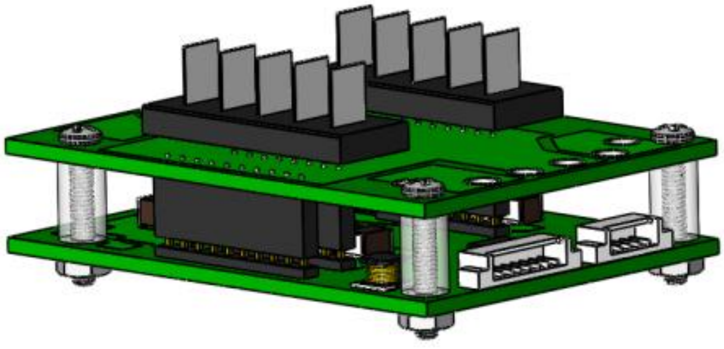
# BC-1022 Intelligent Battery & Power System (IBPS) User Guide



REVISION 1.5C

www.ocean-server.com    Sales@ocean-server.com

# Table of Contents

# 1

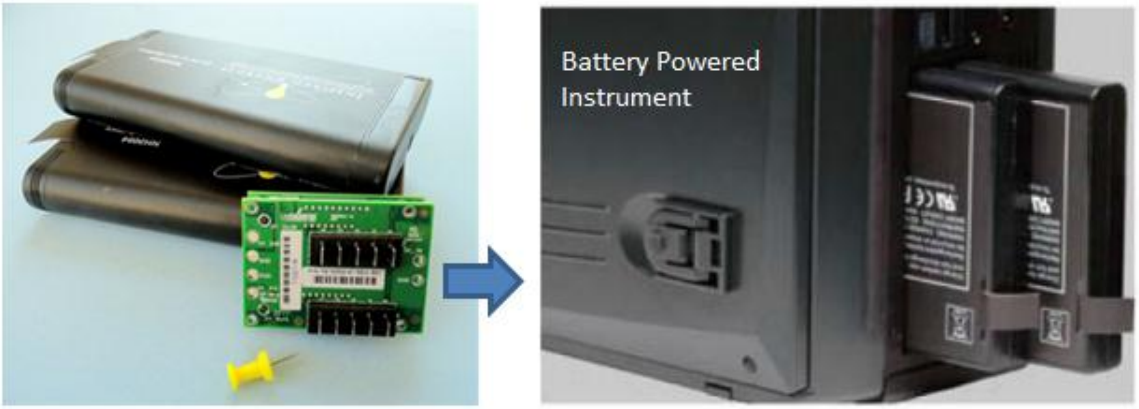
## Introduction to the BC-1022 Quick Start Guide

The Intelligent Battery & Power System (IBPS) BC-1022 Quick Start Guide is provided to assist new users with basic interconnect procedures and to answer most of the commonly asked set-up questions.

The IBPS can manage all aspects of powering a system with battery power. The BC-1022 provides the charging circuits for two smart battery packs as well as all of the monitoring capability and circuits that allow you to parallel up to 32 packs together to build higher battery capacity mobile systems.

Key Features:

- Switches power to load between external power and battery power
- Can simultaneously power system and charge batteries
- Fully configurable to meet specific needs of your system



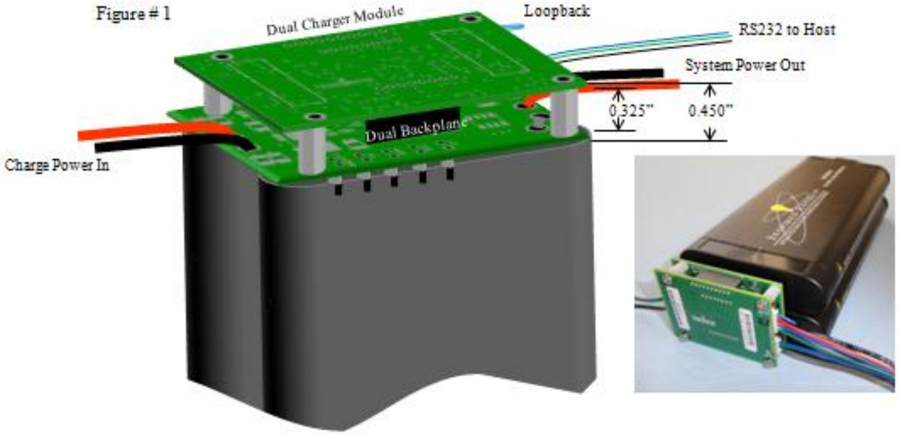
## BC-1022 Specifications

- Max discharge current:  60%-80% of pack specification (max 7A)
- Max charge current: 1A per pack, (airflow is required to reach maximum charge current)
- Input Charge Voltage for 14.4V Battery Packs: 20.1V-30V
- Vin_min_charging = Vflt +3.3V (16.8V + 3.3V = 20.1V example for 14.4V batteries)
  - Maximum controllers in an RS232 string: 4 (8 battery packs) Supports small battery clusters, see BC-1123 for large clusters.
- Charge method: constant current the constant voltage (CC-CV)
- Operating temperature: limited by the battery packs, typically 0-45C (charging), -10 to 50C (discharge)
- Host interface: RS232, no-handshake, 8bit, N, 115k baud to 9600 baud
- High discharge rates may require airflow around the battery packs.

## BC-1022 Module Design

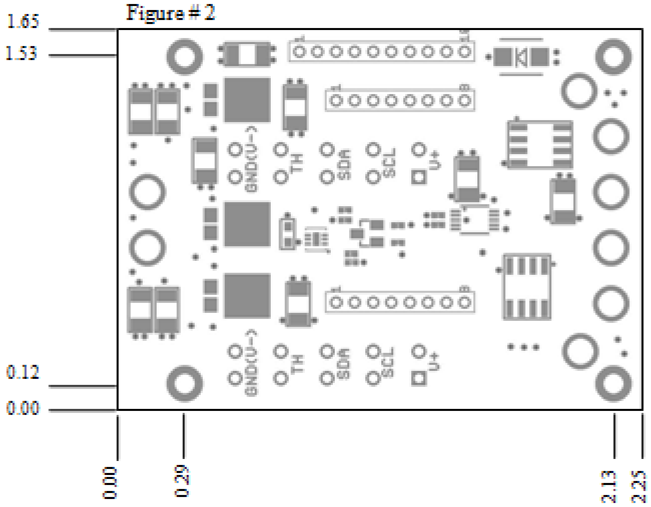
The BC-1022 is a tiny, two module stack for connecting up to two battery packs. The backplane module connects directly to the battery packs and includes solder points for connectivity.  The second module includes the critical charging circuitry and communications connectors.  Figure #1 shows a dual stack board with both power and communications cables.

Figure # 1



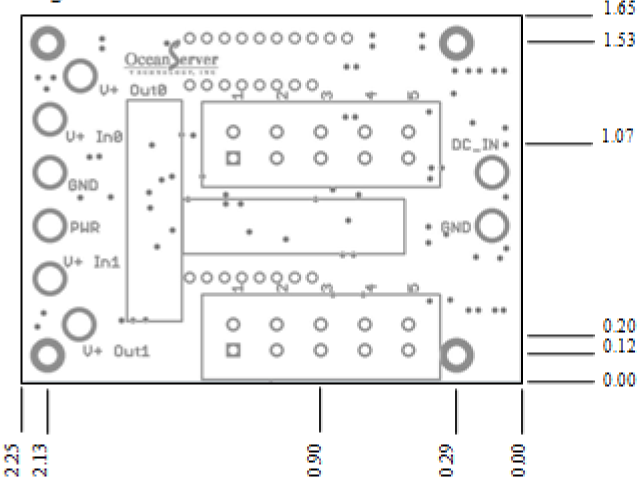
Drawing for Side 1: Backplane Module

Figure # 2



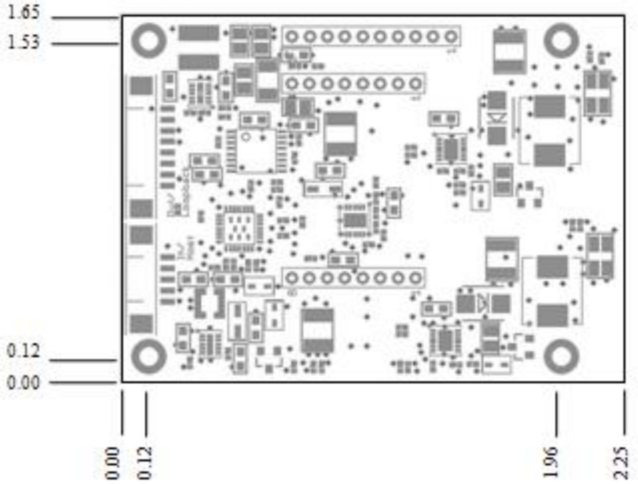
Drawing for Side 2: Backplane Module

Figure # 3



Battery Connectors: Tyco 5787444-1 or equivalent

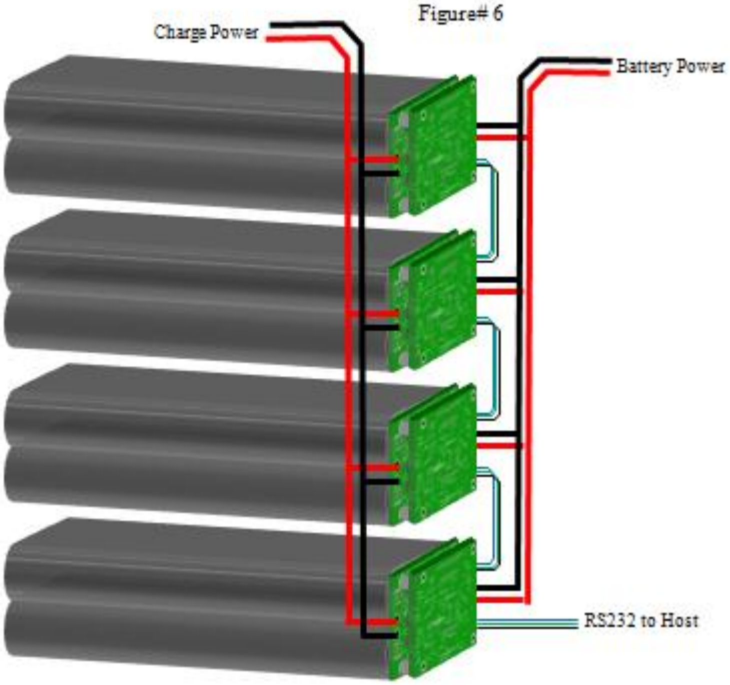
Figure # 5



## Interconnect Diagrams for a Single Module or a Cluster

The BC-1022 supports up to 4 controllers chained together for a total of 8 battery packs. This scalable system allows the user to utilize a single battery pack or build a larger battery cluster (figure #6) with up to 800 watt-hours of energy. The BC-1123 supports large battery clusters up to 32 packs. The battery information is transmitted to the host via a single RS232 serial port with NMEA style strings. The host can parse these strings and monitor the voltage, current, remaining capacity, temperature, faults as well as many other key parameters.

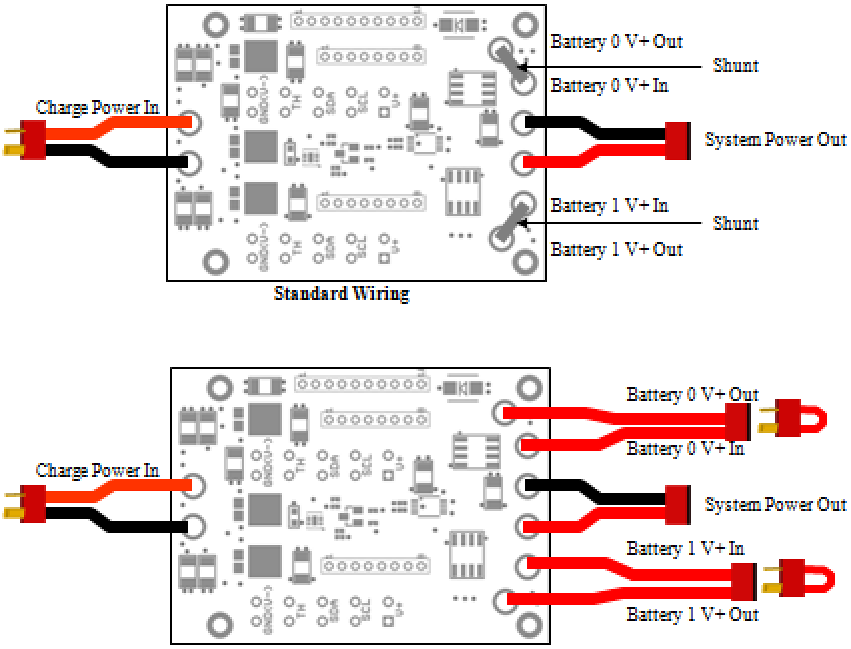
Figure# 6



### *Single Module Setup:*

There are two wiring approaches for installation into a user system. One option is to create a system with external loopbacks and the other being standard wiring. The loopback approach allows the battery packs to remain engaged to the battery controller but with the loopbacks removed the electrical connection between BC-1022 and battery packs is broken. This is an important consideration for shipping to comply with shipping regulations. The standard wiring approach is more common with customers who ship the battery packs separate from the system and have the user install the packs on initial setup for their mobile system (See Figure #7).

Figure # 7

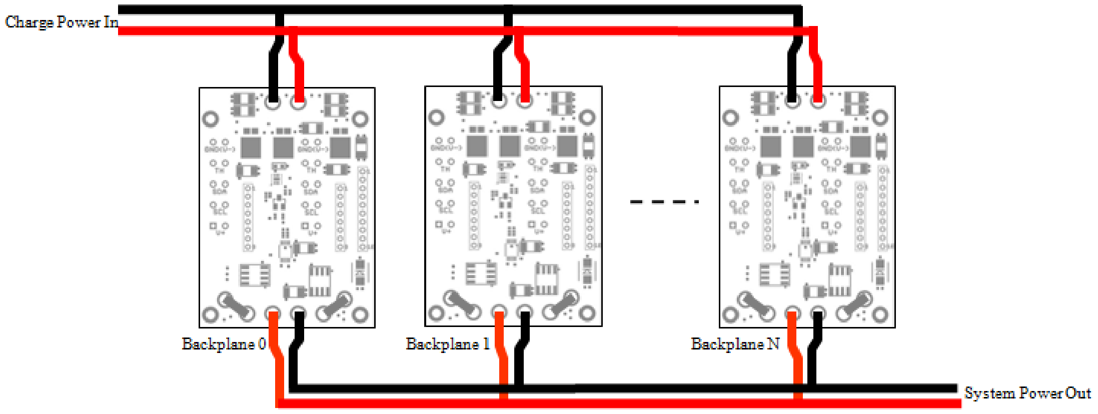


**Standard Wiring**

**External Loopback Wiring**

## Multiple BC-1022 Modules Wired Together

Figure# 8



### BC-1022 Power In Considerations

Raw battery power or power from the external power source is connected through the BC-1022 to your system through the System Power IN (DC_IN) connections.The Base Battery Management Module BC-1022 will handle all aspects of power to your system.

The Base Battery Management Module will monitor all power inputs and connect the appropriate power source to your system. The controller will automatically switch the power being supplied to the system from the batteries to the external power when an external power source is connected. Any remaining power capacity from the external power supply will be used to charge the batteries (if necessary). The power switching happens fast enough so the system experiences no interruption in power.

AC adapters or power supplies can be connected to DC_IN. An external power source is not needed, but if attached, the power can be used to run the system and to charge the batteries (if needed). The AC adapter or power supply should provide Vflt+3.3V up to 30V for charging batteries.

10 Amps of power can be delivered to the system from an external power supply.

Up to 1 Amp of power will be supplied to each battery for charging (if necessary).

Power supplies need to be at least 20.1V to charge 14.4V batteries.

**OceanServer Power Supply Units for Charging** (Vin_min_charging = Vflt +3.3V)

The PS-90 is a 90W, (19V @ 4.74A) power supply with a 2.5/5.5mm barrel connector. Use cable 19-00003-12 to connect to the controller for 10.6V Nominal battery packs. OceanServer offers a 24V version for 14.4 Nominal battery packs.

The PS-150 is a 150W, (18V @ 8.33A) power supply with output DIN-8 Connector. Use DC IN Cable p/n 19-00194-12 to connect to the controller.

The PS-320 is a 320W, (20V @ 10.5A) power supply. Use cable 19-00020-24 (1 per battery pair) to connect to the controller.

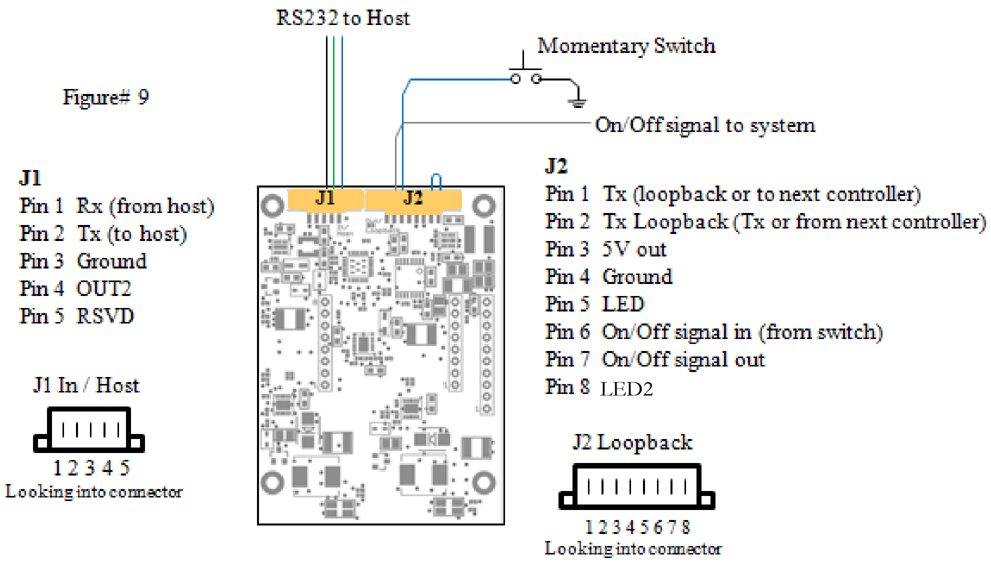
The PS-600 is a 600W, (18V @ 33A) power supply. This supply can be connected in parallel for large battery systems. Use cable 19-00020-24 (1 per battery pair) to connect to the controller. AC into the PS-600 can be either 110 or 220 VAC, selectable via a slide switch on the case.

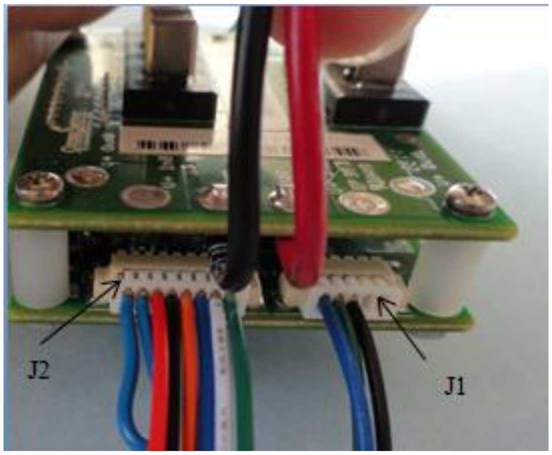
## BC-1022 Communicating & Control

Figure# 9 shows the two connectors for communication and control. J1 is a RS232 connection to the host for monitoring the BC-1022 battery system status and also used to update settings. J2 is used for external status and control of the Base Module. Note, the modules will control all aspects of the charge and discharge cycle so a connection to a serial port is not required.

J2.6 can be attached to a switch and along with J1.4 and J2.7 can be used to generate On/Off signals. Three switch types are supported, a toggle switch, a momentary pushbutton switch, and a PC style momentary switch. When configured as a PC style switch, momentarily pushing the switch will turn the system on and when held for 3 seconds, it turns the system off.

Figure# 9





J1.4 and J2.7 will provide a TTL output that can be used for external control. The output can be configured to be high when the system is turned on and will go low when the external system needs to be turned off. The parameters of this output can be configured through the RS232 port.

A 2 pin header connected to J2.5 and J2.4 (ground) can be used to control an external LED to monitor the percent of charge in the battery system when the system is discharging. The LED will blink about once a second and the ratio of the time on vs. the time off will indicate the percent of charge left on the batteries. Always on will indicate fully charged

### Interconnect Diagram for Multiple BC-1022 Modules in a Cluster

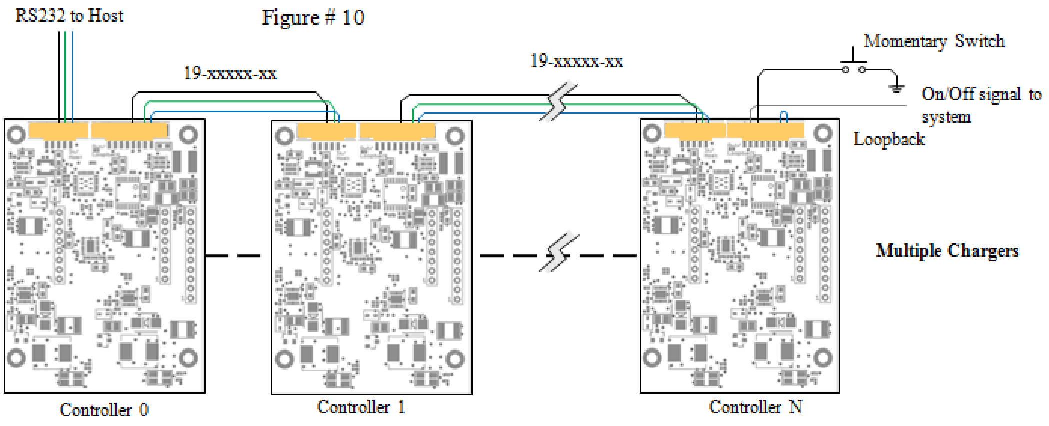


Figure # 10

## System Monitoring & Connecting to a Host

The Base Battery Management Module can be connected to a PC or other system through an RS232 serial port when connected to J1 through OceanServer's serial port conversion cable (19-00212-24). The conversion cable connects to J1 and has a female DB9 style connector to connect to a standard serial port.

To connect to the serial port, the loopback connector cable (19-00212-24) also needs to be installed into J2.

Connecting the BC-1022 to a host system allows users to monitor the battery/power system and to set up and configure advanced features of the IBPS Power System.

A Hyperterminal (or a terminal program such as Tera Term) can be used to connect directly to the controller to monitor status and to customize the behavior of the controller. Some of the items that can be configured are power on and power down policies, display options, and configuration parameters.

Below is an example output from the BC-1022 (Host interface: RS232, no-handshake, 8bit, N, 1 stop bit, 57600 baud):

### Example of Battery controller output in text mode with one battery connected:

```
Bat(model)     Volts    Curr    Power   Pnct     Cap   Temp  Timeto*  Status-flags
--------------------------------------------------------------------------------------
| B02 NI2040HD > 12.49V | 0.000 |    0.0W |  93% | 6.8AH | 24C |   ..   | Fc
--------------------------------------------------------------------------------------

System;  Current:  0.000 Amps discharging), Power:   0.0 W, SWout=OFF()1

Capacity:6.8 AH (93%), 84.3 WHrs, Packs:1, Fd:0,

Bat(model)     Volts    Curr    Power   Pnct     Cap   Temp  Timeto*  Status-flags
--------------------------------------------------------------------------------------
| B02 NI2040HD > 12.48V | 0.000 |    0.0W |  93% | 6.8AH | 24C |   ..   | Fc
--------------------------------------------------------------------------------------

System;  Current:  0.000 Amps discharging), Power:   0.0 W, SWout=ON()1

Capacity:6.8 AH (93%), 84.3 WHrs, Packs:1, Fd:0,

Bat(model)     Volts    Curr    Power   Pnct     Cap   Temp  Timeto*  Status-flags
--------------------------------------------------------------------------------------
| B02 NI2040HD > 12.48V | 0.000 |    0.0W |  93% | 6.8AH | 24C |   ..   | Fc
--------------------------------------------------------------------------------------

System;  Current:  0.000 Amps discharging), Power:   0.0 W, SWout=OFF()1

Capacity:6.8 AH (93%), 84.3 WHrs, Packs:1, Fd:0,
```

### Configuring the battery Help Menu <ESC> H:

```
CMD:H

*********************************
* OceanServer Technology, Inc    *
* www.ocean-server.com (c) 2012  *
* help: support@ocean-server.com *
*********************************

 S - Setup, configure controller settings.
 M - Set SMB fields to display.
 P - LED and IO+Switch policy.
 b - Change the BAUD rate, must be done one EACH controller separately.
 c - Set LCD branding,company
 @ - Checksum Type, *-SUM,%-XOR.
 F - Reset the first Controller to FACTORY DEFAULTS.
 & - Dump controller information. Send output with bug reports.
 H - Help Menu.

   Notes: Switch and output state decoder:   SWout=OFF(%L.Fd.Oc)1 ->
          OFF=extSwitchState (%L=%too low, Fd=FullyDischarged,
                              Oc=OverCurrent)Switch-input.


 Hit <space> to continue
```

**Configuring the battery controller using <ESC> S:** This command allows you to do the initial setup of the battery controller. This command needs to be executed on all new modules to have the controller start outputting data.

```
CMD:S

** System Configuration **


 -> First Controller ID=0, Each controller supports 2 battery packs

 -> Total number of controllers connected together
  cur=16; new(1..16): 1

 -> Shutdown Flag
      b0: Shutdown upon reaching an Average remaining capacity Percent,  +1
      b1: Shutdown when packs reach FULLY DISCHARGED STATE, value later, +2
      b2: Shutdown when current exceeds a value, (integer) value later,  +4
  cur=1; new(0..7):

 -> Percent for shutdown, Integer
  cur=8; new(0..80):
```

```
 -> Sentence Output, 1=HEX(X), 2-TEXT(B), 3-$OPOW(Y), 4-$Mfg(M), 5-LCD1(L)
  cur=2; new(1..5):

 -> Output Rate
  cur=3; new(-2..15):
57600
Flash Write
```

**Total number of controllers connected together:** Input the number of battery controllers in your system.

**Shutdown Flag:** Conditions which will cause an on/off signal to deassert. Add the numbers for each flag and enter the value. For example, to enable the b0 and the b2 flag, set the value to 5. The example above has a current value of one indicating that only the b0 flag is enabled, shutting down based on the battery charge percentage. Below that, the desired charge percentage input. In the above example the value has been set to 8%.

**Percent for shutdown, Integer:** Input the desired battery percentage for shutdown.

**Sentence output:** Input the desired format of the output data. Sentences that end with a %xx or *xx have a checksum, where "xx" is the checksum value. If there is an "%", the checksum is an unsigned byte wide XOR of every character value between the "$" and the "%". If there is an "*", the checksum is an unsigned byte wide sum of every character value between the "$" and the "*".

HEX: Used for the outputting the data used by the Fullbats program

```
$C0,01,03,02,00,03,03,05,00,06,00,07,00,08,03%7C
$B01,04,0000,05,FFFF,06,FFFF,07,0001,08,0B88,09,2FD1%30
$B01,0A,0000,0B,0000,0C,0001,0D,0000,0E,0000,0F,00FA%42
$B01,10,17AC,11,7FFF,12,7FFF,13,7FFF,14,0BB8,15,41A0%4B
$B01,16,0AD0,17,0003,18,1838,19,3840,21, NH2054HD ,1B,3F17%42
$B01,1C,0006%37
$B02,04,0000,05,FFFF,06,FFFF,07,0001,08,0B83,09,4092%36
$B02,0A,FFEE,0B,FFEE,0C,0001,0D,0061,0E,005B,0F,00FA%31
$B02,10,1514,11,4402,12,4402,13,7FFF,14,0000,15,41A0%45
$B02,16,40E0,17,000F,18,16A8,19,3840,21, NH2054HD ,1B,3EE1%4D
$B02,1C,0005%37
```

- $Cx indicates the controller data where "x" is the controller number. The first controller is C0. The controller data is set up as a bit mask were the LSB (on the right) correlates to battery 1.
    - The value after 01 indicates which batteries are present
    - The value after 02 indicates which batteries are charging.
    - The value after 03 indicates which batteries are supplying power to the system.

- The value after 04 is reserved (not presently output)
- The value after 05 indicates which batteries have charge power present
- The value after 06 indicates which batteries have a fault condition
- The value after 07 indicates which batteries are charge inhibited
- The value after 08 indicates the maximum number of batteries supported by the controller

- $Bcb indicates battery data where "c" is the controller number the battery is connected to and "b" is the battery number. The byte values correspond to register addresses followed by the value read from that register. The registers are defined in the Smart Battery Data Specification. See the Smart Battery Data Specification for more information.

TEXT: Used to output the data in an easily read format.

```
Bat(model)     Volts    Curr    Power   Pnct     Cap   Temp  Timeto*  Status-flags
-------------------------------------------------------------------------------------
| B02 NI2040HD > 12.49V | 0.000 |   0.0W |  93% | 6.8AH | 24C |   ..    | Fc
-------------------------------------------------------------------------------------

System;  Current:  0.000 Amps discharging), Power:   0.0 W, SWout=OFF()1

Capacity:6.8 AH (93%), 84.3 WHrs, Packs:1, Fd:0,
```

- SWout shows the state of the on/off signal OUT1 and OUT2. Values in the parentheses indicate the reason the system was shut off by the controller. Refer to the Help menu for definitions of these codes. The value after the parentheses will inidicate the state of the switch signal.

$OPOW: NMEA style output.

```
$OPOW,2,0,-0.0,0.0,5.2,86.3,48.5,D%41
```

- $OPOW identifies the sentence.
    - The 1st value indicates (integer) Batteries present count
    - The 2nd value indicates (integer) Batteries with a abs(current) > 0.1 Amps, charging or discharging
    - The 3rd value indicates (float) Total system current, sum(all packs) [negative=discharge, positive=charge]
    - The 4th value indicates (float) Total power in Watts flowing
    - The 5th value indicates (float) Total Amp Hours, sum(all packs)
    - The 6th value indicates (float) Total Watt Hours, sum(all packs)
    - The 7th value indicates Average battery % capacity, average all packs
    - The 8th value indicates (ASCII CHAR), E=error, C=Charging, D=Discharging

$Mfg: Reserved for testing.

**Output Rate:** Determines how frequently data is output. In the example above the controller is set to output data every 3 seconds.

**Configuring the battery controller using <ESC> M:** This command allows you to enable or disable what battery parameters you want the battery controller to output in the HEX data format. In the example all parameters have been enabled. If you are running the Fullbats software, you want all fields enabled.

```
CMD:M

** Configure MASK for Data Fields to Output  **
  Fields Enabled for Output:

Bit: 6 [AtRateTimeToEmpty(min)]  --> Enabled
Bit: 7 [AtRateOK]  --> Enabled
Bit: 8 [Temperature(0.1K)]  --> Enabled
Bit: 9 [Voltage(mV)]  --> Enabled
Bit: 10 [Current(mA)]  --> Enabled
Bit: 11 [AverageCurrent(mA)]  --> Enabled
Bit: 13 [RelativeStateOfCharge(%)]  --> Enabled
Bit: 14 [AbsoluteStateOfCharge(%)]  --> Enabled
Bit: 19 [AverageTimeToFull(min)]  --> Enabled
Bit: 21 [ChargingVoltage(mV)]  --> Enabled
Bit: 23 [CycleCount(count)]  --> Enabled
Bit: 24 [DesignCapacity(mA)]  --> Enabled
Bit: 25 [DesignVoltage(mV)]  --> Enabled
Bit: 26 [SpecificationInfo]  --> Enabled
Bit: 28 [SerialNumber]  --> Enabled

 Would you like to change any fields [1=Yes,0=No,9=all,8=none]?
  cur=0; new(0..9): 1

Bit: 4 [AtRate(mA)]  --------------> disabled
  1=Enable, 0=Disable  cur=0; new(0..1): 1

Bit: 5 [AtRateTimeToFull(min)]  --------------> disabled
  1=Enable, 0=Disable  cur=0; new(0..1): 1

Bit: 6 [AtRateTimeToEmpty(min)]  --------------> ENABLED
)  1=Enable, 0=Disable  cur=1; new(0..1): 1

Bit: 7 [AtRateOK]  --------------> ENABLED
)  1=Enable, 0=Disable  cur=1; new(0..1): 1

Bit: 8 [Temperature(0.1K)]  --------------> ENABLED
)  1=Enable, 0=Disable  cur=1; new(0..1): 1

Bit: 9 [Voltage(mV)]  --------------> ENABLED
)  1=Enable, 0=Disable  cur=1; new(0..1): 1

Bit: 10 [Current(mA)]  --------------> ENABLED
```

```
)  1=Enable, 0=Disable  cur=1; new(0..1): 1

Bit: 11 [AverageCurrent(mA)]  --------------> ENABLED
)  1=Enable, 0=Disable  cur=1; new(0..1): 1

Bit: 12 [MaxError(%)]  --------------> disabled
  1=Enable, 0=Disable  cur=0; new(0..1): 1

Bit: 13 [RelativeStateOfCharge(%)]  --------------> ENABLED
)  1=Enable, 0=Disable  cur=1; new(0..1): 1

Bit: 14 [AbsoluteStateOfCharge(%)]  --------------> ENABLED
)  1=Enable, 0=Disable  cur=1; new(0..1): 1

Bit: 15 [RemainingCapacity(mAh)]  --------------> disabled
  1=Enable, 0=Disable  cur=0; new(0..1): 1

Bit: 16 [FullChargeCapacity(mAh)]  --------------> disabled
  1=Enable, 0=Disable  cur=0; new(0..1): 1

Bit: 17 [RunTimeToEmpty(min)]  --------------> disabled
  1=Enable, 0=Disable  cur=0; new(0..1): 1

Bit: 18 [AverageTimeToEmpty(min)]  --------------> disabled
  1=Enable, 0=Disable  cur=0; new(0..1): 1

Bit: 19 [AverageTimeToFull(min)]  --------------> ENABLED
)  1=Enable, 0=Disable  cur=1; new(0..1): 1

Bit: 20 [ChargingCurrent(mA)]  --------------> disabled
  1=Enable, 0=Disable  cur=0; new(0..1): 1

Bit: 21 [ChargingVoltage(mV)]  --------------> ENABLED
)  1=Enable, 0=Disable  cur=1; new(0..1): 1

Bit: 22 [BatteryStatus(bits)]  --------------> disabled
  1=Enable, 0=Disable  cur=0; new(0..1): 1

Bit: 23 [CycleCount(count)]  --------------> ENABLED
)  1=Enable, 0=Disable  cur=1; new(0..1): 1

Bit: 24 [DesignCapacity(mA)]  --------------> ENABLED
)  1=Enable, 0=Disable  cur=1; new(0..1): 1

Bit: 25 [DesignVoltage(mV)]  --------------> ENABLED
)  1=Enable, 0=Disable  cur=1; new(0..1): 1

Bit: 26 [SpecificationInfo]  --------------> ENABLED
)  1=Enable, 0=Disable  cur=1; new(0..1): 1

Bit: 27 [ManufactureDate]  --------------> disabled
  1=Enable, 0=Disable  cur=0; new(0..1): 1
```

```
Bit: 28 [SerialNumber]  --------------> ENABLED
)  1=Enable, 0=Disable  cur=1; new(0..1): 1


 Review of enabled fields:

Bit: 4 [AtRate(mA)]  --> Enabled
Bit: 5 [AtRateTimeToFull(min)]  --> Enabled
Bit: 6 [AtRateTimeToEmpty(min)]  --> Enabled
Bit: 7 [AtRateOK]  --> Enabled
Bit: 8 [Temperature(0.1K)]  --> Enabled
Bit: 9 [Voltage(mV)]  --> Enabled
Bit: 10 [Current(mA)]  --> Enabled
Bit: 11 [AverageCurrent(mA)]  --> Enabled
Bit: 12 [MaxError(%)]  --> Enabled
Bit: 13 [RelativeStateOfCharge(%)]  --> Enabled
Bit: 14 [AbsoluteStateOfCharge(%)]  --> Enabled
Bit: 15 [RemainingCapacity(mAh)]  --> Enabled
Bit: 16 [FullChargeCapacity(mAh)]  --> Enabled
Bit: 17 [RunTimeToEmpty(min)]  --> Enabled
Bit: 18 [AverageTimeToEmpty(min)]  --> Enabled
Bit: 19 [AverageTimeToFull(min)]  --> Enabled
Bit: 20 [ChargingCurrent(mA)]  --> Enabled
Bit: 21 [ChargingVoltage(mV)]  --> Enabled
Bit: 22 [BatteryStatus(bits)]  --> Enabled
Bit: 23 [CycleCount(count)]  --> Enabled
Bit: 24 [DesignCapacity(mA)]  --> Enabled
Bit: 25 [DesignVoltage(mV)]  --> Enabled
Bit: 26 [SpecificationInfo]  --> Enabled
Bit: 27 [ManufactureDate]  --> Enabled
Bit: 28 [SerialNumber]  --> Enabled7ißÿÿòd
 Hit <space> to continue

Flash Write
```

### Configuring the battery controller external control <ESC> P:

```
CMD:P

Switch and DC-DC enable output configuration
--------------------------------------------

SWITCH connections input J2-Pin7 (GND J2-Pin4)

  Switch Type; 0=Momentary, 1=Toggle  cur=1; new(0..1):
  Contacts; 0=Normally-Open(NO), 1=Normally-Closed(NC)  cur=0; new(0..1):

LEDs-- Anodes; J2-5(Charge), J2-8(Discharge), Cathode to J2-4(GND)

  LightCoding; 0=Blinks capacity%, 1=ON/OFF  cur=0; new(0..1):
```

```
  Duty cycle for LED ON/OFF pattern in Seconds  cur=4; new(1..10):
  Enable LED signals; 1-Enable, 0=Disable (GND=LED off)  cur=1; new(0..1):

  Sleep when ON/OFF switch is OFF to save power (1=sleep, 0=run):   cur=0;
               new(0..1):

Control Signals for external converters or systems.  Set by
ON/OFF switch and shutdown policy. Connect to last controller(TTL 3.3V level)

  OUT1 Signal (J2-7) Polarity;0=actLOW(ON=0V), 1=actHIGH(ON=5V)  cur=1;
               new(0..1):
  OUT2 Signal (J1-4) Polarity;0=actLOW(ON=0V), 1=actHIGH(ON=5V)  cur=1;
               new(0..1):
Flash Write
```

**Switch type:** Identifies the type of external switch, either momentary or toggle.  Note that turning the switch off doesn't shut off the batteries or the output power.  It is used to generate external on/off signals that can be used by the system to control DC-DC regulators or other components.

**Contacts:** Identifies the normal position of the switch, either open or closed.

**LEDs:** Identifies the behavior of external LEDs during charging/discharging.  If set up for Blinks showing the capacity percentage, the ratio of the on time vs. the off time will indicate the percentage of battery power.  You can set the blink rate in seconds (the default is every 4 seconds).  ON/OFF will have the LED on to indicate charging or discharging.

**Enable LED signals:** Enables or disables the LED functionality.

**Sleep:** Sleep mode is not currently supported.

**OUT1 Signal:** This sets up the behavior of an external logic signal based on an external switch.  It can be used to turn your system on/off.  This signal will toggle every time the external switch is activated.  This signal is by default, active high.  It will go high when the switch indicates the "on" state.  The behavior can be change to be active low.

**OUT2 Signal:** This sets up the behavior of an external logic signal based on an external switch.  It can be used to turn your system on/off.  This signal will toggle every time the external switch is activated.  This signal is by default, active high.  It will go high when the switch indicates the "on" state. The behavior can be change to be active low.

**Configuring the battery controller external control <ESC> b:**

```
CMD:b

** WARNING ** To change baud rate you must break apart all controllers
    and do this operation on each one locally with each controller looped back.

 ** If you have two or more controllers chained you CANNOT SET THE BAUD, All data will
                 be corrupt after this **

    Enter the number 4931 to confirm you understand and want to proceed  cur=0;
                 new(0..5000): 4931
/r -> Baud Rate,              0=115K, 1=57K, 2=19.2K, 3=9600
  cur=1; new(0..3): 1


 ** Baud Rate Reprogrammed **
```

**Baud Rate:** The battery controller supports 4 different baud rates, 115K, 57K, 19.2K, and 9600. If more than one module is connected together, then each modules baud rate needs to be change individually. Data will get corrupted if you try and change the baud rate of multiple modules connected together.

**Configuring the battery controller checksum <ESC> @:** Allows you to change the way the checksum is computed in the output strings.

```
$C1,01,02,02,00,03,02,05,02,06,00,07,30,08,30%7C

CMD:@

Select Checksum type, 1=*>SUM>NMEA, 0=%>XOR>Oceanserver style
: (0..1) cur=0: 1

Flash Write
```

**Select Checksum type:** Input the desired format of the output data. Sentences that end with a %xx or *xx have a checksum, where "xx" is the checksum value. If there is a "%", the checksum is ChecksumP += x; for every character value between the "$" and the "%". If there is an "*", the checksum is ChecksumX = ChecksumX ^ x; for every character value between the "$" and the "*". The sum is kept for all characters in the sentence between the "$" and the "*" or "%", not including these characters. All math is done with UNSIGNED Bytes.

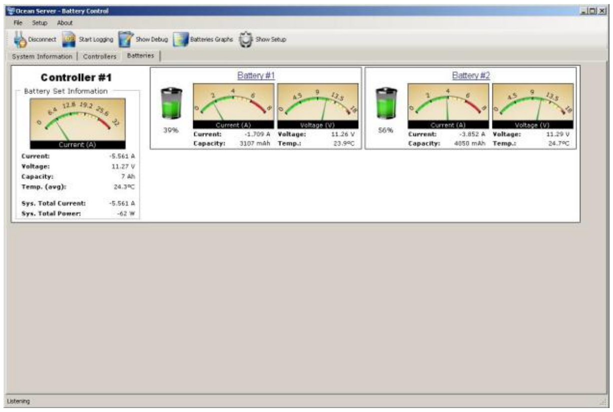
**Outputting setup parameters <ESC> &:** Outputs the setup parameters of the battery controller.

```
CMD:&

Model=BC-1022
FW_Version=V1.3 RC2
FW_Date=27-Feb-2012
```

```
Serial_number=123456789999
Test_date=01 Jan 11
Baud_Rate=1 (57600)
myID=0
LastID=0
ShutdownBitMask=01
ShutdownPercent=8
ShutdownFDcount=2
ShutdownAMPS=30
SentenceFormat=2 (TEXT Human readable)
SecondsPerSentence=3

C_IOMAP=0x69
  (Switch_Type_Toggle)
  (Switch_Active_Low)
  (DISPLAY_is_LED)
  (LED_Blink_DutyCyc)=4 sec
  (ALT_ONOFF_ActiveHigh)
  (SW_OUT_POLARITY_ACTIVE_High)

Fieldmask=00001FFF
Power_ON=OFF, switch_input=1
Module: BC-1022, FW Rev:V1.3
 SN#123456789999, 01 Jan 11


 Hit <space> to continue
```

FULLBATS™ is a Windows-based software program that provides advanced detailed status of the battery powered system. This is a very powerful tool used for monitoring and logging power information for systems with multiple controllers.

The battery controller needs to have its Sentence Output set up for HEX when using Fullbats to monitor the batteries.



Controller #1

Current: -5.561 A
Voltage: 11.27 V
Capacity: 7 Ah
Temp. (avg): 24.3ºC

Sys. Total Current: -5.561 A
Sys. Total Power: -62 W

Battery #1: 39%
Current: -1.709 A
Capacity: 3107 mAh
Voltage: 11.26 V
Temp.: 23.9ºC

Battery #2: 56%
Current: -3.852 A
Capacity: 4050 mAh
Voltage: 11.29 V
Temp.: 24.7ºC

## Connecting to the LCD display

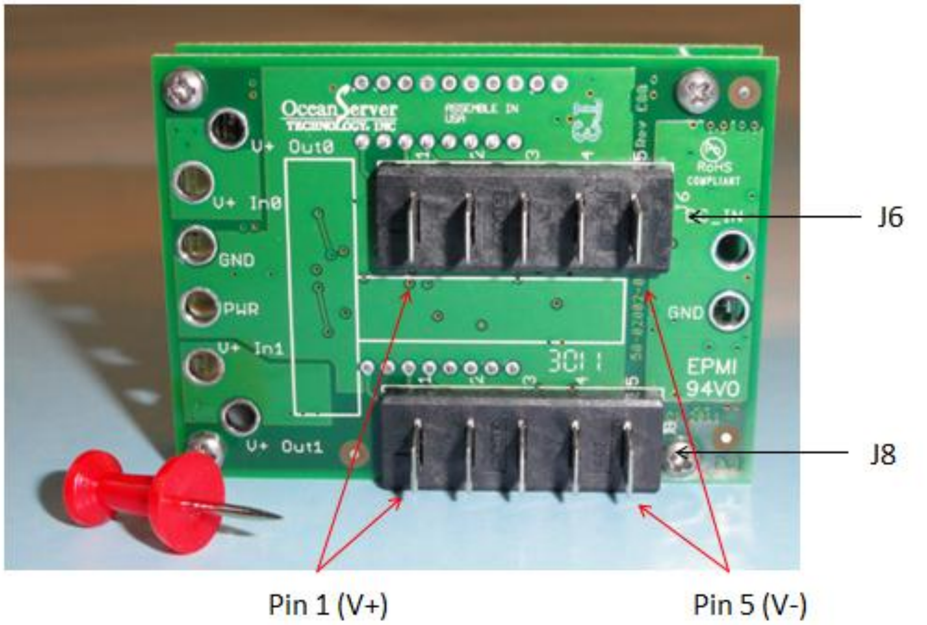
1) Set the baud rate of the controller to 9600 by using the <ESC>b command.
2) Reconnect your terminal window at 9600 baud.
3) Set the output to LCD by using the <ESC>S command. Hit return until you come down to "Sentence Output" and then enter "5" to set it to LCD.

**Wiring to the LCD display**

1) The +5 signal coming from the LCD needs to be connected to J2.3 (only one connection needs to installed).
2) The GND signal coming from the LCD needs to be connected to J2.4 (only one connection needs to installed).
3) The SER signal coming from the LCD needs to be connected to J2.1.

## Connecting Smart Battery Packs to the BC-1022

Up to 2 Smart Battery Packs can be connected directly to the individual BC-1022 Battery Backplane Module. Battery 0 is connected to J6. Battery 1 is connected to J8. The controller will autonomously handle and monitor all aspects of the battery behavior, switching them into the circuit to supply power to the system and charging them when needed.



**Warning be sure to plug battery packs in correctly or damage will result**

The BA-89HC is a 14.4V, 6.2Ah, 8Amp max discharge rate, 89Wh battery that attaches through a backplane. This is an 8 cell battery pack that allows for battery installation and removal similar to what is found in a notebook computer.

Battery Pack Link: http://www.oceanserver-store.com/basmbapalire.html

WARNING: ALWAYS BE CAREFUL TO PLUG IN THE BATTERY PACK IN THE CORRECT POSITION DURING TESTING OR DAMAGE WILL RESULT

## BC-1022 Accessories: Cables, DC/DC Converters, LCDs..etc.

OceanServer offers a variety of options to go with the BC-1022 Battery Controllers. For high efficiency converters please visit OceanServer's online store:

Converter Link: http://www.oceanserver-store.com/dccomo.html

Other Options:

| Part # | Description |
|---|---|

| | |
|---|---|
| 19-00043-12 | CABLE, SINGLE DC_IN JUMPER CABLE, 12" |
| 19-00009-00 | PS-90  POWER CORD, US GROUNDED, 6' |
| PS-90 | POWER SUPPLY, 90W, 19V, 4.74A (24V or  20V versions available for 14.4V Packs) |
| 19-00210-12 | CABLE, DAISY CHAIN (for clustering multiple BC-1022s) |
| 19-00211-24 | CABLE, LOOPBACK/CONTROL (for J2) |
| 19-00212-24 | CABLE, SERIAL COM HOST (for J1) |
| 65-00012-00 | picoPSU, 125W, 6V TO 24V, DC-DC (Included in EK-07 kit) |
| 19-00215-12 | CABLE, picoPSU TO BC-1022 |
| LCD Options | Visit: http://www.oceanserver-store.com/accessories.html |

Image Showing BC-1022 J1 & J2 Connectors with Cables

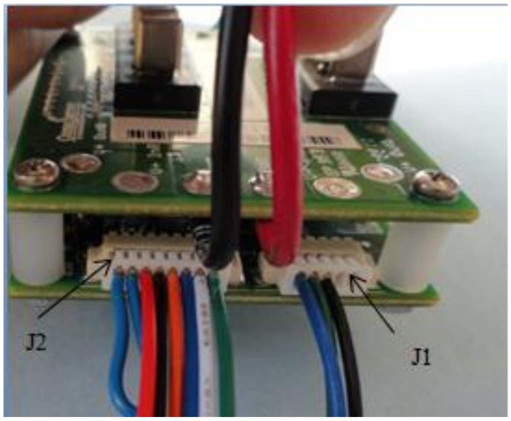


Image below shown with 19-00043-12 CABLE, SINGLE DC_IN JUMPER CABLE, 12" sitting next to BC-1022 (kit version has power in and out cables connected).

To purchase a BC-1022 Development Kit click on below link:

Unregulated Development Kits:

http://www.oceanserver-store.com/ekunbcibpoki.html

Regulated Development Kit with ATX Output:

http://www.oceanserver-store.com/ekunbcibbaki.html

## Controlling the picoPSU with the BC-1022:

If you are not connecting the picoPSU to a PC, you need to use the battery controller to turn the picoPSU on/off.

Connect an on/off switch to J2.6.

Connect J2.7 to the PS_ON# signal of the picoPSU (this is pin 14 of the ATX connector). If you are unfamiliar with an ATX power supply, you should refer to the ATX spec.

Set up the switch type and on/off signal control by using the <ESC> P command.

Set up the switch type and contacts to match your on/off switch.

Set the polarity of the on/off signal coming from the battery controller to be active low. Go down to "OUT1 Signal" and make sure it is set to "0" (active low).

# Customer Support

Send technical questions to: support@ocean-server.com

Or call us at 508-678-0550 during normal business hours.

For mechanical drawings and 3D Models:

http://www.ocean-server.com/download.html