

DC Motor Integration in BEMOSS

Brian Lauer

Advisor: Dr. Suruz Miah

Department of Electrical and Computer Engineering
Bradley University
1501 W. Bradley Avenue
Peoria, IL, 61625, USA

Friday, August 14, 2019

Outline

- 1 Device API Translator
- 2 Steps in adding device API
- 3 Steps in adding new device type
- 4 Problems Fixed
- 5 Additions
- 6 Priorities

Device API Translator

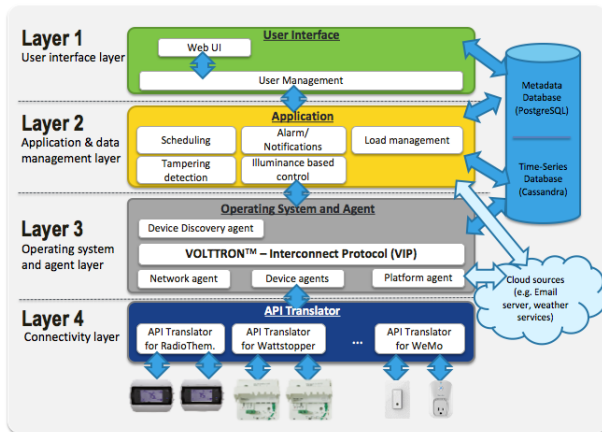


Figure: Courtesy of BEMOSS developer resources

Steps in adding device API

Methods to implement

- `API_info` returns list of supported device attributes (`device_model`, `vendor_name`, `html_template`, `chart_template` etc.)
- `dashboard_view` returns parameters of device to be shown on the BEMOSS dashboard
- `ontology` maps parameters from device to BEMOSS parameters
- `discover` is called by the device discovery agent and returns attributes of discovered devices
- `getDataFromDevice` returns dictionary of device data
- `identifyDevice` visually identifies the device, triggered by web UI and sends message to control agent
- `setDeviceStatus` sends control commands to device

Steps in adding device API

Methods in motor API

- `argsToPi` uses `ssh` to run `XBEEcontrol.py` with arguments to control motor and retrieve data
- `toggleDeviceStatus` queries the device for on/off status then changes settings of the motor, uses `getDataFromDevice` and `setDeviceStatus`

Steps in adding new device type

Steps

- Select a device type id (1, 2, 3, 4 and 7 are reserved)
- Register new device type in BEMOSS db by updating `PROJECT_DIR/Web_Server/run/defaultDB.py` then running the script
- Update `discover_devices` function in `Web_Server/webapps/discovery/views.py`
- Add widget for new device in `Web_Server/webapps/discovery/templates/discovery/manual_discovery.html`
- Update `Web_Server/static/app_js/discovery.js` with code to enable use of the "Set all to 'Approved'" button in the Device Discovery page

Steps in adding new device type

The screenshot shows a web application interface for device discovery. The browser address bar indicates the URL is `localhost:8082/dashboard/discover`. The left sidebar contains a navigation menu with the following items: Nodes (1), Devices (0), NETWORK STATUS, ALARMS & NOTIFICATIONS, MANAGE USERS (0), MISC SETTINGS, APPLICATIONS, NODE1 (2), and LOG OUT. The main content area is titled "Approved Devices" and features a settings gear icon in the top right. Below the title, there are two tabs: "Plugload" (1) and "Motor" (1). The "Motor" tab is active. A search bar is located above a table. The table has columns: Nickname, Vendor, Model, MAC Address, Date Added, Device Authorization, Assigned Node, Current Node, and Approval Status. A single entry is displayed: Nickname "Motor2", Vendor "Pittman", Model "GM8224D201", MAC Address "b827ebbfade", Date Added "Aug. 14, 2019, 2:24 p.m.", Device Authorization "Node1", Current Node "Node1", and Approval Status "Approved". A green button "Set all to 'Approved'" is in the top right of the table. Below the table, it says "Showing 1 to 1 of 1 entries" and includes pagination buttons: First, Previous, 1, Next, Last. At the bottom of the main content area, there are two buttons: "Save Changes to Motor Controllers" and "Cancel". A section titled "Non-BEMOSS Devices" is visible at the very bottom.

Nickname	Vendor	Model	MAC Address	Date Added	Device Authorization	Assigned Node	Current Node	Approval Status
Motor2	Pittman	GM8224D201	b827ebbfade	Aug. 14, 2019, 2:24 p.m.	Node1	Node1	Node1	Approved

Figure: Device discovery page

Problems Fixed

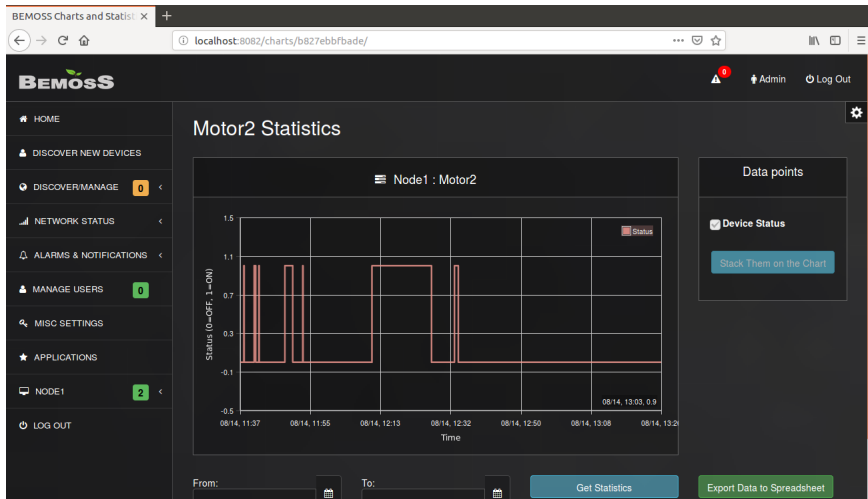


Figure: Fully functional jQuery plot

ssh keys

- ssh-keygen generates a public and private key
- Public key copied to RPi, private key kept on host machine
- More secure than using password
- <https://www.raspberrypi.org/documentation/remote-access/ssh/passwordless.md>

Priorities

- Write paper
- Speed control algorithm
- Still looking for new device