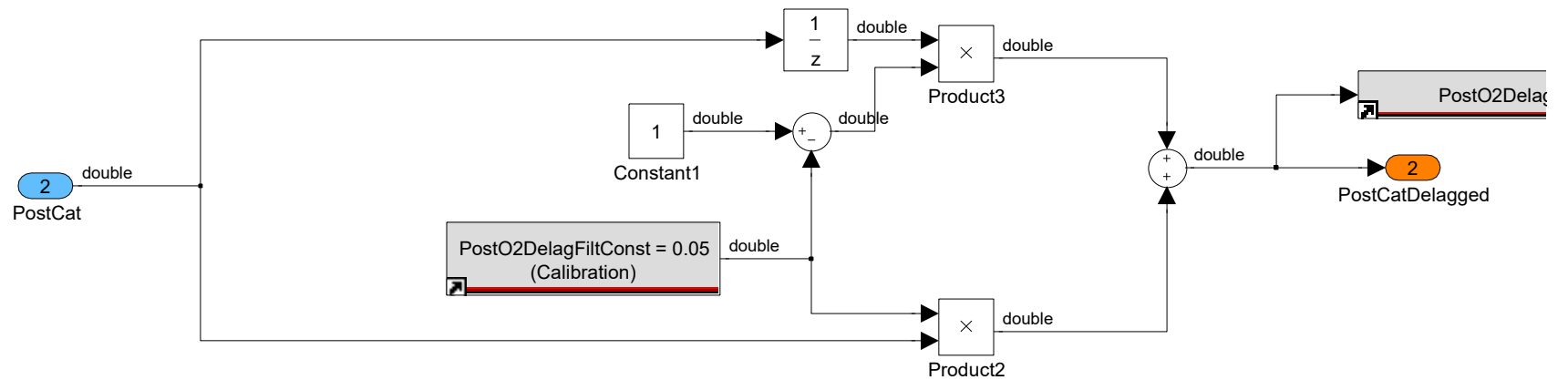
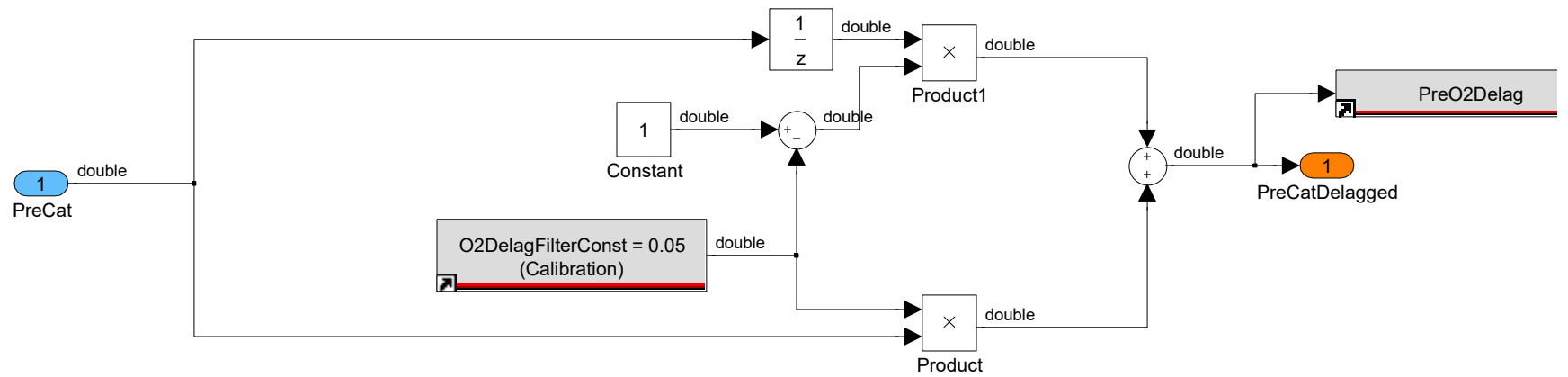




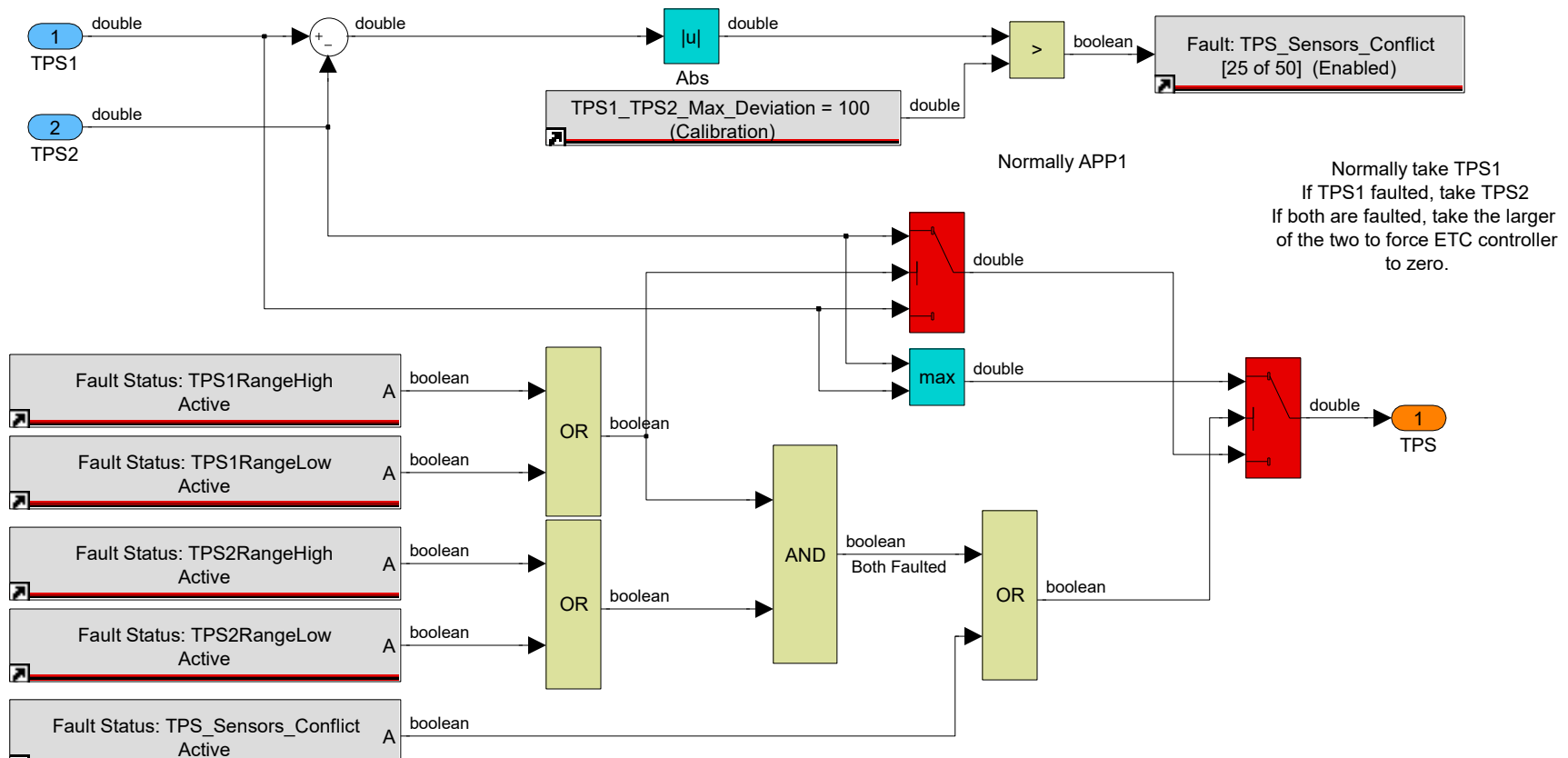




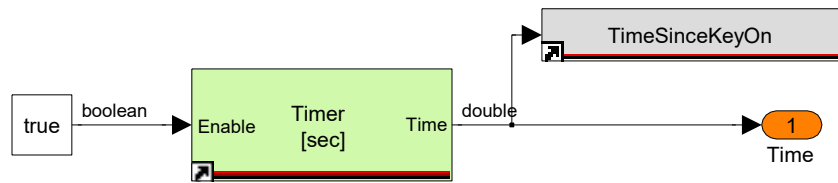




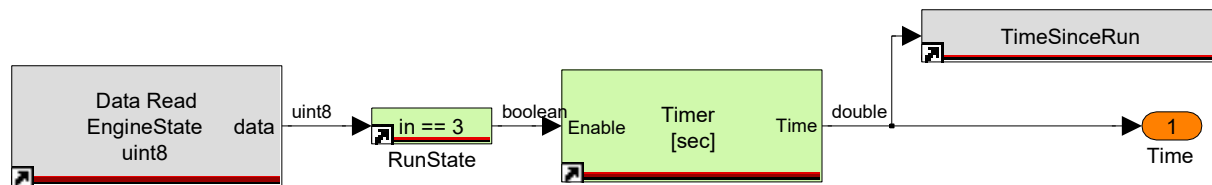


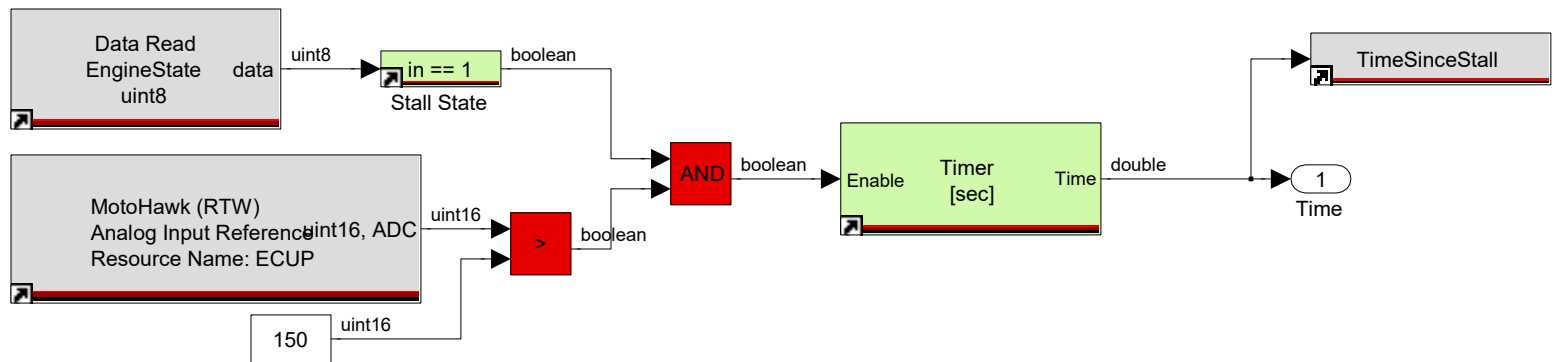


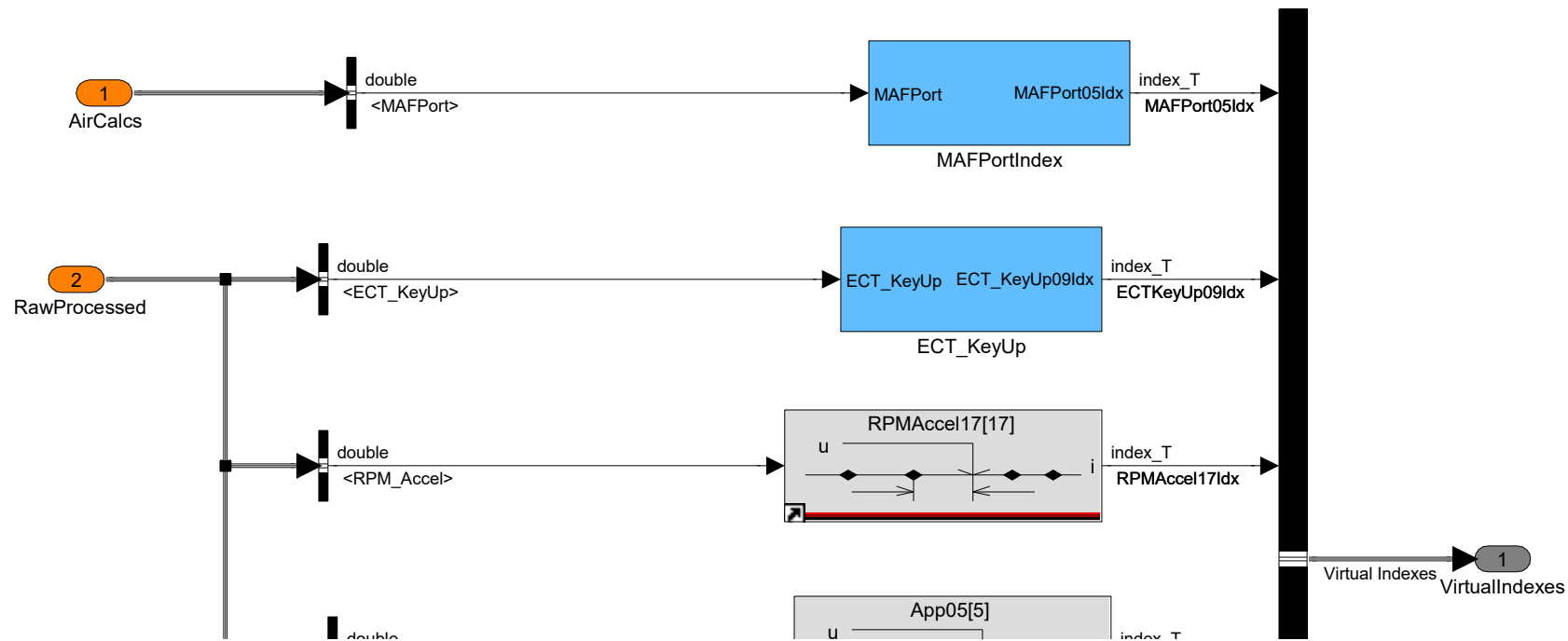


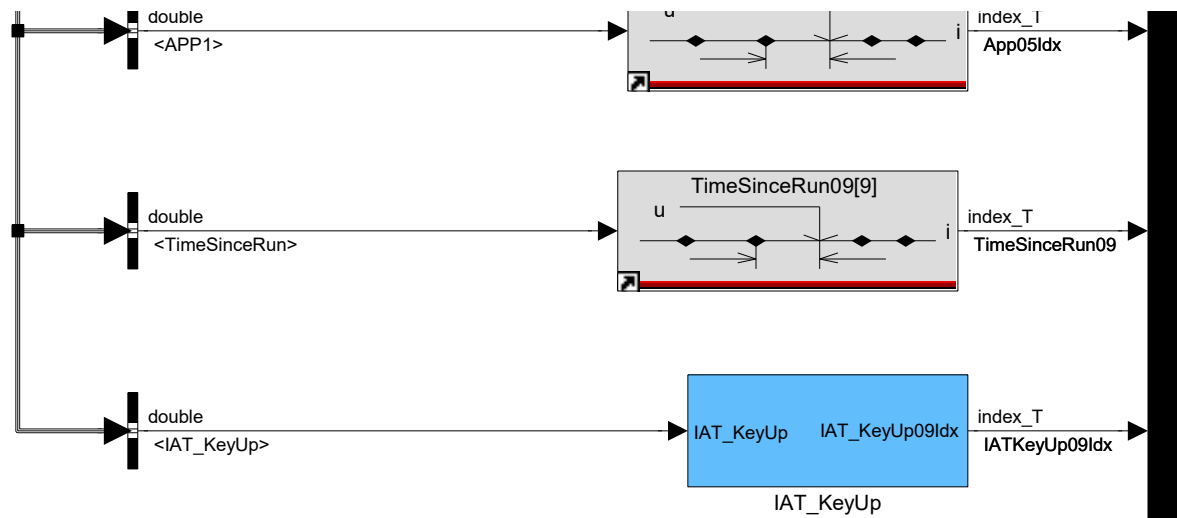


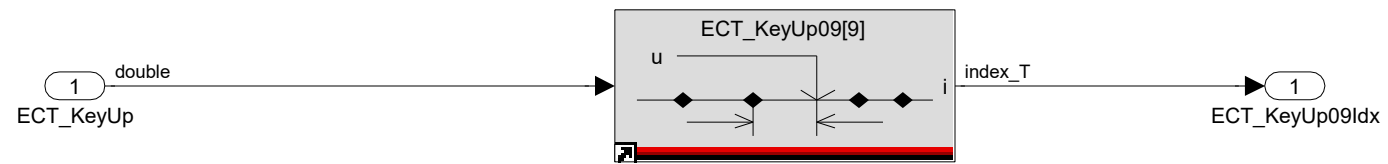


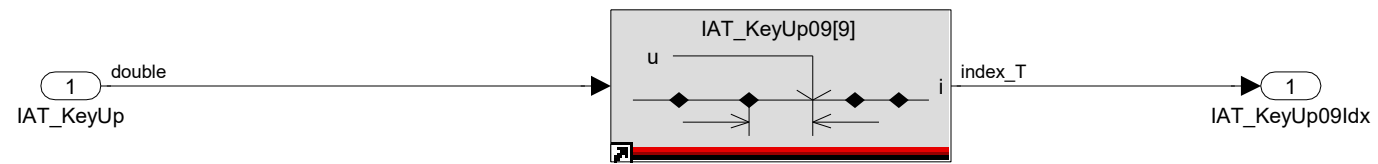


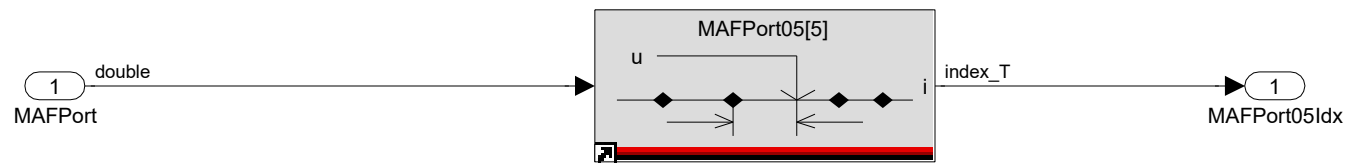


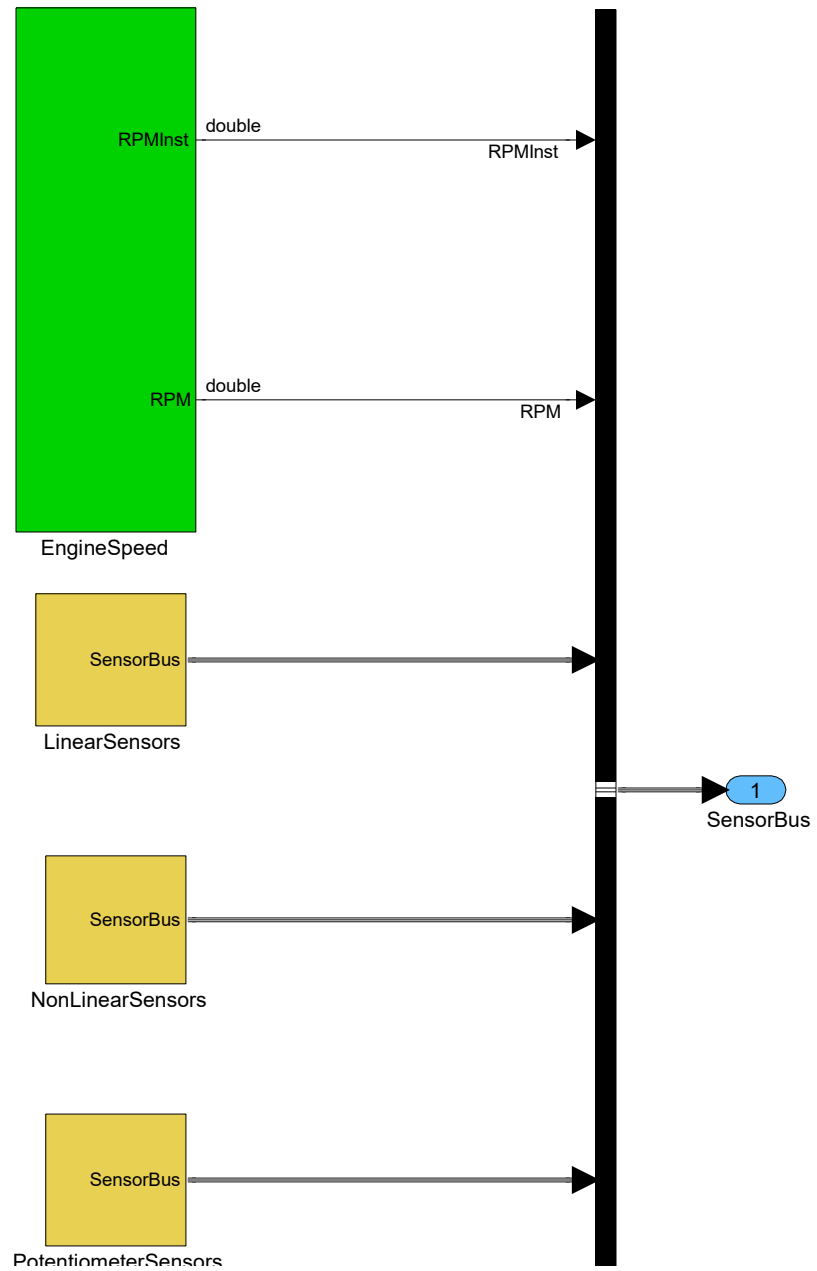






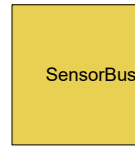








Potentiometer Sensors



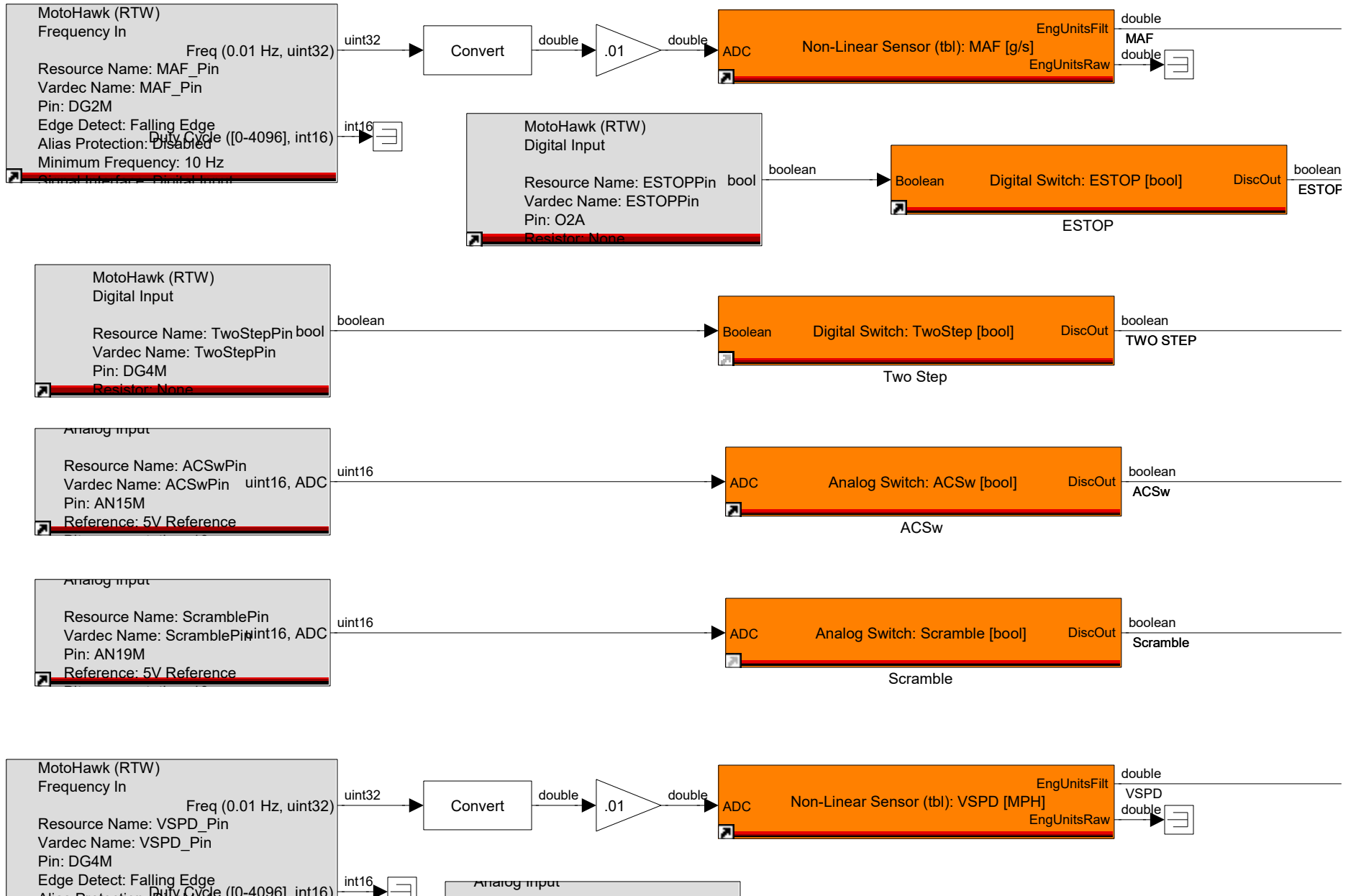
Digital

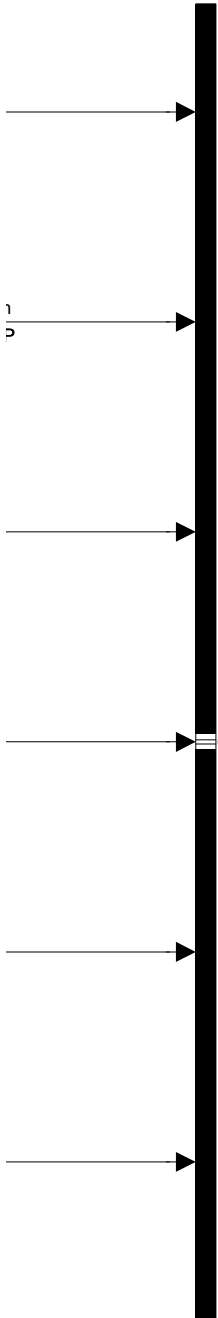


Read from H/W

Convert to Engineering Units

Create a Bus of Sensors





Digital

SensorBus

1

Alias Protection: Disabled

Minimum Frequency: 10 Hz

Resource Name: ClutchSw\_Pin

Vardec Name: ClutchSw\_Pin16, ADC

Pin: AN15M

Reference: 5V Reference

uint16

ADC

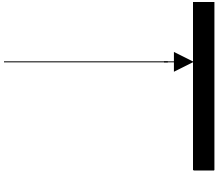
Analog Switch: ClutchSw [bool]

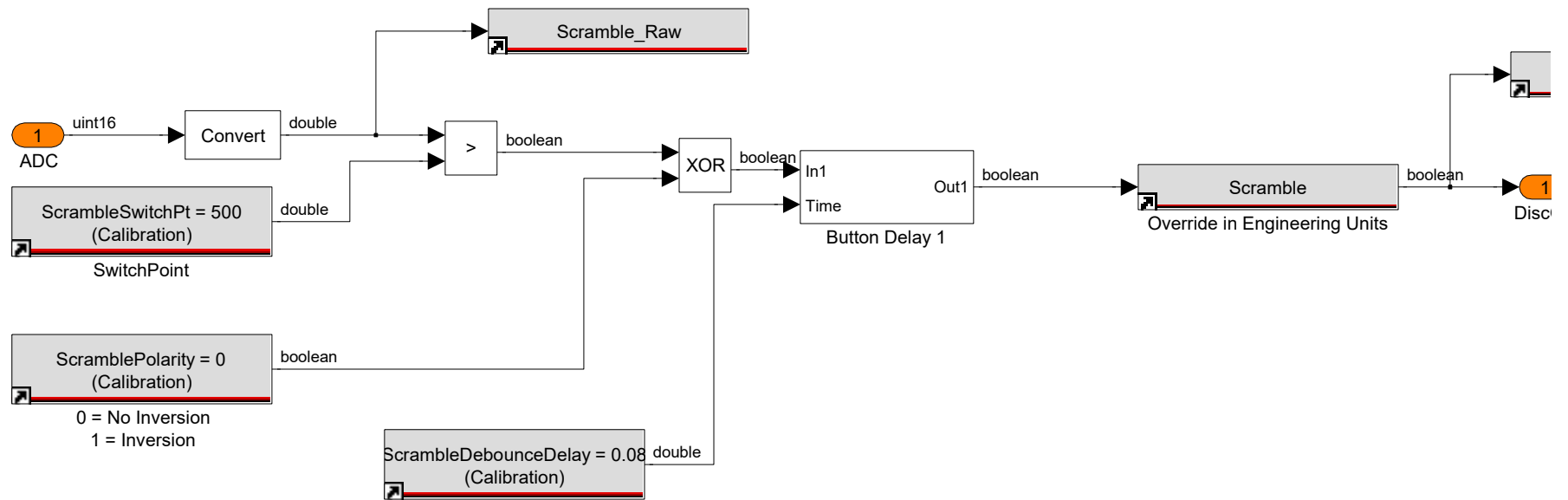
DiscOut

boolean

ClutchSw

ClutchSw

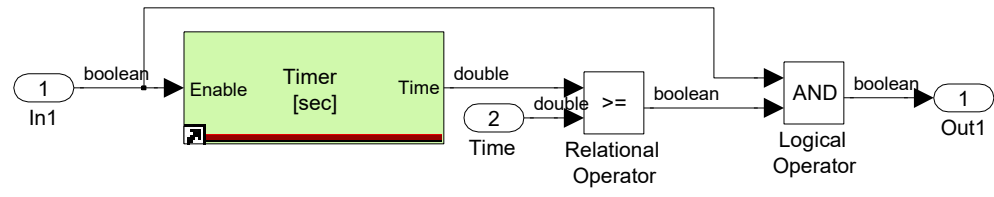




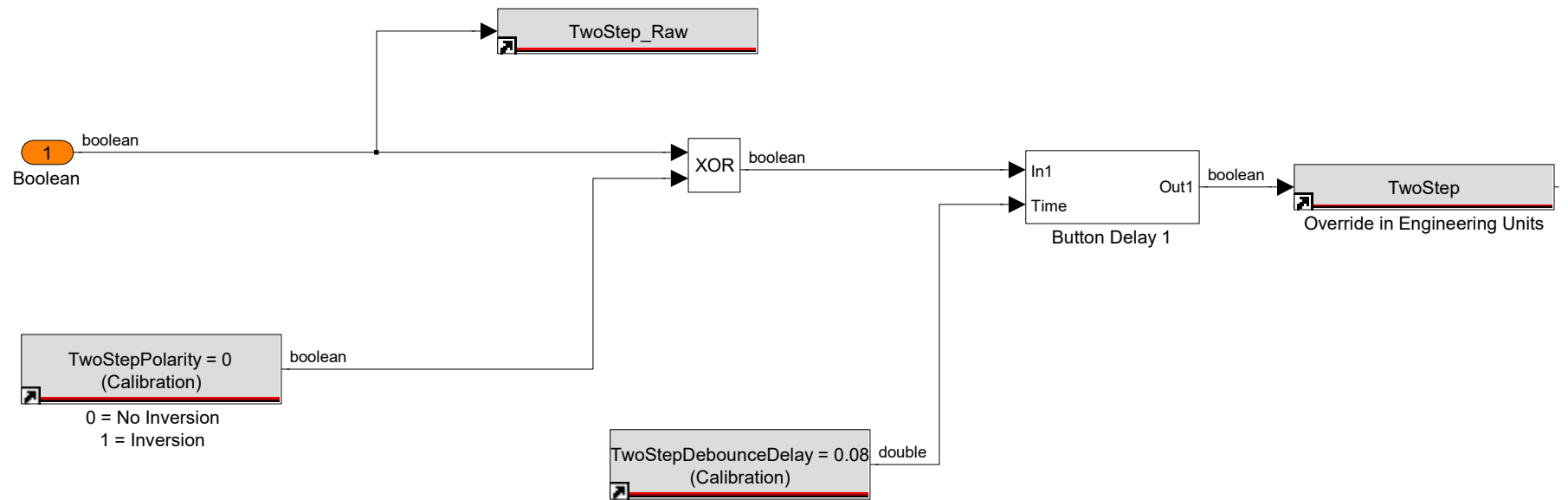
Scramble

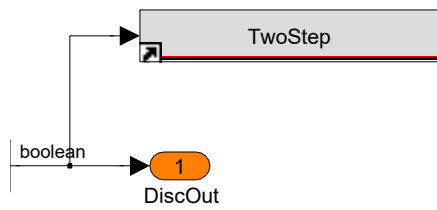


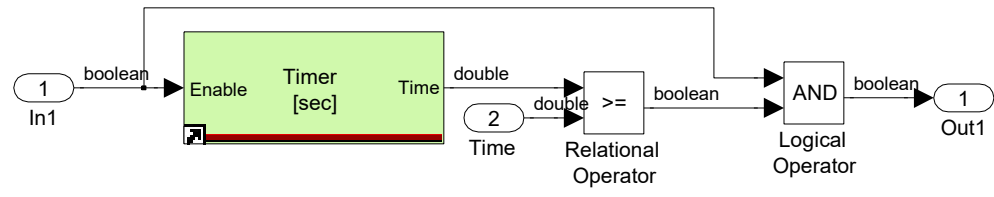
:Out



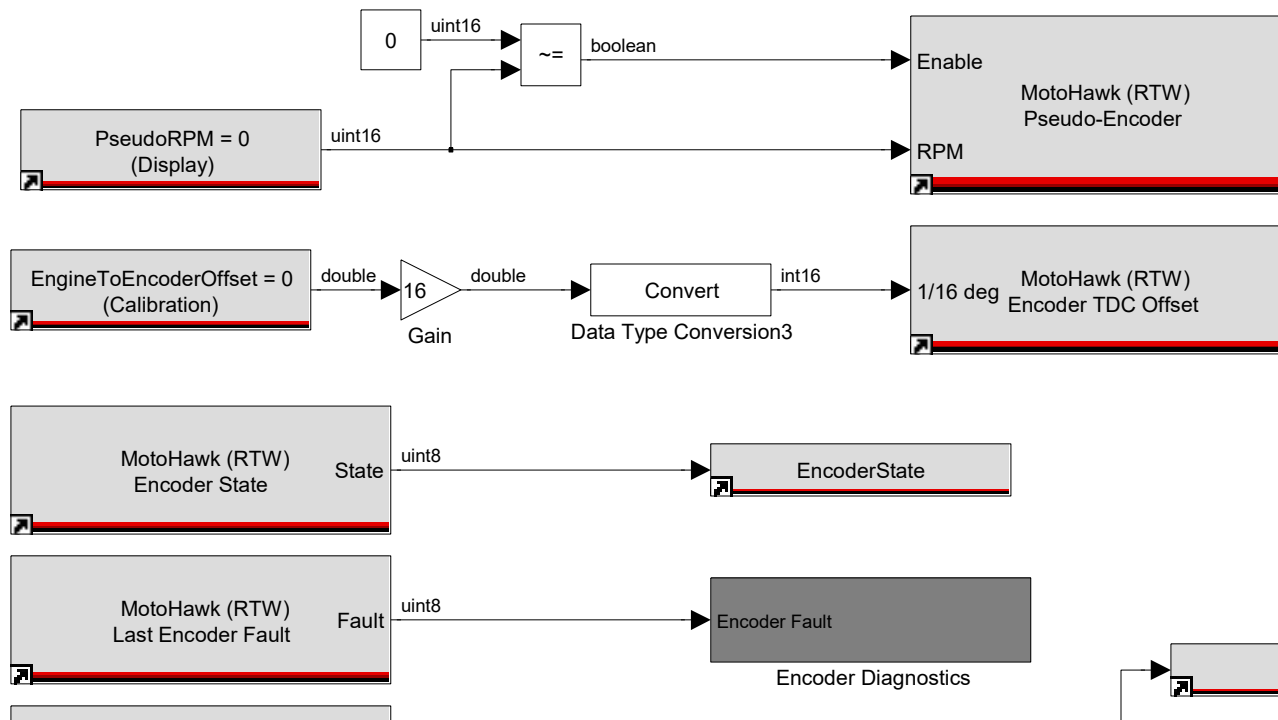








Set PseudoRPM from MotoTune  
to a non-zero value to cause  
simulated RPM

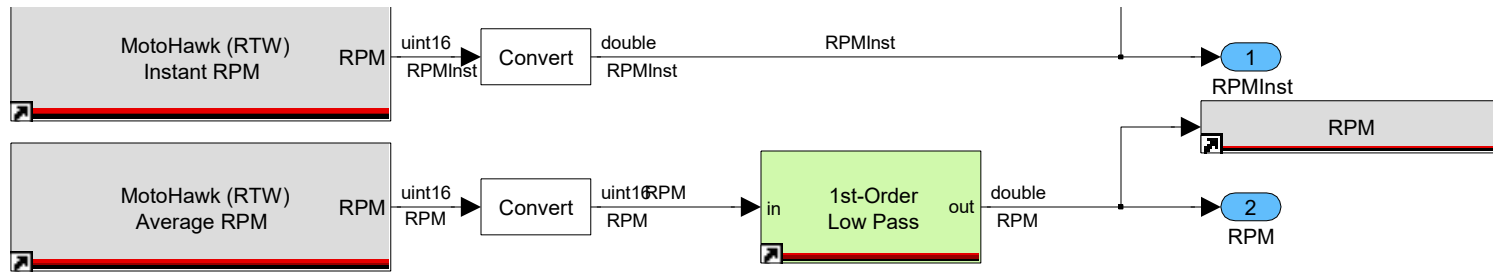


MotoHawk (RTW)  
Encoder Definition

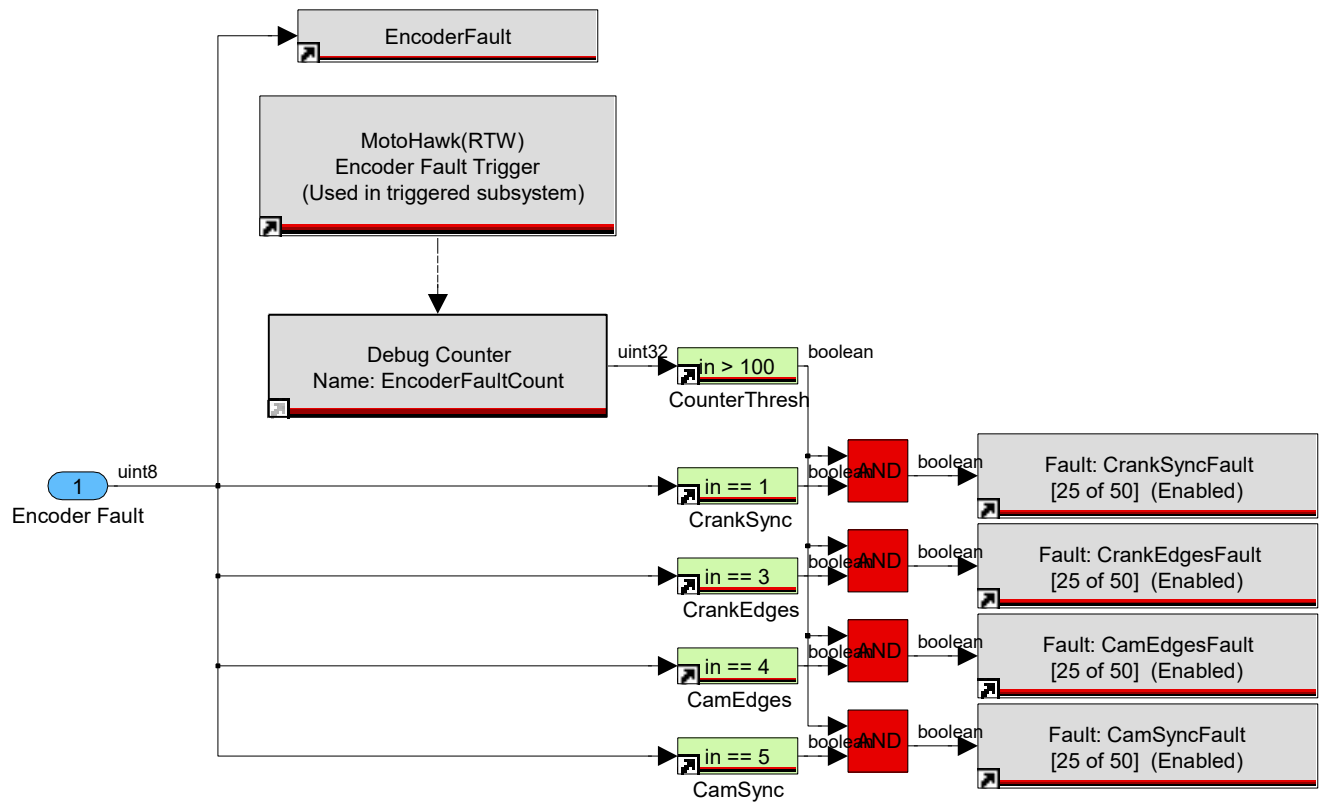
Encoder Type: 24 Minus 1 - 4-Stroke - Half-Mo  
Redundancy: None  
Signal Interface: VR Crank - Digital Can  
Crank Synchronization: Rising Edge  
Crank Pullup: Strong Pullup  
Cam Synchronization: Rising Edge  
Cam Pullup: Strong Pullup  
Number of Cylinders: 4  
TDC Angles: [0 180 360 540 0 0 0  
TDC Angle Offset: 27  
Teeth to ignore before attempting synchroniza  
Cam Delay: 0  
Calibratable Max Cylinders: 8

Archive: TPU\_N\_12.a









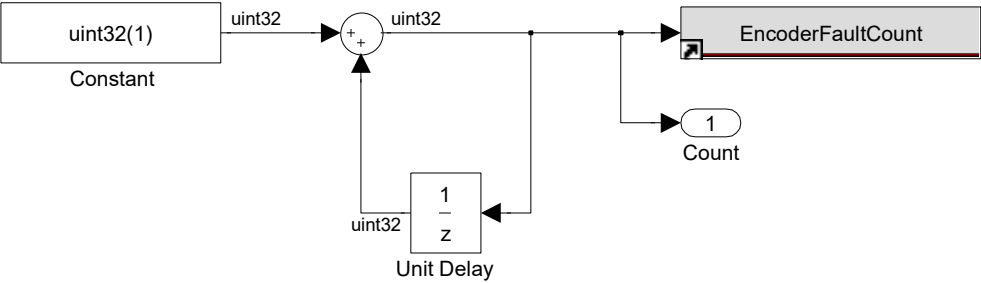


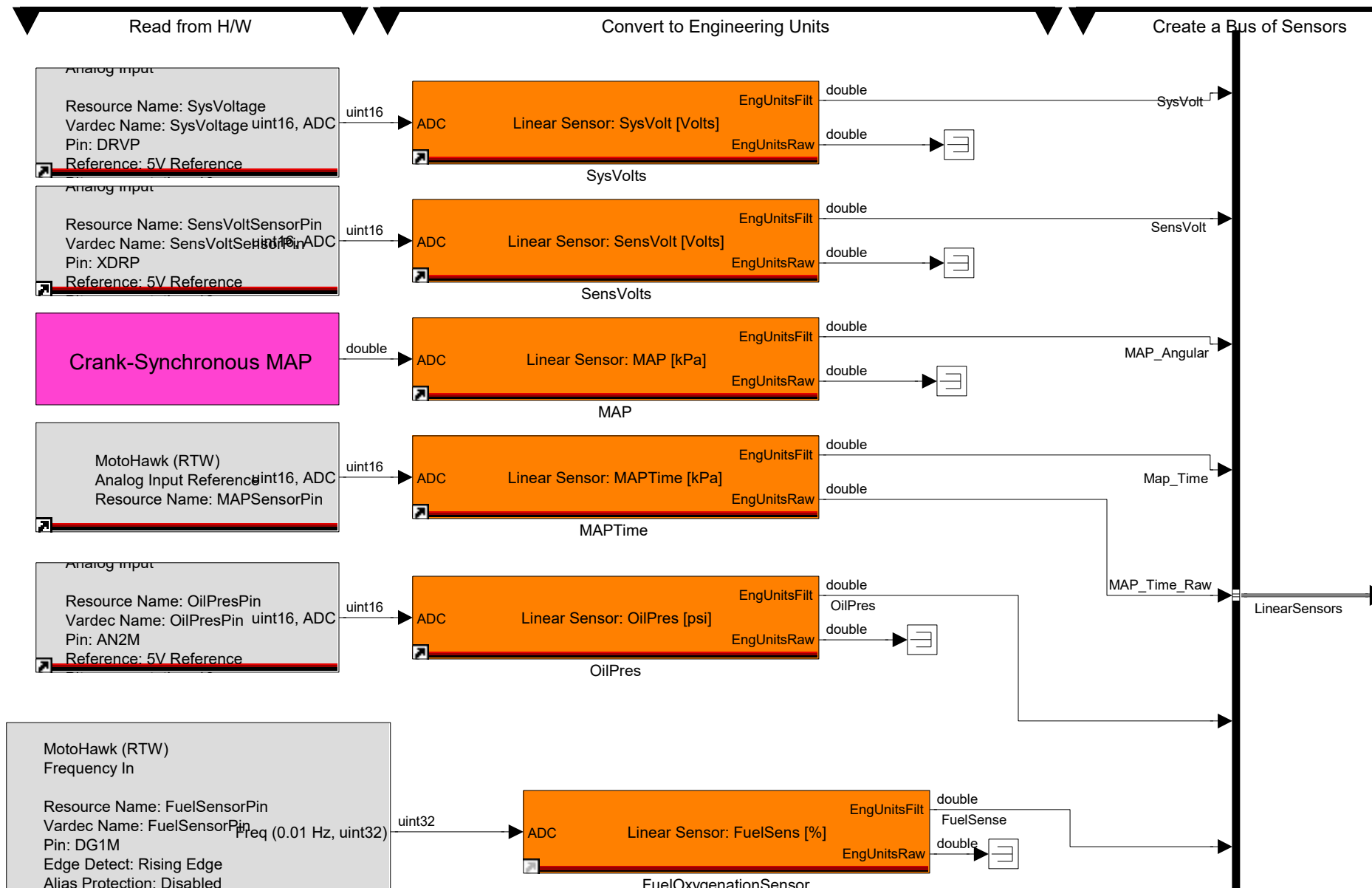
Simple Model Probe

f()

SCIL4

MotoHawk (RTW)  
Code Coverage Test Bit







Alias Protection: Disabled  
Minimum Frequency: 10 Hz  
Signal Interface: Digital Input

FuelOxygenationSensor

Analog Input

Resource Name: FuelPresPin  
Vardec Name: FuelPresPin uint16, ADC  
Pin: AN3M  
Reference: 5V Reference

uint16

ADC Linear Sensor: FuelPres [kPa Gauge]

EngUnitsFilt  
EngUnitsRaw

FuelPres

double

FuelPres

double



Analog Input

Resource Name: ACPresPin  
Vardec Name: ACPresPin uint16, ADC  
Pin: AN8M  
Reference: 5V Reference

uint16

ADC Linear Sensor: ACPres [kPa Gauge]

EngUnitsFilt  
EngUnitsRaw

ACPres

double

ACPres

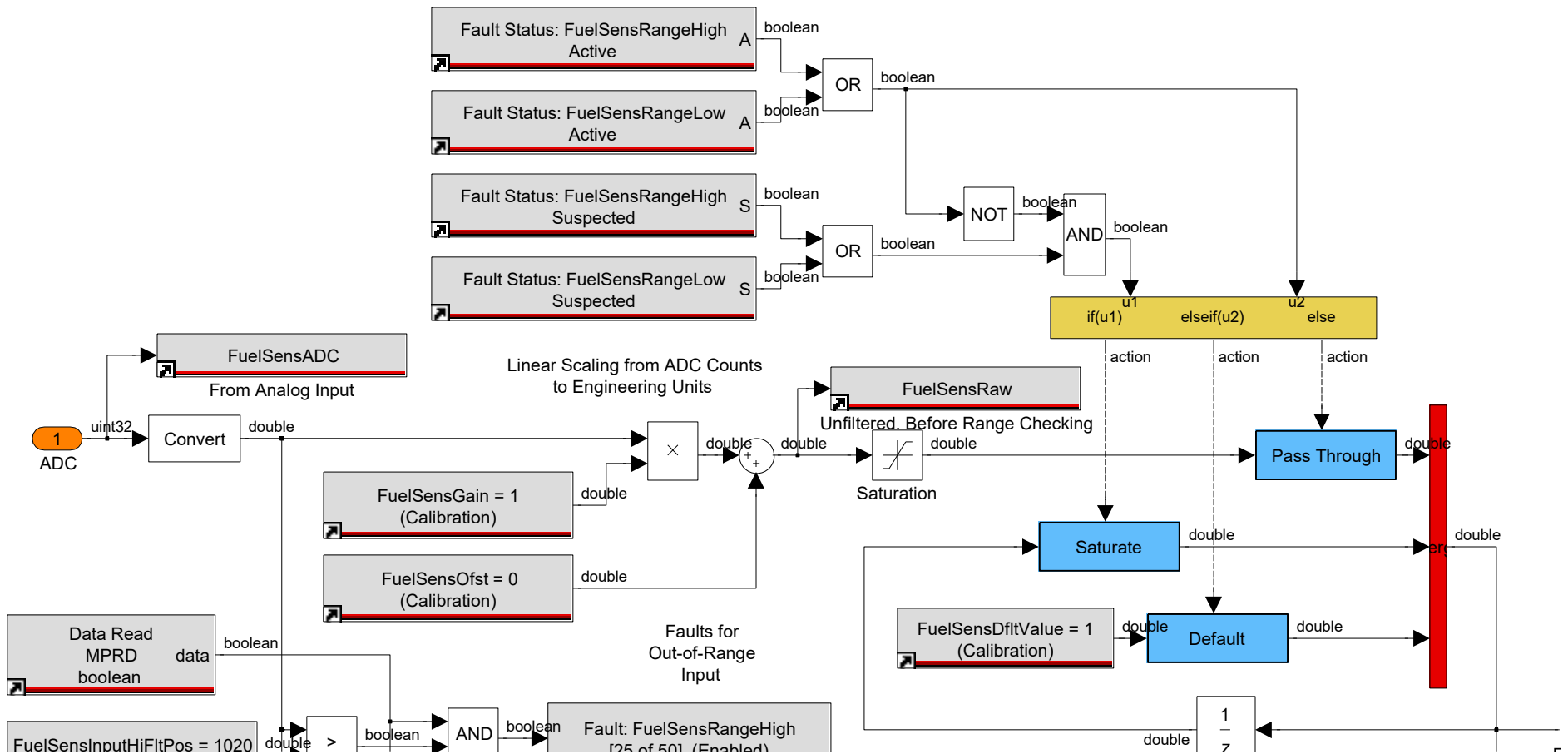
double



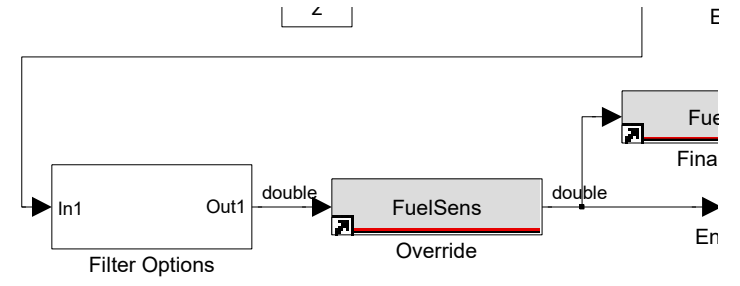
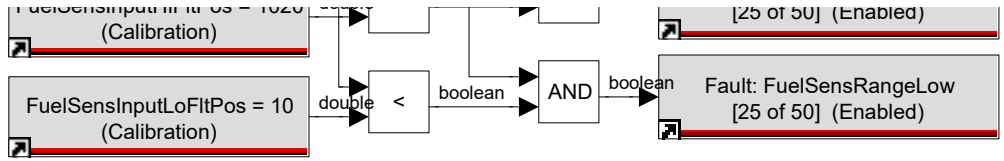
ACPres













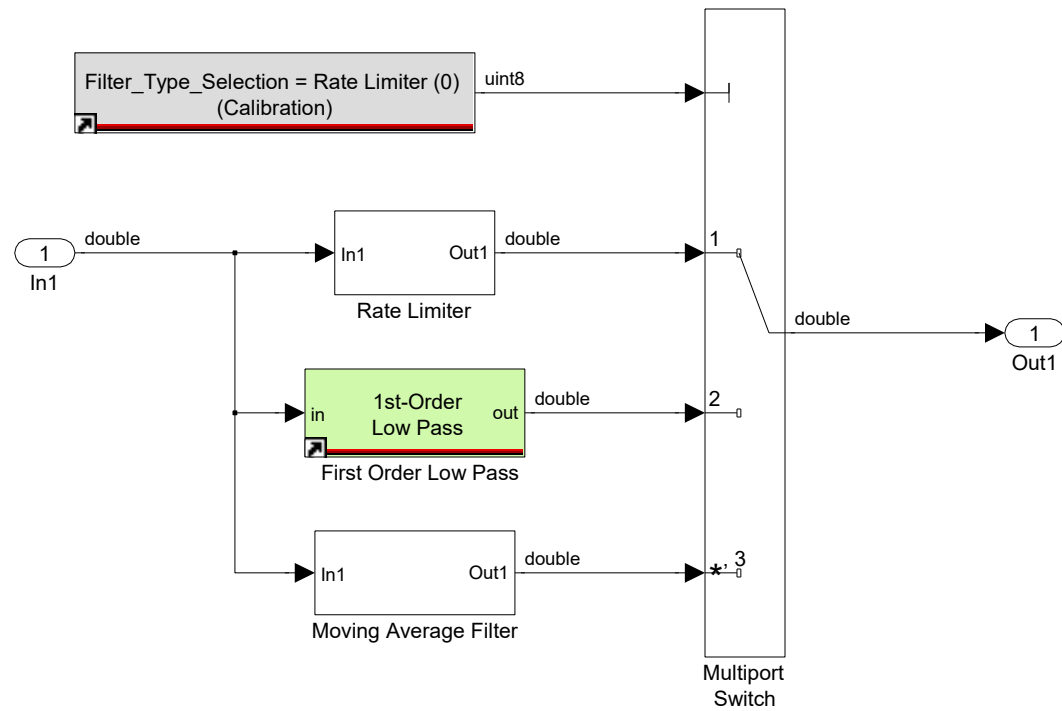
EngUnitsRaw

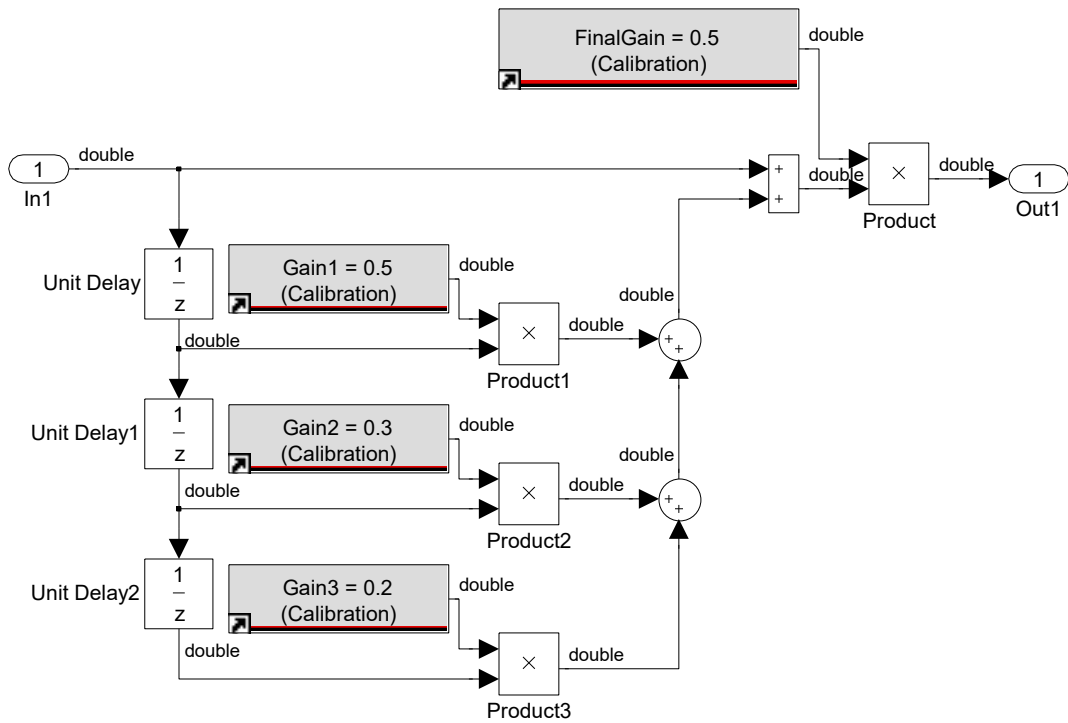
elSens

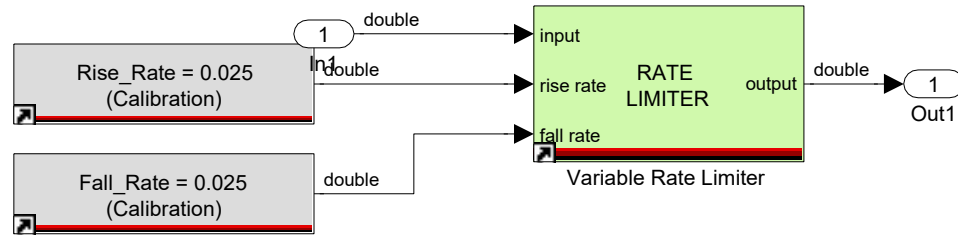
al Value

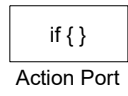
1

ngUnitsFilt



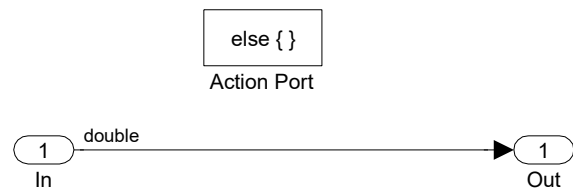


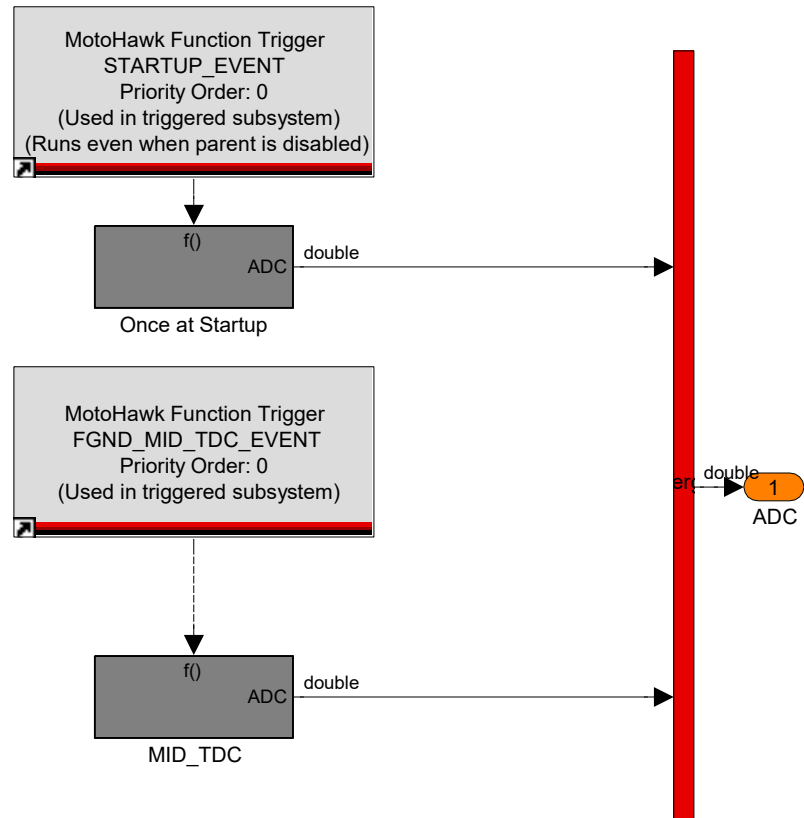




elseif { }  
Action Port



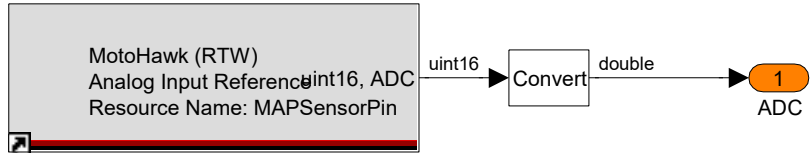








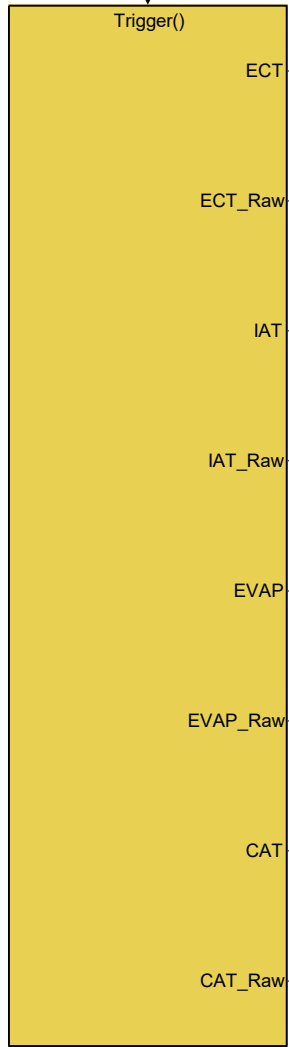
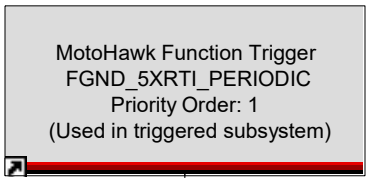
f()  
f



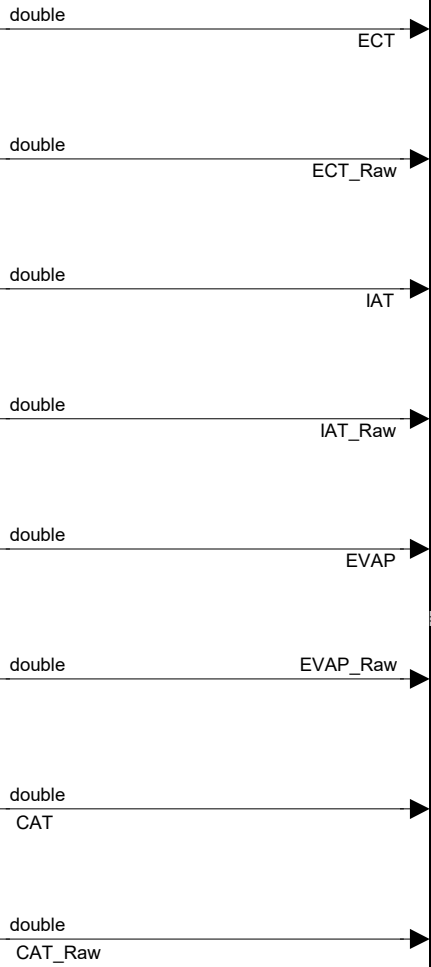
f()  
f

## Sample Crank-Synchronous MAP Once at Startup

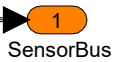




Temp Sensors



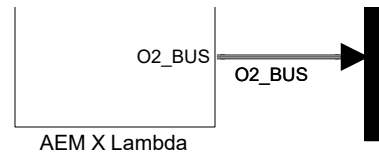
NonLinearSensors



O2\_PreCat

double

O2\_PreCat



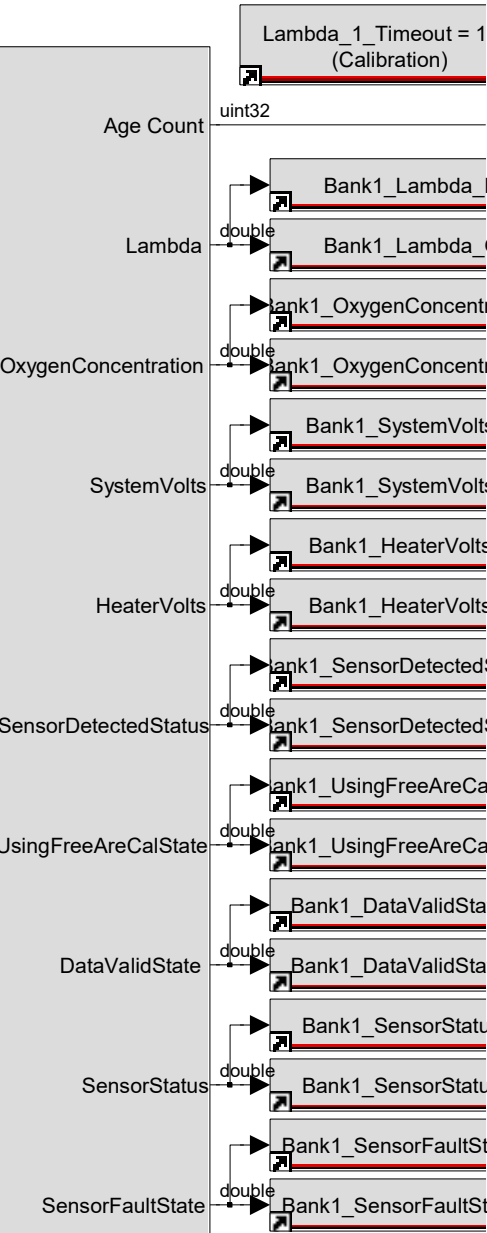
MotoHawk Read CAN Message

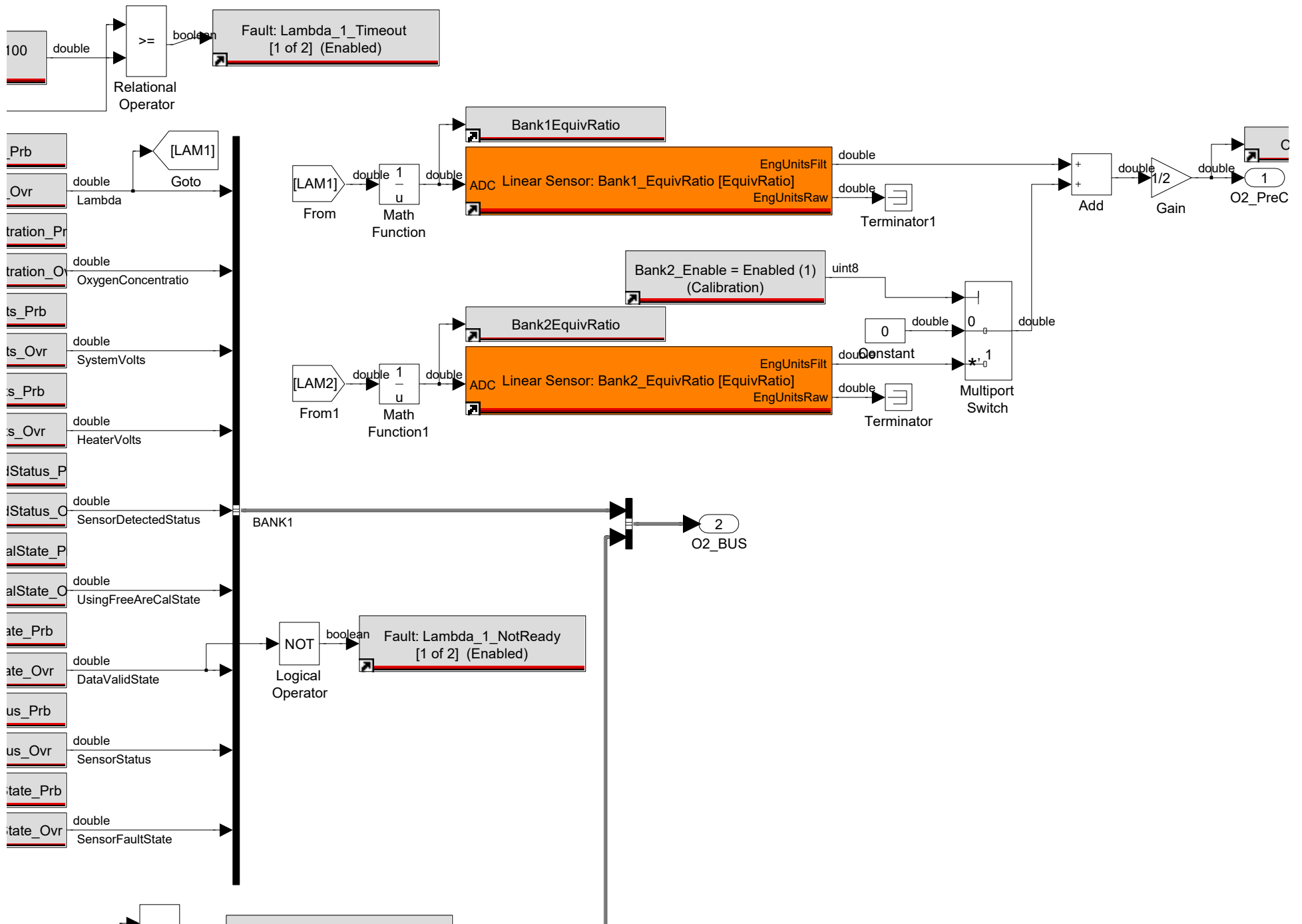
Name: CAN\_1  
Source Module: UEGO  
Interval: 10 ms (100 Hz)  
Queue Size: 1  
ID: 0x00000180 (STANDARD)  
Mask: 0x000007ff  
RTR: 0

Message: AEM X-Series UEGO  
Description: UEGO Num1  
Payload Size: 8

Payload Contents:

Name	Units	LSB	Len	Type	Byte Order	Gain	Offset
Lambda	Lam	48	16	UNSIGNED	BIG_ENDIAN	1/10000.000	0.000
OxygenConcentration	%	32	16	SIGNED	BIG_ENDIAN	1/1000.000	0.000
SystemVolts	V	24	8	UNSIGNED	BIG_ENDIAN	1/10.000	0.000
HeaterVolts	V	16	8	UNSIGNED	BIG_ENDIAN	1/10.000	0.000
SensorDetectedStatus	enum	8	4	UNSIGNED	BIG_ENDIAN	1.000	0.000
UsingFreeAreCalState	On/Off	13	1	UNSIGNED	BIG_ENDIAN	1.000	0.000
DataValidState	On/Off	15	1	UNSIGNED	BIG_ENDIAN	1.000	0.000
SensorStatus	enum	0	5	UNSIGNED	BIG_ENDIAN	1.000	0.000
SensorFaultState	On/Off	6	1	UNSIGNED	BIG_ENDIAN	1.000	0.000





O2\_EquivRatio\_Avg

.

at



## MotoHawk Read CAN Message

**Name:** CAN\_1  
**Source Module:** UEGO  
**Interval:** 10 ms (100 Hz)  
**Queue Size:** 1  
**ID:** 0x00000181 (STANDARD)  
**Mask:** 0x000007ff  
**RTR:** 0

**Message:** AEM X-Series UEGO  
**Description:** UEGO Num2  
**Payload Size:** 8

### Payload Contents:

Name	Units	LSB	Len	Type	Byte Order	Gain	Offset
Lambda	Lam	48	16	UNSIGNED	BIG_ENDIAN	1/10000.000	0.000
OxygenConcentration	%	32	16	SIGNED	BIG_ENDIAN	1/1000.000	0.000
SystemVolts	V	24	8	UNSIGNED	BIG_ENDIAN	1/10.000	0.000
HeaterVolts	V	16	8	UNSIGNED	BIG_ENDIAN	1/10.000	0.000
SensorDetectedStatus	enum	8	4	UNSIGNED	BIG_ENDIAN	1.000	0.000
UsingFreeAreCalState	On/Off	13	1	UNSIGNED	BIG_ENDIAN	1.000	0.000
DataValidState	On/Off	15	1	UNSIGNED	BIG_ENDIAN	1.000	0.000
SensorStatus	enum	0	5	UNSIGNED	BIG_ENDIAN	1.000	0.000
SensorFaultState	On/Off	6	1	UNSIGNED	BIG_ENDIAN	1.000	0.000

Lambda\_2\_Timeout =  
(Calibration)

uint32

Age Count

Bank2\_Lambda

Lambda

Bank2\_Lambda

Bank2\_OxygenConce

OxygenConcentration

Bank2\_OxygenConce

Bank2\_SystemVo

SystemVolts

Bank2\_SystemVo

Bank2\_HeaterVo

HeaterVolts

Bank2\_HeaterVo

Bank2\_SensorDetecte

SensorDetectedStatus

Bank2\_SensorDetecte

Bank2\_UsingFreeAreC

UsingFreeAreCalState

Bank2\_UsingFreeAreC

Bank2\_DataValidS

DataValidState

Bank2\_DataValidS

Bank2\_SensorSta

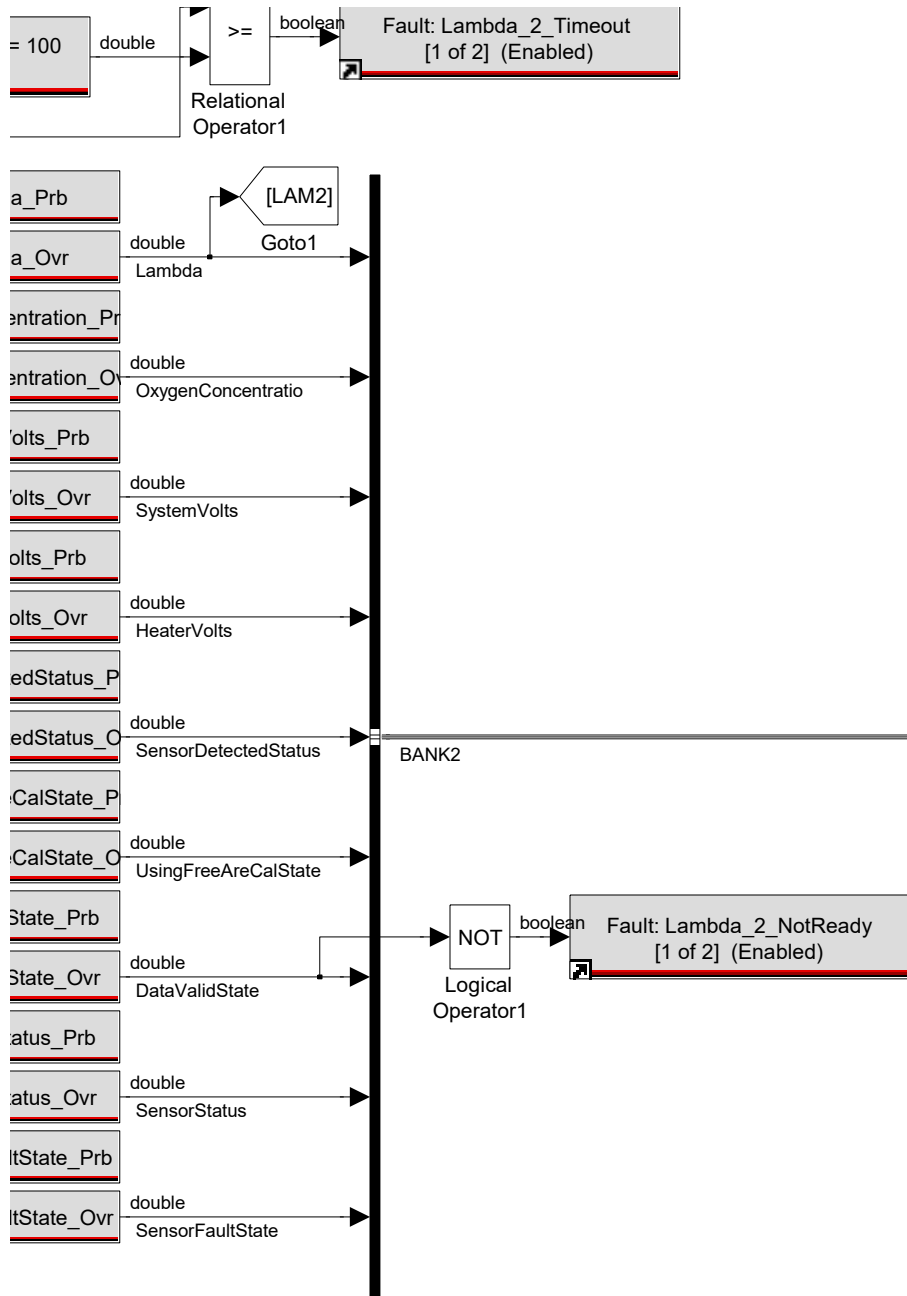
SensorStatus

Bank2\_SensorSta

Bank2\_SensorFault

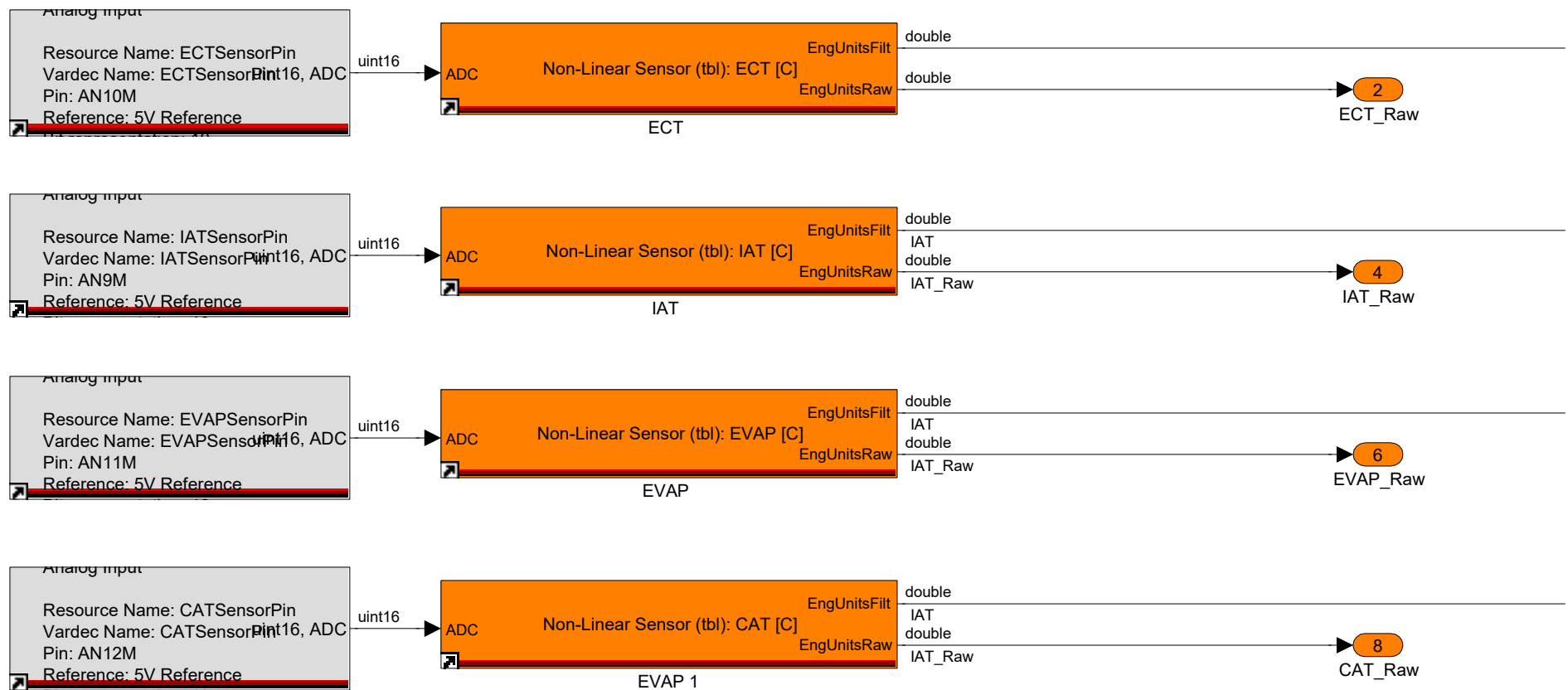
SensorFaultState

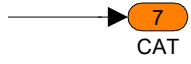
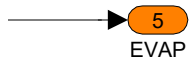
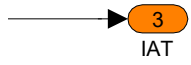
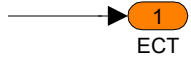
Bank2\_SensorFault

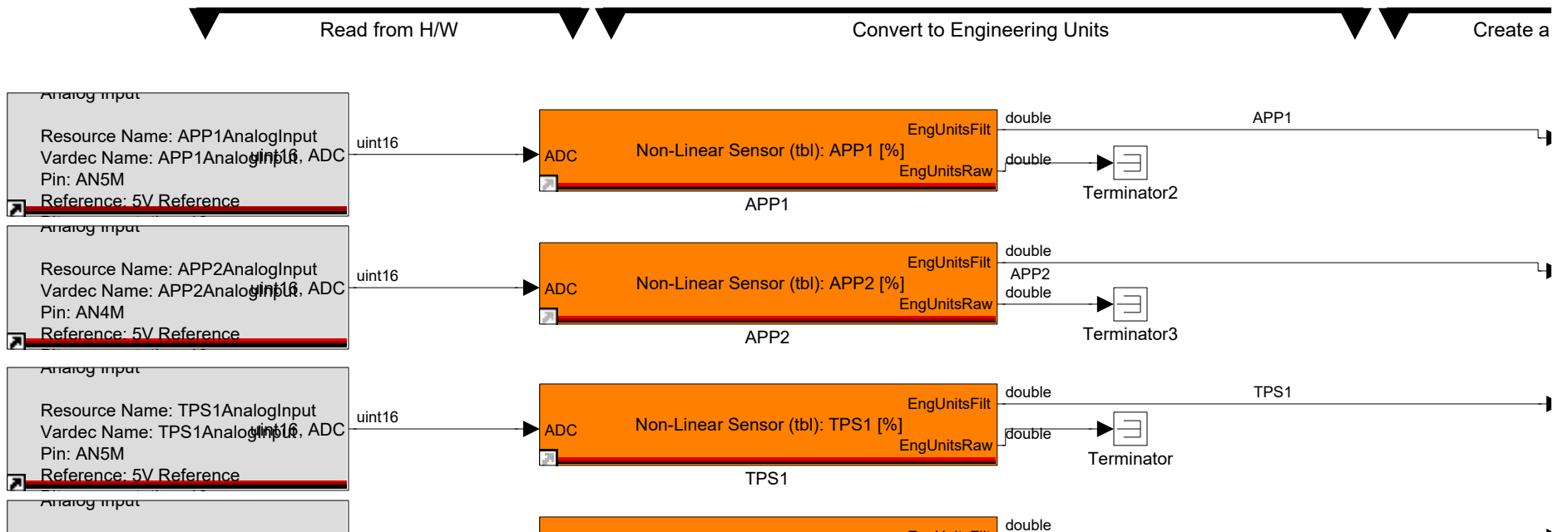




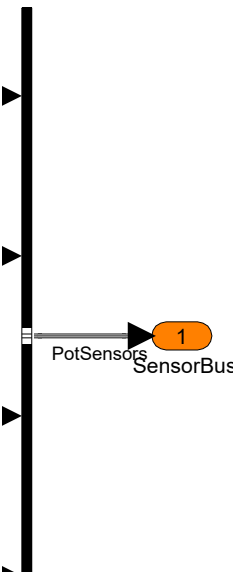
f()







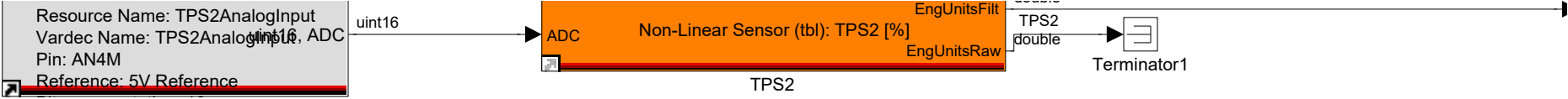
Bus of Sensors



PotSensors

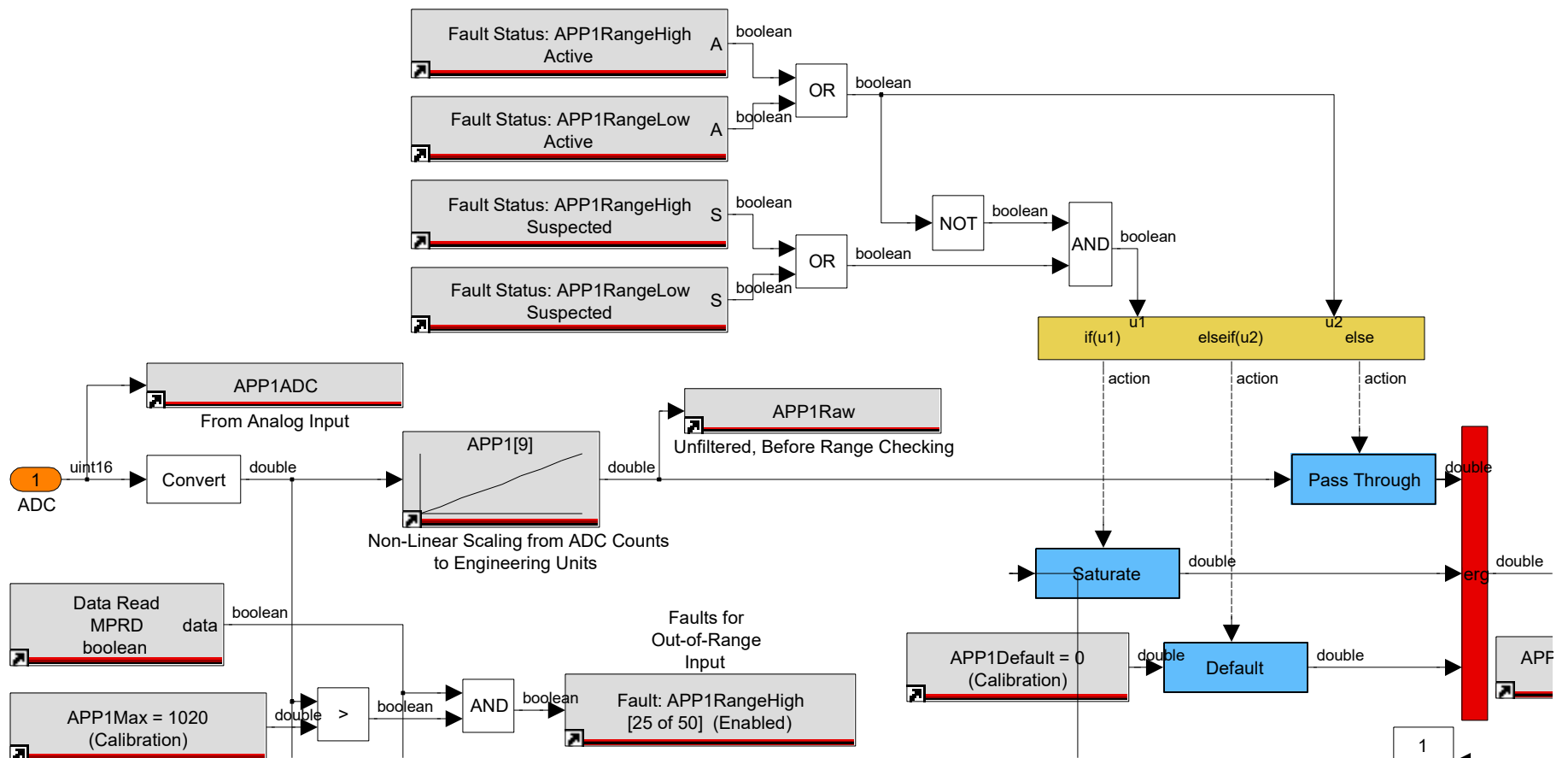
1

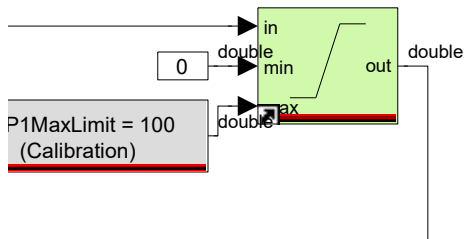
SensorBus

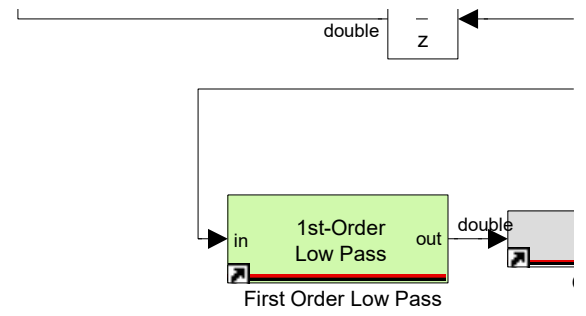
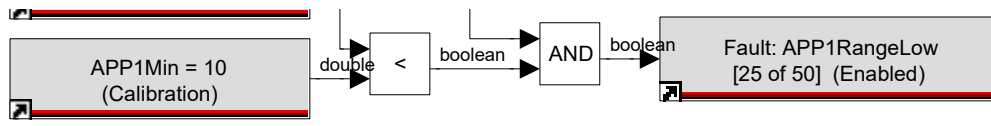


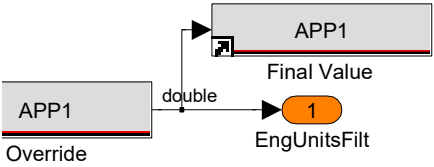


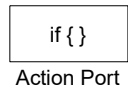






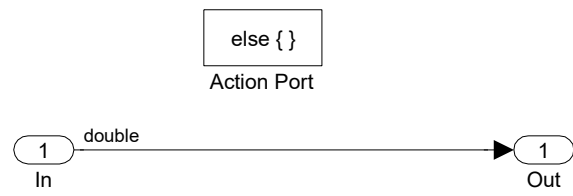




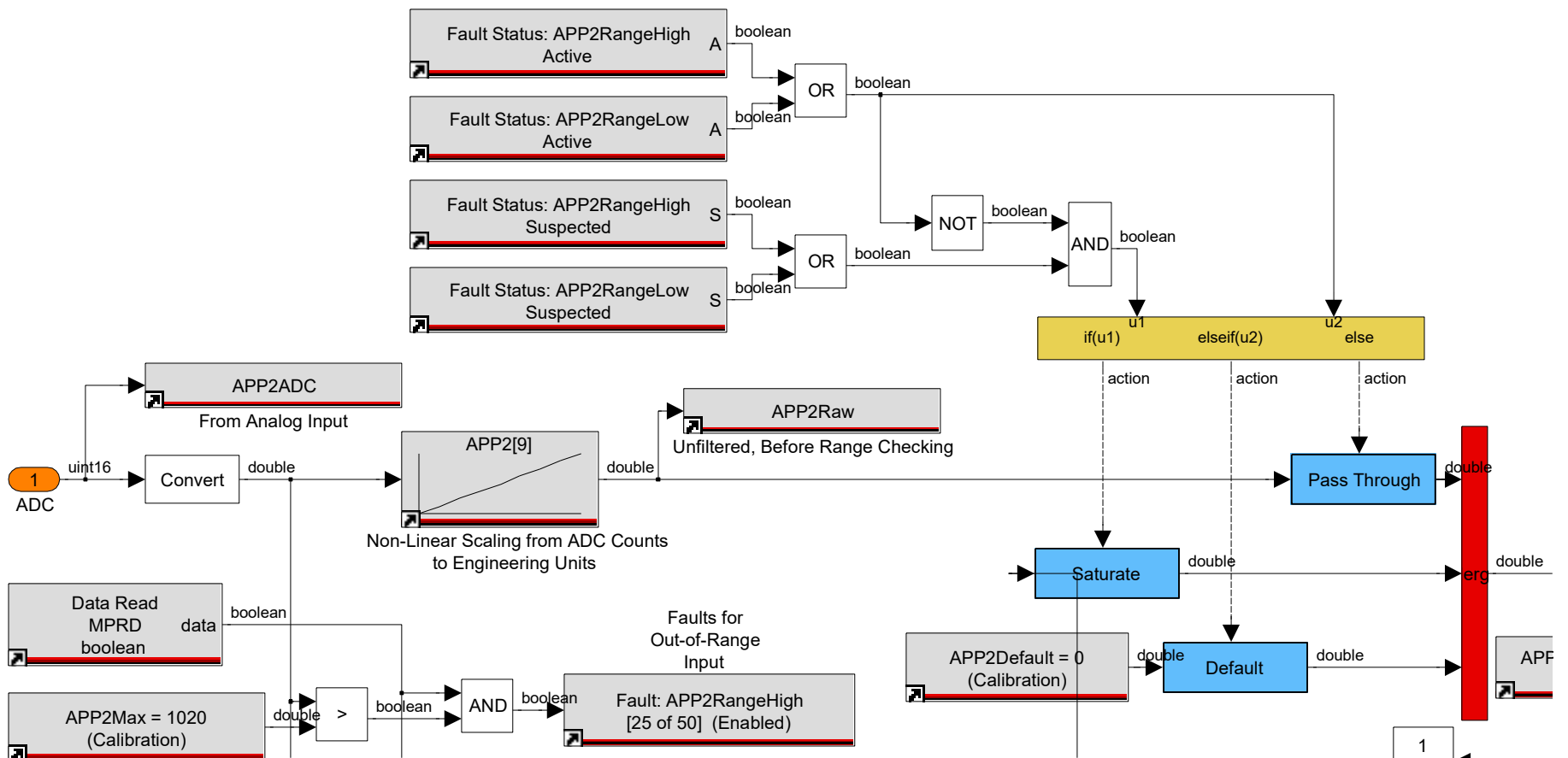


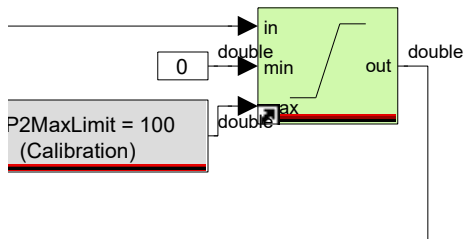
elseif { }  
Action Port

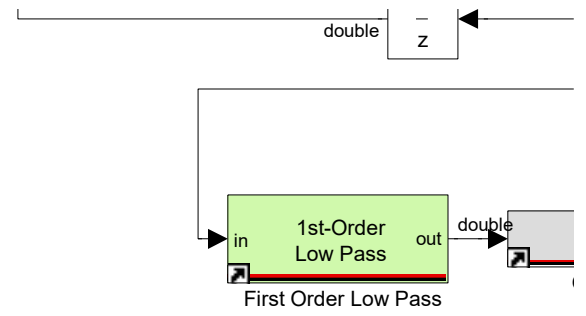
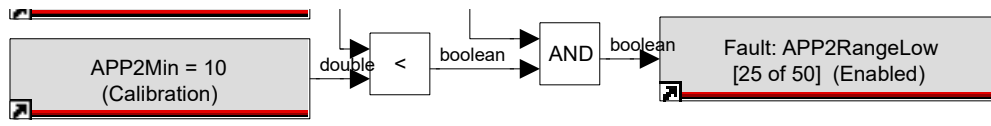


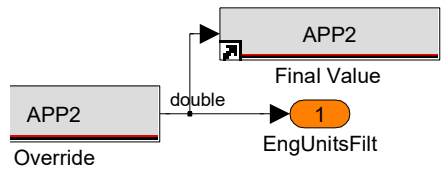


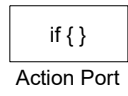






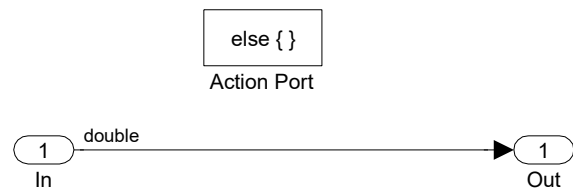


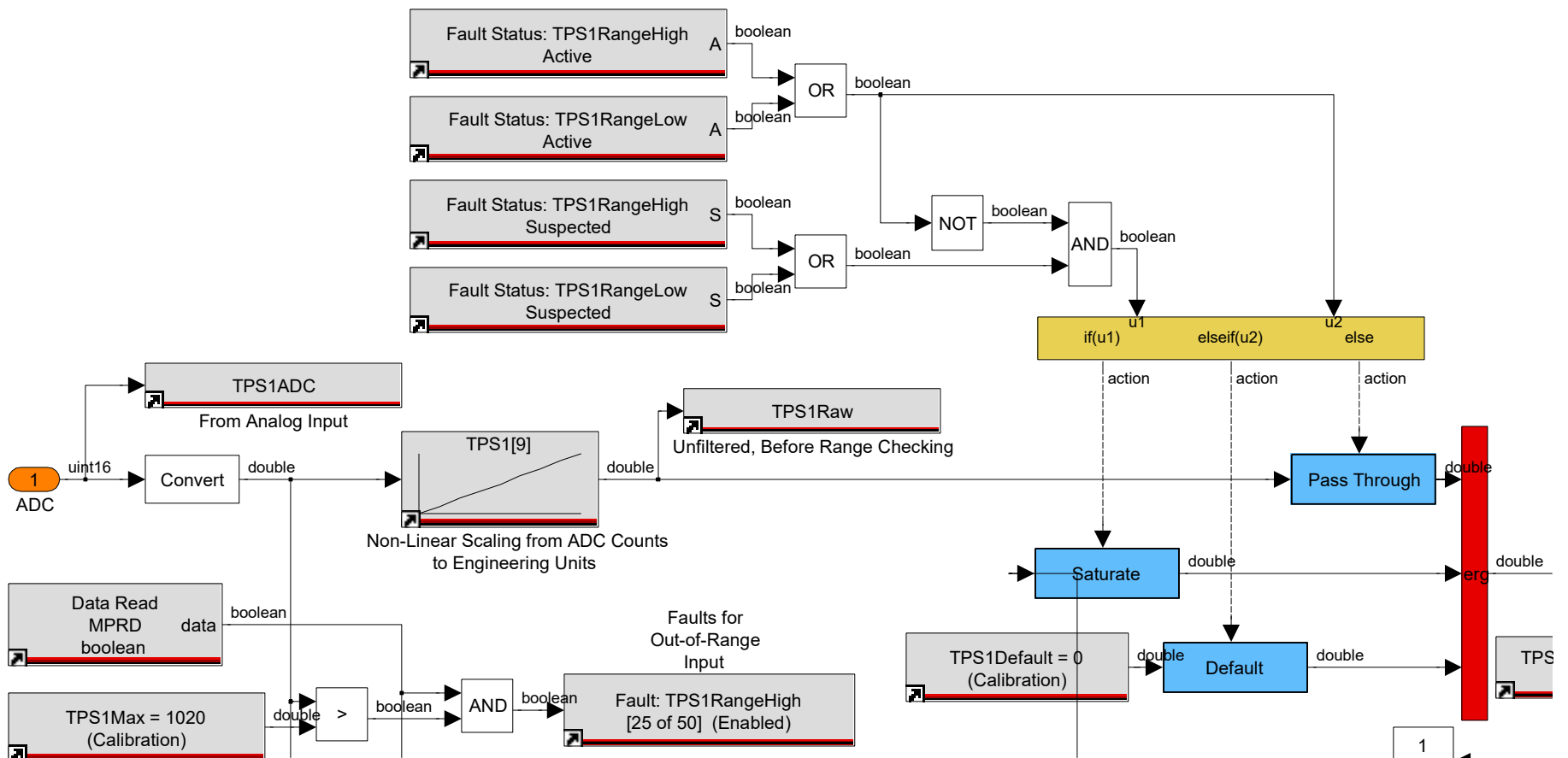




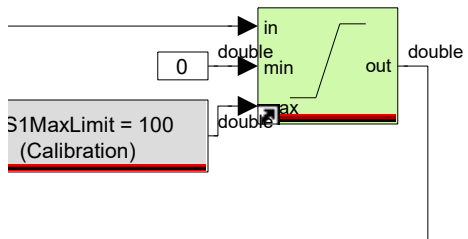
elseif { }  
Action Port

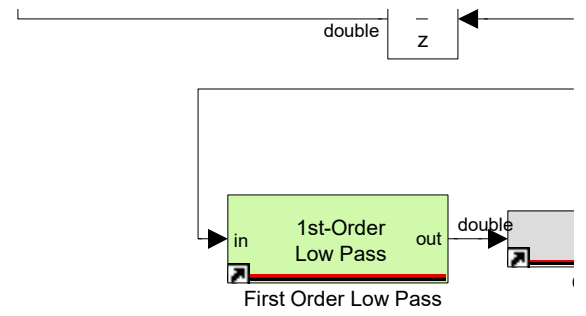
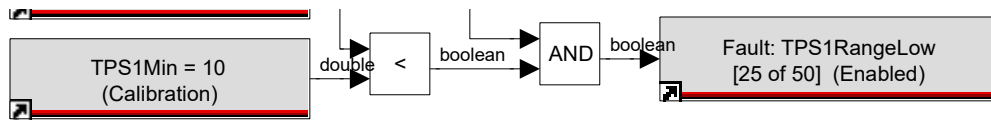


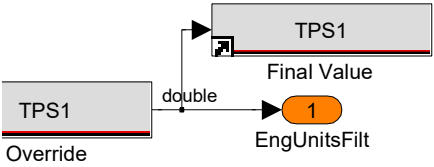


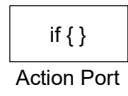






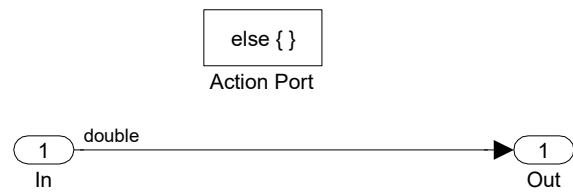


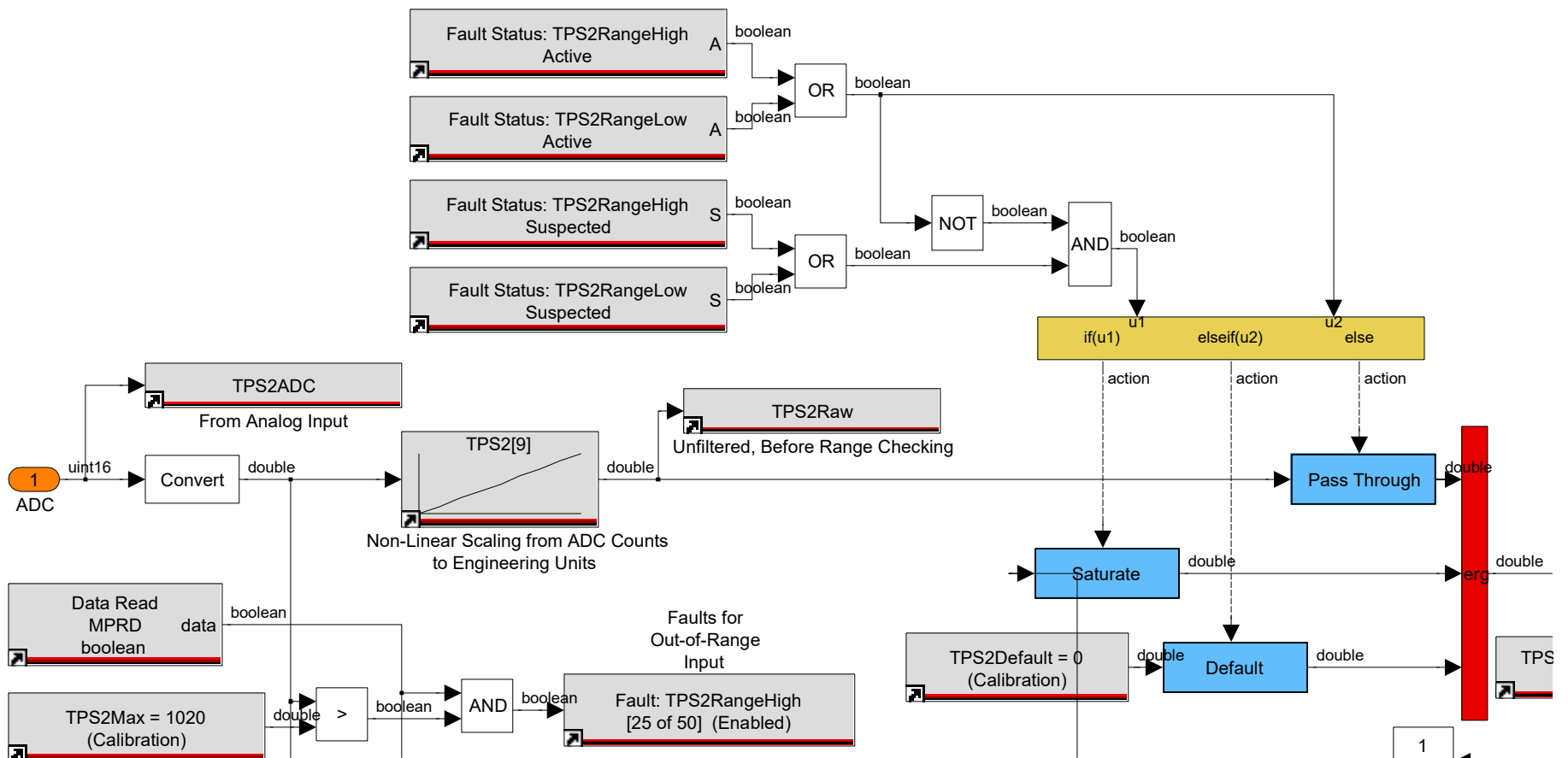


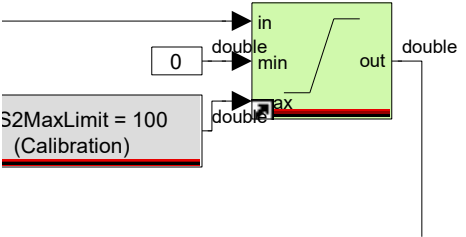


elseif { }  
Action Port

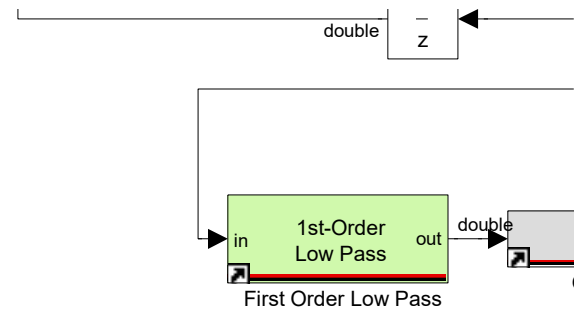
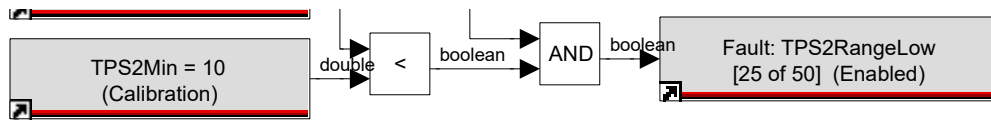


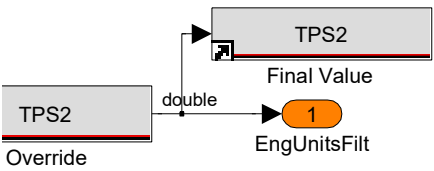


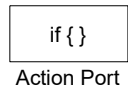






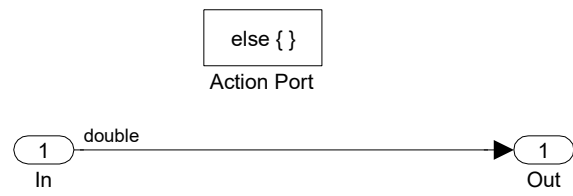


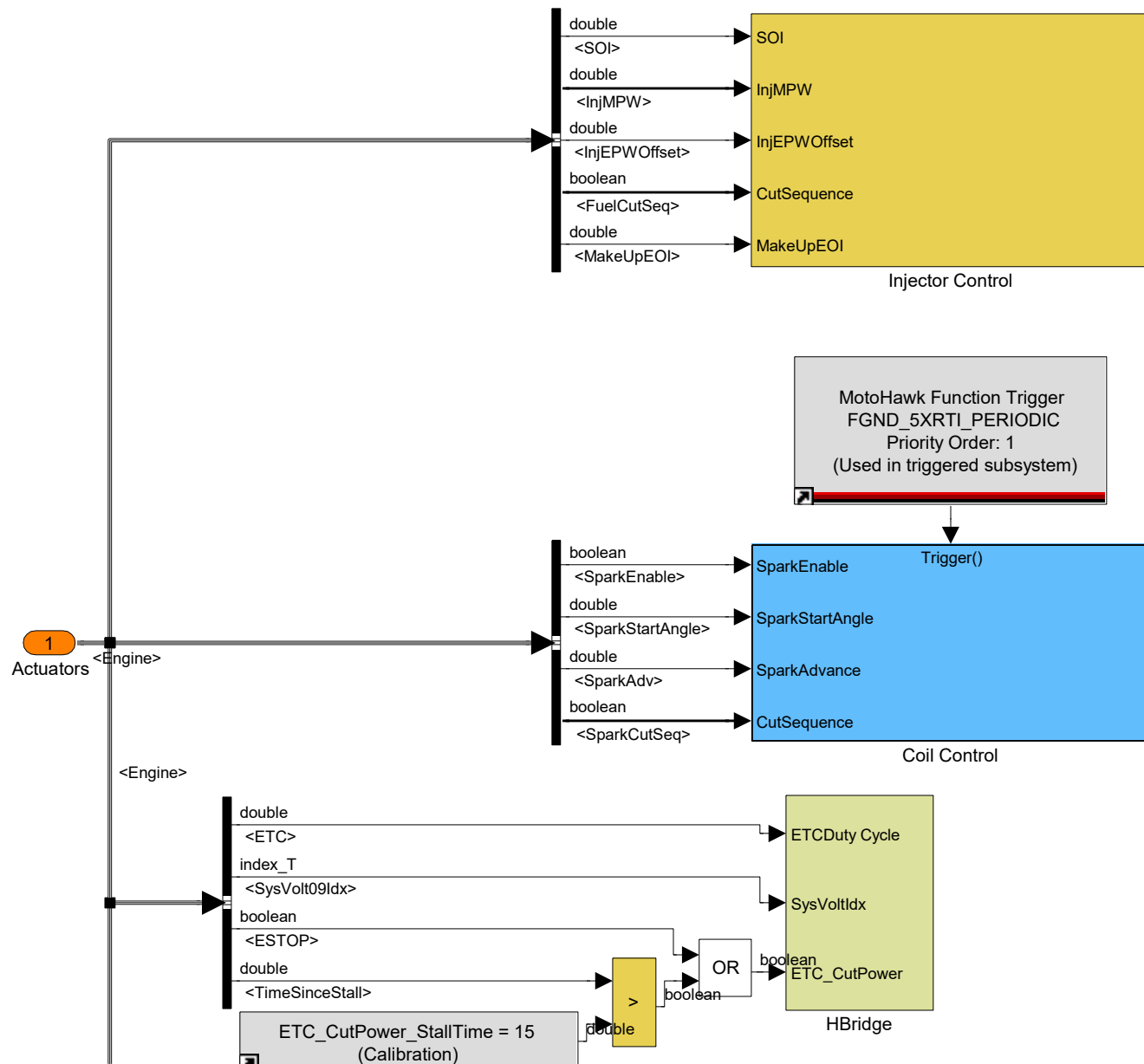




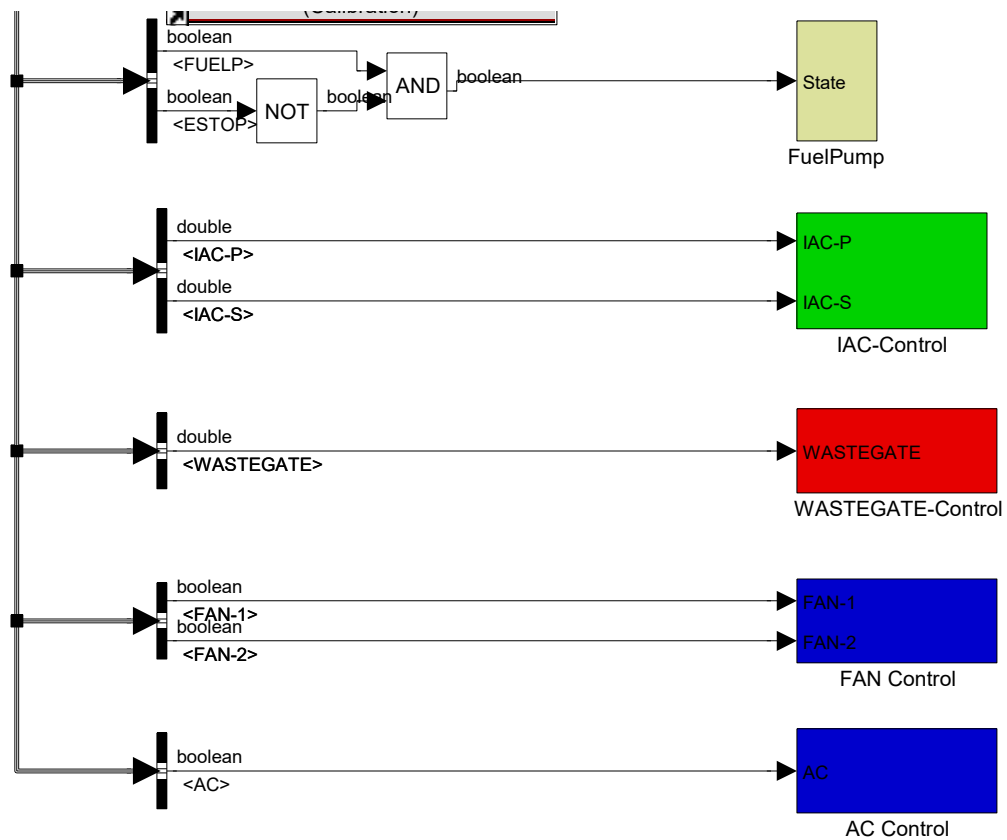
elseif { }  
Action Port













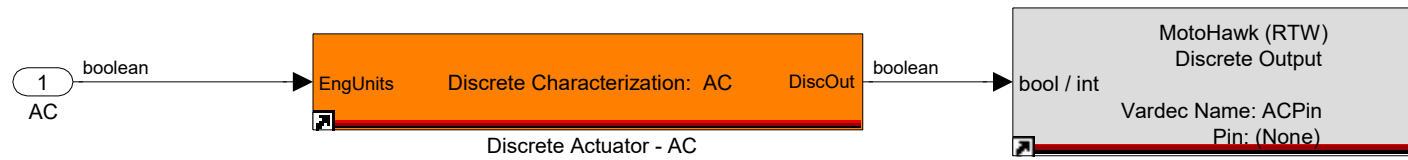




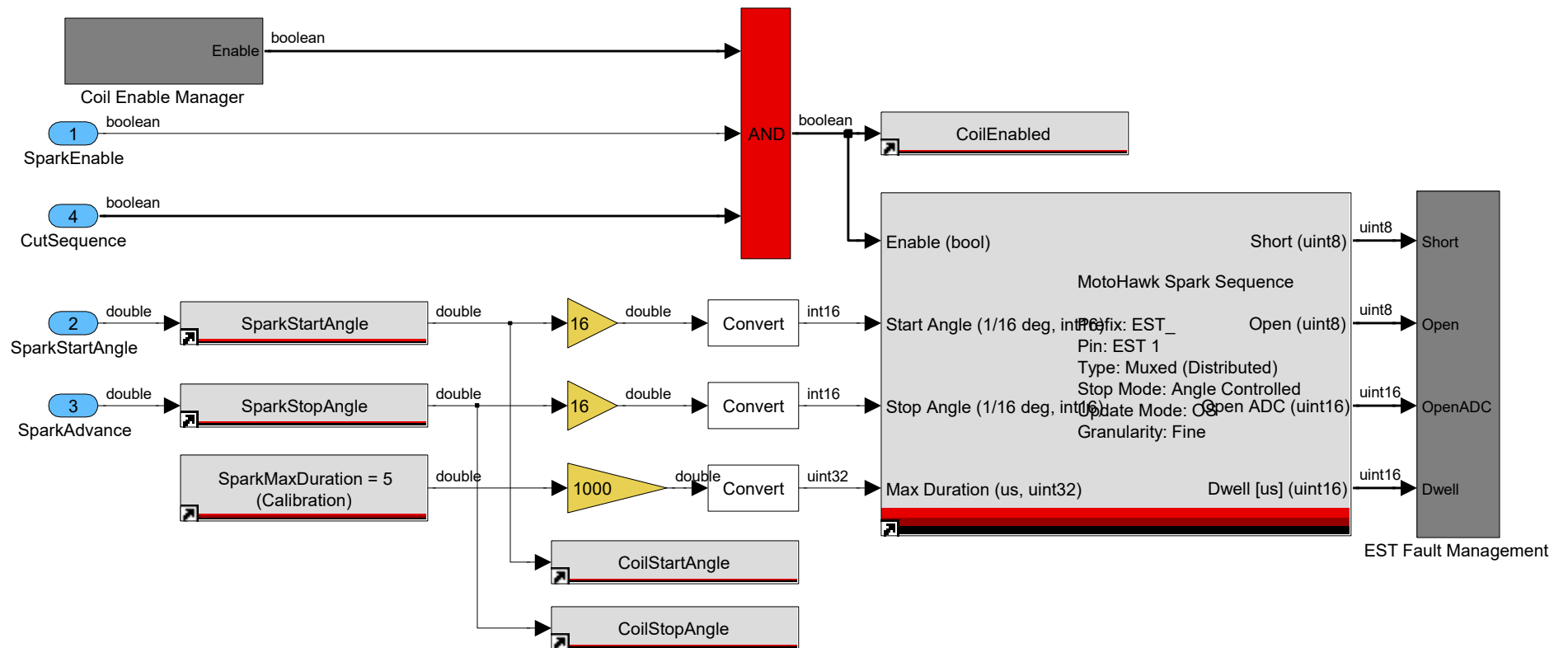


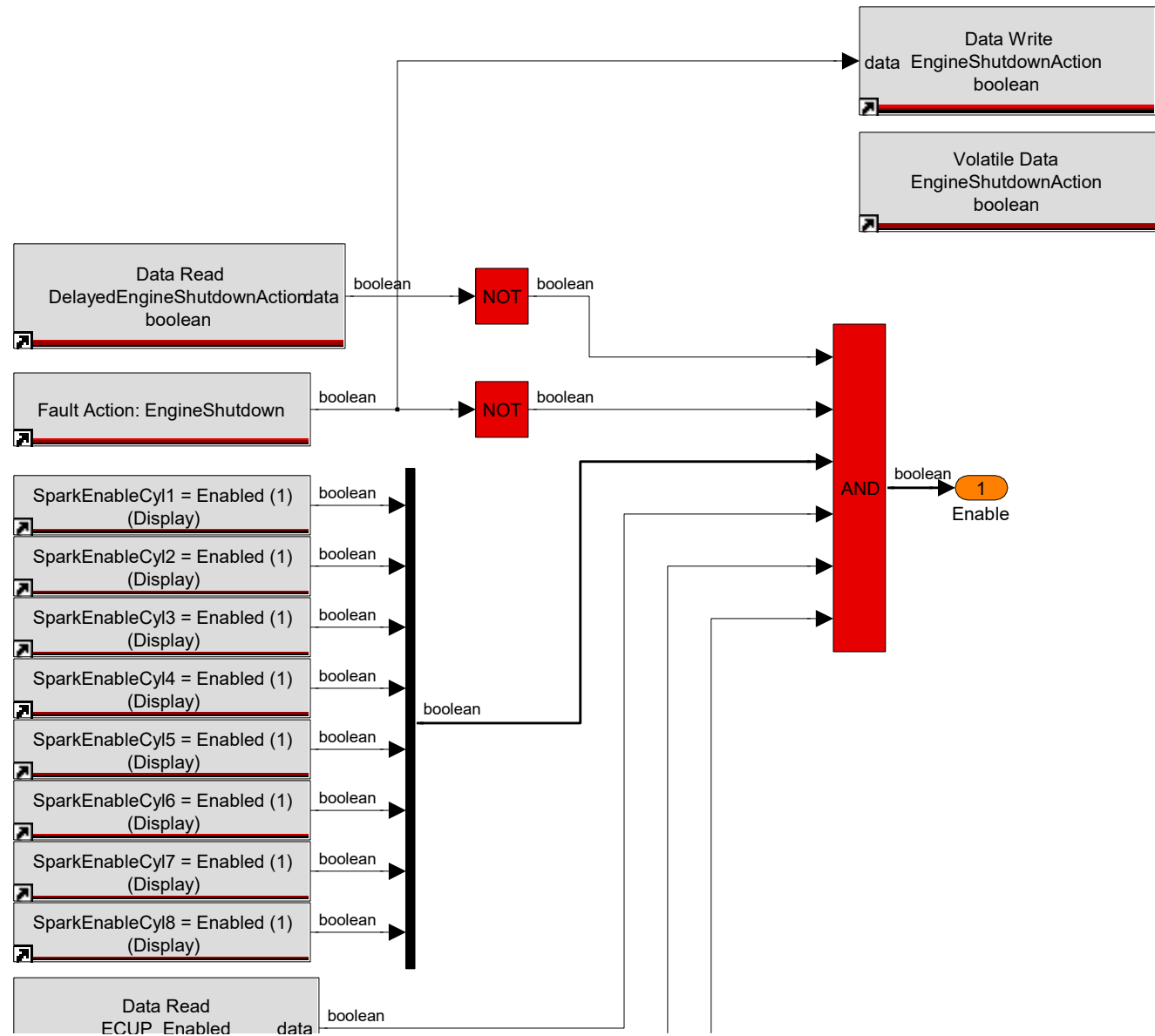
--



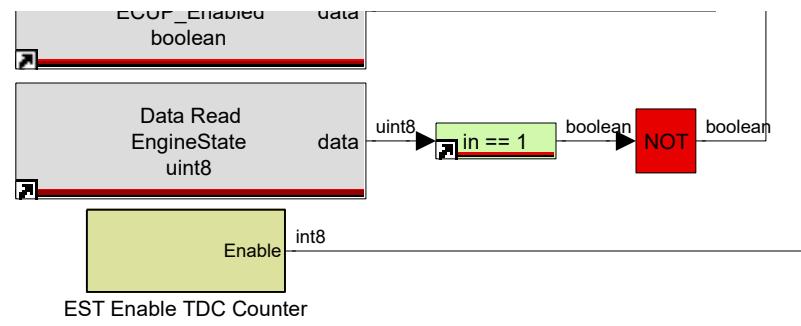


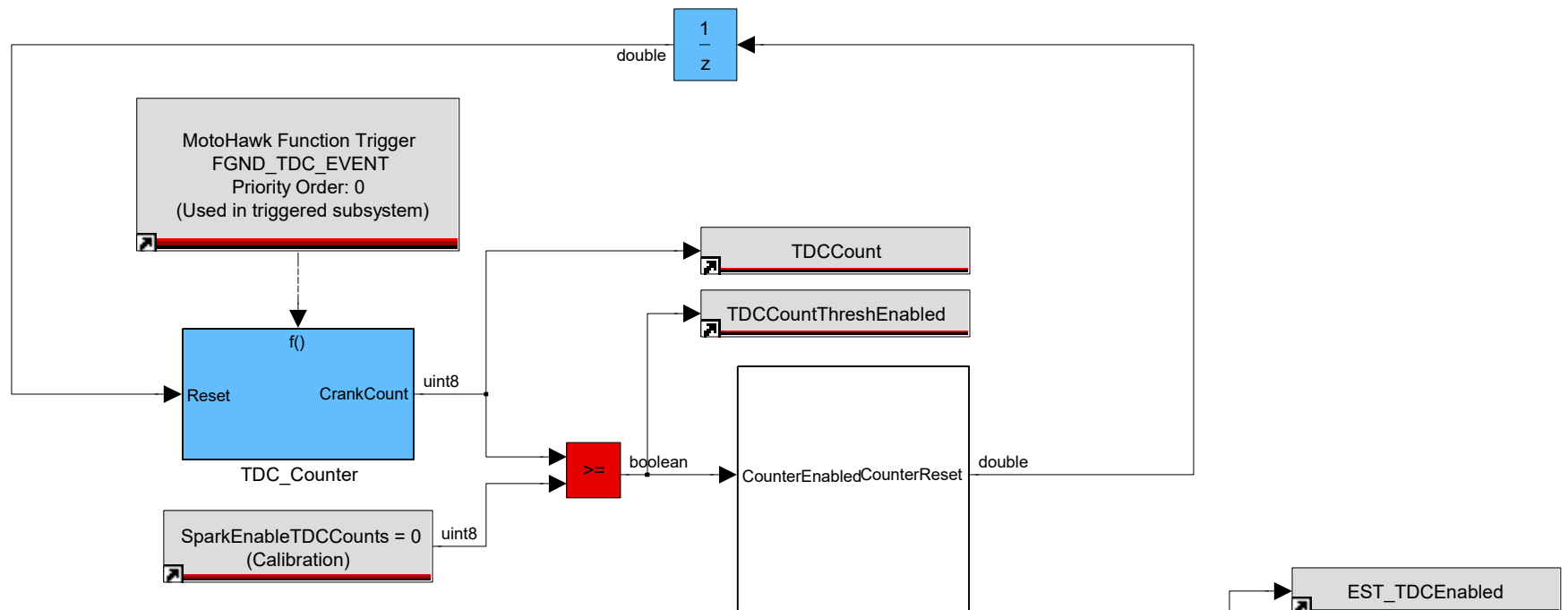
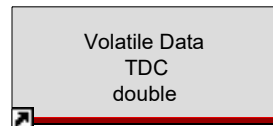
f()

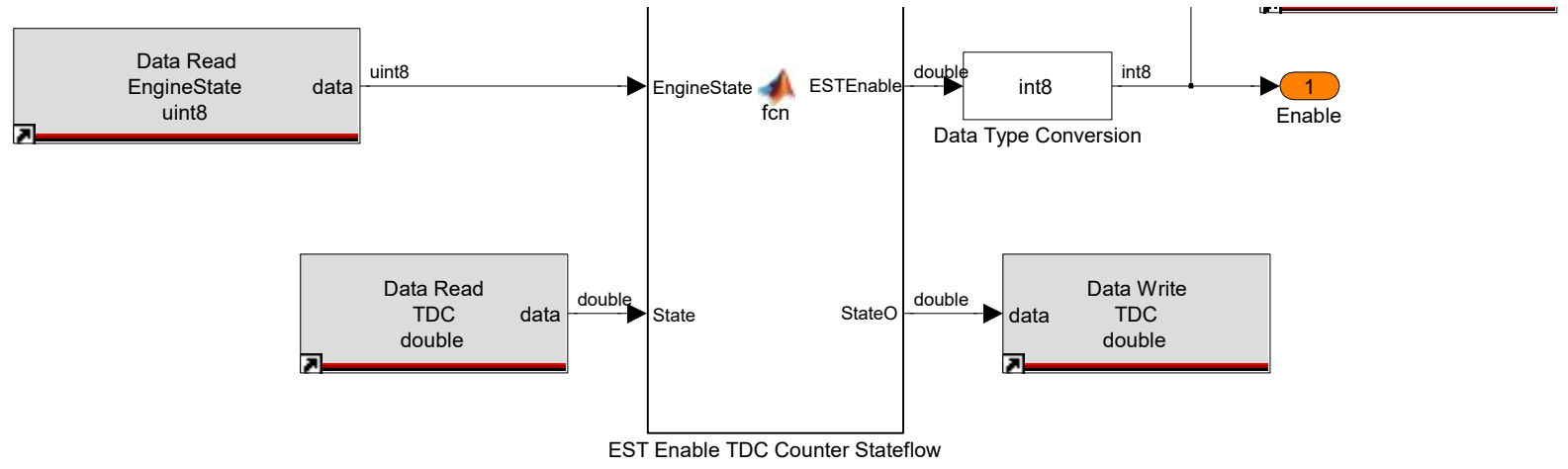




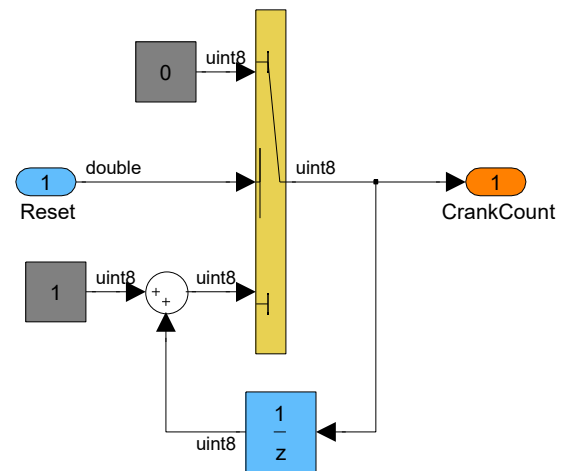


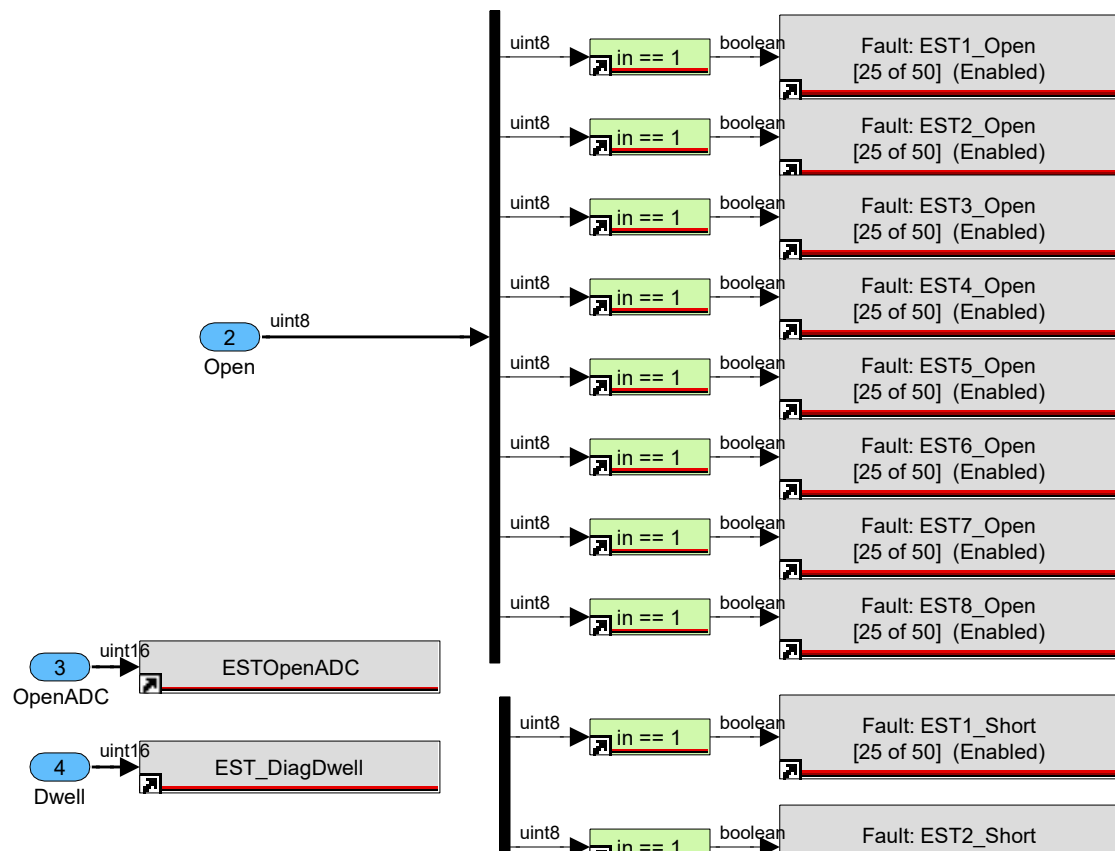


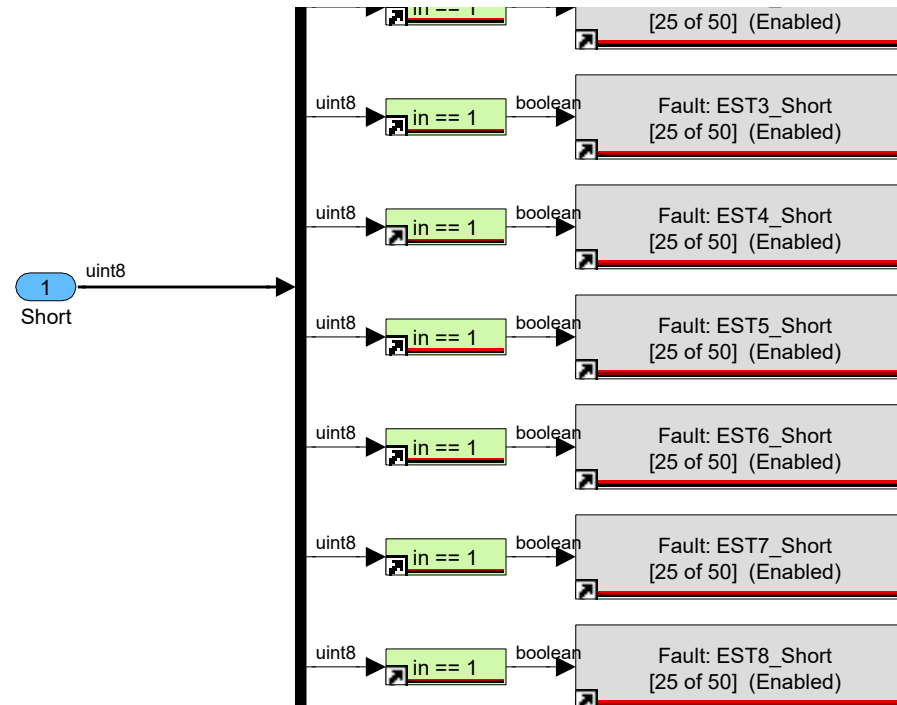


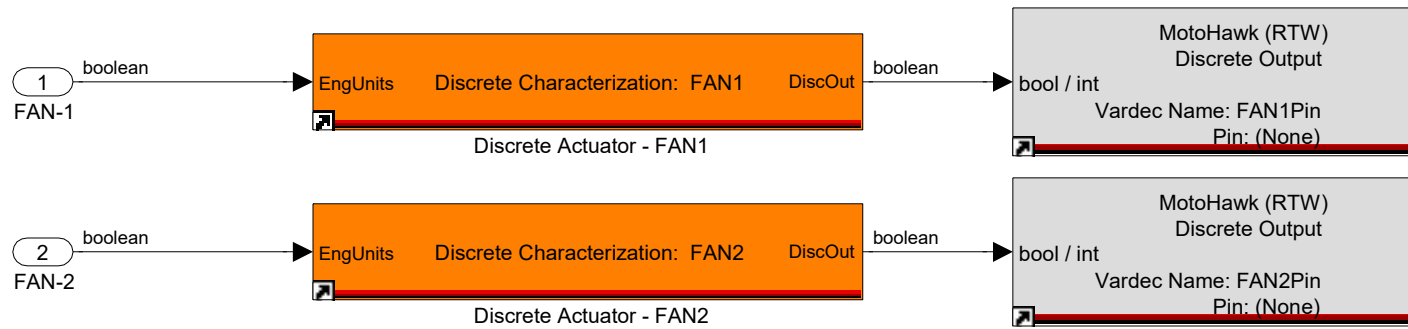


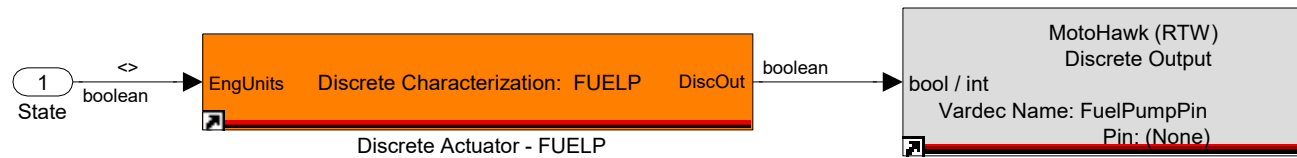
f()  
f



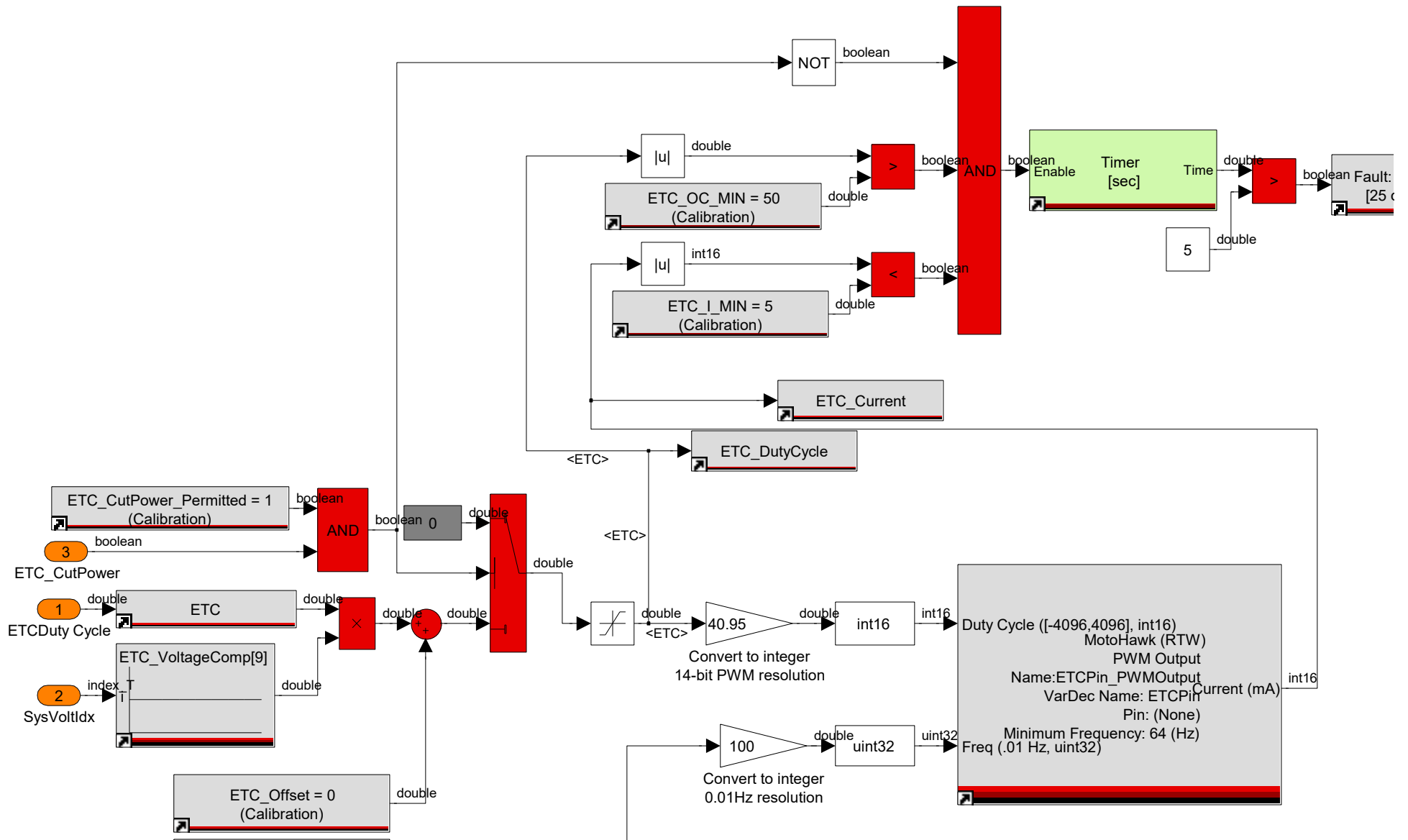










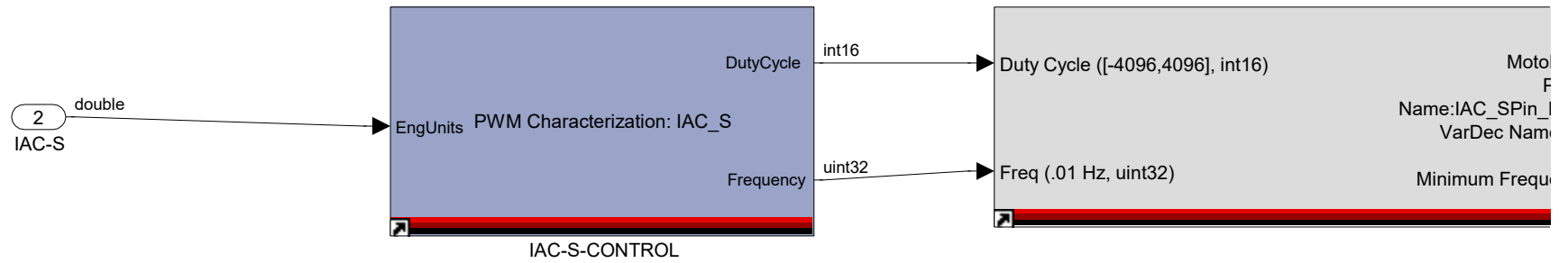
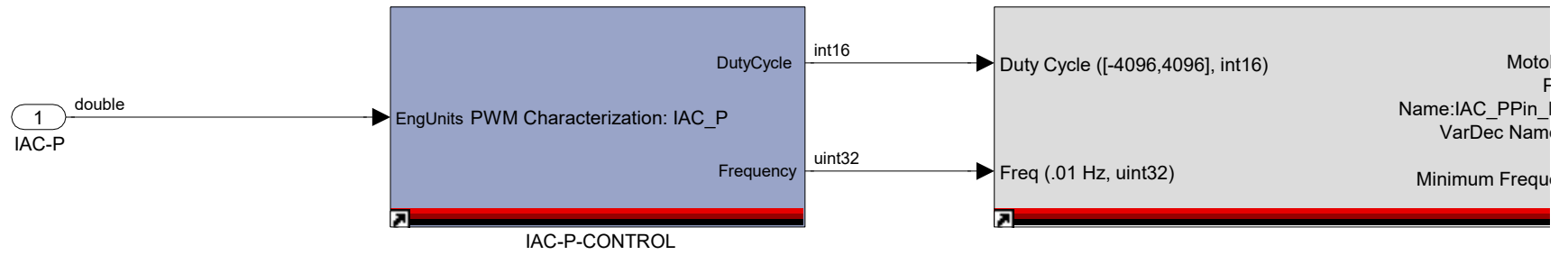


: ETC\_Open\_Fault  
of 50] (Enabled)

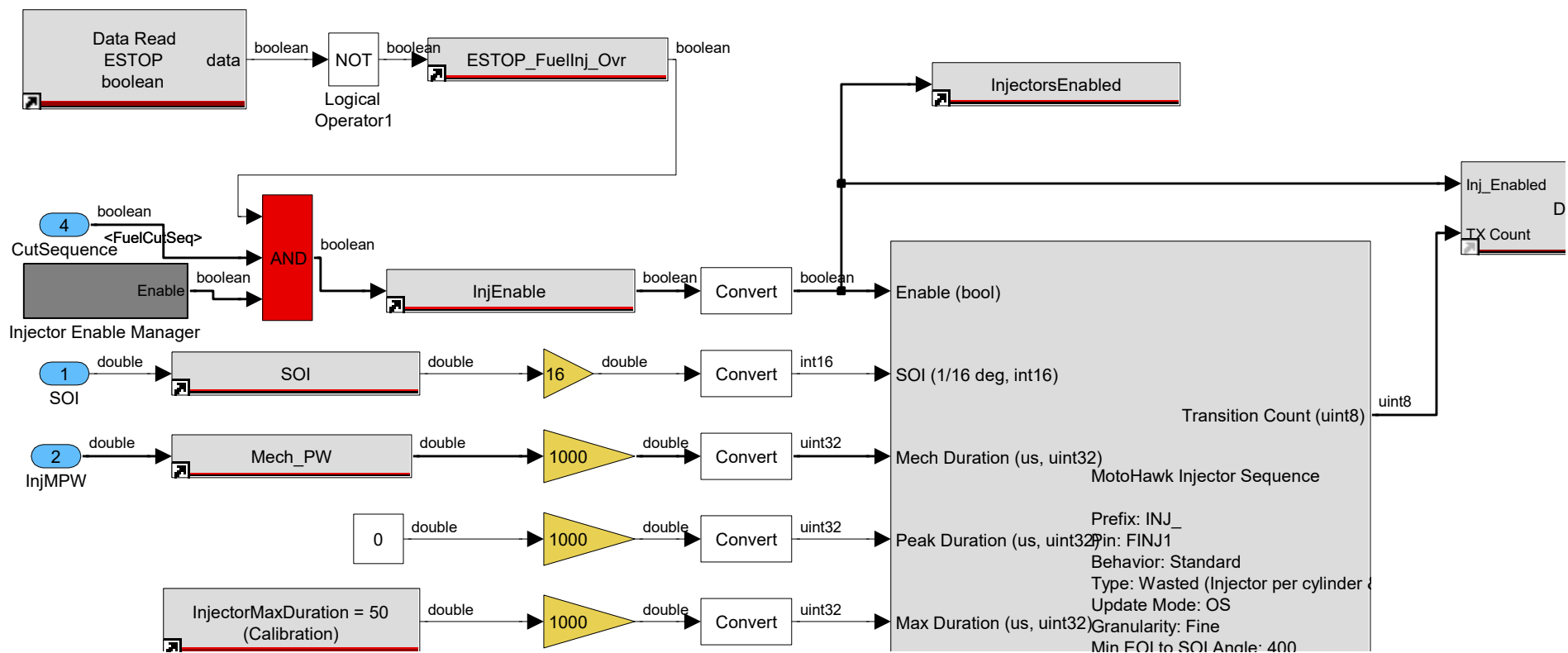
ETC\_Frequency = 2500  
(Calibration)

double



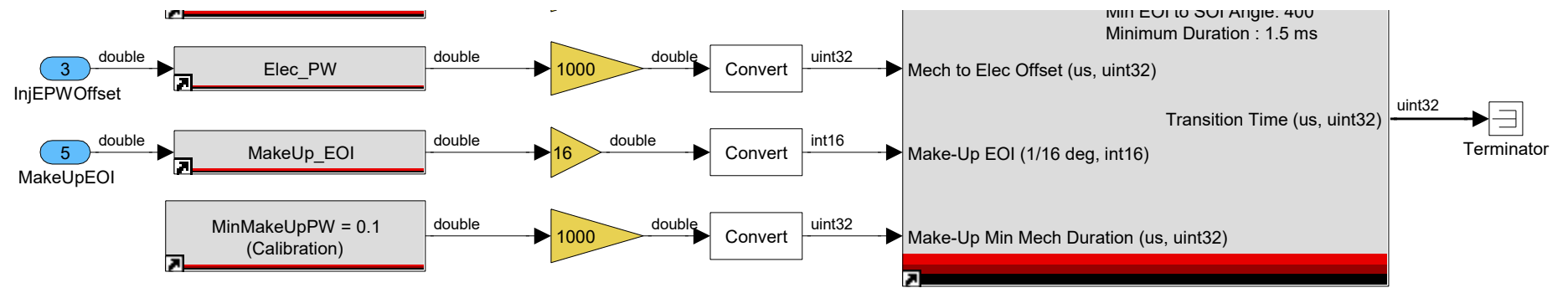




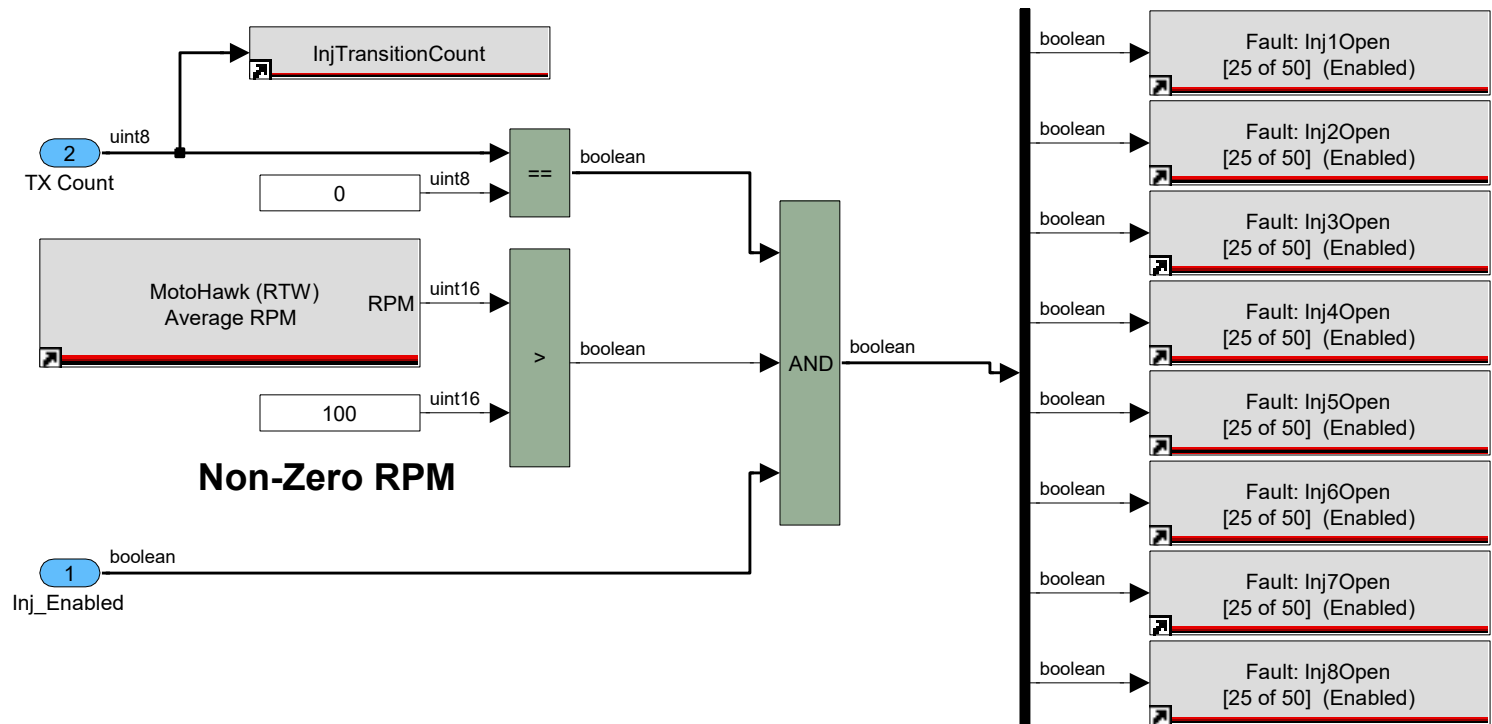


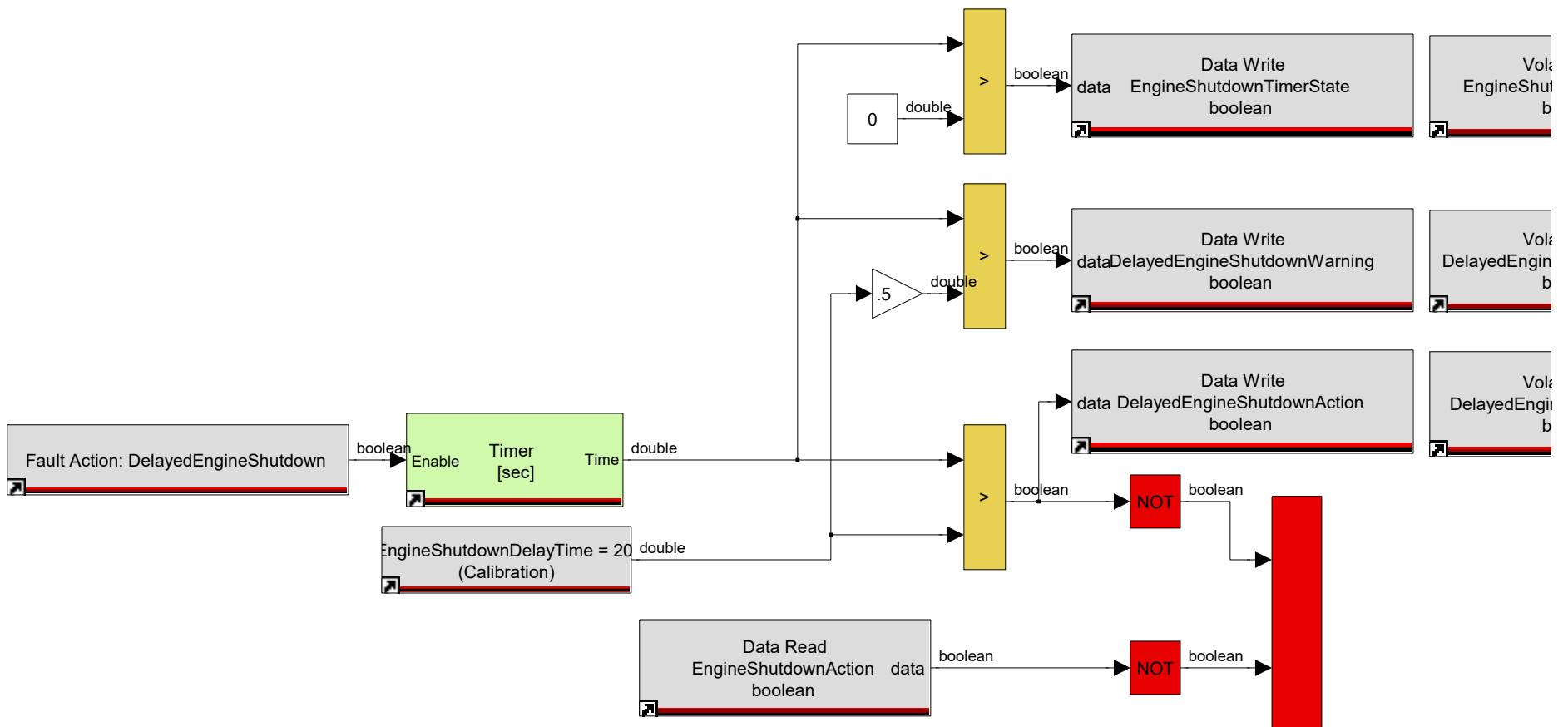
Injector  
Diagnostics  
(Inj)







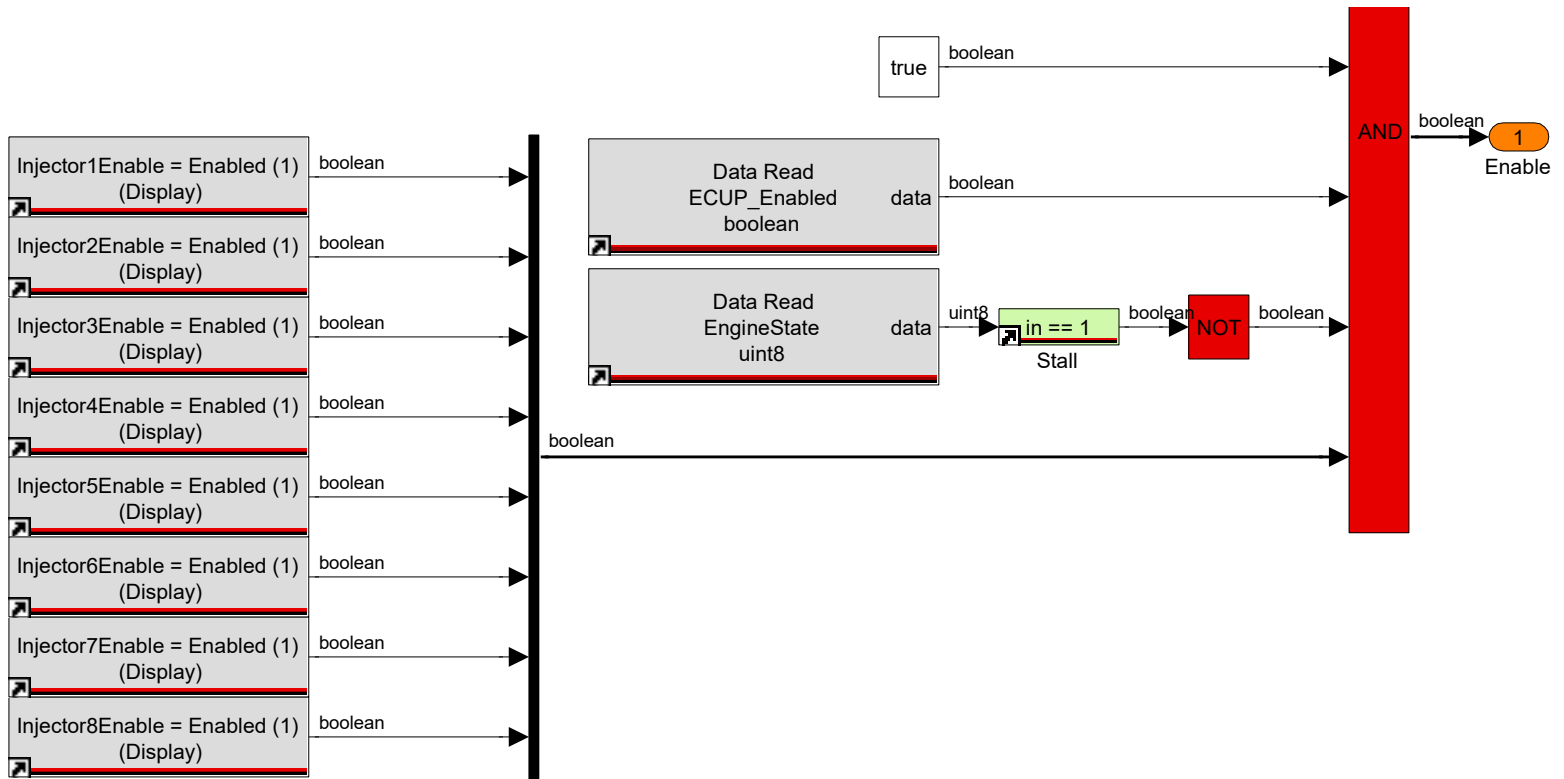




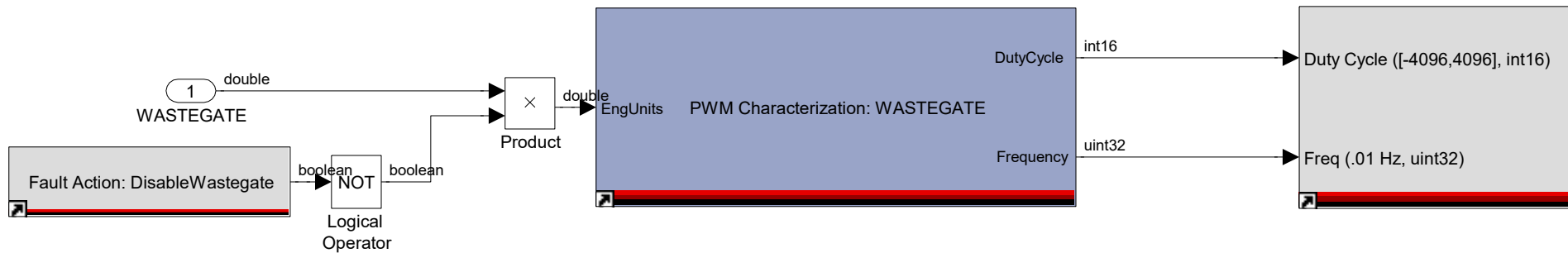
olatile Data  
utdownTimerState  
oolean

olatile Data  
neShutdownWarning  
oolean

olatile Data  
ineShutdownAction  
oolean









MotoHawk (RTW)

PWM Output

Name:WASTEGATE\_PWMOutput

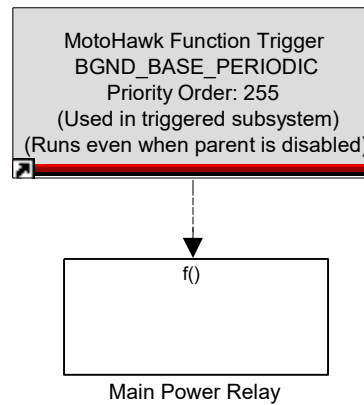
VarDec Name: WASTEGATE

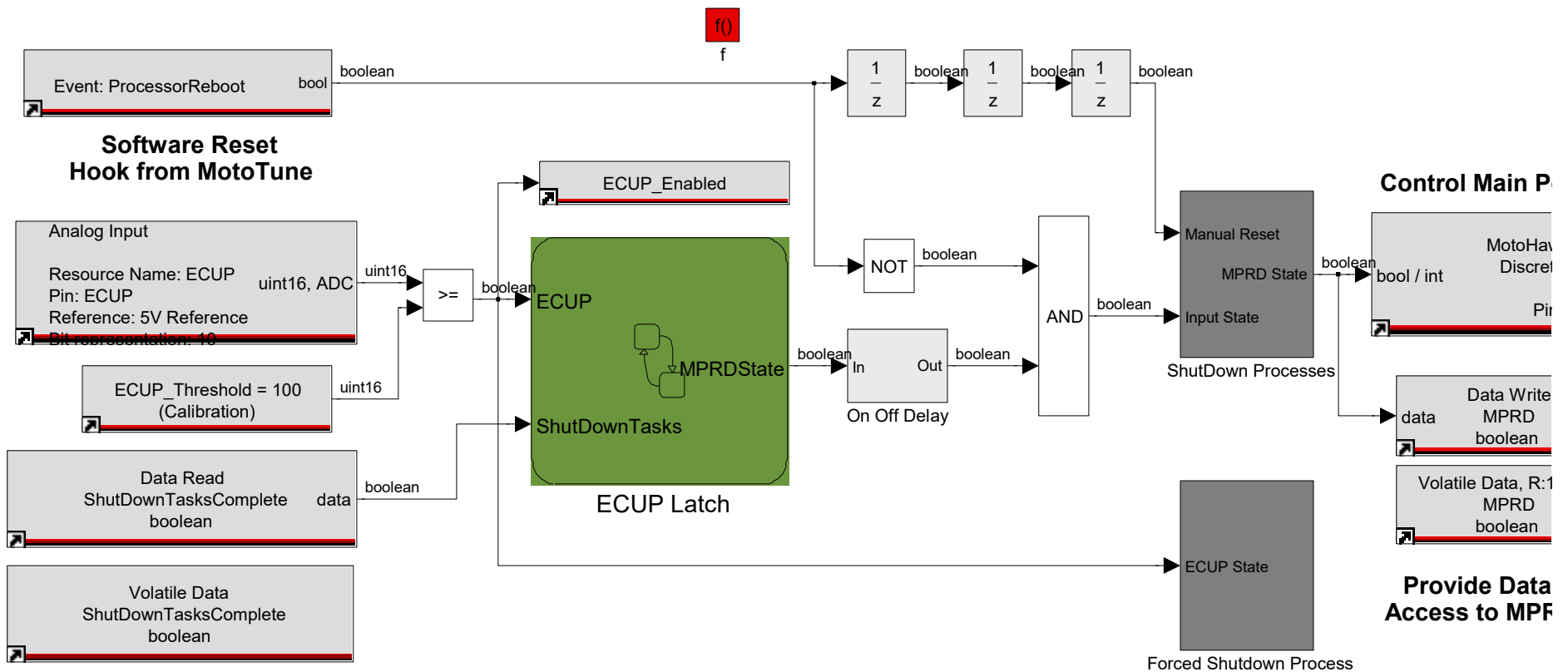
Pin: (None)

Minimum Frequency: 2 (Hz)

## Always execute Main Power Relay control in the background.

The saving and restoring of non-volatile variables must be called from a background priority task.





Power Relay

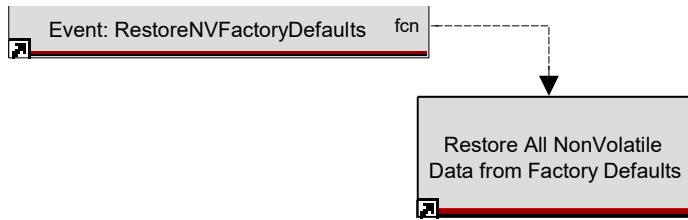
Work (RTW)

State Output

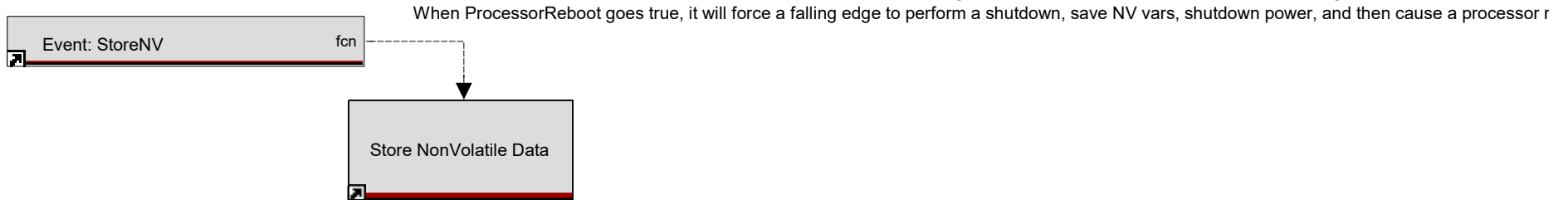
in: MPRD

1, W:0

Store  
RD State

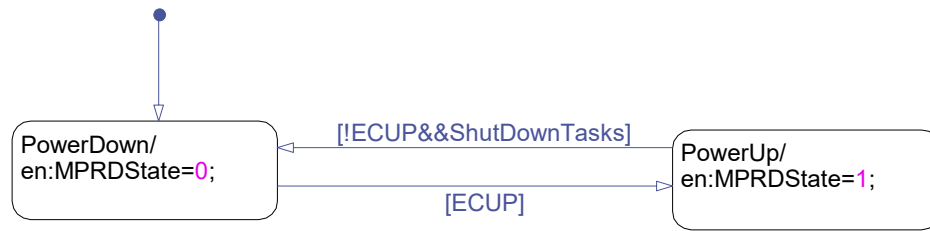


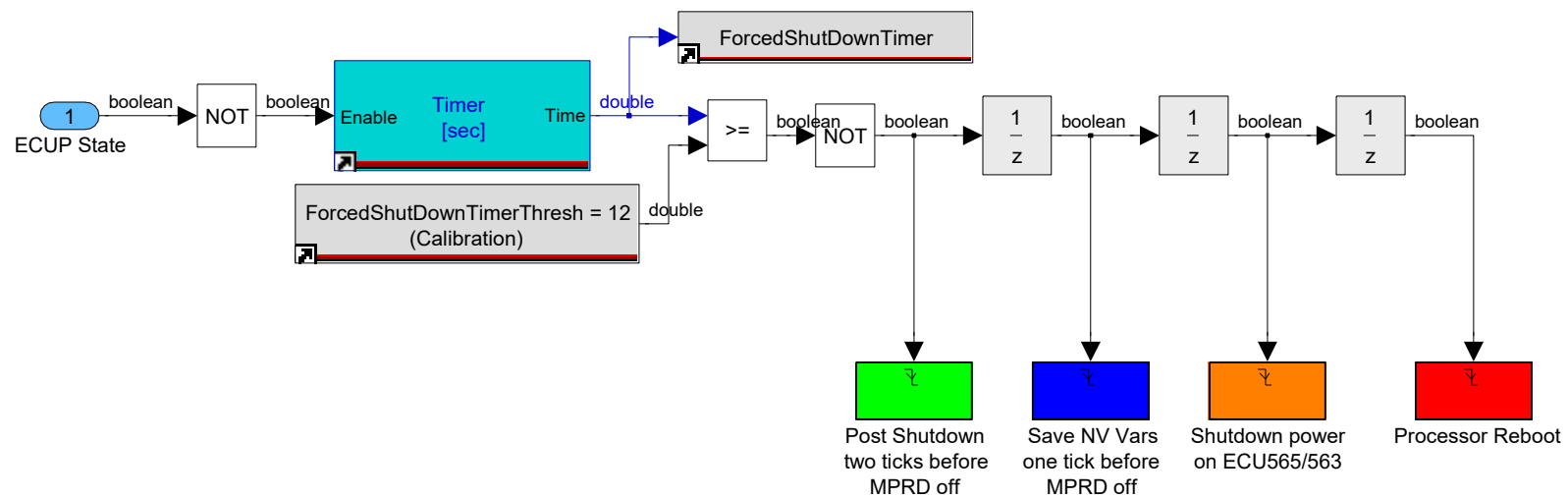
**After a delay, turn on or off Main Power Relay**



**NonVolatile Memory Store/Restore  
Hooks from MotoTune**

reset if still true

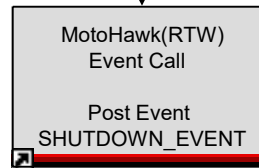
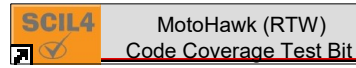








Trigger



Post Shutdown two ticks before MPRD off



true

boolean

clear

Clear Fault: ETCSpringTest



Trigger

Inline Code

Include:  
Start:  
Output: while (1);

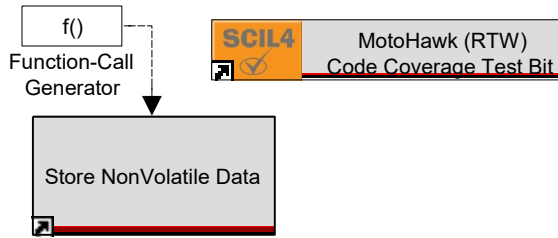
Loop Forever  
Causing Watchdog Reset

SCIL4

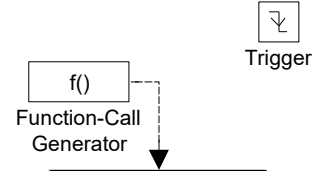
MotoHawk (RTW)  
Code Coverage Test Bit



Trigger



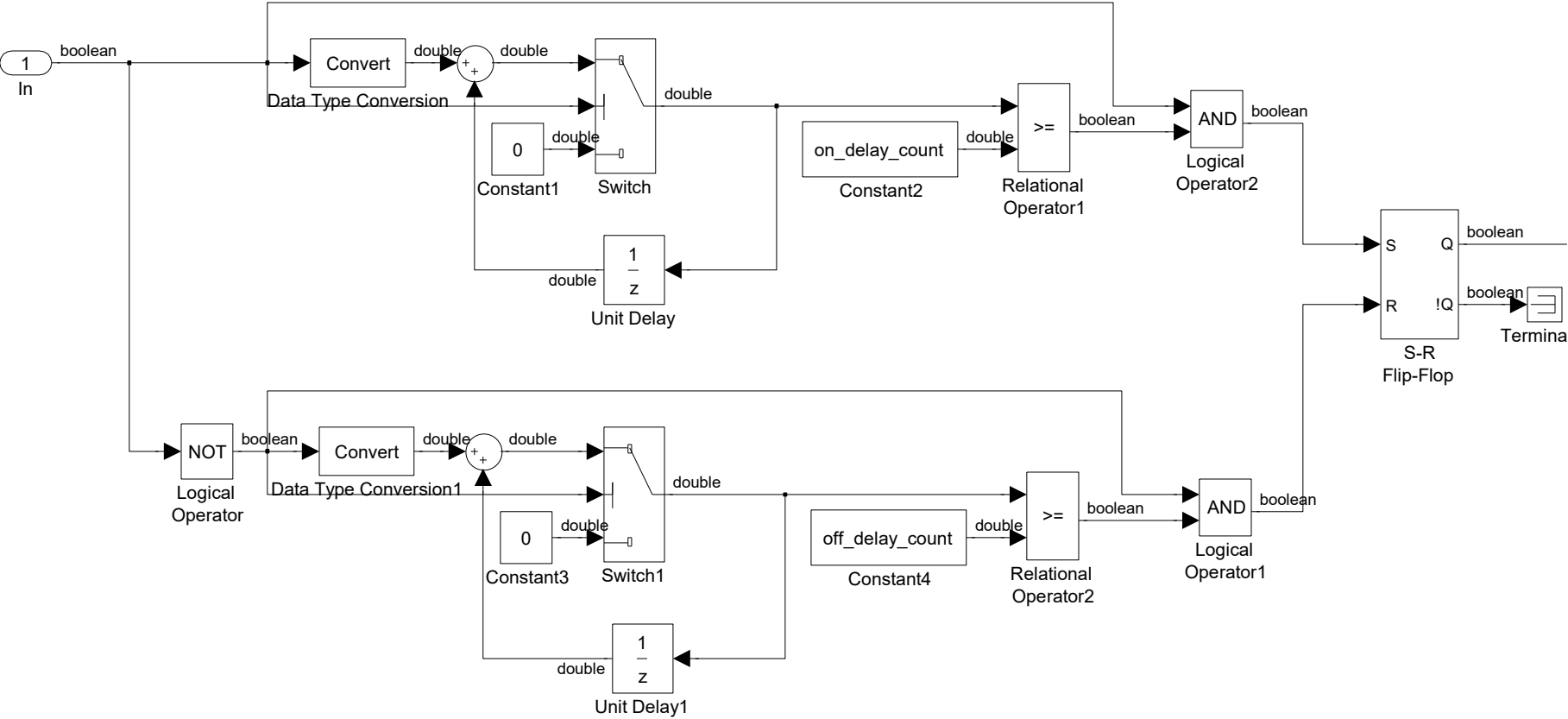
Save NV Vars one tick before MPRD off

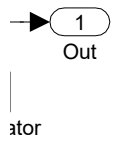


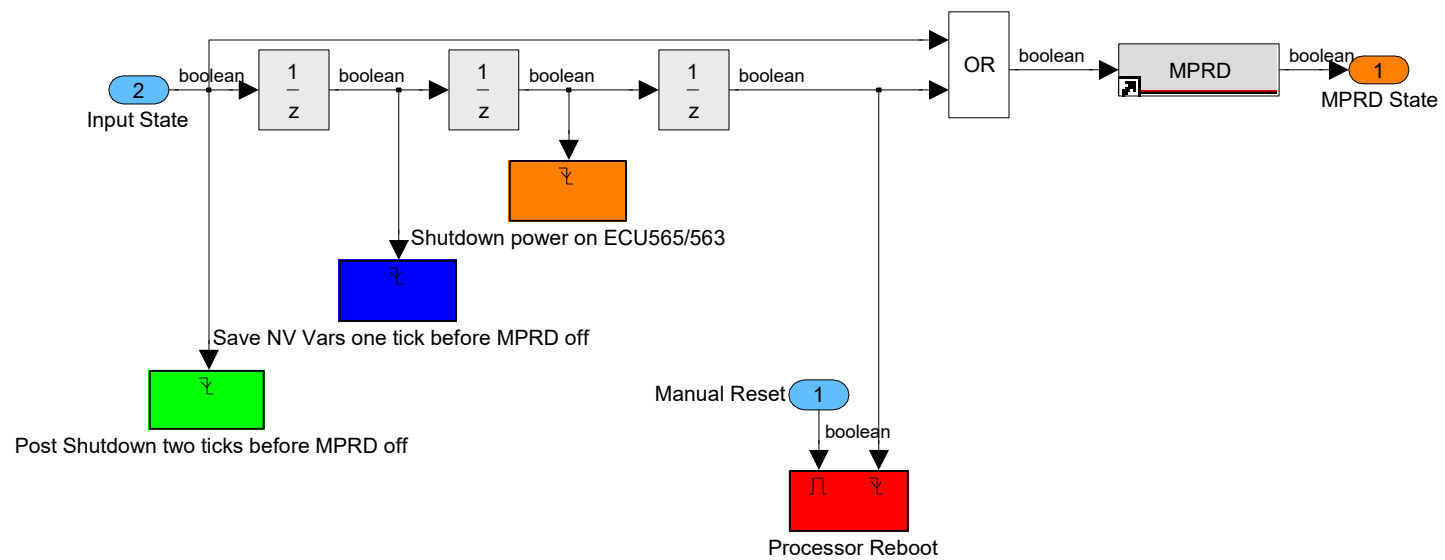
<b>SCIL4</b>	MotoHawk (RTW)
	Code Coverage Test Bit

Shutdown power on ECU565-128

Delay the rising and falling of a boolean signal



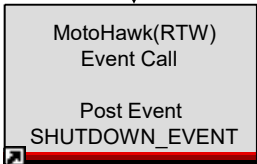






Trigger

f()  
Function-Call  
Generator



Post Shutdown two ticks before MPRD off






Trigger



Enable

Inline Code

Include:  
Start:  
Output: while (1);



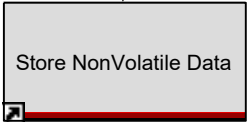
Loop Forever  
Causing Watchdog Reset



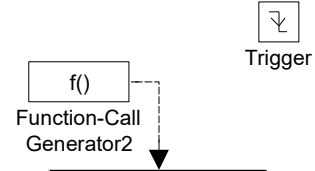
Trigger

f()

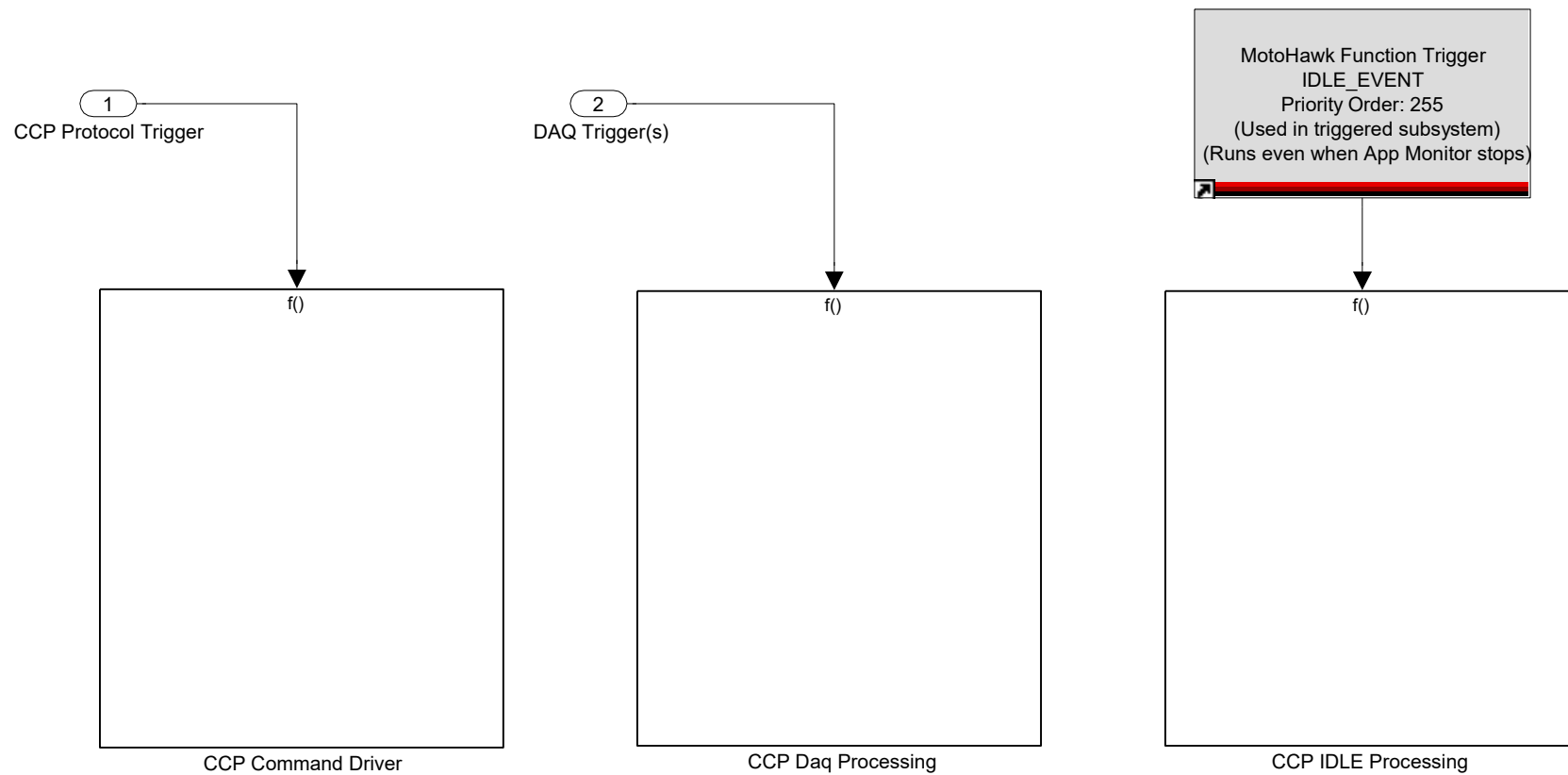
Function-Call  
Generator1



Save NV Vars one tick before MPRD off



Shutdown power on ECU565-128



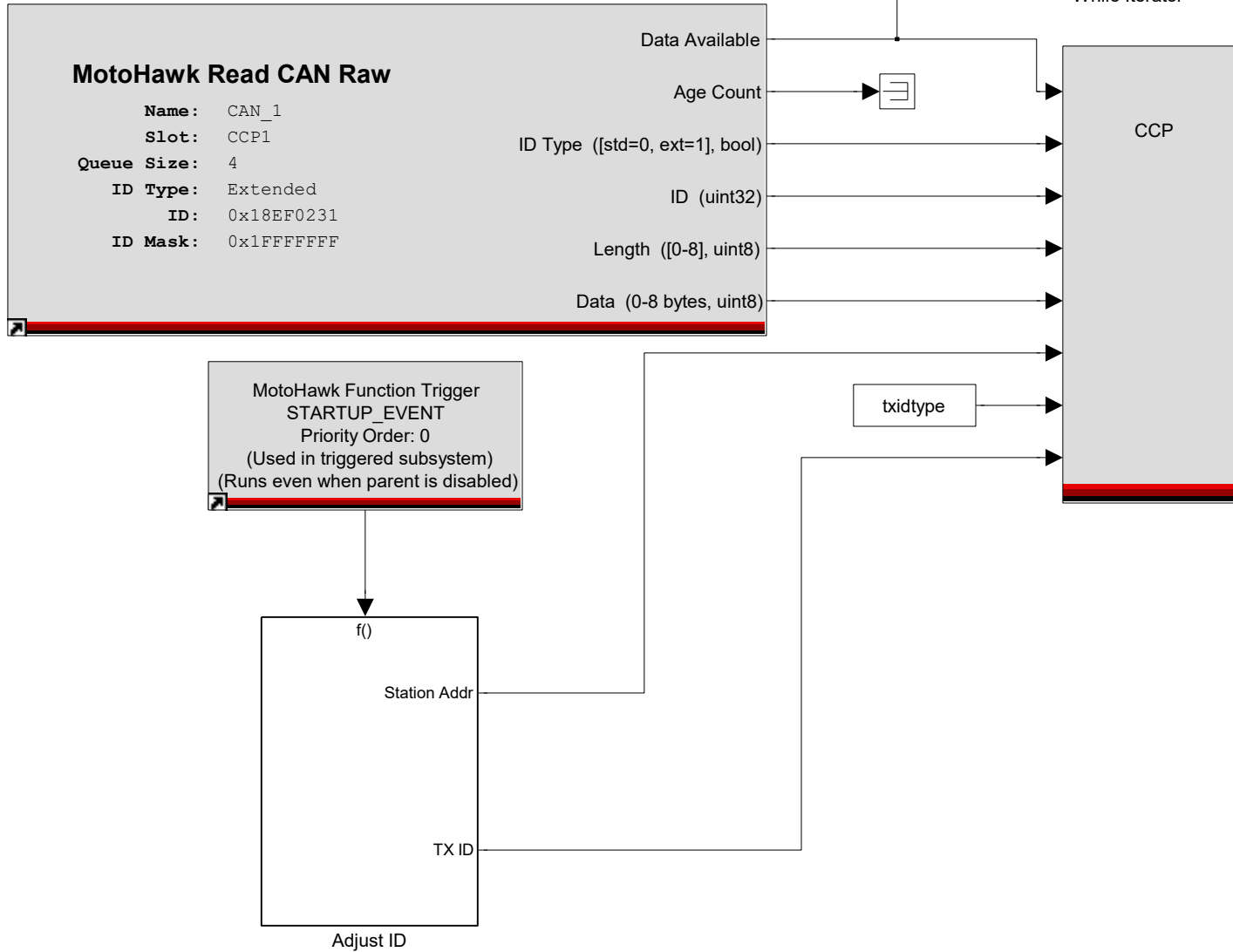
f()  
f

do { ... } while

While new CCP Command

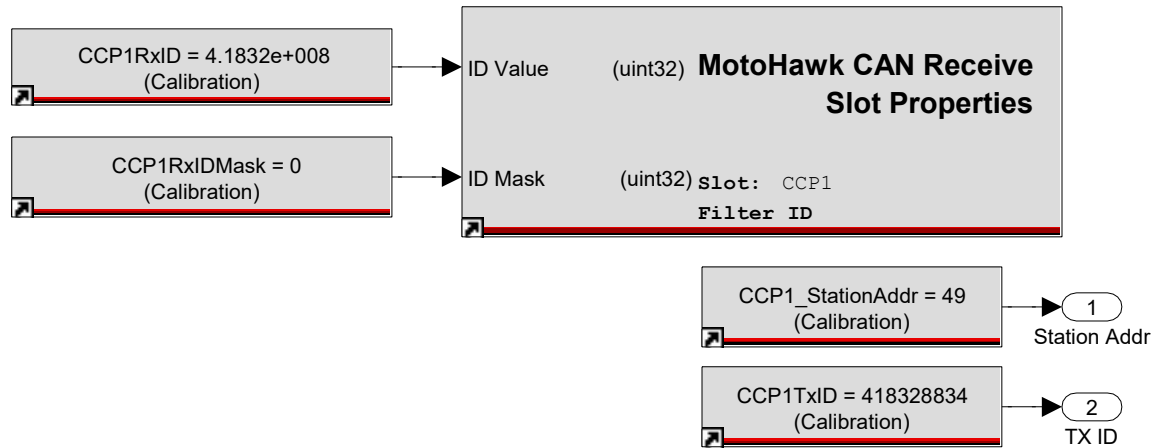
SCIL4

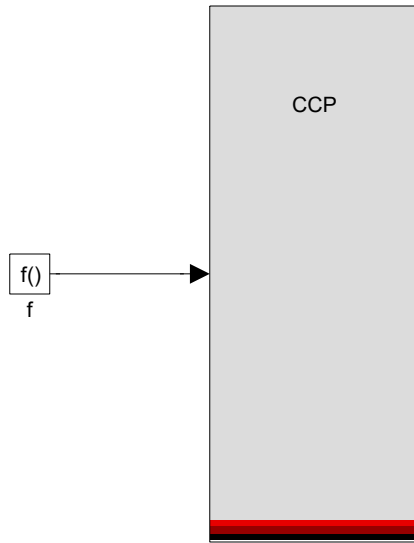
MotoHawk (RTW)  
Code Coverage Test Bit



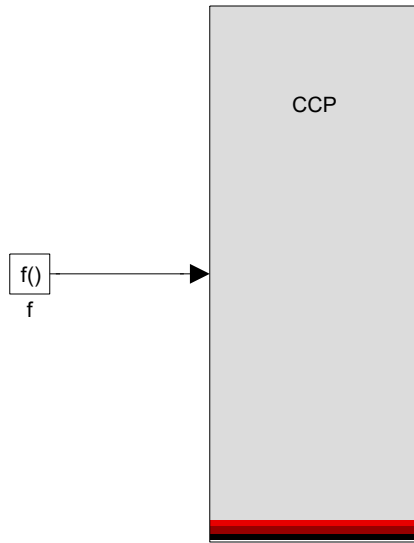
f()  
f

**SCIL4** MotoHawk (RTW)  
Code Coverage Test Bit

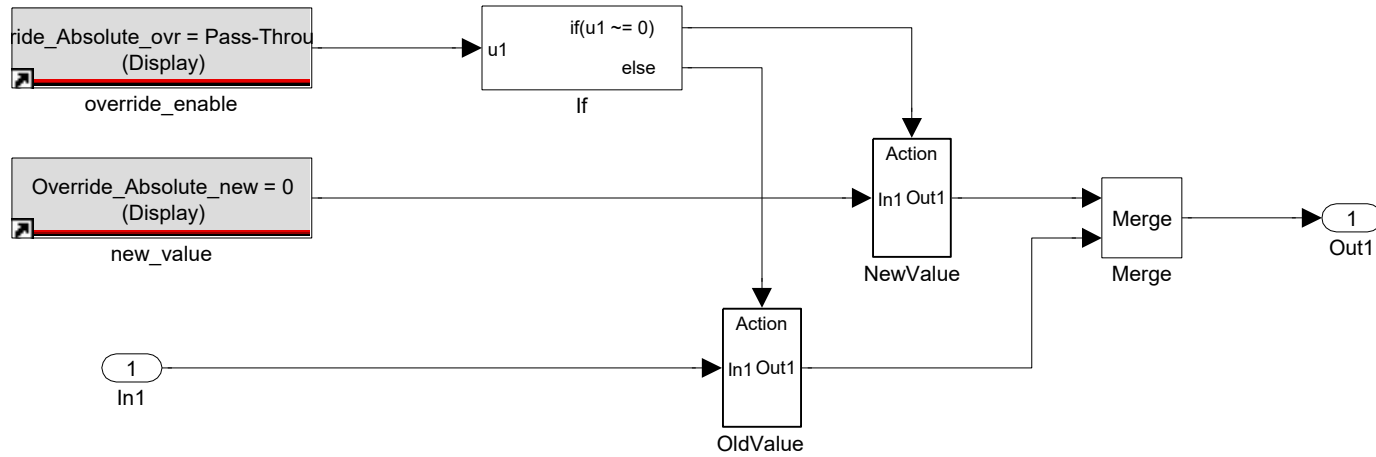








## MotoHawk Absolute Override



Action

Action Port

SCIL4

MotoHawk (RTW)  
Code Coverage Test Bit


1  
In1



1  
Out1

Action  
Action Port

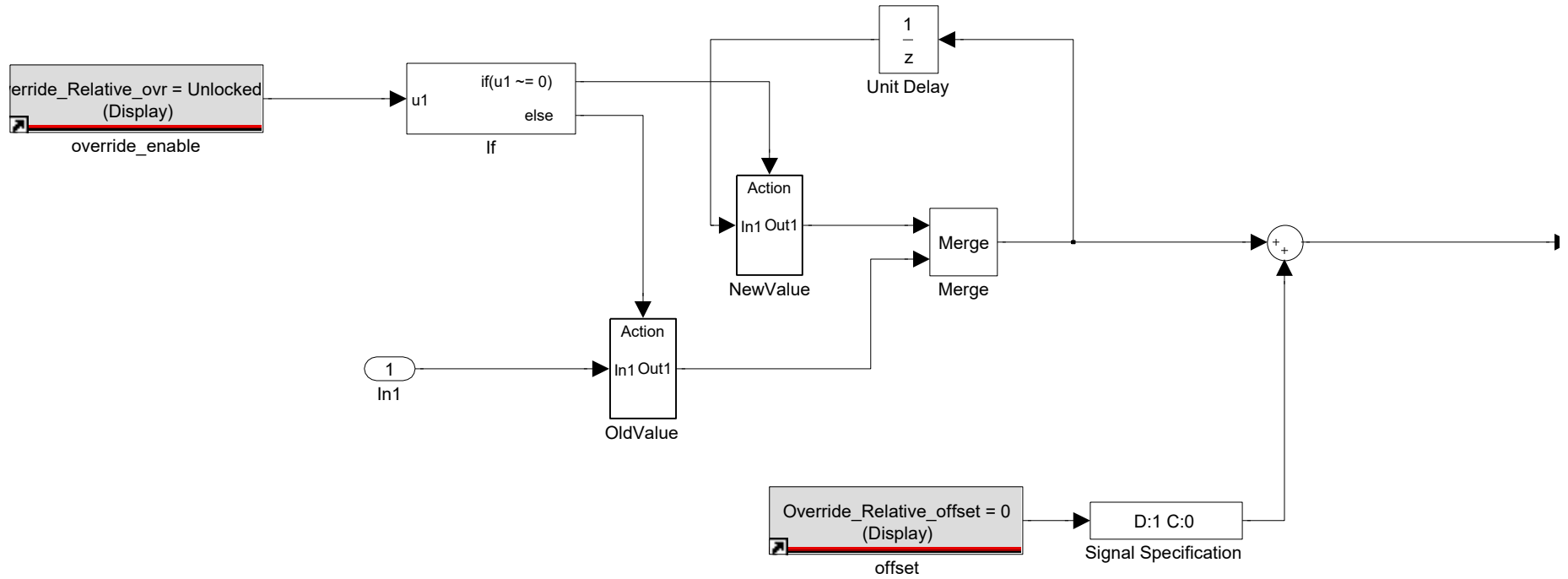
SCIL4



MotoHawk (RTW)  
Code Coverage Test Bit



## MotoHawk Relative Override








Action  
Action Port

SCIL4



MotoHawk (RTW)  
Code Coverage Test Bit






f()

Trigger

motohawk\_sfund\_restore\_nvmem

SCIL4



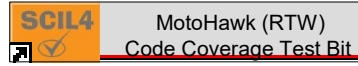
MotoHawk (RTW)  
Code Coverage Test Bit

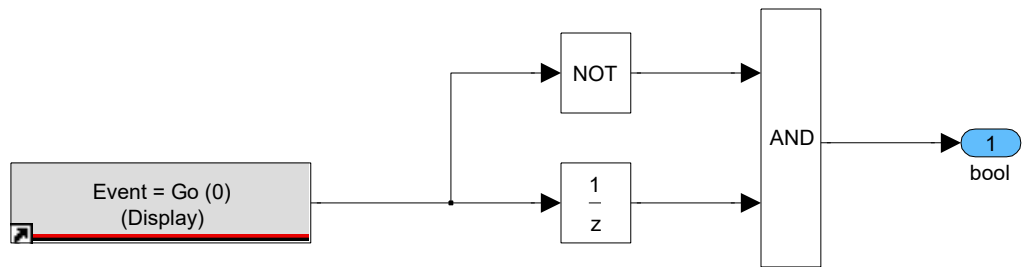


f()

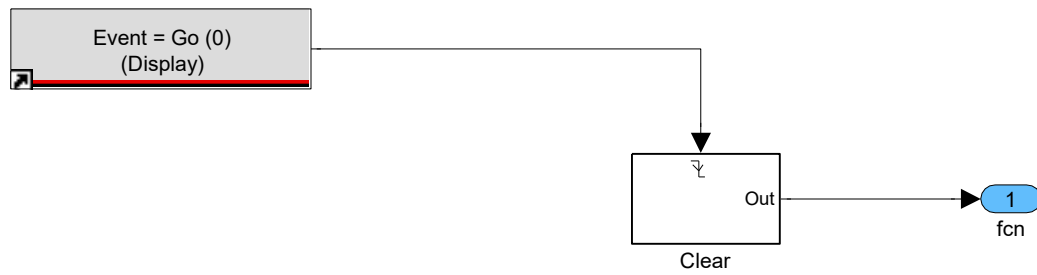
Trigger

motohawk\_sfun\_store\_nvmem

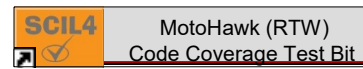


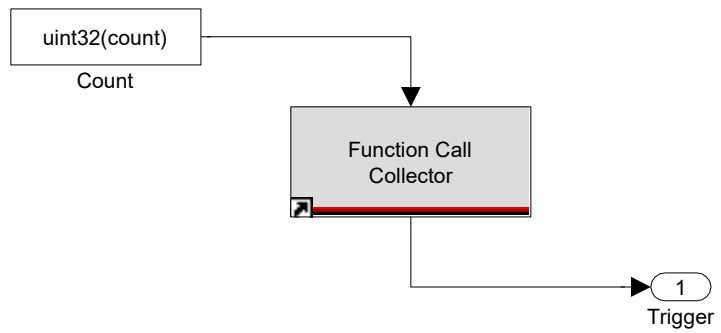


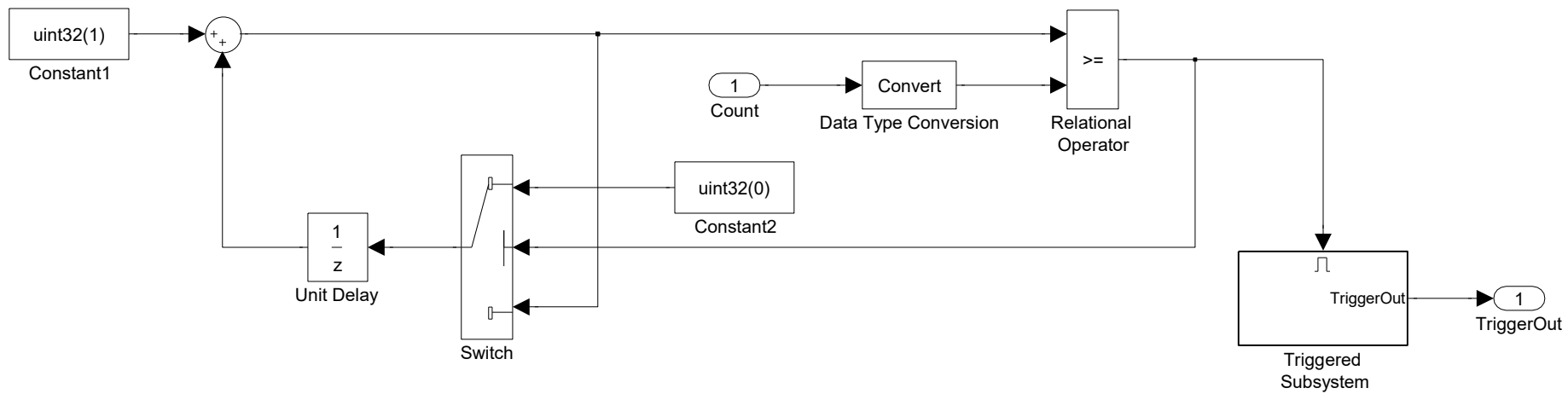
**Output true once on falling edge of event display variable**



**Output function-call once on falling edge of event display variable**













Enable

SCIL4

MotoHawk (RTW)  
Code Coverage Test Bit

f()

Function-Call  
Generator

1

TriggerOut

f()

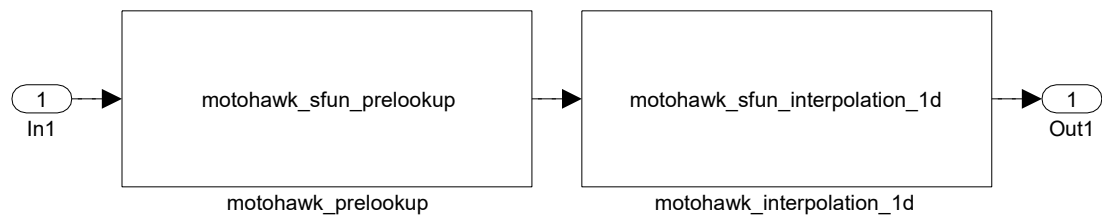
function



MotoHawk (RTW)  
Code Coverage Test Bit

motohawk\_sfun\_shutdown\_power

motohawk\_shutdown\_power




f()

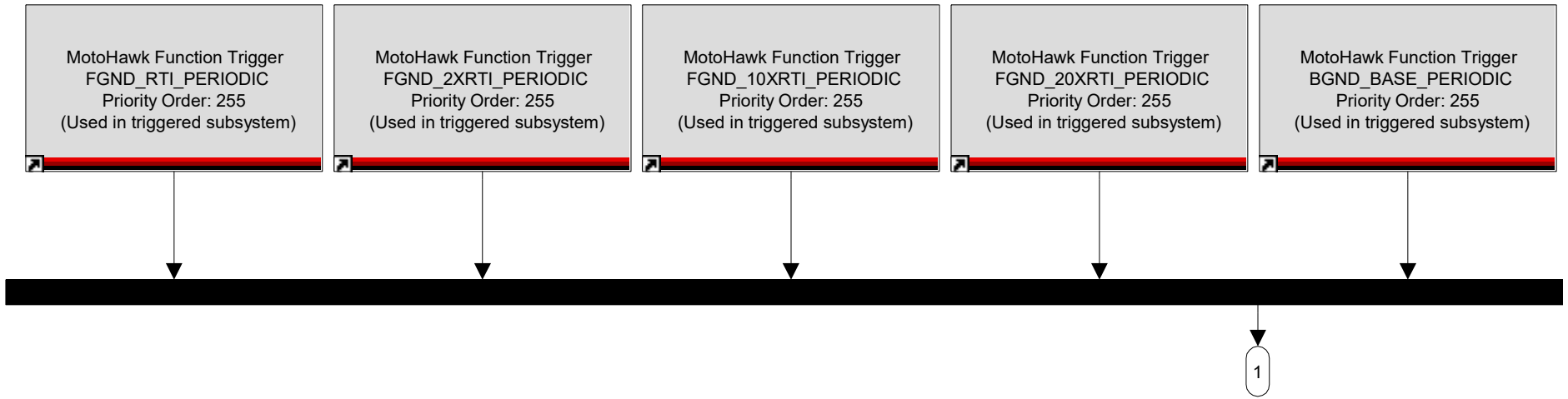
Trigger

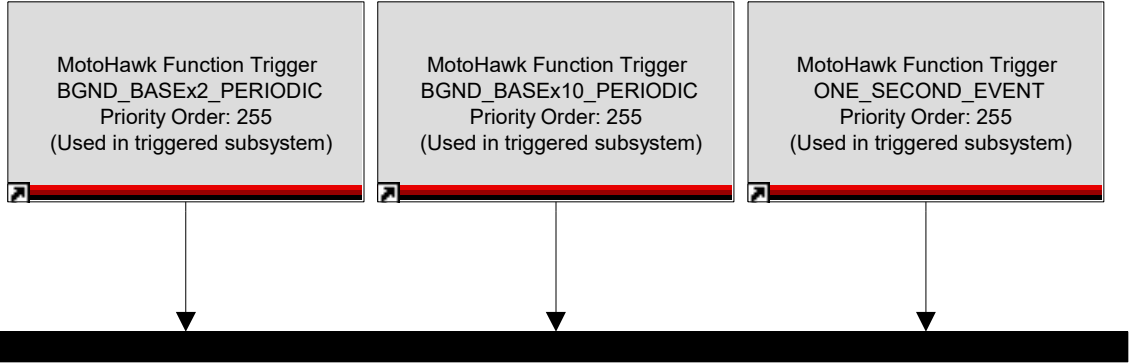
MotoHawk(RTW)  
Event Call

SCIL4

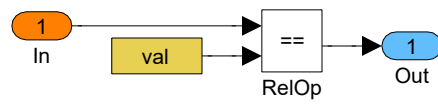


MotoHawk (RTW)  
Code Coverage Test Bit

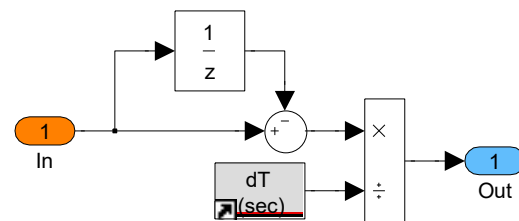




**Check how input compares to 'val'**

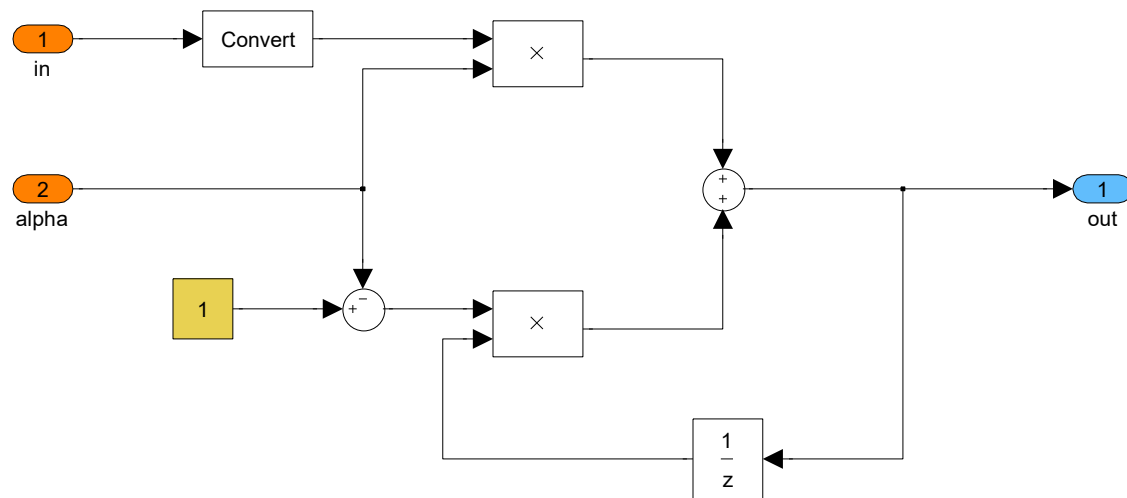


## Discrete Derivative





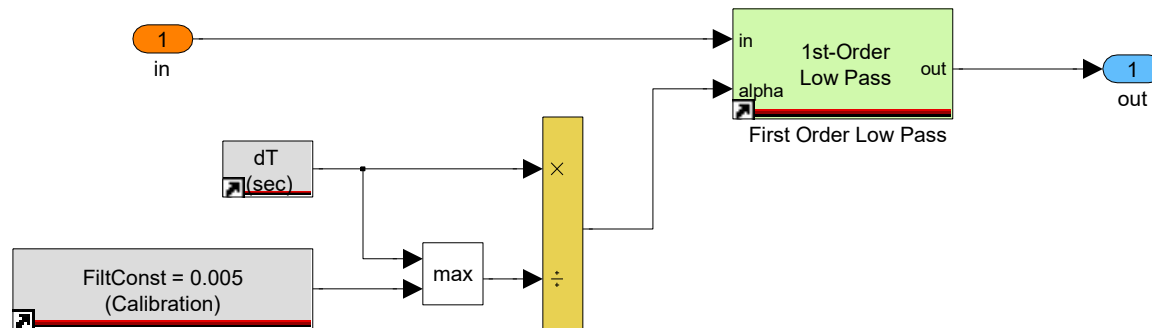
## 1st-Order Low-Pass Filter

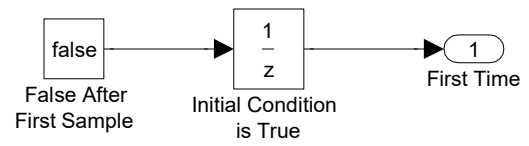


$$y[k] = a \cdot x[k] + (1-a) \cdot y[k-1]$$

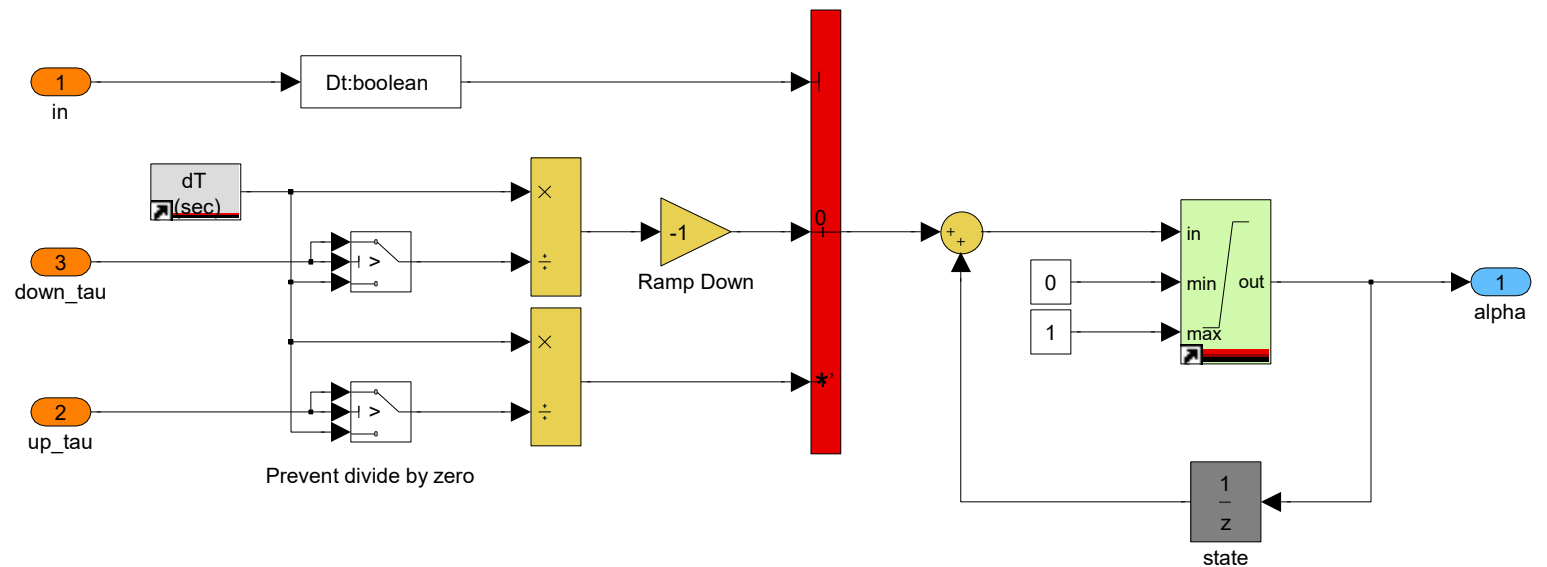
where  $a = t/T$

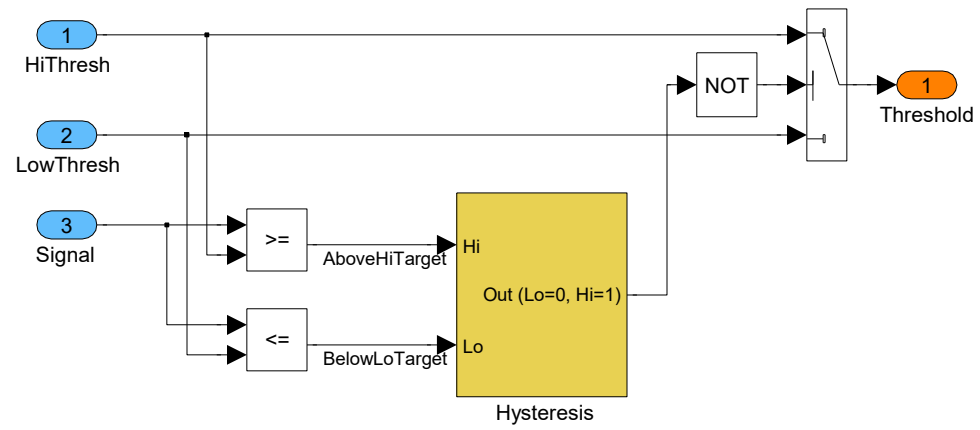
## Tunable 1st-Order Low-Pass Filter



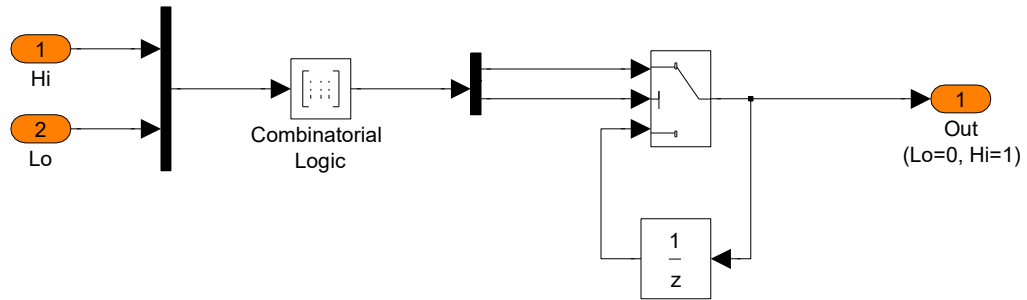


**Convert a boolean input signal  
to a more slowly ramping 'alpha' from 0 to 1**

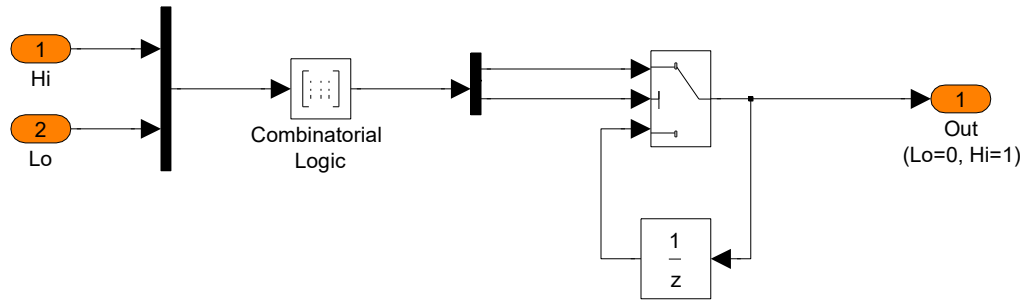




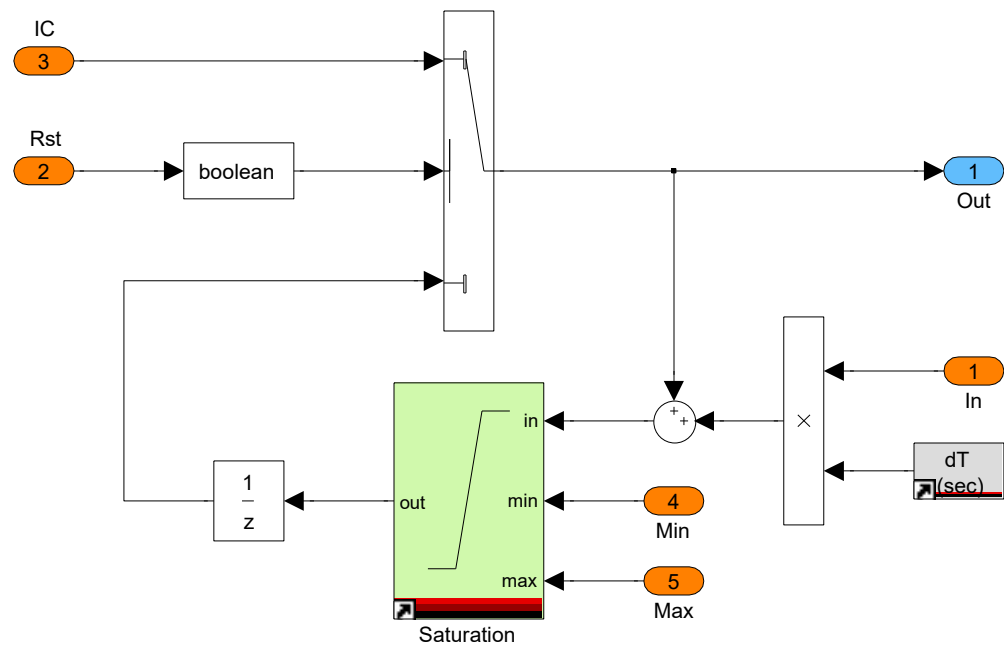
$Hi \ \& \ !Lo \Rightarrow Hi$   
 $Lo \ \& \ !Hi \Rightarrow Lo$   
Otherwise, Hold



$Hi \ \& \ !Lo \Rightarrow Hi$   
 $Lo \ \& \ !Hi \Rightarrow Lo$   
Otherwise, Hold

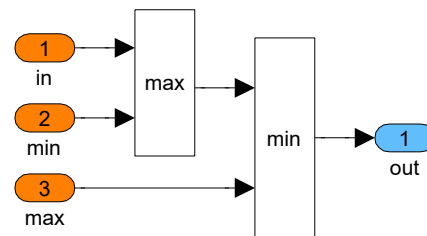


## Discrete Integrator, with output and state saturation.

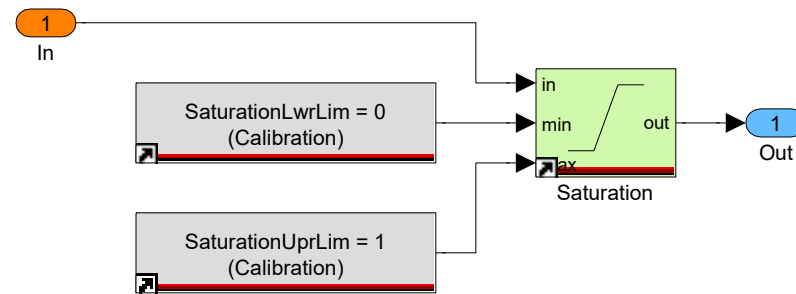




**Output 'In' when between Min and Max,  
and saturate against limits otherwise.**



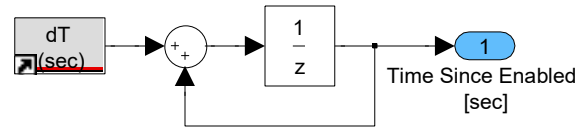
## Calibratable wrapper around the Saturation block

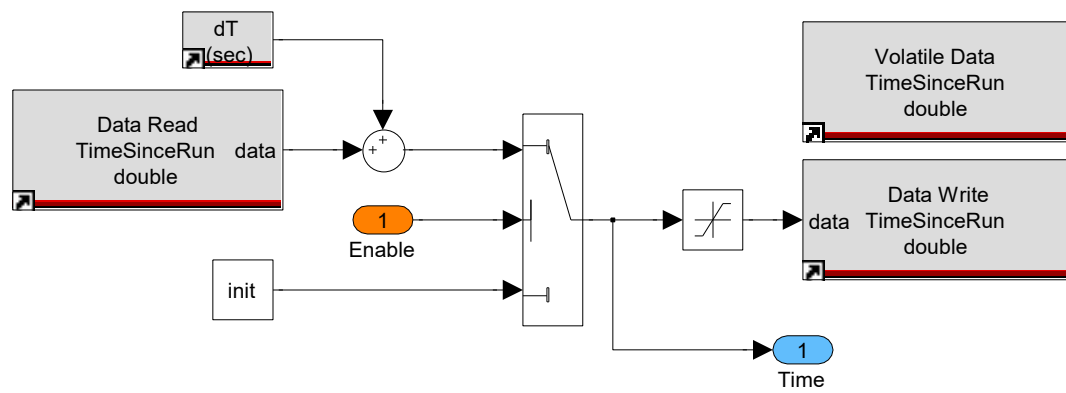


## Output the time since enabled, by summing up 'dt'.

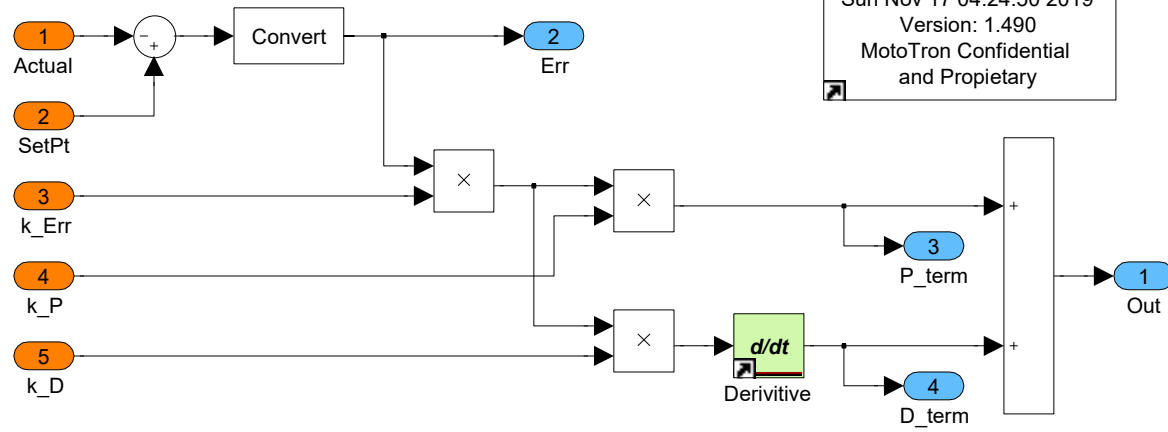
If this block is in an enabled subsystem that resets its states, then the count will reset as well. Otherwise, the count will resume from where it left off.

Note that when the Sample Time of the the 'dt' is non-positive, the block will only output its Initial Value if the enabled subsystem is set to reset its state. This means that the count will immediately 'catch up' if the enabled subsystem holds its states.



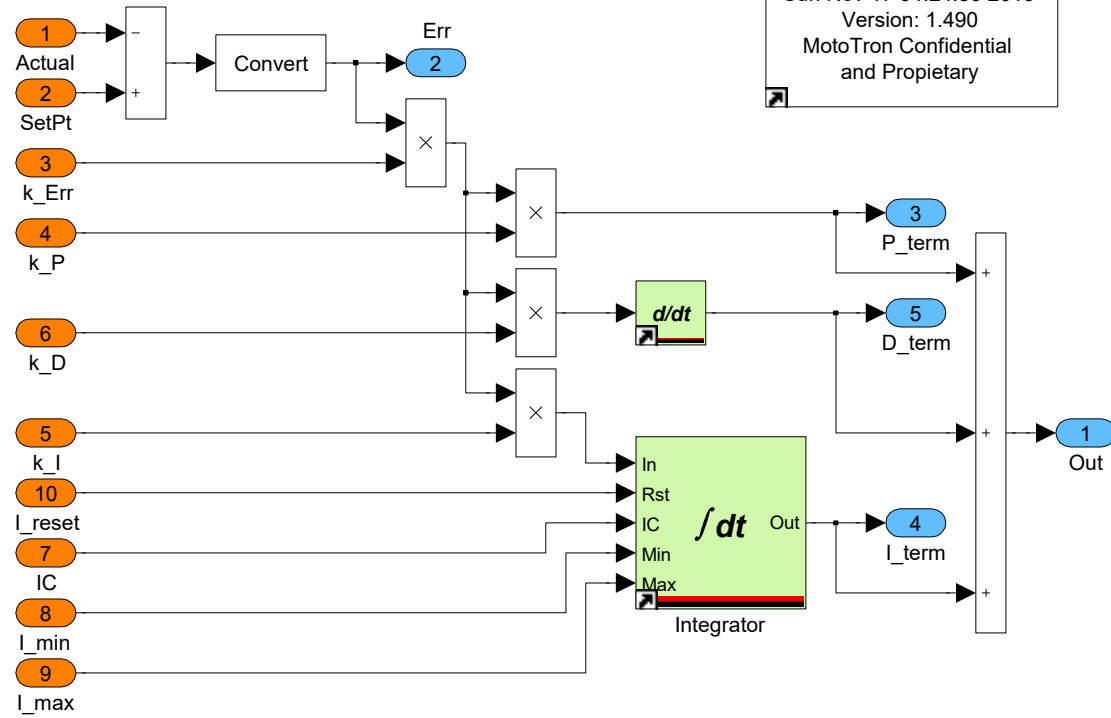


# Universal PD Controller

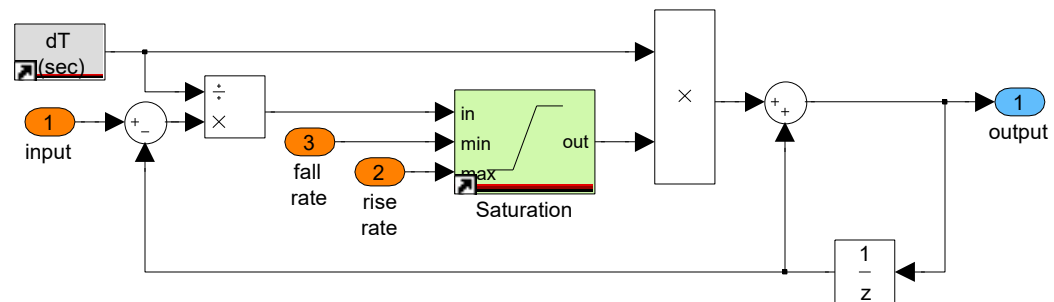


# Universal PID Controller

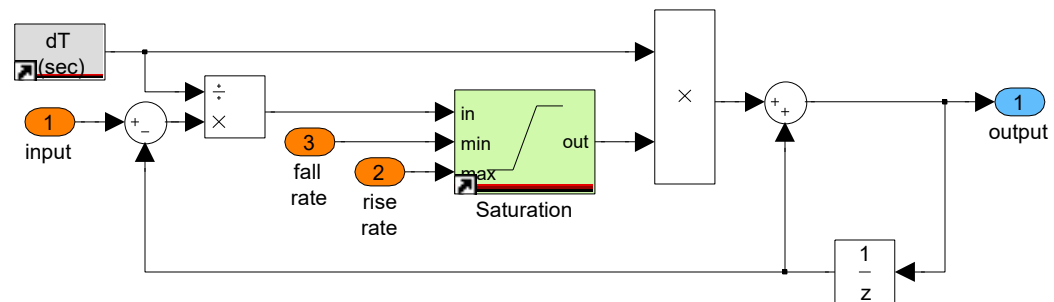
Model Info:  
Author: Suhre, Poorman  
Thu Mar 18 14:13:06 2004  
Sun Nov 17 04:24:50 2019  
Version: 1.490  
MotoTron Confidential  
and Proprietary



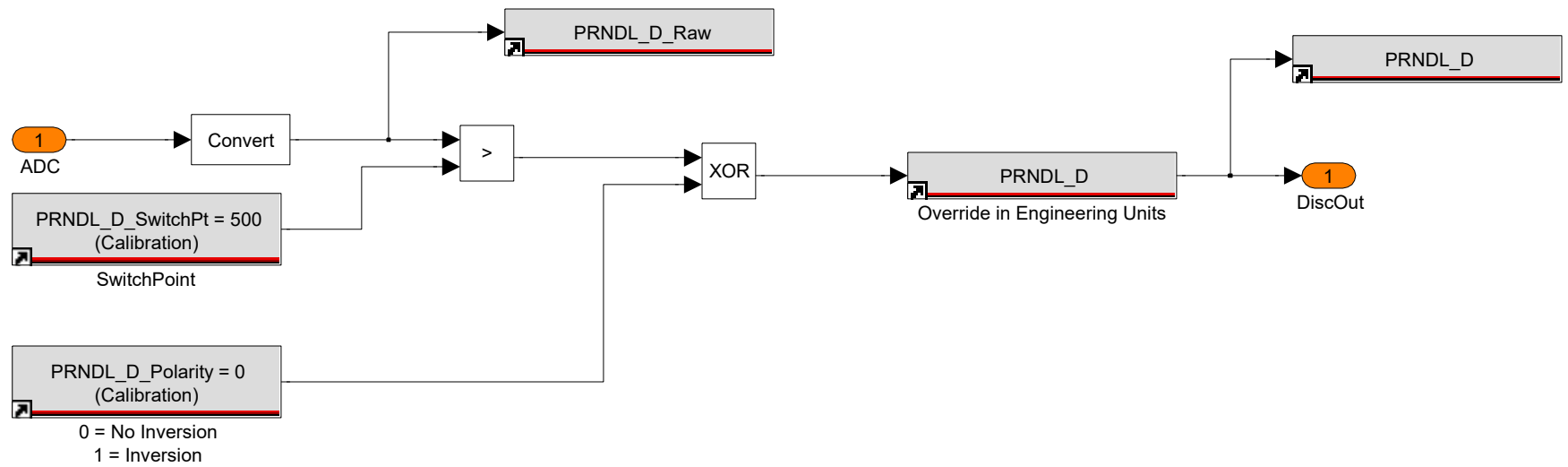
### Rate Limiter - Limit allowable change in signal per timestep

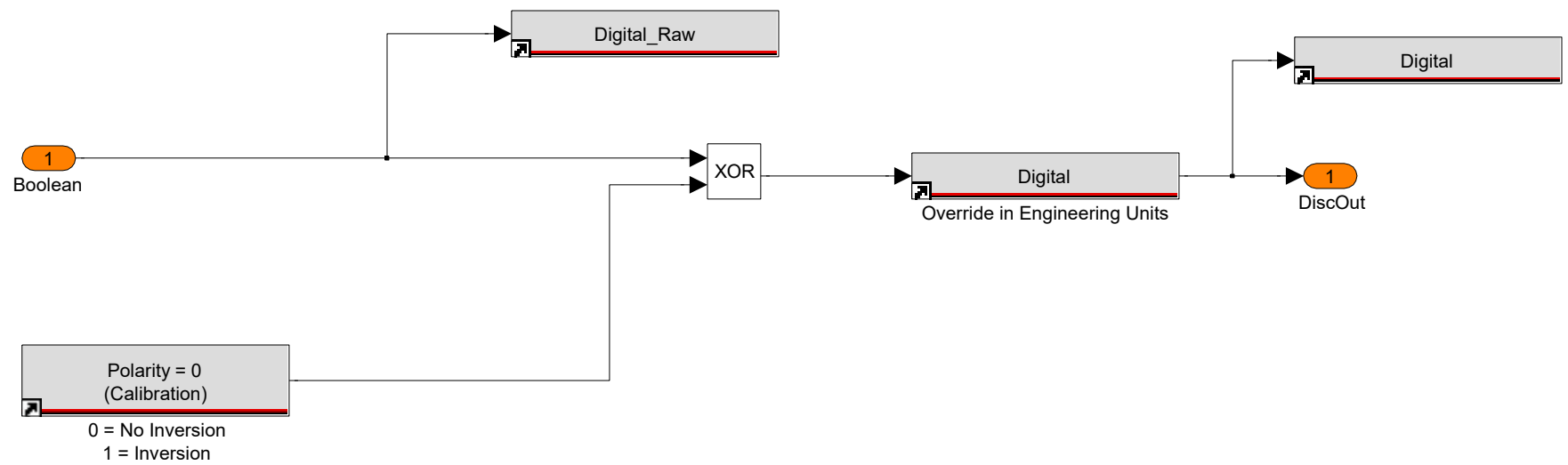


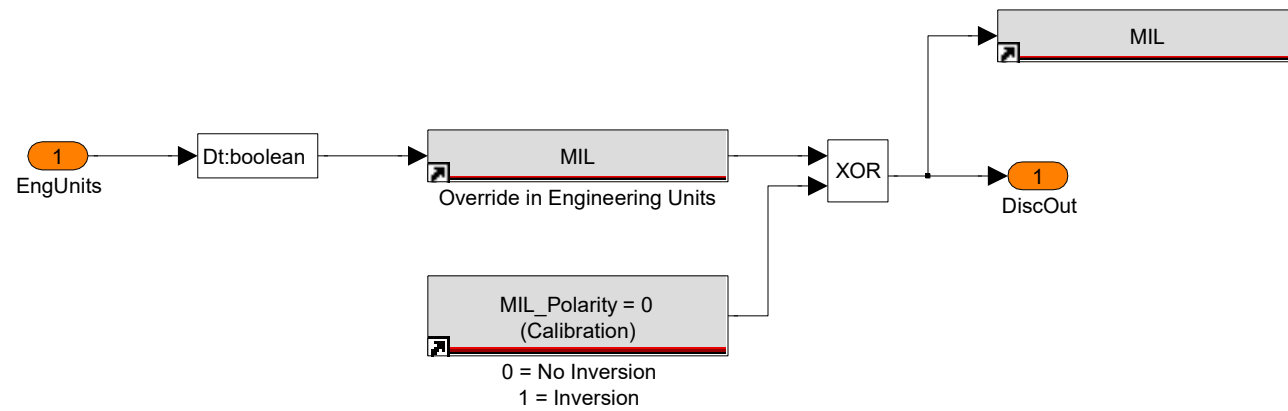
### Rate Limiter - Limit allowable change in signal per timestep

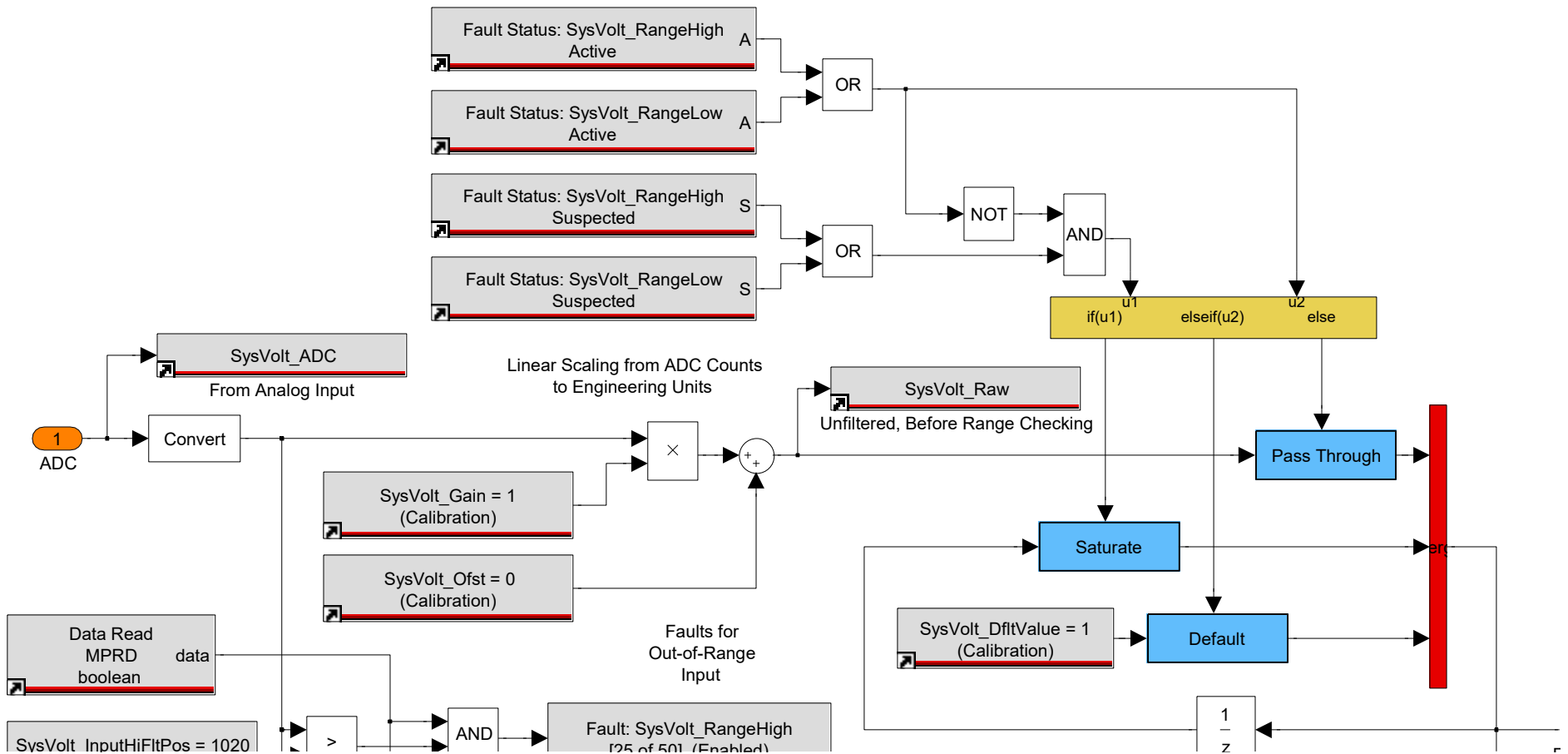




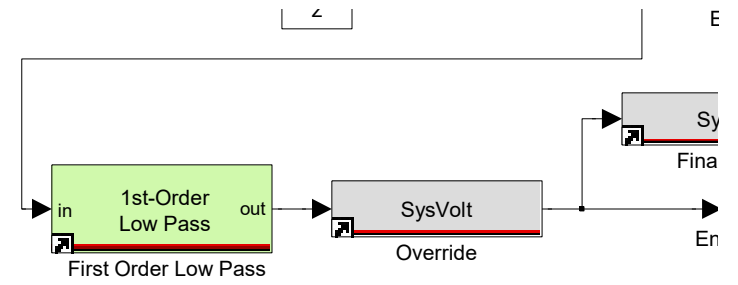
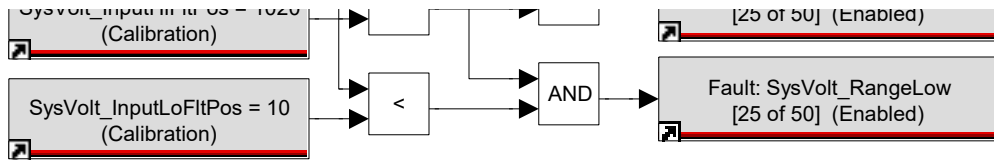












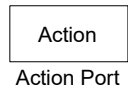
EngUnitsRaw

ysVolt

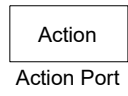
al Value

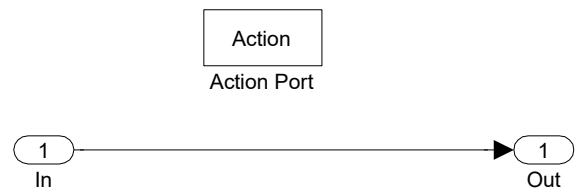
► 1

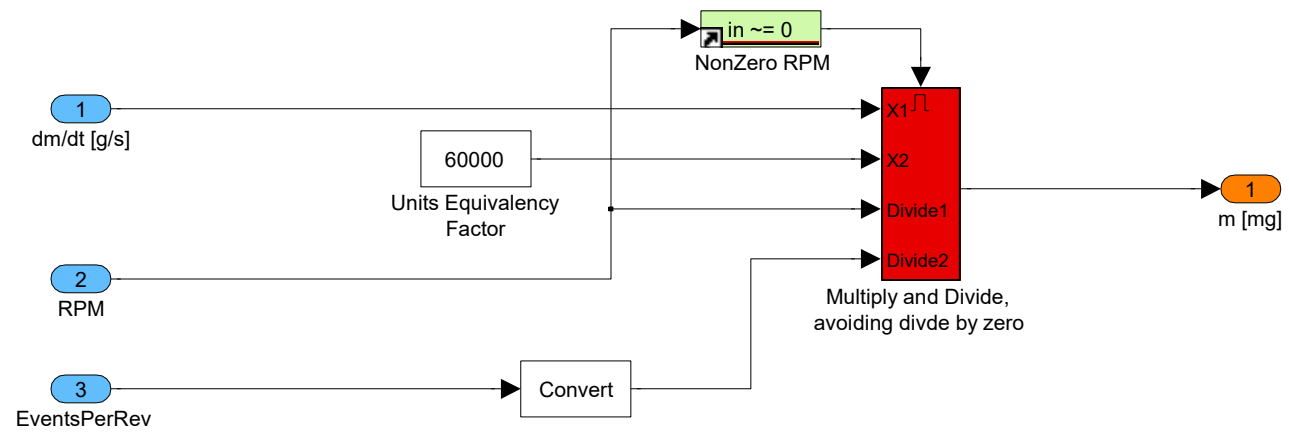
ngUnitsFilt

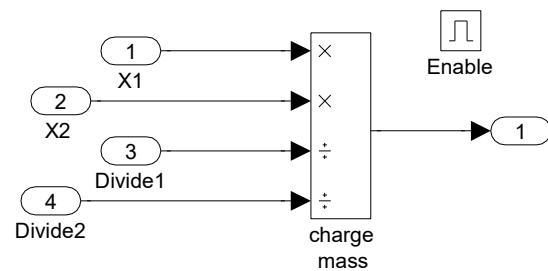


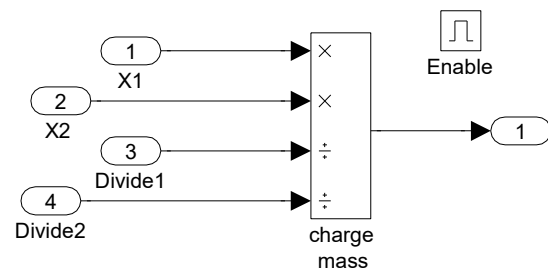


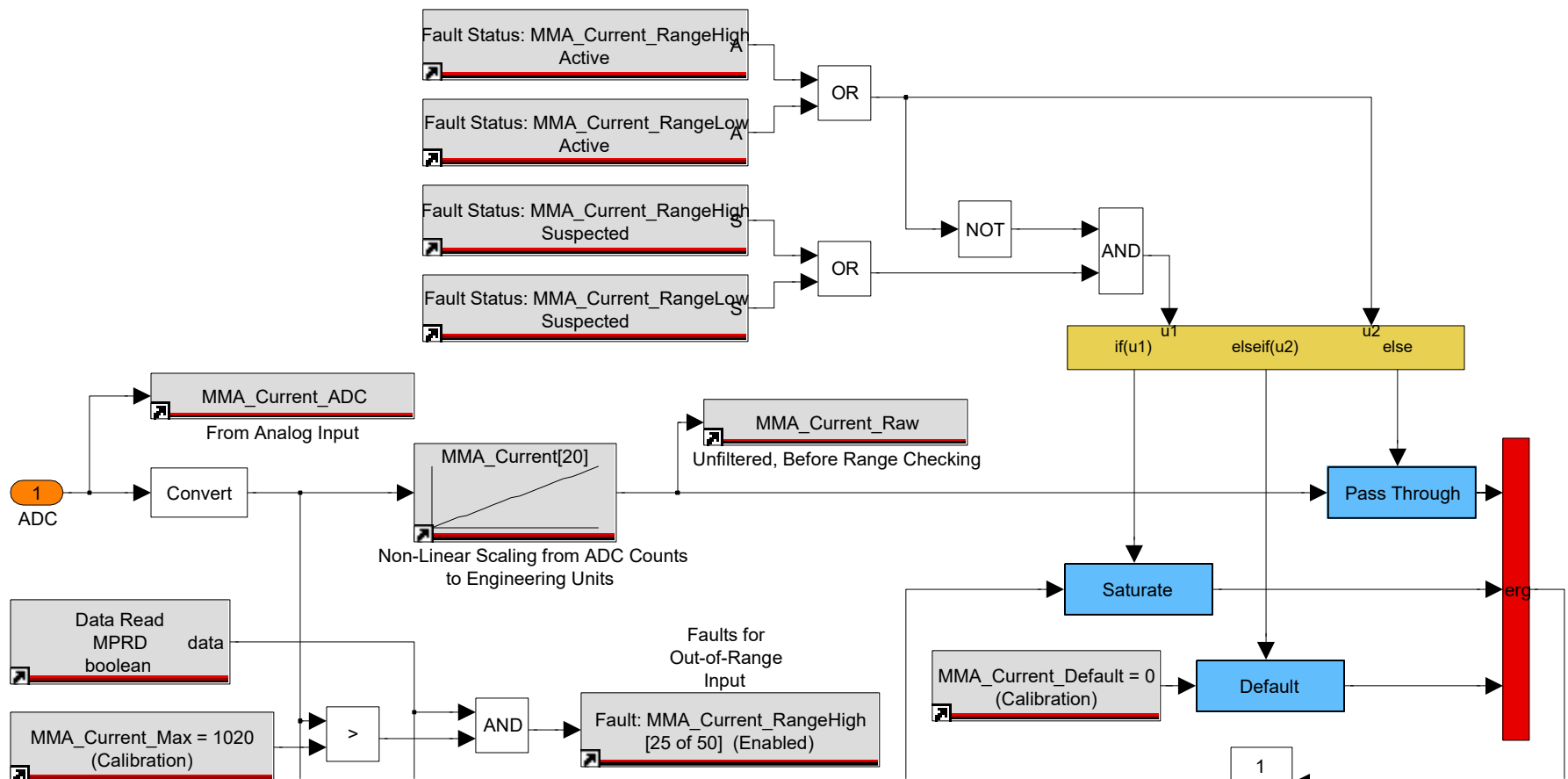




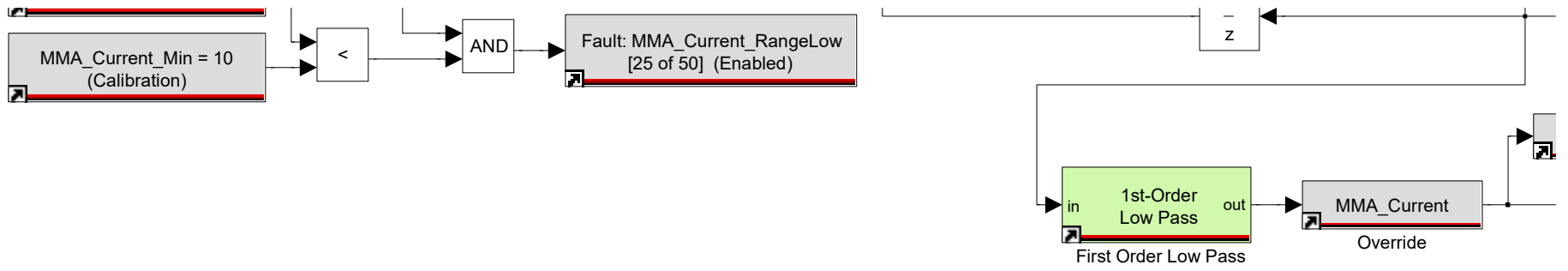










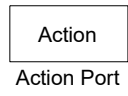


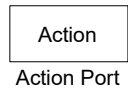


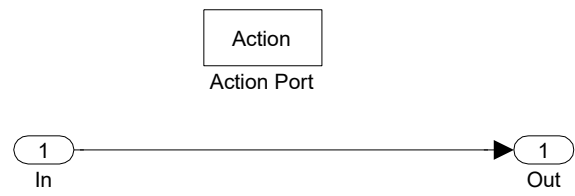
→ 2  
EngUnitsRaw

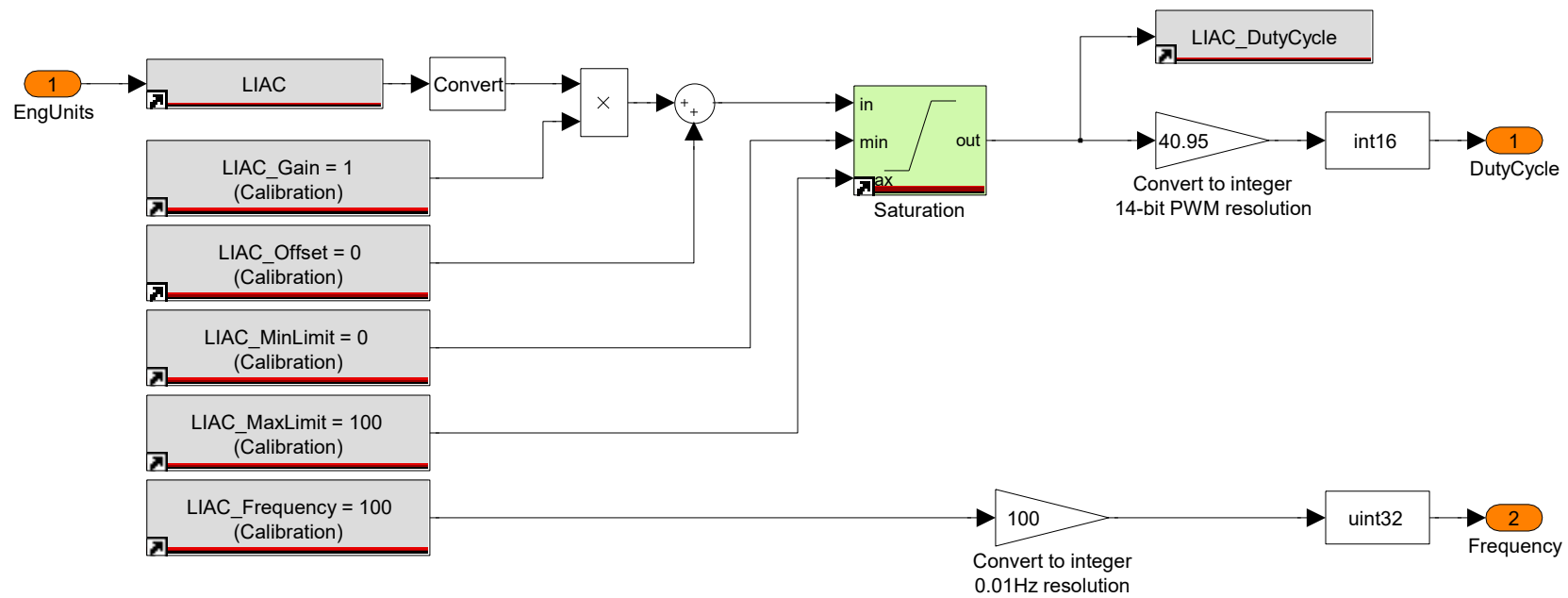
MMA\_Current  
Final Value

→ 1  
EngUnitsFilt

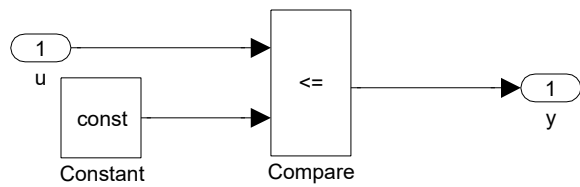








The outputs of this block are designed to be directly connected to a MotoHawk PWM block.





EmptySubsystem

