

# LVV-T2290

March 8, 2022

## 1 Slew, Track and Image taking with ComCam

This notebook is used for the level 3 integration tests from test plan LVV-P81 (<https://jira.lsstcorp.org/secure/Tests.jspa#/testPlan/LVV-P81>) as part of test cycle LVV-C176 (<https://jira.lsstcorp.org/secure/Tests.jspa#/testCycle/LVV-C176>). The following tests are currently run as part of this notebook:

- LVV-T2290 (<https://jira.lsstcorp.org/secure/Tests.jspa#/testCase/LVV-T2290>)

Execution steps are separated by horizontal lines. Upon completion, save the notebook and its output as a pdf file to be attached to the test execution in JIRA.

Last executed by E. Dennihy 20210928

---

Run the setup.ipynb notebook to bring all components up and in their enabled position. Check Chronograph.

---

Bring ComCom online and transition it to EnabledState. Check Chronograph.

```
[1]: %load_ext autoreload
      %autoreload 2
```

```
[2]: import rubin_jupyter_utils.lab.notebook as nb
      nb.utils.get_node()
```

```
/tmp/ipykernel_49447/1665379685.py:2: DeprecationWarning: Call to deprecated
function (or staticmethod) get_node. (Please use lsst.rsp.get_node())
      nb.utils.get_node()
```

```
[2]: 'yagan06'
```

```
[3]: import os
      import sys
      import asyncio
      import logging

      import pandas as pd
      import numpy as np
```

```

from matplotlib import pyplot as plt

from lsst.ts import salobj
from lsst.ts.observatory.control.maintel import MTCS, ComCam
from lsst.ts.observatory.control import RotType

```

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

```
[4]: logging.basicConfig(format="%(name)s: %(message)s", level=logging.DEBUG)
```

```
[5]: log = logging.getLogger("setup")
log.level = logging.DEBUG
```

```
[6]: domain = salobj.Domain()
```

```
[7]: mtcs = MTCS(domain=domain, log=log)
mtcs.set_rem_loglevel(40)
```

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

```
[8]: await mtcs.start_task
```

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

```
[8]: [None, None, None, None, None, None, None, None, None]
```

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

```
[9]: comcam = ComCam(domain=domain, log=log)
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
[10]: comcam.set_rem_loglevel(40)
```

```
[11]: await comcam.start_task
```

```
[11]: [None, None, None]
```

```
[12]: await comcam.standby()
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
[13]: await comcam.enable()
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

---

Find four targets separated by  $5^\circ$  in azimuth and elevation in a square pattern around  $az = 120^\circ$  and  $el = 60^\circ$  and rotator angle at PhysicalSky and  $1.8^\circ$ .

At this position, the rotator stays within a couple of degrees of its initial position. This is because the CCW is not running (MTmount in simulation mode).

```
target_1 -> az = 117.5°, el = 57.5°
target_2 -> az = 122.5°, el = 57.5°
target_3 -> az = 122.5°, el = 62.5°
target_4 -> az = 117.5°, el = 62.5°
```

```
[25]: target_1 = await mtcs.find_target(az=117.5, el=57.5, mag_limit=8)
      await asyncio.sleep(15)
      target_2 = await mtcs.find_target(az=122.5, el=57.5, mag_limit=8)
      await asyncio.sleep(15)
      target_3 = await mtcs.find_target(az=122.5, el=62.5, mag_limit=8)
      await asyncio.sleep(15)
      target_4 = await mtcs.find_target(az=117.5, el=62.5, mag_limit=8)

      print(f"Target 1: {target_1}\n"
            f"Target 2: {target_2}\n"
            f"Target 3: {target_3}\n"
            f"Target 4: {target_4}\n")
```

```
Target 1: HD 20983
Target 2: HD 20723
Target 3: HD 18169
Target 4: HD 18264
```

---

Slew to target 1:

```
[26]: await mtcs.slew_object(target_1, rot_type=RotType.PhysicalSky, rot=1.9)
```

```
<IPython.core.display.HTML object>
<IPython.core.display.HTML object>
<IPython.core.display.HTML object>
<IPython.core.display.HTML object>
<IPython.core.display.HTML object>
<IPython.core.display.HTML object>
<IPython.core.display.HTML object>
<IPython.core.display.HTML object>
<IPython.core.display.HTML object>
<IPython.core.display.HTML object>
<IPython.core.display.HTML object>
<IPython.core.display.HTML object>
```

[illegible]

<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>

---

Once on target\_1 and tracking, take an image with ComCam

```
[27]: exp1 = await comcam.take_object(15)  
      print(f"Target 1 exposure: {exp1}")
```

<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
Target 1 exposure: [2022030800007]

---

Slew to target\_2:

```
[28]: await mtcs.slew_object(target_2, rot_type=RotType.PhysicalSky, rot=1.9)
```

<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>



```
exp2 = await comcam.take_object(15)
print(f"Target 1 exposure: {exp2}")
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

Target 1 exposure: [2022030800008]

```
await mtcs.slew_object(target_3, rot_type=RotType.PhysicalSky, rot=1.9)
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```



[illegible]

Once on target\_3 and tracking, take an image with ComCam

```
[31]: exp3 = await comcam.take_object(15)
print(f"Target 1 exposure: {exp3}")
```

```
<IPython.core.display.HTML object>
<IPython.core.display.HTML object>
<IPython.core.display.HTML object>
Target 1 exposure: [2022030800009]
```

Slew to target 4

```
[32]: await mtcs.slew_object(target_4, rot_type=RotType.PhysicalSky, rot=1.9)
```

[illegible]

<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>

---

Once on target\_4 and tracking, take an image with ComCam

```
[33]: exp4 = await comcam.take_object(15)
      print(f"Target 1 exposure: {exp4}")
```

<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
<IPython.core.display.HTML object>  
Target 1 exposure: [2022030800010]

---

Stop tracking to prevent hitting the Rotator soft limit.

```
[34]: await mtcs.stop_tracking()
```

<IPython.core.display.HTML object>

---

Use ComCam recent images CCS to ensure that the images were taken (<http://ccs.lsst.org/RecentImages/comcam.html>).

---

Query the butler to verify that the images are there and check the metadata. This step must be verified using a separate notebook.

---

### Wrap Up and Shut Down

This cell is not currently included as part of the test execution, but included here as needed to shutdown the systems

```
[ ]: await mtcs.set_state(salobj.State.STANDBY, components=["mtaos"])
```

```
[ ]: await mtcs.lower_m1m3()
```

```
[ ]: await mtcs.set_state(salobj.State.STANDBY, components=["mtm1m3"])
```

```
[ ]: await mtcs.set_state(salobj.State.STANDBY, components=["mtm2"])
```

```
[ ]: await mtcs.set_state(salobj.State.STANDBY, components=["mthexapod_1"])
```

```
[ ]: await mtcs.set_state(salobj.State.STANDBY, components=["mthexapod_2"])
```

```
[ ]: await mtcs.standby()
```

```
[ ]: await comcam.standby()
```