NETWORK OWL API [PUBLIC]

REVISION HISTORY

Date	Author	Description
07/09/2011	Barry Webb	Document Created
26/04/2012	Barry Webb	Added List of commands available in Beta 1
09/05/2012	Barry Webb	Added documentation of the production process
13/03/2013	Luke Knight	Added up to date documentation of most commands
14/03/2013	Luke Knight	Added further documentation of more commands
25/03/2013	Luke Knight	Updated to public version
08/04/2013	Barry Webb	Added documentation for network comms

TABLE OF CONTENTS

NETWORK COMMUNICATIONS

AVAILABLE TRANSPORTS

The Network OWL current has two methods of communicating live data to remote applications.

1. UDP Multicast (Local Network Only)
2. UDP Push (Local Network or WAN)

The Network OWL transmits the same format data on both transports and both can be used at the same time. i.e. Multicast for in home display and UDP Push for server side data collection. For details of the XML format of the live data, please see the document "Network OWL Multicast".

Applications can also send commands to the Network OWL via the UDP interface. Commands can be sent from anywhere (LAN or WAN), but will only be processed if a valid UDP key is appended to the end of the command. Each Network OWL has its own UDP key, which can be requested from The OWL Customer services department.

Both multicast and push can be used to discover the IP address of the Network OWL. As data is transmitted regularly from the Network OWL, the receiving application can capture the IP address that transmitted the packet and store it for later use. This is then used when sending commands to the Network OWL.

UDP MULTICAST

The Network OWL automatically sends UDP packets to the following multicast group:

Address: 224.192.32.19

Port: 22600

There are a number of multicast examples available on the internet, some listed below. The values above can be used with these examples to get you started receiving multicast data from the Network OWL.

Java
http://lycog.com/programming/multicast-programming-java/

C/C++
http://ntrg.cs.tcd.ie/undergrad/4ba2/multicast/antony/example.html

Multicast packets are generally limited to the local subnet, but this may be different depending on router configurations. For more information see the "Network OWL Multicast" document and the example multicast server application.

UDP PUSH

The Network OWL can be configured to send unicast UDP packets to a specified IP address and port. These packets are the same format as the multicast packets and are sent at the same time. A server application needs to be set up to listen on a UDP port. The server needs to be accessible from the same network as the Network OWL, this may involve configuring firewalls.

By default UDP push is disabled on the Network OWL. You can enable this feature with the SET,UDP command (see below for command format). This command can be sent to the Network OWL via UDP if you know the Network OWL's IP address, or The OWL customer services can configure this remotely.

Note: Users will be able to configure this via the Owl Intuition web interface in a future version.

COMMAND PROCESSING

The Network OWL constantly listens for commands on UDP port 5100. A command must be sent in a single UDP packet for it to be processed by the Network OWL. If a command is accepted and processed, the Network OWL will send a UDP response to the IP address and port that sent the request. This way any application can configure the Network OWL, not just the server application defined in the UDP push settings.

For a command to be accepted, it must contain the UDP key. For example:

VERSION,11AA22BB

The UDP key **MUST** be in upper case for it to be accepted.

CONFIGURATION COMMANDS

Configuration commands on this device can be one of five types, SET/GET/ADD/DEL/action. To set the configuration parameters, prepend the string “SET,” to the command. To get the current configuration settings, prepend the string “GET,” to the command. To perform the command action, send the command with no prefix.

For some commands SET, GET, ADD, DEL or action are not appropriate. In these cases the documentation for the command will reflect this.

VERSION

Description: Retrieves the version information of the device.

Responses:

OK,VERSION,(firmware),(version),(build)

DEVICE

Description: Manages internal device list. Allows user to add, delete and view device details.

Get Parameters: GET,DEVICE,(device index)

- Device Index: 0 to 9 or ALL. ALL will return a comma separated list of paired device types showing which device is in which location. 0 to 9 will return device details for that index.

Responses (get):

ALL (example): OK,DEVICE,CMR180,ROOM_STAT,NONE,NONE,NONE,NONE,NONE,NONE,NONE,NONE

0 to 9: OK,DEVICE,(index),(device address),(device type),(seconds since last rx packet),(device state),(RSSI),(LQI),(battery volts/state),(rx packets),(tx packets)

Del Parameters: DEL,DEVICE,(device index)

- Device index: the index of the device to be deleted

Responses (add & del):

OK,DEVICE

SCAN

Description: Puts the device into wireless scanning mode and starts looking for wireless devices. Once 1 device is found, scanning mode is cancelled.

Action Parameters (optional): SCAN,(device_type)

- Device Type: 119, 160 or 180 to scan for electricity transmitters. If left blank, the Network OWL will scan for heating or hot water devices (Room Stat, Tank Sensor, Relay Box, etc)

Responses:

OK,SCAN

UPTIME

Description: Retrieves the run time of the device in days, hours, minutes and seconds.

Responses:

OK,UPTIME,(uptime)

CLOCK

Description: Allows for the retrieval of the network owl's clock data (in seconds).

Responses:

OK,CLOCK,(UTC time in seconds),(Local time in seconds)

DAYHEATINGPERIODS

Description: Retrieves list of heating periods for a specified day

Get Parameters: GET,DAYHEATINGPERIODS,(day)

- Day: 0 – 6 = Monday – Sunday.

Responses:

OK,DAYHEATINGPERIODS,(day),[(start time in seconds),(end time in seconds),(temperature)],[. . .],[. . .]

DAYHWPERIODS

Description: Retrieves hot water periods for a specified day

Get Parameters: GET,DAYHWPERIODS,(day)

- Day: 0 – 6 = Monday – Sunday.

Responses:

OK,DAYHWPERIODS,(day),1,[(flags),(start time in seconds),(end time in seconds),(temperature)],[. . .],[. . .]

HEATINGDAY

Description: Deletes heating periods for a specific day.

Del Parameters: DEL,HEATINGDAY,(day)

- Day: 0 – 6 = Monday – Sunday

Responses:

OK,HEATINGDAY,(day)

HWDAY

Description: Deletes hot water periods for a specific day.

Del Parameters: DEL,HWDAY,(day)

- Day: 0 – 6 = Monday – Sunday

Responses:

OK,HWDAY,(day)

BOOST

Description: Boosts the heating temperature.

Set Parameters: SET,BOOST,(ON/OFF)

Get Parameters: GET,BOOST

Responses (all):

OK,BOOST,ON

OK,BOOST,OFF

MOREHW

Description: Overrides the hot water controls.

Set Parameters: SET,MOREHW,(ON/OFF),(temperature)

- ON = overriding hot water. OFF = not overriding.
- Temperature: the temperature to be maintained while overriding.

Reponses:

OK,MOREHW

ELECTRICITY

Description: Configures electricity device settings for standard, solar and 3 phase.

Set Parameters: SET,ELECTRICITY,(mode),(flags),(voltage),(power factor)

- Mode: 0 for single phase, 1 for 3 phase, 2 for PV
- Flags:
  - (3Phase) Bit mask of channels to use when calculating property consumption
  - (PV)
    - 0 = standard PV
    - 2 = Type 2 PV (Y-Cable)

        - 3 = PV Gross (everything generated is exported)
- Voltage: system voltage (default 230V)
- Power factor: default 1.0

Get Parameters: GET,ELECTRICITY

Responses:

OK,ELECTRICITY,(mode),(flags),(voltage),(power factor)

REBOOT

Description: Reboots the device.

Responses:

OK,REBOOT

Note: The device may reboot before this response is sent, if the command has been issued over the network.

PVCONFIG

Description: Configures solar parameters. (sunrise/sunset offsets).

Get Parameters: GET,PVCONFIG

Set Parameters: SET,PVCONFIG,(sunrise offset),(sunset offset),(lower limit),(use phantom power),(phantom power)

- Sunrise Offset: Number of seconds before or after sunrise to start using PV data
- Sunset Offset: Number of seconds before or after sunset to stop using PV data
- Lower limit: (watts) Any PV reading below this value is discarded and the reading is set to 0.
    - **[IMPORTANT]** When using phantom power detection, set this to 1.
- Use phantom power: (0/1) whether or not new phantom power algorithm should be used
- Phantom power: how much the algorithm will compensate for on the current day

Responses:

OK,PVCONFIG,(sunrise offset),(sunset offset),(lower limit),(use phantom power),(phantom power)

TARIFF

Description: Configures electricity tariff data

Get Parameters: GET,TARIFF

Set Parameters: SET,TARIFF,(tariff type),(standing charge),(standard cost),[economy7 cost],[economy7 start],[economy7 end]

- Tariff type: (1/2) = standard/economy 7 method of charging
- Standing charge: the standard charge taken per day
- Standard cost: the standard cost per unit
- Economy7 cost: [only if Tariff type = 1] the off-peak cost per unit

- Economy7 start: [only if Tariff type = 1] the time that off-peak begins
- Economy7 end: [only if Tariff type = 1] the time that off-peak ends

Responses:

OK,TARIFF,(tariff type),(standing charge),(standard cost),[economy7 cost],[economy7 start],[economy7 end]

TZ

Description: Configures timezone data.

Set Parameters: SET,TZ,(offset)

- Offset: The offset from GMT in seconds (-43200 to 43200)

Get Parameters: GET,TZ

Responses:

OK,TZ,(offset)

HOLIDAY

Description: Configures holiday settings.

Set Parameters: SET,HOLIDAY,(holiday start),(holiday end)

- Holiday start: the day of the year to start the holiday. (1-365)
- Holiday start: the day of the year to end the holiday. (1-365)

Get Parameters: GET,HOLIDAY

Responses:

OK,HOLIDAY,(holiday start),(holiday end)

AWAY

Description: Toggles Away mode & sets times.

Set Parameters: SET,AWAY,[start time],[end time]

- Start time: [optional] sets the time in seconds that the away mode should begin.
- End time: [optional] sets the time in seconds that the away mode should end.

If no optional parameters entered, the command will toggle Away mode between on and off. To disable away mode manually, set both start and end times to 0.

Get Parameters: GET,AWAY

Responses:

OK,AWAY,(start time),(end time)

COMFORT

Description: Sets Comfort mode to on.

Responses:

OK,COMFORT

STANDBY

Description: Sets Standby mode to on.

Responses:

OK,STANDBY

HEATING

Description: Configures heating device settings.

Set Parameters: SET,HEATING,(upper temp),(lower temp),(boost temp),(standby unoccupied temp),(standby occupied temp), (standby sleep temp),(away temp),(holiday temp),(boost time),(min boiler time),(warm up offset),(cool down offset),(max warm up time),(max cool down time),(comfort report interval),(standby report interval), (away report interval),(summer report interval),(temp hysteresis)

- upper temp: Not used, set to 35.0
- lower temp: Not used, set to 5.0
- boost temp: Number of degrees (C) to increase the required temperature by when boosting the heating
- standby unoccupied temp: Required temperature in standby when property unoccupied
- standby occupied temp: Required temperature in standby when property occupied
- standby sleep temp: Required temperature in standby when in sleep mode
- away temp: Required temperature when in away mode
- holiday temp: Required temperature when in holiday mode
- boost time: Number of seconds to remain boosted for
- min boiler time: No used, set to 120
- warm up offset: Percentage modifier for calculated warm up time (100% = use calculated warm up time, 50% = use ½ calculated warm up time)
- cool down offset: Percentage modifier for calculated cool down time
- max warm up time: Maximum number of seconds for warm up duration
- max cool down time: Maximum number of seconds for cool down duration
- comfort report interval: Number of seconds between reports from remote devices when in a comfort period.
- standby report interval: Number of seconds between reports from remote devices when in a standby period.
- away report interval: Number of seconds between reports from remote devices when in away mode
- summer report interval: Number of seconds between reports from remote devices when in summer mode.
- temp hysteresis: Number of degrees (or fraction of) the temperature has to drop, below the required temperature, before the heating will turn on again.

Responses:

OK,HEATING,(upper temp),(lower temp),(boost temp),(standby temp),(comfort temp), (sleep temp),(away temp),(holiday temp),(boost time),(min boiler time),(warmup offset),(cooldown offset),(max warmup time),(max cooldown time),(comfort report interval),(standby report interval),(away report interval),(summer report interval),(temp hysteresis)

SUMMER

Description: Configures summer start and end dates.

Set Parameters: SET,SUMMER,(start date),(end date)

- Start date: the day of the year to start summer mode. (1-365)
- End date: the day of the year to end summer mode. (1-365)

Get Parameters: GET,SUMMER

Responses:

OK,SUMMER,(start date),(end date)

PROPERTY

Description: Sets EPC rating, latitude and longitude of the property the system is installed at. Latitude and longitude are used for sunrise/sunset calculations. They are stored on the Network OWL and not transmitted anywhere except as a response to this command. (OWL Servers do not request/use this information)

Set Parameters: SET,PROPERTY,(epc),(lat),(lon)

- Epc: (A – G) sets the EPC rating of the property.
- Lat: sets the latitude of the property.
- Lon: sets the longitude of the property.

Get Parameters: GET,PROPERTY

Responses:

OK,PROPERTY,(epc),(lat),(lon)

RESET

Description: Resets network owl settings to factory defaults.

Responses:

OK,RESET

Note: The Network Owl will reboot following this reset.

MAC

Description: Returns the MAC ID of this network owl.

Responses:

OK,MAC,(mac id)

DST

Description: Gets and sets whether or not Daylight Savings is used

Set Parameters: SET,DST,(used)

- Used: if ‘1’ then DST will be used – any other number will result in DST not being used.

Get Parameters: GET,DST

Responses:

OK,DST,(used),(dst start time in secs from 1/1/1970 00:00:00),(dst end time in secs)

UDP

Description: Configures the UDP Push functionality

Set Parameters: SET,UDP,(hostname),(ip address),(port)

- Used: if ‘1’ then DST will be used – any other number will result in DST not being used.

Get Parameters: GET,DST

Responses:

OK,DST,(used),(dst start time in secs from 1/1/1970 00:00:00),(dst end time in secs)

SAVE

Description: Saves network owl settings.

Responses:

OK,SAVE

MANAGING THE TIME CLOCKS

To Be Documented.