# Encoding

All multi-byte numeric fields are encoded in network byte order (big-endian). Note that this also applies to the floating point representations.

When implementing interface code on little-endian systems (such as x86, x86-64, ARMv7-A), care should be taken to byte swap all numerical values wider than 1 byte. This includes 16 bit and 32 bit signed and unsigned numbers, along with the IEEE 754 32 bit floating point numbers.

# Command Definitions

TODO: section description and brief command mapping table

## CMD\_GET\_PROTOCOL\_VERSION (0x00)

This command returns the current protocol version.

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** 2 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Major | | | | | | | |
| 1 | Minor | | | | Subminor | | | |

**Major:** 8 bits unsigned

**Minor:** 4 bits unsigned

**Subminor:** 4 bits unsigned

The version is encoded in USB’s binary coded decimal (BCD) version format. A response with Major=2, Minor=0, Subminor=1 represents a protocol version of “2.0.1”.

## CMD\_GET\_BOARD\_INFO (0x01)

This command returns the board’s name and revision number (e.g. “RF Power Transmitter” Revision 3).

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Board Revision Number | | | | | | | |
| 1 | Board Name | | | | | | | |
| … |

**Board Revision Number:** 8 bits unsigned

**Board Name:** variable length string (UTF-8)

## CMD\_GET\_USER\_TAG\_LOCATIONS (0x02)

This command returns the locations of the various tags in the NVM region of the device. The NVM commands can be used to read and write the tag values. The locations are guaranteed to not overlap.

Tag usage is left to the user; the device does not utilize the data in these locations in any way. They’re set aside as a convenient way to store persistent information on the device (rather than on the computer talking to the device). An example is to use the 1st tag for customer information, using the 2nd tag for the physical installation location, and using the general storage to store device defaults packed in binary.

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** 12 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Tag 1 Offset | | | | | | | |
| 1 |
| 2 | Tag 1 Length | | | | | | | |
| 3 |
| 4 | Tag 2 Offset | | | | | | | |
| 5 |
| 6 | Tag 2 Length | | | | | | | |
| 7 |
| 8 | General Storage Offset | | | | | | | |
| 9 |
| 10 | General Storage Size | | | | | | | |
| 11 |

**Tag 1 Offset:** 16 bits unsigned

**Tag 1 Length:** 16 bit unsigned

**Tag 2 Offset:** 16 bits unsigned

**Tag 2 Length:** 16 bit unsigned

**General Storage Offset:** 16 bits unsigned

**General Storage Size:** 16 bit unsigned

The offsets, lengths and sizes are all given in NVM words (32-bit). To get the lengths in bytes the relevant value should be multiplied by 4.

There is no technical difference between tags and general storage. The different names reflect the expected usage.

NOTE: because of hardware limitations, the lengths may be limited. Implementations will provide a minimum of 8 NVM words (32 bytes) for each location.

## CMD\_GET\_BUILD\_INFO (0x03)

This command returns the build information of the firmware. The build information can be useful for debugging.

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Build Info | | | | | | | |
| … |

**Build Info:** variable length string (UTF-8)

## CMD\_GET\_BUILD\_DATE (0x04)

This command returns the build date of the firmware. The build date can be useful for debugging, along with the build info (see CMD\_GET\_BUILD\_INFO).

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Build Date | | | | | | | |
| … |

**Build Date:** variable length string (UTF-8)

## CMD\_GET\_CHIP\_FAMILY (0x05)

This command returns the chip family of the device’s main processor. Some examples of chip families are: “XMega”, “EFM32” and “nRF24LE1”.

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Chip Family | | | | | | | |
| … |

**Chip Family:** variable length string (UTF-8)

## CMD\_GET\_CHIP\_MODEL (0x06)

This command returns the chip model number of the device’s main processor. Some examples are: “ATXMega128A4U” and “EFM32WG295F256”.

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Chip Model | | | | | | | |
| … |

**Chip Model:** variable length string (UTF-8)

## CMD\_GET\_CHIP\_ID (0x07)

This command returns the unique chip id of the device’s main processor. Note that this is distinct from the device’s serial number. This is primarily used during factory calibration, and is not needed for normal device interaction.

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Chip ID | | | | | | | |
| … |

**Chip ID:** variable length byte array

The chip ID is a sequence of bytes, rather than a string. To convert to string representation, uppercase hexadecimal formatting should be used. This will yield a string with two characters for every byte in the byte array.

## CMD\_GET\_NVM\_SIZE (0x08)

This command returns the size of the NVM region in the device.

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** 2 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | NVM Size | | | | | | | |
| 1 |

**NVM Size:** 16 bits unsigned

NVM Size is given in NVM words (32-bits). To get the size in bytes, the value should be multiplied by 4.

## CMD\_ERASE\_NVM (0x09)

This command erases the entire NVM region in the device. This must be done before writing updated NVM data.

Outgoing Parameters:

None

Incoming Parameters:

None

## CMD\_WRITE\_NVM (0x0A)

This command writes data to the NVM region in the device. Before writing updated data, the NVM region must be erased (see CMD\_ERASE\_NVM). Writes should not be performed at addresses that have been written since the last erase.

Outgoing Parameters:

**Total length:** variable (≥6 bytes)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | NVM Offset | | | | | | | |
| 1 |
| 2 | Data Word 0 | | | | | | | |
| 3 |
| 4 |
| 5 |
| … | Data Word 1 to n | | | | | | | |

**NVM Offset:** 16 bits unsigned

**Data Word:** 32 bits unsigned

Incoming Parameters:

None

The NVM Offset is given in NVM words (32-bits). This offset is the first address to be written with the new data. Data must be provided in multiples of 32-bits.

## CMD\_READ\_NVM (0x0B)

This command reads data from the NVM region of the device.

Outgoing Parameters:

**Total length:** 2 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | NVM Offset | | | | | | | |
| 1 |

**NVM Offset:** 16 bits unsigned

Incoming Parameters:

**Total length:** variable (≥4 bytes)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Data Word 0 | | | | | | | |
| 1 |
| 2 |
| 3 |
| … | Data Word 1 to n | | | | | | | |

**Data Word:** 32 bits unsigned

The NVM Offset is given in NVM words (32-bits). This offset is the first address to be read. Data will be provided in multiples of 32-bits. As many data words will be provided as are available, while taking into account the constraints imposed by the maximum packet size.

## CMD\_FLUSH (0x0C)

This command causes the device to perform a “soft” reset, flushing any communication buffers, and disabling any enabled streams. It may also, at the device’s discretion perform other operations such as resetting LEDs, disabling RF Power, and resetting control variables.

Outgoing Parameters:

None

Incoming Parameters:

None

## CMD\_RESET (0x0D)

This command causes the device to perform a complete reset, as soon as it receives the command. No response will be sent. The exact manner of the subsequent disconnect depends on the encapsulating transport method.

Outgoing Parameters:

None

Incoming Parameters:

None

## CMD\_GET\_BOOTLOADER\_INFO (0x0E)

This command returns an informational string describing the nature of the device’s bootloader. An empty reply means that no bootloader is present on the device, and that consequently CMD\_BOOTLOADER\_JUMP is non-functional.

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Bootloader Type | | | | | | | |
| … |

**Bootloader Type:** variable length string (UTF-8)

The Bootloader Type string identifies the bootloader. Bootloader types and their usages is outside the scope of this document.

## CMD\_BOOTLOADER\_JUMP (0x0F)

This command causes the device to immediately start its bootloader. No response will be sent. The exact manner of the subsequent disconnect depends on the encapsulating transport method. Connecting to the bootloader is device specific, and outside the scope of this document. In devices that do not have bootloaders present (and return empty strings for CMD\_GET\_BOOTLOADER\_INFO), for simplicity of implementation, this command will perform the same operation as CMD\_RESET.

Outgoing Parameters:

None

Incoming Parameters:

None

## CMD\_GET\_RGB\_COUNT (0x10)

This command returns the number of RGB LEDs present on the device. This number may be zero.

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Count | | | | | | | |

**Count:** 8 bits unsigned

## CMD\_GET\_RGB\_VALUES (0x11)

This command returns the current setting of a specific RGB LED on the device.

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters:

**Total length:** 3 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Red | | | | | | | |
| 1 | Green | | | | | | | |
| 2 | Blue | | | | | | | |

**Red:** 8 bits unsigned

**Green:** 8 bits unsigned

**Blue:** 8 bits unsigned

The index should be less than the number of RGB LEDs present on the device (see CMD\_GET\_RGB\_COUNT).

## CMD\_SET\_RGB (0x12)

This command sets the target color for one of the RGB LEDs present on the device. The device will smooth the color transition. See CMD\_SET\_RGB\_INSTANT for changing the color without any transition.

Outgoing Parameters:

**Total length:** 4 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |
| 1 | Red | | | | | | | |
| 2 | Green | | | | | | | |
| 3 | Blue | | | | | | | |

**Index:** 8 bits unsigned

**Red:** 8 bits unsigned

**Green:** 8 bits unsigned

**Blue:** 8 bits unsigned

Incoming Parameters:

None

The index should be less than the number of RGB LEDs present on the device (see CMD\_GET\_RGB\_COUNT).

## CMD\_SET\_RGB\_INSTANT (0x13)

This command sets the immediate color for one of the RGB LEDs present on the device. The device will not attempt to smooth the color transition. See CMD\_SET\_RGB for changing the color with a smooth transition.

Outgoing Parameters:

**Total length:** 4 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |
| 1 | Red | | | | | | | |
| 2 | Green | | | | | | | |
| 3 | Blue | | | | | | | |

**Index:** 8 bits unsigned

**Red:** 8 bits unsigned

**Green:** 8 bits unsigned

**Blue:** 8 bits unsigned

Incoming Parameters:

None

The index should be less than the number of RGB LEDs present on the device (see CMD\_GET\_RGB\_COUNT).

## CMD\_GET\_LED\_COUNT (0x14)

This command returns the number of individually addressable LEDs present on the device. This number may be zero.

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Count | | | | | | | |

**Count:** 8 bits unsigned

## CMD\_GET\_LED\_VALUE (0x15)

This command returns the current setting of one of the individually addressable LEDs on the device.

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Value | | | | | | | |

**Value:** 8 bits unsigned

The index should be less than the number of LEDs present on the device (see CMD\_GET\_LED\_COUNT).

## CMD\_SET\_LED (0x16)

This command sets the target value for one of the individually addressable LEDs present on the device. The device will smooth the transition. See CMD\_SET\_LED\_INSTANT for changing the value without any transition.

Outgoing Parameters:

**Total length:** 2 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |
| 1 | Value | | | | | | | |

**Index:** 8 bits unsigned

**Value:** 8 bits unsigned

Incoming Parameters:

None

The index should be less than the number of LEDs present on the device (see CMD\_GET\_LED\_COUNT). A value of 0 represents fully off, and a value of 255 represents fully on.

## CMD\_SET\_LED\_INSTANT (0x17)

This command sets the immediate value for one of the individually addressable LEDs present on the device. The device will not attempt to smooth the transition. See CMD\_SET\_LED for changing the value with a smooth transition.

Outgoing Parameters:

**Total length:** 2 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |
| 1 | Value | | | | | | | |

**Index:** 8 bits unsigned

**Value:** 8 bits unsigned

Incoming Parameters:

None

The index should be less than the number of LEDs present on the device (see CMD\_GET\_LED\_COUNT). A value of 0 represents fully off, and a value of 255 represents fully on.

## CMD\_GET\_STREAM\_COUNT\_AND\_ID (0x20)

This command returns the number of streams present on the device along with the number of bits in the stream ID at the beginning of each stream packet. The number of streams on a device may be zero.

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** 3 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Count | | | | | | | |
| 1 | Leading Filler Bits | | | | | | | |
| 2 | ID Bits | | | | | | | |

**Count:** 8 bits unsigned

**Leading Filler Bits:** 8 bits unsigned

**ID Bits:** 8 bits unsigned

If the number of ID bits is zero, then each stream packet belongs to Stream 0.

## CMD\_GET\_STREAM\_CHANNELS (0x21)

This command returns a list of indexes of channels contained on the specified stream. These channel indexes are guaranteed to not be contained on any other stream.

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Channel Index 0 | | | | | | | |
| … | Channel Index n | | | | | | | |

**Channel Index:** 8 bits unsigned

The index should be less than the number of streams present on the device (see CMD\_GET\_STREAM\_COUNT).

## CMD\_GET\_STREAM\_FORMAT (0x22)

This command returns format and rate information about the specified stream.

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters:

**Total length:** 14 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Leading Filler Bits | | | | | | | |
| 1 | Counter Bits | | | | | | | |
| 2 | Rate | | | | | | | |
| 3 |
| 4 |
| 5 |
| 6 | Rate Error | | | | | | | |
| 7 |
| 8 |
| 9 |
| 10 | Warm Up Delay | | | | | | | |
| 11 |
| 12 |
| 13 |

**Leading Filler Bits:** 8 bits unsigned

**Counter Bits:** 8 bits unsigned

**Rate:** 32 bits floating point (IEEE 754 binary32)

**Rate Error:** 32 bits floating point (IEEE 754 binary32)

**Warm Up Delay:** 32 bits floating point (IEEE 754 binary32)

The index should be less than the number of streams present on the device (see CMD\_GET\_STREAM\_COUNT).

The Counter Bits parameter is the number of bits dedicated to the counter at the beginning of the stream packet. Bits following the counter are channel data.

Rate is the number of stream packets per second (i.e. in Hz). Rate error is the theoretical maximum deviation from the rate. For example, a 5% maximum deviation would be encoded as 0.05.

## CMD\_ENABLE\_STREAM (0x23)

This command enables or disables a specific stream.

Outgoing Parameters:

**Total length:** 2 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |
| 1 | Enable | | | | | | | |

**Index:** 8 bits unsigned

**Enable:** Boolean (0/1)

Incoming Parameters:

None

The index should be less than the number of streams present on the device (see CMD\_GET\_STREAM\_COUNT). If the Enable parameter is 0, the stream will be disabled. Otherwise the stream will be enabled.

## CMD\_WARM\_UP\_STREAM (0x24)

This command enables or disables the warm up functionality of a specific stream. A warmed up stream may provide data more rapidly when streaming is enabled.

Outgoing Parameters:

**Total length:** 2 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |
| 1 | Warm Up | | | | | | | |

**Index:** 8 bits unsigned

**Warm Up:** Boolean (0/1)

Incoming Parameters:

None

The index should be less than the number of streams present on the device (see CMD\_GET\_STREAM\_COUNT). If the Warm Up parameter is 0, the stream will have its warm up disabled. Otherwise the stream will have its warm up enabled.

## CMD\_GET\_STREAM\_STATUS (0x25)

This command returns the enable and warm up status of a specific stream.

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters:

**Total length:** 2 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Enabled | | | | | | | |
| 1 | Warm Up | | | | | | | |

**Enabled:** Boolean (0/1)

**Warm Up:** Boolean (0/1)

The index should be less than the number of streams present on the device (see CMD\_GET\_STREAM\_COUNT).

## CMD\_GET\_CHANNEL\_COUNT (0x30)

This command returns the number of streaming channels present on the device. This number may be zero.

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Count | | | | | | | |

**Count:** 8 bits unsigned

## CMD\_GET\_CHANNEL\_NAME (0x31)

This command returns the name of a specific streaming channel on the device.

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Channel Name | | | | | | | |
| … |

**Channel Name:** variable length string (UTF-8)

The index should be less than the number of channels present on the device (see CMD\_GET\_CHANNEL\_COUNT).

## CMD\_GET\_CHANNEL\_INFO (0x32)

This command returns channel type and decoding information about the specified channel.

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters:

**Total length:** 22 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Channel Type | | | | | | | |
| 1 | Unit Type | | | | | | | |
| 2 | Leading Filler Bits | | | | | | | |
| 3 |
| 4 | Data Bits | | | | | | | |
| 5 |
| 6 | Samples | | | | | | | |
| 7 | Bits Per Sample | | | | | | | |
| 8 |
| 9 | Minimum | | | | | | | |
| 10 |
| 11 |
| 12 |
| 13 | Maximum | | | | | | | |
| 14 |
| 15 |
| 16 |
| 17 | Resolution | | | | | | | |
| 18 |
| 19 |
| 20 |
| 21 | Chunk Count | | | | | | | |

**Channel Type:** 8 bits unsigned

**Unit Type:** 8 bits unsigned

**Leading Filler Bits:** 16 bits unsigned

**Data Bits:** 16 bits unsigned

**Samples:** 8 bits unsigned

**Bits Per Sample:** 16 bits signed

**Minimum:** 32 bit floating point (IEEE 754 binary32)

**Maximum:** 32 bit floating point (IEEE 754 binary32)

**Resolution:** 32 bit floating point (IEEE 754 binary32)

**Chunk Count:** 8 bits unsigned

The index should be less than the number of channels present on the device (see CMD\_GET\_CHANNEL\_COUNT).

The Channel Type parameter specifies the type of the channel, and dictates how the channel extra information (see CMD\_GET\_CHANNEL\_EXTRA\_INFO) and channel coefficients (see CMD\_GET\_CHANNEL\_COEFFICIENTS) are interpreted. Likewise, it dictates what channel specific transactions (see CMD\_CHANNEL\_SPECIFIC) are available, if any. See the relevant channel type documentation for more information.

Unit Type corresponds to an entry in the unit table definition, and defines the units of the output data.

The Leading Filler Bits parameter determines how many bits to skip over before reading valid data.

Data Bits determines how many bits to decode in the stream packet.

The Samples parameter is used to determine how many sample points of the channel are packed into the data bits on each stream packet.

The Bits Per Sample parameter is channel type dependent, but the general convention is that positive numbers represent unsigned values, and negative numbers represent signed values. For example a value of 24 would mean that each sample is encoded as a 24-bit unsigned value. Likewise, a value of -18 would mean that each sample is encoded as an 18-bit signed value.

The Minimum, Maximum and Resolution parameters are useful generalities that can be utilized to auto scale graphs and determine the number of digits after a decimal point to display to a user. The minimum and maximum are not hard limits, but they instead define the expected operational range. The resolution specifies the difference between measurements that can be reasonably distinguished from one another given the limitations of the sensor and measurement system. If the resolution is 0, then it is unknown or unspecified and the parameter should be ignored.

The Chunk Count parameter defines the number of chunks present on this channel. See CMD\_GET\_CHANNEL\_CHUNK for details for how to query chunks.

## CMD\_GET\_CHANNEL\_COEFFICIENTS (0x33)

This command returns the channel coefficients for a specific channel. The channel coefficients are channel-type specific and their usage is defined the relevant channel type documentation. Note: the number of coefficients may be zero.

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Channel Coefficient 0 | | | | | | | |
| 1 |
| 2 |
| 3 |
| … | Channel Coefficient n | | | | | | | |

**Channel Coefficient n:** 32 bit floating point (IEEE 754 binary32)

The index should be less than the number of channels present on the device (see CMD\_GET\_CHANNEL\_COUNT).

## CMD\_GET\_CHANNEL \_CHUNK (0x34)

This command returns the additional channel formatting information for a specific channel. The channel information chunks are channel-type specific and their usage is defined the relevant channel type documentation.

Outgoing Parameters:

**Total length:** 2 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |
| 1 | Chunk Number | | | | | | | |

**Index:** 8 bits unsigned

**Chunk Number:** 8 bits unsigned

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Chunk | | | | | | | |
| … |

**Chunk:** variable length byte array

The index should be less than the number of channels present on the device (see CMD\_GET\_CHANNEL\_COUNT).

The chunk number should be less than the number of chunks present on the channel (see CMD\_GET\_CHANNEL\_INFO).

## CMD\_CHANNEL\_SPECIFIC (0x35)

This command performs a channel-type specific action or inquiry on the specified channel. The format of the outgoing and incoming channel specific parameters is defined in the relevant channel type documentation.

Outgoing Parameters:

**Total length:** variable (≥2 bytes)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |
| 1 | Channel Specific Outgoing | | | | | | | |
| … |

**Index:** 8 bits unsigned

**Channel Specific Outgoing:** variable length byte array

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Channel Specific Incoming | | | | | | | |
| … |

**Channel Specific Incoming:** variable length byte array

The index should be less than the number of channels present on the device (see CMD\_GET\_CHANNEL\_COUNT).

## CMD\_GET\_SUPPLY\_COUNT (0x40)

This command returns the number of measurable supplies present on the device. This number may be zero.

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Count | | | | | | | |

**Count:** 8 bits unsigned

## CMD\_GET\_SUPPLY\_NAME (0x41)

This command returns the name of a specific supply on the device.

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Supply Name | | | | | | | |
| … |

**Supply Name:** variable length string (UTF-8)

The index should be less than the number of supplies present on the device (see CMD\_GET\_SUPPLY\_COUNT).

## CMD\_GET\_SUPPLY\_INFO (0x42)

This command returns display and representational information about the specified supply. This information is only needed for displaying supply results for display to a user.

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters:

**Total length:** 14 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Unit Type | | | | | | | |
| 1 | Is Battery | | | | | | | |
| 2 | Nominal | | | | | | | |
| 3 |
| 4 |
| 5 |
| 6 | Scale | | | | | | | |
| 7 |
| 8 |
| 9 |
| 10 | Offset | | | | | | | |
| 11 |
| 12 |
| 13 |

**Unit Type:** 8 bits unsigned

**Is Battery:** Boolean (0/1)

**Nominal:** 32 bits signed

**Scale:** 32 bit floating point (IEEE 754 binary32)

**Offset:** 32 bit floating point (IEEE 754 binary32)

The index should be less than the number of supplies present on the device (see CMD\_GET\_SUPPLY\_COUNT).

Unit Type corresponds to an entry in the unit type table, and defines the output unit after applying the scale and offset.

The Is Battery parameter determines whether the supply represents a battery that the user can replace or recharge. If this bit is set, during a CMD\_CHECK\_SUPPLY the status may have the Low Battery bit set.

The Nominal parameter is the idealized value for the CMD\_CHECK\_SUPPLY’s measurement. The nominal value can be used to display the error percentage for a given measurement (after accounting for scale & offset).

The Scale and Offset parameters are used to change the measurement value (or nominal value) into the given unit type. The measurement is first multiplied by the scale, and then this product is added to the offset to get the final value.

## CMD\_CHECK\_SUPPLY (0x43)

This command starts a supply measurement, and returns the results (or returns status indicating the conversion is still in progress).

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters (Conversion Still In Progress):

None

Incoming Parameters (Conversion Finished):

**Total length:** 5 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Measurement | | | | | | | |
| 1 |
| 2 |
| 3 |
| 4 | Reserved | | | | | Too High | Too Low | Low Battery |

**Measurement:** 32 bits signed

**Too High:** 1-bit Boolean (0/1)

**Too Low:** 1 bit Boolean (0/1)

**Low Battery:** 1 bit Boolean (0/1)

The index should be less than the number of supplies present on the device (see CMD\_GET\_SUPPLY\_COUNT).

If the conversion not finished, no incoming parameters are returned. The host can resubmit the request until the conversion is finished and parameters are returned.

The Measurement parameter can be converted to a human readable value with the values from CMD\_GET\_SUPPLY\_INFO. One can also compute the error percentage from nominal.

The Too High and Too Low bits signal hardware error conditions. Streamed channel values cannot be relied upon when either of these bits is set.

The Low Battery bit signals that the supply is low enough that a human should soon replace or recharge the device’s batteries. The supply, however, is not low enough to cause measurement errors on streamed channels.

## CMD\_GET\_CTRL\_VAR\_COUNT (0x50)

This command returns the number of control variables present on the device. This number may be zero.

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Count | | | | | | | |

**Count:** 8 bits unsigned

## CMD\_GET\_CTRL\_VAR\_NAME (0x51)

This command returns the name of a specific control variable on the device.

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Name | | | | | | | |
| … |

**Name:** variable length string (UTF-8)

The index should be less than the number of control variables present on the device (see CMD\_GET\_CTRL\_VAR\_COUNT).

## CMD\_GET\_CTRL\_VAR\_INFO (0x52)

This command returns representational and bounds information about the specified control variable.

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters:

**Total length:** 17 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Unit Type | | | | | | | |
| 1 | Minimum | | | | | | | |
| 2 |
| 3 |
| 4 |
| 5 | Maximum | | | | | | | |
| 6 |
| 7 |
| 8 |
| 9 | Scale | | | | | | | |
| 10 |
| 11 |
| 12 |
| 13 | Offset | | | | | | | |
| 14 |
| 15 |
| 16 |

**Unit Type:** 8 bits unsigned

**Minimum:** 32 bits signed

**Maximum:** 32 bits signed

**Scale:** 32 bit floating point (IEEE 754 binary32)

**Offset:** 32 bit floating point (IEEE 754 binary32)

The index should be less than the number of control variables present on the device (see CMD\_GET\_CTRL\_VAR\_COUNT).

Unit Type corresponds to an entry in the unit table definition, and defines the units of the variable’s values.

The minimum and maximum define the bounds on values sent with CMD\_SET\_CTRL\_VAR. The minimum and maximum are both inclusive.

The Scale and Offset parameters are used to change the variable’s value (or min/max) into the given unit type. The value is first multiplied by the scale, and then this product is added to the offset to get the final value.

## CMD\_GET\_CTRL\_VAR (0x53)

This command returns the value of a specific control variable on the device.

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters:

**Total length:** 4 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Value | | | | | | | |
| 1 |
| 2 |
| 3 |

**Value:** 32 bits signed

The index should be less than the number of control variables present on the device (see CMD\_GET\_CTRL\_VAR\_COUNT).

## CMD\_SET\_CTRL\_VAR (0x54)

This command sets the value of a specific control variable on the device.

Outgoing Parameters:

**Total length:** 5 bytes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |
| 1 | Value | | | | | | | |
| 2 |
| 3 |
| 4 |

**Index:** 8 bits unsigned

**Value:** 32 bits signed

Incoming Parameters:

None

The index should be less than the number of control variables present on the device (see CMD\_GET\_CTRL\_VAR\_COUNT).

## CMD\_GET\_SETTING\_COUNT (0x60)

This command returns the number of settings present on the device. This number may be zero.

Outgoing Parameters:

None

Incoming Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Count | | | | | | | |

**Count:** 8 bits unsigned

## CMD\_GET\_SETTING\_NAME (0x61)

This command returns the name of a specific setting on the device.

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Name | | | | | | | |
| … |

**Name:** variable length string (UTF-8)

The index should be less than the number of settings present on the device (see CMD\_GET\_SETTING\_COUNT).

## CMD\_GET\_SETTING\_INFO (0x62)

This command returns information about the specified setting.

Outgoing Parameters:

**Total length:** 1 byte

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Index | | | | | | | |

**Index:** 8 bits unsigned

Incoming Parameters:

**Total length:** variable (≥1 byte)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Setting Type | | | | | | | |
| 1 | Setting Info | | | | | | | |
| … |

**Setting Type:** 8 bits unsigned

**Setting Info:** variable length byte array

The index should be less than the number of settings present on the device (see CMD\_GET\_SETTING\_COUNT).

Setting Type corresponds to an entry in the setting table definition, and defines the organization and interpretation of the setting info.

# Error Handling

If the reply packet contains CMD\_REPLY\_ERROR (0xFF) instead of the expected command ID, it means there was an error while processing the command. The incoming parameters contain information about the error.

Incoming Parameters:

**Total length:** variable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte # | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0 | Error Code | | | | | | | |
| … | Error Parameters | | | | | | | |

**Error Code:** 8 bit unsigned

**Error Parameters:** variable length byte array

The error code indicates the type of error that happened. The error parameters are optional, and not intended to be automatically interpreted. The error parameters are included to help with debugging.