Switching Monitoring Protection Module

**Introduction**

Switching Monitoring Protection module is a device to switch load & data monitoring. There are 3 data that can be monitored. They are voltage, current and power. In this document we will refer switching monitoring protection module as SMP. SMP implement a REST API as an interface to exchange data thus easily integrate into systems.

**How it works**

When SMP powered up, it attempts to connect into access point with predefined SSID. Once it is connected, SMP will setting up a webserver as a way for other devices to communicate using REST API. There are many features which is provided and each feature has its own endpoint. The usage of each endpoints will be explained later. Once webserver is ready, SMP begin capture and process data from sensor so it can be served.

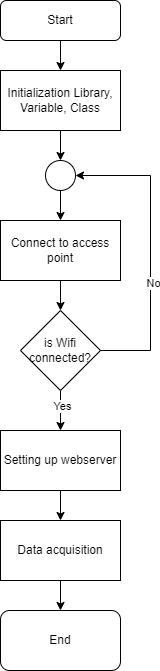


Figure . SMP Flowchart

**REST API**

SMP implements a REST API to exchange data. This will make user could easily integrate it into their own systems. SMP implements a HTTP POST and HTTP GET method, and doesn’t support HTTPS right now, make sure to send a request using HTTP and not the later. The list of each endpoints of POST and GET method are described below

**GET Method**

Note : fill the “.” With ip of device (ex : http://192.168.2.143/get-line-data)

Endpoint usage and explanation for GET Method :

1. get-line-data?line=$,$,$,$,$,$
   * To request multiple line data, append each $ with line number each separated with comma. Support up to 6 lines data per request
   * Send a HTTP GET request with parameter line (ex : 192.168.2.143/get-line-data?line=1,2,3,4)
   * SMP will respond with formatted JSON :

|  |
| --- |
| {  "line-data": [  {  "line": 1,  "voltage": 0,  "current": 0,  "power": 0  },  {  "line": 2,  "voltage": 0,  "current": 0,  "power": 0  },  {  "line": 3,  "voltage": 0,  "current": 0,  "power": 0  },  {  "line": 4,  "voltage": 0,  "current": 0,  "power": 0  }  ]  } |

* “line” : line number, support up to 64 line(s)
* “voltage” : voltage data in millivolt(s) (mV)
* “current” : current data in milliamp(s) (mA)
* “power” : power data in watt(s) (W)

**POST Method**



Note : fill the “.” With ip of device (ex : http://192.168.2.143/relay)

Endpoint usage and explanation for POST Method :

1. relay
   * To control relay on SMP, send a HTTP POST with following JSON format:

|  |
| --- |
| {  "number": 4,  "line": [  1,  6,  7,  12  ],  "value": [  1,  0,  0,  1  ]  } |

* + “number” : represent the quantity of relay that need to be controlled
  + “line” : array that contain the line position
  + “value” : array that contain state of relay. Set 1 to activate relay, set 0 to deactivate relay
  + Example above is a POST request to control 4 relays, which activate relay on line 1 and 12, while deactivate relay on line 6 and 7
  + When POST request successfully received, SMP will respond with JSON format :

|  |
| --- |
| {  "status": 1  } |

* Status 1 indicate that the request is successfully received, any other value indicate abnormality

1. set-addressing
   * To begin addressing on BMS, send the HTTP POST request to SMP with following JSON format :

|  |
| --- |
| {  "addr" : 1  } |

* + Please wait a while, preferebaly around 2 – 3s before sending another request since the SMP need some time to do an addressing
  + The addressing status can be checked on ./get-addressing-status. It lists all the detected device and field “status” represent whether the addressing process is finished(1) or still ongoing(0)
  + When POST request successfully received, SMP will respond with JSON format :

|  |
| --- |
| {  "status": 1  } |

1. set-alarm
   * To set the alarm output (buzzer, relay, etc), send the HTTP POST request to SMP with following JSON format :

|  |
| --- |
| {  "alarm": {  "buzzer": 0,  "power\_relay": 0,  "batt\_relay": 0  }  } |

* + When POST request is successfully received, SMP will respond with JSON format :

|  |
| --- |
| {  "status": 1  } |

1. set-data-collection
   * To begin data capture from BMS, send the HTTP POST request to SMP with following JSON format :

|  |
| --- |
| {  "data\_collection" : 1  } |

* + When POST request is successfully received, SMP will respond with JSON format :

|  |
| --- |
| {  "status": 1  } |

1. set-led
   * To set the led on each BMS, send HTTP POST request to SMP with following JSON format :

|  |
| --- |
| {  "bid": 1,  "ledset": 1,  "num\_of\_led" : 8,  "led\_rgb" : [  [0,0,0],  [0,0,0],  [0,0,0],  [0,0,0],  [0,0,0],  [0,0,0],  [0,0,0],  [0,0,0]  ]  } |

* bid : determine the pack number id
* ledset : 1 to execute, 0 to cancel
* num\_of\_led : number of led in 1 pack
* led\_rgb : array contains the RGB value, each array index represent the position of led (ex : to set the first led into green, set the first array to [0, 255, 0])
  + When POST request is successfully received, SMP will respond with JSON format :

|  |
| --- |
| {  "status": 1  } |

1. set-alarm-parameter
   * To set alarm parameter such as max – min voltage, max – min temperature, send HTTP POST request to SMP with following JSON :

|  |
| --- |
| {  "vcell\_max" : 3600,  "vcell\_min" : 3000,  "temp\_max" : 80000,  "temp\_min" : 20000  } |

* + When POST request is successfully received, SMP will respond with JSON format :

|  |
| --- |
| {  "status": 1  } |

1. set-hardware-alarm
   * To enable / disable internal hardware alarm, send a HTTP POST request to SMP with following JSON format :

|  |
| --- |
| {  "hardware\_alarm" : 0  } |

* + Hardware alarm feature is an internal abnormality checker, if it is enabled SMP will perform a routine check on cell abnormality (voltage and temperature). If an abnormality happens, the SMP will trigger output on buzzer pin. Default value is 0 (disabled) and the value will not be retained when powered off
  + When POST request is successfully received, SMP will respond with JSON format :

|  |
| --- |
| {  "status": 1  } |

1. set-sleep
   * To make CMS sleep, send HTTP POST request to SMP with following JSON format :

|  |
| --- |
| {  "bid": 1,  "shutdown" : 1  } |

* + When POST request is successfully received, SMP will respond with JSON format :

|  |
| --- |
| {  "status": 1  } |

1. set-wakeup
   * To wake CMS, send HTTP POST request to SMP with following JSON format :

|  |
| --- |
| {  "bid": 1,  "wakeup" : 1  } |

* + When POST request is successfully received, SMP will respond with JSON format :

|  |
| --- |
| {  "status": 1  } |

1. restart-cms
   * To restart CMS, send HTTP POST request to SMP with following JSON format :

|  |
| --- |
| {  "bid": 1,  "restart" : 1  } |

* + When POST request is successfully received, SMP will respond with JSON format :

|  |
| --- |
| {  "status": 1  } |

1. Restart
   * To restart SMP, send HTTP POST request to SMP with following JSON format :

|  |
| --- |
| {  "restart" : 1  } |

* + When POST request is successfully received, SMP will respond with JSON format :

|  |
| --- |
| {  "status": 1  } |

1. set-SMP-code
   * To set SMP code, send HTTP POST request to SMP with following JSON format :

|  |
| --- |
| {  "SMP\_code\_write" : 1,  "SMP\_code" : "SMP-000000030"  } |

* + When POST request is successfully received, SMP will respond with JSON format :

|  |
| --- |
| {  "status": 1  } |

1. set-frame
   * To set frame code, send HTTP POST request to SMP with following JSON format :

|  |
| --- |
| {  "bid" : 1,  "frame\_write" : 1,  "frame\_name" : "611800016"  } |

* + When POST request is successfully received, SMP will respond with JSON format :

|  |
| --- |
| {  "status": 1  } |

1. set-cms-code
   * To set cms code, send HTTP POST request to SMP with following JSON format :

|  |
| --- |
| {  "bid" : 1,  "cms\_write" : 1,  "cms\_code" : "0622400013"  } |

* + When POST request is successfully received, SMP will respond with JSON format :

|  |
| --- |
| {  "status": 1  } |

1. set-base-code
   * To set base code, send HTTP POST request to SMP with following JSON format :

|  |
| --- |
| {  "bid" : 1,  "base\_write" : 1,  "base\_code" : "b0622400013"  } |

* + When POST request is successfully received, SMP will respond with JSON format :

|  |
| --- |
| {  "status": 1  } |

1. set-mcu-code
   * To set base code, send HTTP POST request to SMP with following JSON format :

|  |
| --- |
| {  "bid" : 1,  "mcu\_write" : 1,  "mcu\_code" : "m0622400013"  } |

* + When POST request is successfully received, SMP will respond with JSON format :

|  |
| --- |
| {  "status": 1  } |

1. set-site-location
   * To set base code, send HTTP POST request to SMP with following JSON format :

|  |
| --- |
| {  "bid" : 1,  "site\_write" : 1,  "site\_location" : "l0622400013"  } |

* + When POST request is successfully received, SMP will respond with JSON format :

|  |
| --- |
| {  "status": 1  } |