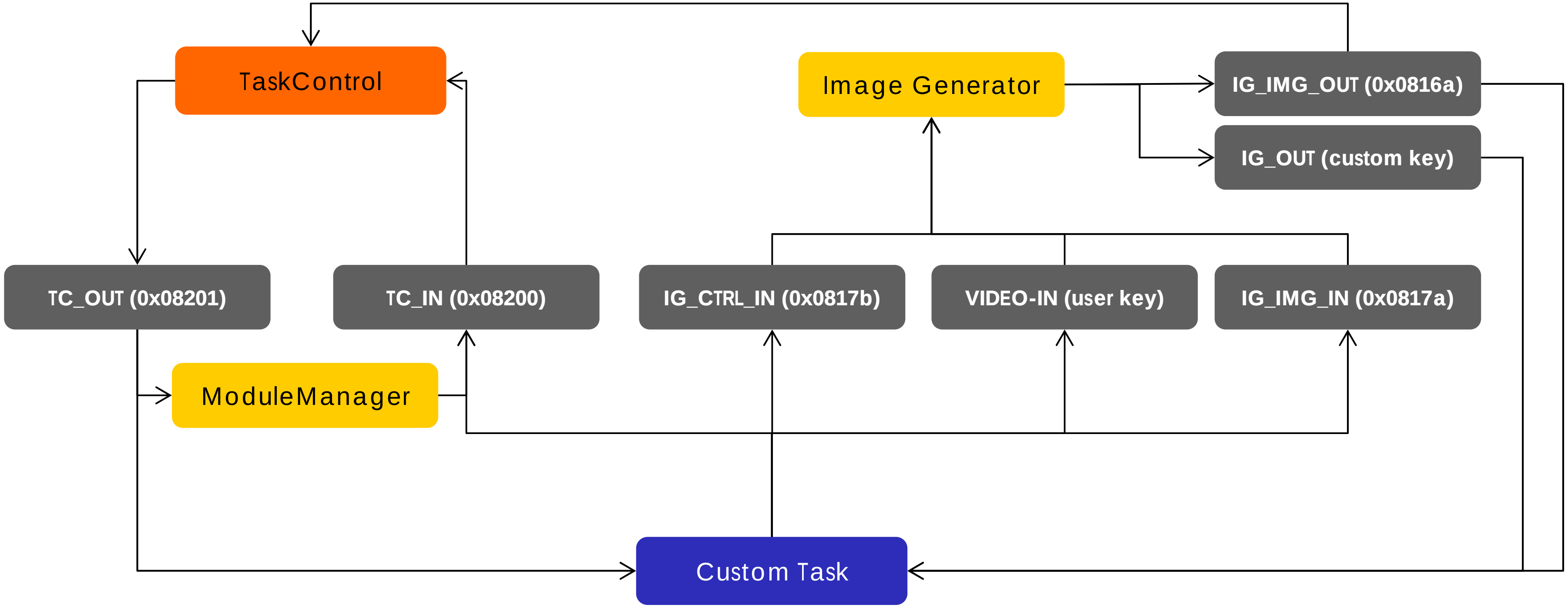
Shared Memory

Shared Memory Overview

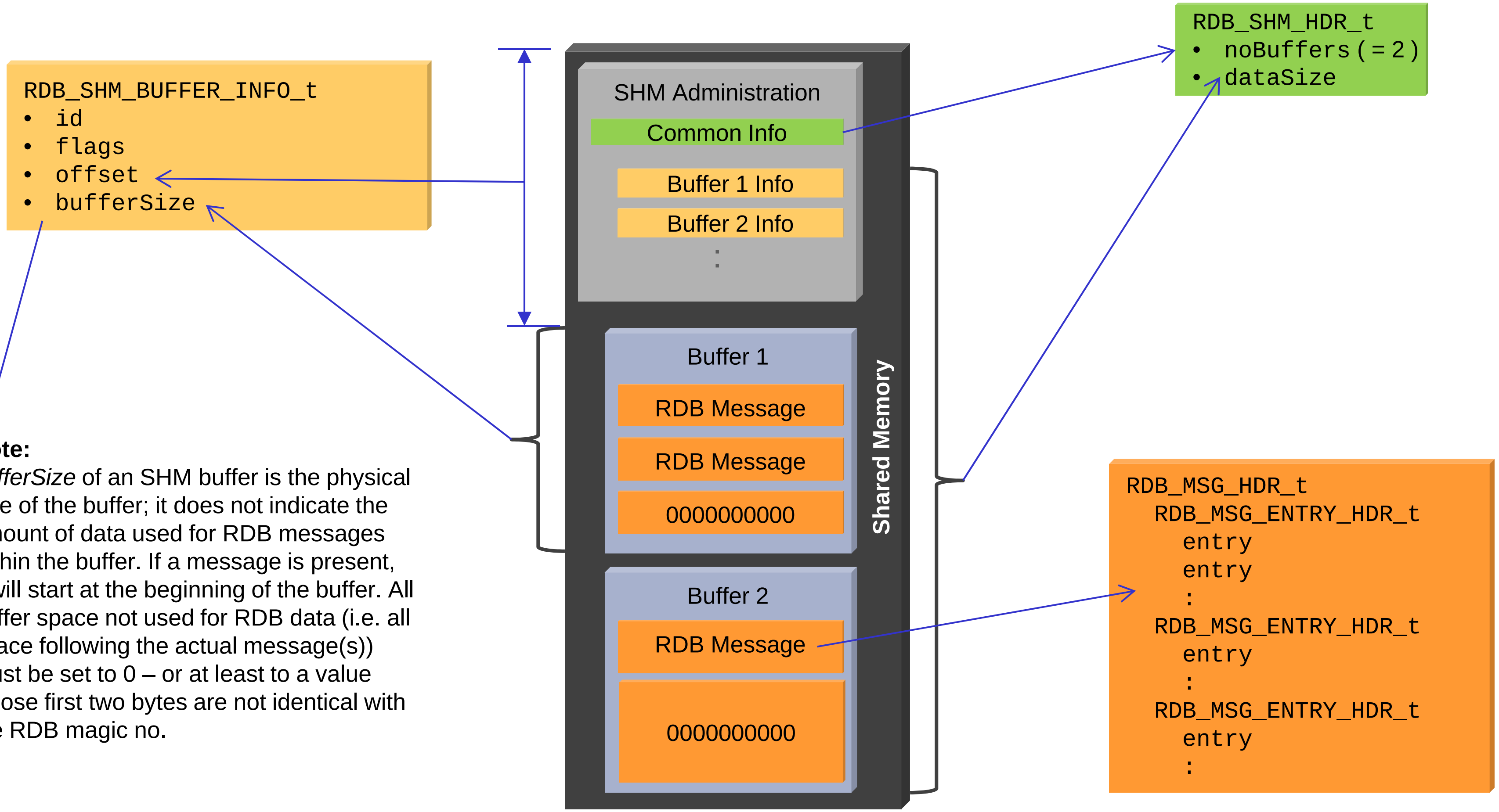


Notes

- Shared Memory connections may be used for image and data transfer
- any data within an SHM is formatted as RDB message
- **IG_IMG_OUT** and (custom) **IG_OUT** are mutually exclusive, i.e. only one of both may exist
- **VIDEO_IN** is used for video streaming via symbols (to be configured in symbol configuration file)
- **IG_CTRL_IN** is used for triggering the IG from an external source
- **IG_IMG_IN** is used for sending vehicle light masks into the IG

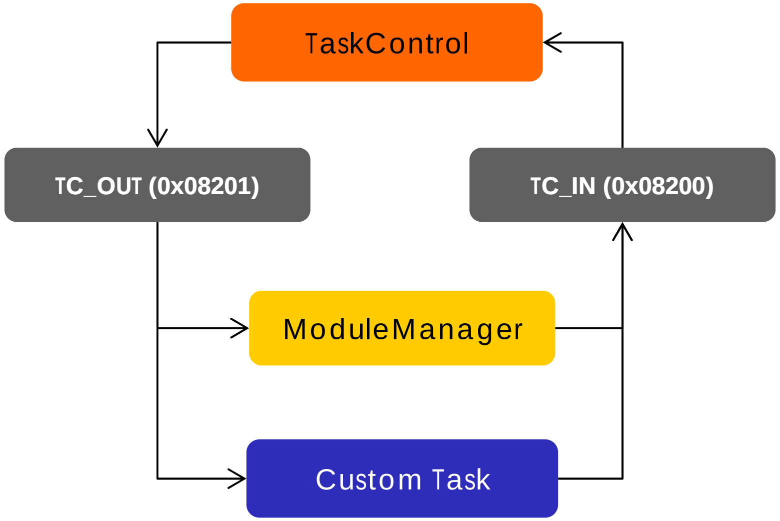
Shared Memory Layout

Example: Dual Buffer SHM (standard use-case for IG buffers)



Note:
bufferSize of an SHM buffer is the physical size of the buffer; it does not indicate the amount of data used for RDB messages within the buffer. If a message is present, it will start at the beginning of the buffer. All buffer space not used for RDB data (i.e. all space following the actual message(s)) must be set to 0 – or at least to a value whose first two bytes are not identical with the RDB magic no.

Shared Memory Access TC IN / OUT



TaskControl

TC_OUT (0x08201)

TC_IN (0x08200)

ModuleManager

Custom Task

taskControl.xml

```
<RDB name="default"
     enable="true"
     portType="TCP"
     imageTransfer="false"/>
<RDB name="shmIn"
     enable="true"
     portType="SHM"
     imageTransfer="false"
     receive="true"
     doubleBuffer="true"/>
<RDB name="shmOut"
     enable="true"
     portType="SHM"
     imageTransfer="false"
     send="true"
     doubleBuffer="true"/>
```

moduleManager.xml

```
<RDB>
   <Port name="RDBraw" type="SHM"
         receive="true" doubleBuffer="true"/>
   <Port name="RDBraw" type="SHM"
         send="true" doubleBuffer="true"/>
</RDB>
```

SCP Commands:

Request a lock flag for the TC input SHM:

```
<Query entity="SHM"> <Sync source="myTask" target="TC_IN"/></Query>
```

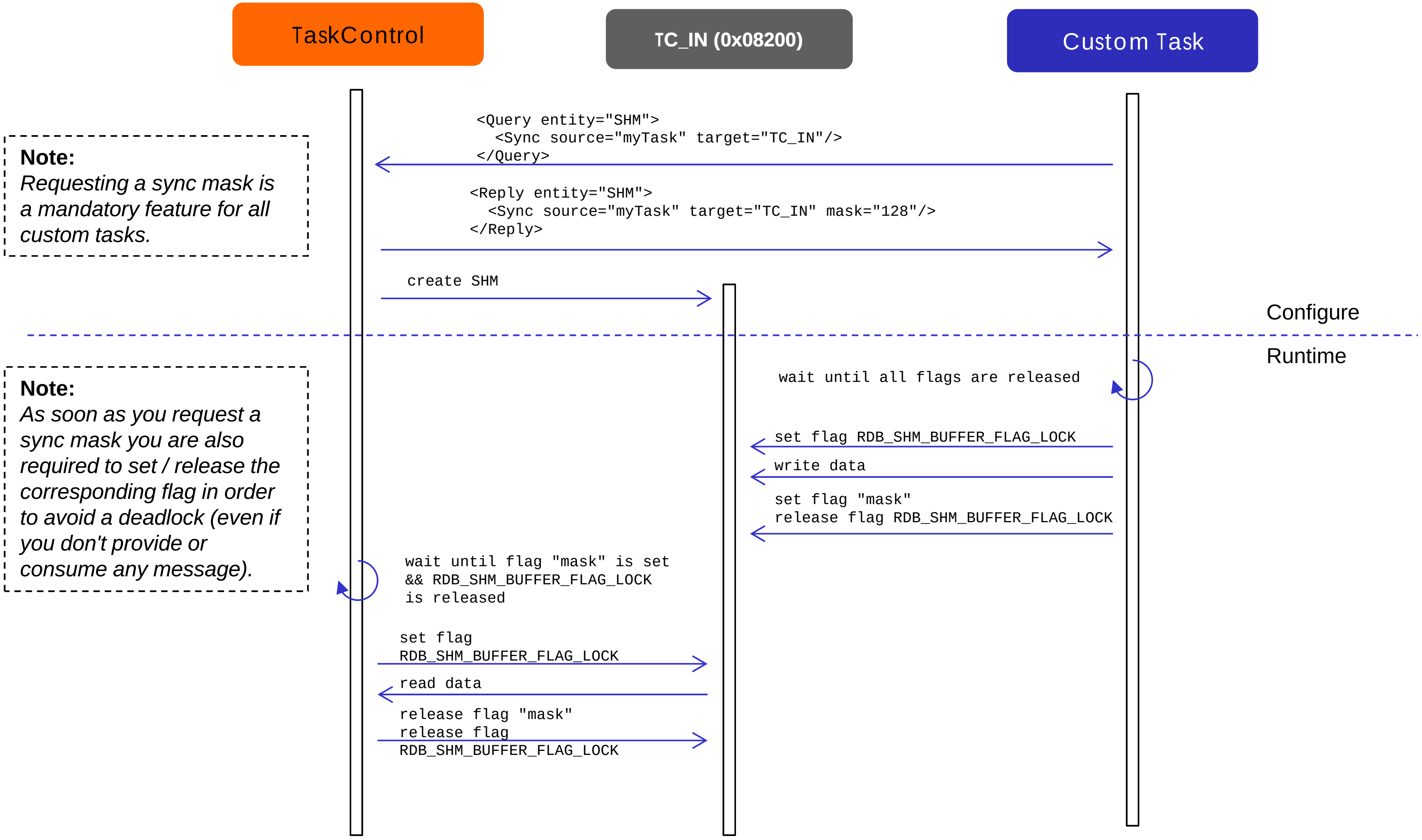
Request a lock flag for the TC output SHM:

```
<Query entity="SHM"> <Sync source="myTask" target="TC_OUT"/></Query>
```

Note: Custom Tasks should always request their individual sync flags with the commands given above!

Note: "DoubleBuffer" settings in TC-configuration and MM-configuration must be consistent!

Shared Memory Access (TC_IN)



TaskControl

TC_IN (0x08200)

Custom Task

Note:
Requesting a sync mask is a mandatory feature for all custom tasks.

```
<Query entity="SHM">
  <Sync source="myTask" target="TC_IN"/>
</Query>
```

```
<Reply entity="SHM">
  <Sync source="myTask" target="TC_IN" mask="128"/>
</Reply>
```

create SHM

Configure

Runtime

Note:
As soon as you request a sync mask you are also required to set / release the corresponding flag in order to avoid a deadlock (even if you don't provide or consume any message).

wait until all flags are released

set flag RDB_SHM_BUFFER_FLAG_LOCK

write data

set flag "mask"
release flag RDB_SHM_BUFFER_FLAG_LOCK

wait until flag "mask" is set
&& RDB_SHM_BUFFER_FLAG_LOCK
is released

set flag
RDB_SHM_BUFFER_FLAG_LOCK

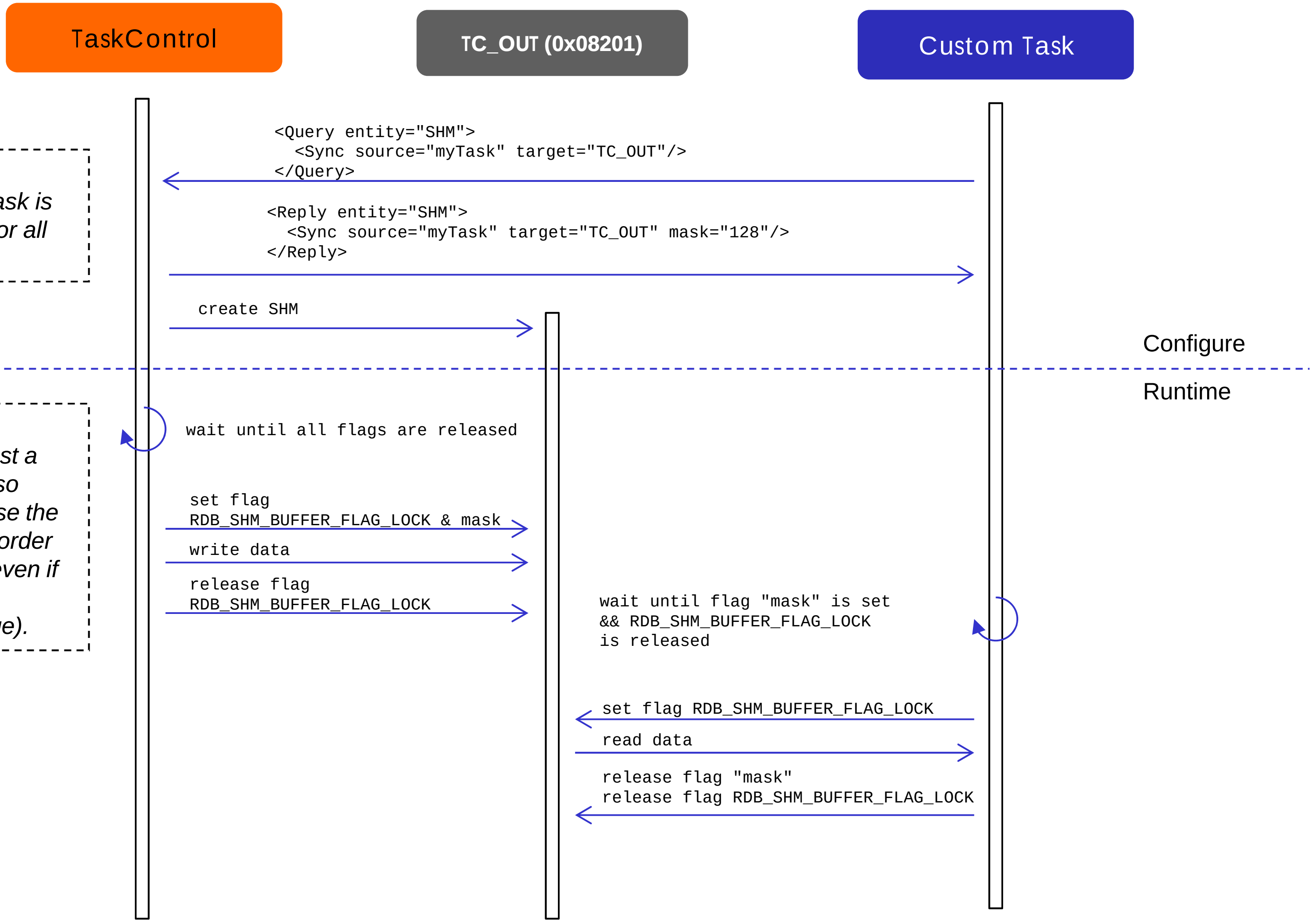
read data

release flag "mask"
release flag
RDB_SHM_BUFFER_FLAG_LOCK

Shared Memory Access (TC_OUT)

Note:
Requesting a sync mask is a mandatory feature for all custom tasks.

Note:
As soon as you request a sync mask you are also required to set / release the corresponding flag in order to avoid a deadlock (even if you don't provide or consume any message).



TaskControl

TC_OUT (0x08201)

Custom Task

```
<Query entity="SHM">
  <Sync source="myTask" target="TC_OUT"/>
</Query>
```

```
<Reply entity="SHM">
  <Sync source="myTask" target="TC_OUT" mask="128"/>
</Reply>
```

create SHM

Configure

Runtime

wait until all flags are released

set flag
RDB_SHM_BUFFER_FLAG_LOCK & mask

write data

release flag
RDB_SHM_BUFFER_FLAG_LOCK

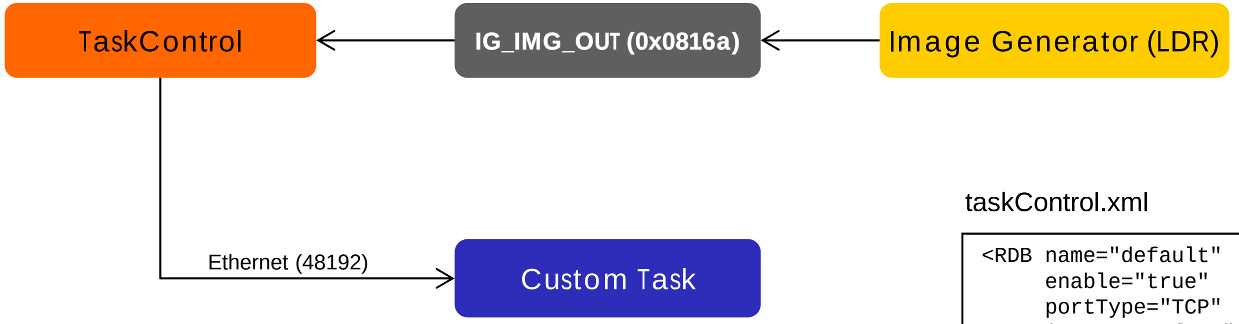
wait until flag "mask" is set
&& RDB_SHM_BUFFER_FLAG_LOCK
is released

set flag RDB_SHM_BUFFER_FLAG_LOCK

read data

release flag "mask"
release flag RDB_SHM_BUFFER_FLAG_LOCK

LDR Video Transfer via TC (Ethernet path)



taskControl.xml

```
<RDB name="default"
     enable="true"
     portType="TCP"
     imageTransfer="true"/>

<Video saveToFile="false"
       buffer="color"
       motionBlurRes="0"
       motionBlurMs="0"
       liveStream="true"
       streamShm="true"
       streamNet="false" />

alternative buffers: colorOnly
                     depth
```

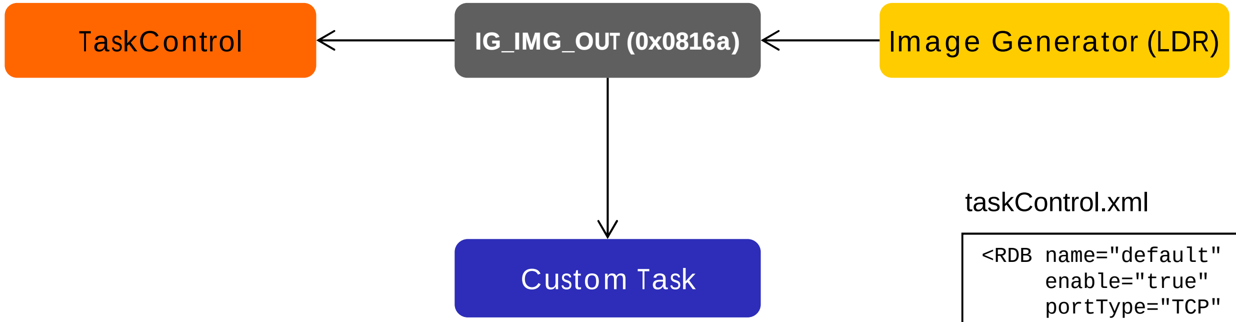
SCP Commands:

```
not applicable
```

Notes:

- TC will block until image from IG is available and has been read by Custom Task
- if `buffer` is set to "`color`", the color AND the depth image will be written to SHM
- if `buffer` is set to "`colorOnly`", only the color image will be written to SHM
- if `buffer` is set to "depth" only the depth image will be written to SHM

LDR Video Transfer via TC (SHM path)



taskControl.xml

```
<RDB name="default"
     enable="true"
     portType="TCP"
     imageTransfer="false"/>

<Video saveToFile="false"
       buffer="color"
       motionBlurRes="0"
       motionBlurMs="0"
       liveStream="true"
       streamShm="true"
       streamNet="false" />

alternative buffers: colorOnly
                     depth
```

SCP Commands:

Request a lock flag for the Image Output of the IG:

```
<Query entity="SHM"> <Sync source="myTask" target="IG_IMG_OUT"/></Query>
```

Notea:

- Custom Task should alway request individual sync flag via SCP
- TC will block until image from IG is available and has been read by Custom Task
- if buffer is set to "color", the color AND the depth image will be written to SHM
- if buffer is set to "colorOnly", only the color image will be written to SHM
- if buffer is set to "depth" only the depth image will be written to SHM

LDR Video Transfer via TC (SHM path)

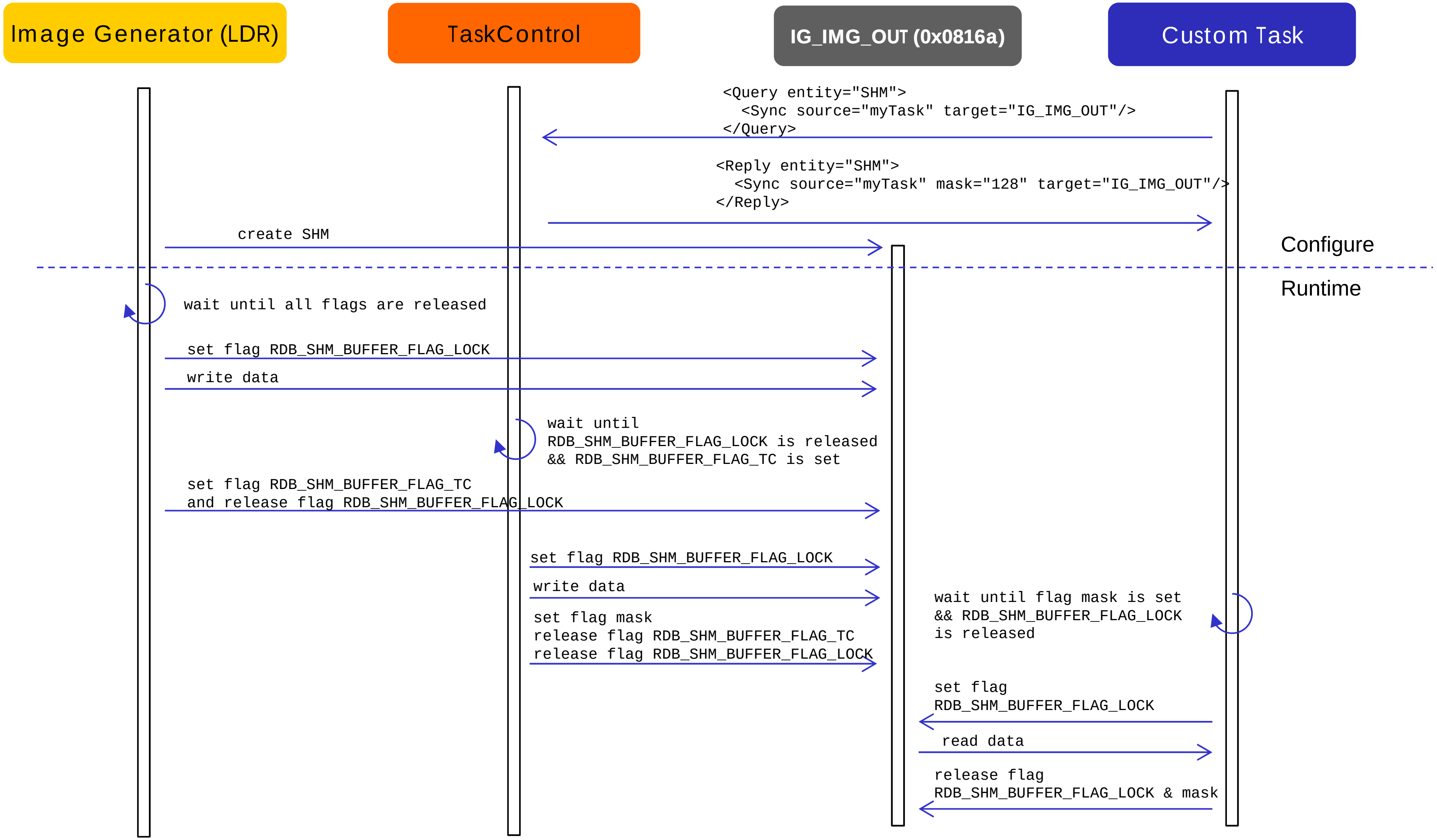


Image Generator (LDR)

TaskControl

IG_IMG_OUT (0x0816a)

Custom Task

```
<Query entity="SHM">
  <Sync source="myTask" target="IG_IMG_OUT"/>
</Query>
```

```
<Reply entity="SHM">
  <Sync source="myTask" mask="128" target="IG_IMG_OUT"/>
</Reply>
```

create SHM

Configure

Runtime

wait until all flags are released

set flag RDB_SHM_BUFFER_FLAG_LOCK

write data

wait until
RDB_SHM_BUFFER_FLAG_LOCK is released
&& RDB_SHM_BUFFER_FLAG_TC is set

set flag RDB_SHM_BUFFER_FLAG_TC
and release flag RDB_SHM_BUFFER_FLAG_LOCK

set flag RDB_SHM_BUFFER_FLAG_LOCK

write data

set flag mask
release flag RDB_SHM_BUFFER_FLAG_TC
release flag RDB_SHM_BUFFER_FLAG_LOCK

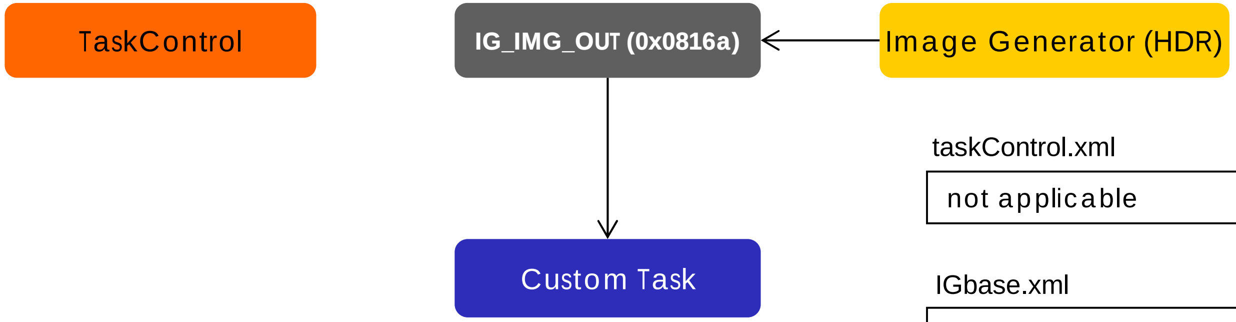
wait until flag mask is set
&& RDB_SHM_BUFFER_FLAG_LOCK
is released

set flag
RDB_SHM_BUFFER_FLAG_LOCK

read data

release flag
RDB_SHM_BUFFER_FLAG_LOCK & mask

HDR Video Transfer (w/o TC)



taskControl.xml

```
not applicable
```

IGbase.xml

```
<Components>
   <RDBInterface     name="MyRDBInterface"
                     printDebugInfo="0"
                     ignoreShmFlags="0">
       <SHMLayoutImage key="0x0816a"
                        releaseFlag="0x0002"
                        freeFlag="0x0000"/>
   </RDBInterface>
   <VCarHDRRenderer name="MyHDRRenderer"
                     writeVideoToShm="1"
                     videoReadbackThreadAffinity="-1"
                     numRenderBuffer="-1"
                     originalSceneBufferFormat="GL_RGB32F"
                     floatBufferFormat="GL_RGB32F"
                     floatBufferScaleFactor="1"
                     outputToFloatTexture="BLOOM_LINEAR"
                     outputToFrameBuffer="BLOOM_REINHARD"
                     enableSRGBOnFrameBuffer="1"
                     frameBufferScaleFactor="0.00024414" />
```

SCP Commands:

```
not applicable
```

Notes:

- Sync mask and free mask have to be defined by user in <SHMLayoutImage .../>
- TC will not block automatically; triggering needs to be done via Custom Task if full frame-sync operation is to be achieved
- only color image will be written to SHM

HDR Video Transfer (w/o TC)

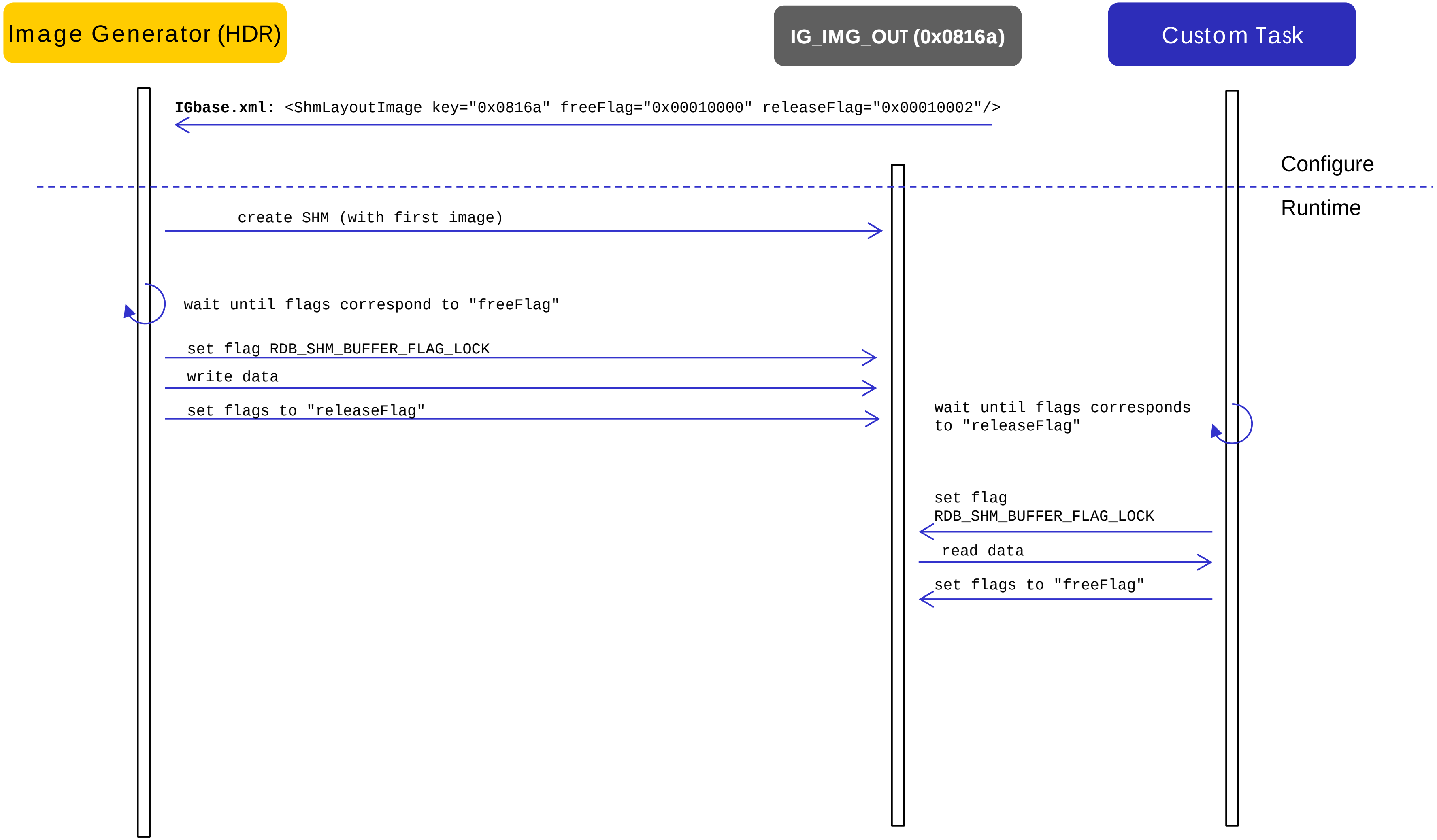
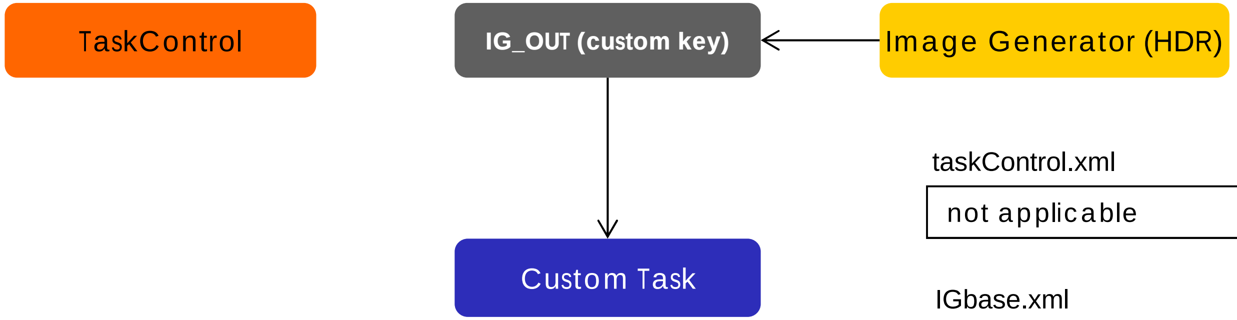


Image Generator (HDR)
IG_IMG_OUT (0x0816a)
Custom Task
IGbase.xml: <ShmLayoutImage key="0x0816a" freeFlag="0x00010000" releaseFlag="0x00010002"/>
Configure
Runtime
create SHM (with first image)
wait until flags correspond to "freeFlag"
set flag RDB_SHM_BUFFER_FLAG_LOCK
write data
set flags to "releaseFlag"
wait until flags corresponds to "releaseFlag"
set flag RDB_SHM_BUFFER_FLAG_LOCK
read data
set flags to "freeFlag"

HDR Video Transfer (custom buffer)



taskControl.xml

```
not applicable
```

IGbase.xml

```
<Components>
   <RDBInterface    name="MyRDBInterface"
                    printDebugInfo="0"
                    ignoreShmFlags="0">
       <SHMLayoutImage key="custom key"
                       releaseFlag="0x0002"
                       freeFlag="0x0000"/>
   </RDBInterface>
   <VCarHDRRenderer name="MyHDRRenderer"
                    writeVideoToShm="1"
                    videoReadbackThreadAffinity="-1"
                    numRenderBuffer="-1"
                    originalSceneBufferFormat="GL_RGB32F"
                    floatBufferFormat="GL_RGB32F"
                    floatBufferScaleFactor="1"
                    outputToFloatTexture="BLOOM_LINEAR"
                    outputToFrameBuffer="BLOOM_REINHARD"
                    enableSRGBOnFrameBuffer="1"
                    frameBufferScaleFactor="0.00024414" />
```

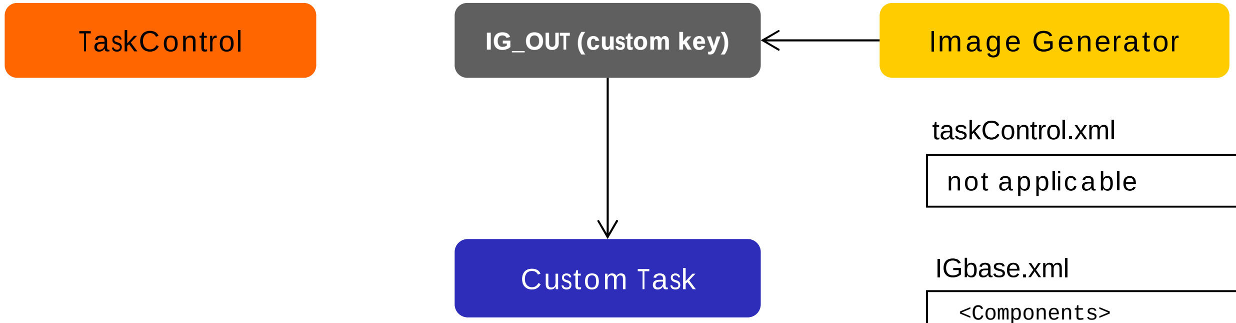
SCP Commands:

```
not applicable
```

Notes:

- Sync mask and free mask have to be defined by user in <SHMLayoutImage .../>
- TC will not block automatically; triggering needs to be done via Custom Task if full frame-sync operation is to be achieved
- only color images will be written to SHM

Video Transfer via PostProcessing (LDR and HDR)



taskControl.xml

not applicable

IGbase.xml

```
<Components>
   <RDBInterface     name="MyRDBInterface"
                     printDebugInfo="0"
                     ignoreShmFlags="0">
       <ShmLayoutImage key="custom key"/>
   </RDBInterface>
   <PostProcessing name="MyPostProcessing"/>
   <PostProcessingPipelineConfigurator
       name="MyPostProcessingPipelineConfigurator">
       <Pipeline hideSceneFromDefaultView="1" >
           <Step type="PPSTextureRect"
                 name="OriginalScene"
                 generateDepthTexture="1">
               <Inputs renderLights="1">
                 <NodeInput inputNo="0" type="scene"/>
               </Inputs>
               <RTT sizeMode="viewport"
                    bufferFormat="GL_RGB8" />
               <SAQ lowerLeftX="0" lowerLeftY="0"
                    width="1.0" height="1.0"
                    isForDebug="0" enableSRGB="0" />
               <Readback readbackThreadAffinity="4">
                   <Image outputSlot="color"/>
                   <Image outputSlot="depth"/>
               </Readback>
           </Step>
       </Pipeline>
   </PostProcessingPipelineConfigurator>
```

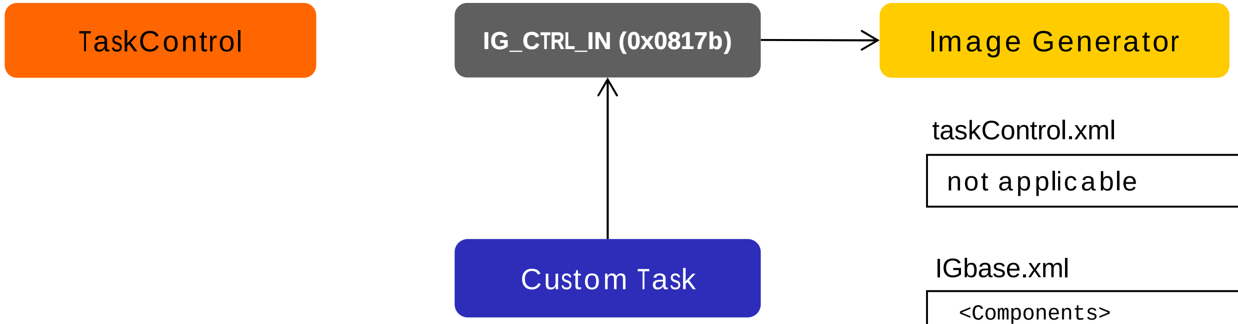
SCP Commands:

not applicable

Notes

- Sync mask and free mask have to be defined by user in <SHMLayoutImage .../>
- TC will not block automatically; triggering needs to be done via Custom Task if full frame-sync operation is to be achieved
- If the tags <Image> are missing within the <Readback> definition, only the color image will be transferred
- This solution is somewhat the "generic" solution for video transfer if the TC is not to be involved; it can be combined with other post-processing steps also

IG render control via SHM (IG_CTRL_IN)



taskControl.xml

not applicable

IGbase.xml

```
<Components>
   <TAKATA …
      updateIGCommandsByShm="1"
      … />
```

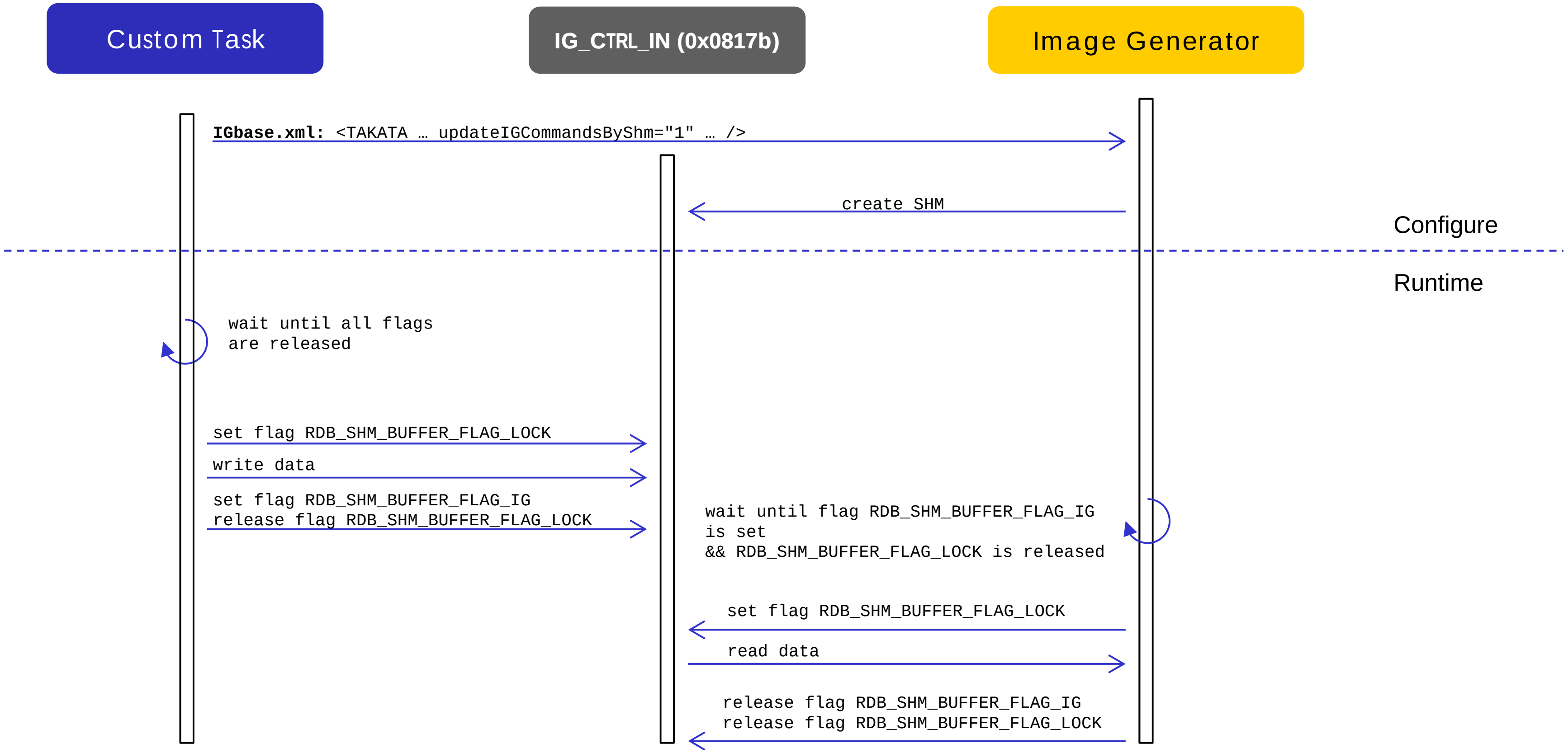
SCP Commands:

not applicable

Notes

- TC may send own data independently to IG
- IG will only update if the respective command is set in the **IG_CTRL_IN** SHM
- IG will be in "continuous" mode per default
- available commands are:
  - RDB_SYNC_CMD_RENDER_CONTINUOUS
  - RDB_SYNC_CMD_RENDER_PAUSE
  - RDB_SYNC_CMD_RENDER_SINGLE_FRAME

IG render control via SHM (IG_CTRL_IN)



Custom Task

IG_CTRL_IN (0x0817b)

Image Generator

IGbase.xml: <TAKATA … updateIGCommandsByShm="1" … />

create SHM

Configure

Runtime

wait until all flags
are released

set flag RDB_SHM_BUFFER_FLAG_LOCK

write data

set flag RDB_SHM_BUFFER_FLAG_IG
release flag RDB_SHM_BUFFER_FLAG_LOCK

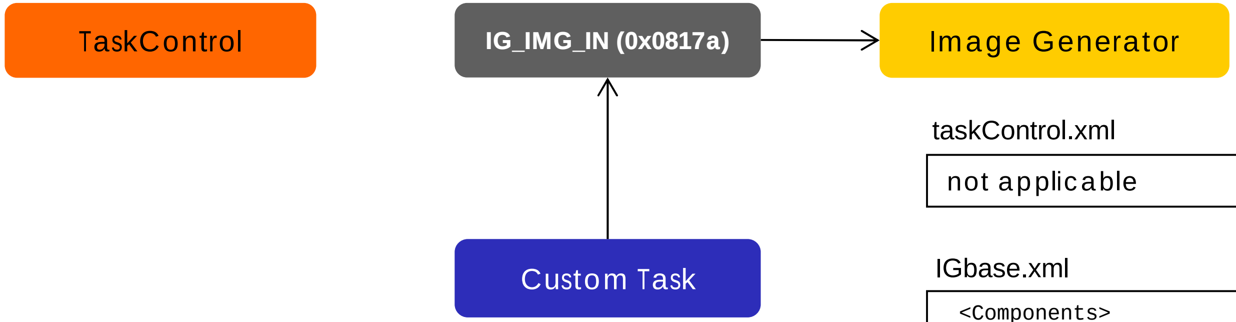
wait until flag RDB_SHM_BUFFER_FLAG_IG
is set
&& RDB_SHM_BUFFER_FLAG_LOCK is released

set flag RDB_SHM_BUFFER_FLAG_LOCK

read data

release flag RDB_SHM_BUFFER_FLAG_IG
release flag RDB_SHM_BUFFER_FLAG_LOCK

IG headlight textures via SHM (IG_IMG_IN)



taskControl.xml

```
not applicable
```

IGbase.xml

```
<Components>
   <TAKATA …
      enableShmReader="1"
      convertLightmapThreadCPUAffinity="4"
      … />
```

SCP Commands:

```
not applicable
```

Notes

- TC may send own data independently to IG
- IG will read light map texture from **IG_IMG_IN** SHM if IG flag has been set