

# nestedSequencer5 Report

4th October 2016

## 1 ID Files

### 1.1 MissionIds

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

*TopMissionMID : MissionID*  
*MidMissionAMID : MissionID*  
*BottomMissionAMID : MissionID*  
*MidMissionBMID : MissionID*  
*BottomMissionBMID : MissionID*

---

*distinct⟨nullMissionId, TopMissionMID, MidMissionAMID,  
BottomMissionAMID, MidMissionBMID,  
BottomMissionBMID⟩*

## 1.2 SchedulablesIds

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

*TopSequencerSID : SchedulableID*

*MT1SID : SchedulableID*

*MidMissionSequencerSID : SchedulableID*

*BottomMissionSequencerASID : SchedulableID*

*OSEHSID : SchedulableID*

*MT2SID : SchedulableID*

*BottomMissionSequencerBSID : SchedulableID*

*APEHSID : SchedulableID*

*PEHSID : SchedulableID*

*distinct (nullSequencerId, nullSchedulableId, TopSequencerSID,  
MT1SID, MidMissionSequencerSID,  
BottomMissionSequencerASID, OSEHSID,  
MT2SID, BottomMissionSequencerBSID,  
APEHSID, PEHSID)*

## 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 . TopMission, done\_mission . TopMission, 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* ==  
 $\bigcup\{ \textit{SafeltAppSync, MissionSequencerAppSync, MissionAppSync, MTAppSync, OSEHSync, APEHSync, PEHSync,} \}$   
{ *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 . MidMissionA, done_mission . MidMissionA,
    initializeRet . MidMissionA, requestTermination . MidMissionA . TopSequencer }

```

```

channelset Tier1Sync ==
  { done_toplevel_sequencer, done_safeletFW,
    start_mission . BottomMissionA, done_mission . BottomMissionA,
    initializeRet . BottomMissionA, requestTermination . BottomMissionA . ,
    start_mission . MidMissionB, done_mission . MidMissionB,
    initializeRet . MidMissionB, requestTermination . MidMissionB . }

```

```

channelset Tier2Sync ==
  { done_toplevel_sequencer, done_safeletFW,
    start_mission . BottomMissionB, done_mission . BottomMissionB,
    initializeRet . BottomMissionB, requestTermination . BottomMissionB . }

```

## 2.2 Locking

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

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

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

**process** *Locking*  $\hat{=}$  *Threads* [ *ThreadSync* ] *Objects*

## 2.3 Program

**section** *Program* **parents** *scj\_prelude, MissionId, MissionIds, SchedulableId, SchedulableIds, MissionChan, SchedulableMethChan, MissionFW, SafeletFW, TopLevelMissionSequencerFW, NetworkChannels, ManagedThreadFW, SchedulableMissionSequencerFW, PeriodicEventHandlerFW, OneShotEventHandlerFW, AperiodicEventHandlerFW, ObjectFW, ThreadFW, MyAppApp, TopSequencerApp, TopMissionApp, MT1App, MidMissionSequencerApp, MidMissionAApp, BottomMissionSequencerAApp, BottomMissionAApp, MT2App, OSEHApp, MidMissionBApp, BottomMissionSequencerBApp, BottomMissionBApp, APEHApp, PEHApp*

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

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

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

$$\left( \begin{array}{l} \text{MissionFW}(\text{TopMissionID}) \\ \llbracket \text{MissionSync} \rrbracket \\ \left( \begin{array}{l} \text{ManagedThreadFW}(\text{MT1ID}) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \text{SchedulableMissionSequencerFW}(\text{MidMissionSequencerID}) \end{array} \right) \end{array} \right)$$

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

$$\left( \begin{array}{l} \text{MissionFW}(\text{MidMissionAID}) \\ \llbracket \text{MissionSync} \rrbracket \\ (\text{OneShotEventHandlerFW}(\text{BottomMissionSequencerAID})) \end{array} \right)$$

**process** *Tier2*  $\hat{=}$

$$\left( \begin{array}{l} \text{MissionFW}(\text{BottomMissionAID}) \\ \llbracket \text{MissionSync} \rrbracket \\ \left( \begin{array}{l} \text{ManagedThreadFW}(\text{MT2ID}) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \text{OneShotEventHandlerFW}(\text{OSEHID}, (\text{time}(60, 0)), (\text{time}(100, 0), \text{null})) \end{array} \right) \\ \llbracket \text{ClusterSync} \rrbracket \\ \left( \begin{array}{l} \text{MissionFW}(\text{MidMissionBID}) \\ \llbracket \text{MissionSync} \rrbracket \\ (\text{OneShotEventHandlerFW}(\text{BottomMissionSequencerBID})) \end{array} \right) \end{array} \right)$$

**process** *Tier3*  $\hat{=}$

$$\left( \begin{array}{l} \text{MissionFW}(\text{BottomMissionBID}) \\ \llbracket \text{MissionSync} \rrbracket \\ \left( \begin{array}{l} \text{OneShotEventHandlerFW}(\text{APEHID}, (\text{time}(5, 0), \text{null})) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \text{AperiodicEventHandlerFW}(\text{PEHID}, (\text{time}(60, 0), \text{time}(5, 0), \text{NULL}, \text{nullSchedulableId})) \end{array} \right) \end{array} \right)$$

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

$$\left( \begin{array}{l} \text{ControlTier} \\ \llbracket \text{TierSync} \rrbracket \\ \left( \begin{array}{l} \text{Tier0} \\ \llbracket \text{Tier0Sync} \rrbracket \end{array} \right) \\ \text{Tier1} \\ \llbracket \text{Tier1Sync} \rrbracket \\ \text{Tier2} \\ \llbracket \text{Tier2Sync} \rrbracket \\ \text{Tier3} \end{array} \right)$$

$$\text{process } Application \hat{=} \left( \begin{array}{l} MyAppApp \\ ||| \\ TopSequencerApp \\ ||| \\ TopMissionApp \\ ||| \\ MT1App \\ ||| \\ MidMissionSequencerApp \\ ||| \\ MidMissionAApp \\ ||| \\ BottomMissionSequencerAApp \\ ||| \\ BottomMissionAApp \\ ||| \\ MT2App \\ ||| \\ OSEHApp(BottomMissionAID) \\ ||| \\ MidMissionBApp \\ ||| \\ BottomMissionSequencerBApp \\ ||| \\ BottomMissionBApp \\ ||| \\ APEHApp(BottomMissionBID) \\ ||| \\ PEHApp(apehID) \end{array} \right)$$

$$\text{process } Program \hat{=} (Framework \llbracket AppSync \rrbracket Application) \llbracket LockingSync \rrbracket Locking$$

### 3 Safelet

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

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

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

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

*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** *TopSequencerApp* **parents** *TopLevelMissionSequencerChan*,  
*MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *TopSequencerClass*, *MethodCallBindingChannels*

**process** *TopSequencerApp*  $\hat{=}$   
*name* : *String* • **begin**

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

**state** *State*

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

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

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

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

**end**

**section** *TopSequencerClass* **parents** *scj\_prelude*, *SchedulableId*, *SchedulableIds*, *SafeletChannels*, *MethodCallBindingChannels*, *MissionId*, *MissionIds*

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

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

**state** *State*

<b>initial</b> <i>Init</i> <i>State</i> '
<i>notReleased</i> = <b>True</b>

**protected** *getNextMission*  $\hat{=}$  **var** *ret* : *MissionID* •

$$\left( \begin{array}{l} \text{if } \text{notReleased} = \mathbf{True} \longrightarrow \\ \quad \left( \text{notReleased} := \mathbf{False}; \right. \\ \quad \left. \text{ret} := \text{TopMissionMID} \right) \\ \quad \square \neg \text{notReleased} = \mathbf{True} \longrightarrow \\ \quad \quad \left( \text{ret} := \text{nullMissionId} \right) \\ \text{fi} \end{array} \right)$$

• **Skip**

**end**

## 5 Missions

### 5.1 TopMission

**section** *TopMissionApp* **parents** *scj\_prelude*, *MissionId*, *MissionIds*,  
*SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *TopMissionMethChan*,  
*MethodCallBindingChannels*

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

*InitializePhase*  $\hat{=}$   

$$\left( \begin{array}{l} \text{initializeCall} . \text{TopMissionMID} \longrightarrow \\ \text{register} ! \text{MT1SID} ! \text{TopMissionMID} \longrightarrow \\ \text{register} ! \text{MidMissionSequencerSID} ! \text{TopMissionMID} \longrightarrow \\ \text{initializeRet} . \text{TopMissionMID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*CleanupPhase*  $\hat{=}$   

$$\left( \begin{array}{l} \mathbf{var} \ \mathbb{B} : \text{ret} \bullet \text{cleanupMissionCall} . \text{TopMissionMID} \longrightarrow \\ \text{cleanupMissionRet} . \text{TopMissionMID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

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

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

**end**

## 5.2 Schedulables of TopMission

**section** *MT1App* **parents** *ManagedThreadChan, SchedulableId, SchedulableIds, MethodCallBindingChannels*

**process** *MT1App*  $\hat{=}$  **begin**

*Run*  $\hat{=}$   
 $\left( \begin{array}{l} \text{runCall} . \text{MT1SID} \longrightarrow \\ \mathbf{Skip}; \\ \text{runRet} . \text{MT1SID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

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

•  $(\text{Methods}) \triangle (\text{end\_managedThread\_app} . \text{MT1SID} \longrightarrow \mathbf{Skip})$

**end**

**section** *MidMissionSequencerApp* **parents** *TopLevelMissionSequencerChan*,  
*MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *MidMissionSequencerClass*, *MethodCallBindingChannels*

**process** *MidMissionSequencerApp*  $\hat{=}$   
*name* : *String* • **begin**

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

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

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

**end**

**section** *MidMissionSequencerClass* **parents** *scj\_prelude*, *SchedulableId*, *SchedulableIds*, *SafeletChannels*, *MethodCallBindingChannels*, *MissionId*, *MissionIds*

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

---

**state** *State*  
*releases* :  $\mathbb{Z}$

---

**state** *State*

---

**initial** *Init*  
*State* '  
*releases* = 0

---

**protected** *getNextMission*  $\hat{=}$  **var** *ret* : *MissionID* •

$$\left( \begin{array}{l} \text{if } releases = 0 \longrightarrow \\ \quad \left( \begin{array}{l} releases := releases + 1; \\ ret := MidMissionAMID \end{array} \right) \\ \quad \square \neg releases = 0 \longrightarrow \\ \quad \quad \text{if } releases = 1 \longrightarrow \\ \quad \quad \quad \left( \begin{array}{l} releases := releases + 1; \\ ret := MidMissionBMID \end{array} \right) \\ \quad \quad \quad \square \neg releases = 1 \longrightarrow \\ \quad \quad \quad \quad (ret := nullMissionId) \\ \quad \text{fi} \\ \text{fi} \end{array} \right)$$

• Skip

**end**

### 5.3 MidMissionA

**section** *MidMissionAApp* **parents** *scj\_prelude*, *MissionId*, *MissionIds*,  
*SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *MidMissionAMethChan*,  
*MethodCallBindingChannels*

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

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

**state** *State*

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

*InitializePhase*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{initializeCall} . \textit{MidMissionAMID} \longrightarrow \\ \textit{register} ! \textit{BottomMissionSequencerASID} ! \textit{MidMissionAMID} \longrightarrow \\ \textit{initializeRet} . \textit{MidMissionAMID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

*CleanupPhase*  $\hat{=}$   
 $\left( \begin{array}{l} \mathbf{var} \ \mathbb{B} : \textit{ret} \bullet \textit{cleanupMissionCall} . \textit{MidMissionAMID} \longrightarrow \\ \textit{cleanupMissionRet} . \textit{MidMissionAMID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

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

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

**end**

## 5.4 Schedulables of MidMissionA

**section** *BottomMissionSequencerAApp* **parents** *TopLevelMissionSequencerChan,*  
*MissionId, MissionIds, SchedulableId, SchedulableIds, BottomMissionSequencerAClass, MethodCallBindingChannels*

**process** *BottomMissionSequencerAApp*  $\hat{=}$   
*name* : *String* • **begin**

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

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

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

**end**



**section** *BottomMissionSequencerAClass* **parents** *scj\_prelude, SchedulableId, SchedulableIds, SafeletChannels, MethodCallBindingChannels, MissionId, MissionIds*

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

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

**state** *State*

<b>initial</b> <i>Init</i> <i>State</i> '
<i>notReleased</i> = <b>True</b>

**protected** *getNextMission*  $\hat{=}$  **var** *ret* : *MissionID* •

$$\left( \begin{array}{l} \text{if } \textit{notReleased} = \mathbf{True} \longrightarrow \\ \quad \left( \textit{notReleased} := \mathbf{False}; \right. \\ \quad \left. \textit{ret} := \textit{BottomMissionAMID} \right) \\ \quad \square \neg \textit{notReleased} = \mathbf{True} \longrightarrow \\ \quad \quad \left( \textit{ret} := \textit{nullMissionId} \right) \\ \text{fi} \end{array} \right)$$

• **Skip**

**end**

## 5.5 BottomMissionA

**section** *BottomMissionAApp* **parents** *scj\_prelude*, *MissionId*, *MissionIds*,  
*SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *BottomMissionAMethChan*,  
*MethodCallBindingChannels*

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

*State*

*this* : **ref** *BottomMissionAClass*

**state** *State*

*Init*

*State'*

*this'* = **new** *BottomMissionAClass*()

*InitializePhase*  $\hat{=}$

$$\left( \begin{array}{l} \text{initializeCall} . \text{BottomMissionAMID} \longrightarrow \\ \text{register} ! \text{OSEHSID} ! \text{BottomMissionAMID} \longrightarrow \\ \text{register} ! \text{MT2SID} ! \text{BottomMissionAMID} \longrightarrow \\ \text{initializeRet} . \text{BottomMissionAMID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*CleanupPhase*  $\hat{=}$

$$\left( \begin{array}{l} \mathbf{var} \ \mathbb{B} : \text{ret} \bullet \text{cleanupMissionCall} . \text{BottomMissionAMID} \longrightarrow \\ \text{cleanupMissionRet} . \text{BottomMissionAMID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

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

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

**end**

## 5.6 Schedulables of BottomMissionA

**section** *MT2App* **parents** *ManagedThreadChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBindingChannels*

**process** *MT2App*  $\hat{=}$  **begin**

*Run*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{runCall} . \textit{MT2SID} \longrightarrow \\ \mathbf{Skip}; \\ \textit{runRet} . \textit{MT2SID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

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

•  $(\textit{Methods}) \triangle (\textit{end\_managedThread\_app} . \textit{MT2SID} \longrightarrow \mathbf{Skip})$

**end**

**section** *OSEHApp* **parents** *OneShotEventHandlerChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBindingChannels*

**process** *OSEHApp*  $\hat{=}$   
     *start* : *HighResolutionTime*,  
     *controllingMission* : *MissionID* • **begin**

*handleAsyncEvent*  $\hat{=}$   

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

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

• (*Methods*)  $\triangle$  (*end\_oneShot\_app* . *OSEHSID*  $\longrightarrow$  **Skip**)

**end**

## 5.7 MidMissionB

**section** *MidMissionBApp* **parents** *scj\_prelude*, *MissionId*, *MissionIds*,  
*SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *MidMissionBMethChan*,  
*MethodCallBindingChannels*

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

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

**state** *State*

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

*InitializePhase*  $\hat{=}$   

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

*CleanupPhase*  $\hat{=}$   

$$\left( \begin{array}{l} \mathbf{var} \ \mathbb{B} : \textit{ret} \bullet \textit{cleanupMissionCall} . \textit{MidMissionBMID} \longrightarrow \\ \textit{cleanupMissionRet} . \textit{MidMissionBMID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

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

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

**end**

## 5.8 Schedulables of MidMissionB

**section** *BottomMissionSequencerBApp* **parents** *TopLevelMissionSequencerChan,*  
*MissionId, MissionIds, SchedulableId, SchedulableIds, BottomMissionSequencerBClass, MethodCallBindingChannels*

**process** *BottomMissionSequencerBApp*  $\hat{=}$   
*name* : *String* • **begin**

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

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

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

**end**

**section** *BottomMissionSequencerBClass* **parents** *scj\_prelude, SchedulableId, SchedulableIds, SafeletChannels, MethodCallBindingChannels, MissionId, MissionIds*

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

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

**state** *State*

<b>initial</b> <i>Init</i> <i>State</i> '
<i>notReleased</i> = <b>True</b>

**protected** *getNextMission*  $\hat{=}$  **var** *ret* : *MissionID* •

$$\left( \begin{array}{l} \text{if } notReleased = \mathbf{True} \longrightarrow \\ \quad \left( notReleased := \mathbf{False}; \right. \\ \quad \left. ret := BottomMissionBMID \right) \\ \quad \Box \neg notReleased = \mathbf{True} \longrightarrow \\ \quad \quad (ret := nullMissionId) \\ \text{fi} \end{array} \right)$$

• **Skip**

**end**

## 5.9 BottomMissionB

**section** *BottomMissionBApp* **parents** *scj\_prelude*, *MissionId*, *MissionIds*,  
*SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *BottomMissionBMethChan*,  
*MethodCallBindingChannels*

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

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

**state** *State*

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

*InitializePhase*  $\hat{=}$   

$$\left( \begin{array}{l} \text{initializeCall} . \text{BottomMissionBMID} \longrightarrow \\ \text{register} ! \text{APEHSID} ! \text{BottomMissionBMID} \longrightarrow \\ \text{register} ! \text{PEHSID} ! \text{BottomMissionBMID} \longrightarrow \\ \text{initializeRet} . \text{BottomMissionBMID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*CleanupPhase*  $\hat{=}$   

$$\left( \begin{array}{l} \mathbf{var} \ \mathbb{B} : \text{ret} \bullet \text{cleanupMissionCall} . \text{BottomMissionBMID} \longrightarrow \\ \text{cleanupMissionRet} . \text{BottomMissionBMID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

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

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

**end**



## 5.10 Schedulables of BottomMissionB

**section** *APEHApp* **parents** *AperiodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBindingChannels*

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

*handleAsyncEvent*  $\hat{=}$   

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

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

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

**end**

**section** *PEHApp* **parents** *PeriodicEventHandlerChan, SchedulableId, SchedulableIds, MethodCallBindingChannels*

**process** *PEHApp*  $\hat{=}$   
     *apeh* : *SchedulableID* • **begin**

*handleAsyncEvent*  $\hat{=}$   

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

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

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

**end**