*SOHO CDS* *Technical Note*

# An EGSE Implementation of the FM Commanding Scheme

CDS-MSSL-TN-0132
Version 1.0

E R Breeveld MSSL, 23 November, 1994.

Distribution: MKC, RAH, DJP, JP, CDP (RAL)
KFJ, RAG, KN, ERB (MSSL)
RAH please distribute further if necessary

## Changes:

| Version | Date | Comment |
|---|---|---|
| 1.0 | 18/5/93 | Completely restructured |

## Contents

## 1. Summary

This document details a suggested EGSE implementation of the FM commanding scheme, with examples. These examples are very simple, using a few commands detailed in the proposed structure for the FM command database.

## 2. Requirements

The EGSE implementation must allow all possible SOHO CDS commands to be sent.

## 3. Command Structure

Most CDS commands must be packed into an On-Board Data Handling (OBDH) command block before transmission to the spacecraft. Exceptions to this rule are for hardware or OBDH commands that are not sent from the EGSE.

A CDS OBDH command block is defined by a collection of 16 bit words, starting with the ESA defined MLA (memory load A) word and followed by at least two 16 bit data words. The two data words fill the MLB (memory load B).

| EID definition | | CDS Terminology | | Size |
|---|---|---|---|---|
| OBDH command block | MLA | MLA | | 1 word (16 bits) |
| | MLB | Command block | Block Header | 1 word |
| | | | Block Body (arguments) | 1 to 29 words |
| | Checksum | Checksum | | 1 word |

The Block Body must have at least one 16 bit word (argument) to make up the full command block. Similar command blocks (with the same block header) can be combined into a single block. The largest block body size is 29 from the ground, and the minimum is one.

The word *command* (as opposed to *command block)* is ambiguous, as it could either mean a whole command block, or an individual argument in the block body. It should be avoided if possible. Unfortunately, the EID definition of a command is an MLA word that is not followed by any MLB words.

In the following sections, the command blocks will be illustrated as individual bits within each word, as follows:

| Bit 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | Bit15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MLA | | | | | | | | | | | | | | | |
| Block Header | | | | | | | | | | | | | | | |
| Argument 1 | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | } Block Body | | | | | |
| Argument 'n' | | | | | | | | | | | | | | | |
| Checksum | | | | | | | | | | | | | | | |

The checksum is as defined in the EID. If it is not supplied, the ground system should automatically append it.

## 4. Block Header

Each block has a header word, which states the on-board *destination* (4 bits), *function* (5 bits), and *number of arguments to follow* (7 bits). There is no end of block argument, other than the checksum.

The *destination* is the module with the CDHS on-board software that the block is intended for. The *function* specifies what type of action is to be taken, and the *arguments* specify the exact action.

Thus the structure is as follows:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MLA | | | | | | | | | | | | | | | |
| Destination | | | | Function | | | | | Number of arguments ('n') | | | | | | |
| Argument 1 | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | |
| Argument 'n'. | | | | | | | | | | | | | | | |
| Checksum | | | | | | | | | | | | | | | |

This structure allows the commanding of up to 16 destinations, with 32 functions at each destination. Note that many more command blocks can be defined by each argument (up to 65536 different combinations per argument).

This is very similar to the previous command type 2 commands. Note that there is no end of block command as in the old type 2 commands, just the checksum.

## 5. Command Database

The suggested FM command database structure is shown below. It is a simplification of the EM database, but unlike the EM database, characters within the command mnemonic do not need to be interpreted.

Suggested structure:

| Mnemonic | Comment | Destination | Function | Command argument | No. of params | Hazardous command | Param 1 bit start | Param 1 no. of bits |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| CBDFILL | ... | 4 | 2 | 0x | 1 | N | 0 | 16 |
| CBDTIM1 | ... | N/A | N/A | 0x | 1 | N | 0 | 16 |
| CBDTIM2 | ... | N/A | N/A | 0x | 1 | N | 0 | 16 |
| | | | | | | | | |
| CBLFILL | ... | 5 | 2 | 0x | 1 | N | 0 | 16 |
| CBLX1 | ... | N/A | N/A | 0x | 1 | N | 0 | 16 |
| CBLX2 | ... | N/A | N/A | 0x | 1 | N | 0 | 16 |
| CBLY1 | ... | N/A | N/A | 0x | 1 | N | 0 | 16 |
| CBLY2 | ... | N/A | N/A | 0x | 1 | N | 0 | 16 |
| | | | | | | | | |
| CBSFILL | ... | 5 | 3 | 0x | 1 | N | 0 | 16 |
| | | | | | | | | |
| CBVNOOP | ... | 2 | 4 | 0x8000 | 0 | N | | |
| | | | | | | | | |
| CBEGHV4N | ... | 2 | 8 | 0xABAB | 0 | Y | | |
| CBEGHV4F | ... | 2 | 8 | 0xCBCB | 0 | N | | |
| | | | | | | | | |

NB: Those commands with "N/A" as destination and action do not have block headers, and are only used within fill commands, see below. The EGSE should not allow such commands to be sent outside fill command blocks.

Further columns would be needed at the right of the table for extra parameters; they are not shown here for brevity.

## 6. Command Implementation

Details of the three basic types of command blocks (single argument, multiple argument, and fill command blocks) are illustrated below.

### 6.1. Single argument command blocks

Single commands can be constructed by taking the destination and function to form the header word; followed by the single command argument word (the command body). There is no end of block command as with the EM commanding.

Thus the command:

| | | |
|---|---|---|
| CBEGHV4N | ! | (16 bit header + 16 bit argument) |

would translate to:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MLA | | | | | | | | | | | | | | | |
| Destination = 2 | | | | Function = 8 | | | | | Number of arguments ('n') = 1 | | | | | | |
| Argument 1 = 0xABAB | | | | | | | | | | | | | | | |
| Checksum | | | | | | | | | | | | | | | |

The final command would then look like:

| | |
|---|---|
| (MLA) | ! MLA as per EID |
| 0x2401 | ! Destination = 2, Function = 8 (5 bits), no = 1 (7 bits) |
| 0xABAB | ! Command CBEGHV4N |
| (checksum) | |

## 6.2. Multiple argument command blocks

Block commands can be implemented by taking the destination and function and following the header by the arguments. All the commands in the block are executed immediately on reception on-board. **Only commands with the same destination and function can be blocked together**.

On the EGSE, multiple argument command blocks are encoded within a *start_block, commands, end_block* construction. If more than 29 command block arguments are attempted, then the EGSE should signal an error.

Thus the command block:

```
start_block
          CBEGHV4N             ! 16 bit header + 16 bit argument
          CBEGHV4F             ! Header not needed, 16 bit argument only
end_block
```

would translate to:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MLA | | | | | | | | | | | | | | | |
| Destination = 2 | | | | Function = 8 | | | | | Number of arguments ('n') = 2 | | | | | | |
| Argument 1 = 0xABAB | | | | | | | | | | | | | | | |
| Argument 2 = 0xCBCB | | | | | | | | | | | | | | | |
| Checksum | | | | | | | | | | | | | | | |

The final command would then look like:

```
(MLA)               ! MLA as per EID
0x2402              ! Destination = 2, Function = 8 (5 bits), no = 2 (7 bits)
0xABAB              ! Command CBEGHV4N
0xCBCB              ! Command CBEGHV4F
(checksum)
```

## 6.3. Fill Command blocks

Fill commands are used to fill CDHS tables with commands or information. On the EGSE, fill command blocks are encoded within a *start_fill, commands, end_fill* construction. There is no restriction on the destination or function of commands within the fill command block. Under come circumstances, for instance with deferred command fill command blocks, the fill blocks may contain multiple command blocks.

For this example, deferred commands are collections of commands, each preceded by a 32 bit 'local on-board time' (as per EID), in seconds. They are blocked with a header and a location. The exact implementation of deferred commands is not relevant, changes can be made via the command database without changing the EGSE code.

Note that the location command (CBDFILL) has the destination 'deferred command' and function 'fill', but individual deferred commands (e.g., CBVNOOP, CBEGHV4N) do not.

Thus the deferred command store fill instructions:

```
start_fill
        CBDFILL = 25               ! Location in the store
        CBDTIM1 = 0x0034           ! High order time for commands
        CBDTIM2 = 0x1682           ! Low order time for commands
        start_block
                CBEGHV4N           ! 1st command
                                   !       (16 bit header + 16 bit argument)
                CBEGHV4F           ! 2nd command (16 bit argument)
        end_block
        CBDTIM1 = 0x0034           ! High order time for command
        CBDTIM2 = 0x17A6           ! Low order time for command
        CBVNOOP                    ! 3rd command (unblocked)
                                   !       (16 bit header + 16 bit argument)
end_fill
```

would be translated into:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MLA | | | | | | | | | | | | | | | |
| Destination = 4 | | | | Function = 2 | | | | | Number of arguments ('n') = 10 | | | | | | |
| Argument 1 = 25 | | | | | | | | | | | | | | | |
| Argument 2 = 0x0034 | | | | | | | | | | | | | | | |
| Argument 3 = 0x1682 | | | | | | | | | | | | | | | |
| Defer. dest. = 2 | | | | Defer. action = 8 | | | | | Defer. no. of arguments ('n') = 2 | | | | | | |
| Defer. command 1 = 0xABAB | | | | | | | | | | | | | | | |
| Defer. command 2 = 0xCBCB | | | | | | | | | | | | | | | |
| Argument 7 = 0x0034 | | | | | | | | | | | | | | | |
| Argument 8 = 0x17A6 | | | | | | | | | | | | | | | |
| Defer. dest. = 2 | | | | Defer. function = 4 | | | | | Defer. no. of arguments ('n') = 1 | | | | | | |
| Defer. command 3 = 0x8000 | | | | | | | | | | | | | | | |
| Checksum | | | | | | | | | | | | | | | |

The final command would then look like:

| | |
|---|---|
| (MLA) | ! MLA as per EID |
| 0x410A | ! Destination = 4, Function = 2 (5 bits), no = 10 (7 bits) |
| 0x0019 | ! Location = 25 (19hex) |
| 0x0034 | ! Time 1 (most sig. word) |
| 0x1682 | ! Time 1 (least sig. word) |
| 0x2402 | ! Destination = 2, Function = 8 (5 bits), no = 2 (7 bits) |
| 0xABAB | ! Command CBEGHV4N |
| 0xCBCB | ! Command CBEGHV4F |
| 0x0034 | ! Time 2 (most sig. word) |
| 0x17A6 | ! Time 2 (least sig. word) |
| 0x2201 | ! Destination = 2, Function = 4 (5 bits), no = 1 (7 bits) |
| 0x8000 | ! Command CBVNOOP |
| (checksum) | |

### 6.4. Other lists

There is much room for expansion here, without changing the EGSE code.

This scheme would easily accommodate Error actions, Raster Parameter Lists, Health Monitor values, Out-of-Limits actions, etc.; all using the *start_fill, end_fill* instructions. Only additional entries in the EGSE database would be needed.

## 7. Reference Documents

| | | |
|---|---|---|
| 1. | CDS-MSSL-TN-0131 | Modified Commanding Scheme, Version 1.2 |
| 2. | PLP/410S/EID A | Experiment Interfaces Document, Issue 1, Rev.0 |