

# nestedSequencer2

Tight Rope v0.88

4th March 2017

## 1 ID Files

### 1.1 MissionIds

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

*TopMission1MID* : *MissionID*

*MyMission1MID* : *MissionID*

*MyMission2MID* : *MissionID*

*MyMission3MID* : *MissionID*

---

*distinct*(*nullMissionId*, *TopMission1MID*, *MyMission1MID*,  
*MyMission2MID*, *MyMission3MID*)

## 1.2 SchedulablesIds

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

<i>MySequencerSID : SchedulableID</i> <i>FirstMissionSequencerSID : SchedulableID</i> <i>SecondMissionSequencerSID : SchedulableID</i> <i>ThirdMissionSequencerSID : SchedulableID</i> <i>MyPEH1SID : SchedulableID</i> <i>MyPEH2SID : SchedulableID</i> <i>MyPEH3SID : SchedulableID</i>
<i>distinct⟨nullSequencerId, nullSchedulableId, MySequencerSID,</i> <i>FirstMissionSequencerSID, SecondMissionSequencerSID,</i> <i>ThirdMissionSequencerSID, MyPEH1SID,</i> <i>MyPEH2SID, MyPEH3SID⟩</i>

### 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*

<i>distinct</i> $\langle \rangle$
-----------------------------------

## 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 . TopMission1, done\_mission . TopMission1, 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* ==  
     $\bigcup$ { *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 . MyMission1, done_mission . MyMission1,
    initializeRet . MyMission1, requestTermination . MyMission1 . MySequencer }

```

```

channelset Tier1Sync ==
  { done_toplevel_sequencer, done_safeletFW,
    start_mission . MyMission2, done_mission . MyMission2,
    initializeRet . MyMission2, requestTermination . MyMission2 . }

```

```

channelset Tier2Sync ==
  { done_toplevel_sequencer, done_safeletFW,
    start_mission . MyMission3, done_mission . MyMission3,
    initializeRet . MyMission3, requestTermination . MyMission3 . }

```

## 2.2 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**)



## 2.3 Program

**section** *Program* **parents** *scj\_prelude, MissionId, MissionIds, SchedulableId, SchedulableIds, MissionChan, SchedulableMethChan, MissionFW, SafeletFW, TopLevelMissionSequencerFW, NetworkChannels, ManagedThreadFW, SchedulableMissionSequencerFW, PeriodicEventHandlerFW, OneShotEventHandlerFW, AperiodicEventHandlerFW, ObjectFW, ThreadFW, MyAppApp, MySequencerApp, TopMission1App, FirstMissionSequencerApp, SecondMissionSequencerApp, ThirdMissionSequencerApp, MyMission1App, MyPEH1App, MyMission2App, MyPEH2App, MyMission3App, MyPEH3App*

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

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

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

$$\left( \begin{array}{l} \text{MissionFW}(\text{TopMission1ID}) \\ \llbracket \text{MissionSync} \rrbracket \\ \left( \begin{array}{l} \text{SchedulableMissionSequencerFW}(\text{FirstMissionSequencerID}) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \text{SchedulableMissionSequencerFW}(\text{SecondMissionSequencerID}) \\ \llbracket \text{SchedulablesSync} \rrbracket \\ \text{SchedulableMissionSequencerFW}(\text{ThirdMissionSequencerID}) \end{array} \right) \end{array} \right)$$

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

$$\left( \begin{array}{l} \text{MissionFW}(\text{MyMission1ID}) \\ \llbracket \text{MissionSync} \rrbracket \\ (\text{PeriodicEventHandlerFW}(\text{MyPEH1ID}, (\text{NULL}, \text{time}(1000, 0), \text{NULL}, \text{nullSchedulableId}))) \end{array} \right)$$

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

$$\left( \begin{array}{l} \text{MissionFW}(\text{MyMission2ID}) \\ \llbracket \text{MissionSync} \rrbracket \\ (\text{PeriodicEventHandlerFW}(\text{MyPEH2ID}, (\text{NULL}, \text{time}(1000, 0), \text{NULL}, \text{nullSchedulableId}))) \end{array} \right)$$

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

$$\left( \begin{array}{l} \text{MissionFW}(\text{MyMission3ID}) \\ \llbracket \text{MissionSync} \rrbracket \\ (\text{PeriodicEventHandlerFW}(\text{MyPEH3ID}, (\text{NULL}, \text{time}(1000, 0), \text{NULL}, \text{nullSchedulableId}))) \end{array} \right)$$

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

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

$$\text{process } Application \hat{=} \left( \begin{array}{l} MyAppApp \\ ||| \\ MySequencerApp \\ ||| \\ TopMission1App \\ ||| \\ FirstMissionSequencerApp \\ ||| \\ SecondMissionSequencerApp \\ ||| \\ ThirdMissionSequencerApp \\ ||| \\ MyMission1App \\ ||| \\ MyPEH1App(MyMission1ID) \\ ||| \\ MyMission2App \\ ||| \\ MyPEH2App(MyMission2ID) \\ ||| \\ MyMission3App \\ ||| \\ MyPEH3App(MyMission3ID) \end{array} \right)$$

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

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

*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** *MySequencerApp* **parents** *TopLevelMissionSequencerChan*,  
*MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *MySequencerClass*, *MethodCallBindingChannels*

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

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

**state** *State*

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

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

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

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

**end**

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

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

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

**state** *State*

<b>initial</b> <i>Init</i> <i>State</i> '
<i>done</i> ' = <i>false</i>

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

$$\left( \begin{array}{l} \text{if } (done = \mathbf{False}) \longrightarrow \\ \quad \left( \begin{array}{l} done := \mathbf{True}; \\ ret := TopMission1MID \end{array} \right) \\ \quad \parallel \neg (done = \mathbf{False}) \longrightarrow \\ \quad \quad (ret := nullMissionId) \\ \text{fi} \end{array} \right)$$

• **Skip**

**end**

## 5 Missions

### 5.1 TopMission1

**section** *TopMission1App* **parents** *scj\_prelude*, *MissionId*, *MissionIds*,  
*SchedulableId*, *SchedulableIds*, *MissionChan*, *SchedulableMethChan*, *TopMission1MethChan*  
*, MethodCallBindingChannels*

**process** *TopMission1App*  $\hat{=}$  **begin**

*InitializePhase*  $\hat{=}$   

$$\left( \begin{array}{l} \textit{initializeCall} . \textit{TopMission1MID} \longrightarrow \\ \textit{register} ! \textit{FirstMissionSequencerSID} ! \textit{TopMission1MID} \longrightarrow \\ \textit{register} ! \textit{SecondMissionSequencerSID} ! \textit{TopMission1MID} \longrightarrow \\ \textit{register} ! \textit{ThirdMissionSequencerSID} ! \textit{TopMission1MID} \longrightarrow \\ \textit{initializeRet} . \textit{TopMission1MID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*CleanupPhase*  $\hat{=}$   

$$\left( \begin{array}{l} \textit{cleanupMissionCall} . \textit{TopMission1MID} \longrightarrow \\ \textit{cleanupMissionRet} . \textit{TopMission1MID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

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

•  $(\textit{Methods}) \triangle (\textit{end\_mission\_app} . \textit{TopMission1MID} \longrightarrow \mathbf{Skip})$

**end**

## 5.2 Schedulables of TopMission1

**section** *FirstMissionSequencerApp* **parents** *TopLevelMissionSequencerChan*,  
*MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *FirstMissionSequencerClass*, *MethodCallBindingChannels*

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

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

**state** *State*

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

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

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

• (*Init* ; *Methods*)  $\triangle$  (*end\_sequencer\_app* . *FirstMissionSequencerSID*  $\longrightarrow$  **Skip**)

**end**

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

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

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

**state** *State*

<b>initial</b> <i>Init</i> <i>State</i> ' <i>done</i> ' = <i>false</i>
--

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

$$\left( \begin{array}{l} \text{if } (done = \mathbf{False}) \longrightarrow \\ \quad \left( \begin{array}{l} done := \mathbf{True}; \\ ret := MyMission1MID \end{array} \right) \\ \parallel \neg (done = \mathbf{False}) \longrightarrow \\ \quad (ret := nullMissionId) \\ \text{fi} \end{array} \right)$$

• **Skip**

**end**



**section** *SecondMissionSequencerApp* **parents** *TopLevelMissionSequencerChan*,  
*MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *SecondMissionSequencerClass*, *MethodCallBindingChannels*

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

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

**state** *State*

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

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

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

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

**end**

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

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

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

**state** *State*

<b>initial</b> <i>Init</i> <i>State</i> ' <i>done</i> ' = <i>false</i>
--

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

$$\left( \begin{array}{l} \text{if } (done = \mathbf{False}) \longrightarrow \\ \quad \left( \begin{array}{l} done := \mathbf{True}; \\ ret := MyMission2MID \end{array} \right) \\ \parallel \neg (done = \mathbf{False}) \longrightarrow \\ \quad (ret := nullMissionId) \\ \text{fi} \end{array} \right)$$

• **Skip**

**end**

**section** *ThirdMissionSequencerApp* **parents** *TopLevelMissionSequencerChan*,  
*MissionId*, *MissionIds*, *SchedulableId*, *SchedulableIds*, *ThirdMissionSequencerClass*, *MethodCallBindingChannels*

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

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

**state** *State*

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

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

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

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

**end**

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

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

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

**state** *State*

<b>initial</b> <i>Init</i> <i>State</i> ' <i>done</i> ' = <i>false</i>
--

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

$$\left( \begin{array}{l} \text{if } (done = \mathbf{False}) \longrightarrow \\ \quad \left( \begin{array}{l} done := \mathbf{True}; \\ ret := MyMission3MID \end{array} \right) \\ \parallel \neg (done = \mathbf{False}) \longrightarrow \\ \quad (ret := nullMissionId) \\ \text{fi} \end{array} \right)$$

• **Skip**

**end**

### 5.3 MyMission1

**section** *MyMission1App* **parents** *scj\_prelude, MissionId, MissionIds,*  
*SchedulableId, SchedulableIds, MissionChan, SchedulableMethChan, MyMission1MethChan*  
*, MethodCallBindingChannels*

**process** *MyMission1App*  $\hat{=}$  **begin**

*InitializePhase*  $\hat{=}$   

$$\left( \begin{array}{l} \textit{initializeCall} . \textit{MyMission1MID} \longrightarrow \\ \textit{register} ! \textit{MyPEH1SID} ! \textit{MyMission1MID} \longrightarrow \\ \textit{initializeRet} . \textit{MyMission1MID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*CleanupPhase*  $\hat{=}$   

$$\left( \begin{array}{l} \textit{cleanupMissionCall} . \textit{MyMission1MID} \longrightarrow \\ \textit{cleanupMissionRet} . \textit{MyMission1MID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

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

•  $(\textit{Methods}) \triangle (\textit{end\_mission\_app} . \textit{MyMission1MID} \longrightarrow \mathbf{Skip})$

**end**

## 5.4 Schedulables of MyMission1

**section** *MyPEH1App* **parents** *PeriodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBindingChannels*

**process** *MyPEH1App*  $\hat{=}$   
     *m* : *MissionID* • **begin**

<i>State</i> <i>this</i> : <b>ref</b> <i>MyPEH1Class</i>
---

**state** *State*

<i>Init</i> <i>State</i> '
<i>this</i> ' = <b>new</b> <i>MyPEH1Class</i> ()

*handleAsyncEvent*  $\hat{=}$

$$\left( \begin{array}{l} \text{handleAsyncEventCall} . \text{MyPEH1SID} \longrightarrow \\ \left( \begin{array}{l} \text{this} . \text{count} := \text{this} . \text{count} + 1; \\ \text{if } (\text{this} . \text{count} = 10) \longrightarrow \\ \left( \begin{array}{l} \text{requestTerminationCall} . m . \text{MyPEH1SID} \longrightarrow \\ \text{requestTerminationRet} . m . \text{MyPEH1SID} ? \text{requestTermination} \longrightarrow \end{array} \right) \\ \text{Skip} \end{array} \right) ; \\ \square \neg (\text{this} . \text{count} = 10) \longrightarrow \text{Skip} \\ \text{fi} \\ \text{handleAsyncEventRet} . \text{MyPEH1SID} \longrightarrow \\ \text{Skip} \end{array} \right)$$

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

• (*Init* ; *Methods*)  $\triangle$  (*end\_periodic\_app* . *MyPEH1SID*  $\longrightarrow$  **Skip**)

**end**

**section** *MyPEH1Class* **parents** *scj\_prelude, SchedulableId, SchedulableIds, SafeletChan*  
*, MethodCallBindingChannels*

**class** *MyPEH1Class*  $\hat{=}$  **begin**

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

**state** *State*

<b>initial</b> <i>Init</i> <i>State'</i>
<i>count'</i> = 0

• **Skip**

**end**

## 5.5 MyMission2

**section** *MyMission2App* **parents** *scj\_prelude, MissionId, MissionIds,*  
*SchedulableId, SchedulableIds, MissionChan, SchedulableMethChan, MyMission2MethChan*  
*, MethodCallBindingChannels*

**process** *MyMission2App*  $\hat{=}$  **begin**

*InitializePhase*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{initializeCall} . \textit{MyMission2MID} \longrightarrow \\ \textit{register} ! \textit{MyPEH2SID} ! \textit{MyMission2MID} \longrightarrow \\ \textit{initializeRet} . \textit{MyMission2MID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

*CleanupPhase*  $\hat{=}$   
 $\left( \begin{array}{l} \textit{cleanupMissionCall} . \textit{MyMission2MID} \longrightarrow \\ \textit{cleanupMissionRet} . \textit{MyMission2MID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$

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

•  $(\textit{Methods}) \triangle (\textit{end\_mission\_app} . \textit{MyMission2MID} \longrightarrow \mathbf{Skip})$

**end**



## 5.6 Schedulables of MyMission2

**section** *MyPEH2App* **parents** *PeriodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBindingChannels*

**process** *MyPEH2App*  $\hat{=}$   
     *m* : *MissionID* • **begin**

<i>State</i> <i>this</i> : <b>ref</b> <i>MyPEH2Class</i>
---

**state** *State*

<i>Init</i> <i>State</i> '
<i>this</i> ' = <b>new</b> <i>MyPEH2Class</i> ()

*handleAsyncEvent*  $\hat{=}$

$$\left( \begin{array}{l} \text{handleAsyncEventCall} . \text{MyPEH2SID} \longrightarrow \\ \left( \begin{array}{l} \text{this} . \text{count} := \text{this} . \text{count} + 1; \\ \text{if } (\text{this} . \text{count} = 10) \longrightarrow \\ \left( \begin{array}{l} \text{requestTerminationCall} . m . \text{MyPEH2SID} \longrightarrow \\ \text{requestTerminationRet} . m . \text{MyPEH2SID} ? \text{requestTermination} \longrightarrow \end{array} \right) \\ \text{Skip} \end{array} \right) ; \\ \square \neg (\text{this} . \text{count} = 10) \longrightarrow \text{Skip} \\ \text{fi} \\ \text{handleAsyncEventRet} . \text{MyPEH2SID} \longrightarrow \\ \text{Skip} \end{array} \right)$$

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

• (*Init* ; *Methods*)  $\triangle$  (*end\_periodic\_app* . *MyPEH2SID*  $\longrightarrow$  **Skip**)

**end**

**section** *MyPEH2Class* **parents** *scj\_prelude*, *SchedulableId*, *SchedulableIds*, *SafeletChan*  
*, MethodCallBindingChannels*

**class** *MyPEH2Class*  $\hat{=}$  **begin**

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

**state** *State*

<b>initial</b> <i>Init</i> <i>State'</i>
<i>count'</i> = 0

• **Skip**

**end**

## 5.7 MyMission3

**section** *MyMission3App* **parents** *scj\_prelude, MissionId, MissionIds,*  
*SchedulableId, SchedulableIds, MissionChan, SchedulableMethChan, MyMission3MethChan*  
*, MethodCallBindingChannels*

**process** *MyMission3App*  $\hat{=}$  **begin**

*InitializePhase*  $\hat{=}$   

$$\left( \begin{array}{l} \textit{initializeCall} . \textit{MyMission3MID} \longrightarrow \\ \textit{register} ! \textit{MyPEH3SID} ! \textit{MyMission3MID} \longrightarrow \\ \textit{initializeRet} . \textit{MyMission3MID} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

*CleanupPhase*  $\hat{=}$   

$$\left( \begin{array}{l} \textit{cleanupMissionCall} . \textit{MyMission3MID} \longrightarrow \\ \textit{cleanupMissionRet} . \textit{MyMission3MID} ! \mathbf{True} \longrightarrow \\ \mathbf{Skip} \end{array} \right)$$

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

•  $(\textit{Methods}) \triangle (\textit{end\_mission\_app} . \textit{MyMission3MID} \longrightarrow \mathbf{Skip})$

**end**

## 5.8 Schedulables of MyMission3

**section** *MyPEH3App* **parents** *PeriodicEventHandlerChan*, *SchedulableId*, *SchedulableIds*, *MethodCallBindingChannels*

**process** *MyPEH3App*  $\hat{=}$   
     *m* : *MissionID* • **begin**

<i>State</i> <i>this</i> : <b>ref</b> <i>MyPEH3Class</i>
---

**state** *State*

<i>Init</i> <i>State</i> '
<i>this</i> ' = <b>new</b> <i>MyPEH3Class</i> ()

*handleAsyncEvent*  $\hat{=}$

$$\left( \begin{array}{l} \text{handleAsyncEventCall} . \text{MyPEH3SID} \longrightarrow \\ \left( \begin{array}{l} \text{this} . \text{count} := \text{this} . \text{count} + 1; \\ \text{if } (\text{this} . \text{count} = 10) \longrightarrow \\ \left( \begin{array}{l} \text{requestTerminationCall} . m . \text{MyPEH3SID} \longrightarrow \\ \text{requestTerminationRet} . m . \text{MyPEH3SID} ? \text{requestTermination} \longrightarrow \end{array} \right) \\ \text{Skip} \end{array} \right) ; \\ \square \neg (\text{this} . \text{count} = 10) \longrightarrow \text{Skip} \\ \text{fi} \\ \text{handleAsyncEventRet} . \text{MyPEH3SID} \longrightarrow \\ \text{Skip} \end{array} \right)$$

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

• (*Init* ; *Methods*)  $\triangle$  (*end\_periodic\_app* . *MyPEH3SID*  $\longrightarrow$  **Skip**)

**end**

**section** *MyPEH3Class* **parents** *scj\_prelude, SchedulableId, SchedulableIds, SafeletChan*  
*, MethodCallBindingChannels*

**class** *MyPEH3Class*  $\hat{=}$  **begin**

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

**state** *State*

<b>initial</b> <i>Init</i> <i>State'</i>
<i>count'</i> = 0

• **Skip**

**end**