```
CONTEXT
   C0
CONSTANTS
   rangeSpeed
AXIOMS
   axm1   :   rangeSpeed = 0‥5000
END
```

```
MACHINE
    M0
SEES
    C0
VARIABLES
    currentSpeed
INVARIANTS
    inv1   :   currentSpeed ∈ rangeSpeed
EVENTS
    INITIALISATION   ≙
    STATUS
      ordinary
    BEGIN
      act1   :   currentSpeed ≔ 0
    END

    updateVehiculeSpeed   ≙
    STATUS
      ordinary
    ANY
      val
    WHERE
      grd1   :   val∈ rangeSpeed
    THEN
      act1   :   currentSpeed≔val
    END

END
```

```
CONTEXT
    C1
EXTENDS
    C0
SETS
    keyStates
    SCSLeverPositions
CONSTANTS
    Neutral
    Downward
    Upward
    Downward5
    Downward7
    Backward
    Forward
    Upward5
    Upward7
    NoKeyInserted
    KeyInserted
    KeyInIgnitionOnPosition
    gasbrakeRange
    updateDur
AXIOMS
    axm1   :   partition(SCSLeverPositions, Upward, Downward, {Backward}, {Forward}, {Neutral})
    axm2   :   partition(Upward,{Upward5}, {Upward7})
    axm3   :   partition(Downward,{Downward5},{Downward7})
    axm4   :   partition(keyStates, {NoKeyInserted}, {KeyInserted}, {KeyInIgnitionOnPosition})
    axm5   :   gasbrakeRange= 0‥225
    axm6   :   updateDur∈ ℕ ∧ updateDur=600
END
```

MACHINE
M1
REFINES
M0
SEES
C1
VARIABLES
currentSpeed
keyState
keyStateP
SCSLeverUD
SCSLeverFB
SCSLeverUDP
SCSLeverFBP
brakePedal
currentTime
gasPedal
speedLimiterSwitchOn
buttonHead
rangeRadarState
nextTest
lastTest
speedLimit
INVARIANTS
inv1 : brakePedal>0 ⇒ gasPedal=0
inv2 : keyState ∈ keyStates ∧ keyStateP ∈ keyStates ∧ SCSLeverUD∈ SCSLeverPositions ∧ SCSLeverUDP∈ SCSLeverPositions ∧ SCSLeverFB∈ SCSLeverPositions ∧ SCSLeverFBP∈ SCSLeverPositions ∧ brakePedal ∈ gasbrakeRange ∧ currentTime ∈ ℕ ∧ gasPedal ∈ gasbrakeRange ∧ buttonHead ∈ BOOL
inv3 : keyState=NoKeyInserted ⇒ keyStateP=NoKeyInserted ∨ keyStateP=KeyInserted
inv4 : keyState=KeyInIgnitionOnPosition ⇒ keyStateP=KeyInIgnitionOnPosition ∨ keyStateP=KeyInserted
inv5 : (SCSLeverUD ∈ Upward ⇒ SCSLeverUDP ∈ Upwardu{Neutral}) ∧ (SCSLeverUD ∈ Downward ⇒ SCSLeverUDP ∈ Downwardu{Neutral})
inv6 : SCSLeverFBP ≠ Neutral ∧ SCSLeverFBP ≠ SCSLeverFB⇒ SCSLeverFB = Neutral
inv7 : speedLimiterSwitchOn ∈ BOOL
inv8 : buttonHead=TRUE ⇒ keyState=KeyInIgnitionOnPosition ∧ SCSLeverFB ≠ Backward // *SCS-35*
inv9 : speedLimiterSwitchOn=TRUE ⇒ buttonHead=TRUE ∧ gasPedal≤90 // *SCS-29*
inv10 : speedLimit ∈ rangeSpeed
inv12 : rangeRadarState∈ BOOL ∧ nextTest∈ ℕ
inv13 : keyState=KeyInIgnitionOnPosition⇒ nextTest>currentTime ∧ nextTest−currentTime≤updateDur
inv14 : keyState=KeyInIgnitionOnPosition ∧ keyStateP≠KeyInIgnitionOnPosition ⇒ nextTest=currentTime+updateDur
inv15 : SCSLeverFB=Backward ⇒speedLimiterSwitchOn=FALSE
inv16 : brakePedal=0 ∨ gasPedal=0
inv17 : keyState≠KeyInIgnitionOnPosition ⇒ brakePedal=0 ∧ gasPedal=0
inv18 : lastTest∈ ℕ ∧ lastTest≤currentTime
inv19 : keyStateP≠KeyInIgnitionOnPosition ∧ keyState=KeyInIgnitionOnPosition⇒lastTest=currentTime
inv20 : keyState≠KeyInIgnitionOnPosition⇒lastTest=0
inv21 : (nextTest=0 ∧ lastTest=0) ∨ nextTest−lastTest=updateDur
EVENTS
INITIALISATION ≙
extended
STATUS
ordinary
BEGIN
act1 : currentSpeed ≔ 0
act2 : keyState≔NoKeyInserted
act3 : keyStateP≔NoKeyInserted
act4 : SCSLeverUD≔Neutral
act5 : SCSLeverUDP≔Neutral
act6 : SCSLeverFB≔Neutral
act7 : SCSLeverFBP≔Neutral
act8 : brakePedal≔0
act9 : currentTime≔0
act10 : gasPedal≔0

```
  act11   :   speedLimiterSwitchOn≔FALSE
  act12   :   rangeRadarState≔TRUE
  act13   :   nextTest≔0
  act14   :   buttonHead≔FALSE
  act15   :   speedLimit≔0
  act16   :   lastTest≔0
END

brakepedal   ≙
STATUS
  ordinary
ANY
  val
WHERE
  grd1   :   keyState=KeyInIgnitionOnPosition
  grd2   :   val∈ gasbrakeRange ∧ val≠brakePedal
  grd3   :   gasPedal=0
THEN
  act1   :   brakePedal≔val
  act2   :   SCSLeverFBP≔SCSLeverFB
END

gaspedal   ≙
STATUS
  ordinary
ANY
  gas
  sw
WHERE
  grd1   :   brakePedal=0 ∧ keyState=KeyInIgnitionOnPosition
  grd2   :   gas∈gasbrakeRange ∧ gas≠gasPedal
  grd3   :   sw={TRUE↦FALSE, FALSE↦speedLimiterSwitchOn}
             (bool(gas>90))
THEN
  act1   :   gasPedal≔gas
  act2   :   keyStateP≔keyState
  act3   :   SCSLeverUDP≔SCSLeverUD
  act4   :   SCSLeverFBP≔SCSLeverFB
  act5   :   speedLimiterSwitchOn≔sw
END

moveKey   ≙
STATUS
  ordinary
ANY
  valkey
  radstate
WHERE
  grd1   :   currentSpeed=0
  grd2   :   valkey ∈ keyStates
             (keyState=NoKeyInserted ⟹ valkey=KeyInserted)
             ∧
  grd3   :   (keyState=KeyInserted ⟹ valkey∈ {NoKeyInserted, KeyInIgnitionOnPosition})
                ∧
             (keyState=KeyInIgnitionOnPosition ⟹ valkey=KeyInserted)
  grd4   :   radstate∈ BOOL
  grd5   :   valkey≠KeyInIgnitionOnPosition⟹radstate=FALSE
THEN
  act1   :   keyState≔valkey
  act2   :   keyStateP≔keyState
  act3   :   SCSLeverUDP≔SCSLeverUD
  act4   :   SCSLeverFBP≔SCSLeverFB
  act5   :   rangeRadarState≔radstate
  act6   :   nextTest≔{TRUE↦currentTime+updateDur, FALSE↦0}
             (bool(valkey=KeyInIgnitionOnPosition))
  act7   :   brakePedal≔ {TRUE↦brakePedal, FALSE↦0}
             (bool(keyState≠KeyInIgnitionOnPosition))
  act8   :   gasPedal≔ {TRUE↦gasPedal, FALSE↦0}
             (bool(keyState≠KeyInIgnitionOnPosition))
  act9   :   speedLimiterSwitchOn≔ FALSE
  act10   :   buttonHead≔FALSE
```

```
  act11   :   lastTest≔{TRUE↦currentTime,FALSE↦0}(bool
              (valkey=KeyInIgnitionOnPosition))
END

moveSCSLeverUD   ≙
STATUS
  ordinary
ANY
  valSCS
WHERE
  grd1   :   valSCS ∈  Upwardu Downwardu {Neutral} ∧ valSCS≠SCSLeverUD
  grd2   :   SCSLeverFB=Neutral
  grd3   :   SCSLeverUD ≠ Neutral ⟹
                (SCSLeverUD ∈ Upward ∧ valSCS ∈ Upward)
             v
                (SCSLeverUD ∈ Downward ∧ valSCS ∈ Downward)
             v
             valSCS = Neutral
THEN
  act1   :   SCSLeverUD≔valSCS
  act2   :   SCSLeverUDP≔SCSLeverUD
  act3   :   SCSLeverFBP≔SCSLeverFB
  act4   :   keyStateP≔keyState
END

moveSCSLeverFB   ≙
STATUS
  ordinary
ANY
  valSCS
  sw
  bh
WHERE
  grd1   :   valSCS ∈ {Backward, Forward, Neutral} ∧ valSCS≠SCSLeverFB
  grd2   :   SCSLeverUD=Neutral
  grd3   :   SCSLeverFB ≠ Neutral ⟹ valSCS = Neutral
  grd4   :   bh={TRUE↦buttonHead, FALSE↦FALSE}(bool(valSCS ≠Backward))
  grd5   :   sw={TRUE↦FALSE, FALSE↦speedLimiterSwitchOn}(bool(bh=FALSE))
THEN
  act1   :   SCSLeverFB≔valSCS
  act2   :   SCSLeverFBP≔SCSLeverFB
  act3   :   keyStateP≔keyState
  act4   :   speedLimiterSwitchOn≔sw
  act5   :   buttonHead≔bh
END

progress   ≙
  extended
STATUS
  ordinary
REFINES
  updateVehiculeSpeed
ANY
  val
  radstate
WHERE
  grd1   :   val∈ rangeSpeed
  grd2   :   radstate∈ BOOL
  grd3   :   keyState≠KeyInIgnitionOnPosition ∨ nextTest≠currentTime+1⟹
             radstate=rangeRadarState
THEN
  act1   :   currentSpeed≔val
  act2   :   currentTime≔currentTime+1
  act3   :   SCSLeverUDP≔SCSLeverUD
  act4   :   SCSLeverFBP≔SCSLeverFB
  act5   :   keyStateP≔keyState
  act6   :   rangeRadarState≔radstate
  act7   :   nextTest≔{TRUE↦currentTime+1+updateDur, FALSE↦nextTest}
               (bool(keyState=KeyInIgnitionOnPosition  ∧ nextTest=currentTime+1))
  act8   :   lastTest≔{TRUE↦currentTime+1, FALSE↦lastTest}
               (bool(keyState=KeyInIgnitionOnPosition  ∧ nextTest=currentTime+1))
```

```
  END

  pushButtonHead   ≙
  STATUS
    ordinary
  ANY
    sl
  WHERE
    grd1   :   buttonHead=FALSE
    grd2   :   speedLimiterSwitchOn=FALSE ∧ SCSLeverFB≠Backward ∧
               keyState=KeyInIgnitionOnPosition ∧ gasPedal≤90
    grd3   :   keyState=KeyInIgnitionOnPosition ∧ SCSLeverFB ≠ Backward
    grd4   :   currentSpeed≤speedLimit
    grd5   :   sl∈rangeSpeed
    grd6   :   currentSpeed≤sl
  THEN
    act1   :   speedLimiterSwitchOn≔TRUE
    act2   :   speedLimit≔sl
    act3   :   SCSLeverUDP≔SCSLeverUD
    act4   :   SCSLeverFBP≔SCSLeverFB
    act5   :   keyStateP≔keyState
    act6   :   buttonHead≔TRUE
  END

  trafficSignDetection   ≙
  STATUS
    ordinary
  BEGIN
    act1   :   SCSLeverUDP≔SCSLeverUD
    act2   :   SCSLeverFBP≔SCSLeverFB
    act3   :   keyStateP≔keyState
  END

END
```

CONTEXT
C2
EXTENDS
C1
CONSTANTS
cruiseControlMode
trafficSignDetectionOn
desiredSpeedMax
gasLimit
speedActiv
AXIOMS
axm1 : ∀A· finite(A) ∧ A≠∅ ⇒ max(A)∈A
axm2 : ∀A· finite(A) ∧ A≠∅ ⇒ min(A)∈A
axm3 : ∀A,a· finite(A)∧ a∈ A⇒max(A)≥a
axm4 : ∀A,a· finite(A)∧ a∈ A⇒min(A)≤a
axm5 : ∀ a, b· a∈ℕ ∧ b∈ℕ ⇒ a ∗ b∈ ℕ
axm6 : ∀ a, b· a∈ℕ ∧ b∈ℕ ⇒ a + b∈ ℕ
axm7 : ∀ a, b· a∈ℕ ∧ b∈ℕ1 ⇒ a ÷ b∈ ℕ
axm8 : ∀a,b,c· a≤b ∧ b≤c ⇒ a≤c
axm9 : ∀a· a∈ ℕ ⇒ a÷10∗10≤a
axm10 : ∀a,b· a∈ℕ ∧ b∈ℕ ⇒ a−b≤a
axm11 : ∀a,b· a∈ℕ1 ∧ b∈ℕ1 ⇒ a∗b∈ℕ1
axm12 : ∀a· a∈ℕ ⇒ −a≤0
axm13 : cruiseControlMode∈ {1,2}
axm14 : trafficSignDetectionOn∈ BOOL
axm15 : desiredSpeedMax∈ rangeSpeed ∧ desiredSpeedMax=2000
axm16 : speedActiv ∈ rangeSpeed ∧ speedActiv=200
axm17 : gasLimit∈ ℕ ∧ gasLimit=90
END

MACHINE
M2
REFINES
M1
SEES
C2
VARIABLES
currentSpeed
desiredSpeed
desiredSpeedP
mandesiredSpeed
keyState
keyStateP
SCSLeverUD
SCSLeverFB
SCSLeverUDP
SCSLeverFBP
adapContr
adapContrP
normContr
normContrP
currentTime
lastTimeSCSLeverUD
lastdesiredSpeed
brakePedal
gasPedal
buttonHead
speedLimiterSwitchOn
speedLimit
detectedTrafficSign
detectedTrafficOn
trafficdetected
rangeRadarState
nextTest
lastTest
INVARIANTS

inv1 : desiredSpeed ∈ ℕ ∧ desiredSpeed≤desiredSpeedMax ∧ desiredSpeedP ∈ ℕ ∧ desiredSpeedP ≤desiredSpeedMax ∧ lastdesiredSpeed ∈ ℕ ∧ lastdesiredSpeed ≤ desiredSpeedMax ∧ lastTimeSCSLeverUD∈ℕ ∧ adapContr ∈ BOOL ∧ adapContrP ∈ BOOL ∧ normContr∈BOOL ∧ normContrP∈BOOL ∧ detectedTrafficOn ∈ BOOL ∧ trafficdetected∈ BOOL

inv2 : speedLimit ∈ rangeSpeed

inv3 : keyStateP=KeyInserted ∧ keyState=KeyInIgnitionOnPosition ⇒ desiredSpeed=0 // *SCS-1*

inv4 : SCSLeverFBP≠Forward ∧ SCSLeverFBP=Forward ⇒ (desiredSpeedP≠0 ∧ desiredSpeed=desiredSpeedP) ∨ (desiredSpeedP=0 ∧ desiredSpeed=min({desiredSpeedMax,currentSpeed})) // *SCS-2*

inv5 : normContr=TRUE ⇔ ((SCSLeverFB = Forward ∧ SCSLeverFBP ≠ Forward ∧ (currentSpeed≥speedActiv ∨ desiredSpeed≠0)) ∨ (normContrP=TRUE ∧ SCSLeverFB ≠Backward)) ∧ cruiseControlMode=1 ∧ brakePedal=0 // *SCS-3/SCS16/SCS-17*

inv6 : adapContr=TRUE ⇔ ((SCSLeverFB = Forward ∧ SCSLeverFBP ≠ Forward ∧ (currentSpeed≥speedActiv ∨ desiredSpeed≠0)) ∨ (adapContrP=TRUE ∧ SCSLeverFB ≠Backward)) ∧ cruiseControlMode=2 ∧ brakePedal=0 // *SCS-3/SCS16/SCS-17*

inv7 : SCSLeverUDP≠Upward5 ∧ SCSLeverUD=Upward5 ∧(adapContrP =TRUE ∨ normContrP=TRUE) ∧ trafficdetected=FALSE ⇒ desiredSpeed=min({desiredSpeedMax, desiredSpeedP+10}) // *SCS-4*

inv8 : SCSLeverUDP≠Upward7 ∧ SCSLeverUD=Upward7 ∧ (adapContrP =TRUE ∨ normContrP=TRUE) ∧ trafficdetected=FALSE ⇒ desiredSpeed=min({desiredSpeedMax,(desiredSpeedP÷100)∗100 +100}) // *SCS-5*

inv9 : SCSLeverUDP≠Downward5 ∧ SCSLeverUD=Downward5 ∧ (adapContrP =TRUE ∨ normContrP=TRUE)∧ trafficdetected=FALSE ⇒ desiredSpeed=max({0, desiredSpeedP−10}) // *SC-61*

inv10 : SCSLeverUDP≠Downward7 ∧ SCSLeverUD=Downward7 ∧ (adapContrP =TRUE ∨ normContrP=TRUE) ∧ trafficdetected=FALSE ⇒ desiredSpeed=max({10,(desiredSpeedP÷100)∗100 −100}) // *SC-62*

inv11 : lastTimeSCSLeverUD≠0 ∧ (normContr=TRUE ∨ adapContr=TRUE) ∧ SCSLeverUD=Upward5 ∧ currentTime−lastTimeSCSLeverUD≥2 ∧ trafficdetected=FALSE ⇒ desiredSpeed=min({desiredSpeedMax, lastdesiredSpeed+(currentTime−lastTimeSCSLeverUD−1)∗10}) // *SCS-7*

inv12 : (normContr=TRUE ∨ adapContr=TRUE) ∧ SCSLeverUD=Upward7 ∧ // *SCS-8*

```
                currentTime−lastTimeSCSLeverUD≥2 ∧trafficdetected=FALSE
                 ⇒
                desiredSpeed=min({desiredSpeedMax,
                (lastdesiredSpeed÷100∗100)+((currentTime−lastTimeSCSLeverUD)÷2)∗100})
                (normContr=TRUE ∨ adapContr=TRUE) ∧ SCSLeverUD=Downward5 ∧                    //   SCS-9
    inv13   :   currentTime−lastTimeSCSLeverUD≥2 ∧ trafficdetected=FALSE
                 ⇒
                desiredSpeed=max({10,lastdesiredSpeed−(currentTime−lastTimeSCSLeverUD−1)∗10})
                lastTimeSCSLeverUD≠0 ∧ (normContr=TRUE ∨ adapContr=TRUE) ∧          //   SCS-10
                SCSLeverUD=Downward7 ∧ currentTime−lastTimeSCSLeverUD≥2 ∧
    inv14   :   trafficdetected=FALSE
                 ⇒
                desiredSpeed=max({10,(lastdesiredSpeed÷100)∗100−
                                    ((currentTime−lastTimeSCSLeverUD)÷2)∗100})
                SCSLeverUDP=Neutral ∧ SCSLeverUDP≠Neutral ∧       //   SCS-11
    inv15   :   (normContr=FALSE ∧ adapContr=FALSE)
                 ⇒
                desiredSpeed=currentSpeed
                (normContr=FALSE ∧ adapContr=FALSE) ∨ SCSLeverUD=Neutral
    inv16   :   ⇒
                lastTimeSCSLeverUD=0
    inv17   :   keyState≠KeyInIgnitionOnPosition⇒ currentSpeed=0 ∧ desiredSpeed=0
    inv18   :   normContr=FALSE ⇒ normContrP=FALSE
    inv19   :   adapContr=FALSE ⇒ adapContrP=FALSE
                speedLimiterSwitchOn =TRUE ∧ gasPedal≤gasLimit          //   SCS-32
    inv20   :   ⇒                                                       //   SCS-33
                currentSpeed≤speedLimit                                 //   SCS-34
    inv21   :   lastTimeSCSLeverUD≤currentTime
    inv22   :   detectedTrafficOn=TRUE ⇔adapContr=TRUE ∧ trafficSignDetectionOn=TRUE       //   SCS-36
    inv23   :   detectedTrafficSign∈ ℕ ∧ detectedTrafficSign ≤desiredSpeedMax  ∧
                mandesiredSpeed∈ ℕ ∧ mandesiredSpeed≤desiredSpeedMax
    inv24   :   trafficdetected=TRUE⇒trafficSignDetectionOn=TRUE
                trafficdetected=TRUE ∧ gasPedal>0                                              //   SCS-39
                ⇒
                (
                (detectedTrafficSign<200⇒desiredSpeed=desiredSpeedP)
                ∧
                (detectedTrafficSign∈200‥1300⇒desiredSpeed=detectedTrafficSign)
                ∧
    inv25   :   (detectedTrafficSign>1300⇒
                  (desiredSpeedP<1200⇒desiredSpeed=1200)
                 ∧
                  (desiredSpeedP≥1200 ∧ mandesiredSpeed≥1200⇒desiredSpeed=mandesiredSpeed)
                 ∧
                 (desiredSpeedP≥1200 ∧ mandesiredSpeed=0⇒desiredSpeed=desiredSpeedP)
                )
                )
                trafficdetected=TRUE ∧ gasPedal=0       //   SCS-37
    inv26   :   ⇒
                desiredSpeed=detectedTrafficSign
    inv27   :   (adapContr=TRUE ∨ normContr=TRUE)⇒ keyState=KeyInIgnitionOnPosition
EVENTS
    INITIALISATION   ≙
      extended
    STATUS
      ordinary
    BEGIN
      act1   :   currentSpeed ≔ 0
      act2   :   keyState≔NoKeyInserted
      act3   :   keyStateP≔NoKeyInserted
      act4   :   SCSLeverUD≔Neutral
      act5   :   SCSLeverUDP≔Neutral
      act6   :   SCSLeverFB≔Neutral
      act7   :   SCSLeverFBP≔Neutral
      act8   :   brakePedal≔0
      act9   :   currentTime≔0
      act10   :   gasPedal≔0
      act11   :   speedLimiterSwitchOn≔FALSE
      act12   :   rangeRadarState≔TRUE
      act13   :   nextTest≔0
      act14   :   buttonHead≔FALSE
      act15   :   speedLimit≔0
      act16   :   lastTest≔0
      act17   :   desiredSpeed≔0
      act18   :   desiredSpeedP≔0
      act19   :   lastTimeSCSLeverUD≔0
      act20   :   lastdesiredSpeed≔0
      act21   :   adapContr≔FALSE
      act22   :   adapContrP≔FALSE
      act23   :   normContr≔FALSE
      act24   :   normContrP≔FALSE
      act25   :   detectedTrafficSign≔0
      act26   :   mandesiredSpeed≔0
      act27   :   detectedTrafficOn≔FALSE
      act28   :   trafficdetected≔FALSE
    END

    gaspedal   ≙
      extended
    STATUS
      ordinary
    REFINES
```

```
  gaspedal
ANY
  gas
  sw
  desspeed
WHERE
  grd1   :   brakePedal=0 ∧ keyState=KeyInIgnitionOnPosition
  grd2   :   gas∈gasbrakeRange ∧ gas≠gasPedal
  grd3   :   sw={TRUE↦FALSE, FALSE↦speedLimiterSwitchOn}
             (bool(gas>90))
  grd4   :   desspeed ∈ ℕ ∧ desspeed≤desiredSpeedMax
             lastTimeSCSLeverUD≠0 ∧ (normContr=TRUE ∨ adapContr=TRUE) ∧
             SCSLeverUD=Upward5 ∧
  grd5   :   currentTime−lastTimeSCSLeverUD≥2
              ⇒
             desspeed=min({desiredSpeedMax,lastdesiredSpeed+
             (currentTime−lastTimeSCSLeverUD−1)∗10})
             (normContr=TRUE ∨ adapContr=TRUE) ∧ SCSLeverUD=Upward7 ∧
             currentTime−lastTimeSCSLeverUD≥2
  grd6   :    ⇒
             desspeed=min({desiredSpeedMax,
             lastdesiredSpeed÷100∗100+((currentTime−lastTimeSCSLeverUD)÷2)∗100})
             (normContr=TRUE ∨ adapContr=TRUE) ∧ SCSLeverUD=Downward5 ∧
             currentTime−lastTimeSCSLeverUD≥2
  grd7   :    ⇒
             desspeed=max({10,lastdesiredSpeed−
             (currentTime−lastTimeSCSLeverUD−1)∗10})
             lastTimeSCSLeverUD≠0 ∧ (normContr=TRUE ∨ adapContr=TRUE) ∧
             SCSLeverUD=Downward7 ∧ currentTime−lastTimeSCSLeverUD≥2
  grd8   :    ⇒
             desspeed=max({10,lastdesiredSpeed÷100∗100−
             ((currentTime−lastTimeSCSLeverUD)÷2)∗100})
             speedLimiterSwitchOn =TRUE ∧ gas≤90
  grd9   :    ⇒
              currentSpeed≤speedLimit
THEN
  act1   :   gasPedal≔gas
  act2   :   keyStateP≔keyState
  act3   :   SCSLeverUDP≔SCSLeverUD
  act4   :   SCSLeverFBP≔SCSLeverFB
  act5   :   speedLimiterSwitchOn≔sw
  act6   :   adapContrP≔adapContr
  act7   :   normContrP≔normContr
  act8   :   desiredSpeed≔desspeed
  act9   :   desiredSpeedP≔desiredSpeed
  act10   :   trafficdetected≔FALSE
END

brakepedal   ≙
  extended
STATUS
  ordinary
REFINES
  brakepedal
ANY
  val
WHERE
  grd1   :   keyState=KeyInIgnitionOnPosition
  grd2   :   val∈ gasbrakeRange ∧ val≠brakePedal
  grd3   :   gasPedal=0
THEN
  act1   :   brakePedal≔val
  act2   :   SCSLeverFBP≔SCSLeverFB
  act3   :   normContr≔FALSE
  act4   :   normContrP≔FALSE
  act5   :   adapContr≔FALSE
  act6   :   adapContrP≔FALSE
  act7   :   lastTimeSCSLeverUD≔0
  act8   :   detectedTrafficOn≔FALSE
  act9   :   trafficdetected≔FALSE
END

moveKey   ≙
  extended
STATUS
  ordinary
REFINES
  moveKey
ANY
  valkey
  radstate
WHERE
  grd1   :   currentSpeed=0
  grd2   :   valkey ∈ keyStates
             (keyState=NoKeyInserted ⇒ valkey=KeyInserted)
             ∧
  grd3   :   (keyState=KeyInserted ⇒ valkey∈ {NoKeyInserted, KeyInIgnitionOnPosition})
                ∧
             (keyState=KeyInIgnitionOnPosition ⇒ valkey=KeyInserted)
  grd4   :   radstate∈ BOOL
  grd5   :   valkey≠KeyInIgnitionOnPosition⇒radstate=FALSE
THEN
```

```
  act1   :   keyState≔valkey
  act2   :   keyStateP≔keyState
  act3   :   SCSLeverUDP≔SCSLeverUD
  act4   :   SCSLeverFBP≔SCSLeverFB
  act5   :   rangeRadarState≔radstate
  act6   :   nextTest≔{TRUE↦currentTime+updateDur, FALSE↦0}
             (bool(valkey=KeyInIgnitionOnPosition))
  act7   :   brakePedal≔ {TRUE↦brakePedal, FALSE↦0}
             (bool(keyState≠KeyInIgnitionOnPosition))
  act8   :   gasPedal≔ {TRUE↦gasPedal, FALSE↦0}
             (bool(keyState≠KeyInIgnitionOnPosition))
  act9   :   speedLimiterSwitchOn≔ FALSE
  act10   :    buttonHead≔FALSE
  act11   :    lastTest≔{TRUE↦currentTime,FALSE↦0}(bool(valkey=KeyInIgnitionOnPosition))
  act12   :   desiredSpeed ≔ 0
  act13   :   adapContr≔FALSE
  act14   :   adapContrP≔FALSE
  act15   :   normContr≔FALSE
  act16   :   normContrP≔FALSE
  act17   :   lastTimeSCSLeverUD≔0
  act18   :   detectedTrafficOn≔{TRUE↦FALSE, FALSE↦detectedTrafficOn}
              (bool(valkey≠KeyInIgnitionOnPosition))
  act19   :   trafficdetected≔FALSE
END

moveSCSLeverUD   ≙
  extended
STATUS
  ordinary
REFINES
  moveSCSLeverUD
ANY
  valSCS
  newDesiredSpeed
  trafdet
WHERE
  grd1   :   valSCS ∈  Upwardu Downwardu {Neutral} ∧ valSCS≠SCSLeverUD
  grd2   :   SCSLeverFB=Neutral
             SCSLeverUD ≠ Neutral ⟹
                (SCSLeverUD ∈ Upward ∧ valSCS ∈ Upward)
  grd3   :   ∨
                (SCSLeverUD ∈ Downward ∧ valSCS ∈ Downward)
             ∨
             valSCS = Neutral
             trafdet=
             {
             TRUE↦{
             TRUE↦detectedTrafficSign,
                     FALSE↦{TRUE↦1200,
                            FALSE↦{TRUE↦mandesiredSpeed, FALSE↦desiredSpeed}
                                                              (bool(mandesiredSpeed≥1200))
  grd4   :                        }(bool(desiredSpeed<1200))
             }(bool(detectedTrafficSign∈20‥130)),
             FALSE↦desiredSpeed
             }(bool(
             trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧
             adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition ∧
             detectedTrafficSign≥20
             ))
             newDesiredSpeed={TRUE↦
             {TRUE↦min({desiredSpeedMax,currentSpeed}),
               FALSE ↦{TRUE↦{TRUE↦min({desiredSpeedMax,desiredSpeed+10}),
             FALSE↦min({desiredSpeedMax,(desiredSpeed÷100)∗100 +100})}
  grd5   :             (bool(valSCS= Upward5)),
                   FALSE↦{TRUE↦max({0,desiredSpeed−10}),FALSE↦max({10,(desiredSpeed÷100)∗100−100})}
                                                          (bool(valSCS= Downward5))}(bool(valSCS∈ Upward))
               }(bool(adapContr =FALSE ∧ normContr=FALSE)),
             FALSE↦trafdet}(bool(valSCS≠Neutral))
THEN
  act1   :   SCSLeverUD≔valSCS
  act2   :   SCSLeverUDP≔SCSLeverUD
  act3   :   SCSLeverFBP≔SCSLeverFB
  act4   :   keyStateP≔keyState
  act5   :   desiredSpeed≔newDesiredSpeed
  act6   :   desiredSpeedP≔desiredSpeed
             lastTimeSCSLeverUD≔{TRUE↦0, FALSE↦currentTime}
  act7   :   (bool(
             (normContr=FALSE ∧ adapContr=FALSE) ∨
             valSCS=Neutral))
  act8   :   lastdesiredSpeed≔newDesiredSpeed
  act9   :   mandesiredSpeed≔{TRUE↦newDesiredSpeed, FALSE↦mandesiredSpeed}(bool(newDesiredSpeed≥120))
  act10   :    trafficdetected≔FALSE
END

moveSCSLeverFB   ≙
  extended
STATUS
  ordinary
REFINES
  moveSCSLeverFB
ANY
  valSCS
  sw
```

```
  bh
  newnormCont
  newadapCont
  desSpeed
WHERE
  grd1   :   valSCS ∈ {Backward, Forward, Neutral} ∧ valSCS≠SCSLeverFB
  grd2   :   SCSLeverUD=Neutral
  grd3   :   SCSLeverFB ≠ Neutral ⟹ valSCS = Neutral
  grd4   :   bh={TRUE↦buttonHead, FALSE↦FALSE}(bool(valSCS ≠Backward))
  grd5   :   sw={TRUE↦FALSE, FALSE↦speedLimiterSwitchOn}(bool(bh=FALSE))
  grd6   :   desSpeed ∈  ℕ  ∧desSpeed ≤desiredSpeedMax
  grd7   :   keyState≠KeyInIgnitionOnPosition⟹ desSpeed=0
             valSCS = Forward  ⟹                                                                                    //   Req SCS-2
                         (
  grd8   :               (desiredSpeed=0 ∧ currentSpeed≠0⟹desSpeed=min({desiredSpeedMax,currentSpeed÷10}))
                         ∧
                         (currentSpeed=0 ⟹desSpeed=desiredSpeed)
                         )
             valSCS = Neutral
             ⟹
             desSpeed=
             {
             TRUE↦{
             TRUE↦detectedTrafficSign,
                     FALSE↦{TRUE↦120,
                            FALSE↦{TRUE↦mandesiredSpeed, FALSE↦desiredSpeed}
  grd9   :                                                  (bool(mandesiredSpeed≥120))
                                }(bool(desiredSpeed<120))
             }(bool(detectedTrafficSigne≥20‥130)),
             FALSE↦desiredSpeed
             }(bool(
             trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧ SCSLeverUD=Neutral ∧
             adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition ∧
             detectedTrafficSign≥20
             ))
              newnormCont=bool(
              (
              (valSCS = Forward ∧ cruiseControlMode=1∧  (currentSpeed≥speedActiv ∨ desSpeed≠0))
                ∨
  grd10  :    (normContr=TRUE ∧ valSCS ≠Backward)
              )
               ∧
              brakePedal=0
              )
              newadapCont=bool(
              (
              (valSCS = Forward ∧ cruiseControlMode=2 ∧ (currentSpeed≥speedActiv ∨ desSpeed≠0))
                ∨
  grd11  :    (adapContr=TRUE ∧ valSCS ≠Backward)
              )
               ∧
              brakePedal=0
              )
THEN
  act1   :   SCSLeverFB≔valSCS
  act2   :   SCSLeverFBP≔SCSLeverFB
  act3   :   keyStateP≔keyState
  act4   :   speedLimiterSwitchOn≔sw
  act5   :   buttonHead≔bh
  act6   :   desiredSpeed ≔ desSpeed
  act7   :   desiredSpeedP ≔ desiredSpeed
  act8   :   normContr≔newnormCont
  act9   :   normContrP≔{TRUE↦normContr, FALSE↦FALSE}(newnormCont)
  act10   :   adapContr≔newadapCont
  act11   :   adapContrP≔{TRUE↦adapContr, FALSE↦FALSE}(newadapCont)
  act12   :   mandesiredSpeed≔{TRUE↦desSpeed, FALSE↦mandesiredSpeed}(bool(desSpeed≥120))
  act13   :   detectedTrafficOn≔bool(newadapCont=TRUE ∧ trafficSignDetectionOn=TRUE)
  act14   :   trafficdetected≔FALSE
END

progress   ≙
  extended
STATUS
  ordinary
REFINES
  progress
ANY
  val
  radstate
  despeed
WHERE
  grd1   :   val∈ rangeSpeed
  grd2   :   radstate∈ BOOL
  grd3   :   keyState≠KeyInIgnitionOnPosition ∨ nextTest≠currentTime+1⟹
             radstate=rangeRadarState
  grd4   :   keyState≠KeyInIgnitionOnPosition ⟹ val=0
  grd5   :   despeed∈ ℕ ∧ despeed≤desiredSpeedMax
             (normContr=FALSE ∧ adapContr=FALSE)
  grd6   :   ⟹
              despeed=0
  grd7   :   (normContr=TRUE ∨ adapContr=TRUE) ∧
             (SCSLeverUD=Neutral ∨ currentTime+1−lastTimeSCSLeverUD<2)
             ⟹
```

```
            despeed=desiredSpeed
  grd8   :  speedLimiterSwitchOn =TRUE ∧ gasPedal≤gasLimit ⇒ val≤speedLimit
            (normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
            currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward5
  grd9   :  ⇒
            despeed=min({desiredSpeedMax,lastdesiredSpeed+
            (currentTime+1−lastTimeSCSLeverUD−1)∗10})
            (normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
            SCSLeverUD=Upward7
  grd10  :  ⇒
            despeed=min({desiredSpeedMax,lastdesiredSpeed÷100∗100+
                 ((currentTime+1−lastTimeSCSLeverUD)÷2)∗100})
            (normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
            currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward5
  grd11  :  ⇒
            despeed=max({10,lastdesiredSpeed−
            (currentTime+1−lastTimeSCSLeverUD−1)∗10})
            (normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
             SCSLeverUD=Downward7
  grd12  :  ⇒
            despeed=max({10,(lastdesiredSpeed÷100)∗100−
            ((currentTime+1−lastTimeSCSLeverUD)÷2)∗100})
            trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧ SCSLeverUD=Neutral ∧
            SCSLeverFB=Neutral ∧
            adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition
            ⇒
            (
            (detectedTrafficSign<20⇒despeed=desiredSpeed)
            ∧
            (detectedTrafficSign∈20‥130⇒despeed=detectedTrafficSign)
  grd13  :  ∧
            (detectedTrafficSign>130⇒
              (desiredSpeed<120⇒despeed=120)
             ∧
              (desiredSpeed≥120 ∧ mandesiredSpeed≥120⇒despeed=mandesiredSpeed)
             ∧
             (desiredSpeed≥120 ∧ mandesiredSpeed=0⇒despeed=desiredSpeed)
            )
            )
THEN
  act1   :  currentSpeed≔val
  act2   :  currentTime≔currentTime+1
  act3   :  SCSLeverUDP≔SCSLeverUD
  act4   :  SCSLeverFBP≔SCSLeverFB
  act5   :  keyStateP≔keyState
  act6   :  rangeRadarState≔radstate
  act7   :  nextTest≔{TRUE↦currentTime+1+updateDur, FALSE↦nextTest}
              (bool(keyState=KeyInIgnitionOnPosition  ∧ nextTest=currentTime+1))
  act8   :  lastTest≔{TRUE↦currentTime+1, FALSE↦lastTest}
              (bool(keyState=KeyInIgnitionOnPosition  ∧ nextTest=currentTime+1))
  act9   :  normContrP≔normContr
  act10  :  adapContrP≔adapContr
  act11  :  desiredSpeed≔despeed
  act12  :  desiredSpeedP≔desiredSpeed
  act13  :  trafficdetected≔FALSE
END

pushButtonHead  ≙
  extended
STATUS
  ordinary
REFINES
  pushButtonHead
ANY
  sl
  desspeed
WHERE
  grd1   :  buttonHead=FALSE
  grd2   :  speedLimiterSwitchOn=FALSE ∧ SCSLeverFB≠Backward ∧
            keyState=KeyInIgnitionOnPosition ∧ gasPedal≤90
  grd3   :  keyState=KeyInIgnitionOnPosition ∧ SCSLeverFB ≠ Backward
  grd4   :  currentSpeed≤speedLimit
  grd5   :  sl≤rangeSpeed
  grd6   :  currentSpeed≤sl
  grd7   :  sl≤rangeSpeed
  grd8   :  desiredSpeed≥0⇒  sl=desspeed
  grd9   :  currentSpeed≤sl
  grd10  :  desspeed ∈ ℕ ∧ desspeed≤desiredSpeedMax
            lastTimeSCSLeverUD≠0 ∧ (normContr=TRUE ∨ adapContr=TRUE) ∧
            SCSLeverUD=Upward5 ∧
  grd11  :  currentTime−lastTimeSCSLeverUD≥2
             ⇒
            desspeed=min({desiredSpeedMax,
                        lastdesiredSpeed+(currentTime−lastTimeSCSLeverUD−1)∗10})
            (normContr=TRUE ∨ adapContr=TRUE) ∧ SCSLeverUD=Upward7 ∧
            currentTime−lastTimeSCSLeverUD≥2
  grd12  :   ⇒
            desspeed=min({desiredSpeedMax,lastdesiredSpeed÷100∗100+
                            ((currentTime−lastTimeSCSLeverUD)÷2)∗100})
  grd13  :  (normContr=TRUE ∨ adapContr=TRUE) ∧ SCSLeverUD=Downward5 ∧
            currentTime−lastTimeSCSLeverUD≥2
             ⇒
            desspeed=max({10,lastdesiredSpeed−
```

```
              (currentTime−lastTimeSCSLeverUD−1)∗10})
              lastTimeSCSLeverUD≠0 ∧ (normContr=TRUE ∨ adapContr=TRUE) ∧
              SCSLeverUD=Downward7 ∧ currentTime−lastTimeSCSLeverUD≥2
  grd14   :    ⇒
              desspeed=max({10,lastdesiredSpeed÷100∗100−
              ((currentTime−lastTimeSCSLeverUD)÷2)∗100})
  grd15   :   keyState≠KeyInIgnitionOnPosition⇒ desspeed=0
THEN
  act1    :   speedLimiterSwitchOn≔TRUE
  act2    :   speedLimit≔sl
  act3    :   SCSLeverUDP≔SCSLeverUD
  act4    :   SCSLeverFBP≔SCSLeverFB
  act5    :   keyStateP≔keyState
  act6    :   buttonHead≔TRUE
  act7    :   adapContrP≔adapContr
  act8    :   normContrP≔normContr
  act9    :   trafficdetected≔FALSE
  act10   :    desiredSpeed≔desspeed
END

trafficSignDetection   ≙
  extended
STATUS
  ordinary
REFINES
  trafficSignDetection
ANY
  val
  desSpeed
WHERE
              val∈N ∧ val ≤desiredSpeedMax ∧
  grd1    :   detectedTrafficSign≠val ∧ desSpeede∈ N ∧
              desSpeed≤desiredSpeedMax
  grd2    :   detectedTrafficOn=TRUE
              gasPedal>0
              ⇒(
              (val<200⇒desSpeed=desiredSpeed)
              ∧
              (val∈200‥1300⇒desSpeed=val)
              ∧
              (val>1300⇒(desiredSpeed<1200⇒desSpeed=1200)
  grd3    :    ∧
                (desiredSpeed≥1200 ∧ mandesiredSpeed≥1200⇒
                                desSpeed=mandesiredSpeed)
               ∧
               (desiredSpeed≥1200 ∧ mandesiredSpeed=0⇒desSpeed=desiredSpeed)
              )
              )
              gasPedal=0
  grd4    :   ⇒
              desSpeed=val
THEN
  act1    :   SCSLeverUDP≔SCSLeverUD
  act2    :   SCSLeverFBP≔SCSLeverFB
  act3    :   keyStateP≔keyState
  act4    :   detectedTrafficSign≔val
  act5    :   trafficdetected≔TRUE
  act6    :   normContrP≔normContr
  act7    :   adapContrP≔adapContr
  act8    :   desiredSpeed≔desSpeed
  act9    :   desiredSpeedP≔desiredSpeed
END

END
```

```
CONTEXT
    C3
EXTENDS
    C2
CONSTANTS
    headLevels
    rangeRadarSensorValues
    accel
AXIOMS
    axm1   :   headLevels={0,20,25,30}
    axm2   :   rangeRadarSensorValues={0}∪1‥200∪{255}
    axm3   :   accel={0↦0,50↦15, 100↦30}
END
```

MACHINE
M3
REFINES
M2
SEES
C3
VARIABLES
currentSpeed
desiredSpeed
desiredSpeedP
mandesiredSpeed
keyState
keyStateP
SCSLeverUD
SCSLeverFB
SCSLeverUDP
SCSLeverFBP
adapContr
adapContrP
normContr
normContrP
currentTime
lastTimeSCSLeverUD
lastdesiredSpeed
brakePedal
gasPedal
safetyDistance
securedistanceToHead
rangeRadarSensor
speedOfHead
speedOfHeadP
accVeh
speedLimiterSwitchOn
speedLimit
detectedTrafficSign
detectedTrafficOn
trafficdetected
acousticWarningOn
visualWarningOn
rangeRadarState
nextTest
setVehicleSpeed
buttonHead
lastTest
INVARIANTS

inv1 : safetyDistance ∈headLevels ∧ securedistanceToHead ∈ ℕ ∧ speedOfHead∈ rangeSpeed ∧ speedOfHeadP∈ rangeSpeed ∧ accVeh∈−60‥30 ∧ rangeRadarSensor ∈ rangeRadarSensorValues ∧ setVehicleSpeed∈−60‥30

inv2 : normContr=FALSE ∧ adapContr=FALSE ⟹setVehicleSpeed =0

inv3 : brakePedal≠0 ∧ currentSpeed>0 ⟹ accVeh=max({−60,−(brakePedal∗10)÷375})

inv4 : keyState≠KeyInIgnitionOnPosition ⟹ accVeh = 0

inv5 : currentSpeed<desiredSpeed ∧ ((rangeRadarSensor≠255 ∧ rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0) ∧ adapContr=TRUE ⟹ accVeh∈ 0‥10 // SCS-22

inv6 : currentSpeed=desiredSpeed ∧ ((rangeRadarSensor≠255 ∧ rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0) ∧ adapContr=TRUE ⟹ accVeh=0 // SCS-18

inv7 : speedOfHead≤speedActiv ∧ speedOfHeadP>speedOfHead ∧ speedOfHead≠0 ∧ adapContr=TRUE ∧ rangeRadarSensor∉{0,255} ⟹ securedistanceToHead=25∗(currentSpeed÷36) // SCS23_1

inv8 : speedOfHead= 0 ∧ currentSpeed=0 ∧ adapContr=TRUE ∧ rangeRadarSensor∉{0,255} ⟹ securedistanceToHead=2 // SCS23_2

inv9 : speedOfHeadP<speedOfHead ∧ speedOfHead≠0 ∧ speedOfHead≤speedActiv ∧ adapContr=TRUE ∧ rangeRadarSensor∉{0,255} ⟹ // SCS23_3

```
                securedistanceToHead=3∗(currentSpeed∗10÷36)
            speedOfHead>speedActiv ∧ adapContr=TRUE                          //   SCS24
inv10   :   ⇒
            securedistanceToHead=safetyDistance ∗(currentSpeed∗10÷36)
inv11   :   visualWarningOn∈ BOOL ∧ acousticWarningOn∈ BOOL
            (currentSpeed÷36)∗15< rangeRadarSensor ∧ adapContr=TRUE          //   SCS25
inv12   :   ⇔
            visualWarningOn=TRUE
            (currentSpeed÷36)∗8< rangeRadarSensor ∧ adapContr=TRUE           //   SCS26
inv13   :   ⇔
            acousticWarningOn=TRUE
inv14   :   rangeRadarState=FALSE ⇔rangeRadarSensor=255
inv15   :   currentSpeed=desiredSpeed ∧ normContr=TRUE  ⇒ setVehicleSpeed=0       //   SCS-14
inv16   :   currentSpeed<desiredSpeed ∧ normContr=TRUE  ⇒ setVehicleSpeed∈0‥10       //   SCS-14
            gasPedal>0 ∧ (currentSpeed≠desiredSpeed ∨                                                    //   SCS-
             ((rangeRadarSensor=255 ∨  rangeRadarSensor<securedistanceToHead) ∧ rangeRadarSensor≠0) ∨      15
inv17   :    adapContr=FALSE)
            ⇒
            accVeh=min({30,max({(gasPedal∗10)÷375,setVehicleSpeed})})
            rangeRadarSensor<securedistanceToHead ∧ rangeRadarSensor∉{0,255}       //   SCS-20
            ∧ adapContr=TRUE
inv18   :           ⇒
            accVeh<0 ∧ setVehicleSpeed=0
            brakePedal≠0 ∧ currentSpeed=0
inv19   :   ⇒
            accVeh=0
EVENTS
  INITIALISATION   ≙
    extended
  STATUS
    ordinary
  BEGIN
    act1   :   currentSpeed ≔ 0
    act2   :   keyState≔NoKeyInserted
    act3   :   keyStateP≔NoKeyInserted
    act4   :   SCSLeverUD≔Neutral
    act5   :   SCSLeverUDP≔Neutral
    act6   :   SCSLeverFB≔Neutral
    act7   :   SCSLeverFBP≔Neutral
    act8   :   brakePedal≔0
    act9   :   currentTime≔0
    act10   :   gasPedal≔0
    act11   :   speedLimiterSwitchOn≔FALSE
    act12   :   rangeRadarState≔TRUE
    act13   :   nextTest≔0
    act14   :   buttonHead≔FALSE
    act15   :   speedLimit≔0
    act16   :   lastTest≔0
    act17   :   desiredSpeed≔0
    act18   :   desiredSpeedP≔0
    act19   :   lastTimeSCSLeverUD≔0
    act20   :   lastdesiredSpeed≔0
    act21   :   adapContr≔FALSE
    act22   :   adapContrP≔FALSE
    act23   :   normContr≔FALSE
    act24   :   normContrP≔FALSE
    act25   :   detectedTrafficSign≔0
    act26   :   mandesiredSpeed≔0
    act27   :   detectedTrafficOn≔FALSE
    act28   :   trafficdetected≔FALSE
    act29   :   safetyDistance≔0
    act30   :   securedistanceToHead≔ 0
    act31   :   rangeRadarSensor≔ 0
    act32   :   speedOfHead≔120
    act33   :   speedOfHeadP≔0
    act34   :   accVeh≔0
    act35   :   visualWarningOn≔FALSE
    act36   :   acousticWarningOn≔FALSE
    act37   :   setVehicleSpeed≔0
  END

  brakepedal   ≙
    extended
  STATUS
    ordinary
  REFINES
    brakepedal
```

```
ANY
  val
  valacc
WHERE
  grd1   :   keyState=KeyInIgnitionOnPosition
  grd2   :   val∈ gasbrakeRange ∧ val≠brakePedal
  grd3   :   gasPedal=0
  grd4   :   currentSpeed=0⟹valacc=0
  grd5   :   currentSpeed>0⟹valacc=max({-60,-(val∗10)÷375})
THEN
  act1   :   brakePedal≔val
  act2   :   SCSLeverFBP≔SCSLeverFB
  act3   :   normContr≔FALSE
  act4   :   normContrP≔FALSE
  act5   :   adapContr≔FALSE
  act6   :   adapContrP≔FALSE
  act7   :   lastTimeSCSLeverUD≔0
  act8   :   detectedTrafficOn≔FALSE
  act9   :   trafficdetected≔FALSE
  act10   :   speedOfHeadP≔speedOfHead
  act11   :   accVeh≔valacc
  act12   :   visualWarningOn≔FALSE
  act13   :   acousticWarningOn≔FALSE
  act14   :   setVehicleSpeed≔0
END

gaspedal   ≙
  extended
STATUS
  ordinary
REFINES
  gaspedal
ANY
  gas
  sw
  desspeed
  valacc
  tg
  stv
WHERE
  grd1   :   brakePedal=0 ∧ keyState=KeyInIgnitionOnPosition
  grd2   :   gas∈gasbrakeRange ∧ gas≠gasPedal
  grd3   :   sw={TRUE↦FALSE, FALSE↦speedLimiterSwitchOn}
             (bool(gas>90))
  grd4   :   desspeed ∈ ℕ ∧ desspeed≤desiredSpeedMax
  grd5   :   lastTimeSCSLeverUD≠0 ∧ (normContr=TRUE ∨ adapContr=TRUE) ∧
             SCSLeverUD=Upward5 ∧
             currentTime-lastTimeSCSLeverUD≥2
              ⟹
             desspeed=min({desiredSpeedMax,lastdesiredSpeed+
             (currentTime-lastTimeSCSLeverUD-1)∗10})
  grd6   :   (normContr=TRUE ∨ adapContr=TRUE) ∧ SCSLeverUD=Upward7 ∧
             currentTime-lastTimeSCSLeverUD≥2
              ⟹
             desspeed=min({desiredSpeedMax,
             lastdesiredSpeed÷100∗100+((currentTime-lastTimeSCSLeverUD)÷2)∗100})
  grd7   :   (normContr=TRUE ∨ adapContr=TRUE) ∧ SCSLeverUD=Downward5 ∧
             currentTime-lastTimeSCSLeverUD≥2
              ⟹
             desspeed=max({10,lastdesiredSpeed-
             (currentTime-lastTimeSCSLeverUD-1)∗10})
  grd8   :   lastTimeSCSLeverUD≠0 ∧ (normContr=TRUE ∨ adapContr=TRUE) ∧
             SCSLeverUD=Downward7 ∧ currentTime-lastTimeSCSLeverUD≥2
              ⟹
             desspeed=max({10,lastdesiredSpeed÷100∗100-
             ((currentTime-lastTimeSCSLeverUD)÷2)∗100})
  grd9   :   speedLimiterSwitchOn =TRUE ∧ gas≤90
              ⟹
             currentSpeed≤speedLimit
  grd10   :   valacc∈-60‥30
  grd11   :   gas>0 ⟹valacc=min({30,max({gas∗10 ÷ 375,stv})})
  grd12   :   currentSpeed<desspeed ∧
               ((rangeRadarSensor≠255 ∧
               rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0) ∧
                adapContr=TRUE
                      ⟹
               valacc∈ 0‥10
  grd13   :   currentSpeed=desspeed ∧
```

```
                ((rangeRadarSensor≠255 ∧
                rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0) ∧
                  adapContr=TRUE
                        ⇒
                        valacc=0
  grd14   :   tg∈ℕ
  grd15   :   normContr=TRUE ⇒tg=desiredSpeed
              adapContr=TRUE ∧ rangeRadarSensore∈{0,255}
  grd16   :   ⇒
              tg=desspeed
              adapContr=TRUE ∧ rangeRadarSensore∉{0,255}
  grd17   :            ⇒
              tg=min({speedActiv, speedOfHead÷10})
  grd18   :   stv∈−60‥30
              currentSpeed=desspeed ∧
  grd19   :   normContr=TRUE
              ⇒
              stv=0
              currentSpeed<desspeed ∧
  grd20   :   normContr=TRUE
              ⇒
              stv∈0‥10
  grd21   :   normContr=FALSE ∧ adapContr=FALSE ⇒stv=0
              rangeRadarSensor<securedistanceToHead ∧ rangeRadarSensore∉{0,255}
  grd22   :   ∧ adapContr=TRUE
                        ⇒
              valacc<0 ∧ stv=0
THEN
  act1    :   gasPedal≔gas
  act2    :   keyStateP≔keyState
  act3    :   SCSLeverUDP≔SCSLeverUD
  act4    :   SCSLeverFBP≔SCSLeverFB
  act5    :   speedLimiterSwitchOn≔sw
  act6    :   adapContrP≔adapContr
  act7    :   normContrP≔normContr
  act8    :   desiredSpeed≔desspeed
  act9    :   desiredSpeedP≔desiredSpeed
  act10   :    trafficdetected≔FALSE
  act11   :    speedOfHeadP≔speedOfHead
  act12   :    accVeh≔valacc
  act13   :    setVehicleSpeed≔stv
END

moveKey   ≙
  extended
STATUS
  ordinary
REFINES
  moveKey
ANY
  valkey
  radstate
  rds
  brk
  valacc
WHERE
  grd1    :   currentSpeed=0
  grd2    :   valkey ∈ keyStates
              (keyState=NoKeyInserted ⇒ valkey=KeyInserted)
              ∧
  grd3    :   (keyState=KeyInserted ⇒ valkey∈ {NoKeyInserted, KeyInIgnitionOnPosition})
                 ∧
              (keyState=KeyInIgnitionOnPosition ⇒ valkey=KeyInserted)
  grd4    :   radstate∈ BOOL
  grd5    :   valkey≠KeyInIgnitionOnPosition⇒radstate=FALSE
  grd6    :   brk=max({−50,−(brakePedal∗10)÷375})
  grd7    :   rds∈ rangeRadarSensorValues ∧ (radstate=FALSE⇔rds=255)
  grd8    :   valacc∈−60‥30
  grd9    :   valkey≠KeyInIgnitionOnPosition⇒ valacc=0
  grd10   :    valkey=KeyInIgnitionOnPosition⇒ valacc=accVeh
THEN
  act1    :   keyState≔valkey
  act2    :   keyStateP≔keyState
  act3    :   SCSLeverUDP≔SCSLeverUD
  act4    :   SCSLeverFBP≔SCSLeverFB
  act5    :   rangeRadarState≔radstate
  act6    :   nextTest≔{TRUE↦currentTime+updateDur, FALSE↦0}
              (bool(valkey=KeyInIgnitionOnPosition))
```

```
  act7   :   brakePedal:= {TRUE↦brakePedal, FALSE↦0}
             (bool(keyState≠KeyInIgnitionOnPosition))
  act8   :   gasPedal:= {TRUE↦gasPedal, FALSE↦0}
             (bool(keyState≠KeyInIgnitionOnPosition))
  act9   :   speedLimiterSwitchOn:= FALSE
  act10  :    buttonHead:=FALSE
  act11  :    lastTest:={TRUE↦currentTime,FALSE↦0}(bool(valkey=KeyInIgnitionOnPosition))
  act12  :    desiredSpeed := 0
  act13  :    adapContr:=FALSE
  act14  :    adapContrP:=FALSE
  act15  :    normContr:=FALSE
  act16  :    normContrP:=FALSE
  act17  :    lastTimeSCSLeverUD:=0
  act18  :    detectedTrafficOn:={TRUE↦FALSE, FALSE↦detectedTrafficOn}
              (bool(valkey≠KeyInIgnitionOnPosition))
  act19  :    trafficdetected:=FALSE
  act20  :   accVeh:=valacc
  act21  :   visualWarningOn:=FALSE
  act22  :   acousticWarningOn:=FALSE
  act23  :   rangeRadarSensor:=rds
  act24  :   setVehicleSpeed :=0
END

moveSCSLeverUD   ≙
  extended
STATUS
  ordinary
REFINES
  moveSCSLeverUD
ANY
  valSCS
  newDesiredSpeed
  trafdet
  tag
  stv
  valacc
WHERE
  grd1   :   valSCS ∈  Upwardu Downwardu {Neutral} ∧ valSCS≠SCSLeverUD
  grd2   :   SCSLeverFB=Neutral
             SCSLeverUD ≠ Neutral ⟹
                (SCSLeverUD ∈ Upward ∧ valSCS ∈ Upward)
  grd3   :   v
                (SCSLeverUD ∈ Downward ∧ valSCS ∈ Downward)
             v
             valSCS = Neutral
             trafdet=
             {
             TRUE↦{
             TRUE↦detectedTrafficSign,
                     FALSE↦{TRUE↦1200,
                            FALSE↦{TRUE↦mandesiredSpeed, FALSE↦desiredSpeed}
                                                              (bool(mandesiredSpeed≥1200))
  grd4   :                        }(bool(desiredSpeed<1200))
             }(bool(detectedTrafficSign∈20‥130)),
             FALSE↦desiredSpeed
             }(bool(
             trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧
             adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition ∧
             detectedTrafficSign≥20
             ))
             newDesiredSpeed={TRUE↦
             {TRUE↦min({desiredSpeedMax,currentSpeed}),
               FALSE ↦{TRUE↦{TRUE↦min({desiredSpeedMax,desiredSpeed+10}),
             FALSE↦min({desiredSpeedMax,(desiredSpeed÷100)*100 +100})}
  grd5   :             (bool(valSCS= Upward5)),
                     FALSE↦{TRUE↦max({0,desiredSpeed−10}),FALSE↦max({10,(desiredSpeed÷100)*100−100})}
                                                          (bool(valSCS= Downward5))}(bool(valSCS∈ Upward))
               }(bool(adapContr =FALSE ∧ normContr=FALSE)),
             FALSE↦trafdet}(bool(valSCS≠Neutral))
             (normContr=TRUE ∨ (adapContr=TRUE ∧ rangeRadarSensor∈{0,255}))
  grd6   :   ⟹
             tag=newDesiredSpeed
  grd7   :   stv∈−60 ‥ 30
             currentSpeed < newDesiredSpeed ∧
                ((rangeRadarSensor≠255 ∧  rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0) ∧
  grd8   :        adapContr=TRUE
                  ⟹
             stv∈ 0‥10
  grd9   :   rangeRadarSensor<securedistanceToHead ∧ rangeRadarSensor∉{0,255} ∧ adapContr=TRUE
```

```
                ⇒
            stv=0
  grd10   :   adapContr=FALSE ⇒ stv=0
            ¬ (currentSpeed < newDesiredSpeed ∧
               ((rangeRadarSensor≠255 ∧  rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0))
  grd11   :   ∧ (¬(rangeRadarSensor<securedistanceToHead ∧ rangeRadarSensor∉{0,255}))
            ∧ adapContr=TRUE
            ⇒
            stv=setVehicleSpeed
  grd12   :   keyState≠KeyInIgnitionOnPosition ⇒ valacc = 0
  grd13   :   valacc∈−60‥30
            brakePedal≠0 ⇒
  grd14   :   (currentSpeed ≠0⇒valacc=max({−60,−(brakePedal∗10)÷375}))
            ∧
            (currentSpeed =0⇒valacc=0)
            currentSpeed<newDesiredSpeed ∧
             ((rangeRadarSensor≠255 ∧  rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0) ∧
  grd15   :    adapContr=TRUE
                ⇒
              valacc∈ 0‥10
  grd16   :   valacc∈−60 ‥ 30
            currentSpeed=newDesiredSpeed ∧
             ((rangeRadarSensor≠255 ∧
  grd17   :   rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0) ∧
              adapContr=TRUE
                ⇒
                valacc=0
  grd18   :   currentSpeed=0⇒valacc≥0
            gasPedal>0
  grd19   :   ⇒
            valacc=min({30,max({(gasPedal∗10)÷375,stv})})
            rangeRadarSensor<securedistanceToHead ∧ rangeRadarSensor∉{0,255}
  grd20   :   ∧ adapContr=TRUE
                ⇒
            valacc<0 ∧ stv=0
THEN
  act1    :   SCSLeverUD≔valSCS
  act2    :   SCSLeverUDP≔SCSLeverUD
  act3    :   SCSLeverFBP≔SCSLeverFB
  act4    :   keyStateP≔keyState
  act5    :   desiredSpeed≔newDesiredSpeed
  act6    :   desiredSpeedP≔desiredSpeed
            lastTimeSCSLeverUD≔{TRUE↦0, FALSE↦currentTime}
  act7    :   (bool(
            (normContr=FALSE ∧ adapContr=FALSE) ∨
            valSCS=Neutral))
  act8    :   lastdesiredSpeed≔newDesiredSpeed
  act9    :   mandesiredSpeed≔{TRUE↦newDesiredSpeed, FALSE↦mandesiredSpeed}(bool(newDesiredSpeed≥120))
  act10   :    trafficdetected≔FALSE
  act11   :   accVeh≔valacc
  act12   :   setVehicleSpeed≔stv
END

moveSCSLeverFB   ≙
  extended
STATUS
  ordinary
REFINES
  moveSCSLeverFB
ANY
  valSCS
  sw
  bh
  newnormCont
  newadapCont
  desSpeed
  stv
  valacc
  secdis
WHERE
  grd1    :   valSCS ∈ {Backward, Forward, Neutral} ∧ valSCS≠SCSLeverFB
  grd2    :   SCSLeverUD=Neutral
  grd3    :   SCSLeverFB ≠ Neutral ⇒ valSCS = Neutral
  grd4    :   bh={TRUE↦buttonHead, FALSE↦FALSE}(bool(valSCS ≠Backward))
  grd5    :   sw={TRUE↦FALSE, FALSE↦speedLimiterSwitchOn}(bool(bh=FALSE))
  grd6    :   desSpeed ∈  N  ∧desSpeed ≤desiredSpeedMax
  grd7    :   keyState≠KeyInIgnitionOnPosition⇒ desSpeed=0
  grd8    :   valSCS = Forward  ⇒                                                    //   Req SCS-2
```

```
                    (
                    (desiredSpeed=0 ∧ currentSpeed≠0⟹desSpeed=min
                ({desiredSpeedMax,currentSpeed÷10}))
                    ∧
                    (currentSpeed=0 ⟹desSpeed=desiredSpeed)
                    )
                valSCS = Neutral
                ⟹
                desSpeed=
                {
                TRUE↦{
                TRUE↦detectedTrafficSign,
                        FALSE↦{TRUE↦120,
                             FALSE↦{TRUE↦mandesiredSpeed, FALSE↦desiredSpeed}
grd9   :                                                          (bool(mandesiredSpeed≥120))
                                  }(bool(desiredSpeed<120))
                }(bool(detectedTrafficSign∈20‥130)),
                FALSE↦desiredSpeed
                }(bool(
                trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧ SCSLeverUD=Neutral ∧
                adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition ∧
                detectedTrafficSign≥20
                ))
                 newnormCont=bool(
                 (
                 (valSCS = Forward ∧ cruiseControlMode=1∧  (currentSpeed≥speedActiv ∨ desSpeed≠0))
                   ∨
grd10  :         (normContr=TRUE ∧ valSCS ≠Backward)
                 )
                  ∧
                 brakePedal=0
                 )
                 newadapCont=bool(
                 (
                 (valSCS = Forward ∧ cruiseControlMode=2 ∧ (currentSpeed≥speedActiv ∨ desSpeed≠0))
                   ∨
grd11  :         (adapContr=TRUE ∧ valSCS ≠Backward)
                 )
                  ∧
                 brakePedal=0
                 )
grd12  :         newnormCont=FALSE ∧ newadapCont=FALSE ⟹stv =0
grd13  :         stv∈−60‥30
grd14  :         valacc∈−60 … 30
grd15  :         currentSpeed=0⟹valacc=0
grd16  :         brakePedal≠0 ⟹ valacc=max({−60,−(brakePedal∗10)÷375})
                 currentSpeed=desSpeed ∧
                  ((rangeRadarSensor≠255 ∧  rangeRadarSensor≥secdis) ∨ rangeRadarSensor=0) ∧
grd17  :           newadapCont=TRUE
                         ⟹
                         valacc=0
grd18  :         stv∈−60 … 30
grd19  :         keyState≠KeyInIgnitionOnPosition ⟹ valacc = 0
                 currentSpeed<desSpeed ∧
                  ((rangeRadarSensor≠255 ∧  rangeRadarSensor≥secdis) ∨
                 rangeRadarSensor=0) ∧
grd20  :           newadapCont=TRUE
                         ⟹
                         valacc∈ 0‥10
                 gasPedal≠0
grd21  :         ⟹
                 valacc=min({30,max({(gasPedal∗10)÷375,stv})})
grd22  :         secdis∈ ℕ
                 speedOfHead≤speedActiv ∧ speedOfHeadP>speedOfHead ∧ speedOfHead≠0 ∧
                  newadapCont=TRUE ∧
grd23  :                   rangeRadarSensor∉{0,255}
                           ⟹
                           secdis=25∗(currentSpeed÷36)
                 speedOfHeadP<speedOfHead ∧ speedOfHead≠0 ∧ speedOfHead≤speedActiv  ∧
                 newadapCont=TRUE ∧ rangeRadarSensor∉{0,255}
grd24  :                   ⟹
                           secdis=3∗(currentSpeed∗10÷36)
                 speedOfHead>speedActiv ∧ newadapCont=TRUE
grd25  :         ⟹
                 secdis=safetyDistance ∗(currentSpeed∗10÷36)
                 speedOfHead= 0 ∧ currentSpeed=0 ∧ newadapCont=TRUE  ∧ rangeRadarSensor∉{0,255}
grd26  :         ⟹
                 secdis=2
grd27  :         currentSpeed=desSpeed ∧ newnormCont=TRUE
                  ⟹
```

```
            stv=0
            currentSpeed<desSpeed ∧ newnormCont=TRUE
  grd28  :   ⇒
            stv∈ 0‥10
            rangeRadarSensor<secdis ∧ rangeRadarSensor∉{0,255}
  grd29  :  ∧ newadapCont=TRUE
                      ⇒
            valacc<0 ∧ stv=0
THEN
  act1   :  SCSLeverFB≔valSCS
  act2   :  SCSLeverFBP≔SCSLeverFB
  act3   :  keyStateP≔keyState
  act4   :  speedLimiterSwitchOn≔sw
  act5   :  buttonHead≔bh
  act6   :  desiredSpeed ≔ desSpeed
  act7   :  desiredSpeedP ≔ desiredSpeed
  act8   :  normContr≔newnormCont
  act9   :  normContrP≔{TRUE↦normContr, FALSE↦FALSE}(newnormCont)
  act10  :   adapContr≔newadapCont
  act11  :   adapContrP≔{TRUE↦adapContr, FALSE↦FALSE}(newadapCont)
  act12  :   mandesiredSpeed≔{TRUE↦desSpeed, FALSE↦mandesiredSpeed}(bool(desSpeed≥120))
  act13  :   detectedTrafficOn≔bool(newadapCont=TRUE ∧ trafficSignDetectionOn=TRUE)
  act14  :   trafficdetected≔FALSE
  act15  :   visualWarningOn≔bool((currentSpeed÷36)∗15< rangeRadarSensor ∧ newadapCont=TRUE)
  act16  :   acousticWarningOn≔bool((currentSpeed÷36)∗8< rangeRadarSensor ∧ newadapCont=TRUE)
  act17  :   accVeh≔valacc
  act18  :   setVehicleSpeed≔stv
  act19  :   securedistanceToHead≔secdis
END

progress   ≙
  extended
STATUS
  ordinary
REFINES
  progress
ANY
  val
  radstate
  despeed
  stv
  tag
  rgs
  valacc
  secdis
WHERE
  grd1   :  val∈ rangeSpeed
  grd2   :  radstate∈ BOOL
  grd3   :  keyState≠KeyInIgnitionOnPosition ∨ nextTest≠currentTime+1⇒
            radstate=rangeRadarState
  grd4   :  keyState≠KeyInIgnitionOnPosition ⇒ val=0
  grd5   :  despeed∈ ℕ ∧ despeed≤desiredSpeedMax
            (normContr=FALSE ∧ adapContr=FALSE)
  grd6   :  ⇒
             despeed=0
            (normContr=TRUE ∨ adapContr=TRUE) ∧
  grd7   :  (SCSLeverUD=Neutral ∨ currentTime+1−lastTimeSCSLeverUD<2)
            ⇒
            despeed=desiredSpeed
  grd8   :  speedLimiterSwitchOn =TRUE ∧ gasPedal≤gasLimit ⇒ val≤speedLimit
            (normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
            currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward5
  grd9   :  ⇒
            despeed=min({desiredSpeedMax,lastdesiredSpeed+
            (currentTime+1−lastTimeSCSLeverUD−1)∗10})
             (normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
             SCSLeverUD=Upward7
  grd10  :   ⇒
             despeed=min({desiredSpeedMax,lastdesiredSpeed÷100∗100+
                  ((currentTime+1−lastTimeSCSLeverUD)÷2)∗100})
             (normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
             currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward5
  grd11  :   ⇒
             despeed=max({10,lastdesiredSpeed−
             (currentTime+1−lastTimeSCSLeverUD−1)∗10})
  grd12  :   (normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
              SCSLeverUD=Downward7
             ⇒
```

```
                despeed=max({10,(lastdesiredSpeed÷100)*100−
                ((currentTime+1−lastTimeSCSLeverUD)÷2)*100})
                trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧ SCSLeverUD=Neutral ∧
                SCSLeverFB=Neutral ∧
                adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition
                ⇒
                (
                (detectedTrafficSign<20⇒despeed=desiredSpeed)
                ∧
                (detectedTrafficSign∈20‥130⇒despeed=detectedTrafficSign)
grd13   :       ∧
                (detectedTrafficSign>130⇒
                  (desiredSpeed<120⇒despeed=120)
                 ∧
                  (desiredSpeed≥120 ∧ mandesiredSpeed≥120⇒despeed=mandesiredSpeed)
                 ∧
                 (desiredSpeed≥120 ∧ mandesiredSpeed=0⇒despeed=desiredSpeed)
                )
                )
                secdis∈ ℕ ∧ stv∈−60‥30 ∧ tag∈ ℕ ∧
grd14   :       rgs∈rangeRadarSensorValues ∧
                valacc∈−60‥30
                val=despeed ∧
                 ((rgs≠255 ∧   rgs≥secdis) ∨ rgs=0) ∧
grd15   :        adapContr=TRUE
                       ⇒
                    valacc=0
                brakePedal≠0 ⇒
                 (val≠0⇒ valacc=max({−60,−brakePedal*10 ÷ 375}))
grd16   :       ∧
                 (val=0⇒ valacc=0)
                gasPedal>0 ∧ (val≠despeed ∨
                 ((rgs=255 ∨   rgs<secdis) ∧ rgs≠0) ∨  adapContr=FALSE)
grd17   :       ⇒
                valacc=min({30,max({(gasPedal*10)÷375,stv})})
                val<despeed ∧
                 ((rgs≠255 ∧  rgs≥secdis) ∨ rgs=0) ∧ adapContr=TRUE
grd18   :              ⇒
                      valacc∈ 0‥10
                accVeh≥0 ∧ speedLimiterSwitchOn=FALSE
                ⇒
                ((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⇒
grd19   :            val=min({desiredSpeed,(accVeh+currentSpeed*10÷36)*36÷10})) ∧
                ((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0⇒
                     val=(accVeh+currentSpeed*10÷36)*36÷10)
                accVeh≥0∧ speedLimiterSwitchOn=TRUE
                ⇒
grd20   :       ((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⇒
                val=min({speedLimit,desiredSpeed,(accVeh+currentSpeed*10÷36)*36÷10})) ∧
                ((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0 ⇒
                val=min({speedLimit,(accVeh+currentSpeed*10÷36)*36÷10}))
grd21   :       accVeh<0⇒val=max({0,(accVeh+currentSpeed*10÷36)*36÷10})
grd22   :       normContr=FALSE ∧ adapContr=FALSE ⇒stv =0
grd23   :       keyState≠KeyInIgnitionOnPosition ⇒ valacc = 0
                speedOfHead≤speedActiv ∧ speedOfHeadP>speedOfHead ∧
                speedOfHead≠0 ∧ adapContr=TRUE ∧
grd24   :                rgs∉{0,255}
                         ⇒
                         secdis=25*(val÷36)
                speedOfHead= 0 ∧ val=0 ∧ adapContr=TRUE  ∧ rgs∉{0,255}
grd25   :                ⇒
                secdis=2
                speedOfHeadP<speedOfHead ∧ speedOfHead≠0 ∧
                speedOfHead≤speedActiv  ∧ adapContr=TRUE ∧
grd26   :                rgs∉{0,255}
                         ⇒
                secdis=3*(val*10÷36)
                speedOfHead>speedActiv ∧ adapContr=TRUE
grd27   :       ⇒
                secdis=safetyDistance *(val*10÷36)
grd28   :       radstate=FALSE ⇔rgs=255
                val=despeed ∧ normContr=TRUE
grd29   :        ⇒
                stv=0
                val<despeed ∧ normContr=TRUE
grd30   :        ⇒
                stv∈ 0‥10
grd31   :       rgs<secdis ∧ rgs∉{0,255}
                ∧ adapContr=TRUE
```

```
                    ⇒
             valacc<0 ∧ stv=0
THEN
  act1   :   currentSpeed≔val
  act2   :   currentTime≔currentTime+1
  act3   :   SCSLeverUDP≔SCSLeverUD
  act4   :   SCSLeverFBP≔SCSLeverFB
  act5   :   keyStateP≔keyState
  act6   :   rangeRadarState≔radstate
  act7   :   nextTest≔{TRUE↦currentTime+1+updateDur, FALSE↦nextTest}
               (bool(keyState=KeyInIgnitionOnPosition  ∧ nextTest=currentTime+1))
  act8   :   lastTest≔{TRUE↦currentTime+1, FALSE↦lastTest}
               (bool(keyState=KeyInIgnitionOnPosition  ∧ nextTest=currentTime+1))
  act9   :   normContrP≔normContr
  act10   :   adapContrP≔adapContr
  act11   :   desiredSpeed≔despeed
  act12   :   desiredSpeedP≔desiredSpeed
  act13   :   trafficdetected≔FALSE
  act14   :   securedistanceToHead≔secdis
  act15   :   accVeh≔valacc
  act16   :   visualWarningOn≔bool((val÷36)∗15< rgs ∧ adapContr=TRUE)
  act17   :   acousticWarningOn≔bool((val÷36)∗8< rgs ∧ adapContr=TRUE)
  act18   :   rangeRadarSensor≔rgs
  act19   :   setVehicleSpeed≔stv
END

turnHead   ≙
STATUS
  ordinary
ANY
  val
WHERE
  grd1   :   val∈ headLevels ∧ val≠safetyDistance
             speedOfHead>speedActiv ∧ adapContr=TRUE
  grd2   :   ⇒
             securedistanceToHead=val ∗(currentSpeed∗10÷36)
THEN
  act1   :   safetyDistance ≔val
END

pushButtonHead   ≙
  extended
STATUS
  ordinary
REFINES
  pushButtonHead
ANY
  sl
  desspeed
  stv
WHERE
  grd1   :   buttonHead=FALSE
  grd2   :   speedLimiterSwitchOn=FALSE ∧ SCSLeverFB≠Backward ∧
             keyState=KeyInIgnitionOnPosition ∧ gasPedal≤90
  grd3   :   keyState=KeyInIgnitionOnPosition ∧ SCSLeverFB ≠ Backward
  grd4   :   currentSpeed≤speedLimit
  grd5   :   sl∈rangeSpeed
  grd6   :   currentSpeed≤sl
  grd7   :   sl∈rangeSpeed
  grd8   :   desiredSpeed≥0⇒  sl=desspeed
  grd9   :   currentSpeed≤sl
  grd10   :   desspeed ∈ ℕ ∧ desspeed≤desiredSpeedMax
             lastTimeSCSLeverUD≠0 ∧ (normContr=TRUE ∨ adapContr=TRUE) ∧
             SCSLeverUD=Upward5 ∧
  grd11   :  currentTime−lastTimeSCSLeverUD≥2
              ⇒
             desspeed=min({desiredSpeedMax,
                          lastdesiredSpeed+(currentTime−lastTimeSCSLeverUD−1)∗10})
             (normContr=TRUE ∨ adapContr=TRUE) ∧ SCSLeverUD=Upward7 ∧
             currentTime−lastTimeSCSLeverUD≥2
  grd12   :   ⇒
             desspeed=min({desiredSpeedMax,lastdesiredSpeed÷100∗100+
                           ((currentTime−lastTimeSCSLeverUD)÷2)∗100})
             (normContr=TRUE ∨ adapContr=TRUE) ∧ SCSLeverUD=Downward5 ∧
             currentTime−lastTimeSCSLeverUD≥2
  grd13   :   ⇒
             desspeed=max({10,lastdesiredSpeed−
             (currentTime−lastTimeSCSLeverUD−1)∗10})
```

```
  grd14   :   lastTimeSCSLeverUD≠0 ∧ (normContr=TRUE ∨ adapContr=TRUE) ∧
              SCSLeverUD=Downward7 ∧ currentTime−lastTimeSCSLeverUD≥2
               ⇒
              desspeed=max({10,lastdesiredSpeed÷100∗100−
              ((currentTime−lastTimeSCSLeverUD)÷2)∗100})
  grd15   :   keyState≠KeyInIgnitionOnPosition⇒ desspeed=0
              currentSpeed<desspeed ∧
               ((rangeRadarSensor≠255 ∧
  grd16   :   rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0) ∧
                adapContr=TRUE
                      ⇒
                      accVeh∈ 0‥10
              currentSpeed=desspeed ∧
               ((rangeRadarSensor≠255 ∧  rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0) ∧
  grd17   :    adapContr=TRUE
                      ⇒
                      accVeh=0
  grd18   :   stv∈ −60‥30
              currentSpeed=desspeed ∧ normContr=TRUE
  grd19   :    ⇒
              stv=0
              currentSpeed<desspeed ∧ normContr=TRUE
  grd20   :    ⇒
              stv∈ 0‥10
  grd21   :   normContr=FALSE ∧ adapContr=FALSE ⇒stv =0
              gasPedal>0
  grd22   :   ⇒
              accVeh=min({30,max({(gasPedal∗10)÷375,stv})})
              rangeRadarSensor<securedistanceToHead ∧
              rangeRadarSensor∉{0,255}
  grd23   :   ∧ adapContr=TRUE
                       ⇒
              accVeh<0 ∧ stv=0
THEN
  act1    :   speedLimiterSwitchOn≔TRUE
  act2    :   speedLimit≔sl
  act3    :   SCSLeverUDP≔SCSLeverUD
  act4    :   SCSLeverFBP≔SCSLeverFB
  act5    :   keyStateP≔keyState
  act6    :   buttonHead≔TRUE
  act7    :   adapContrP≔adapContr
  act8    :   normContrP≔normContr
  act9    :   trafficdetected≔FALSE
  act10   :    desiredSpeed≔desspeed
  act11   :   setVehicleSpeed≔stv
END

trafficSignDetection   ≙
  extended
STATUS
  ordinary
REFINES
  trafficSignDetection
ANY
  val
  desSpeed
  accval
WHERE
              val∈N ∧ val ≤desiredSpeedMax ∧
  grd1    :   detectedTrafficSign≠val ∧ desSpeed∈ N ∧
              desSpeed≤desiredSpeedMax
  grd2    :   detectedTrafficOn=TRUE
              gasPedal>0
              ⇒(
              (val<200⇒desSpeed=desiredSpeed)
              ∧
              (val∈200‥1300⇒desSpeed=val)
              ∧
              (val>1300⇒(desiredSpeed<1200⇒desSpeed=1200)
               ∧
                (desiredSpeed≥1200 ∧ mandesiredSpeed≥1200⇒
                                 desSpeed=mandesiredSpeed)
               ∧
               (desiredSpeed≥1200 ∧ mandesiredSpeed=0⇒desSpeed=desiredSpeed)
              )
              )
              gasPedal=0
  grd4    :   ⇒
              desSpeed=val
```

```
  grd5   :   accval∈−60‥
             30
             gasPedal≠0
  grd6   :   ⇒
             accval=min({30,max({(gasPedal∗10)÷375,setVehicleSpeed})})
             currentSpeed<desSpeed ∧
              ((rangeRadarSensor≠255 ∧  rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0) ∧
  grd7   :     adapContr=TRUE
                     ⇒
               accval∈ 0‥10
             currentSpeed=desSpeed ∧
              ((rangeRadarSensor≠255 ∧  rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0) ∧
  grd8   :     adapContr=TRUE
                     ⇒
              accval=0
             rangeRadarSensor<securedistanceToHead ∧
             rangeRadarSensor∉{0,255}
  grd9   :   ∧ adapContr=TRUE
                      ⇒
             accval<0 ∧ setVehicleSpeed=0
THEN
  act1   :   SCSLeverUDP≔SCSLeverUD
  act2   :   SCSLeverFBP≔SCSLeverFB
  act3   :   keyStateP≔keyState
  act4   :   detectedTrafficSign≔val
  act5   :   trafficdetected≔TRUE
  act6   :   normContrP≔normContr
  act7   :   adapContrP≔adapContr
  act8   :   desiredSpeed≔desSpeed
  act9   :   desiredSpeedP≔desiredSpeed
  act10   :   accVeh≔accval
END

VehicHeadDetect   ≙
STATUS
  ordinary
ANY
  val
  stv
  secdis
  speh
  accv
WHERE
  grd1   :   secdis∈ℕ ∧ stv∈ −60‥30  ∧ accv∈ −60‥30
             speh>200 ∧ adapContr=TRUE
  grd2   :                ⇒
                          secdis=safetyDistance ∗(currentSpeed∗10÷36)
             bool(
                          (speh≤200 ∧ speedOfHead>speh ∧ speh≠0 ∧ adapContr=TRUE ∧ val∉{0,255})
                          ∨
                          (speh= 0 ∧ currentSpeed=0 ∧ adapContr=TRUE  ∧ val∉{0,255})
                          ∨
  grd3   :                (speedOfHead<speh ∧ speh≠0 ∧ speh≤200  ∧ adapContr=TRUE ∧ val∉{0,255})
                          ∨
                          (speh>200 ∧ adapContr=TRUE)
                          )=FALSE
                          ⇒
                          secdis=securedistanceToHead
             adapContr=TRUE ∧ val∉{0,255}
  grd4   :                ⇒
                          speh∈0‥2000
             adapContr=FALSE ∨ val∈{0,255}
  grd5   :                ⇒
                          speh=speedOfHead
             val ∈ rangeRadarSensorValues ∧
  grd6   :                (rangeRadarState=FALSE ⇔val=255)
  grd7   :   speh∈rangeSpeed
             currentSpeed<desiredSpeed ∧
                         ((val≠255 ∧  val≥secdis) ∨ val=0) ∧
  grd8   :               adapContr=TRUE
                         ⇒
                         stv∈ 0‥10
  grd9   :   adapContr=FALSE⇒stv=0
  grd10   :   bool(
                          (currentSpeed<desiredSpeed ∧
                          ((val≠255 ∧  val≥secdis) ∨ val=0) ∧ adapContr=TRUE)
                          ∨
                          (adapContr=FALSE)) =FALSE
                          ⇒
```

```
                    stv=setVehicleSpeed
               val<secdis ∧ val∉{0,255} ∧ adapContr=TRUE
  grd11   :         ⇒
                    brakePedal∈−30‥−1 ∧ stv=0
               speh≤speedActiv ∧ speedOfHead>speh ∧ speh≠0 ∧ adapContr=TRUE ∧
                    val∉{0,255}
  grd12   :         ⇒
                    secdis=25∗(currentSpeed÷36)
               speh= 0 ∧ currentSpeed=0 ∧ adapContr=TRUE  ∧ val∉{0,255}
  grd13   :         ⇒
                    secdis=2
               speedOfHead<speh ∧ speh≠0 ∧ speh≤speedActiv
                 ∧ adapContr=TRUE ∧ val∉{0,255}
  grd14   :          ⇒
                     secdis=3∗(currentSpeed∗10÷36)
               currentSpeed<desiredSpeed ∧
                       ((val≠255 ∧  val≥securedistanceToHead) ∨
                       val=0) ∧
  grd15   :            adapContr=TRUE
                       ⇒
                       accv∈ 0‥10
               brakePedal≠0 ∧ currentSpeed=0
  grd16   :          ⇒
               accv=0
               brakePedal≠0 ∧ currentSpeed>0
  grd17   :          ⇒
                       accv=max({−60,−(brakePedal∗10)÷375})
  grd18   :    keyState≠KeyInIgnitionOnPosition ⇒ accv = 0
               currentSpeed<desiredSpeed ∧
                       ((val≠255 ∧  val≥secdis) ∨
                       val=0) ∧
  grd19   :            adapContr=TRUE
                       ⇒
                       accv∈ 0‥10
               currentSpeed=desiredSpeed ∧
                       ((val≠255 ∧ val≥secdis) ∨ val=0) ∧
  grd20   :            adapContr=TRUE
                       ⇒
                       accv=0
               speh>speedActiv ∧ adapContr=TRUE
  grd21   :    ⇒
               secdis=safetyDistance ∗(currentSpeed∗10÷36)
               gasPedal>0 ∧ (currentSpeed≠desiredSpeed ∨
                ((val=255 ∨  val<secdis) ∧ val≠0) ∨
  grd22   :     adapContr=FALSE)
               ⇒
               accv=min({30,max({(gasPedal∗10)÷375,stv})})
               val<secdis ∧ val∉{0,255} ∧ adapContr=TRUE
  grd23   :          ⇒
               accv<0 ∧ stv=0
  THEN
    act1   :   acousticWarningOn≔bool((currentSpeed÷36)∗8<val ∧ adapContr=TRUE)
    act2   :   speedOfHead≔speh
    act3   :   speedOfHeadP≔speedOfHead
    act4   :   setVehicleSpeed≔stv
    act5   :   accVeh≔accv
    act6   :   securedistanceToHead≔secdis
    act7   :   rangeRadarSensor ≔ val
    act8   :   visualWarningOn≔bool((currentSpeed÷36)∗15<val ∧ adapContr=TRUE)
  END

END
```

```
MACHINE
    M4
REFINES
    M3
SEES
    C3
VARIABLES
    currentSpeed
    desiredSpeed
    desiredSpeedP
    mandesiredSpeed
    keyState
    keyStateP
    SCSLeverUD
    SCSLeverFB
    SCSLeverUDP
    SCSLeverFBP
    adapContr
    adapContrP
    normContr
    normContrP
    currentTime
    lastTimeSCSLeverUD
    lastdesiredSpeed
    brakePedal
    gasPedal
    safetyDistance
    securedistanceToHead
    rangeRadarSensor
    speedOfHead
    speedOfHeadP
    accVeh
    speedLimiterSwitchOn
    speedLimit
    detectedTrafficSign
    detectedTrafficOn
    trafficdetected
    acousticWarningOn
    visualWarningOn
    rangeRadarState
    nextTest
    setVehicleSpeed
    buttonHead
    lastTest
INVARIANTS
    inv1   :   SCSLeverUD∈Upward ∪Downward ∪{Neutral}
EVENTS
    INITIALISATION   ≙
      extended
    STATUS
      ordinary
    BEGIN
      act1   :   currentSpeed ≔ 0
      act2   :   keyState≔NoKeyInserted
      act3   :   keyStateP≔NoKeyInserted
      act4   :   SCSLeverUD≔Neutral
      act5   :   SCSLeverUDP≔Neutral
      act6   :   SCSLeverFB≔Neutral
      act7   :   SCSLeverFBP≔Neutral
      act8   :   brakePedal≔0
      act9   :   currentTime≔0
      act10   :   gasPedal≔0
      act11   :   speedLimiterSwitchOn≔FALSE
      act12   :   rangeRadarState≔TRUE
      act13   :   nextTest≔0
      act14   :   buttonHead≔FALSE
      act15   :   speedLimit≔0
      act16   :   lastTest≔0
      act17   :   desiredSpeed≔0
      act18   :   desiredSpeedP≔0
      act19   :   lastTimeSCSLeverUD≔0
      act20   :   lastdesiredSpeed≔0
```

```
  act21   :   adapContr≔FALSE
  act22   :   adapContrP≔FALSE
  act23   :   normContr≔FALSE
  act24   :   normContrP≔FALSE
  act25   :   detectedTrafficSign≔0
  act26   :   mandesiredSpeed≔0
  act27   :   detectedTrafficOn≔FALSE
  act28   :   trafficdetected≔FALSE
  act29   :   safetyDistance≔0
  act30   :   securedistanceToHead≔ 0
  act31   :   rangeRadarSensor≔ 0
  act32   :   speedOfHead≔120
  act33   :   speedOfHeadP≔0
  act34   :   accVeh≔0
  act35   :   visualWarningOn≔FALSE
  act36   :   acousticWarningOn≔FALSE
  act37   :   setVehicleSpeed≔0
END

brakepedal   ≙
  extended
STATUS
  ordinary
REFINES
  brakepedal
ANY
  val
  valacc
WHERE
  grd1   :   keyState=KeyInIgnitionOnPosition
  grd2   :   val∈ gasbrakeRange ∧ val≠brakePedal
  grd3   :   gasPedal=0
  grd4   :   currentSpeed=0⟹valacc=0
  grd5   :   currentSpeed>0⟹valacc=max({-60,-(val∗10)÷375})
THEN
  act1   :   brakePedal≔val
  act2   :   SCSLeverFBP≔SCSLeverFB
  act3   :   normContr≔FALSE
  act4   :   normContrP≔FALSE
  act5   :   adapContr≔FALSE
  act6   :   adapContrP≔FALSE
  act7   :   lastTimeSCSLeverUD≔0
  act8   :   detectedTrafficOn≔FALSE
  act9   :   trafficdetected≔FALSE
  act10   :   speedOfHeadP≔speedOfHead
  act11   :   accVeh≔valacc
  act12   :   visualWarningOn≔FALSE
  act13   :   acousticWarningOn≔FALSE
  act14   :   setVehicleSpeed≔0
END

gaspedal   ≙
  extended
STATUS
  ordinary
REFINES
  gaspedal
ANY
  gas
  sw
  desspeed
  valacc
  tg
  stv
WHERE
  grd1   :   brakePedal=0 ∧ keyState=KeyInIgnitionOnPosition
  grd2   :   gas∈gasbrakeRange ∧ gas≠gasPedal
  grd3   :   sw={TRUE↦FALSE, FALSE↦speedLimiterSwitchOn}
             (bool(gas>90))
  grd4   :   desspeed ∈ ℕ ∧ desspeed≤desiredSpeedMax
  grd5   :   lastTimeSCSLeverUD≠0 ∧ (normContr=TRUE ∨ adapContr=TRUE) ∧
             SCSLeverUD=Upward5 ∧
             currentTime−lastTimeSCSLeverUD≥2
              ⟹
             desspeed=min({desiredSpeedMax,lastdesiredSpeed+
```

```
            (currentTime−lastTimeSCSLeverUD−1)∗10})
            (normContr=TRUE ∨ adapContr=TRUE) ∧ SCSLeverUD=Upward7 ∧
            currentTime−lastTimeSCSLeverUD≥2
  grd6   :   ⟹
            desspeed=min({desiredSpeedMax,
            lastdesiredSpeed÷100∗100+((currentTime−lastTimeSCSLeverUD)÷2)∗100})
            (normContr=TRUE ∨ adapContr=TRUE) ∧ SCSLeverUD=Downward5 ∧
            currentTime−lastTimeSCSLeverUD≥2
  grd7   :   ⟹
            desspeed=max({10,lastdesiredSpeed−
            (currentTime−lastTimeSCSLeverUD−1)∗10})
            lastTimeSCSLeverUD≠0 ∧ (normContr=TRUE ∨ adapContr=TRUE) ∧
            SCSLeverUD=Downward7 ∧ currentTime−lastTimeSCSLeverUD≥2
  grd8   :   ⟹
            desspeed=max({10,lastdesiredSpeed÷100∗100−
            ((currentTime−lastTimeSCSLeverUD)÷2)∗100})
            speedLimiterSwitchOn =TRUE ∧ gas≤90
  grd9   :   ⟹
            currentSpeed≤speedLimit
  grd10  :  valacc∈−60‥30
  grd11  :  gas>0 ⟹valacc=min({30,max({gas∗10 ÷ 375,stv})})
            currentSpeed<desspeed ∧
             ((rangeRadarSensor≠255 ∧
  grd12  :   rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0) ∧
              adapContr=TRUE
                   ⟹
            valacc∈ 0‥10
            currentSpeed=desspeed ∧
             ((rangeRadarSensor≠255 ∧
  grd13  :  rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0) ∧
              adapContr=TRUE
                   ⟹
                   valacc=0
  grd14  :  tg∈ℕ
  grd15  :  normContr=TRUE ⟹tg=desiredSpeed
            adapContr=TRUE ∧ rangeRadarSensore∈{0,255}
  grd16  :  ⟹
            tg=desspeed
            adapContr=TRUE ∧ rangeRadarSensore∉{0,255}
  grd17  :          ⟹
            tg=min({speedActiv, speedOfHead÷10})
  grd18  :  stv∈−60‥30
            currentSpeed=desspeed ∧
  grd19  :  normContr=TRUE
            ⟹
            stv=0
            currentSpeed<desspeed ∧
  grd20  :  normContr=TRUE
            ⟹
            stv∈0‥10
  grd21  :  normContr=FALSE ∧ adapContr=FALSE ⟹stv=0
            rangeRadarSensor<securedistanceToHead ∧ rangeRadarSensore∉{0,255}
  grd22  :  ∧ adapContr=TRUE
                   ⟹
            valacc<0 ∧ stv=0
THEN
  act1   :  gasPedal≔gas
  act2   :  keyStateP≔keyState
  act3   :  SCSLeverUDP≔SCSLeverUD
  act4   :  SCSLeverFBP≔SCSLeverFB
  act5   :  speedLimiterSwitchOn≔sw
  act6   :  adapContrP≔adapContr
  act7   :  normContrP≔normContr
  act8   :  desiredSpeed≔desspeed
  act9   :  desiredSpeedP≔desiredSpeed
  act10  :   trafficdetected≔FALSE
  act11  :   speedOfHeadP≔speedOfHead
  act12  :   accVeh≔valacc
  act13  :   setVehicleSpeed≔stv
END

moveKey   ≙
  extended
STATUS
  ordinary
REFINES
```

```
  moveKey
ANY
  valkey
  radstate
  rds
  brk
  valacc
WHERE
  grd1   :   currentSpeed=0
  grd2   :   valkey ∈ keyStates
             (keyState=NoKeyInserted ⇒ valkey=KeyInserted)
             ∧
  grd3   :   (keyState=KeyInserted ⇒ valkey∈ {NoKeyInserted, KeyInIgnitionOnPosition})
                ∧
             (keyState=KeyInIgnitionOnPosition ⇒ valkey=KeyInserted)
  grd4   :   radstate∈ BOOL
  grd5   :   valkey≠KeyInIgnitionOnPosition⇒radstate=FALSE
  grd6   :   brk=max({-50,-(brakePedal∗10)÷375})
  grd7   :   rds∈ rangeRadarSensorValues ∧ (radstate=FALSE⇔rds=255)
  grd8   :   valacc∈-60‥30
  grd9   :   valkey≠KeyInIgnitionOnPosition⇒ valacc=0
  grd10   :    valkey=KeyInIgnitionOnPosition⇒ valacc=accVeh
THEN
  act1   :   keyState≔valkey
  act2   :   keyStateP≔keyState
  act3   :   SCSLeverUDP≔SCSLeverUD
  act4   :   SCSLeverFBP≔SCSLeverFB
  act5   :   rangeRadarState≔radstate
  act6   :   nextTest≔{TRUE↦currentTime+updateDur, FALSE↦0}
             (bool(valkey=KeyInIgnitionOnPosition))
  act7   :   brakePedal≔ {TRUE↦brakePedal, FALSE↦0}
             (bool(keyState≠KeyInIgnitionOnPosition))
  act8   :   gasPedal≔ {TRUE↦gasPedal, FALSE↦0}
             (bool(keyState≠KeyInIgnitionOnPosition))
  act9   :   speedLimiterSwitchOn≔ FALSE
  act10   :    buttonHead≔FALSE
  act11   :    lastTest≔{TRUE↦currentTime,FALSE↦0}(bool(valkey=KeyInIgnitionOnPosition))
  act12   :    desiredSpeed ≔ 0
  act13   :    adapContr≔FALSE
  act14   :    adapContrP≔FALSE
  act15   :    normContr≔FALSE
  act16   :    normContrP≔FALSE
  act17   :    lastTimeSCSLeverUD≔0
  act18   :    detectedTrafficOn≔{TRUE↦FALSE, FALSE↦detectedTrafficOn}
               (bool(valkey≠KeyInIgnitionOnPosition))
  act19   :    trafficdetected≔FALSE
  act20   :    accVeh≔valacc
  act21   :    visualWarningOn≔FALSE
  act22   :    acousticWarningOn≔FALSE
  act23   :    rangeRadarSensor≔rds
  act24   :    setVehicleSpeed ≔0
END

moveSCSLeverUD   ≙
  extended
STATUS
  ordinary
REFINES
  moveSCSLeverUD
ANY
  valSCS
  newDesiredSpeed
  trafdet
  tag
  stv
  valacc
WHERE
  grd1   :   valSCS ∈  Upwardu Downwardu {Neutral} ∧ valSCS≠SCSLeverUD
  grd2   :   SCSLeverFB=Neutral
             SCSLeverUD ≠ Neutral ⇒
                (SCSLeverUD ∈ Upward ∧ valSCS ∈ Upward)
             ∨
  grd3   :      (SCSLeverUD ∈ Downward ∧ valSCS ∈ Downward)
             ∨
             valSCS = Neutral
```

```
grd4   :  trafdet=
          {
          TRUE↦{
          TRUE↦detectedTrafficSign,
                  FALSE↦{TRUE↦1200,
                         FALSE↦{TRUE↦mandesiredSpeed, FALSE↦desiredSpeed}
                                                              (bool(mandesiredSpeed≥1200))
                               }(bool(desiredSpeed<1200))
          }(bool(detectedTrafficSign∈20‥130)),
          FALSE↦desiredSpeed
          }(bool(
          trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧
          adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition ∧
          detectedTrafficSign≥20
          ))
          newDesiredSpeed={TRUE↦
          {TRUE↦min({desiredSpeedMax,currentSpeed}),
            FALSE ↦{TRUE↦{TRUE↦min({desiredSpeedMax,desiredSpeed+10}),
          FALSE↦min({desiredSpeedMax,(desiredSpeed÷100)*100 +100})}
grd5   :            (bool(valSCS= Upward5)),
                FALSE↦{TRUE↦max({0,desiredSpeed−10}),FALSE↦max({10,(desiredSpeed÷100)*100−100})}
                                                        (bool(valSCS= Downward5))}(bool(valSCS∈ Upward))
            }(bool(adapContr =FALSE ∧ normContr=FALSE)),
          FALSE↦trafdet}(bool(valSCS≠Neutral))
          (normContr=TRUE ∨ (adapContr=TRUE ∧ rangeRadarSensor∈{0,255}))
grd6   :  ⇒
          tag=newDesiredSpeed
grd7   :  stv∈−60 ‥ 30
          currentSpeed < newDesiredSpeed ∧
             ((rangeRadarSensor≠255 ∧  rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0) ∧
grd8   :       adapContr=TRUE
               ⇒
          stv∈ 0‥10
          rangeRadarSensor<securedistanceToHead ∧ rangeRadarSensor∉{0,255} ∧ adapContr=TRUE
grd9   :       ⇒
           stv=0
grd10  :   adapContr=FALSE ⇒ stv=0
           ¬ (currentSpeed < newDesiredSpeed ∧
              ((rangeRadarSensor≠255 ∧  rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0))
grd11  :   ∧ (¬(rangeRadarSensor<securedistanceToHead ∧ rangeRadarSensor∉{0,255}))
           ∧ adapContr=TRUE
           ⇒
           stv=setVehicleSpeed
grd12  :   keyState≠KeyInIgnitionOnPosition ⇒ valacc = 0
grd13  :   valacc∈−60‥30
           brakePedal≠0 ⇒
grd14  :   (currentSpeed ≠0⇒valacc=max({−60,−(brakePedal*10)÷375}))
           ∧
           (currentSpeed =0⇒valacc=0)
           currentSpeed<newDesiredSpeed ∧
            ((rangeRadarSensor≠255 ∧  rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0) ∧
grd15  :     adapContr=TRUE
               ⇒
           valacc∈ 0‥10
grd16  :   valacc∈−60 ‥ 30
           currentSpeed=newDesiredSpeed ∧
            ((rangeRadarSensor≠255 ∧
grd17  :   rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0) ∧
             adapContr=TRUE
               ⇒
               valacc=0
grd18  :   currentSpeed=0⇒valacc≥0
           gasPedal>0
grd19  :   ⇒
           valacc=min({30,max({(gasPedal*10)÷375,stv})})
           rangeRadarSensor<securedistanceToHead ∧ rangeRadarSensor∉{0,255}
grd20  :   ∧ adapContr=TRUE
                 ⇒
           valacc<0 ∧ stv=0
THEN
act1   :   SCSLeverUD≔valSCS
act2   :   SCSLeverUDP≔SCSLeverUD
act3   :   SCSLeverFBP≔SCSLeverFB
act4   :   keyStateP≔keyState
act5   :   desiredSpeed≔newDesiredSpeed
act6   :   desiredSpeedP≔desiredSpeed
act7   :   lastTimeSCSLeverUD≔{TRUE↦0, FALSE↦currentTime}
```

```
            (bool(
            (normContr=FALSE ∧ adapContr=FALSE) ∨
            valSCS=Neutral))
  act8   :  lastdesiredSpeed≔newDesiredSpeed
  act9   :  mandesiredSpeed≔{TRUE↦newDesiredSpeed, FALSE↦mandesiredSpeed}(bool(newDesiredSpeed≥120))
  act10  :   trafficdetected≔FALSE
  act11  :   accVeh≔valacc
  act12  :   setVehicleSpeed≔stv
END

moveSCSLeverFB   ≙
  extended
STATUS
  ordinary
REFINES
  moveSCSLeverFB
ANY
  valSCS
  sw
  bh
  newnormCont
  newadapCont
  desSpeed
  stv
  valacc
  secdis
WHERE
  grd1   :  valSCS ∈ {Backward, Forward, Neutral} ∧ valSCS≠SCSLeverFB
  grd2   :  SCSLeverUD=Neutral
  grd3   :  SCSLeverFB ≠ Neutral ⟹ valSCS = Neutral
  grd4   :  bh={TRUE↦buttonHead, FALSE↦FALSE}(bool(valSCS ≠Backward))
  grd5   :  sw={TRUE↦FALSE, FALSE↦speedLimiterSwitchOn}(bool(bh=FALSE))
  grd6   :  desSpeed ∈  ℕ  ∧desSpeed ≤desiredSpeedMax
  grd7   :  keyState≠KeyInIgnitionOnPosition⟹ desSpeed=0
            valSCS = Forward  ⟹                                                    //   Req SCS-2
                        (
                        (desiredSpeed=0 ∧ currentSpeed≠0⟹desSpeed=min
  grd8   :  ({desiredSpeedMax,currentSpeed÷10}))
                        ∧
                        (currentSpeed=0 ⟹desSpeed=desiredSpeed)
                        )
            valSCS = Neutral
            ⟹
            desSpeed=
            {
            TRUE↦{
            TRUE↦detectedTrafficSign,
                    FALSE↦{TRUE↦120,
                            FALSE↦{TRUE↦mandesiredSpeed, FALSE↦desiredSpeed}
  grd9   :                                                    (bool(mandesiredSpeed≥120))
                                }(bool(desiredSpeed<120))
            }(bool(detectedTrafficSign∈20‥130)),
            FALSE↦desiredSpeed
            }(bool(
            trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧ SCSLeverUD=Neutral ∧
            adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition ∧
            detectedTrafficSign≥20
            ))
             newnormCont=bool(
             (
             (valSCS = Forward ∧ cruiseControlMode=1∧  (currentSpeed≥speedActiv ∨ desSpeed≠0))
               ∨
  grd10  :   (normContr=TRUE ∧ valSCS ≠Backward)
             )
              ∧
             brakePedal=0
             )
             newadapCont=bool(
             (
             (valSCS = Forward ∧ cruiseControlMode=2 ∧ (currentSpeed≥speedActiv ∨ desSpeed≠0))
               ∨
  grd11  :   (adapContr=TRUE ∧ valSCS ≠Backward)
             )
              ∧
             brakePedal=0
             )
  grd12  :   newnormCont=FALSE ∧ newadapCont=FALSE ⟹stv =0
```

```
  grd13   :   stv∈-60…
              30
  grd14   :   valacc∈-60 … 30
  grd15   :   currentSpeed=0⇒valacc=0
  grd16   :   brakePedal≠0 ⇒ valacc=max({-60,-(brakePedal∗10)÷375})
              currentSpeed=desSpeed ∧
               ((rangeRadarSensor≠255 ∧  rangeRadarSensor≥secdis) ∨ rangeRadarSensor=0) ∧
  grd17   :     newadapCont=TRUE
                      ⇒
                      valacc=0
  grd18   :   stv∈-60 … 30
  grd19   :   keyState≠KeyInIgnitionOnPosition ⇒ valacc = 0
              currentSpeed<desSpeed ∧
               ((rangeRadarSensor≠255 ∧  rangeRadarSensor≥secdis) ∨
  grd20   :   rangeRadarSensor=0) ∧
                newadapCont=TRUE
                      ⇒
                      valacc∈ 0…10
              gasPedal≠0
  grd21   :   ⇒
              valacc=min({30,max({(gasPedal∗10)÷375,stv})})
  grd22   :   secdis∈ ℕ
              speedOfHead≤speedActiv ∧ speedOfHeadP>speedOfHead ∧ speedOfHead≠0 ∧
               newadapCont=TRUE ∧
  grd23   :            rangeRadarSensor∉{0,255}
                       ⇒
                       secdis=25∗(currentSpeed÷36)
              speedOfHeadP<speedOfHead ∧ speedOfHead≠0 ∧ speedOfHead≤speedActiv  ∧
  grd24   :   newadapCont=TRUE ∧ rangeRadarSensor∉{0,255}
                       ⇒
                       secdis=3∗(currentSpeed∗10÷36)
              speedOfHead>speedActiv ∧ newadapCont=TRUE
  grd25   :   ⇒
              secdis=safetyDistance ∗(currentSpeed∗10÷36)
              speedOfHead= 0 ∧ currentSpeed=0 ∧ newadapCont=TRUE  ∧ rangeRadarSensor∉{0,255}
  grd26   :   ⇒
              secdis=2
              currentSpeed=desSpeed ∧ newnormCont=TRUE
  grd27   :    ⇒
              stv=0
              currentSpeed<desSpeed ∧ newnormCont=TRUE
  grd28   :    ⇒
              stv∈ 0…10
              rangeRadarSensor<secdis ∧ rangeRadarSensor∉{0,255}
  grd29   :   ∧ newadapCont=TRUE
                       ⇒
              valacc<0 ∧ stv=0
THEN
  act1    :   SCSLeverFB≔valSCS
  act2    :   SCSLeverFBP≔SCSLeverFB
  act3    :   keyStateP≔keyState
  act4    :   speedLimiterSwitchOn≔sw
  act5    :   buttonHead≔bh
  act6    :   desiredSpeed ≔ desSpeed
  act7    :   desiredSpeedP ≔ desiredSpeed
  act8    :   normContr≔newnormCont
  act9    :   normContrP≔{TRUE↦normContr, FALSE↦FALSE}(newnormCont)
  act10   :    adapContr≔newadapCont
  act11   :    adapContrP≔{TRUE↦adapContr, FALSE↦FALSE}(newadapCont)
  act12   :    mandesiredSpeed≔{TRUE↦desSpeed, FALSE↦mandesiredSpeed}(bool(desSpeed≥120))
  act13   :    detectedTrafficOn≔bool(newadapCont=TRUE ∧ trafficSignDetectionOn=TRUE)
  act14   :    trafficdetected≔FALSE
  act15   :    visualWarningOn≔bool((currentSpeed÷36)∗15< rangeRadarSensor ∧ newadapCont=TRUE)
  act16   :    acousticWarningOn≔bool((currentSpeed÷36)∗8< rangeRadarSensor ∧ newadapCont=TRUE)
  act17   :    accVeh≔valacc
  act18   :    setVehicleSpeed≔stv
  act19   :    securedistanceToHead≔secdis
END

progress   ≙
  extended
STATUS
  ordinary
REFINES
  progress
```

```
ANY
  val
  radstate
  despeed
  stv
  tag
  rgs
  valacc
  secdis
WHERE
  grd1   :   val∈ rangeSpeed
  grd2   :   radstate∈ BOOL
  grd3   :   keyState≠KeyInIgnitionOnPosition ∨ nextTest≠currentTime+1⇒
             radstate=rangeRadarState
  grd4   :   keyState≠KeyInIgnitionOnPosition ⇒ val=0
  grd5   :   despeed∈ ℕ ∧ despeed≤desiredSpeedMax
             (normContr=FALSE ∧ adapContr=FALSE)
  grd6   :   ⇒
              despeed=0
             (normContr=TRUE ∨ adapContr=TRUE) ∧
  grd7   :   (SCSLeverUD=Neutral ∨ currentTime+1−lastTimeSCSLeverUD<2)
             ⇒
             despeed=desiredSpeed
  grd8   :   speedLimiterSwitchOn =TRUE ∧ gasPedal≤gasLimit ⇒ val≤speedLimit
             (normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
             currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward5
  grd9   :   ⇒
             despeed=min({desiredSpeedMax,lastdesiredSpeed+
             (currentTime+1−lastTimeSCSLeverUD−1)∗10})
              (normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
             SCSLeverUD=Upward7
  grd10  :   ⇒
             despeed=min({desiredSpeedMax,lastdesiredSpeed÷100∗100+
                  ((currentTime+1−lastTimeSCSLeverUD)÷2)∗100})
             (normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
             currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward5
  grd11  :   ⇒
             despeed=max({10,lastdesiredSpeed−
             (currentTime+1−lastTimeSCSLeverUD−1)∗10})
             (normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
              SCSLeverUD=Downward7
  grd12  :   ⇒
             despeed=max({10,(lastdesiredSpeed÷100)∗100−
             ((currentTime+1−lastTimeSCSLeverUD)÷2)∗100})
             trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧ SCSLeverUD=Neutral ∧
             SCSLeverFB=Neutral ∧
             adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition
             ⇒
             (
             (detectedTrafficSign<20⇒despeed=desiredSpeed)
             ∧
             (detectedTrafficSign∈20‥130⇒despeed=detectedTrafficSign)
  grd13  :   ∧
             (detectedTrafficSign>130⇒
               (desiredSpeed<120⇒despeed=120)
              ∧
               (desiredSpeed≥120 ∧ mandesiredSpeed≥120⇒despeed=mandesiredSpeed)
              ∧
              (desiredSpeed≥120 ∧ mandesiredSpeed=0⇒despeed=desiredSpeed)
             )
             )
             secdis∈ ℕ ∧ stv∈−60‥30 ∧ tag∈ ℕ ∧
  grd14  :   rgs∈rangeRadarSensorValues ∧
             valacc∈−60‥30
             val=despeed ∧
              ((rgs≠255 ∧   rgs≥secdis) ∨ rgs=0) ∧
  grd15  :    adapContr=TRUE
                    ⇒
                valacc=0
             brakePedal≠0 ⇒
  grd16  :    (val≠0⇒ valacc=max({−60,−brakePedal∗10 ÷ 375}))
             ∧
              (val=0⇒ valacc=0)
  grd17  :   gasPedal>0 ∧ (val≠despeed ∨
              ((rgs=255 ∨   rgs<secdis) ∧ rgs≠0) ∨  adapContr=FALSE)
             ⇒
```

valacc=min({30,max({(gasPedal∗10)÷375,stv})})

grd18 : val<despeed ∧ ((rgs≠255 ∧ rgs≥secdis) ∨ rgs=0) ∧ adapContr=TRUE ⟹ valacc∈ 0‥10

grd19 : accVeh≥0 ∧ speedLimiterSwitchOn=FALSE ⟹ ((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⟹ val=min({desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧ ((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0⟹ val=(accVeh+currentSpeed∗10÷36)∗36÷10)

grd20 : accVeh≥0∧ speedLimiterSwitchOn=TRUE ⟹ ((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⟹ val=min({speedLimit,desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧ ((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0 ⟹ val=min({speedLimit,(accVeh+currentSpeed∗10÷36)∗36÷10}))

grd21 : accVeh<0⟹val=max({0,(accVeh+currentSpeed∗10÷36)∗36÷10})

grd22 : normContr=FALSE ∧ adapContr=FALSE ⟹stv =0

grd23 : keyState≠KeyInIgnitionOnPosition ⟹ valacc = 0

grd24 : speedOfHead≤speedActiv ∧ speedOfHeadP>speedOfHead ∧ speedOfHead≠0 ∧ adapContr=TRUE ∧ rgs∉{0,255} ⟹ secdis=25∗(val÷36)

grd25 : speedOfHead= 0 ∧ val=0 ∧ adapContr=TRUE ∧ rgs∉{0,255} ⟹ secdis=2

grd26 : speedOfHeadP<speedOfHead ∧ speedOfHead≠0 ∧ speedOfHead≤speedActiv ∧ adapContr=TRUE ∧ rgs∉{0,255} ⟹ secdis=3∗(val∗10÷36)

grd27 : speedOfHead>speedActiv ∧ adapContr=TRUE ⟹ secdis=safetyDistance ∗(val∗10÷36)

grd28 : radstate=FALSE ⇔rgs=255

grd29 : val=despeed ∧ normContr=TRUE ⟹ stv=0

grd30 : val<despeed ∧ normContr=TRUE ⟹ stv∈ 0‥10

grd31 : rgs<secdis ∧ rgs∉{0,255} ∧ adapContr=TRUE ⟹ valacc<0 ∧ stv=0

grd32 :
keyState≠KeyInIgnitionOnPosition⟹(
∃ val1, radstate1, despeed1,rgs1,secdis1,stv1,valacc1,tag1·
(
val1∈ rangeSpeed ∧ radstate1∈ BOOL
∧(keyState≠KeyInIgnitionOnPosition ∨ nextTest≠currentTime+1⟹
radstate1=rangeRadarState)

∧(keyState≠KeyInIgnitionOnPosition ⟹ val1=0)
∧(despeed1∈ ℕ ∧ despeed1≤desiredSpeedMax)
∧((normContr=FALSE ∧ adapContr=FALSE) ⟹ despeed1=0)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧
(SCSLeverUD=Neutral ∨ currentTime+1−lastTimeSCSLeverUD<2)
⟹
despeed1=desiredSpeed)

∧(speedLimiterSwitchOn =TRUE ∧ gasPedal≤gasLimit ⟹ val1≤speedLimit)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧ SCSLeverUD≠Neutral ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward5
⟹
despeed1=min({desiredSpeedMax,lastdesiredSpeed+
(currentTime+1−lastTimeSCSLeverUD−1)∗10}))

∧
((normContr=TRUE ∨ adapContr=TRUE) ∧ SCSLeverUD≠Neutral ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
SCSLeverUD=Upward7
⟹
despeed1=min({desiredSpeedMax,lastdesiredSpeed÷100∗100+
((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}))
∧(
(normContr=TRUE ∨ adapContr=TRUE) ∧ SCSLeverUD≠Neutral ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward5

```
⇒
despeed1=max({10,lastdesiredSpeed−
(currentTime+1−lastTimeSCSLeverUD−1)∗10})  )

∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
 SCSLeverUD=Downward7
⇒
despeed1=max({10,(lastdesiredSpeed÷100)∗100−
((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )

∧(trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧ SCSLeverUD=Neutral ∧
SCSLeverFB=Neutral ∧
adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition
⇒
(
(detectedTrafficSign<20⇒despeed1=desiredSpeed)
∧
(detectedTrafficSign∈20‥130⇒despeed1=detectedTrafficSign)
∧
(detectedTrafficSign>130⇒
  (desiredSpeed<120⇒despeed1=120)
 ∧
  (desiredSpeed≥120 ∧ mandesiredSpeed≥120⇒despeed1=mandesiredSpeed)
 ∧
 (desiredSpeed≥120 ∧ mandesiredSpeed=0⇒despeed1=desiredSpeed)
)
)
)
∧(
secdis1∈ ℕ ∧ stv1∈−60‥30 ∧ tag1∈ ℕ ∧
rgs1∈rangeRadarSensorValues ∧
valacc1∈−60‥30
)
∧(val1=despeed1 ∧  ((rgs1≠255 ∧   rgs1≥secdis1) ∨ rgs1=0) ∧  adapContr=TRUE ⇒  valacc1=0)
∧(brakePedal≠0 ⇒
 (val1≠0 ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
 ⇒ valacc1=max({−60,−brakePedal∗10 ÷ 375}))
∧
 (val1=0⇒ valacc1=0)
)
∧(gasPedal>0  ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
⇒
valacc1=min({30,max({(gasPedal∗10)÷375,stv1})})
)
∧(val1<despeed1 ∧  ((rgs1≠255 ∧  rgs1≥secdis1) ∨ rgs1=0) ∧ adapContr=TRUE
        ⇒
      valacc1∈ 0‥10
 )
∧(accVeh≥0 ∧ speedLimiterSwitchOn=FALSE
⇒
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⇒
     val1=min({desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0⇒
     val1=(accVeh+currentSpeed∗10÷36)∗36÷10)
 )
∧(accVeh≥0∧ speedLimiterSwitchOn=TRUE
⇒
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⇒
val1=min({speedLimit,desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0 ⇒
val1=min({speedLimit,(accVeh+currentSpeed∗10÷36)∗36÷10}))
)
∧(accVeh<0⇒val1=max({0,(accVeh+currentSpeed∗10÷36)∗36÷10}))
∧(normContr=FALSE ∧ adapContr=FALSE ⇒stv1 =0)
∧(keyState≠KeyInIgnitionOnPosition ⇒ valacc1 = 0 )
∧(speedOfHead= 0 ∧ val1=0 ∧ adapContr=TRUE  ∧ rgs1∉{0,255}
         ⇒  secdis1=2)
∧(speedOfHead≤speedActiv ∧ speedOfHeadP>speedOfHead ∧   speedOfHead≠0 ∧ adapContr=TRUE ∧
         rgs1∉{0,255}
         ⇒   secdis1=25∗(val1÷36))


∧(speedOfHead≤speedActiv  ∧ speedOfHeadP<speedOfHead ∧ speedOfHead≠0 ∧    adapContr=TRUE ∧
         rgs1∉{0,255} ⇒  secdis1=3∗(val1∗10÷36) )

∧
(speedOfHead>speedActiv ∧ adapContr=TRUE  ⇒  secdis1=safetyDistance ∗(val1∗10÷36) )

∧(radstate1=FALSE ⇔rgs1=255)
∧(val1=despeed1 ∧ normContr=TRUE    ⇒   stv1=0)
```

```
∧(val1<despeed1 ∧ normContr=TRUE   ⇒   stv1∈ 0‥10 )

∧(rgs1<secdis1 ∧ rgs1∉{0,255}
∧ adapContr=TRUE
         ⇒
valacc1<0 ∧ stv1=0
)
)
)
grd33   :  keyState=KeyInIgnitionOnPosition ∧
(normContr=FALSE ∧ adapContr=FALSE)
⇒(
∃ val1, radstate1, despeed1,rgs1,secdis1,stv1,valacc1,tag1·
(
val1∈ rangeSpeed ∧ radstate1∈ BOOL
∧(keyState≠KeyInIgnitionOnPosition ∨ nextTest≠currentTime+1⇒
radstate1=rangeRadarState)

∧(keyState≠KeyInIgnitionOnPosition ⇒ val1=0)
∧(despeed1∈ ℕ ∧ despeed1≤desiredSpeedMax)
∧((normContr=FALSE ∧ adapContr=FALSE)   ⇒  despeed1=0)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧
(SCSLeverUD=Neutral ∨ currentTime+1−lastTimeSCSLeverUD<2)
⇒
despeed1=desiredSpeed)

∧(speedLimiterSwitchOn =TRUE ∧ gasPedal≤gasLimit ⇒ val1≤speedLimit)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward5
⇒
despeed1=min({desiredSpeedMax,lastdesiredSpeed+
(currentTime+1−lastTimeSCSLeverUD−1)∗10}) )

∧
((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
SCSLeverUD=Upward7
⇒
despeed1=min({desiredSpeedMax,lastdesiredSpeed÷100∗100+
     ((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )
 ∧(
(normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward5
⇒
despeed1=max({10,lastdesiredSpeed−
(currentTime+1−lastTimeSCSLeverUD−1)∗10})  )

∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
 SCSLeverUD=Downward7
⇒
despeed1=max({10,(lastdesiredSpeed÷100)∗100−
((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )

∧(trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧ SCSLeverUD=Neutral ∧
SCSLeverFB=Neutral ∧
adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition
⇒
(
(detectedTrafficSign<20⇒despeed1=desiredSpeed)
∧
(detectedTrafficSign∈20‥130⇒despeed1=detectedTrafficSign)
∧
(detectedTrafficSign>130⇒
  (desiredSpeed<120⇒despeed1=120)
 ∧
  (desiredSpeed≥120 ∧ mandesiredSpeed≥120⇒despeed1=mandesiredSpeed)
 ∧
 (desiredSpeed≥120 ∧ mandesiredSpeed=0⇒despeed1=desiredSpeed)
)
)
)
∧(
secdis1∈ ℕ ∧ stv1∈−60‥30 ∧ tag1∈ ℕ ∧
rgs1∈rangeRadarSensorValues ∧
valacc1∈−60‥30
)
∧(val1=despeed1 ∧  ((rgs1≠255 ∧   rgs1≥secdis1) ∨ rgs1=0) ∧  adapContr=TRUE ⇒  valacc1=0)
∧(brakePedal≠0 ⇒
 (val1≠0 ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
 ⇒ valacc1=max({−60,−brakePedal∗10 ÷ 375}))
```

```
∧
 (val1=0⟹ valacc1=0)
)
∧(gasPedal>0  ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
⟹
valacc1=min({30,max({(gasPedal∗10)÷375,stv1})})
)
∧(val1<despeed1 ∧  ((rgs1≠255 ∧  rgs1≥secdis1) ∨ rgs1=0) ∧ adapContr=TRUE
       ⟹
      valacc1∈ 0‥10
 )
∧(accVeh≥0 ∧ speedLimiterSwitchOn=FALSE
⟹
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⟹
     val1=min({desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0⟹
     val1=(accVeh+currentSpeed∗10÷36)∗36÷10)
 )
∧(accVeh≥0∧ speedLimiterSwitchOn=TRUE
⟹
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⟹
val1=min({speedLimit,desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0 ⟹
val1=min({speedLimit,(accVeh+currentSpeed∗10÷36)∗36÷10}))
)
∧(accVeh<0⟹val1=max({0,(accVeh+currentSpeed∗10÷36)∗36÷10}))
∧(normContr=FALSE ∧ adapContr=FALSE ⟹stv1 =0)
∧(keyState≠KeyInIgnitionOnPosition ⟹ valacc1 = 0 )
∧(speedOfHead= 0 ∧ val1=0 ∧ adapContr=TRUE  ∧ rgs1∉{0,255}
         ⟹  secdis1=2)
∧(speedOfHead≤speedActiv ∧ speedOfHeadP>speedOfHead ∧   speedOfHead≠0 ∧ adapContr=TRUE ∧
         rgs1∉{0,255}
         ⟹   secdis1=25∗(val1÷36))


∧(speedOfHead≤speedActiv  ∧ speedOfHeadP<speedOfHead ∧ speedOfHead≠0 ∧    adapContr=TRUE ∧
         rgs1∉{0,255} ⟹  secdis1=3∗(val1∗10÷36) )

∧
(speedOfHead>speedActiv ∧ adapContr=TRUE  ⟹  secdis1=safetyDistance ∗(val1∗10÷36) )

∧(radstate1=FALSE ⇔rgs1=255)
∧(val1=despeed1 ∧ normContr=TRUE    ⟹   stv1=0)

∧(val1<despeed1 ∧ normContr=TRUE   ⟹   stv1∈ 0‥10 )

∧(rgs1<secdis1 ∧ rgs1∉{0,255}
∧ adapContr=TRUE
         ⟹
valacc1<0 ∧ stv1=0
)
)
)
```

grd34 :

```
keyState=KeyInIgnitionOnPosition ∧
(normContr=TRUE ∨ adapContr=TRUE) ∧
(SCSLeverUD=Neutral ∨ currentTime+1−lastTimeSCSLeverUD<2) ∧
accVeh<0
⟹(
∃ val1, radstate1, despeed1,rgs1,secdis1,stv1,valacc1,tag1·
(
val1∈ rangeSpeed ∧ radstate1∈ BOOL
∧(keyState≠KeyInIgnitionOnPosition ∨ nextTest≠currentTime+1⟹
radstate1=rangeRadarState)

∧(keyState≠KeyInIgnitionOnPosition ⟹ val1=0)
∧(despeed1∈ ℕ ∧ despeed1≤desiredSpeedMax)
∧((normContr=FALSE ∧ adapContr=FALSE)   ⟹  despeed1=0)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧
(SCSLeverUD=Neutral ∨ currentTime+1−lastTimeSCSLeverUD<2)
⟹
despeed1=desiredSpeed)

∧(speedLimiterSwitchOn =TRUE ∧ gasPedal≤gasLimit ⟹ val1≤speedLimit)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward5
⟹
despeed1=min({desiredSpeedMax,lastdesiredSpeed+
(currentTime+1−lastTimeSCSLeverUD−1)∗10}) )
```

```
∧
((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
SCSLeverUD=Upward7
⟹
despeed1=min({desiredSpeedMax,lastdesiredSpeed÷100∗100+
     ((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )
 ∧(
(normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward5
⟹
despeed1=max({10,lastdesiredSpeed−
(currentTime+1−lastTimeSCSLeverUD−1)∗10})  )

∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
 SCSLeverUD=Downward7
⟹
despeed1=max({10,(lastdesiredSpeed÷100)∗100−
((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )

∧(trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧ SCSLeverUD=Neutral ∧
SCSLeverFB=Neutral ∧
adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition
⟹
(
(detectedTrafficSign<20⟹despeed1=desiredSpeed)
∧
(detectedTrafficSign∈20…130⟹despeed1=detectedTrafficSign)
∧
(detectedTrafficSign>130⟹
  (desiredSpeed<120⟹despeed1=120)
 ∧
  (desiredSpeed≥120 ∧ mandesiredSpeed≥120⟹despeed1=mandesiredSpeed)
 ∧
 (desiredSpeed≥120 ∧ mandesiredSpeed=0⟹despeed1=desiredSpeed)
)
)
)
∧(
secdis1∈ ℕ ∧ stv1∈−60…30 ∧ tag1∈ ℕ ∧
rgs1∈rangeRadarSensorValues ∧
valacc1∈−60…30
)
∧(val1=despeed1 ∧  ((rgs1≠255 ∧   rgs1≥secdis1) ∨ rgs1=0) ∧  adapContr=TRUE ⟹  valacc1=0)
∧(brakePedal≠0 ⟹
 (val1≠0 ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
 ⟹ valacc1=max({−60,−brakePedal∗10 ÷ 375}))
∧
 (val1=0⟹ valacc1=0)
)
∧(gasPedal>0  ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
⟹
valacc1=min({30,max({(gasPedal∗10)÷375,stv1})})
)
∧(val1<despeed1 ∧  ((rgs1≠255 ∧  rgs1≥secdis1) ∨ rgs1=0) ∧ adapContr=TRUE
        ⟹
      valacc1∈ 0…10
 )
∧(accVeh≥0 ∧ speedLimiterSwitchOn=FALSE
⟹
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⟹
     val1=min({desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0⟹
     val1=(accVeh+currentSpeed∗10÷36)∗36÷10)
 )
∧(accVeh≥0∧ speedLimiterSwitchOn=TRUE
⟹
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⟹
val1=min({speedLimit,desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0 ⟹
val1=min({speedLimit,(accVeh+currentSpeed∗10÷36)∗36÷10}))
)
∧(accVeh<0⟹val1=max({0,(accVeh+currentSpeed∗10÷36)∗36÷10}))
∧(normContr=FALSE ∧ adapContr=FALSE ⟹stv1 =0)
∧(keyState≠KeyInIgnitionOnPosition ⟹ valacc1 = 0 )
∧(speedOfHead= 0 ∧ val1=0 ∧ adapContr=TRUE  ∧ rgs1∉{0,255}
         ⟹  secdis1=2)
∧(speedOfHead≤speedActiv ∧ speedOfHeadP>speedOfHead ∧   speedOfHead≠0 ∧ adapContr=TRUE ∧
         rgs1∉{0,255}
         ⟹   secdis1=25∗(val1÷36))
```

```
∧(speedOfHead≤speedActiv  ∧ speedOfHeadP<speedOfHead ∧ speedOfHead≠0 ∧    adapContr=TRUE ∧
         rgs1∉{0,255} ⇒  secdis1=3∗(val1∗10÷36) )

∧
(speedOfHead>speedActiv ∧ adapContr=TRUE  ⇒  secdis1=safetyDistance ∗(val1∗10÷36) )

∧(radstate1=FALSE ⇔rgs1=255)
∧(val1=despeed1 ∧ normContr=TRUE    ⇒   stv1=0)

∧(val1<despeed1 ∧ normContr=TRUE   ⇒   stv1∈ 0‥10 )

∧(rgs1<secdis1 ∧ rgs1∉{0,255}
∧ adapContr=TRUE
         ⇒
valacc1<0 ∧ stv1=0
)
)
)
grd35   :  keyState=KeyInIgnitionOnPosition ∧
(normContr=TRUE ∨ adapContr=TRUE) ∧
(SCSLeverUD=Neutral ∨ currentTime+1−lastTimeSCSLeverUD<2) ∧
accVeh≥0
⇒(
∃ val1, radstate1, despeed1,rgs1,secdis1,stv1,valacc1,tag1·
(
val1∈ rangeSpeed ∧ radstate1∈ BOOL
∧(keyState≠KeyInIgnitionOnPosition ∨ nextTest≠currentTime+1⇒
radstate1=rangeRadarState)

∧(keyState≠KeyInIgnitionOnPosition ⇒ val1=0)
∧(despeed1∈ ℕ ∧ despeed1≤desiredSpeedMax)
∧((normContr=FALSE ∧ adapContr=FALSE)   ⇒  despeed1=0)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧
(SCSLeverUD=Neutral ∨ currentTime+1−lastTimeSCSLeverUD<2)
⇒
despeed1=desiredSpeed)

∧(speedLimiterSwitchOn =TRUE ∧ gasPedal≤gasLimit ⇒ val1≤speedLimit)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward5
⇒
despeed1=min({desiredSpeedMax,lastdesiredSpeed+
(currentTime+1−lastTimeSCSLeverUD−1)∗10}) )

∧
((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
SCSLeverUD=Upward7
⇒
despeed1=min({desiredSpeedMax,lastdesiredSpeed÷100∗100+
     ((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )
 ∧(
(normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward5
⇒
despeed1=max({10,lastdesiredSpeed−
(currentTime+1−lastTimeSCSLeverUD−1)∗10})  )

∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
 SCSLeverUD=Downward7
⇒
despeed1=max({10,(lastdesiredSpeed÷100)∗100−
((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )

∧(trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧ SCSLeverUD=Neutral ∧
SCSLeverFB=Neutral ∧
adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition
⇒
(
(detectedTrafficSign<20⇒despeed1=desiredSpeed)
∧
(detectedTrafficSign∈20‥130⇒despeed1=detectedTrafficSign)
∧
(detectedTrafficSign>130⇒
  (desiredSpeed<120⇒despeed1=120)
 ∧
  (desiredSpeed≥120 ∧ mandesiredSpeed≥120⇒despeed1=mandesiredSpeed)
 ∧
 (desiredSpeed≥120 ∧ mandesiredSpeed=0⇒despeed1=desiredSpeed)
)
```

```
)
)
∧(
secdis1∈ ℕ ∧ stv1∈−60‥30 ∧ tag1∈ ℕ ∧
rgs1∈rangeRadarSensorValues ∧
valacc1∈−60‥30
)
∧(val1=despeed1 ∧  ((rgs1≠255 ∧   rgs1≥secdis1) ∨ rgs1=0) ∧  adapContr=TRUE ⟹  valacc1=0)
∧(brakePedal≠0 ⟹
 (val1≠0 ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
 ⟹ valacc1=max({−60,−brakePedal∗10 ÷ 375}))
∧
 (val1=0⟹ valacc1=0)
)
∧(gasPedal>0  ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
⟹
valacc1=min({30,max({(gasPedal∗10)÷375,stv1})})
)
∧(val1<despeed1 ∧  ((rgs1≠255 ∧  rgs1≥secdis1) ∨ rgs1=0) ∧ adapContr=TRUE
        ⟹
      valacc1∈ 0‥10
 )
∧(accVeh≥0 ∧ speedLimiterSwitchOn=FALSE
⟹
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⟹
     val1=min({desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0⟹
     val1=(accVeh+currentSpeed∗10÷36)∗36÷10)
 )
∧(accVeh≥0∧ speedLimiterSwitchOn=TRUE
⟹
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⟹
val1=min({speedLimit,desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0 ⟹
val1=min({speedLimit,(accVeh+currentSpeed∗10÷36)∗36÷10}))
)
∧(accVeh<0⟹val1=max({0,(accVeh+currentSpeed∗10÷36)∗36÷10}))
∧(normContr=FALSE ∧ adapContr=FALSE ⟹stv1 =0)
∧(keyState≠KeyInIgnitionOnPosition ⟹ valacc1 = 0 )
∧(speedOfHead= 0 ∧ val1=0 ∧ adapContr=TRUE  ∧ rgs1∉{0,255}
         ⟹  secdis1=2)
∧(speedOfHead≤speedActiv ∧ speedOfHeadP>speedOfHead ∧   speedOfHead≠0 ∧ adapContr=TRUE ∧
         rgs1∉{0,255}
         ⟹   secdis1=25∗(val1÷36))


∧(speedOfHead≤speedActiv  ∧ speedOfHeadP<speedOfHead ∧ speedOfHead≠0 ∧    adapContr=TRUE ∧
         rgs1∉{0,255} ⟹  secdis1=3∗(val1∗10÷36) )

∧
(speedOfHead>speedActiv ∧ adapContr=TRUE  ⟹  secdis1=safetyDistance ∗(val1∗10÷36) )

∧(radstate1=FALSE ⇔rgs1=255)
∧(val1=despeed1 ∧ normContr=TRUE    ⟹   stv1=0)

∧(val1<despeed1 ∧ normContr=TRUE   ⟹   stv1∈ 0‥10 )

∧(rgs1<secdis1 ∧ rgs1∉{0,255}
∧ adapContr=TRUE
         ⟹
valacc1<0 ∧ stv1=0
)
)
)
grd36   :  keyState=KeyInIgnitionOnPosition ∧
(normContr=TRUE ∨ adapContr=TRUE) ∧
SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward5 ∧
accVeh<0
⟹(
∃ val1, radstate1, despeed1,rgs1,secdis1,stv1,valacc1,tag1·
(
val1∈ rangeSpeed ∧ radstate1∈ BOOL
∧(keyState≠KeyInIgnitionOnPosition ∨ nextTest≠currentTime+1⟹
radstate1=rangeRadarState)

∧(keyState≠KeyInIgnitionOnPosition ⟹ val1=0)
∧(despeed1∈ ℕ ∧ despeed1≤desiredSpeedMax)
∧((normContr=FALSE ∧ adapContr=FALSE)   ⟹  despeed1=0)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧
```

```
(SCSLeverUD=Neutral ∨ currentTime+1−lastTimeSCSLeverUD<2)
⇒
despeed1=desiredSpeed)

∧(speedLimiterSwitchOn =TRUE ∧ gasPedal≤gasLimit ⇒ val1≤speedLimit)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward5
⇒
despeed1=min({desiredSpeedMax,lastdesiredSpeed+
(currentTime+1−lastTimeSCSLeverUD−1)∗10}) )

∧
((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
SCSLeverUD=Upward7
⇒
despeed1=min({desiredSpeedMax,lastdesiredSpeed÷100∗100+
     ((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )
 ∧(
(normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward5
⇒
despeed1=max({10,lastdesiredSpeed−
(currentTime+1−lastTimeSCSLeverUD−1)∗10})  )

∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
 SCSLeverUD=Downward7
⇒
despeed1=max({10,(lastdesiredSpeed÷100)∗100−
((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )

∧(trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧ SCSLeverUD=Neutral ∧
SCSLeverFB=Neutral ∧
adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition
⇒
(
(detectedTrafficSign<20⇒despeed1=desiredSpeed)
∧
(detectedTrafficSign∈20…130⇒despeed1=detectedTrafficSign)
∧
(detectedTrafficSign>130⇒
  (desiredSpeed<120⇒despeed1=120)
 ∧
  (desiredSpeed≥120 ∧ mandesiredSpeed≥120⇒despeed1=mandesiredSpeed)
 ∧
 (desiredSpeed≥120 ∧ mandesiredSpeed=0⇒despeed1=desiredSpeed)
)
)
)
∧(
secdis1∈ ℕ ∧ stv1∈−60…30 ∧ tag1∈ ℕ ∧
rgs1∈rangeRadarSensorValues ∧
valacc1∈−60…30
)
∧(val1=despeed1 ∧  ((rgs1≠255 ∧   rgs1≥secdis1) ∨ rgs1=0) ∧  adapContr=TRUE ⇒  valacc1=0)
∧(brakePedal≠0 ⇒
 (val1≠0 ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
 ⇒ valacc1=max({−60,−brakePedal∗10 ÷ 375}))
∧
 (val1=0⇒ valacc1=0)
)
∧(gasPedal>0  ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
⇒
valacc1=min({30,max({(gasPedal∗10)÷375,stv1})})
)
∧(val1<despeed1 ∧  ((rgs1≠255 ∧  rgs1≥secdis1) ∨ rgs1=0) ∧ adapContr=TRUE
        ⇒
      valacc1∈ 0…10
 )
∧(accVeh≥0 ∧ speedLimiterSwitchOn=FALSE
⇒
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⇒
     val1=min({desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0⇒
     val1=(accVeh+currentSpeed∗10÷36)∗36÷10)
 )
∧(accVeh≥0∧ speedLimiterSwitchOn=TRUE
⇒
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⇒
val1=min({speedLimit,desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0 ⇒
```

```
val1=min({speedLimit,(accVeh+currentSpeed∗10÷36)∗36÷10}))
)
∧(accVeh<0⟹val1=max({0,(accVeh+currentSpeed∗10÷36)∗36÷10}))
∧(normContr=FALSE ∧ adapContr=FALSE ⟹stv1 =0)
∧(keyState≠KeyInIgnitionOnPosition ⟹ valacc1 = 0 )
∧(speedOfHead= 0 ∧ val1=0 ∧ adapContr=TRUE  ∧ rgs1∉{0,255}
         ⟹  secdis1=2)
∧(speedOfHead≤speedActiv ∧ speedOfHeadP>speedOfHead ∧   speedOfHead≠0 ∧ adapContr=TRUE ∧
         rgs1∉{0,255}
         ⟹   secdis1=25∗(val1÷36))


∧(speedOfHead≤speedActiv  ∧ speedOfHeadP<speedOfHead ∧ speedOfHead≠0 ∧    adapContr=TRUE ∧
         rgs1∉{0,255} ⟹  secdis1=3∗(val1∗10÷36) )

∧
(speedOfHead>speedActiv ∧ adapContr=TRUE  ⟹  secdis1=safetyDistance ∗(val1∗10÷36) )

∧(radstate1=FALSE ⇔rgs1=255)
∧(val1=despeed1 ∧ normContr=TRUE    ⟹   stv1=0)

∧(val1<despeed1 ∧ normContr=TRUE   ⟹   stv1∈ 0‥10 )

∧(rgs1<secdis1 ∧ rgs1∉{0,255}
∧ adapContr=TRUE
         ⟹
valacc1<0 ∧ stv1=0
)
)
)
```

grd37 :

```
keyState=KeyInIgnitionOnPosition ∧
(normContr=TRUE ∨ adapContr=TRUE) ∧
SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward5 ∧
accVeh≥0
⟹(
∃ val1, radstate1, despeed1,rgs1,secdis1,stv1,valacc1,tag1·
(
val1∈ rangeSpeed ∧ radstate1∈ BOOL
∧(keyState≠KeyInIgnitionOnPosition ∨ nextTest≠currentTime+1⟹
radstate1=rangeRadarState)

∧(keyState≠KeyInIgnitionOnPosition ⟹ val1=0)
∧(despeed1∈ ℕ ∧ despeed1≤desiredSpeedMax)
∧((normContr=FALSE ∧ adapContr=FALSE)   ⟹  despeed1=0)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧
(SCSLeverUD=Neutral ∨ currentTime+1−lastTimeSCSLeverUD<2)
⟹
despeed1=desiredSpeed)

∧(speedLimiterSwitchOn =TRUE ∧ gasPedal≤gasLimit ⟹ val1≤speedLimit)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward5
⟹
despeed1=min({desiredSpeedMax,lastdesiredSpeed+
(currentTime+1−lastTimeSCSLeverUD−1)∗10}) )

∧
((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
SCSLeverUD=Upward7
⟹
despeed1=min({desiredSpeedMax,lastdesiredSpeed÷100∗100+
     ((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )
 ∧(
(normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward5
⟹
despeed1=max({10,lastdesiredSpeed−
(currentTime+1−lastTimeSCSLeverUD−1)∗10})  )

∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
 SCSLeverUD=Downward7
⟹
despeed1=max({10,(lastdesiredSpeed÷100)∗100−
((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )

∧(trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧ SCSLeverUD=Neutral ∧
SCSLeverFB=Neutral ∧
adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition
⟹
```

```
(
(detectedTrafficSign<20⟹despeed1=desiredSpeed)
∧
(detectedTrafficSign∈20‥130⟹despeed1=detectedTrafficSign)
∧
(detectedTrafficSign>130⟹
  (desiredSpeed<120⟹despeed1=120)
 ∧
  (desiredSpeed≥120 ∧ mandesiredSpeed≥120⟹despeed1=mandesiredSpeed)
 ∧
 (desiredSpeed≥120 ∧ mandesiredSpeed=0⟹despeed1=desiredSpeed)
)
)
)
∧(
secdis1∈ ℕ ∧ stv1∈−60‥30 ∧ tag1∈ ℕ ∧
rgs1∈rangeRadarSensorValues ∧
valacc1∈−60‥30
)
∧(val1=despeed1 ∧  ((rgs1≠255 ∧   rgs1≥secdis1) ∨ rgs1=0) ∧  adapContr=TRUE ⟹  valacc1=0)
∧(brakePedal≠0 ⟹
 (val1≠0 ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
 ⟹ valacc1=max({−60,−brakePedal∗10 ÷ 375}))
∧
 (val1=0⟹ valacc1=0)
)
∧(gasPedal>0  ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
⟹
valacc1=min({30,max({(gasPedal∗10)÷375,stv1})})
)
∧(val1<despeed1 ∧  ((rgs1≠255 ∧  rgs1≥secdis1) ∨ rgs1=0) ∧ adapContr=TRUE
        ⟹
      valacc1∈ 0‥10
 )
∧(accVeh≥0 ∧ speedLimiterSwitchOn=FALSE
⟹
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⟹
     val1=min({desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0⟹
     val1=(accVeh+currentSpeed∗10÷36)∗36÷10)
 )
∧(accVeh≥0∧ speedLimiterSwitchOn=TRUE
⟹
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⟹
val1=min({speedLimit,desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0 ⟹
val1=min({speedLimit,(accVeh+currentSpeed∗10÷36)∗36÷10}))
)
∧(accVeh<0⟹val1=max({0,(accVeh+currentSpeed∗10÷36)∗36÷10}))
∧(normContr=FALSE ∧ adapContr=FALSE ⟹stv1 =0)
∧(keyState≠KeyInIgnitionOnPosition ⟹ valacc1 = 0 )
∧(speedOfHead= 0 ∧ val1=0 ∧ adapContr=TRUE  ∧ rgs1∉{0,255}
         ⟹  secdis1=2)
∧(speedOfHead≤speedActiv ∧ speedOfHeadP>speedOfHead ∧   speedOfHead≠0 ∧ adapContr=TRUE ∧
         rgs1∉{0,255}
         ⟹   secdis1=25∗(val1÷36))


∧(speedOfHead≤speedActiv  ∧ speedOfHeadP<speedOfHead ∧ speedOfHead≠0 ∧    adapContr=TRUE ∧
         rgs1∉{0,255} ⟹  secdis1=3∗(val1∗10÷36) )

∧
(speedOfHead>speedActiv ∧ adapContr=TRUE  ⟹  secdis1=safetyDistance ∗(val1∗10÷36) )

∧(radstate1=FALSE ⇔rgs1=255)
∧(val1=despeed1 ∧ normContr=TRUE    ⟹   stv1=0)

∧(val1<despeed1 ∧ normContr=TRUE   ⟹   stv1∈ 0‥10 )

∧(rgs1<secdis1 ∧ rgs1∉{0,255}
∧ adapContr=TRUE
         ⟹
valacc1<0 ∧ stv1=0
)
)
)
grd38   :   keyState=KeyInIgnitionOnPosition ∧
            (normContr=TRUE ∨ adapContr=TRUE) ∧
            SCSLeverUD≠Neutral  ∧
            currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward7 ∧
```

```
accVeh<0
⇒(
∃ val1, radstate1, despeed1,rgs1,secdis1,stv1,valacc1,tag1·
(
val1∈ rangeSpeed ∧ radstate1∈ BOOL
∧(keyState≠KeyInIgnitionOnPosition ∨ nextTest≠currentTime+1⇒
radstate1=rangeRadarState)

∧(keyState≠KeyInIgnitionOnPosition ⇒ val1=0)
∧(despeed1∈ ℕ ∧ despeed1≤desiredSpeedMax)
∧((normContr=FALSE ∧ adapContr=FALSE)   ⇒  despeed1=0)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧
(SCSLeverUD=Neutral ∨ currentTime+1−lastTimeSCSLeverUD<2)
⇒
despeed1=desiredSpeed)

∧(speedLimiterSwitchOn =TRUE ∧ gasPedal≤gasLimit ⇒ val1≤speedLimit)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward5
⇒
despeed1=min({desiredSpeedMax,lastdesiredSpeed+
(currentTime+1−lastTimeSCSLeverUD−1)∗10}) )

∧
((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
SCSLeverUD=Upward7
⇒
despeed1=min({desiredSpeedMax,lastdesiredSpeed÷100∗100+
     ((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )
 ∧(
(normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward5
⇒
despeed1=max({10,lastdesiredSpeed−
(currentTime+1−lastTimeSCSLeverUD−1)∗10})  )

∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
 SCSLeverUD=Downward7
⇒
despeed1=max({10,(lastdesiredSpeed÷100)∗100−
((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )

∧(trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧ SCSLeverUD=Neutral ∧
SCSLeverFB=Neutral ∧
adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition
⇒
(
(detectedTrafficSign<20⇒despeed1=desiredSpeed)
∧
(detectedTrafficSign∈20…130⇒despeed1=detectedTrafficSign)
∧
(detectedTrafficSign>130⇒
  (desiredSpeed<120⇒despeed1=120)
 ∧
  (desiredSpeed≥120 ∧ mandesiredSpeed≥120⇒despeed1=mandesiredSpeed)
 ∧
 (desiredSpeed≥120 ∧ mandesiredSpeed=0⇒despeed1=desiredSpeed)
)
)
)
∧(
secdis1∈ ℕ ∧ stv1∈−60…30 ∧ tag1∈ ℕ ∧
rgs1∈rangeRadarSensorValues ∧
valacc1∈−60…30
)
∧(val1=despeed1 ∧  ((rgs1≠255 ∧   rgs1≥secdis1) ∨ rgs1=0) ∧  adapContr=TRUE ⇒  valacc1=0)
∧(brakePedal≠0 ⇒
 (val1≠0 ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
 ⇒ valacc1=max({−60,−brakePedal∗10 ÷ 375}))
∧
 (val1=0⇒ valacc1=0)
)
∧(gasPedal>0  ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
⇒
valacc1=min({30,max({(gasPedal∗10)÷375,stv1})})
)
∧(val1<despeed1 ∧  ((rgs1≠255 ∧  rgs1≥secdis1) ∨ rgs1=0) ∧ adapContr=TRUE
        ⇒
      valacc1∈ 0…10
 )
```

```
∧(accVeh≥0 ∧ speedLimiterSwitchOn=FALSE
⇒
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⇒
     val1=min({desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0⇒
     val1=(accVeh+currentSpeed∗10÷36)∗36÷10)
 )
∧(accVeh≥0∧ speedLimiterSwitchOn=TRUE
⇒
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⇒
val1=min({speedLimit,desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0 ⇒
val1=min({speedLimit,(accVeh+currentSpeed∗10÷36)∗36÷10}))
)
∧(accVeh<0⇒val1=max({0,(accVeh+currentSpeed∗10÷36)∗36÷10}))
∧(normContr=FALSE ∧ adapContr=FALSE ⇒stv1 =0)
∧(keyState≠KeyInIgnitionOnPosition ⇒ valacc1 = 0 )
∧(speedOfHead= 0 ∧ val1=0 ∧ adapContr=TRUE  ∧ rgs1∉{0,255}
         ⇒  secdis1=2)
∧(speedOfHead≤speedActiv ∧ speedOfHeadP>speedOfHead ∧   speedOfHead≠0 ∧ adapContr=TRUE ∧
         rgs1∉{0,255}
         ⇒   secdis1=25∗(val1÷36))


∧(speedOfHead≤speedActiv  ∧ speedOfHeadP<speedOfHead ∧ speedOfHead≠0 ∧    adapContr=TRUE ∧
         rgs1∉{0,255} ⇒  secdis1=3∗(val1∗10÷36) )

∧
(speedOfHead>speedActiv ∧ adapContr=TRUE  ⇒  secdis1=safetyDistance ∗(val1∗10÷36) )

∧(radstate1=FALSE ⇔rgs1=255)
∧(val1=despeed1 ∧ normContr=TRUE    ⇒   stv1=0)

∧(val1<despeed1 ∧ normContr=TRUE   ⇒   stv1∈ 0‥10 )

∧(rgs1<secdis1 ∧ rgs1∉{0,255}
∧ adapContr=TRUE
         ⇒
valacc1<0 ∧ stv1=0
)
)
)
```

grd39 :

```
keyState=KeyInIgnitionOnPosition ∧
(normContr=TRUE ∨ adapContr=TRUE) ∧
SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward7 ∧
accVeh≥0
⇒(
∃ val1, radstate1, despeed1,rgs1,secdis1,stv1,valacc1,tag1·
(
val1∈ rangeSpeed ∧ radstate1∈ BOOL
∧(keyState≠KeyInIgnitionOnPosition ∨ nextTest≠currentTime+1⇒
radstate1=rangeRadarState)

∧(keyState≠KeyInIgnitionOnPosition ⇒ val1=0)
∧(despeed1∈ ℕ ∧ despeed1≤desiredSpeedMax)
∧((normContr=FALSE ∧ adapContr=FALSE)   ⇒  despeed1=0)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧
(SCSLeverUD=Neutral ∨ currentTime+1−lastTimeSCSLeverUD<2)
⇒
despeed1=desiredSpeed)

∧(speedLimiterSwitchOn =TRUE ∧ gasPedal≤gasLimit ⇒ val1≤speedLimit)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward5
⇒
despeed1=min({desiredSpeedMax,lastdesiredSpeed+
(currentTime+1−lastTimeSCSLeverUD)∗10}) )

∧
((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
SCSLeverUD=Upward7
⇒
despeed1=min({desiredSpeedMax,lastdesiredSpeed÷100∗100+
     ((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )
 ∧(
(normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward5
⇒
despeed1=max({10,lastdesiredSpeed−
```

```
(currentTime+1−lastTimeSCSLeverUD)∗10})  )

∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
 SCSLeverUD=Downward7
⇒
despeed1=max({10,(lastdesiredSpeed÷100)∗100−
((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )

∧(trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧ SCSLeverUD=Neutral ∧
SCSLeverFB=Neutral ∧
adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition
⇒
(
(detectedTrafficSign<20⇒despeed1=desiredSpeed)
∧
(detectedTrafficSign∈20…130⇒despeed1=detectedTrafficSign)
∧
(detectedTrafficSign>130⇒
  (desiredSpeed<120⇒despeed1=120)
 ∧
  (desiredSpeed≥120 ∧ mandesiredSpeed≥120⇒despeed1=mandesiredSpeed)
 ∧
 (desiredSpeed≥120 ∧ mandesiredSpeed=0⇒despeed1=desiredSpeed)
)
)
)
∧(
secdis1∈ ℕ ∧ stv1∈−60…30 ∧ tag1∈ ℕ ∧
rgs1∈rangeRadarSensorValues ∧
valacc1∈−60…30
)
∧(val1=despeed1 ∧  ((rgs1≠255 ∧   rgs1≥secdis1) ∨ rgs1=0) ∧  adapContr=TRUE ⇒  valacc1=0)
∧(brakePedal≠0 ⇒
 (val1≠0 ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
 ⇒ valacc1=max({−60,−brakePedal∗10 ÷ 375}))
∧
 (val1=0⇒ valacc1=0)
)
∧(gasPedal>0  ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
⇒
valacc1=min({30,max({(gasPedal∗10)÷375,stv1})})
)
∧(val1<despeed1 ∧  ((rgs1≠255 ∧  rgs1≥secdis1) ∨ rgs1=0) ∧ adapContr=TRUE
        ⇒
      valacc1∈ 0…10
 )
∧(accVeh≥0 ∧ speedLimiterSwitchOn=FALSE
⇒
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⇒
     val1=min({desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0⇒
     val1=(accVeh+currentSpeed∗10÷36)∗36÷10)
 )
∧(accVeh≥0∧ speedLimiterSwitchOn=TRUE
⇒
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⇒
val1=min({speedLimit,desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0 ⇒
val1=min({speedLimit,(accVeh+currentSpeed∗10÷36)∗36÷10}))
)
∧(accVeh<0⇒val1=max({0,(accVeh+currentSpeed∗10÷36)∗36÷10}))
∧(normContr=FALSE ∧ adapContr=FALSE ⇒stv1 =0)
∧(keyState≠KeyInIgnitionOnPosition ⇒ valacc1 = 0 )
∧(speedOfHead= 0 ∧ val1=0 ∧ adapContr=TRUE  ∧ rgs1∉{0,255}
         ⇒  secdis1=2)
∧(speedOfHead≤speedActiv ∧ speedOfHeadP>speedOfHead ∧   speedOfHead≠0 ∧ adapContr=TRUE ∧
         rgs1∉{0,255}
         ⇒   secdis1=25∗(val1÷36))


∧(speedOfHead≤speedActiv  ∧ speedOfHeadP<speedOfHead ∧ speedOfHead≠0 ∧    adapContr=TRUE ∧
         rgs1∉{0,255} ⇒  secdis1=3∗(val1∗10÷36) )

∧
(speedOfHead>speedActiv ∧ adapContr=TRUE  ⇒  secdis1=safetyDistance ∗(val1∗10÷36) )

∧(radstate1=FALSE ⇔rgs1=255)
∧(val1=despeed1 ∧ normContr=TRUE    ⇒   stv1=0)

∧(val1<despeed1 ∧ normContr=TRUE   ⇒   stv1∈ 0…10 )
```

```
∧(rgs1<secdis1 ∧ rgs1∉{0,255}
∧ adapContr=TRUE
          ⇒
valacc1<0 ∧ stv1=0
)
)
)
grd40   :  keyState=KeyInIgnitionOnPosition ∧
(normContr=TRUE ∨ adapContr=TRUE) ∧
SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward5 ∧
accVeh≥0
⇒(
∃ val1, radstate1, despeed1,rgs1,secdis1,stv1,valacc1,tag1·
(
val1∈ rangeSpeed ∧ radstate1∈ BOOL
∧(keyState≠KeyInIgnitionOnPosition ∨ nextTest≠currentTime+1⇒
radstate1=rangeRadarState)

∧(keyState≠KeyInIgnitionOnPosition ⇒ val1=0)
∧(despeed1∈ ℕ ∧ despeed1≤desiredSpeedMax)
∧((normContr=FALSE ∧ adapContr=FALSE)   ⇒  despeed1=0)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧
(SCSLeverUD=Neutral ∨ currentTime+1−lastTimeSCSLeverUD<2)
⇒
despeed1=desiredSpeed)

∧(speedLimiterSwitchOn =TRUE ∧ gasPedal≤gasLimit ⇒ val1≤speedLimit)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward5
⇒
despeed1=min({desiredSpeedMax,lastdesiredSpeed+
(currentTime+1−lastTimeSCSLeverUD−1)∗10}) )

∧
((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
SCSLeverUD=Upward7
⇒
despeed1=min({desiredSpeedMax,lastdesiredSpeed÷100∗100+
     ((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )
 ∧(
(normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward5
⇒
despeed1=max({10,lastdesiredSpeed−
(currentTime+1−lastTimeSCSLeverUD−1)∗10})  )

∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
 SCSLeverUD=Downward7
⇒
despeed1=max({10,(lastdesiredSpeed÷100)∗100−
((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )

∧(trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧ SCSLeverUD=Neutral ∧
SCSLeverFB=Neutral ∧
adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition
⇒
(
(detectedTrafficSign<20⇒despeed1=desiredSpeed)
∧
(detectedTrafficSign∈20‥130⇒despeed1=detectedTrafficSign)
∧
(detectedTrafficSign>130⇒
  (desiredSpeed<120⇒despeed1=120)
 ∧
  (desiredSpeed≥120 ∧ mandesiredSpeed≥120⇒despeed1=mandesiredSpeed)
 ∧
 (desiredSpeed≥120 ∧ mandesiredSpeed=0⇒despeed1=desiredSpeed)
)
)
)
∧(
secdis1∈ ℕ ∧ stv1∈−60‥30 ∧ tag1∈ ℕ ∧
rgs1∈rangeRadarSensorValues ∧
valacc1∈−60‥30
)
∧(val1=despeed1 ∧  ((rgs1≠255 ∧   rgs1≥secdis1) ∨ rgs1=0) ∧  adapContr=TRUE ⇒  valacc1=0)
∧(brakePedal≠0 ⇒
 (val1≠0 ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
```

```
 ⇒ valacc1=max({-60,-brakePedal*10 ÷ 375}))
∧
 (val1=0⇒ valacc1=0)
)
∧(gasPedal>0  ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
⇒
valacc1=min({30,max({(gasPedal*10)÷375,stv1})})
)
∧(val1<despeed1 ∧  ((rgs1≠255 ∧  rgs1≥secdis1) ∨ rgs1=0) ∧ adapContr=TRUE
        ⇒
      valacc1∈ 0‥10
 )
∧(accVeh≥0 ∧ speedLimiterSwitchOn=FALSE
⇒
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⇒
     val1=min({desiredSpeed,(accVeh+currentSpeed*10÷36)*36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0⇒
     val1=(accVeh+currentSpeed*10÷36)*36÷10)
 )
∧(accVeh≥0∧ speedLimiterSwitchOn=TRUE
⇒
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⇒
val1=min({speedLimit,desiredSpeed,(accVeh+currentSpeed*10÷36)*36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0 ⇒
val1=min({speedLimit,(accVeh+currentSpeed*10÷36)*36÷10}))
)
∧(accVeh<0⇒val1=max({0,(accVeh+currentSpeed*10÷36)*36÷10}))
∧(normContr=FALSE ∧ adapContr=FALSE ⇒stv1 =0)
∧(keyState≠KeyInIgnitionOnPosition ⇒ valacc1 = 0 )
∧(speedOfHead= 0 ∧ val1=0 ∧ adapContr=TRUE  ∧ rgs1∉{0,255}
         ⇒  secdis1=2)
∧(speedOfHead≤speedActiv ∧ speedOfHeadP>speedOfHead ∧   speedOfHead≠0 ∧ adapContr=TRUE ∧
         rgs1∉{0,255}
         ⇒   secdis1=25*(val1÷36))


∧(speedOfHead≤speedActiv  ∧ speedOfHeadP<speedOfHead ∧ speedOfHead≠0 ∧    adapContr=TRUE ∧
         rgs1∉{0,255} ⇒  secdis1=3*(val1*10÷36) )

∧
(speedOfHead>speedActiv ∧ adapContr=TRUE  ⇒  secdis1=safetyDistance *(val1*10÷36) )

∧(radstate1=FALSE ⇔rgs1=255)
∧(val1=despeed1 ∧ normContr=TRUE    ⇒   stv1=0)

∧(val1<despeed1 ∧ normContr=TRUE   ⇒   stv1∈ 0‥10 )

∧(rgs1<secdis1 ∧ rgs1∉{0,255}
∧ adapContr=TRUE
         ⇒
valacc1<0 ∧ stv1=0
)
)
)
```

grd41 :

```
keyState=KeyInIgnitionOnPosition ∧
(normContr=TRUE ∨ adapContr=TRUE) ∧
SCSLeverUD≠Neutral  ∧
currentTime+1-lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward5 ∧
accVeh<0
⇒(
∃ val1, radstate1, despeed1,rgs1,secdis1,stv1,valacc1,tag1·
(
val1∈ rangeSpeed ∧ radstate1∈ BOOL
∧(keyState≠KeyInIgnitionOnPosition ∨ nextTest≠currentTime+1⇒
radstate1=rangeRadarState)

∧(keyState≠KeyInIgnitionOnPosition ⇒ val1=0)
∧(despeed1∈ ℕ ∧ despeed1≤desiredSpeedMax)
∧((normContr=FALSE ∧ adapContr=FALSE)   ⇒  despeed1=0)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧
(SCSLeverUD=Neutral ∨ currentTime+1-lastTimeSCSLeverUD<2)
⇒
despeed1=desiredSpeed)

∧(speedLimiterSwitchOn =TRUE ∧ gasPedal≤gasLimit ⇒ val1≤speedLimit)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1-lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward5
⇒
despeed1=min({desiredSpeedMax,lastdesiredSpeed+
(currentTime+1-lastTimeSCSLeverUD-1)*10}) )
```

```
∧
((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
SCSLeverUD=Upward7
⟹
despeed1=min({desiredSpeedMax,lastdesiredSpeed÷100∗100+
     ((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )
 ∧(
(normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward5
⟹
despeed1=max({10,lastdesiredSpeed−
(currentTime+1−lastTimeSCSLeverUD−1)∗10})  )

∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
 SCSLeverUD=Downward7
⟹
despeed1=max({10,(lastdesiredSpeed÷100)∗100−
((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )

∧(trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧ SCSLeverUD=Neutral ∧
SCSLeverFB=Neutral ∧
adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition
⟹
(
(detectedTrafficSign<20⟹despeed1=desiredSpeed)
∧
(detectedTrafficSign∈20…130⟹despeed1=detectedTrafficSign)
∧
(detectedTrafficSign>130⟹
  (desiredSpeed<120⟹despeed1=120)
 ∧
  (desiredSpeed≥120 ∧ mandesiredSpeed≥120⟹despeed1=mandesiredSpeed)
 ∧
 (desiredSpeed≥120 ∧ mandesiredSpeed=0⟹despeed1=desiredSpeed)
)
)
)
∧(
secdis1∈ ℕ ∧ stv1∈−60…30 ∧ tag1∈ ℕ ∧
rgs1∈rangeRadarSensorValues ∧
valacc1∈−60…30
)
∧(val1=despeed1 ∧  ((rgs1≠255 ∧   rgs1≥secdis1) ∨ rgs1=0) ∧  adapContr=TRUE ⟹  valacc1=0)
∧(brakePedal≠0 ⟹
 (val1≠0 ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
 ⟹ valacc1=max({−60,−brakePedal∗10 ÷ 375}))
∧
 (val1=0⟹ valacc1=0)
)
∧(gasPedal>0  ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
⟹
valacc1=min({30,max({(gasPedal∗10)÷375,stv1})})
)
∧(val1<despeed1 ∧  ((rgs1≠255 ∧  rgs1≥secdis1) ∨ rgs1=0) ∧ adapContr=TRUE
        ⟹
      valacc1∈ 0…10
 )
∧(accVeh≥0 ∧ speedLimiterSwitchOn=FALSE
⟹
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⟹
     val1=min({desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0⟹
     val1=(accVeh+currentSpeed∗10÷36)∗36÷10)
 )
∧(accVeh≥0∧ speedLimiterSwitchOn=TRUE
⟹
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⟹
val1=min({speedLimit,desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0 ⟹
val1=min({speedLimit,(accVeh+currentSpeed∗10÷36)∗36÷10}))
)
∧(accVeh<0⟹val1=max({0,(accVeh+currentSpeed∗10÷36)∗36÷10}))
∧(normContr=FALSE ∧ adapContr=FALSE ⟹stv1 =0)
∧(keyState≠KeyInIgnitionOnPosition ⟹ valacc1 = 0 )
∧(speedOfHead= 0 ∧ val1=0 ∧ adapContr=TRUE  ∧ rgs1∉{0,255}
         ⟹  secdis1=2)
∧(speedOfHead≤speedActiv ∧ speedOfHeadP>speedOfHead ∧   speedOfHead≠0 ∧ adapContr=TRUE ∧
         rgs1∉{0,255}
         ⟹   secdis1=25∗(val1÷36))
```

```
∧(speedOfHead≤speedActiv  ∧ speedOfHeadP<speedOfHead ∧ speedOfHead≠0 ∧    adapContr=TRUE ∧
         rgs1∉{0,255} ⇒  secdis1=3∗(val1∗10÷36) )

∧
(speedOfHead>speedActiv ∧ adapContr=TRUE  ⇒  secdis1=safetyDistance ∗(val1∗10÷36) )

∧(radstate1=FALSE ⇔rgs1=255)
∧(val1=despeed1 ∧ normContr=TRUE    ⇒   stv1=0)

∧(val1<despeed1 ∧ normContr=TRUE   ⇒   stv1∈ 0‥10 )

∧(rgs1<secdis1 ∧ rgs1∉{0,255}
∧ adapContr=TRUE
         ⇒
valacc1<0 ∧ stv1=0
)
)
)
grd42   :  keyState=KeyInIgnitionOnPosition ∧
(normContr=TRUE ∨ adapContr=TRUE) ∧
SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward7 ∧
accVeh≥0
⇒(
∃ val1, radstate1, despeed1,rgs1,secdis1,stv1,valacc1,tag1·
(
val1∈ rangeSpeed ∧ radstate1∈ BOOL
∧(keyState≠KeyInIgnitionOnPosition ∨ nextTest≠currentTime+1⇒
radstate1=rangeRadarState)

∧(keyState≠KeyInIgnitionOnPosition ⇒ val1=0)
∧(despeed1∈ ℕ ∧ despeed1≤desiredSpeedMax)
∧((normContr=FALSE ∧ adapContr=FALSE)   ⇒  despeed1=0)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧
(SCSLeverUD=Neutral ∨ currentTime+1−lastTimeSCSLeverUD<2)
⇒
despeed1=desiredSpeed)

∧(speedLimiterSwitchOn =TRUE ∧ gasPedal≤gasLimit ⇒ val1≤speedLimit)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward5
⇒
despeed1=min({desiredSpeedMax,lastdesiredSpeed+
(currentTime+1−lastTimeSCSLeverUD−1)∗10}) )

∧
((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
SCSLeverUD=Upward7
⇒
despeed1=min({desiredSpeedMax,lastdesiredSpeed÷100∗100+
     ((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )
 ∧(
(normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward5
⇒
despeed1=max({10,lastdesiredSpeed−
(currentTime+1−lastTimeSCSLeverUD−1)∗10})  )

∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
 SCSLeverUD=Downward7
⇒
despeed1=max({10,(lastdesiredSpeed÷100)∗100−
((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )

∧(trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧ SCSLeverUD=Neutral ∧
SCSLeverFB=Neutral ∧
adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition
⇒
(
(detectedTrafficSign<20⇒despeed1=desiredSpeed)
∧
(detectedTrafficSign∈20‥130⇒despeed1=detectedTrafficSign)
∧
(detectedTrafficSign>130⇒
  (desiredSpeed<120⇒despeed1=120)
 ∧
  (desiredSpeed≥120 ∧ mandesiredSpeed≥120⇒despeed1=mandesiredSpeed)
 ∧
```

```
 (desiredSpeed≥120 ∧ mandesiredSpeed=0⇒despeed1=desiredSpeed)
)
)
)
∧(
secdis1∈ ℕ ∧ stv1∈−60‥30 ∧ tag1∈ ℕ ∧
rgs1∈rangeRadarSensorValues ∧
valacc1∈−60‥30
)
∧(val1=despeed1 ∧  ((rgs1≠255 ∧   rgs1≥secdis1) ∨ rgs1=0) ∧  adapContr=TRUE ⇒  valacc1=0)
∧(brakePedal≠0 ⇒
 (val1≠0 ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
 ⇒ valacc1=max({−60,−brakePedal∗10 ÷ 375}))
∧
 (val1=0⇒ valacc1=0)
)
∧(gasPedal>0  ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
⇒
valacc1=min({30,max({(gasPedal∗10)÷375,stv1})})
)
∧(val1<despeed1 ∧  ((rgs1≠255 ∧  rgs1≥secdis1) ∨ rgs1=0) ∧ adapContr=TRUE
        ⇒
      valacc1∈ 0‥10
 )
∧(accVeh≥0 ∧ speedLimiterSwitchOn=FALSE
⇒
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⇒
     val1=min({desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0⇒
     val1=(accVeh+currentSpeed∗10÷36)∗36÷10)
 )
∧(accVeh≥0∧ speedLimiterSwitchOn=TRUE
⇒
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⇒
val1=min({speedLimit,desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0 ⇒
val1=min({speedLimit,(accVeh+currentSpeed∗10÷36)∗36÷10}))
)
∧(accVeh<0⇒val1=max({0,(accVeh+currentSpeed∗10÷36)∗36÷10}))
∧(normContr=FALSE ∧ adapContr=FALSE ⇒stv1 =0)
∧(keyState≠KeyInIgnitionOnPosition ⇒ valacc1 = 0 )
∧(speedOfHead= 0 ∧ val1=0 ∧ adapContr=TRUE  ∧ rgs1∉{0,255}
         ⇒  secdis1=2)
∧(speedOfHead≤speedActiv ∧ speedOfHeadP>speedOfHead ∧   speedOfHead≠0 ∧ adapContr=TRUE ∧
         rgs1∉{0,255}
         ⇒   secdis1=25∗(val1÷36))


∧(speedOfHead≤speedActiv  ∧ speedOfHeadP<speedOfHead ∧ speedOfHead≠0 ∧    adapContr=TRUE ∧
         rgs1∉{0,255} ⇒  secdis1=3∗(val1∗10÷36) )

∧
(speedOfHead>speedActiv ∧ adapContr=TRUE  ⇒  secdis1=safetyDistance ∗(val1∗10÷36) )

∧(radstate1=FALSE ⇔rgs1=255)
∧(val1=despeed1 ∧ normContr=TRUE    ⇒   stv1=0)

∧(val1<despeed1 ∧ normContr=TRUE   ⇒   stv1∈ 0‥10 )

∧(rgs1<secdis1 ∧ rgs1∉{0,255}
∧ adapContr=TRUE
         ⇒
valacc1<0 ∧ stv1=0
)
)
)
grd43   :   keyState=KeyInIgnitionOnPosition ∧
(normContr=TRUE ∨ adapContr=TRUE) ∧
SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward7 ∧
accVeh<0
⇒(
∃ val1, radstate1, despeed1,rgs1,secdis1,stv1,valacc1,tag1·
(
val1∈ rangeSpeed ∧ radstate1∈ BOOL
∧(keyState≠KeyInIgnitionOnPosition ∨ nextTest≠currentTime+1⇒
radstate1=rangeRadarState)

∧(keyState≠KeyInIgnitionOnPosition ⇒ val1=0)
∧(despeed1∈ ℕ ∧ despeed1≤desiredSpeedMax)
```

```
∧((normContr=FALSE ∧ adapContr=FALSE)   ⟹  despeed1=0)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧
(SCSLeverUD=Neutral ∨ currentTime+1−lastTimeSCSLeverUD<2)
⟹
despeed1=desiredSpeed)

∧(speedLimiterSwitchOn =TRUE ∧ gasPedal≤gasLimit ⟹ val1≤speedLimit)
∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward5
⟹
despeed1=min({desiredSpeedMax,lastdesiredSpeed+
(currentTime+1−lastTimeSCSLeverUD)∗10}) )

∧
((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
SCSLeverUD=Upward7
⟹
despeed1=min({desiredSpeedMax,lastdesiredSpeed÷100∗100+
     ((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )
 ∧(
(normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward5
⟹
despeed1=max({10,lastdesiredSpeed−
(currentTime+1−lastTimeSCSLeverUD)∗10})  )

∧((normContr=TRUE ∨ adapContr=TRUE) ∧   SCSLeverUD≠Neutral  ∧ currentTime+1−lastTimeSCSLeverUD≥2 ∧
 SCSLeverUD=Downward7
⟹
despeed1=max({10,(lastdesiredSpeed÷100)∗100−
((currentTime+1−lastTimeSCSLeverUD)÷2)∗100}) )

∧(trafficSignDetectionOn=TRUE ∧ gasPedal=0 ∧ SCSLeverUD=Neutral ∧
SCSLeverFB=Neutral ∧
adapContr=TRUE ∧ keyState=KeyInIgnitionOnPosition
⟹
(
(detectedTrafficSign<20⟹despeed1=desiredSpeed)
∧
(detectedTrafficSign∈20‥130⟹despeed1=detectedTrafficSign)
∧
(detectedTrafficSign>130⟹
  (desiredSpeed<120⟹despeed1=120)
 ∧
  (desiredSpeed≥120 ∧ mandesiredSpeed≥120⟹despeed1=mandesiredSpeed)
 ∧
 (desiredSpeed≥120 ∧ mandesiredSpeed=0⟹despeed1=desiredSpeed)
)
)
)
∧(
secdis1∈ ℕ ∧ stv1∈−60‥30 ∧ tag1∈ ℕ ∧
rgs1∈rangeRadarSensorValues ∧
valacc1∈−60‥30
)
∧(val1=despeed1 ∧  ((rgs1≠255 ∧   rgs1≥secdis1) ∨ rgs1=0) ∧  adapContr=TRUE ⟹  valacc1=0)
∧(brakePedal≠0 ⟹
 (val1≠0 ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
 ⟹ valacc1=max({−60,−brakePedal∗10 ÷ 375}))
∧
 (val1=0⟹ valacc1=0)
)
∧(gasPedal>0  ∧ (val1≠despeed1 ∨  ((rgs1=255 ∨   rgs1<secdis1) ∧ rgs1≠0) ∨  adapContr=FALSE)
⟹
valacc1=min({30,max({(gasPedal∗10)÷375,stv1})})
)
∧(val1<despeed1 ∧  ((rgs1≠255 ∧  rgs1≥secdis1) ∨ rgs1=0) ∧ adapContr=TRUE
        ⟹
      valacc1∈ 0‥10
 )
∧(accVeh≥0 ∧ speedLimiterSwitchOn=FALSE
⟹
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⟹
     val1=min({desiredSpeed,(accVeh+currentSpeed∗10÷36)∗36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0⟹
     val1=(accVeh+currentSpeed∗10÷36)∗36÷10)
 )
∧(accVeh≥0∧ speedLimiterSwitchOn=TRUE
⟹
((normContr=TRUE ∨ adapContr=TRUE) ∧ desiredSpeed≠0⟹
```

```
val1=min({speedLimit,desiredSpeed,(accVeh+currentSpeed*10÷36)*36÷10})) ∧
((normContr=FALSE ∧ adapContr=FALSE) ∨ desiredSpeed=0 ⟹
val1=min({speedLimit,(accVeh+currentSpeed*10÷36)*36÷10}))
)
∧(accVeh<0⟹val1=max({0,(accVeh+currentSpeed*10÷36)*36÷10}))
∧(normContr=FALSE ∧ adapContr=FALSE ⟹stv1 =0)
∧(keyState≠KeyInIgnitionOnPosition ⟹ valacc1 = 0 )
∧(speedOfHead= 0 ∧ val1=0 ∧ adapContr=TRUE  ∧ rgs1∉{0,255}
         ⟹  secdis1=2)
∧(speedOfHead≤speedActiv ∧ speedOfHeadP>speedOfHead ∧   speedOfHead≠0 ∧ adapContr=TRUE ∧
         rgs1∉{0,255}
         ⟹   secdis1=25∗(val1÷36))


∧(speedOfHead≤speedActiv  ∧ speedOfHeadP<speedOfHead ∧ speedOfHead≠0 ∧    adapContr=TRUE ∧
         rgs1∉{0,255} ⟹  secdis1=3∗(val1∗10÷36) )

∧
(speedOfHead>speedActiv ∧ adapContr=TRUE  ⟹  secdis1=safetyDistance ∗(val1∗10÷36) )

∧(radstate1=FALSE ⇔rgs1=255)
∧(val1=despeed1 ∧ normContr=TRUE    ⟹   stv1=0)

∧(val1<despeed1 ∧ normContr=TRUE   ⟹   stv1∈ 0‥10 )

∧(rgs1<secdis1 ∧ rgs1∉{0,255}
∧ adapContr=TRUE
         ⟹
valacc1<0 ∧ stv1=0
)
)
)
grd44   :  (keyState≠KeyInIgnitionOnPosition)
∨
(keyState=KeyInIgnitionOnPosition ∧
(normContr=FALSE ∧ adapContr=FALSE))
∨
(keyState=KeyInIgnitionOnPosition ∧
(normContr=TRUE ∨ adapContr=TRUE) ∧
(SCSLeverUD=Neutral ∨ currentTime+1−lastTimeSCSLeverUD<2) ∧
accVeh<0)
∨
(keyState=KeyInIgnitionOnPosition ∧
(normContr=TRUE ∨ adapContr=TRUE) ∧
(SCSLeverUD=Neutral ∨ currentTime+1−lastTimeSCSLeverUD<2) ∧
accVeh≥0)
∨
(keyState=KeyInIgnitionOnPosition ∧
(normContr=TRUE ∨ adapContr=TRUE) ∧
SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward5 ∧
accVeh<0)
∨
(keyState=KeyInIgnitionOnPosition ∧
(normContr=TRUE ∨ adapContr=TRUE) ∧
SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward5 ∧
accVeh≥0)
∨
(keyState=KeyInIgnitionOnPosition ∧
(normContr=TRUE ∨ adapContr=TRUE) ∧
SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward7 ∧
accVeh<0)
∨
(keyState=KeyInIgnitionOnPosition ∧
(normContr=TRUE ∨ adapContr=TRUE) ∧
SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Upward7 ∧
accVeh≥0)
∨
(keyState=KeyInIgnitionOnPosition ∧
(normContr=TRUE ∨ adapContr=TRUE) ∧
SCSLeverUD≠Neutral  ∧
currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward5 ∧
accVeh≥0)
∨
(keyState=KeyInIgnitionOnPosition ∧
(normContr=TRUE ∨ adapContr=TRUE) ∧
SCSLeverUD≠Neutral  ∧
```

```
              currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward5 ∧
              accVeh<0)
              ∨
              (keyState=KeyInIgnitionOnPosition ∧
              (normContr=TRUE ∨ adapContr=TRUE) ∧
              SCSLeverUD≠Neutral  ∧
              currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward7 ∧
              accVeh<0)
              ∨
              (keyState=KeyInIgnitionOnPosition ∧
              (normContr=TRUE ∨ adapContr=TRUE) ∧
              SCSLeverUD≠Neutral  ∧
              currentTime+1−lastTimeSCSLeverUD≥2 ∧ SCSLeverUD=Downward7 ∧
              accVeh≥0)
THEN
  act1   :   currentSpeed≔val
  act2   :   currentTime≔currentTime+1
  act3   :   SCSLeverUDP≔SCSLeverUD
  act4   :   SCSLeverFBP≔SCSLeverFB
  act5   :   keyStateP≔keyState
  act6   :   rangeRadarState≔radstate
  act7   :   nextTest≔{TRUE↦currentTime+1+updateDur, FALSE↦nextTest}
               (bool(keyState=KeyInIgnitionOnPosition  ∧ nextTest=currentTime+1))
  act8   :   lastTest≔{TRUE↦currentTime+1, FALSE↦lastTest}
               (bool(keyState=KeyInIgnitionOnPosition  ∧ nextTest=currentTime+1))
  act9   :   normContrP≔normContr
  act10   :   adapContrP≔adapContr
  act11   :   desiredSpeed≔despeed
  act12   :   desiredSpeedP≔desiredSpeed
  act13   :   trafficdetected≔FALSE
  act14   :   securedistanceToHead≔secdis
  act15   :   accVeh≔valacc
  act16   :   visualWarningOn≔bool((val÷36)∗15< rgs ∧ adapContr=TRUE)
  act17   :   acousticWarningOn≔bool((val÷36)∗8< rgs ∧ adapContr=TRUE)
  act18   :   rangeRadarSensor≔rgs
  act19   :   setVehicleSpeed≔stv
END

turnHead   ≙
  extended
STATUS
  ordinary
REFINES
  turnHead
ANY
  val
WHERE
  grd1   :   val∈ headLevels ∧ val≠safetyDistance
             speedOfHead>speedActiv ∧ adapContr=TRUE
  grd2   :   ⟹
             securedistanceToHead=val ∗(currentSpeed∗10÷36)
THEN
  act1   :   safetyDistance ≔val
END

VehicHeadDetect   ≙
  extended
STATUS
  ordinary
REFINES
  VehicHeadDetect
ANY
  val
  stv
  secdis
  speh
  accv
WHERE
  grd1   :   secdis∈ℕ ∧ stv∈ −60‥30  ∧ accv∈ −60‥30
             speh>200 ∧ adapContr=TRUE
  grd2   :                ⟹
                          secdis=safetyDistance ∗(currentSpeed∗10÷36)
  grd3   :   bool(
                          (speh≤200 ∧ speedOfHead>speh ∧ speh≠0 ∧ adapContr=TRUE ∧ val∉{0,255})
                          ∨
                          (speh= 0 ∧ currentSpeed=0 ∧ adapContr=TRUE  ∧ val∉{0,255})
```

```
                    ∨
                    (speedOfHead<speh ∧ speh≠0 ∧ speh≤200  ∧ adapContr=TRUE ∧ val∉{0,255})
                    ∨
                    (speh>200 ∧ adapContr=TRUE)
                    )=FALSE
                    ⟹
                    secdis=securedistanceToHead
          adapContr=TRUE ∧ val∉{0,255}
grd4   :            ⟹
                    speh∈0…2000
          adapContr=FALSE ∨ val∈{0,255}
grd5   :            ⟹
                    speh=speedOfHead
          val ∈ rangeRadarSensorValues ∧
grd6   :            (rangeRadarState=FALSE ⟺val=255)
grd7   :  speh∈rangeSpeed
          currentSpeed<desiredSpeed ∧
                    ((val≠255 ∧  val≥secdis) ∨ val=0) ∧
grd8   :            adapContr=TRUE
                    ⟹
                    stv∈ 0…10
grd9   :  adapContr=FALSE⟹stv=0
          bool(
                    (currentSpeed<desiredSpeed ∧
                    ((val≠255 ∧  val≥secdis) ∨ val=0) ∧ adapContr=TRUE)
grd10  :            ∨
                    (adapContr=FALSE)) =FALSE
                    ⟹
                    stv=setVehicleSpeed
          val<secdis ∧ val∉{0,255} ∧ adapContr=TRUE
grd11  :            ⟹
                    brakePedale∈−30…−1 ∧ stv=0
          speh≤speedActiv ∧ speedOfHead>speh ∧ speh≠0 ∧ adapContr=TRUE ∧
grd12  :            val∉{0,255}
                    ⟹
                    secdis=25∗(currentSpeed÷36)
          speh= 0 ∧ currentSpeed=0 ∧ adapContr=TRUE  ∧ val∉{0,255}
grd13  :            ⟹
                    secdis=2
          speedOfHead<speh ∧ speh≠0 ∧ speh≤speedActiv
grd14  :    ∧ adapContr=TRUE ∧ val∉{0,255}
                     ⟹
                     secdis=3∗(currentSpeed∗10÷36)
          currentSpeed<desiredSpeed ∧
                    ((val≠255 ∧  val≥securedistanceToHead) ∨
grd15  :            val=0) ∧
                    adapContr=TRUE
                    ⟹
                    accv∈ 0…10
          brakePedal≠0 ∧ currentSpeed=0
grd16  :             ⟹
          accv=0
          brakePedal≠0 ∧ currentSpeed>0
grd17  :            ⟹
                    accv=max({−60,−(brakePedal∗10)÷375})
grd18  :  keyState≠KeyInIgnitionOnPosition ⟹ accv = 0
          currentSpeed<desiredSpeed ∧
                    ((val≠255 ∧  val≥secdis) ∨
grd19  :            val=0) ∧
                    adapContr=TRUE
                    ⟹
                    accv∈ 0…10
          currentSpeed=desiredSpeed ∧
                    ((val≠255 ∧ val≥secdis) ∨ val=0) ∧
grd20  :            adapContr=TRUE
                    ⟹
                    accv=0
          speh>speedActiv ∧ adapContr=TRUE
grd21  :  ⟹
          secdis=safetyDistance ∗(currentSpeed∗10÷36)
          gasPedal>0 ∧ (currentSpeed≠desiredSpeed ∨
           ((val=255 ∨  val<secdis) ∧ val≠0) ∨
grd22  :    adapContr=FALSE)
          ⟹
          accv=min({30,max({(gasPedal∗10)÷375,stv})})
grd23  :  val<secdis ∧ val∉{0,255} ∧ adapContr=TRUE
```

```
                    ⇒
              accv<0 ∧ stv=0
THEN
  act1   :   acousticWarningOn≔bool((currentSpeed÷36)∗8<val ∧ adapContr=TRUE)
  act2   :   speedOfHead≔speh
  act3   :   speedOfHeadP≔speedOfHead
  act4   :   setVehicleSpeed≔stv
  act5   :   accVeh≔accv
  act6   :   securedistanceToHead≔secdis
  act7   :   rangeRadarSensor ≔ val
  act8   :   visualWarningOn≔bool((currentSpeed÷36)∗15<val ∧ adapContr=TRUE)
END

pushButtonHead   ≙
  extended
STATUS
  ordinary
REFINES
  pushButtonHead
ANY
  sl
  desspeed
  stv
WHERE
  grd1   :   buttonHead=FALSE
  grd2   :   speedLimiterSwitchOn=FALSE ∧ SCSLeverFB≠Backward ∧
             keyState=KeyInIgnitionOnPosition ∧ gasPedal≤90
  grd3   :   keyState=KeyInIgnitionOnPosition ∧ SCSLeverFB ≠ Backward
  grd4   :   currentSpeed≤speedLimit
  grd5   :   sl∈rangeSpeed
  grd6   :   currentSpeed≤sl
  grd7   :   sl∈rangeSpeed
  grd8   :   desiredSpeed≥0⇒  sl=desspeed
  grd9   :   currentSpeed≤sl
  grd10   :   desspeed ∈ ℕ ∧ desspeed≤desiredSpeedMax
             lastTimeSCSLeverUD≠0 ∧ (normContr=TRUE ∨ adapContr=TRUE) ∧
             SCSLeverUD=Upward5 ∧
  grd11   :  currentTime−lastTimeSCSLeverUD≥2
              ⇒
             desspeed=min({desiredSpeedMax,
                          lastdesiredSpeed+(currentTime−lastTimeSCSLeverUD−1)∗10})
             (normContr=TRUE ∨ adapContr=TRUE) ∧ SCSLeverUD=Upward7 ∧
             currentTime−lastTimeSCSLeverUD≥2
  grd12   :   ⇒
             desspeed=min({desiredSpeedMax,lastdesiredSpeed÷100∗100+
                           ((currentTime−lastTimeSCSLeverUD)÷2)∗100})
             (normContr=TRUE ∨ adapContr=TRUE) ∧ SCSLeverUD=Downward5 ∧
             currentTime−lastTimeSCSLeverUD≥2
  grd13   :   ⇒
             desspeed=max({10,lastdesiredSpeed−
             (currentTime−lastTimeSCSLeverUD−1)∗10})
             lastTimeSCSLeverUD≠0 ∧ (normContr=TRUE ∨ adapContr=TRUE) ∧
             SCSLeverUD=Downward7 ∧ currentTime−lastTimeSCSLeverUD≥2
  grd14   :   ⇒
             desspeed=max({10,lastdesiredSpeed÷100∗100−
             ((currentTime−lastTimeSCSLeverUD)÷2)∗100})
  grd15   :   keyState≠KeyInIgnitionOnPosition⇒ desspeed=0
             currentSpeed<desspeed ∧
              ((rangeRadarSensor≠255 ∧
  grd16   :  rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0) ∧
               adapContr=TRUE
                    ⇒
                    accVeh∈ 0‥10
             currentSpeed=desspeed ∧
              ((rangeRadarSensor≠255 ∧  rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0) ∧
  grd17   :    adapContr=TRUE
                    ⇒
                    accVeh=0
  grd18   :   stv∈ −60‥30
             currentSpeed=desspeed ∧ normContr=TRUE
  grd19   :   ⇒
             stv=0
             currentSpeed<desspeed ∧ normContr=TRUE
  grd20   :   ⇒
             stv∈ 0‥10
  grd21   :   normContr=FALSE ∧ adapContr=FALSE ⇒stv =0
```

```
  grd22   :   gasPedal>0
              ⟹
              accVeh=min({30,max({(gasPedal∗10)÷375,stv})})
              rangeRadarSensor<securedistanceToHead ∧
              rangeRadarSensor∉{0,255}
  grd23   :   ∧ adapContr=TRUE
                        ⟹
              accVeh<0 ∧ stv=0
THEN
  act1    :   speedLimiterSwitchOn≔TRUE
  act2    :   speedLimit≔sl
  act3    :   SCSLeverUDP≔SCSLeverUD
  act4    :   SCSLeverFBP≔SCSLeverFB
  act5    :   keyStateP≔keyState
  act6    :   buttonHead≔TRUE
  act7    :   adapContrP≔adapContr
  act8    :   normContrP≔normContr
  act9    :   trafficdetected≔FALSE
  act10   :    desiredSpeed≔desspeed
  act11   :    setVehicleSpeed≔stv
END

trafficSignDetection   ≙
  extended
STATUS
  ordinary
REFINES
  trafficSignDetection
ANY
  val
  desSpeed
  accval
WHERE
              val∈ℕ ∧ val ≤desiredSpeedMax ∧
  grd1    :   detectedTrafficSign≠val ∧ desSpeed∈ ℕ ∧
              desSpeed≤desiredSpeedMax
  grd2    :   detectedTrafficOn=TRUE
              gasPedal>0
              ⟹(
              (val<200⟹desSpeed=desiredSpeed)
              ∧
              (val∈200‥1300⟹desSpeed=val)
              ∧
              (val>1300⟹(desiredSpeed<1200⟹desSpeed=1200)
  grd3    :    ∧
                (desiredSpeed≥1200 ∧ mandesiredSpeed≥1200⟹
                                  desSpeed=mandesiredSpeed)
               ∧
               (desiredSpeed≥1200 ∧ mandesiredSpeed=0⟹desSpeed=desiredSpeed)
              )
              )
              gasPedal=0
  grd4    :   ⟹
              desSpeed=val
  grd5    :   accval∈−60‥30
              gasPedal≠0
  grd6    :   ⟹
              accval=min({30,max({(gasPedal∗10)÷375,setVehicleSpeed})})
              currentSpeed<desSpeed ∧
               ((rangeRadarSensor≠255 ∧  rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0) ∧
  grd7    :     adapContr=TRUE
                      ⟹
                accval∈ 0‥10
              currentSpeed=desSpeed ∧
               ((rangeRadarSensor≠255 ∧  rangeRadarSensor≥securedistanceToHead) ∨ rangeRadarSensor=0) ∧
  grd8    :     adapContr=TRUE
                      ⟹
               accval=0
              rangeRadarSensor<securedistanceToHead ∧
              rangeRadarSensor∉{0,255}
  grd9    :   ∧ adapContr=TRUE
                        ⟹
              accval<0 ∧ setVehicleSpeed=0
THEN
  act1    :   SCSLeverUDP≔SCSLeverUD
  act2    :   SCSLeverFBP≔SCSLeverFB
```

```
    act3   :   keyStateP=keyState
    act4   :   detectedTrafficSign=val
    act5   :   trafficdetected=TRUE
    act6   :   normContrP=normContr
    act7   :   adapContrP=adapContr
    act8   :   desiredSpeed=desSpeed
    act9   :   desiredSpeedP=desiredSpeed
    act10   :   accVeh=accval
  END

END
```