

# aircraft

Tight Rope v0.88

5th March 2017

## 1 ID Files

### 1.1 MissionIds

**section** *MissionIds* **parents** *scj\_prelude*, *MissionId*

*MainMissionMID : MissionID*  
*TakeOffMissionMID : MissionID*  
*CruiseMissionMID : MissionID*  
*LandMissionMID : MissionID*

---

*distinct*(*nullMissionId*, *MainMissionMID*, *TakeOffMissionMID*,  
*CruiseMissionMID*, *LandMissionMID*)

## 1.2 SchedulablesIds

**section** *SchedulableIds* **parents** *scj\_prelude*, *SchedulableId*

*MainMissionSequencerSID* : *SchedulableID*  
*ACModeChanger2SID* : *SchedulableID*  
*EnvironmentMonitorSID* : *SchedulableID*  
*ControlHandlerSID* : *SchedulableID*  
*FlightSensorsMonitorSID* : *SchedulableID*  
*CommunicationsHandlerSID* : *SchedulableID*  
*LandingGearHandlerSID* : *SchedulableID*  
*TakeOffMonitorSID* : *SchedulableID*  
*TakeOffFailureHandlerSID* : *SchedulableID*  
*BeginLandingHandlerSID* : *SchedulableID*  
*NavigationMonitorSID* : *SchedulableID*  
*GroundDistanceMonitorSID* : *SchedulableID*  
*LandingGearHandlerLandSID* : *SchedulableID*  
*InstrumentLandingSystemMonitorSID* : *SchedulableID*  
*SafeLandingHandlerSID* : *SchedulableID*

*distinct*(*nullSequencerId*, *nullSchedulableId*, *MainMissionSequencerSID*,  
*ACModeChanger2SID*, *EnvironmentMonitorSID*,  
*ControlHandlerSID*, *FlightSensorsMonitorSID*,  
*CommunicationsHandlerSID*, *LandingGearHandlerSID*,  
*TakeOffMonitorSID*, *TakeOffFailureHandlerSID*,  
*BeginLandingHandlerSID*, *NavigationMonitorSID*,  
*GroundDistanceMonitorSID*, *LandingGearHandlerLandSID*,  
*InstrumentLandingSystemMonitorSID*, *SafeLandingHandlerSID*)

### 1.3 Non-Paradigm Objects

## 1.4 ThreadIds

**section** *ThreadId* **parents** *scj\_prelude, GlobalTypes*

*SafeletTid* : *ThreadID*  
*nullThreadId* : *ThreadID*

---

*distinct*(*SafeletTid*, *nullThreadId*)

## 1.5 ObjectIds

**section** *ObjectIds* **parents** *scj\_prelude, GlobalTypes*

$distinct \langle \rangle$
----------------------------

## 2 Network

### 2.1 Network Channel Sets

```
section NetworkChannels parents scj_prelude, MissionId, MissionIds,  
    SchedulableId, SchedulableIds, MissionChan, TopLevelMissionSequencerFWChan,  
    FrameworkChan, SafeletChan, AperiodicEventHandlerChan, ManagedThreadChan,  
    OneShotEventHandlerChan, PeriodicEventHandlerChan, MissionSequencerMethChan  
  
channelset TerminateSync ==  
    { schedulables_terminated, schedulables_stopped, get_activeSchedulables }  
  
channelset ControlTierSync ==  
    { start_toplevel_sequencer, done_toplevel_sequencer, done_safeletFW }  
  
channelset TierSync ==  
    { start_mission . MainMission, done_mission . MainMission,  
      done_safeletFW, done_toplevel_sequencer }  
  
channelset MissionSync ==  
    { done_safeletFW, done_toplevel_sequencer, register,  
      signalTerminationCall, signalTerminationRet, activate_schedulables, done_schedulable,  
      cleanupSchedulableCall, cleanupSchedulableRet }  
  
channelset SchedulablesSync ==  
    { activate_schedulables, done_safeletFW, done_toplevel_sequencer }  
  
channelset ClusterSync ==  
    { done_toplevel_sequencer, done_safeletFW }  
  
channelset SafeltAppSync  $\hat{=}$   
    { getSequencerCall, getSequencerRet, initializeApplicationCall, initializeApplicationRet, end_safelet_app }  
  
channelset MissionSequencerAppSync ==  
    { getNextMissionCall, getNextMissionRet, end_sequencer_app }  
  
channelset MissionAppSync ==  
    { initializeCall, register, initializeRet, cleanupMissionCall, cleanupMissionRet }  
  
channelset AppSync ==  
    { SafeltAppSync, MissionSequencerAppSync, MissionAppSync,  
      MTAppSync, OSEHSync, APEHSync, PEHSync,  
      { getSequencer, end_mission_app, end_managedThread_app,  
        setCeilingPriority, requestTerminationCall, requestTerminationRet, terminationPendingCall,  
        terminationPendingRet, handleAsyncEventCall, handleAsyncEventRet } }  
  
channelset ThreadSync ==  
    { raise_thread_priority, lower_thread_priority, isInterruptedCall, isInterruptedRet, get_priorityLevel }  
  
channelset LockingSync ==  
    { lockAcquired, startSyncMeth, endSyncMeth, waitCall, waitRet, notify, isInterruptedCall, isInterruptedRet,  
      interruptedCall, interruptedRet, done_toplevel_sequencer, get_priorityLevel }  
  
channelset Tier0Sync ==  
    { done_toplevel_sequencer, done_safeletFW,  
      start_mission . TakeOffMission, done_mission . TakeOffMission,  
      initializeRet . TakeOffMission, requestTermination . TakeOffMission . MainMissionSequencer,  
      start_mission . CruiseMission, done_mission . CruiseMission,  
      initializeRet . CruiseMission, requestTermination . CruiseMission . MainMissionSequencer,  
      start_mission . LandMission, done_mission . LandMission,  
      initializeRet . LandMission, requestTermination . LandMission . MainMissionSequencer }
```

## 2.2 MethodCallBinder

**section** *MethodCallBindingChannels* **parents** *scj\_prelude, GlobalTypes, FrameworkChan, MissionId, MissionIds, SchedulableId, SchedulableIds, ThreadIds*

**channel** *binder\_setCabinPressureCall* : *MissionID*  $\times$  *SchedulableID*  $\times$   $\mathbb{P} \mathbb{A}$

**channel** *binder\_setCabinPressureRet* : *MissionID*  $\times$  *SchedulableID*

*setCabinPressureLocs* == { *MainMissionMID* }

*setCabinPressureCallers* == { *EnvironmentMonitorSID* }

**channel** *binder\_setFuelRemainingCall* : *MissionID*  $\times$  *SchedulableID*  $\times$   $\mathbb{P} \mathbb{A}$

**channel** *binder\_setFuelRemainingRet* : *MissionID*  $\times$  *SchedulableID*

*setFuelRemainingLocs* == { *MainMissionMID* }

*setFuelRemainingCallers* == { *EnvironmentMonitorSID* }

**channel** *binder\_getAltitudeCall* : *MissionID*  $\times$  *SchedulableID*

**channel** *binder\_getAltitudeRet* : *MissionID*  $\times$  *SchedulableID*  $\times$   $\mathbb{P} \mathbb{A}$

*getAltitudeLocs* == { *MainMissionMID* }

*getAltitudeCallers* == { *NavigationMonitorSID, TakeOffMonitorSID, GroundDistanceMonitorSID, SafeLandingHandlerS*

**channel** *binder\_setHeadingCall* : *MissionID*  $\times$  *SchedulableID*  $\times$   $\mathbb{P} \mathbb{A}$

**channel** *binder\_setHeadingRet* : *MissionID*  $\times$  *SchedulableID*

*setHeadingLocs* == { *MainMissionMID* }

*setHeadingCallers* == { *FlightSensorsMonitorSID* }

**channel** *binder\_stowLandingGearCall* : *MissionID*  $\times$  *SchedulableID*

**channel** *binder\_stowLandingGearRet* : *MissionID*  $\times$  *SchedulableID*

*stowLandingGearLocs* == { *TakeOffMissionMID, LandMissionMID* }

*stowLandingGearCallers* == { *LandingGearHandlerSID, LandingGearHandlerLandSID* }

**channel** *binder\_takeOffAbortCall* : *MissionID*  $\times$  *SchedulableID*

**channel** *binder\_takeOffAbortRet* : *MissionID*  $\times$  *SchedulableID*

*takeOffAbortLocs* == { *TakeOffMissionMID* }

*takeOffAbortCallers* == { *TakeOffFailureHandlerSID* }

**channel** *binder\_setAltitudeCall* : *MissionID*  $\times$  *SchedulableID*  $\times$   $\mathbb{P} \mathbb{A}$

**channel** *binder\_setAltitudeRet* : *MissionID*  $\times$  *SchedulableID*

*setAltitudeLocs* == { *MainMissionMID* }

*setAltitudeCallers* == { *FlightSensorsMonitorSID* }

**channel** *binder\_getHeadingCall* : *MissionID*  $\times$  *SchedulableID*  
**channel** *binder\_getHeadingRet* : *MissionID*  $\times$  *SchedulableID*  $\times$   $\mathbb{P}\mathbb{A}$

*getHeadingLocs* == { *MainMissionMID* }  
*getHeadingCallers* == { *NavigationMonitorSID* }

**channel** *binder\_getAirSpeedCall* : *MissionID*  $\times$  *SchedulableID*  
**channel** *binder\_getAirSpeedRet* : *MissionID*  $\times$  *SchedulableID*  $\times$   $\mathbb{P}\mathbb{A}$

*getAirSpeedLocs* == { *MainMissionMID* }  
*getAirSpeedCallers* == { *NavigationMonitorSID*, *TakeOffFailureHandlerSID* }

**channel** *binder\_deployLandingGearCall* : *MissionID*  $\times$  *SchedulableID*  
**channel** *binder\_deployLandingGearRet* : *MissionID*  $\times$  *SchedulableID*

*deployLandingGearLocs* == { *TakeOffMissionMID*, *LandMissionMID* }  
*deployLandingGearCallers* == { *LandingGearHandlerSID*, *LandingGearHandlerLandSID* }

**channel** *binder\_setEmergencyOxygenCall* : *MissionID*  $\times$  *SchedulableID*  $\times$   $\mathbb{P}\mathbb{A}$   
**channel** *binder\_setEmergencyOxygenRet* : *MissionID*  $\times$  *SchedulableID*

*setEmergencyOxygenLocs* == { *MainMissionMID* }  
*setEmergencyOxygenCallers* == { *EnvironmentMonitorSID* }

**channel** *binder\_setAirSpeedCall* : *MissionID*  $\times$  *SchedulableID*  $\times$   $\mathbb{P}\mathbb{A}$   
**channel** *binder\_setAirSpeedRet* : *MissionID*  $\times$  *SchedulableID*

*setAirSpeedLocs* == { *MainMissionMID* }  
*setAirSpeedCallers* == { *FlightSensorsMonitorSID* }

**channel** *binder\_isLandingGearDeployedCall* : *MissionID*  $\times$  *SchedulableID*  
**channel** *binder\_isLandingGearDeployedRet* : *MissionID*  $\times$  *SchedulableID*  $\times$   $\mathbb{B}$

*isLandingGearDeployedLocs* == { *TakeOffMissionMID*, *LandMissionMID* }  
*isLandingGearDeployedCallers* == { *LandingGearHandlerSID*, *LandingGearHandlerLandSID* }

**channelset** *MethodCallBinderSync* == { *done\_toplevel\_sequencer*,  
*binder\_setCabinPressureCall*, *binder\_setCabinPressureRet*,  
*binder\_setFuelRemainingCall*, *binder\_setFuelRemainingRet*,  
*binder\_getAltitudeCall*, *binder\_getAltitudeRet*,  
*binder\_setHeadingCall*, *binder\_setHeadingRet*,  
*binder\_stowLandingGearCall*, *binder\_stowLandingGearRet*,  
*binder\_takeOffAbortCall*, *binder\_takeOffAbortRet*,  
*binder\_setAltitudeCall*, *binder\_setAltitudeRet*,  
*binder\_getHeadingCall*, *binder\_getHeadingRet*,  
*binder\_getAirSpeedCall*, *binder\_getAirSpeedRet*,  
*binder\_deployLandingGearCall*, *binder\_deployLandingGearRet*,  
*binder\_setEmergencyOxygenCall*, *binder\_setEmergencyOxygenRet*,  
*binder\_setAirSpeedCall*, *binder\_setAirSpeedRet*,  
*binder\_isLandingGearDeployedCall*, *binder\_isLandingGearDeployedRet* }



**section** *MethodCallBinder* **parents** *scj\_prelude, MissionId, MissionIds,*  
*SchedulableId, SchedulableIds, MethodCallBindingChannels*  
*, MainMissionMethChan, LandMissionMethChan*

**process** *MethodCallBinder*  $\hat{=}$  **begin**

*setCabinPressure\_MethodBinder*  $\hat{=}$   
 $\left( \begin{array}{l} \text{binder\_setCabinPressureCall} ? \text{loc} : (\text{loc} \in \text{setCabinPressureLocs}) ? \text{caller} : (\text{caller} \in \text{setCabinPressureCallers}) ? p1 \longrightarrow \\ \text{setCabinPressureCall} . \text{loc} . \text{caller} ! p1 \longrightarrow \\ \text{setCabinPressureRet} . \text{loc} . \text{caller} \longrightarrow \\ \text{binder\_setCabinPressureRet} . \text{loc} . \text{caller} \longrightarrow \\ \text{setCabinPressure\_MethodBinder} \end{array} \right)$

*setFuelRemaining\_MethodBinder*  $\hat{=}$   
 $\left( \begin{array}{l} \text{binder\_setFuelRemainingCall} ? \text{loc} : (\text{loc} \in \text{setFuelRemainingLocs}) ? \text{caller} : (\text{caller} \in \text{setFuelRemainingCallers}) ? p1 \longrightarrow \\ \text{setFuelRemainingCall} . \text{loc} . \text{caller} ! p1 \longrightarrow \\ \text{setFuelRemainingRet} . \text{loc} . \text{caller} \longrightarrow \\ \text{binder\_setFuelRemainingRet} . \text{loc} . \text{caller} \longrightarrow \\ \text{setFuelRemaining\_MethodBinder} \end{array} \right)$

*getAltitude\_MethodBinder*  $\hat{=}$   
 $\left( \begin{array}{l} \text{binder\_getAltitudeCall} ? \text{loc} : (\text{loc} \in \text{getAltitudeLocs}) ? \text{caller} : (\text{caller} \in \text{getAltitudeCallers}) \longrightarrow \\ \text{getAltitudeCall} . \text{loc} . \text{caller} \longrightarrow \\ \text{getAltitudeRet} . \text{loc} . \text{caller} ? \text{ret} \longrightarrow \\ \text{binder\_getAltitudeRet} . \text{loc} . \text{caller} ! \text{ret} \longrightarrow \\ \text{getAltitude\_MethodBinder} \end{array} \right)$

*setHeading\_MethodBinder*  $\hat{=}$   
 $\left( \begin{array}{l} \text{binder\_setHeadingCall} ? \text{loc} : (\text{loc} \in \text{setHeadingLocs}) ? \text{caller} : (\text{caller} \in \text{setHeadingCallers}) ? p1 \longrightarrow \\ \text{setHeadingCall} . \text{loc} . \text{caller} ! p1 \longrightarrow \\ \text{setHeadingRet} . \text{loc} . \text{caller} \longrightarrow \\ \text{binder\_setHeadingRet} . \text{loc} . \text{caller} \longrightarrow \\ \text{setHeading\_MethodBinder} \end{array} \right)$

*stowLandingGear\_MethodBinder*  $\hat{=}$   
 $\left( \begin{array}{l} \text{binder\_stowLandingGearCall} ? \text{loc} : (\text{loc} \in \text{stowLandingGearLocs}) ? \text{caller} : (\text{caller} \in \text{stowLandingGearCallers}) \longrightarrow \\ \text{stowLandingGearCall} . \text{loc} . \text{caller} \longrightarrow \\ \text{stowLandingGearRet} . \text{loc} . \text{caller} \longrightarrow \\ \text{binder\_stowLandingGearRet} . \text{loc} . \text{caller} \longrightarrow \\ \text{stowLandingGear\_MethodBinder} \end{array} \right)$

*takeOffAbort\_MethodBinder*  $\hat{=}$   
 $\left( \begin{array}{l} \text{binder\_takeOffAbortCall} ? \text{loc} : (\text{loc} \in \text{takeOffAbortLocs}) ? \text{caller} : (\text{caller} \in \text{takeOffAbortCallers}) \longrightarrow \\ \text{takeOffAbortCall} . \text{loc} . \text{caller} \longrightarrow \\ \text{takeOffAbortRet} . \text{loc} . \text{caller} \longrightarrow \\ \text{binder\_takeOffAbortRet} . \text{loc} . \text{caller} \longrightarrow \\ \text{takeOffAbort\_MethodBinder} \end{array} \right)$

*setAltitude\_MethodBinder*  $\hat{=}$   
 $\left( \begin{array}{l} \text{binder\_setAltitudeCall} ? \text{loc} : (\text{loc} \in \text{setAltitudeLocs}) ? \text{caller} : (\text{caller} \in \text{setAltitudeCallers}) ? p1 \longrightarrow \\ \text{setAltitudeCall} . \text{loc} . \text{caller} ! p1 \longrightarrow \\ \text{setAltitudeRet} . \text{loc} . \text{caller} \longrightarrow \\ \text{binder\_setAltitudeRet} . \text{loc} . \text{caller} \longrightarrow \\ \text{setAltitude\_MethodBinder} \end{array} \right)$

$$\text{getHeading\_MethodBinder} \hat{=} \left( \begin{array}{l} \text{binder\_getHeadingCall} ? \text{loc} : (\text{loc} \in \text{getHeadingLocs}) ? \text{caller} : (\text{caller} \in \text{getHeadingCallers}) \longrightarrow \\ \text{getHeadingCall} . \text{loc} . \text{caller} \longrightarrow \\ \text{getHeadingRet} . \text{loc} . \text{caller} ? \text{ret} \longrightarrow \\ \text{binder\_getHeadingRet} . \text{loc} . \text{caller} ! \text{ret} \longrightarrow \\ \text{getHeading\_MethodBinder} \end{array} \right)$$

$$\text{getAirSpeed\_MethodBinder} \hat{=} \left( \begin{array}{l} \text{binder\_getAirSpeedCall} ? \text{loc} : (\text{loc} \in \text{getAirSpeedLocs}) ? \text{caller} : (\text{caller} \in \text{getAirSpeedCallers}) \longrightarrow \\ \text{getAirSpeedCall} . \text{loc} . \text{caller} \longrightarrow \\ \text{getAirSpeedRet} . \text{loc} . \text{caller} ? \text{ret} \longrightarrow \\ \text{binder\_getAirSpeedRet} . \text{loc} . \text{caller} ! \text{ret} \longrightarrow \\ \text{getAirSpeed\_MethodBinder} \end{array} \right)$$

$$\text{deployLandingGear\_MethodBinder} \hat{=} \left( \begin{array}{l} \text{binder\_deployLandingGearCall} ? \text{loc} : (\text{loc} \in \text{deployLandingGearLocs}) ? \text{caller} : (\text{caller} \in \text{deployLandingGearCallers}) \longrightarrow \\ \text{deployLandingGearCall} . \text{loc} . \text{caller} \longrightarrow \\ \text{deployLandingGearRet} . \text{loc} . \text{caller} \longrightarrow \\ \text{binder\_deployLandingGearRet} . \text{loc} . \text{caller} \longrightarrow \\ \text{deployLandingGear\_MethodBinder} \end{array} \right)$$

$$\text{setEmergencyOxygen\_MethodBinder} \hat{=} \left( \begin{array}{l} \text{binder\_setEmergencyOxygenCall} ? \text{loc} : (\text{loc} \in \text{setEmergencyOxygenLocs}) ? \text{caller} : (\text{caller} \in \text{setEmergencyOxygenCallers}) \longrightarrow \\ \text{setEmergencyOxygenCall} . \text{loc} . \text{caller} ! p1 \longrightarrow \\ \text{setEmergencyOxygenRet} . \text{loc} . \text{caller} \longrightarrow \\ \text{binder\_setEmergencyOxygenRet} . \text{loc} . \text{caller} \longrightarrow \\ \text{setEmergencyOxygen\_MethodBinder} \end{array} \right)$$

$$\text{setAirSpeed\_MethodBinder} \hat{=} \left( \begin{array}{l} \text{binder\_setAirSpeedCall} ? \text{loc} : (\text{loc} \in \text{setAirSpeedLocs}) ? \text{caller} : (\text{caller} \in \text{setAirSpeedCallers}) ? p1 \longrightarrow \\ \text{setAirSpeedCall} . \text{loc} . \text{caller} ! p1 \longrightarrow \\ \text{setAirSpeedRet} . \text{loc} . \text{caller} \longrightarrow \\ \text{binder\_setAirSpeedRet} . \text{loc} . \text{caller} \longrightarrow \\ \text{setAirSpeed\_MethodBinder} \end{array} \right)$$

$$\text{isLandingGearDeployed\_MethodBinder} \hat{=} \left( \begin{array}{l} \text{binder\_isLandingGearDeployedCall} ? \text{loc} : (\text{loc} \in \text{isLandingGearDeployedLocs}) ? \text{caller} : (\text{caller} \in \text{isLandingGearDeployedCallers}) \longrightarrow \\ \text{isLandingGearDeployedCall} . \text{loc} . \text{caller} \longrightarrow \\ \text{isLandingGearDeployedRet} . \text{loc} . \text{caller} ? \text{ret} \longrightarrow \\ \text{binder\_isLandingGearDeployedRet} . \text{loc} . \text{caller} ! \text{ret} \longrightarrow \\ \text{isLandingGearDeployed\_MethodBinder} \end{array} \right)$$

$$\begin{array}{l}
\text{BinderActions} \triangleq \\
\left( \begin{array}{l}
\text{setCabinPressure\_MethodBinder} \\
||| \\
\text{setFuelRemaining\_MethodBinder} \\
||| \\
\text{getAltitude\_MethodBinder} \\
||| \\
\text{setHeading\_MethodBinder} \\
||| \\
\text{stowLandingGear\_MethodBinder} \\
||| \\
\text{takeOffAbort\_MethodBinder} \\
||| \\
\text{setAltitude\_MethodBinder} \\
||| \\
\text{getHeading\_MethodBinder} \\
||| \\
\text{getAirSpeed\_MethodBinder} \\
||| \\
\text{deployLandingGear\_MethodBinder} \\
||| \\
\text{setEmergencyOxygen\_MethodBinder} \\
||| \\
\text{setAirSpeed\_MethodBinder} \\
||| \\
\text{isLandingGearDeployed\_MethodBinder}
\end{array} \right)
\end{array}$$

- $\text{BinderActions} \triangleq (\text{done\_toplevel\_sequencer} \longrightarrow \mathbf{Skip})$

**end**

## 2.3 Locking

**section** *NetworkLocking* **parents** *scj\_prelude, GlobalTypes, FrameworkChan, MissionId, MissionIds, ThreadIds, NetworkChannels, ObjectFW, ThreadFW, Priority*

**process** *Threads*  $\hat{=}$   
(**Skip**)

**process** *Objects*  $\hat{=}$   
(**Skip**)

**process** *Locking*  $\hat{=}$  (*Threads*  $\llbracket$  *ThreadSync*  $\rrbracket$  *Objects*)  $\triangle$  (*done\_toplevel\_sequencer*  $\longrightarrow$  **Skip**)

## 2.4 Program

**section** *Program* **parents** *scj\_prelude, MissionId, MissionIds, SchedulableId, SchedulableIds, MissionChan, SchedulableMethChan, MissionFW, SafeletFW, TopLevelMissionSequencerFW, NetworkChannels, ManagedThreadFW, SchedulableMissionSequencerFW, PeriodicEventHandlerFW, OneShotEventHandlerFW, AperiodicEventHandlerFW, ObjectFW, ThreadFW, ACSafeletApp, MainMissionSequencerApp, MainMissionApp, AModeChanger2App, ControlHandlerApp, CommunicationsHandlerApp, EnvironmentMonitorApp, FlightSensorsMonitorApp, TakeOffMissionApp, LandingGearHandlerApp, TakeOffFailureHandlerApp, TakeOffMonitorApp, CruiseMissionApp, BeginLandingHandlerApp, NavigationMonitorApp, LandMissionApp, LandingGearHandlerLandApp, SafeLandingHandlerApp, GroundDistanceMonitorApp, InstrumentLandingSystemMonitorApp*

**process** *ControlTier*  $\hat{=}$   

$$\left( \begin{array}{l} \text{SafeletFW} \\ \llbracket \text{ControlTierSync} \rrbracket \\ \text{TopLevelMissionSequencerFW}(\text{MainMissionSequencer}) \end{array} \right)$$

**process** *Tier0*  $\hat{=}$   

$$\left( \begin{array}{l} \text{MissionFW}(\text{MainMissionID}) \\ \llbracket \text{MissionSync} \rrbracket \\ \left( \begin{array}{l} \text{SchedulableMissionSequencerFW}(\text{AModeChanger2ID}) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \text{AperiodicEventHandlerFW}(\text{ControlHandlerID}, \text{aperiodic}, (\text{time}(10, 0), \text{nullSchedulableId})) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \text{AperiodicEventHandlerFW}(\text{CommunicationsHandlerID}, \text{aperiodic}, (\text{NULL}, \text{nullSchedulableId})) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \text{PeriodicEventHandlerFW}(\text{EnvironmentMonitorID}, (\text{time}(10, 0), \text{NULL}, \text{NULL}, \text{nullSchedulableId})) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \text{PeriodicEventHandlerFW}(\text{FlightSensorsMonitorID}, (\text{time}(10, 0), \text{NULL}, \text{NULL}, \text{nullSchedulableId})) \end{array} \right) \end{array} \right)$$

**process** *Tier1*  $\hat{=}$   

$$\left( \begin{array}{l} \text{MissionFW}(\text{TakeOffMissionID}) \\ \llbracket \text{MissionSync} \rrbracket \\ \left( \begin{array}{l} \text{AperiodicEventHandlerFW}(\text{LandingGearHandlerID}, \text{aperiodic}, (\text{NULL}, \text{nullSchedulableId})) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \text{AperiodicEventHandlerFW}(\text{TakeOffFailureHandlerID}, \text{aperiodic}, (\text{NULL}, \text{nullSchedulableId})) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \text{PeriodicEventHandlerFW}(\text{TakeOffMonitorID}, (\text{time}(0, 0), \text{time}(500, 0), \text{NULL}, \text{nullSchedulableId})) \end{array} \right) \\ \llbracket \text{ClusterSync} \rrbracket \\ \left( \begin{array}{l} \text{MissionFW}(\text{CruiseMissionID}) \\ \llbracket \text{MissionSync} \rrbracket \\ \left( \begin{array}{l} \text{AperiodicEventHandlerFW}(\text{BeginLandingHandlerID}, \text{aperiodic}, (\text{NULL}, \text{nullSchedulableId})) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \text{PeriodicEventHandlerFW}(\text{NavigationMonitorID}, (\text{time}(0, 0), \text{time}(10, 0), \text{NULL}, \text{nullSchedulableId})) \end{array} \right) \\ \llbracket \text{ClusterSync} \rrbracket \\ \left( \begin{array}{l} \text{MissionFW}(\text{LandMissionID}) \\ \llbracket \text{MissionSync} \rrbracket \\ \left( \begin{array}{l} \text{AperiodicEventHandlerFW}(\text{LandingGearHandlerLandID}, \text{aperiodic}, (\text{NULL}, \text{nullSchedulableId})) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \text{AperiodicEventHandlerFW}(\text{SafeLandingHandlerID}, \text{aperiodic}, (\text{NULL}, \text{nullSchedulableId})) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \text{PeriodicEventHandlerFW}(\text{GroundDistanceMonitorID}, (\text{time}(0, 0), \text{time}(10, 0), \text{NULL}, \text{nullSchedulableId})) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \text{PeriodicEventHandlerFW}(\text{InstrumentLandingSystemMonitorID}, (\text{time}(0, 0), \text{time}(10, 0), \text{NULL}, \text{nullSchedulableId})) \end{array} \right) \end{array} \right) \end{array} \right)$$

$$\text{process Framework} \hat{=} \left( \begin{array}{c} \text{ControlTier} \\ \llbracket \text{TierSync} \rrbracket \\ \left( \begin{array}{c} \text{Tier0} \\ \llbracket \text{Tier0Sync} \rrbracket \end{array} \right) \\ \text{Tier1} \end{array} \right)$$

$$\text{process Application} \hat{=} \left( \begin{array}{l} \text{ACSafeletApp} \\ ||| \\ \text{MainMissionSequencerApp} \\ ||| \\ \text{MainMissionApp} \\ ||| \\ \text{ACModeChanger2App}(\text{MainMissionID}) \\ ||| \\ \text{ControlHandlerApp} \\ ||| \\ \text{CommunicationsHandlerApp} \\ ||| \\ \text{EnvironmentMonitorApp}(\text{MainMissionID}) \\ ||| \\ \text{FlightSensorsMonitorApp}(\text{MainMissionID}) \\ ||| \\ \text{TakeOffMissionApp} \\ ||| \\ \text{LandingGearHandlerApp}(\text{TakeOffMissionID}) \\ ||| \\ \text{TakeOffFailureHandlerApp}(\text{TakeOffMissionID}, 10.0) \\ ||| \\ \text{TakeOffMonitorApp}(\text{TakeOffMissionID}, 10.0, \text{landingGearHandlerID}) \\ ||| \\ \text{CruiseMissionApp} \\ ||| \\ \text{BeginLandingHandlerApp} \\ ||| \\ \text{NavigationMonitorApp} \\ ||| \\ \text{LandMissionApp} \\ ||| \\ \text{LandingGearHandlerLandApp}(\text{LandMissionID}) \\ ||| \\ \text{SafeLandingHandlerApp}(10.0) \\ ||| \\ \text{GroundDistanceMonitorApp}(0.0) \\ ||| \\ \text{InstrumentLandingSystemMonitorApp}(\text{LandMissionID}) \end{array} \right)$$

**process** *Bound\_Application*  $\hat{=}$  *Application*  $\llbracket \text{MethodCallBinderSync} \rrbracket \text{MethodCallBinder}$   
**process** *Program*  $\hat{=}$  (*Framework*  $\llbracket \text{AppSync} \rrbracket \text{Bound\_Application}$ )  $\llbracket \text{LockingSync} \rrbracket \text{Locking}$

### 3 Safelet

**section** *ACSafeletApp* **parents** *scj\_prelude, SchedulableId, SchedulableIds, SafeletChan, MethodCallBindingChannels*

**process** *ACSafeletApp*  $\hat{=}$  **begin**

*InitializeApplication*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{initializeApplicationCall} \longrightarrow \\ \textit{initializeApplicationRet} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

*GetSequencer*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{getSequencerCall} \longrightarrow \\ \textit{getSequencerRet} ! \textit{MainMissionSequencerSID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

*Methods*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{GetSequencer} \\ \square \\ \textit{InitializeApplication} \end{array} \right); \textit{Methods}$

•  $(\textit{Methods}) \triangle (\textit{end\_safelet\_app} \longrightarrow \mathbf{Skip})$

**end**

## 4 Top Level Mission Sequencer

**section** *MainMissionSequencerApp* **parents** *TopLevelMissionSequencerChan*,  
*MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *MainMissionSequencerClass*, *MethodCallBindingChannels*

**process** *MainMissionSequencerApp*  $\hat{=}$  **begin**

<i>State</i> <i>this</i> : <b>ref</b> <i>MainMissionSequencerClass</i>
---

**state** *State*

<i>Init</i> <i>State</i> '
<i>this</i> ' = <b>new</b> <i>MainMissionSequencerClass</i> ()

*GetNextMission*  $\hat{=}$  **var** *ret* : *MissionID* •  
 $\left( \begin{array}{l} \textit{getNextMissionCall} . \textit{MainMissionSequencerSID} \longrightarrow \\ \textit{ret} := \textit{this} . \textit{getNextMission}(); \\ \textit{getNextMissionRet} . \textit{MainMissionSequencerSID} ! \textit{ret} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

*Methods*  $\hat{=}$   
 $(\textit{GetNextMission}) ; \textit{Methods}$

•  $(\textit{Init} ; \textit{Methods}) \triangle (\textit{end\_sequencer\_app} . \textit{MainMissionSequencerSID} \longrightarrow \mathbf{Skip})$

**end**



**section** *MainMissionSequencerClass* **parents** *scj\_prelude*, *SchedulableId*, *SchedulableIds*, *SafeletChan*, *MethodCallBindingChannels*, *MissionId*, *MissionIds*

**class** *MainMissionSequencerClass*  $\hat{=}$  **begin**

<b>state</b> <i>State</i> <i>returnedMission</i> : $\mathbb{B}$
--

**state** *State*

<b>initial</b> <i>Init</i> <i>State</i> '
<i>returnedMission</i> ' = <b>False</b>

**protected** *getNextMission*  $\hat{=}$

$$\left( \begin{array}{l} \text{if } (\neg \text{returnedMission}) \longrightarrow \\ \quad \left( \begin{array}{l} \text{returnedMission} := \mathbf{True}; \\ \text{ret} := \text{MainMissionMID} \end{array} \right) \\ \parallel \neg (\neg \text{returnedMission}) \longrightarrow \\ \quad (\text{ret} := \text{nullMissionId}) \\ \text{fi} \end{array} \right)$$

• **Skip**

**end**

## 5 Missions

### 5.1 MainMission

**section** *MainMissionApp* **parents** *scj\_prelude*, *MissionId*, *MissionIds*,  
*SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *MainMissionMethChan*,  
*MainMissionClass*, *MethodCallBindingChannels*

**process** *MainMissionApp*  $\hat{=}$  **begin**

<i>State</i> <i>this</i> : <b>ref</b> <i>MainMissionClass</i>
--

**state** *State*

<i>Init</i> <i>State'</i>
<i>this'</i> = <b>new</b> <i>MainMissionClass</i> ()

*InitializePhase*  $\hat{=}$

$$\left( \begin{array}{l} \textit{initializeCall} . \textit{MainMissionMID} \longrightarrow \\ \textit{register} ! \textit{ACModeChanger2SID} ! \textit{MainMissionMID} \longrightarrow \\ \textit{register} ! \textit{EnvironmentMonitorSID} ! \textit{MainMissionMID} \longrightarrow \\ \textit{register} ! \textit{ControlHandlerSID} ! \textit{MainMissionMID} \longrightarrow \\ \textit{register} ! \textit{FlightSensorsMonitorSID} ! \textit{MainMissionMID} \longrightarrow \\ \textit{register} ! \textit{CommunicationsHandlerSID} ! \textit{MainMissionMID} \longrightarrow \\ \textit{initializeRet} . \textit{MainMissionMID} \longrightarrow \\ \textbf{Skip} \end{array} \right)$$

*CleanupPhase*  $\hat{=}$

$$\left( \begin{array}{l} \textit{cleanupMissionCall} . \textit{MainMissionMID} \longrightarrow \\ \textit{cleanupMissionRet} . \textit{MainMissionMID} ! \textbf{True} \longrightarrow \\ \textbf{Skip} \end{array} \right)$$

*getAirSpeedMeth*  $\hat{=}$  **var** *ret* :  $\mathbb{P} \mathbb{A} \bullet$

$$\left( \begin{array}{l} \textit{getAirSpeedCall} . \textit{MainMissionMID} ? \textit{caller} \longrightarrow \\ \textit{ret} := \textit{this} . \textit{getAirSpeed}(); \\ \textit{getAirSpeedRet} . \textit{MainMissionMID} . \textit{caller} ! \textit{ret} \longrightarrow \\ \textbf{Skip} \end{array} \right)$$

*getAltitudeMeth*  $\hat{=}$  **var** *ret* :  $\mathbb{P} \mathbb{A} \bullet$

$$\left( \begin{array}{l} \textit{getAltitudeCall} . \textit{MainMissionMID} ? \textit{caller} \longrightarrow \\ \textit{ret} := \textit{this} . \textit{getAltitude}(); \\ \textit{getAltitudeRet} . \textit{MainMissionMID} . \textit{caller} ! \textit{ret} \longrightarrow \\ \textbf{Skip} \end{array} \right)$$

*getCabinPressureMeth*  $\hat{=}$  **var** *ret* :  $\mathbb{P} \mathbb{A} \bullet$

$$\left( \begin{array}{l} \textit{getCabinPressureCall} . \textit{MainMissionMID} \longrightarrow \\ \textit{ret} := \textit{this} . \textit{getCabinPressure}(); \\ \textit{getCabinPressureRet} . \textit{MainMissionMID} ! \textit{ret} \longrightarrow \\ \textbf{Skip} \end{array} \right)$$

$$\text{getEmergencyOxygenMeth} \hat{=} \mathbf{var} \text{ ret} : \mathbb{P} \mathbb{A} \bullet \left( \begin{array}{l} \text{getEmergencyOxygenCall} . \text{MainMissionMID} \longrightarrow \\ \text{ret} := \text{this} . \text{getEmergencyOxygen}(); \\ \text{getEmergencyOxygenRet} . \text{MainMissionMID} ! \text{ret} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

$$\text{getFuelRemainingMeth} \hat{=} \mathbf{var} \text{ ret} : \mathbb{P} \mathbb{A} \bullet \left( \begin{array}{l} \text{getFuelRemainingCall} . \text{MainMissionMID} \longrightarrow \\ \text{ret} := \text{this} . \text{getFuelRemaining}(); \\ \text{getFuelRemainingRet} . \text{MainMissionMID} ! \text{ret} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

$$\text{getHeadingMeth} \hat{=} \mathbf{var} \text{ ret} : \mathbb{P} \mathbb{A} \bullet \left( \begin{array}{l} \text{getHeadingCall} . \text{MainMissionMID} ? \text{caller} \longrightarrow \\ \text{ret} := \text{this} . \text{getHeading}(); \\ \text{getHeadingRet} . \text{MainMissionMID} . \text{caller} ! \text{ret} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

$$\text{setAirSpeedMeth} \hat{=} \left( \begin{array}{l} \text{setAirSpeedCall} . \text{MainMissionMID} ? \text{caller} ? \text{newAirSpeed} \longrightarrow \\ \text{this} . \text{setAirSpeed}(\text{newAirSpeed}); \\ \text{setAirSpeedRet} . \text{MainMissionMID} . \text{caller} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

$$\text{setAltitudeMeth} \hat{=} \left( \begin{array}{l} \text{setAltitudeCall} . \text{MainMissionMID} ? \text{caller} ? \text{newAltitude} \longrightarrow \\ \text{this} . \text{setAltitude}(\text{newAltitude}); \\ \text{setAltitudeRet} . \text{MainMissionMID} . \text{caller} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

$$\text{setCabinPressureMeth} \hat{=} \left( \begin{array}{l} \text{setCabinPressureCall} . \text{MainMissionMID} ? \text{caller} ? \text{newCabinPressure} \longrightarrow \\ \text{this} . \text{setCabinPressure}(\text{newCabinPressure}); \\ \text{setCabinPressureRet} . \text{MainMissionMID} . \text{caller} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

$$\text{setEmergencyOxygenMeth} \hat{=} \left( \begin{array}{l} \text{setEmergencyOxygenCall} . \text{MainMissionMID} ? \text{caller} ? \text{newEmergencyOxygen} \longrightarrow \\ \text{this} . \text{setEmergencyOxygen}(\text{newEmergencyOxygen}); \\ \text{setEmergencyOxygenRet} . \text{MainMissionMID} . \text{caller} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

$$\text{setFuelRemainingMeth} \hat{=} \left( \begin{array}{l} \text{setFuelRemainingCall} . \text{MainMissionMID} ? \text{caller} ? \text{newFuelRemaining} \longrightarrow \\ \text{this} . \text{setFuelRemaining}(\text{newFuelRemaining}); \\ \text{setFuelRemainingRet} . \text{MainMissionMID} . \text{caller} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

$$\text{setHeadingMeth} \hat{=} \left( \begin{array}{l} \text{setHeadingCall} . \text{MainMissionMID} ? \text{caller} ? \text{newHeading} \longrightarrow \\ \text{this} . \text{setHeading}(\text{newHeading}); \\ \text{setHeadingRet} . \text{MainMissionMID} . \text{caller} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

$$Methods \hat{=} \left( \begin{array}{l} InitializePhase \\ \square \\ CleanupPhase \\ \square \\ getAirSpeedMeth \\ \square \\ getAltitudeMeth \\ \square \\ getCabinPressureMeth \\ \square \\ getEmergencyOxygenMeth \\ \square \\ getFuelRemainingMeth \\ \square \\ getHeadingMeth \\ \square \\ setAirSpeedMeth \\ \square \\ setAltitudeMeth \\ \square \\ setCabinPressureMeth \\ \square \\ setEmergencyOxygenMeth \\ \square \\ setFuelRemainingMeth \\ \square \\ setHeadingMeth \end{array} \right) ; Methods$$

- $(Init ; Methods) \triangle (end\_mission\_app . MainMissionMID \longrightarrow \mathbf{Skip})$

**end**

**section** *MainMissionClass* **parents** *scj\_prelude*, *SchedulableId*, *SchedulableIds*, *SafeletChan*, *MethodCallBindingChannels*

**class** *MainMissionClass*  $\hat{=}$  **begin**

**state** *State*

---

*cabinPressure* :  $\mathbb{P} \mathbb{A}$   
*emergencyOxygen* :  $\mathbb{P} \mathbb{A}$   
*fuelRemaining* :  $\mathbb{P} \mathbb{A}$   
*altitude* :  $\mathbb{P} \mathbb{A}$   
*airSpeed* :  $\mathbb{P} \mathbb{A}$   
*heading* :  $\mathbb{P} \mathbb{A}$

---

**state** *State*

**initial** *Init*

---

*State*'

---

**public** *getAirSpeed*  $\hat{=}$   
(*ret* := *airSpeed*)

**public** *getAltitude*  $\hat{=}$   
(*ret* := *altitude*)

**public** *getCabinPressure*  $\hat{=}$   
(*ret* := *cabinPressure*)

**public** *getEmergencyOxygen*  $\hat{=}$   
(*ret* := *emergencyOxygen*)

**public** *getFuelRemaining*  $\hat{=}$   
(*ret* := *fuelRemaining*)

**public** *getHeading*  $\hat{=}$   
(*ret* := *heading*)

**public** *setAirSpeed*  $\hat{=}$  **var** *newAirSpeed* :  $\mathbb{P} \mathbb{A}$  •  
(*airSpeed* := *newAirSpeed*)

**public** *setAltitude*  $\hat{=}$  **var** *newAltitude* :  $\mathbb{P} \mathbb{A}$  •  
(*altitude* := *newAltitude*)

**public** *setCabinPressure*  $\hat{=}$  **var** *newCabinPressure* :  $\mathbb{P} \mathbb{A}$  •  
(*cabinPressure* := *newCabinPressure*)

**public** *setEmergencyOxygen*  $\hat{=}$  **var** *newEmergencyOxygen* :  $\mathbb{P} \mathbb{A}$  •  
(*emergencyOxygen* := *newEmergencyOxygen*)

**public** *setFuelRemaining*  $\hat{=}$  **var** *newFuelRemaining* :  $\mathbb{P}\mathbb{A}$  •  
(*fuelRemaining* := *newFuelRemaining*)

**public** *setHeading*  $\hat{=}$  **var** *newHeading* :  $\mathbb{P}\mathbb{A}$  •  
(*heading* := *newHeading*)

• Skip

**end**

## 5.2 Schedulables of MainMission

**section** *ACModeChanger2App* **parents** *TopLevelMissionSequencerChan*,  
*MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *ACModeChanger2Class*, *MethodCallBindingChannels*

**process** *ACModeChanger2App*  $\hat{=}$   
*controllingMission* : *MissionID* • **begin**

*GetNextMission*  $\hat{=}$  **var** *ret* : *MissionID* •  
 $\left( \begin{array}{l} \textit{getNextMissionCall} . \textit{ACModeChanger2SID} \longrightarrow \\ \textit{ret} := \textit{this} . \textit{getNextMission}(); \\ \textit{getNextMissionRet} . \textit{ACModeChanger2SID} ! \textit{ret} \longrightarrow \\ \textbf{Skip} \end{array} \right)$

*Methods*  $\hat{=}$   
 $(\textit{GetNextMission}) ; \textit{Methods}$

•  $(\textit{Methods}) \triangle (\textit{end\_sequencer\_app} . \textit{ACModeChanger2SID} \longrightarrow \textbf{Skip})$

**end**

**section** *ACModeChanger2Class* **parents** *scj\_prelude, SchedulableId, SchedulableIds, SafeletChan*  
*, MethodCallBindingChannels, MissionId, MissionIds*

**class** *ACModeChanger2Class*  $\hat{=}$  **begin**

<b>state</b> <i>State</i> <i>modesLeft</i> : $\mathbb{Z}$
--

**state** *State*

<b>initial</b> <i>Init</i> <i>State</i> ' <i>modesLeft</i> ' = 3
--

**protected** *getNextMission*  $\hat{=}$

$$\left( \begin{array}{l} \text{if } (modesLeft = 3) \longrightarrow \\ \quad \left( \begin{array}{l} modesLeft := modesLeft - 1; \\ ret := TakeOffMissionMID \end{array} \right) \\ \parallel \neg (modesLeft = 3) \longrightarrow \\ \quad \text{if } (modesLeft = 2) \longrightarrow \\ \quad \quad \left( \begin{array}{l} modesLeft := modesLeft - 1; \\ ret := CruiseMissionMID \end{array} \right) \\ \parallel \neg (modesLeft = 2) \longrightarrow \\ \quad \text{if } (modesLeft = 1) \longrightarrow \\ \quad \quad \left( \begin{array}{l} modesLeft := modesLeft - 1; \\ ret := LandMissionMID \end{array} \right) \\ \parallel \neg (modesLeft = 1) \longrightarrow \\ \quad (ret := nullMissionId) \\ \text{fi} \\ \text{fi} \\ \text{fi} \end{array} \right)$$

• Skip

**end**



**section** *ControlHandlerApp* **parents** *AperiodicEventHandlerChan, SchedulableId, SchedulableIds, MethodCallBindingCh*

**process** *ControlHandlerApp*  $\hat{=}$  **begin**

*handleAsyncEvent*  $\hat{=}$   

$$\left( \begin{array}{l} \text{handleAsyncEventCall} . \text{ControlHandlerSID} \longrightarrow \\ \mathbf{Skip}; \\ \text{handleAsyncEventRet} . \text{ControlHandlerSID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*Methods*  $\hat{=}$   
 $(\text{handleAsyncEvent}) ; \text{Methods}$

$\bullet (\text{Methods}) \triangle (\text{end\_aperiodic\_app} . \text{ControlHandlerSID} \longrightarrow \mathbf{Skip})$

**end**

**section** *CommunicationsHandlerApp* **parents** *AperiodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*, *MethodCallB*

**process** *CommunicationsHandlerApp*  $\hat{=}$  **begin**

*handleAsyncEvent*  $\hat{=}$   

$$\left( \begin{array}{l} \textit{handleAsyncEventCall} . \textit{CommunicationsHandlerSID} \longrightarrow \\ \mathbf{Skip}; \\ \textit{handleAsyncEventRet} . \textit{CommunicationsHandlerSID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*Methods*  $\hat{=}$   
 $(\textit{handleAsyncEvent}) ; \textit{Methods}$

$\bullet (\textit{Methods}) \triangle (\textit{end\_aperiodic\_app} . \textit{CommunicationsHandlerSID} \longrightarrow \mathbf{Skip})$

**end**

**section** *EnvironmentMonitorApp* **parents** *PeriodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBinding*, *MainMissionMethChan*

**process** *EnvironmentMonitorApp*  $\hat{=}$   
*controllingMission* : *MissionID* • **begin**

*handleAsyncEvent*  $\hat{=}$   

$$\left( \begin{array}{l} \text{handleAsyncEventCall} . \text{EnvironmentMonitorSID} \longrightarrow \\ \left( \begin{array}{l} \text{binder\_setCabinPressureCall} . \text{controllingMission} . \text{EnvironmentMonitorSID} ! 0 \longrightarrow \\ \text{binder\_setCabinPressureRet} . \text{controllingMission} . \text{EnvironmentMonitorSID} \longrightarrow \\ \mathbf{Skip}; \\ \text{binder\_setEmergencyOxygenCall} . \text{controllingMission} . \text{EnvironmentMonitorSID} ! 0 \longrightarrow \\ \text{binder\_setEmergencyOxygenRet} . \text{controllingMission} . \text{EnvironmentMonitorSID} \longrightarrow \\ \mathbf{Skip}; \\ \text{binder\_setFuelRemainingCall} . \text{controllingMission} . \text{EnvironmentMonitorSID} ! 0 \longrightarrow \\ \text{binder\_setFuelRemainingRet} . \text{controllingMission} . \text{EnvironmentMonitorSID} \longrightarrow \\ \mathbf{Skip} \end{array} \right) ; \\ \text{handleAsyncEventRet} . \text{EnvironmentMonitorSID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*Methods*  $\hat{=}$   
(*handleAsyncEvent*) ; *Methods*

• (*Methods*)  $\triangle$  (*end\\_periodic\\_app* . *EnvironmentMonitorSID*  $\longrightarrow$  **Skip**)

**end**

**section** *FlightSensorsMonitorApp* **parents** *PeriodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBinding*, *MainMissionMethChan*

**process** *FlightSensorsMonitorApp*  $\hat{=}$   
*mainMission* : *MissionID* • **begin**

<i>State</i> <i>controllingMission</i> : <i>MainMission</i>
--

**state** *State*

<i>Init</i> <i>State'</i>
<i>controllingMission'</i> =

*handleAsyncEvent*  $\hat{=}$

$$\left( \begin{array}{l} \text{handleAsyncEventCall} . \text{FlightSensorsMonitorSID} \longrightarrow \\ \left( \begin{array}{l} \text{binder\_setAirSpeedCall} . \text{controllingMission} . \text{FlightSensorsMonitorSID} ! 0 \longrightarrow \\ \text{binder\_setAirSpeedRet} . \text{controllingMission} . \text{FlightSensorsMonitorSID} \longrightarrow \\ \mathbf{Skip}; \\ \text{binder\_setAltitudeCall} . \text{controllingMission} . \text{FlightSensorsMonitorSID} ! 0 \longrightarrow \\ \text{binder\_setAltitudeRet} . \text{controllingMission} . \text{FlightSensorsMonitorSID} \longrightarrow \\ \mathbf{Skip}; \\ \text{binder\_setHeadingCall} . \text{controllingMission} . \text{FlightSensorsMonitorSID} ! 0 \longrightarrow \\ \text{binder\_setHeadingRet} . \text{controllingMission} . \text{FlightSensorsMonitorSID} \longrightarrow \\ \mathbf{Skip} \end{array} \right) ; \\ \text{handleAsyncEventRet} . \text{FlightSensorsMonitorSID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*Methods*  $\hat{=}$   
(*handleAsyncEvent*) ; *Methods*

• (*Init* ; *Methods*)  $\triangle$  (*end\\_periodic\\_app* . *FlightSensorsMonitorSID*  $\longrightarrow$  **Skip**)

**end**

### 5.3 TakeOffMission

**section** *TakeOffMissionApp* **parents** *scj\_prelude, MissionId, MissionIds, SchedulableId, SchedulableIds, MissionChan, SchedulableMethChan, TakeOffMissionMethChan, TakeOffMissionClass, MethodCallBindingChannels*

**process** *TakeOffMissionApp*  $\hat{=}$   
*controllingMission* : *MissionID* • **begin**

---

*State*  
*this* : **ref** *TakeOffMissionClass*

---

**state** *State*

---

*Init*  
*State'*  


---

*this'* = **new** *TakeOffMissionClass*()

---

*InitializePhase*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{initializeCall} . \textit{TakeOffMissionMID} \longrightarrow \\ \textit{register! LandingGearHandlerSID! TakeOffMissionMID} \longrightarrow \\ \textit{register! TakeOffMonitorSID! TakeOffMissionMID} \longrightarrow \\ \textit{register! TakeOffFailureHandlerSID! TakeOffMissionMID} \longrightarrow \\ \textit{initializeRet} . \textit{TakeOffMissionMID} \longrightarrow \\ \textbf{Skip} \end{array} \right)$

*CleanupPhase*  $\hat{=}$  **var**  $\mathbb{B}$  : *ret* •  
 $\left( \begin{array}{l} \textit{cleanupMissionCall} . \textit{TakeOffMissionMID} \longrightarrow \\ (\textit{ret} := (\neg \textit{this} . \textit{abort})) \\ \textit{cleanupMissionRet} . \textit{TakeOffMissionMID! ret} \longrightarrow \\ \textbf{Skip} \end{array} \right)$

*takeOffAbortMeth*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{takeOffAbortCall} . \textit{TakeOffMissionMID? caller} \longrightarrow \\ \textit{this} . \textit{takeOffAbort}(); \\ \textit{takeOffAbortRet} . \textit{TakeOffMissionMID} . \textit{caller} \longrightarrow \\ \textbf{Skip} \end{array} \right)$

*deployLandingGearMeth*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{deployLandingGearCall} . \textit{TakeOffMissionMID? caller} \longrightarrow \\ \textit{this} . \textit{deployLandingGear}(); \\ \textit{deployLandingGearRet} . \textit{TakeOffMissionMID} . \textit{caller} \longrightarrow \\ \textbf{Skip} \end{array} \right)$

*stowLandingGearMeth*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{stowLandingGearCall} . \textit{TakeOffMissionMID? caller} \longrightarrow \\ \textit{this} . \textit{stowLandingGear}(); \\ \textit{stowLandingGearRet} . \textit{TakeOffMissionMID} . \textit{caller} \longrightarrow \\ \textbf{Skip} \end{array} \right)$

$isLandingGearDeployedMeth \hat{=} \mathbf{var} \ ret : \mathbb{B} \bullet$   
 $\left( \begin{array}{l} isLandingGearDeployedCall . TakeOffMissionMID ? caller \longrightarrow \\ ret := this . isLandingGearDeployed(); \\ isLandingGearDeployedRet . TakeOffMissionMID . caller ! ret \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

$Methods \hat{=} \left( \begin{array}{l} InitializePhase \\ \square \\ CleanupPhase \\ \square \\ takeOffAbortMeth \\ \square \\ deployLandingGearMeth \\ \square \\ stowLandingGearMeth \\ \square \\ isLandingGearDeployedMeth \end{array} \right) ; Methods$

$\bullet (Init ; Methods) \triangle (end\_mission\_app . TakeOffMissionMID \longrightarrow \mathbf{Skip})$

**end**

**section** *TakeOffMissionClass* **parents** *scj\_prelude*, *SchedulableId*, *SchedulableIds*, *SafeletChan*, *MethodCallBindingChannels*

**class** *TakeOffMissionClass*  $\hat{=}$  **begin**

**state** *State*

---

*SAFE\_AIRSPEED\_THRESHOLD* :  $\mathbb{P} \mathbb{A}$   
*TAKEOFF\_ALTITUDE* :  $\mathbb{P} \mathbb{A}$   
*abort* :  $\mathbb{B}$   
*landingGearDeployed* :  $\mathbb{B}$

---

**state** *State*

**initial** *Init*

---

*State'*  


---

*SAFE\_AIRSPEED\_THRESHOLD'* = 10.0  
*TAKEOFF\_ALTITUDE'* = 10.0  
*abort'* = *false*

---

**public** *takeOffAbort*  $\hat{=}$   
(*abort* := **True**)

**public** *deployLandingGear*  $\hat{=}$   
(*landingGearDeployed* := **True**)

**public** *stowLandingGear*  $\hat{=}$   
(*landingGearDeployed* := **False**)

**public** *isLandingGearDeployed*  $\hat{=}$   
(*ret* := *landingGearDeployed*)

• **Skip**

**end**

## 5.4 Schedulables of TakeOffMission

**section** *LandingGearHandlerApp* **parents** *AperiodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBinder*, *TakeOffMissionMethChan*

**process** *LandingGearHandlerApp*  $\hat{=}$   
*mission* : *MissionID* • **begin**

*handleAsyncEvent*  $\hat{=}$   

$$\left( \begin{array}{l} \text{handleAsyncEventCall} . \text{LandingGearHandlerSID} \longrightarrow \\ \left( \begin{array}{l} \text{binder\_isLandingGearDeployedCall} . \text{mission} . \text{LandingGearHandlerSID} \longrightarrow \\ \text{binder\_isLandingGearDeployedRet} . \text{mission} . \text{LandingGearHandlerSID} ? \text{isLandingGearDeployed} \longrightarrow \\ \mathbf{Skip} ; \mathbf{var} \text{landingGearIsDeployed} : \mathbb{B} \bullet \text{landingGearIsDeployed} := \text{isLandingGearDeployed}; \\ \mathbf{if} \text{landingGearIsDeployed} \longrightarrow \\ \left( \begin{array}{l} \text{binder\_stowLandingGearCall} . \text{mission} . \text{LandingGearHandlerSID} \longrightarrow \\ \text{binder\_stowLandingGearRet} . \text{mission} . \text{LandingGearHandlerSID} \longrightarrow \\ \mathbf{Skip} \end{array} \right) \\ \parallel \neg \text{landingGearIsDeployed} \longrightarrow \\ \left( \begin{array}{l} \text{binder\_deployLandingGearCall} . \text{mission} . \text{LandingGearHandlerSID} \longrightarrow \\ \text{binder\_deployLandingGearRet} . \text{mission} . \text{LandingGearHandlerSID} \longrightarrow \\ \mathbf{Skip} \end{array} \right) \\ \mathbf{fi} \end{array} \right) ; \end{array} \right)$$

*Methods*  $\hat{=}$   
 $(\text{handleAsyncEvent}) ; \text{Methods}$

•  $(\text{Methods}) \triangle (\text{end\_aperiodic\_app} . \text{LandingGearHandlerSID} \longrightarrow \mathbf{Skip})$

**end**



**section** *TakeOffFailureHandlerApp* **parents** *AperiodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBin*  
*, MainMissionMethChan*, *TakeOffMissionMethChan*

**process** *TakeOffFailureHandlerApp*  $\hat{=}$   
*mainMission* : *MissionID*,  
*takeoffMission* : *MissionID*,  
*threshold* :  $\mathbb{P} \mathbb{A}$  • **begin**

*handleAsyncEvent*  $\hat{=}$   

$$\left( \begin{array}{l} \text{handleAsyncEventCall} . \text{TakeOffFailureHandlerSID} \longrightarrow \\ \left( \begin{array}{l} \text{binder\_getAirSpeedCall} . \text{mainMission} . \text{TakeOffFailureHandlerSID} \longrightarrow \\ \text{binder\_getAirSpeedRet} . \text{mainMission} . \text{TakeOffFailureHandlerSID} ? \text{getAirSpeed} \longrightarrow \\ \mathbf{Skip} ; \mathbf{var} \text{currentSpeed} : \mathbb{P} \mathbb{A} \bullet \text{currentSpeed} := \text{getAirSpeed} ; \\ \mathbf{if} (\text{currentSpeed} < \text{threshold}) \longrightarrow \\ \left( \begin{array}{l} \text{binder\_takeOffAbortCall} . \text{takeoffMission} . \text{TakeOffFailureHandlerSID} \longrightarrow \\ \text{binder\_takeOffAbortRet} . \text{takeoffMission} . \text{TakeOffFailureHandlerSID} \longrightarrow \\ \mathbf{Skip} ; \\ \text{requestTerminationCall} . \text{takeoffMission} . \text{TakeOffFailureHandlerSID} \longrightarrow \\ \text{requestTerminationRet} . \text{takeoffMission} . \text{TakeOffFailureHandlerSID} ? \text{requestTermination} \longrightarrow \\ \mathbf{Skip} \end{array} \right) ; \\ \mathbb{I} \neg (\text{currentSpeed} < \text{threshold}) \longrightarrow \\ \mathbf{Skip} \end{array} \right) \\ \mathbf{fi} \\ \text{handleAsyncEventRet} . \text{TakeOffFailureHandlerSID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*Methods*  $\hat{=}$   
(*handleAsyncEvent*) ; *Methods*

• (*Methods*)  $\triangle$  (*end\_aperiodic\_app* . *TakeOffFailureHandlerSID*  $\longrightarrow$  **Skip**)

**end**

**section** *TakeOffMonitorApp* **parents** *PeriodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBindingChan*, *MainMissionMethChan*

**process** *TakeOffMonitorApp*  $\hat{=}$   
*mainMission* : *MissionID*,  
*takeOffMission* : *MissionID*,  
*takeOffAltitude* :  $\mathbb{P} \mathbb{A}$ ,  
*landingGearHandler* : *SchedulableID* • **begin**

*handleAsyncEvent*  $\hat{=}$   

$$\left( \begin{array}{l} \text{handleAsyncEventCall} . \text{TakeOffMonitorSID} \longrightarrow \\ \left( \begin{array}{l} \text{binder\_getAltitudeCall} . \text{mainMission} . \text{TakeOffMonitorSID} \longrightarrow \\ \text{binder\_getAltitudeRet} . \text{mainMission} . \text{TakeOffMonitorSID} ? \text{getAltitude} \longrightarrow \\ \mathbf{Skip} ; \mathbf{var} \text{altitude} : \mathbb{P} \mathbb{A} \bullet \text{altitude} := \text{getAltitude}; \\ \mathbf{if} (\text{altitude} > \text{takeOffAltitude}) \longrightarrow \\ \left( \begin{array}{l} \text{release} . \text{landingGearHandler} \longrightarrow \\ \mathbf{Skip}; \\ \text{requestTerminationCall} . \text{takeOffMission} . \text{TakeOffMonitorSID} \longrightarrow \\ \text{requestTerminationRet} . \text{takeOffMission} . \text{TakeOffMonitorSID} ? \text{requestTermination} \longrightarrow \\ \mathbf{Skip} \end{array} \right) ; \\ \mathbb{I} \neg (\text{altitude} > \text{takeOffAltitude}) \longrightarrow \mathbf{Skip} \end{array} \right) \\ \mathbf{fi} \\ \text{handleAsyncEventRet} . \text{TakeOffMonitorSID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*Methods*  $\hat{=}$   
(*handleAsyncEvent*) ; *Methods*

• (*Methods*)  $\triangle$  (*end\\_periodic\\_app* . *TakeOffMonitorSID*  $\longrightarrow$  **Skip**)

**end**

## 5.5 CruiseMission

**section** *CruiseMissionApp* **parents** *scj\_prelude*, *MissionId*, *MissionIds*,  
*SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *CruiseMissionMethChan*  
*, MethodCallBindingChannels*

**process** *CruiseMissionApp*  $\hat{=}$   
*controllingMission* : *MissionID* • **begin**

*InitializePhase*  $\hat{=}$   

$$\left( \begin{array}{l} \textit{initializeCall} . \textit{CruiseMissionMID} \longrightarrow \\ \textit{register} ! \textit{BeginLandingHandlerSID} ! \textit{CruiseMissionMID} \longrightarrow \\ \textit{register} ! \textit{NavigationMonitorSID} ! \textit{CruiseMissionMID} \longrightarrow \\ \textit{initializeRet} . \textit{CruiseMissionMID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*CleanupPhase*  $\hat{=}$   

$$\left( \begin{array}{l} \textit{cleanupMissionCall} . \textit{CruiseMissionMID} \longrightarrow \\ \textit{cleanupMissionRet} . \textit{CruiseMissionMID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*Methods*  $\hat{=}$   $\left( \begin{array}{c} \textit{InitializePhase} \\ \square \\ \textit{CleanupPhase} \end{array} \right) ; \textit{Methods}$

• (*Methods*)  $\triangle$  (*end\_mission\_app* . *CruiseMissionMID*  $\longrightarrow$  **Skip**)

**end**

## 5.6 Schedulables of CruiseMission

**section** *BeginLandingHandlerApp* **parents** *AperiodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBind*

**process** *BeginLandingHandlerApp*  $\hat{=}$   
*controllingMission* : *MissionID* • **begin**

*handleAsyncEvent*  $\hat{=}$   

$$\left( \begin{array}{l} \textit{handleAsyncEventCall} . \textit{BeginLandingHandlerSID} \longrightarrow \\ \left( \begin{array}{l} \textit{requestTerminationCall} . \textit{controllingMission} . \textit{BeginLandingHandlerSID} \longrightarrow \\ \textit{requestTerminationRet} . \textit{controllingMission} . \textit{BeginLandingHandlerSID} ? \textit{requestTermination} \longrightarrow \end{array} \right) ; \\ \mathbf{Skip} \\ \textit{handleAsyncEventRet} . \textit{BeginLandingHandlerSID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*Methods*  $\hat{=}$   
(*handleAsyncEvent*) ; *Methods*

• (*Methods*)  $\triangle$  (*end\_aperiodic\_app* . *BeginLandingHandlerSID*  $\longrightarrow$  **Skip**)

**end**

**section** *NavigationMonitorApp* **parents** *PeriodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBinding*  
*, MainMissionMethChan*

**process** *NavigationMonitorApp*  $\hat{=}$   
*mainMission* : *MissionID* • **begin**

*handleAsyncEvent*  $\hat{=}$   

$$\left( \begin{array}{l} \text{handleAsyncEventCall} . \text{NavigationMonitorSID} \longrightarrow \\ \left( \begin{array}{l} \text{binder\_getHeadingCall} . \text{mainMission} . \text{NavigationMonitorSID} \longrightarrow \\ \text{binder\_getHeadingRet} . \text{mainMission} . \text{NavigationMonitorSID} ? \text{getHeading} \longrightarrow \\ \mathbf{Skip} ; \mathbf{var} \text{ heading} : \mathbb{P} \mathbb{A} \bullet \text{heading} := \text{getHeading}; \\ \text{binder\_getAirSpeedCall} . \text{mainMission} . \text{NavigationMonitorSID} \longrightarrow \\ \text{binder\_getAirSpeedRet} . \text{mainMission} . \text{NavigationMonitorSID} ? \text{getAirSpeed} \longrightarrow \\ \mathbf{Skip} ; \mathbf{var} \text{ airSpeed} : \mathbb{P} \mathbb{A} \bullet \text{airSpeed} := \text{getAirSpeed}; \\ \text{binder\_getAltitudeCall} . \text{mainMission} . \text{NavigationMonitorSID} \longrightarrow \\ \text{binder\_getAltitudeRet} . \text{mainMission} . \text{NavigationMonitorSID} ? \text{getAltitude} \longrightarrow \\ \mathbf{Skip} ; \mathbf{var} \text{ altitude} : \mathbb{P} \mathbb{A} \bullet \text{altitude} := \text{getAltitude} \end{array} \right) ; \\ \text{handleAsyncEventRet} . \text{NavigationMonitorSID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*Methods*  $\hat{=}$   
(*handleAsyncEvent*) ; *Methods*

• (*Methods*)  $\triangle$  (*end\\_periodic\\_app* . *NavigationMonitorSID*  $\longrightarrow$  **Skip**)

**end**

## 5.7 LandMission

**section** *LandMissionApp* **parents** *scj\_prelude, MissionId, MissionIds, SchedulableId, SchedulableIds, MissionChan, SchedulableMethChan, LandMissionMethChan, LandMissionClass, MethodCallBindingChannels*

**process** *LandMissionApp*  $\hat{=}$   
*controllingMission* : *MissionID* • **begin**

*InitializePhase*  $\hat{=}$   

$$\left( \begin{array}{l} \text{initializeCall} . \text{LandMissionMID} \longrightarrow \\ \text{register} ! \text{GroundDistanceMonitorSID} ! \text{LandMissionMID} \longrightarrow \\ \text{register} ! \text{LandingGearHandlerLandSID} ! \text{LandMissionMID} \longrightarrow \\ \text{register} ! \text{InstrumentLandingSystemMonitorSID} ! \text{LandMissionMID} \longrightarrow \\ \text{register} ! \text{SafeLandingHandlerSID} ! \text{LandMissionMID} \longrightarrow \\ \text{initializeRet} . \text{LandMissionMID} \longrightarrow \\ \text{Skip} \end{array} \right)$$

*CleanupPhase*  $\hat{=}$  **var**  $\mathbb{B}$  : *ret* •  

$$\left( \begin{array}{l} \text{cleanupMissionCall} . \text{LandMissionMID} \longrightarrow \\ (\text{ret} := \text{False}) \\ \text{cleanupMissionRet} . \text{LandMissionMID} ! \text{ret} \longrightarrow \\ \text{Skip} \end{array} \right)$$

*deployLandingGearMeth*  $\hat{=}$   

$$\left( \begin{array}{l} \text{deployLandingGearCall} . \text{LandMissionMID} ? \text{caller} \longrightarrow \\ \text{this} . \text{deployLandingGear}(); \\ \text{deployLandingGearRet} . \text{LandMissionMID} . \text{caller} \longrightarrow \\ \text{Skip} \end{array} \right)$$

*stowLandingGearMeth*  $\hat{=}$   

$$\left( \begin{array}{l} \text{stowLandingGearCall} . \text{LandMissionMID} ? \text{caller} \longrightarrow \\ \text{this} . \text{stowLandingGear}(); \\ \text{stowLandingGearRet} . \text{LandMissionMID} . \text{caller} \longrightarrow \\ \text{Skip} \end{array} \right)$$

*isLandingGearDeployedMeth*  $\hat{=}$  **var** *ret* :  $\mathbb{B}$  •  

$$\left( \begin{array}{l} \text{isLandingGearDeployedCall} . \text{LandMissionMID} ? \text{caller} \longrightarrow \\ \text{ret} := \text{this} . \text{isLandingGearDeployed}(); \\ \text{isLandingGearDeployedRet} . \text{LandMissionMID} . \text{caller} ! \text{ret} \longrightarrow \\ \text{Skip} \end{array} \right)$$

*Methods*  $\hat{=}$  
$$\left( \begin{array}{l} \text{InitializePhase} \\ \square \\ \text{CleanupPhase} \\ \square \\ \text{deployLandingGearMeth} \\ \square \\ \text{stowLandingGearMeth} \\ \square \\ \text{isLandingGearDeployedMeth} \end{array} \right); \text{Methods}$$

• (*Methods*)  $\triangle$  (*end\_mission\_app* . *LandMissionMID*  $\longrightarrow$  **Skip**)

**end**

**section** *LandMissionClass* **parents** *scj\_prelude*, *SchedulableId*, *SchedulableIds*, *SafeletChan*, *MethodCallBindingChannels*

**class** *LandMissionClass*  $\hat{=}$  **begin**

**state** *State*

---

*SAFE\_LANDING\_ALTITUDE* :  $\mathbb{P}\mathbb{A}$   
*ALTITUDE\_READING\_ON\_GROUND* :  $\mathbb{P}\mathbb{A}$   
*abort* :  $\mathbb{B}$   
*landingGearDeployed* :  $\mathbb{B}$

---

**state** *State*

**initial** *Init*

---

*State* '  


---

*SAFE\_LANDING\_ALTITUDE*' = 10.0  
*ALTITUDE\_READING\_ON\_GROUND*' = 0.0  
*abort*' = *false*

---

**public** *deployLandingGear*  $\hat{=}$   
(*landingGearDeployed* := **True**)

**public** *stowLandingGear*  $\hat{=}$   
(*landingGearDeployed* := **False**)

**public** *isLandingGearDeployed*  $\hat{=}$   
(*ret* := *landingGearDeployed*)

• **Skip**

**end**

## 5.8 Schedulables of LandMission

**section** *LandingGearHandlerLandApp* **parents** *AperiodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*, *MethodCall*, *LandMissionMethChan*

**process** *LandingGearHandlerLandApp*  $\hat{=}$   
*mission* : *MissionID* • **begin**

*handleAsyncEvent*  $\hat{=}$

$$\left( \begin{array}{l} \text{handleAsyncEventCall} . \text{LandingGearHandlerLandSID} \longrightarrow \\ \left( \begin{array}{l} \text{binder\_isLandingGearDeployedCall} . \text{mission} . \text{LandingGearHandlerLandSID} \longrightarrow \\ \text{binder\_isLandingGearDeployedRet} . \text{mission} . \text{LandingGearHandlerLandSID} ? \text{isLandingGearDeployed} \longrightarrow \\ \text{Skip} ; \text{var landingGearIsDeployed} : \mathbb{B} \bullet \text{landingGearIsDeployed} := \text{isLandingGearDeployed}; \\ \text{if landingGearIsDeployed} \longrightarrow \\ \left( \begin{array}{l} \text{binder\_stowLandingGearCall} . \text{mission} . \text{LandingGearHandlerLandSID} \longrightarrow \\ \text{binder\_stowLandingGearRet} . \text{mission} . \text{LandingGearHandlerLandSID} \longrightarrow \\ \text{Skip} \end{array} \right) \\ \square \neg \text{landingGearIsDeployed} \longrightarrow \\ \left( \begin{array}{l} \text{binder\_deployLandingGearCall} . \text{mission} . \text{LandingGearHandlerLandSID} \longrightarrow \\ \text{binder\_deployLandingGearRet} . \text{mission} . \text{LandingGearHandlerLandSID} \longrightarrow \\ \text{Skip} \end{array} \right) \end{array} \right) \\ \text{fi} \\ \text{handleAsyncEventRet} . \text{LandingGearHandlerLandSID} \longrightarrow \\ \text{Skip} \end{array} \right) ;$$

*Methods*  $\hat{=}$   
(*handleAsyncEvent*) ; *Methods*

• (*Methods*)  $\triangle$  (*end\_aperiodic\_app* . *LandingGearHandlerLandSID*  $\longrightarrow$  **Skip**)

**end**



**section** *SafeLandingHandlerApp* **parents** *AperiodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBinding*, *MainMissionMethChan*

**process** *SafeLandingHandlerApp*  $\hat{=}$   
*mainMission* : *MissionID*,  
*threshold* :  $\mathbb{P}\mathbb{A}$  • **begin**

*handleAsyncEvent*  $\hat{=}$   

$$\left( \begin{array}{l} \text{handleAsyncEventCall} . \text{SafeLandingHandlerSID} \longrightarrow \\ \left( \begin{array}{l} \text{binder\_getAltitudeCall} . \text{mainMission} . \text{SafeLandingHandlerSID} \longrightarrow \\ \text{binder\_getAltitudeRet} . \text{mainMission} . \text{SafeLandingHandlerSID} ? \text{getAltitude} \longrightarrow \\ \mathbf{Skip} ; \mathbf{var} \text{altitude} : \mathbb{P}\mathbb{A} \bullet \text{altitude} := \text{getAltitude}; \\ \mathbf{if} (\text{altitude} < \text{threshold}) \longrightarrow \\ \quad \mathbf{Skip} \\ \quad \square \neg (\text{altitude} < \text{threshold}) \longrightarrow \\ \quad \mathbf{Skip} \\ \mathbf{fi} \end{array} \right) ; \\ \text{handleAsyncEventRet} . \text{SafeLandingHandlerSID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*Methods*  $\hat{=}$   
(*handleAsyncEvent*) ; *Methods*

• (*Methods*)  $\triangle$  (*end\_aperiodic\_app* . *SafeLandingHandlerSID*  $\longrightarrow$  **Skip**)

**end**

**section** *GroundDistanceMonitorApp* **parents** *PeriodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBin*  
*, MainMissionMethChan*

**process** *GroundDistanceMonitorApp*  $\hat{=}$   
*mainMission* : *MissionID*,  
*readingOnGround* :  $\mathbb{P}\mathbb{A}$  • **begin**

*handleAsyncEvent*  $\hat{=}$   

$$\left( \begin{array}{l} \text{handleAsyncEventCall} . \text{GroundDistanceMonitorSID} \longrightarrow \\ \left( \begin{array}{l} \text{binder\_getAltitudeCall} . \text{mainMission} . \text{GroundDistanceMonitorSID} \longrightarrow \\ \text{binder\_getAltitudeRet} . \text{mainMission} . \text{GroundDistanceMonitorSID} ? \text{getAltitude} \longrightarrow \\ \mathbf{Skip} ; \mathbf{var} \text{ distance} : \mathbb{P}\mathbb{A} \bullet \text{distance} := \text{getAltitude}; \\ \mathbf{if} (\text{distance} = \text{readingOnGround}) \longrightarrow \\ \left( \begin{array}{l} \text{requestTerminationCall} . \text{mainMission} . \text{GroundDistanceMonitorSID} \longrightarrow \\ \text{requestTerminationRet} . \text{mainMission} . \text{GroundDistanceMonitorSID} ? \text{requestTermination} \longrightarrow \\ \mathbf{Skip} \end{array} \right) \\ \mathbb{I} \neg (\text{distance} = \text{readingOnGround}) \longrightarrow \mathbf{Skip} \\ \mathbf{fi} \end{array} \right) ; \\ \text{handleAsyncEventRet} . \text{GroundDistanceMonitorSID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*Methods*  $\hat{=}$   
 $(\text{handleAsyncEvent}) ; \text{Methods}$

•  $(\text{Methods}) \triangle (\text{end\_periodic\_app} . \text{GroundDistanceMonitorSID} \longrightarrow \mathbf{Skip})$

**end**

**section** *InstrumentLandingSystemMonitorApp* **parents** *PeriodicEventHandlerChan, SchedulableId, SchedulableIds, Meth*

**process** *InstrumentLandingSystemMonitorApp*  $\hat{=}$   
*mission : MissionID* • **begin**

*handleAsyncEvent*  $\hat{=}$   

$$\left( \begin{array}{l} \text{handleAsyncEventCall} . \text{InstrumentLandingSystemMonitorSID} \longrightarrow \\ \mathbf{Skip}; \\ \text{handleAsyncEventRet} . \text{InstrumentLandingSystemMonitorSID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*Methods*  $\hat{=}$   
 $(\text{handleAsyncEvent}) ; \text{Methods}$

•  $(\text{Methods}) \triangle (\text{end\_periodic\_app} . \text{InstrumentLandingSystemMonitorSID} \longrightarrow \mathbf{Skip})$

**end**