Mobile Power Supply
User Manual
14 April 2021

The MBLSRVPR power supply is designed to provide regulated power to a mother board and external devices. The input is the “12 volt” battery from a vehicle. There are two 50W Boost-Buck power supplies on the circuit board. A serial port allows for configuration of the supply.

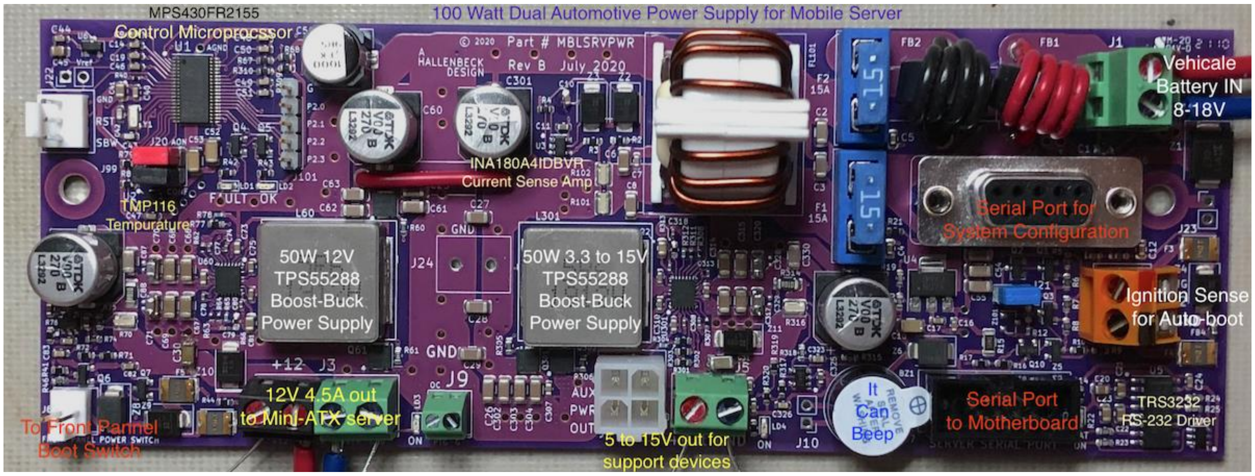
An on-board microprocessor (uP) and sensors for voltage, current and ambient temperature can boot and shut down the server by mimicking the chassis front panel “power” button. Several measurements have their maximum values saved in non-volatile memory that can be examined using the serial port.

The uP ensures that the motherboard will not be booted if the ambient temperature is outside of the user set limits. It will also take down the server if during operation the temperature gets outside of the limits after a user programable period. Similarly, if the battery voltage goes below a user defined limit, the server will be taken down and the supplies shut off after a user programable period.

In addition to the battery input to the power supply, the “ignition” wire from the vehicle is connected to the supply. This is used to boot the server when the vehicle is started after a programable delay. When ignition goes away (there is no voltage applied) the server will be taken down after a programable delay.

The net result is from the users’ point of view, the server is available if the vehicle is running with no action needed by the user. The vehicle battery can’t be drain down so low that the engine can’t be started. The motherboard won’t be booted if it’s outside of its operating temperature range. This allows the motherboard to be an inexpensive grade consumer or small server product.

Overview of the Supply PCB:



The power supply form factor is such that it can be placed next to a Mini-ATX mother board in a chassis that is designed for a Flex-ATX mother board. Airflow should be from left to right w/r/t/ the above picture, which works since most of these chassis, like the SuperMicro SCE300, have the fans in the front of the chassis. The battery and ignition come into the supply from the “rear” of the chassis. A connector such an Anderson Power Pole style is suggested. 16 gauge wire connected to the power supply can be routed through the vent holes in the chassis and then the connector attached to the wires. A color coding of black for ground, red for battery plus and orange for ignition is suggested. This results in a 3 pin connector, making it easy to remove the entire server from the vehicle for servicing.

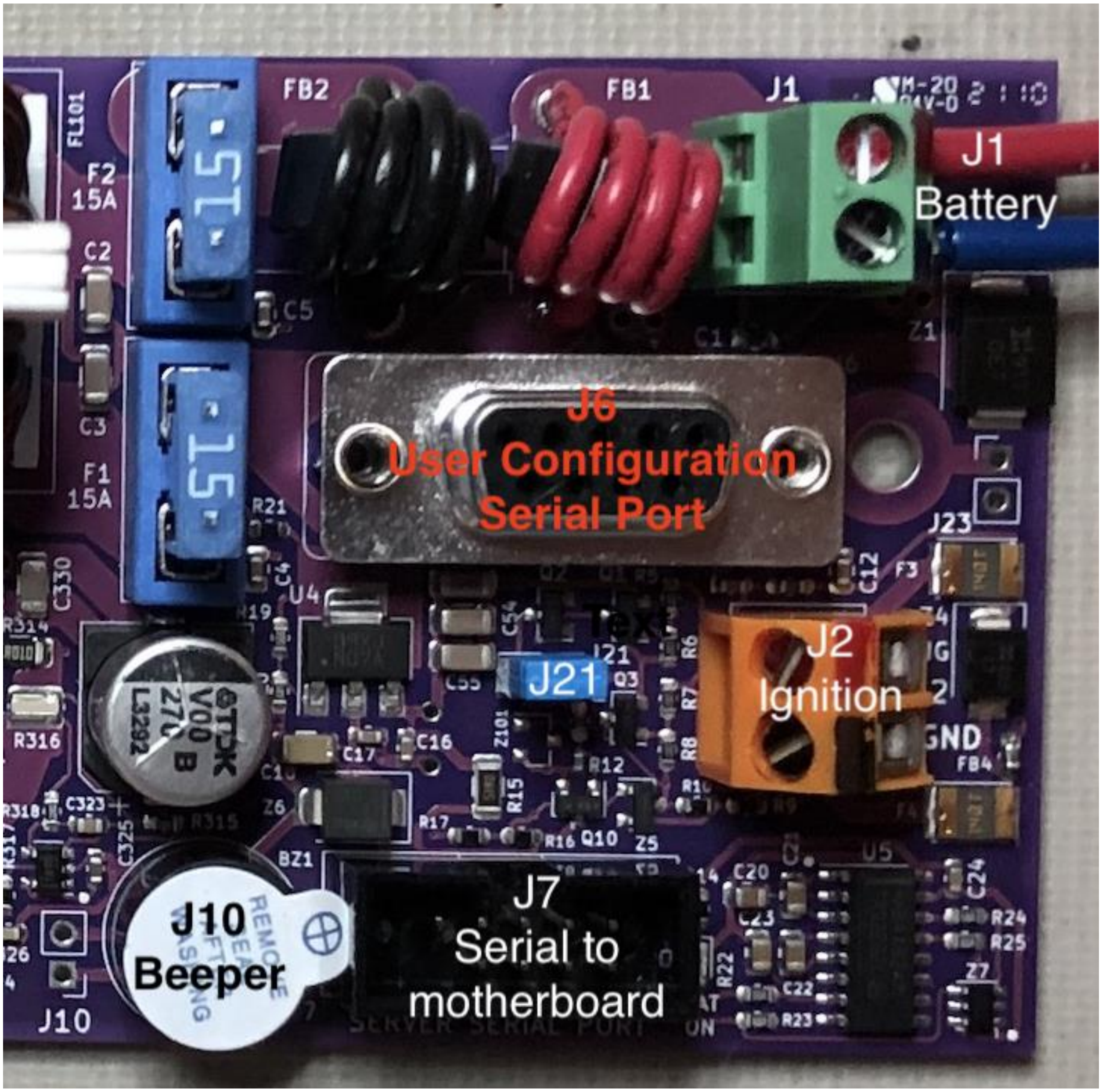
All user connections on the power supply have voltage spike protection, and many also have fuses. Use static protection when handling the board.

When removing the 15 amp fuses, press down on the large white choke/coil and Serial Port connector. Then lift up on the fuses to avoid flexing the printed circuit board (PCB). Rubber feet on the bottom of the boards allow fuses and the 10 pin connector to be inserted by pressing down on the board without the risk of flexing too much.

When removing the serial port connector, press on the fuses and orange Ignition connector to avoid flexing the PCB too much.

Connections

There are a number of user connections to the power supply. On the left side of the printed circuit board (PCB) we see:



J1 is the vehicle battery input. The screw terminals are marked on the PCB and the screw heads may be colored as shown.

J2 is the input from the ignition wire on the vehicle. Both a “positive” and “Ground” connection are provided. Note that Battery Ground and Ignition Ground are the same signal in a vehicle. As such, it is sufficient to simply connect ignition to the top terminal labeled as “IG” the PCB which may also be colored as shown.

J6 is a DB-9 female serial port used to configure the supply. Use a standard DB-9 Male to USB adaptor cable to connect to your PC. The baud rate is 57600. “8-none-1”, no flow control.

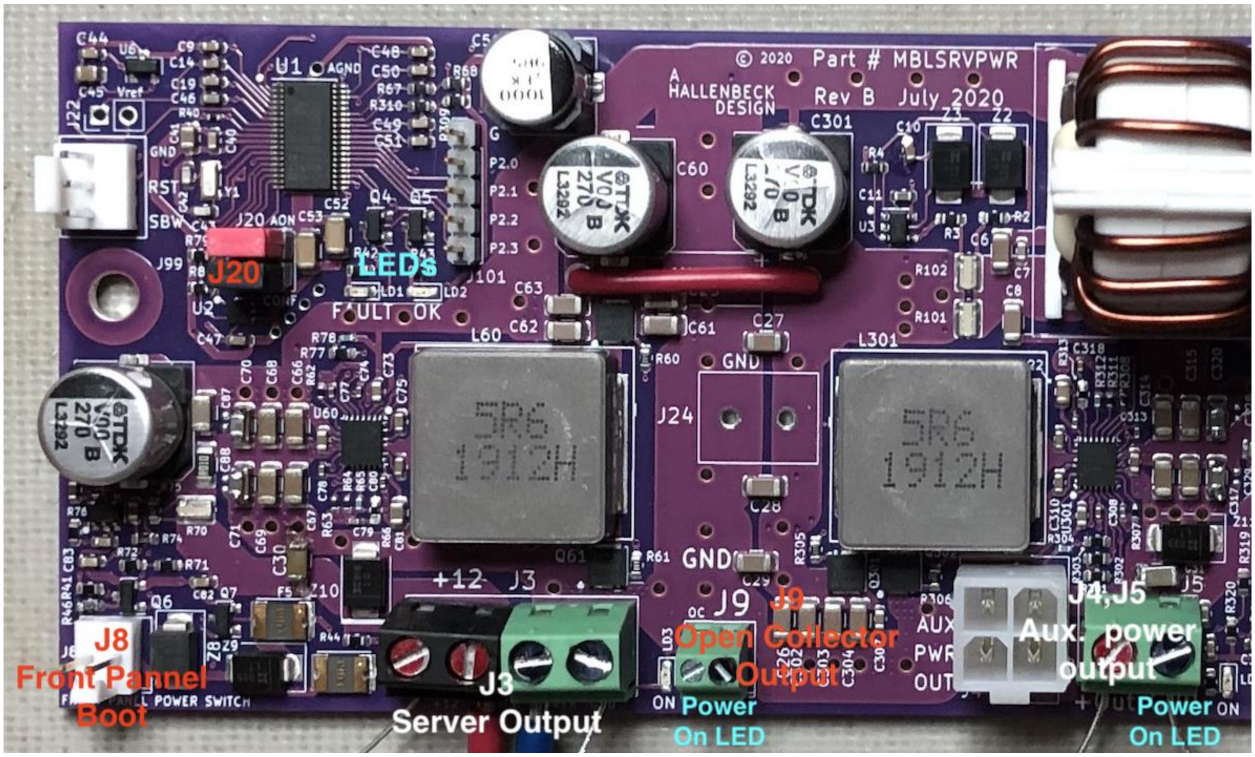
J7 is a 10-pin ribbon connector that connects to your mother board so that the power supply can tell the OS what is going on. This is a one-to-one match for the SuperMicro A2SDI-xC-HLN4F series motherboards. The “xx” is the core count, and can be -2C, -4C or -8C. The pinout for J7 is:

DB9	Header		DB9
6	2	1	1
7	4	3	2
8	6	5	3
9	8	7	4
Nc	10	9	5

Note that other SuperMicro server motherboards have a different pinout for the 10-pin serial headers. Be sure to check your motherboard manual.

J10 is for the “beeper”. Typically the two wires will go to a 4 pin header connector that is on the mother board. There is no polarity on the beeper.

On the Left side of the supply PCB there are more user connections:



The power supply form factor is such that it occupies the empty space in a SuperMicro SCE300 chassis. That chassis it for a Flex-ATX mother board. When using the A2SDI-xC-HLN4F series Mini-ATX mother boards, there is a space left in the chassis. The power supply mounts in this space. That chassis has a 5.5mm/2.5mm “barrel connector” jack on the back with two of the 4-pin connectors on wires. One of those sets of connectors is cut from the barrel connector, the bare ends are insulated with heat sink at the barrel connector, and the wires on the cut-off connector goes to J3. The two yellow wires to the +12, the two back wires to the GND. The 4 pin connector then goes to the motherboard to provide power. This is the Server supply.

The remaining 4-pin connector, which is still wired to the rear 5.5mm/2.5mm barrel connector plugs into J4, the white 4-pin connector, to provide power out of the server chassis to user devices that need power This is the 2nd 50 W supply called the Auxiliary (or AUX) supply. The screw terminal connector J5 can also be used to connect to the Auxiliary (AUX) power supply.

There are small green LEDs next to the server supply output and the aux. supply output that are illuminated when the supply has voltage. There is also a small green LED next to the J7 10 pin Server Serial Port connector that is dimly lit when the uP power supply is on, indicating that the on board microprocessor is up and running.

J8 connects to the chassis front panel. There is a 16 wire ribbon connector that goes from the panel to the mother board. The “power” button is a “shorts when you press it” button that has one side at Ground and the other shows about 3.3V. This is connected to J8 by extending the 2 wires on the ribbon cable. You have to make this “special” version of the front panel ribbon connector. The “right” pin (PCB oriented as in the above picture) is Ground, as indicated by a small ‘G’ on the PCB. The “left” pin is the active signal to and from the switch. The supply uP can both pull this pin to ground to boot or take down the server and can monitor the pin in case the user presses the button to take the server down manually.

The connector is a Molex connector with these part numbers:

2 pin header: 22-01-2027

Pins: 08-50-0113 (cut strip with 100 pins on it)

While this connector is hard to solder the wires to, and difficult to insert the pins into the header, when done right the connector can only be inserted one way on the supply board. As such, it is hard to get the polarity swapped by mistake. Note that the pin spacing is .1”, so other connectors with screw terminals could be user installed.

J9 is an “Open Collector” connector. Under software control, the supply can pull a signal to ground. The user can configure what state the supply must be in to activate the pull to ground transistor. This output has a Transient Protection Diode (abbreviated TVS, go figure). It also has 140 mA self-resetting fuses on both the output and the ground connection. The ground connection is at power supply and server mother board ground. The maximum current that can be pulled to ground is 50 mA, enough for a small relay if you need to work at higher currents.

Header Shorting Pins:

There are two areas where users can install “shorting headers” onto the .1” pins:

J20 is on the left side of the boards. There are two headers with horizontal orientation that can be installed. If the “top” header, which is red in the above picture, is installed the supply will always be “on” when there is battery applied. This makes the supply look like a more conventional supply without the ignition turn-on feature. There is “AON” text above this header on the PCB.

The “bottom” header, which is black in the above picture, is used to enable changes in the configuration. It is above the “CONF” text on the PCB. When installed, the user configuration serial port can be used to change the supply voltages and current limits. The other parameters described in the software section can be changed. When not installed, information can be viewed but not changed.

J21 is on the right side of the board. If this jumper/shorting header is installed, whenever there is battery power the uP is running. This is useful for turning on the uP so you can configure the device with the serial port. It also must be installed for the system to work when configured to look like an “ordinary” power supply.

If you want the server to boot when battery voltage is applied, you should short out both J20 “top”/”AON” and J21. Note that you must take the server down manually with the front panel button, and wait for the server to go down before disconnecting the battery voltage.

Other Connections you probably shouldn’t use:

For the sake of completeness, here are other connections that are not intended to be used by the typical user:

J99 is a 3-pin Molex connector used to “flash” (download) the software into the microprocessor.

J101 is a 5 pin header that has Ground and 4 signals. It is used for debugging when doing software development and can be used to output other signals depending on the power supplies state. Connect to these at your own risk if you are using these advanced features.

J23 can be used to connect the power supply ground directly to the chassis if you carefully scrape off the solder mask on the screw hole. This connection is after the EMI filtering and fuses. In some situations, this may improve EMI performance.

J24 is between the two big gray inductors. It provides access to the filtered battery supply in case that is needed for some other device. This is after the fuses, and as such there is not much power available without starting to impact the current available for the two power supplies. A standard .2” screw terminal connect can be applied here.

Finally, there are four small holes on a diagonal around the “CONF” text. These connect to 3.0 volts, Ground, and the I2C signals of SDA and SCL. The intent here is to have some way to add another sensor to the board for some use that was not anticipated at the time of design. If you’re using this feature, and using the J101 connections, you are also writing your own software for this platform and you’re on your own with no warranty.

The Fault and OK diagnostic LEDs:

There is a red “Fault” and green “OK” led on the PCB as seen on the picture above. The indicate the state of the system. Each lead can display any of the following patterns:

Always on or off.

Fast or slow “blinking” (50 percent duty cycle).

Fast or slow “Winking” (80% on, 20% off)

Fast or slow “Flashing” (20% on, 80% off)

The pattern seen indicates the state of the software. For example, when the green OK led is blinking fast, the ignition is on and the system is waiting to turn on the supplies. When the LED blinks slow, the supplies are on and soon the mother board will be “booted”. When the green LED is on all the time, the system is up.

The red fault LED can be on and yet everything is OK. It just means that at some point in time, there was a brief fault that occurred. The user configuration serial port can be used to see what the problem was.

Technicians who install and maintain servers using this power supply can use these two LEDs to have some idea of what is going on. Note that the serial port can also be configured output what state the system is in, providing detailed feedback about the system. This is done via the Hardware Configuration Bitmask, described later in this document.

Electrical Specifications:

Input voltage:

Range: 7.5 to 18V. The input can briefly (< 100 mS) dip down to 6.5 volts during vehicle cranking. The limitation here is exceeding the 15 amp input fuse rating for too long.

Maximum continuous battery current: 14 Amps

Note that at 12V output and 7.5 V input, an individual supply will draw 7.2 amps.

Standby Current: When the ignition is off, the board draws less than 10 micro amps of current.

Quiescent Current drawn by the microprocessor: < 40 mA

Server Supply:

Output Range: 1.8 to 13 V

Server Supply Current Max: 8.1 to 13V: 4.8 Amps
1.8 to 8V: 5.8 Amps

Note: 4 A continuous load is recommended. (5 A for <8.1 V)

Auxiliary (AUX) Supply:

Output Range: 1.8 to 18V

Aux Supply Current Max: 8.1 to 13V: 4.8 Amps
1.8 to 8V: 5.8 Amps

Note: 4 A continuous load is recommended. (5 A for <8.1 V)

Both supplies can provide currents up to 15 amps for 3 mS before an over-current event is logged and the supplies “fold back” to a current limited mode of 5 (Vout > 8) or 6 (Vout <= 8) amps. This allows for step changes in load, particularly for capacitive or high inrush currents, to be accommodated while maintaining sufficient voltage regulation.

The power supplies can be shorted with no damage to supplies.

If the output voltage is set to exceed 12 V, there will be some current derating when the supply is operating in Boost mode and the battery voltage is less than the set output voltage. User must test to determine if supply is suitable in these situations. Making 18V on only 7.5 volts in will draw a substantial amount of current, much more so that making 12V out.

As such, the available current when running in boost mode is derated as the set output voltage goes above the typically battery input voltage.

Temperature Range:

The range over which the supplies will be turned on (produce output voltage) and boot the server is user configurable, but can not exceed a range of 32 to 138 degrees F. The power supply board can function from minus 40 to + 185 degrees F, but if the ambient is outside of the user configurable range the uP will just wait for the temperature to be in range before turning on the power supplies. The on board uP and it’s power supply will work over that temperature range. Note that the environmental and ambient temperature restrictions of the mother board should dictate the operational range for the supplies, and some degree of margin should be used when setting the operation range. The default range of the supply (prior to user modification) is 35 to 130 degrees F. This is for the SuperMicro A2SDI-xC-HLN4F series motherboards. If the ambient temperature is out of range, the uP will wait until the temperature is in range, and only then turn on the power supplies and boot the system.

Humidity must not be such that condensation on the PCB can occur. Like most motherboards, the power supply is not conformal coated.

Vibration:

The components on the power supply are similar to the components used on a motherboard. They are not AEC-Q200 automotive grade parts. It is suggested that the chassis be mounted in the vehicle using some form of shock absorbing material, such as shock mounts or a cushion. Note also that the motherboard/OS should use SSDs for storage.

Front Panel Power Switch:

Maximum sink current: 50 mA

Maximum input voltage: 3.6 V

Pull to Ground (PTG) Output:

Maximum sink current: 50 mA

(fused at 140 mA on both the output and ground connections)

Maximum input voltage: 18 V

Voltage and Current Measurement Accuracy:

Voltage Measurements: ± 1%

Power Supply Current Measurements: ± 2%

Battery Current Measurements: ± 3% (current > 1 amp)

Battery current can be calibrated at a specific current level which improves accuracy to 1% when operating at currents within ±25% of the calibrated level.

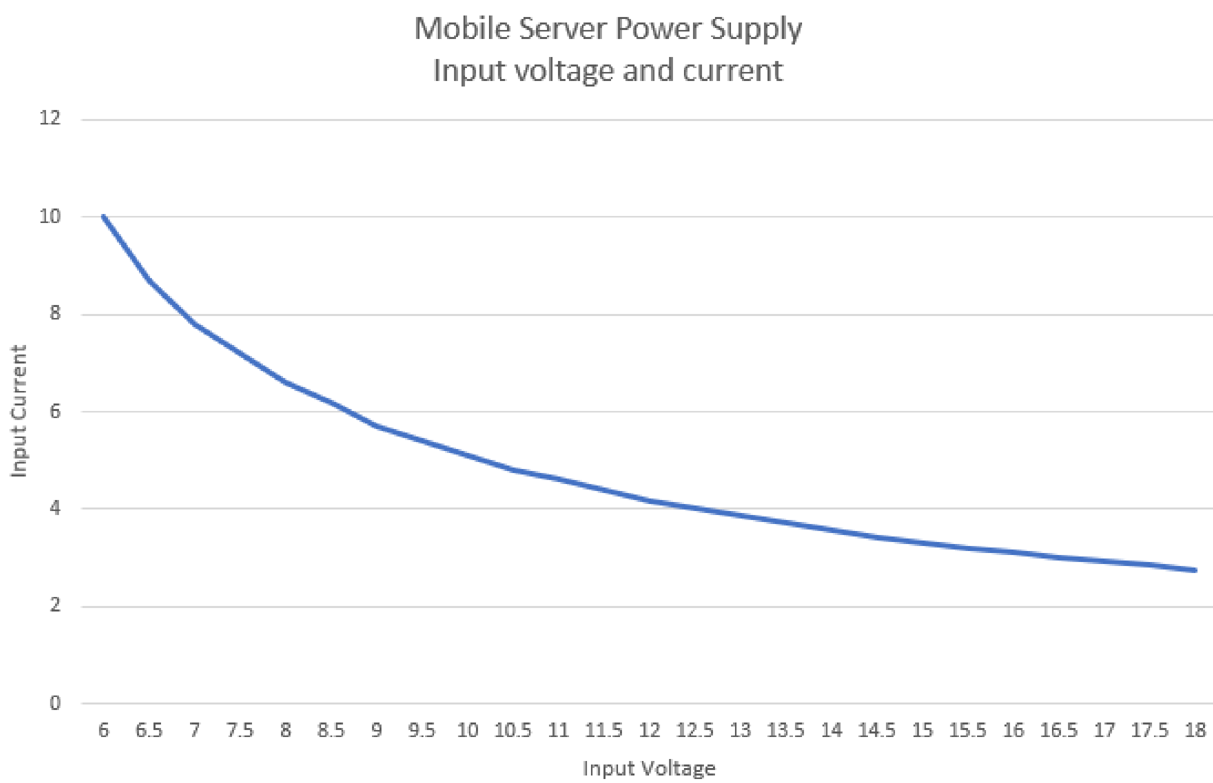
Ignition Input Voltage Threshold:

Vin < 1 Volt → Off, Vin > 2 Volt → On, Vthreshold-Typical = 1.6 V

Input Impedance: Approximately 5K Ohms

Input Voltage vs. Input Current:

Note: Power supply output at 12V with 4 Amp load



Note that for both supplies at 12V with 4A load, the lowest continuous sustained input voltage should not be less than 7.5 volts. At that input voltage and load, the battery input current will be about 14.5 amps.

The supplies will be turned off if the battery voltage in dips below 6.3 volts for any period of time. As such, the 6.5 to 7.5 voltage range allows for a brief dip in voltage during initial 100 mS period when the starter motor is engaged and at stall. When at stall the maximum current draw causes the battery to dip to its minimum voltage.

Functional Description

Use case: Battery and Ignition connections. If the vehicle is running, the server should be operational and booted. In this discussion, times and other constants are taken from the default values of the power supply control system that runs on the on-board microprocessor.

Note: values that are user programable are blue.

The vehicle and the server are off. The driver cranks the vehicle. In the process, the ignition wire goes hot for a brief time (100 mS or so), then goes off (no voltage) while the engine cranks. This would be an “old fashion” rotating system where you “pass through” ignition to get to “crank”. Once the engine starts and is running, the ignition wire goes hot again. Push-to-start more current vehicles tend to not bump the ignition before cranking.

The uP waits for the ignition to be hot (have voltage > 2 volts) for 2 seconds (This is user programable delay). During this time, the OK LED slow flashes green (20% on, 80% off). It also waits for the battery to go above 12 volts (This is a user programable threshold). Once it is above 12 volts the OK LED displays a fast blink. With ignition good and the battery OK, the uP asserts a line that keeps it’s power on even if ignition goes away. The processor waits until the ambient temperature is in range.

It then waits for 3 seconds before trying to turn on the supplies (This is a user programable delay). If at any time the ignition drops or the battery voltage goes < 12 Volts, the entire process starts all over again. After 3 seconds, the power supply chips are turned on. The OK Led displays a slow blink.

Throughout the early start process, if the ignition goes away, the temperature goes out of range or the battery goes low, the process starts all over.

If the ignition and battery are still good, it then checks to see if the ambient temperature is > 35 and < 130 degrees F (These are user programable limits). If not it waits for the temperature to be withing limits. If the temperature is OK, it turns on the power supply chips. It then waits for 300 mS for the power supply chips to reset and stabilize. It then turns on the server supply to 12 volts (This is a user programable voltage). The green OK LED is still winking. It then waits 500 mS and then checks to be sure the ignition is still hot. If so, it turns on the Auxiliary supply. Both supplies are now on.

The uP then waits for 5 seconds before it tries to boot the motherboard (This is a user programable delay). The Green OK LED is now on all the time. Then the “boot” button (the front panel power button) is “pressed” via J8 for 1.1 seconds. Once the server is told to boot, the systems wait for either 60 seconds or for the current consumed by the mother board to exceed 800 mA (This is a user programable current threshold). When the mother board is turned on, the current is around 400 mA and the fans are running, and when Unix boots, the current is about 1.6 A.

At this point, everything is up and running. This is the state that the uP will spend most if its time in. All next steps revolve around when to take the system down.

If the ignition goes off, the system waits for 10 minutes before starting the operating system shutdown procedure (This is a programable delay). If the ambient temperature goes out of range, the system waits for 2 minutes and starts the shutdown procedure. If the battery falls below 12 volts, the system waits for 10 minutes and then starts the shutdown procedure (This is a user programable delay). The shutdown procedure is different depending on what event triggered the need to shut down. If ignition went away and then came back, the system goes back to its steady state. Similarly, if the temperature or battery come back into range, the system goes back to its steady state.

Once the “turn off” event (whatever it was) timeout has passed, the process of shutting down the system starts. The front panel switch is asserted for 2 seconds to start the shutdown process. The system is considered shut down if 90 seconds passes, or if the supply current drops below 800 mA. The power supplies stay up for 10 more seconds so that the fans in the chassis can continue to cool the motherboard components.

The power supplies are then turned off. 2 seconds later, the power supply chips are turned off. 300 mS after that, the uP turns off it’s own power supply and subsequently turns off, waiting for the ignition to go hot again and the whole process starts anew. This last state where the uP turns its power off is the “Goodbye Cruel World” state. When fully off, the power supply current is the leakage current of the capacitors, typically around 5 microAmps.

The state machine that runs all this has over 25 states.

Use Case: Battery power is applied, ignition is ignored, the entire system just needs to power up and boot the system.

This process is much simpler. The hardware is configured with a shorting jump on J21 and on the “AON” pins of J20. The system waits until the battery voltage and temperature are withing range and boots the system.

If the battery voltage drops too low (below 12 volts) or if the ambient temperature goes out of range, the above time delays occur and the system is taken down. It then starts the process all over. The uP is always on when in this state.

Before the user that provided power to the battery can remove the power, the user must to press the front panel power button manually for the 2 seconds. Timing is everything here, the motherboard will shut down without any system OS controlled shutdown if the button is pressed for too long (typically > 4 seconds). When the system is finally down as determined by the user power can be removed.

This would be used in the case where the server was plugged into a power source and not necessarily hardwired into the vehicle. A “Portable sever you can plug into a vehicles’ power system” scenario.

Configuration Settings:

The uP has nonvolatile memory that does not require a battery. In that memory, various configuration parameters are stored. Using the serial interface, the current settings can be viewed. Here is what might be seen when displaying the settings:

```
  Config/EEPROM settings:
Configured by pete
Server Supply Voltage=12000 mV
Aux Supply Voltage=5000 mV
Ignition to Power On Delay=3 Sec
Power On to Boot Delay=5 Sec
Bad Battery to Power Off Delay=600 Sec
Bad Temperature to Power Off Delay=120 Sec
Minimum Operating Temp.=35 F
Maximum Operating Temp.=130 F
Battery Max Current Threshold=14900 mA
```

```
  Server Supply Max Current Threshold=4700 mA
  Aux Supply Max Current Threshold=4500 mA
  Ambient Max Temp. Seen=78 F
  Ambient Min Temp. Seen=76 F
  Ignition Loss to Shutdown=10 Minutes
  Current when OS Running=800 mA
  Battery OK Threshold=12000 mV
  Fan Turn On =105 F
  Fan Turn Off=90 F
  Boot Press=111 x 10mS F:
  Shutdown Press=210 x 10mS F:
  Hardware Config. Bitmask=0x800
```

Given the above description of how the system works, many of these settings should be clear. A few need an explanation though. Some of the configuration values are used to establish *thresholds* for over current events that are logged. There is an option that can be set to run a fan if the ambient temperature is too hot. Finally, there is a hardware configuration bitmask to enable a number of other functions that we will look at in a later section. One of those options displays the state the system is in along with the time spent in a state before transitioning to the next state. For a normal “boot”, that state transition display looks like this:

```
0.0 State  0 ->  1  Have Ignition, wait to check Vbat...
1.5 State  1 ->  2  Wait to re-check Ignition...
3.3 State  2 ->  3  Monitor temp. and battery
0.0 State  3 ->  4  Turn on supply chips...
1.3 State  4 ->  5  Server Supply ON
0.0 State  5 ->  6  Server Supply stabilize
0.5 State  6 ->  7  Aux Supply ON, monitor Vbat & temp...
5.0 State  7 ->  8  Pre-boot Check
0.0 State  8 ->  9  Assert Boot Button
0.0 State  9 -> 10  Button Depress Delay
1.1 State 10 -> 11  90 seconds or Sever Booted wait...
90.0 State 11 -> 12  Everything running OK
```

State 12 is the steady state where everything is up and running OK. Note that some delays are configurable, some are not. If the system is not booting or appears to be hung up, seeing the states can help determine what the problem is. Future firmware may not show the English description, or they may be shorter is program memory space gets tight. (“may” = “almost always”).

The User Configuration Serial Port

When the user has the serial port connection made, and the power supply is turned on either by applying power to the ignition or by jumper-ing J20 “AON” and J21, the serial console is up and ready to be used. If the “CONF” shorting jumper is installed, changes can be made. The user sees this:

```
PS CTRL V 1.0
>
```

The user can type a command. All commands are a single letter, with any letters after that ignored. So “h” and “help” are the same command. That command shows:

```
> h
    Terminate all commands with an 'Enter'.
  Type 'c' to configure
  'A' for current analog measurements.
  'i' to see Server supply I2C registers,
  'j' for Aux supply registers
  'd' to display current EEPROM settings
  'L' to blink LEDs briefly
  'P' On/Off:  ps1=turn on Server, ps0=turn off Server, pa1=power on Aux
  'x' to clear all system faults, max currents & max temperatures
  '~' to reset EEPROM to default values
  'b' to calibrate battery current
  'r' to reset and re-boot
```

The ‘c’onfigure command lets you see and change configuration values. You will see a value displayed and can either hit Enter/Return to go to the next, or type in a new value and hit Enter/Return. You can quit at any time by typing a ‘q’. Changes made prior to that will still be made, this is just handy if you only want to change one value and don’t want to go through the list. Here is an example of what you would see if you just wanted to change the auxiliary supply output voltage:

```
> c
    Config Mode. 'q' to quit, Enter for next,
    or Type in value and Enter to change.
   'A' to see current measurements.
  Configured by pete:
  Server Supply Voltage=12000 mV:
  Aux Supply Voltage=12000 mV: 5000
  Ignition to Power On Delay=3 Sec: q


 --- Exit Config Mode. ---
 Configuration done & saved. Changed 2 bytes.
```

Now displaying the configuration settings shows:

```
> d
    Config/EEPROM settings:
  Configured by pete
  Server Supply Voltage=12000 mV
  Aux Supply Voltage=5000 mV
  Ignition to Power On Delay=3 Sec
  Power On to Boot Delay=5 Sec
  Bad Battery to Power Off Delay=600 Sec
  Bad Temperature to Power Off Delay=120 Sec
  Minimum Operating Temp.=35 F
  Maximum Operating Temp.=130 F
  Battery Max Current Threshold=14900 mA
  Server Supply Max Current Threshold=4700 mA
  Aux Supply Max Current Threshold=4500 mA
  Ambient Max Temp. Seen=78 F
  Ambient Min Temp. Seen=76 F
  Ignition Loss to Shutdown=10 Minutes
  Current when OS Running=800 mA
  Battery OK Threshold=12000 mV
  Fan Turn On =105 F
  Fan Turn Off=90 F
  Boot Press=111 x 10mS F:
  Shutdown Press=210 x 10mS F:
  Hardware Config. Bitmask=0x800
```

You can see that the aux supply voltage is now 5 volts. The units are always in milliVolts or milliAmps. All the values have units after them.

The Boot and Shutdown Press set how long the front panel button on your server is pressed to boot and take-down the system. The units are in “Jiffy”, which is a 10 mS clock. A time of 111 means 1.11 seconds.

Now let us look at the current analog measurements:

```
> a
  Bat V=13897 mV, I=236 mA
  Server V=12017 mV, I=254 mA
  Aux V=5008 mV, I=0 mA
  Max Batt. I=3412 mA, Max Server I=2956 mA, Max Aux I=115 mA
  Current Temp=78, Max Ambient.=78 F, Min Ambient=72 F
  No Faults.
  Low Battery Shutdown Count=2
  I2C bus: 49 TMP116, 74 TPS55288 Server, 75 TPS55288 Aux
  Power Supply State=12  Everything running OK
```

```
  Configured By: pete
```

We can see that the aux supply is at 5 volts, and has no load on it. The server supply is at 12V and has a small 254 mA load. The battery is at 13.9 volts and has a 236 mA load.

Note the maximum values logged for the battery, server supply, and aux supply along with the temperature logs. Any faults will be displayed with the appropriate English text. The I2C bus has found the temperature sensor (TMP116 is the part number for the sensor) and the two power supply chips (The power supply chip is a Texas Instruments TPS55288). The `configured by` is a 31-character field that can be set to anything you can type. It can show who last configured the unit and possibly a date- it is whatever you care to type.

The types of faults you can see are (extracted from the software):

```
"Over Temperature", "Under Temperature", "Battery Current Over Limit",
"Server Current Over Limit", "Aux Current Over Limit", "Server Chip Short",
"Server Chip Over Current", "Server Chip Over Voltage", "Aux Chip Short",
"Aux Chip Over Current", "Aux Chip Over Voltage", "Faulty Temperature Sensor",
"Low Battery Shutdown"
```

As you can see, a lot of conditions are monitored. Sometimes a fault can be ignored. If you connect a device with a large inrush current (due to a heavy capacitive load), you my may get a Over Current fault. None the less, these faults can be useful for figuring out what the supply has been subjected to while it was in use. The ‘x’ command (described later) can be used to clear all the faults.

For the geeks in the crowd (and to help software developers) you can display the values of the registers in the TPS55288 power supply chip:

```
> i
    Server Supply I2C Registers:
  R0-1 Vref=759   0x2F7 (901 mV)
  R2 I Limit=229  0xE5 (50.5 mV, with .010 Ohm=5050 mA)
  R3 Vout Slew Rate=17 (OCP_DELAY=1 -> 3mS, SR=1 -> 2.5mV/uS)
  R4 Vout Feedback=2 (Internal feedback, internal ratio=2 -> 0.0752)
  R5 CDC= E0  1110 0000  No CDC compensation
  R6 Mode=B0  1011 0000  -- Output is ON --
  R7 Status=1  Buck
```

The ‘l’ (lower case’ l’ as in ‘l’eds) command briefly turns on both LEDs and is for diagnostics to be sure the LEDs are working.

The ‘p’ command can turn the power supplies on and off once they have been turned on. This allows you to turn off a supply to wire up a device without having to power down the entire system.

We saw with the ‘a’ command:

```
> a
  Bat V=13897 mV, I=236 mA
  Server V=12017 mV, I=254 mA
  Aux V=5008 mV, I=0 mA
  Max Batt. I=3412 mA, Max Server I=2956 mA, Max Aux I=115 mA
  Current Temp=78, Max Ambient.=78 F, Min Ambient=72 F
  No Faults.
  Low Battery Shutdown Count=2
  I2C bus: 49 TMP116, 74 TPS55288 Server, 75 TPS55288 Aux
  Power Supply State=12  Everything running OK
  Configured By: pete
```

The items in red are values logged by the system in non-volatile memory. The ‘x’ command lets you clear/reset those values. You type the ‘x’, then as per the printed instructions, type a control-D (^D). Any other character aborts the clear. This command also clears the faults listed above in red.

```
> x
 type ^D to clear all faults:
--- All faults have been cleared.
PS CTRL V 1.0
```

The tilde ‘~’ command is similar, but it resets all the configuration/EEPROM values to their factory defaults.

Finally, there is the ‘b’ command to calibrate the current sensor for the battery. For technical people, the current sense resistor is two 3 milliOhm resistors in parallel (1.5 mOhm). With this low value, small changes in the solder depth as the part was soldered on the board can introduce errors. While the accuracy is good enough for knowing if you’re within the limits of the supply, if you are typically operating at a given current, you can calibrate the battery current sense. Note you will need an accurate amp meter to measure the current. When you do this, you will also be trimming out the quiescent current of the microprocessor. As such, at low currents (<

.5 amps) the battery current reading is less accurate and includes both the power supply and microprocessor quiescent current. Here is what that process looks like:

```
> b

  type + or - to adjust, q or ^D to quit
   Battery I=234 mA
  type + or - to adjust, q or ^D to quit
   Battery I=230 mA
-  Battery I=232 mA      (user typed a '-' here)
-  Battery I=240 mA
=  Battery I=232 mA      (user typed a '=' for plus here)
=  Battery I=238 mA
=  Battery I=236 mA     (user typed a control-D here)

Calibration Done
```

Note that you can type an ‘=’ sign in addition to a plus so you don’t have to use the shift key.

Hardware Configuration Bitmask

A 16 bit “Hardware Config. Bitmask” can be configured by the user. The default value of this bitmask is all zeros. From the software, here are the meanings of the bits:

```
//// Bit assignments for hardwareConfigBitmask:
// D0  Take open collector output low if need a fan running
// D1  Take open collector output low if uP power is ON (independent of Ignition)
// D2  Open collector output is a mirror of the front panel push button to
       boot (as driven by the uP). D6 & D7 affect this too
       This can also be configured via D5 to be a "PowerOK"/"~Reset"
       (active low reset, open collector/drain)
// D3  Take open collector output low if battery voltage is
       OK (above the batteryOkVoltage)
//      Note: All battery voltage measurements have 200 mV of hysteresis.
//  --------------------
// D4  Make P2.0-3 reflect all of the above D0 to D3 bits. D0-3 can still be set for
the Open Collector output.
        // Note: the P2 pins are active high to set-up for driving an open collector
transistor for "pull to ground" low.
// D5  Front panel push button output used as a "Power Good"/"nReset".  Goes low
       before power up on boot, and goes low before power down on system take-down.
       Can be used as "enable" for other systems. Functions as an
        active high enable driven by an open collector transistor or FET
// D6  Open Collector output is a "PowerOk" or "~Reset". Allows use of P2.0-3pins
     via the D4 bit and lets the Open Collector output be used for reset so all
     5 control signals are available
// D7  (not used)
// ---------------------
// D8   Don't "push" the front panel button on boot
       (for systems that boot on power up)
// D9   Don't "push" the front panel button on power down
// D10  Don't use the server running current to determine if OS is stopped,
      just wait for the OS_TAKE_DOWN_TIME (typically 90 seconds)
      seconds boot time and assume system is down.
// D11  Print state transitions on Configuration Serialpport (Uart 1)
```

These bits can be used to define the function of the Pull to Ground output connector. They can also be used to drive the J101 pins. Some change the usage of the Front Panel Power Switch J8 to account for the default action of some motherboards w/r/t/ when and how they boot. Some motherboards boot on power up, others need to see the front panel “power” button pressed. This is often configurable in the Bios.

The D8/0x100 and D9/0x200 bits are the most commonly set by a user. In particular, if your system boots on power up, the D8 bit should be set.

Another useful bit is the D11 or 0x800 bit, which has the current state printed out as it changes.

Someday there will be a state diagram in this document, however the code may be updated or get bug fixes not reflected in the state diagram. It is provided to illustrate the over concepts used by the microprocessor and the various sensors to control the power supply and motherboard. Currently, there are about 27 states in the state machine for the uP to control the power supply system.

Composite Fault Bitmask

The next section shows the serial string that is sent from the power supply to the server. One of the data elements in that string is the Composite Fault Bitmask. The strings you can see have been covered above. Here are the bit definitions for that mask:

0x0001 Over Temperature
0x0002 Under Temperature
0x0004 Battery Current Over Limit
0x0008 Server Current Over Limit
0x0010 Auxiliary Current Over Limit
0x0020 Server Chip Fault (the server chip had an over current, over voltage, or short at some point)
0x0040 Server Chip Short
0x0080 Server Chip Over Current
0x0100 Server Chip Over Voltage
0x0200 Auxiliary Chip Fault
0x0400 Auxiliary Chip Short
0x0800 Auxiliary Chip Over Current
0x1000 Auxiliary Chip Over Voltage
0x2000 Faulty Temperature Sensor
0x4000 Emergency Low Battery Supply Shutdown
0x0800 Not assigned

The bits are set when there is an event. There is no count of how often the fault happened, or of the duration of the fault.

Over and Under Temperature are set if the ambient temperature is outside of the user configured limit at any time, including initial power up. Note that this may not be an error, it just means that at some point in time the supplies either had to wait to be powered up, or it was running and had to power down. These bits being set just to advise technicians that the device has been used in some extreme temperature conditions that would have temporarily prevented the system from booting or would have taken the system down after the configurable time delay.

The first 3 Over Current faults (0x4, 0x8, and 0x10) are set when the current exceeds the user configurable current limit. The chip based over-currents are set when the current drawn exceed the limit in the chip for more than 3 mS. That limit is set by the software and is 6 amps for output voltages < 8.1 volts, and 5 amps for output voltages >= 8.1 volts. The power supply chip can provide substantially more current (15 amps) for up to 3 mS in order to provide for a large increase in load (especially capacitor inrush current) while maintaining voltage regulation.

The power supply outputs can be short circuited. The chip will periodically (about every 78 mS) check to see if the short is still there. This reduces power consumption and heating of the chip during a short circuit event. When the short is cleared, normal operation continues (but devices using the supply may be in an unknown state due to the power interruption).

Over-voltage occurs if the output is > 23.5 volts. Since the software limits the range of output voltages (to 13V for the server, and 18 volts for the auxiliary supply), this would be some spike or back feeding of the power supply. The source of the overage must be tracked down.

If the battery drops to 6.3 volts or lower, both supplies are immediately shut down. This is done while the power supply chips have good voltage and will “behave correctly”. Note that your server will not have a change to be taken down correctly. This action includes if you just “pull the plug” and disconnect the battery.

Server Serial Port Description

The 10 pin ribbon connector is a serial port running at 38.4K Baud, “8-none-1”. About every 4 seconds it supplies the server with information about the state of the power supply, and if the server will be going down soon. The C code for creating the string that gets sent to the motherboard looks similar to this:

```
/////////////////////////////
//  Talk to the Unix system running on the motherboard
/////////////////////////////
void doTalkToMotherboard(void){
    printf("{\"Id\":\"MBLSRVPWR\"");
    printf(",\"Bv\":%u", batteryVoltage);
    sprintf(s, ",\"Bi\":%u", batteryCurrent);
    printf(",\"Sv\":%u", serverVoltage);
```

```
    printf(",\"Si\":%u", serverCurrent);
    printf(",\"Av\":%u", aux12Voltage);
    printf(",\"Ai\":%u", aux12Current);
    printf(",\"So\":%u", supplyServerSupplyIsOn);
    printf(",\"Ao\":%u", supplyAuxSupplyIsOn);
    printf(",\"Ps\":%u", serverPowerSupplyState);
    printf(",\"Bk\":%u", batteryOkVoltage);
    printf(",\"Tc\":%u", currentAmbientTemperature);
    printf(",\"Tm\":%u", minOperatingTemp);
    printf(",\"Tx\":%u", maxOperatingTemp);
    printf(",\"Th\":%u", maxTemperatureSeen);
    printf(",\"Tl\":%u", minTemperatureSeen);
    printf(",\"Bt\":%u", batterySupplyMaxCurrent);
    printf(",\"Bp\":%u", batterySupplyPeakCurrent);
    printf(",\"Sc\":%u", serverSupplyVoltage);
    printf(",\"St\":%u", serverSupplyMaxCurrent);
    printf(",\"Sp\":%u", serverSupplyPeakCurrent);
    printf(",\"Ac\":%u", auxiliaryySupplyVoltage);
    printf(",\"At\":%u", auxiliarySupplyMaxCurrent);
    printf(",\"Ap\":%u", auxiliarySupplyPeakCurrent);
    printf(",\"Fb\":%u", compositeFaultBitmask);
    printf(",\"Fs\":%u", lowBattShutdownCount);
    stringPtr = addToEndOfString(stringPtr, ",\"Ds\":");
    if(timeLeftUntilShutdown > 0xFFF0){
        printf("-1");
    } else {
        printf("%u", timeLeftUntilShutdown);

    }
    Printf("}\n");
}
```

The format is a group of data elements, which each element being comma delimited. The 1 or 2 character name of the data element is followed by a colon, and then by the decimal value or a string value. The length of the string can be up to 380 bytes, typically it’s about 300 bytes long. A sample output string looks like:

{"Id":"MBLSRVPWR","Bv":13888,"Bi":228,"Sv":12017,"Si":246,"Av":12025,"Ai":2,”So”:1,"Ao":1,"Ps":11,"Bk":12000,"Tc":77,"Tm":35,"Tx":130,"Th":77,"Tl":73,"Bt":14900,"Bp":960,"Sc":12000,"St":4900,"Sp":256,"Ac":12000,"At":4700,"Ap":38,"Fb":16384,"Fs":1,"Ds":-1}

This is a JSON encoded data object.

The first data element looks like: `″Id″:″MBLSRVPWR″`. This is the Power Supply ID, which is MBLSRVPWR for this circuit board. This lets server-side software work with any future supplies that might have different data elements that are sent to the server. This is also an example of when the value of the data object is indeed a string and not a hex number.

Most of the data elements are self-explanatory from the names. The "`timeLeftUntilShutdown`" is how long (in seconds) until the "Power Off" button on the front panel is pressed, initiating a system shutdown. A value of -1 means the supplies are not scheduled to go down. This lets the OS on the mother board issue a warning to all users. The front panel switch will be pressed (via J8) in order to start the system take-down process after the number of seconds have elapsed.

Detailed Description of JSON data elements:

Color Coding of Values is:

Blue – user programable value

Red - Measurement of the value "right now"

Green – Log Value. A value logged by the system, typically a Peak or minimum value

Black – Other values

All voltages are displayed and entered in milliVolts.

All currents are displayed and entered in milliAmps.

Bv – `batteryVoltage` - This is the battery voltage as seen on the power supply board right now.

Bi - `batteryCurrent` - This is the amount of current being drawn from the battery "right now".

Sv - `serverVoltage` - This is voltage output for the server supply right now. Units are Millivolts

Si - `serverCurrent` - This is the current the server supply is providing right now. Units are milliAmps

Av - `aux12Voltage` - This is the voltage output of the Auxiliary supply right now. Units are milliVolts

Ai - `aux12Current` - This is the maximum current coming out of the the Auxiliary supply right now. Units are milliAmps

So - `supplyServerSupplyIsOn` - This is a Boolean for if the server supply is turned on. While this may seem silly, since if the server is not up what is reading this serial port, it allows for an external system to know the status of the server power supply. Zero is off, any other number is on.

Ao - `supplyAuxSupplyIsOn` - This is a Boolean for if the Auxiliary supply is turned on.

Ps - `serverPowerSupplyState` - This is the current state of the power supply control state machine

Bk - `batteryOkVoltage` - This is the programed voltage for the threshold used to determine if the battery voltage is OK (> the threshold) or not. The range is 6.5 to 14.5 volts. Keep in mind that the supply does a “hard and fast” turn off of the power supplies at a battery input voltage of 6.3 volts. This value is set to avoid draining a battery so far that the vehicle can not be cranked/started.

Tc - - This is the ambient temperature right now

Tm - `minOperatingTemp` - This is the programable lowest temperature that the supplies can be operating. The range is limited between 0 and 60 degrees F.

Tx - `maxOperatingTemp` - This is the programable highest temperature that the supplies can be operating. Together with the Tm, it defines the temperature range over which the supplies can be operating. The range is limited between 85 and 138 degrees F

Th - `maxTemperatureSeen` - This is the ‘h’ighest temperature seen by the system at any point in time. This logs the maximum temperature seen.

Tl - `minTemperatureSeen` - This is the ‘l’owest temperature seen by the system. Note that both the Th and Tc are the extreme temperatures seen when there was ignition or whenever the supplies were up. This logs the boundaries of the environment for the server.

Bt - `batterySupplyMaxCurrent` - This is the programed threshold for battery current above which a battery overcurrent exists. The range is 0 to 19 amps.

Bp - `batterySupplyPeakCurrent` - This is the peak/maximum battery current the supply has ever drawn. A log of the maximum current seen.

Sc - `serverSupplyVoltage` - This is the programable voltage for the server supply. This is the output voltage that the supply should be running at. The range is 1.8 to 13 Volts, the units used are millivolts.

St - `serverSupplyMaxCurrent` - The server supply current threshold for determining that an “overcurrent” has occurred. The range is 0 to 5 amps.

Sp - `serverSupplyPeakCurrent` - This is the peak current the server supply has ever provided. Think “Peak current ever seen” log entry

Ac - `auxiliaryySupplyVoltage` - This is the programable voltage for the auxiliary supply. This is the output voltage that the supply should be running at. The range is 1.8 to 18 Volts, the units used are millivolts.

At - `auxiliarySupplyMaxCurrent` - This is the programable threshold at which an overcurrent exists. The range is 0 to 6 amps.

Ap - `auxiliarySupplyPeakCurrent` - This is the peak current the auxiliary supply has ever provided.

Fb - `compositeFaultBitmask` - This is the fault bit mask, as defined above

Fs - `lowBattShutdownCount` - This is the number times the supply has ever had to do an emergency shut-down and turn off the supplies due to the battery voltage going below 6.3 volts. If the server seems to “keep crashing”, this can be a clue that the input battery voltage keeps dipping too low, even if it’s just for 10 mS.

Ds - `timeLeftUntilShutdown` - This is the number of seconds until the front panel power button is pressed to start the server shut-down. A value of -1 means no shutdown is anticipated.

Summary and Future Work

The microprocessor used here is a Texas Instruments MPS430FR2155. It has 32 kB of MRAM (like Flash) for the program and 4 KB of ram. Currently, the program takes about 30 K, and ram use is about 2500 bytes. Some of the program space must be reserved for big fixes.

As the supply is used, there may be additional parameters for the Config/EEPROM settings that need to be created. There is no easy upgrade path for the uP in the package used- it’s about maxed out for this series of part.

As hinted at in this data sheet, the part is heavily targeted around the SuperMicro SCE300 chassis with the A2SDI-xx-HLN4F series motherboards in the -2C (two core), -4C (four core) and -8C (eight core) processors. Do not use the 12 and 16 core parts unless you really know what your software will be doing. Doing computationally intense operations such as video transcoding can use all the processors on the chip and that would require more current than this supply can provide.

Here is a picture of the entire system, mother board and supply:

A2SDI-8C-HLN4F in the SCE300 Chassis