

## LVV-T2232 - M1M3 Integration with SAL

The objective of this test case is to verify the latest M1M3 commands, events, and telemetry defined by the latest version of the XML.

This test case will exercise the functionality of the M1M3 on the 3rd level of the Summit and meets the following criteria:

- Only requires the most current version of SAL
- Only requires the M1M3 surrogate to be loaded on the cell
- Requires the use of the DDS and the EFD

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

```
In [2]: from lsst.sitcom import vandv
```

```
exec_info = vandv.ExecutionInfo()
print(exec_info)
```

```
| lsst.ts.utils.tai INFO: Update leap second table
```

```
| lsst.ts.utils.tai INFO: current_tai uses the system TAI clock
```

```
Executed by blquint on 2022-06-14T20:20:59.382.
```

```
Running in yagan07 at summit
```

## LVV-T1996 (1.0) M1M3 DDS Startup Procedure

```
In [3]: import asyncio
import os
import yaml

import astropy.units as u
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

from astropy import time
from astropy.coordinates import AltAz, ICRS, EarthLocation, Angle, FK5
from datetime import datetime, timedelta

from lsst_efd_client import EfdClient
from lsst.ts import utils, salobj
from lsst.ts.cRIOpy import M1M3FATable
from lsst.ts.observatory.control.maintel.mtcs import MTCS, MTCSUsages
from lsst.ts.observatory.control import RotType

import lsst.sitcom.vandv as vandv
```

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

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

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

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

```
| setup.MTCS DEBUG: mtmount: Adding all resources.
| setup.MTCS DEBUG: mtptg: Adding all resources.
| setup.MTCS DEBUG: mtaos: Adding all resources.
| setup.MTCS DEBUG: mtm1m3: Adding all resources.
| setup.MTCS DEBUG: mtm2: Adding all resources.
| setup.MTCS DEBUG: mthexapod_1: Adding all resources.
| setup.MTCS DEBUG: mthexapod_2: Adding all resources.
| setup.MTCS DEBUG: mtrotator: Adding all resources.
| setup.MTCS DEBUG: mtdome: Adding all resources.
| setup.MTCS DEBUG: mtdometrajectory: Adding all resources.
| MTHexapod INFO: Read historical data in 0.05 sec
| MTHexapod INFO: Read historical data in 0.08 sec
```

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

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

```
In [9]: index = 22321285 # Test Case + Test Execution

start_time = datetime.now()
script = salobj.Controller("Script", index=index)
```

```
| Script INFO: Read historical data in 0.00 sec
```

```
In [10]: await mtcs.set_state(state=salobj.State.DISABLED, components=["mtm1m3"], overri
```

```
| setup.MTCS DEBUG: [mtm1m3]: [<State.STANDBY: 5>, <State.DISABLED: 1>]
| setup.MTCS INFO: All components in <State.DISABLED: 1>.
```

```
In [11]: await mtcs.set_state(state=salobj.State.ENABLED, components=["mtm1m3"])
```

```
| setup.MTCS DEBUG: [mtm1m3]: [<State.DISABLED: 1>, <State.ENABLED: 2>]
| setup.MTCS INFO: All components in <State.ENABLED: 2>.
```

```
In [12]: script.log.info("LVV-T12232 - LVV-E1285 - Start")
```

```
| Script INFO: LVV-T12232 - LVV-E1285 - Start
```

## Telemetry Verification

Verify the MTM1M3\_forceActuatorData telemetry data is being published to the EFD with the following parameters:

- primaryCylinderForce
- secondaryCylinderForce
- xForce
- yForce
- zForce
- fx
- fy
- fz
- mx
- my
- mz
- timestamp
- forceMagnitude

Check [Chronograph - M1M3 Status](#).

```
In [15]: if exec_info.loc == "summit":
         client = EfdClient("summit_efd")
       elif location == "tucson":
         client = EfdClient("tucson_teststand_efd")
       else:
         raise ValueError(
             "Location does not match any valid options {summit|tucson}"
         )
```

```
In [16]: start = time.Time("2022-06-14T20:20", scale="utc", format="isot")
         end = time.Time("2022-06-14T20:30", scale="utc", format="isot")
```

```
In [19]: df = await client.select_time_series(
         "lsst.sal.MTM1M3.forceActuatorData",
         fields="*",
         start=start.utc,
         end=end.utc,
       )
```

```
In [22]: df
```

Out [22]:

forceMagnitude

fx

fy

fz

mx

2022-06-14 20:23:30.431000+00:00	3809.418457	85.224068	-1504.604858	-3498.652588	-715.060120
2022-06-14 20:23:30.443000+00:00	3807.615479	84.975540	-1503.668823	-3497.098145	-705.331177
2022-06-14 20:23:30.463000+00:00	3811.184814	85.337730	-1504.894043	-3500.448730	-725.897766
2022-06-14 20:23:30.483000+00:00	3808.606689	84.858192	-1504.350830	-3497.886719	-707.882690
2022-06-14 20:23:30.503000+00:00	3811.266113	85.243378	-1504.893555	-3500.539795	-709.605957
...	...	...	...	...	...
2022-06-14 20:29:59.908000+00:00	0.517302	-0.003042	0.423209	0.297466	-3.725654
2022-06-14 20:29:59.929000+00:00	1.844737	-0.073257	-1.232412	-1.370711	-6.511684
2022-06-14 20:29:59.949000+00:00	0.405582	0.132096	0.378276	0.062885	-1.638163
2022-06-14 20:29:59.969000+00:00	1.033179	0.129351	-0.939686	-0.409531	-3.692890
2022-06-14 20:29:59.989000+00:00	0.916157	0.241840	-0.353054	0.810068	-8.234219

19479 rows × 552 columns

In [20]:

df.iloc[0]

Out [20]:

forceMagnitude3809.418457

fx85.224068

fy-1504.604858

fz-3498.652588

mx-715.06012

...

zForce95-51.223316

zForce96-11.185081

zForce97-54.310261

zForce98-43.609318

zForce99-23.358471

Name: 2022-06-14 20:23:30.431000+00:00, Length: 552, dtype: object

MTM1M3.powerSupplyData ERROR: tel\_powerSupplyData DDS read queue is full (100 elements); data may be lost

MTM1M3.inclinometerData ERROR: tel\_inclinometerData DDS read queue is full (100 elements); data may be lost

MTM1M3.imsData ERROR: tel\_imsData DDS read queue is full (100 elements); data may be lost

MTM1M3.hardpointMonitorData ERROR: tel\_hardpointMonitorData DDS read queue is full (100 elements); data may be lost

MTM1M3.hardpointActuatorData ERROR: tel\_hardpointActuatorData DDS read queue is full (100 elements); data may be lost

```

MTM1M3.forceActuatorData ERROR: tel_forceActuatorData DDS read queue is full (100 elements); data may be lost
MTM1M3.accelerometerData ERROR: tel_accelerometerData DDS read queue is full (100 elements); data may be lost
MTM1M3.logevent_forceActuatorWarning ERROR: evt_forceActuatorWarning DDS read queue is full (100 elements); data may be lost
MTM1M3.logevent_appliedForces ERROR: evt_appliedForces DDS read queue is full (100 elements); data may be lost
MTM1M3.logevent_appliedCylinderForces ERROR: evt_appliedCylinderForces DDS read queue is full (100 elements); data may be lost

```

---

Verify the MTM1M3\_forceActuatorPressure telemetry data is being published to the EFD with the following parameters:

- timestamps
- primaryCylinderPullPressures
- primaryCylinderPushPressures
- secondaryCylinderPullPressures
- secondaryCylinderPushPressures

```

In [23]: fap_df = await client.select_time_series(
            "lsst.sal.MTM1M3.forceActuatorPressure",
            fields="*",
            start=start.utc,
            end=end.utc,
        )

```

```
In [24]: fap_df
```

```
Out[24]: —
```

---

Verify the MTM1M3\_inclinometerDatatelemetry data is being published to the EFD with the following parameters:

- timestamp
- inclinometerAngle

```

In [33]: df_id = await client.select_time_series(
            "lsst.sal.MTM1M3.outerLoopData",
            fields="*",
            start=start.utc,
            end=end.utc,
        )

```

```
In [34]: df_id
```

```
Out[34]: —
```

```
In [ ]: script.log.info("LVV-T12232 - LVV-E1285 - END")
```

---

## 61 - Engineering Mode Test

While the M1M3 is enabled and in the PARKED state, send an `MTM1M3_command_enterEngineering` command.

`PARKED` means that it is not raised.

In [ ]: