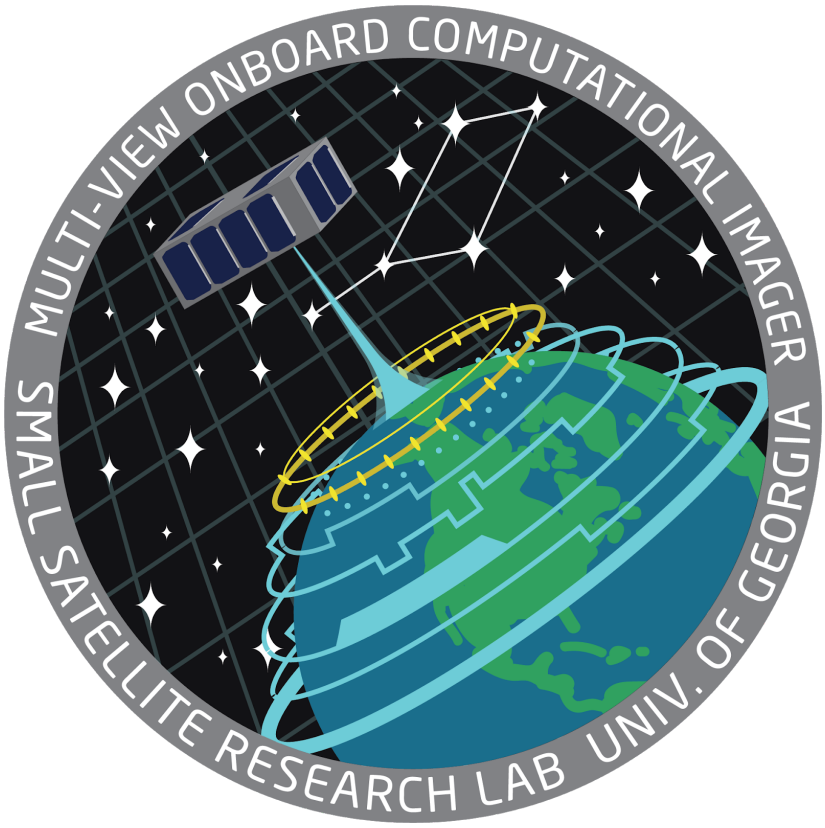
Small Satellite Research Laboratory
Franklin College of Arts and Sciences
UNIVERSITY OF GEORGIA

Author0: Messitte, Joshua
Author1: McFerren, W. Conor

Date: 2019/10/25

MOCI Command Execution Test



Revision Table

Changes	Authors	Date	Version
Added overview and description of test requirements.	Josh Messitte	11/17/2020	0.0.1
Added scenarios of tests we plan on conducting.	Josh Messitte	11/18/2020	0.0.2
Added Testing Procedures and reviewed for Interim Review 4	W. Conor McFerren	11/18/2020	0.1.0
Updated commands from ADSC/TX2i actions/parameters	Josh Messitte	12/24/2020	0.1.1
Continued to update actions/parameters	Josh Messitte	12/25/2020	0.1.2

Introduction & Overview

The purpose of the Command Execution Test (CET) is to execute every command that will be sent to the spacecraft by operators, and to the subsystems by the Command and Data Handling (C&DH) subsystem. It is important that this be done on the ground in order to make sure that the commands work and do not put the satellite into any unknown error states. The test is full execution of the Command and Telemetry List (CTL). All commands should be sent to the spacecraft and an effort be made to observe the spacecraft's physical response to the command (meaning not only the successful transmission, but also execution). Further, all internal commanding of the C&DH subsystem to subsystems should be demonstrated (e.g. voltage and current thresholds on EPS for different operations modes). These should be tested in operational use cases if configuration changes from system modes. The test should also capture if the system is appropriately storing data. Note that the CET should capture all non-nominal operations. Therefore, one might notice that there is overlap between testing objectives of the CET and DitL. Note that any nominal commands that have been successfully executed in the DitL test do not have to be repeated in the CET if the DitL is performed first. However, if nominal commands are not explicitly tested in CET but are tested in DitL, these commands should be referenced accordingly in the test report. CET should capture all edge cases in addition to the nominal commands during DitL.

Prerequisite Tests

**Add later*

Test Requirements

Requirement	Description
1	Demonstrate execution of every command (including non-nominal) that could be sent to the satellite.
2	Demonstrate execution of every internal command on the satellite.
3	Test must be run on fully integrated satellite (or FlatSat for PIR).
4	Test must be performed through RF transmission of commands.

Setup

In order to make sure no commands send the satellite into an error state, we need to test commands under different circumstances. Whenever possible, we try to specify the circumstance under which each command is sent and should therefore be tested under as well.

1. Test the following actions/parameters on each subsystem and record the results:

   HKBeacon *(chd.tmtc)*

   Actions:
   - reset
   - send

   Parameters:
   - enabled
   - frameDefinitions
   - frameParams
   - frameEnabled
   - frameEnabledBitmask
   - framePeriodMultipliers
   - frameSid

   TMDebug *(chd.tmtc)*

   Actions:
   - reset

   Parameters
   - debugLevel
   - includeSourceRef
   - includeThreadID
   - messageCount
   - messageDropped
   - transactionsLost

   TMTCEvent *(chd.tmtc)*

   Actions:
   - reset
   - forwardEvent

   Parameters:
   - enabled
   - eventEnable

   TMTCTransfer *(chd.tmtc)*

   Actions:
   - clearGetTransfer
   - clearSetTansfer

   Parameters:
   - enabled
   - lastTcSignature
   - getTransferInProgress
   - getTransferTransactionId
   - setTransferInProgress

- setTransferTransactionId

WatchdogPeriodicAction *(chd)*

Actions:

- clear
- reset

Parameters:

- enabled
- idIsAction
- actionListPacked
- actionListPeriodMultiplier
- actionListCurrentPeriods
- actionListIds
- actionListEnabled
- actionListInvoked
- actionListAck
- paramRowList

Storage *(core)*

Actions:

- wipe
- getParamToChannel
- setParamFromChannel

Parameters:

- description
- isCircular
- isClaimed
- isFull
- maxBytes
- rowLength
- numRows

ConfigurationManager *(core)*

Actions:

- resetAll
- loadAll
- sotreAll
- loadProfile
- loaf
- store
- erase
- eraseConfig
- eraseAll

Parameters:

- enabled
- configurationIds
- configSizes
- profileCount

OBT *(core)*

Actions:

- reset
- update

Parameters:

- time
- uptime

EventDispatcher *(core)*

Actions:

- reset

Parameters:

- enabled
- pendingCount
- eventCount
- overflowCount
- lastEvent

OBC *(platform.obc)*

Actions:

- reset
- kickWatchdog
- markCurrentImageStable
- clearImage
- updateImageCrc
- resetGPS
- reset Gyros

Parameters:

- firmwareVersion
- partNumber
- mpuEnable
- currBootImage
- nextBootImage
- image1
- image2
- imageCrc
- imageValid
- imagePriority
- imageIsStable
- dtmfSequence
- dtmfSequenceLocked
- dtmfFrequency
- mramCorrectableErrorCount
- mramNonCorrectableErrorCount
- flashCorrectableErrorCount
- flashNonCorrectableErrorCount
- gpsEnable
- adcEnable
- gyroEnable
- magnetometerEnable

Time *(platform.obc)*
Actions:
- reset
- refresh

Parameters:
- enabled
- time

GPIO *(platform.obc)*
Actions:
- reset

Parameters:
- pins

PlatformI2C *(platform.obc)*
Actions:
- resetStatistics

Parameters:
- enabled
- channelAddress
- nackAddrCount
- nackDataCount
- ackAddrCount
- ackDataCount
- hwStateErrorCount
- isErrorCount

PlatformSPI *(platform.obc)*
Actions:
- resetErrors

Parameters:
- errorCount
- lastError

STX *(platform)*
Actions:
- reset
- preloadBuffers
- enableTransmit
- disableTransmit
- clearFrameBuffer

Parameters:
- version
- status
- mode
- txReady
- powerAmplifierEnable
- powerAmplifierLevel

- encodingRate
- modulation
- pulseShapingFilterEnable
- offsetFrequency
- bufferUnderrun
- bufferOverrun
- bufferCount
- powerAmplifierTemperature
- rfOutputPower
- powerAmplifierCurrent
- powerAmplifierVoltage
- boardTemperature
- batteryCurrent
- batteryVoltage
- debug
- bytesEnqueued
- bytesDequeued
- frameBufferUnderrun
- rawAccess

CMC *(platform)*

Actions:

- reset
- resetBeacon
- resetRx
- clearBeaconTimer
- send
- SetInactivityBeacon

Parameters:

- enabled
- version
- mode
- beaconEnable
- beaconTimeout
- beaconInterval
- beaconData
- txTransparent
- txConvEnabled
- txFrequencyOffset
- txLock
- txPower
- txFlagsCount
- txSyncCount
- txReady
- txReadyThreshold
- txSpaceAvailable
- txPacketCount
- txOverrunErrorCount

- rxFrequencyOffset
- rxLock
- rxDtmfAndCounter
- rxRssi
- rxReady
- rxDataWaiting
- rxFrameCount
- rxPacketCount
- rxDroppedErrorCount
- rxCrcErrorCount
- rxOverrunErrorCount
- rxProtocolErrorCount
- temperatureSmps
- temperaturePa
- voltage3V3
- voltage5V
- current3V3
- current5V
- debugShowLock
- debugFlags
- paForwardPower
- paReversePower

EPS *(platform)*

Actions:

- resetWatchDog
- switchOnAllSwitches
- switchOffAllSwitches
- resetSwitchStates
- turnOnSwitches
- turnOffSwitches
- cycleBus
- reset

Parameters

- status
- lastError
- version
- romChecksum
- revision
- watchDogTimeout
- brownOutResetCount
- autoResetCount
- manualResetCount
- watchdogResetCount
- actualSwitchStates
- actualSwitchStatesBitmap
- expectedSwitchStates
- expectedSwitchStatesBitmap

- initialSwitchStates
- initialSwitchStatesBitmap
- switchOvercurrentBitmap
- switchTimerLimit
- switchTimerElapsed
- BcrOutputCurrent
- BcrOutputVoltage
- currentDrawOfEps
- busVoltages
- busCurrents
- switchVoltages
- switchCurrents
- boardTemperatures
- solarArrayVoltages
- solarArrayCurrents
- solarArrayTemperatures
- solarArrayIllumination

ISIS *(platform)*

Actions:

- reset
- deploy
- deployAll
- cancel

Parameters:

- armState
- timeout
- override
- status
- activationCount
- activationTime
- temperature

BAT1 *(platform)*

Actions:

- reset

Parameters:

- batteryCurrentDir
- batteryCurrent
- batteryVoltage
- boardTemperature
- batteryTemperature
- batteryHeaterStatus
- version
- status
- lastError
- romChecksum
- manualSoftResetCount
- autoResetCount

- brownOutResetCount
- heaterAutomationEnabled

BAT2 *(platform)*

Actions:

- reset

Parameters:

- batteryCurrentDir
- batteryCurrent
- batteryVoltage
- boardTemperature
- batteryTemperature
- batteryHeaterStatus
- version
- status
- lastError
- romChecksum
- manualSoftResetCount
- autoResetCount
- brownOutResetCount
- heaterAutomationEnabled

FlashDevice *(platform.obc)*

Actions:

- erase
- blockErase

Parameters:

- pageAddress
- writeRepeats
- pageData

FileSystem *(platform.obc)*

Actions:

- format

Parameters:

- enabled
- freeFileHandles
- freeDirEntries

CubeSpaceADCS *(platform)*

Actions:

- Reset
- ReserLogPointer
- AdvanceLogPointer
- ReserBootREgisters
- DeployMagnetometerBoom
- ADCSRunMode
- ADCSPowerControl
- ClearErrors
- TriggerADCSLoop
- TriggerADCSLoopWithSimulatedSensorData

- ConvertToJPGFile
- DefaultConfiguration
- SaveConfiguration
- SaveOrbitParameters
- SaveImage
- SDLog1Configuration
- SDLog2Configuration
- UARTLogConfiguration
- EraseFile
- LoadFileDownloadBlock
- AdvanceFileListReadPointer
- InitiateFileUpload
- FinalizeUploadBlock
- ResetUploadBlock
- InitiateDownloadBurst
- InitializeUploadComplete
- UploadBlockComplete
- GenerateTLE

Parameters:

- CurrentUnixTime
- SRAMScrubParameters
- AttitudeControlMode
- AttitudeEstimationMode
- MagnetorquerOutput
- WheelSpeed
- ADCSConfiguration
- MagnetorquerConfiguration
- WheelConfiguration
- RateGyroConfiguration
- CSSAlignmentConfiguration
- CSSScaleFactorConfiguration
- Cam1SensorConfiguration
- Cam2SensorConfiguration
- NadirSensorMaskConfiguration1
- NadirSensorMaskConfiguration2
- NadirSensorMaskConfiguration3
- NadirSensorMaskConfiguration4
- NadirSensorMaskConfiguration5
- MagnetometerMountingConfiguration
- MagnetometerOffsetAndScalingConfiguration
- MagnetometerSensitivityConfiguration
- RateSensorConfiguration
- StarTrackerConfiguration
- DetumblingControlParameters
- YWheelControlParameters
- ReactionWheelControlParameters
- MomentsOfInertia
- ProductsOfInertia

- EstimationParameters1
- EstimationParameters2
- SGP4OrbitParameters
- SGP4OrbitInclination
- SGP4OrbitEccentricity
- SGP4OrbitRAAN
- SGP4OrbitArgumentofPerigee
- SGP4OrbitBStarDragTerm
- SGP4OrbitMeanMotion
- SGP4OrbitMeanAnomaly
- SGP4OrbitEpoch
- TrackingControllerGainParameters
- TrackingControllerTargetReference
- ModeOfMagnetometerOperation
- FileUploadPacket
- HoleMap1
- HoleMap2
- HoleMap3
- HoleMap4
- HoleMap5
- HoleMap6
- HoleMap7
- HoleMap8
- Identification
- BootIndexAndStatus
- CacheEnabledState
- BootAndRunningProgramStatus
- JPGConversionProgress
- LastLoggedEvent
- SRAMLatchupCounters
- EDACErrorCounters
- CommunicationStatus
- CurrentADCSState
- EstimatedAttitudeAngles
- EstimatedAngularRates
- SatellitePositionECI
- SatelliteVelocityECI
- SatellitePositionLLH
- MagneticFieldVector
- CoarseSunVector
- FineSunVector
- NadirVector
- RateSensorRates
- WheelSpeed
- MagnetorquerCommand
- WheelSpeedCommands
- IGRFModelledMagneticFieldVector
- ModelledSunVector

- EstimatedGyroBias
- EstimationInnovationVector
- QuaternionErrorVector
- QuaternionCovariance
- AngularRateCovariance
- RawCam2Sensor
- RawCam1Sensor
- RawCSS1To6
- RawCSS7To10
- RawMagnetometer
- CubeSenseCurrentMeasurements
- CubeControlCurrentMeasurements
- WheelCurrents
- ADCSTemperatures
- RateSensorTemperatures
- RawGPSStatus
- RawGPSTime
- RawGPSX
- RawGPSY
- RawGPSZ
- Star1BodyVector
- Star2BodyVector
- Star3BodyVector
- Star1OrbitVector
- Star2OrbitVector
- Star3OrbitVector
- StarMagnitude
- StarPerformance
- StarTiming
- ADCSState
- ADCSMeasurements
- ActuatorCommands
- EstimationData
- RawSensorMeasurements
- PowerandTemperatureMeasurements
- ADCSExecutionTimes
- ADCSPowerControl
- ADCSMiscCurrrentMeasurements
- CommandedAttitudeAngles
- TrackingConrollerTargetReference
- FineEstimatedAngularRates
- SGP4OrbitParameters
- RawGPSMeasurements
- RawStarTracker
- Star1RawData
- Star2RawData
- Star3RawData
- RedundantMagnetometerRawMeasurements

- RawRateSensor
- EstimatedQuaternion
- ECEFPosition
- ACPExecutionState
- StatusOfImageCaptureAndSaveOperation
- SDLog1Configuration
- SDLog2Configuration
- UARTLogConfiguration
- TelecommandAcknowledge
- DownloadBlockReady
- FileInformation
- UnixTimeSaveToFlash
- CommandedAttitudeAngles
- FileDownloadBufferWithFileContents
- BlockChecksum
- MagicNumber
- PowerControl
- TriggerADCSLoopWithSimulatedSensorData

SDLS *(comms)*

Actions:

- CreateDownlinkSA
- DeleteDownlinkSA
- ExpireDownlinkSA
- CreateUplinkSA
- DeleteUplinkSA
- ExpireUplinkSA

Parameters:

1. DownLinkSA
2. ActiveDownlinkSA
3. UpinkSA
4. FrameCounters

UmbilicalSerial *(comms)*

Actions:

- reset

Parameters:

- enabled
- txFailures
- framingErrors
- partyErrors
- overrunErrors

StxDownlink *(comms.STX)*

Actions:

- StxDownlink.requestData
- StxDownlink.suspend
- StxDownlink.resume
- StxDownlink.abort
- StxDownlink.reset

Parameters:

- state
- currRequest
- currId
- currRowRange
- requestCount
- requestSuccessCount
- requestFailureCount
- transactionSuccessCount
- transactionFailureCount
- requestLeakCount
- transactionLeakCount

TMTCBuffer *(comms)*

Actions:

- reset
- clear

Parameters:

- spacelinkTxEnable
- umbilicalTxEnable
- umbilicalRxEnable
- umbilicalTimeout
- txCount
- rxCount

PUSLDT *(comms.pus)*

Actions:

- txAbort
- txSuspend
- txResume
- txReset
- rxAbort
- rxSuspend
- rxResume
- rxReset

Parameters:

- txTransferState
- txTransferTransactionId
- txTransferTimeoutCount
- rxTransferState
- rxTransferTransactionId
- rxTransferTimeoutCount
- txTimeout
- txMaxTimeouts
- rxTimeout
- rxMaxTimeouts
- txLostTransactoins
- rxLostTransactions
- rxProcessingErrors

PayloadSerial *(comms)*
Actions:
- reset

Parameters:
- enabled
- txFailures
- rxFailures
- framingErrors
- parityErrors
- overrunErrors

TX2 *(payload)*
Actions:
- write
- powerHard
- powerOnSoft
- powerOffSoft
- ping
- getTelemetry
- setProgrammaticState
- setPowerState
- getData
- sendData
- reset
- restartAPI

Parameters:
- sendFileContent
- receiveData

DataPool *(cdh.telemetry)*
Actions:
- reset
- refresh

Parameters:
- lifeTime
- paramRefreshTime
- paramRefreshStatus
- paramValid

DataPoolBaseSampler *(cdh.telemetry)*
Actions:
- sample

Parameters:
- paramList
- periodMultiplier
- burstSize
- enabled

DataPoolHighResSampler *(cdh.telemetry)*
Actions:
- sample

Parameters:
- paramList
- periodMultiplier
- burstSize
- enabled

ParamStats *(cdh.telemetry)*

Actions:
- refreshAll
- refreshList
- resetAll
- resetList
- clear

Parameters:
- enabled
- statisticsPacked
- paramListPAcked
- resetInStatisticREad
- paramEnabled
- statisticValid
- isSigned
- maxSignedValue
- maxUnsignedValue
- maxTime
- timeSinceMax
- minSignedValue
- minUnsignedValue
- minTime
- timeSinceMin
- changeTime
- timeSinceChange
- meanSignedValue
- meanUnsgnedValue
- periodMultiplier

PoolLogger *(cdh.logging)*

Actions:
- reset
- flush
- log

Parameters:
- enabled
- channelId
- paramId
- periodMultiplier

TCLogger *(cdh.logging)*

Actions:
- reset
- flush

- log

Parameters:

- enabled
- channelId

EventLogger *(cdh.logging)*

Actions:

- reset
- flush
- log

Parameters:

- enabled
- channelId
- severityFilterType
- severityFiler

STXLogger *(cdh.logging)*

Actions:

- reset
- flush
- log

Parameters:

- enabled
- channelId
- paramId
- periodMultiplier

WarningMonitor *(cdh.monitoring)*

Actions:

- restartChecks
- refresh

Parameters:

- enabled
- readOnly
- groupFailThreshold
- groupFailCount
- checkListPacked
- checkEnabled
- checkIsValid
- checkIsDelta
- checkCanWrap
- checkChecked
- checkFailed
- checkFailThreshold
- checkGetStatus
- checkParameterId
- checkParameterRow
- checkIsSigned
- checkUpperBoundSigned
- checkUpperBoundUnsigned

- checkLowerBoundSigned
- checkLowerBoundUnsigned
- checkUpperEventId
- checkLowerEventId
- checkGroup

CriticalMonitor *(cdh.monitoring)*

Actions:

- restartChecks
- refresh

Parameters:

- enabled
- readOnly
- groupFailThreshold
- groupFailCount
- checkListPacked
- checkEnabled
- checkIsValid
- checkIsDelta
- checkCanWrap
- checkChecked
- checkFailed
- checkFailThreshold
- checkGetStatus
- checkParameterId
- checkParameterRow
- checkIsSigned
- checkUpperBoundSigned
- checkUpperBoundUnsigned
- checkLowerBoundSigned
- checkLowerBoundUnsigned
- checkUpperEventId
- checkLowerEventId
- checkGroup

StxMonitor *(cdh.monitoring)*

Actions:

- restartChecks
- refresh

Parameters:

- enabled
- readOnly
- groupFailThreshold
- groupFailCount
- checkListPacked
- checkEnabled
- checkIsValid
- checkIsDelta
- checkCanWrap

- checkChecked
- checkFailed
- checkFailThreshold
- checkGetStatus
- checkParameterId
- checkParameterRow
- checkIsSigned
- checkUpperBoundSigned
- checkUpperBoundUnsigned
- checkLowerBoundSigned
- checkLowerBoundUnsigned
- checkUpperEventId
- checkLowerEventId
- checkGroup

EventAction *(cdh.scheduling)*

Actions:

- clear

Parameters:

- enabled
- entryListPacked
- entryEventId
- entryEventInfo
- entryId
- entryEnabled
- entrySeverity
- entryOnce
- entryExecuted
- entryAck
- entryStatus
- entryIdIsAction
- entryParamRowList

CriticalEventAction *(cdh.scheduling)*

Actions:

- clear

Parameters:

- enabled
- entryListPacked
- entryEventId
- entryEventInfo
- entryId
- entryEnabled
- entrySeverity
- entryOnce
- entryExecuted
- entryAck
- entryStatus
- entryIdIsAction

- entryParamRowList

DPEventAction *(cdh.scheduling)*

Actions:

- clear

Parameters:

- enabled
- entryListPacked
- entryEventId
- entryEventInfo
- entryId
- entryEnabled
- entrySeverity
- entryOnce
- entryExecuted
- entryAck
- entryStatus
- entryIdIsAction
- entryParamRowList

ScanEventAction *(cdh.scheduling)*

Actions:

- clear

Parameters:

- enabled
- entryListPacked
- entryEventId
- entryEventInfo
- entryId
- entryEnabled
- entrySeverity
- entryOnce
- entryExecuted
- entryAck
- entryStatus
- entryIdIsAction
- entryParamRowList

DownlinkEventAction *(cdh.scheduling)*

Actions:

- clear

Parameters:

- enabled
- entryListPacked
- entryEventId
- entryEventInfo
- entryId
- entryEnabled
- entrySeverity
- entryOnce

- entryExecuted
- entryAck
- entryStatus
- entryIdIsAction
- entryParamRowList

TimeAction *(cdh.scheduling)*

Actions:

- clear
- timeShift
- restartRelativeEntries

Parameters:

- enabled
- entryListPacked
- entryTime
- entryId
- entryEnabled
- entryisRelative
- entryOneShot
- entryExpired
- entryExecuted
- entryStatus
- entryIdIsAction
- entryParamRowLsit
- entryRepeatDelay
- entryRepeatCount

ModeTimeAction *(cdh.scheduling)*

Actions:

- clear
- timeShift
- restartRelativeEntries

Parameters:

- enabled
- entryListPacked
- entryTime
- entryId
- entryEnabled
- entryisRelative
- entryOneShot
- entryExpired
- entryExecuted
- entryStatus
- entryIdIsAction
- entryParamRowLsit
- entryRepeatDelay
- entryRepeatCount

Scheduler *(mission)*

Actions:

- addEntry
- recoverToMode
- enableTimeAction
- disableTimeAction
- clearSchedule
- clearScheduleEntry
- setSchedule

Parameters:

- ModeID
- TimeShift
- SubmodeID

DownlinkManager *(mission)*

Actions:

- prepDownlink
- startDownlink
- endDownlink

Parameters:

- downlinkInProgress
- downlinkFrequencyOffset

SeparationSequence *(mission)*

Actions:

- resume
- halt
- reset
- restart

Parameters:

- state
- checkpoint
- antennaDeploymentState
- admFireProducedCounter
- admHardFireCounter
- admTelemetryLog
- separationTimestamp
- depolymentInhibit

ModeManager *(mission)*

Actions:

- modeTimeout
- recoverToSafe
- recoverToCruise
- setModeWithScheduleID
- recoverToMode

Parameters:

- startUpMode
- IsDeployment
- mode

- checkpointTime
- modeTimeoutPeriod
- groundStationLLA
- downlinkModeTimeoutPeriod
- scanModeTimeoutPeriod
- dataProcModeTimeoutPeriod

Testing Procedures

This test will be performed in a unit test style. Using the TMTCLab Python API, we will automate most of the commands and assert that they are functional and appropriately executed. The reaction to a given command will either be programmatic or physical. Thus, some commands may require assertions within the testing script and some will require visual confirmation.

The script should generate a log file with the commands called, a timestamp, the result (pass or fail) and any errors that might follow, and any necessary information that is required to fully understand the command.

While this test must be automated to the largest degree that it can be, it is necessary for some command to have real time intervention from an operator.

Results and Conclusions

**Add after testing commands*

Citations

- UNP NS10 users guide
- Telecommand List_v2.0.0