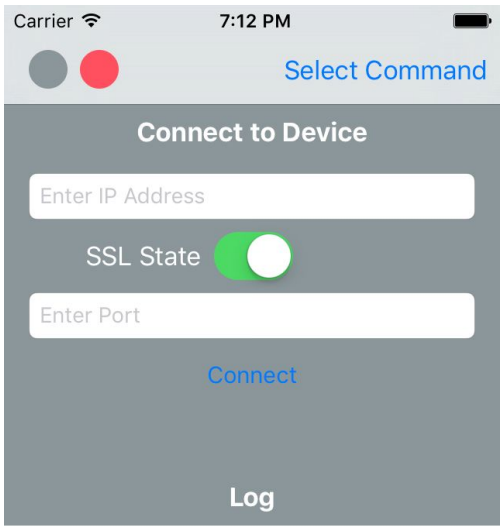
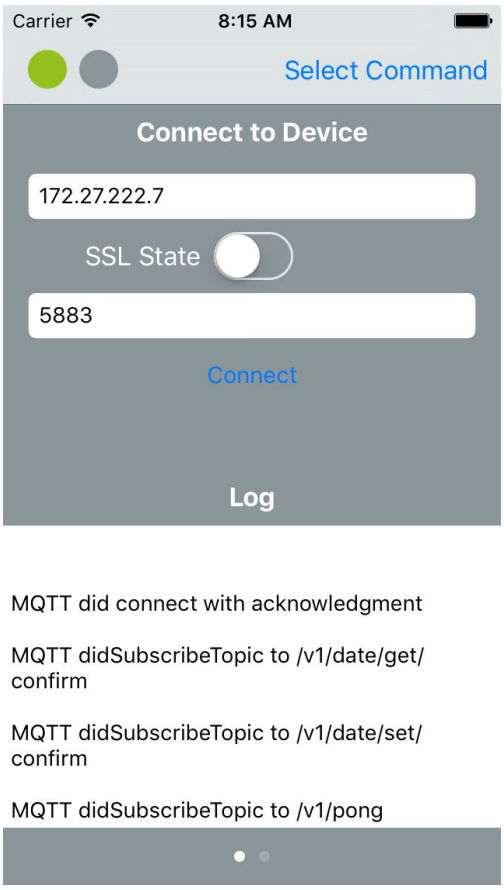
Somabar Prototype App

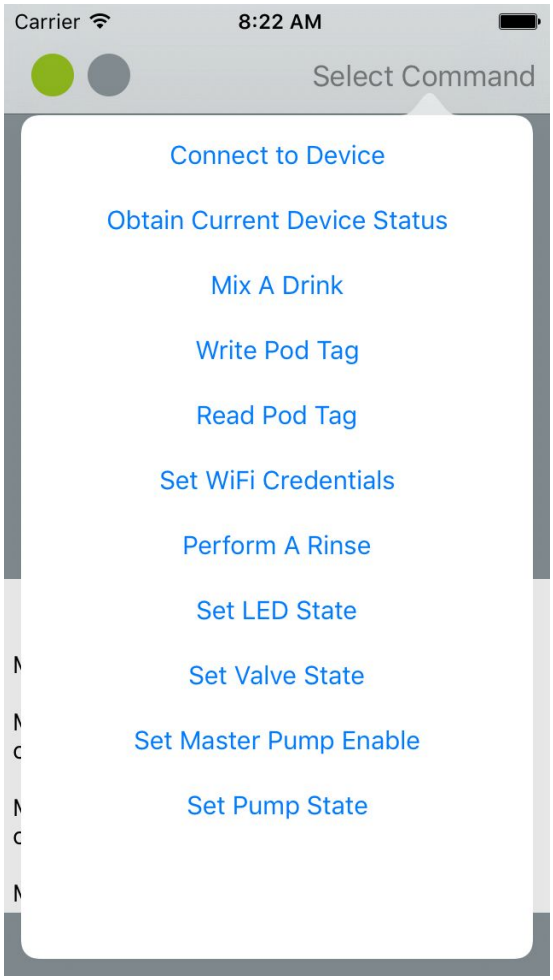
Connecting to the Somabar





To connect to the Somabar, enter the correct IP Address and Port in the designated fields and select if the connection will be using SSL or not. Then, click on “Connect”. After selecting connect, you should see the green light appear in the top right corner and a confirmation of connection to the mqtt client, in addition to notifications of the subscription to the necessary topics (see image right)

Selecting and Sending a Command



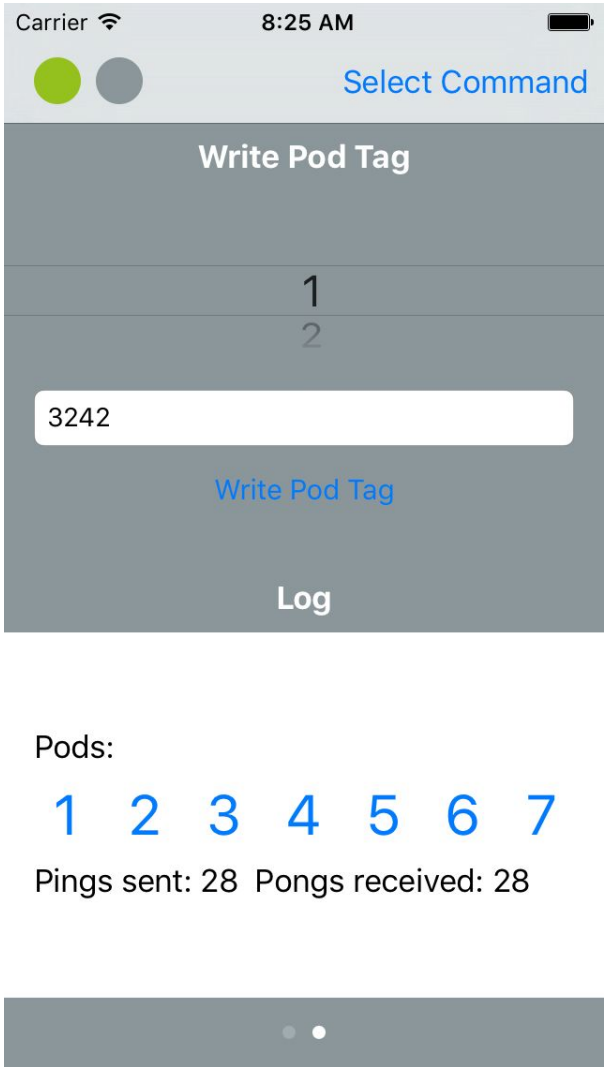
To select a command, click “Select Command” in the top right corner. This will bring up a list of possible commands that can be executed (see image left). Selecting a command will bring up a new page, which will allow you to enter arguments that will be sent to the Somabar (if necessary) and a blue button to initiate the sending. Once sent, you should see a notification in the log confirming that your command has been published with the desired message and to the desired path.

Monitoring the Log and Pods

The log is the bottom half of the screen, and is available regardless of which command you are currently on. It is a scrollable view with two pages:

1. The first page is a vertically scrollable log of all commands that have been sent and received by your device. Here you can see when you have sent a command, and also if you have received any information in return from the Somabar. See “The Commands” section for a full list of each command and its expected bahaviour.

2. The second scrollable view displays how many pings have been sent to the somabar and how many pongs have been returned. It also has a button for each pod which, when clicked, displays a notification with that pod's most recently sent status.



The Commands

Basic Connectivity Check: Send a ping via MQTT to ellicit a Pong

Parameters:
Takes no parameters

Response:
{
  "timestamp": 1471616873.4102029,
  "token": "16bba3ab661911e6a75aacbc32d2a437"
}

Obtain Current Device Status: Retrieve status information about the Somabar

Parameters:
Takes no parameters

Response:
{
  "status": "ready",
  "uptime": "2 days 23:24:45.050000",
  "version": "0.0.17",
  "token": "412c0e14684811e6ab81acbc32d2a437",
  "last_drink": null,
  "timestamp": 1434351836.5730679,
  "date_time_utc": "2015-06-15T07:03:56.572428"
}

Mix A Drink: Mix a drink from a recipe description

Parameters:
recipe_id : list of ingredients to be used to create the drink. pod numbers are 1-7 (see above for explanation), pump times are in seconds
Ingredients: integer id of recipe to be made, used to track last drink made

Response:
{
  "timestamp": 1471616873.4102029,

```
  "token": "16bba3ab661911e6a75aacbc32d2a437"
}
```


Write Pod Tag: Write a JSON payload to a pod RFID tag. NOTE: The entire body of this request will be written to the tag, however the parameters listed below must be contained for a write to occur. Care must be taken not to exceed the storage capacity of the tags.

Parameters:
Pod_id Pod number to be written. pod numbers are 1-7.
Ingredient_id Integer representing the ingredient ID

Response:


```
{
  "value": {
     "ingredent_id": 123456
   },
   "timestamp": 1471616873.4102029,
   "token": "16bba3ab661911e6a75aacbc32d2a437"
}
```


Read Pod Tag: Read current value from drink pod(s)

Parameters:
Pod_id Pod number to read. pod numbers are 1-7. if "all" is provided, all pods will be read. One message will be generated per pod

Response:


```
{
  "value": {
     "ingredent_id": 123456
   },
   "timestamp": 1471616873.4102029,
```

"token": "16bba3ab661911e6a75aacbc32d2a437"
}

Set WiFi Credentials: Set wifi credentials and immediately join the new network

Parameters:
Ssid : SSID to be used. Must be 1-31 numbers, letters, or underscores
Password: Password to be used. Must be 1-31 numbers, letters, or underscores

Response:
No Response

Perform A Rinse: Initiate a rinse procedure (default to 5s)

Parameters:
Duration Rinse duration in seconds

Response:
{
"timestamp": 1471616873.4102029,
"token": "16bba3ab661911e6a75aacbc32d2a437"
}

TEST COMMAND - Set LED State: Manually set the state of an LED. May cause undefined behavior if used in normal operation. This command WILL be removed before final consumer release.

Parameters:
Led_name LED to command. Must be "blue", "green", or "white"
State: State for the LED. Must be "on", "off", or "blink"

Response:
No Response

TEST COMMAND - Set Valve State: Manually set the state of a valve. May cause undefined behavior if used in normal operation. This command WILL be removed before final consumer release.

Parameters:
State: State for the valve. Must be "on" or "off"

Response:
No Response

TEST COMMAND - Set Master Pump Enable: Manually set the state of the Master Pump Enable pin, enabling the 50Hz oscillator necessary for pumps to operate. Must be set to 'on' for "Set Pump State" command to work. May cause undefined behavior if used in normal operation. This command WILL be removed before final consumer release.

Parameters:
State: State for the Master Enable. Must be "on" or "off"

Response:
No Response

TEST COMMAND - Set Pump State: Manually set the state of a pump. May cause undefined behavior if used in normal operation. This command WILL be removed before final consumer release.

Parameters:
Pump_id Pump to command. See table above.
State: State for the pump. Must be "on" or "off"

Response:
No Response