

iShower API

Conventions

The content of the exchanged messages is specified within quotes in this document for clarity. The actual messages sent or received from the shower will not have the quotes.

Discovery

The shower can be discovered by broadcasting a UDP packet on the local subnet to port 30303. The message in the packet should contain 'ua' as the first two characters. The shower will respond back with a UDP packet on port 30303 and the following message content `SWRvvvvvMMMMMMMMMM\0' where vvvvv is the revision number and MMM...MMM is the ethernet mac address

Connection and subscription for status updates

Once the IP address of the shower is discovered or is known in advance a TCP connection to port 21825 must be established. Upon connecting the shower will respond with +OK message (see the uaapi reference).

After receipt of the +OK connection message a subscription for the shower status updated must be made by sending the `+/shower/o/status` message. Upon successful subscription the shower will send periodic status update messages. These must be received and processed.

States

The status update messages contain a field for the state of the shower. The field header is 'state:' followed by the state mnemonic. The shower will report one of the following states:

- "ini" -- initialization (only on device startup)
- "mns" -- main state
- "cls" -- power saver
- "pst" -- preset selection
- "msg" -- pending alert confirmation
- "mnu" -- menu
- "del" -- preset deletion selection
- "pas" -- password entry
- "val" -- valve assignment

All commands must be send from the main state "mns". To transition from the various states to the main state the following messages must be send:

- "ini" -- the device will transition to the main state after initialization
- "cls" -- `@/shower/i/slider/t/cw<1`
- "pst" -- `@/shower/i/button/nw<1`

Proprietary and Confidential

```
"msg" -- `@/shower/i/slider/t/cw<1'  
"mnu" -- `@/shower/i/button/sw<1'  
"del" -- `@/shower/i/button/sw<1'  
"pas" -- `@/shower/i/button/sw<1'  
"val" -- `@/shower/i/button/sw<1'
```

LED commands

The LED on/off state is shown in a field with a header of `led:' followed by `0' for OFF and `1' for ON. The color of the LED is broken down into its red, blue and green components and available in fields with the `r:', `g:' and `b:' headings correspondingly followed by the value of the color component. The following 37 colors are available ({ r, g, b}):

```
{ 255, 255, 255 }  
{ 255, 255, 0 }  
{ 255, 170, 0 }  
{ 255, 128, 0 }  
{ 255, 102, 0 }  
{ 255, 0, 0 }  
{ 255, 0, 128 }  
{ 255, 0, 255 }  
{ 128, 0, 255 }  
{ 0, 0, 255 }  
{ 0, 255, 102 }  
{ 0, 255, 0 }  
{ 128, 255, 0 }  
{ 170, 170, 0 }  
{ 170, 113, 0 }  
{ 170, 85, 0 }  
{ 170, 68, 0 }  
{ 170, 0, 0 }  
{ 170, 0, 85 }  
{ 170, 0, 170 }  
{ 85, 0, 170 }  
{ 0, 0, 170 }  
{ 0, 170, 68 }  
{ 0, 170, 0 }  
{ 85, 170, 0 }  
{ 85, 85, 0 }  
{ 85, 57, 0 }  
{ 85, 43, 0 }  
{ 85, 34, 0 }  
{ 85, 0, 0 }  
{ 85, 0, 43 }  
{ 85, 0, 85 }  
{ 43, 0, 85 }
```

Proprietary and Confidential

```
{ 0, 0, 85 }  
{ 0, 85, 34 }  
{ 0, 85, 0 }  
{ 43, 85, 0 }
```

To toggle the LED on/off send the following message:

```
'@/shower/i/button/8<1'
```

To change to the previous colour send the following message:

```
'@/shower/i/button/7<1'
```

To change to the next colour send the following message:

```
'@/shower/i/button/9<1'
```

Aroma commands

The Aroma state is shown in a field with header `oil:' followed by a number specifying which oil pump is currently engaged -- 0 for No pump working, 1 for oil pump 1, 2 for oil pump 2, and finally 3 for oil pump 3. In order to turn on for 10 seconds a particular oil pump the following messages can be send:

```
'@/shower/i/button/4<1' -- for oil pump 1  
'@/shower/i/button/5<1' -- for oil pump 2  
'@/shower/i/button/6<1' -- for oil pump 3
```

Outlet commands

The shower supports up to 7 outlets numbered from 0 to 6. The currently selected outlet is shown in the status messages under a heading of `o:' followed by the outlet number.

To change the selected outlet the following messages can be send:

```
'@/shower/i/button/n<1' -- for outlet 0  
'@/shower/i/button/w<1' -- for outlet 1  
'@/shower/i/button/s<1' -- for outlet 2  
'@/shower/i/button/0<1' -- for outlet 3  
'@/shower/i/button/1<1' -- for outlet 4  
'@/shower/i/button/2<1' -- for outlet 5  
'@/shower/i/button/3<1' -- for outlet 6
```

The current water temperature for a particular outlet is shown under a field with a heading of `cTX:' where X is the outlet number followed by the value in degrees Celsius. Similarly, the current flow settings is under a field with heading of `cFX:' where X is the outlet number and followed by a value from 0 (minimum flow) to 7 (maximum flow). The current outlet open/close mode is shown in a field with heading of `cSX:' where X is the outlet number and the field value is 0 for closed outlet and 1 for open (water running). Each outlet' desired values are shown in fields with headings of `dTX:', `dFX:', and `dSX:' for the temperature, flow, and mode values respectively.

To toggle an outlet between open / close the following command should be send:

```
'@/shower/i/button/c<1'
```

To decrease the outlet' desired temperature the following command should be send:

```
'@/shower/i/slider/b/ccw<1'
```

To increase the outlet' desired flow rate the following command should be send:

MEMO
STEAMOMATIC API - r.100

Proprietary and Confidential

UNIQUE
automation

`@/shower/i/slider/t/cw<1'

To decrease the outlet' desired flow rate the following command should be send:

`@/shower/i/slider/t/ccw<1'