CorsairHMI Battery Monitoring

2/20/2017

Manuals in this series

CHMI_Overview.pdf

CHMI_Operator.pdf

CHMI_Developer.pdf

CHMI_Designer.pdf

CHMI_Experts.pdf

CHMI_Corrections.pdf

CHMI_BatteryMon.pdf – This Manual

CHMI_Glossary.pdf

Contents

Corsair Battery Monitoring

The CorsairHMI program has special features that make it possible to use it as a part of Battery Monitoring Systems. It communicates with BMS (Battery Monitoring System) hardware. A BMS is used to detect problems with a string of battery cells. The Corsair software reads battery data from the BMS

and reports it to an operator. The operator may be at the same site as the batteries or miles away using the Internet.

Any version of the Corsair program can talk to a BMS but the special BMS license option is needed to take advantage of the full range of Corsair battery monitoring features.

Streaming Serial Security

Datacenters are very concerned about internet security. There is a resistance to having internet-based outside monitoring of the battery equipment. One approach is to eliminate real-time operation. The data could be built up on the local Corsair machine and someone could transfer it to an outside firm for analysis using a USB drive. The first problem with this approach is that it requires manual intervention. A larger concern is that it does not provide for real-time alarming. A battery post temperature in a thermal runaway condition will not be discovered in time.

There are two conditions in this situation that allow for another possibility. The first condition is that no information needs to travel from outside on the internet into the battery monitoring system. The second condition is that battery status data does not have extremely high security requirements. A data centers business is not threatened if a hacker finds out the cell voltage of one of its batteries.

A solution for this situation comes from an understanding of the electrical characteristics of some types of computer data communication systems. The first is an older serial communication system called RS-232. An RS-232 port on a computer had a number of pins on a connector. Some systems used pin 2 for 'Transmit Data'. Pin 3 was used for 'Receive Data'. Pin 7 was used as a common connection. There were other pins but they are not important for this discussion. Communications could be accomplished with just pins 2, 3, and 7.

Suppose that there are two computers with identical serial ports. If someone wanted for them to communicate he may hook them up like this:

Transmit Data 2---------- 2 Transmit Data

Receive Data 3---------- 3 Receive Data

Common 7---------- 7 Common

This is a straight-through cable where each pin on one end is connected to the same number pin on the other end. Straight-through cables cannot be used to communicate between identical computers using RS-232. The problem is that a transmit is hooked to a transmit and a receive is hooked to a receive. Data can only be communicated when a transmit is hooked to a receive.

Transmit Data 2---------- 3 Receive Data

Receive Data 3---------- 2 Transmit Data

Common 7---------- 7 Common

This type of crossover cable is called a 'null-modem' cable. It makes the proper connections for data communication between two identical computers. It is needed because the RS-232 Transmit pin only knows how to transmit, it cannot receive. The Receive pin only knows how to receive, it cannot transmit.

Ethernet is a serial communication system like RS-232 except with different wiring and much higher speeds. The pins on modern Ethernet connectors can assume a transmit or a receive function. It is possible to connect a straight-through Ethernet cable between two identical computers and they will communicate. This is because most ports now support 'auto-crossing' where the computers figure out among themselves what wires are used for transmission in one direction and what wires are used for transmission in the other direction.

Any Ethernet system – including so-called 'one-way' routers – replies on electrical data transmission in both directions. Security comes from software and how the equipment on each end is programmed. All software security is vulnerable to new attacks as they occur. Software security is a never-ending battle. It is very difficult to verify that the battle has been won.

With old-fashioned RS-232 serial ports another approach is possible:

Transmit Data 2---------- 3 Receive Data

Receive Data 3 no connection 2 Transmit Data

Common 7---------- 7 Common

Cutting (or not hooking up) the one wire makes it so that data can only be transmitted in one direction. The RS-232 serial port hardware makes sure that data can only flow in one direction. This is hardware security. It can be enhanced with the addition of a 1-way fiber optic link.

Inside Computer Fiber Outside Computer

Transmit Data 2 ------- Fiber Converter ----------Fiber Converter -------- 3 Receive Data

The short piece of fiber may be thought of as an optical isolator. Any sort of electrical noise coming from either computer will not be seen by the other. The only thing connecting them is a piece of fiber with a beam of light going through it. It only goes one way through the fiber. The data only goes one way on the wired pins of the RS-232 connectors.

The 'Inside Computer' is running a copy of the Corsair software. It communicates with one or more pieces of BMS and building automation equipment. It could have the ability operate things on that equipment. It can be used as an operator control station.

The Inside Computer sends information from the equipment through the fiber to the other computer. This information is sent as a continuous stream of data which is why the technique is called 'streaming serial'. The 'Outside Computer' is also running a copy of Corsair software. It is connected to the Internet. It has whatever firewall and security protections the owner desires. The computers may be located within a few feet of each other.

It may be possible for someone to get through the security of the Outside computer and learn the cell voltage of the facility's batteries. Most importantly he cannot get through to the Inside computer to corrupt anything inside the data center or to probe for information other than what is sent from the Inside computer. This security is not dependent on any version of any firewall software. It does not have to be updated whenever new intrusions are found.

The streaming serial system accomplishes isolation of the 'Inside' from the 'Outside'. For this reason, the inside computer could be called the 'Transmit Isolator' and the outside computer the 'Receive Isolator'.

The data format that Corsair sends for streaming serial transmission is available from CorsairHMI in case a company wants to evaluate it.

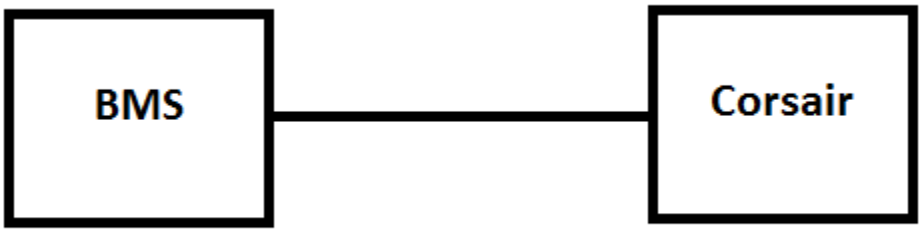
Streaming serial security may be of no value in an isolated small system where there is nothing that can be corrupted from the outside and there is no data that must be kept secret.

Architectures

One of the most important aspects of selecting a battery monitoring interface is choosing the architecture. The architecture specifies the hardware and software that is used and how they are interconnected. The selection process is much more complex than many people realize. It varies greatly with the chosen BMS hardware, network security issues, alarming, cost, and the data storage, analysis, and viewing requirements. Communication speeds can dictate architecture choices. CorsairHMI can be used in a wide variety of architectures. In some cases it is the only software that is needed. In other cases, software must be added for web hosting, for data storage and analysis, or for security considerations.

The following list of possible architectures is only a starting point for a discussion as to the best solution for a customer.

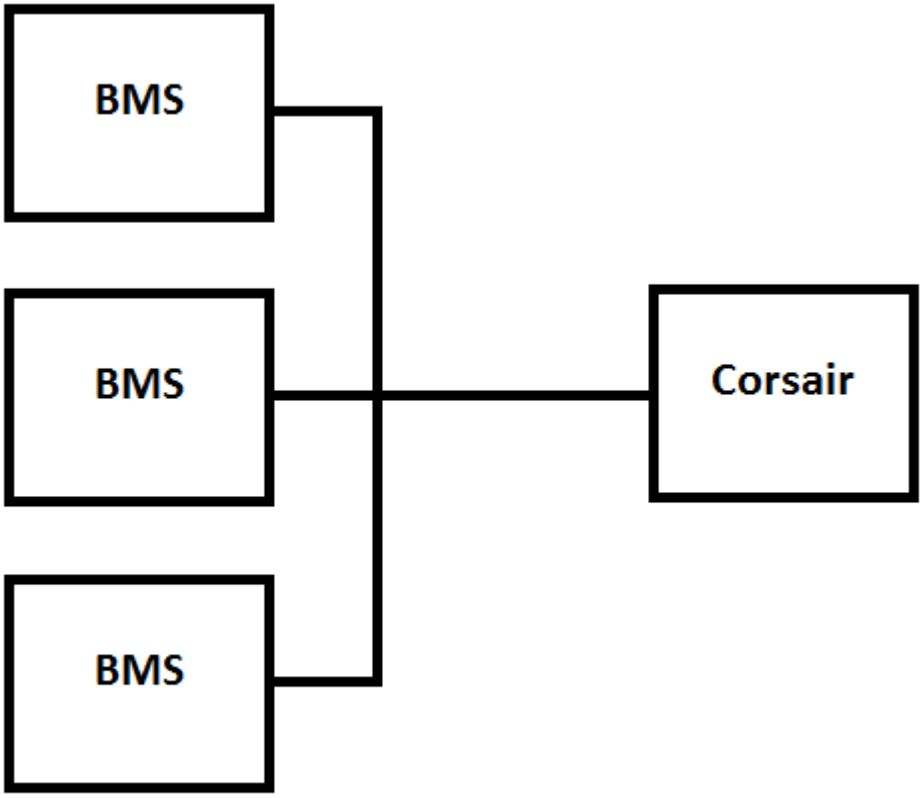
Architecture A Local View



This is the simplest system with one BMS and one Corsair computer. It provides a simple dashboard that may be adequate for small systems. The BMS connection depends on the manufacturer. It may be Ethernet, RS-232 serial, or RS-485 serial. The computer is typically a Windows computer with a monitor, a mouse, and a keyboard.

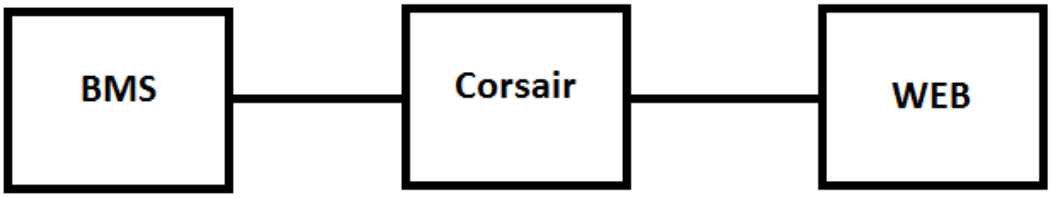
BMS equipment can generate large amounts of data when battery discharge events occur. A CorsairHMI computer can be used for local storage of that data beyond the storage that is provided with the BMS equipment.

Architecture B Multiple BMSs



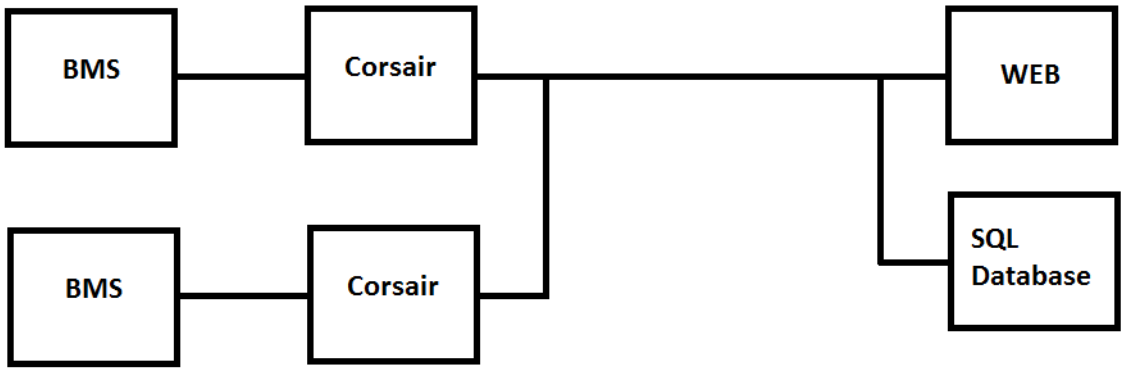
A Corsair computer can usually handle more than 1 BMS. The possibilities depend on the type of communication, the protocols that are involved, and data storage requirements. Serial communication may involve a separate cable from each port on the computer to each BMS or the equipment may be daisy-chained on a multidrop serial line. Ethernet communication is another possibility.

Architecture C Web Access



Adding an Internet connection enables the Corsair computer to serve web pages to operators that are not on the site. Corsair can send email messages or text messages to cell phones for alarm conditions. The local monitor and keyboard can be used at the same time. The web view is not as complete as the local view but it may be adequate for some system.

Architecture D Remote Web Server



In this system a Web Server computer and a database computer work together for long-term data storage and analysis for several BMSs. The Web Server and SQL Database computer may be at a different location than the BMS equipment.

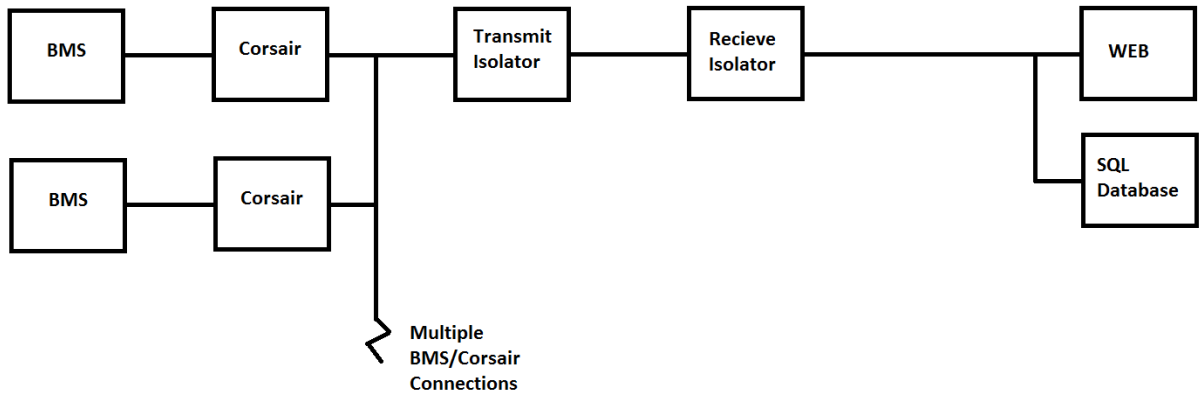
A data center may have installed equipment from several BMS manufacturers. Corsair can serve the important function of normalizing the data from each BMS as it appears to the web server computer. This means that the data from each BMS appears in the same format to the server.

This architecture has the possibility of being the most powerful of all the architectures with the best web interface, the most powerful analytical tools, the largest data storage, and the best ways to compare data across multiple BMSs. It could be the worst architecture if the web server software is not well-written. CorsairHMI does not provide this software but can provide assistance with developing it. Corsair has a partner that already has an excellent tested solution. It features several innovations that maximize real-time performance while minimizing communications traffic and data storage requirements. Contact Corsair for information concerning this partner

Architecture E Remote Web Server with FTP Push

This architecture has the same drawing as D. The difference is that the data is periodically pushed by the Corsair computer to the server using the FTP protocol. The Real-Time performance is not the same as D but it may be more acceptable from a security standpoint. The Corsair partner also has a solution for this approach.

Architecture F Remote Web Server with Streaming Serial Security

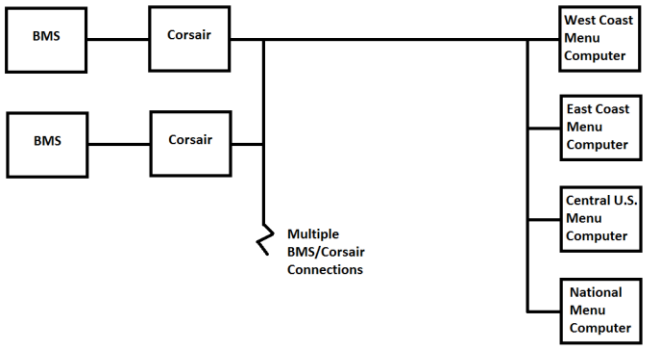


This system uses a Transmit Isolator computer running CorsairHMI software to concentrate data from several BMSs. It streams this data out to a Receive Isolator computer that also runs Corsair. It then sends the data to the web server. The web server software is the same as for architectures D or E.

Architecture G Large Quantity System with Remote Web Server

This architecture is the same as D and E except that is uses an inexpensive ARM processor running Linux hooked to each BMS. This may be for monitoring batteries in cell phone towers or wherever a large quantity of similar sites exists. Monitors, keyboards, and mice may be left off the Corsair computers to save space and costs.

Architecture H Distributed System with Regional and National Menus



This is an expansion of Architecture C. It features a lot of BMS systems scattered across the US. Four instances of the Corsair program serve for menu access to all of the systems. Web users with appropriate credentials can access the West coast, East Coast, and Center region menus. From these

menus they can access any Corsair computer in the region. A national menu can be used to access the regional menus. One computer can be used to run all of the menus. A key difference between this and the D architecture is that the menu computers do not store data. They only supply links to the viewer's web browser.

Architecture I Modbus Repeater

CorsairHMI can read data from a BMS using several different protocols. This data can be repeated from Corsair to another computer using the MBHR system Modbus protocol. A possible use for this is to allow Corsair BMS monitoring at the same time that another computer is using the manufacturers configuration software. This capability has a wide variety of specialized applications.

State Determination

Any battery monitoring system must determine the state of the battery bank – if it is Discharge, Charge, or Float. The determination may be more complex than simply seeing the polarity and magnitude of the battery string current. Most BMS equipment has that manufacturer's version of string state determination.

CorsairHMI has a proprietary method for state determination embedded into the BMS version of the software. Special arrangements may be possible for a technical discussion of this method. One advantage of it is that it brings unity to systems that have multiple brands of BMS equipment.

Using CorsairHMI for state determination depends on communications update rates. It cannot catch a 1 second discharge event if it is only reading BMS data at a 3 second rate. These issues can be a part of determining the architecture.

BMS Data Items

Each BMS measures data items from the battery string. The items that are available vary with the equipment manufacturer. CorsairHMI can deal with any item that can be read through the communications protocol. The BMS version of the Corsair program has special features for dealing with common data items. They include:

Primary String Current – 'PStCu' – per string

Primary String Voltage – 'PStVo' – per string

Secondary String Current – 'SStCu' – per string

Secondary String Voltage – 'SStVo' – per string

Cell Voltage – 'ClVo' – per cell

Ambient Temperature – 'AmbT' – per BMS

Aux Temperatures – ‘AuxT’ – designated with ‘D’

Cell Temperatures – ‘Clt’ – per cell

Specific Gravity – ‘SpecG’ – per cell

String Ripple Current – ‘RiCu’ – per string

String Ripple Voltage – ‘StRiVo’ – per string

Cell Ripple Voltage – ‘ClRiVo’ – per cell

Cell Equalization – ‘ClEQ’ – per cell

Strap Resistance – ‘StrapR’ – per cell

Discrete Inputs – ‘DiscIn’ – designated with ‘D’

Analog Points – ‘AnaIn’ – designated with ‘D’

Electrolyte Level – ‘E_Lev’ – per cell

A data item that is ‘per BMS’ has one element for each BMS. A ‘per string’ data item has one element for each string of battery cells so there may be multiple elements per BMS. A ‘per cell’ data items has one element for each of the cells that are in a string.

Each make and model of BMS will offer different combinations of these data items. Corsair can handle other types of data items but additional specialized development work will be required.

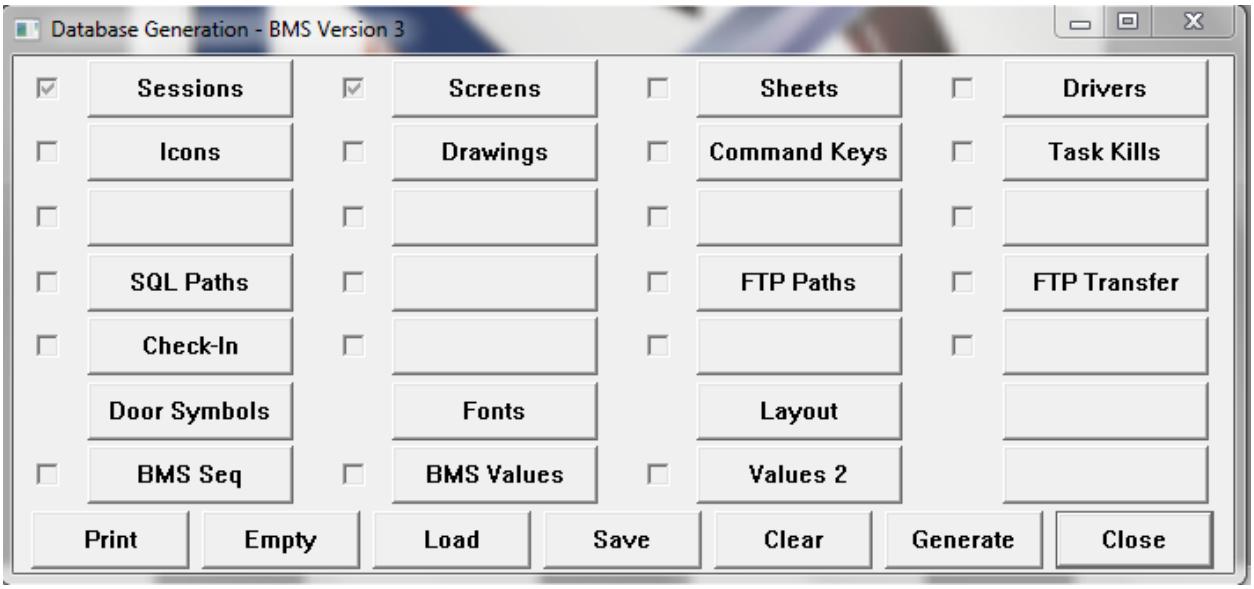
BMS Database Generation

CorsairHMI includes a database generation expert that is used as an aid to developing Corsair database (‘Model’) files. If the system has the BMS option enabled on its license BMS database generation capabilities are available. The generator is not capable of handling all possible systems but generated databases normally require minimal modification before they are used in Battery Monitoring systems.

It is recommended that the developer learns how to use the database generator for non-BMS systems first before trying specialized BMS generation. This information is in the CorsairHMI Experts manual.

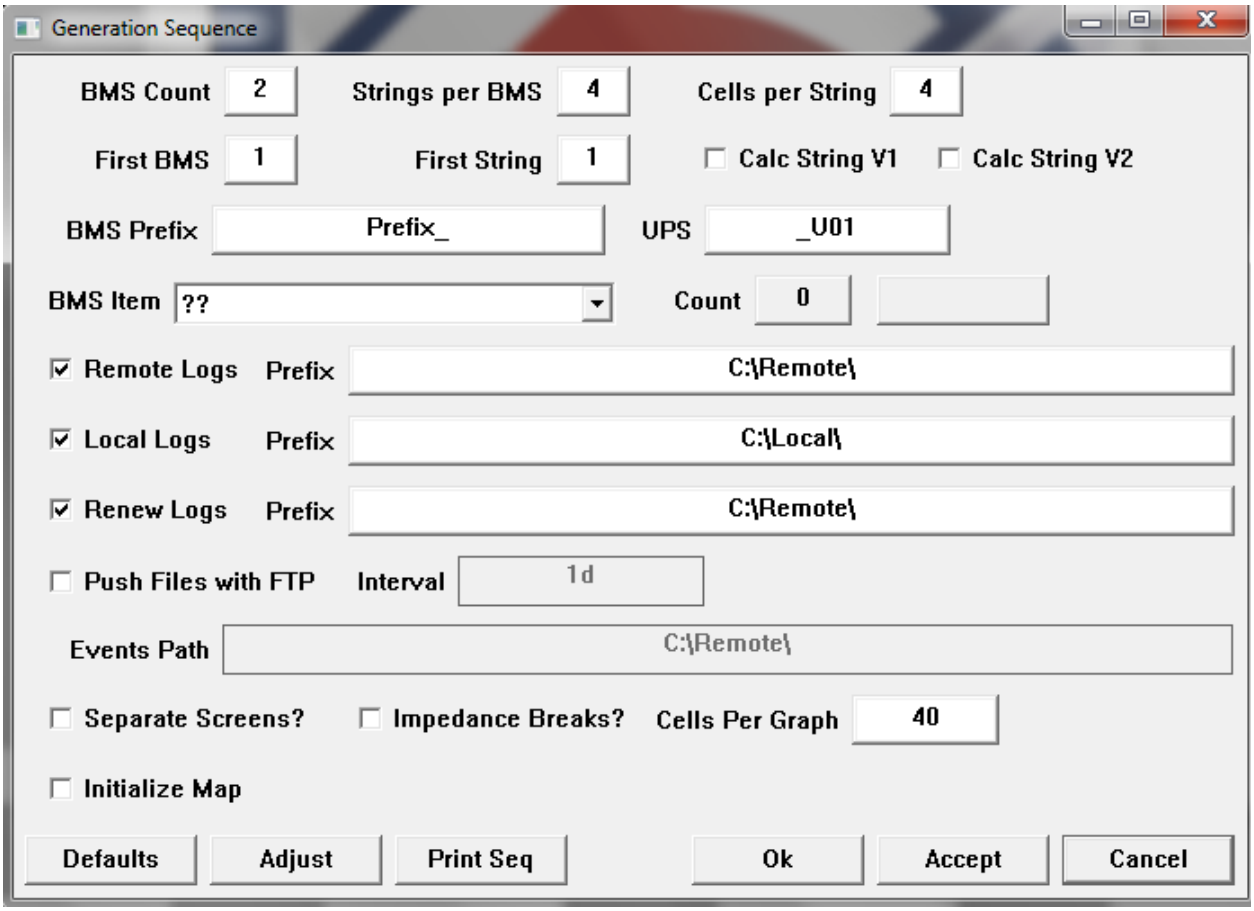
Generator Operation

Database generation is opened from the ‘Database’,’Generation’ option on the main corsair menu.



This window is used to create, load, and save database generation templates. The three buttons for 'BMS Sequence', 'BMS Values', and 'Values 2' are only visible when the license has BMS capabilities.

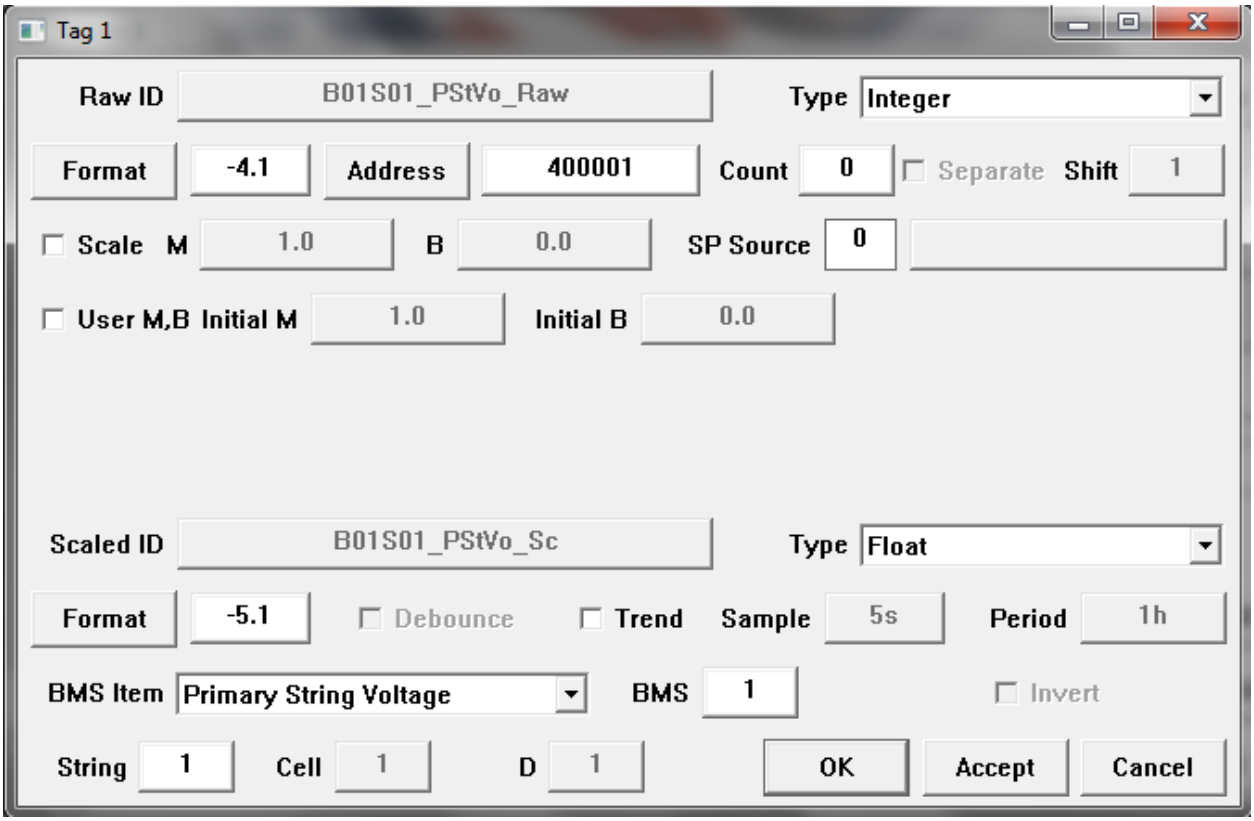
The BMS sequence window is an early step in generating databases for battery monitoring projects.



'Strings Per BMS' is how many battery strings are on each piece of BMS equipment. 'Cells per String' is how many battery modules are hooked in series to create the string. 'Strings Per BMS' times 'Cells per String' gives the total number of cells.

Corsair Tag Creation

Tags for BMS data items are created slightly differently than the tags that are normally created by the generator.

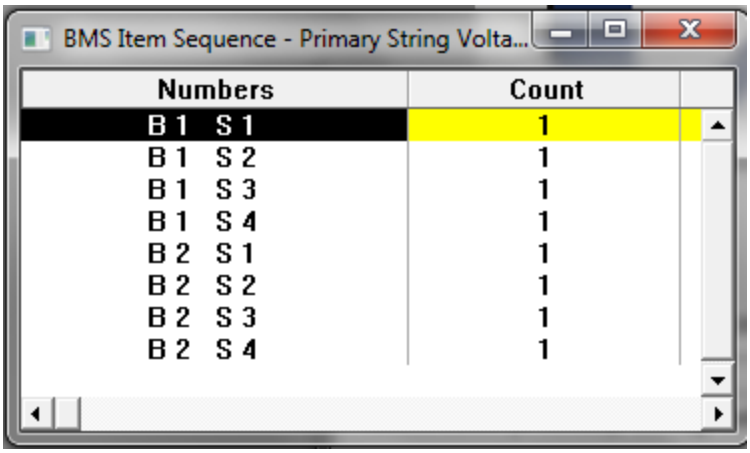


They must be assigned a BMS item and given a starting BMS number. The type of the BMS item determines if the String, Cell, or D number must also be entered.

The scaled and raw ID tags cannot be manually entered for a BMS item. They are generated automatically using the generators tagging convention.

Name Sequencing

The generation sequence window is used to specify how many BMSs are on the system, how many strings per BMS, and how many cells per string. There is an item sequencing window that shows the progression of sequence numbers for each data item. This sequence system is used for tag names within Corsair, for naming csv data files, and for alarm names in event records.



The letter 'B' refers to BMS. The letter 'S' refers to String. Data items that are per cell will have an additional 'C' designation. Other BMS-related items have a 'D' designation.

If the BMS has a sequence value of 1, and there are 4 strings per BMS, and the strings start at 1, a 'Per String' data item will be sequenced with these designations:

B1 S1, B1 S2, B1 S3, B1 S4

If there are 4 cells per string in this system a 'Per Cell' data item will be sequenced with these designations:

B1 S1 C1, B1 S1 C2, B1 S1 C3, B1 S1 C4,

B1 S2, C1, B1 S2 C2, B1 S2 C3, B1 S2 C4,

B1 S3 C1, B1 S3 C2, B1 S3 C3, B1 S3 C4

B1 S4 C1, B1 S4 C2, B1 S4 C3, B1 S4 C4

The count of 'D' items on a BMS is not dependent on the string and cell configuration. They may be sequenced like this:

B1 D1, B1 D2, B1 D3, B1 D4, etc

Generator Naming Convention

The tag names of BMS data items created by the generator are a combination of the type of the item, where it appears in the sequencing convention, and another part for description. B and S designations are followed by a 2-digit number with leading zero padding. C designations are followed by a 3-digit number with leading zero padding.

A tag that begins with 'B02S03_ PStCu_' is related to the primary string current of the third string on a BMS with sequence number 2.

A tag that begins with ‘B01S01C001_Imp_’ is related to the cell impedance of the first cell on the first string on the first BMS.

A tag that begins with ‘B1D1_DiscIn_’ is related to the first discrete input on the first BMS.

CSV Files

When the Corsair interface goes into run mode the generated database will create a number of Comma-Quote separated text files for BMS data. Each line (record) of each file will begin with 6 fields that define the date and time of the data sample.

“1941”,”12”,”7”,”7”,”10”,”23.245”, . . .

The first field is the year. The second is the month expressed as 1 through 12. The third is the day of the month from 1 through 31. The fourth is the hour expressed as 0 through 23. The fifth is the minute from 0 to 59. The sixth is the seconds from 0 to 59. The seconds field may include up to three decimal places to indicate fractional seconds. The generator defaults to using GMT time for the record date and time information. It is not advisable to use the local time zone time for these records.

CSV File Types

The generator can create logs for three different types of files. ‘Remote’ logs generate files that are designed to be transferred to a remote computer for analysis and storage. ‘Local’ logs generate files that are designed to be kept on the hard drive on the Corsair computer. Corsair writes to Remote and Local logs with an append operation that adds each new record to the end of the file. The triggering of Remote and Local logs uses an algorithm that is not documented here.

‘Renew’ logs are written with a replace operation that means that the file will only have one time-stamped line in it at all times. This record is the latest current value of the data items in that file. Renew log files are used to show nearly real-time data to a centralized monitoring system. They are triggered at a fixed rate with a short trigger interval depending on the type of the data.

The file names for each type of file begin with the file name prefix that is entered on the Generation Sequence window. This prefix typically contains a full path specification for where the files are stored on the hard drive of the Corsair computer. The rest of the file name depends on the type of log and the type of the data that it contains. Local and remote log file names are suffixed with ‘.CSV’. Renew log file names are suffixed with ‘R.CSV’.

Local logs use Corsair ‘Monthly Auto’ file naming. The file name has a component like ‘201702’ for a file generated in February of 2017. This is to speed up historical review on the Corsair computer by reducing total file size. Remote logs do not use this sort of naming because the data is typically transferred to some sort of host machine that uses its own file naming conventions. Renew file names are fixed because the file never grows in size.

String Data Logs

The file name for the string data logs consists of the file name prefix followed by the BMS prefix followed by the UPS designation. This is followed by '_S01' for string 1. A unique file is created for each string.

After the date and time the next fields are the Primary String Current, Primary String Voltage, Secondary String Current, Secondary String Voltage, String Ripple Current, and String Ripple Voltage. There are always six fields here. They will be shown as an empty Quote-Quote-Comma set if the data item is not implemented in a system.

Cell Data Logs

To be continued

Temperature Data Logs

To be Continued

SQL Events

The generator creates alarm records that are used to send event data to an SQL database. The database configuration standard for this event logging is described in the Designer manual.

Alarms include:

String Discharge, Charge, and Float State

BMS Communications failure

Ambient temperature high and low

To be continued . . . .