

Write memory area
File: H_CRP_DHS_3023.xls
Author: S. Manganelli



Procedure Summary

Objectives

This procedure describes the steps needed to write the following memory areas/registers.

- CPU RAM
- Communication RAM
- PM EEPROM
- TTR RAM 0 (non-write protected part)
- ERC 32 registers
- PM COCOS registers

SGM and TTR CROME registers are written via dedicated procedures.

Summary of Constraints

Memory areas/registers are written through TC(6,2); this TC will be delayed when there is an ongoing:

- TC(6,2) Load Memory Using Absolute Addresses
- TC(6,5) Dump Memory Using Absolute Addresses
- TC(6,9) Check Memory Using Absolute Addresses
- TC(8,4,1,1) Copy Memory

Moreover:

- writing to CPU/COCOS registers or EEPROM is enabled and disabled through TC(8,4,7,1).
- writing to ASW/BSW code and constants is enabled and disabled through TC(8,4,7,2).

Spacecraft Configuration

Start of Procedure

n/a

End of Procedure

n/a

Reference File(s)

Input Command Sequences

Output Command Sequences

HRD3023B
HRD3023C
HRD3023D
HRD3023E
HRD3023F
HRD3023G

Referenced Displays

ANDs GRDs SLDs
ZAD51999

Configuration Control Information

Status : Version 2 - Unchanged
Last Checkin: 08/12/08

Write memory area
File: H_CRP_DHS_3023.xls
Author: S. Manganelli

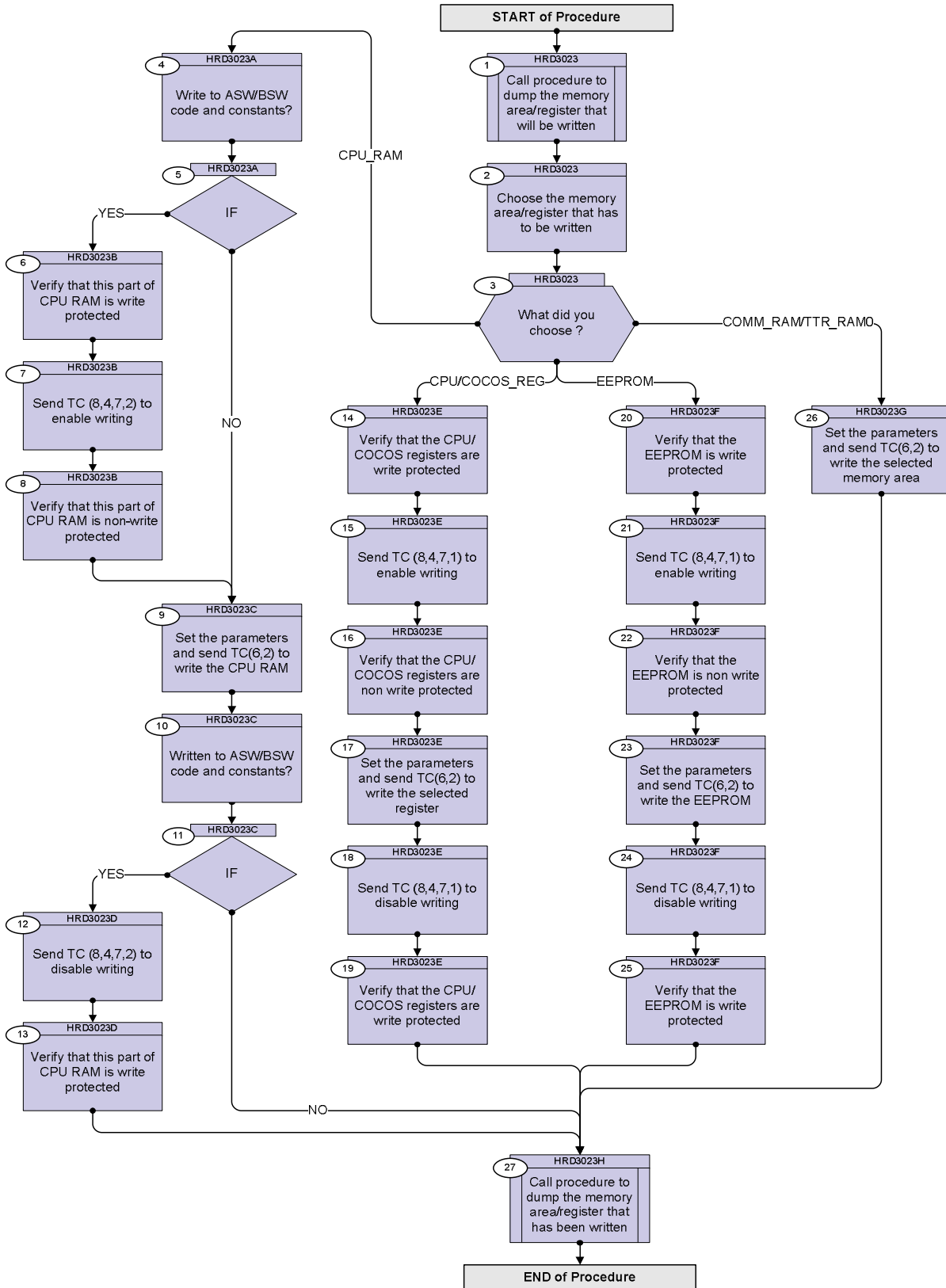


DATE	FOP ISSUE	VERSION	MODIFICATION DESCRIPTION	AUTHOR	SPR REF
28/01/08	1	1	Created	cmevi-hp	
08/12/08	2	2	DB check against OBSW 3_6_2	S. Manganelli	

Write memory area
 File: H_CRP_DHS_3023.xls
 Author: S. Manganelli



Procedure Flowchart Overview



Write memory area
 File: H_CRP_DHS_3023.xls
 Author: S. Manganelli



Step No.	Time	Activity/Remarks	TC/TLM	Display/ Branch
Beginning of Procedure				
<p><i>TC Seq. Name :HRD3023 (Dump memory area.)</i></p> <p><i>TimeTag Type:</i> <i>Sub Schedule ID:</i></p> <p style="text-align: center;"><input type="checkbox"/></p>				
1		Call procedure to dump the memory area/register that will be written		Next Step: 2
		Execute H_CRP_DHS_3022.		
2		Choose the memory area/register that has to be written		Next Step: 3
3		What did you choose ?		Next Step: CPU_RAM 4 CPU/COCOS_REG 14 EEPROM 20 COMM_RAM/TTR_RA M0 26
<p><i>TC Seq. Name :HRD3023A (Dummy sequence.)</i></p> <p><i>TimeTag Type:</i> <i>Sub Schedule ID:</i></p> <p style="text-align: center;"><input type="checkbox"/></p>				
4		Write to ASW/BSW code and constants?		Next Step: 5
5		IF		Next Step: YES 6 NO 9
<p><i>TC Seq. Name :HRD3023B (Enable writing.)</i></p> <p><i>TimeTag Type:</i> <i>Sub Schedule ID:</i></p> <p style="text-align: center;"><input type="checkbox"/></p>				
6		Verify that this part of CPU RAM is write protected		Next Step: 7
		Note that the parameter CPU RAM write protection status (DID_BSW_MEM_ACCESS) is part of the default CDMU diagnostic packet (BSW1); thus to acquire this parameter the packet must be enabled.		

Write memory area
 File: H_CRP_DHS_3023.xls
 Author: S. Manganelli



Step No.	Time	Activity/Remarks	TC/TLM	Display/ Branch															
		<p>In the TC(6,2) it is necessary to set the following parameters:</p> <ul style="list-style-type: none"> - Memory ID: identifier of the destination memory block. - Start Address: start address (in SAUs, with the count starting from zero) within the memory block for loading the data. - N: number of SAUs to be loaded. As the overall length of a TC packet cannot exceed 248 octets, the maximum length of the field "Data" will be 228 octets. N, expressed as number of SAUs, must be compatible with this boundary. - Data: repeated N times, data block to be loaded (in increasing order of SAU). - Checksum: CRC checksum that is used by the on-board user to verify the integrity of the data being loaded. This checksum is generated over the unpadded Data block to be loaded, (i.e. : excluding the optional spare octet) and is additional and different to the CRC word at the end of each packet. 																	
		The fields Memory ID and Start Address are treated as one 32-bit field where the 16 least significant bits of the address is stored in Start Address and the 16 most significant bit in the Memory ID field.																	
		WARNING: The specified address range must not span over several types of memory.																	
		WARNING: only one of the following TCs must be sent.																	
		WARNING: the following TC has to be sent in case of even number of bytes; it is a variable lenght TC that MOIS cannot handle and it is intended to be just an example.																	
		<p>Execute Telecommand</p> <p style="text-align: center;">LoadMem_AbsAdd_EvenByte</p> <p>Command Parameter(s) :</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: right;">Memory_ID</td> <td style="text-align: right;">DH003180</td> <td style="text-align: right;"><hex> (Def)</td> </tr> <tr> <td style="text-align: right;">Start_Address</td> <td style="text-align: right;">DH004180</td> <td style="text-align: right;"><hex> (Def)</td> </tr> <tr> <td style="text-align: right;">N</td> <td style="text-align: right;">DH005180</td> <td style="text-align: right;">1 <hex> (Def)</td> </tr> <tr> <td style="text-align: right;">Data</td> <td style="text-align: right;">DH006180</td> <td style="text-align: right;"><hex> (Def)</td> </tr> <tr> <td style="text-align: right;">Checksum</td> <td style="text-align: right;">DH007180</td> <td style="text-align: right;"><hex> (Def)</td> </tr> </table> <p>TC Control Flags :</p> <p style="text-align: right;">GBM IL DSE --Y -- YYY</p> <p>Subsch. ID : 10 Det. descr. : Load Even number of bytes inMemory Using Absolute Addresses</p> <p>This Telecommand will not be included in the export</p>	Memory_ID	DH003180	<hex> (Def)	Start_Address	DH004180	<hex> (Def)	N	DH005180	1 <hex> (Def)	Data	DH006180	<hex> (Def)	Checksum	DH007180	<hex> (Def)	DC601180	
Memory_ID	DH003180	<hex> (Def)																	
Start_Address	DH004180	<hex> (Def)																	
N	DH005180	1 <hex> (Def)																	
Data	DH006180	<hex> (Def)																	
Checksum	DH007180	<hex> (Def)																	

Write memory area
 File: H_CRP_DHS_3023.xls
 Author: S. Manganelli



Step No.	Time	Activity/Remarks	TC/TLM	Display/ Branch																				
		WARNING: the following TC has to be sent in case of odd number of bytes; it is a variable lenght TC that MOIS cannot handle and it is intended to be just an example.																						
		Execute Telecommand <p style="text-align: right;">LoadMem_AbsAdd_OddByte</p> Command Parameter(s) : <table style="width: 100%; border: none;"> <tr> <td style="width: 30%;">Memory_ID</td> <td style="width: 20%;">DH003180</td> <td style="width: 20%;"></td> <td style="width: 30%; text-align: right;"><hex> (Def)</td> </tr> <tr> <td>Start_Address</td> <td>DH004180</td> <td></td> <td style="text-align: right;"><hex> (Def)</td> </tr> <tr> <td>N</td> <td>DH005180</td> <td></td> <td style="text-align: right;">1 <hex> (Def)</td> </tr> <tr> <td>Data</td> <td>DH006180</td> <td></td> <td style="text-align: right;"><hex> (Def)</td> </tr> <tr> <td>Checksum</td> <td>DH007180</td> <td></td> <td style="text-align: right;"><hex> (Def)</td> </tr> </table> TC Control Flags : <p style="text-align: right;">GBM IL DSE --Y -- YYY</p> Subsch. ID : 10 Det. descr. : Load Odd number of bytes inMemory Using Absolute Addresses This Telecommand will not be included in the export	Memory_ID	DH003180		<hex> (Def)	Start_Address	DH004180		<hex> (Def)	N	DH005180		1 <hex> (Def)	Data	DH006180		<hex> (Def)	Checksum	DH007180		<hex> (Def)	DC600180	
Memory_ID	DH003180		<hex> (Def)																					
Start_Address	DH004180		<hex> (Def)																					
N	DH005180		1 <hex> (Def)																					
Data	DH006180		<hex> (Def)																					
Checksum	DH007180		<hex> (Def)																					
10		Written to ASW/BSW code and constants?		Next Step: 11																				
11		IF		Next Step: YES 12 NO 27																				
TC Seq. Name :HRD3023D (Disable writing.) TimeTag Type: Sub Schedule ID: <input type="checkbox"/>																								
12		Send TC (8,4,7,2) to disable writing		Next Step: 13																				
		Execute Telecommand <p style="text-align: right;">DisableCodeWrite</p> TC Control Flags : <p style="text-align: right;">GBM IL DSE --Y -- YYY</p> Subsch. ID : 10 Det. descr. : Disable write to ASW/BSW code and constants	DC802180																					
13		Verify that this part of CPU RAM is write protected		Next Step: 27																				
		Note that the parameter CPU RAM write protection status (DID_BSW_MEM_ACCESS) is part of the default CDMU diagnostic packet (BSW1); thus to acquire this parameter the packet must be enabled.																						

Write memory area
 File: H_CRP_DHS_3023.xls
 Author: S. Manganelli



Step No.	Time	Activity/Remarks	TC/TLM	Display/ Branch
		When the CDMU receives this Telecommand, it shall store the checksum of the received data, write the data block to the memory at the specified start address and re-read the memory area just written to, calculate and compare the checksum and report successful execution, TM(1,7), or an error report TM(1,8). In case of an error the loaded data shall be discarded.		
		In the TC(6,2) it is necessary to set the following parameters: <ul style="list-style-type: none"> - Memory ID: identifier of the destination memory block. - Start Address: start address (in SAUs, with the count starting from zero) within the memory block for loading the data. - N: number of SAUs to be loaded. As the overall length of a TC packet cannot exceed 248 octets, the maximum length of the field "Data" will be 228 octets. N, expressed as number of SAUs, must be compatible with this boundary. - Data: repeated N times, data block to be loaded (in increasing order of SAU). - Checksum: CRC checksum that is used by the on-board user to verify the integrity of the data being loaded. This checksum is generated over the unpadded Data block to be loaded, (i.e. : excluding the optional spare octet) and is additional and different to the CRC word at the end of each packet. 		
		The fields Memory ID and Start Address are treated as one 32-bit field where the 16 least significant bits of the address is stored in Start Address and the 16 most significant bit in the Memory ID field.		
		WARNING: <ul style="list-style-type: none"> - The specified address range must not span over several types of memory. - No checking is performed that the provided address is a valid register or that register is writeable. Moreover the value of the checksum is irrelevant as no readback and verification of written data is done. - For ERC32 only the mapped registers are accessible. 		
		WARNING: only one of the following TCs must be sent.		
		WARNING: the following TC has to be sent in case of even number of bytes; it is a variable length TC that MOIS cannot handle and it is intended to be just an example.		

Write memory area
 File: H_CRP_DHS_3023.xls
 Author: S. Manganelli



Step No.	Time	Activity/Remarks	TC/TLM	Display/ Branch
		Execute Telecommand <p style="text-align: right;">LoadMem_AbsAdd_EvenByte</p> DC601180 Command Parameter(s) : Memory_ID DH003180 <hex> (Def) Start_Address DH004180 <hex> (Def) N DH005180 1 <hex> (Def) Data DH006180 <hex> (Def) Checksum DH007180 <hex> (Def) TC Control Flags : <p style="text-align: right;"> GBM IL DSE --Y -- YYY</p> Subsch. ID : 10 Det. descr. : Load Even number of bytes inMemory Using Absolute Addresses This Telecommand will not be included in the export		
		WARNING: the following TC has to be sent in case of odd number of bytes; it is a variable length TC that MOIS cannot handle and it is intended to be just an example.		
		Execute Telecommand <p style="text-align: right;">LoadMem_AbsAdd_OddByte</p> DC600180 Command Parameter(s) : Memory_ID DH003180 <hex> (Def) Start_Address DH004180 <hex> (Def) N DH005180 1 <hex> (Def) Data DH006180 <hex> (Def) Checksum DH007180 <hex> (Def) TC Control Flags : <p style="text-align: right;"> GBM IL DSE --Y -- YYY</p> Subsch. ID : 10 Det. descr. : Load Odd number of bytes inMemory Using Absolute Addresses This Telecommand will not be included in the export		
18		Send TC (8,4,7,1) to disable writing		Next Step: 19
		Execute Telecommand <p style="text-align: right;">DisableRegWrite</p> DC801180 TC Control Flags : <p style="text-align: right;"> GBM IL DSE --Y -- YYY</p> Subsch. ID : 10 Det. descr. : Disable write to CPU and COCOS register		
19		Verify that the CPU/COCOS registers are write protected		Next Step: 27

Write memory area
 File: H_CRP_DHS_3023.xls
 Author: S. Manganelli



Step No.	Time	Activity/Remarks	TC/TLM	Display/ Branch
		Note that the parameter CPU/COCOS write protection status (DID_BSW_MEM_ACCESS) is part of the default CDMU diagnostic packet (BSW1); thus to acquire this parameter the packet must be enabled.		
		Verify Telemetry CpuCocosWrtProt DEL0G160	= ENABLED	AND=ZAD51999
<p>TC Seq. Name :HRD3023F (Write EEPROM.)</p> <p>TimeTag Type: Sub Schedule ID: □</p>				
20		Verify that the EEPROM is write protected		Next Step: 21
		Note that the parameter EEPROM write protection status (DID_BSW_MEM_ACCESS) is part of the default CDMU diagnostic packet (BSW1); thus to acquire this parameter the packet must be enabled.		
		Verify Telemetry EEPromWriteProt DEL0J160	= ENABLED	AND=ZAD51999
21		Send TC (8,4,7,1) to enable writing		Next Step: 22
		Execute Telecommand TC Control Flags : Subsch. ID : 10 Det. descr. : Enable write to EEPROM	EnableEEWrite DC806180 GBM IL DSE --Y -- YYY	
22		Verify that the EEPROM is non write protected		Next Step: 23
		Note that the parameter EEPROM write protection status (DID_BSW_MEM_ACCESS) is part of the default CDMU diagnostic packet (BSW1); thus to acquire this parameter the packet must be enabled.		
		Verify Telemetry EEPromWriteProt DEL0J160	= DISABLED	AND=ZAD51999
23		Set the parameters and send TC(6,2) to write the EEPROM		Next Step: 24

Write memory area
 File: H_CRP_DHS_3023.xls
 Author: S. Manganelli



Step No.	Time	Activity/Remarks	TC/TLM	Display/ Branch
		When the CDMU receives this Telecommand, it shall store the checksum of the received data, write the data block to the memory at the specified start address and re-read the memory area just written to, calculate and compare the checksum and report successful execution, TM(1,7), or an error report TM(1,8). In case of an error the loaded data shall be discarded.		
		In the TC(6,2) it is necessary to set the following parameters: <ul style="list-style-type: none"> - Memory ID: identifier of the destination memory block. - Start Address: start address (in SAUs, with the count starting from zero) within the memory block for loading the data. - N: number of SAUs to be loaded. As the overall length of a TC packet cannot exceed 248 octets, the maximum length of the field "Data" will be 228 octets. N, expressed as number of SAUs, must be compatible with this boundary. - Data: repeated N times, data block to be loaded (in increasing order of SAU). - Checksum: CRC checksum that is used by the on-board user to verify the integrity of the data being loaded. This checksum is generated over the unpadded Data block to be loaded, (i.e. : excluding the optional spare octet) and is additional and different to the CRC word at the end of each packet. 		
		The fields Memory ID and Start Address are treated as one 32-bit field where the 16 least significant bits of the address is stored in Start Address and the 16 most significant bit in the Memory ID field.		
		WARNING: The specified address range must not span over several types of memory.		
		WARNING: only one of the following TCs must be sent.		
		WARNING: the following TC has to be sent in case of even number of bytes; it is a variable lenght TC that MOIS cannot handle and it is intended to be just an example.		

Write memory area
 File: H_CRP_DHS_3023.xls
 Author: S. Manganelli



Step No.	Time	Activity/Remarks	TC/TLM	Display/ Branch
		Execute Telecommand <p style="text-align: right;">LoadMem_AbsAdd_EvenByte</p> Command Parameter(s) : Memory_ID DH003180 <hex> (Def) Start_Address DH004180 <hex> (Def) N DH005180 1 <hex> (Def) Data DH006180 <hex> (Def) Checksum DH007180 <hex> (Def) TC Control Flags : <p style="text-align: right;"> GBM IL DSE --Y -- YYY</p> Subsch. ID : 10 Det. descr. : Load Even number of bytes inMemory Using Absolute Addresses This Telecommand will not be included in the export	DC601180	
		WARNING: the following TC has to be sent in case of odd number of bytes; it is a variable length TC that MOIS cannot handle and it is intended to be just an example.		
		Execute Telecommand <p style="text-align: right;">LoadMem_AbsAdd_OddByte</p> Command Parameter(s) : Memory_ID DH003180 <hex> (Def) Start_Address DH004180 <hex> (Def) N DH005180 1 <hex> (Def) Data DH006180 <hex> (Def) Checksum DH007180 <hex> (Def) TC Control Flags : <p style="text-align: right;"> GBM IL DSE --Y -- YYY</p> Subsch. ID : 10 Det. descr. : Load Odd number of bytes inMemory Using Absolute Addresses This Telecommand will not be included in the export	DC600180	
24		Send TC (8,4,7,1) to disable writing		Next Step: 25
		Execute Telecommand <p style="text-align: right;">DisableEEWrite</p> TC Control Flags : <p style="text-align: right;"> GBM IL DSE --Y -- YYY</p> Subsch. ID : 10 Det. descr. : Disable write to EEPROM	DC805180	
25		Verify that the EEPROM is write protected		Next Step: 27
		Note that the parameter EEPROM write protection status (DID_BSW_MEM_ACCESS) is part of the default CDMU diagnostic packet (BSW1); thus to acquire this parameter the packet must be enabled.		

Write memory area
 File: H_CRP_DHS_3023.xls
 Author: S. Manganelli



Step No.	Time	Activity/Remarks	TC/TLM	Display/ Branch
		Verify Telemetry EEPromWriteProt DEL0J160	= ENABLED	AND=ZAD51999
<p>TC Seq. Name :HRD3023G (Write COMM/TTR RAM)</p> <p>TimeTag Type: Sub Schedule ID:</p> <p style="text-align: center;">□</p>				
26		Set the parameters and send TC(6,2) to write the selected memory area		Next Step: 27
		When the CDMU receives this Telecommand, it shall store the checksum of the received data, write the data block to the memory at the specified start address and re-read the memory area just written to, calculate and compare the checksum and report successful execution, TM(1,7), or an error report TM(1,8). In case of an error the loaded data shall be discarded.		
		<p>In the TC(6,2) it is necessary to set the following parameters:</p> <ul style="list-style-type: none"> - Memory ID: identifier of the destination memory block. - Start Address: start address (in SAUs, with the count starting from zero) within the memory block for loading the data. - N: number of SAUs to be loaded. As the overall length of a TC packet cannot exceed 248 octets, the maximum length of the field "Data" will be 228 octets. N, expressed as number of SAUs, must be compatible with this boundary. - Data: repeated N times, data block to be loaded (in increasing order of SAU). - Checksum: CRC checksum that is used by the on-board user to verify the integrity of the data being loaded. This checksum is generated over the unpadded Data block to be loaded, (i.e. : excluding the optional spare octet) and is additional and different to the CRC word at the end of each packet. 		
		The fields Memory ID and Start Address are treated as one 32-bit field where the 16 least significant bits of the address is stored in Start Address and the 16 most significant bit in the Memory ID field.		
		WARNING: The specified address range must not span over several types of memory.		
		WARNING: only one of the following TCs must be sent.		

Write memory area
 File: H_CRP_DHS_3023.xls
 Author: S. Manganelli



Step No.	Time	Activity/Remarks	TC/TLM	Display/ Branch																				
		WARNING: the following TC has to be sent in case of even number of bytes; it is a variable lenght TC that MOIS cannot handle and it is intended to be just an example.																						
		Execute Telecommand <p style="text-align: right;">LoadMem_AbsAdd_EvenByte</p> DC601180 Command Parameter(s) : <table style="width: 100%; border: none;"> <tr> <td style="width: 30%;">Memory_ID</td> <td style="width: 20%;">DH003180</td> <td style="width: 20%;"></td> <td style="width: 30%; text-align: right;"><hex> (Def)</td> </tr> <tr> <td>Start_Address</td> <td>DH004180</td> <td></td> <td style="text-align: right;"><hex> (Def)</td> </tr> <tr> <td>N</td> <td>DH005180</td> <td></td> <td style="text-align: right;">1 <hex> (Def)</td> </tr> <tr> <td>Data</td> <td>DH006180</td> <td></td> <td style="text-align: right;"><hex> (Def)</td> </tr> <tr> <td>Checksum</td> <td>DH007180</td> <td></td> <td style="text-align: right;"><hex> (Def)</td> </tr> </table> TC Control Flags : <p style="text-align: right;">GBM IL DSE --Y -- YYY</p> Subsch. ID : 10 Det. descr. : Load Even number of bytes inMemory Using Absolute Addresses This Telecommand will not be included in the export	Memory_ID	DH003180		<hex> (Def)	Start_Address	DH004180		<hex> (Def)	N	DH005180		1 <hex> (Def)	Data	DH006180		<hex> (Def)	Checksum	DH007180		<hex> (Def)		
Memory_ID	DH003180		<hex> (Def)																					
Start_Address	DH004180		<hex> (Def)																					
N	DH005180		1 <hex> (Def)																					
Data	DH006180		<hex> (Def)																					
Checksum	DH007180		<hex> (Def)																					
		WARNING: the following TC has to be sent in case of odd number of bytes; it is a variable lenght TC that MOIS cannot handle and it is intended to be just an example.																						
		Execute Telecommand <p style="text-align: right;">LoadMem_AbsAdd_OddByte</p> DC600180 Command Parameter(s) : <table style="width: 100%; border: none;"> <tr> <td style="width: 30%;">Memory_ID</td> <td style="width: 20%;">DH003180</td> <td style="width: 20%;"></td> <td style="width: 30%; text-align: right;"><hex> (Def)</td> </tr> <tr> <td>Start_Address</td> <td>DH004180</td> <td></td> <td style="text-align: right;"><hex> (Def)</td> </tr> <tr> <td>N</td> <td>DH005180</td> <td></td> <td style="text-align: right;">1 <hex> (Def)</td> </tr> <tr> <td>Data</td> <td>DH006180</td> <td></td> <td style="text-align: right;"><hex> (Def)</td> </tr> <tr> <td>Checksum</td> <td>DH007180</td> <td></td> <td style="text-align: right;"><hex> (Def)</td> </tr> </table> TC Control Flags : <p style="text-align: right;">GBM IL DSE --Y -- YYY</p> Subsch. ID : 10 Det. descr. : Load Odd number of bytes inMemory Using Absolute Addresses This Telecommand will not be included in the export	Memory_ID	DH003180		<hex> (Def)	Start_Address	DH004180		<hex> (Def)	N	DH005180		1 <hex> (Def)	Data	DH006180		<hex> (Def)	Checksum	DH007180		<hex> (Def)		
Memory_ID	DH003180		<hex> (Def)																					
Start_Address	DH004180		<hex> (Def)																					
N	DH005180		1 <hex> (Def)																					
Data	DH006180		<hex> (Def)																					
Checksum	DH007180		<hex> (Def)																					
TC Seq. Name :HRD3023H (Read area/register)																								
TimeTag Type: Sub Schedule ID: <input type="checkbox"/>																								
27		Call procedure to dump the memory area/register that has been written		Next Step: END																				
		Execute H_CRP_DHS_3022.																						
End of Procedure																								