**RMD Telemetry Dictionary**

This document will serve to define and explain the format and syntax of all possible FSW commands to the Mini-NS.

Applicable Versions of the FSW:

|  |  |
| --- | --- |
| FSW Version | Applicable (Yes/No) |
| Version 1 | No |
| Version 2 | No |
| Version 3 | No |
| Pre-Version 4 | Yes |

[1. FSW Command Template 2](#_Toc22723638)

[2. Format Requirements 3](#_Toc22723639)

[3. Command Dictionary 4](#_Toc22723640)

[3.1 Get Status (NO-OP) 4](#_Toc22723641)

[3.2 Initialize Data Acquisition Run 9](#_Toc22723642)

[3.3 LS Files Command 10](#_Toc22723643)

[4. Input Command Packet Structure 12](#_Toc22723644)

[4.1 General Structure 12](#_Toc22723645)

[5. Data Product Packet Type Dictionary 14](#_Toc22723646)

[5.1 Event-by-event Data Product (EVT) 14](#_Toc22723647)

[6. Appendix 15](#_Toc22723648)

[6.1 Byte-by-byte Breakdown of Mini-NS CCSDS Packet 15](#_Toc22723649)

[6.2 APID Table for Mini-NS CCSDS Packets 16](#_Toc22723650)

[6.3 Mini-NS States and Modes 16](#_Toc22723651)

[7. SD Card File Format 18](#_Toc22723652)

[7.1 Files on the SD Card 18](#_Toc22723653)

[7.2 System File Format Breakdown 18](#_Toc22723654)

# FSW Command Template

**Function Name**

Description: A brief description of the command.

Syntax: This will give a general idea for the syntax of an FSW command. All fields will be specified and keywords/symbols for those fields are identified.

Example: An example of the syntax to use.

Command Length: The length of the command in bytes. Fields in the commands have fixed length. The length of a command includes the newline character which is used to terminate Mini-NS commands.

Field Format:

* A breakdown of each field listed in the command syntax. The data type (ie. INT, FLOAT, CHAR), the allowed range, as well as a description for that field are provided.
* Each command starts with a naming sequence, which is an ASCII character string
* Most commands include a detector number field, which allows commands to be sent to one of the two Mini-NS boards.

Byte Conversion: A conversion value to convert the ASCII command to a hex sequence for packing into a Payload Command Packet, RAW\_BYTES section.

Loop Status: Yes/no whether the system will begin a looping sequence after receiving this command.

Latency: The latency associated with this command. This number indicates the amount of time to wait before issuing any further commands.

Sub-commands: A list of commands which may be issued while this command is looping, if applicable.

Return Type: All Mini-NS return values come in the form of CCSDS packets. The APID of that packet indicates what type it is. Possible values are listed in this section.

Byte Breakdown: A description of all of the byte fields within the event or packet.

# Format Requirements

Commands are sent to the instrument as ASCII statements. Bytes received by the RS422 are interpreted as ASCII characters.

All commands have parameters which follow the command. Those parameters can be either integers, floats, or strings, as documented below. All parameters are required, and will be parsed as the expected data type. Any input given to the command will be parsed as the data type given. If the number of parameters is too few or too many, the command will be rejected with a Command Failure packet (0x01).

* **The parsing relies on the “\_” (underscore) character.**

Parameters and naming sequences not separated by the underscore will not be recognized by the system.

The number which follows almost all the commands is the Detector ID (0 or 1) indicated as “#” (pound) in the command list below. The Mini-NS is comprised of two separate boards and commands need to be addressed to one or the other board.

Each board will parse, but silently reject commands which are addressed to the other board.

* **Commands must be terminated with a newline “\n” character.**

On the completion of a command, the return value will be written to the RS422 wrapped in a CCSDS packet; these return values are listed below each command. The success or failure of a command is indicated via the APID bytes (ie. the type of packet returned), see the appendix. All commanded functions return a CCSDS packet upon completion, but some output a packet other than SUCCESS if successful (eg. GETSTAT returns an SOH packet instead of a SUCCESS packet).

# Command Dictionary

## Get Status (NO-OP)

Description: Request the Statement of Health information of the instrument. The Mini-NS will output a single SOH packet.

Syntax: MNS\_GETSTAT\_#

Example: MNS\_GETSTAT\_0\n

Command Length: 14 bytes

Field Format:

|  |  |  |  |
| --- | --- | --- | --- |
| Format | Description | Data Type | Value Range |
| MNS\_GETSTAT | Naming Sequence | ASCII string | - |
| # | Detector Number | Integer | 0, 1 |

Byte Conversion: N/A

Loop Status: No

Latency: N/A

Sub-commands: None

Return Type: Statement of Health (0x22)

Byte Breakdown: Table 2 lays out the bytes for the first packet only for the Level 3 FSW currently on the RMD engineering board (9-30-2019).

The value of byte 4 depends on the CCSDS flags and the detector number:

* Detector 0 = 0x0A
* Detector 1 = 0x0B
* NOTE: the Neutron-1 will always show 0x0A

The value for byte 5 is the APID which is 0x22 for SOH packets.

The value for byte 6 will always be 0xC0, as SOH packets are always unsegmented (group flags) and the sequence count is always 1, leaving the MSB for sequence count zeroes. The value for byte 7 will always be 0x01.

The packet length (byte 8, 9) will always be 41 for SOH packets so byte 8 is always 0x00 and byte 9 is always 0x29.

The first 3 fields within an SOH packet are comprised of the current temperature of each temperature sensor within the detector; each detector has a temperature sensor on the digital board, on the analog board, and on one module. The temperature is reported as a signed 4-byte wide field.

Aggregated neutrons is a signed 4-byte wide field which reports the aggregated total neutron counts over the most recent science run. This number will be incrementing during DAQ mode, but static all other times.

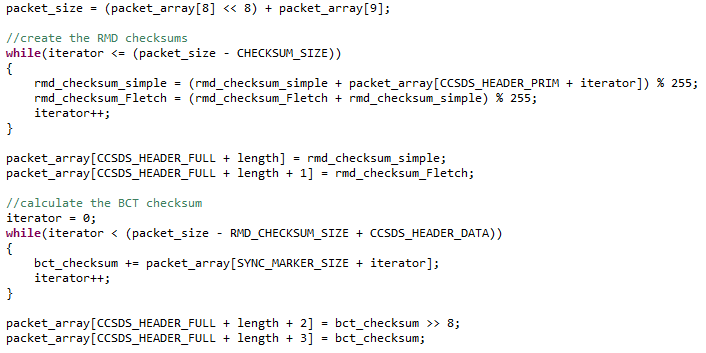
The aggregated neutrons field will be reset to 0 when “MNS\_DAQ” is commanded.

Local time is an unsigned 4-byte wide field which reports the processor time in units of clock cycles.

The Mode Byte informs the flight computer what Mode the detector is currently in. There are 7 power-on modes (also referred to as states) which the detector can be in and each state accepts a subset of the Mini-NS commands. See Table 3 and Figure 2 for the specifics.

The ID Number and Run Number fields are only relevant when the SOH reports that it is in Pre-DAQ, DAQ, or WF mode. The ID and Run numbers tell the user what the file and folder names for the DAQ or WF runs are. When in Pre-DAQ or DAQ mode, both the ID and Run numbers will be reported. When in WF mode, the ID number will be reported, but the Run number will always be set to 0. When in any other mode, the ID and Run numbers will be set to 0.

Below is the formula for calculating the checksums that the Mini-NS reports with each packet:



*Figure 1: Checksum Calculation*

To calculate the RMD checksums (2 8-bit checksums), we iterate over the secondary CCSDS header and the payload data bytes and calculate a simple and Fletcher checksum. To find the BCT checksum (1 16-bit checksum), we add the byte values from byte 5 through the RMD checksums together.

*Table 1: RMD General Statement of Health Format*

|  |  |  |
| --- | --- | --- |
| Byte | Description | Group |
| 0-3 | Sync Marker | Primary CCSDS Header |
| 4 | Flags and Detector Number: |
| 5 | APID |
| 6 | Group Flags and Sequence Count MSB |
| 7 | Sequence Count LSB |
| 8 | Packet Length MSB |
| 9 | Packet Length LSB |
| 10 | Reset Request Flag | Secondary CCSDS Header |
| 11-14 | Analog Board Temp. | Data |
| 15 | “\t” |
| 16-19 | Digital Board Temp. |
| 20 | “\t” |
| 21-24 | Module Temp. Sensor |
| 25 | “\t” |
| 26-29 | Agg. Neutrons w/PSD |
| 30 | “\t” |
| 31-34 | Local Time |
| 35 | “\t” |
| 36 | Mode Byte |
| 37 | “\t” |
| 38-41 | ID Number |
| 42 | “\t” |
| 43-46 | Run Number |
| 47 | “\n” |
| 48 | Simple Checksum | RMD Data Checksums |
| 49 | Fletcher Checksum |
| 50 | CCSDS Checksum MSB | Checksum |
| 51 | CCSDS Checksum LSB |

*Table 2: Values for an SOH Packet*

|  |  |  |
| --- | --- | --- |
| Byte | Description | Group |
| 0 | 0x35 | Sync Marker |
| 1 | 0x2E |
| 2 | 0xF8 |
| 3 | 0x53 |
| 4 | 0x0A | Primary CCSDS Header |
| 5 | 0x22 |
| 6 | 0xC0 |
| 7 | 0x01 |
| 8 | 0x00 |
| 9 | 0x29 |
| 10 | 0x00 | Secondary CCSDS Header |
| 11-14 | Byte 11 = 0x00  Byte 12 = 0x00  Byte 13 = 0x00  Byte 14 = 0x19 | Mini-NS Data Bytes |
| 15 | 0x09 |
| 16-19 | Byte 16 = 0x00  Byte 17 = 0x00  Byte 18 = 0x00  Byte 19 = 0x01 |
| 20 | 0x09 |
| 21-24 | Byte 21 = 0x00  Byte 22 = 0x00  Byte 23 = 0x00  Byte 24 = 0x19 |
| 25 | 0x09 |
| 26-29 | Byte 26 = 0x00  Byte 27 = 0x00  Byte 28 = 0x00  Byte 29 = 0x32 |
| 30 | 0x09 |
| 31-34 | Byte 31 = 0x00  Byte 32 = 0x00  Byte 33 = 0x00  Byte 34 = 0x02 |
| 35 | 0x09 |
| 36 | 0x11 |
| 37 | 0x0A |
| 38-41 | Byte 38 = 0x00  Byte 39 = 0x00  Byte 40 = 0x00  Byte 41 = 0x00 |  |
| 42 | 0x0A |  |
| 43-46 | Byte 43 = 0x00  Byte 44 = 0x00  Byte 45 = 0x00  Byte 46 = 0x00 |  |
| 47 | 0x0A |  |
| 48 | CS1 | RMD Simple Checksum |
| 49 | CS2 | RMD Fletcher Checksum |
| 50 | CS3 | CCSDS Checksum MSB |
| 51 | CS4 | CCSDS Checksum LSB |

## Initialize Data Acquisition Run

Description: Initialize the system and prepare to take science data. The active components are energized, data objects are created in the code, and parameters for operation are loaded (e.g. HV is set). Data files are generated and opened. The system will be in a new polling loop.

Syntax: MNS\_DAQ\_#\_IDNumber

Example: MNS\_DAQ\_0\_000001\n

Command Length: 17 bytes

Field Format:

|  |  |  |  |
| --- | --- | --- | --- |
| Format | Description | Data Type | Value Range |
| MNS\_DAQ | Naming Sequence | ASCII string | - |
| # | Detector Number | Integer | 0, 1 |
| ID Number | Unique Run Number | Integer | 0 - 999999 |

Byte Conversion: N/A

Loop Status: Yes

Latency: N/A

Sub-commands: MNS\_START, MNS\_END, MNS\_BREAK, MNS\_READTEMP

Return Type: Command Success (0x00), Command Failure (0x11)

## LS Files Command

Description: Request system file list packets which list all the files on either SD card. As long as the SD cards have not reported any errors, then the default is to access SD 0.

Syntax: MNS\_LS\_FILES\_#\_SDNumber

Example: MNS\_LS\_FILES\_0\_1\n

Command Length: 17 bytes

Field Format:

|  |  |  |  |
| --- | --- | --- | --- |
| Format | Description | Data Type | Value Range |
| MNS\_LS\_FILES | Naming Sequence | ASCII string | - |
| # | Detector Number | Integer | 0, 1 |
| SD Number | SD Card ID Number | Integer | 0, 1 |

Byte Conversion: N/A

Loop Status: Yes

Latency: N/A

Sub-commands: MNS\_BREAK

Return Type: LS Files Return (0x33), Command Failure (0x11)

Byte Breakdown: Packet(s) returned by this function will have the following format:

|  |  |  |
| --- | --- | --- |
| Byte | Description | Group |
| 0-3 | 0x35 2E F8 53 | Sync Marker |
| 4 | … | Primary CCSDS Header |
| 5 | APID LSB = 0x33 |
| 8 | Packet Length MSB = |
| 9 | Packet Length LSB = |
| 10 | Reset Request Flag | Secondary CCSDS Header |
| 11-12 | Total Folders on SD Card | Mini-NS Data Bytes |
| 13-14 | Total Files on SD Card (including Root Directory) |  |
| 15-16 | SD Card ID Number (0 or 1) |  |
| 17-27 | Folder Name 1 (first folder item in directory) |  |
| 28 | ASCII Tab Character (0x09) |  |
| 29-36 | File Name 1-1 (first item in directory) |  |
| 37 | ASCII Space Character (0x20) |  |
| 38-41 | File Size 1-1 |  |
| 42 | ASCII Tab Character (0x09) |  |
| 43-50 | File Name 1-2 (second item in directory) |  |
| 51 | ASCII Space Character (0x20) |  |
| 52-55 | File Size 1-2 |  |
| 56 | ASCII Tab Character (0x09) |  |
| … | … |  |
| 2036 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| 88 |  |  |
| 89-97 |  |  |

# Input Command Packet Structure

## General Structure

Description: This outlines the input telecommand packets (command packets) which the board will accept. In addition to command packets, the board will accept plain ASCII strings which are terminated with a newline (‘\n’) character.

Field Format:

|  |  |  |
| --- | --- | --- |
| Byte | Description | Group |
| 0 | 0x35 | Sync Marker |
| 1 | 0x2E |
| 2 | 0xF8 |
| 3 | 0x53 |
| 4 | 7:5 - CCSDS Version Number = 000  4 - Packet Type = 1  3 - Secondary Header Flag = 0  2:0 – APID MSB = 010 for Mini-NS Detector 0  = 011 for Mini-NS Detector 1  0x12 = Detector 0  0x13 = Detector 1 | Primary CCSDS Header |
| 5 | APID LSB = TBD |
| 6 | 7:6 – Sequence Flags = 00 for intermediate packet  = 01 for first packet  = 10 for last packet  = 11 for unsegmented packet  5:0 – Sequence Count MSB |
| 7 | Sequence Count LSB |
| 8 | Packet Length MSB |
| 9 | Packet Length LSB |
| 10 | Telecommand ASCII Raw Bytes | Mini-NS Data Bytes |
| … | … |
| N-2 | Newline ASCII character “\n” |
| N-1 | Simple Checksum | RMD Data Checksums |
| N | Fletcher Checksum |

Return Type: When an ASCII Raw Bytes command is received by the board, the parameters and syntax are validated. If the command is accepted, then it is passed to the system for execution, otherwise it is rejected with a Command Failure packet. When a telecommand packet is received by the board, the board calculates checksums for it and the CCSDS header is validated. If the checksums match what is in bytes “N-1” and “N” and the header is accepted, then the command is parsed out of the Data Bytes field. The parameters and syntax are then validated and must also be accepted by the board. Otherwise the packet is rejected with a Command Failure packet.

* Exception: if an ASCII Raw Bytes command is rejected because it has an APID MSB for the other detector, then the command is rejected silently.

Byte Breakdown: Command packets have a Packet Type of 1 to indicate telecommand.

The APIDs for telecommand packets have not been determined as of writing.

Telecommand packets do not have a secondary CCSDS header, so the secondary header flag is 0.

Instead of four checksums, there will only be the RMD simple and Fletcher checksums at the end of each telecommand packet. These checksums are calculated the same as for SOH packets, seen above in Section 3.1.

The table below gives a breakdown of the available ASCII Raw Bytes commands which the Mini-NS will accept.

Example Commands:

|  |  |  |
| --- | --- | --- |
| Format | Description | Size Range (bytes) |
| MNS\_DAQ\_#\_ID | Data Acquistion |  |
| MNS\_WF\_#\_WFType\_NumWaves | Capture Waveforms |  |
| MNS\_READTEMP\_# | Send temperature Packet |  |
| MNS\_GETSTAT\_# | Send SOH Packet |  |
| MNS\_DISABLE\_ACT\_# | Disable Active Components |  |
| MNS\_ENABLE\_ACT\_# | Enable Active Components |  |
| MNS\_BREAK\_# | Break from a Process |  |
| MNS\_TX\_#\_... | Transfer a File |  |
| MNS\_DEL\_#\_FileName | Delete a File |  |
| MNS\_LS\_FILES\_# | Transfer File List |  |
| MNS\_TXLOG\_# | Transfer Log File |  |
| MNS\_CONF\_# | Transfer Configuration File |  |
| MNS\_TRG\_#\_Threshold | Change Trigger Threshold |  |
| MNS\_ECAL\_#\_Slope\_Intercept | Set Energy Calibration Params |  |
| MNS\_NGATES\_#\_... | Set Neutron Cut Gates |  |
| MNS\_HV\_#\_PMTID\_Taps | Set High Voltage Pot Values |  |
| MNS\_INT\_#\_Base\_Short\_Long\_Full | Set Integration Times |  |
| MNS\_START\_#\_RealTime\_MIN | Start Data Acquisition |  |
| MNS\_END\_#\_RealTime | End Data Acquisition |  |

Each command in the table is listed with a Size Range which indicates the minimum and maximum length, in bytes, that the command will be when placed into a command packet.

Each command has at least one parameter associated with it, that being the # which stands for detector number. The Mini-NS has two detectors numbered 0 and 1.

Commands with multiple parameters either have a descriptor of the field or a “…” which indicates there are many parameters. These parameters are described in full detail in the Mini-NS ICD.

The accepted command parameter ranges are listed in the Mini-NS ICD.

# Data Product Packet Type Dictionary

## Event-by-event Data Product (EVT)

Description: The full context of a data acquisition run.

How to Acquire: MNS\_TX command

Packet Length: 2037 bytes

Field Format:

|  |  |  |  |
| --- | --- | --- | --- |
| Format | Description | Data Type | Value Range |
| MNS\_TX | Naming Sequence | ASCII string | - |
| # | Detector Number | Integer | 0, 1 |
| Type | Data Product Type | Integer | 0-8 |
| IDNum | DAQ ID Number | Integer | 0-9999 |
| RunNum | DAQ Run Number | Integer | 0-9999 |
| SetNum | DAQ Set Number | Integer | 0-75 (estimated) |

Byte Conversion: N/A

Return Type: 0x77

Byte Breakdown: This data product is characterized by having a CCSDS header, a secondary header, a data header, data bytes, and checksums.

|  |  |  |  |
| --- | --- | --- | --- |
| Section | Description | Length (bytes) | Grouping |
| CCSDS Header | Header for acceptance by XB-1 | 10 | CCSDS Header |
| Secondary CCSDS Header | Reset Request Bytes | 1 | Data Bytes |
| RMD Data Header | System configuration parameters | 70 | Data Bytes |
| Data Bytes | Data product events | 1952 | Data Bytes |
| RMD Checksums | Simple, Fletcher | 2 | Data Bytes |
| Checksums | Checksums for XB-1 | 2 | Checksums |

# Appendix

## Byte-by-byte Breakdown of Mini-NS CCSDS Packet

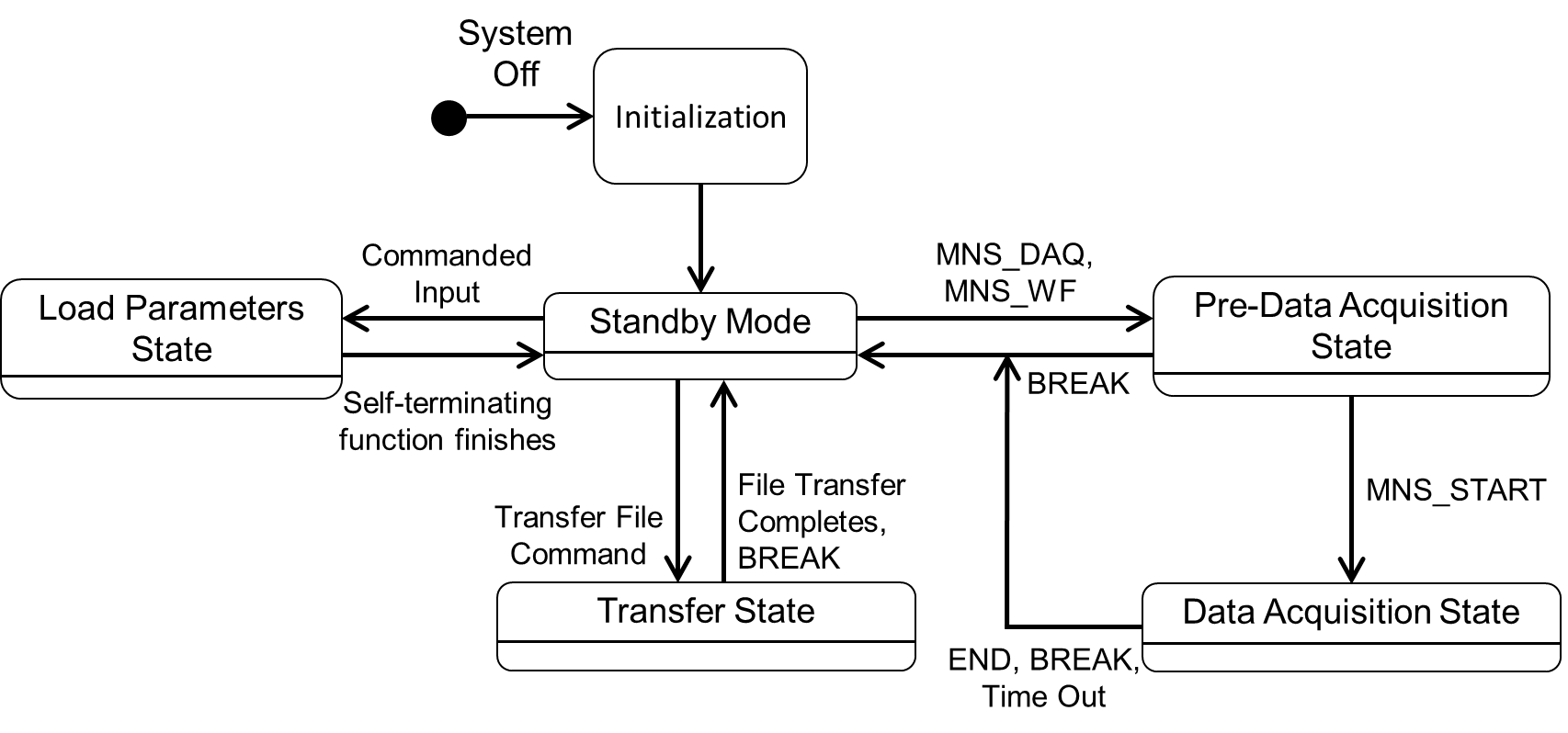
|  |  |  |
| --- | --- | --- |
| Byte | Description | Group |
| 0 | 0x35 | Sync Marker |
| 1 | 0x2E |
| 2 | 0xF8 |
| 3 | 0x53 |
| 4 | 7:5 - CCSDS Version Number = 000  4 - Packet Type (P.T.) = 0  3 - Secondary Header Flag (S.H.) = 1  2:0 – APID MSB = 010 for Mini-NS Detector 0  = 011 for Mini-NS Detector 1  0x0A = Detector 0  0x0B = Detector 1 | Primary CCSDS Header |
| 5 | APID LSB = 0x00 – 0xFF |
| 6 | 7:6 – Group Flags = 00 for intermediate packet  = 01 for first packet  = 10 for last packet  = 11 for unsegmented packet  5:0 – Sequence Count MSB |
| 7 | Sequence Count LSB |
| 8 | Packet Length MSB |
| 9 | Packet Length LSB |
| 10 | Reset Request = 0x00 – No request for reset  = 0x0F – Power off the detector  = 0xFF – Power cycle the detector to STDBY  = 0xF0 – Power cycle the detector to DAQ | Secondary CCSDS Header |
| 11 | Payload Data Byte 0 | Mini-NS Data Bytes |
| … | Payload Data Byte 2026 (max N = 2026) |
| N-1 | Simple Checksum | RMD Data Checksums |
| N | Fletcher Checksum |
| N+1 | Checksum MSB | Checksums |
| N+2 | Checksum LSB |

## APID Table for Mini-NS CCSDS Packets

|  |  |  |
| --- | --- | --- |
| APID LSB | Value | Description |
| Command Success | 0x00 | The input command was successfully executed |
| Command Failure | 0x11 | The input command failed somewhere during execution |
| Statement of Health | 0x22 | Packet contains statement of health information |
| LS Files Return | 0x33 | Packet contains filenames and file sizes for all the file on the Mini-NS SD card |
| Temperature | 0x44 | Packet with the most current temperatures. |
| MNS\_CPS | 0x55 | Neutron Counts per Second Data |
| MNS\_WAV | 0x66 | Waveform Data |
| MNS\_EVTS | 0x77 | Event by Event Data |
| MNS\_2DH | 0x88 | 2D Histogram Data |
| Log File | 0x99 | Log File |
| Configuration File | 0xAA | Configuration File |

## Mini-NS States and Modes

The Mode Byte in payload SOH packets informs the flight computer what mode (also referred to as states) the detector is currently in. There are 6 modes which the detector can be in at some point within a commanded run.



*Figure 2: Mini-NS State Diagram*

By sending valid command packets to the board, the flight computer can change the mode of the detector. See the following table for a list of the modes.

*Table 3: Mini-NS Mode Descriptions*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mode | Mode Byte | Description | Self-Terminating | Commands |
| System Off | - | No power to the detector | No | None |
| Initialization | - | Power applied, board is booting from selected EEPROM and performing startup operations | Yes, when finished board is in Standby Mode | None |
| Standby Mode | 0x11 | Detector waiting for input | No | All, **except** Start,  End |
| Load Parameters State | 0x22 | Validating and updating system run parameters | Yes, terminates to Standby Mode | None |
| Transfer State | 0x33 | Transfer a file | Yes, terminates to Standby Mode | Break |
| Pre-Data Acquisition | 0x44 | Have received the MNS\_DAQ command; processing the ID number and creating the folder & files for the run; waiting for MNS\_START command | No | ReadTemp,  Break,  Start |
| Data Acquisition | 0x55 | Collect neutron data | Yes, terminates to Standby Mode | ReadTemp  Start,  End,  Break |
| Waveform Collection | 0x66 | Collect Waveform data | Yes, terminates to Standby mode | None |

Each mode is listed with a byte value (or – indicating none) which will be reported in an SOH packet while the detector is in that mode or state.

The commands column lists which commands each mode will accept.

The Load Parameters state is a short-lived state and will not report SOH

# SD Card File Format

## Files on the SD Card

There are both data product files and system files which can be found on the SD card of the Mini-NS. Data product files are created in folders while the system files are created in the root directory of the SD card. The files are:

* CPS data product files
* WF data product files
* EVT data product files
* 2DH data product files
* System log file
* System configuration file
* System differentiation file \*
* System transfer log file\*

The system differentiation file and the system transfer log file are both not found on the SD card for this release. They will potentially be used in a future update.

The format of these files is different than what will be transferred using the MNS\_TX command. This is due to header and footer bytes which store other information about the data product (such as ID, Run numbers, etc) and which can be found in the data header for the transferred data products.

The EVT data product file has an additional header which allows it to be written faster during science operation. These bytes are spacing bytes and are blank. This allows the system to write on the cluster edges when writing this data product since it is much larger than the others and will be written more often.

## System File Format Breakdown

Here the format for the system files above are shown and discussed.

### System Log File

The Log File records every commanded transaction with the input command and the input parameters. The data is written as ASCII characters to a text file which is stored on the root directory of the SD card. This file will grow in size over the lifetime of the experiment.

The Log file records all commanded input which is either accepted or rejected by the system. The full context of the commanded input is recorded, including the detector number and all parameters specified.

**Commands which are ignored by a Detector are not recorded in the command log.**

If the file grows too large or needs to be reset, it can be deleted using the MNS\_DEL command. The system will detect that it has been deleted upon receiving the next properly formed command or upon power cycle and will generate a new log file.

If during power up the system detects that the log file doesn’t exist, it will create a new log file. It will write in “FIRST POWER ON” as the only contents and save the file.

Each time the system is power cycled the system will write in to the log file “POWER RESET” to indicate that the power was turned off then on again.

*Table 4: Mini-NS System Log File Format*

|  |  |  |
| --- | --- | --- |
| Byte | Description | Type |
|  | **FIRST POWER ON** | ASCII |
|  | **“\n”** | ASCII |
|  | Command 0 | ASCII |
|  | “\n” | ASCII |
|  | Command 1 | ASCII |
|  | ”\n” | ASCII |

The possible entries written into the log file are “First power on”, “Power reset”, and every Mini-NS command along with their parameters.

The exact format for this file to be transferred is yet to be determined. The following table shows what it might look like.

*Table 5: Mini-NS System Log Transfer Format*

|  |  |  |
| --- | --- | --- |
| Byte | Description | Group |
| 0-8 | Primary CCSDS Header  **Byte 5 = 0x99** | Primary CCSDS Header |
| 9 | **Packet Length = N+3** |
| 10 | Reset Request Flag | Secondary CCSDS Header |
| 11… | Command 0 | Data |
|  | “\n” |
|  | Command 1 |
| …N-2 | …”\n” |
| N-1 | Simple Checksum | Mini-NS Data Checksums |
| N | Fletcher Checksum |
| N+1 | CCSDS Checksum MSB | Checksum |
| N+2 | CCSDS Checksum LSB |

The second file is the Mini-NS configuration file and is a binary file which contains the current system configuration parameters. The full list of system parameters which are configurable is found in TABLE\_XX. Any time that commanded input changes a system parameter, the change is recorded and that value is stored in the configuration file. As a back up measure, the Mini-NS has a set of hard-coded default system parameters that are used if the configuration file on either SD card cannot be accessed.

### System Configuration File

The configuration file holds a list of parameters which are the most current system configuration parameters. The following table displays the format of that file

Table : Mini-NS System Configuration File Format

|  |  |  |
| --- | --- | --- |
| Byte | Field Type | Description |
| 0-3 | Float | Energy Calibration Curve Slope |
| 4-7 | Float | Energy Calibration Curve Intercept |
| 8-11 | Int | Trigger threshold |
| 12-15 | Int | Baseline integration time |
| 16-19 | Int | Short Integration time |
| 20-23 | Int | Long Integration time |
| 24-27 | Int | Full Integration Time |
| 28-31 | Int | High voltage tap value for PMT 0 |
| 32-35 | Int | High voltage tap value for PMT 1 |
| 36-39 | Int | High voltage tap value for PMT 2 |
| 40-43 | Int | High voltage tap value for PMT 3 |
| 44-47 | Double | X scale factor for Ellipse 1, PMT 0 |
| 48-51 | Double | X scale factor for Ellipse 2, PMT 0 |
| 52-55 | Double | X scale factor for Ellipse 1, PMT 1 |
| 56-59 | Double | X scale factor for Ellipse 2, PMT 1 |
| 60-63 | Double | X scale factor for Ellipse 1, PMT 2 |
| 64-67 | Double | X scale factor for Ellipse 2, PMT 2 |
| 68-71 | Double | X scale factor for Ellipse 1, PMT 3 |
| 72-75 | Double | X scale factor for Ellipse 2, PMT 3 |
| 76-79 | Double | Y scale factor for Ellipse 1, PMT 0 |
| 80-83 | Double | Y scale factor for Ellipse 2, PMT 0 |
| 84-87 | Double | Y scale factor for Ellipse 1, PMT 1 |
| 88-91 | Double | Y scale factor for Ellipse 2, PMT 1 |
| 92-95 | Double | Y scale factor for Ellipse 1, PMT 2 |
| 96-99 | Double | Y scale factor for Ellipse 2, PMT 2 |
| 100-103 | Double | Y scale factor for Ellipse 1, PMT 3 |
| 104-107 | Double | Y scale factor for Ellipse 2, PMT 3 |
| 108-111 | Double | X offset for Ellipse 1, PMT 0 |
| 112-115 | Double | X offset for Ellipse 2, PMT 0 |
| 116-119 | Double | X offset for Ellipse 1, PMT 1 |
| 120-123 | Double | X offset for Ellipse 2, PMT 1 |
| 124-127 | Double | X offset for Ellipse 1, PMT 2 |
| 128-131 | Double | X offset for Ellipse 2, PMT 2 |
| 132-135 | Double | X offset for Ellipse 1, PMT 3 |
| 136-139 | Double | X offset for Ellipse 2, PMT 3 |
| 140-143 | Double | Y offset for Ellipse 1, PMT 0 |
| 144-147 | Double | Y offset for Ellipse 2, PMT 0 |
| 148-151 | Double | Y offset for Ellipse 1, PMT 1 |
| 152-155 | Double | Y offset for Ellipse 2, PMT 1 |
| 156-159 | Double | Y offset for Ellipse 1, PMT 2 |
| 160-163 | Double | Y offset for Ellipse 2, PMT 2 |
| 164-167 | Double | Y offset for Ellipse 1, PMT 3 |
| 168-171 | Double | Y offset for Ellipse 2, PMT 3 |

The configuration file is a total of 172 bytes, all of which hold a system parameter. Every time a system parameter is updated via a valid MNS\_ command, the corresponding value in the configuration file is also updated. In this way, each time the Mini-NS is power cycled, it will be initialized with the most recent set of system parameters with no further input.

Every time MNS\_DAQ or MNS\_WF are commanded, this configuration is reapplied to the system.

Table 7 Configuration Transfer Format

| Byte | Field Type | Description | Group |
| --- | --- | --- | --- |
| 0-8 | Byte-by-byte | Primary CCSDS Header  **Byte 5 = 0xAA** | Primary CCSDS Header |
| 9 |  | **Packet Length = 60** |
| 10 | Unsigned Char | Reset Request Flag | Secondary CCSDS Header |
| 11-14 | Float | Energy Calibration Slope | Configuration Data |
| 15-18 | Float | Energy Calibration Intercept |
| 19-22 | Int | Trigger Threshold |
| 23-26 | Int | Baseline Integration Time |
| 27-30 | Int | Short Integration Time |
| 31-34 | Int | Long Integration Time |
| 35-38 | Int | Full Integration Time |
| 39-42 | Int | High Voltage of PMT 0 |
| 43-46 | Int | High Voltage of PMT 1 |
| 47-50 | Int | High Voltage of PMT 2 |
| 51-54 | Int | High Voltage of PMT 3 |
| 55-58 | Double | X scale factor for Ellipse 1, PMT 0 |
| 59-62 | Double | X scale factor for Ellipse 2, PMT 0 |
| 63-66 | Double | X scale factor for Ellipse 1, PMT 1 |
| 67-70 | Double | X scale factor for Ellipse 2, PMT 1 |
| 71-74 | Double | X scale factor for Ellipse 1, PMT 2 |
| 75-78 | Double | X scale factor for Ellipse 2, PMT 2 |
| 79-82 | Double | X scale factor for Ellipse 1, PMT 3 |
| 83-86 | Double | X scale factor for Ellipse 2, PMT 3 |
| 87-90 | Double | Y scale factor for Ellipse 1, PMT 0 |
| 91-94 | Double | Y scale factor for Ellipse 2, PMT 0 |
| 95-98 | Double | Y scale factor for Ellipse 1, PMT 1 |
| 99-102 | Double | Y scale factor for Ellipse 2, PMT 1 |
| 103-106 | Double | Y scale factor for Ellipse 1, PMT 2 |
| 107-110 | Double | Y scale factor for Ellipse 2, PMT 2 |
| 111-114 | Double | Y scale factor for Ellipse 1, PMT 3 |
| 115-118 | Double | Y scale factor for Ellipse 2, PMT 3 |
| 119-122 | Double | X offset for Ellipse 1, PMT 0 |
| 123-126 | Double | X offset for Ellipse 2, PMT 0 |
| 127-130 | Double | X offset for Ellipse 1, PMT 1 |
| 131-134 | Double | X offset for Ellipse 2, PMT 1 |
| 135-138 | Double | X offset for Ellipse 1, PMT 2 |
| 139-142 | Double | X offset for Ellipse 2, PMT 2 |
| 143-146 | Double | X offset for Ellipse 1, PMT 3 |
| 147-150 | Double | X offset for Ellipse 2, PMT 3 |
| 151-154 | Double | Y offset for Ellipse 1, PMT 0 |
| 155-158 | Double | Y offset for Ellipse 2, PMT 0 |
| 159-162 | Double | Y offset for Ellipse 1, PMT 1 |
| 163-166 | Double | Y offset for Ellipse 2, PMT 1 |
| 167-170 | Double | Y offset for Ellipse 1, PMT 2 |
| 171-174 | Double | Y offset for Ellipse 2, PMT 2 |
| 175-178 | Double | Y offset for Ellipse 1, PMT 3 |
| 179-182 | Double | Y offset for Ellipse 2, PMT 3 |
| 183 | Unsigned Char | Simple Checksum | Mini-NS Data Checksums |
| 184 | Unsigned Char | Fletcher Checksum |
| 185 | Unsigned Char | CCSDS Checksum MSB | Checksum |
| 186 | Unsigned Char | CCSDS Checksum LSB |

### CPS Data Product File Format