

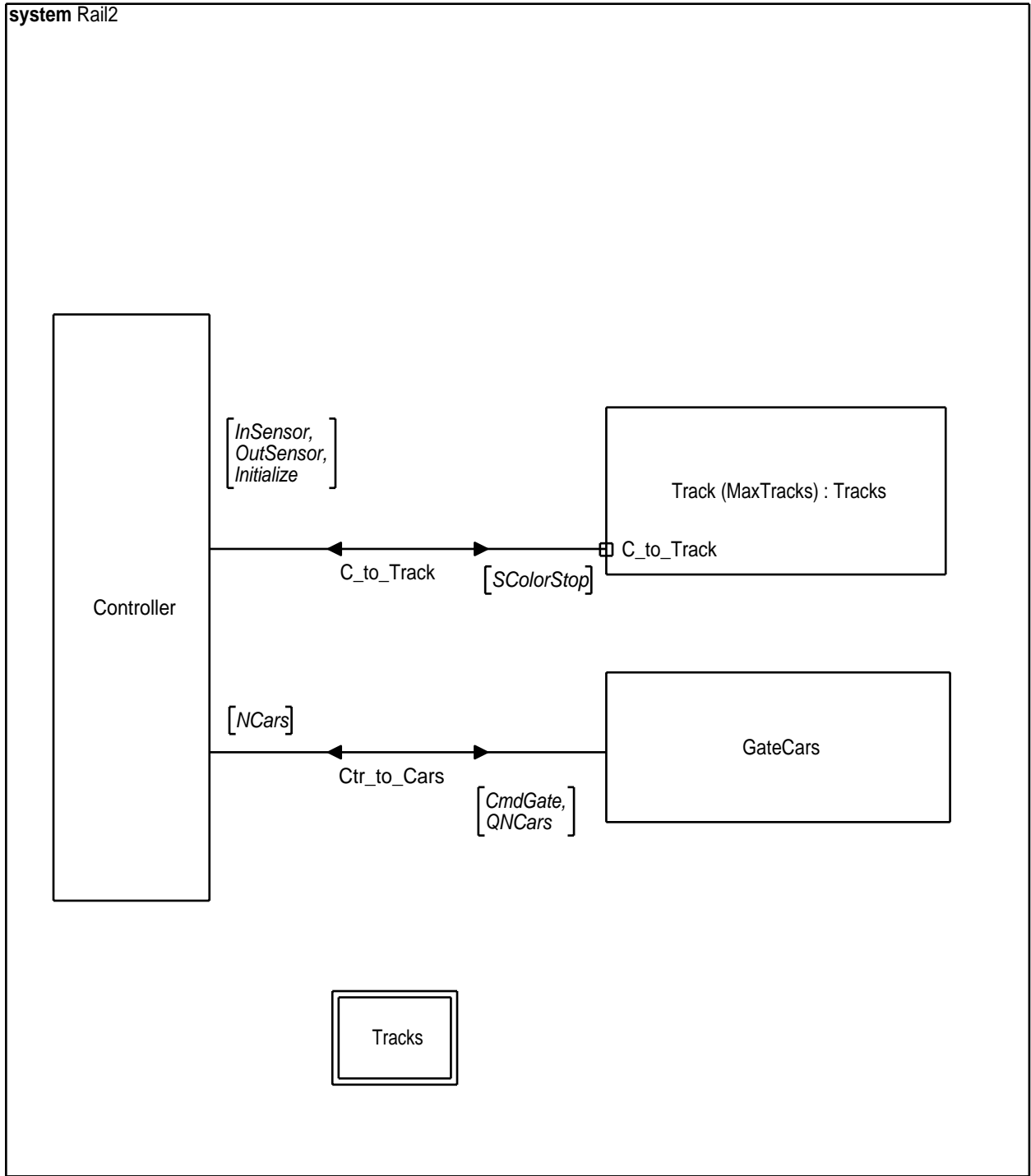
**system Rail2**

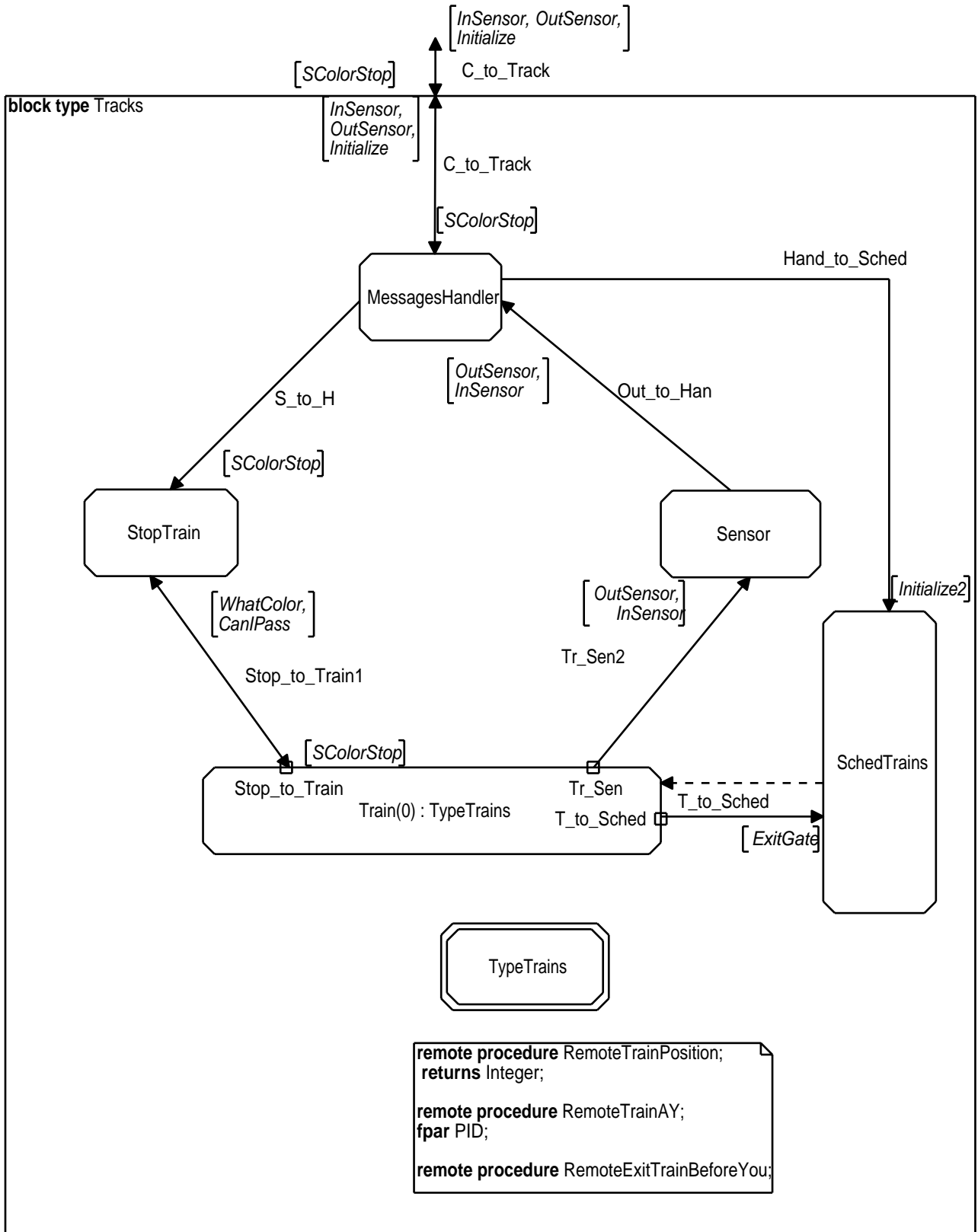
```
/* General type declarations */  
  
NEWTYPE ColorStop  
LITERALS  
  Red, Green, PassOneTrain, DontPassOneTrain, NotDefined;  
OPERATORS ORDERING;  
ENDNEWTYPE ColorStop;  
  
NEWTYPE CGate  
LITERALS  
  CloseGate, OpenGate;  
OPERATORS ORDERING;  
ENDNEWTYPE CGate;  
  
NewType TypeSituation          /* situations to controller */  
LITERALS  
  TrainsPriorityPass, TrainsPriorityGOpenNotPass, TrainsPriorityGCloseNotPass,  
  FewTrainsManyCarsGClosePass, FewTrainsFewCarsGClosePass,  
  FewTrainsFewCarsGCloseNotPass, NoTrainsGClosePass, FewTrainsFewCarsGOpenNotPass,  
  FewTrainsManyCarsGOpenNotPass, FewTrainsManyCarsGCloseNotPass, NoTrainsGCloseNotPass,  
  NoTrainsGOpenNotPass, FewTrainsZeroCarsGOpenNotPass,  
  FewTrainsVeryFewCarsGClosePass  
OPERATORS ORDERING;  
ENDNEWTYPE TypeSituation;  
  
NEWTYPE TSpeed  
LITERALS  
  slow, fast;  
OPERATORS ORDERING;  
ENDNEWTYPE TSpeed;  
  
SYNTYPE TypeTrackNumber = Integer  
CONSTANTS 0:4          /* Max Number of tracks */  
ENDSYNTYPE TypeTrackNumber;  
  
SYNTYPE SpeedRank = Integer  
CONSTANTS 0:4  
ENDSYNTYPE SpeedRank;  
  
/* Signals declaration */  
  
SIGNAL InSensor (TSpeed), OutSensor (TSpeed);  
SIGNAL Ncars(Integer), QNcars;  
SIGNAL SColorStop(ColorStop), WhatColor;  
SIGNAL CmdGate(CGate), Initialize2;  
SIGNAL Initialize, ExitGate, GateStatus;  
SIGNAL CanIPass;
```

**system Rail2**

```
/* parameters of the system */
```

```
SYNONYM MaxTracks =1;  
SYNONYM MaxTrains = 2; /* max trains for track */  
SYNONYM DistSecurity = 3;  
SYNONYM short =1;  
SYNONYM large = 2;  
SYNONYM t_limite = 60 ;  
SYNONYM DistanceSlow = 14, DistanceFast = 15;  
SYNONYM SpeedSlow = 2, SpeedFast = 4;  
SYNONYM CarsArrival = 10, CarsDepart = 10;  
SYNONYM CarsBorneSup = 30, TrainsWaitingBorneSup = 4;  
SYNONYM TGeneration=2;
```





**process type** TypeTrains

**FPAR** TrainSpeed TSpeed, BeforeTrain PID, LongTrain Integer

**DCL** distance Integer;  
**DCL** speed SpeedRank;  
**DCL** Pass ColorStop;  
**DCL** MaxSpeed SpeedRank;  
**DCL** Evaluation Integer;  
**DCL** AfterTrain PID;

**synonym** t1 = 1;

**synonym** Accelerate = 1;

**synonym** Slowdown = 2;

**synonym** Continue = 3, ICantStop = 4;

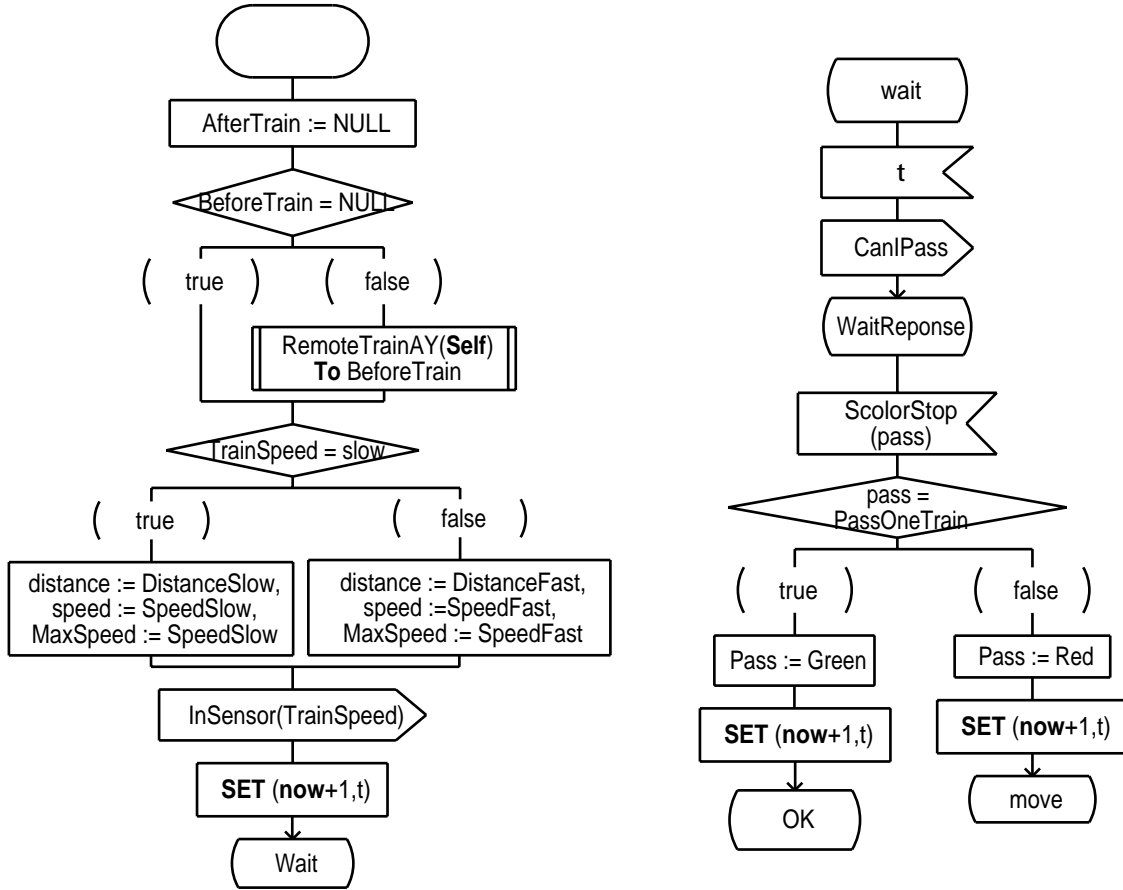
**timer** t;

**imported procedure** RemoteTrainPosition;  
**returns** Integer;

**imported procedure** RemoteExitTrainBeforeYou;

**imported procedure** RemoteTrainAY;  
**FPAR** PID;

process type TypeTrains  
FPAR TrainSpeed TSpeed, BeforeTrain PID, LongTrain Integer



ActualizeSpeed

EvaluateSpeed

exported as  
RemoteTrainAY  
TrainAfterYou

exported as  
RemoteExitTrainBeforeYou  
TrainBeforeYouExit

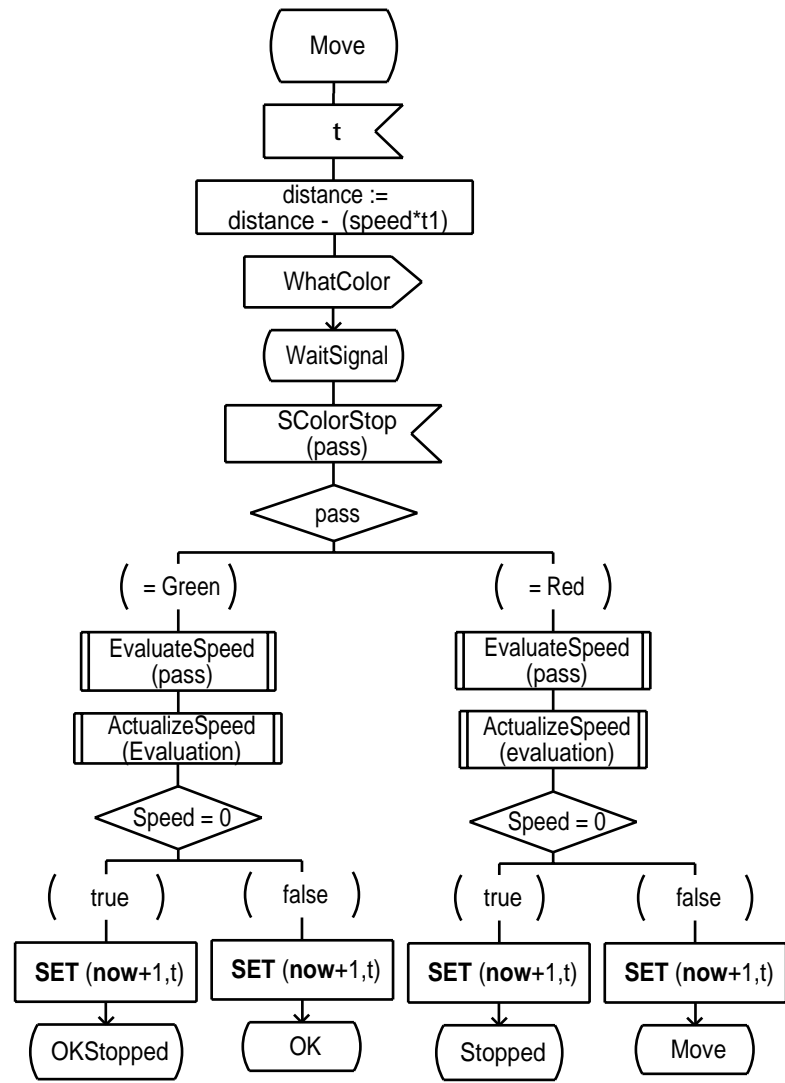
exported as  
RemoteTrainPosition  
TrainPosition

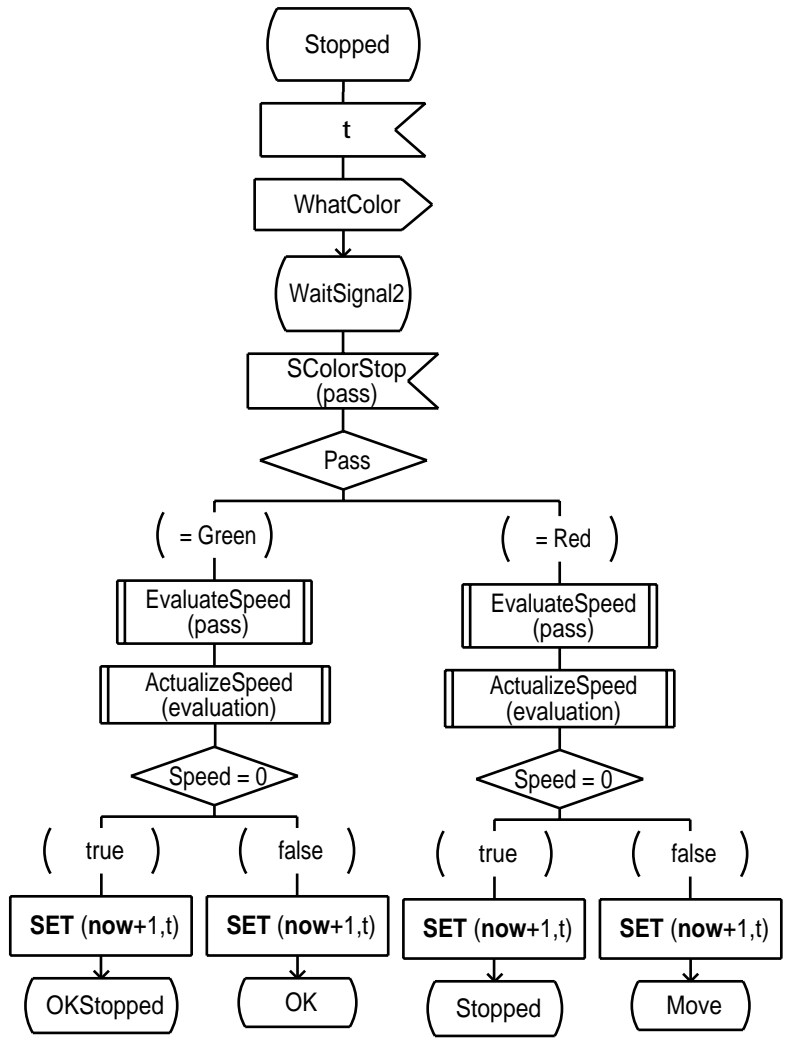
Stop\_to\_train [SColorStop]  
[WhatColor, CanIPass]

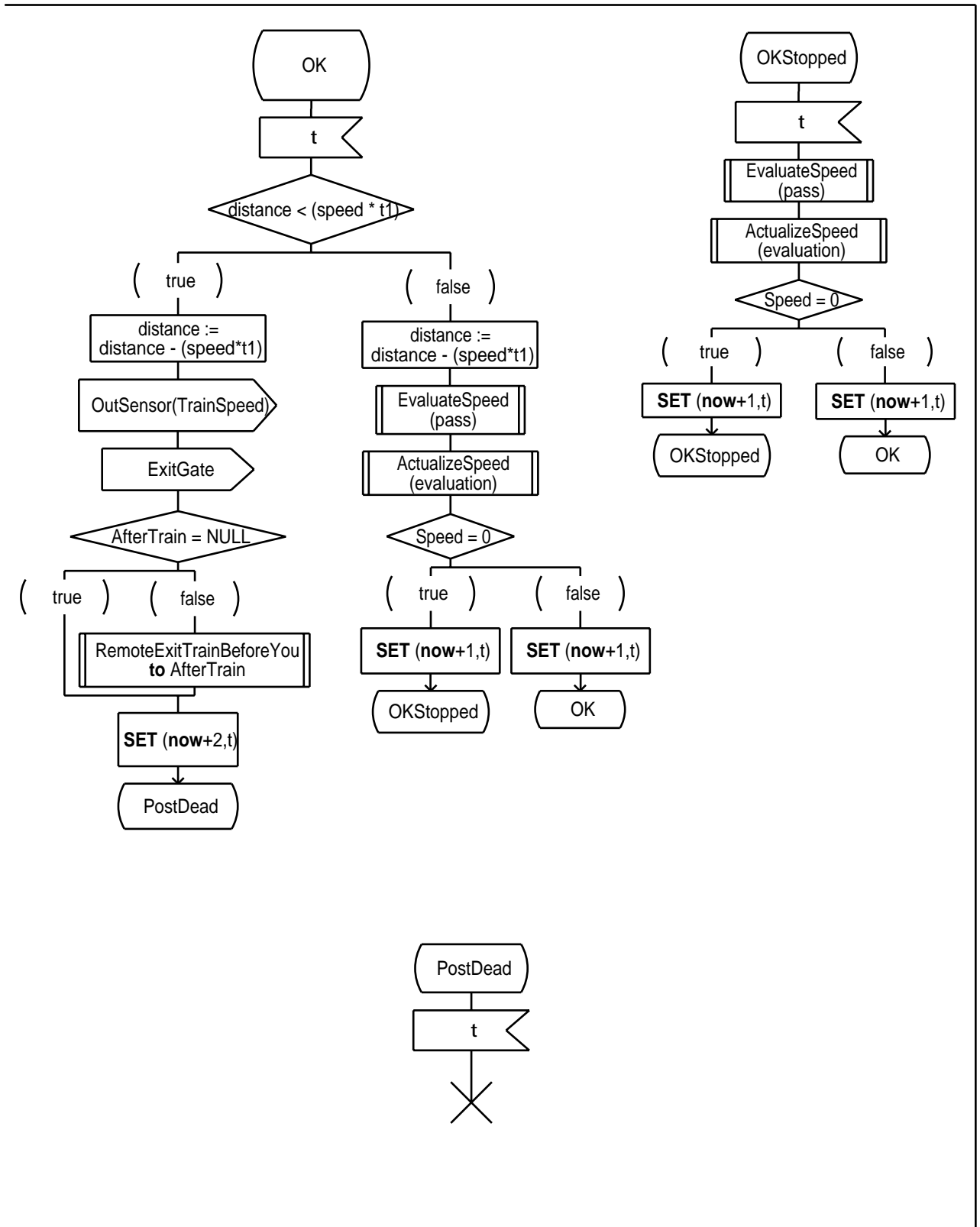
Tr\_Sen [OutSensor, InSensor]

T\_to\_Sched [ExitGate]

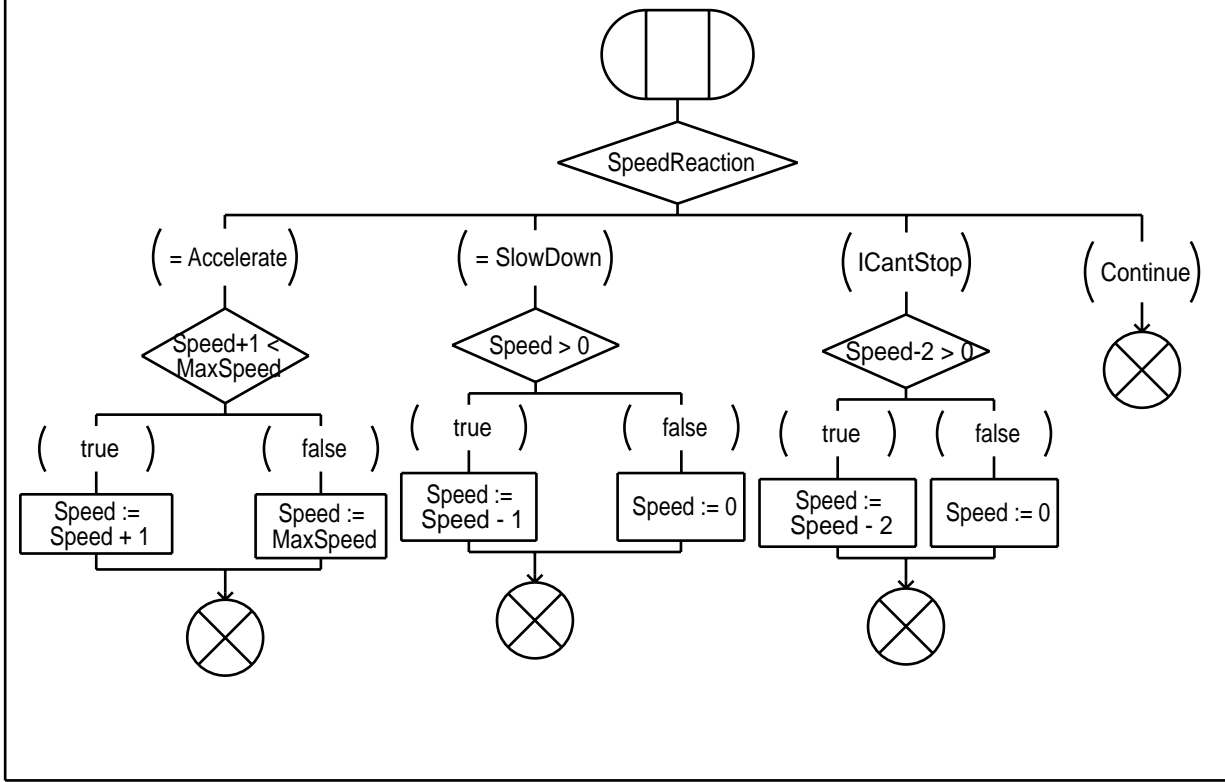




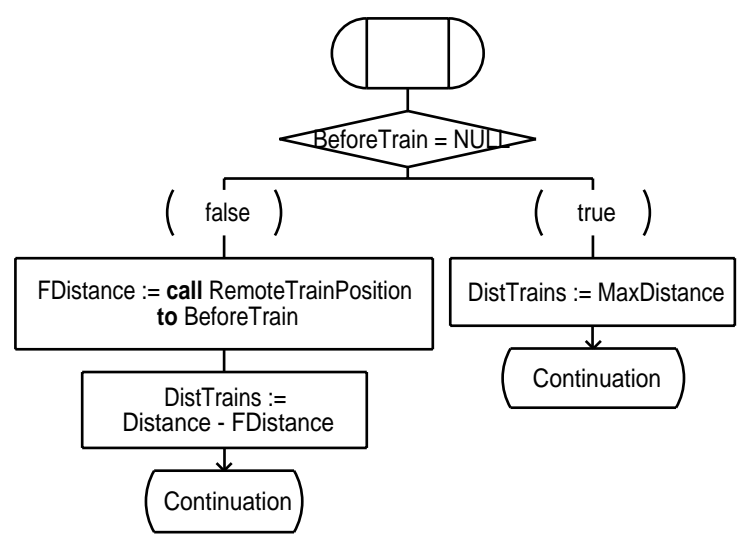




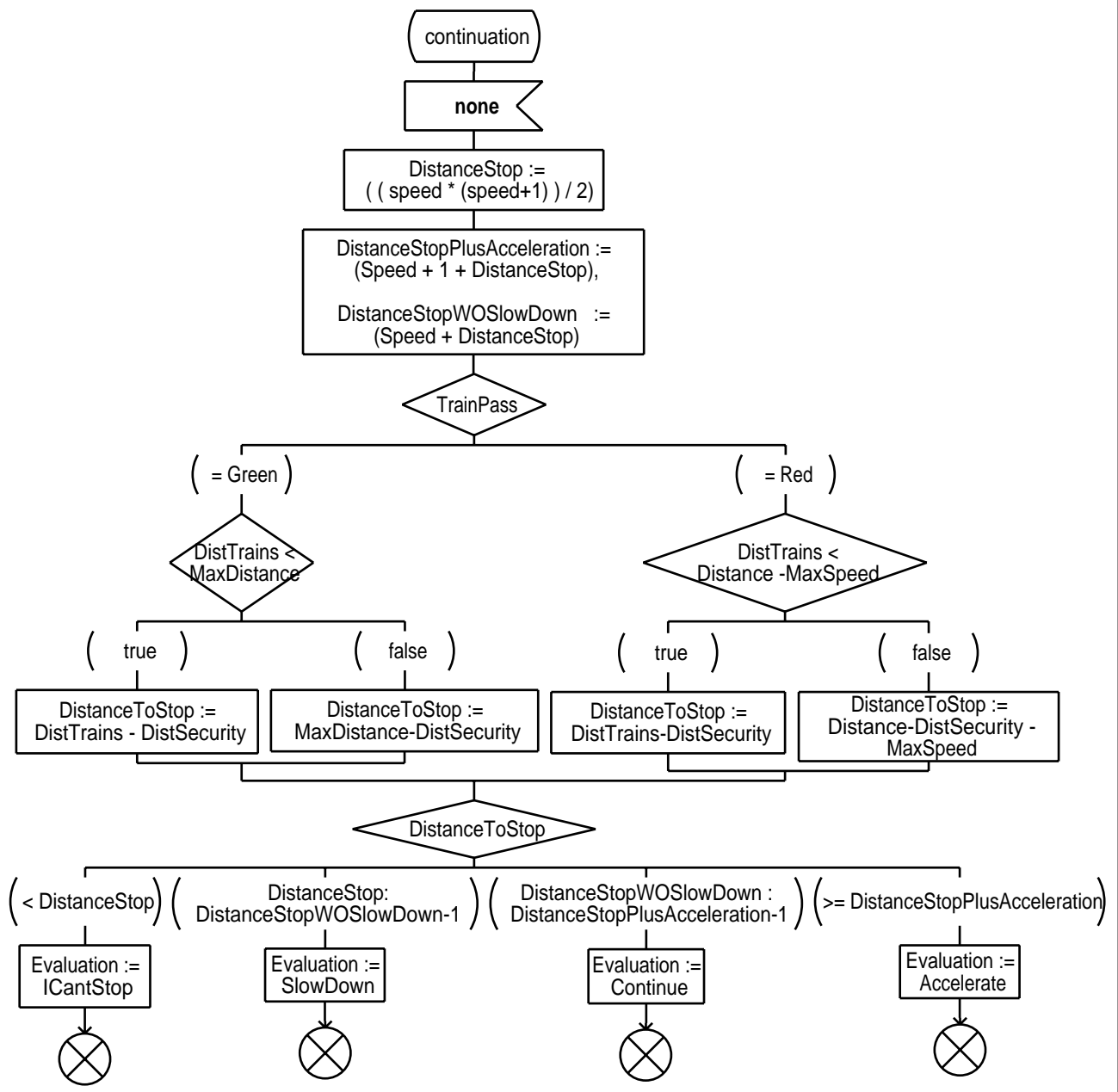
**procedure** ActualizeSpeed  
**FPAR** SpeedReaction Integer



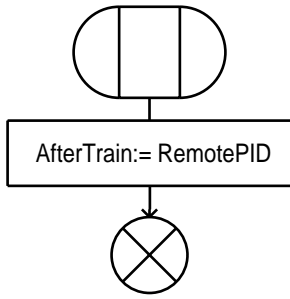
**procedure** EvaluateSpeed  
**FPAR** TrainPass ColorStop



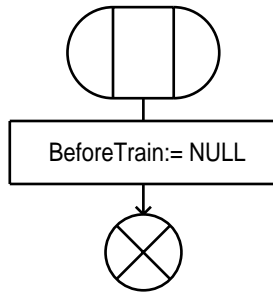
**DCL** DistanceStop, DistanceStopPlusAcceleration Integer;  
**DCL** DistanceStopWOSlowDown Integer;  
**DCL** DistanceToStop Integer;  
**DCL** DistTrains, FDistance Integer;  
**SYNONYM** MaxDistance = 100;



**exported as**  
RemoteTrainAY  
**procedure** TrainAfterYou  
**fpar** RemotePID PID

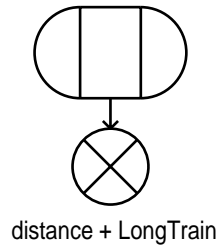


**exported as**  
RemoteExitTrainBeforeYou  
**procedure** TrainBeforeYouExit

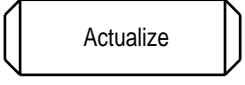
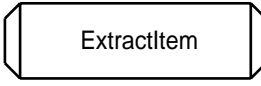
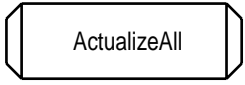
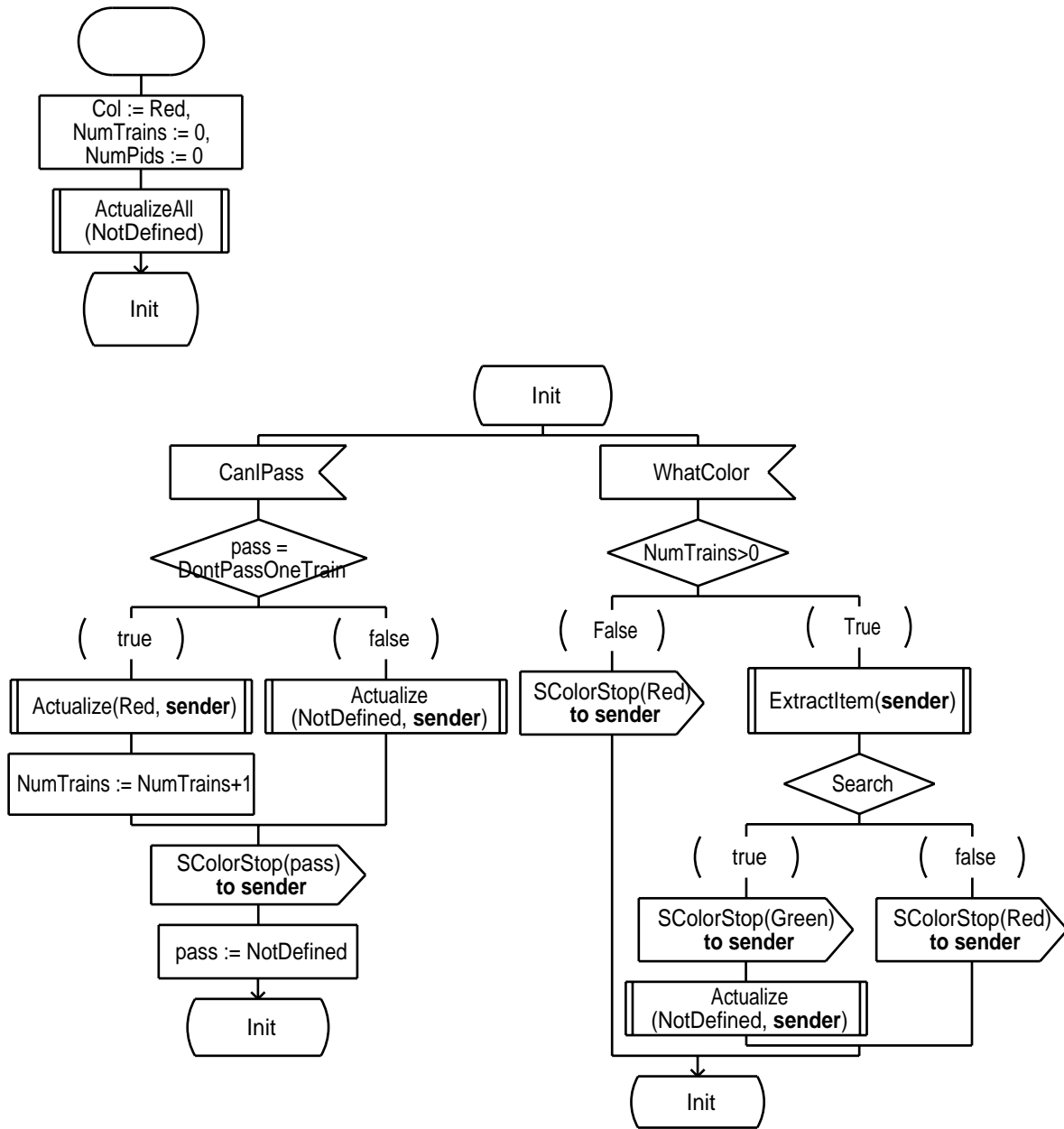




**exported as**  
RemoteTrainPosition  
**procedure** TrainPosition  
**returns** Integer



process StopTrain



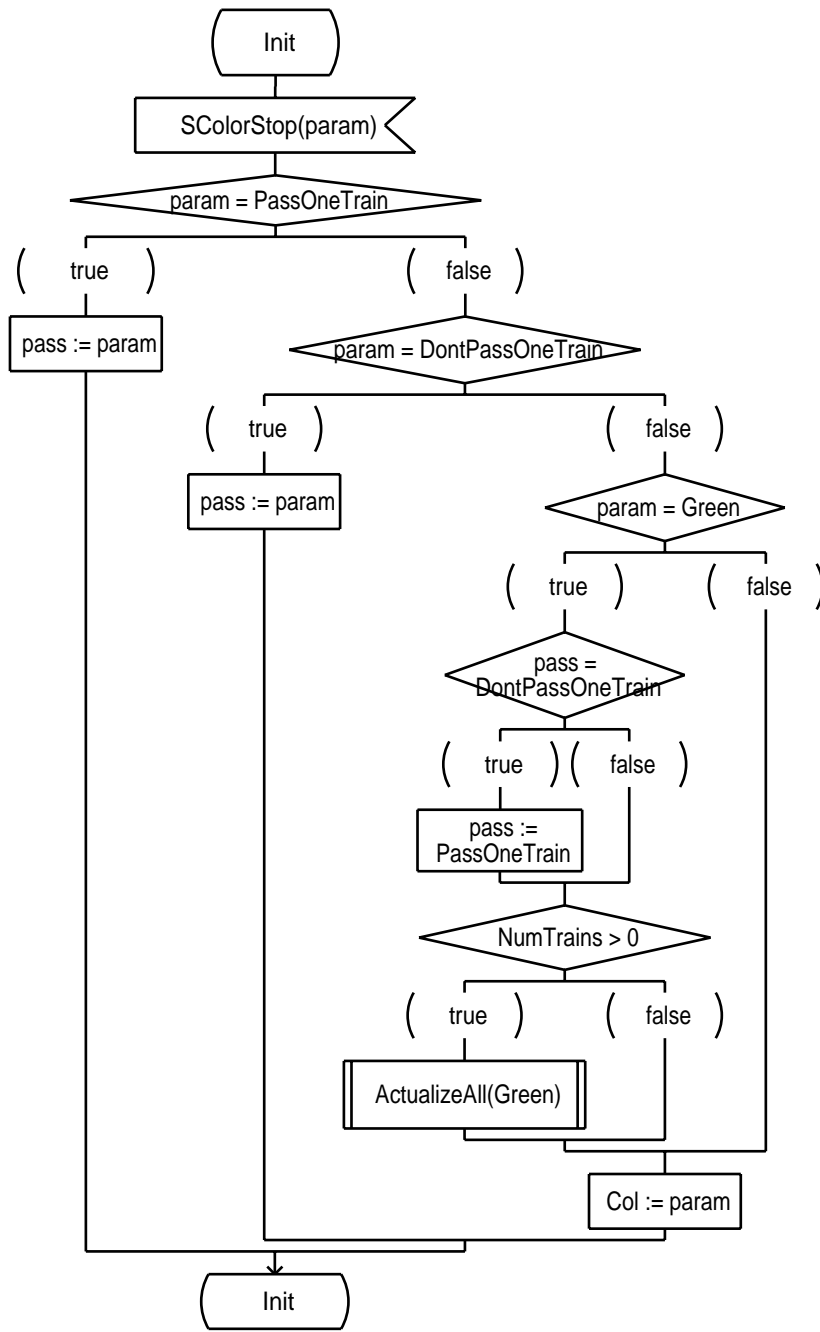
```

DCL param, Col, pass ColorStop;

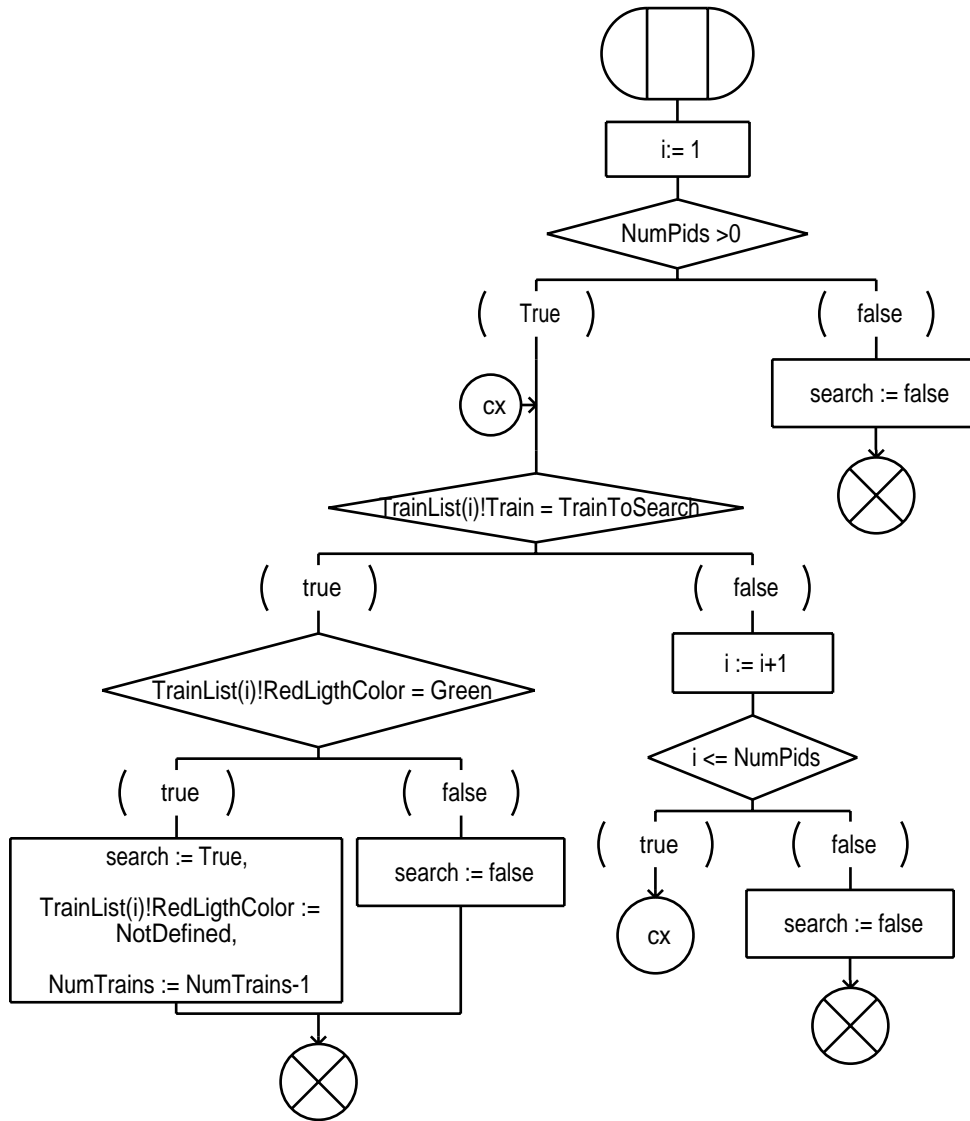
NEWTYPE TypeRedLigth
STRUCT
  Train PID;
  RedLigthColor ColorStop;
ENDNEWTYPE TypeRedLigth;

Newtype ListRedLigth Array(TypeTrackNumber, TypeRedLigth);
Endnewtype;

DCL NumTrains, NumPids Integer;
DCL TrainList ListRedLigth;
DCL Search Boolean;
  
```

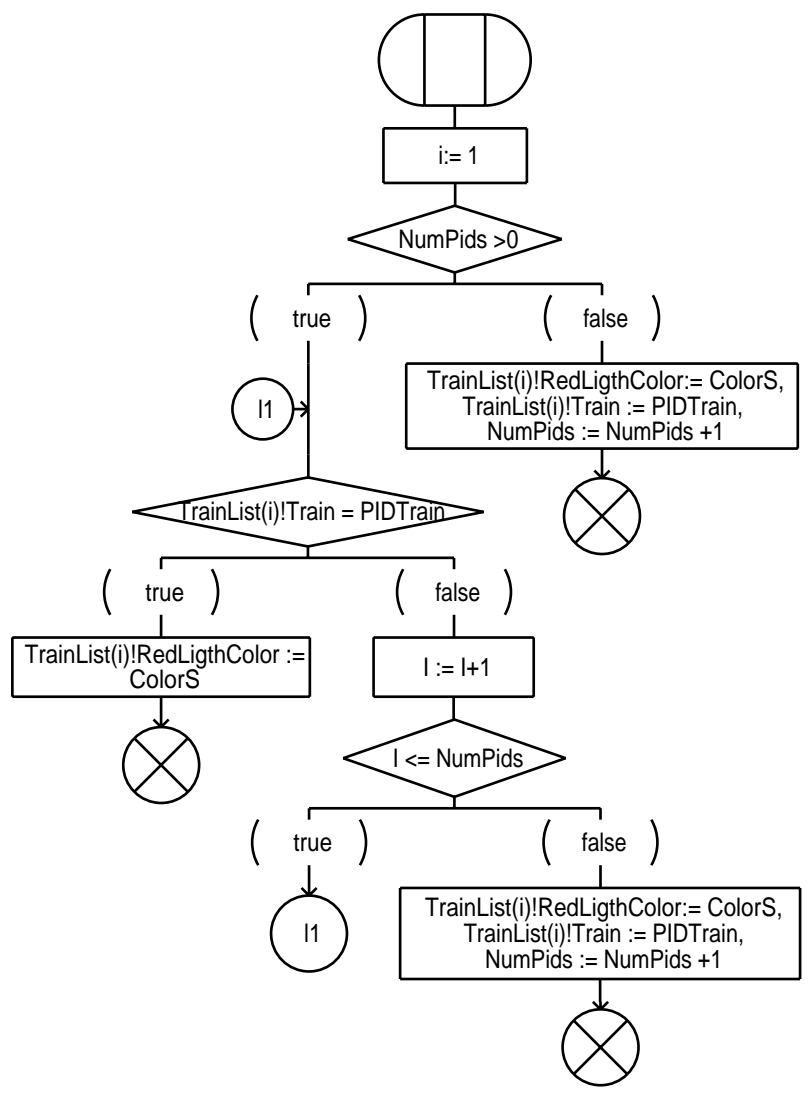


**procedure** ExtractItem  
**Fpar** TrainToSearch PID



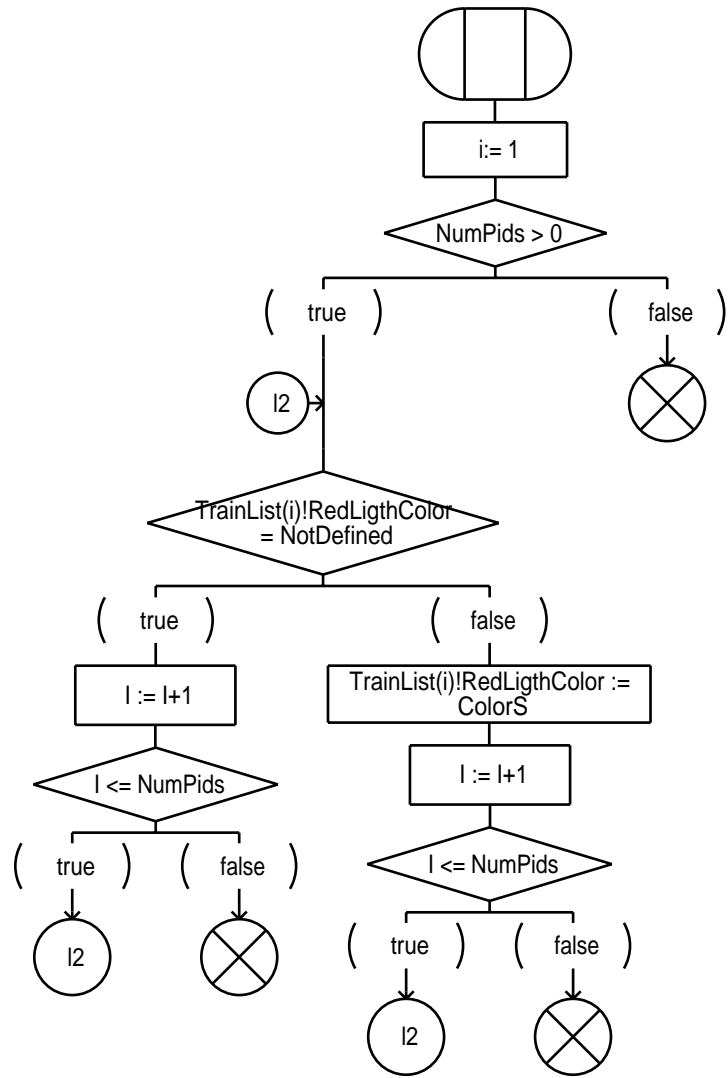
**DCL** i Integer;

**procedure** Actualize  
FPAR ColorS ColorStop, PIDTrain PID



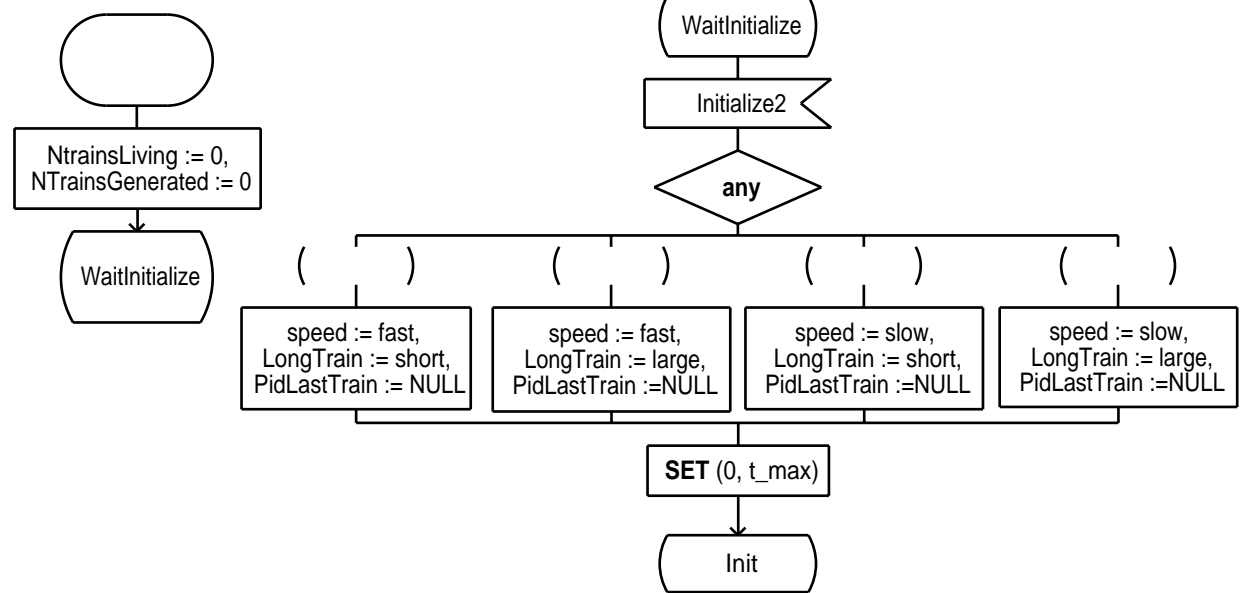
**DCL** i Integer;

**procedure** ActualizeAll  
FPAR ColorS ColorStop



**DCL** i Integer;

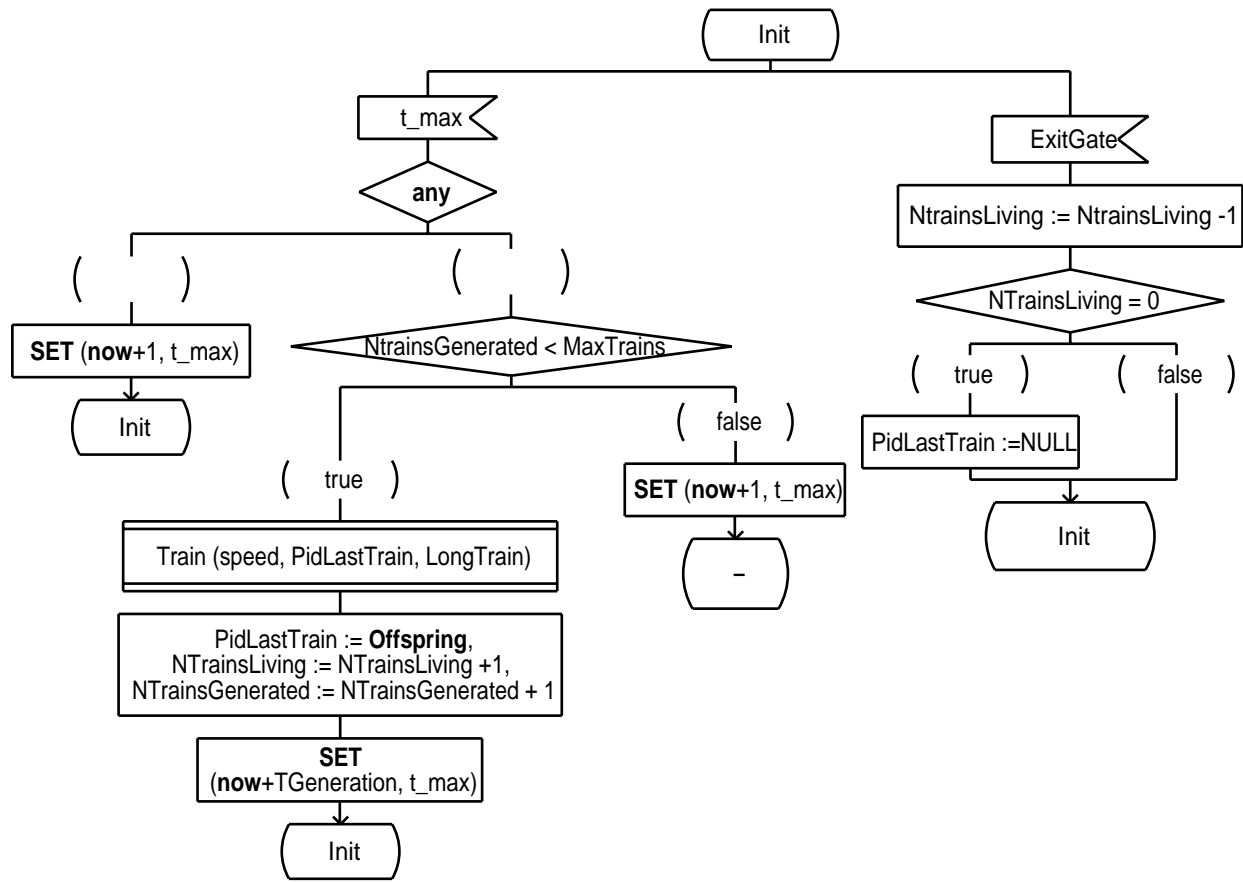
process SchedTrains



```

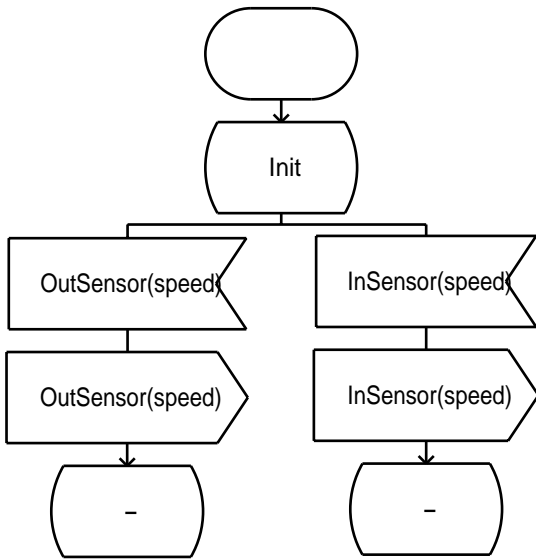
DCL LongTrain, NTrainsLiving Integer;
DCL PidLastTrain PID;
DCL speed TSpeed;
DCL NTrainsGenerated Integer;

timer t_max;
  
```





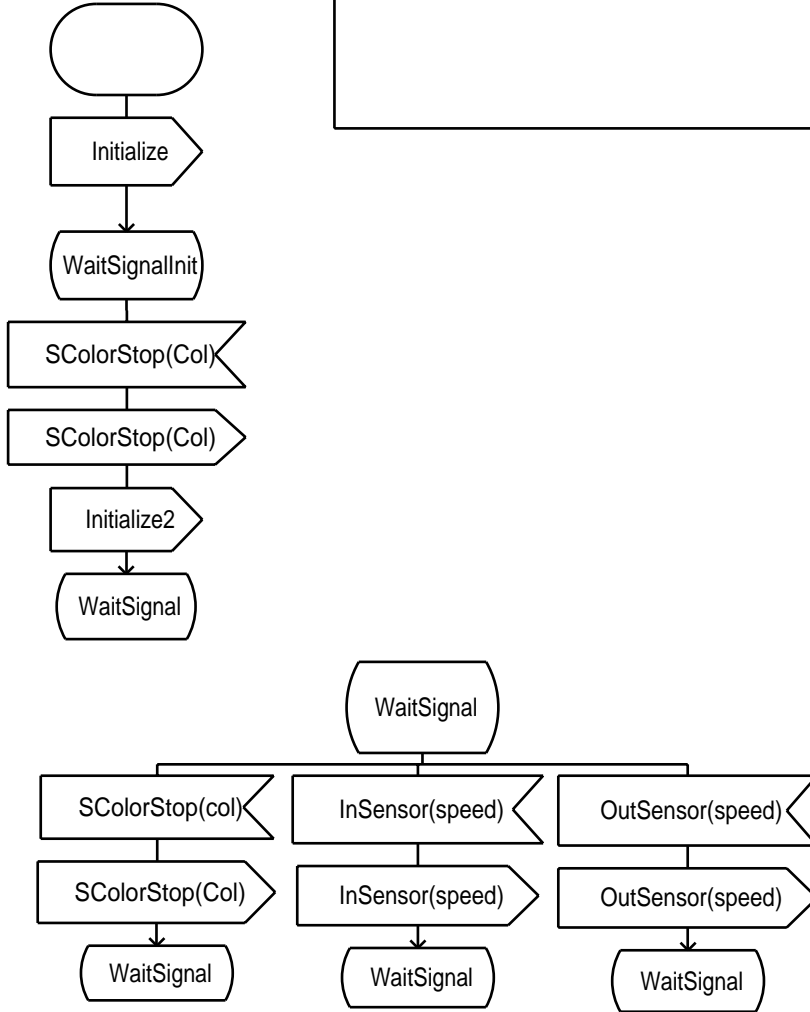
**process** Sensor

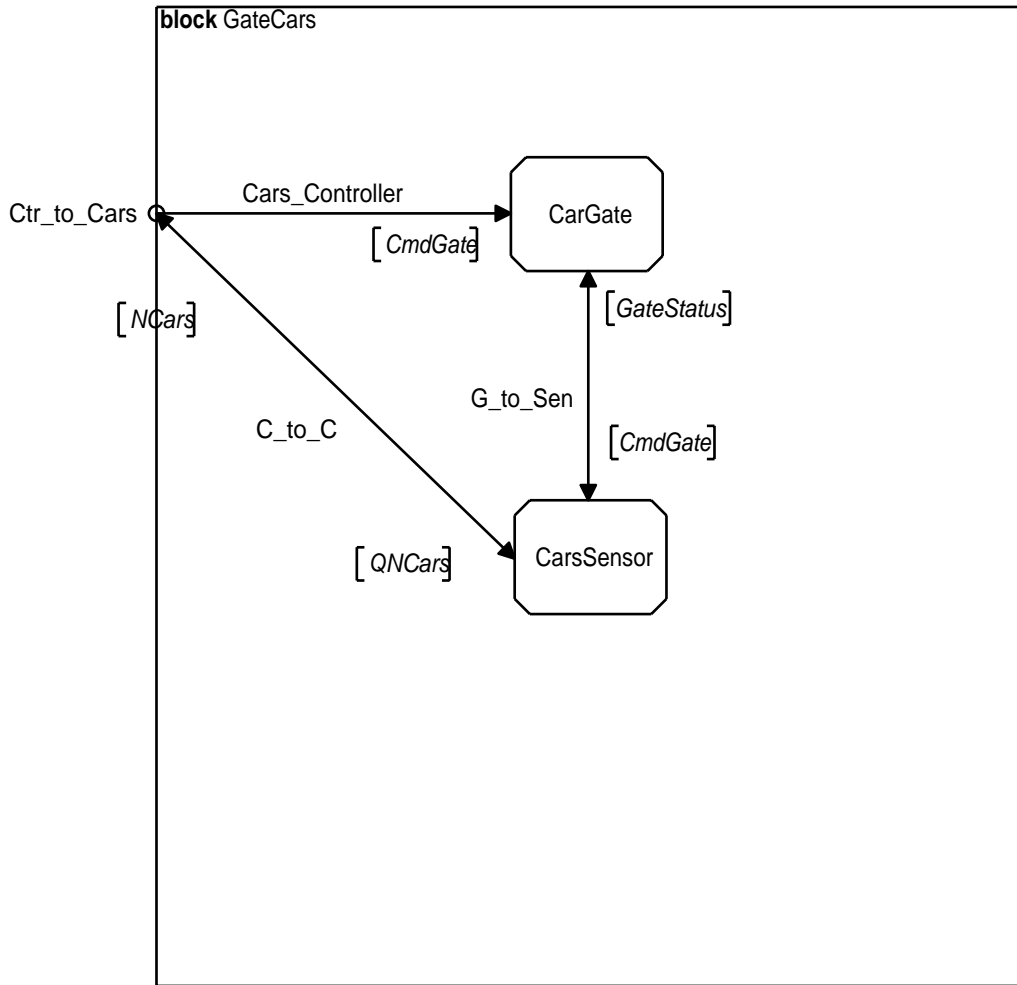


DCL speed TSpeed;

process MessagesHandler

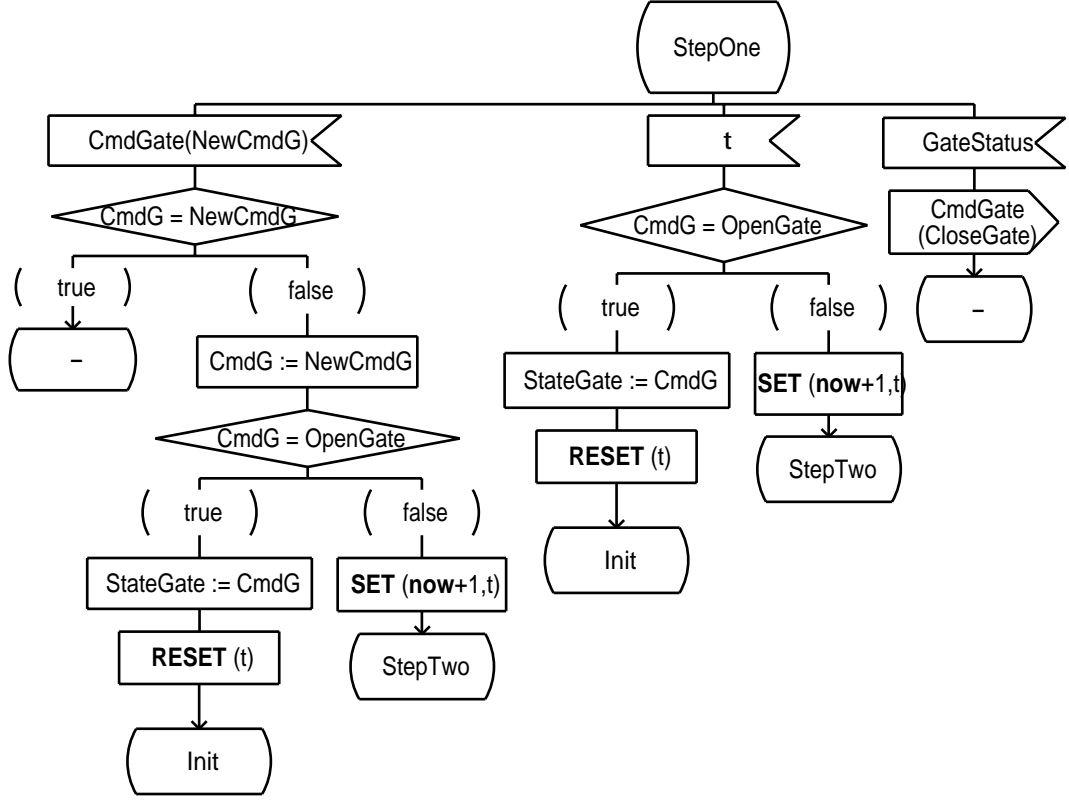
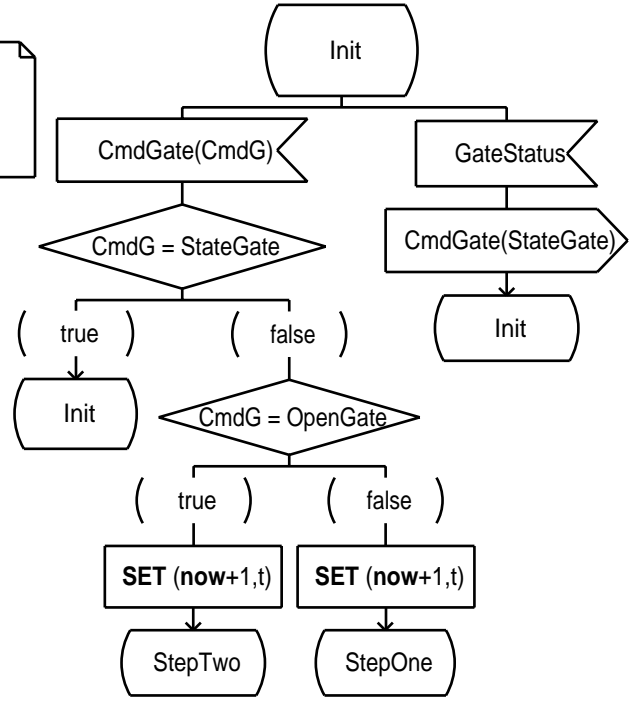
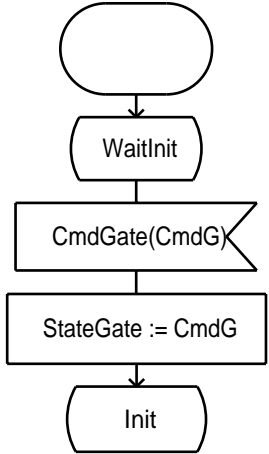
```
DCL Col ColorStop;  
DCL speed TSpeed;
```

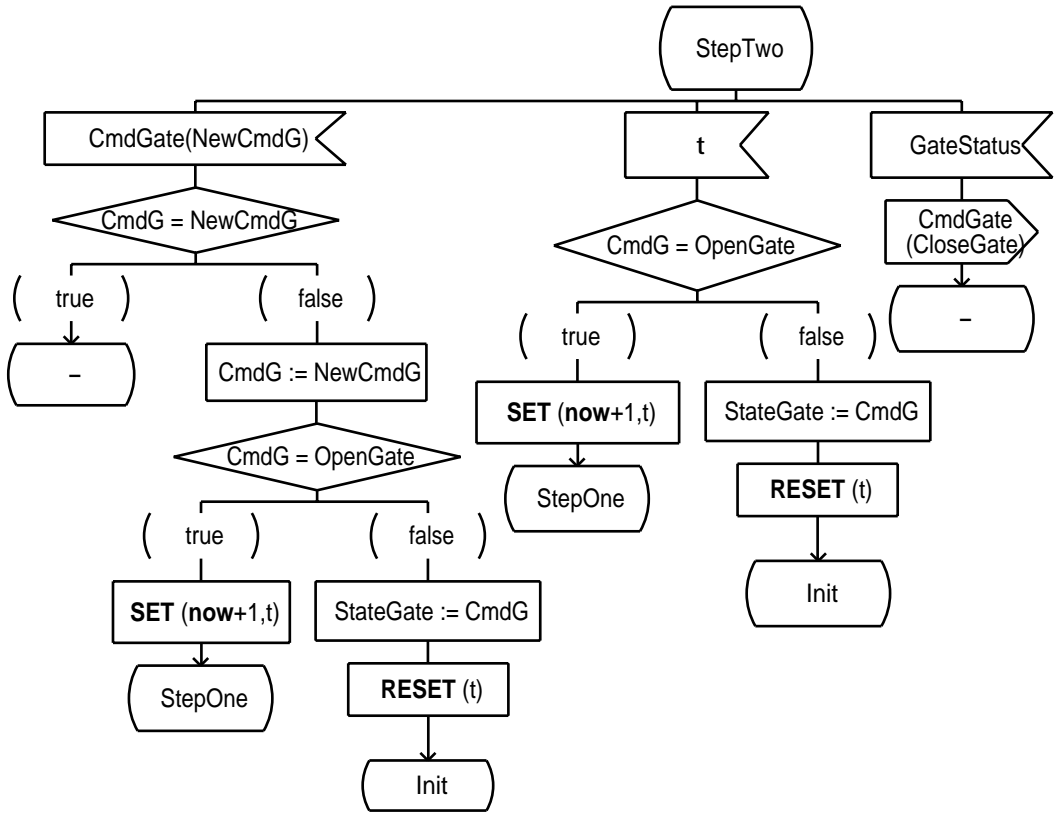




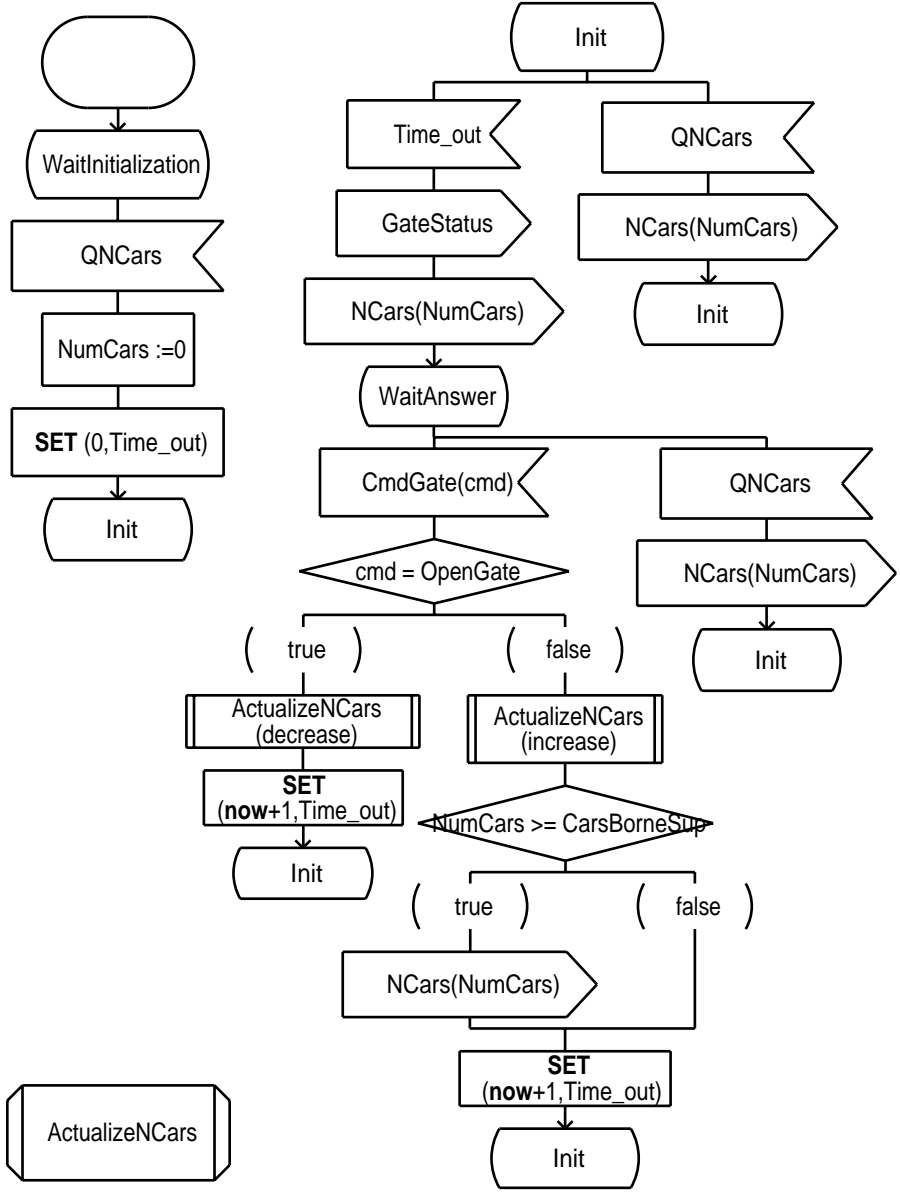
process CarGate

DCL CmdG, NewCmdG, StateGate CGate;  
 Timer t;





process CarsSensor

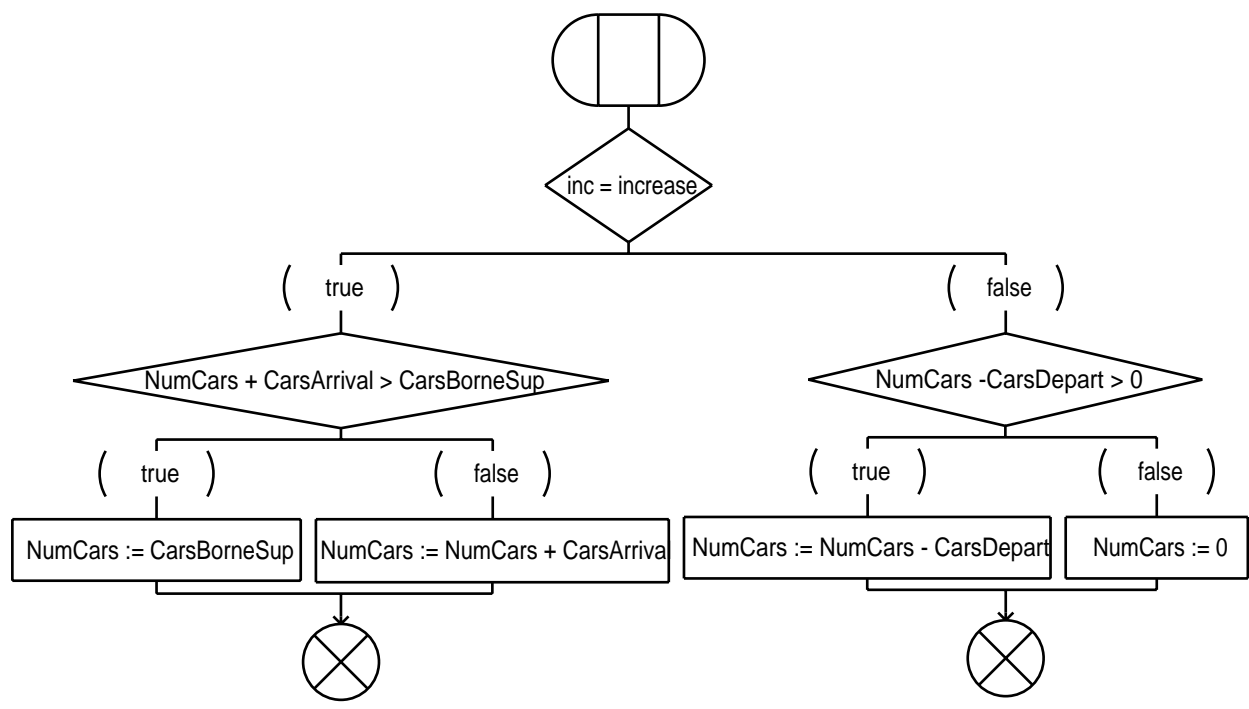


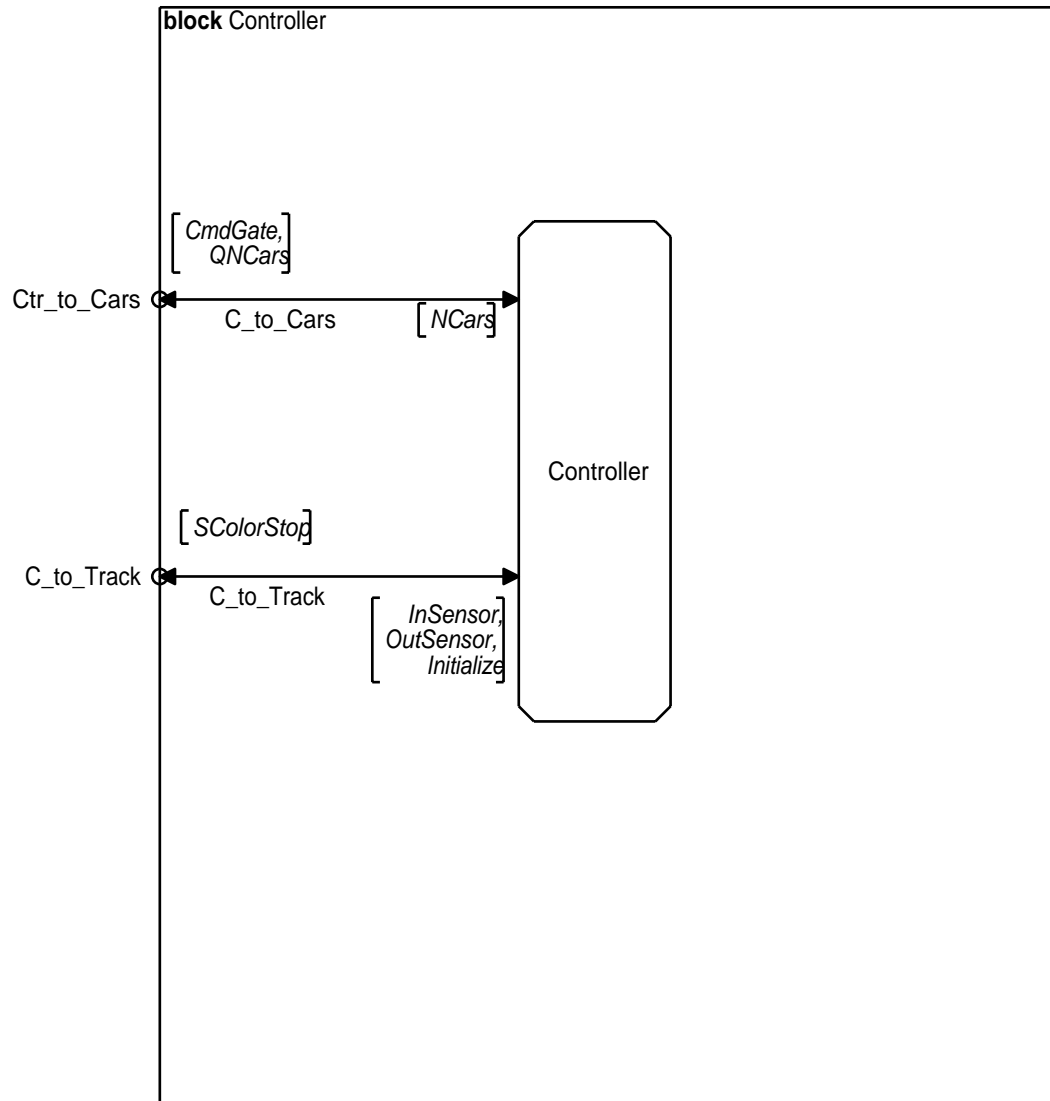
```

DCL NumCars Integer;
DCL cmd CGate;

TIMER Time_out;
synonym increase = 1;
synonym decrease = -1;
  
```

**procedure** ActualizeNCars  
FPA inc Integer







**process** Controller

```
NEWTYPE PidTable
  ARRAY (TypeTrackNumber, PID)
ENDNEWTYPE PidTable;

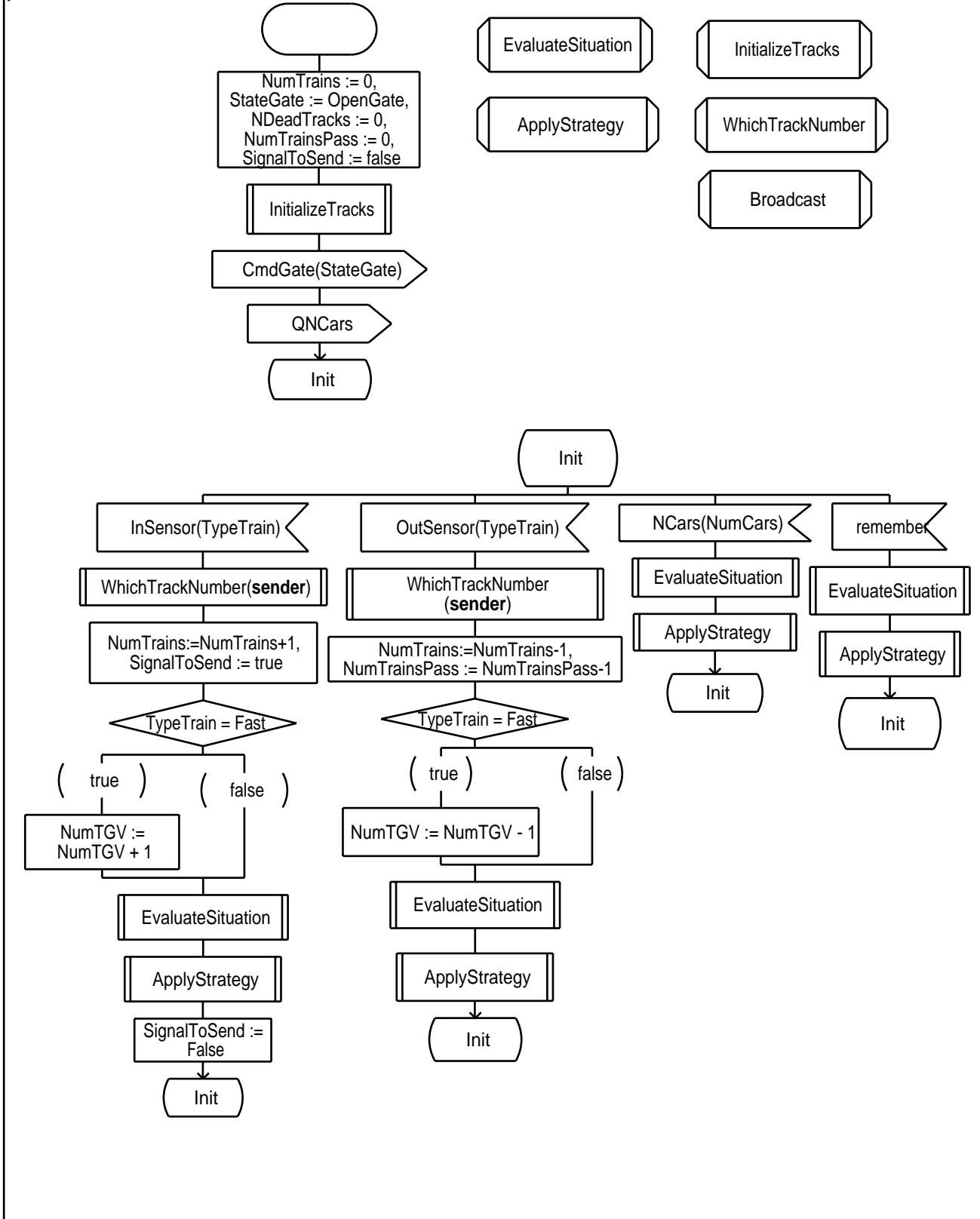
NEWTYPE t_element_track
  STRUCT
    NumTrack TypeTrackNumber;
    NumTrains Integer;
    PidStop PID;
ENDNEWTYPE t_element_track;

NEWTYPE TypeTrackList Array (TypeTrackNumber, t_element_track)
ENDNEWTYPE;

DCL Pass Boolean;
DCL NumTrains, NumTrainsPass, NumTrainsWaiting Integer;
DCL TypeTrain TSpeed;
DCL NumCars Integer;
DCL NumTrack, NDeadTracks TypeTrackNumber;
DCL TrackList TypeTrackList;
DCL NumTGV Integer;
DCL StateGate CGate;
DCL Situation TypeSituation;
DCL PidsTable PidTable;
DCL t_trains Integer;
DCL Tpriority Boolean;
DCL SignalToSend Boolean;
```

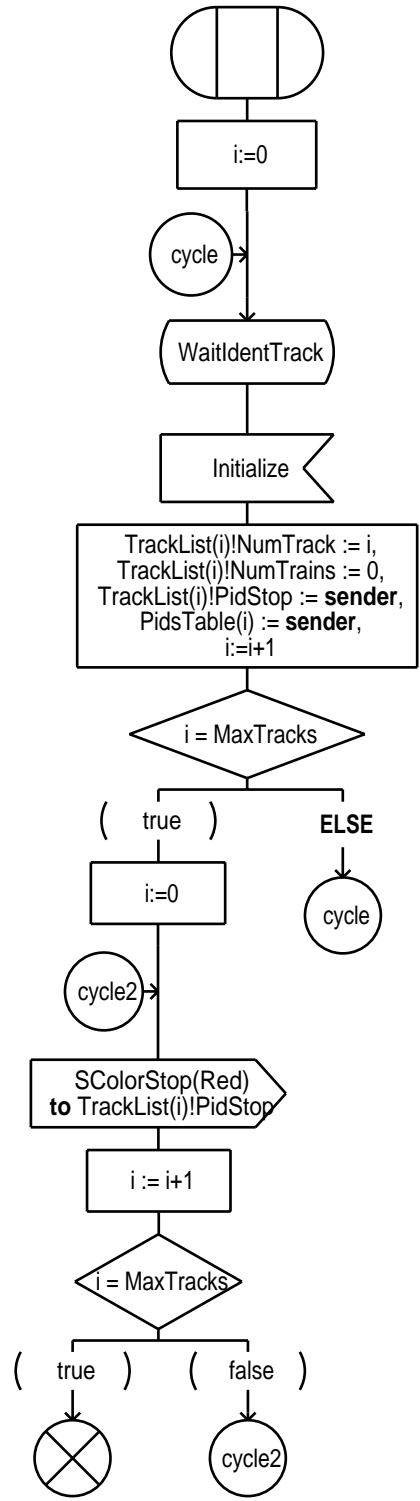
```
Timer remember;
```

process Controller

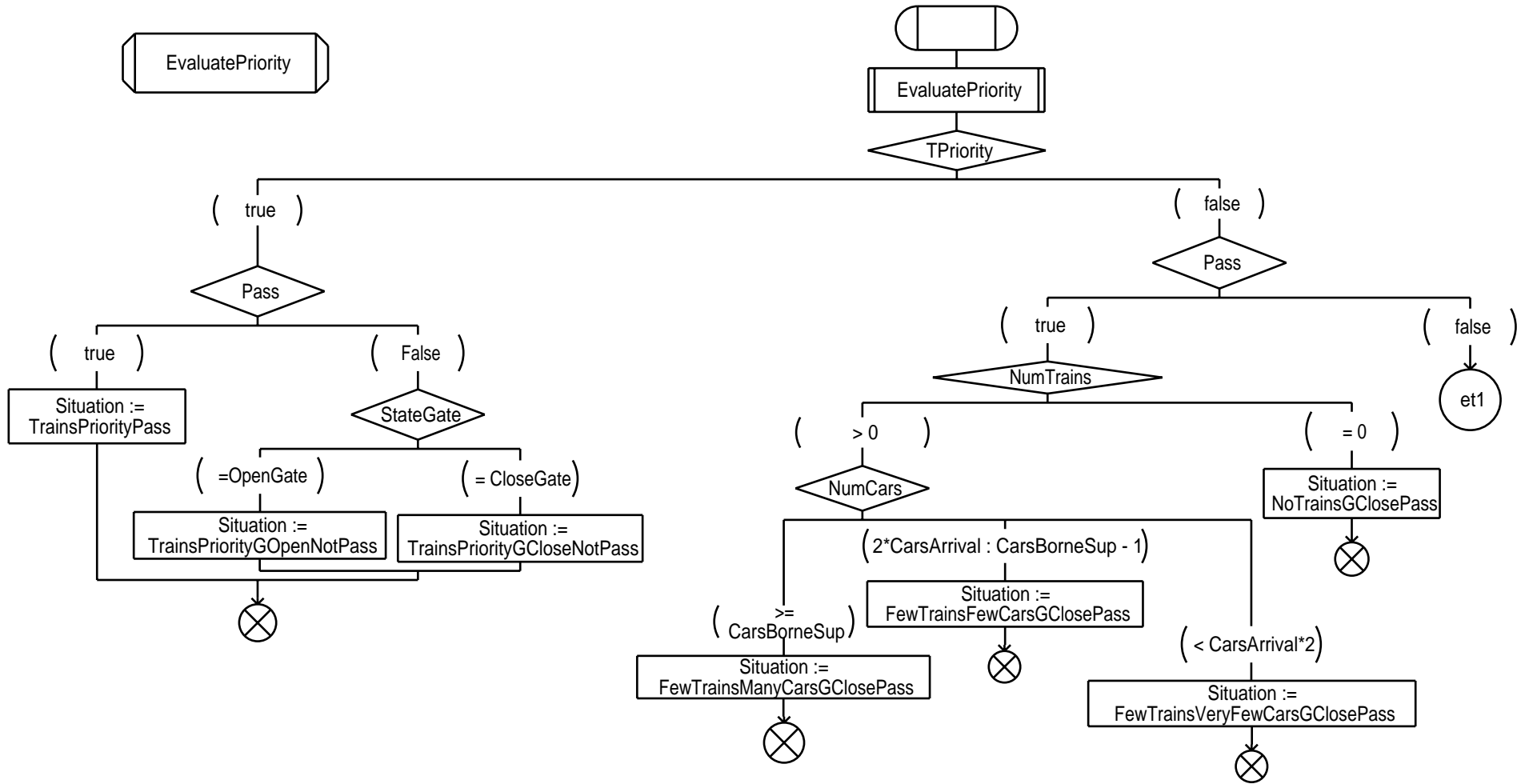


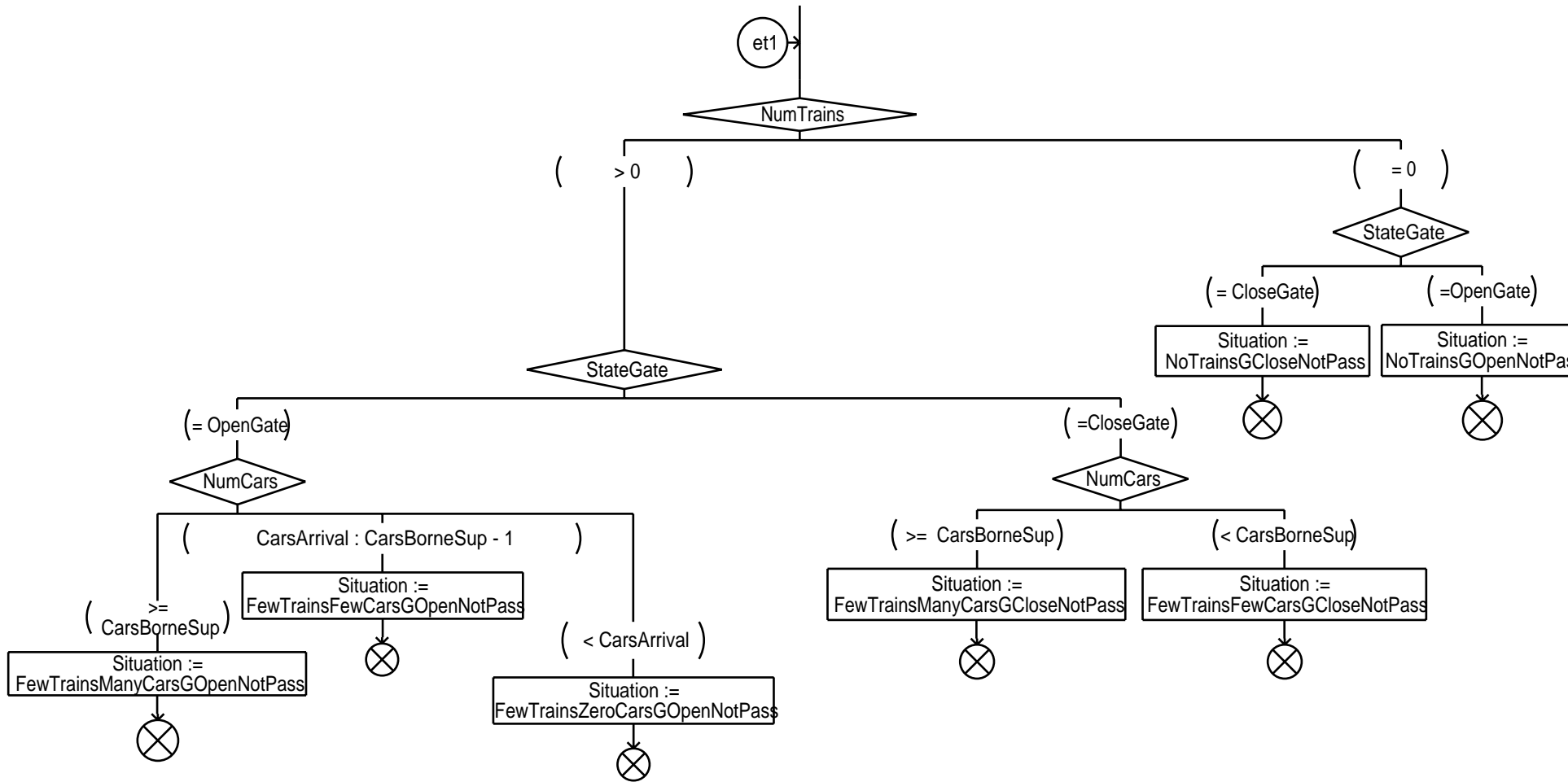
procedure InitializeTracks

DCL i TypeTrackNumber;

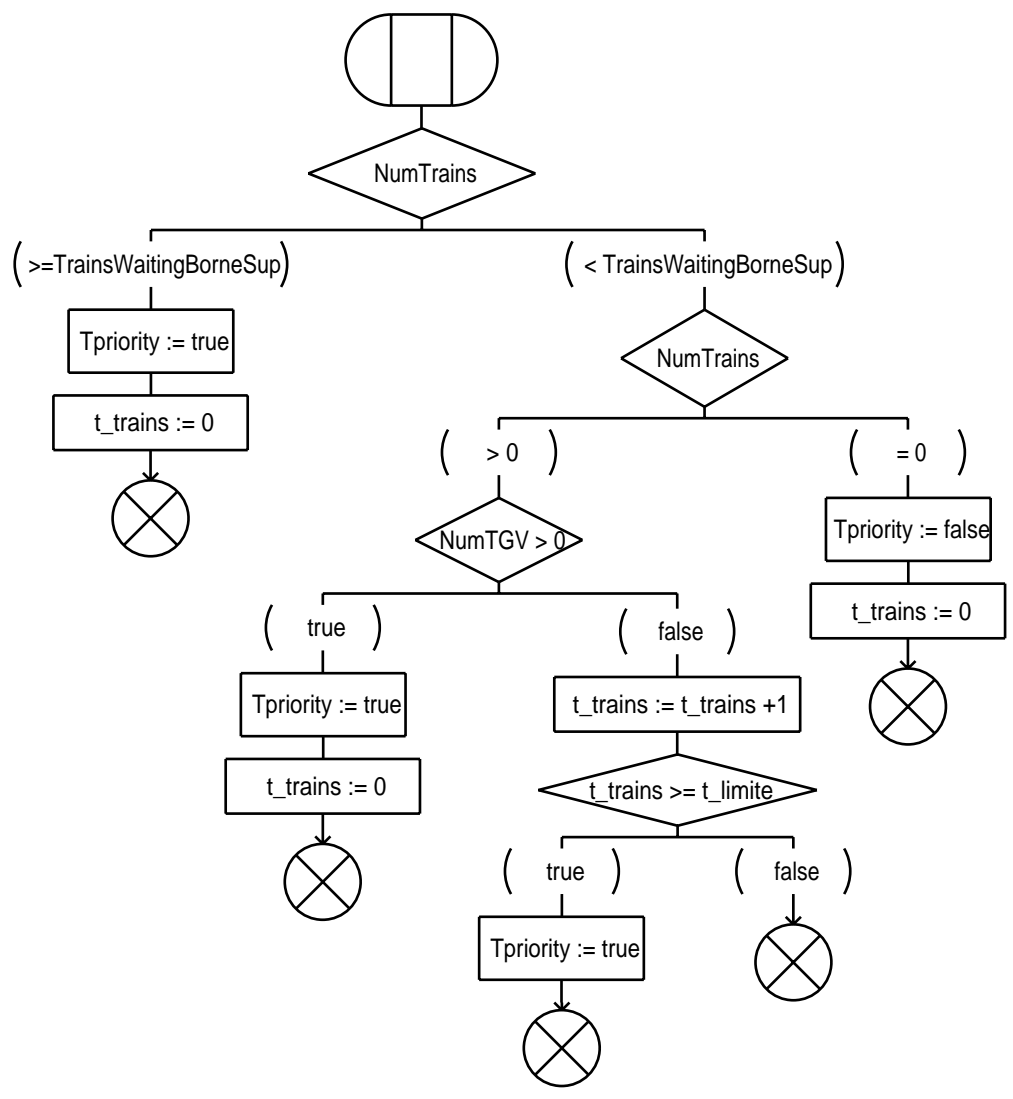


**procedure** EvaluateSituation

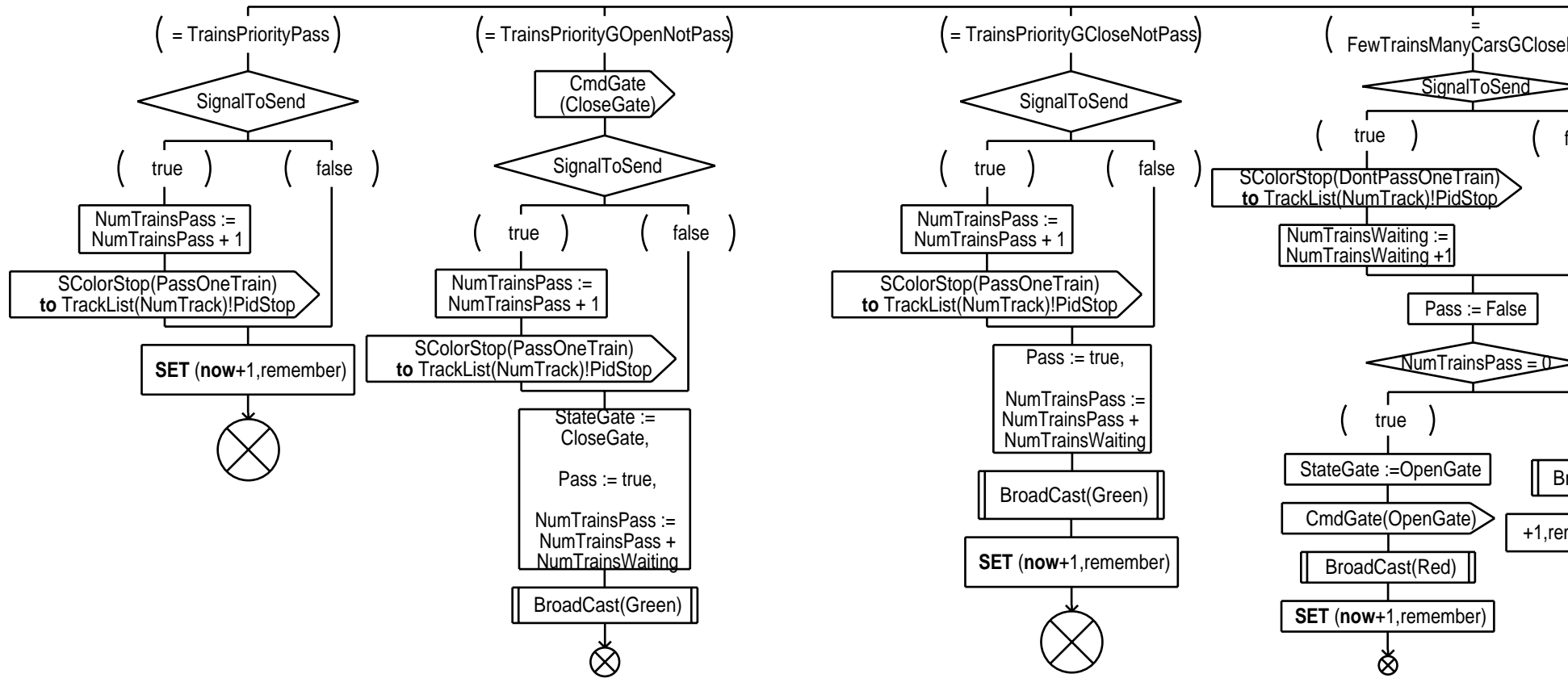


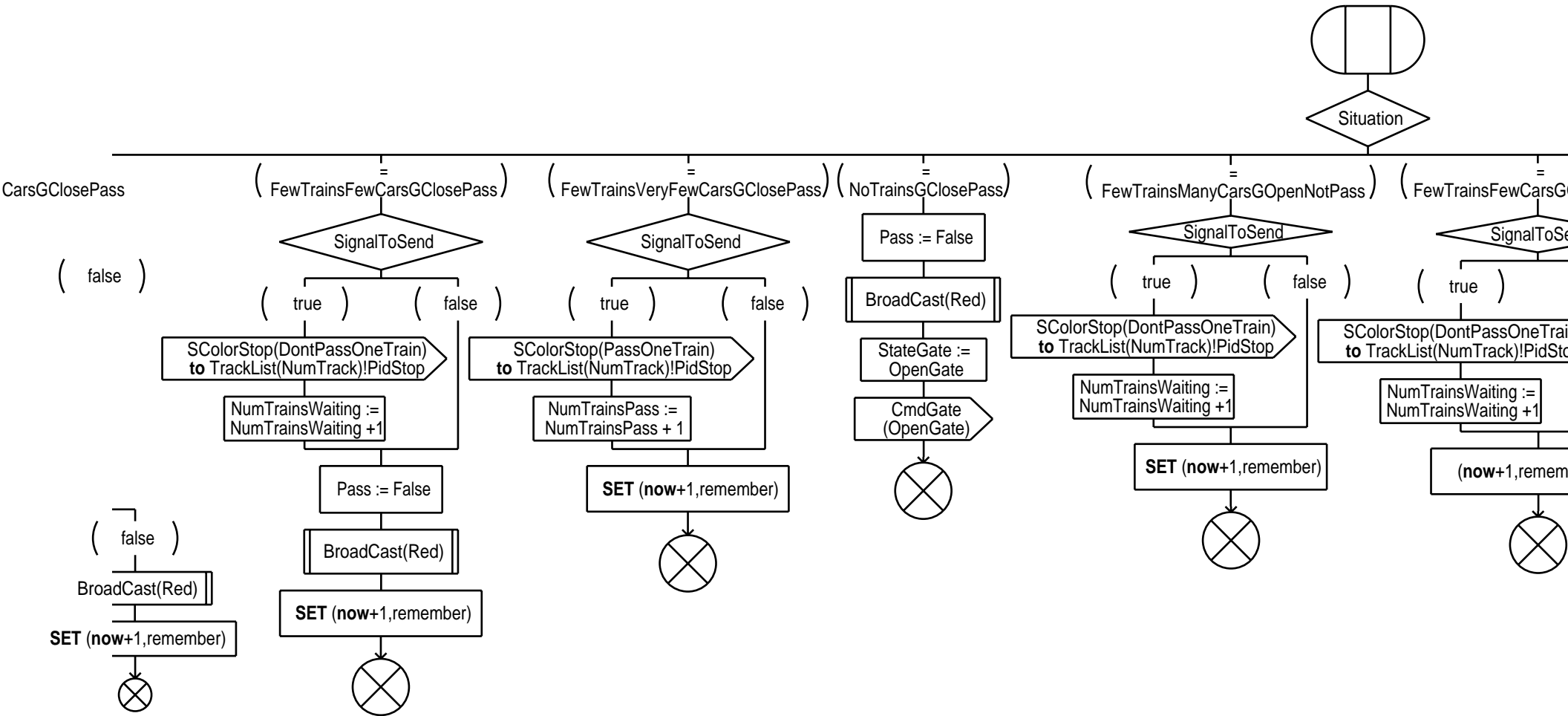


procedure EvaluatePriority

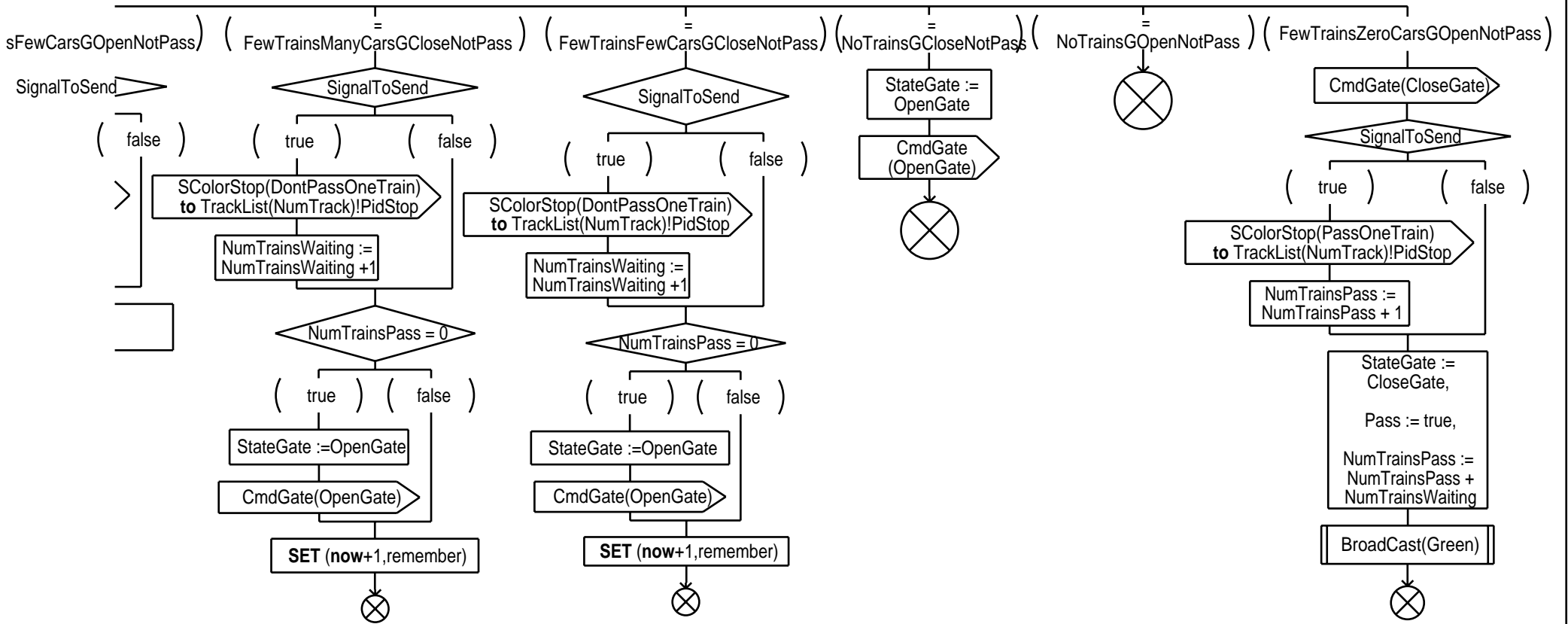


**procedure** ApplyStrategy

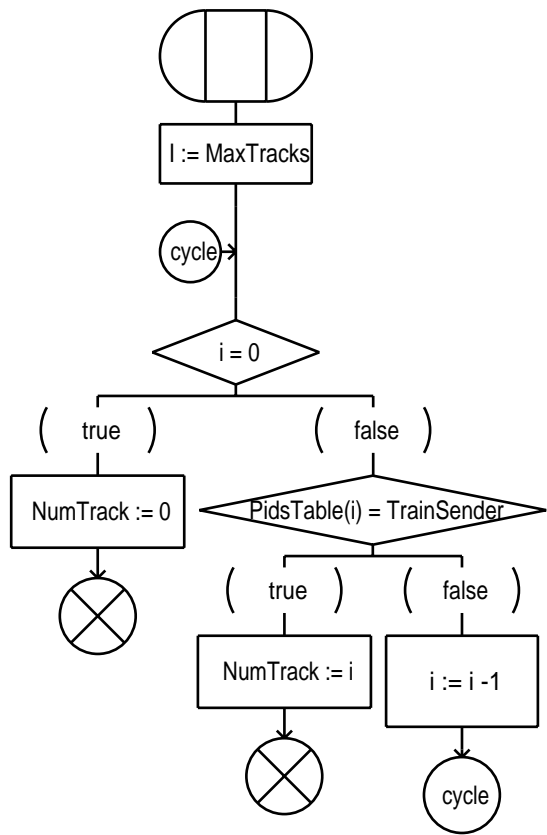








**procedure** WhichTrackNumber  
**fpar** TrainSender PID



**DCL** i Integer;

**procedure Broadcast**  
**FPAR col ColorStop**

**DCL** i Integer;

