



Airemote Battery Monitor Users' Guide

SUG-SCR-BATMON v4.00 | May 2007

Copyright and Confidentiality

The contents of this paper contain valuable proprietary information and are protected under the copyright laws of the United States and by international treaties. All rights in these materials are reserved. No part of this paper may be copied, photocopied, reproduced, transmitted, translated or reduced to any electronic medium or machine readable form, in whole or in part, without the prior written consent of the author.

Copyright © 2007 **Applied Innovation Inc.** All rights reserved. Duplication in whole or part is not permitted without express written permission.

Copyright and Confidentiality 2

1. Introduction..... 4

2. Script Installation..... 5

3. Script Properties Description 6

4. Configuring Required Properties 9

5. Starting the Script 11

6. Script Alarms 11

1. Introduction

This is the application note for the Battery Monitoring Algorithm, designed to operate on the AIremote. The algorithm was developed using the Telcordia Battery Specification, specifically using the Coup de Fouet, which is the name given to the transient response of a cell (or battery) when it is taken off the float and used to supply the load. It also states that the Coup de Fouet voltage at the five hour rate should not drop below 1.89 volts/cell throughout the life of the battery. A typical response of the voltage response of a single cell battery can be seen below in **Figure 1**.

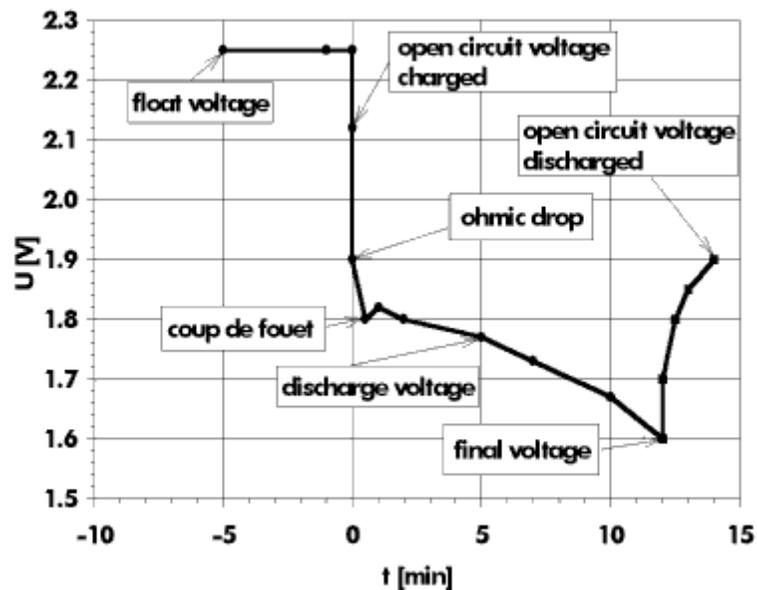


Figure 1

As shown by the figure, there are several areas of concern for us to monitor the battery. First, we need to monitor when the battery is taken off float. Second, we need to find out what the Coup de Fouet voltage is when taken off float in order to meet the specification set above. Finally, we need to determine when the battery is in the linear discharge state, and then approximate the amount of time before the battery would reach its final voltage, End of Discharge (EOD).

To predict the amount of time until the EOD voltage, the script uses slope calculation between two voltage points (which are calculated using windowing functions) to extrapolate the EOD. Thus, in order to quickly predict when we are close to the EOD (the voltage drops will be greater) we need to check the slopes between the points. However, the one drawback is that if there is a lot of noise in the readings, the estimations can vary.

The script follows the curve of the voltage and can report alarms along the way. The script also has several optionally configurable properties for which one can fine tune the script. The alarms and properties are discussed in greater detail in the following sections.

Also, there are four discharge levels for which the script can follow. These discharge rates are 4 hour, 8 hour, 12 hour, and 24 hour.

2. Script Installation

FTP (using binary mode) the Battery Monitor Package (batteryMon-4.0.0.pkg) to the AIremote. Once you have the packages on your AIremote, make sure you are logged into the CLI so you can begin job configuration.

First we need to install the package into the MUSE environment.

```
config pkgs install all
```

Once we have the package installed we need to configure a job for the script to run. The job name specified here is the same name as specified in the Package Manifest.

```
config jobs name batterymon
```

Once the job is configured, we need to tell the job from which package to load the scripts.

```
config jobs name batterymon script batterymon batterymon 4.0.0
```

Enable the script to output data.

```
config jobs name batterymon capture
```

Set the amount of memory the script needs.

```
config jobs name batterymon memory 8192
```

3. Script Properties Description

Now you need to set the job properties for the script. For every property that is deemed ‘required’ in the Package Manifest, you must enter that property with the corresponding value you want it to be instantiated with. If the property is deemed ‘optional’ then it does not have to be instantiated with a value as it has a default value which is not critical for the script’s operation. The generic command format is

```
config jobs name BatteryMon property <name> <value>
```

NOTE: If you do NOT enter a ‘required’ property or enter an incorrect value type, the script will catch the error and NOT begin execution. Also, if the required property is INVALID, the script will catch the error and execution will not start. All property errors are logged in the BM_active.log in the /jobdata directory on the AIremote.

The Battery Monitor script has one **required** property, float_volt. Default values are supplied for the rest, and can be changed while the script is stopped. It is a good idea to get familiar with the following properties and their default values in order to know which values you may need to change for your specific situation.

Battery Monitor Script Properties Table		
Name	Default	Description
float_volt	NONE	Typical input float voltage (REQUIRED)
float_max	57	Voltage at which battery leaves the float state and begins to enter the max state
discharge_rate	8	The approximate discharge rate of the battery. Valid values are 24, 12, 8, 4.
battery_input	power	Battery input to sample. Valid values are <i>power</i> , <i>powerB</i> , 1, 2, 3, 4, 5, 6
eod_volt	42.0	Voltage for the end of discharge for the battery. <i>Must be greater than 0.</i>
cdf_volt	45.36	Voltage for the CdF comparison. <i>Must be greater than 0.</i>
float_hysteresis	0.4	Voltage for allowable flux in the float state. <i>Must be greater than or equal to 0.</i>
discharge_hysteresis	0.4	Voltage for allowable flux in the discharge state. <i>Must be greater than or equal to 0.</i>
recharge_hysteresis	0.2	Voltage for allowable flux in the recharge state. <i>Must be greater than or equal to 0.</i>
enable_estimate	1	Flag for enabling EOD estimation. Valid values are 1 and 0 (for backwards compatibility), <i>enable</i> , and <i>disable</i> . (1 = Enabled, 0 = Disabled)

estimate_start_time	2200	Time at which the EOD estimations will begin. <i>Must be greater than 700.</i>
window_size	10	The size of the window for averaging samples. Valid range is 5 to 100
sample_time	5	Interval for sampling voltage. <i>Must be greater than or equal to 0.</i>
nag_interval	60	Interval for nagging SNMP traps. <i>Must be greater than or equal to 0.</i>
log_level	50	Logging level. Valid values are 200 (INSANE), 100 (VERBOSE), 50 (INFO), 6 (WARNING), 3 (ERROR), 0 (EMERGENCY)
trend	disable	Indicates whether or not a voltage trending file should be created during power events.
snmp	enable	Indicates whether or not SNMP traps should be enabled for the alarm table entries.
t11	disable	Indicates whether or not t11 traps should be enabled for the alarm table entries.
local_display	disable	Indicates whether or not the local display should be enabled for the alarm table entries.
config_mediation	enable	Indicates whether or not the script should configure the alarm table entries and discrete inputs. This must be enabled to run the script in a standalone fashion.

As mentioned earlier, the script supports 4, 8, 12 and 24 hour discharge rates. The script *defaults to the 8 hour discharge rate*. Thus in order to customize the script to a different discharge rate you need to make a few changes via the properties to the script. You need to set the `discharge_rate` and the `estimate_start_time`. The suggested values for these properties are shown below.

4 Hour Discharge Script Properties	
Property	Value
discharge_rate	4
estimate_start_time	1100

8 Hour Discharge Script Properties	
Property	Value
discharge_rate	8
estimate_start_time	2200

12 Hour Discharge Script Properties	
--	--

Property	Value
discharge_rate	12
estimate_start_time	3000

24 Hour Discharge Script Properties	
Property	Value
discharge_rate	24
estimate_start_time	6000

NOTE: If your first several estimations are quite a bit off, chances are that the `estimate_start_time` property needs to be increased. The possible reasons for this could be that your battery discharge rate is not close enough to any of the 4 defaults supported, that the estimations are just beginning too soon, before the linear discharge. Also, note that healthy versus unhealthy batteries will also discharge differently, and that suggested estimated start times supplied above take both unhealthy and healthy batteries into consideration.

4. Configuring Required Properties

In order to start the script, you must configure the `float_volt` property as mentioned above. Most sites will have slightly varying voltages, and the starting point for the algorithm is heavily dependent on this value.

To configure the `float_volt` property, you will need to determine the float voltage that the **AI**remote is reading. To do this, first log into your **AI**remote and at the command line type the command to show all the analog input values.

```
(AI)>show discrete analogs
```

You should see a display similar to the following:

```
(AI)>show discrete analogs
ID           State      Value      Mode      Interval Description
0/1          disable   0.00 mA    current   0
0/2          disable   0.00 mA    current   0
0/3          disable   0.00 mA    current   0
0/4          disable   0.02 mA    current   0
0/5          disable   0.01 mA    current   0
0/6          disable   0.01 mA    current   0
0/power      disable   54.61 V    voltage   0
0/powerB     disable   54.60 V    voltage   0
0/temperature disable   32 C      current   5
```

Take notice of the value for the power analog that you plan to monitor: *power*, *powerB*, *1*, *2*, *3*, *4*, *5*, or *6*. (Note that if you are interested in monitoring points 1-6, you will need to make sure that the analog point has been set to measure voltage rather than current.)

In order to configure the analog point to measure voltage, type the following:

```
(AI)>config discrete analog 0/<point#> voltage
```

Repeat the voltage display command a few times to make sure the reading is stable. Once you have determined what your float voltage is on the **AI**remote, you should enter that value as the script property. The optional `float_max` parameter has a default

value that will suffice for most cases, but you should ensure that it is *at least* a few tenths higher than the float voltage.

```
(AI)>config jobs name batterymonitor properties float_volt 54.45  
(AI)>config jobs name batterymonitor properties float_max 57.5
```

5. Starting the Script

After you have set all of the properties, you need to enter the command to start execution of the job. This is done by:

```
exec-job batterymon run
```

If all commands were entered correctly, you will see the following result on your session, which shows that the job is currently running.

```
Run ID: n
```

Where n is a valid integer.

6. Script Alarms

Once the script has started and has initialized, by default it will add an entry to the Central Alarm Table (If you are not familiar with the Central Alarm Table, please see the **AIremote** User Guide). The entry will be added to the table with the name that you gave the job followed by a 1. In this case the alarm entry will be *batterymon_1*. The entry is added by default with SNMP alarm reporting. You can change the alarm entry to receive the alarms in whatever form you prefer.

The following are the possible alarms that can be reported by the Battery Monitor.

Battery Monitor Status Alarms (<jobname>_1)		
State	Severity	Message
FLOAT	Normal	The battery is in normal operation: <voltage>
MAX	Warning	Over max float voltage: <voltage>
TRANSIENT	Major	The battery is below Float Voltage: <voltage>
TRANSIENT	Critical	Below CdF voltage. Battery needs replaced: <voltage>
TRANSIENT	Critical	Below EOD voltage. Battery needs replaced: <voltage>
DISCHARGE	Minor	CdF found(<value>), moving from transient to discharge state: <voltage>
DISCHARGE	Minor	In discharge state: <voltage>
DISCHARGE	Critical	Estimated time to reach EOD is <minutes> minutes
DISCHARGE	Critical	Voltage below EOD: <voltage>
RECHARGE	Minor	In recharge state: <voltage>
TRANSIENT	Major	CdF not found, moving from transient to discharge state: <voltage> (<i>This can occur in an unhealthy or not fully charged battery</i>)
RECHARGE	Major	Moving from recharge to discharge state: <voltage>
FLOAT	Minor	Moving from recharge to float state: <voltage>

If the 'trend' job property is set to 'enable', a second entry will be added to the central alarm table with its name being the jobname followed by a _2. (For example: *batterymon_2*) This point will be used to convey information about the trending file status, and when it should be obtained. The trending behavior is as follows: The file will begin to record voltage data once the voltage drops below the float voltage. It will continue logging data to this file until the voltage comes back to normal for a period of at least 60 seconds. At this point, the power event has been logged to the voltage history file, and the SNMP battery monitor trending trap will be sent, indicating the name of the voltage history file in the trap. This file will be located in the jobdata/<jobname> directory, and will be available until the next power event completes, which overwrites the previous voltage history file.

The following alarms will be generated when trending has been enabled:

Battery Monitor Trending Alarms (<jobname>_2)	
--	--

Severity	Message
Warning	A voltage history file has been created:<filename>
Normal	CLEARED: A voltage history file has been created
Major	Voltage is very low, get voltage history file:<filename>
Normal	CLEARED: Voltage is very low, get voltage history file