```
CONTEXT
   GuiStates
SETS
   Modes
CONSTANTS
   ModesG
   StartUp
   Start
   SelfTest
   Menu
   Settings
   Ventilation
   Off
   possTransG
   PCV
   PSV
   batteryRange
AXIOMS
   axm1   :   ModesG⊆Modes
   axm2   :   partition(ModesG, {StartUp}, {Start} ,{Menu},{SelfTest}, {Settings},Ventilation, {Off})
   axm3   :   partition(Ventilation,{PCV},{PSV})
   axm4   :   possTransG∈ BOOL⟶ℙ(ModesG×ModesG)
              possTransG={TRUE↦{
                      Off↦StartUp,
                      StartUp↦Start
                      }∪
                      ({StartUp}×Ventilation) ∪
   axm5   :           {Start↦Menu,Start↦SelfTest,
                      Menu↦Settings, SelfTest↦SelfTest,
                      SelfTest↦Menu,Settings↦Menu} ∪ ({Off}×Ventilation) ∪
                      ({Menu,Settings}×Ventilation)∪
                         (Ventilation×({Menu,Settings}∪Ventilation)),
                      FALSE↦(ModesG×{Off})}
   axm6   :   batteryRange=0‥120*60*10
END
```

```
MACHINE
    GuiSM
SEES
    GuiStates
VARIABLES
    power
    onAC
    batLev
    switchover
    crashed
    modeG
    modeGP
    curTime
    batFail
INVARIANTS
    inv1   :   powerє BOOL ∧ crashed ∈ BOOL ∧ batLevє batteryRange ∧ curTimeєℕ
               ∧ batFailє BOOL
               crashed=TRUE ∨ power=FALSE ∨ (onAC=FALSE ∧ (switchover=FALSE ∨ batLev=0 ∨ batFail=TRUE))
    inv2   :    ⇔
               modeG=Off
    inv3   :   modeGє ModesG  ∧ modeGP ∈ ModesG
               modeGP ≠ modeG⟹modeGP↦modeGє
    inv4   :               possTransG(bool(power=TRUE ∧ crashed=FALSE ∧
                               (onAC=TRUE ∨ (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE))))
    inv5   :   modeG=Settings⟹ modeGP∈{Menu}∪Ventilation
EVENTS
    INITIALISATION   ≙
    STATUS
      ordinary
    BEGIN
      act1   :   power≔FALSE
      act2   :   batLev:∈batteryRange
      act3   :   onAC≔TRUE
      act4   :   switchover≔TRUE
      act5   :   crashed≔FALSE
      act6   :   modeG≔Off
      act7   :   modeGP≔Off
      act8   :   curTime≔0
      act9   :   batFail:∈BOOL
    END

    powerON   ≙          //   CONT.2
    STATUS
      ordinary
    WHEN
      grd1   :   power=FALSE
      grd2   :   onAC=TRUE ∨ (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)
    THEN
      act1   :   power≔TRUE
      act2   :   modeG≔{FALSE↦StartUp, TRUE↦Off}(crashed)
      act3   :   modeGP≔modeG
    END

    powerOff   ≙
    STATUS
      ordinary
    WHEN
      grd1   :   power=TRUE
    THEN
      act1   :   power≔FALSE
      act2   :   modeG≔Off
      act3   :   modeGP≔modeG
    END

    startUpEndedGui   ≙
    STATUS
      ordinary
    ANY
      modeg
```

```
WHERE
  grd1   :   modeG=StartUp
  grd2   :   modeg∈{Start}∪Ventilation
THEN
  act1   :   modeG≔modeg
  act2   :   modeGP≔modeG
END

crash   ≙
STATUS
  ordinary
WHEN
  grd1   :   crashed=FALSE
THEN
  act1   :   crashed≔TRUE
  act2   :   modeG≔Off
  act3   :   modeGP≔Off
END

repare   ≙
STATUS
  ordinary
ANY
  modeg
WHERE
  grd1   :   crashed=TRUE
             power=FALSE ∨  (onAC=FALSE ∧ (switchover=FALSE ∨ batLev=0 ∨ batFail=TRUE))
  grd2   :   ⇒
             modeg=Off
             power=TRUE ∧ (onAC=TRUE ∨ (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE))
  grd3   :   ⇒
             modeg=StartUp
THEN
  act1   :   crashed≔FALSE
  act2   :   modeG≔modeg
  act3   :   modeGP≔modeG
END

newPatient   ≙
STATUS
  ordinary
WHEN
  grd1   :   modeG=Start
THEN
  act1   :   modeG≔SelfTest
  act2   :   modeGP≔modeG
END

resumeVent   ≙
STATUS
  ordinary
WHEN
  grd1   :   modeG=Start
THEN
  act1   :   modeG≔Menu
  act2   :   modeGP≔modeG
END

runAbortSelfTest   ≙
STATUS
  ordinary
WHEN
  grd1   :   modeG=SelfTest
THEN
  act1   :   modeG≔SelfTest
  act2   :   modeGP≔modeG
END

selfTestPassed   ≙          //   CONT4-4.1
STATUS
  ordinary
```

```
WHEN
  grd1   :   modeG=SelfTest
THEN
  act1   :   modeG≔Menu
  act2   :   modeGP≔modeG
END

setParam   ≙
STATUS
  ordinary
WHEN
  grd1   :   modeG∈{Menu}∪Ventilation
THEN
  act1   :   modeG≔Settings
  act2   :   modeGP≔modeG
END

saveBackAbort   ≙
STATUS
  ordinary
ANY
  modeg
WHERE
  grd1   :   modeG=Settings ∧ modeg∈ModesG
  grd2   :   modeg∈Ventilation∪{Menu}
THEN
  act1   :   modeG≔modeg
  act2   :   modeGP≔modeG
END

startStopPCVPSV   ≙
STATUS
  ordinary
ANY
  modeg
WHERE
  grd1   :   modeG∈{Menu}∪Ventilation
  grd2   :   modeg∈Ventilation∪{Menu}
  grd3   :   modeG=Menu⇒modeg∈Ventilation
  grd4   :   modeG∈Ventilation⇒modeg=Menu
THEN
  act1   :   modeG≔modeg
  act2   :   modeGP≔modeG
END

changeMode   ≙
STATUS
  ordinary
ANY
  modeg
WHERE
  grd1   :   modeG∈ Ventilation
  grd2   :   modeg∈Ventilation∪{Settings}
THEN
  act1   :   modeG≔modeg
  act2   :   modeGP≔modeG
END

failExternalPower   ≙
STATUS
  ordinary
ANY
  modeg
  modegp
WHERE
  grd1   :   onAC=TRUE
  grd2   :   modeg ∈ ModesG ∧ modegp ∈ ModesG
  grd3   :   ¬(power=TRUE ∧ crashed=FALSE) ⇒ modeg=Off ∧ modegp=Off
  grd4   :   power=TRUE ∧ crashed=FALSE ⇒
             (
                ((switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE ⇒ modeg=modeG ∧ modegp=modeGP) ∧
```

```
                (¬(switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)⇒ modeg=Off ∧ modegp=modeG))
            )
  THEN
    act1   :   onAC≔FALSE
    act2   :   modeG≔modeg
    act3   :   modeGP≔modegp
  END

  progress   ≙
  STATUS
    ordinary
  ANY
    step
    l
    modeg
    batf
  WHERE
    grd1   :   step∈ℕ1 ∧ l∈batteryRange ∧ batf∈BOOL
    grd2   :   batFail=TRUE ⇒ l=batLev
    grd3   :   onAC=TRUE ∧ batFail=FALSE ⇔ l=batLev+step
    grd4   :   onAC=FALSE ∧ batFail=FALSE ⇔ l=max({0,batLev−step})
               (onAC=TRUE ∨ (l>0 ∧ switchover=TRUE ∧ batf=FALSE)) ∧
               power=TRUE ∧  modeG=Off ∧
    grd5   :   crashed=FALSE
               ⇒
               modeg=StartUp
    grd6   :   (l=0 ∨ batf=TRUE ∨ switchover=FALSE) ∧ onAC=FALSE ⇒modeg=Off
               onAC=TRUE  ∨
    grd7   :   ((l>0 ∧ batLev>0) ∧ switchover=TRUE ∧ batf=FALSE)
               ⇒
               modeg=modeG
  THEN
    act1   :   curTime≔curTime+step
    act2   :   batLev≔l
    act3   :   modeG≔modeg
    act4   :   modeGP≔{TRUE↦modeGP, FALSE↦modeG}(bool(modeg=modeG))
    act5   :   batFail≔batf
  END

  switchoverFail   ≙
  STATUS
    ordinary
  ANY
    modeg
  WHERE
    grd1   :   switchover=TRUE
    grd2   :   onAC=FALSE⇒modeg=Off
    grd3   :   onAC=TRUE⇒modeg=modeG
  THEN
    act1   :   switchover≔FALSE
    act2   :   modeG≔modeg
    act3   :   modeGP≔{TRUE↦modeG,FALSE↦modeGP}(bool(modeG≠modeg))
  END

END
```

```
CONTEXT
    ContStates
EXTENDS
    GuiStates
CONSTANTS
    ModesC
    FailSafe
    VentilationOff
    possTransC
AXIOMS
    axm1   :   partition(Modes,{StartUp}, {Start} ,{Menu},{SelfTest}, {FailSafe}, {Settings},{PSV},{PCV}, {Off},{VentilationOff})
    axm2   :   ModesC⊆Modes
    axm3   :   partition(ModesC, {StartUp}, {SelfTest}, {FailSafe},{VentilationOff}, {Off},{PCV},{PSV})        //   CONT1
    axm4   :   partition(Ventilation,{PCV},{PSV})
    axm5   :   possTransC∈ BOOL⟶ℙ(ModesC×ModesC)
               possTransC={                                                                                    //   CONT2:Off↦StartUp
                       TRUE↦
                       {Off↦StartUp, StartUp↦SelfTest, StartUp↦FailSafe,
                       SelfTest↦FailSafe, SelfTest↦VentilationOff,VentilationOff↦FailSafe, PSV↦PCV,PCV↦PSV}
    axm6   :
               ∪
                       (Ventilation×{VentilationOff})∪ ({VentilationOff}×Ventilation)∪
                       (Ventilation×{FailSafe}),
                       FALSE↦((ModesC\{Off})×{Off})}
END
```

```
MACHINE
    ContSM
REFINES
    GuiSM
SEES
    ContStates
VARIABLES
    power
    onAC
    batLev
    switchover
    crashed
    modeG
    modeGP
    batFail
    modeC
    modeCP
    curTime
INVARIANTS
                power=FALSE ∨ (onAC=FALSE ∧ (switchover=FALSE ∨ batLev=0 ∨ batFail=TRUE))
    inv1    :   ⇔
                modeC=Off
    inv2    :   modeCP ≠ modeC⇒modeCP↦modeC∈
                possTransC(bool(power=TRUE ∧ (onAC=TRUE ∨ (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE))))
    inv3    :   modeC∈ ModesC  ∧ modeCP ∈ ModesC
    inv4    :   power=TRUE ∧ (onAC=TRUE ∨ (switchover=TRUE ∧ batLev>0)) ∧ modeCP=Off⇒modeC=StartUp
                modeG∈{Settings,Menu}∪Ventilation
    inv5    :   ⇒
                modeC∈Ventilation ∪{FailSafe,VentilationOff}
    inv6    :   modeG∈Ventilation ⇒
                modeC∈Ventilation∪{FailSafe}
    inv7    :   modeC=StartUp⇒modeG≠Settings
EVENTS
    INITIALISATION   ≙
      extended
    STATUS
      ordinary
    BEGIN
      act1   :   power≔FALSE
      act2   :   batLev:∈batteryRange
      act3   :   onAC≔TRUE
      act4   :   switchover≔TRUE
      act5   :   crashed≔FALSE
      act6   :   modeG≔Off
      act7   :   modeGP≔Off
      act8   :   curTime≔0
      act9   :   batFail:∈BOOL
      act10   :   modeC≔Off
      act11   :   modeCP≔Off
    END

    powerON   ≙          //   CONT.2
      extended
    STATUS
      ordinary
    REFINES
      powerON
    WHEN
      grd1   :   power=FALSE
      grd2   :   onAC=TRUE ∨ (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)
    THEN
      act1   :   power≔TRUE
      act2   :   modeG≔{FALSE↦StartUp, TRUE↦Off}(crashed)
      act3   :   modeGP≔modeG
      act4   :   modeC≔StartUp
      act5   :   modeCP≔modeC
    END

    powerOff   ≙
```

```
  extended
STATUS
  ordinary
REFINES
  powerOff
WHEN
  grd1   :   power=TRUE
THEN
  act1   :   power≔FALSE
  act2   :   modeG≔Off
  act3   :   modeGP≔modeG
  act4   :   modeC≔Off
  act5   :   modeCP≔modeC
END

startUpEndedGui   ≙
  extended
STATUS
  ordinary
REFINES
  startUpEndedGui
ANY
  modeg
WHERE
  grd1   :   modeG=StartUp
  grd2   :   modeg∈{Start}∪Ventilation
  grd3   :   modeC∈Ventilation ⇒ modeg=modeC
  grd4   :   modeC∉Ventilation ⇒ modeg=Start
THEN
  act1   :   modeG≔modeg
  act2   :   modeGP≔modeG
END

startUpEndedCont   ≙
STATUS
  ordinary
WHEN
  grd1   :   modeC=StartUp
THEN
  act1   :   modeC≔SelfTest
  act2   :   modeCP≔modeC
END

crash   ≙
  extended
STATUS
  ordinary
REFINES
  crash
WHEN
  grd1   :   crashed=FALSE
THEN
  act1   :   crashed≔TRUE
  act2   :   modeG≔Off
  act3   :   modeGP≔Off
END

repare   ≙
  extended
STATUS
  ordinary
REFINES
  repare
ANY
  modeg
WHERE
  grd1   :   crashed=TRUE
             power=FALSE ∨  (onAC=FALSE ∧ (switchover=FALSE ∨ batLev=0 ∨ batFail=TRUE))
  grd2   :   ⇒
             modeg=Off
  grd3   :   power=TRUE ∧ (onAC=TRUE ∨ (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE))
```

```
            ⟹
            modeg=StartUp
  grd4   :  modeC∈Ventilation ⟹ modeg=modeC
            power=TRUE ∧ (onAC=TRUE ∨ (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE))∧
  grd5   :  modeC∉Ventilation
            ⟹
            modeg=StartUp
THEN
  act1   :  crashed≔FALSE
  act2   :  modeG≔modeg
  act3   :  modeGP≔modeG
END

newPatient   ≙
  extended
STATUS
  ordinary
REFINES
  newPatient
WHEN
  grd1   :  modeG=Start
  grd2   :  modeC=SelfTest
THEN
  act1   :  modeG≔SelfTest
  act2   :  modeGP≔modeG
END

resumeVent   ≙
  extended
STATUS
  ordinary
REFINES
  resumeVent
WHEN
  grd1   :  modeG=Start
  grd2   :  modeC=SelfTest
THEN
  act1   :  modeG≔Menu
  act2   :  modeGP≔modeG
  act3   :  modeC≔VentilationOff
  act4   :  modeCP≔modeC
END

runAbortSelfTest   ≙
  extended
STATUS
  ordinary
REFINES
  runAbortSelfTest
WHEN
  grd1   :  modeG=SelfTest
  grd2   :  modeC=SelfTest
THEN
  act1   :  modeG≔SelfTest
  act2   :  modeGP≔modeG
END

selfTestPassed   ≙          //   CONT4-4.1
  extended
STATUS
  ordinary
REFINES
  selfTestPassed
WHEN
  grd1   :  modeG=SelfTest
  grd2   :  modeC=SelfTest
THEN
  act1   :  modeG≔Menu
  act2   :  modeGP≔modeG
  act3   :  modeC≔VentilationOff
  act4   :  modeCP≔VentilationOff
END
```

```
setParam   ≙
  extended
STATUS
  ordinary
REFINES
  setParam
WHEN
  grd1   :   modeG∈{Menu}∪Ventilation
THEN
  act1   :   modeG≔Settings
  act2   :   modeGP≔modeG
END

saveBackAbort   ≙
  extended
STATUS
  ordinary
REFINES
  saveBackAbort
ANY
  modeg
  modec
WHERE
  grd1   :   modeG=Settings ∧ modeg∈ModesG
  grd2   :   modeg∈Ventilation∪{Menu}
  grd3   :   modeC≠FailSafe
  grd4   :   modec∈ModesC
  grd5   :   modeC∈Ventilation ⇒ modec∈Ventilation ∧ modeg=modec
  grd6   :   modeC∉Ventilation ⇒modec=modeC ∧ modeg=Menu
THEN
  act1   :   modeG≔modeg
  act2   :   modeGP≔modeG
  act3   :   modeC≔modec
  act4   :   modeCP≔modeC
END

startStopPCVPSV   ≙
  extended
STATUS
  ordinary
REFINES
  startStopPCVPSV
ANY
  modeg
  modec
WHERE
  grd1   :   modeG∈{Menu}∪Ventilation
  grd2   :   modeg∈Ventilation∪{Menu}
  grd3   :   modeG=Menu⇒modeg∈Ventilation
  grd4   :   modeG∈Ventilation⇒modeg=Menu
  grd5   :   modeC≠FailSafe
  grd6   :   modeg∈Ventilation ⇒ modec=modeg
  grd7   :   modeg∉Ventilation ⇒ modec=VentilationOff
THEN
  act1   :   modeG≔modeg
  act2   :   modeGP≔modeG
  act3   :   modeC≔modec
  act4   :   modeCP≔{TRUE↦modeC,FALSE↦modeCP}(bool(modeC∈Ventilation ∪{VentilationOff}))
END

changeMode   ≙
  extended
STATUS
  ordinary
REFINES
  changeMode
ANY
  modeg
  modec
WHERE
```

```
  grd1   :   modeG∈ Ventilation
  grd2   :   modeg∈Ventilation∪{Settings}
  grd3   :   modeC∈Ventilation\{FailSafe}
  grd4   :   modec∈Ventilation
THEN
  act1   :   modeG≔modeg
  act2   :   modeGP≔modeG
  act3   :   modeC≔modec
  act4   :   modeCP≔modeC
END

failExternalPower   ≙
  extended
STATUS
  ordinary
REFINES
  failExternalPower
ANY
  modeg
  modegp
  modec
  modecp
WHERE
  grd1   :   onAC=TRUE
  grd2   :   modeg ∈ ModesG ∧ modegp ∈ ModesG
  grd3   :   ¬(power=TRUE ∧ crashed=FALSE) ⇒ modeg=Off ∧ modegp=Off
             power=TRUE ∧ crashed=FALSE ⇒
             (
  grd4   :      ((switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE ⇒ modeg=modeG ∧ modegp=modeGP) ∧
                (¬(switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)⇒ modeg=Off ∧ modegp=modeG))
             )
  grd5   :   modec ∈ ModesC ∧ modecp ∈ ModesC
  grd6   :   power= FALSE ⇒ modec=modeC ∧ modecp=modeCP
              power= TRUE ⇒
              (
  grd7   :       ((switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE ⇒modec=modeC ∧ modecp=modeCP) ∧
                  (¬(switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)⇒modec=Off ∧ modecp=modeC))
                 )
THEN
  act1   :   onAC≔FALSE
  act2   :   modeG≔modeg
  act3   :   modeGP≔modegp
  act4   :   modeC≔modec
  act5   :   modeCP≔modecp
END

failSelfTest   ≙
STATUS
  ordinary
WHEN
  grd1   :   modeC=SelfTest
THEN
  act1   :   modeC≔FailSafe
  act2   :   modeCP≔modeC
END

progress   ≙
  extended
STATUS
  ordinary
REFINES
  progress
ANY
  step
  l
  modeg
  batf
  modec
WHERE
  grd1   :   step∈N1 ∧ l∈batteryRange ∧ batf∈BOOL
  grd2   :   batFail=TRUE ⇒ l=batLev
  grd3   :   onAC=TRUE ∧ batFail=FALSE ⇔ l=batLev+step
```

```
    grd4   :   onAC=FALSE ∧ batFail=FALSE ⇔ l=max({0,batLev−step})
               (onAC=TRUE ∨ (l>0 ∧ switchover=TRUE ∧ batf=FALSE)) ∧
               power=TRUE ∧  modeG=Off ∧
    grd5   :   crashed=FALSE
               ⇒
               modeg=StartUp
    grd6   :   (l=0 ∨ batf=TRUE ∨ switchover=FALSE) ∧ onAC=FALSE ⇒modeg=Off
               onAC=TRUE  ∨
    grd7   :   ((l>0 ∧ batLev>0) ∧ switchover=TRUE ∧ batf=FALSE)
               ⇒
               modeg=modeG
    grd8   :   modec∈{FailSafe,modeC, Off,StartUp}
    grd9   :   modec=FailSafe⇒modeC≠Off
               (onAC=TRUE ∨ (l>0 ∧ switchover=TRUE ∧ batf=FALSE)) ∧
    grd10  :   power=TRUE ∧  modeC=Off
               ⇒
               modec=StartUp
               (batLev>0 ∧ l>0)∨  switchover=FALSE ∨ onAC=TRUE ∨
    grd11  :   power=FALSE
               ⇒
               modec∈{modeC,FailSafe}
               (l=0 ∨ batf=TRUE ∨ switchover=FALSE) ∧ onAC=FALSE
    grd12  :   ⇒
               modec=Off
  THEN
    act1   :   curTime≔curTime+step
    act2   :   batLev≔l
    act3   :   modeG≔modeg
    act4   :   modeGP≔{TRUE↦modeGP, FALSE↦modeG}(bool(modeg=modeG))
    act5   :   batFail≔batf
    act6   :   modeC≔modec
    act7   :   modeCP≔{TRUE↦modeCP,FALSE↦modeC}(bool(modeC=modec))
  END

  switchoverFail   ≙
    extended
  STATUS
    ordinary
  REFINES
    switchoverFail
  ANY
    modeg
    modec
  WHERE
    grd1   :   switchover=TRUE
    grd2   :   onAC=FALSE⇒modeg=Off
    grd3   :   onAC=TRUE⇒modeg=modeG
    grd4   :   onAC=FALSE⇒modec=Off
    grd5   :   onAC=TRUE⇒modec=modeC
  THEN
    act1   :   switchover≔FALSE
    act2   :   modeG≔modeg
    act3   :   modeGP≔{TRUE↦modeG,FALSE↦modeGP}(bool(modeG≠modeg))
    act4   :   modeC≔modec
    act5   :   modeCP≔{TRUE↦modeC,FALSE↦modeCP}(bool(modeC≠modec))
  END

END
```

CONTEXT

ComParams

SETS

parameters
Cycles

CONSTANTS

comParams
PEEP
FiO2
PRM
timerRM
domcomParams
defcomParams
inspPauseMax
expPauseMax
resumeTime

AXIOMS

axm1 : comParams ⊆ parameters
axm2 : partition(comParams, {PEEP}, {FiO2},{PRM},{timerRM})
axm3 : domcomParams={PEEP↦5‥20,FiO2↦21‥100, PRM↦1‥50,timerRM↦1‥30}
axm4 : defcomParams={PEEP↦5,FiO2↦21, PRM↦20,timerRM↦10}
axm5 : domcomParams∈ comParams ⟶ ℙ1(ℕ1)
axm6 : defcomParams∈ comParams ⟶ ℕ1
axm7 : ∀p·p∈ comParams⇒defcomParams(p)∈domcomParams(p)
axm8 : ∀y· y∈comParams ⇒ (Ux·x∈comParams | {x ↦ defcomParams(x)})(y)∈domcomParams(y)
axm9 : inspPauseMax=40∗10
axm10 : expPauseMax=60∗10
axm11 : resumeTime∈ℕ1
axm12 : ∀ a,A, f,x· a∈ ℕ ∧ A⊆ parameters⇸ℕ ∧ x∈ A ∧ f∈ 0‥a⇸A ⇒ f⩤{a↦x}∈ 0‥a⇸A
axm13 : ∀a,b,c,d· a∈ℕ ∧ b∈ℕ1 ∧ c∈ℕ ∧ d∈ℕ1 ∧ a≥c ∧ b≤d ⇒ a÷b≥c÷d
axm14 : ∀a· a∈ℕ ⇒ a÷2∈ℕ
axm15 : finite(Cycles)

END

CONTEXT
PCVParams
EXTENDS
ComParams
CONSTANTS
pcvParams
dompcvParams
defpcvParams
RRPCV
IEPCV
PinspPCV
ITSPCV
AXIOMS
axm1 : pcvParams⊆parameters
axm2 : partition(pcvParams,{RRPCV}, {IEPCV}, {PinspPCV}, {ITSPCV})
axm3 : dompcvParams∈ pcvParams ⟶ ℙ1(N1)
axm4 : dompcvParams={RRPCV↦4‥50,IEPCV↦1‥4,PinspPCV↦2‥50,ITSPCV↦1‥9}
axm5 : defpcvParams∈ pcvParams ⟶ ℕ1
axm6 : ∀p·p∈ pcvParams⇒defpcvParams(p)∈dompcvParams(p)
axm7 : defpcvParams={RRPCV↦12,IEPCV↦2,PinspPCV↦15,ITSPCV↦3}
axm12 : ∀a,b· a∈ ℕ ∧ b∈ ℕ ∧ a>0 ∧ b>0 ⇒a÷b≥0
axm8 : pcvParams∩comParams=∅
axm9 : card(pcvParams)=4
axm10 : ∀y· y∈pcvParams
⇒
(Ux·x∈pcvParams | {x ↦ defpcvParams(x)})(y)∈dompcvParams(y)
axm11 : ∀a,b· a∈ ℕ ∧ b∈ ℕ ∧ a>0 ∧ b>0 ⇒a∗b>0
END

CONTEXT
PSVParams
EXTENDS
PCVParams
CONSTANTS
psvParams
PinspPSV
ITSPSV
ETS
ApneaLag
RRAP
IEAP
PinspAP
dompsvParams
defpsvParams
max_insp_time_psv
VERange
AXIOMS
axm1 : psvParams⊆parameters
axm2 : partition(psvParams,{PinspPSV},{ITSPSV},{ETS},{ApneaLag},{RRAP},{PinspAP})
axm3 : partition(parameters, comParams,pcvParams,psvParams)
axm4 : dompsvParams∈ psvParams ⟶ ℙ1(ℕ1)
axm5 : dompsvParams={PinspPSV↦2‥50, ITSPSV↦1‥9,ETS↦5‥60,ApneaLag↦100‥600,RRAP↦4‥50,PinspAP↦2‥50}
axm6 : IEAP∈ ℕ ∧ IEAP=2
axm7 : defpsvParams∈ psvParams \{RRAP,PinspAP}⟶ ℕ1
axm8 : ∀p·p∈ dom(defpsvParams)⇒defpsvParams(p)∈dompsvParams(p)
axm9 : {PinspPSV↦15, ITSPSV↦3,ETS↦30,ApneaLag↦300}⊆defpsvParams
axm10 : max_insp_time_psv∈ ℕ ∧ max_insp_time_psv=70
axm11 : VERange⊆ℕ
END

MACHINE
Ventilation
REFINES
ContSM
SEES
ContStates
PSVParams
VARIABLES
power
onAC
batLev
switchover
crashed
modeG
modeGP
modeC
modeCP
batFail
PCV2PSV
comParamsValC
comParamsValG
pcvParamsValC
pcvParamsValG
psvParamsValC
psvParamsValG
curTime
offT
INVARIANTS

inv2	:	comParamsValC∈ 0‥curTime⇸(comParams⟶ℕ1)
inv3	:	comParamsValG∈ 0‥curTime ⇸(comParams⟶ℕ1)
inv4	:	pcvParamsValC∈ 0‥curTime⇸(pcvParams⟶ℕ1)
inv5	:	pcvParamsValG∈ 0‥curTime ⇸(pcvParams⟶ℕ1)
inv6	:	psvParamsValC∈ 0‥curTime⇸(psvParams⇸ℕ1) ∧ (∀x· x∈ran(psvParamsValC)⇒ psvParams\{RRAP,PinspAP}⊆dom(x))
inv7	:	psvParamsValG∈ 0‥curTime ⇸(psvParams⇸ℕ1) ∧ (∀x· x∈ran(psvParamsValG)⇒ psvParams\{RRAP,PinspAP}⊆dom(x))
inv8	:	modeG∉{Off,StartUp} ⇒ pcvParamsValG≠ø ∧ psvParamsValG≠ø ∧ comParamsValG≠ø
inv9	:	modeC∉{Off,StartUp, FailSafe} ⇒ pcvParamsValC≠ø ∧ psvParamsValC≠ø ∧ comParamsValC≠ø
inv10	:	offT∈ ℕ ∧ offT ≤curTime
inv11	:	offT>0 ⇒ comParamsValC≠ø ∧ pcvParamsValC≠ø ∧ psvParamsValC≠ø
inv12	:	(comParamsValC≠ø ⇔ pcvParamsValC≠ø) ∧ (pcvParamsValC≠ø⇔psvParamsValC≠ø)
inv13	:	dom(comParamsValC)=dom(pcvParamsValC) ∧ dom(pcvParamsValC)=dom(psvParamsValC)
inv14	:	PCV2PSV∈ BOOL ∧ PCV2PSV=TRUE ⇒ modeG=Settings
inv16	:	PCV2PSV=TRUE ⇒ modeC∈{PCV,FailSafe}
inv17	:	∀t· t∈ dom(pcvParamsValC) ⇒ ((pcvParamsValC(t))(RRPCV) ∗(1 +(pcvParamsValC(t))(IEPCV)))>0
inv18	:	modeC=PSV ⇒ dom(psvParamsValC(max(dom(psvParamsValC))))=psvParams ∧ (psvParamsValC(max(dom(psvParamsValC))))(ApneaLag)≥max_insp_time_psv÷2
inv19	:	modeC=PCV⇒ 10∗60∗(pcvParamsValC(max(dom(pcvParamsValC))))(IEPCV)÷ ((pcvParamsValC(max(dom(pcvParamsValC))))(RRPCV)∗ (1 +(pcvParamsValC(max(dom(pcvParamsValC))))(IEPCV))) >0

EVENTS
INITIALISATION ≙
extended
STATUS
ordinary

```
BEGIN
  act1   :   power≔FALSE
  act2   :   batLev:∈batteryRange
  act3   :   onAC≔TRUE
  act4   :   switchover≔TRUE
  act5   :   crashed≔FALSE
  act6   :   modeG≔Off
  act7   :   modeGP≔Off
  act8   :   curTime≔0
  act9   :   batFail:∈BOOL
  act10   :   modeC≔Off
  act11   :   modeCP≔Off
  act12   :   comParamsValC≔ø
  act13   :   comParamsValG≔ø
  act14   :   pcvParamsValC≔ø
  act15   :   pcvParamsValG≔ø
  act16   :   psvParamsValG≔ø
  act17   :   psvParamsValC≔ø
  act18   :   offT≔0
  act19   :   PCV2PSV≔FALSE
END

powerON   ≙          //   CONT.2
  extended
STATUS
  ordinary
REFINES
  powerON
WHEN
  grd1   :   power=FALSE
  grd2   :   onAC=TRUE ∨ (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)
THEN
  act1   :   power≔TRUE
  act2   :   modeG≔{FALSE↦StartUp, TRUE↦Off}(crashed)
  act3   :   modeGP≔modeG
  act4   :   modeC≔StartUp
  act5   :   modeCP≔modeC
END

powerOff   ≙
  extended
STATUS
  ordinary
REFINES
  powerOff
ANY
  val
WHERE
  grd1   :   power=TRUE
             comParamsValC=ø ∨  comParamsValC(max(dom(comParamsValC)))=defcomParams
  grd2   :   ⇒
             val=0
             comParamsValC≠ø ∧
  grd3   :   comParamsValC(max(dom(comParamsValC)))≠defcomParams
             ⇒
             val=curTime
THEN
  act1   :   power≔FALSE
  act2   :   modeG≔Off
  act3   :   modeGP≔modeG
  act4   :   modeC≔Off
  act5   :   modeCP≔modeC
  act6   :   offT≔val
  act7   :   PCV2PSV≔FALSE
END

startUpEndedGui   ≙
  extended
STATUS
  ordinary
REFINES
```

```
  startUpEndedGui
ANY
  modeg
  pcvG
  comG
  psvG
WHERE
  grd1   :   modeG=StartUp
  grd2   :   modeg∈{Start}∪Ventilation
  grd3   :   modeC∈Ventilation ⇒ modeg=modeC
  grd4   :   modeC∉Ventilation ⇒ modeg=Start
             modeC∈Ventilation ∨ (offT≠0 ∧ curTime−offT ≤ resumeTime)
             ⇒
                comG=comParamsValG ⩤
                                {curTime ↦comParamsValC(max(dom(comParamsValC)))} ∧
  grd5   :      pcvG=pcvParamsValG ⩤
                                {curTime ↦pcvParamsValC(max(dom(pcvParamsValC)))} ∧
                psvG=psvParamsValG ⩤
                                {curTime ↦psvParamsValC(max(dom(psvParamsValC)))}
             modeC∉Ventilation ∧ (offT=0 ∨  curTime−offT > resumeTime)
             ⇒
                comG=comParamsValG ⩤
                                {curTime ↦defcomParams} ∧
  grd6   :      pcvG=pcvParamsValG ⩤
                                {curTime ↦defpcvParams} ∧
                psvG=psvParamsValG ⩤
                                {curTime ↦defpsvParams}
THEN
  act1   :   modeG≔modeg
  act2   :   modeGP≔modeG
  act3   :   comParamsValG≔comG
  act4   :   pcvParamsValG≔pcvG
  act5   :   psvParamsValG≔psvG
END

startUpEndedCont   ≙
  extended
STATUS
  ordinary
REFINES
  startUpEndedCont
ANY
  comC
  pcvC
  psvC
WHERE
  grd1   :   modeC=StartUp
             offT=0 ∨  curTime−offT > resumeTime
             ⇒
             comC=comParamsValC ⩤
                                {curTime ↦defcomParams} ∧
  grd2   :   pcvC=pcvParamsValC ⩤
                                {curTime ↦defpcvParams} ∧
             psvC=psvParamsValC ⩤
                                {curTime ↦defpsvParams}
             offT≠0 ∧ curTime−offT ≤ resumeTime
             ⇒
             comC=comParamsValC ⩤
                                {curTime ↦comParamsValC(max(dom(comParamsValC)))} ∧
  grd3   :   pcvC=pcvParamsValC ⩤
                                {curTime ↦pcvParamsValC(max(dom(pcvParamsValC)))} ∧
             psvC=psvParamsValC ⩤
                                {curTime ↦psvParamsValC(max(dom(psvParamsValC)))}
THEN
  act1   :   modeC≔SelfTest
  act2   :   modeCP≔modeC
  act3   :   comParamsValC≔comC
  act4   :   pcvParamsValC≔pcvC
  act5   :   psvParamsValC≔psvC
END

crash   ≙
  extended
```

```
STATUS
  ordinary
REFINES
  crash
WHEN
  grd1   :   crashed=FALSE
THEN
  act1   :   crashed≔TRUE
  act2   :   modeG≔Off
  act3   :   modeGP≔Off
  act4   :   PCV2PSV≔FALSE
END

repare   ≙
  extended
STATUS
  ordinary
REFINES
  repare
ANY
  modeg
WHERE
  grd1   :   crashed=TRUE
             power=FALSE ∨  (onAC=FALSE ∧ (switchover=FALSE ∨ batLev=0 ∨ batFail=TRUE))
  grd2   :   ⟹
             modeg=Off
             power=TRUE ∧ (onAC=TRUE ∨ (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE))
  grd3   :   ⟹
             modeg=StartUp
  grd4   :   modeC∈Ventilation ⟹ modeg=modeC
             power=TRUE ∧ (onAC=TRUE ∨ (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE))∧
  grd5   :   modeC∉Ventilation
             ⟹
             modeg=StartUp
THEN
  act1   :   crashed≔FALSE
  act2   :   modeG≔modeg
  act3   :   modeGP≔modeG
END

newPatient   ≙
  extended
STATUS
  ordinary
REFINES
  newPatient
WHEN
  grd1   :   modeG=Start
  grd2   :   modeC=SelfTest
THEN
  act1   :   modeG≔SelfTest
  act2   :   modeGP≔modeG
END

resumeVent   ≙
  extended
STATUS
  ordinary
REFINES
  resumeVent
WHEN
  grd1   :   modeG=Start
  grd2   :   modeC=SelfTest
  grd3   :   offT≠0 ∧ curTime−offT ≤ resumeTime
THEN
  act1   :   modeG≔Menu
  act2   :   modeGP≔modeG
  act3   :   modeC≔VentilationOff
  act4   :   modeCP≔modeC
END
```

```
runAbortSelfTest   ≙
  extended
STATUS
  ordinary
REFINES
  runAbortSelfTest
WHEN
  grd1   :   modeG=SelfTest
  grd2   :   modeC=SelfTest
THEN
  act1   :   modeG≔SelfTest
  act2   :   modeGP≔modeG
END

selfTestPassed   ≙          //   CONT4-4.1
  extended
STATUS
  ordinary
REFINES
  selfTestPassed
WHEN
  grd1   :   modeG=SelfTest
  grd2   :   modeC=SelfTest
THEN
  act1   :   modeG≔Menu
  act2   :   modeGP≔modeG
  act3   :   modeC≔VentilationOff
  act4   :   modeCP≔VentilationOff
END

setParam   ≙
  extended
STATUS
  ordinary
REFINES
  setParam
WHEN
  grd1   :   modeG∈{Menu}∪Ventilation
THEN
  act1   :   modeG≔Settings
  act2   :   modeGP≔modeG
END

settingParams   ≙
STATUS
  ordinary
ANY
  psvP
  pcvP
  comP
WHERE
  grd1   :   modeG=Settings
  grd2   :   max(dom(psvParamsValG))<curTime
  grd3   :   psvP∈ psvParams⇸ℕ1
  grd4   :   psvParams\{RRAP,PinspAP}⊆dom(psvP)
  grd5   :   pcvP∈ pcvParams⟶ℕ1
  grd6   :   comP∈ comParams⟶ℕ1
THEN
  act1   :   psvParamsValG(curTime)≔psvP
  act2   :   pcvParamsValG(curTime)≔pcvP
  act3   :   comParamsValG(curTime)≔comP
END

saveBackAbort   ≙
  extended
STATUS
  ordinary
REFINES
  saveBackAbort
ANY
  modeg
```

```
  modec
  sv
  pcvC
  psvC
  comC
  pcvG
  psvG
  comG
WHERE
  grd1   :   modeG=Settings ∧ modeg∈ModesG
  grd2   :   modeg∈Ventilation∪{Menu}
  grd3   :   modeC≠FailSafe
  grd4   :   modec∈ModesC
  grd5   :   modeC∈Ventilation ⇒ modec∈Ventilation ∧ modeg=modec
  grd6   :   modeC∉Ventilation ⇒modec=modeC ∧ modeg=Menu
  grd7   :   sv∈{TRUE,FALSE}
             psvG={TRUE↦
  grd8   :   psvParamsValG(max(dom(psvParamsValG))),
              FALSE↦psvParamsValC(max(dom(psvParamsValC)))}(sv)
             psvC={TRUE↦
  grd9   :   psvParamsValG(max(dom(psvParamsValG))),
              FALSE↦psvParamsValC(max(dom(psvParamsValC)))}(sv)
              pcvG={TRUE↦
  grd10   :   pcvParamsValG(max(dom(pcvParamsValG))),
               FALSE↦pcvParamsValC(max(dom(pcvParamsValC)))}(sv)
              pcvC={TRUE↦
  grd11   :   pcvParamsValG(max(dom(pcvParamsValG))),
               FALSE↦pcvParamsValC(max(dom(pcvParamsValC)))}(sv)
              comG={TRUE↦
  grd12   :   comParamsValG(max(dom(comParamsValG))),
               FALSE↦comParamsValC(max(dom(comParamsValC)))}(sv)
              comC={TRUE↦
  grd13   :   comParamsValG(max(dom(comParamsValG))),
               FALSE↦comParamsValC(max(dom(comParamsValC)))}(sv)
  grd14   :   PCV2PSV=TRUE ∧ modeC=PCV⇒modec=PSV
  grd15   :   PCV2PSV=FALSE ∨ modeC≠PCV⇒modec=modeC
              modec=PSV
              ⇒
  grd16   :   dom(psvC)=psvParams ∧
              psvC(ApneaLag)≥max_insp_time_psv÷2
  grd17   :   modeC≠modec ⇒PCV2PSV=TRUE ∧ modeC=PCV ∧ modec=PSV
              modec=PCV∧ sv=TRUE⇒
  grd18   :   10∗60∗pcvC(IEPCV)÷ (pcvC(RRPCV)∗(1 +pcvC(IEPCV))) >0
  grd19   :   curTime∉dom(pcvParamsValC)
THEN
  act1   :   modeG≔modeg
  act2   :   modeGP≔modeG
  act3   :   modeC≔modec
  act4   :   modeCP≔modeC
  act5   :   psvParamsValG(curTime)≔psvG
  act6   :   psvParamsValC(curTime)≔psvC
  act7   :   pcvParamsValG(curTime)≔pcvG
  act8   :   pcvParamsValC(curTime)≔pcvC
  act9   :   comParamsValG(curTime)≔comG
  act10   :   comParamsValC(curTime)≔comC
  act11   :   PCV2PSV≔FALSE
END

startStopPCVPSV   ≙
  extended
STATUS
  ordinary
REFINES
  startStopPCVPSV
ANY
  modeg
  modec
WHERE
  grd1   :   modeG∈{Menu}∪Ventilation
  grd2   :   modeg∈Ventilation∪{Menu}
  grd3   :   modeG=Menu⇒modeg∈Ventilation
```

```
  grd4   :   modeG∈Ventilation⇒modeg=Menu
  grd5   :   modeC≠FailSafe
  grd6   :   modeg∈Ventilation ⇒ modec=modeg
  grd7   :   modeg∉Ventilation ⇒ modec=VentilationOff
             modeg=PSV
  grd8   :   ⇒
             dom(psvParamsValC(max(dom(psvParamsValC))))=psvParams
             modeg=PSV
  grd9   :   ⇒
             (psvParamsValC(max(dom(psvParamsValC))))(ApneaLag)≥max_insp_time_psv÷2
              modec=PCV⇒
  grd10  :    10∗60∗(pcvParamsValC(max(dom(pcvParamsValC))))(IEPCV)÷
                ((pcvParamsValC(max(dom(pcvParamsValC))))(RRPCV)∗
              (1 +(pcvParamsValC(max(dom(pcvParamsValC))))(IEPCV))) >0
THEN
  act1   :   modeG≔modeg
  act2   :   modeGP≔modeG
  act3   :   modeC≔modec
  act4   :   modeCP≔{TRUE↦modeC,FALSE↦modeCP}(bool(modeC∈Ventilation ∪{VentilationOff}))
END

moveToPSV   ≙
  extended
STATUS
  ordinary
REFINES
  changeMode
ANY
  modeg
  modec
WHERE
  grd1   :   modeG∈ Ventilation
  grd2   :   modeg∈Ventilation∪{Settings}
  grd3   :   modeC∈Ventilation\{FailSafe}
  grd4   :   modec∈Ventilation
  grd5   :   modeG=PCV
  grd6   :   modeC=PCV
  grd7   :   modeg=Settings
  grd8   :   modec=modeC
THEN
  act1   :   modeG≔modeg
  act2   :   modeGP≔modeG
  act3   :   modeC≔modec
  act4   :   modeCP≔modeC
  act5   :   PCV2PSV≔TRUE
END

changeMode   ≙
  extended
STATUS
  ordinary
REFINES
  changeMode
ANY
  modeg
  modec
  pcv
WHERE
  grd1   :   modeG∈ Ventilation
  grd2   :   modeg∈Ventilation∪{Settings}
  grd3   :   modeC∈Ventilation\{FailSafe}
  grd4   :   modec∈Ventilation
  grd5   :   pcv ∈ pcvParams⟶ℕ1
  grd6   :   modeC=PCV ⇒ modec=PCV
             modeC=PSV ∧ modec=PCV
             ⇒
  grd7   :   pcv=pcvParamsValC(max(dom(pcvParamsValC)))⩤
             {RRPCV↦(psvParamsValC(max(dom(psvParamsValC))))(RRAP),
              PinspPCV↦(psvParamsValC(max(dom(psvParamsValC))))(PinspAP),
             IEPCV↦IEAP}
  grd8   :   modeC=PCV ∨ modec=PSV
```

```
            ⇒
            pcv=pcvParamsValC(max(dom(pcvParamsValC)))
            modec=PCV
  grd9   :  ⇒
            10*60*pcv(IEPCV)÷ (pcv(RRPCV)*(1 +pcv(IEPCV))) >0
THEN
  act1   :  modeG≔modeg
  act2   :  modeGP≔modeG
  act3   :  modeC≔modec
  act4   :  modeCP≔modeC
  act5   :  pcvParamsValC(curTime)≔pcv
  act6   :  psvParamsValC(curTime)≔psvParamsValC(max(dom(psvParamsValC)))
  act7   :  comParamsValC(curTime)≔comParamsValC(max(dom(comParamsValC)))
END

failExternalPower   ≙
  extended
STATUS
  ordinary
REFINES
  failExternalPower
ANY
  modeg
  modegp
  modec
  modecp
  pcv2psv
WHERE
  grd1   :  onAC=TRUE
  grd2   :  modeg ∈ ModesG ∧ modegp ∈ ModesG
  grd3   :  ¬(power=TRUE ∧ crashed=FALSE) ⇒ modeg=Off ∧ modegp=Off
            power=TRUE ∧ crashed=FALSE ⇒
            (
  grd4   :     ((switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE ⇒ modeg=modeG ∧ modegp=modeGP) ∧
               (¬(switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)⇒ modeg=Off ∧ modegp=modeG))
            )
  grd5   :  modec ∈ ModesC ∧ modecp ∈ ModesC
  grd6   :  power= FALSE ⇒ modec=modeC ∧ modecp=modeCP
             power= TRUE ⇒
             (
  grd7   :      ((switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE ⇒modec=modeC ∧ modecp=modeCP) ∧
                 (¬(switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)⇒modec=Off ∧ modecp=modeC))
                )
  grd8   :  ¬(power=TRUE ∧ crashed=FALSE) ⇒ pcv2psv=FALSE
            power=TRUE ∧ crashed=FALSE ⇒
            (
  grd9   :  ((switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE⇒ pcv2psv=PCV2PSV) ∧
            (¬(switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)⇒ pcv2psv=FALSE))
                    )
THEN
  act1   :  onAC≔FALSE
  act2   :  modeG≔modeg
  act3   :  modeGP≔modegp
  act4   :  modeC≔modec
  act5   :  modeCP≔modecp
  act6   :  PCV2PSV≔pcv2psv
END

failSelfTest   ≙
  extended
STATUS
  ordinary
REFINES
  failSelfTest
WHEN
  grd1   :  modeC=SelfTest
THEN
  act1   :  modeC≔FailSafe
  act2   :  modeCP≔modeC
END

progress   ≙
```

```
  extended
STATUS
  ordinary
REFINES
  progress
ANY
  step
  l
  modeg
  batf
  modec
  paw
WHERE
  grd1   :   step∈N1 ∧ l∈batteryRange ∧ batf∈BOOL
  grd2   :   batFail=TRUE ⟹ l=batLev
  grd3   :   onAC=TRUE ∧ batFail=FALSE ⟺ l=batLev+step
  grd4   :   onAC=FALSE ∧ batFail=FALSE ⟺ l=max({0,batLev−step})
  grd5   :   (onAC=TRUE ∨ (l>0 ∧ switchover=TRUE ∧ batf=FALSE)) ∧
             power=TRUE ∧  modeG=Off ∧
             crashed=FALSE
             ⟹
             modeg=StartUp
  grd6   :   (l=0 ∨ batf=TRUE ∨ switchover=FALSE) ∧ onAC=FALSE ⟹modeg=Off
  grd7   :   onAC=TRUE  ∨
             ((l>0 ∧ batLev>0) ∧ switchover=TRUE ∧ batf=FALSE)
             ⟹
             modeg=modeG
  grd8   :   modec∈{FailSafe,modeC, Off,StartUp}
  grd9   :   modec=FailSafe⟹modeC≠Off
  grd10  :   (onAC=TRUE ∨ (l>0 ∧ switchover=TRUE ∧ batf=FALSE)) ∧
             power=TRUE ∧  modeC=Off
             ⟹
             modec=StartUp
  grd11  :   (batLev>0 ∧ l>0)∨  switchover=FALSE ∨ onAC=TRUE ∨
             power=FALSE
             ⟹
             modec∈{modeC,FailSafe}
  grd12  :   (l=0 ∨ batf=TRUE ∨ switchover=FALSE) ∧ onAC=FALSE
             ⟹
             modec=Off
  grd13  :   step∈N1 ∧ paw∈N
THEN
  act1   :   curTime≔curTime+step
  act2   :   batLev≔l
  act3   :   modeG≔modeg
  act4   :   modeGP≔{TRUE↦modeGP, FALSE↦modeG}(bool(modeg=modeG))
  act5   :   batFail≔batf
  act6   :   modeC≔modec
  act7   :   modeCP≔{TRUE↦modeCP,FALSE↦modeC}(bool(modeC=modec))
  act8   :   PCV2PSV≔{TRUE↦PCV2PSV, FALSE↦FALSE}(bool(
             (batLev>0 ∧ l>0∧  switchover=FALSE) ∨ onAC=TRUE))
END

switchoverFail   ≙
  extended
STATUS
  ordinary
REFINES
  switchoverFail
ANY
  modeg
  modec
WHERE
  grd1   :   switchover=TRUE
  grd2   :   onAC=FALSE⟹modeg=Off
  grd3   :   onAC=TRUE⟹modeg=modeG
  grd4   :   onAC=FALSE⟹modec=Off
  grd5   :   onAC=TRUE⟹modec=modeC
THEN
  act1   :   switchover≔FALSE
  act2   :   modeG≔modeg
```

```
    act3   :   modeGP:={TRUE↦modeG,FALSE↦modeGP}(bool
               (modeG≠modeg))
    act4   :   modeC:=modec
    act5   :   modeCP:={TRUE↦modeC,FALSE↦modeCP}(bool(modeC≠modec))
    act6   :   PCV2PSV:={TRUE↦PCV2PSV, FALSE↦FALSE}(bool(onAC=TRUE))
  END

END
```

```
CONTEXT
   VentilStates
EXTENDS
   GuiStates
SETS
   ventSates
CONSTANTS
   inspBeg
   inspEnd
   expBeg
   expEnd
   inspPauseBeg
   inspPauseEnd
   expPauseBeg
   expPauseEnd
   rmBeg
   rmEnd
AXIOMS
   axm1   :   partition(ventSates,{inspBeg},{inspEnd},{expBeg},{expEnd},{inspPauseBeg},
                {inspPauseEnd},{expPauseBeg},{expPauseEnd},{rmBeg}, {rmEnd})
END
```

MACHINE
VentilationPhases
REFINES
Ventilation
SEES
ContStates
VentilStates
PSVParams
VARIABLES
power
onAC
batLev
switchover
crashed
modeG
modeGP
modeC
modeCP
PCV2PSV
comParamsValC
comParamsValG
pcvParamsValC
pcvParamsValG
psvParamsValC
psvParamsValG
batFail
curTime
offT
cycles
cycleMode
ventilPhase
inspEndT
inspBegT
inspPauseBegT
inspPauseEndT
PAW
VE
rmBegT
rmEndT
expBegT
expEndT
expPauseBegT
expPauseEndT
INVARIANTS
inv1 : cycles⊆Cycles ∧ cycleMode∈cycles⟶Ventilation
inv2 : cycles≠ø⟹pcvParamsValC≠ø
inv3 : inspBegT ∈ cycles ⟶ℕ
inv4 : inspEndT ∈ cycles ⟶ℕ
inv5 : ∀ c· c∈cycles ⟹ (∃ x·(x∈dom(pcvParamsValC) ∧ x∈dom(comParamsValC) ∧ x≤inspBegT(c)))
inv6 : ∀c· c∈ cycles ∧ cycleMode(c)=PSV ⟹ inspEndT(c)−inspBegT(c)≤ max_insp_time_psv ∧ inspEndT(c)>inspBegT(c)
inv7 : ventilPhase∈cycles ⟶ventSates
inv8 : ∀c· c∈ cycles ⟹ max({x·x∈dom(pcvParamsValC)∧x≤inspBegT(c) | x}) ∈ dom(pcvParamsValC) ∧ max({x·x∈dom(psvParamsValC)∧x≤inspBegT(c) | x}) ∈ dom(psvParamsValC)
inv9 : ∀t,c· c∈ cycles ∧ t∈ ℕ ∧ t=max({x|x∈ dom(pcvParamsValC) ∧ x≤ inspBegT(c)}) ∧ cycleMode(c)=PCV ⟹ inspEndT(c)−inspBegT(c)=10∗60∗(pcvParamsValC(t))(IEPCV)÷ ((pcvParamsValC(t))(RRPCV)∗(1 +(pcvParamsValC(t))(IEPCV)))
inv10 :

```
           ∀c· c∈ cycles ∧ ventilPhase(c)≠inspBeg
           ⇒
           inspBegT(c) <curTime
           modeG=Settings
inv11   :  ⇒
           modeC∈Ventilationu{VentilationOff,FailSafe}
           ∀t,c· c∈ cycles ∧ t∈ ℕ ∧ t=max({x|x∈ dom(pcvParamsValC) ∧ x≤  inspBegT(c)}) ∧
           cycleMode(c)=PCV
inv12   :  ⇒
           10∗60∗(pcvParamsValC(t))(IEPCV)÷
             ((pcvParamsValC(t))(RRPCV)∗(1 +(pcvParamsValC(t))(IEPCV))) >0
           ∀c· c∈cycles
inv13   :  ⇒
           inspBegT(c)<inspEndT(c)
           ∀c· c∈cycles ∧ ventilPhase(c)≠inspBeg
inv14   :  ⇒
           inspEndT(c)≤curTime
inv15   :  PAW∈ 0‥curTime⇸ℕ
inv16   :  VE∈ 0‥curTime⇸ℕ
inv17   :  cycles≠ø⇒ PAW≠ø ∧ VE≠ø
           modeC∉Ventilationu{FailSafe}
inv18   :           ⇒
           cycles=ventilPhase~[{expEnd,expPauseEnd}]
           cycles≠ø
inv19   :   ⇒
           comParamsValC≠ø ∧ pcvParamsValC≠ø ∧ psvParamsValC≠ø
           ∀c· c∈cycles ∧ ventilPhase(c)=inspBeg
inv20   :  ⇒
           inspEndT(c)≥curTime
           ∀c· c∈cycles
inv21   :  ⇒
           inspBegT(c)≤curTime
inv22   :  inspPauseBegT∈ ventilPhase~[ventSates\{inspBeg}]⇸ℕ
inv23   :  inspPauseEndT∈dom(inspPauseBegT) ⟶ℕ
           ∀c· c∈dom(inspPauseBegT)⇒
inv24   :  inspPauseEndT(c)≥inspPauseBegT(c)∧
           inspPauseEndT(c)−inspPauseBegT(c)≤inspPauseMax
           ∀c·c∈dom(inspPauseBegT)
inv25   :  ⇒
           inspPauseBegT(c)≥inspEndT(c)
           ventilPhase~[{inspPauseBeg}] ⊆ dom(inspPauseBegT) ∧
           ventilPhase~[{inspPauseEnd}] ⊆ dom(inspPauseEndT) ∧
inv26   :  ventilPhase~[{inspEnd}] ∩ dom(inspPauseEndT)=ø ∧
           ventilPhase~[{inspBeg}] ∩ dom(inspPauseEndT)=ø
inv27   :  rmBegT∈ ventilPhase~[ventSates\{inspBeg}] ⇸ ℕ
inv28   :  rmEndT∈dom(rmBegT) ⟶ℕ
           ∀c,t· c∈dom(rmBegT) ∧
           t=max({x|x∈ dom(comParamsValC) ∧ x≤  inspBegT(c)}) ∧
inv29   :     cycleMode(c)=PCV
           ⇒
           rmEndT(c)− rmBegT(c)=(comParamsValC(t))(timerRM)
           ∀c·c∈dom(rmBegT)
inv30   :  ⇒
           rmBegT(c)≥ inspEndT(c)
           ∀c·c∈dom(rmBegT) ∧ c∈dom(inspPauseEndT)
inv31   :  ⇒
           rmBegT(c)≥ inspPauseEndT(c)
           ventilPhase~[{rmBeg}]⊆dom(rmBegT) ∧
inv32   :  ventilPhase~[{inspEnd,inspBeg,inspPauseBeg,inspPauseEnd}]∩dom(rmBegT)=ø
           ∀c· c∈ ventilPhase~[{inspPauseEnd}]
inv33   :  ⇒
           inspPauseEndT(c)≤curTime
           ∀c,t· c∈ cycles ∧ cycleMode(c)=PSV ∧
           t=max({x|x∈ dom(psvParamsValC) ∧ x≤  inspBegT(c)})
inv34   :  ⇒
           (dom(psvParamsValC(t))=psvParams ∧
           (psvParamsValC(t))(ApneaLag)≥max_insp_time_psv÷2)
           expBegT ∈ventilPhase~[{expBeg,expEnd,expPauseBeg,expPauseEnd}]
inv35   :  ⟶
           0‥curTime
inv36   :  expEndT ∈ dom(expBegT) ⟶ℙ1(N)
```

```
    inv37   :   ∀t,c· c∈ cycles ∧ t∈ ℕ ∧ c∈dom(expBegT) ∧
                t=max({x|x∈ dom(pcvParamsValC) ∧ x≤  inspBegT(c)}) ∧
                c∈dom(expBegT)  ∧ cycleMode(c)=PCV
                ⇒
                (∃ k·(k∈ ℕ ∧  expEndT(c)={k} ∧
                 k−expBegT(c)≤ 60÷((pcvParamsValC(t))(RRPCV) ∗
                                   (1 +(pcvParamsValC(t))(IEPCV)))))
                ∀t,c· c∈ cycles ∧ t∈ ℕ ∧
                t=max({x|x∈ dom(pcvParamsValC) ∧ x≤  inspBegT(c)}) ∧
                c∈dom(expBegT) ∧  cycleMode(c)=PSV
    inv38   :    ⇒
                (∃ k1,k2· k1∈ ℕ ∧ k2∈ℕ ∧ k1≤k2 ∧expEndT(c)=k1‥k2 ∧
                (k1−expBegT(c)≥ (inspEndT(c)−inspBegT(c))÷2)  ∧
                (k2−expBegT(c)≤((psvParamsValC(t)))(ApneaLag)))
    inv39   :   expPauseBegT∈ ventilPhase~[{expPauseBeg, expPauseEnd}]⟶ℕ
    inv40   :   expPauseEndT∈dom(expPauseBegT) ⟶ℕ
                ∀c· c∈dom(expPauseBegT)⇒
    inv41   :   expPauseEndT(c)≥expPauseBegT(c)∧
                expPauseEndT(c)−expPauseBegT(c)≤expPauseMax
                modeC∈Ventilation
                ⇒
    inv42   :   card({c|c∈cycles ∧
                        ventilPhase(c)∉{expEnd,expPauseEnd}})≤1
EVENTS
    INITIALISATION   ≙
      extended
    STATUS
      ordinary
    BEGIN
      act1   :   power≔FALSE
      act2   :   batLev:∈batteryRange
      act3   :   onAC≔TRUE
      act4   :   switchover≔TRUE
      act5   :   crashed≔FALSE
      act6   :   modeG≔Off
      act7   :   modeGP≔Off
      act8   :   curTime≔0
      act9   :   batFail:∈BOOL
      act10   :   modeC≔Off
      act11   :   modeCP≔Off
      act12   :   comParamsValC≔∅
      act13   :   comParamsValG≔∅
      act14   :   pcvParamsValC≔∅
      act15   :   pcvParamsValG≔∅
      act16   :   psvParamsValG≔∅
      act17   :   psvParamsValC≔∅
      act18   :   offT≔0
      act19   :   PCV2PSV≔FALSE
      act20   :   ventilPhase≔∅
      act21   :   inspBegT≔∅
      act22   :   inspEndT≔∅
      act23   :   cycles≔∅
      act24   :   cycleMode≔∅
      act25   :   PAW≔∅
      act26   :   VE≔∅
      act27   :   inspPauseBegT≔∅
      act28   :   inspPauseEndT≔∅
      act29   :   rmBegT≔∅
      act30   :   rmEndT≔∅
      act31   :   expBegT≔∅
      act32   :   expEndT≔∅
      act33   :   expPauseBegT≔∅
      act34   :   expPauseEndT≔∅
    END

    powerON   ≙          //   CONT.2
      extended
    STATUS
      ordinary
    REFINES
      powerON
    WHEN
```

```
  grd1   :   power=FALSE
  grd2   :   onAC=TRUE ∨ (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)
THEN
  act1   :   power≔TRUE
  act2   :   modeG≔{FALSE↦StartUp, TRUE↦Off}(crashed)
  act3   :   modeGP≔modeG
  act4   :   modeC≔StartUp
  act5   :   modeCP≔modeC
END

powerOff   ≙
  extended
STATUS
  ordinary
REFINES
  powerOff
ANY
  val
WHERE
  grd1   :   power=TRUE
             comParamsValC=∅ ∨  comParamsValC(max(dom(comParamsValC)))=defcomParams
  grd2   :   ⟹
             val=0
             comParamsValC≠∅ ∧
  grd3   :   comParamsValC(max(dom(comParamsValC)))≠defcomParams
             ⟹
             val=curTime
THEN
  act1   :   power≔FALSE
  act2   :   modeG≔Off
  act3   :   modeGP≔modeG
  act4   :   modeC≔Off
  act5   :   modeCP≔modeC
  act6   :   offT≔val
  act7   :   PCV2PSV≔FALSE
  act8   :   ventilPhase≔ventilPhase⩥{expEnd,expPauseEnd}
  act9   :   inspBegT≔ventilPhase~[{expEnd,expPauseEnd}]⩤inspBegT
  act10   :   inspEndT≔ventilPhase~[{expEnd,expPauseEnd}]⩤inspEndT
  act11   :   cycles≔ventilPhase~[{expEnd,expPauseEnd}]
  act12   :   cycleMode≔ ventilPhase~[{expEnd,expPauseEnd}]⩤cycleMode
  act13   :   inspPauseBegT≔ventilPhase~[{expEnd,expPauseEnd}]⩤inspPauseBegT
  act14   :   inspPauseEndT≔ventilPhase~[{expEnd,expPauseEnd}]⩤inspPauseEndT
  act15   :   rmBegT≔ventilPhase~[{expEnd,expPauseEnd}]⩤rmBegT
  act16   :   rmEndT≔ventilPhase~[{expEnd,expPauseEnd}]⩤rmEndT
  act17   :   expBegT≔ventilPhase~[{expEnd,expPauseEnd}]⩤expBegT
  act18   :   expEndT≔ventilPhase~[{expEnd,expPauseEnd}]⩤expEndT
  act19   :   expPauseBegT≔ventilPhase~[{expEnd,expPauseEnd}]⩤expPauseBegT
  act20   :   expPauseEndT≔ventilPhase~[{expEnd,expPauseEnd}]⩤expPauseEndT
END

startUpEndedGui   ≙
  extended
STATUS
  ordinary
REFINES
  startUpEndedGui
ANY
  modeg
  pcvG
  comG
  psvG
WHERE
  grd1   :   modeG=StartUp
  grd2   :   modeg∈{Start}∪Ventilation
  grd3   :   modeC∈Ventilation ⟹ modeg=modeC
  grd4   :   modeC∉Ventilation ⟹ modeg=Start
  grd5   :   modeC∈Ventilation ∨ (offT≠0 ∧ curTime−offT ≤ resumeTime)
             ⟹
                comG=comParamsValG ⩤
                                {curTime ↦comParamsValC(max(dom(comParamsValC)))} ∧
                pcvG=pcvParamsValG ⩤
```

```
                                {curTime ↦pcvParamsValC(max(dom(pcvParamsValC)))} ∧
            psvG=psvParamsValG ⩤
                                {curTime ↦psvParamsValC(max(dom(psvParamsValC)))}
          modeC∉Ventilation ∧ (offT=0 ∨  curTime−offT > resumeTime)
          ⇒
            comG=comParamsValG ⩤
  grd6  :                       {curTime ↦defcomParams} ∧
            pcvG=pcvParamsValG ⩤
                                {curTime ↦defpcvParams} ∧
            psvG=psvParamsValG ⩤
                                {curTime ↦defpsvParams}
THEN
  act1  :  modeG≔modeg
  act2  :  modeGP≔modeG
  act3  :  comParamsValG≔comG
  act4  :  pcvParamsValG≔pcvG
  act5  :  psvParamsValG≔psvG
END

startUpEndedCont   ≙
  extended
STATUS
  ordinary
REFINES
  startUpEndedCont
ANY
  comC
  pcvC
  psvC
WHERE
  grd1  :  modeC=StartUp
           offT=0 ∨  curTime−offT > resumeTime
           ⇒
           comC=comParamsValC ⩤
  grd2  :                       {curTime ↦defcomParams} ∧
           pcvC=pcvParamsValC ⩤
                                {curTime ↦defpcvParams} ∧
           psvC=psvParamsValC ⩤
                                {curTime ↦defpsvParams}
           offT≠0 ∧ curTime−offT ≤ resumeTime
           ⇒
           comC=comParamsValC ⩤
                                {curTime ↦comParamsValC(max(dom(comParamsValC)))} ∧
  grd3  :  pcvC=pcvParamsValC ⩤
                                {curTime ↦pcvParamsValC(max(dom(pcvParamsValC)))} ∧
           psvC=psvParamsValC ⩤
                                {curTime ↦psvParamsValC(max(dom(psvParamsValC)))}
THEN
  act1  :  modeC≔SelfTest
  act2  :  modeCP≔modeC
  act3  :  comParamsValC≔comC
  act4  :  pcvParamsValC≔pcvC
  act5  :  psvParamsValC≔psvC
END

crash   ≙
  extended
STATUS
  ordinary
REFINES
  crash
WHEN
  grd1  :  crashed=FALSE
THEN
  act1  :  crashed≔TRUE
  act2  :  modeG≔Off
  act3  :  modeGP≔Off
  act4  :  PCV2PSV≔FALSE
END

repare   ≙
  extended
STATUS
```

```
  ordinary
REFINES
  repare
ANY
  modeg
WHERE
  grd1   :   crashed=TRUE
  grd2   :   power=FALSE ∨  (onAC=FALSE ∧ (switchover=FALSE ∨ batLev=0 ∨ batFail=TRUE))
             ⇒
             modeg=Off
  grd3   :   power=TRUE ∧ (onAC=TRUE ∨ (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE))
             ⇒
             modeg=StartUp
  grd4   :   modeC∈Ventilation ⇒ modeg=modeC
  grd5   :   power=TRUE ∧ (onAC=TRUE ∨ (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE))∧
             modeC∉Ventilation
             ⇒
             modeg=StartUp
THEN
  act1   :   crashed′=FALSE
  act2   :   modeG′=modeg
  act3   :   modeGP′=modeG
END

newPatient   ≙
  extended
STATUS
  ordinary
REFINES
  newPatient
WHEN
  grd1   :   modeG=Start
  grd2   :   modeC=SelfTest
THEN
  act1   :   modeG′=SelfTest
  act2   :   modeGP′=modeG
END

resumeVent   ≙
  extended
STATUS
  ordinary
REFINES
  resumeVent
WHEN
  grd1   :   modeG=Start
  grd2   :   modeC=SelfTest
  grd3   :   offT≠0 ∧ curTime−offT ≤ resumeTime
THEN
  act1   :   modeG′=Menu
  act2   :   modeGP′=modeG
  act3   :   modeC′=VentilationOff
  act4   :   modeCP′=modeC
END

runAbortSelfTest   ≙
  extended
STATUS
  ordinary
REFINES
  runAbortSelfTest
WHEN
  grd1   :   modeG=SelfTest
  grd2   :   modeC=SelfTest
THEN
  act1   :   modeG′=SelfTest
  act2   :   modeGP′=modeG
END

selfTestPassed   ≙          //   CONT4-4.1
  extended
STATUS
```

```
  ordinary
REFINES
  selfTestPassed
WHEN
  grd1   :   modeG=SelfTest
  grd2   :   modeC=SelfTest
THEN
  act1   :   modeG≔Menu
  act2   :   modeGP≔modeG
  act3   :   modeC≔VentilationOff
  act4   :   modeCP≔VentilationOff
END

setParam   ≙
  extended
STATUS
  ordinary
REFINES
  setParam
WHEN
  grd1   :   modeG∈{Menu}∪Ventilation
THEN
  act1   :   modeG≔Settings
  act2   :   modeGP≔modeG
END

settingParams   ≙
  extended
STATUS
  ordinary
REFINES
  settingParams
ANY
  psvP
  pcvP
  comP
WHERE
  grd1   :   modeG=Settings
  grd2   :   max(dom(psvParamsValG))<curTime
  grd3   :   psvP∈ psvParams⇸N1
  grd4   :   psvParams\{RRAP,PinspAP}⊆dom(psvP)
  grd5   :   pcvP∈ pcvParams⟶N1
  grd6   :   comP∈ comParams⟶N1
THEN
  act1   :   psvParamsValG(curTime)≔psvP
  act2   :   pcvParamsValG(curTime)≔pcvP
  act3   :   comParamsValG(curTime)≔comP
END

saveBackAbort   ≙
  extended
STATUS
  ordinary
REFINES
  saveBackAbort
ANY
  modeg
  modec
  sv
  pcvC
  psvC
  comC
  pcvG
  psvG
  comG
WHERE
  grd1   :   modeG=Settings ∧ modeg∈ModesG
  grd2   :   modeg∈Ventilation∪{Menu}
  grd3   :   modeC≠FailSafe
  grd4   :   modec∈ModesC
  grd5   :   modeC∈Ventilation ⇒ modec∈Ventilation ∧ modeg=modec
```

```
  grd6   :   modeC∉Ventilation ⇒modec=modeC ∧ modeg=Menu
  grd7   :   sv∈{TRUE,FALSE}
             psvG={TRUE↦
  grd8   :   psvParamsValG(max(dom(psvParamsValG))),
              FALSE↦psvParamsValC(max(dom(psvParamsValC)))}(sv)
             psvC={TRUE↦
  grd9   :   psvParamsValG(max(dom(psvParamsValG))),
              FALSE↦psvParamsValC(max(dom(psvParamsValC)))}(sv)
              pcvG={TRUE↦
  grd10  :   pcvParamsValG(max(dom(pcvParamsValG))),
               FALSE↦pcvParamsValC(max(dom(pcvParamsValC)))}(sv)
              pcvC={TRUE↦
  grd11  :   pcvParamsValG(max(dom(pcvParamsValG))),
               FALSE↦pcvParamsValC(max(dom(pcvParamsValC)))}(sv)
              comG={TRUE↦
  grd12  :   comParamsValG(max(dom(comParamsValG))),
               FALSE↦comParamsValC(max(dom(comParamsValC)))}(sv)
              comC={TRUE↦
  grd13  :   comParamsValG(max(dom(comParamsValG))),
               FALSE↦comParamsValC(max(dom(comParamsValC)))}(sv)
  grd14  :   PCV2PSV=TRUE ∧ modeC=PCV⇒modec=PSV
  grd15  :   PCV2PSV=FALSE ∨ modeC≠PCV⇒modec=modeC
             modec=PSV
             ⇒
  grd16  :   dom(psvC)=psvParams ∧
             psvC(ApneaLag)≥max_insp_time_psv÷2
  grd17  :   modeC≠modec ⇒PCV2PSV=TRUE ∧ modeC=PCV ∧ modec=PSV
             modec=PCV∧ sv=TRUE⇒
  grd18  :   10*60*pcvC(IEPCV)÷ (pcvC(RRPCV)*(1 +pcvC(IEPCV))) >0
  grd19  :   curTime∉dom(pcvParamsValC)
  grd20  :   ∀c·c∈cycles⇒ inspBegT(c)≠curTime
THEN
  act1   :   modeG≔modeg
  act2   :   modeGP≔modeG
  act3   :   modeC≔modec
  act4   :   modeCP≔modeC
  act5   :   psvParamsValG(curTime)≔psvG
  act6   :   psvParamsValC(curTime)≔psvC
  act7   :   pcvParamsValG(curTime)≔pcvG
  act8   :   pcvParamsValC(curTime)≔pcvC
  act9   :   comParamsValG(curTime)≔comG
  act10  :   comParamsValC(curTime)≔comC
  act11  :   PCV2PSV≔FALSE
END

startStopPCVPSV   ≙
  extended
STATUS
  ordinary
REFINES
  startStopPCVPSV
ANY
  modeg
  modec
WHERE
  grd1   :   modeG∈{Menu}∪Ventilation
  grd2   :   modeg∈Ventilation∪{Menu}
  grd3   :   modeG=Menu⇒modeg∈Ventilation
  grd4   :   modeG∈Ventilation⇒modeg=Menu
  grd5   :   modeC≠FailSafe
  grd6   :   modeg∈Ventilation ⇒ modec=modeg
  grd7   :   modeg∉Ventilation ⇒ modec=VentilationOff
             modeg=PSV
  grd8   :   ⇒
             dom(psvParamsValC(max(dom(psvParamsValC))))=psvParams
             modeg=PSV
  grd9   :   ⇒
             (psvParamsValC(max(dom(psvParamsValC))))(ApneaLag)≥max_insp_time_psv÷2
  grd10  :   modec=PCV⇒
             10*60*(pcvParamsValC(max(dom(pcvParamsValC))))(IEPCV)÷
               ((pcvParamsValC(max(dom(pcvParamsValC))))(RRPCV)*
```

```
            (1 +(pcvParamsValC(max(dom(pcvParamsValC))))(IEPCV))) >0
  grd11   :   cycles=ventilPhase~[{expEnd,expPauseEnd}]
THEN
  act1    :   modeG≔modeg
  act2    :   modeGP≔modeG
  act3    :   modeC≔modec
  act4    :   modeCP≔{TRUE↦modeC,FALSE↦modeCP}(bool(modeC∈Ventilation ∪{VentilationOff}))
END

moveToPSV   ≙
  extended
STATUS
  ordinary
REFINES
  moveToPSV
ANY
  modeg
  modec
WHERE
  grd1    :   modeG∈ Ventilation
  grd2    :   modeg∈Ventilation∪{Settings}
  grd3    :   modeC∈Ventilation\{FailSafe}
  grd4    :   modec∈Ventilation
  grd5    :   modeG=PCV
  grd6    :   modeC=PCV
  grd7    :   modeg=Settings
  grd8    :   modec=modeC
THEN
  act1    :   modeG≔modeg
  act2    :   modeGP≔modeG
  act3    :   modeC≔modec
  act4    :   modeCP≔modeC
  act5    :   PCV2PSV≔TRUE
END

failExternalPower   ≙
  extended
STATUS
  ordinary
REFINES
  failExternalPower
ANY
  modeg
  modegp
  modec
  modecp
  pcv2psv
  cycs
WHERE
  grd1    :   onAC=TRUE
  grd2    :   modeg ∈ ModesG ∧ modegp ∈ ModesG
  grd3    :   ¬(power=TRUE ∧ crashed=FALSE) ⇒ modeg=Off ∧ modegp=Off
              power=TRUE ∧ crashed=FALSE ⇒
              (
  grd4    :      ((switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE ⇒ modeg=modeG ∧ modegp=modeGP) ∧
                 (¬(switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)⇒ modeg=Off ∧ modegp=modeG))
              )
  grd5    :   modec ∈ ModesC ∧ modecp ∈ ModesC
  grd6    :   power= FALSE ⇒ modec=modeC ∧ modecp=modeCP
               power= TRUE ⇒
               (
  grd7    :      ((switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE ⇒modec=modeC ∧ modecp=modeCP) ∧
                  (¬(switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)⇒modec=Off ∧ modecp=modeC))
                 )
  grd8    :   ¬(power=TRUE ∧ crashed=FALSE) ⇒ pcv2psv=FALSE
              power=TRUE ∧ crashed=FALSE ⇒
              (
  grd9    :   ((switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE⇒ pcv2psv=PCV2PSV) ∧
              (¬(switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)⇒ pcv2psv=FALSE))
                      )
  grd10   :   (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)
                ⇒
```

```
            cycs=cycles
            (switchover=FALSE ∨ batLev=0 ∨  batFail=TRUE)
  grd11   :  ⇒
            cycs=ventilPhase~[{expEnd,expPauseEnd}]
  grd12   :  cycs⊆cycles
THEN
  act1    :  onAC≔FALSE
  act2    :  modeG≔modeg
  act3    :  modeGP≔modegp
  act4    :  modeC≔modec
  act5    :  modeCP≔modecp
  act6    :  PCV2PSV≔pcv2psv
  act7    :  ventilPhase≔cycs◁ventilPhase
  act8    :  inspBegT≔cycs◁inspBegT
  act9    :  inspEndT≔cycs◁inspEndT
  act10   :  cycles≔cycs
  act11   :  cycleMode≔ cycs◁cycleMode
  act12   :  inspPauseBegT≔cycs◁inspPauseBegT
  act13   :  inspPauseEndT≔cycs◁inspPauseEndT
  act14   :  rmBegT≔cycs◁rmBegT
  act15   :  rmEndT≔cycs◁rmEndT
  act16   :  expBegT≔cycs◁expBegT
  act17   :  expEndT≔cycs◁expEndT
  act18   :  expPauseBegT≔cycs◁expPauseBegT
  act19   :  expPauseEndT≔cycs◁expPauseEndT
END

failSelfTest   ≙
  extended
STATUS
  ordinary
REFINES
  failSelfTest
WHEN
  grd1    :  modeC=SelfTest
THEN
  act1    :  modeC≔FailSafe
  act2    :  modeCP≔modeC
END

progress   ≙
  extended
STATUS
  ordinary
REFINES
  progress
ANY
  step
  l
  modeg
  batf
  modec
  paw
  cycs
WHERE
  grd1    :  step∈N1 ∧ l∈batteryRange ∧ batf∈BOOL
  grd2    :  batFail=TRUE ⇒ l=batLev
  grd3    :  onAC=TRUE ∧ batFail=FALSE ⇔ l=batLev+step
  grd4    :  onAC=FALSE ∧ batFail=FALSE ⇔ l=max({0,batLev−step})
  grd5    :  (onAC=TRUE ∨ (l>0 ∧ switchover=TRUE ∧ batf=FALSE)) ∧
             power=TRUE ∧  modeG=Off ∧
             crashed=FALSE
             ⇒
             modeg=StartUp
  grd6    :  (l=0 ∨ batf=TRUE ∨ switchover=FALSE) ∧ onAC=FALSE ⇒modeg=Off
  grd7    :  onAC=TRUE  ∨
             ((l>0 ∧ batLev>0) ∧ switchover=TRUE ∧ batf=FALSE)
             ⇒
             modeg=modeG
  grd8    :  modec∈{FailSafe,modeC, Off,StartUp}
  grd9    :  modec=FailSafe⇒modeC≠Off
```

```
  grd10   :   (onAC=TRUE ∨ (l>0 ∧ switchover=TRUE ∧ batf=FALSE)) ∧
              power=TRUE ∧  modeC=Off
              ⇒
              modec=StartUp
              (batLev>0 ∧ l>0)∨  switchover=FALSE ∨ onAC=TRUE ∨
  grd11   :   power=FALSE
              ⇒
              modec∈{modeC,FailSafe}
              (l=0 ∨ batf=TRUE ∨ switchover=FALSE) ∧ onAC=FALSE
  grd12   :   ⇒
              modec=Off
  grd13   :   step∈N1 ∧ paw∈N
              ∀c· c∈cycles ∧ ventilPhase(c)=inspBeg
  grd14   :   ⇒
              curTime+step≤inspEndT(c)
              ∀c· c∈cycles ∧ ventilPhase(c)=inspPauseBeg
  grd15   :   ⇒
              curTime+step≤inspPauseEndT(c)
              ∀c· c∈cycles ∧ ventilPhase(c)=rmBeg
  grd16   :   ⇒
              curTime+step≤ rmEndT(c)
              ∀c· c∈cycles ∧ ventilPhase(c)=expBeg
  grd17   :   ⇒
              curTime+step∈expEndT(c)
              ∀c· c∈cycles ∧ ventilPhase(c)=expPauseBeg
  grd18   :   ⇒
              curTime+step ≤ expPauseEndT(c)
  grd19   :   modec∉Ventilationu{FailSafe} ⇒ cycs=ventilPhase~[{expEnd,expPauseEnd}]
  grd20   :   modec∈Ventilationu{FailSafe}  ⇒ cycs=cycles
  grd21   :   cycs⊆cycles
THEN
  act1    :   curTime≔curTime+step
  act2    :   batLev≔l
  act3    :   modeG≔modeg
  act4    :   modeGP≔{TRUE↦modeGP, FALSE↦modeG}(bool(modeg=modeG))
  act5    :   batFail≔batf
  act6    :   modeC≔modec
  act7    :   modeCP≔{TRUE↦modeCP,FALSE↦modeC}(bool(modeC=modec))
  act8    :   PCV2PSV≔{TRUE↦PCV2PSV, FALSE↦FALSE}(bool(
              (batLev>0 ∧ l>0∧  switchover=FALSE) ∨ onAC=TRUE))
  act9    :   ventilPhase≔cycs◁ventilPhase
  act10   :    inspBegT≔cycs◁inspBegT
  act11   :    inspEndT≔cycs◁inspEndT
  act12   :    cycles≔cycs
  act13   :    cycleMode≔cycs◁cycleMode
  act14   :    inspPauseBegT≔cycs◁inspPauseBegT
  act15   :    inspPauseEndT≔cycs◁inspPauseEndT
  act16   :    rmBegT≔cycs◁rmBegT
  act17   :    rmEndT≔cycs◁rmEndT
  act18   :    expBegT≔cycs◁expBegT
  act19   :    expEndT≔cycs◁expEndT
  act20   :    expPauseBegT≔cycs◁expPauseBegT
  act21   :    expPauseEndT≔cycs◁expPauseEndT
END

switchoverFail   ≙
  extended
STATUS
  ordinary
REFINES
  switchoverFail
ANY
  modeg
  modec
  cycs
WHERE
  grd1    :   switchover=TRUE
  grd2    :   onAC=FALSE⇒modeg=Off
  grd3    :   onAC=TRUE⇒modeg=modeG
  grd4    :   onAC=FALSE⇒modec=Off
  grd5    :   onAC=TRUE⇒modec=modeC
```

```
  grd6   :   modec∉Ventilationu{FailSafe} ⟹ cycs=ventilPhase~
             [{expEnd,expPauseEnd}]
  grd7   :   modec∈Ventilationu{FailSafe}  ⟹ cycs=cycles
  grd8   :   cycs⊆cycles
THEN
  act1   :   switchover≔FALSE
  act2   :   modeG≔modeg
  act3   :   modeGP≔{TRUE↦modeG,FALSE↦modeGP}(bool(modeG≠modeg))
  act4   :   modeC≔modec
  act5   :   modeCP≔{TRUE↦modeC,FALSE↦modeCP}(bool(modeC≠modec))
  act6   :   PCV2PSV≔{TRUE↦PCV2PSV, FALSE↦FALSE}(bool(onAC=TRUE))
  act7   :   ventilPhase≔cycs⩤ventilPhase
  act8   :   inspBegT≔cycs⩤inspBegT
  act9   :   inspEndT≔cycs⩤inspEndT
  act10   :   cycles≔cycs
  act11   :   cycleMode≔cycs⩤cycleMode
  act12   :   inspPauseBegT≔cycs⩤inspPauseBegT
  act13   :   inspPauseEndT≔cycs⩤inspPauseEndT
  act14   :   rmBegT≔cycs⩤rmBegT
  act15   :   rmEndT≔cycs⩤rmEndT
  act16   :   expBegT≔cycs⩤expBegT
  act17   :   expEndT≔cycs⩤expEndT
  act18   :   expPauseBegT≔cycs⩤expPauseBegT
  act19   :   expPauseEndT≔cycs⩤expPauseEndT
END

inspStart   ≙
STATUS
  ordinary
ANY
  cy
  paw
  ve
  inspT
WHERE
  grd1   :   modeC∈Ventilation ∧ inspT∈ℕ
  grd2   :   cy∈Cycles\cycles
  grd3   :   ran(ventilPhase)⊆{expEnd,expPauseEnd}
  grd4   :   modeC=PCV⟹inspT=curTime + 10∗60∗
              (pcvParamsValC(max(dom(pcvParamsValC))))(IEPCV)÷
                    ((pcvParamsValC(max(dom(pcvParamsValC))))(RRPCV)∗
                  (1 +(pcvParamsValC(max(dom(pcvParamsValC))))(IEPCV)))
             modeC=PSV⟹
  grd5   :   inspT>curTime ∧
             inspT≤curTime + max_insp_time_psv
  grd6   :   paw∈ℕ ∧ ve∈ ℕ
  grd7   :   pcvParamsValC≠ø ∧ psvParamsValC≠ø ∧ comParamsValC≠ø
THEN
  act1   :   ventilPhase(cy)≔inspBeg
  act2   :   cycles≔cyclesu{cy}
  act3   :   inspBegT(cy)≔curTime
  act4   :   inspEndT(cy)≔inspT
  act5   :   cycleMode(cy)≔modeC
  act6   :   PAW(curTime)≔paw
  act7   :   VE(curTime)≔ve
END

inspEnd   ≙
STATUS
  ordinary
ANY
  cy
  ve
  peakVE
WHERE
  grd1   :   cy∈ cycles ∧ ventilPhase(cy)=inspBeg ∧ ve∈ℕ
  grd2   :   cycleMode(cy)=PSV⟹curTime>inspBegT(cy)
  grd3   :   cycleMode(cy)=PCV ⟹curTime=inspEndT(cy)
  grd4   :   peakVE=VE(max(dom(VE)))
  grd5   :   (curTime=inspEndT(cy)) ∨
             (cycleMode(cy)=PSV ⟹
```

```
            peakVE>ve ∧
            ve<(psvParamsValC(max(dom(psvParamsValC))))(ETS)∗peakVE)
THEN
  act1   :   ventilPhase(cy)≔inspEnd
  act2   :   VE(curTime)≔ve
  act3   :   inspEndT(cy)≔curTime
END

inspPauseStart   ≙
STATUS
  ordinary
ANY
  cy
WHERE
  grd1   :   cy∈cycles ∧ ventilPhase(cy)=inspEnd
THEN
  act1   :   ventilPhase(cy)≔inspPauseBeg
  act2   :   inspPauseBegT(cy)≔curTime
  act3   :   inspPauseEndT(cy)≔curTime+inspPauseMax
END

inspPauseEnd   ≙
STATUS
  ordinary
ANY
  cy
WHERE
  grd1   :   cy∈cycles ∧ ventilPhase(cy)=inspPauseBeg
  grd2   :   curTime≤inspPauseEndT(cy)
  grd3   :   curTime>inspPauseBegT(cy)
THEN
  act1   :   ventilPhase(cy)≔inspPauseEnd
  act2   :   inspPauseEndT(cy)≔curTime
END

rmStart   ≙
STATUS
  ordinary
ANY
  cy
  t
WHERE
  grd1   :   cy∈cycles
  grd2   :   ventilPhase(cy)∈{inspEnd, inspPauseEnd}
  grd3   :   t=max({x|x∈ dom(comParamsValC) ∧ x≤  inspBegT(cy)})
THEN
  act1   :   ventilPhase(cy)≔rmBeg
  act2   :   rmBegT(cy)≔curTime
  act3   :   rmEndT(cy)≔curTime+(comParamsValC(t))(timerRM)
END

rmEnd   ≙
STATUS
  ordinary
ANY
  cy
WHERE
  grd1   :   cy∈cycles
  grd2   :   ventilPhase(cy)=rmBeg
  grd3   :   curTime=rmEndT(cy)
THEN
  act1   :   ventilPhase(cy)≔rmEnd
END

expStart   ≙
STATUS
  ordinary
ANY
  cy
  t
  expT
```

```
WHERE
  grd1   :   cy∈cycles ∧  ventilPhase(cy)∈{inspEnd, inspPauseEnd,rmEnd}
  grd2   :   t=max({x|x∈ dom(pcvParamsValC) ∧ x≤  inspBegT(cy)})
  grd3   :   expT⊆ℕ
             cycleMode(cy)=PCV ⟹
  grd4   :   expT={curTime +60÷((pcvParamsValC(t))(RRPCV) *
                       (1 +(pcvParamsValC(t))(IEPCV)))}
             cycleMode(cy)=PSV ⟹
             expT=
  grd5   :   (curTime+(inspEndT(cy)−inspBegT(cy))÷2)
             ‥
             (curTime +((psvParamsValC(t)))(ApneaLag))
THEN
  act1   :   ventilPhase(cy):=expBeg
  act2   :   expBegT(cy):=curTime
  act3   :   expEndT(cy):=expT
END

expEnd   ≙
  extended
STATUS
  ordinary
REFINES
  changeMode
ANY
  modeg
  modec
  pcv
  paw
  cy
  lastTime
  t
WHERE
  grd1   :   modeG∈ Ventilation
  grd2   :   modeg∈Ventilation∪{Settings}
  grd3   :   modeC∈Ventilation\{FailSafe}
  grd4   :   modec∈Ventilation
  grd5   :   pcv ∈ pcvParams⟶ℕ1
  grd6   :   modeC=PCV ⟹ modec=PCV
             modeC=PSV ∧ modec=PCV
             ⟹
  grd7   :   pcv=pcvParamsValC(max(dom(pcvParamsValC)))⩤
             {RRPCV↦(psvParamsValC(max(dom(psvParamsValC))))(RRAP),
              PinspPCV↦(psvParamsValC(max(dom(psvParamsValC))))(PinspAP),
             IEPCV↦IEAP}
             modeC=PCV ∨ modec=PSV
  grd8   :   ⟹
             pcv=pcvParamsValC(max(dom(pcvParamsValC)))
             modec=PCV
  grd9   :   ⟹
             10*60*pcv(IEPCV)÷ (pcv(RRPCV)*(1 +pcv(IEPCV))) >0
  grd11  :   modeG≠Off⟹ modeg=modec
  grd12  :   cy∈cycles
  grd13  :   ventilPhase(cy)=expBeg
  grd14  :   lastTime= max(dom(PAW))
  grd15  :   paw∈ℕ
  grd16  :   t=max({x | x∈dom(psvParamsValC)∧x≤inspBegT(cy)})
             (
             cycleMode(cy)=PCV ∧
                 (curTime∈ expEndT(cy) ∨
                   (curTime<max(expEndT(cy)) ∧
                     (PAW(lastTime)<paw ∧ PAW(lastTime)−paw>(pcvParamsValC(t))(ITSPCV)))
                 )
  grd17  :   )
             ∨
             (
             cycleMode(cy)=PSV ∧
                 (curTime= max(expEndT(cy)) ∨
                   (curTime≥min(expEndT(cy)) ∧ curTime<max(expEndT(cy)) ∧ PAW(lastTime)<paw ∧ PAW
             (lastTime)−paw>(psvParamsValC(t))(ITSPSV)))
             )
  grd18  :   curTime=max(expEndT(cy))∧ cycleMode(cy)=PSV
```

```
                ⇒
                modec=PCV
                curTime≠max(expEndT(cy))∧ cycleMode(cy)=PSV
  grd19   :     ⇒
                modec=PSV
                cycleMode(cy)=PCV
  grd20   :     ⇒
                modec=PCV
THEN
  act1    :   modeG≔modeg
  act2    :   modeGP≔modeG
  act3    :   modeC≔modec
  act4    :   modeCP≔modeC
  act5    :   pcvParamsValC(curTime)≔pcv
  act6    :   psvParamsValC(curTime)≔psvParamsValC(max(dom(psvParamsValC)))
  act7    :   comParamsValC(curTime)≔comParamsValC(max(dom(comParamsValC)))
  act8    :   ventilPhase(cy)≔expEnd
  act9    :   PAW(curTime)≔paw
  act10   :    expEndT(cy)≔curTime‥curTime
END

expPauseStart   ≙
STATUS
  ordinary
ANY
  cy
WHERE
  grd1    :   cy∈cycles ∧ ventilPhase(cy)=expEnd
  grd2    :   modeC∈Ventilation
  grd3    :   ran(ventilPhase)⊆{expEnd,expPauseEnd}
THEN
  act1    :   ventilPhase(cy)≔expPauseBeg
  act2    :   expPauseBegT(cy)≔curTime
  act3    :   expPauseEndT(cy)≔curTime+expPauseMax
END

expPauseEnd   ≙
STATUS
  ordinary
ANY
  cy
WHERE
  grd1    :   cy∈cycles ∧ ventilPhase(cy)=expPauseBeg
  grd2    :   curTime≤expPauseEndT(cy)
  grd3    :   curTime>expPauseBegT(cy)
THEN
  act1    :   ventilPhase(cy)≔expPauseEnd
  act2    :   expPauseEndT(cy)≔curTime
END

END
```

MACHINE
Valves
REFINES
VentilationPhases
SEES
ContStates
VentilStates
PSVParams
VARIABLES
power
onAC
batLev
batFail
switchover
crashed
modeG
modeGP
modeC
modeCP
PCV2PSV
comParamsValC
comParamsValG
pcvParamsValC
pcvParamsValG
psvParamsValC
psvParamsValG
curTime
offT
cycles
cycleMode
ventilPhase
inspEndT
inspBegT
inspPauseBegT
inspPauseEndT
PAW
VE
rmBegT
rmEndT
expBegT
expEndT
expPauseBegT
expPauseEndT
inValve
outValve
inValveF
outValveF
INVARIANTS

inv1 : inValve ∈ BOOL ∧ outValve∈ BOOL ∧ inValveF ∈ BOOL ∧ outValveF∈ BOOL

inv3 : modeC≠Ventilation ∧ outValveF=FALSE ⇒ outValve=TRUE

inv4 : (∃ c· c ∈ cycles ∧ ventilPhase(c)∈{inspBeg,inspEnd}) ∧ inValveF=FALSE ∧ modeC∈Ventilation ⇒ inValve=TRUE

inv5 : (∃ c· c ∈ cycles ∧ ventilPhase(c)∈{inspBeg,inspEnd}) ∧ outValveF=FALSE ∧ modeC∈Ventilation ⇒ outValve=FALSE

inv6 : (∃ c· c ∈ cycles ∧ ventilPhase(c)=expBeg) ∧ inValveF=FALSE ⇒ inValve=FALSE

inv7 : (∃ c· c ∈ cycles ∧ ventilPhase(c)=expBeg) ∧ outValveF=FALSE ⇒ outValve=TRUE

inv8 : (∃ c· c ∈ cycles ∧ ventilPhase(c)∈ {expPauseBeg,inspPauseBeg,inspPauseEnd}) ∧ inValveF=FALSE

```
                ⇒
                inVal ve=FALSE
                (∃ c· c ∈ cycles ∧ ventilPhase(c)∈
                   {expPauseBeg,inspPauseBeg,inspPauseEnd}) ∧
    inv9   :       outValveF=FALSE ∧ modeC∈Ventilation
                ⇒
                outValve=FALSE
                (∃ c· c ∈ cycles ∧ ventilPhase(c)∈{rmBeg,rmEnd}) ∧ inValveF=FALSE ∧
    inv10  :    modeC∈Ventilation
                ⇒
                inValve=TRUE
                (∃ c· c ∈ cycles ∧ ventilPhase(c)∈{rmBeg,rmEnd}) ∧ outValveF=FALSE
    inv11  :    ∧ modeC∈Ventilation
                ⇒
                outValve=FALSE
EVENTS
    INITIALISATION   ≙
      extended
    STATUS
      ordinary
    BEGIN
      act1   :   power≔FALSE
      act2   :   batLev:∈batteryRange
      act3   :   onAC≔TRUE
      act4   :   switchover≔TRUE
      act5   :   crashed≔FALSE
      act6   :   modeG≔Off
      act7   :   modeGP≔Off
      act8   :   curTime≔0
      act9   :   batFail:∈BOOL
      act10   :   modeC≔Off
      act11   :   modeCP≔Off
      act12   :   comParamsValC≔∅
      act13   :   comParamsValG≔∅
      act14   :   pcvParamsValC≔∅
      act15   :   pcvParamsValG≔∅
      act16   :   psvParamsValG≔∅
      act17   :   psvParamsValC≔∅
      act18   :   offT≔0
      act19   :   PCV2PSV≔FALSE
      act20   :   ventilPhase≔∅
      act21   :   inspBegT≔∅
      act22   :   inspEndT≔∅
      act23   :   cycles≔∅
      act24   :   cycleMode≔∅
      act25   :   PAW≔∅
      act26   :   VE≔∅
      act27   :   inspPauseBegT≔∅
      act28   :   inspPauseEndT≔∅
      act29   :   rmBegT≔∅
      act30   :   rmEndT≔∅
      act31   :   expBegT≔∅
      act32   :   expEndT≔∅
      act33   :   expPauseBegT≔∅
      act34   :   expPauseEndT≔∅
      act35   :   inValve≔FALSE
      act36   :   outValve≔TRUE
      act37   :   inValveF≔FALSE
      act38   :   outValveF≔FALSE
    END

    powerON   ≙          //   CONT.2
      extended
    STATUS
      ordinary
    REFINES
      powerON
    WHEN
      grd1   :   power=FALSE
      grd2   :   onAC=TRUE ∨ (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)
    THEN
```

```
  act1   :   power≔TRUE
  act2   :   modeG≔{FALSE↦StartUp, TRUE↦Off}(crashed)
  act3   :   modeGP≔modeG
  act4   :   modeC≔StartUp
  act5   :   modeCP≔modeC
END

powerOff   ≙
  extended
STATUS
  ordinary
REFINES
  powerOff
ANY
  val
WHERE
  grd1   :   power=TRUE
             comParamsValC=∅ ∨  comParamsValC(max(dom(comParamsValC)))=defcomParams
  grd2   :   ⇒
             val=0
             comParamsValC≠∅ ∧
  grd3   :   comParamsValC(max(dom(comParamsValC)))≠defcomParams
             ⇒
             val=curTime
THEN
  act1   :   power≔FALSE
  act2   :   modeG≔Off
  act3   :   modeGP≔modeG
  act4   :   modeC≔Off
  act5   :   modeCP≔modeC
  act6   :   offT≔val
  act7   :   PCV2PSV≔FALSE
  act8   :   ventilPhase≔ventilPhase⩥{expEnd,expPauseEnd}
  act9   :   inspBegT≔ventilPhase~[{expEnd,expPauseEnd}]⩤inspBegT
  act10   :   inspEndT≔ventilPhase~[{expEnd,expPauseEnd}]⩤inspEndT
  act11   :   cycles≔ventilPhase~[{expEnd,expPauseEnd}]
  act12   :   cycleMode≔ ventilPhase~[{expEnd,expPauseEnd}]⩤cycleMode
  act13   :   inspPauseBegT≔ventilPhase~[{expEnd,expPauseEnd}]⩤inspPauseBegT
  act14   :   inspPauseEndT≔ventilPhase~[{expEnd,expPauseEnd}]⩤inspPauseEndT
  act15   :   rmBegT≔ventilPhase~[{expEnd,expPauseEnd}]⩤rmBegT
  act16   :   rmEndT≔ventilPhase~[{expEnd,expPauseEnd}]⩤rmEndT
  act17   :   expBegT≔ventilPhase~[{expEnd,expPauseEnd}]⩤expBegT
  act18   :   expEndT≔ventilPhase~[{expEnd,expPauseEnd}]⩤expEndT
  act19   :   expPauseBegT≔ventilPhase~[{expEnd,expPauseEnd}]⩤expPauseBegT
  act20   :   expPauseEndT≔ventilPhase~[{expEnd,expPauseEnd}]⩤expPauseEndT
  act21   :   inValve≔FALSE
  act22   :   outValve≔TRUE
END

startUpEndedGui   ≙
  extended
STATUS
  ordinary
REFINES
  startUpEndedGui
ANY
  modeg
  pcvG
  comG
  psvG
WHERE
  grd1   :   modeG=StartUp
  grd2   :   modeg∈{Start}∪Ventilation
  grd3   :   modeC∈Ventilation ⇒ modeg=modeC
  grd4   :   modeC∉Ventilation ⇒ modeg=Start
  grd5   :   modeC∈Ventilation ∨ (offT≠0 ∧ curTime−offT ≤ resumeTime)
             ⇒
                comG=comParamsValG ⩤
                                {curTime ↦comParamsValC(max(dom(comParamsValC)))} ∧
                pcvG=pcvParamsValG ⩤
                                {curTime ↦pcvParamsValC(max(dom(pcvParamsValC)))} ∧
                psvG=psvParamsValG ⩤
```

```
                                {curTime ↦psvParamsValC(max(dom(psvParamsValC)))}
            modeC≠Ventilation ∧ (offT=0 ∨  curTime−offT > resumeTime)
            ⇒
               comG=comParamsValG ⩤
                                {curTime ↦defcomParams} ∧
  grd6   :     pcvG=pcvParamsValG ⩤
                                {curTime ↦defpcvParams} ∧
               psvG=psvParamsValG ⩤
                                {curTime ↦defpsvParams}
THEN
  act1   :   modeG≔modeg
  act2   :   modeGP≔modeG
  act3   :   comParamsValG≔comG
  act4   :   pcvParamsValG≔pcvG
  act5   :   psvParamsValG≔psvG
END

startUpEndedCont   ≙
  extended
STATUS
  ordinary
REFINES
  startUpEndedCont
ANY
  comC
  pcvC
  psvC
WHERE
  grd1   :   modeC=StartUp
             offT=0 ∨  curTime−offT > resumeTime
             ⇒
             comC=comParamsValC ⩤
                                {curTime ↦defcomParams} ∧
  grd2   :   pcvC=pcvParamsValC ⩤
                                {curTime ↦defpcvParams} ∧
             psvC=psvParamsValC ⩤
                                {curTime ↦defpsvParams}
             offT≠0 ∧ curTime−offT ≤ resumeTime
             ⇒
             comC=comParamsValC ⩤
                                {curTime ↦comParamsValC(max(dom(comParamsValC)))} ∧
  grd3   :   pcvC=pcvParamsValC ⩤
                                {curTime ↦pcvParamsValC(max(dom(pcvParamsValC)))} ∧
             psvC=psvParamsValC ⩤
                                {curTime ↦psvParamsValC(max(dom(psvParamsValC)))}
THEN
  act1   :   modeC≔SelfTest
  act2   :   modeCP≔modeC
  act3   :   comParamsValC≔comC
  act4   :   pcvParamsValC≔pcvC
  act5   :   psvParamsValC≔psvC
END

crash   ≙
  extended
STATUS
  ordinary
REFINES
  crash
WHEN
  grd1   :   crashed=FALSE
THEN
  act1   :   crashed≔TRUE
  act2   :   modeG≔Off
  act3   :   modeGP≔Off
  act4   :   PCV2PSV≔FALSE
END

repare   ≙
  extended
STATUS
  ordinary
```

```
REFINES
  repare
ANY
  modeg
WHERE
  grd1   :   crashed=TRUE
  grd2   :   power=FALSE ∨  (onAC=FALSE ∧ (switchover=FALSE ∨ batLev=0 ∨ batFail=TRUE))
             ⇒
             modeg=Off
  grd3   :   power=TRUE ∧ (onAC=TRUE ∨ (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE))
             ⇒
             modeg=StartUp
  grd4   :   modeC∈Ventilation ⇒ modeg=modeC
  grd5   :   power=TRUE ∧ (onAC=TRUE ∨ (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE))∧
             modeC∉Ventilation
             ⇒
             modeg=StartUp
THEN
  act1   :   crashed′=FALSE
  act2   :   modeG≔modeg
  act3   :   modeGP≔modeG
END

newPatient   ≙
  extended
STATUS
  ordinary
REFINES
  newPatient
WHEN
  grd1   :   modeG=Start
  grd2   :   modeC=SelfTest
THEN
  act1   :   modeG≔SelfTest
  act2   :   modeGP≔modeG
END

resumeVent   ≙
  extended
STATUS
  ordinary
REFINES
  resumeVent
WHEN
  grd1   :   modeG=Start
  grd2   :   modeC=SelfTest
  grd3   :   offT≠0 ∧ curTime−offT ≤ resumeTime
THEN
  act1   :   modeG≔Menu
  act2   :   modeGP≔modeG
  act3   :   modeC≔VentilationOff
  act4   :   modeCP≔modeC
END

runAbortSelfTest   ≙
  extended
STATUS
  ordinary
REFINES
  runAbortSelfTest
WHEN
  grd1   :   modeG=SelfTest
  grd2   :   modeC=SelfTest
THEN
  act1   :   modeG≔SelfTest
  act2   :   modeGP≔modeG
END

selfTestPassed   ≙          //   CONT4-4.1
  extended
STATUS
  ordinary
```

```
REFINES
  selfTestPassed
WHEN
  grd1   :   modeG=SelfTest
  grd2   :   modeC=SelfTest
THEN
  act1   :   modeG≔Menu
  act2   :   modeGP≔modeG
  act3   :   modeC≔VentilationOff
  act4   :   modeCP≔VentilationOff
END

setParam   ≙
  extended
STATUS
  ordinary
REFINES
  setParam
WHEN
  grd1   :   modeG∈{Menu}∪Ventilation
THEN
  act1   :   modeG≔Settings
  act2   :   modeGP≔modeG
END

settingParams   ≙
  extended
STATUS
  ordinary
REFINES
  settingParams
ANY
  psvP
  pcvP
  comP
WHERE
  grd1   :   modeG=Settings
  grd2   :   max(dom(psvParamsValG))<curTime
  grd3   :   psvP∈ psvParams⇸N1
  grd4   :   psvParams\{RRAP,PinspAP}⊆dom(psvP)
  grd5   :   pcvP∈ pcvParams⟶N1
  grd6   :   comP∈ comParams⟶N1
THEN
  act1   :   psvParamsValG(curTime)≔psvP
  act2   :   pcvParamsValG(curTime)≔pcvP
  act3   :   comParamsValG(curTime)≔comP
END

saveBackAbort   ≙
  extended
STATUS
  ordinary
REFINES
  saveBackAbort
ANY
  modeg
  modec
  sv
  pcvC
  psvC
  comC
  pcvG
  psvG
  comG
WHERE
  grd1   :   modeG=Settings ∧ modeg∈ModesG
  grd2   :   modeg∈Ventilation∪{Menu}
  grd3   :   modeC≠FailSafe
  grd4   :   modec∈ModesC
  grd5   :   modeC∈Ventilation ⇒ modec∈Ventilation ∧ modeg=modec
  grd6   :   modeC∉Ventilation ⇒modec=modeC ∧ modeg=Menu
```

```
  grd7   :   sv∈
             {TRUE,FALSE}
             psvG={TRUE↦
  grd8   :   psvParamsValG(max(dom(psvParamsValG))),
              FALSE↦psvParamsValC(max(dom(psvParamsValC)))}(sv)
             psvC={TRUE↦
  grd9   :   psvParamsValG(max(dom(psvParamsValG))),
              FALSE↦psvParamsValC(max(dom(psvParamsValC)))}(sv)
              pcvG={TRUE↦
  grd10  :   pcvParamsValG(max(dom(pcvParamsValG))),
               FALSE↦pcvParamsValC(max(dom(pcvParamsValC)))}(sv)
             pcvC={TRUE↦
  grd11  :   pcvParamsValG(max(dom(pcvParamsValG))),
              FALSE↦pcvParamsValC(max(dom(pcvParamsValC)))}(sv)
             comG={TRUE↦
  grd12  :   comParamsValG(max(dom(comParamsValG))),
              FALSE↦comParamsValC(max(dom(comParamsValC)))}(sv)
             comC={TRUE↦
  grd13  :   comParamsValG(max(dom(comParamsValG))),
              FALSE↦comParamsValC(max(dom(comParamsValC)))}(sv)
  grd14  :   PCV2PSV=TRUE ∧ modeC=PCV⇒modec=PSV
  grd15  :   PCV2PSV=FALSE ∨ modeC≠PCV⇒modec=modeC
             modec=PSV
             ⇒
  grd16  :   dom(psvC)=psvParams ∧
             psvC(ApneaLag)≥max_insp_time_psv÷2
  grd17  :   modeC≠modec ⇒PCV2PSV=TRUE ∧ modeC=PCV ∧ modec=PSV
             modec=PCV∧ sv=TRUE⇒
  grd18  :   10*60*pcvC(IEPCV)÷ (pcvC(RRPCV)*(1 +pcvC(IEPCV))) >0
  grd19  :   curTime∉dom(pcvParamsValC)
  grd20  :   ∀c·c∈cycles⇒ inspBegT(c)≠curTime
THEN
  act1   :   modeG≔modeg
  act2   :   modeGP≔modeG
  act3   :   modeC≔modec
  act4   :   modeCP≔modeC
  act5   :   psvParamsValG(curTime)≔psvG
  act6   :   psvParamsValC(curTime)≔psvC
  act7   :   pcvParamsValG(curTime)≔pcvG
  act8   :   pcvParamsValC(curTime)≔pcvC
  act9   :   comParamsValG(curTime)≔comG
  act10  :   comParamsValC(curTime)≔comC
  act11  :   PCV2PSV≔FALSE
END

startStopPCVPSV   ≙
  extended
STATUS
  ordinary
REFINES
  startStopPCVPSV
ANY
  modeg
  modec
WHERE
  grd1   :   modeG∈{Menu}∪Ventilation
  grd2   :   modeg∈Ventilation∪{Menu}
  grd3   :   modeG=Menu⇒modeg∈Ventilation
  grd4   :   modeG∈Ventilation⇒modeg=Menu
  grd5   :   modeC≠FailSafe
  grd6   :   modeg∈Ventilation ⇒ modec=modeg
  grd7   :   modeg∉Ventilation ⇒ modec=VentilationOff
             modeg=PSV
  grd8   :   ⇒
             dom(psvParamsValC(max(dom(psvParamsValC))))=psvParams
             modeg=PSV
  grd9   :   ⇒
             (psvParamsValC(max(dom(psvParamsValC))))(ApneaLag)≥max_insp_time_psv÷2
  grd10  :   modec=PCV⇒
              10*60*(pcvParamsValC(max(dom(pcvParamsValC))))(IEPCV)÷
                ((pcvParamsValC(max(dom(pcvParamsValC))))(RRPCV)*
```

```
            (1 +(pcvParamsValC(max(dom(pcvParamsValC))))(IEPCV))) >0
  grd11   :   cycles=ventilPhase~[{expEnd,expPauseEnd}]
THEN
  act1    :   modeG≔modeg
  act2    :   modeGP≔modeG
  act3    :   modeC≔modec
  act4    :   modeCP≔{TRUE↦modeC,FALSE↦modeCP}(bool(modeC∈Ventilation ∪{VentilationOff}))
              inValve≔{TRUE↦
  act18   :                {FALSE↦FALSE, TRUE↦inValve}(inValveF),
                       FALSE↦inValve}(bool(modec∉Ventilation))
              outValve≔{TRUE↦
  act19   :                 {FALSE↦TRUE, TRUE↦outValve}(outValveF),
                        FALSE↦outValve}(bool(modec∉Ventilation))
END

moveToPSV   ≙
  extended
STATUS
  ordinary
REFINES
  moveToPSV
ANY
  modeg
  modec
WHERE
  grd1    :   modeG∈ Ventilation
  grd2    :   modeg∈Ventilation∪{Settings}
  grd3    :   modeC∈Ventilation\{FailSafe}
  grd4    :   modec∈Ventilation
  grd5    :   modeG=PCV
  grd6    :   modeC=PCV
  grd7    :   modeg=Settings
  grd8    :   modec=modeC
THEN
  act1    :   modeG≔modeg
  act2    :   modeGP≔modeG
  act3    :   modeC≔modec
  act4    :   modeCP≔modeC
  act5    :   PCV2PSV≔TRUE
END

failExternalPower   ≙
  extended
STATUS
  ordinary
REFINES
  failExternalPower
ANY
  modeg
  modegp
  modec
  modecp
  pcv2psv
  cycs
WHERE
  grd1    :   onAC=TRUE
  grd2    :   modeg ∈ ModesG ∧ modegp ∈ ModesG
  grd3    :   ¬(power=TRUE ∧ crashed=FALSE) ⇒ modeg=Off ∧ modegp=Off
              power=TRUE ∧ crashed=FALSE ⇒
              (
  grd4    :      ((switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE ⇒ modeg=modeG ∧ modegp=modeGP) ∧
              (¬(switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)⇒ modeg=Off ∧ modegp=modeG))
              )
  grd5    :   modec ∈ ModesC ∧ modecp ∈ ModesC
  grd6    :   power= FALSE ⇒ modec=modeC ∧ modecp=modeCP
               power= TRUE ⇒
               (
  grd7    :      ((switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE ⇒modec=modeC ∧ modecp=modeCP) ∧
                (¬(switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)⇒modec=Off ∧ modecp=modeC))
               )
  grd8    :   ¬(power=TRUE ∧ crashed=FALSE) ⇒ pcv2psv=FALSE
```

```
  grd9   :   power=TRUE ∧ crashed=FALSE ⇒
             (
             ((switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE⇒ pcv2psv=PCV2PSV) ∧
             (¬(switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)⇒ pcv2psv=FALSE))
                     )
              (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)
  grd10  :     ⇒
              cycs=cycles
              (switchover=FALSE ∨ batLev=0 ∨  batFail=TRUE)
  grd11  :   ⇒
              cycs=ventilPhase~[{expEnd,expPauseEnd}]
  grd12  :   cycs⊆cycles
THEN
  act1   :   onAC≔FALSE
  act2   :   modeG≔modeg
  act3   :   modeGP≔modegp
  act4   :   modeC≔modec
  act5   :   modeCP≔modecp
  act6   :   PCV2PSV≔pcv2psv
  act7   :   ventilPhase≔cycs◁ventilPhase
  act8   :   inspBegT≔cycs◁inspBegT
  act9   :   inspEndT≔cycs◁inspEndT
  act10   :   cycles≔cycs
  act11   :   cycleMode≔ cycs◁cycleMode
  act12   :   inspPauseBegT≔cycs◁inspPauseBegT
  act13   :   inspPauseEndT≔cycs◁inspPauseEndT
  act14   :   rmBegT≔cycs◁rmBegT
  act15   :   rmEndT≔cycs◁rmEndT
  act16   :   expBegT≔cycs◁expBegT
  act17   :   expEndT≔cycs◁expEndT
  act18   :   expPauseBegT≔cycs◁expPauseBegT
  act19   :   expPauseEndT≔cycs◁expPauseEndT
              inValve≔{TRUE↦
  act20   :                {FALSE↦FALSE, TRUE↦inValve}(inValveF),
                     FALSE↦inValve}(bool(modec=Off))
              outValve≔{TRUE↦
  act21   :                  {FALSE↦TRUE, TRUE↦outValve}(outValveF),
                      FALSE↦outValve}(bool(modec=Off))
END

failSelfTest   ≙
  extended
STATUS
  ordinary
REFINES
  failSelfTest
WHEN
  grd1   :   modeC=SelfTest
THEN
  act1   :   modeC≔FailSafe
  act2   :   modeCP≔modeC
END

progress   ≙
  extended
STATUS
  ordinary
REFINES
  progress
ANY
  step
  l
  modeg
  batf
  modec
  paw
  cycs
WHERE
  grd1   :   step∈N1 ∧ l∈batteryRange ∧ batf∈BOOL
  grd2   :   batFail=TRUE ⇒ l=batLev
  grd3   :   onAC=TRUE ∧ batFail=FALSE ⇔ l=batLev+step
```

```
  grd4   :   onAC=FALSE ∧ batFail=FALSE ⇔ l=max
             ({0,batLev−step})
             (onAC=TRUE ∨ (l>0 ∧ switchover=TRUE ∧ batf=FALSE)) ∧
             power=TRUE ∧  modeG=Off ∧
  grd5   :   crashed=FALSE
             ⇒
             modeg=StartUp
  grd6   :   (l=0 ∨ batf=TRUE ∨ switchover=FALSE) ∧ onAC=FALSE ⇒modeg=Off
             onAC=TRUE  ∨
  grd7   :   ((l>0 ∧ batLev>0) ∧ switchover=TRUE ∧ batf=FALSE)
             ⇒
             modeg=modeG
  grd8   :   modec∈{FailSafe,modeC, Off,StartUp}
  grd9   :   modec=FailSafe⇒modeC≠Off
             (onAC=TRUE ∨ (l>0 ∧ switchover=TRUE ∧ batf=FALSE)) ∧
             power=TRUE ∧  modeC=Off
  grd10  :   ⇒
             modec=StartUp
             (batLev>0 ∧ l>0)∨  switchover=FALSE ∨ onAC=TRUE ∨
             power=FALSE
  grd11  :   ⇒
             modec∈{modeC,FailSafe}
             (l=0 ∨ batf=TRUE ∨ switchover=FALSE) ∧ onAC=FALSE
  grd12  :   ⇒
             modec=Off
  grd13  :   step∈N1 ∧ paw∈N
             ∀c· c∈cycles ∧ ventilPhase(c)=inspBeg
  grd14  :   ⇒
             curTime+step≤inspEndT(c)
             ∀c· c∈cycles ∧ ventilPhase(c)=inspPauseBeg
  grd15  :   ⇒
             curTime+step≤inspPauseEndT(c)
             ∀c· c∈cycles ∧ ventilPhase(c)=rmBeg
  grd16  :   ⇒
             curTime+step≤ rmEndT(c)
             ∀c· c∈cycles ∧ ventilPhase(c)=expBeg
  grd17  :   ⇒
             curTime+step∈expEndT(c)
             ∀c· c∈cycles ∧ ventilPhase(c)=expPauseBeg
  grd18  :   ⇒
             curTime+step ≤ expPauseEndT(c)
  grd19  :   modec∉Ventilationu{FailSafe} ⇒ cycs=ventilPhase~[{expEnd,expPauseEnd}]
  grd20  :   modec∈Ventilationu{FailSafe}  ⇒ cycs=cycles
  grd21  :   cycs⊆cycles
THEN
  act1   :   curTime≔curTime+step
  act2   :   batLev≔l
  act3   :   modeG≔modeg
  act4   :   modeGP≔{TRUE↦modeGP, FALSE↦modeG}(bool(modeg=modeG))
  act5   :   batFail≔batf
  act6   :   modeC≔modec
  act7   :   modeCP≔{TRUE↦modeCP,FALSE↦modeC}(bool(modeC=modec))
  act8   :   PCV2PSV≔{TRUE↦PCV2PSV, FALSE↦FALSE}(bool(
             (batLev>0 ∧ l>0∧  switchover=FALSE) ∨ onAC=TRUE))
  act9   :   ventilPhase≔cycs◁ventilPhase
  act10  :   inspBegT≔cycs◁inspBegT
  act11  :   inspEndT≔cycs◁inspEndT
  act12  :   cycles≔cycs
  act13  :   cycleMode≔cycs◁cycleMode
  act14  :   inspPauseBegT≔cycs◁inspPauseBegT
  act15  :   inspPauseEndT≔cycs◁inspPauseEndT
  act16  :   rmBegT≔cycs◁rmBegT
  act17  :   rmEndT≔cycs◁rmEndT
  act18  :   expBegT≔cycs◁expBegT
  act19  :   expEndT≔cycs◁expEndT
  act20  :   expPauseBegT≔cycs◁expPauseBegT
  act21  :   expPauseEndT≔cycs◁expPauseEndT
  act22  :   inValve≔{TRUE↦inValve,FALSE↦FALSE}(bool(modec=modeC))
  act23  :   outValve≔{TRUE↦outValve,FALSE↦TRUE}(bool(modec=modeC))
END
```

switchoverFail ≙
extended
STATUS
ordinary
REFINES
switchoverFail
ANY
modeg
modec
cycs
WHERE
grd1 : *switchover=TRUE*
grd2 : *onAC=FALSE⟹modeg=Off*
grd3 : *onAC=TRUE⟹modeg=modeG*
grd4 : *onAC=FALSE⟹modec=Off*
grd5 : *onAC=TRUE⟹modec=modeC*
grd6 : *modec∉Ventilationu{FailSafe} ⟹ cycs=ventilPhase~[{expEnd,expPauseEnd}]*
grd7 : *modec∈Ventilationu{FailSafe} ⟹ cycs=cycles*
grd8 : *cycs⊆cycles*
THEN
act1 : *switchover≔FALSE*
act2 : *modeG≔modeg*
act3 : *modeGP≔{TRUE↦modeG,FALSE↦modeGP}(bool(modeG≠modeg))*
act4 : *modeC≔modec*
act5 : *modeCP≔{TRUE↦modeC,FALSE↦modeCP}(bool(modeC≠modec))*
act6 : *PCV2PSV≔{TRUE↦PCV2PSV, FALSE↦FALSE}(bool(onAC=TRUE))*
act7 : *ventilPhase≔cycs⩤ventilPhase*
act8 : *inspBegT≔cycs⩤inspBegT*
act9 : *inspEndT≔cycs⩤inspEndT*
act10 : *cycles≔cycs*
act11 : *cycleMode≔cycs⩤cycleMode*
act12 : *inspPauseBegT≔cycs⩤inspPauseBegT*
act13 : *inspPauseEndT≔cycs⩤inspPauseEndT*
act14 : *rmBegT≔cycs⩤rmBegT*
act15 : *rmEndT≔cycs⩤rmEndT*
act16 : *expBegT≔cycs⩤expBegT*
act17 : *expEndT≔cycs⩤expEndT*
act18 : *expPauseBegT≔cycs⩤expPauseBegT*
act19 : *expPauseEndT≔cycs⩤expPauseEndT*
act20 : inValve≔{TRUE↦inValve,FALSE↦FALSE}(bool(modec=modeC))
act21 : outValve≔{TRUE↦outValve,FALSE↦TRUE}(bool(modec=modeC))
END

inspStart ≙
extended
STATUS
ordinary
REFINES
inspStart
ANY
cy
paw
ve
inspT
WHERE
grd1 : *modeC∈Ventilation ∧ inspT∈ℕ*
grd2 : *cy∈Cycles\cycles*
grd3 : *ran(ventilPhase)⊆{expEnd,expPauseEnd}*
grd4 : *modeC=PCV⟹inspT=curTime + 10∗60∗*
(pcvParamsValC(max(dom(pcvParamsValC))))(IEPCV)÷
((pcvParamsValC(max(dom(pcvParamsValC))))(RRPCV)∗
(1 +(pcvParamsValC(max(dom(pcvParamsValC))))(IEPCV)))
modeC=PSV⟹
grd5 : *inspT>curTime ∧*
inspT≤curTime + max_insp_time_psv
grd6 : *paw∈ℕ ∧ ve∈ ℕ*
grd7 : *pcvParamsValC≠∅ ∧ psvParamsValC≠∅ ∧ comParamsValC≠∅*
THEN
act1 : *ventilPhase(cy)≔inspBeg*
act2 : *cycles≔cyclesu{cy}*
act3 : *inspBegT(cy)≔curTime*

```
  act4   :   inspEndT
             (cy)≔inspT
  act5   :   cycleMode(cy)≔modeC
  act6   :   PAW(curTime)≔paw
  act7   :   VE(curTime)≔ve
  act8   :   inValve≔{FALSE↦TRUE, TRUE↦inValve}(inValveF)
  act9   :   outValve≔{FALSE↦FALSE, TRUE↦outValve}(outValveF)
END

inspEnd   ≙
  extended
STATUS
  ordinary
REFINES
  inspEnd
ANY
  cy
  ve
  peakVE
WHERE
  grd1   :   cy∈ cycles ∧ ventilPhase(cy)=inspBeg ∧ ve∈N
  grd2   :   cycleMode(cy)=PSV⟹curTime>inspBegT(cy)
  grd3   :   cycleMode(cy)=PCV ⟹curTime=inspEndT(cy)
  grd4   :   peakVE=VE(max(dom(VE)))
  grd5   :   (curTime=inspEndT(cy)) ∨
             (cycleMode(cy)=PSV ⟹
             peakVE>ve ∧
             ve<(psvParamsValC(max(dom(psvParamsValC))))(ETS)∗peakVE)
  grd6   :   modeC∈Ventilation
THEN
  act1   :   ventilPhase(cy)≔inspEnd
  act2   :   VE(curTime)≔ve
  act3   :   inspEndT(cy)≔curTime
END

inspPauseStart   ≙
  extended
STATUS
  ordinary
REFINES
  inspPauseStart
ANY
  cy
WHERE
  grd1   :   cy∈cycles ∧ ventilPhase(cy)=inspEnd
  grd2   :   modeC∈Ventilation
THEN
  act1   :   ventilPhase(cy)≔inspPauseBeg
  act2   :   inspPauseBegT(cy)≔curTime
  act3   :   inspPauseEndT(cy)≔curTime+inspPauseMax
  act4   :   inValve≔{FALSE↦FALSE, TRUE↦inValve}(inValveF)
END

inspPauseEnd   ≙
  extended
STATUS
  ordinary
REFINES
  inspPauseEnd
ANY
  cy
WHERE
  grd1   :   cy∈cycles ∧ ventilPhase(cy)=inspPauseBeg
  grd2   :   curTime≤inspPauseEndT(cy)
  grd3   :   curTime>inspPauseBegT(cy)
  grd4   :   modeC∈Ventilation
THEN
  act1   :   ventilPhase(cy)≔inspPauseEnd
  act2   :   inspPauseEndT(cy)≔curTime
END

rmStart   ≙
```

```
  extended
STATUS
  ordinary
REFINES
  rmStart
ANY
  cy
  t
WHERE
  grd1   :   cy∈cycles
  grd2   :   ventilPhase(cy)∈{inspEnd, inspPauseEnd}
  grd3   :   t=max({x|x∈ dom(comParamsValC) ∧ x≤  inspBegT(cy)})
  grd4   :   modeC∈Ventilation
THEN
  act1   :   ventilPhase(cy):=rmBeg
  act2   :   rmBegT(cy):=curTime
  act3   :   rmEndT(cy):=curTime+(comParamsValC(t))(timerRM)
  act4   :   inValve:={FALSE↦TRUE, TRUE↦inValve}(inValveF)
END

rmEnd   ≙
  extended
STATUS
  ordinary
REFINES
  rmEnd
ANY
  cy
WHERE
  grd1   :   cy∈cycles
  grd2   :   ventilPhase(cy)=rmBeg
  grd3   :   curTime=rmEndT(cy)
  grd4   :   modeC∈Ventilation
THEN
  act1   :   ventilPhase(cy):=rmEnd
END

expStart   ≙
  extended
STATUS
  ordinary
REFINES
  expStart
ANY
  cy
  t
  expT
WHERE
  grd1   :   cy∈cycles ∧  ventilPhase(cy)∈{inspEnd, inspPauseEnd,rmEnd}
  grd2   :   t=max({x|x∈ dom(pcvParamsValC) ∧ x≤  inspBegT(cy)})
  grd3   :   expT⊆N
             cycleMode(cy)=PCV ⟹
  grd4   :   expT={curTime +60÷((pcvParamsValC(t))(RRPCV) *
                        (1 +(pcvParamsValC(t))(IEPCV)))}
             cycleMode(cy)=PSV ⟹
             expT=
  grd5   :   (curTime+(inspEndT(cy)−inspBegT(cy))÷2)
             ‥
             (curTime +((psvParamsValC(t)))(ApneaLag))
  grd6   :   modeC∈Ventilation
THEN
  act1   :   ventilPhase(cy):=expBeg
  act2   :   expBegT(cy):=curTime
  act3   :   expEndT(cy):=expT
  act4   :   inValve:={FALSE↦FALSE, TRUE↦inValve}(inValveF)
  act5   :   outValve:={FALSE↦TRUE, TRUE↦outValve}(outValveF)
END

expEnd   ≙
  extended
STATUS
```

```
  ordinary
REFINES
  expEnd
ANY
  modeg
  modec
  pcv
  paw
  cy
  lastTime
  t
WHERE
  grd1   :   modeG∈ Ventilation
  grd2   :   modeg∈Ventilation∪{Settings}
  grd3   :   modeC∈Ventilation\{FailSafe}
  grd4   :   modec∈Ventilation
  grd5   :   pcv ∈ pcvParams⟶N1
  grd6   :   modeC=PCV ⇒ modec=PCV
             modeC=PSV ∧ modec=PCV
             ⇒
  grd7   :   pcv=pcvParamsValC(max(dom(pcvParamsValC)))⩤
             {RRPCV↦(psvParamsValC(max(dom(psvParamsValC))))(RRAP),
              PinspPCV↦(psvParamsValC(max(dom(psvParamsValC))))(PinspAP),
             IEPCV↦IEAP}
             modeC=PCV ∨ modec=PSV
  grd8   :   ⇒
             pcv=pcvParamsValC(max(dom(pcvParamsValC)))
             modec=PCV
  grd9   :   ⇒
             10*60*pcv(IEPCV)÷ (pcv(RRPCV)*(1 +pcv(IEPCV))) >0
  grd11   :   modeG≠Off⇒ modeg=modec
  grd12   :   cy∈cycles
  grd13   :   ventilPhase(cy)=expBeg
  grd14   :   lastTime= max(dom(PAW))
  grd15   :   paw∈N
  grd16   :   t=max({x | x∈dom(psvParamsValC)∧x≤inspBegT(cy)})
              (
              cycleMode(cy)=PCV ∧
                  (curTime∈ expEndT(cy) ∨
                    (curTime<max(expEndT(cy)) ∧
                      (PAW(lastTime)<paw ∧ PAW(lastTime)−paw>(pcvParamsValC(t))(ITSPCV)))
                  )
              )
  grd17   :   ∨
              (
              cycleMode(cy)=PSV ∧
                  (curTime= max(expEndT(cy)) ∨
                    (curTime≥min(expEndT(cy)) ∧ curTime<max(expEndT(cy)) ∧ PAW(lastTime)<paw ∧ PAW
             (lastTime)−paw>(psvParamsValC(t))(ITSPSV)))
              )
              curTime=max(expEndT(cy))∧ cycleMode(cy)=PSV
  grd18   :   ⇒
              modec=PCV
              curTime≠max(expEndT(cy))∧ cycleMode(cy)=PSV
  grd19   :   ⇒
              modec=PSV
              cycleMode(cy)=PCV
  grd20   :   ⇒
              modec=PCV
  grd21   :   modeC∈Ventilation
THEN
  act1   :   modeG≔modeg
  act2   :   modeGP≔modeG
  act3   :   modeC≔modec
  act4   :   modeCP≔modeC
  act5   :   pcvParamsValC(curTime)≔pcv
  act6   :   psvParamsValC(curTime)≔psvParamsValC(max(dom(psvParamsValC)))
  act7   :   comParamsValC(curTime)≔comParamsValC(max(dom(comParamsValC)))
  act8   :   ventilPhase(cy)≔expEnd
  act9   :   PAW(curTime)≔paw
  act10   :   expEndT(cy)≔curTime‥curTime
END
```

```
  expPauseStart   ≙
    extended
  STATUS
    ordinary
  REFINES
    expPauseStart
  ANY
    cy
  WHERE
    grd1   :   cy∈cycles ∧ ventilPhase(cy)=expEnd
    grd2   :   modeC∈Ventilation
    grd3   :   ran(ventilPhase)⊆{expEnd,expPauseEnd}
  THEN
    act1   :   ventilPhase(cy):=expPauseBeg
    act2   :   expPauseBegT(cy):=curTime
    act3   :   expPauseEndT(cy):=curTime+expPauseMax
    act4   :   inValve:={FALSE↦FALSE, TRUE↦inValve}(inValveF)
    act5   :   outValve:={FALSE↦FALSE, TRUE↦outValve}(outValveF)
  END

  expPauseEnd   ≙
    extended
  STATUS
    ordinary
  REFINES
    expPauseEnd
  ANY
    cy
  WHERE
    grd1   :   cy∈cycles ∧ ventilPhase(cy)=expPauseBeg
    grd2   :   curTime≤expPauseEndT(cy)
    grd3   :   curTime>expPauseBegT(cy)
    grd4   :   modeC∈Ventilation
  THEN
    act1   :   ventilPhase(cy):=expPauseEnd
    act2   :   expPauseEndT(cy):=curTime
  END

  inValveF   ≙
  STATUS
    ordinary
  WHEN
    grd1   :   inValveF=FALSE
  THEN
    act1   :   inValveF:=TRUE
  END

  outValveF   ≙
  STATUS
    ordinary
  WHEN
    grd1   :   outValveF=FALSE
  THEN
    act1   :   outValveF:=TRUE
  END

END
```

```
CONTEXT
   Alarms
SETS
   Alarms
CONSTANTS
   patConnected
   FI1Failure
   FI2Failure
   oxygenSensorFailure
   switchoverFailure
   conguiComFailure
   inValveFailure
   outsideValue
   outValveFailure
AXIOMS
              partition(Alarms,{conguiComFailure}, {inValveFailure},{patConnected},
   axm1   :           {FI1Failure}, {FI2Failure}, {oxygenSensorFailure},{switchoverFailure},
              {outsideValue},{outValveFailure})
END
```

MACHINE
MVLAlarms
REFINES
Valves
SEES
ContStates
VentilStates
PSVParams
Alarms
VARIABLES
power
onAC
batLev
batFail
switchover
crashed
modeG
modeGP
modeC
modeCP
PCV2PSV
comParamsValC
comParamsValG
pcvParamsValC
pcvParamsValG
psvParamsValC
psvParamsValG
curTime
offT
cycles
cycleMode
ventilPhase
inspEndT
inspBegT
inspPauseBegT
inspPauseEndT
PAW
VE
rmBegT
rmEndT
expBegT
expEndT
expPauseBegT
expPauseEndT
inValve
outValve
inValveF
outValveF
alarmRaised
patientConnected
FI1
FI2
oxygenSensor
guiContCom
inValveP
INVARIANTS
inv1 : alarmRaised ∈ Alarms⟶BOOL ∧ inValveP∈BOOL
inv2 : power=FALSE ⟹ alarmRaised=Alarms×{FALSE}
alarmRaised(patConnected)
inv3 : =
bool(patientConnected=TRUE ∧ modeC∈{StartUp,SelfTest})
inv4 : alarmRaised(inValveFailure)=
bool(∃c· (c∈cycles ∧
(
(
(
(ventilPhase(c)=inspBeg ∧ curTime >inspBegT(c))∨
(ventilPhase(c)=rmBeg ∧ curTime > rmBegT(c))
) ∧ inValveP=FALSE)

```
                    ∨
                    ((
                    (ventilPhase(c)=expPauseBeg ∧ curTime >expPauseBegT(c))
                    ∨
                   (ventilPhase(c)=expBeg ∧ curTime >expBegT(c))
                    ∨
                    (ventilPhase(c)=inspPauseBeg ∧ curTime >inspPauseBegT(c))
                  )
                  ∧ inValveP=TRUE))
               )
            )
            alarmRaised(inValveFailure)=TRUE
inv5   :    ⇒
            modeC=FailSafe
inv6   :    oxygenSensor∈ BOOL ∧ FI1∈ BOOL ∧ FI2∈ BOOL
inv7   :    oxygenSensor=FALSE ∧ modeC∉{Off, StartUp,SelfTest}⇔alarmRaised(oxygenSensorFailure)=TRUE
inv8   :    switchover=FALSE  ∧ modeC∉{Off, StartUp,SelfTest} ⇔alarmRaised(switchoverFailure)=TRUE
inv9   :    FI1=FALSE ∧  modeC∉{Off, StartUp,SelfTest} ⇔alarmRaised(FI1Failure)=TRUE
inv10   :    FI2=FALSE ∧ modeC∉{Off, StartUp, SelfTest}  ⇔alarmRaised(FI2Failure)=TRUE
             (
                 modeC∉{Off,StartUp, FailSafe} ∨
inv14   :       (modeCP≠StartUp ∧ modeC=FailSafe)
             )
             ⇒comParamsValC≠ø
             (
                 modeC∉{Off,StartUp, FailSafe} ∨
                (modeCP≠StartUp ∧ modeC=FailSafe)
             )
             ⇒
             alarmRaised(outsideValue)=bool(
             (∃ p· p ∈ comParams  ∧
inv11   :     (comParamsValC(max(dom(comParamsValC))))(p)∉domcomParams(p))
             ∨
             (∃ p· p ∈ pcvParams  ∧
              (pcvParamsValC(max(dom(pcvParamsValC))))(p)∉dompcvParams(p))
             ∨
             (∃ p· p ∈ dom((psvParamsValC(max(dom(psvParamsValC)))))  ∧
              (psvParamsValC(max(dom(psvParamsValC))))(p)∉dompsvParams(p))
             )
inv12   :    guiContCom∈BOOL
             guiContCom=FALSE ∧ modeC∉{Off, StartUp, SelfTest}
inv13   :     ⇔
             alarmRaised(conguiComFailure)=TRUE
EVENTS
  INITIALISATION   ≙
    extended
  STATUS
    ordinary
  BEGIN
    act1   :   power≔FALSE
    act2   :   batLev:∈batteryRange
    act3   :   onAC≔TRUE
    act4   :   switchover≔TRUE
    act5   :   crashed≔FALSE
    act6   :   modeG≔Off
    act7   :   modeGP≔Off
    act8   :   curTime≔0
    act9   :   batFail:∈BOOL
    act10   :   modeC≔Off
    act11   :   modeCP≔Off
    act12   :   comParamsValC≔ø
    act13   :   comParamsValG≔ø
    act14   :   pcvParamsValC≔ø
    act15   :   pcvParamsValG≔ø
    act16   :   psvParamsValG≔ø
    act17   :   psvParamsValC≔ø
    act18   :   offT≔0
    act19   :   PCV2PSV≔FALSE
    act20   :   ventilPhase≔ø
    act21   :   inspBegT≔ø
    act22   :   inspEndT≔ø
    act23   :   cycles≔ø
```

```
  act24   :   cycleMode≔ø
  act25   :   PAW≔ø
  act26   :   VE≔ø
  act27   :   inspPauseBegT≔ø
  act28   :   inspPauseEndT≔ø
  act29   :   rmBegT≔ø
  act30   :   rmEndT≔ø
  act31   :   expBegT≔ø
  act32   :   expEndT≔ø
  act33   :   expPauseBegT≔ø
  act34   :   expPauseEndT≔ø
  act35   :   inValve≔FALSE
  act36   :   outValve≔TRUE
  act37   :   inValveF≔FALSE
  act38   :   outValveF≔FALSE
  act39   :   alarmRaised≔Alarms×{FALSE}
  act40   :   patientConnected≔FALSE
  act41   :   oxygenSensor≔TRUE
  act42   :   FI1≔TRUE
  act43   :   FI2≔TRUE
  act44   :   guiContCom:∈BOOL
  act45   :   inValveP≔FALSE
END

powerON   ≙          //   CONT.2
  extended
STATUS
  ordinary
REFINES
  powerON
WHEN
  grd1   :   power=FALSE
  grd2   :   onAC=TRUE ∨ (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)
THEN
  act1   :   power≔TRUE
  act2   :   modeG≔{FALSE↦StartUp, TRUE↦Off}(crashed)
  act3   :   modeGP≔modeG
  act4   :   modeC≔StartUp
  act5   :   modeCP≔modeC
             alarmRaised≔alarmRaised⩤{
  act6   :   patConnected↦patientConnected
             }
END

powerOff   ≙
  extended
STATUS
  ordinary
REFINES
  powerOff
ANY
  val
WHERE
  grd1   :   power=TRUE
             comParamsValC=ø ∨  comParamsValC(max(dom(comParamsValC)))=defcomParams
  grd2   :   ⟹
             val=0
             comParamsValC≠ø ∧
  grd3   :   comParamsValC(max(dom(comParamsValC)))≠defcomParams
             ⟹
             val=curTime
THEN
  act1   :   power≔FALSE
  act2   :   modeG≔Off
  act3   :   modeGP≔modeG
  act4   :   modeC≔Off
  act5   :   modeCP≔modeC
  act6   :   offT≔val
  act7   :   PCV2PSV≔FALSE
  act8   :   ventilPhase≔ventilPhase⩥{expEnd,expPauseEnd}
  act9   :   inspBegT≔ventilPhase~[{expEnd,expPauseEnd}]⩤inspBegT
```

```
  act10   :   inspEndT≔ventilPhase~[{expEnd,expPauseEnd}]
              ⩤inspEndT
  act11   :   cycles≔ventilPhase~[{expEnd,expPauseEnd}]
  act12   :   cycleMode≔ ventilPhase~[{expEnd,expPauseEnd}]⩤cycleMode
  act13   :   inspPauseBegT≔ventilPhase~[{expEnd,expPauseEnd}]⩤inspPauseBegT
  act14   :   inspPauseEndT≔ventilPhase~[{expEnd,expPauseEnd}]⩤inspPauseEndT
  act15   :   rmBegT≔ventilPhase~[{expEnd,expPauseEnd}]⩤rmBegT
  act16   :   rmEndT≔ventilPhase~[{expEnd,expPauseEnd}]⩤rmEndT
  act17   :   expBegT≔ventilPhase~[{expEnd,expPauseEnd}]⩤expBegT
  act18   :   expEndT≔ventilPhase~[{expEnd,expPauseEnd}]⩤expEndT
  act19   :   expPauseBegT≔ventilPhase~[{expEnd,expPauseEnd}]⩤expPauseBegT
  act20   :   expPauseEndT≔ventilPhase~[{expEnd,expPauseEnd}]⩤expPauseEndT
  act21   :   inValve≔FALSE
  act22   :   outValve≔TRUE
  act23   :   alarmRaised≔Alarms×{FALSE}
END

startUpEndedGui   ≙
  extended
STATUS
  ordinary
REFINES
  startUpEndedGui
ANY
  modeg
  pcvG
  comG
  psvG
WHERE
  grd1   :   modeG=StartUp
  grd2   :   modeg∈{Start}∪Ventilation
  grd3   :   modeC∈Ventilation ⇒ modeg=modeC
  grd4   :   modeC∉Ventilation ⇒ modeg=Start
             modeC∈Ventilation ∨ (offT≠0 ∧ curTime−offT ≤ resumeTime)
             ⇒
                comG=comParamsValG ⩤
                                {curTime ↦comParamsValC(max(dom(comParamsValC)))} ∧
  grd5   :      pcvG=pcvParamsValG ⩤
                                {curTime ↦pcvParamsValC(max(dom(pcvParamsValC)))} ∧
                psvG=psvParamsValG ⩤
                                {curTime ↦psvParamsValC(max(dom(psvParamsValC)))}
             modeC∉Ventilation ∧ (offT=0 ∨  curTime−offT > resumeTime)
             ⇒
                comG=comParamsValG ⩤
                                {curTime ↦defcomParams} ∧
  grd6   :      pcvG=pcvParamsValG ⩤
                                {curTime ↦defpcvParams} ∧
                psvG=psvParamsValG ⩤
                                {curTime ↦defpsvParams}
  grd7   :   guiContCom=TRUE
THEN
  act1   :   modeG≔modeg
  act2   :   modeGP≔modeG
  act3   :   comParamsValG≔comG
  act4   :   pcvParamsValG≔pcvG
  act5   :   psvParamsValG≔psvG
END

startUpEndedCont   ≙
  extended
STATUS
  ordinary
REFINES
  startUpEndedCont
ANY
  comC
  pcvC
  psvC
  outs
WHERE
  grd1   :   modeC=StartUp
  grd2   :   offT=0 ∨  curTime−offT > resumeTime
```

```
                ⇒
                comC=comParamsValC ⩤
                                   {curTime ↦defcomParams} ∧
                pcvC=pcvParamsValC ⩤
                                   {curTime ↦defpcvParams} ∧
                psvC=psvParamsValC ⩤
                                   {curTime ↦defpsvParams}
                offT≠0 ∧ curTime−offT ≤ resumeTime
                ⇒
                comC=comParamsValC ⩤
                                   {curTime ↦comParamsValC(max(dom(comParamsValC)))} ∧
  grd3   :      pcvC=pcvParamsValC ⩤
                                   {curTime ↦pcvParamsValC(max(dom(pcvParamsValC)))} ∧
                psvC=psvParamsValC ⩤
                                   {curTime ↦psvParamsValC(max(dom(psvParamsValC)))}
  grd4   :      oxygenSensor=TRUE ∧ FI1=TRUE ∧ FI2=TRUE ∧ guiContCom=TRUE
                ∧ inValve=FALSE ∧ outValve=TRUE
                offT≠0 ∧ curTime−offT ≤ resumeTime
                ⇒
                outs=bool(
                (
                (∃ p·p ∈ dom(comParamsValC(max(dom(comParamsValC))))
                  ∧ (comParamsValC(max(dom(comParamsValC))))(p)∉domcomParams(p))
  grd5   :      ∨
                (∃ p·p ∈ dom(pcvParamsValC(max(dom(pcvParamsValC))))
                  ∧ (pcvParamsValC(max(dom(pcvParamsValC))))(p)∉dompcvParams(p))
                ∨
                 (∃ p·p ∈ dom(psvParamsValC(max(dom(psvParamsValC))))
                              ∧ (psvParamsValC(max(dom(psvParamsValC))))(p)∉dompsvParams(p))
                )
                )
                offT=0 ∨  curTime−offT > resumeTime
  grd6   :      ⇒
                outs=FALSE
THEN
  act1   :      modeC≔SelfTest
  act2   :      modeCP≔modeC
  act3   :      comParamsValC≔comC
  act4   :      pcvParamsValC≔pcvC
  act5   :      psvParamsValC≔psvC
  act6   :      alarmRaised(outsideValue)≔outs
END

crash   ≙
  extended
STATUS
  ordinary
REFINES
  crash
WHEN
  grd1   :      crashed=FALSE
THEN
  act1   :      crashed≔TRUE
  act2   :      modeG≔Off
  act3   :      modeGP≔Off
  act4   :      PCV2PSV≔FALSE
END

repare   ≙
  extended
STATUS
  ordinary
REFINES
  repare
ANY
  modeg
WHERE
  grd1   :      crashed=TRUE
                power=FALSE ∨  (onAC=FALSE ∧ (switchover=FALSE ∨ batLev=0 ∨ batFail=TRUE))
  grd2   :      ⇒
                modeg=Off
  grd3   :      power=TRUE ∧ (onAC=TRUE ∨ (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE))
                ⇒
```

```
            modeg=StartUp
  grd4   :  modeC∈Ventilation ⟹ modeg=modeC
            power=TRUE ∧ (onAC=TRUE ∨ (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE))∧
  grd5   :  modeC∉Ventilation
            ⟹
            modeg=StartUp
THEN
  act1   :  crashed≔FALSE
  act2   :  modeG≔modeg
  act3   :  modeGP≔modeG
END

newPatient   ≙
  extended
STATUS
  ordinary
REFINES
  newPatient
WHEN
  grd1   :  modeG=Start
  grd2   :  modeC=SelfTest
THEN
  act1   :  modeG≔SelfTest
  act2   :  modeGP≔modeG
END

resumeVent   ≙
  extended
STATUS
  ordinary
REFINES
  resumeVent
WHEN
  grd1   :  modeG=Start
  grd2   :  modeC=SelfTest
  grd3   :  offT≠0 ∧ curTime−offT ≤ resumeTime
THEN
  act1   :  modeG≔Menu
  act2   :  modeGP≔modeG
  act3   :  modeC≔VentilationOff
  act4   :  modeCP≔modeC
            alarmRaised≔alarmRaised⩤(
              {patConnected↦FALSE}∪
              {oxygenSensorFailure↦bool(oxygenSensor=FALSE),
  act5   :     switchoverFailure↦bool(switchover=FALSE),
               FI1Failure↦bool(FI1=FALSE),
               FI2Failure↦bool(FI2=FALSE),
               conguiComFailure↦bool(guiContCom=FALSE)})
END

runAbortSelfTest   ≙
  extended
STATUS
  ordinary
REFINES
  runAbortSelfTest
WHEN
  grd1   :  modeG=SelfTest
  grd2   :  modeC=SelfTest
  grd3   :  ¬(switchover=TRUE ∧ outValve=TRUE ∧ FI1=TRUE ∧ FI2=TRUE ∧
            oxygenSensor= TRUE)
THEN
  act1   :  modeG≔SelfTest
  act2   :  modeGP≔modeG
END

selfTestPassed   ≙          //   CONT4-4.1
  extended
STATUS
  ordinary
REFINES
  selfTestPassed
```

```
WHEN
  grd1   :   modeG=SelfTest
  grd2   :   modeC=SelfTest
  grd3   :   switchover=TRUE ∧ outValve=TRUE ∧ FI1=TRUE ∧ FI2=TRUE ∧
             oxygenSensor= TRUE ∧ guiContCom=TRUE
THEN
  act1   :   modeG≔Menu
  act2   :   modeGP≔modeG
  act3   :   modeC≔VentilationOff
  act4   :   modeCP≔VentilationOff
  act5   :   alarmRaised(patConnected)≔FALSE
END

setParam   ≙
  extended
STATUS
  ordinary
REFINES
  setParam
WHEN
  grd1   :   modeG∈{Menu}∪Ventilation
THEN
  act1   :   modeG≔Settings
  act2   :   modeGP≔modeG
END

settingParams   ≙
  extended
STATUS
  ordinary
REFINES
  settingParams
ANY
  psvP
  pcvP
  comP
WHERE
  grd1   :   modeG=Settings
  grd2   :   max(dom(psvParamsValG))<curTime
  grd3   :   psvP∈ psvParams⇸N1
  grd4   :   psvParams\{RRAP,PinspAP}⊆dom(psvP)
  grd5   :   pcvP∈ pcvParams⟶N1
  grd6   :   comP∈ comParams⟶N1
THEN
  act1   :   psvParamsValG(curTime)≔psvP
  act2   :   pcvParamsValG(curTime)≔pcvP
  act3   :   comParamsValG(curTime)≔comP
END

saveBackAbort   ≙
  extended
STATUS
  ordinary
REFINES
  saveBackAbort
ANY
  modeg
  modec
  sv
  pcvC
  psvC
  comC
  pcvG
  psvG
  comG
  outs
WHERE
  grd1   :   modeG=Settings ∧ modeg∈ModesG
  grd2   :   modeg∈Ventilation∪{Menu}
  grd3   :   modeC≠FailSafe
  grd4   :   modec∈ModesC
```

```
  grd5   :   modeC∈Ventilation ⇒ modec∈Ventilation ∧ modeg=modec
  grd6   :   modeC∉Ventilation ⇒modec=modeC ∧ modeg=Menu
  grd7   :   sv∈{TRUE,FALSE}
             psvG={TRUE↦
  grd8   :   psvParamsValG(max(dom(psvParamsValG))),
              FALSE↦psvParamsValC(max(dom(psvParamsValC)))}(sv)
             psvC={TRUE↦
  grd9   :   psvParamsValG(max(dom(psvParamsValG))),
              FALSE↦psvParamsValC(max(dom(psvParamsValC)))}(sv)
             pcvG={TRUE↦
  grd10  :   pcvParamsValG(max(dom(pcvParamsValG))),
              FALSE↦pcvParamsValC(max(dom(pcvParamsValC)))}(sv)
             pcvC={TRUE↦
  grd11  :   pcvParamsValG(max(dom(pcvParamsValG))),
              FALSE↦pcvParamsValC(max(dom(pcvParamsValC)))}(sv)
             comG={TRUE↦
  grd12  :   comParamsValG(max(dom(comParamsValG))),
              FALSE↦comParamsValC(max(dom(comParamsValC)))}(sv)
             comC={TRUE↦
  grd13  :   comParamsValG(max(dom(comParamsValG))),
              FALSE↦comParamsValC(max(dom(comParamsValC)))}(sv)
  grd14  :   PCV2PSV=TRUE ∧ modeC=PCV⇒modec=PSV
  grd15  :   PCV2PSV=FALSE ∨ modeC≠PCV⇒modec=modeC
             modec=PSV
             ⇒
  grd16  :   dom(psvC)=psvParams ∧
             psvC(ApneaLag)≥max_insp_time_psv÷2
  grd17  :   modeC≠modec ⇒PCV2PSV=TRUE ∧ modeC=PCV ∧ modec=PSV
             modec=PCV∧ sv=TRUE⇒
  grd18  :   10*60*pcvC(IEPCV)÷ (pcvC(RRPCV)*(1 +pcvC(IEPCV))) >0
  grd19  :   curTime∉dom(pcvParamsValC)
  grd20  :   ∀c·c∈cycles⇒ inspBegT(c)≠curTime
             sv=TRUE
             ⇒
             outs=bool(
             (
             (∃ p·p ∈ comParams  ∧ comC(p)∉domcomParams(p))
  grd21  :   ∨
             (∃ p·p ∈ pcvParams  ∧ pcvC(p)∉dompcvParams(p))
             ∨
              (∃ p·p ∈ dom(psvC)  ∧ psvC(p)∉dompsvParams(p))
             )
             )
             sv=FALSE
  grd22  :   ⇒
             outs=alarmRaised(outsideValue)
THEN
  act1   :   modeG≔modeg
  act2   :   modeGP≔modeG
  act3   :   modeC≔modec
  act4   :   modeCP≔modeC
  act5   :   psvParamsValG(curTime)≔psvG
  act6   :   psvParamsValC(curTime)≔psvC
  act7   :   pcvParamsValG(curTime)≔pcvG
  act8   :   pcvParamsValC(curTime)≔pcvC
  act9   :   comParamsValG(curTime)≔comG
  act10  :   comParamsValC(curTime)≔comC
  act11  :   PCV2PSV≔FALSE
  act12  :   alarmRaised(outsideValue)≔outs
END

startStopPCVPSV   ≙
  extended
STATUS
  ordinary
REFINES
  startStopPCVPSV
ANY
  modeg
  modec
WHERE
  grd1   :   modeG∈{Menu}∪Ventilation
```

```
  grd2   :   modeg∈Ventilation∪
             {Menu}
  grd3   :   modeG=Menu⇒modeg∈Ventilation
  grd4   :   modeG∈Ventilation⇒modeg=Menu
  grd5   :   modeC≠FailSafe
  grd6   :   modeg∈Ventilation ⇒ modec=modeg
  grd7   :   modeg∉Ventilation ⇒ modec=VentilationOff
             modeg=PSV
  grd8   :   ⇒
             dom(psvParamsValC(max(dom(psvParamsValC))))=psvParams
             modeg=PSV
  grd9   :   ⇒
             (psvParamsValC(max(dom(psvParamsValC))))(ApneaLag)≥max_insp_time_psv÷2
              modec=PCV⇒
  grd10  :   10*60*(pcvParamsValC(max(dom(pcvParamsValC))))(IEPCV)÷
               ((pcvParamsValC(max(dom(pcvParamsValC))))(RRPCV)*
             (1 +(pcvParamsValC(max(dom(pcvParamsValC))))(IEPCV))) >0
  grd11  :   cycles=ventilPhase~[{expEnd,expPauseEnd}]
THEN
  act1   :   modeG≔modeg
  act2   :   modeGP≔modeG
  act3   :   modeC≔modec
  act4   :   modeCP≔{TRUE↦modeC,FALSE↦modeCP}(bool(modeC∈Ventilation ∪{VentilationOff}))
              inValve≔{TRUE↦
  act18  :                {FALSE↦FALSE, TRUE↦inValve}(inValveF),
                      FALSE↦inValve}(bool(modec∉Ventilation))
             outValve≔{TRUE↦
  act19  :                  {FALSE↦TRUE, TRUE↦outValve}(outValveF),
                       FALSE↦outValve}(bool(modec∉Ventilation))
END

moveToPSV   ≙
  extended
STATUS
  ordinary
REFINES
  moveToPSV
ANY
  modeg
  modec
WHERE
  grd1   :   modeG∈ Ventilation
  grd2   :   modeg∈Ventilation∪{Settings}
  grd3   :   modeC∈Ventilation\{FailSafe}
  grd4   :   modec∈Ventilation
  grd5   :   modeG=PCV
  grd6   :   modeC=PCV
  grd7   :   modeg=Settings
  grd8   :   modec=modeC
THEN
  act1   :   modeG≔modeg
  act2   :   modeGP≔modeG
  act3   :   modeC≔modec
  act4   :   modeCP≔modeC
  act5   :   PCV2PSV≔TRUE
END

failExternalPower   ≙
  extended
STATUS
  ordinary
REFINES
  failExternalPower
ANY
  modeg
  modegp
  modec
  modecp
  pcv2psv
  cycs
WHERE
```

```
  grd1   :   onAC=TRUE
  grd2   :   modeg ∈ ModesG ∧ modegp ∈ ModesG
  grd3   :   ¬(power=TRUE ∧ crashed=FALSE) ⇒ modeg=Off ∧ modegp=Off
             power=TRUE ∧ crashed=FALSE ⇒
             (
  grd4   :      ((switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE ⇒ modeg=modeG ∧ modegp=modeGP) ∧
                (¬(switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)⇒ modeg=Off ∧ modegp=modeG))
             )
  grd5   :   modec ∈ ModesC ∧ modecp ∈ ModesC
  grd6   :   power= FALSE ⇒ modec=modeC ∧ modecp=modeCP
              power= TRUE ⇒
              (
  grd7   :       ((switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE ⇒modec=modeC ∧ modecp=modeCP) ∧
                  (¬(switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)⇒modec=Off ∧ modecp=modeC))
                 )
  grd8   :   ¬(power=TRUE ∧ crashed=FALSE) ⇒ pcv2psv=FALSE
             power=TRUE ∧ crashed=FALSE ⇒
             (
  grd9   :   ((switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE⇒ pcv2psv=PCV2PSV) ∧
             (¬(switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)⇒ pcv2psv=FALSE))
                     )
              (switchover=TRUE ∧ batLev>0 ∧ batFail=FALSE)
  grd10  :      ⇒
              cycs=cycles
              (switchover=FALSE ∨ batLev=0 ∨  batFail=TRUE)
  grd11  :   ⇒
              cycs=ventilPhase~[{expEnd,expPauseEnd}]
  grd12  :   cycs⊆cycles
THEN
  act1   :   onAC≔FALSE
  act2   :   modeG≔modeg
  act3   :   modeGP≔modegp
  act4   :   modeC≔modec
  act5   :   modeCP≔modecp
  act6   :   PCV2PSV≔pcv2psv
  act7   :   ventilPhase≔cycs◁ventilPhase
  act8   :   inspBegT≔cycs◁inspBegT
  act9   :   inspEndT≔cycs◁inspEndT
  act10  :   cycles≔cycs
  act11  :   cycleMode≔ cycs◁cycleMode
  act12  :   inspPauseBegT≔cycs◁inspPauseBegT
  act13  :   inspPauseEndT≔cycs◁inspPauseEndT
  act14  :   rmBegT≔cycs◁rmBegT
  act15  :   rmEndT≔cycs◁rmEndT
  act16  :   expBegT≔cycs◁expBegT
  act17  :   expEndT≔cycs◁expEndT
  act18  :   expPauseBegT≔cycs◁expPauseBegT
  act19  :   expPauseEndT≔cycs◁expPauseEndT
              inValve≔{TRUE↦
  act20  :                {FALSE↦FALSE, TRUE↦inValve}(inValveF),
                      FALSE↦inValve}(bool(modec=Off))
              outValve≔{TRUE↦
  act21  :                 {FALSE↦TRUE, TRUE↦outValve}(outValveF),
                       FALSE↦outValve}(bool(modec=Off))
  act22  :   alarmRaised≔{TRUE↦alarmRaised, FALSE↦Alarms×{FALSE}}(bool(modec≠Off))
END

failSelfTest   ≙
  extended
STATUS
  ordinary
REFINES
  failSelfTest
WHEN
  grd1   :   modeC=SelfTest
  grd2   :   FI1=FALSE ∨ FI2=FALSE ∨ oxygenSensor=FALSE ∨ switchover=FALSE ∨
             guiContCom=FALSE ∨ inValve=TRUE ∨ outValve=FALSE
THEN
  act1   :   modeC≔FailSafe
  act2   :   modeCP≔modeC
  act3   :   alarmRaised≔ alarmRaised◃
```

```
            {patConnected↦FALSE, FI1Failure↦ bool(FI1=FALSE), FI2Failure↦bool(FI2=FALSE),
             oxygenSensorFailure↦bool(oxygenSensor=FALSE),
             switchoverFailure↦bool(switchover=FALSE),
             conguiComFailure ↦bool(guiContCom=FALSE)}
END

progress  ≙
  extended
STATUS
  ordinary
REFINES
  progress
ANY
  step
  l
  modeg
  batf
  modec
  paw
  cycs
  alarmInV
  outside
WHERE
  grd1   :  step∈N1 ∧ l∈batteryRange ∧ batf∈BOOL
  grd2   :  batFail=TRUE ⇒ l=batLev
  grd3   :  onAC=TRUE ∧ batFail=FALSE ⇔ l=batLev+step
  grd4   :  onAC=FALSE ∧ batFail=FALSE ⇔ l=max({0,batLev−step})
            (onAC=TRUE ∨ (l>0 ∧ switchover=TRUE ∧ batf=FALSE)) ∧
            power=TRUE ∧  modeG=Off ∧
  grd5   :  crashed=FALSE
            ⇒
            modeg=StartUp
  grd6   :  (l=0 ∨ batf=TRUE ∨ switchover=FALSE) ∧ onAC=FALSE ⇒modeg=Off
            onAC=TRUE  ∨
  grd7   :  ((l>0 ∧ batLev>0) ∧ switchover=TRUE ∧ batf=FALSE)
            ⇒
            modeg=modeG
  grd8   :  modec∈{FailSafe,modeC, Off,StartUp}
  grd9   :  modec=FailSafe⇒modeC≠Off
            (onAC=TRUE ∨ (l>0 ∧ switchover=TRUE ∧ batf=FALSE)) ∧
  grd10  :  power=TRUE ∧  modeC=Off
            ⇒
            modec=StartUp
            (batLev>0 ∧ l>0)∨  switchover=FALSE ∨ onAC=TRUE ∨
  grd11  :  power=FALSE
            ⇒
            modec∈{modeC,FailSafe}
            (l=0 ∨ batf=TRUE ∨ switchover=FALSE) ∧ onAC=FALSE
  grd12  :  ⇒
            modec=Off
  grd13  :  step∈N1 ∧ paw∈N
            ∀c· c∈cycles ∧ ventilPhase(c)=inspBeg
  grd14  :  ⇒
            curTime+step≤inspEndT(c)
            ∀c· c∈cycles ∧ ventilPhase(c)=inspPauseBeg
  grd15  :  ⇒
            curTime+step≤inspPauseEndT(c)
            ∀c· c∈cycles ∧ ventilPhase(c)=rmBeg
  grd16  :  ⇒
            curTime+step≤ rmEndT(c)
            ∀c· c∈cycles ∧ ventilPhase(c)=expBeg
  grd17  :  ⇒
            curTime+step∈expEndT(c)
            ∀c· c∈cycles ∧ ventilPhase(c)=expPauseBeg
  grd18  :  ⇒
            curTime+step ≤ expPauseEndT(c)
  grd19  :  modec∉Ventilationu{FailSafe} ⇒ cycs=ventilPhase~[{expEnd,expPauseEnd}]
  grd20  :  modec∈Ventilationu{FailSafe}  ⇒ cycs=cycles
  grd21  :  cycs⊆cycles
  grd22  :  modec∉{Off, StartUp,SelfTest}
            ⇒
            alarmInV=
```

```
            bool(∃c· (c∈cycles ∧
                      (
                      (
                      (
                      (ventilPhase(c)=inspBeg ∧ curTime+step >inspBegT(c))∨
                               (ventilPhase(c)=rmBeg ∧ curTime+step > rmBegT(c))
                      ) ∧
                       inValve=FALSE)
                      ∨
                      ((
                      (ventilPhase(c)=expPauseBeg ∧ curTime+step >expPauseBegT(c))
                      ∨
                     (ventilPhase(c)=expBeg ∧ curTime+step >expBegT(c))
                      ∨
                      (ventilPhase(c)=inspPauseBeg ∧ curTime+step >inspPauseBegT(c))
                    )
                    ∧ inValve=TRUE))
                )
            )
            modec∈{Off, StartUp,SelfTest}
  grd23  :  ⇒
            alarmInV=FALSE
            alarmInV=TRUE
  grd24  :  ⇒
            modec=FailSafe
            ¬(oxygenSensor=TRUE ∧ FI1=TRUE ∧ FI2=TRUE ∧ guiContCom=TRUE)
  grd25  :  ⇒
            modec=FailSafe
            modec≠Off
  grd26  :  ⇒
            outside=alarmRaised(outsideValue)
            modec=Off
  grd27  :  ⇒
            outside=FALSE
THEN
  act1   :  curTime≔curTime+step
  act2   :  batLev≔l
  act3   :  modeG≔modeg
  act4   :  modeGP≔{TRUE↦modeGP, FALSE↦modeG}(bool(modeg=modeG))
  act5   :  batFail≔batf
  act6   :  modeC≔modec
  act7   :  modeCP≔{TRUE↦modeCP,FALSE↦modeC}(bool(modeC=modec))
  act8   :  PCV2PSV≔{TRUE↦PCV2PSV, FALSE↦FALSE}(bool(
            (batLev>0 ∧ l>0∧  switchover=FALSE) ∨ onAC=TRUE))
  act9   :  ventilPhase≔cycs◁ventilPhase
  act10  :  inspBegT≔cycs◁inspBegT
  act11  :  inspEndT≔cycs◁inspEndT
  act12  :  cycles≔cycs
  act13  :  cycleMode≔cycs◁cycleMode
  act14  :  inspPauseBegT≔cycs◁inspPauseBegT
  act15  :  inspPauseEndT≔cycs◁inspPauseEndT
  act16  :  rmBegT≔cycs◁rmBegT
  act17  :  rmEndT≔cycs◁rmEndT
  act18  :  expBegT≔cycs◁expBegT
  act19  :  expEndT≔cycs◁expEndT
  act20  :  expPauseBegT≔cycs◁expPauseBegT
  act21  :  expPauseEndT≔cycs◁expPauseEndT
  act22  :  inValve≔{TRUE↦inValve,FALSE↦FALSE}(bool(modec=modeC))
  act23  :  outValve≔{TRUE↦outValve,FALSE↦TRUE}(bool(modec=modeC))
            alarmRaised≔alarmRaised◁{patConnected↦
                bool(patientConnected=TRUE ∧ modec∈{StartUp,SelfTest}),
            inValveFailure ↦ alarmInV,
            oxygenSensorFailure ↦bool(
            oxygenSensor=FALSE ∧ modec∉{Off, StartUp,SelfTest}),
            switchoverFailure↦bool(switchover=FALSE  ∧ modec∉{Off, StartUp,SelfTest}),
  act24  :  FI1Failure↦bool(FI1=FALSE ∧  modec∉{Off, StartUp,SelfTest}),
            FI2Failure↦bool(FI2=FALSE ∧  modec∉{Off, StartUp,SelfTest}),
            outsideValue↦outside,
            conguiComFailure↦bool(guiContCom=FALSE ∧
                                  modec∉{Off, StartUp,SelfTest})
            }
  act25  :  inValveP≔inValve
END
```

```
switchoverFail   ≙
  extended
STATUS
  ordinary
REFINES
  switchoverFail
ANY
  modeg
  modec
  cycs
WHERE
  grd1   :   switchover=TRUE
  grd2   :   onAC=FALSE⟹modeg=Off
  grd3   :   onAC=TRUE⟹modeg=modeG
  grd4   :   onAC=FALSE⟹modec=Off
  grd5   :   onAC=TRUE⟹modec=modeC
  grd6   :   modec∉Ventilationu{FailSafe} ⟹ cycs=ventilPhase~[{expEnd,expPauseEnd}]
  grd7   :   modec∈Ventilationu{FailSafe}  ⟹ cycs=cycles
  grd8   :   cycs⊆cycles
THEN
  act1   :   switchover≔FALSE
  act2   :   modeG≔modeg
  act3   :   modeGP≔{TRUE↦modeG,FALSE↦modeGP}(bool(modeG≠modeg))
  act4   :   modeC≔modec
  act5   :   modeCP≔{TRUE↦modeC,FALSE↦modeCP}(bool(modeC≠modec))
  act6   :   PCV2PSV≔{TRUE↦PCV2PSV, FALSE↦FALSE}(bool(onAC=TRUE))
  act7   :   ventilPhase≔cycs⩤ventilPhase
  act8   :   inspBegT≔cycs⩤inspBegT
  act9   :   inspEndT≔cycs⩤inspEndT
  act10   :   cycles≔cycs
  act11   :   cycleMode≔cycs⩤cycleMode
  act12   :   inspPauseBegT≔cycs⩤inspPauseBegT
  act13   :   inspPauseEndT≔cycs⩤inspPauseEndT
  act14   :   rmBegT≔cycs⩤rmBegT
  act15   :   rmEndT≔cycs⩤rmEndT
  act16   :   expBegT≔cycs⩤expBegT
  act17   :   expEndT≔cycs⩤expEndT
  act18   :   expPauseBegT≔cycs⩤expPauseBegT
  act19   :   expPauseEndT≔cycs⩤expPauseEndT
  act20   :   inValve≔{TRUE↦inValve,FALSE↦FALSE}(bool(modec=modeC))
  act21   :   outValve≔{TRUE↦outValve,FALSE↦TRUE}(bool(modec=modeC))
              alarmRaised≔{FALSE↦Alarms×{FALSE},
  act22   :     TRUE↦alarmRaised⩤
                          {switchoverFailure↦bool(modeC∉{Off, StartUp,SelfTest})}
                }(onAC)
END

inspStart   ≙
  extended
STATUS
  ordinary
REFINES
  inspStart
ANY
  cy
  paw
  ve
  inspT
WHERE
  grd1   :   modeC∈Ventilation ∧ inspT∈N
  grd2   :   cy∈Cycles\cycles
  grd3   :   ran(ventilPhase)⊆{expEnd,expPauseEnd}
             modeC=PCV⟹inspT=curTime + 10*60*
  grd4   :    (pcvParamsValC(max(dom(pcvParamsValC))))(IEPCV)÷
                    ((pcvParamsValC(max(dom(pcvParamsValC))))(RRPCV)*
                  (1 +(pcvParamsValC(max(dom(pcvParamsValC))))(IEPCV)))
             modeC=PSV⟹
  grd5   :   inspT>curTime ∧
             inspT≤curTime + max_insp_time_psv
  grd6   :   paw∈N ∧ ve∈ N
```

```
  grd7   :   pcvParamsValC≠ø ∧ psvParamsValC≠ø ∧ comParamsValC≠ø
THEN
  act1   :   ventilPhase(cy):=inspBeg
  act2   :   cycles:=cyclesu{cy}
  act3   :   inspBegT(cy):=curTime
  act4   :   inspEndT(cy):=inspT
  act5   :   cycleMode(cy):=modeC
  act6   :   PAW(curTime):=paw
  act7   :   VE(curTime):=ve
  act8   :   inValve:={FALSE↦TRUE, TRUE↦inValve}(inValveF)
  act9   :   outValve:={FALSE↦FALSE, TRUE↦outValve}(outValveF)
END

inspEnd   ≙
  extended
STATUS
  ordinary
REFINES
  inspEnd
ANY
  cy
  ve
  peakVE
WHERE
  grd1   :   cy∈ cycles ∧ ventilPhase(cy)=inspBeg ∧ ve∈N
  grd2   :   cycleMode(cy)=PSV⟹curTime>inspBegT(cy)
  grd3   :   cycleMode(cy)=PCV ⟹curTime=inspEndT(cy)
  grd4   :   peakVE=VE(max(dom(VE)))
  grd5   :   (curTime=inspEndT(cy)) ∨
             (cycleMode(cy)=PSV ⟹
             peakVE>ve ∧
             ve<(psvParamsValC(max(dom(psvParamsValC))))(ETS)∗peakVE)
  grd6   :   modeC∈Ventilation
THEN
  act1   :   ventilPhase(cy):=inspEnd
  act2   :   VE(curTime):=ve
  act3   :   inspEndT(cy):=curTime
END

inspPauseStart   ≙
  extended
STATUS
  ordinary
REFINES
  inspPauseStart
ANY
  cy
WHERE
  grd1   :   cy∈cycles ∧ ventilPhase(cy)=inspEnd
  grd2   :   modeC∈Ventilation
THEN
  act1   :   ventilPhase(cy):=inspPauseBeg
  act2   :   inspPauseBegT(cy):=curTime
  act3   :   inspPauseEndT(cy):=curTime+inspPauseMax
  act4   :   inValve:={FALSE↦FALSE, TRUE↦inValve}(inValveF)
END

inspPauseEnd   ≙
  extended
STATUS
  ordinary
REFINES
  inspPauseEnd
ANY
  cy
WHERE
  grd1   :   cy∈cycles ∧ ventilPhase(cy)=inspPauseBeg
  grd2   :   curTime≤inspPauseEndT(cy)
  grd3   :   curTime>inspPauseBegT(cy)
  grd4   :   modeC∈Ventilation
THEN
```

```
  act1   :   ventilPhase(cy)
             ≔inspPauseEnd
  act2   :   inspPauseEndT(cy)≔curTime
END

rmStart   ≙
  extended
STATUS
  ordinary
REFINES
  rmStart
ANY
  cy
  t
WHERE
  grd1   :   cy∈cycles
  grd2   :   ventilPhase(cy)∈{inspEnd, inspPauseEnd}
  grd3   :   t=max({x|x∈ dom(comParamsValC) ∧ x≤  inspBegT(cy)})
  grd4   :   modeC∈Ventilation
THEN
  act1   :   ventilPhase(cy)≔rmBeg
  act2   :   rmBegT(cy)≔curTime
  act3   :   rmEndT(cy)≔curTime+(comParamsValC(t))(timerRM)
  act4   :   inValve≔{FALSE↦TRUE, TRUE↦inValve}(inValveF)
END

rmEnd   ≙
  extended
STATUS
  ordinary
REFINES
  rmEnd
ANY
  cy
WHERE
  grd1   :   cy∈cycles
  grd2   :   ventilPhase(cy)=rmBeg
  grd3   :   curTime=rmEndT(cy)
  grd4   :   modeC∈Ventilation
THEN
  act1   :   ventilPhase(cy)≔rmEnd
END

expStart   ≙
  extended
STATUS
  ordinary
REFINES
  expStart
ANY
  cy
  t
  expT
WHERE
  grd1   :   cy∈cycles ∧  ventilPhase(cy)∈{inspEnd, inspPauseEnd,rmEnd}
  grd2   :   t=max({x|x∈ dom(pcvParamsValC) ∧ x≤  inspBegT(cy)})
  grd3   :   expT⊆N
             cycleMode(cy)=PCV ⟹
  grd4   :   expT={curTime +60÷((pcvParamsValC(t))(RRPCV) *
                        (1 +(pcvParamsValC(t))(IEPCV)))}
             cycleMode(cy)=PSV ⟹
             expT=
  grd5   :   (curTime+(inspEndT(cy)−inspBegT(cy))÷2)
             ‥
             (curTime +((psvParamsValC(t)))(ApneaLag))
  grd6   :   modeC∈Ventilation
THEN
  act1   :   ventilPhase(cy)≔expBeg
  act2   :   expBegT(cy)≔curTime
  act3   :   expEndT(cy)≔expT
  act4   :   inValve≔{FALSE↦FALSE, TRUE↦inValve}(inValveF)
  act5   :   outValve≔{FALSE↦TRUE, TRUE↦outValve}(outValveF)
```

```
END

expEnd   ≙
  extended
STATUS
  ordinary
REFINES
  expEnd
ANY
  modeg
  modec
  pcv
  paw
  cy
  lastTime
  t
  alarmOutValue
WHERE
  grd1   :   modeG∈ Ventilation
  grd2   :   modeg∈Ventilation∪{Settings}
  grd3   :   modeC∈Ventilation\{FailSafe}
  grd4   :   modec∈Ventilation
  grd5   :   pcv ∈ pcvParams⟶N1
  grd6   :   modeC=PCV ⇒ modec=PCV
             modeC=PSV ∧ modec=PCV
             ⇒
  grd7   :   pcv=pcvParamsValC(max(dom(pcvParamsValC)))⩤
             {RRPCV↦(psvParamsValC(max(dom(psvParamsValC))))(RRAP),
              PinspPCV↦(psvParamsValC(max(dom(psvParamsValC))))(PinspAP),
             IEPCV↦IEAP}
             modeC=PCV ∨ modec=PSV
  grd8   :   ⇒
             pcv=pcvParamsValC(max(dom(pcvParamsValC)))
             modec=PCV
  grd9   :   ⇒
             10*60*pcv(IEPCV)÷ (pcv(RRPCV)*(1 +pcv(IEPCV))) >0
  grd11   :   modeG≠Off⇒ modeg=modec
  grd12   :   cy∈cycles
  grd13   :   ventilPhase(cy)=expBeg
  grd14   :   lastTime= max(dom(PAW))
  grd15   :   paw∈N
  grd16   :   t=max({x | x∈dom(psvParamsValC)∧x≤inspBegT(cy)})
              (
              cycleMode(cy)=PCV ∧
                  (curTime∈ expEndT(cy) ∨
                    (curTime<max(expEndT(cy)) ∧
                      (PAW(lastTime)<paw ∧ PAW(lastTime)−paw>(pcvParamsValC(t))(ITSPCV)))
                  )
              )
  grd17   :   ∨
              (
              cycleMode(cy)=PSV ∧
                  (curTime= max(expEndT(cy)) ∨
                    (curTime≥min(expEndT(cy)) ∧ curTime<max(expEndT(cy)) ∧ PAW(lastTime)<paw ∧ PAW
              (lastTime)−paw>(psvParamsValC(t))(ITSPSV)))
              )
              curTime=max(expEndT(cy))∧ cycleMode(cy)=PSV
  grd18   :   ⇒
              modec=PCV
              curTime≠max(expEndT(cy))∧ cycleMode(cy)=PSV
  grd19   :   ⇒
              modec=PSV
              cycleMode(cy)=PCV
  grd20   :   ⇒
              modec=PCV
  grd21   :   modeC∈Ventilation
  grd22   :   modeC=PSV ∧ modec=PCV
              ⇒
              alarmOutValue=bool(
              (∃ p· p ∈ pcvParams  ∧ pcv(p)∉dompcvParams(p))
              ∨
              (∃ p· p ∈ comParams  ∧
```

```
                (comParamsValC(max(dom(comParamsValC))))(p)∉domcomParams(p))
               ∨
               (∃ p· p ∈ dom((psvParamsValC(max(dom(psvParamsValC))))  ∧
                (psvParamsValC(max(dom(psvParamsValC))))(p)∉dompsvParams(p))
               )
               modeC=PCV ∨  modec=PSV
  grd23   :    ⇒
               alarmOutValue=alarmRaised(outsideValue)
THEN
  act1    :   modeG≔modeg
  act2    :   modeGP≔modeG
  act3    :   modeC≔modec
  act4    :   modeCP≔modeC
  act5    :   pcvParamsValC(curTime)≔pcv
  act6    :   psvParamsValC(curTime)≔psvParamsValC(max(dom(psvParamsValC)))
  act7    :   comParamsValC(curTime)≔comParamsValC(max(dom(comParamsValC)))
  act8    :   ventilPhase(cy)≔expEnd
  act9    :   PAW(curTime)≔paw
  act10    :   expEndT(cy)≔curTime‥curTime
  act11    :   alarmRaised(outsideValue)≔alarmOutValue
END

expPauseStart   ≙
  extended
STATUS
  ordinary
REFINES
  expPauseStart
ANY
  cy
WHERE
  grd1    :   cy∈cycles ∧ ventilPhase(cy)=expEnd
  grd2    :   modeC∈Ventilation
  grd3    :   ran(ventilPhase)⊆{expEnd,expPauseEnd}
THEN
  act1    :   ventilPhase(cy)≔expPauseBeg
  act2    :   expPauseBegT(cy)≔curTime
  act3    :   expPauseEndT(cy)≔curTime+expPauseMax
  act4    :   inValve≔{FALSE↦FALSE, TRUE↦inValve}(inValveF)
  act5    :   outValve≔{FALSE↦FALSE, TRUE↦outValve}(outValveF)
END

expPauseEnd   ≙
  extended
STATUS
  ordinary
REFINES
  expPauseEnd
ANY
  cy
WHERE
  grd1    :   cy∈cycles ∧ ventilPhase(cy)=expPauseBeg
  grd2    :   curTime≤expPauseEndT(cy)
  grd3    :   curTime>expPauseBegT(cy)
  grd4    :   modeC∈Ventilation
THEN
  act1    :   ventilPhase(cy)≔expPauseEnd
  act2    :   expPauseEndT(cy)≔curTime
END

inValveF   ≙
  extended
STATUS
  ordinary
REFINES
  inValveF
WHEN
  grd1    :   inValveF=FALSE
THEN
  act1    :   inValveF≔TRUE
END
```

```
  outValveF   ≙
    extended
  STATUS
    ordinary
  REFINES
    outValveF
  WHEN
    grd1   :   outValveF=FALSE
  THEN
    act1   :   outValveF≔TRUE
  END

  connectPatient   ≙
  STATUS
    ordinary
  WHEN
    grd1   :   patientConnected=FALSE
  THEN
    act1   :   patientConnected≔TRUE
    act2   :   alarmRaised(patConnected)≔bool(modeC∈{StartUp,SelfTest})
  END

  disconnectPatient   ≙
  STATUS
    ordinary
  WHEN
    grd1   :   patientConnected=TRUE
  THEN
    act1   :   patientConnected≔FALSE
    act2   :   alarmRaised(patConnected)≔FALSE
  END

  FI1Fail   ≙
  STATUS
    ordinary
  WHEN
    grd1   :   FI1=TRUE
  THEN
    act1   :   FI1≔FALSE
    act2   :   alarmRaised(FI1Failure)≔bool(modeC∉{Off, StartUp, SelfTest})
  END

  FI2Fail   ≙
  STATUS
    ordinary
  WHEN
    grd1   :   FI2=TRUE
  THEN
    act1   :   FI2≔FALSE
    act2   :   alarmRaised(FI2Failure)≔bool(modeC∉{Off, StartUp,SelfTest})
  END

  oxygenSensorFail   ≙
  STATUS
    ordinary
  WHEN
    grd1   :   oxygenSensor=TRUE
  THEN
    act1   :   oxygenSensor≔FALSE
    act2   :   alarmRaised(oxygenSensorFailure)≔bool(modeC∉{Off, StartUp,SelfTest})
  END

END
```