cFS MM Requirements

Code 582
Flight Software Systems Branch

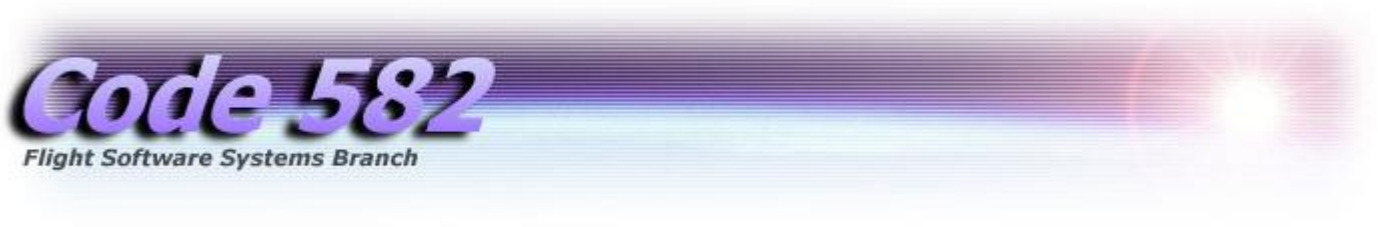
Apr 3, 2015

ID	Summary
1384	CFS MM Requirements Document

SR_Contains:

ID	ReqID	Text	Rationale	Heritage Reference
1386		# CFS Memory Manager (MM) Requirements ## 1.0 Introduction ### 1.1 Document Purpose The Core Flight Software System (CFS) Memory Manager Application will be developed by the Flight Software Branch (FSB) of the Software Engineering Division (SED). The purpose of this requirements specification is to define the requirements to be satisfied by the Memory Manager Application. This application is developed for re-use. For this reason, several nomenclatures are used in this document to identify configurations for a mission. The CFS is specified as a multi-platform product. Mission-specific features and customization requirements which are applicable for all platforms are tagged with <MISSION_DEFINED>. Platform-specific features and customizations requirements are		

cFS MM Requirements

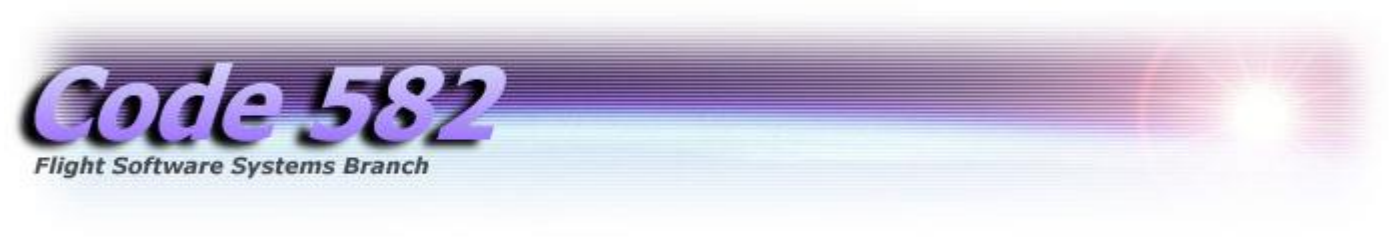


Apr 3, 2015

ID	Summary

		tagged with either "<PLATFORM_DEFINED>" or "<OPTIONAL>." Additional nomenclature is used along with the tag to specify a CFS default value for the platform-specific feature: "<PLATFORM_DEFINED, Default_Value>". Reference platforms (single processor and multi-processor architectures) are defined to supply the default CFS application configuration. These configurations define the "maximum" CFS Application deployments such that any refined deployment is a subset of a reference platform. ## 1.2 Document Scope The scope of this document is limited to the specification of requirements for the Memory Manager Software requirements. These include functional, performance, qualification, and design requirements. ## 1.3 Document Organization This document is organized into three additional sections and several appendices. Section 2 gives the Memory Manager context. Section 3 documents the Memory Manager system design decisions and constraints.		

cFS MM Requirements



Apr 3, 2015

ID	Summary

		Section 4 contains the Memory Manager functional and performance requirements. Appendix A contains a list of abbreviations and acronyms used in this document. **1.4 Relevant Documents** **1.4.1 Parent Documents** CFS Memory Manager Application Heritage Analysis 582-2007-011 582-2007-011 **1.4.2 Reference Documents** 1. Operating System Abstraction Layer (OSAL) Library 2. cFE Application Developer's Guide 582-2007-001 3. cFE User's Guide **2.0 CFS Memory Manager Application Context**		

cFS MM Requirements

Code 582
Flight Software Systems Branch

Apr 3, 2015

ID		Summary		
		The Memory Manager (MM) component of the Core Flight System (CFS) is responsible for the loading and dumping of flight system memory. MM is basically the operator interface for the OSAL memory manipulation. Memory Manager provides the ability to load and dump memory via commands as well as from files. If the operating system supports symbolic addressing, Memory Manager supports specifying the memory address using a symbolic address.		

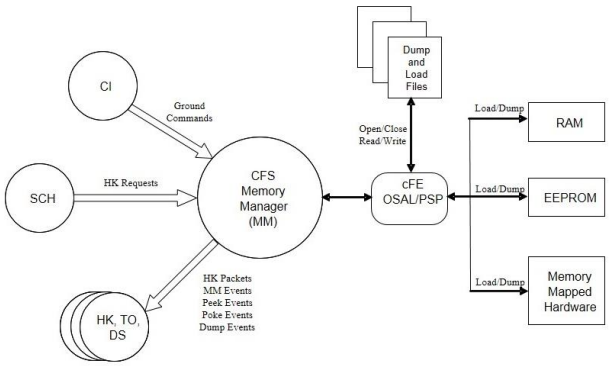
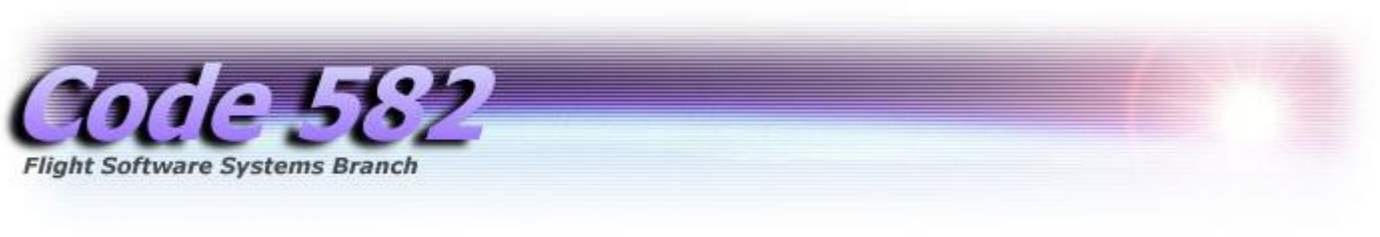


Figure 2.0 – CFS MM Context

Memory Manager makes use of the OSAL when interfacing to memory. Memory Manager assumes that the OSAL will provide routines to access processor memory as well as memory that is not directly accessible (i.e. requires address translation). Address checking is performed using the OSAL. Any addresses specified outside of the valid address range will be considered invalid.

MM performs data transfers between memory and files, but does not handle file dumps or loads. That

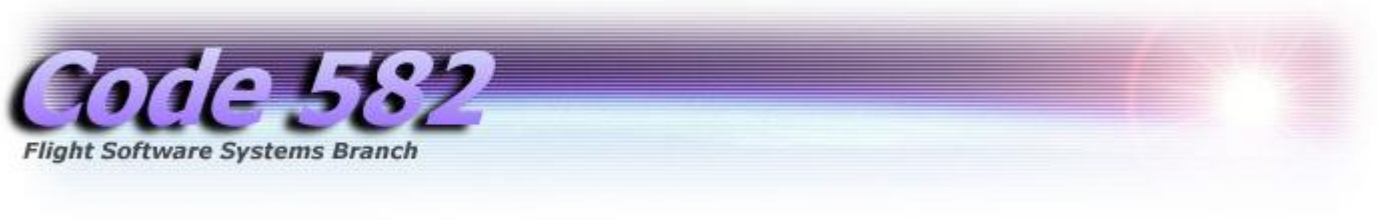
cFS MM Requirements



Apr 3, 2015

ID	Summary			
		function must be done with a file transfer application such as the CFS CFDP application. Some of the Memory Manager requirements relate to the use of files. MM does not have responsibility for file management operations or directory manipulations. That function is allocated to the CFS FM (File Manager) application. It should be noted that Memory Manager assumes that the files are binary. There are 4 types of memory that are referred to in Memory Manager : • RAM – processor memory. Generic term for RAM including DRAM, and SRAM • EEPROM – Generic term used for non-volatile memory including EEPROM, Flash, PROM, etc. • Memory Mapped I/O - Addressable Memory that must be read from and written to in 8, 16 or 32 bits at a time Note that for the Memory Mapped I/O that is byte addressable and requires no special code to support will be accessed as RAM. • Port I/O – I/O that must be read from and written to in 8, 16 or 32 bits at a time. Note that this type of I/O is not supported in Memory Manager as it is not required in any current future missions. Most missions in the past and the foreseeable future use memory mapped I/O		

cFS MM Requirements



Apr 3, 2015

ID	Summary

2.1 Assumptions

The following list summarizes the assumptions made by the CFS Memory Manager Application:

· cFE API and OSAL are being used

· OSAL provides memory validation routines in order to validate the memory addresses

· OSAL/ OS BSP handles banked memory

· OSAL/ OS BSP handles EEPROM alignment requirements

· Memory Manager is not responsible for uploading or downloading file. Files generated by or loaded for Memory Manager are transferred using a file transfer application such as CFDP.

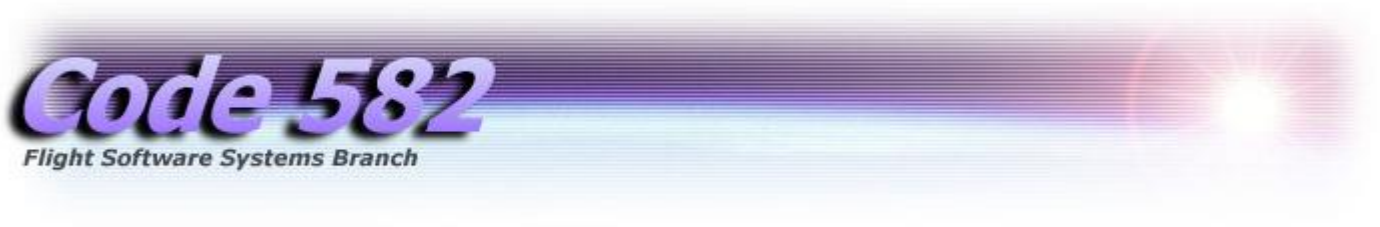
· All files are in BINARY format.

· Memory Manager will prevent CPU hogging by segmenting the amount of data written to or read from a file in a cycle.

· CRC API provided by cFE which computes CRC using the desired (mission_defined) CRC algorithm.

· Memory Mapped I/O does not support symbolic addressing.

cFS MM Requirements



Apr 3, 2015

ID	Summary

3.0 Design Specifications

The Memory Manager Application's requirements and design are based on the results of the CFS heritage analysis effort. The results of the heritage analysis are document in the CFS Memory Manager Application Heritage Analysis document.

The Memory Manager Application is based on the Core Flight Executive (cFE) and the OSAL. In addition, MM exists in the context of the CFS architecture.

3.1 Design Constraints

Since the OSAL will provide address range checking, Memory Manager will only be allowed to address memory that is within the address range. The OSAL's design supports the ability to modify the address range in order to facilitate accessing "unadvertised" memory or to account for differences in the hardware configurations (e.g. ETU vs. Flight processor).

Memory Manager segments loads and dumps in order to prevent CPU hogging. Careful consideration should be made when loading and dumping large amounts of memory.

Memory Manager depends on the OSAL for access to the various memory types.

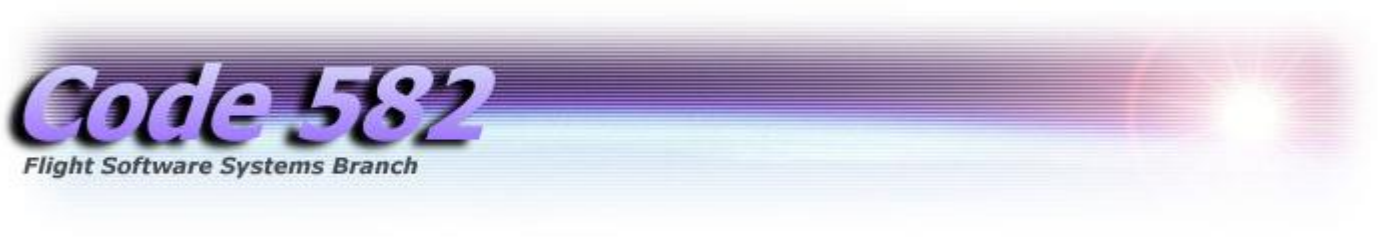
cFS MM Requirements

Code 582
Flight Software Systems Branch

Apr 3, 2015

ID		Summary		
		4.0 Subsystem Requirements The following section includes the high level requirements for Memory Manager. These subsystem requirements shall trace to spacecraft level requirements for a particular spacecraft/mission.		
1601	CFS-001	The CFS shall provide the capability to load code and data to RAM.	Provides capability of being "patched" while executing in RAM.	LRO, BAT, SDO
1603	CFS-002	The CFS shall provide the capability to load code and data to EEPROM.	Once an update is validated, we may want to make it permanent	LRO, BAT, SDO
1605	CFS-003	The CFS shall provide the capability to load data to any hardware memory mapped I/O addresses.	Useful for debugging	LRO, BAT, SDO
1609	CFS-005	The CFS shall provide the capability to dump code and data from RAM	Need to validate FSW updates in temporary storage.	LRO, BAT, SDO
1611	CFS-006	The CFS shall provide the capability to dump code and data from EEPROM	Obtain contents of non-volatile memory for debugging or verification of a load	LRO, BAT, SDO

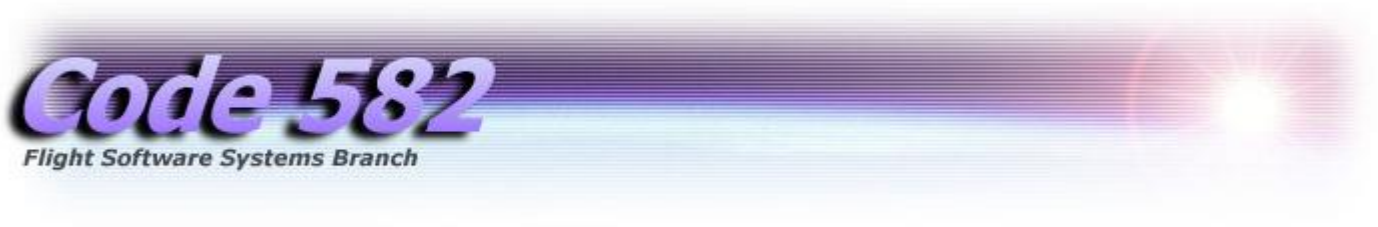
cFS MM Requirements



Apr 3, 2015

ID		Summary		
1613	CFS-007	The CFS shall provide the capability to dump data from any hardware memory mapped I/O addresses.	Obtain contents of a memory mapped device . Useful for debugging	LRO, BAT, SDO
1617	CFS-009	<OPTIONAL> The CFS shall provide symbolic address resolution	Need to cross reference symbols and addresses	BAT
1619		**5.0 Detailed Requirements** This section itemizes the detailed requirements for Memory Manager. These requirements are shall be verified as part of the build verification test effort. **5.1 Basic Command Requirements** The following requirements apply to all Memory Manager commands. Rather than repeating these requirements for each applicable requirement, they have been grouped together to cover all requirements.		
1465	MM1000	Upon receipt of a No-Op command, MM shall increment the MM Valid Command Counter and generate an event message.	Confirms path of communication to the component software.	LRO, SDO, BAT
1467	MM1001	Upon receipt of a Reset command, MM shall reset the following housekeeping variables to a value of	Important for testing and on-orbit flight	LRO, SDO, BAT

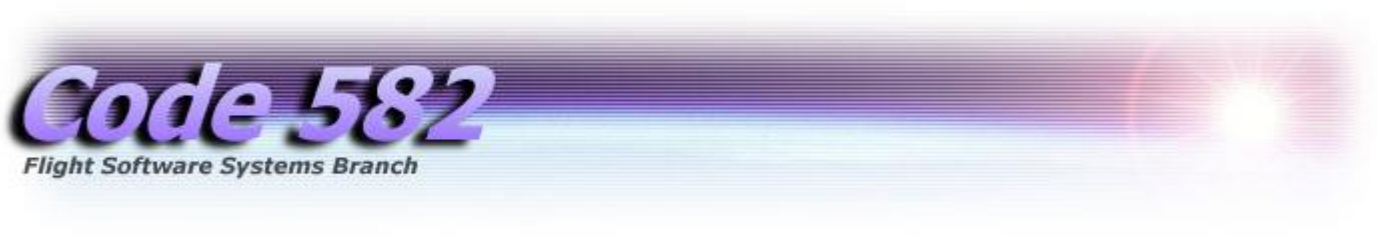
cFS MM Requirements



Apr 3, 2015

ID		Summary		
		zero: a) MM Valid Command Counter b) MM Command Rejected Counter	operations in order to start with a “clean slate”	
1469	MM1006	For all MM commands, if the length contained in the message header is not equal to the expected length, MM shall reject the command.	Basic command verification in the event of SEU or memory corruption	LRO, SDO, BAT
1471	MM1007	If the address specified in any MM command fails validation, MM shall reject the command	Covers source and/or destination address. Protects against operator error.	SDO
1473	MM1008	If the filename specified in any MM command is not valid, MM shall reject the command	Erroneous filename makes the entire command erroneous and not trustworthy.	LRO
1475	MM1009	If MM accepts any command as valid, MM shall execute the command, increment the MM Valid Command Counter and issue an event message	Provides command verification	LRO, SDO, BAT
1477	MM1010	If MM rejects any command, MM shall abort the command execution, increment the MM Command Rejected Counter and issue an error event message	Provides an alert of a command error	LRO, SDO, BAT
1479	MM1011	<OPTIONAL> Symbol Name and offset can be used in lieu an absolute address in any RAM command	If symbolic addressing supported, symbolic names and offset can be used instead of an absolute address in RAM	BAT

cFS MM Requirements



Apr 3, 2015

ID		Summary		
1481	MM1012	<OPTIONAL> Symbol Name and offset can be used in lieu an absolute address in any EEPROM command	If symbolic addressing supported, symbolic names and offset can be used instead of an absolute address in EEPROM	BAT
1483	MM1013	The MM application shall generate an error event message if symbol table operations are initiated but not supported in the current target environment.	Got to do something graceful if we're running without a system symbol table to query	Derived
1684		**5.2 RAM operations**		
1485	MM2000	Upon receipt of a Poke command , MM shall write 8, 16, or 32 bits of data to the command-specified RAM address	Memory poke to any RAM address. Typically the peek function is used to verify the poke.	LRO, BAT
1487	MM2000.1	MM shall confirm a write to the RAM address by issuing an event message which includes: a) address written b) length of data written c) value of the data written	Confirms memory poke operations	LRO, BAT
1489	MM2002	Upon receipt of a Peek command, MM shall read 8, 16, or 32 bits of data from the command-specified RAM address and generate an event message containing the following data: a) address read b) length of data read c) value of the data read	Memory peek to any Memory Type	LRO, BAT

cFS MM Requirements

Code 582
Flight Software Systems Branch

Apr 3, 2015

ID		Summary		
1491	MM2003	Upon receipt of a Write With Interrupts Disable command, MM shall write up to <PLATFORM_DEFINED, TBD> bytes to the command-specified RAM memory address with interrupts disabled.	Load with interrupts disabled so that can patch software that is executing .	LRO, SDO
1493	MM2003.1	MM shall verify that the command-specified <MISSION_DEFINED> CRC matches the computed CRC of the data	Verify the integrity of the data specified in the command to insure no corruption of the data to be written. Note that the cFE provides functions to calculate the CRC using a desired CRC	LRO
1495	MM2003.2	If the command-specified CRC fails validation, MM shall reject the command	Verify data before changing memory	LRO
1497	MM2004	Upon receipt of a Read command, MM shall read the command-specified number of consecutive bytes from the command-specified RAM memory address and generate an event message containing the data.	Dump in event message in order to provide a quick display (alleviates post processing)	LRO, SDO
1499	MM2004.1	If the number of bytes exceeds the maximum event message size then the command shall be rejected.	Cannot exceed the event message size maximum	Derived
1501	MM2100	Upon receipt of a Load From File command, MM shall load RAM, with interrupts enabled during the actual load, based on the following information contained in the command-specified file: a) Destination Address b) Destination Memory Type c) <MISSION_DEFINED> CRC (data only) d) Number of Bytes to Load	Standard load from file into RAM memory. Files are assumed to be binary	LRO, BAT (used command parameters instead of secondary file header)

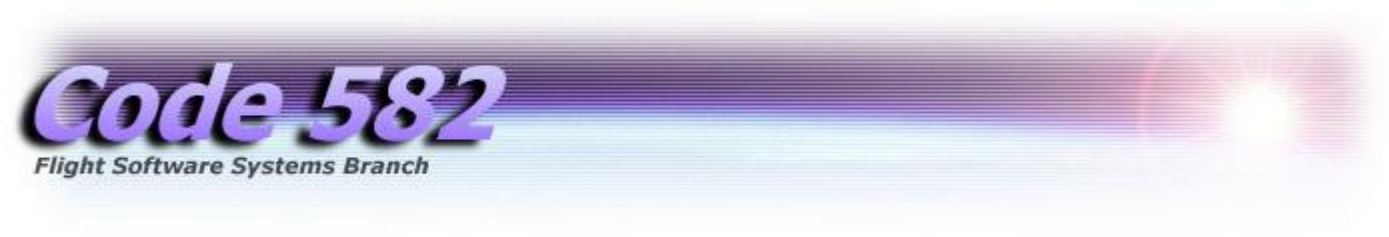
cFS MM Requirements

Code 582
Flight Software Systems Branch

Apr 3, 2015

ID		Summary		
1503	MM2100.1	If the CRC contained in the file fails validation, MM shall reject the command	Verifies load file integrity	LRO
1505	MM2100.2	If the number of bytes exceeds <PLATFORM_DEFINED, TBD> then the command shall be rejected.	<Platform_defined> maximum should be sized to be able to handle the largest app. Requirement prevents exceeding memory bounds.	LRO
1507	MM2104	Upon receipt of a Dump to File command, MM shall write the data associated with the command-specified RAM address , command-specified number of bytes and calculated <MISSION_DEFINED> CRC to the command-specified file.	Standard dump from RAM memory to a file.	LRO, BAT
1509	MM2104.1	If the command-specified number of bytes exceeds <PLATFORM_DEFINED, TBD> then the command shall be rejected.	Keep from running beyond memory type address range.	LRO
1511	MM2300	Upon receipt of a Fill command, MM shall fill RAM with the contents based on the following command-specified parameters: a) Destination Address b) Destination Memory Type c) Number of Bytes to Fill d) 32-bit Fill Pattern	Easy way to write a simple pattern to memory (or tables/data structures). Use Load from File if using a more complicated pattern. SET filled unused memory with the TRAP instruction (if the processor gets lost and wanders off into the weeds, the TRAP instruction will immediately	BAT

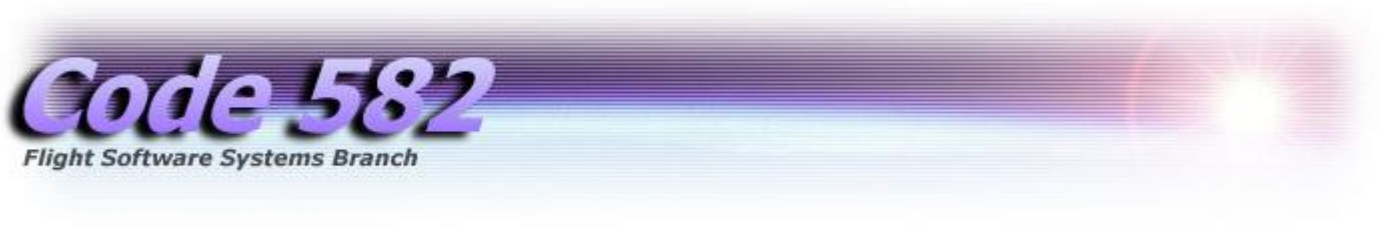
cFS MM Requirements



Apr 3, 2015

ID		Summary		
			cause an exception/interrupt)	
1513	MM2300.1	If the command-specified number of bytes exceeds <PLATFORM_DEFINED, TBD> then the command shall be rejected.	Keeps from running beyond memory type address range	Derived
1515	MM2500	When writing data to RAM memory, MM shall write a maximum of <PLATFORM_DEFINED, TBD> bytes per execution cycle	Segments the load in order to prevent CPU hogging.	LRO
1517	MM2501	When writing RAM data to a file, MM shall write a maximum of <PLATFORM_DEFINED, TBD> bytes per execution cycle	Writing to a file can be very slow. Need to segment writing the file.	LRO
1686		**5.3 EEPROM Operations**		
1519	MM3000	Upon receipt of a Poke command , MM shall write 8, 16, or 32 bits of data to the command-specified EEPROM address	Memory poke to any EEPROM address. Typically the peek function is used to verify the poke.	LRO, BAT
1521	MM3000.1	MM shall confirm a write to the EEPROM address by issuing an event message which includes: a) address written b) length of data written c) value of the data written	Confirms memory poke operations	LRO, BAT
1523	MM3001	Upon receipt of a Peek command, MM shall read 8, 16, or 32 bits of data from the command-specified EEPROM address and generate an event message containing the following data:	Memory peek to any Memory Type	LRO, BAT

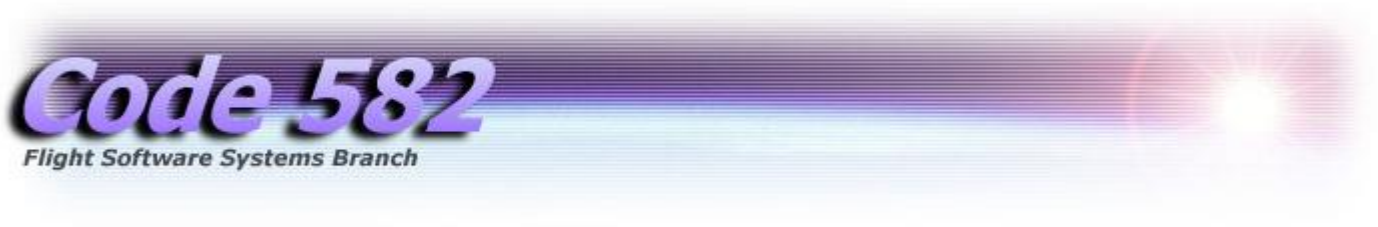
cFS MM Requirements



Apr 3, 2015

ID		Summary		
		a) address read b) length of data read c) value of the data read		
1525	MM3002	Upon receipt of a Read command, MM shall read the command-specified number of consecutive bytes from the command-specified EEPROM memory address and generate an event message containing the data.	Dump in event message in order to provide a quick display (alleviates post processing)	LRO, SDO
1527	MM3002.1	If the number of bytes exceeds the maximum event message size then the command shall be rejected.	Cannot exceed the event message size maximum	derived
1529	MM3100	Upon receipt of a Load from File command, MM shall load EEPROM memory based on the following information contained in the command-specified file: a) Destination Address b) Destination Memory Type c) <MISSION_DEFINED> CRC (data only) d) Number of Bytes to Load	Commit changes to EEPROM memory .	LRO, BAT (used command parameters instead of secondary file header)
1531	MM3100.1	If the CRC contained in the file fails validation, MM shall reject the command	Verifies load file integrity	LRO
1533	MM3100.2	If the number of bytes exceeds <PLATFORM_DEFINED, TBD> then the command shall be rejected.	Keeps from running beyond memory type address range. Need to be able to load largest app	LRO
1535	MM3104	Upon receipt of a Dump to File command, MM shall write the data associated with the command-specified EEPROM address ,command-specified number of bytes and calculated	Standard dump from EEPROM memory to a file.	LRO, BAT

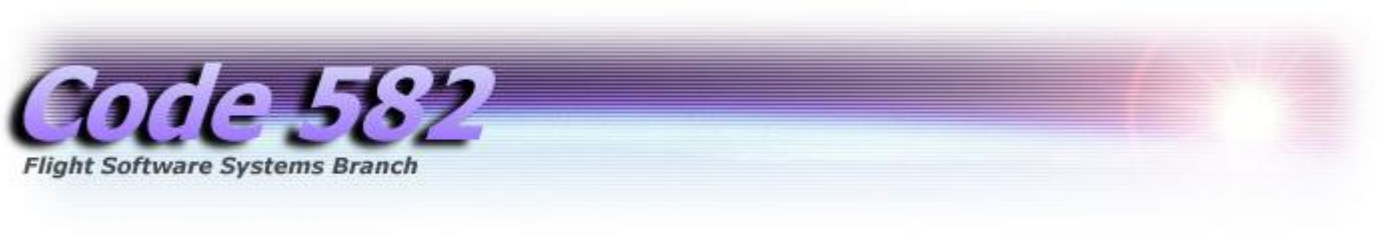
cFS MM Requirements



Apr 3, 2015

ID		Summary		
		<MISSION_DEFINED> CRC to the command-specified file.		
1537	MM3104.1	If the command-specified number of bytes exceeds <PLATFORM_DEFINED, TBD> then the command shall be rejected.	Keep from running beyond memory type address range.	LRO
1539	MM3200	Upon receipt of a Fill command, MM shall fill EEPROM memory with the contents based on the following command-specified parameters: a) Destination Address b) Destination Memory Type c) Number of Bytes to Fill d) 32-bit Fill Pattern	Easy way to write a simple pattern to memory (or tables/data structures). Use Load from File if using a more complicated pattern. SET filled unused memory with the TRAP instruction (if the processor gets lost and wanders off into the weeds, the TRAP instruction will immediately cause an exception/interrupt)	BAT
1541	MM3200.1	If the command-specified number of bytes exceeds <PLATFORM_DEFINED, TBD> then the command shall be rejected.	Keeps from running beyond memory type address range	Derived
1543	MM3300	When writing data to EEPROM memory, MM shall write a maximum of <PLATFORM_DEFINED, TBD> bytes per execution cycle	Segments the load in order to prevent CPU hogging.	LRO
1545	MM3301	When writing EEPROM data to a file, MM shall write a maximum of <PLATFORM_DEFINED, TBD> bytes per execution cycle	Writing to a file can be very slow. Need to segment writing the file.	LRO

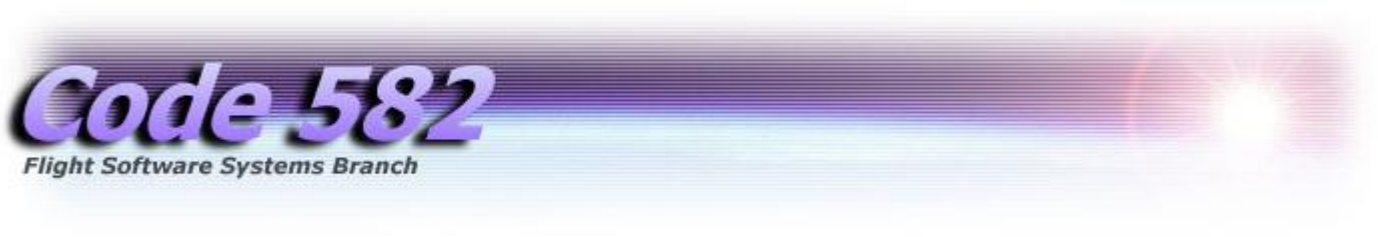
cFS MM Requirements



Apr 3, 2015

ID		Summary		
22947	MM3400	Upon receipt of an Enable EEPROM command, MM shall enable the command specified bank of EEPROM for writing.	Provide capability to patch default FSW image	Derived
22949	MM3500	Upon receipt of a Disable EEPROM command, MM shall disable/lock the command specified bank of EEPROM from being written to.	Protect the EEPROM bank from being inadvertently written	Derived
1688		**5.4 RAM and EEPROM Operations– Copy** These requirements were DELETED.		
1690		**5.5 <OPTIONAL> Memory Mapped I/O** Some hardware requires special interfacing such that the data is written to and read from in very specific ways. Hardware might, for example, need to be written to in 8, 16 or 32 bit chunks of data These interfaces require special processing which is taken care of by the OSAL where it can be abstracted from the specific hardware requirements. Memory mapped I/O that is byte addressable and requires no special code support will be accessed as standard RAM. These		

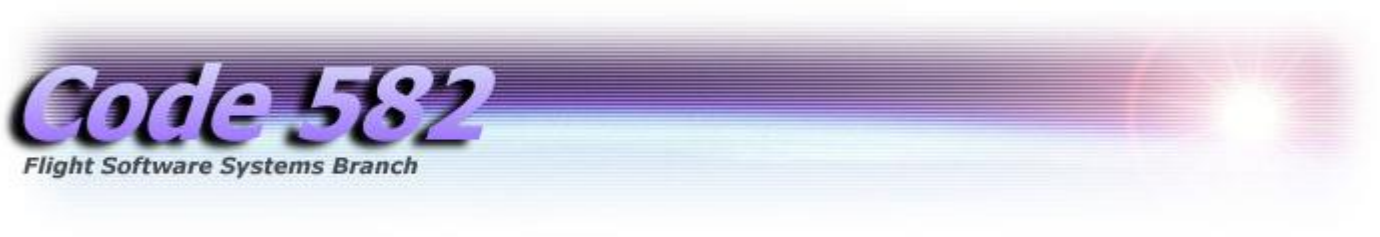
cFS MM Requirements



Apr 3, 2015

ID		Summary		
		requirements, therefore, are OPTIONAL. Missions that don't need these requirements should remove these requirements.		
1553	MM5000	Upon receipt of a Poke command , MM shall write <PLATFORM_DEFINED> bytes of data to the command-specified Memory Mapped I/O address	Memory poke to any Memory Mapped I/O address. The data is written in the required data sizes (8, 16, or 32). Typically the peek function is used to verify the poke.	LRO, BAT
1555	MM5000.1	MM shall confirm a write to the Memory Mapped I/O address by issuing an event message which includes: a) address written b) length of data written c) value of the data written	Confirms memory poke operations	LRO, BAT
1557	MM5002	Upon receipt of a Peek command, MM shall read <PLATFORM_DEFINED> bytes of data from the command-specified Memory Mapped I/O address and generate an event message containing the following data: a) address read b) length of data read c) value of the data read	Memory peek to any Memory Type	LRO, BAT
1559	MM5004	Upon receipt of a Read command, MM shall read the command-specified number of consecutive bytes from the command-specified Memory Mapped I/O memory address and generate an event message containing the data.	Dump in event message in order to provide a quick display (alleviates post processing)	LRO, SDO

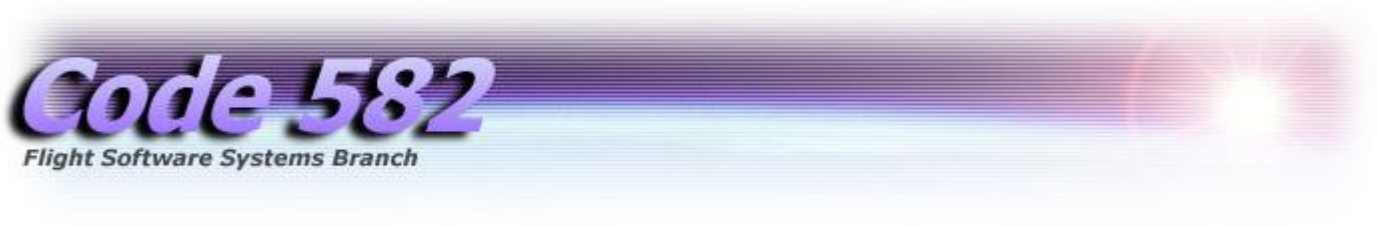
cFS MM Requirements



Apr 3, 2015

ID		Summary		
1561	MM5004.1	If the number of bytes exceeds the maximum event message size then the command shall be rejected.	Cannot exceed the event message size maximum	Derived
1563	MM5100	Upon receipt of a Load from File command, MM shall load Memory mapped I/O, with interrupts enabled during the actual load, based on the following information contained in the command-specified file: a) Destination Address b) Destination Memory Type c) <MISSION_DEFINED> CRC (data only) d) Number of Bytes to Load	Standard load from file into Memory mapped I/O memory. Files are assumed to be binary	LRO, BAT(used command parameters instead of secondary file header)
1565	MM5100.1	If the command-specified CRC fails validation, MM shall reject the command	Verifies load file integrity. Note that the cFE provides functions to calculate the CRC using a desired CRC	LRO
1567	MM5100.2	If the number of bytes exceeds <PLATFORM_DEFINED, TBD> then the command shall be rejected.	Keeps from running beyond memory type address range. Need to be able to load largest app	LRO
1569	MM5104	Upon receipt of a Dump to File command, MM shall write the data associated with the command-specified Memory mapped I/O address, command-specified number of bytes and calculated <MISSION_DEFINED> CRC to the command-specified file.	Standard dump from Memory mapped I/O memory to a file.	LRO, BAT

cFS MM Requirements



Apr 3, 2015

ID		Summary		
1571	MM5104.1	If the command-specified number of bytes exceeds <PLATFORM_DEFINED, TBD> then the command shall be rejected.	Keep from running beyond memory type address range.	LRO
1573	MM5300	Upon receipt of a Fill command, MM shall fill Memory mapped I/O with the contents based on the following command-specified parameters: a) Destination Address b) Destination Memory Type c) Number of Bytes to Fill d) 32-bit Fill Pattern	Fill function. See previous fill requirements for details.	BAT
1575	MM5300.1	If the command-specified number of bytes exceeds <PLATFORM_DEFINED, TBD> then the command shall be rejected.	Keeps from running beyond memory type address range	Derived
1692		**5.6 <OPTIONAL> Symbol Table Support**		
1577	MM7001	Upon receipt of a Write Symbol Table command, MM shall save the system symbol table to an onboard data file	Format of symbol table file will be target OS dependent. Check with Alan to see if should be in cFE or OSAL	BAT
1579	MM7002	Upon receipt of a Symbol-to-Address command, MM shall report the resolved address in telemetry for the command-specified symbol name	Lookup a symbol name. Should this be an event message or a tlm packet?	BAT
1581	MM7004	The MM application shall generate an error event and abort the current operation if any symbolic name argument cannot be resolved to a valid address	Catches bad or misspelled symbol names	Derived

cFS MM Requirements

Code 582
Flight Software Systems Branch

Apr 3, 2015

ID		Summary		
1694		**5.7 Status Reporting**		
1583	MM8000	MM shall generate a housekeeping message containing the following: a) Valid Command Counter b) Command Rejected Counter c) Last command executed d) Address for last command e) Memory Type for last command f) Number of bytes specified by last command g) Filename used in last command h) Data Value for last command (may be fill pattern or peek/poke value)	Housekeeping telemetry to indicate basic MM status including information about the last valid command processed. Note: 'i' is useful for segmented loads and dumps.	Derived
1585	MM9000	Upon initialization of the MM Application, MM shall initialize the following data to Zero a) Valid Command Counter b) Command Rejected Counter c) Last command executed d) address for last command e) Memory Type for last command f) Number of bytes processed by last command g) filename used in last command h) fill pattern specified in last command (if command was a fill command)	No information is preserved across an MM initialization. If, for example, a memory load is interrupted by an App restart or Processor reset then want to abort the load to avoid partial/ erroneous load.	Derived