































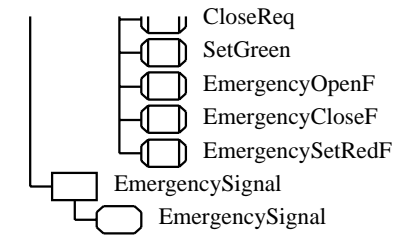


## Table of Contents

 rw C:\Documents and Settings\Administrator\My Documents\SAM2002contest\Li\_probert\_williams\Railway.sdt

 rw C:\Documents and Settings\Administrator\My Documents\SAM2002contest\Li\_probert\_williams\

—	High Level Requirements Model MSCs	4		
	HL01	4	rw	HL01.msc
	HL02	5	rw	HL02.msc
	HL03	6	rw	HL03.msc
	HL04	7	rw	HL04.msc
—	Critical Scenario Model MSCs	8		
	CS01	8	rw	CS01.msc
	CS02	9	rw	CS02.msc
	CS03	10	rw	CS03.msc
—	Manual Override Model MSCs	11		
	MC01	11	rw	MC01.msc
	MC02	12	rw	MC02.msc
—	Emergency Handling MSCs	13		
	EM01	13	rw	EM01.msc
	EM02	14	rw	EM02.msc
	EM03	15	rw	EM03.msc
	EM04	16	rw	EM04.msc
	EM05	17	rw	EM05.msc
—	SDL System Structure	18		
	RailwayCrossing	18	rw	RailwayCrossing.ssy
	tr (...) : Tracks			
	Tracks	20	rw	Tracks.sbt
	OutSensor	21	rw	OutSensor.spr
	InSensor	22	rw	InSensor.spr
	TrainSignal	23	rw	TrainSignal.spr
	TrackControl	24	rw	TrackControl.spr
	CarSensor	26	rw	CarSensor.sbk
	CarSensor	27	rw	CarSensor.spr
	TheGate	28	rw	TheGate.sbk
	TheGate	29	rw	TheGate.spr
	Controller	30	rw	Controller.sbk
	Controller	31	rw	Controller.spr
	ResetData	46	rw	ResetData.spd
	ResetTimer	47	rw	ResetTimer.spd
	FindSender	48	rw	FindSender.spd



49 rw CloseReq.spd  
 50 rw SetGreen.spd  
 51 rw EmergencyHandle.spd  
 52 rw EmergencyCloseF.spd  
 53 rw EmergencySetRedF.spd  
 54 rw EmergencySignal.sbk  
 55 rw EmergencySignal.spr

ModifiedCarSensor

56 rw ModifiedCarSensor.spr

TestCases

rw TestCases.ssc

— Simulator Trace MSCs

TestCase1\_1

rw TestCase1\_1.msc

TestCase1\_2

rw TestCase1\_2.msc

TestCase1\_3

rw TestCase1\_3.msc

TestCase1\_4

rw TestCase1\_4.msc

TestCase1\_5

rw TestCase1\_5.msc

TestCase1\_6

rw TestCase1\_6.msc

TestCase1\_7

rw TestCase1\_7.msc

TestCase2\_1

rw TestCase2\_1.msc

TestCase2\_2

rw TestCase2\_2.msc

TestCase2\_3

rw TestCase2\_3.msc

TestCase2\_4

rw TestCase2\_4.msc

TestCase3\_1

rw TestCase3\_1.msc

TestCase3\_2

rw TestCase3\_2.msc

TestCase3\_3

rw TestCase3\_3.msc

TestCase3\_4

rw TestCase3\_4.msc

TestCase3\_5

rw TestCase3\_5.msc

TestCase4\_1

rw TestCase4\_1.msc

TestCase4\_2

rw TestCase4\_2.msc

TestCase4\_3

rw TestCase4\_3.msc

TestCase4\_4

rw TestCase4\_4.msc

TestCase4\_5

rw TestCase4\_5.msc

TestCase4\_6

rw TestCase4\_6.msc

TestCase4\_7

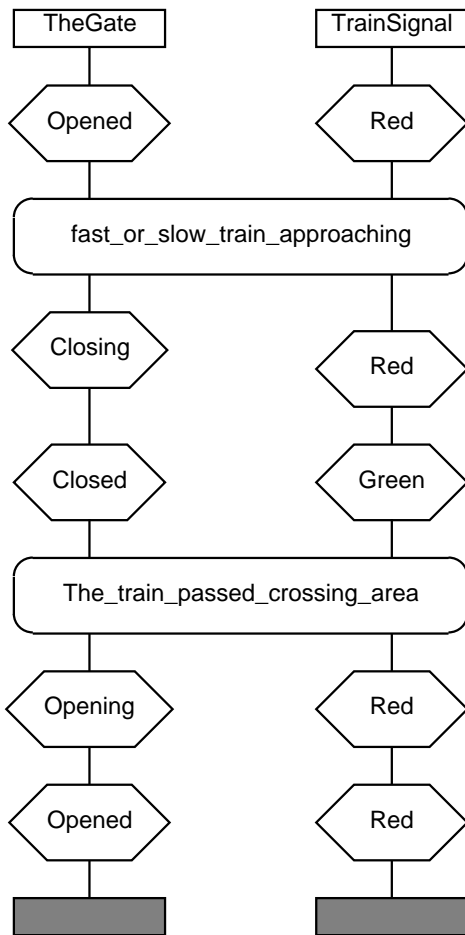
rw TestCase4\_7.msc

TestCase4\_8

rw TestCase4\_8.msc

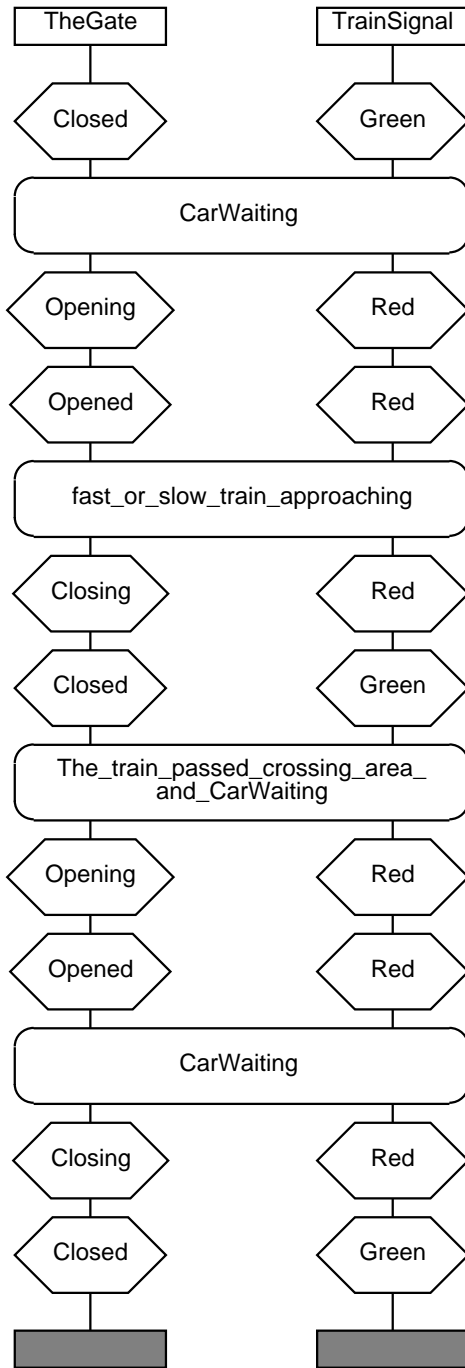


# MSC HL01



/\* High level requirement model #1  
Simplest Generic Scenario (no car)  
-- Applicable to all strategies  
Start State: Gate is Open and TrainSignal is Red  
End State: Gate is Open and TrainSignal is Red  
Actions:  
1. When a train (fast or slow) is approaching,  
close the gate, and set TrainSignal to Green.  
2. After the train passed the crossing area,  
set TrainSignal to Red, and open the gate.  
\*/

# MSC HL02



/\* High level requirement model #2

All trains have precedence scenario  
 --- Applicable to Strategy 2

(In Strategy 2, normally the gate is closed)

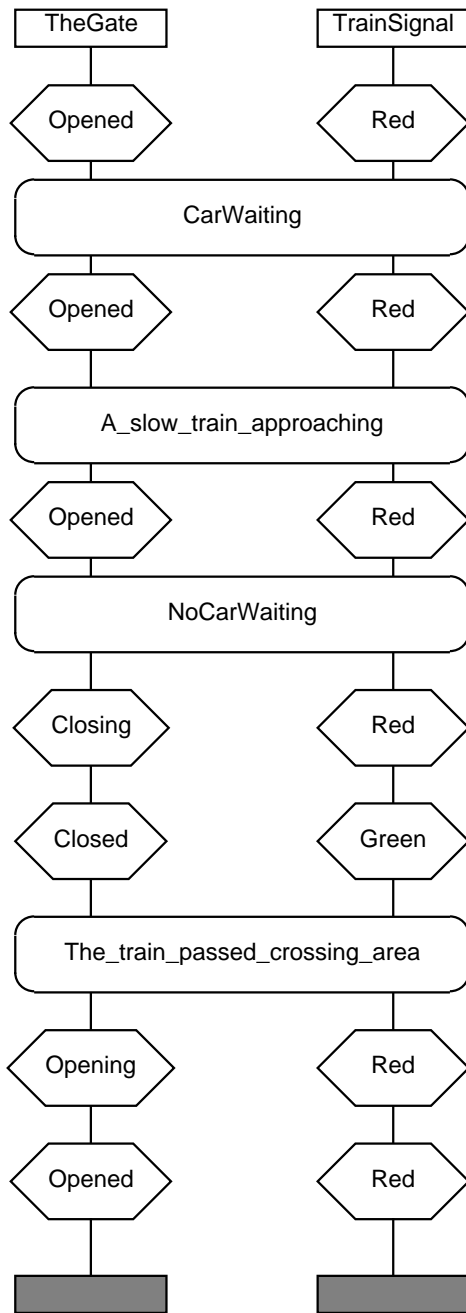
Start State: Gate is Closed and TrainSignal is Green  
 End State: Gate is Closed and TrainSignal is Green

Actions:

1. Car waiting and no train, open the gate
2. A train (fast/slow) is approaching, close the gate
3. The train passed the crossing area, and car waiting, open the gate
4. No car waiting, close the gate.

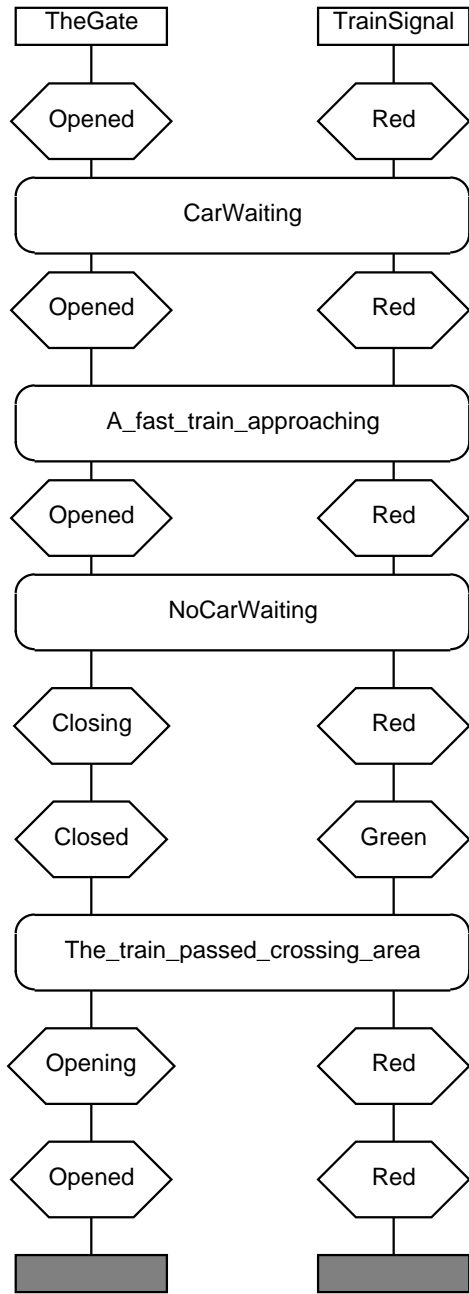
\*/

# MSC HL03



/\* High level requirement model #3  
 Normal slow train urban scenario  
 --- Applicable to Strategy 3  
 (In Strategy 3, normally the gate is open to expedite car traffic)  
 Start State: Gate is Open and TrainSignal is Red  
 End State: Gate is Open and TrainSignal is Red  
 Actions:  
 1. Car waiting, keep the gate open.  
 2. A slow train is approaching when car waiting, stop the train and keep the gate open.  
 3. No car waiting and a slow train is waiting, close the gate, and set TrainSignal Green.  
 4. The train has passed crossing area, set TrainSignal Red, and open the gate  
 \*/

# MSC HL04



/\* High level requirement model #4

Car has precedence scenario  
 -- Applicable to Strategy 4

(In Strategy 4, normally the gate is open to expedite car traffic)

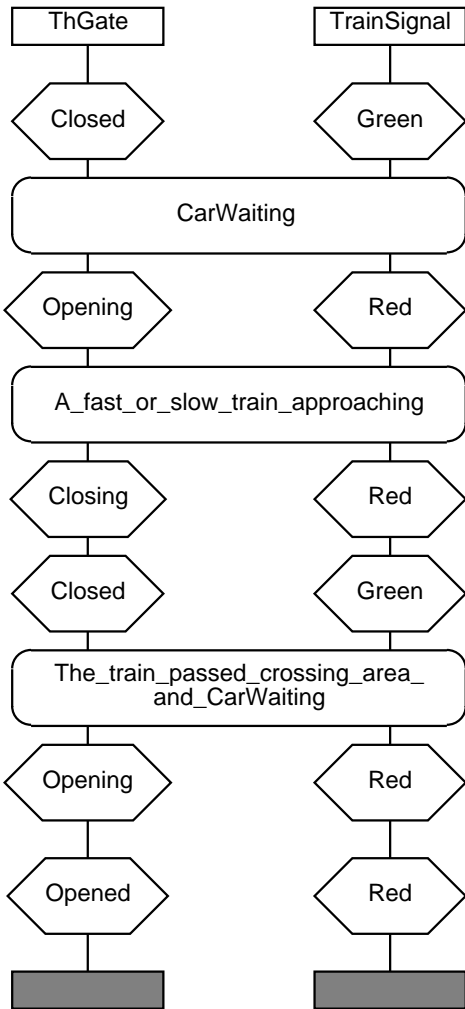
Start State: Gate is Opened and TrainSignal is Red  
 End State: Gate is Opened and TrainSignal is Red

Actions:

1. Car waiting, keep the gate open.
2. A fast train is approaching and car waiting, stop the train and keep the gate open
3. No car waiting, close the gate, and set TrainSignal Green
4. The train has passed crossing area, set TrainSignal Red and open the gate

\*/

# MSC CS01



/\* Critical Scenario model #1

High Risk Critical Race Scenario – Trains have precedence  
 -- applicable to Strategy 2

(In Strategy 2, normally the gate is closed)

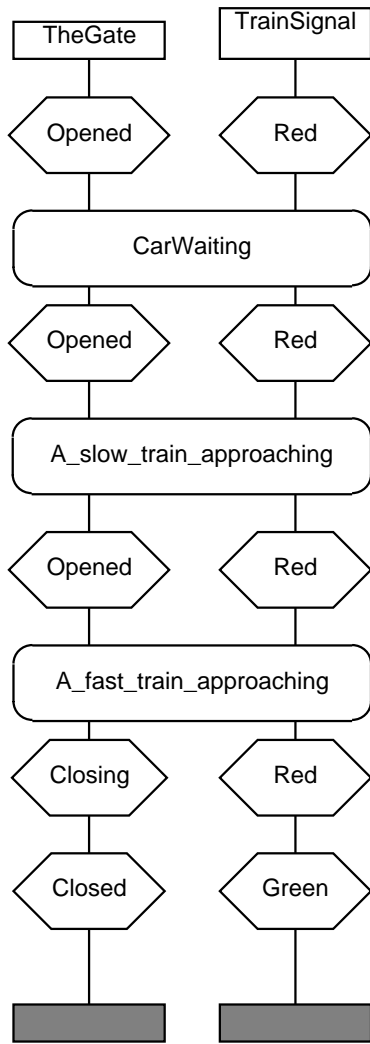
Start State: Gate is Closed and TrainSignal is Green  
 End State: Gate is Open and TrainSignal is Red

Actions:

1. Car waiting and no train,  
 set TrainSignal Red and open the gate
2. While the gate is raising,  
 a train (fast/slow) approaches.  
 Stop raising the gate, and close it instead.  
 Then set TrainSignal Green
3. When the train has passed crossing area,  
 set TrainSignal Red and open the gate

\*/

# MSC CS02

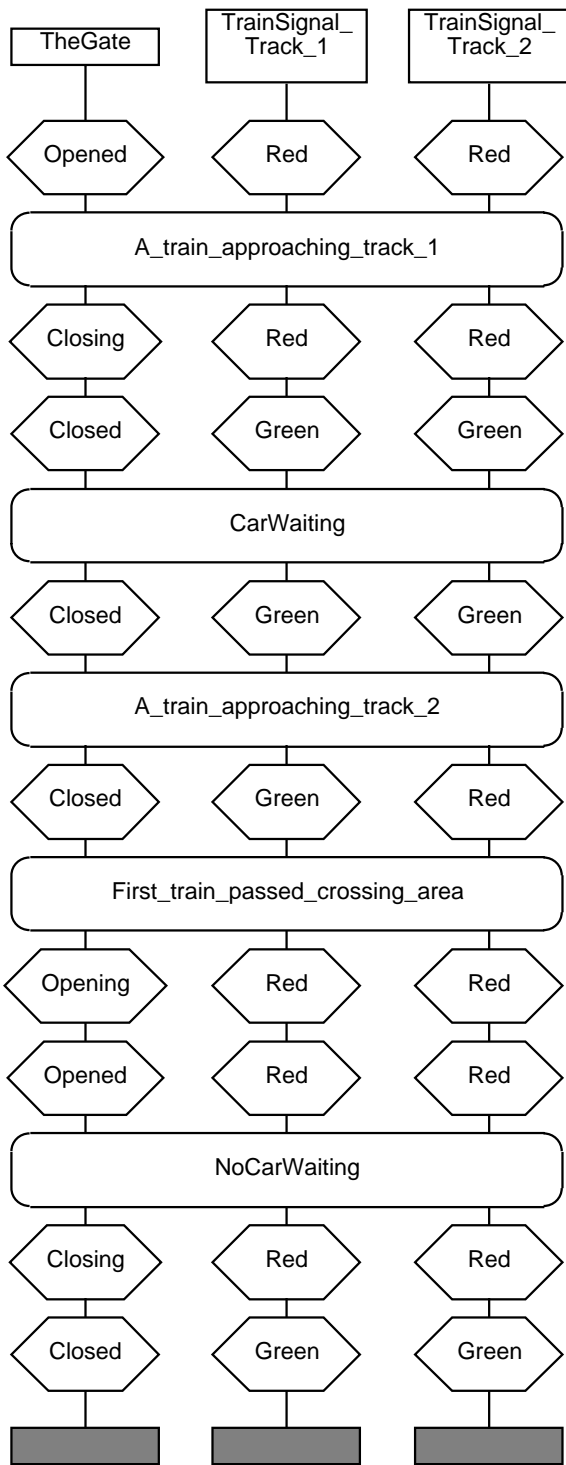


```
/* Critical Scenario model #2
Critical Race Scenario – Slow and Fast Train Arriving
-- Applicable to Strategy 3
(In Strategy 3, normally the gate is open to expedite car traffic)

Start State: Gate is Open and TrainSignal is Red
End State: Gate is Closed and TrainSignal is Green

Actions:
1. Car waiting, keep the gate open.
2. A slow train approaches and car waiting,
stop the train and keep the gate open
3. A fast train approaches,
close the gate, set TrainSignal to Green and let all trains go
*/
```

# MSC CS03



/\* Critical Scenario model #3

Critical Race – Multiple Tracks and Trains -- Cars Have Precedence

-- Applicable to Strategy 4

(In Strategy 4, normally the gate is open to expedite car traffic)

Start State: Gate is Open and TrainSignal is Red

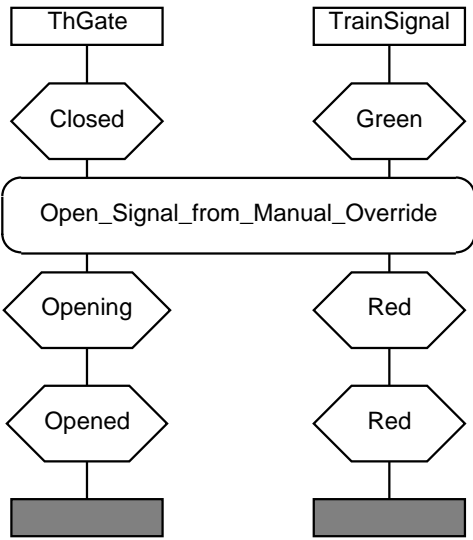
End State: Gate is Closed and TrainSignal is Green

Actions:

1. A train approaches Track 1, close the gate, and set TrainSignal to Green
2. Car waiting and the train is still in the crossing area, keep cars waiting, until the train leaves
3. Another train approaches Track 2. Since car waiting and cars have precedence, stop the train by setting TrainSignal on Track 2 to Red, and keep the gate closed
4. When the train on Track 1 has passed the crossing area, set TrainSignal on Track 1 to Red, and open the gate for cars
5. When no car waiting, close the gate for the waiting train, and set TrainSignal on all tracks to Green.

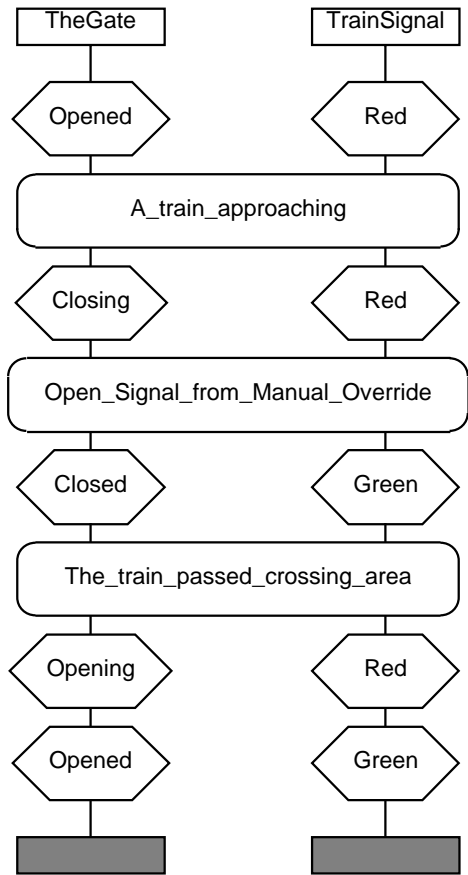
\*/

# MSC MC01



```
/* Manual Override Scenario model #1  
Manually Open Gate – Safe Scenario  
Start State: Gate is Closed and TrainSignal is Green  
End State: Gate is Open and TrainSignal is Red  
Actions:  
Send Open signal manually  
-- No train in crossing area, open the gate  
*/
```

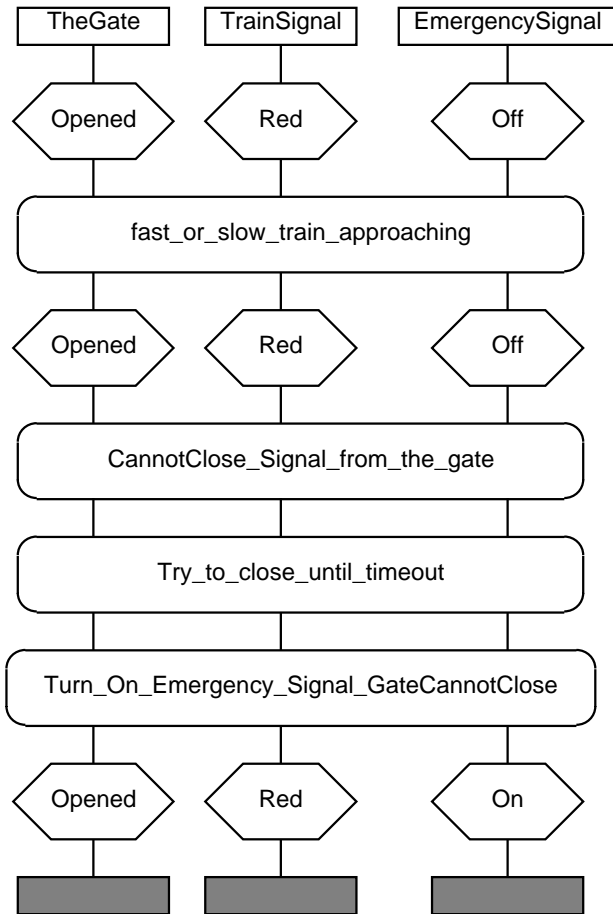
# MSC MC02



```

/* Manual Override Scenario model #2
Unsafe Manually Stopping Gate Closing – Train Approaching
Start State: Gate is Open and TrainSignal is Red
End State: Gate is Open and TrainSignal is Red
Actions:
Send Open signal manually when the gate is lowering down
-- Check if there are trains approaching.
If there is one, continue closing the gate,
set TrainSignal to Green and let the train pass.
After the train passes, open the gate
Note: The unsafe action is not allowed by RCS
*/
    
```

# MSC EM01



/\* Emergency Handling Model #1

Gate Cannot Close Scenario  
 -- Applicable to all strategies

Start State: Gate is Open  
 TrainSignal is Red  
 EmergencySignal is Off

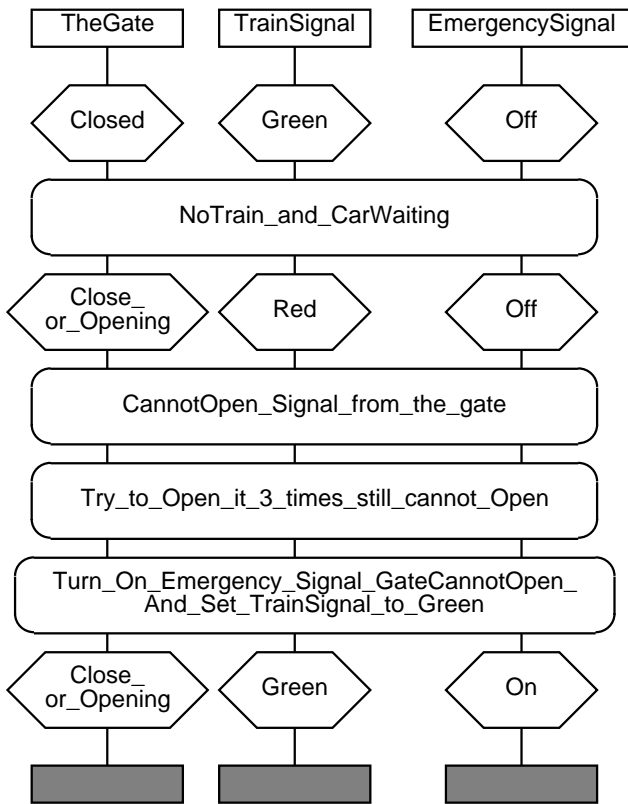
End State: Gate is Open  
 TrainSignal is Red  
 EmergencySignal is On

Actions:

1. A train approaches, try to close the gate
2. Gate cannot close -- either half way down or cannot move at all
3. Try to close it again until timeout (Train cannot stop safely after this moment)
4. Turn on Emergency Signal "GateCannotClose" to inform Trains and Cars and maintenance personnel  
 RCS is locked

\*/

# MSC EM02



```

/* Emergency Handling Model #2
Gate Cannot Open Scenario
-- Applicable to all strategies

Start State: Gate is Closed
              TrainSignal is Green
              EmergencySignal is Off

End State:   Gate is Closed/half way up
              TrainSignal is Green
              EmergencySignal is On

Actions:

1. No train in crossing area and Car Waiting
   set TrainSignal to Red and try to open the gate

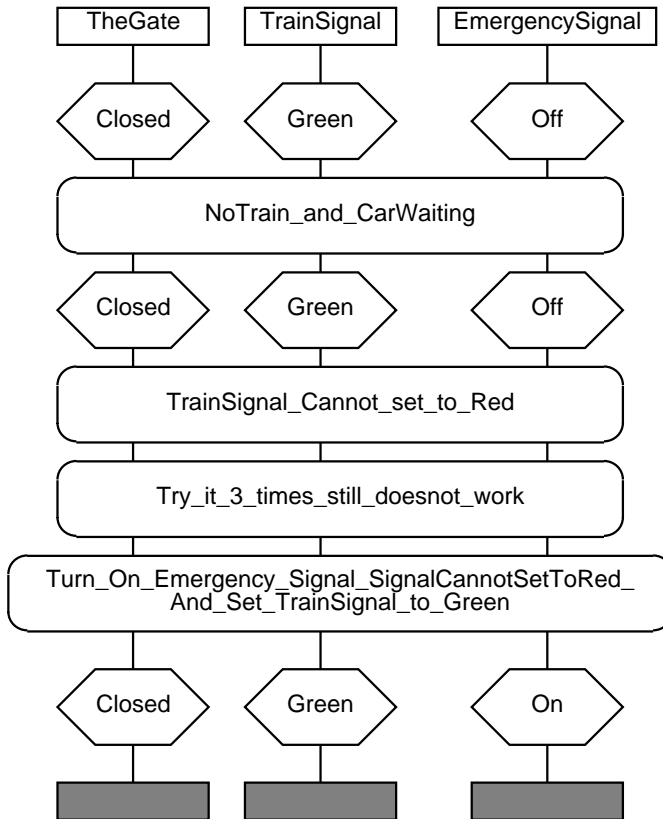
2. Gate cannot open
   -- either half way up or cannot move at all

3. Try to open it again for 3 times

4. Still cannot open,
   Turn on Emergency Signal "GateCannotOpen"
   to inform maintenance personnel
   and set TrainSignal to Green
   to let all upcoming train pass

*/
    
```

# MSC EM03



/\* Emergency Handling Model #3  
 TrainSignal Cannot Set to Red Scenario  
 -- Applicable to all strategies

Start State: Gate is Closed  
 TrainSignal is Green  
 EmergencySignal is Off

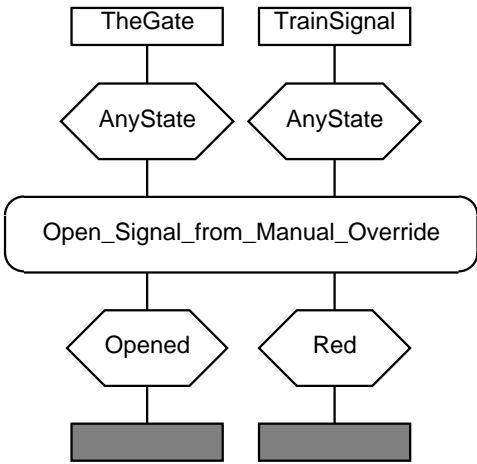
End State: Gate is Closed  
 TrainSignal is Green  
 EmergencySignal is On

Actions:

1. No train in crossing area and Car Waiting set TrainSignal to Red
2. One of the TrainSignals cannot be set to Red
3. Try it again for 3 times
4. Still cannot set to Red, Turn on Emergency Signal "TrainSignal Cannot Set to Red" to inform maintenance personnel and set TrainSignal to Green to let all upcoming train pass

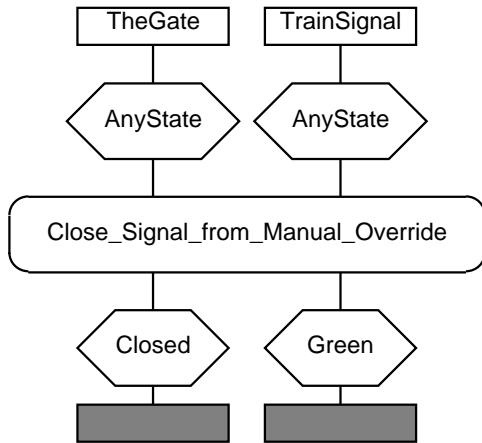
\*/

# MSC EM04



/\* Emergency Handling Model #4  
Unlock RCS After Emergency Scenario -- Open gate  
-- Applicable to all strategies  
Start State: Any state  
End State: Gate is Open  
          TrainSignal is Red  
Actions:  
A state after any emergency, send an Open signal  
from Manual override, the gate will open  
and TrainSignal is set Red. RCS is unlocked.  
\*/

# MSC EM05



/\* Emergency Handling Model #5

Unlock RCS After Emergency Scenario -- Close gate  
-- Applicable to all strategies

Start State: Any state  
End State: Gate is Closed  
TrainSignal is Green

Actions:

A state after any emergency, send a Close signal from Manual override, the gate will close and TrainSignal is set Green. RCS is unlocked.

\*/

## system RailwayCrossing

1(2)

```
/* RailwayCross system consists of a controller, a car sensor, a gate, a set of
emergency signal, and a number of tracks */
/* Each track consists of a track controller, an InSensor, an OutSensor and a set
of train signals */
```

```
/* Four Strategies can be used in this system, and they can be switched
dynamically when the system is running. They are:
```

- 1-- Manual Override
- 2 --Trains take precedence
- 3 -- Fast train - many cars - slow train
- 4 -- Many cars - train \*/

```
SIGNAL Yes, No,YesE,NoE; /* Yes--more than one car waiting*/
SIGNAL TrainC, Coming;
SIGNAL TrainL, Left;
SIGNAL GreenRQ, RedRQ;
SIGNAL Green, Red;
SIGNAL Open, Close;
SIGNAL Opened, Closed, Raising, Lowering;
```

```
SIGNALLIST TrainSignalRQ=GreenRQ, RedRQ;
SIGNALLIST TrainSignal=Green, Red;
SIGNALLIST GateSignal=Opened, Closed, Raising, Lowering;
SIGNALLIST GateControl=Open, Close;
```

```
SYNONYM NoOfTracks Integer =2; /* Constant--No of tracks*/
SYNONYM NoOfFastTracks Integer =1; /* Constant -- No of fast train tracks*/
```

```
/* Emergency Signals */
```

```
SIGNAL OpenF, CloseF, SetRedF;
SIGNAL OpenFail, CloseFail, SetRedFail;
```

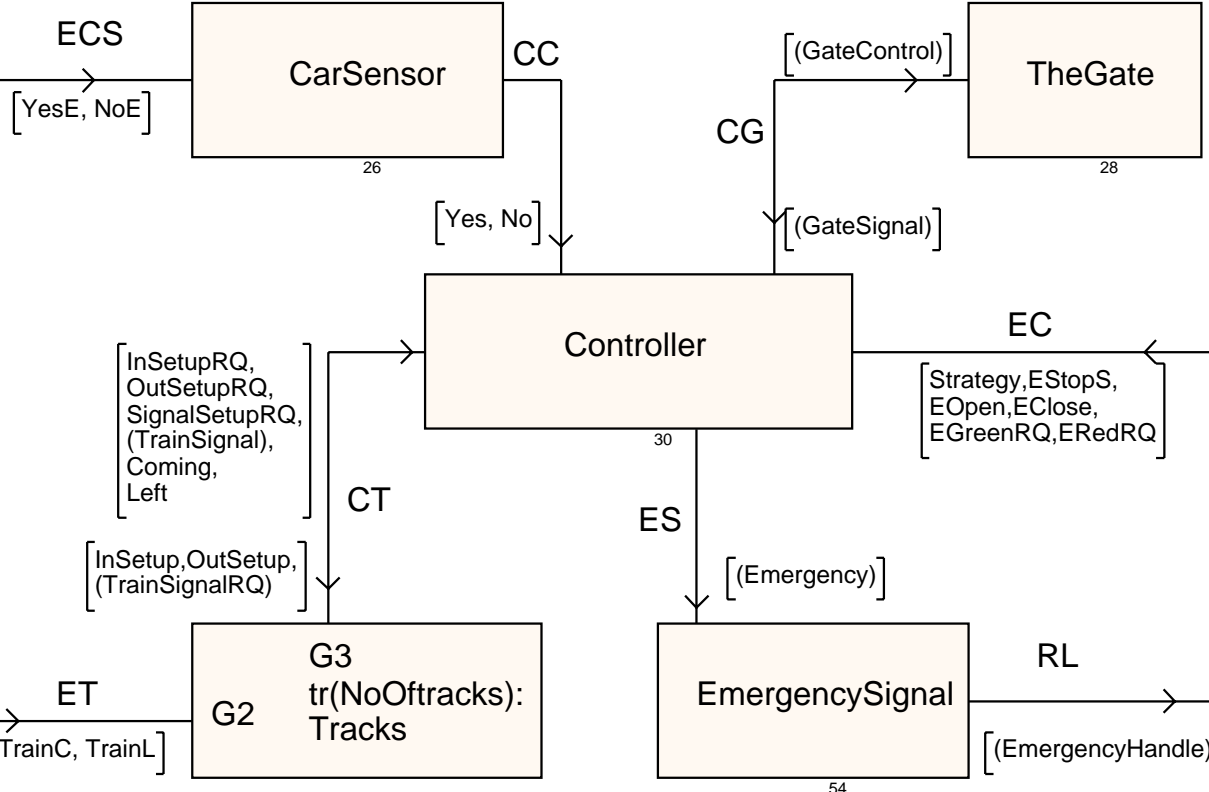
```
SIGNALLIST Emergency=OpenF, CloseF, SetRedF;
SIGNALLIST EmergencyHandle=OpenFail, CloseFail, SetRedFail;
```

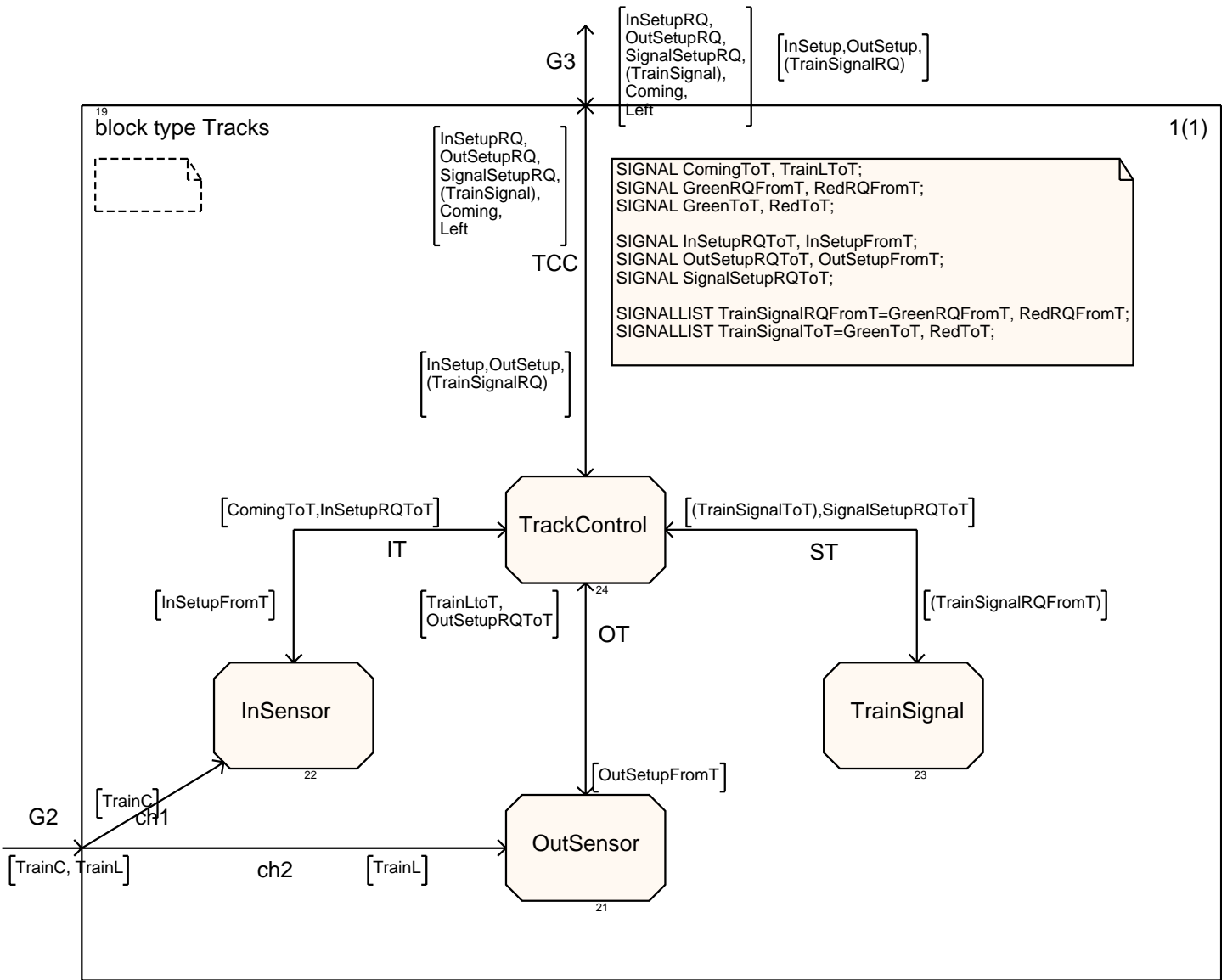
```
/* Manual Override signals */
```

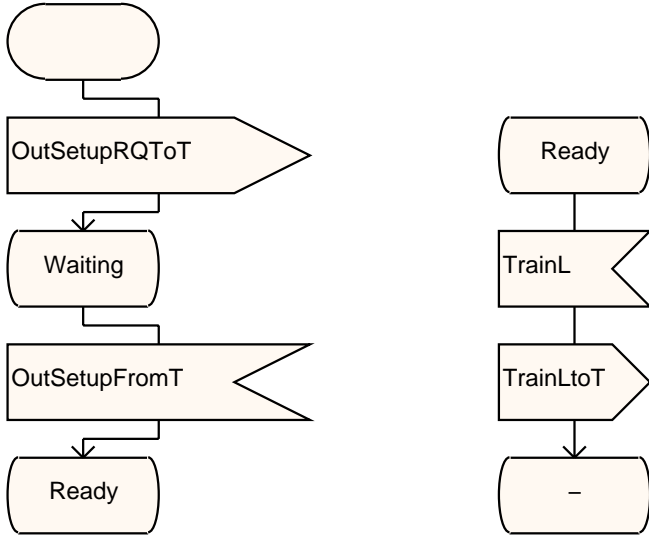
```
SIGNAL Strategy(Natural);
SIGNAL EStopS;
SIGNAL EOpen, EClose;
SIGNAL EGreenRQ, ERedRQ;
```

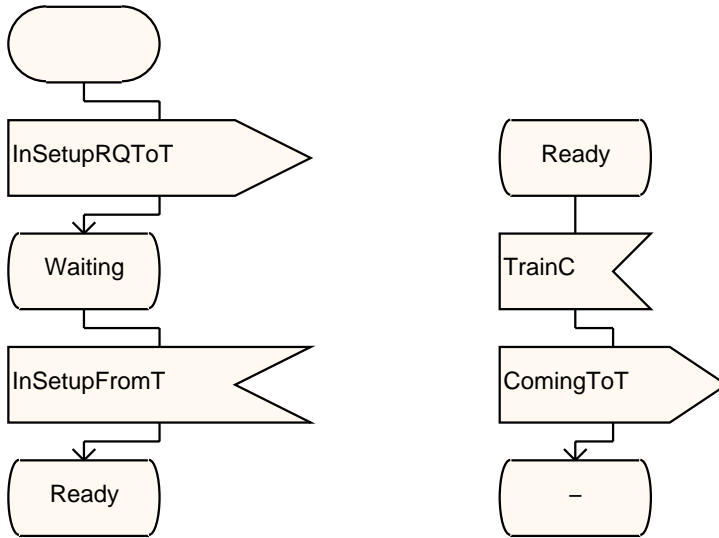
```
/* Setup signals */
```

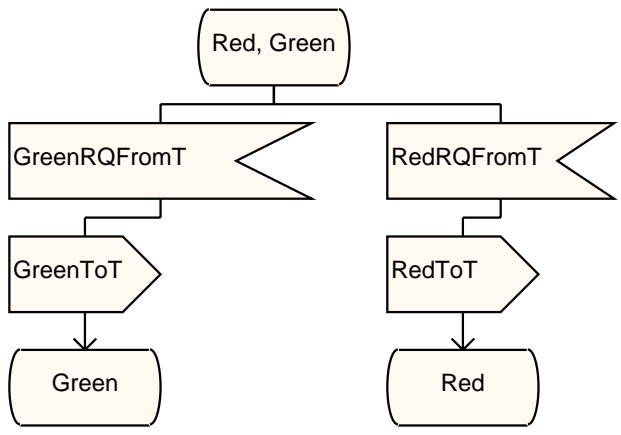
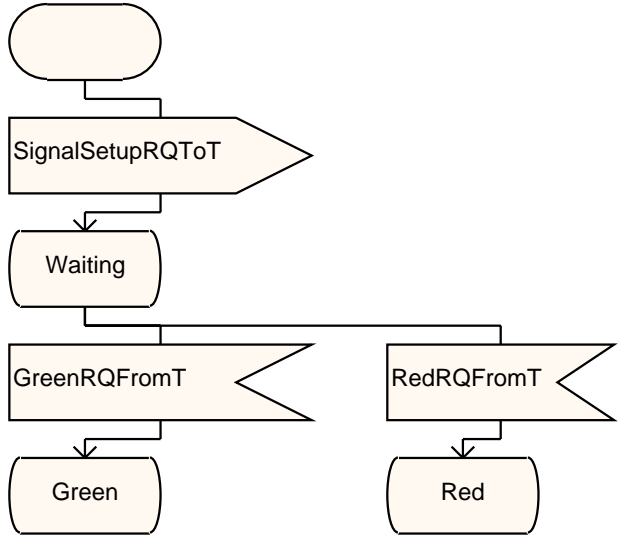
```
SIGNAL InSetupRQ, InSetup;
SIGNAL OutSetupRQ, OutSetup;
SIGNAL SignalSetupRQ;
```





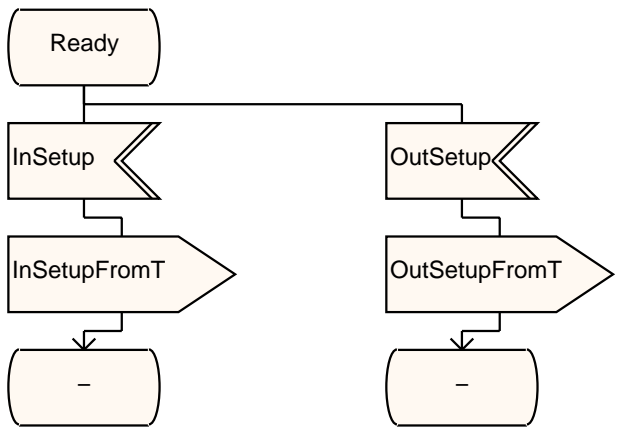
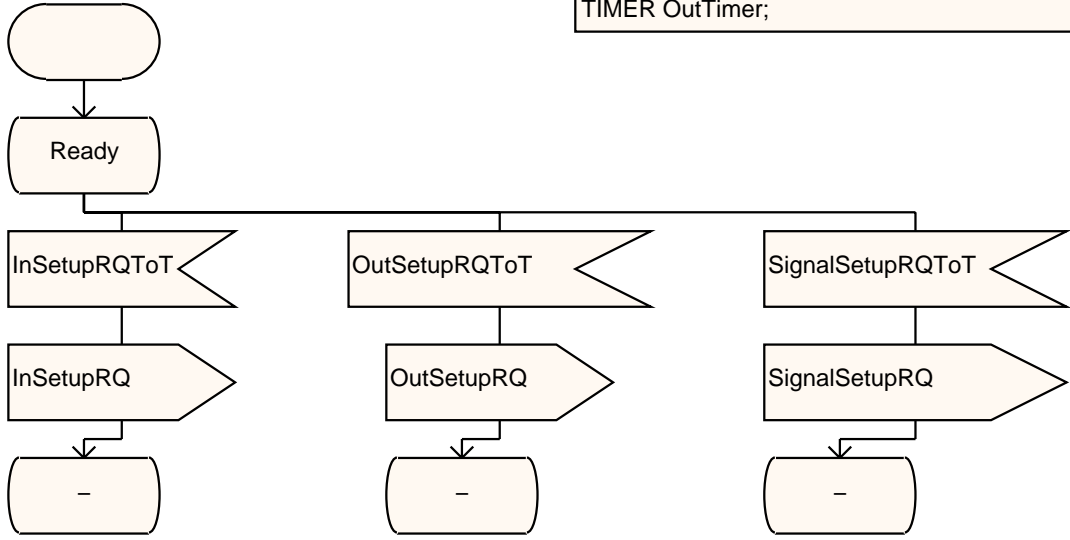


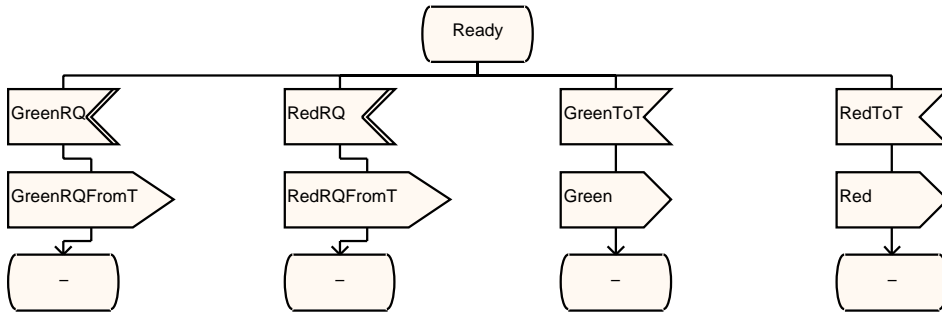
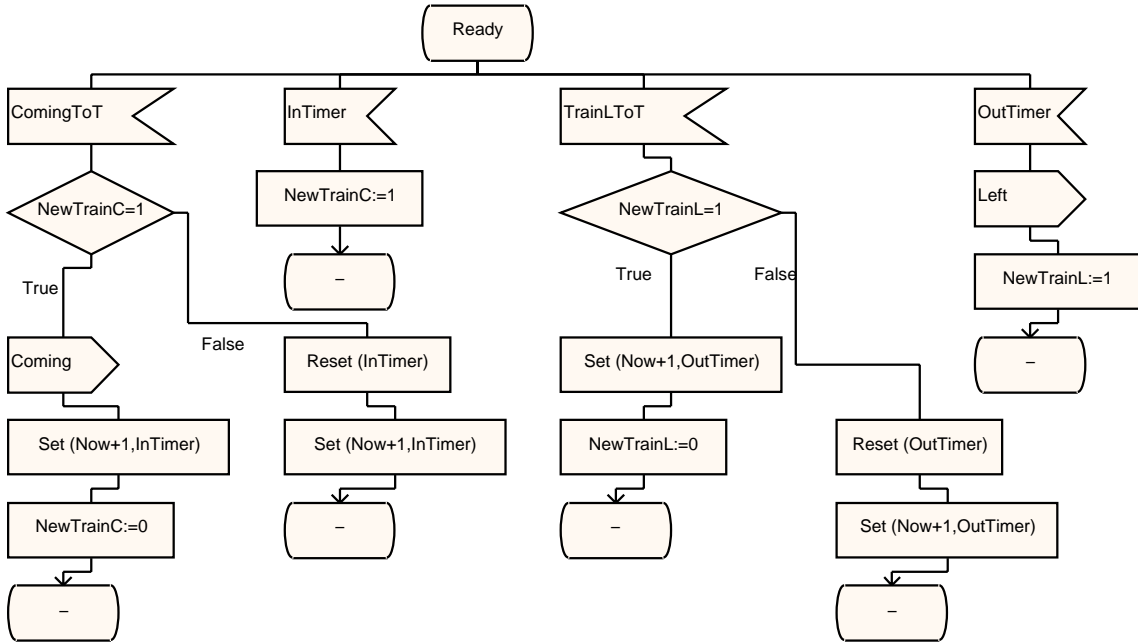






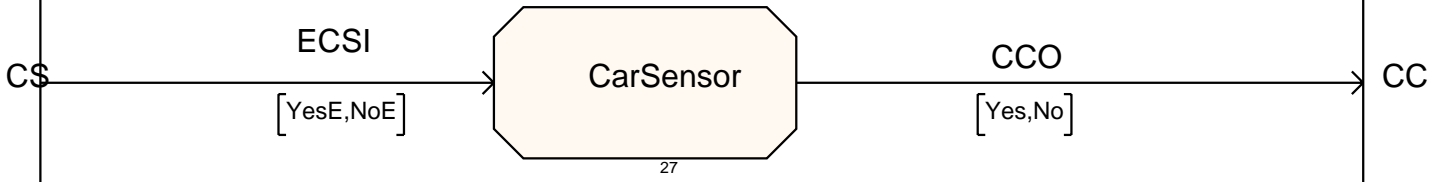
DCL NewTrainC Integer:=1; /\*InSensor detects a new train\*/  
DCL NewTrainL Integer:=1; /\*OutSensor detects a new Train\*/  
  
TIMER InTimer;  
TIMER OutTimer;

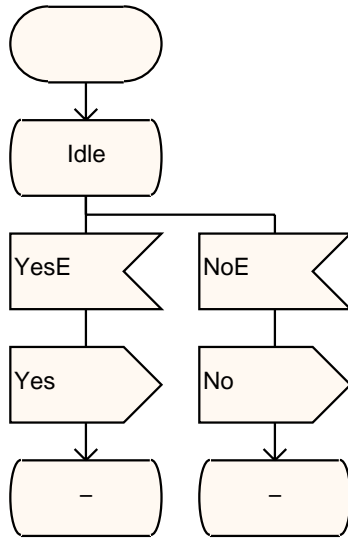


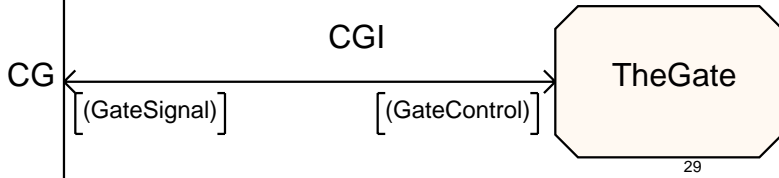


<sup>19</sup> block CarSensor

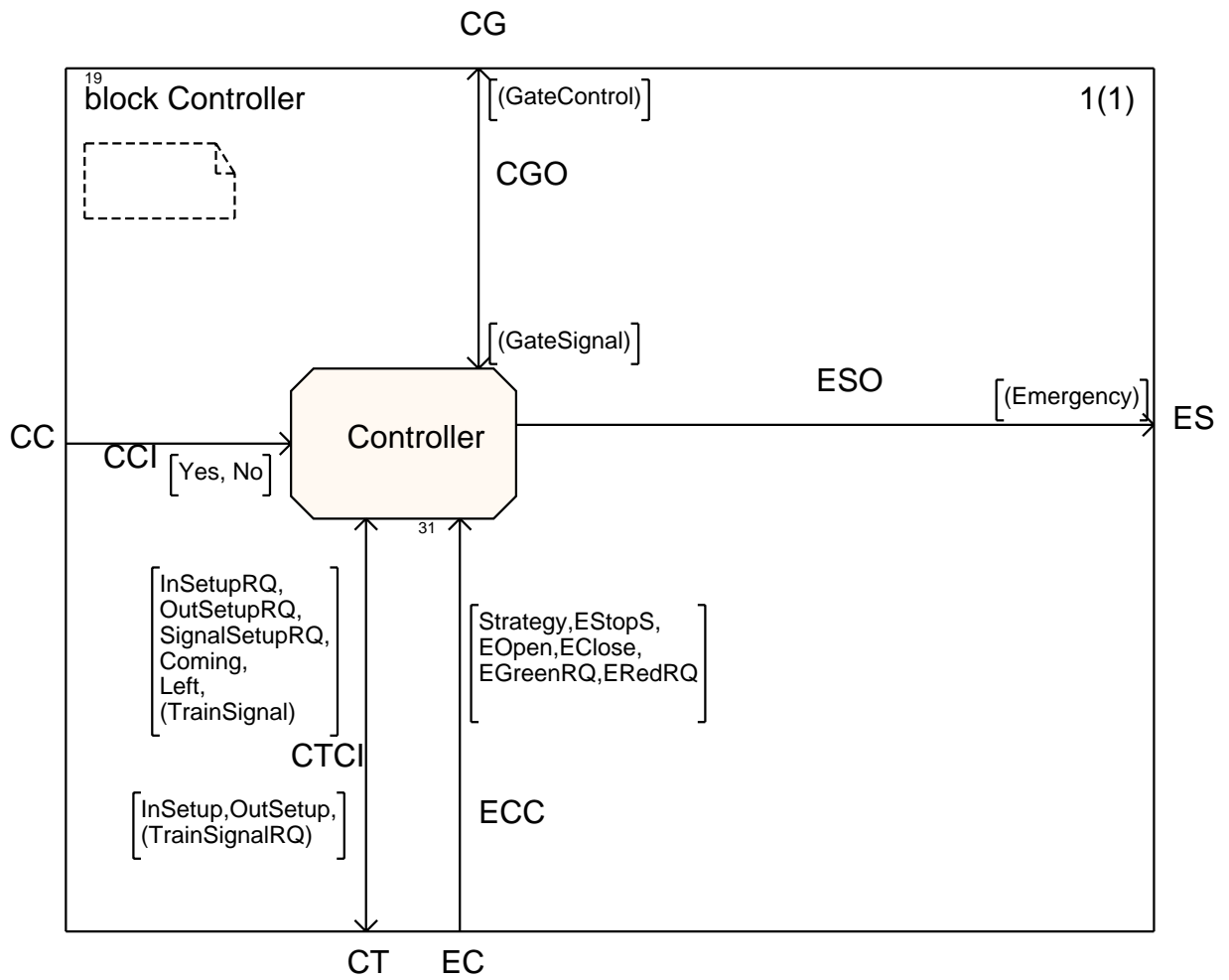
1(1)











## process &lt;&lt;block Controller&gt;&gt; Controller

1(15)

/\* Strategies:

- 1 Manual Override --- any signal can be sent as long as it is safe
- 2 Trains take precedence --- gate is normally closed
- 3 Fast train - many cars - slow train --- gate is normally open
  - Fast train --- close the gate
  - SlowTrain + NoCarWaiting --- close the gate
  - SlowTrain + CarWaiting --- Leave the gate open
- 4 Many cars - train --- gate is normally open
  - Any train comes, check if car Waiting.
  - NoCarWaiting --- close the gate
  - CarWaiting --- Leave the gate open

When gate is open, it is for cars, which means no train is moving in the Crossing area, although some trains may be stopped and waiting. Therefore, checking if "NoOfTrains=0" is not necessary \*/

```
Newtype TrackPid Array(Integer, Pid)
endnewtype;
DCL TrackNo TrackPid; /* Pid of track control process */
```

```
Newtype TrackCount Array(Integer, Integer)
endnewtype;
DCL TrackCounter TrackCount; /* No of trains in a track*/
```

```
Newtype TrackLightType Array(Integer, Character)
endnewtype;
DCL TrackLight TrackLightType; /* Record each track's current light*/
```

```
DCL CarWaiting Integer:=0; /*CarWaiting(1), NoCarWaiting(0)*/
DCL NoOfTrains Integer:=0; /*In the crossing area*/
DCL NoOfSTrains Integer:=0; /* No. of trains stopped */

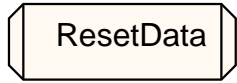
DCL n Integer :=0; /*counter*/
DCL m Integer :=0; /*counter*/
DCL ni Integer :=0; /*counter ---No. of InSensors already setup*/
DCL nO Integer :=0; /*counter ---No. of OutSensors already setup*/
DCL ns Integer :=0; /*counter ---No. of train signals already setup*/
DCL rs Integer :=0; /*counter ---Times of RedRQ sent, but no response*/

DCL SenderNo Integer := 0; /* Sender's track no */
```

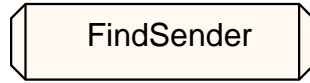
```
TIMER GateClose_TimeOut1;
TIMER GateClose_TimeOut2;
TIMER GateOpen_TimeOut1;
TIMER GateOpen_TimeOut2;
TIMER StopWaiting_TimeOut;
TIMER RedWaiting_TimeOut;
```

```
SYNTYPE sValue =Natural
DEFAULT 3
CONSTANTS 1:4
ENDSYNTYPE sValue;
```

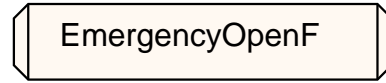
```
DCL Strategies sValue;
DCL S Natural;
```



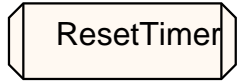
46



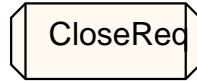
48



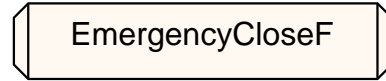
51



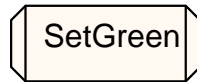
47



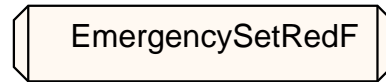
49



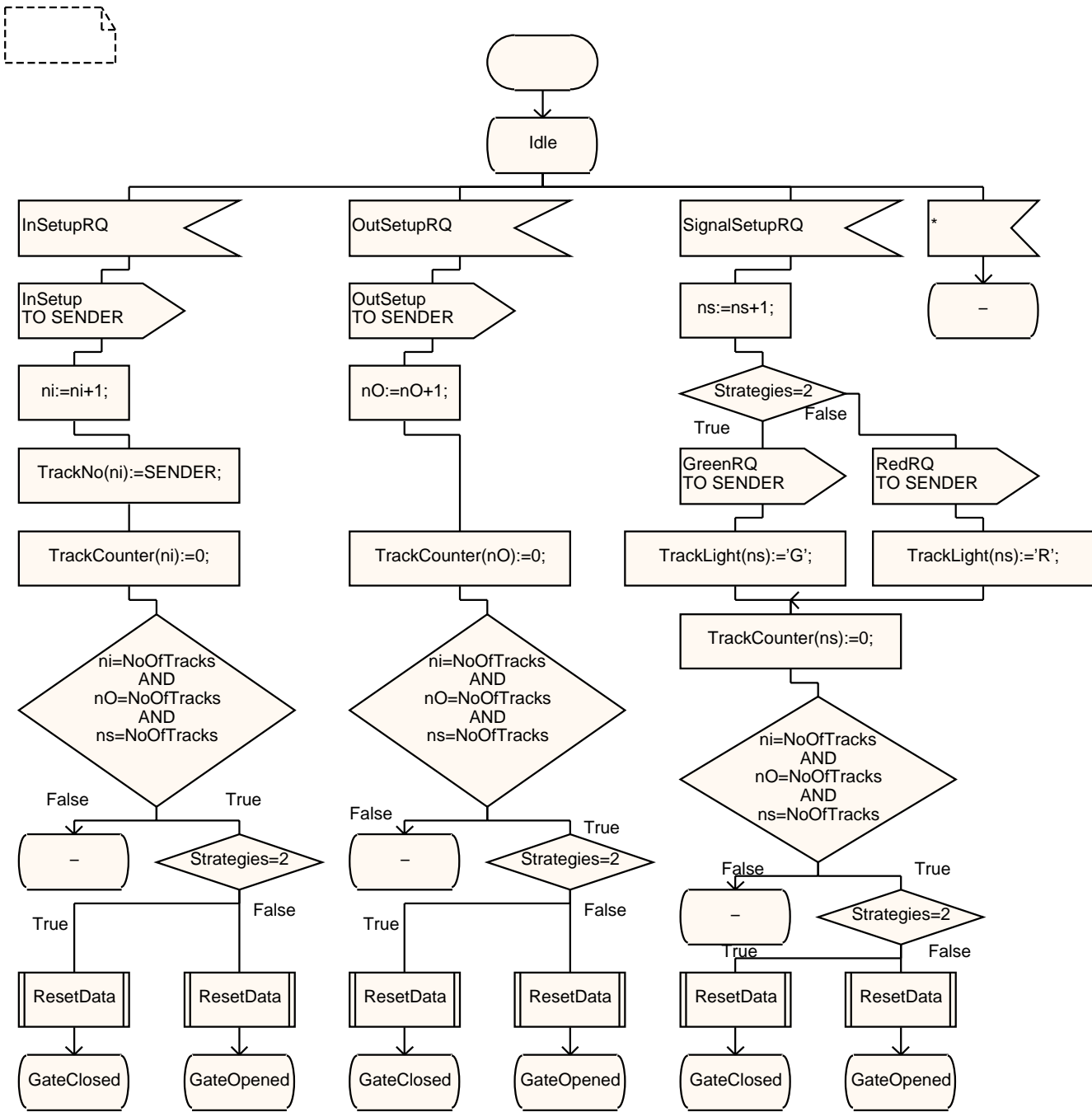
52

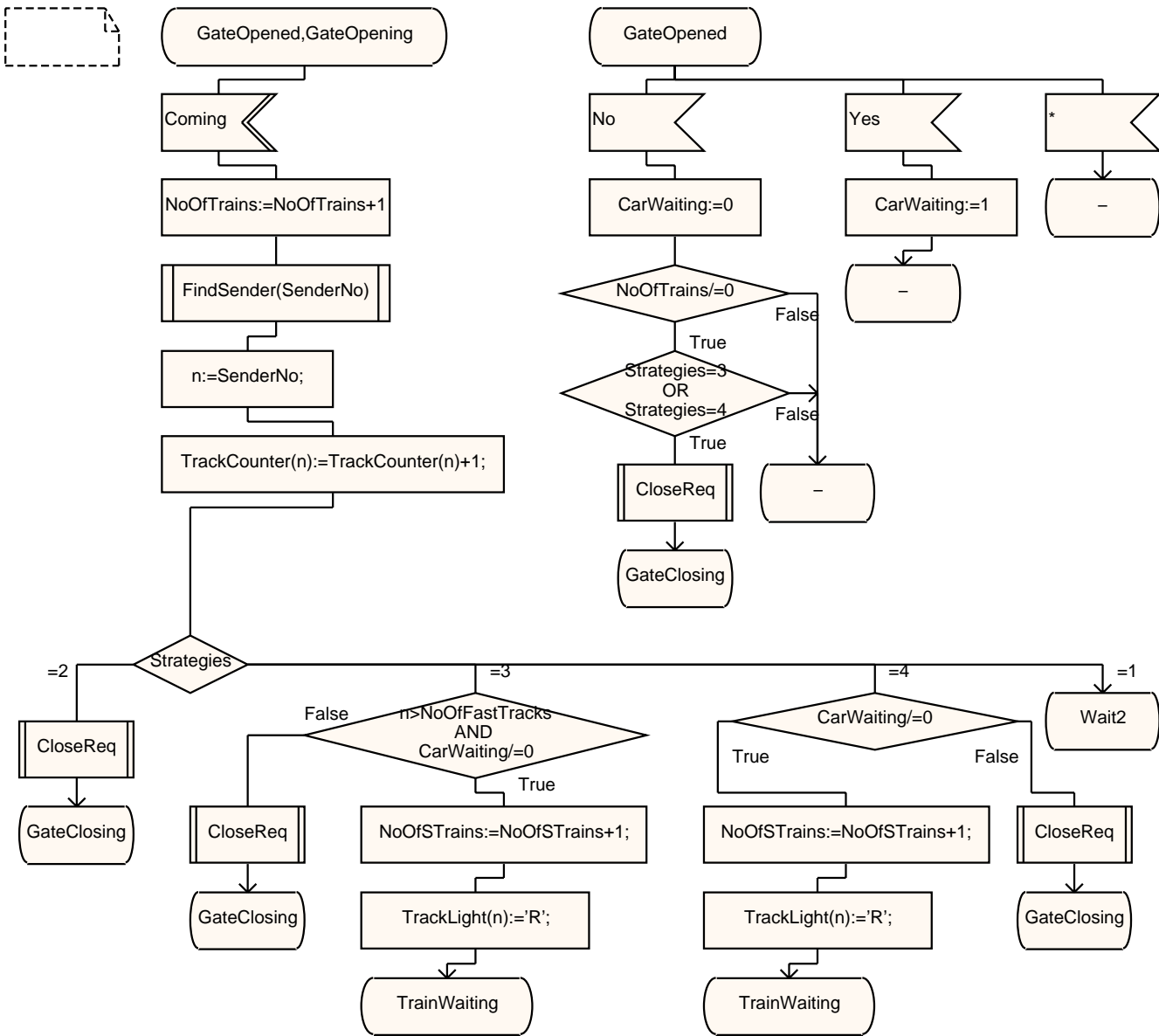


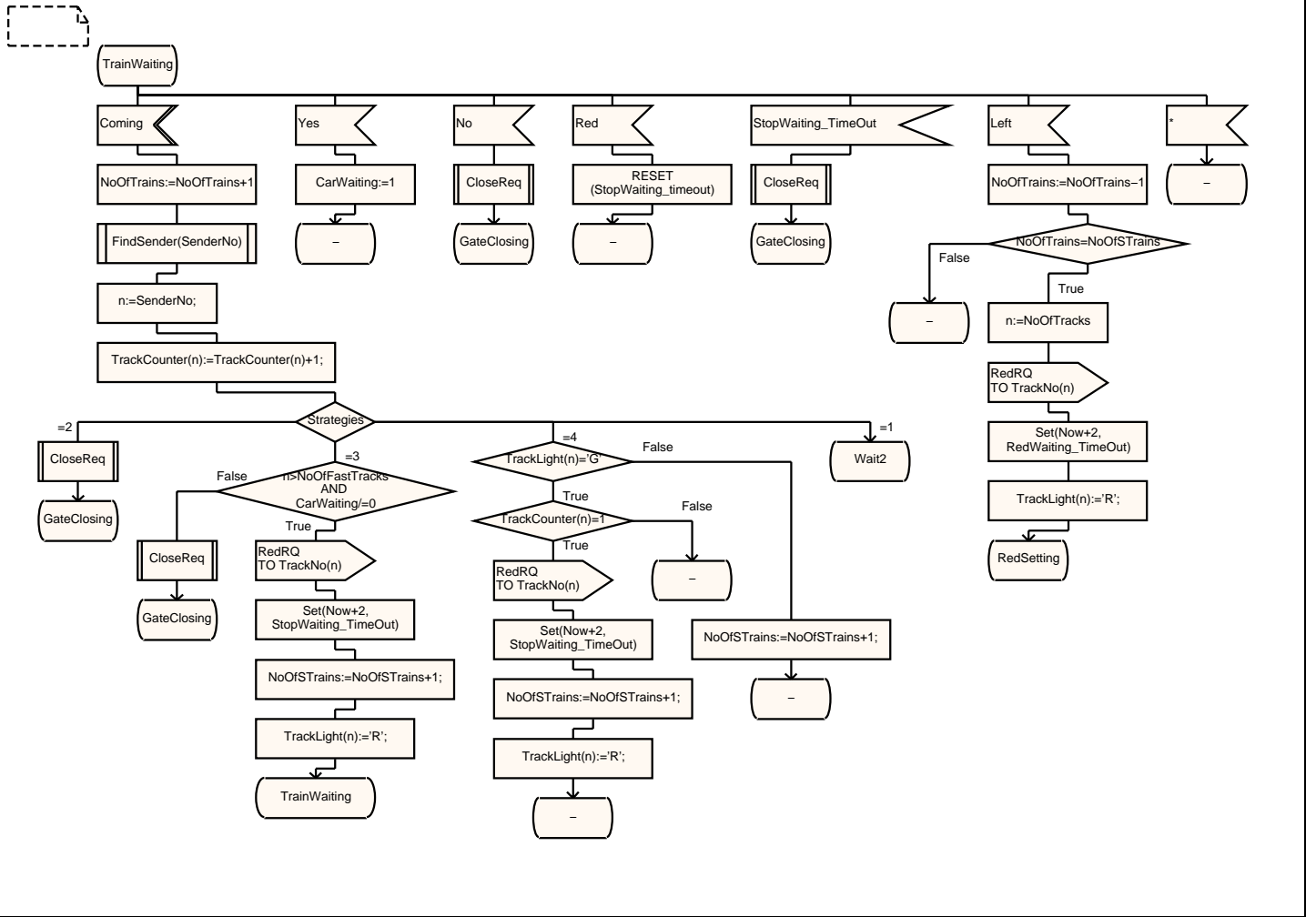
50



53

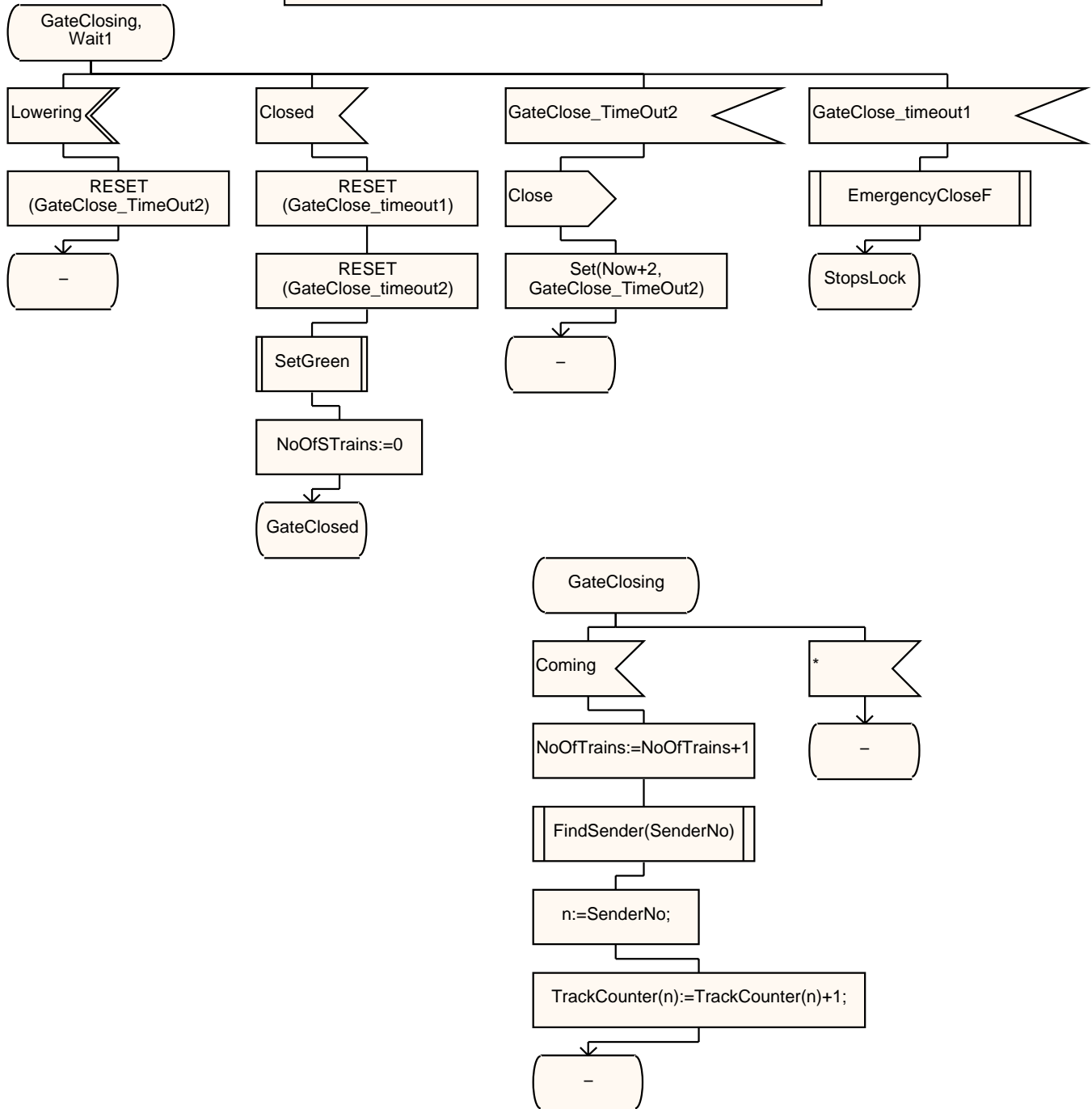




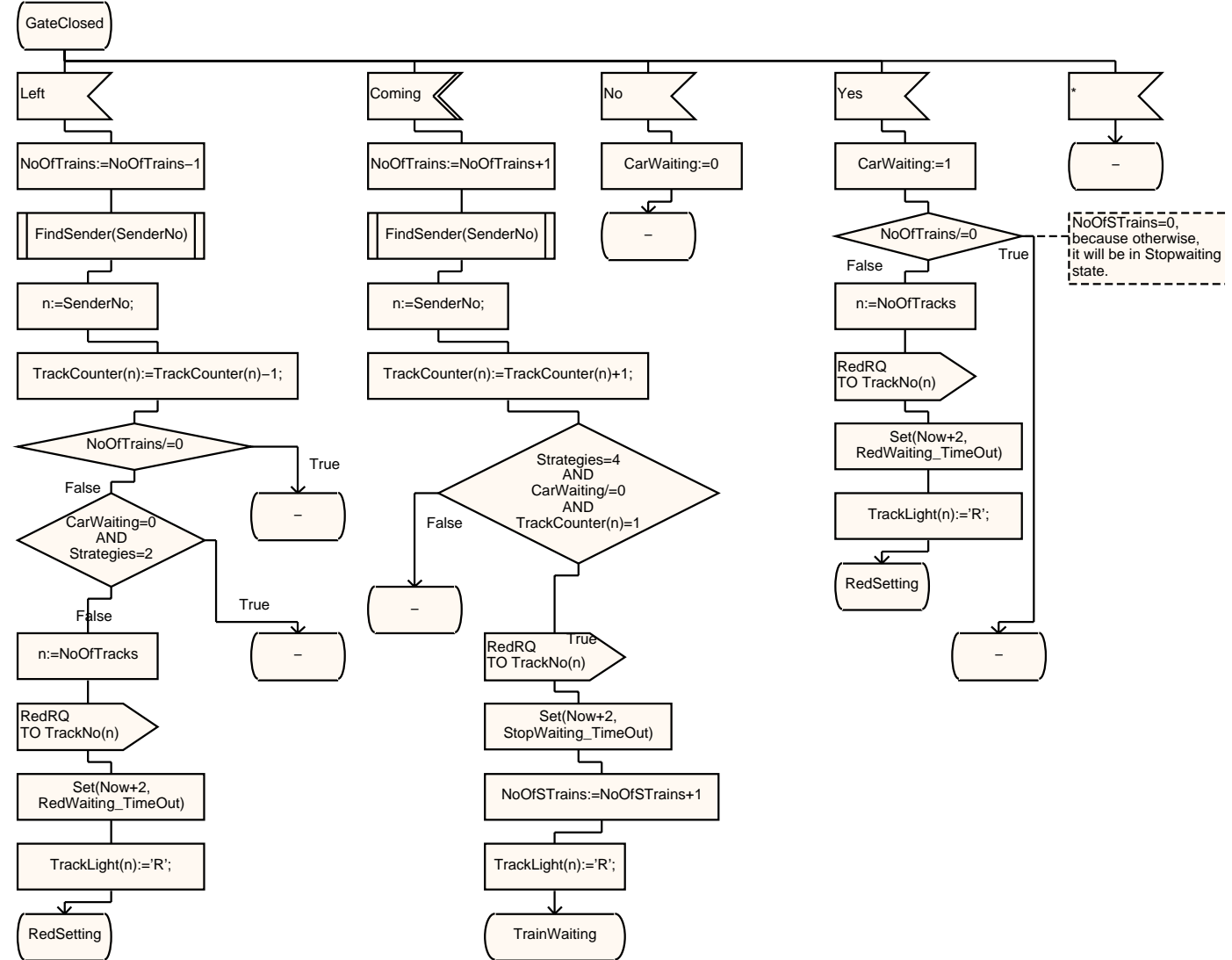


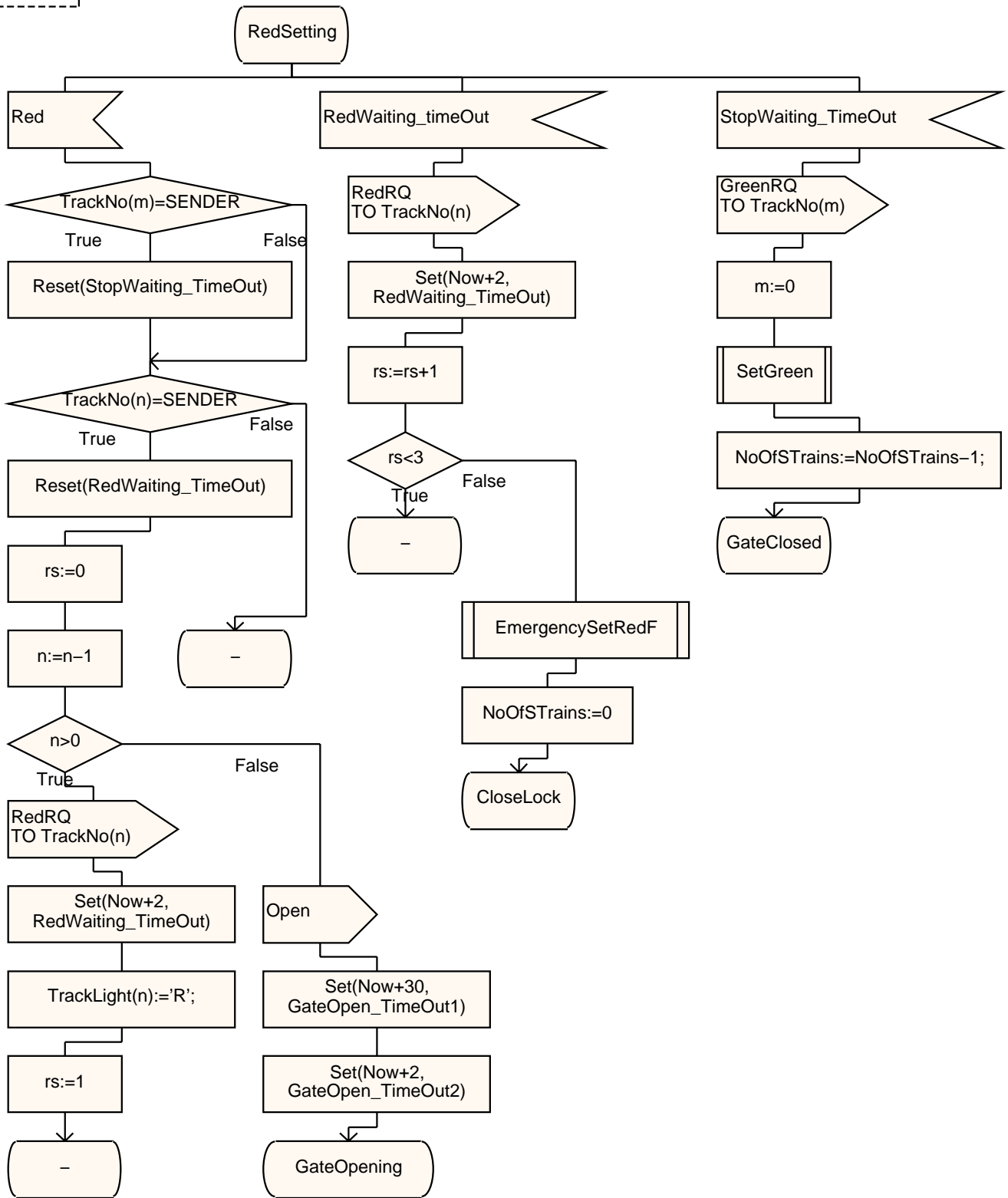


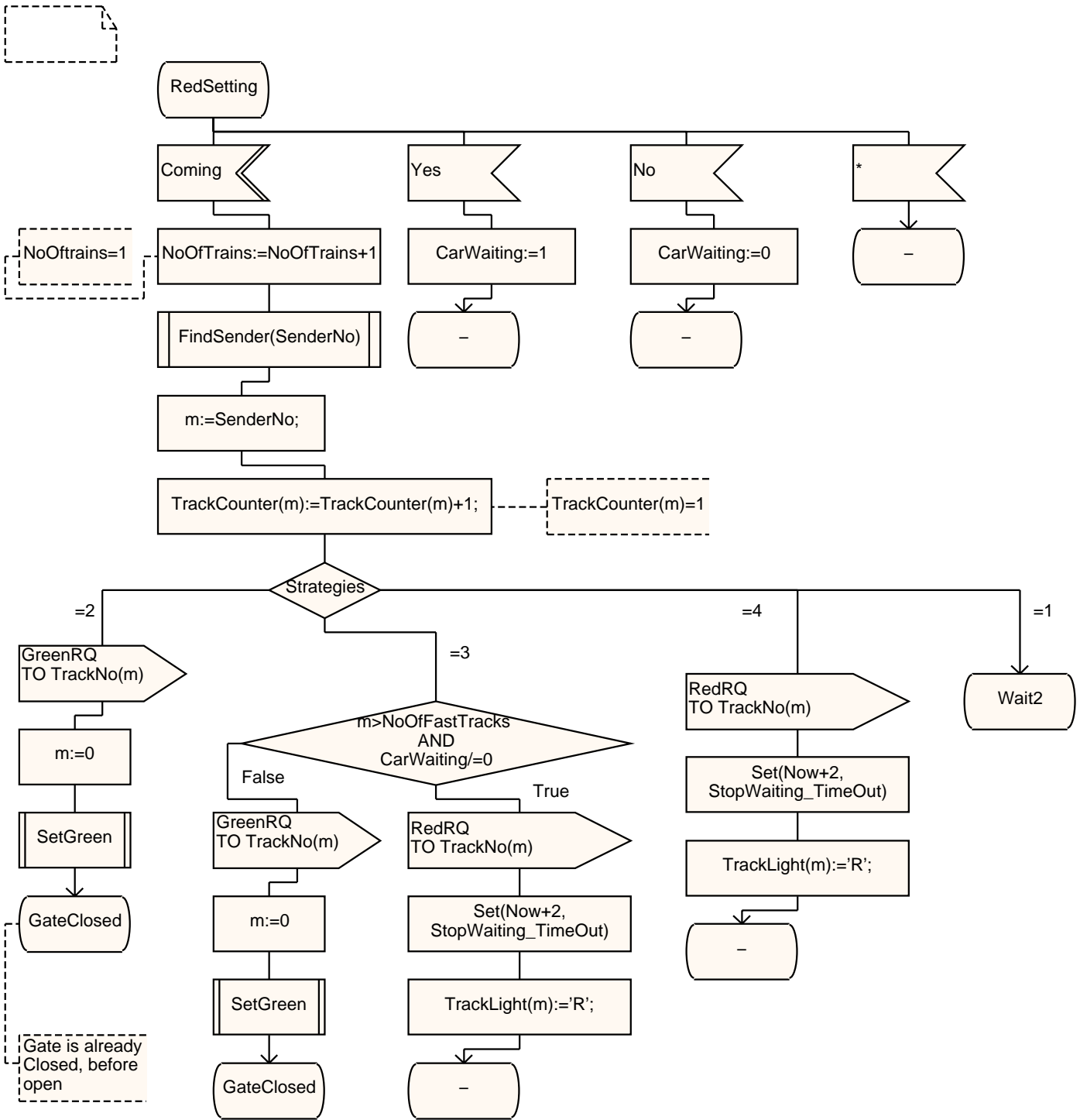
/\* Don't check for Strategy 4, because "close" signal was just sent, which means not too many cars waiting \*/

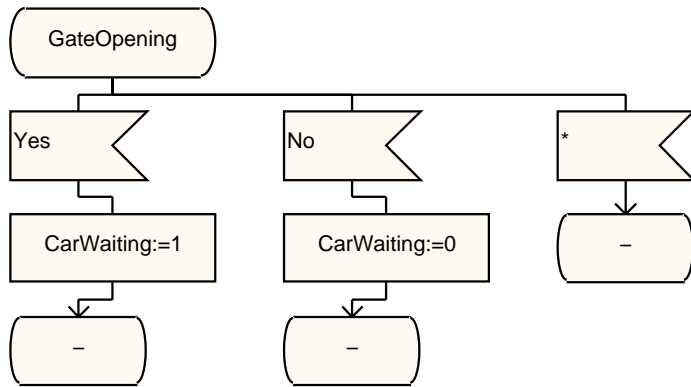
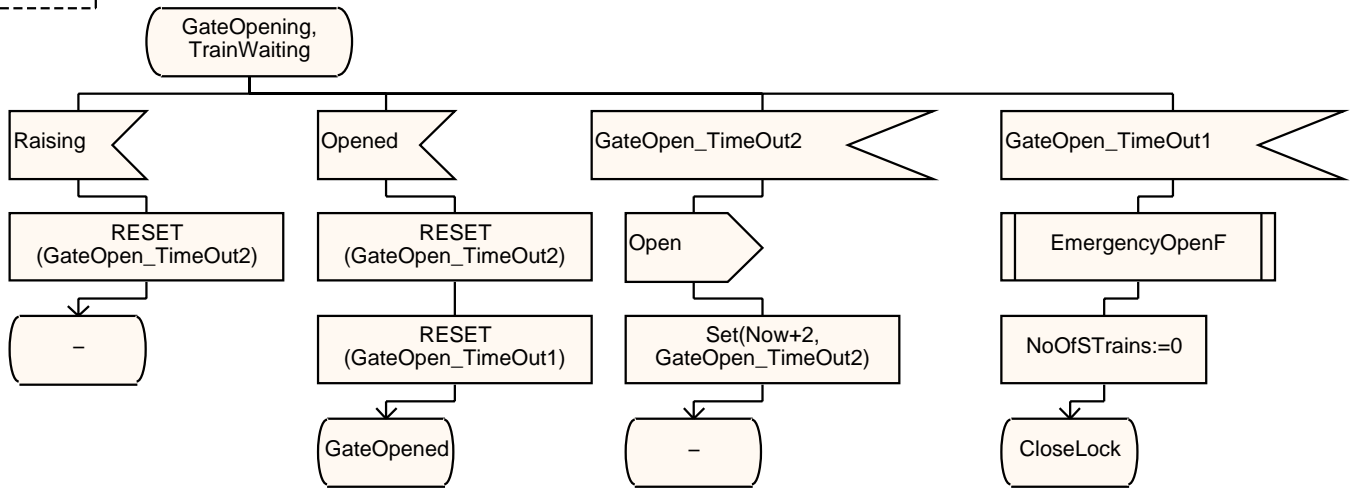


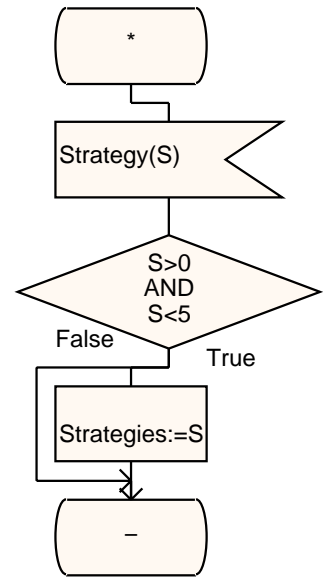
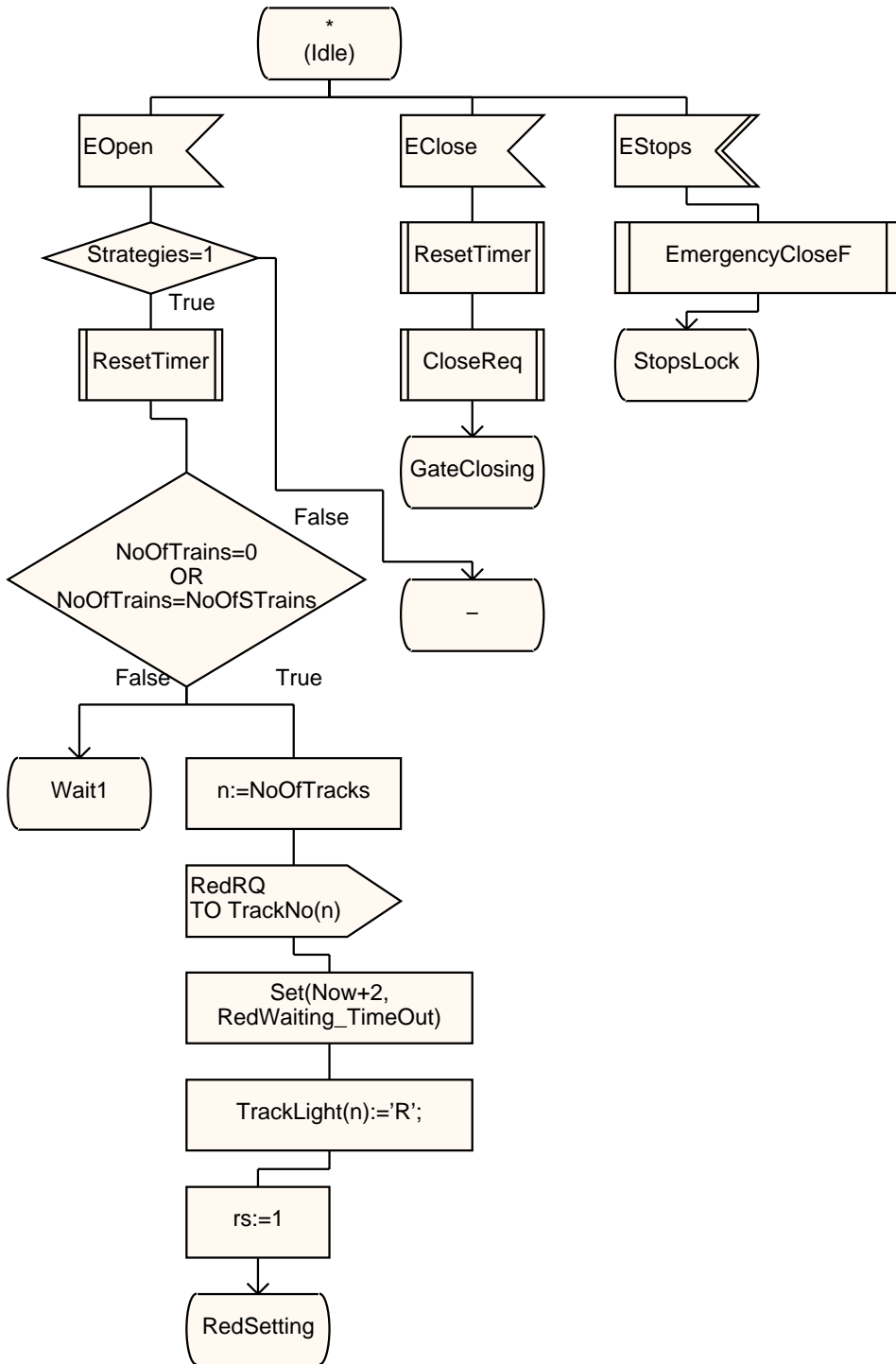
/\* Strategy 4: if carWaiting, and no train running on the same track, stop the train, and try to open the gate  
 Other than this,  
 Always let train go, until no train in gate, then open the gate \*/  
 /\* RedSetting---Send "red" signal one by one, before send "Open" signal \*/

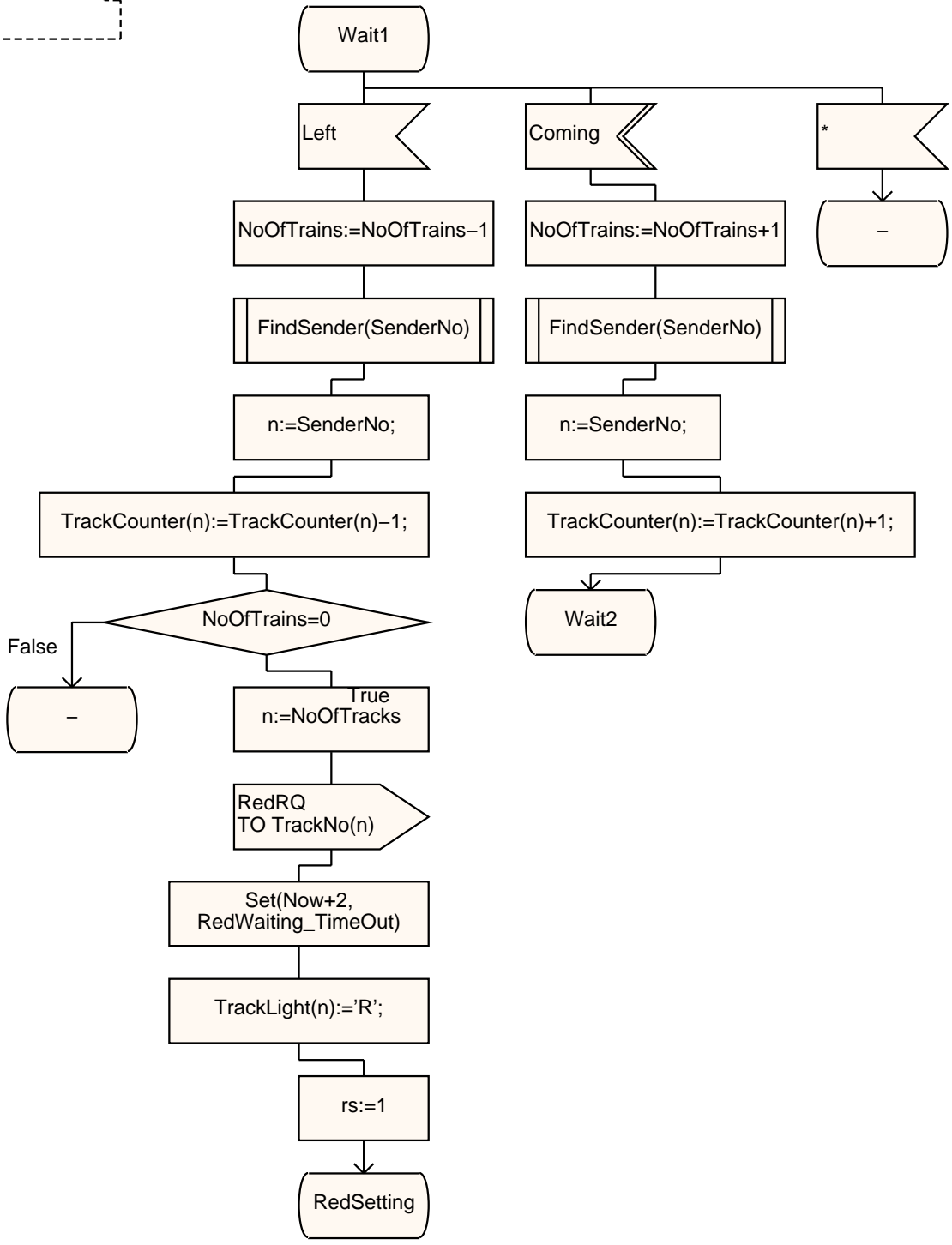


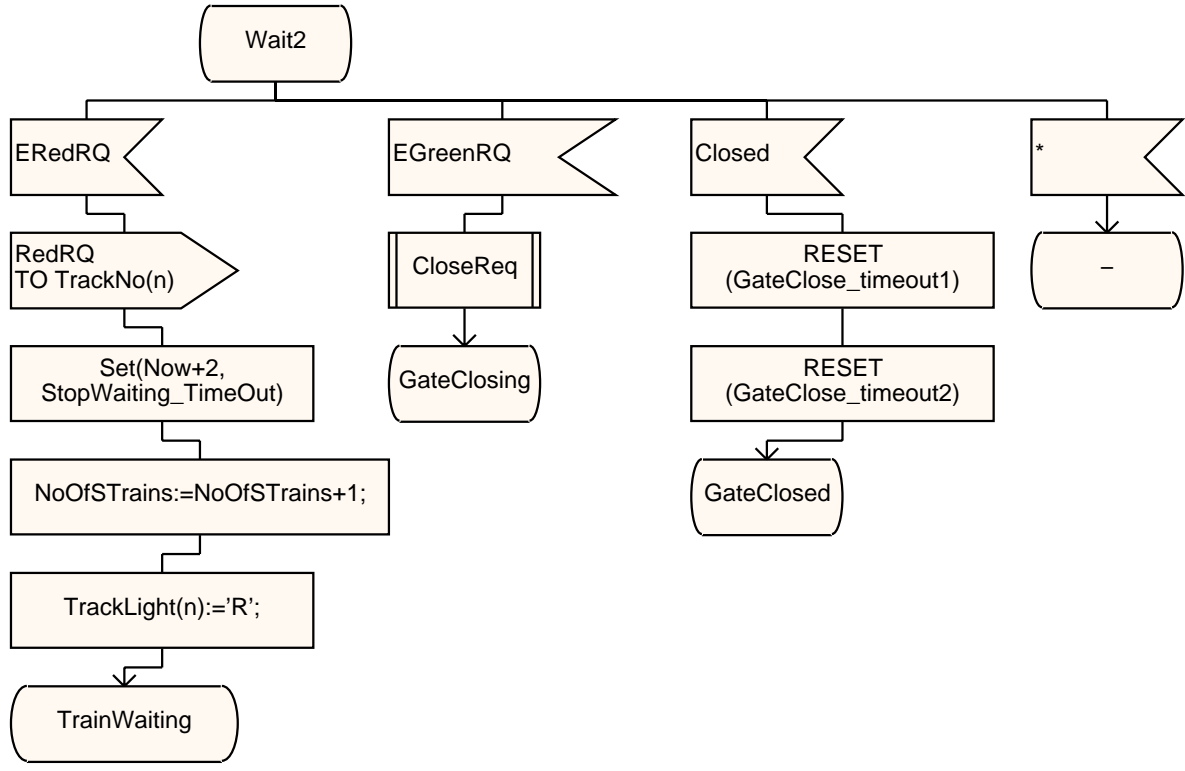


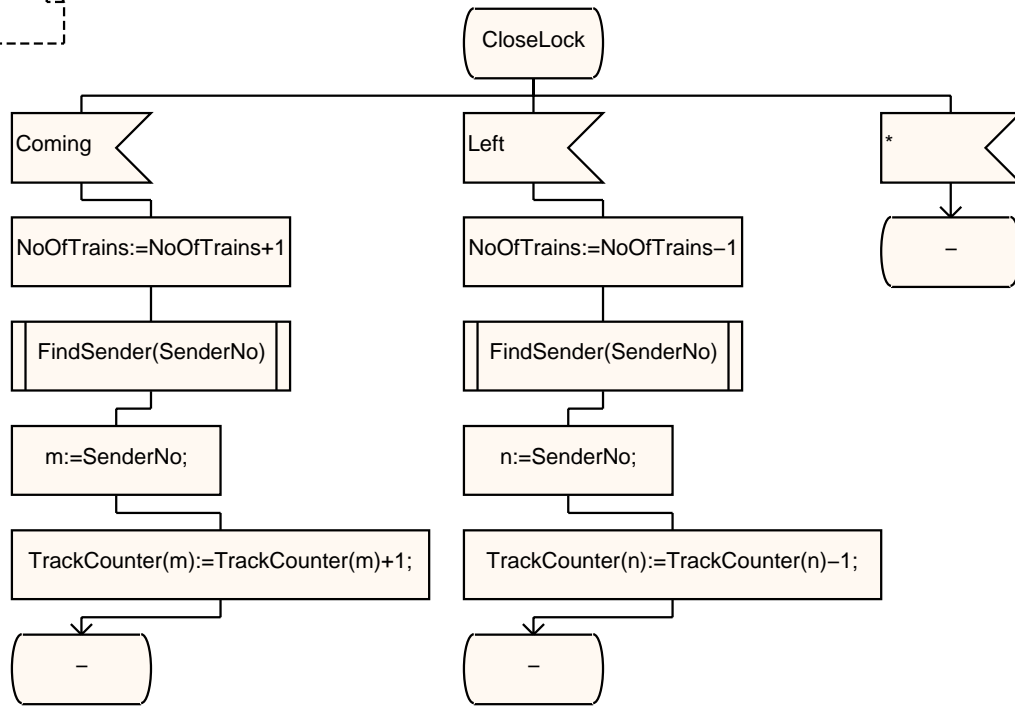


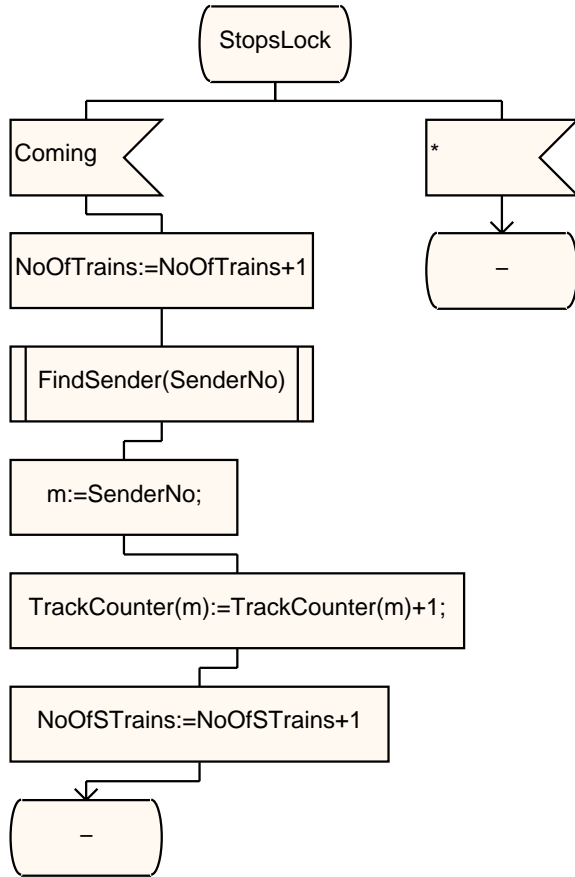


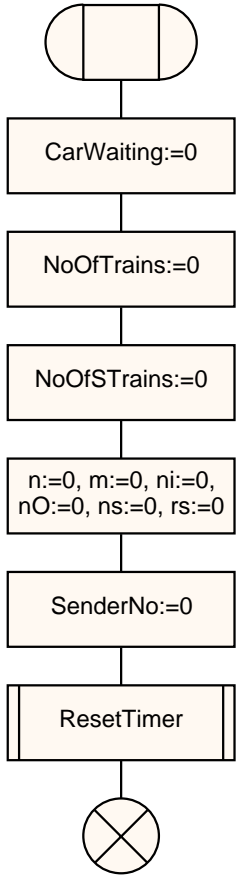


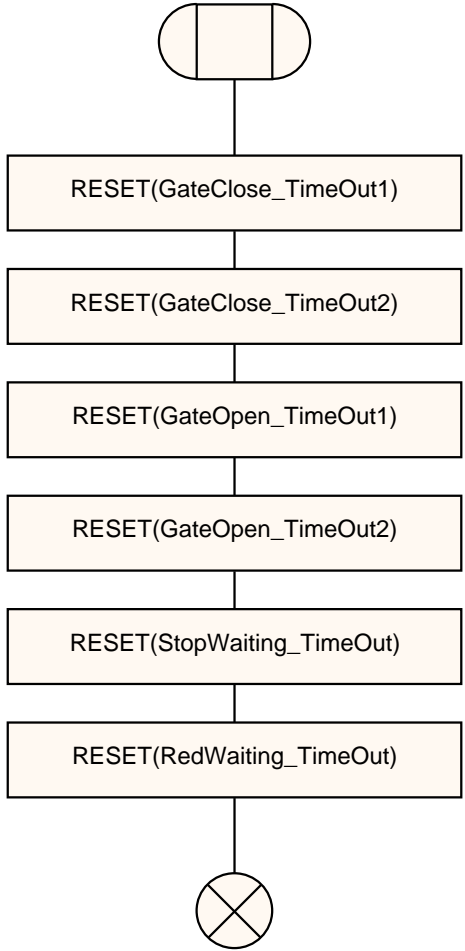






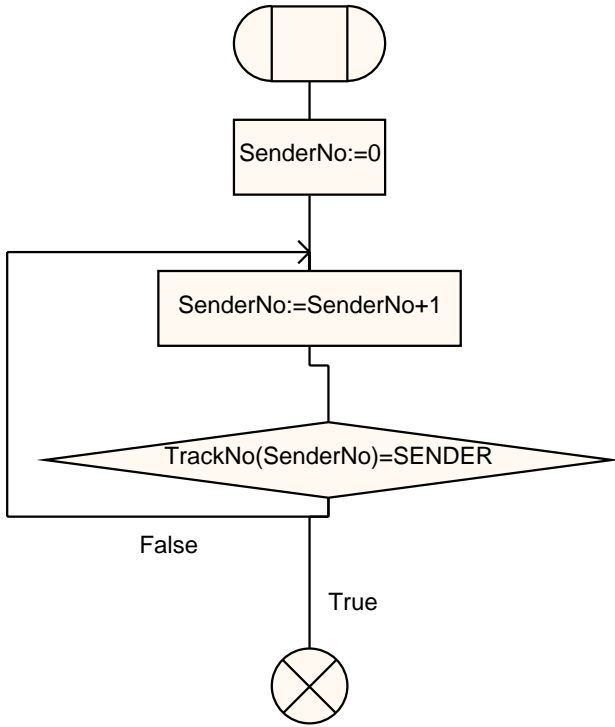


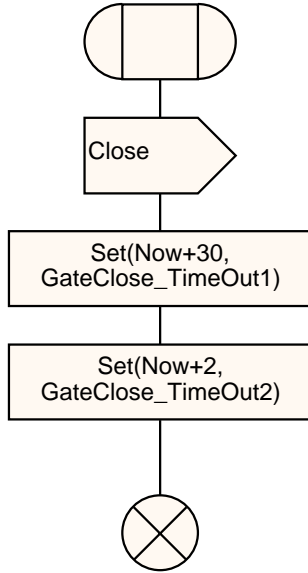


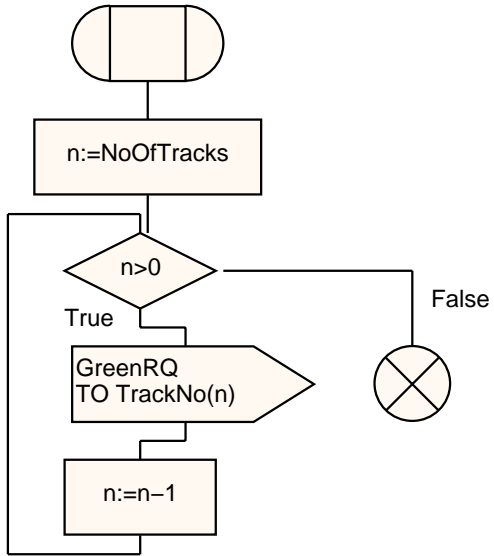


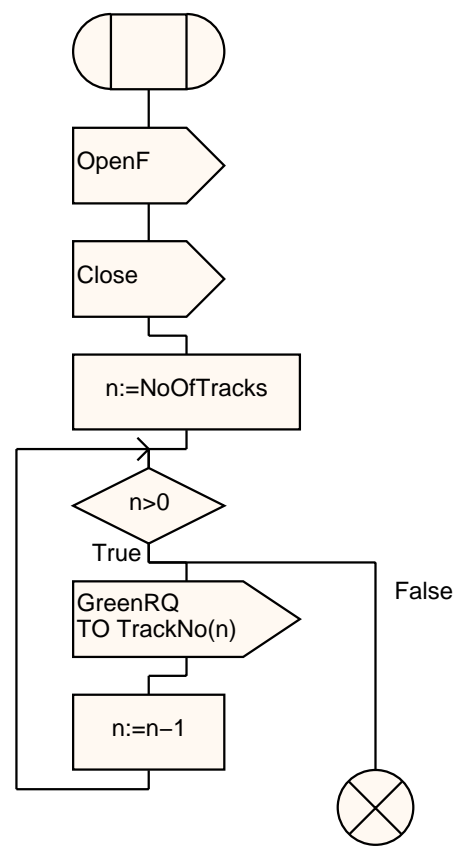
# procedure FindSender

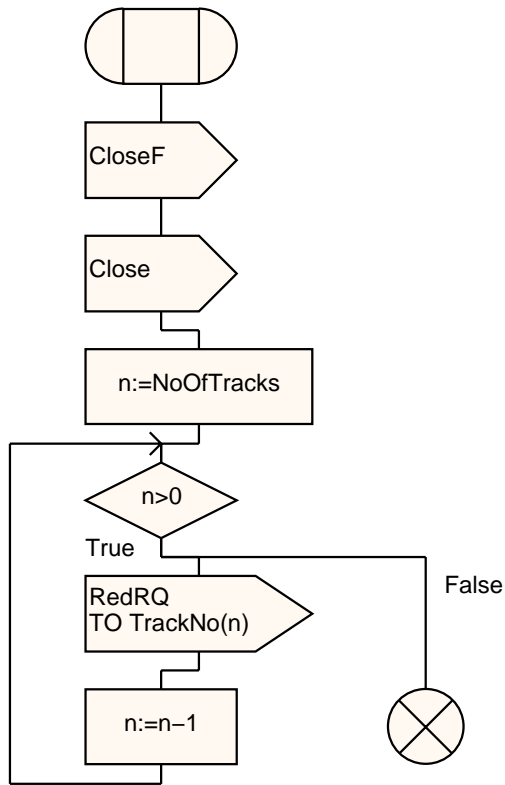
;FPAR  
IN/OUT SenderNo Integer;

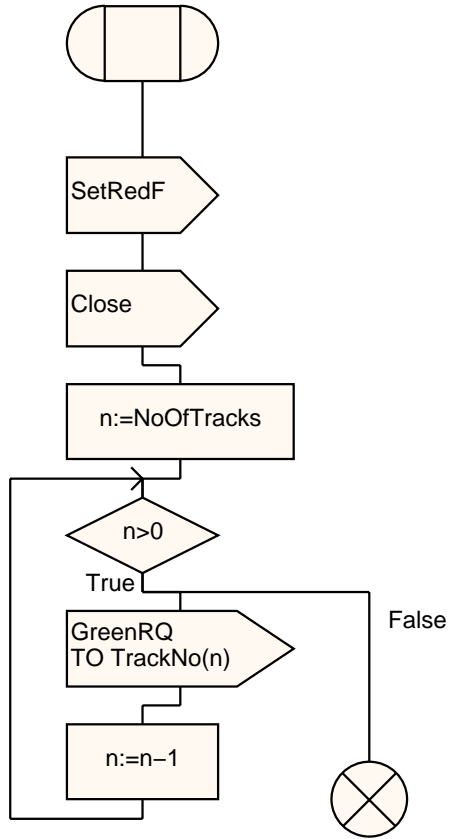






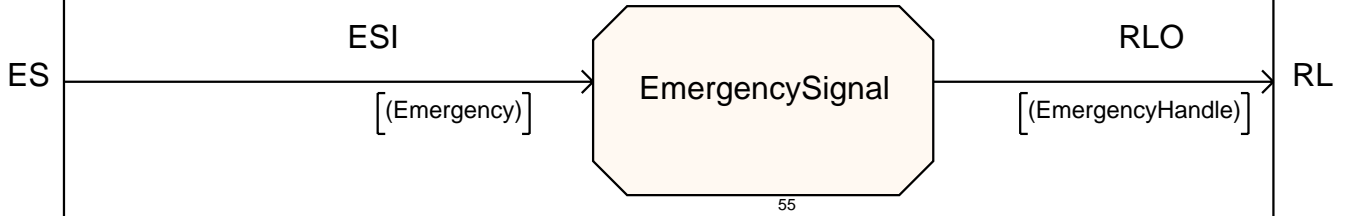


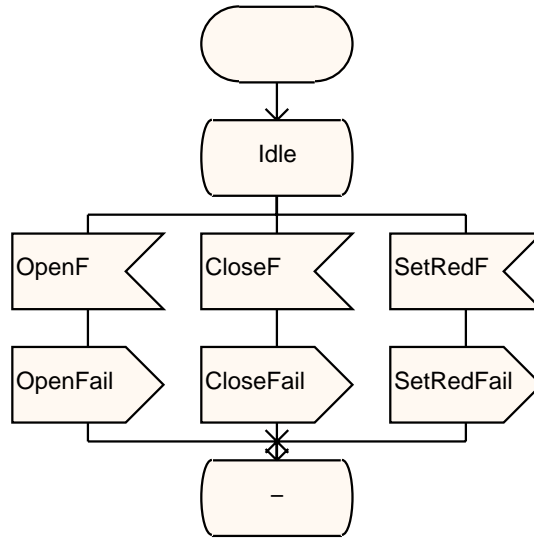




<sup>19</sup> block EmergencySignal

1(1)





# process ModifiedCarSensor

1(1)

