

LVV-T2290

March 2, 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 tranistion 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_50545/1665379685.py:2: DeprecationWarning: Call to deprecated
function (or staticmethod) get_node. (Please use lsst.rsp.get_node())
      nb.utils.get_node()
```

```
[2]: 'yagan07'
```

```
[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>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

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

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

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

<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>

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

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

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

```
[12]: 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° in azimuth and elevation in a square pattern around $az = 120^\circ$ and $el = 60^\circ$ and rotator angle at PhysicalSky and 1.8° .

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 -> $az = 117.5^\circ$, $el = 57.5^\circ$

target_2 -> $az = 122.5^\circ$, $el = 57.5^\circ$

target_3 -> $az = 122.5^\circ$, $el = 62.5^\circ$

target_4 -> $az = 117.5^\circ$, $el = 62.5^\circ$

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

Target 1: HD 10038 Target 2: HD 10027 Target 3: HD 7246 Target 4: HD 7221

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

4

[illegible]

Once on target_1 and tracking, take an image with ComCam

```
[20]: 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: [2022030200001]
```

```
await mtcs.slew_object(target_2, 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>

Once on target_2 and tracking, take an image with ComCam

```
[22]: 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: [2022030200002]

Slew to target_3

```
[23]: 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>

```

```

-----
TimeoutError                                Traceback (most recent call last)
Input In [23], in <module>
----> 1 await mtcs.slew_object(target_3, rot_type=RotType.PhysicalSky, rot=1.9)

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/base_tcs.py:415, in BaseTCS.slew_object(self, name, rot, rot_type, dra,
ddec, offset_x, offset_y, az_wrap_strategy, time_on_target, slew_timeout)
    411 object_table = self.object_list_get(name)
    413 self.log.info(f"Slewing to {name}: {object_table['RA']}
    ↪{object_table['DEC']}")
--> 415 await self.slew_icrs(
    416     ra=object_table["RA"],
    417     dec=object_table["DEC"],
    418     rot=rot,
    419     rot_type=rot_type,
    420     target_name=name,
    421     dra=dra,
    422     ddec=ddec,
    423     offset_x=offset_x,

```

```

424     offset_y=offset_y,
425     az_wrap_strategy=az_wrap_strategy,
426     time_on_target=time_on_target,
427     slew_timeout=slew_timeout,
428 )

```

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/base_tcs.py:589, in BaseTCS.slew_icrs(self, ra, dec, rot, rot_type,
target_name, dra, ddec, offset_x, offset_y, az_wrap_strategy, time_on_target,
slew_timeout, stop_before_slew, wait_settle)

```

584     valid_rottypes = ", ".join(repr(rt) for rt in RotType)
585     raise RuntimeError(
586         f"Unrecognized rottype {rot_type}. Should be one of
{valid_rottypes}"
587     )
--> 589 await self.slew(
590     radec_icrs.ra.hour,
591     radec_icrs.dec.deg,
592     rotPA=rot_angle.deg,
593     target_name=target_name,
594     frame=self.CoordFrame.ICRS,
595     epoch=2000,
596     equinox=2000,
597     parallax=0,
598     pmRA=0,
599     pmDec=0,
600     rv=0,
601     dRA=dra,
602     dDec=ddec,
603     rot_frame=rot_frame,
604     rot_track_frame=rot_track_frame,
605     az_wrap_strategy=az_wrap_strategy,
606     time_on_target=time_on_target,
607     rot_mode=self.RotMode.FIELD,
608     slew_timeout=slew_timeout,
609     stop_before_slew=stop_before_slew,
610     wait_settle=wait_settle,
611     offset_x=offset_x,
612     offset_y=offset_y,
613 )
615 return radec_icrs, rot_angle

```

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/base_tcs.py:761, in BaseTCS.slew(self, ra, dec, rotPA, target_name,
frame, epoch, equinox, parallax, pmRA, pmDec, rv, dRA, dDec, rot_frame,
rot_track_frame, rot_mode, az_wrap_strategy, time_on_target, slew_timeout,
stop_before_slew, wait_settle, offset_x, offset_y)

```

754 getattr(self.rem, self.ptg_name).cmd_poriginOffset.set(
755     dx=offset_x * self.plate_scale,

```

```

756     dy=offset_y * self.plate_scale,
757     num=0,
758 )
760 try:
--> 761     await self._slew_to(
762         getattr(self.rem, self.ptg_name).cmd_raDecTarget,
763         slew_timeout=slew_timeout,
764         offset_cmd=getattr(self.rem, self.ptg_name).cmd_poriginOffset,
765         stop_before_slew=stop_before_slew,
766         wait_settle=wait_settle,
767     )
768 except salobj.AckError as ack_err:
769     self.log.error(
770         f"Command to track target {target_name} rejected: {ack_err}.
↪ackcmd.result}"
771     )

```

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
↪control/maintel/mtcs.py:292, in MTCS._slew_to(self, slew_cmd, slew_timeout, ↪
↪offset_cmd, stop_before_slew, wait_settle, check)

```

287         getattr(self.rem, comp).evt_summaryState.flush()
288         self.scheduled_coro.append(
289             asyncio.create_task(self.check_component_state(comp))
290         )
--> 292 await self.process_as_completed(self.scheduled_coro)

```

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
↪control/remote_group.py:1157, in RemoteGroup.process_as_completed(self, tasks)

```

1155 except Exception as e:
1156     await self.cancel_not_done(tasks)
-> 1157     raise e
1158 else:
1159     await self.cancel_not_done(tasks)

```

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
↪control/remote_group.py:1154, in RemoteGroup.process_as_completed(self, tasks)

```

1152 for res in asyncio.as_completed(tasks):
1153     try:
-> 1154         ret_val = await res
1155     except Exception as e:
1156         await self.cancel_not_done(tasks)

```

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-2.0
↪0/lib/python3.8/asyncio/tasks.py:619, in as_completed.<locals>._wait_for_one(

```

616 if f is None:
617     # Dummy value from _on_timeout().
618     raise exceptions.TimeoutError
--> 619 return f.result()

```

```

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
↳control/maintel/mtcs.py:344, in MTCS.wait_for_inposition(self, timeout,
↳cmd_ack, wait_settle, check)
    339     status.append(
    340         asyncio.create_task(self.wait_for_rotator_inposition(timeout,
↳cmd_ack))
    341     )
    343     ret_val = ""
--> 344     for s in await asyncio.gather(*status):
    345         ret_val += f"{s!r}"
    347     return ret_val

```

```

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
↳control/maintel/mtcs.py:537, in MTCS.wait_for_rotator_inposition(self,
↳timeout, cmd_ack, wait_settle)
    533     self.log.debug("Wait for rotator in position event.")
    535     while True:
--> 537         in_position = await self.rem.mtrotator.evt_inPosition.next(
    538             flush=False, timeout=timeout
    539         )
    541         # make sure timestamp of event is after command was acknowledged.
    542         if (
    543             cmd_ack is not None
    544             and in_position.private_sndStamp < cmd_ack.private_sndStamp
    545         ):

```

```

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-2.0
↳0/lib/python3.8/site-packages/lsst/ts/salobj/topics/read_topic.py:677, in
↳ReadTopic.next(self, flush, timeout)
    675     if flush:
    676         self.flush()
--> 677     return await self._next(timeout=timeout)

```

```

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-2.0
↳0/lib/python3.8/site-packages/lsst/ts/salobj/topics/read_topic.py:691, in
↳ReadTopic._next(self, timeout)
    689     if self._next_task.done():
    690         self._next_task = asyncio.Future()
--> 691     return await asyncio.wait_for(self._next_task, timeout=timeout)

```

```

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-2.0
↳0/lib/python3.8/asyncio/tasks.py:501, in wait_for(fut, timeout, loop)
    497         # We must ensure that the task is not running
    498         # after wait_for() returns.
    499         # See https://bugs.python.org/issue32751
    500         await _cancel_and_wait(fut, loop=loop)
--> 501         raise exceptions.TimeoutError()
    502     finally:

```

```
503     timeout_handle.cancel()
```

```
TimeoutError:
```

Once on target_3 and tracking, take an image with ComCam

```
[24]: 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: [2022030200003]
```

Slew to target 4

```
[25]: await mtcs.slew_object(target_4, 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>
```

[illegible]

[illegible]

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

```
-----
TimeoutError                                Traceback (most recent call last)
Input In [25], in <module>
----> 1 await mtcs.slew_object(target_4, rot_type=RotType.PhysicalSky, rot=1.9)

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/base_tcs.py:415, in BaseTCS.slew_object(self, name, rot, rot_type,
dra, ddec, offset_x, offset_y, az_wrap_strategy, time_on_target, slew_timeout)
    411 object_table = self.object_list_get(name)
    413 self.log.info(f"Slewing to {name}: {object_table['RA']}
    ↪{object_table['DEC']}")
--> 415 await self.slew_icrs(
    416     ra=object_table["RA"],
    417     dec=object_table["DEC"],
    418     rot=rot,
    419     rot_type=rot_type,
    420     target_name=name,
    421     dra=dra,
    422     ddec=ddec,
    423     offset_x=offset_x,
    424     offset_y=offset_y,
    425     az_wrap_strategy=az_wrap_strategy,
    426     time_on_target=time_on_target,
    427     slew_timeout=slew_timeout,
    428 )

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
control/base_tcs.py:589, in BaseTCS.slew_icrs(self, ra, dec, rot, rot_type,
target_name, dra, ddec, offset_x, offset_y, az_wrap_strategy, time_on_target,
slew_timeout, stop_before_slew, wait_settle)
    584 valid_rottypes = ", ".join(repr(rt) for rt in RotType)
    585 raise RuntimeError(
    586     f"Unrecognized rottype {rot_type}. Should be one of
    ↪{valid_rottypes}"
    587 )
--> 589 await self.slew(
    590     radec_icrs.ra.hour,
    591     radec_icrs.dec.deg,
    592     rotPA=rot_angle.deg,
    593     target_name=target_name,
    594     frame=self.CoordFrame.ICRS,
    595     epoch=2000,
    596     equinox=2000,
    597     parallax=0,
    598     pmRA=0,
```

```

599     pmDec=0,
600     rv=0,
601     dRA=dra,
602     dDec=ddec,
603     rot_frame=rot_frame,
604     rot_track_frame=rot_track_frame,
605     az_wrap_strategy=az_wrap_strategy,
606     time_on_target=time_on_target,
607     rot_mode=self.RotMode.FIELD,
608     slew_timeout=slew_timeout,
609     stop_before_slew=stop_before_slew,
610     wait_settle=wait_settle,
611     offset_x=offset_x,
612     offset_y=offset_y,
613 )
615 return radec_icrs, rot_angle

```

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
 ↪control/base_tcs.py:761, in BaseTCS.slew(self, ra, dec, rotPA, target_name,
 ↪frame, epoch, equinox, parallax, pmRA, pmDec, rv, dRA, dDec, rot_frame,
 ↪rot_track_frame, rot_mode, az_wrap_strategy, time_on_target, slew_timeout,
 ↪stop_before_slew, wait_settle, offset_x, offset_y)

```

754     getattr(self.rem, self.ptg_name).cmd_poriginOffset.set(
755         dx=offset_x * self.plate_scale,
756         dy=offset_y * self.plate_scale,
757         num=0,
758     )
760     try:
--> 761         await self._slew_to(
762             getattr(self.rem, self.ptg_name).cmd_raDecTarget,
763             slew_timeout=slew_timeout,
764             offset_cmd=getattr(self.rem, self.ptg_name).cmd_poriginOffset,
765             stop_before_slew=stop_before_slew,
766             wait_settle=wait_settle,
767         )
768     except salobj.AckError as ack_err:
769         self.log.error(
770             f"Command to track target {target_name} rejected: {ack_err}.
↪ackcmd.result}"
771         )

```

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
 ↪control/maintel/mtcs.py:292, in MTCS._slew_to(self, slew_cmd, slew_timeout,
 ↪offset_cmd, stop_before_slew, wait_settle, check)

```

287         getattr(self.rem, comp).evt_summaryState.flush()
288         self.scheduled_coro.append(
289             asyncio.create_task(self.check_component_state(comp))
290         )
--> 292 await self.process_as_completed(self.scheduled_coro)

```

```

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
↳control/remote_group.py:1157, in RemoteGroup.process_as_completed(self, tasks)
    1155 except Exception as e:
    1156     await self.cancel_not_done(tasks)
-> 1157     raise e
    1158 else:
    1159     await self.cancel_not_done(tasks)

```

```

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
↳control/remote_group.py:1154, in RemoteGroup.process_as_completed(self, tasks)
    1152 for res in asyncio.as_completed(tasks):
    1153     try:
-> 1154         ret_val = await res
    1155     except Exception as e:
    1156         await self.cancel_not_done(tasks)

```

```

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-2.0
↳0/lib/python3.8/asyncio/tasks.py:619, in as_completed.<locals>._wait_for_one(
    616 if f is None:
    617     # Dummy value from _on_timeout().
    618     raise exceptions.TimeoutError
--> 619 return f.result()

```

```

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
↳control/maintel/mtcs.py:344, in MTCS.wait_for_inposition(self, timeout,
↳cmd_ack, wait_settle, check)
    339 status.append(
    340     asyncio.create_task(self.wait_for_rotator_inposition(timeout,
↳cmd_ack))
    341 )
    343 ret_val = ""
-> 344 for s in await asyncio.gather(*status):
    345     ret_val += f"{s!r}"
    347 return ret_val

```

```

File ~/auto-op-env-packages/ts_observatory_control/python/lsst/ts/observatory/
↳control/maintel/mtcs.py:537, in MTCS.wait_for_rotator_inposition(self,
↳timeout, cmd_ack, wait_settle)
    533 self.log.debug("Wait for rotator in position event.")
    535 while True:
-> 537     in_position = await self.rem.mtrotator.evt_inPosition.next(
    538         flush=False, timeout=timeout
    539     )
    541     # make sure timestamp of event is after command was acknowledged.
    542     if (
    543         cmd_ack is not None
    544         and in_position.private_sndStamp < cmd_ack.private_sndStamp

```

```

545     ):

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-2.0
↳0/lib/python3.8/site-packages/lsst/ts/salobj/topics/read_topic.py:677, in
↳ReadTopic.next(self, flush, timeout)
    675 if flush:
    676     self.flush()
--> 677 return await self._next(timeout=timeout)

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-2.0
↳0/lib/python3.8/site-packages/lsst/ts/salobj/topics/read_topic.py:691, in
↳ReadTopic._next(self, timeout)
    689 if self._next_task.done():
    690     self._next_task = asyncio.Future()
--> 691 return await asyncio.wait_for(self._next_task, timeout=timeout)

File /opt/lsst/software/stack/conda/miniconda3-py38_4.9.2/envs/lsst-scipipe-2.0
↳0/lib/python3.8/asyncio/tasks.py:501, in wait_for(fut, timeout, loop)
    497     # We must ensure that the task is not running
    498     # after wait_for() returns.
    499     # See https://bugs.python.org/issue32751
    500     await _cancel_and_wait(fut, loop=loop)
--> 501     raise exceptions.TimeoutError()
    502 finally:
    503     timeout_handle.cancel()

TimeoutError:

```

Once on target_4 and tracking, take an image with ComCam

```
[26]: 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: [2022030200004]

Stop tracking to prevent hitting the Rotator soft limit.

```
[27]: 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()
```