

STEREO *IMPACT*

IDPU Command Verification System

CmdVerification_A.doc
Version A – 2002-May-29

David Curtis, UCB IMPACT Project Manager

Document Revision Record

Rev.	Date	Description of Change	Approved By
A	2002-May-29	Preliminary Draft	-

Distribution List

Dave Curtis, UCB

Mike Hashii, UCB

Peter Schroeder, UCB

Table of Contents

Document Revision Record	i
Distribution List	i
1. Introduction	1
1.1. <i>Document Conventions</i>	<i>1</i>
1.2. <i>Applicable Documents</i>	<i>1</i>
2. STEREO IMPACT/PLASTIC Command Routing	1
3. IDPU Command Verification	1
4. GSE Command Verification	2

1. Introduction

This document describes the command verification scheme to be used by the STEREO IMPACT IDPU, and how the GSE/POCC should handle command verification. The idea is that the GSE should keep track of what commands have been verified by the IDPU and report a warning if a command is dropped or an extra command is received.

1.1. Document Conventions

In this document, **TBD** (To Be Determined) means that no data currently exists. A value followed by **TBR** (To Be Resolved) means that this value is preliminary. In either case, the value is typically followed by UCB, indicating who is responsible for providing the data, and a unique reference number.

1.2. Applicable Documents

The following documents include drawings and STEREO Project policies, and are part of this specification. In the event of a conflict between this Specification and the following documents, this Specification takes precedence. Most documents and drawings can be found on the Berkeley STEREO/IMPACT FTP site:

<http://sprg.ssl.berkeley.edu/impact/dwc/>

Or on the APL STEREO web site at:

<https://sd-forum.jhuapl.edu/stereo/>

1. IMPACT_ICD_rev11-16-01 - Spacecraft to IMPACT ICD
2. MOC-POC_ICD_update_209-10-01 - MOC to POC ICD
3. Specifications/CmdFMT_A - IMPACT Command Formats

2. STEREO IMPACT/PLASTIC Command Routing

All STEREO commands are formatted into CCSDS packets as described in Reference 2 and 3. IMPACT and PLASTIC GSE/POCC send commands to the IDPU via the MOC and spacecraft (or via the APL Spacecraft Emulator). Each command contains a CCSDS header that includes an ApID and a sequence counter. The ApID for IMPACT must be in the range 200-27F hex, and for PLASTIC in the range 300-37F hex. Different ApIDs are routed to different tasks in the IDPU, as described in reference 3. The command sequence counter shall be generated by the GSE/POCC, and shall increment by one for each command sent, irrespective of ApID (note that separate sequence counters shall be used for time-tagged and delayed commands).

3. IDPU Command Verification

The IDPU responds to received commands with three parameters in the housekeeping packet:

- CommandCount – a count of commands received since IDPU reset, modulo 256
- CommandLastID – the coded ApID from the latest command received by the IDPU
- CommandLastSeq – the 8LSB of the sequence counter from the last command received by the IDPU

The CommandLastID is an 8-bit quantity related to the ApID as follows:

Bits 6:0 of CommandLastID = Bits6:0 of the command ApID

Bit 7 of CommandLastID = Bit 8 of the command ApID

Thus PLASTIC commands will have bit 7 if CommandLastID set, and IMPACT will have that bit reset.

The IDPU does not differentiate between time-tagged and real-time commands. Further, it is possible that commands may be received by the IDPU from more than one POCC/GSE (IMPACT or PLASTIC). So there it is possible that command verification will be confused by commands in the IDPU verification not recently sent by the POCC/GSE. However, most of the time, at least during ground test and commissioning, there will be very few time-tagged commands and only one POCC/GSE will command at a time. The verification scheme needs to handle the special cases when unexpected commands are verified.

4. GSE Command Verification

The GSE shall maintain a FIFO containing all real-time commands sent (and verified by the MOC or Emulator ARR message), including the ApID and sequence counter. A GSE display shall indicate how many commands are in the verification FIFO at any time, and a display listing the contents of the FIFO shall be available.

When a housekeeping packet containing the IDPU command verification is received, the following will be performed by the GSE:

NumCommands = CommandCount – PreviousCommandCount

If (NumCommads == 0) then {

 If ((CommandLastSeq == PreviousCommandSeq) AND
 (CommandLastID == PreviousCommandID)) then

 Do Nothing – no commands received.

 Else {

 /* Assuming the command counter has over-flowed, look in command
 queue back 256*N commands, up to the size of the FIFO. */

 If ((FIFOSeq[256*N] == CommandLastSeq) AND
 (FIFOID[256*N] == CommandLastID)) then {

 Release 256*N commands from the FIFO

 Display 256*N commands verified.

 }

 Else call VerificationError

 }

```

}
Else {
    /* Look in command queue for command (check every 256 commands in case of
    counter over-flow) */
    If ( NumCommands < 0 ) then NumCommands = NumCommands+256
    If (( CommandLastSeq == FIFOSeq[NumCommands + 256*N]) AND
        ( CommandLastID == FIFOID[NumCommands + 256*N])) then
        Release NumCommands + 256*N commands from the FIFO
        Display NumCommands + 256*N commands verified
    }
    Else call VerificationError
}
Display CommandLastID, CommandLastSeq, and CommandCount
    (use the command database to decode the CommandID)
PreviousCommandCount = CommandCount
PreviousSeq = CommandLastSeq
PreviousID = CommandLastID

VerificationError Handler {
    If ( Verification FIFO is empty) then {
        Report Warning: NumCommands Unexpected Commands; append the
        decoded CommandlastID, CommandLastSeq and CommandCount in the
        error message.
    }
    Else {
        /* Search the whole queue (starting with the oldest command) for a
        command matching CommandLastSeq and CommandLastID. */
        If (( CommandLastSeq == FIFOSeq[N]) AND
            ( CommandLastID == FIFOID[N])) then {
            Release N commands from the Verification FIFO
            Display N commands Verified
            Report Warning: N-NumCommands Unexpected Commands (or if
            N-NumCommands is negative, report NumCommands-N
            commands dropped); append the decoded CommandlastID,
            CommandLastSeq and CommandCount in the error message.
        }
        Else {
            /* Here we have a problem. We don't know how many of the
            commands in the verification FIFO are included in
            NumCommands. Assume none and report extra commands; if
            some were included, we will get a "commands dropped" warning
            in the future. */
            Report Warning: NumCommands Unexpected Commands;
            append the decoded CommandlastID, CommandLastSeq and
            CommandCount in the error message.
        }
    }
}

```

```
    }  
  }  
}
```

Here I use FIFO[n] to indicate the nth command from the top of the FIFO, so that FIFO[1] is the oldest unverified command. Also I don't explicitly call out the loop over N through the FIFO.