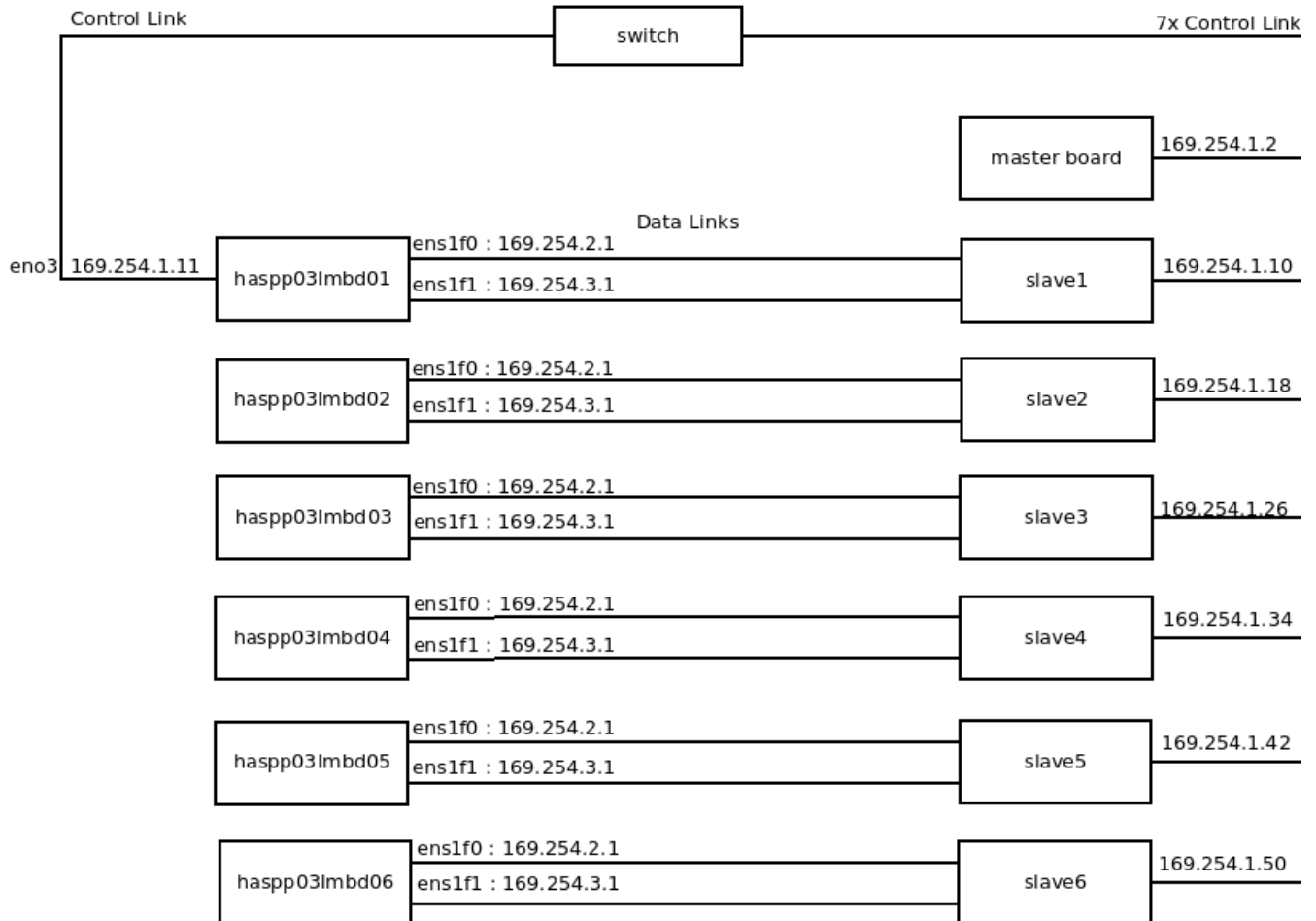


Lambda 4.5M

Layout

Hardware

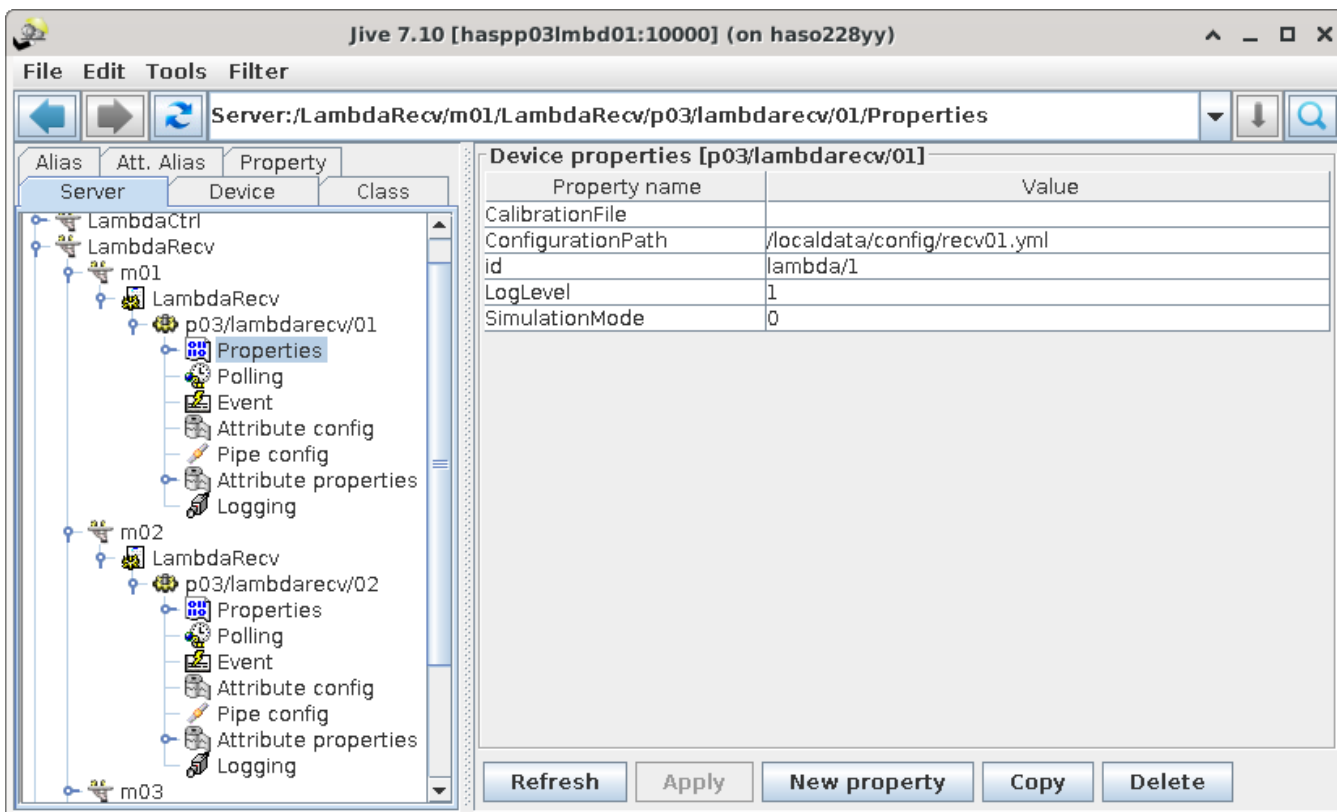
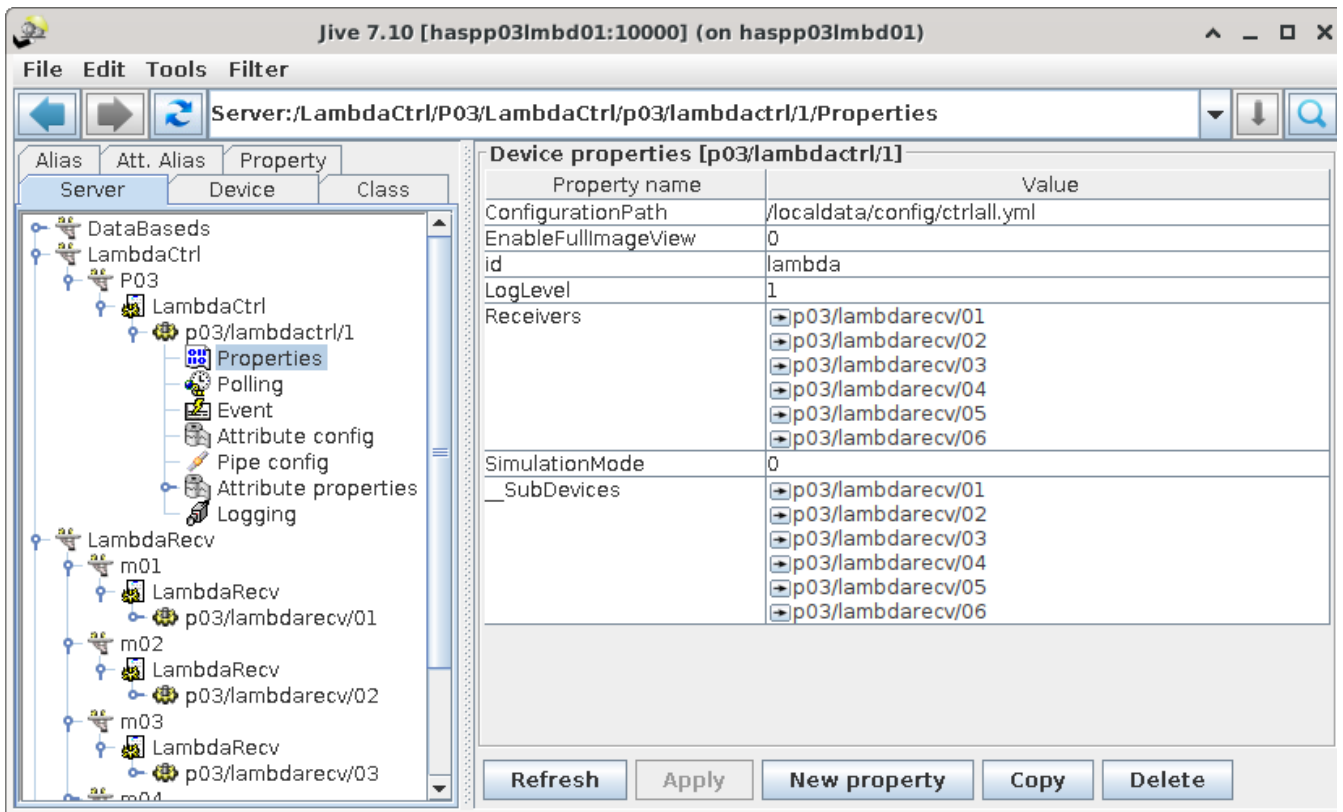
Current the control links of the detector system are connected to switch and the synchronization between detector PCs are done via DESY detector net (192.168.138.*). The data links of each slave module are connected to one detector PC.



Software

On each server, a receiver tango server is running to receive data from data links and save them in storage. The control tango server is running on haspp03lmbd01. This tango server is for

1. communicate with the detectors via eno3
2. communicate with all the receivers



Detector configuration files

The detector configurations are done via yaml file. Each tango server (receiver, control) has it's own file, the file can be found in /localdata/config/ on corresponding detector PCs.

Example control yaml configuration for 6 modules

```
system: # host synchronization for receivers and controllers
  id: SYS
  control:
    ip: 192.168.138.77 # IP of haspp03lmbd01-eno1
  sync:
    master: 192.168.138.77 # IP of haspp03lmbd01
    slaves: [ 192.168.138.77, 192.168.138.78, 192.168.138.79, 192.168.138.80, 192.168.138.93, 192.16
8.138.94 ] # IPs from haspp03lmbd01->06-eno1
    #slaves: [ 192.168.138.77, 192.168.138.78 ]

detectors: # detector related settings
- id: lambda
  type: Lambda
  operation-mode:
    polarity: holes
    bit-depth: 24
  master:
    control:
      ip: 169.254.1.2
      port: 4321
  modules:
    - directory: Module_2019-008_Si # module 1
      control:
        ip: 169.254.1.10
        port: 4321
        position: { x: 0.0, y: 1317.0, z: 0.0 }
    - directory: Module_2018-005_Si # module 2
      control:
        ip: 169.254.1.18
        port: 4321
        position: { x: 1.0, y: 1975.0, z: 0.0 }
    - directory: Module_2018-014_Si # module 3
      control:
        ip: 169.254.1.26
        port: 4321
        position: { x: 1587.0, y: 0.0, z: 0.0 }
    - directory: Module_2019-005_Si # module 4
      control:
        ip: 169.254.1.34
        port: 4321
        position: { x: 1586.0, y: 658.0, z: 0.0 }
    - directory: Module_2019-012_Si # module 5
      control:
        ip: 169.254.1.42
        port: 4321
        position: { x: 1586.0, y: 1315.0, z: 0.0 }
    - directory: Module_2019-014_Si # module 6
      control:
        ip: 169.254.1.50
        port: 4321
        position: { x: 1583.0, y: 1978.0, z: 0.0 }
```

Example receiver yaml configuration for receiver 1

```
system: # host synchronization for receivers and controllers
  id: SYS
  control:
    ip: 192.168.138.77          # IP of haspp03lmbd01
  sync:
    master: 192.168.138.77    # IP of haspp03lmbd01

receivers:
- ref: lambda/1
  type: Lambda
  numa: 0 # if this is set only maximal half of RAM in ram-use for receiver can be used
  decoding:
    ram-use: 30GB # ram configuration
    threads: 12
  compression: # compression
    compressor: zlib
    level: 2
    threads: 8
  udp-buffer: 256000000
  links: # data links
    - ip: 169.254.3.1
      mac: f8:f2:1e:87:b0:e1
    - ip: 169.254.2.1
      mac: f8:f2:1e:87:b0:e0
```

Run the detector

Configurations

Before running the detector, some configurations might need to be changed depending on the use case.

Disable specific module (use disabling module 2 as an example here):

Specific module can be removed during initialization.

1. In the corresponding yaml file which can found in Properties of LambdaCtrl tango server in Jive, one may comment out the module entry under detectors section, so that module will not be connected by software.

Comment out the following part in ctrlall.yml in /localdata/config

Disable module 2

```
- directory: Module_2018-005_Si # module 2
  control:
    ip: 169.254.1.18
    port: 4321
    position: { x: 1.0, y: 1975.0, z: 0.0 }
```

2. Remove the IP of the PC where this module is connected from slaves list in sync section

```
slaves: [ 192.168.138.77, 192.168.138.78, 192.168.138.79, 192.168.138.80, 192.168.138.93, 192.168.138.94 ] # IPs from haspp03lmbd01->06-en01
```

3. Remove "p03/lambdarecv/02" the Receivers property in LambdaCtrl tango server and remove that receiver too.
4. Comment out the line which starts the receiver tango server (ssh -n haspp03lmbd03 /usr/lib/tango/server/LambdaRecv m03 -v2 &) in startallambda.sh.

Change buffer size:

The buffer size decides how much RAM needs to be reserved for receiver during data acquisition. This shows how many images can be taken for each acquisition. This can be checked by FreeBuffer attribute in the receiver tango server.

NOTE: allocating large amount of RAM can last long (current benchmark shows the memory allocation speed is ~1.2GB/s)

Start/Stop

```
#start lambda

$ startalllambda.sh

wait until the following is shown (example), the whole procedure may take some time:module id is:w502-A06,
CRN,0x7db1f616

firmware version:01-80-00 [ctrl=v0, data=v2, feat=0x02]
library version:1.2.1
p03/lambdarecv/01
p03/lambdarecv/02
p03/lambdarecv/03
p03/lambdarecv/04
p03/lambdarecv/05
p03/lambdarecv/06
Ready to accept request

#stop lambda

$ killalllambda.sh
```

Control the detector

Once the tango server is ready, the detector can be controlled via LambdaCtrl tango server.

Be aware that changing some attributes(EnergyThreshold, OperatingMode) may take some time (more than 3 seconds) and it's recommended to check the status of the tango server when changing those attributes.

Live Image

At the moment, there are two sources of live images, one from receiver side(haspp03lmbd01:10000/p03/lambdarecv/#/LiveLastImageData), which is taken from data stream written to Nexus file, this live image is only for each slave module. The other one from control tango server(haspp03lmbd01:10000/p03/lambdactrl/1/LiveLastImageData) stitches the slave module images into one large image with translation information.

Both live image has refresh rate 1 frame/second by default. Lavue can be used to view the live images. ATKpanel is not recommended.

NOTE: The stitched image is relatively large. If one uses this, the ATKpanel should not be used, otherwise it causes Java heap memory error.

By default, the stitched image update is disabled, this is controlled by EnableFullImageView. If this is set true, the stitched live image is available in LiveLastImageData attribute of the LambdaCtrl tango server. However DO NOT use this together with ATKpanel as mentioned above.