## Command Protocol

The host can send the device commands to read and write settings in the device. The host can send the configuration messages via the serial port, MQTT or a LoRa message. The configuration messages are snippets of Json. Every command will receive an acknowledgement. Note that not all configuration commands take effect immediately. Some require the device to be reset for the new setting to take effect. These commands are identified in the fowling documentation. There is a remote command that can be used to force a reset of the device.

## Command Structure

Each command is a json snippet with the following overall structure:

{"v":***ver***,"t":"***target***","c":"***family***","o":"***option***","val":***value***,"seq":***num***}

The **v** element of a command is the version of the protocol. The ***ver*** value gives the protocol version expressed as an integer. At present there is only one version of the protocol, this is version 1. If an incorrect version number is used by the host the device will reply with an error message which includes the version number of the protocol in use by that device.

The **t** element of a command identifies the target device for the command. Each device has a unique name. The ***target*** value gives the name of the target device. If this name does not match that in the device the command is rejected with an error. The only command that does not require a target value is the **getdevname** command which can be used to determine the name of a device.

The **c** element is the command family of the command. The ***family*** value gives the command family that this command is part of. There are four command families:

* node – commands to configure the node
* wifi – commands to configure wifi settings
* lora – commands to configure LoRa settings
* mqtt – commands to configure MQTT settings

The **o** element is the option of the command family that is in use. The ***option*** value of the command specifies the option from the command family which is being used.

The **val** element is provided to set the value of a given command option. The contents of ***value*** are provided to set the value of a given option. They may be numeric or a string. This element is optional for many commands. If the **val** element is left off the command is usually interpreted as a request to read the given setting.

The **seq** element is provided to allow the sender to uniquely identify a given command that has been issued. The contents of ***num*** are an integer sequence number which is set by the host. The reply message from the device will contain the same sequence number. This element is optional for all commands.

## Reply structure

Each command will receive a reply from the device. The reply will contain an error number which will be zero if the command was completed successfully. Appendix 1 of this document gives the meaning of each error number.

The reply may also contain values that have been requested, along with a sequence number if one was given in the command.

## Sample Command

The command getdevname gets the name of a

{"v":1, "t" : "sensor01", "c" : "node", "o" : "devname", "val":"sensor03"}

{"v":1, "c":"node", "o":"getdevname", "seq":99}

## Get Protocol Version

Commands are pre-ceded by a version number. A given device uses a particular version of the protocol and will reject any messages with a different version number. You can find the protocol version in a device by using the Get Protocol version command. This command is unique in that it does not require a version number:

### Reply

{"version":1,"error":0}

## Get Device Name

Gets the name of the device. This is the name that is used to send commands to the device. Note that this is not the MQTT or LoRa name of the device.

{"v":1, "c" : "node", "o" : "getdevname"}

### Reply

The reply is the name of the device.

{"nodename":sensor01,"error":0}

## Set the Device Name

Sets the name of the device. The name of the device is used to identify commands that are to be performed by this device. It is **not** the same as the MQTT name of the device, which is set using an MQTT command.

{"v":1, "t":"sensor01","c":"node","o":"devname","val":"sensor02"}

The above command would change the name of the device from sensor01 to sensor02.