

0.0.3-alpha2 document

Jeeva Kandasamy

2016-07-29

Table of Contents

Introduction	1
Technology.....	1
Installation.....	2
System Requirements	2
Supported Platforms	2
Download.....	2
Configuration	2
Login	4
Menus	4
Utility menu	5
Main menu	5
Sub menu.....	5
Across pages	5
Language (locale)	6
Support	6
Utility menu.....	7
Dashboards	7
Add new widgets	7
Sensors	8
A sensor graphical view	8
Grouped sensors graph	9
Mixed sensors graph.....	10
Sensors bullet graph	10
Heatmap chart	11
Sensors custom button	12
Display image file	15
Groups	16
MyController time	17
Sunrise and Sunset time	17
News	17
Custom widgets	18
Rooms detail	18
Topology	18
Resources	19
Gateways	19
Node	20
Sensors	22
Rules	24

Operations	27
Special operations	29
Timers	31
Forward Payload	33
Groups	33
Rooms	34
Action board	35
Sensors action	35
Send raw message	36
Status	37
About	37
System status	38
Resources logs	41
MyController server log	41
Utilities	41
Scripts	41
Templates	44
HTML additional headers	47
Variables repository	47
UID Tags	48
Firmware	49
External servers	50
Resources data	53
Settings	54
Profile	54
System	54
Notifications	56
MQTT broker	57
Metrics	58
MySensors	59
Variable mapper	59
Users	60
Roles	60
Backup	62
Source Code	63
Issue management	63
License	63



MyController.org

The Open Source Controller

Author(s)	Product Version	Date
Jeeva Kandasamy	0.0.3-alpha2	July-2016

Copyright ©2015-2018, All Rights Reserved, <http://www.mycontroller.org>

Introduction

MyController.org is a controller for the sensors world! Primarily it was developed to MySensors.org project. Later changed it's architecture to support other projects also. Currently supports for [MySensors](#) and [Sparkfun](#). Considered system resources in mind on development. As a result even we can run it on Raspberry PI very first model.

You can control/monitor various sensors with this controller. As this controller is a web application server you can access from anywhere if you have internet/intranet connection. It has various features like, firmware control for nodes, powerful rule engine, timers, custom scripts support(JavaScript, Python, Ruby, Groovy, etc.,). For further details dig into each section.



If you think [MyController.org](#) helps you someway on your daily life, kindly consider to [donate](#) something to MyController. Your donation will help MyController to keep growing. Thank you!



Do you have question? Post your questions on [forum](#) or you can ping at [gitter chat](#)

Technology

MyController.org is Java based web application server.

- Web Server : [TJWS](#)
- REST-API : [Jboss RestEasy](#)
- Database : [h2 database](#)
- MQTT Broker : [andsel/moquette](#)
- Front-end : [AngularJS](#), [PatternFly](#)

Installation

System Requirements

MyController.org server is very lightweight, It required very less resource,

- Disk : 50 MB (may require more space, when we store metrics data for long time)
- RAM : 100 MB
- Java SE : 1.8 or later



Test done up to 10 nodes and 30 sensors with the above configuration._

Supported Platforms

We can run it in any platform which supports Java. So far it has been tested in the following platforms, Kindly share your success stories on other platforms we can add it here.

- Linux
- Windows
- Raspberry PI (Oracle Java recommend)

Download

Executable download is available in two formats, zip and tar.gz. You can download suitable compressed flavor of yours.

- [mycontroller-dist-standalone-0.0.3.Alpha2-bundle.zip](#)
- [mycontroller-dist-standalone-0.0.3.Alpha2-bundle.tar.gz](#)

Kindly visit [releases page](#) of MyController.org to get latest version.

Configuration

Extract downloaded bundle where exactly do you want to run. Configuration file is located in **mycontroller/conf**

File name: **mycontroller.properties**

Temporary file

```
mcc.tmp.location=tmp/
```

You can change default location and file name. This file used as server temporary location and for the operation such as backup, restore, etc.,

Database Configuration

For supports only for H2DB, We have a [request](#) to support other RDMS databases

```
mcc.db.h2db.location=../conf/mycontroller ①
```

- ① You can change default location and file name. File will be stored with the file extension **.h2.db**. Do not add file extension here.

Database username and password are pre-configured. We cannot change it for now. Username: **mycontroller** and password is **mycontroller**

Web server configuration

```
mcc.web.bind.address=0.0.0.0 ①  
mcc.web.enable.https=true ②  
mcc.web.http.port=8443 ③  
mcc.web.file.location=../www/ ④  
mcc.web.ssl.keystore.file=../conf/keystore.jks ⑤  
mcc.web.ssl.keystore.password=mycontroller ⑤  
mcc.web.ssl.keystore.type=JKS ⑤
```

- ① bind interface address. by default it will bind with all the available interface.
- ② Enable/disable https. Only one protocol supported at a time. true - **https**, false - **http**.
- ③ Port number of **http/https** to access MyController.org server.
- ④ web files location, no need to touch this one.
- ⑤ If **https** is enabled these fields are mandatory.

Default URL: <https://<ip>:8443> (ex: <https://localhost:8443>)



Default username/password: **admin/admin**

Important: Change default **mcc.web.ssl.keystore.file** and **mcc.web.ssl.keystore.password** and **https** protocol is recommended

Logger configuration

Configuration File Name: **logback.xml**

Default log file location: **logs/mycontroller.log**

When to change log level, you can change it for specific package. Supported log levels are

- **TRACE** - It prints all the available logs
- **DEBUG** - It prints only debug, info, warn and error logs
- **INFO** - It prints only info, warn and error logs

- **WARN** - It prints only warn and error logs
- **ERROR** - It prints only error logs

Start/Stop Server

Executable scripts are located under `mycontroller/bin/`

- Linux
 - Start : `./start.sh`
 - Stop : `./stop.sh`
- Windows
 - Start : Double click on `start.bat`
 - Stop : `Ctrl+C`
- Other Platforms
 - Execute from 'mycontroller/bin/'
 - `java -Xms8m -Xmx100m -Dlogback.configurationFile=../conf/logback.xml -Dmc.conf.file=../conf/mycontroller.properties -cp org.mycontroller.standalone.StartApp "../lib/*"`



Assigning `-cp classpath` might vary on platforms, kindly refer java documents to know for your platform

Login

LOG IN TO MYCONTROLLER.ORG

Username

Password

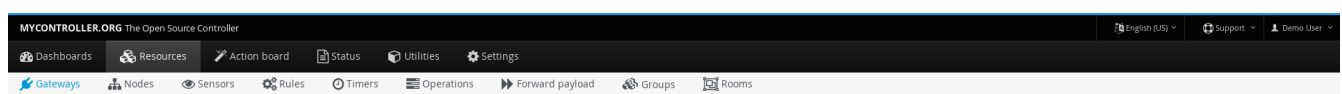
Log In

Welcome to **MyController.org**! The Open Source Controller.
Version: **0.0.3.Alpha2-SNAPSHOT**
Demo account: username: **demo**, password: **demo**

Enter valid username and password to log in to MyController.org server.

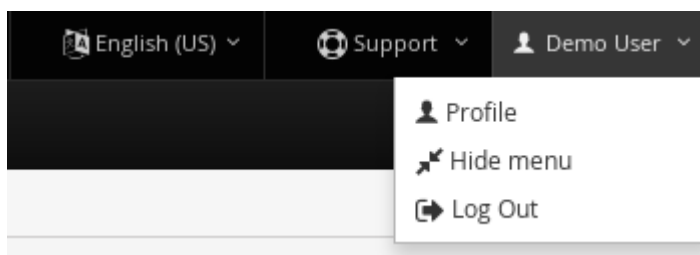
Default username is `admin` and password is `admin`

Menus



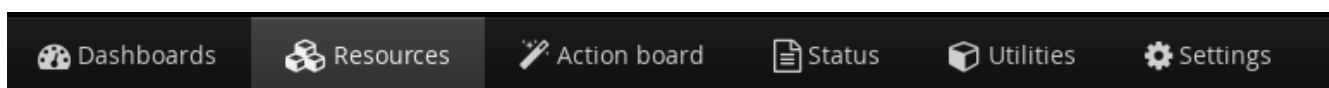
Menu divided in to two parts.

Utility menu



You can perform changing display language(locale), changing password, email of logged in user, logout actions

Main menu

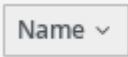




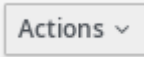


Sub menu

When you click on main menu you get list of sub menus list related to main menu.

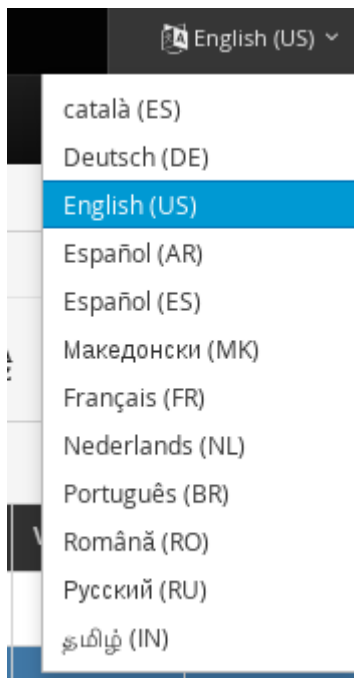
Across pages

You can see the following icons across pages.

-  *Filter by Name* - Filters
 - You change filter with available options.
 - Filter works with AND operation and case sensitive.
-  - Sort
 - you can sort based on fields listed in this menu
 -  - Ascending order
 -  - Descending order
 - To change order click on this icon
-  - View details
 - click this icon to know more about specified item
-  - Actions

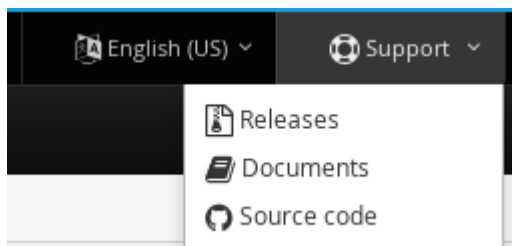
- page to page list of actions will be different
- Select item(s) and perform action with this menu
- Icons
 - ● - Enabled
 - ○ - Disabled
 - ✓ - Up
 - ✗ - Down
 - ? - Unavailable

Language (locale)



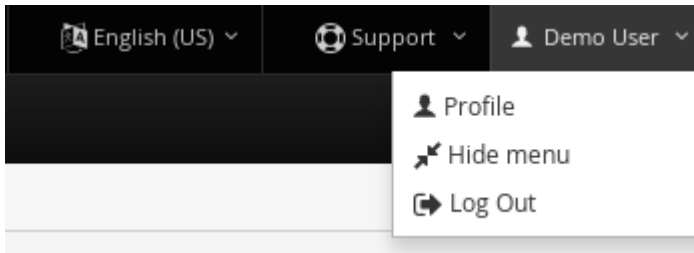
MyController supports multiple locales. You can change to your language by selecting top right corner of language menu.

Support



- **Releases** - take you to MyController.org releases page
- **Documents** - take you to MyController.org documents page
- **Source code** - take you to MyController.org source code page

Utility menu



- **Profile** - take you to profile page. you can change password, email id and name.
- **Log out** - safely log out from MyController.org server

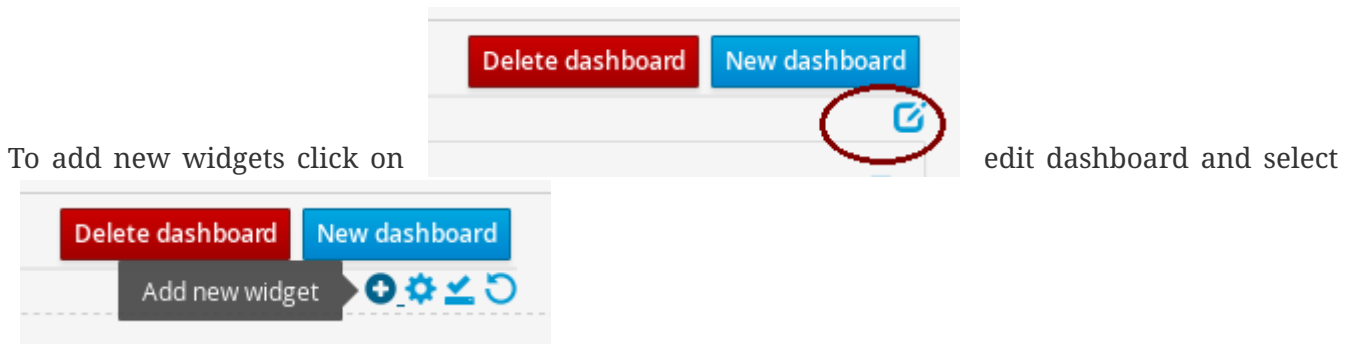
Dashboards

MyController dashboard supports for many number of widgets. If you do not find suitable on for you, you can use custom widget and use scripts for your requirements.

You can keep any number of dashboards. By default maximum dashboard count is set as 5. You can change this settings under **Settings >> System >> MyController**.

To remove a dashboard, select dashboard and click on **Delete dashboard**.

Add new widgets



Type of widgets

- **Sensors**
- **A sensor graphical view**
- **Grouped sensors graph**
- **Mixed sensors graph**
- **Sensors bullet graph**
- **Heatmap chart**
- **Sensors custom button**
- **Display image file**
- **Groups**
- **MyController time**
- **Sunrise and Sunset time**

- News
- Custom widget

Common settings across widgets

- Title - widget title
- Refresh time - refresh widget content, in seconds

Sensors

You can select any number of sensors to be displayed on this widget. You can limit number items per row by setting **Items per row**

Sensors

×

Title

Water tank

Refresh time (seconds)

30

Items per row

1

Sensor variables

Serial-gateway > 3:Water Level Sensor Node > 1:Water level > Volume

Cancel

Apply

A sensor graphical view

When you want to show only one **Sensor variable** graph you can use this widget.

A sensor graphical view
×

Title

Water level last 24 hours

Refresh time (seconds)

30

☐ Enable Min/Max

Time range

Last day

Sensor variable

Serial-gateway > 3:Water Level Sensor Node > 1:Water level > Volume

Cancel

Apply

Grouped sensors graph

You can group similar type of **Sensor variable** with this widget. In this example I have added all the **Temperature** sensor variables.

Grouped sensors graph
×

Title

Grouped sensors graph

Refresh time (seconds)

30

☒ Use interactive guide line

Time range

Last hour

Sensor variable type

Temperature

Sensor variable

Serial-gateway > 2:Outsidenode > 1:s-temp > Temperature, Serial-gateway > 34:q

Cancel

Apply

Mixed sensors graph

You can mix two types of sensors with this widget. One will be shown in **Y1** axis and another will be in **Y2** axis

Mixed sensors graph

Title

Mixed sensors graph

Refresh time (seconds)

30

☐ Use interactive guide line

Interpolate type

Linear

Time range

Last hour

Sensor variable type

Temperature, Humidity

Sensor variable

Serial-gateway > 2:Outsidenode > 1:s-temp > Temperature, Serial-gateway > 2:Ou

Cancel

Apply



There is an open [issue](#) on this widget

Sensors bullet graph

Bullet graph used to watch current status, last value and average value.

Sensors bullet graph

Title

Sensors bullet graph

Refresh time (seconds)

30

Color

↑

#3f9c35

↓

#c00000

↔

Time range

Last hour

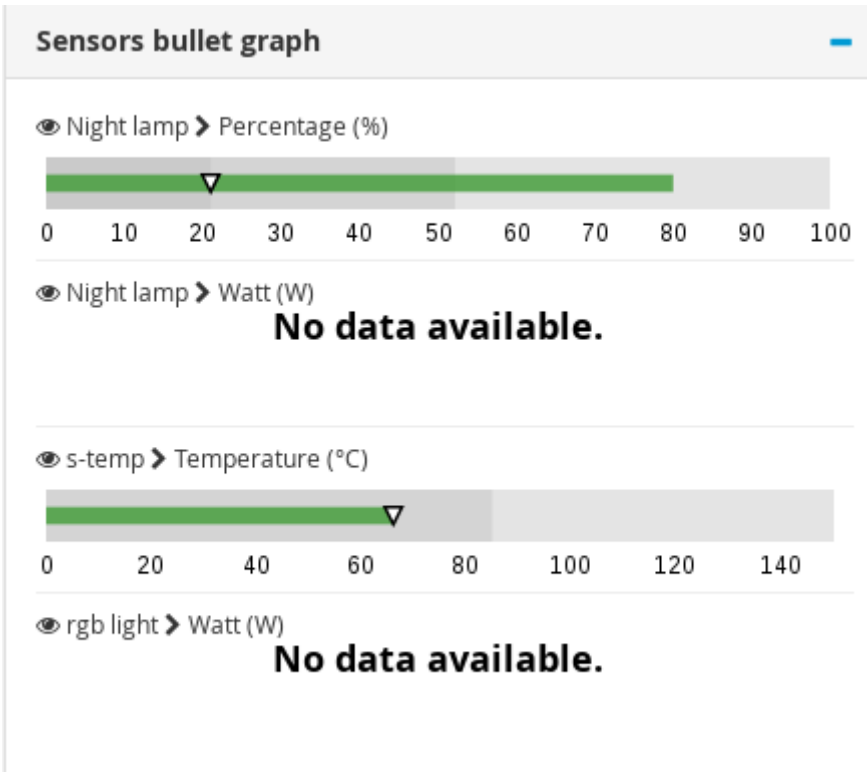
Sensor variables

Serial-gateway > 1:node-1 > 2:Night lamp > Percentage, Serial-gateway > 1:node-1 > 2:Night lamp > Percentage

Cancel

Apply

Displays like this,



Heatmap chart

Heatmap chart used to show resources status in single with with different colors.

Heatmap chart

×

Title

Heatmap chart

Refresh time (seconds)

30

Height

200

Maximum block size

50

☒ Show legends

Data type

Node status

Color

↔

Down

Unavailable

Up

Nodes

Serial-gateway > 2:Outsidenode, Serial-gateway > 155:node-testg, ether2 > 1:N

Cancel

Apply

Sensors custom button

Sensor custom buttons

×

Title

TV remote

Refresh time (seconds)

30

Sensor variable

Serial-gateway > 5:Node in hall > 21:IR sensor > IR send

Button settings:

Minimum height

30

Minimum width

120

JSON

```
[
  {
    "name": "<i class='fa fa-power-off'></i>",
    "payload": "P",
    "btnType": "danger"
  },
  {
    "name": "<i class='fa fa-play'></i>",
    "payload": "PL",
    "btnType": "success"
  },
  {
```

Cancel

Apply

Buttons detail should be in **JSON** format. To create **json** configuration error free way use <http://www.jsoneditoronline.org/> We can include button names as HTML or icons(recommended: font awesome icons)

In configuration **name** and **payload** are mandatory fields. **btnType** is optional and can be any one of **default**, **primary**, **success**, **info**, **warning**, **danger**

Example **json** as follows,

```
[
  {
    "name": "<i class='fa fa-power-off'></i>",
    "payload": "P",
    "btnType": "danger"
  },
  {
```



```

{
  "name": "<i class='fa fa-play'></i>",
  "payload": "PL",
  "btnType": "success"
},
{
  "name": "<i class='fa fa-pause'></i>",
  "payload": "PA",
  "btnType": "warning"
},
{
  "name": "1",
  "payload": "1"
},
{
  "name": "2",
  "payload": "2"
},
{
  "name": "3",
  "payload": "3"
},
{
  "name": "4",
  "payload": "4"
},
{
  "name": "5",
  "payload": "5"
},
{
  "name": "6",
  "payload": "6"
},
{
  "name": "7",
  "payload": "7"
},
{
  "name": "8",
  "payload": "8"
},
{
  "name": "9",
  "payload": "9"
},
{
  "name": "TXT",
  "payload": "TXT",
  "btnType": "primary"
},

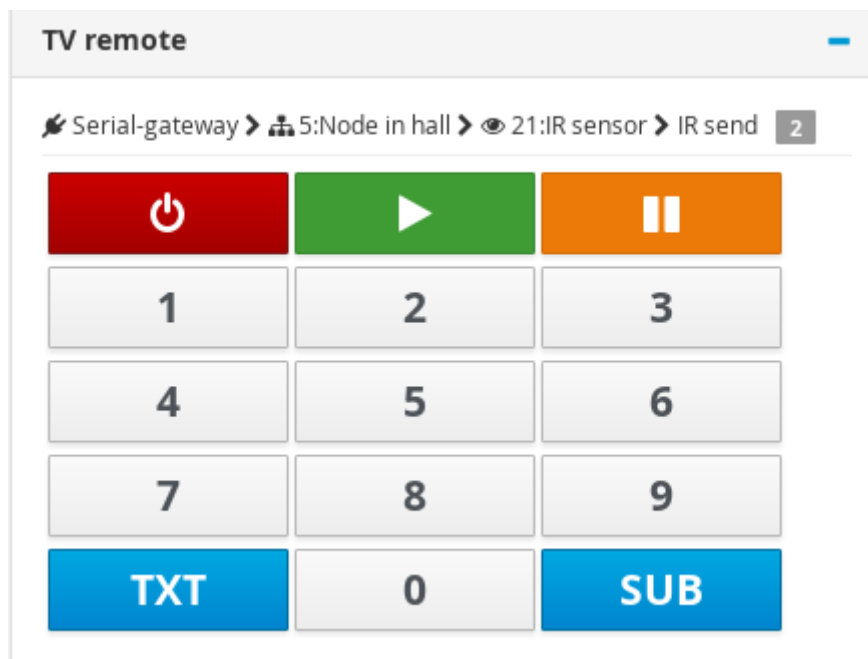
```

```

{
  "name": "0",
  "payload": "0"
},
{
  "name": "SUB",
  "payload": "SUB",
  "btnType": "primary"
}
]

```

Example produces,



Display image file

When you want to display image file from an **url** for from local **disk** you can use this widget. This can be used when you are updating image file with some other services and want to show in MyController dashboard.

Display image file

Title

Display image file

Refresh time (seconds)

30

Type

url

URL

<https://www.google.co.in/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ah>

Cancel

Apply

Groups

When you want to control you groups from dashboard, you can use this widget.

Groups

Title

Groups





Refresh time (seconds)

30

Items per row

1


Groups

 morning,  Sunrise group (scene),  test,  1 Floor

Cancel

Apply


Groups



morning

turn on this group on morning


ON



Sunrise group (sce...

Turn ON this group when sunrises and...


ON



test

-

ON



1 Floor

-

ON

MyController time

Displays MyController time where MyController server is running.

This server time


10:26:25 PM

Jul 29, 2016, IST (+0530)


Sunrise and Sunset time

Displays **Sunrise** and **Sunset** time based on location settings


Sunrise and sunset time



06:05 AM



06:43 PM



11.2333, 78.1667

News

Add **rss** feed and displays news from the feed.

General Discussion

- [Graphical widgets loading forever](#)
- [Restore backup on new mycontroller instance](#)
- [Inclusion mode not supported by this controller](#)
- [Relay Node](#)

Custom widgets

With the custom widget you can add on your interest. You have to write script for your custom widget.

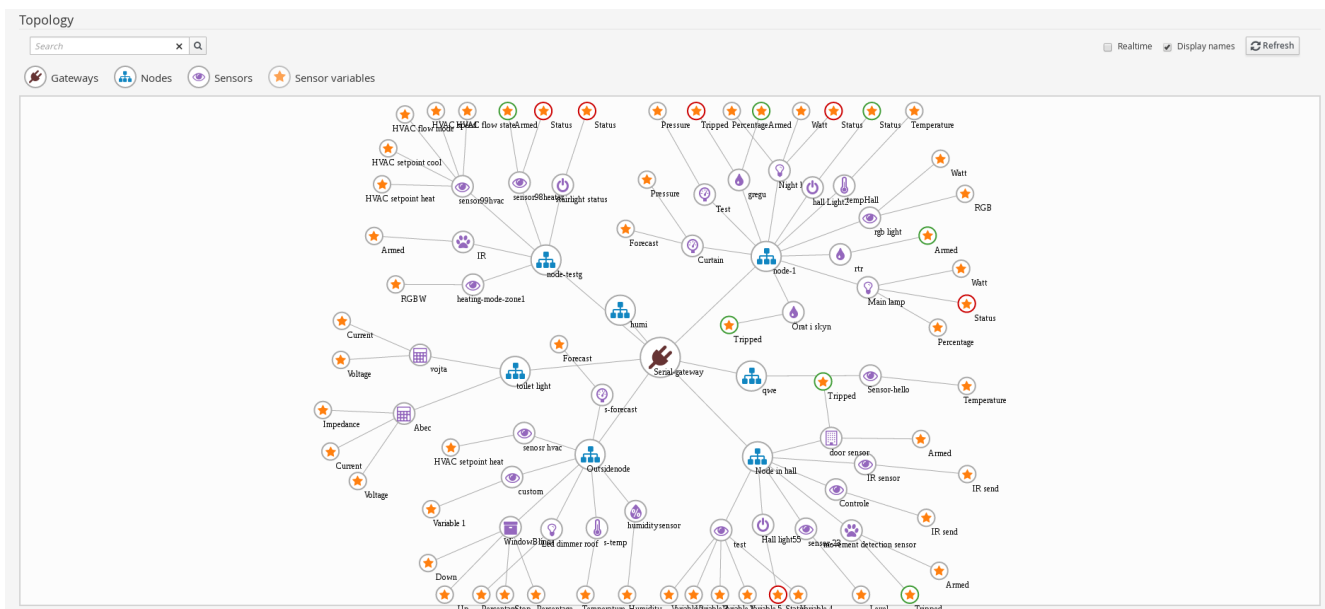
Rooms detail

When you mapped your sensors with room. You can access your room and get your sensors easily. You can perform actions also on sensor.

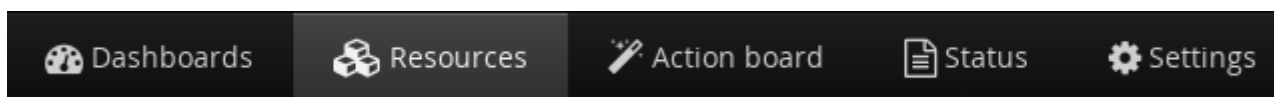
Topology

In topology graph you can see how the things connected in a quick view.

- **Realtime** - How is connected on real time (for now mysensors only supports) When you uncheck this option, shows how the things connected in MyController database.
- By clicking on top legends, you can show hide type of resource.
- You may notices that circle color for nodes and gateways, If node or gateways is **UP** shows in green, other wise based on status.
- Mouse over to resource, displays status of resource as tooltip
- **Display names** - Check/Uncheck to display/hide resources names
- When you click refresh topology graph gets refreshed (also doing automatic refresh also)
- You can filter resources on graph based on name
- When you double click on resources it will take you to resource details page.

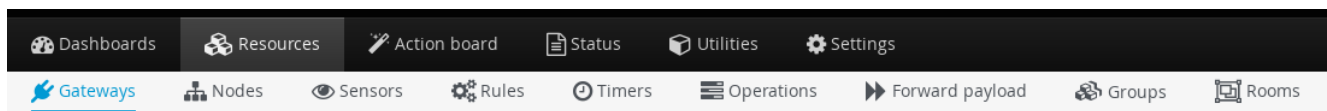


Resources



Under resources you can do almost all actions related to sensors. can do actions on Gateways, Nodes, Sensors, Rules, Timers, Operations, Forward payload, Groups, Rooms.

Gateways



You can add remove any number of gateways. Supports 2 type of networks

MySensors

When you choose MySensors network type it supports 3 types

1. Serial gateway
2. Ethernet gateway
3. MQTT gateway

Sparkfun [phant.io]

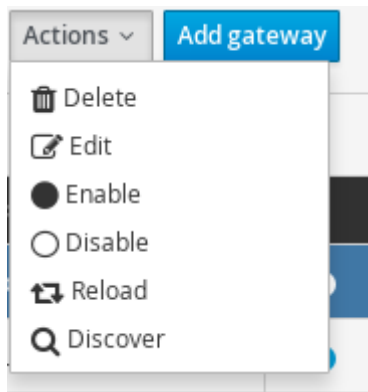
It support REST API. Polls on configured REST API for every N seconds with X limits of data. It will not add duplicate data

Add gateway



Navigate to **Gateways** menu and click **Add gateway**.

Gateways actions



Gateway supports multiple actions. Select an item from the list and choose the desired action.

- **Reload** - reload operation performs stop and start of the gateways.



Reload can happen only for enabled gateways.



When you delete gateway. All the resources belongs to the gateways will be removed permanently. Always do backup when you perform this kind of operation.

Discover

Gateways detail

Name Filter by Name

Name Showing 1 to 3 of 3 items

Actions

3 Results

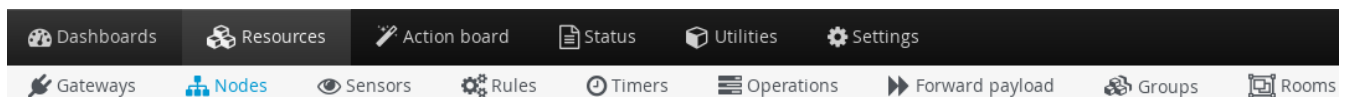
Enabled	Name	Type	Network type	Connection details	Status	Status since	Status message
<input checked="" type="checkbox"/>	Serial-gateway	Serial	MySensors	Port:dev/ttyUSB0, BaudRate:115200, Driver:Auto[SerialComm], RetryFrequency:6000 Seconds(s)		Feb 29, 2016 04:06:14	ERROR: Unrat
<input type="checkbox"/>	ether1	Ethernet	MySensors	Host:192.168.1.2, Port:5003		Feb 19, 2016 05:38:01	Disabled by
<input type="checkbox"/>	ether2	Ethernet	MySensors	Host:192.168.1.3, Port:5003		Feb 19, 2016 05:38:06	Disabled by user

Sends broadcast packet to all the node on this gateway. Will receive parent node details. Used in Topology page to show real time connection



Discover action supports only for MySensors

Node



Under this page you can see node information's like, Node EUI(Node Id), Node Name, Node Type,

Firmware Version, Core lib Version, Battery Level, Assigned firmware, etc.,

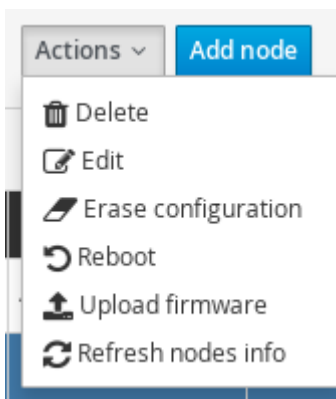


By default **Register nodes automatically** will be enabled under **Settings >> System >> MyController**. If this option is disabled, sensor data of the node will not be added in to MyController until node **Registration status** changed to **Registered**. You can also change a node **Registration status** to one of **New**, **Registered** or **Blocked**

Add manually

Navigate to node page, by clicking  button you can add new node.

Actions

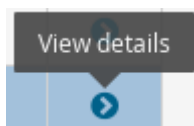


- **Delete** - delete node
- **Erase configuration** - Ask node to erase complete configuration.
- **Reboot** - reboot the node
- **Upload firmware** - Request node to get firmware update
- **Refresh nodes info** - Requests presentation details from selected nodes



When you delete a node. All the resources belongs to the node will be removed permanently. Always do backup when you perform this kind of operation.

Node details

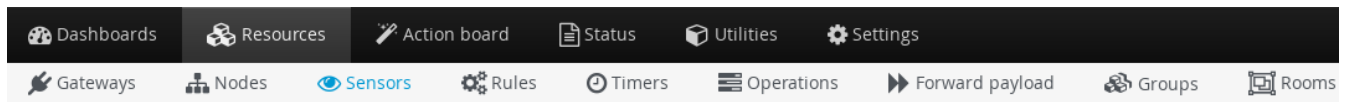


When you click  icon you will complete node details. includes battery usage report.

Mapping Firmware

Before doing this action, firmware should be added in MyController.org. Refer Firmware section to add new firmware. To update selected firmware in to selected node perform **Reboot** or **Update firmware** action.

Sensors



Navigate to sensors Page. In this page you see sensor details like Gateway Id, Node EUI, Sensor Id, Name, Type, Variable Type, Last seen. You can add, edit and delete sensors from here.

Add sensor

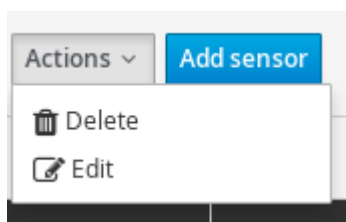
We can add sensors in two methods, via node and manually. If node sends any data related to sensors and if the sensor detail is not available in MyController.org new sensor will added

automatically. To add new sensor manually click on the button



If MyController.org receives any data related to sensor that you have added already will be overwritten.

Actions



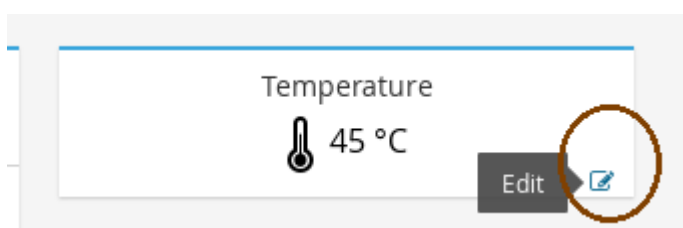
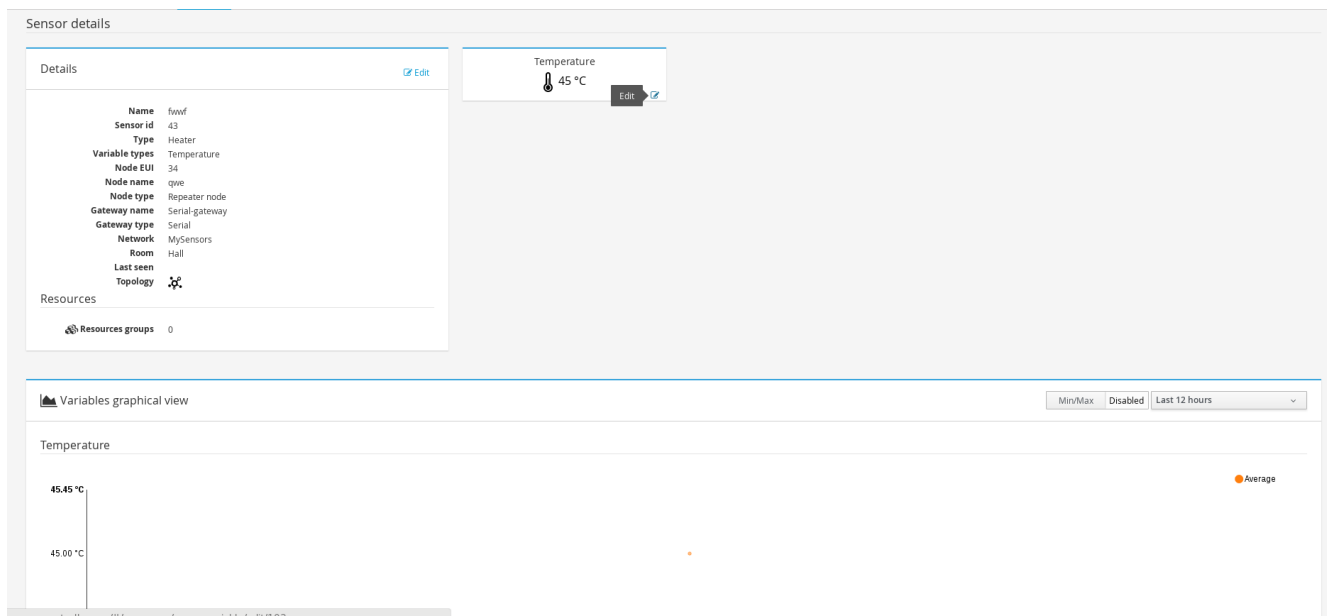
- **Delete** - delete selected sensor
- **Edit** - edit selected sensor



Deletion sensor will delete all the data relevance to that sensor. We cannot recover it back.

Edit Sensor variable

You can change sensor variable type, metric type, read only, graph type, etc., Go to sensors detail page.






Click on edit on **Sensor variable**

You can change the following settings,

- **Read only** - Makes this variable read only, you cannot send payload.
- **Metric type** - You can change metric type
 - **None** - When you do not want to record metric of a variable
 - **Double** - Used for double typed variables, like Temperature, Pressure, etc.,
 - **Binary** - Used for ON and OFF devices
 - **Counter** - Used to record counter type devices, like energy meters
- **Unit** - You can change any type of units. Can select **None** of no units measurement
- **Offset** - Only available for **Double** types metrics. You can enter **+** or **-** values. When MyController.org receives a data from this variable this **offset** will be added with actual value. When sending **offset** will be ignored(offset will not be included).
- **Priority** - This value used to display variables on different locations. When you give lowest number will get higher priority and vice versa.
- **Use global** - When this option set, will use global graph settings. Or you can override with your custom settings.

Update sensor variable

Sensor  Serial-gateway >  34:qwe >  43:Sensor-hello > Temperature

Type Temperature

Read only ☐

Metric type Double

Unit Temperature (°C)

Offset 0

Priority 100

Graph settings

Use global ☐

Line chart

Linear

#ff7f0e

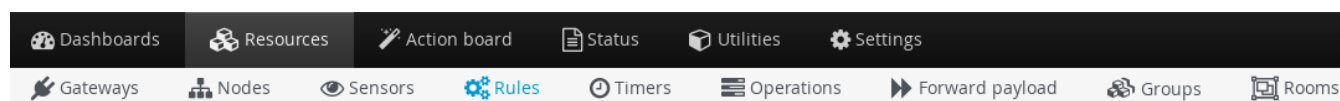
☒ Line ☐ Area ☐ Bar

Cancel

Save

Rules

Formally called **Alarms**



Click on the button  to add new rule.

- **Name** - name of the rule definition
- **Enabled** - enable/disable this rule definition
- **Disable when trigger** - will be disabled automatically when triggers
- **Ignore duplicate** - ignore subsequent triggers. Will trigger again at least condition should not satisfy once.
- **Resource** - Select a resource, Supported resources,
 - **Gateway** - based on gateway status
 - **Node** - based on node status
 - **Sensor variable** - based on sensor variable value
 - **Resources group** - based on resources group status
- **Trigger when** - is a condition.
- **Dampening** - With dampening feature we can control trigger further. Types,
 - **None** - dampening option disabled
 - **Consecutive** - If the condition satisfies continuously for N occurrences.

- **Last N evaluations** - If the condition mets N occurrences in X evaluations.
- **Active time** - If the condition stays active in the specified time.
- **Operations** - We can add any number of operations for a rule definition.

Rules are used to trigger set of **Operations** when a condition met the specification. MyController supports multiple conditions.

Conditions type:

- Compare
- State
- Script
- String
- Threshold
- Threshold range

To add new **Rule**, Navigate to **Resources >> Rules >> Add rule**.

![0_1461341569745_rules-conditions.png](/uploads/files/1461341571011-rules-conditions.png)

Condition - Compare

We can compare **Sensor variable1** with **Sensor variable2**. **Example:** Hall-temperature >= 20% outside-temperature

Condition

Condition type: Compare

Trigger when: <= 15

Condition - State

With this condition we can monitor state of **Binary** type sensor variables, **Node**, **Gateway** and **Resources group**. **Example for Sensor variable**

Condition

Condition type: State

Trigger when: Serial-gateway > 5:Node in hall > 1:movement detection sensor > Tripped == On

Example for Node

Condition

Condition type: State

Trigger when: Serial-gateway > 2:Outsidenode !=> Up

Example for Gateway

Condition

Condition type: State Gateway

Trigger when: ether1 != Up

Example for Resources group

Condition

Condition type: State Resources group

Trigger when: Sunrise group (scene) == On

Condition - Threshold

With this condition we can monitor **Sensor variable** values with a static value or with another **Sensor variable** value

Example with static value:

Condition

Condition type: Threshold Serial-gateway > 2:Outsidenode > 34:humiditysensor > Humidity

Trigger when: <= Value 56

Example with another Sensor variable:

Condition

Condition type: Threshold Serial-gateway > 1:node-1 > 2:Night lamp > Percentage

Trigger when: >= Sensor variable Serial-gateway > 1:node-1 > 44:Test > Percentage

Condition - Threshold range

With this condition we can monitor **Sensor variable** value **Is in range?** or **Is in outside range?**

- **Threshold value low** (1) - low value of threshold
- **Threshold value high** (2) - high value of threshold
- **Include threshold low** - When we check this field included (1) in boundary
- **Include threshold high** - When we check this field included (2) in boundary
- **In range** - When you check this box will be act as **In range** check, otherwise **Out side range** check

Example: X inside [10,56], X outside [45, 200]

Example for In range:

Condition

Condition type: Threshold range Serial-gateway > 2:Outsidenode > 34:humiditysensor > Humidity

Trigger when: ☒ In range ☒ Include threshold low ☒ Include threshold high 60 90

Example for Out side range:

Condition

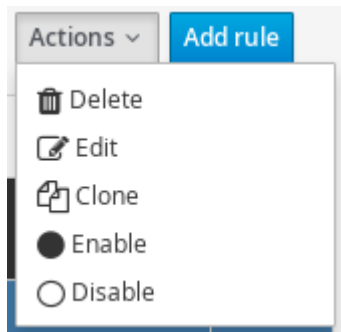
Condition type: Threshold range

Serial-gateway > 2:Outsidenode > 34:humiditysensor > Humidity

Trigger when: ☒ In range ☒ Include threshold low ☒ Include threshold high

60 90

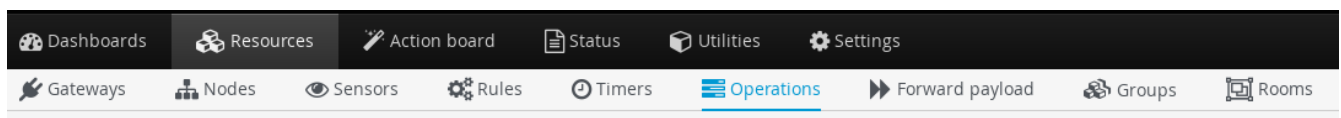
Actions



- **Enable** - enable selected items
- **Disable** - disable selected items
- **Delete** - delete selected item
- **Clone** - clone the selected rule and create duplicate

Operations

Formally called **Notifications**



You can define any number of operations here. These operations will be mapped with **Rules** and **Timers**.



When a operation is in **Disabled** state, Even it's configured active **Rules** or **Timers** will no trigger disabled operations.

Add operation

To add new operation click on the button

Add operation

- **Name** - name of the notification
- **Enabled** - enable/disable notification.
- **Public access** - allow to use other users
- **Type** - type of operations
 - **Send payload** - Do resource operations.

- **Request payload** - Do resource operations.
- **Send SMS** - send SMS
- **Send email** - send email
- **Pushbullet note** - send pushbullet note
- **Execute script** - Executes mentioned script

Send payload

- Select resource and update payload
- **delay time in seconds** update this field with delay time. Using this option payload will send to the resource after x seconds of delay. If you do not want delay leave this field as blank.
- on the payload you can use **Special operations**. Refer Special operations section for further details.

Request payload

- Select a sensor variable to request current payload



This type supports only for MySensors

Send SMS

For SMS we can use [Plivo](#) or [Twilio](#) vendors. When you create trail account you will get some credit. To use SMS notifications you should configure either Plivo or Twilio SMS API settings under **Settings-→Notifications-→SMS**.

- **Phone numbers** - Give destination numbers with '+' with country code then mobile number without any space. If you want to give more than one number use comma(,)
- **Custom message** - If you leave this field blank, default message will be generated. ===== Send email To send email, SMTP email settings should be configured under "Settings-→ Notifications-→Email"
- **Subject** - subject of this email. Mandatory field. You can apply **keys** for this filed.
- **Email(s)** - list of email address with comma(,) separated. Email subject as well as body message supports for `key's

Pushbullet note

To send pushbullet note, Pushbullet settings should be configured under "Settings-→ Notifications-→Pushbullet"

- **Devices** - comma(,) separated device **iden**. If you leave blank will send to all the devices.
- **Title** - Title of the pushbullet note. You can apply **keys** for this filed.
- **Custom message** - You can add your own custom message. If you leave this field blank default message will be taken. You can apply **keys** for this filed.



SMS, Email, Pushbullet note supports for custom messages with keys. `key's will be replaced with actual value on runtime.'

Supported keys on custom messages

You can apply the following keys for this filed.

- `notification.ruleName` - Rule definition name
- `notification.ruleCondition` - Rule condition details
- `notification.actualValue` - Actual value of the resource at the time of trigger
- `notification.triggeredAt` - Rule trigger time
- `notification.operationName` - Operation name

Custom message example: Inside temperature goes too high! Current temperature is `${notification.actualValue}`

Special operations

While defining payload you can assign following special operations, All the special operation reads last received/sent value from target sensor and doing this operation on top of that value and sends to target sensor.

Toggle

By assigning the value: `Toggle` You can select this operation. It is doing toggle operation. This will be useful for 'BINARY' devices. For example if switch is `ON` it will be turned `OFF` vice versa.

Increment

By assigning the value: `++` You can select this operation. Adding 1 with the value. Example: last rx/tx value is 45, on resulting this operation will send 46 to target sensor.

Decrement

By assigning the value: `--` You can select this operation. Subtracting 1 with the value. Example: last rx/tx value is 45, on resulting this operation will send 44 to target sensor.

Addition

By assigning the value: `+3` You can select this operation. Here + meant for addition and 3 is the value should add. Example: last rx/tx value is 45, on resulting this operation will send 48 to target sensor.

Subtraction

By assigning the value: `-4` You can select this operation. Here - meant for subtraction and 4 is the value should subtract. Example: last rx/tx value is 45, on resulting this operation will send 41 to target sensor.

Multiplication

By assigning the value: ***2** You can select this operation. Here * meant for multiplication and 2 is the value should multiple. Example: last rx/tx value is 45, on resulting this operation will send 90 to target sensor.

Division

By assigning the value: **/3** You can select this operation. Here / meant for division and 3 is the value should divide by. Example: last rx/tx value is 45, on resulting this operation will send 15 to target sensor.

Modulus

By assigning the value: **%4** You can select this operation. Here % meant for modulus and 3 is the value used for modulus. Example: last rx/tx value is 45, on resulting this operation will send 1 to target sensor.

Start

By assigning the value: **start** You can select this operation. On this operation target resource will be started.

Stop

By assigning the value: **stop** You can select this operation. On this operation target resource will be rebooted.

Reboot

By assigning the value: **reboot** You can select this operation. On this operation target resource will be rebooted.

Reload

By assigning the value: **reload** You can select this operation. On this operation target resource will be rebooted.

Enable

By assigning the value: **enable** You can select this operation. On this operation target resource will be enabled.

Disabled

By assigning the value: **disable** You can select this operation. On this operation target resource will be disabled.

ON

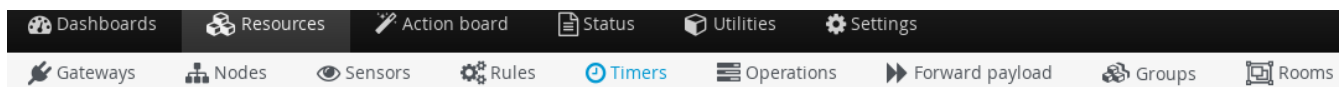
By assigning the value: **ON** You can select this operation. On this operation target resource will be ON.

OFF


By assigning the value: **OFF** You can select this operation. On this operation target resource will be OFF.

Timers

MyController.org using [Sundial](#) scheduler which forked from [Quartz Scheduler](#)

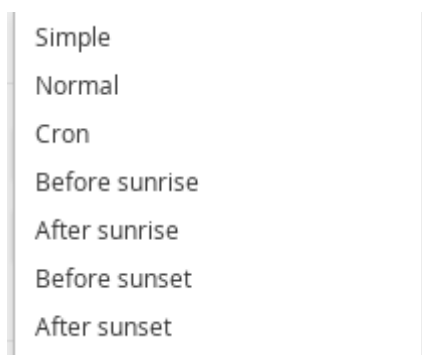


With the timer you can schedule timely operations for resources which configurable in **Operations**. Different types of timers are supported by MyController.org. You can schedule a timer till seconds(via API only, in GUI supports only till minutes only except **cron** type). Means you can

schedule a task for time **21:45:23** like that. By Clicking button  you will be taken to add timer page.

- **Name** - name of the timer
- **Enabled** - enable/disable the timer
- **Timer type** - type of the timer
- **Validity** - validity of the timer. If you leave blank never get expired.
- **Operations** - Select list of operations that you want to run when this timer triggers

Timer types



Simple

Simple timer operates with **Repeat interval** and **Repeat count**. Specify repeat interval in seconds. If you want to run this job count less specify **Repeat count** as **-1**

Normal

Normal is a very basic and classic timer. You can select **Normal** in the type drop down.

Cron

Cron is for advanced users. It is simple and easy. Visit [Quartz-Scheduler](#) page for further detailed configuration



Quartz-Scheduler cron supports from seconds.

Before Sunrise, After Sunrise, Before Sunset and After Sunset

If your task based on Sunrise or Sunset you have to go with this option. Say you want to control your garden light based on your location sun rise and sun set time. You can use this option. You can specify time offset.

Frequency

3 types of frequencies supported by MyController.org

1. Daily – you can select all the 7 days or day(s) only you want to run
2. Weekly – Select a day in week
3. Monthly – Select a day in month

Time

Time format: **HH:mm:ss**

HH – hour in 24 hours format (0~23)

mm – minutes (0~59)

ss – seconds (0~59)



For sunrise and sunset options "Time" will react differently. If you select After Sunrise and After Sunset time offset will be added with "Time" mentioned in task + "Sunrise" or "Sunset" time. If you select "Before Sunrise or Before Sunset" "Time" mentioned in task time offset will be subtracted from actual "Sunrise or Sunset" time.

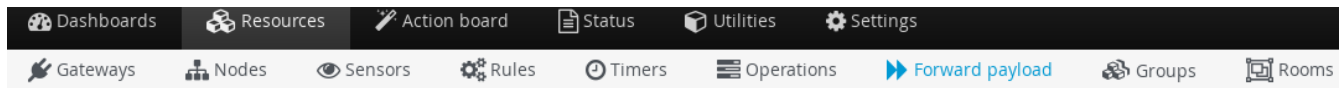
Validity

You may feel you do not want to run this job all the time and want to run only on particular window period. In that case you can select validity. You can select "Validity From" and "Validity To" or only either or nothing. If you do not select any validity that job will be treating like never end. If you select only "Validity From" job will run from that date and never end. If you select only "Validity To" that job will start immediately and will end on the specified date.

Operations

You can select **N** number of operations for a timer. All the operations will be executed when a timer triggers

Forward Payload



You can forward the data received by this sensor to another sensor directly without any condition. This will be useful when you want to send your sensor data to multiple sensors. No need to do any code change on sensor side. Simply add an entry in MyController.org and be happy, MyController.org will take care rest of the things.



This operation supported across gateways too. Which means you can forward data from one network to another network.

Add new entry

To add new entry click on the button

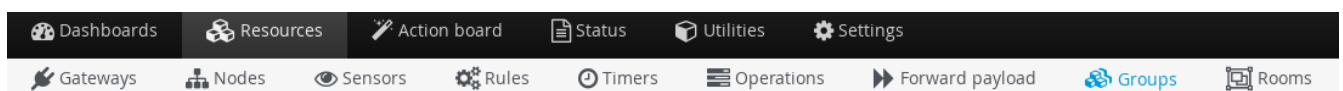
Add an entry

A screenshot of the 'Add forward payload entry' form in the MyController.org interface. The form has a light gray background. It contains three main sections: 'Enabled' with a checked checkbox, 'Source' with a dropdown menu showing 'ether2 > 1:MyNode > 1:aaa > Tripped', and 'Destination' with a dropdown menu showing 'Serial-gateway > 1:node-1 > 2:Night lamp > Status'. At the bottom right, there are 'Cancel' and 'Save' buttons.

Actions


- **Delete** - Delete selected entries
- **Edit** - If there is a change required
- **Disable** - Temporarily you do not want to forward
- **Enable** - Enable disabled entries

Groups



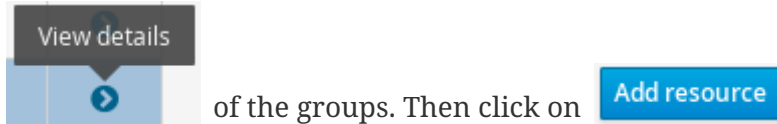
Groups is the place where you want to keep group of resources and perform actions on all resources on single click. Generally it needs **On payload** and **Off payload** for each resource.

Add group

To add new group click on the button  and update **Name** of the group and group **Description**

Add resource

Supports for resource **Gateway**, **Node** and **Sensor variable**. To add resources in to a groups click on



Add an entry

Resource

Resource

Payload

ON

OFF

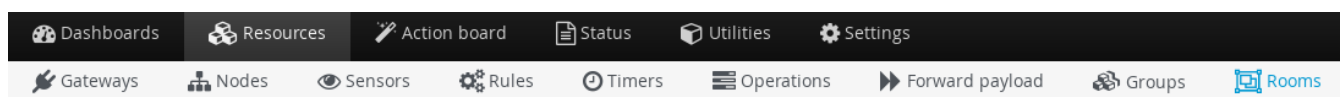
Payloads for Gateway

- **Start**
- **Stop**
- **Enable**
- **Disable**
- **Reload**

Payloads for Node


- **Reboot**

Rooms



Rooms is used to organize your sensors as you see in your home/office. You can create nested rooms also. Example: **Floor1 >> Hall >> TV**

Add room

To add new room click on the button  and update **Name** of the room and room

Description If it is nested room, select parent room. You can select **Sensor** from here as well as from sensors page.



A sensor can be mapped with only one room!

Name

Hall

Description

Sensors located in hall

Parent

Nothing selected

Sensors

Showing all 15

↔

[G]:Serial-gateway >> [N]:5:Node in hall >> [S]:2:Hall light
[G]:Sparkfun-test-2 >> [N]:0IQoo4pEggH6X6O2zyj4:null >
[G]:Sparkfun-test-2 >> [N]:0IQoo4pEggH6X6O2zyj4:null >
[G]:Sparkfun-test-2 >> [N]:0IQoo4pEggH6X6O2zyj4:null >
[G]:Sparkfun-test-2 >> [N]:0IQoo4pEggH6X6O2zyj4:null >
[G]:Sparkfun-test-1 >> [N]:YGYQj85jgDIQylEjg3wj:null >>
[G]:Sparkfun-test-1 >> [N]:YGYQj85jgDIQylEjg3wj:null >>
[G]:Sparkfun-test-1 >> [N]:YGYQj85jgDIQylEjg3wj:null >>
[G]:Sparkfun-test-1 >> [N]:YGYQj85jgDIQylEjg3wj:null >>
[G]:Sparkfun-test-1 >> [N]:YGYQj85jgDIQylEjg3wj:null >>
[G]:Sparkfun-test-1 >> [N]:YGYQj85jgDIQylEjg3wj:null >>
[G]:Sparkfun-test-1 >> [N]:YGYQj85jgDIQylEjg3wj:null >>
[G]:Sparkfun-test-1 >> [N]:YGYQj85jgDIQylEjg3wj:null >>
[G]:Sparkfun-test-1 >> [N]:YGYQj85jgDIQylEjg3wj:null >>
[G]:Sparkfun-test-1 >> [N]:YGYQj85jgDIQylEjg3wj:null >>
[G]:Sparkfun-test-1 >> [N]:YGYQj85jgDIQylEjg3wj:null >>

Showing all 18

↔

[G]:Serial-gateway >> [N]:1:node-1 >> [S]:1:Main lamp
[G]:Serial-gateway >> [N]:1:node-1 >> [S]:2:Night lamp
[G]:Serial-gateway >> [N]:1:node-1 >> [S]:34:rgb light
[G]:Serial-gateway >> [N]:2:Outsidenode >> [S]:33:humid
[G]:Serial-gateway >> [N]:155:node-testg >> [S]:98:senso
[G]:Serial-gateway >> [N]:1:node-1 >> [S]:3:Curtain
[G]:Serial-gateway >> [N]:5:Node in hall >> [S]:1:move
[G]:Serial-gateway >> [N]:1:node-1 >> [S]:121:Örat i skyn
[G]:Serial-gateway >> [N]:5:Node in hall >> [S]:21:IR sens
[G]:ether2 >> [N]:1:MyNode >> [S]:1:aaa
[G]:Serial-gateway >> [N]:1:node-1 >> [S]:222:hall Light2
[G]:ether2 >> [N]:1:MyNode >> [S]:00001:pressure
[G]:Serial-gateway >> [N]:5:Node in hall >> [S]:121:door s
[G]:ether1 >> [N]:122:Node >> [S]:122:son 122

Cancel

Save

Action board

You can perform set of actions here.

Sensors action

Dashboards
Resources
Action board
Status
Utilities
Settings

Sensors action
Send raw message

All available sensors will be listed here. You can perform any action on any sensor. This board can be changed to **List view** (default) or in **Card view**. You can change view by selecting menu on top right corner of the page.

List view

Sensors action

Send raw message

Sensor name

Filter by Sensor name

Last seen

12

51 Results

Sparkfun-test-1

YGYQ85jgDIQyEjg3wj

soilmoisture

Undefined

a few seconds ago

V1: 1017

Sparkfun-test-1

YGYQ85jgDIQyEjg3wj

soiltempf

Undefined

a few seconds ago

V1: -40.0000

Sparkfun-test-1

YGYQ85jgDIQyEjg3wj

winddir

Undefined

a few seconds ago

V1: -1

Sparkfun-test-1

YGYQ85jgDIQyEjg3wj

barotemp

Undefined

a few seconds ago

V1: 27.0000

Sparkfun-test-1

YGYQ85jgDIQyEjg3wj

rainin

Undefined

a few seconds ago

V1: 0.0000

Showing 1 to 5 of 51 items

1

2

3

4

5

6

7

...

11

Card view

Sensors action

Send raw message

Sensor name

Filter by Sensor name

Last seen

12

51 Results

1: heating-mode-zone1

Serial-gateway 155node-testg

RGBW light a month ago

#FF536F

231: tempHall

Serial-gateway 1node-1

Temperature 2 months ago

22 °C

43: Test

Serial-gateway 1node-1

Barometer 2 months ago

12 psi

3: Curtain

Serial-gateway 1node-1

Barometer 2 months ago

15 psi

N/A

20: custom

Serial-gateway 2:Outsidenode

Custom 3 months ago

V1: -

3: s-forecast

Serial-gateway 2:Outsidenode

Barometer 3 months ago

123: vojta

Serial-gateway 25stoleit ligh...

Multimeter 4 months ago

V 234.56 V

I 0.67 A

254: sensor hvac

Serial-gateway 2:Outsidenode

HVAC 5 months ago

34 °C

123: test

Serial-gateway 5Node in hal...

Custom

V1: -

V2: -

V3: -

V4: -

V5: -

00001: pressure

ether2 1MyNode

Barometer

8: sensor-23

Serial-gateway 5Node in hal...

Air quality

Level: -

55: Abec

Serial-gateway 25stoleit ligh...

Multimeter

V - V

I - A

Ohm - Ω

Showing 37 to 48 of 51 items

1

2

3

4

5

...


Send raw message

Dashboards	Resources	Action board	Status	Utilities	Settings
Sensors action		Send raw message			

You can send any type of message from here to your sensor. Goal of this page to send not supported types by MyController and user can do some manual test on sensors network

Example

Sensors action

 [Send raw message](#)

Send raw message

Gateway

Serial-gateway

Node EUI/
Sensor id

21

34

Type/ Sub type

Set

Variable 2

Acknowledge/
Payload

No ack

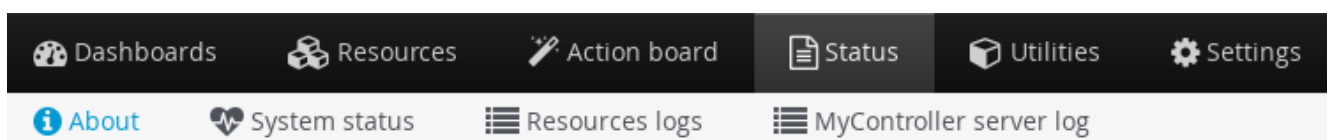
hello

Send

Status

With status pages to can get status about your sensors and MyController. In simple word this is the place to debug the things.

About



About page lists quick details about MyController's configurations.

MyController

Version	0.0.3.Alpha2-SNAPSHOT
Database version	1.02.09 - 2016 Jul 16
Git commit:branch	360026d9877fcd40ba77f478b7fa3e8e52ea31ac:development
Built on	2016-07-25T23:11:37+0530
Build JDK	1.8.0_92
Build tool	Apache Maven 3.3.9


Java virtual machine specification


VM vendor	Oracle Corporation
VM name	Java HotSpot(TM) Client VM
VM version	1.8.0_65-b17
Java home	/home/pi/jdk1.8.0_65/jre


Operating system


Operating system	Linux
Architecture	arm
Version/Kernel	4.4.11-v7+
Location	/home/pi/mycontroller


System status


 Dashboards

 Resources

 Action board

 Status

 Utilities

 Settings


 About System status Resources logs MyController server log

System status page lists bit more about MyController different configurations like,

- Java virtual machine specification
- Operating system
- Script engines

Java virtual machine specification

Lists detail about JVM. You can perform **Run garbage collection** from this page by clicking on

 Run garbage collection

Java home	/home/pi/jdk1.8.0_65/jre
Java vendor url	http://java.oracle.com/
VM name	Java HotSpot(TM) Client VM
VM vendor	Oracle Corporation
VM version	25.65-b01
Specification vendor	Oracle Corporation
Specification version	1.8
Management specification version	1.2
Up time	3 days 13 hours 50 minutes 51 seconds
Heap memory	Used: 47 MB, Committed: 75 MB, Init: 8 MB, Max: 96 MB
Non heap memory	Used: 31 MB, Committed: 34 MB, Init: 0 MB, Max: n/a
Threads count	Current: 39, Peak: 68, Total started: 90409, Daemon: 19
Class loading details	Loaded: 8113, Unloaded: 1583, Total loaded: 9696
Garbage collector - Copy	Collection time: 49499, Collection count: 2934, Last GC duration: 21
Garbage collector - MarkSweepCompact	Collection time: 6288, Collection count: 18, Last GC duration: 539

[Run garbage collection](#)

Operating system

Gives detail about current operating system.

Operating system	Linux
Architecture	arm
Version/Kernel	4.4.11-v7+
Available process	4
System load average	9.00 %
System cpu load	0.00 %
Process cpu	Load: 0.00 %, Time: 9681930 milliseconds
Physical memory (RAM)	Free: 499 MB, Total: 925 MB
Swap space	Free: 99 MB, Total: 99 MB
Committed virtual memory size	242 MB
User working directory	/home/pi/mycontroller/bin

Script engines

Gives detail about supported script engines. By default following script engines are shipped with MyController,

- **Oracle Nashorn** - Supports to execute **Java scripts**
- **Groovy Scripting Engine** - Supports to execute **Groovy** scripts
- **freemarker** - It is template engine used across MyController

You can add any number of script engines. If it supports for **JSR 223** specifications.

1) Groovy Scripting Engine

Name	Groovy Scripting Engine
Version	2.0
Language	Groovy
Language version	2.4.6
Extensions	groovy
Alias	groovy, Groovy

2) freemarker

Name	freemarker
Version	2.3.20
Language	freemarker
Language version	2.3.20
Extensions	ftl
Alias	freemarker, Freemarker, FreeMarker

3) Oracle Nashorn

Name	Oracle Nashorn
Version	1.8.0_92
Language	ECMAScript
Language version	ECMA - 262 Edition 5.1
Extensions	js
Alias	nashorn, Nashorn, js, JS, JavaScript, javascript, ECMAScript, ecmascript

4) jython

Name	jython
Version	2.7.0
Language	python
Language version	2.7
Extensions	py
Alias	python, jython

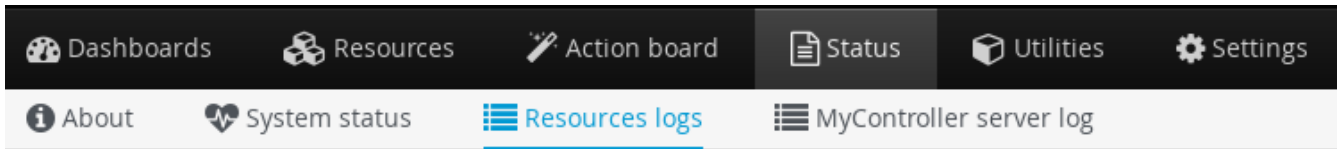
Steps to add new script engine

Download jar file and place it under `mycontroller/lib/` and restart MyController services.

For example if you want to add `Python` support on MyController, you need to add `Jython` library.

- Download `jython-standalone-*.jar` from [Jython](#) website.
- Place it under `mycontroller/lib/`
- Stop MyContorller
- Start MyController

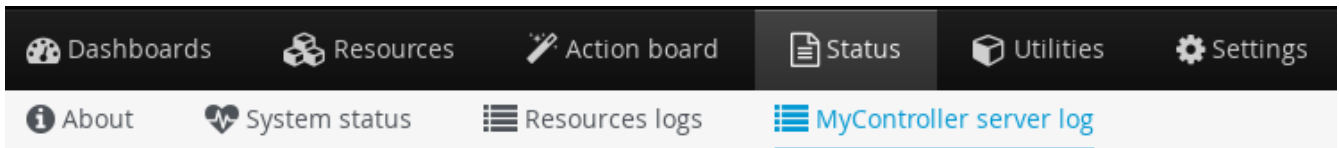
Resources logs



In this page you can understand what's going on about your sensors network. It is a good place to do debugging about your sensors.

By clicking **Purge** button on this page, you can remove all existing logs or selected logs with filter.

MyController server log

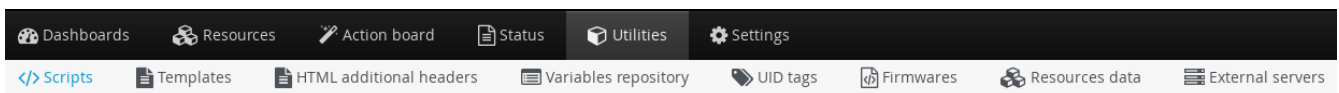


In this page you can watch backend log file `mycontroller/log/mycontroller.log` from GUI.

Utilities

Under utilities you can store scripts, templates, additional headers, variables repository, UID tags, firmwares, external servers details.

Scripts



MyController supports for custom scripts, they can be used to perform any kind of operation or in **Rule**. MyController gives set of API for your needs. With this API's you can query/set data in MyController.

Two types of scripts are available,

- Condition
- Operation

Condition

Condition scripts are used for **Rule** engine. If you are not satisfied with pre defined rule conditions you can write your own script. Your script should return either **true** or **false** or inside script set **mcResult** variable as boolean value.



In python return **true** or **false** will not work. set your result in **mcResult** variable.

Operation

Operation scripts are used for other than **Rule** engine condition type. Can be used on **Operation**, **Dashboard**, etc.,

Script examples

Script's API: [javadoc](#) API's are added in script environment. You can access any supported API's with mcApi object.

Java script

Take an example I want to display last seen of node details in dashboard.

```
var myImports = new JavaImporter(java.io, java.lang, java.util); ①

with(myImports) { ②
    var options = new HashMap(); ③
    //Sort by lastSeen.
    options.put("orderBy", "lastSeen"); ④
    //Order by descending
    options.put("order", "desc"); ⑤
    //Page limit, only 5 result
    options.put("pageLimit", new Long(5)); ⑥
    //Get nodes data
    var nodes = mcApi.node().getAll(options); ⑦
}
```

- ① Import required packages for our coding. **JavaImporter** is used to import packages. You can import any number of packages with comma separated.
- ② Add our imports **with** loop
- ③ Create a **HashMap** to add our query filter values.
- ④ Adding a filter **orderBy** as **lastSeen**
- ⑤ Adding another filter **order** as **desc**
- ⑥ Adding another filter **pageLimit** as 5
- ⑦ Store queried results in the variable **nodes**, can be used in templates(refer templates section)
mcApi already binded with script engine, you can use it to query/set values from/to MyController.

Groovy script

I want to print serial gateways on MyController log file.

```
import org.mycontroller.standalone.gateway.GatewayUtils.GATEWAY_TYPE ①
def filters = [name: 'Serial-gateway', pageLimit: 10L, type: GATEWAY_TYPE.SERIAL] ②
def queryResponse = mcApi.gateway().getAll(filters) ③
mcApi.logger().info("Query Response:{", queryResponse) ④
```

- ① Import `GATEWAY_TYPE` enum, will be used to filter
- ② Create filter list with filter values, Filtering with gateway `name` contains `Serial-gateway` (case sensitive), `type` as `GATEWAY_TYPE.SERIAL` and limiting count on request as 10 `10L` with the key `pageLimit`
- ③ Calling MyController API (`mcApi.gateway().getAll(filters)`) with our filter`
- ④ Printing result under `mycontroller/log/mycontroller.log` as `INFO` log.



When log level set at `ERROR`, `INFO` logs will not be printed!

Python script


In this script we are getting gateways detail and printing in log file.

```
from java.util import HashMap ①
from java.lang import Long ②
options = {'pageLimit': Long(20)} ③
gateways = mcApi.gateway().getAll(HashMap(options)) ④
mcApi.logger().info("Quesry Response:{", gateways) ⑤
```

- ① Import `HashMap` should be used to send filter queries
- ② Import `Long` used to send java `Long` type
- ③ Create filter with `pageLimit` of `20`
- ④ Query MyController with `McApi` and store result in `gateways`
- ⑤ Print result in `mycontroller/log/mycontroller.log` file.

Add script

To add new script click on  Enter `Name` of the script, `Extension` type, script type(`Type`) and script content. You can select script from files also.

Once you have added script you may want to test, is it working as expected. To test script Select your script and click on  `Run now` from actions list. You will get a page to feed bindings options as displayed here,

Run now
✕

Script	operations/get-temperature.js
Script bindings	<input style="width: 90%; border: 1px solid #ccc;" type="text" value="{}"/>

✕ Cancel
▶ Run

You can feed `script bindings` and click on `Run Result(json format)` will be displayed on the same

page as follows,

Run now ✕

Script	operations/get-temperature.js
Script bindings	<input data-bbox="368 333 1291 385" type="text" value='{"myKey": "hello"}'/>

Result

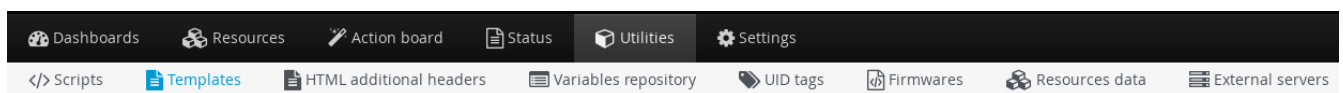
```
{
  "myKey": "hello",
  "uuid": "007027a5-7e21-456d-a50d-7f7d909e8c56",
  "temperatureSen": {
    "id": 35,
    "uid": "temperature",
    "sensorVariable": {
      "id": 7,
      "sensor": {
        "id": 4,
        "sensorId": "1",
        "type": "S_TEMP",
        "name": "s-temp",
        "lastSeen": 1469783363528,
        "node": {
          "id": 2,
          "eui": "2",
          "gatewayTable": {
```

✕ Cancel ▶ Run



script bindings should be in **JSON** format.

Templates



Templates are used to send email and in dashboard for custom widgets. MyController uses [freemarker](#) as template engine. Refer the [docs](#) for the complete format supports.

Syntax example

This template used to display nodes status on dashboard. To run this template you have to select example scripts that queries node status.

```

<table class="table table-hover table-bordered table-striped mc-table">
  <thead>
    <th>Status</th>
    <th>EUI</th>
    <th>Name</th>
    <th>Type</th>
    <th>Battery level</th>
    <th>Last seen</th>
  </thead>
  <tbody>
    <#list nodes.data as item> ❶
      <tr>
        <td class="text-center">
          <#if item.state == "UNAVAILABLE"> ❷
            <i class="pficon pficon-help text-color-gray fa-lg"></i>
          <#elseif item.state == "DOWN">
            <i class="pficon pficon-error-circle-o fa-lg"></i>
          <#elseif item.state == "UP">
            <i class="pficon pficon-ok fa-lg"></i>
          </#if>
        </td>
        <td>${item.eui}</td>
        <td>${item.name}</td>
        <td>${item.type.text}</td> ❸
        <#if item.batteryLevel??>
          <td><span>${(item.batteryLevel)! "-" } %</span></td> ❹
        <#else>
          <td><span>-</span></td>
        </#if>
        <td><span uib-tooltip="{{${(item.lastSeen)!} |
date:mc-helper.cfg.dateFormat:mc-helper.cfg.timezone}}" tooltip-placement="left" am-
time-ago="${(item.lastSeen)! 'Never'}"></span></td>
      </tr>
    </#list>
  </tbody>
</table>


```

- ❶ for loop in template, from script we will get `nodes` object and fetching items one by one.
- ❷ if condition, checks node state (enum type)
- ❸ we can call methods with `.(DOT)` notation. `${item.type.text}` is equalient to `item.getType().getText()`
- ❹ Ignore `null` and assign default value.





To allow `null` you have to add `!` at the end of statement `>> ${item.lastSeen}!`, If you want to put default value on `null` value `>> ${item.lastSeen}! "-"`

Add template

To add new template click on  and save with template **Name** and **Content** as shown in syntax example

Run template


You may want to test your template immediately. For this go to list templates page and select your template and click on  **Run now** you will get a popup. You have select supported script for this template. If script is not required for this template leave this as blank. And add **json** bindings if needed.

Run now 

Template


nodes-last-seen.html


Script

operations/nodes-last-seen.js 


Script bindings

{ }

 Cancel

 Run


When you click **Run** selected script executed with **Script bindings** on back-end and script out put will be passed to template. Final result will be displayed on the pop-up as shown below.

Run now 

Template

nodes-last-seen.html



Script


operations/nodes-last-seen.js 


Script bindings

{ }

Result

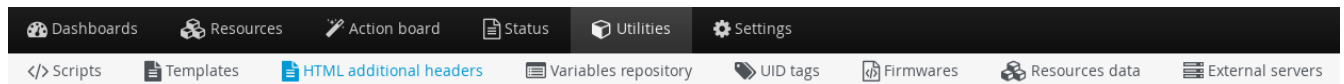
Status	EUI	Name	Type	Battery level	Last seen
	0	Gateway node	Repeater node	-	
	3	Water Level Sensor Node	Repeater node	-	

 Cancel

 Run

if you do not select any script, **Script bindings** directly passed to template engine.

HTML additional headers



When there is situation like you have different widgets available on different sites and you want to use there own **js** or **css** along with MyController, here is the place to add those scripts and style sheets. You can also add your own custom **angular JS controllers**

HTML additional headers

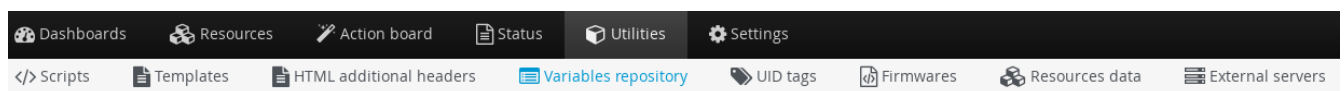
CSS files	<div></div>
Script files	<div>http://bernii.github.io/gauge.js/dist/gauge.min.js</div>
AngularJS custom controllers	<div></div>

Cancel Save



When you do changes in **HTML additional header**, once you saved the changes, to reflect changes on your current browser session do browser refresh **F5**

Variables repository



With the support of variables repository you can keep some configurations here and use it across in scripts and templates.

Each reference has 4 fields, **key**, **value**, **value2** and **value3**.

Add a variable

To add a variable click on

Add variable

 buttons and

Add variable

Key	<input type="text" value="myKey"/>
Value	<input type="text" value="hello"/>
Value #2	<input type="text" value="mycontroller"/>
Value #3	<input type="text" value="awesome"/>

Example(JavaScript):

```
var variable = mcApi.variable().get("myKey"); ①
variable.getKey() ②
variable.getValue() ③
variable.getValue2() ④
variable.getValue3() ⑤
```

① Get variable called **myKey**

② Get the **key** for this variable, must be **myKey**

③ Get **value**

④ Get **value2**

⑤ Get **value3**



There is no field called **value1**. Do not try to call **value1**

UID Tags

[Dashboards](#)
[Resources](#)
[Action board](#)
[Status](#)
[Utilities](#)
[Settings](#)

[</> Scripts](#)
[Templates](#)
[HTML additional headers](#)
[Variables repository](#)
[UID tags](#)
[Firmwares](#)
[Resources data](#)
[External servers](#)

By script calling sensors variables you need lot of info like **gateway** details, **node** details etc., To bypass this we can **tag** a sensor variable with **String UID** and called from your script with this **UID**

Add new entry

To add new entry click on and follow,

Add UID tag entry

UID	<input type="text" value="temperature-out"/>
Sensor variable	<input type="text" value="Serial-gateway > 2:Outsidenode > 1:s-temp > Temperature"/>

In script (Java script)

```
var sensorVariable = mcApi.uidTag().getByUid("temperature-out"); ①  
mcApi.logger().info("Sensor variable Details: {}", sensorVariable); ②
```

① Get **Sensor variable** object with UID **temperature-out**

② Print received object on logger file (**mycontroller/log/mycontroller.log**)

Firmware

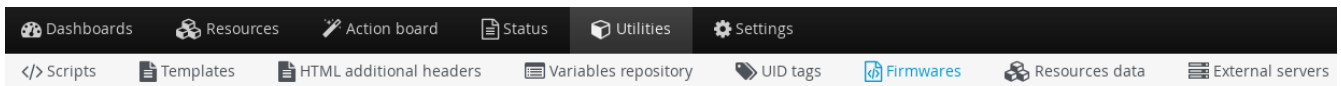
You can control node firmware with the help of MYSBootloader or DualOptiboot-bootloader.



Your node should be running with the bootloader either **MYSBootloader** or **DualOptiboot-bootloader**

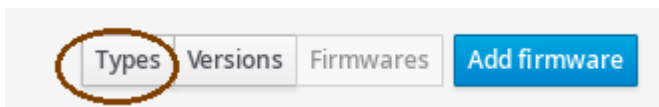
What you can do?

- Upload new firmware
- Change existing firmware to new latest
- Maintain nodes firmware up to date



Each **firmware** is maintainable with **Type** and **Version**. You can keep **N** number of version on each type.

Add Firmware Type



Click on **Type** you will get firmware type page. Clicking **Add firmware type** you can add new firmware type.



Id for the firmware type will be generated automatically. If you leave 'Type Id' as blank.

Add Firmware Version



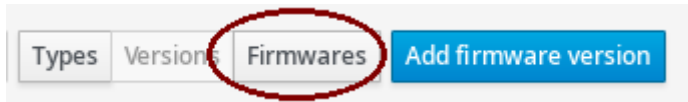
Click on **Version** you will get firmware version page. Clicking on **Add firmware version** you can add new firmware version. While adding firmware version you can leave 'Version Id' as blank. Id will be generated automatically. If you have any requirement for specified id add it.



Id for the firmware version will be generated automatically. If you leave 'Version Id' as blank.

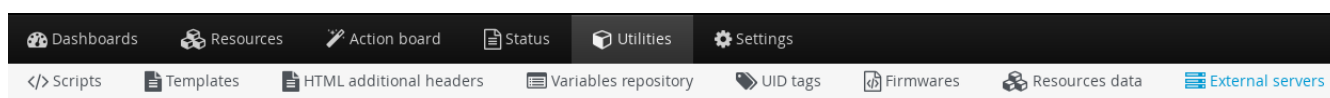
Add Firmware

When you click **Firmwares** on top menu, you will be landed on firmwares page. If you are in firmwares **Type** or **Version** page you will reach by clicking on



Clicking **Add firmware** you can add new firmware. Select Firmware type and version from the pop-up and select .hex from your local computer. Finally click Add. Your firmware is ready for actions.

External servers



When there is need to keep all your sensors metrics on external servers. You have to put an entry here.

Supported external servers

- Emoncms.org
- Influxdb
- Sparkfun [phant.io]

You can add any of the supported server on this page and map your resources with your server. MyController will take care rest of things.

To add external servers click on **Add external server**

Common fields across servers,

- **Name** - Name it your server
- **Type** - select your server type
- **URL** - Your server url
- **Trust host** - select trust host type
- **Key format** - How your resource **key** should be formed, Supported variables **\$nodeName**, **\$nodeEui**, **\$sensorName**, **\$sensorId** and **\$variableType**



When your external server configured with **SSL** certificate, connection will fail. To bypass **SSL** certificate use **Any** as **Trust host**

Emoncms.org server

- **Write API key** - get this key from Emoncms.org server account page.

Name	<input type="text" value="My emoncms.org"/>
Enabled	<input checked="" type="checkbox"/>
Type	<input type="text" value="Emoncms.org"/>
URL	<input type="text" value="https://emoncms.org"/>
Trust host	<input type="text" value="Any"/>
Write API key	<input type="text" value="YG5LEzNZp8i2xbd1dmnb"/>
Key format	<input type="text" value="\$nodeEui_\$sensorId_\$variableType"/>

Influxdb server

- **Username** - username of influxdb
- **Password** - Password of influxdb
- **Database** - Database name of influxdb
- **Tags** - You can assign any number of tags with comma separated.



You can leave **username** and **password** fields as blank if you do not have

Name	<input type="text" value="my influx db server"/>
Enabled	<input checked="" type="checkbox"/>
Type	<input type="text" value="Influxdb"/>
URL	<input type="text" value="https://<host-ip>:8083"/>
Trust host	<input type="text" value="Any"/>
Username	<input type="text" value="influxuser"/>
Password	<input type="text" value="influxpassword"/>
Database	<input type="text" value="mydb"/>
Tags	<input type="text" value="host=mycontroller,location=Zurich"/>
Key format	<input type="text" value="\$nodeEui_\$sensorId_\$variableType"/>

Sparkfun server

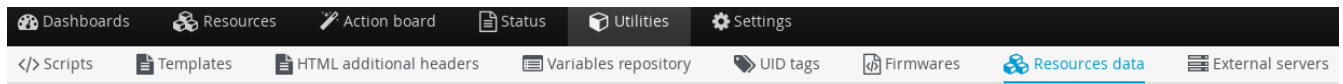
- **Public key** - Public key of your Sparkfun account
- **Private key** - Private key of your Sparkfun account



You have to create Sparkfun account with only one field. If you have more than one field sending will failed. Because MyController sends only one sensor variable data with timestamp. Kindly note **key** is case sensitive in Sparkfun.

Name	<input type="text" value="my sparkfun server"/>
Enabled	<input checked="" type="checkbox"/>
Type	<input type="text" value="Sparkfun [phant.io]"/>
URL	<input type="text" value="https://data.sparkfun.com"/>
Trust host	<input type="text" value="Any"/>
Public key	<input type="text" value="YG5LEzNZp8i2xbd1dmnb"/>
Private key	<input type="text" value="KjylpjWa4MSr3A6rOZ12"/>
Key format	<input type="text" value="\$nodeEui_\$sensorId_\$variableType"/>

Resources data



Once you have created **External servers**, now it's time to configure what are resources data to be sent to external server and to who are all. For all these you have to configure **Resources data**

Supports for 4 types of resources,

- **Gateway** - Sends all the **Sensor variable's** data under this '**Gateway**' to external server(s)
- **Node** - Sends all the **Sensor variable's** data under this '**Node**' to external server(s)
- **Sensor** - Sends all the **Sensor variable's** data under this '**Sensor**' to external server(s)
- **Sensor variable** - Sends only this data to external server(s)

Add resources data

To add new entry click on **Add Resource** and follow as shown bellow,




You can disable sending data at any time. We can do disable in two locations. If you do not want to send data to particular server, do disable on **External servers** page. If you do not want to send particular resource data, do disable on **Resources data** page

Settings

This is the location where you can control various global settings.

Profile

You can change your logged in user **Full name**, **Email**, **Password** here.

 Profile

Username	jeeva
Full name	<input type="text" value="jeeva Kandasamy"/>
Email	<input type="text" value="jeeva123@gmail.com"/>
Current password	<input type="password" value="Current password"/>
New password	<input type="password" value="New password"/>

System


Under system, you can see location settings and MyController settings



Location

You can see **Sunrise** and **Sunset** on **Rule**. Time will be calculated based on this location settings. By clicking **Update** current location will be taken from your browser and will be updated.



You need internet connection to get current location from browser. however to calculate **sunrise** and **sunset** time, internet is not required.

 Location

	06:05 AM
	06:43 PM
Location	<input type="button" value="Update"/>
Latitude	<input type="text" value="11.2333"/>
Longitude	<input type="text" value="78.1667"/>

MyController

Here you can set many settings, which used globally.

☆ MyController

Nodes alive check interval
Minutes

Execute discover interval
Minutes

Units configuration
Metric

Register nodes automatically
☒ Enabled

Resources logs level
Notice

Maximum dashboards allowed

Table rows limit

Global page refresh time
Seconds

Language
English (US)

Time format
12 hours

Grant access to child resources (in roles)
☒ Enabled

Widget image files disk location
../

Login message

This is private controller.

Contact Jeeva (jkandasa at gmail dot com)

Cancel Save

- **Nodes alive check interval** - Every 30 minutes (default) once nodes alive status will be calculated. MyController will send **Ping** to all the nodes. If there is not data for last 30 minutes, node will be marked as down. You can change this interval.
- **Execute discover interval** - Every 30 minutes once this util will be executed. By executing this util real time topology page will be updated.
- **Units configuration** - You have to define here that you want to use **Metric** or **Imperial**
- **Register nodes automatically** - When you disable this button, when new nodes added automatically, **Registration status** will be kept as **New**, data will not be logged from this node until the registration status changed to **Registered**.
- **Resource logs level** - You can define here, what time of sensors logs should be captured.
- **Maximum dashboards allowed** - You can change this settings when you required more than 5 dashboards.
- **Table rows limit** - This settings used across all the tables on MyController
- **Global page refresh time** - Where and all auto page refresh is happening, this interval will be used.
- **Language** - MyController supports many locale. You can set parmenantly your language here.
- **Time format** - You can set MyController should you 12 hours format or 24 hours format
- **Grant access to child resources** - When you create role, if you assign a **Gateway** do you want to control only the **Gateway** or all the resources on the **Gateway**. This is applicable for nodes too.
- **Widget image file disk location** - There is widget on dashboard **display static image**. Local

images are taken from this location. Due to JVM memory issue. There is a restriction on image size. allowed size only up to 7 MB.

- **Login message** - You can sent your own login message. This message will be displayed on login page.



Don't you see you language? Or not fully changed to your locale? Please contribute on [Locale project](#)




Resource logs level is different than MyController logs. To change MyController log level change it on [mycontroller/conf/logback.xml](#)

Notifications

In this page you can set **Email**, **SMS** and **Pushbullet** access settings.

Email

 Email

SMTP host

smtp.gmail.com

SMTP port

465

From address

jkandasa@gmail.com

Enable SSL

Enabled

Use

☒ SSL

☐ STARTTLS

SMTP username

jkandasa@gmail.com

SMTP password

Cancel

Test connection

Save

- **SMTP host** - Enter SMTP host of your emails service provider
- **SMTP port** - port number of email server
- **From address** - When sending email from MyController this address will be added as from address
- **Enable SSL** - If your email server supports for SSL, you should enable this option and select use only **SSL** or **STARTTLS**
- **SMTP username** - username for your email server
- **SMTP password** - password for your email server



Before adding email server settings detail. You can test settings by clicking on **Test connection**. This action will send an email to your **From address**.

Pushbullet

Pushbullet

Access token

mWn

Iden

J8

Active

true

Name

Jeeva Kandasamy

Email

@gmail.com

Cancel

Save

- **Access token** - add access token from your Pushbullet account and click on save. Other settings will be added automatically.

SMS

SMS

Vendor

Plivo

Auth id/sid

WYZ

Auth token

J5

From phone number

+91

97

Cancel

Save

- **Vendor** - Currently supports for **Plivo** and **Twilio**. You can choose either for SMS service.
- **Auth id/sid** - Get from your vendor account details
- **Auth token** - Get from your vendor account details
- **From phone number** - number displayed as from phone number.

MQTT broker

MQTT broker

Status

Enabled

Allow anonymous

No

Bind address

0.0.0.0

HTTP port

1883

Websocket port

7080

Cancel

Save

MQTT broker settings can be changed here. You can do enable or disable MQTT broker without restarting MyController server.

- **Status** - Current status of MQTT broker. You can do enable or disable
- **Allow anonymous** - Allow users without authentication(no user name and password)
- **Bind address** - Broker service should be binded with the IP address. **0.0.0.0** meant bind will all

interfaces

- **HTTP port** - MQTT broker HTTP/TCP port
- **Websocket port** - websocket port of MQTT broker

Metrics

Under metrics settings you can change graph settings and data retention settings

Graphs settings

You can change global graph settings here.

- **Enable Min/Max** - Enable or disable Minimum and Maximum graph settings. If you disable this. shows only Average.
- **Default time range** - default time range in all graphs
- **Battery** - battery graph settings
- **Sensor variables** - set sensor variables graph settings for each type.

Graphs settings

Data retention settings

Cancel

Save

Enable Min/Max

Disabled

Default time range

Last 6 hours

Battery

Line chart

Linear

#ff7f0e

☒ Line ☐ Area ☐ Bar

Individual settings

Armed	Line chart	Step after	#ff7f0e	<input checked="" type="radio"/> Line <input type="radio"/> Area <input type="radio"/> Bar
Current	Line chart	Linear	#ff7f0e	<input checked="" type="radio"/> Line <input type="radio"/> Area <input type="radio"/> Bar
Custom	Line chart	Linear	#ff7f0e	<input checked="" type="radio"/> Line <input type="radio"/> Area <input type="radio"/> Bar
Direction	Line chart	Linear	#ff7f0e	<input checked="" type="radio"/> Line <input type="radio"/> Area <input type="radio"/> Bar
Distance	Line chart	Linear	#ff7f0e	<input checked="" type="radio"/> Line <input type="radio"/> Area <input type="radio"/> Bar
Flow	Line chart	Linear	#ff7f0e	<input checked="" type="radio"/> Line <input type="radio"/> Area <input type="radio"/> Bar
Gust	Line chart	Linear	#ff7f0e	<input checked="" type="radio"/> Line <input type="radio"/> Area <input type="radio"/> Bar
Humidity	Line chart	Linear	#ff7f0e	<input checked="" type="radio"/> Line <input type="radio"/> Area <input type="radio"/> Bar
Impedance	Line chart	Linear	#ff7f0e	<input checked="" type="radio"/> Line <input type="radio"/> Area <input type="radio"/> Bar
KWh	Line chart	Linear	#ff7f0e	<input checked="" type="radio"/> Line <input type="radio"/> Area <input type="radio"/> Bar
Level	Line chart	Linear	#ff7f0e	<input checked="" type="radio"/> Line <input type="radio"/> Area <input type="radio"/> Bar

Data retention settings

This settings tell how long you want to keep data in MyController.

Graphs settings

Data retention settings

Raw data	<input type="text" value="1"/>	Minutes
One minute data	<input type="text" value="6"/>	Hours
Five minutes data	<input type="text" value="2"/>	Days
One hour data	<input type="text" value="30"/>	Days
Six hours data	<input type="text" value="90"/>	Days
Twelve hours data	<input type="text" value="180"/>	Days
One day data	<input type="text" value="5"/>	Years

Cancel

Save

- **Raw data** - how long do you want to keep raw data
- **One minute data** - Every one minute **Raw data** will be calculated
- **Five minutes data** - Five minutes data will be calculated from **One minute data**
- **One hour data** - One hour data will be calculated from **Five minutes data**
- **Six hours data** - calculated from **One hour data**
- **Twelve hours data** - calculated from **Six hours data**
- **One day data** - calculated from **Twelve hours data**



Before do changes in **Data retention settings**, do a MyController backup. If anything you set wrongly can lead data loss. You can avoid this by taking backup.

MySensors

On this page you can set default firmware.

☆ MySensors

Default firmware

Tank water level node:0.0.2

Enable default on no firmware

Disabled

Cancel

Save

- **Default firmware** - select a default firmware, When there is request from MySensors network for firmware. this default firmware will be sent
- **Enable default on no firmware** - When there is request for specified firmware and that firmware is not available in system send default firmware when this option is enabled.

Variable mapper

You can map `Sensor variable's for each sensor types. This can be changed based on your requirement.

Users

You can add/edit/remove users.

Add user

To add new user click on **Add user**.

You can assign roles from here or on roles page.

Enabled ☒

Username

Full name

Email

Password

Roles

Showing all 2

→ →

Super admin role
watertank-user

Showing all 1

← ←

mqtt

Cancel Save

Roles

3 types of roles supported by MyController.

- Super admin
- User
- MQTT user

Super admin

Super admin user can do anything on the controller. Only one super admin role created by controller on installation and you cannot create any new role with **Super admin** power.

Name	Super admin role	
Description	created by system on installation	
Permission	Super admin	
Resources	<div> <div>Users</div> <div>Showing all 1</div> <div> <div>→ →</div> <div>mqtt</div> </div> </div> <div> <div>← ←</div> <div>jeeva</div> </div>	

Cancel Save



Do not delete **Super admin** role. If you delete **Super admin** role, you cannot login.

User

You can add any number of user role(s) with different access. You can assign resources for access when user logged in with this role.

Name	watertank-user	
Description	this user can monitor water tank	
Permission	User	
Resources	<div> <div>Users</div> <div>Gateways</div> <div>Nodes</div> <div>Sensors</div> </div> <div>Showing all 1</div> <div> <div>→ →</div> <div>[G]:Serial-gateway >> [N]:3:Water Level Sensor Node >> [S]:2:Refresh time</div> </div> <div>Showing all 1</div> <div> <div>← ←</div> <div>[G]:Serial-gateway >> [N]:3:Water Level Sensor Node >> [S]:1:Water level</div> </div>	

Cancel Save

MQTT user

You can add any user for **MQTT** authentication. Also you can restrict topics to **subscribe** and **publish**.

Name	mqtt
Description	mqtt
Permission	MQTT user
Topics publish	mgw-out/#, mgw-in/#
Topics subscribe	mgw-out/#, mgw-in/#
Resources	<div> <div>Users</div> <div>Showing all 1</div> <div> <div>→ →</div> <div>jeeva</div> </div> </div> <div> <div></div> <div>Showing all 1</div> <div> <div>← ←</div> <div>mqtt</div> </div> </div>

Cancel
Save

Backup

You can do backup and restore from here. By clicking **Run backup**, backup will be triggered and stored on **backup file location**. can be changed on backup settings. Manual backup always starts with **on-demand_***

Automatic backup

You can set automatic backup on clicking **Settings** on backup page.

Backup settings

Backup location	/home/pi/mc_backup/		
Automatic backup settings			
Status	<div>ON</div>		
Backup file prefix	auto		
Backup every	2	Hours	
Retain maximum #	2		

Cancel
Save

- **Backup location** - location where do you want to keep backup files.
- **Status** - do **ON** or **OFF** automatic backup

- **Backup file prefix** - on automatic backup this prefix will be added on file name.
- **Backup every** - do backup every **Minutes**, **Hours** or **Days**
- **Retain maximum #** - Retain maximum number automatic backups. For example if you set 2, only recent 2 automatic backups only kept.



Retain maximum # settings not applicable for manual backup files.

Restore

To do restore a backup file, select a backup file and click on **Restore** on actions menu. Restore should be triggered immediately.



After a successful restore you have to start the server manually from back-end. You have to check restore status on `mycontroller/log/mycontroller.log`.

Source Code

MyController.org is an Open Source project. You can contribute/download source code from [Github](#) repository

Issue management

Are you facing issue? Wan to file new feature request? Want to give ideas? You are welcome to [Github Issues page](#)

License

[Apache License version 2.0](#)

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensors shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensors regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensors, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensors provide the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Copyright ©2015-2016, All Rights Reserved, <http://www.mycontroller.org>