

Remote Control

Information about the remote control interface of the load, including the underlying protocol, and differences between supported transports.

- [Protocol](#)
- [Properties](#)

Protocol

This page describes the native communications protocol with the device. It is binary based, with a small header, and no other requirements on the data.

By convention, most (if not all) endpoints accept and provide data that is CBOR encoded.

Header

All packets have a simple four byte header that precedes the payload, regardless of the underlying transport:

- 1 byte: Message type
 - Indicates the "endpoint" inside the protocol handler that should receive this message
- 1 byte: Tag
 - Can be used to differentiate multiple outstanding requests and their replies
- 2 bytes: Length
 - Indicates the number of bytes of payload data that follow

Note that all multi-byte values in the header are sent in big endian (network) byte order.

Message Types

Below are all currently implemented message types/endpoints:

Number	Name	Description
0x01	Property Request	Get/set various properties on the device. See <u>Properties</u> for more info.

Transports

This same protocol can be carried over a variety of physical transports. The format of messages, including the packet header, are the same; however, the transport may add additional headers and padding, if needed. At this time, the following transports are supported:

USB

The first interface exposed by the device is a vendor specific interface, which consists of two bulk endpoints (one in and one out) used to transmit and receive the packets and their responses; as well as an interrupt endpoint, used to notify the host that the state of the device changed.

In this case, the device does not send unsolicited data. No additional headers are added to payloads sent over the endpoints, beyond the above packet headers.

Properties

This page describes the "property request" mechanism, as well as the message format used to interact with it.

Message Format

All messages on this endpoint are CBOR encoded.

Requests

Requests are maps, with one or more of the following keys:

- `get`: An array of property IDs to read
- `set`: A map containing properties to set. Keys in the map correspond to property IDs.

Replies

The device responds with a map, which contains the following keys. Which keys are included in the response depends on the request:

- `get`: A map (keyed by property IDs) containing the value of all requested properties. Unsupported/unknown properties are returned as `undefined`.
- `set`: An array containing the property IDs of all properties that were set. Any properties that were requested to be set, but are not supported (or read-only) will not be included.

Supported Properties

Below are all currently supported properties, including their IDs and value types:

ID	Name	R/W	Type	Description
0x01	HwSerial	R	string	Serial number of the hardware
0x02	HwVersion	R	string	Version information (such as revision) for the device

0x03	HwInventory	R	array	<p>Information about all peripherals connected to the load. The array contains maps, which will have the following keys:</p> <ul style="list-style-type: none"> • type: Peripheral type; may be one of ["load", "hmi" or "io"] • sn: Serial number (string; optional) • driver: Driver id (blob; optional)
0x04	SwVersion	R	string	Current software version (including build number)
0x05	MaxVoltage	R	int	Maximum allowable input voltage (mV)
0x06	MaxCurrent	R	int	Maximum allowable input current (mA)
0x07	DefaultVSense	RW	int	<p>Voltage sense source on power-on:</p> <ul style="list-style-type: none"> • -1 = state at last power off • 0 = internal • 1 = external <p>This setting is persistent.</p>

0x08	DefaultMode	RW	int	<p>Operation mode on power-on:</p> <ul style="list-style-type: none"> • -1 = mode at last power off • 0 = Constant current • 1 = Constant voltage • 2 = Constant wattage <p>This setting is persistent.</p>
0x09	DefaultCurrent	RW	int	<p>Current limit value (for constant current mode) to apply on power on, in mA. -1 = last user specified value at power off</p>
0x0A	DefaultVoltage	RW	int	<p>Voltage limit value (for constant voltage mode) to apply at power on, in mV. -1 = last user specified value at power off</p>
0x0B	DefaultWattage	RW	int	<p>Wattage (for constant wattage mode) to apply at power on, in mW. -1 = last user specified value at power off</p>