PUMPKIN™
INCORPORATED
750 Naples Street • San Francisco, CA 94112 • (415) 584-6360 • http://www.pumpkininc.com

UM-31
User Manual

EPSM1 User Manual

TRADEMARKS

The following are Pumpkin trademarks. All other names are the property of their respective owners.

- Pumpkin™and the Pumpkin logo
- Salvo™ and the Salvo logo
- CubeSat Kit™ and the CubeSat Kit logo

DISCLAIMER

PUMPKIN RESERVES THE RIGHT TO MAKE ANY CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO CORRECT ERRORS AND IMPROVE RELIABILITY, FUNCTION, APPEARANCE OR DESIGN. PUMPKIN DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

PUMPKIN INCORPORATED

744 Naples Street
San Francisco, CA 94112 USA
tel: (415) 584-6360
fax: (415) 585-7948

web: http://www.pumpkininc.com/
email: info@pumpkininc.com

web: http://www.cubesatkit.com/
email: info@cubesatkit.com

CHANGELOG

Rev.	Date	Author	Comments
A	20190910	AEK	Initial version.
B	20190915	AEK	Added sections on SAI wiring, output block fuses, additional / new images.
C	20191001	AEK	Added example 2, showing battery charging as a function of available SAI power. Improved status dump formatting.
D	20230212	AEK	Updated to reflect operation under recent firmware releases.

Purpose

This manual describes the Pumpkin EPSM1 and how to use it.

Introduction

The EPSM1 is a powerful, capable and smart switch-mode dc/dc converter, with a variety of features. See the datasheet for more information.

WARNING The EPSM1 is an expensive piece of equipment, which manages high currents and voltages. Configuring, operating or interfacing incorrectly with the EPSM1 may lead to its damage, or the damage of other components. This can happen extremely quickly, and without warning.

The user is afforded a lot of flexibility when working with the EPSM1, but with that comes the possibility of configuring, operating or interfacing to the EPSM1 incorrectly, with serious results. Therefore always work methodically around the EPSM1, especially when any conditions (commands, inputs, loads, etc.) are changing.

Any electrical damage to the EPSM1 shall be assumed to have been done by the user, since the EPSM1 is tested at Pumpkin prior to delivery to an end-user. Such damage is **not** covered under warranty.

Datasheet & Relevant Documents

The EPSM1 datasheet is currently in its preliminary form, and is available to end-users. Refer to its for connector reference numbers, connector pinouts, and other datasheet parameters.

Also, refer to the Pumpkin SupMCU SFR online reference manual for information on SupMCU-based commands and telemetry.

Finally, refer to the Pumpkin BM 2 datasheet and additional manuals for details concerning its operation.

PCBs

The Pumpkin EPSM1 is composed of three PCB assemblies:

- 705-01723 for the power card
- 705-01576 for the lid card
- 705-01961 for the FPGA card

The EPSM1 has at its core an SEU-immune MicroSemi FPGA, and multiple GaNFET-based dc/dc switching blocks. It also has a Microchip PIC24E MCU, running Pumpkin's SupMCU RTOS-based core software architecture, with EPSM-specific firmware extensions. Various connectors are provided to interface to power sources (solar arrays and batteries) and power sinks (loads and batteries).

The FPGA operates in a default mode with hard-coded initial settings. Intervention from the SupMCU is not required for proper FPGA operation. The SupMCU requests telemetry from the FPGA, and can also redefine various settings in the FPGA, while functioning as a "gatekeeper" for the FPGA. All interactions with the command, control and telemetry sides of the FPGA are through the SupMCU.

PCB Details

The PCBs are multilayer and utilize heavy copper cladding.

Tips for Safe Operation

Tip Always observe polarities when attaching devices to the EPSM1.

Tip Always observe current limits and maximum voltages when attaching devices to the EPSM1.

Tip Refer to the SupMCU's splash screen for various information about the firmware revision installed on the EPSM1.

Tip Use the SupMCU's telemetry display (enabled via **`epsm:deb en,<mask16>`**) to see all of the relevant EPSM1 FPGA-driven parameters and values at-a-glance.

Tip Use the EPSM's Non-Volatile Memory (NVM) to permanently define configurable settings for your convenience (e.g., to enable the telemetry display at startup).

Tip When setting external inputs (e.g., a power supply that will feed the SAIs), using a voltage setting that simplifies real-time power calculations is very helpful. E.g. pick 20Vdc in or 25Vdc in, to easily calculate 100W input power (at 5A or 4A in, respectively) from the instantaneous current displayed on the power supply.

Tip Have a good idea of what nominal operation for the EPSM1 system as a whole looks like, so that when you make changes, a drastic change in the system's operation is easy to detect.

Tip Become familiar with issuing SCPI commands via the CLI interface of the SupMCU on the EPSM.

Tip Become familiar with NVM-related operations on the SupMCU, via SCPI commands over I2C or the CLI.

Tip While there is no thermal convection in a vacuum, it's still a wise decision to operate the EPSM1 with a fan over it, in all situations (including stand-alone without being coupled to a structural heatsink).

Warnings

Warning Ring bus voltage is always present on the EPSM1, even when the FPGA is disabled. The ring bus voltage will eventually collapse to 0Vdc only when all sources of input power are completely disconnected from the EPSM).

Warning The EPSM can potentially be destroyed when too high a ring bus voltage is present, with an OVP shunt resistor of inadequate power handling ability. Read and understand the OVP section of the datasheet, and implement an external OVP shunt resistor (connected to the top card of the EPSM) as required, with adequate thermal mass.

Warning Since the EPSM is not delivered in a case or enclosure, it is imperative that extra caution be used when working around the EPSM1, especially with conductive tools. Always turn off and disable all power sources before making any chances to the connections of the EPSM1.

SupMCU

The SupMCU can be reprogrammed ("reflashed") at any time when the EPSM1 is powered. Use a Pumpkin USB Debug Adapter and Pumpkin's firmware update GUI[1] to reprogram the SupMCU.

Tip It is recommended that the EPSM1 run in a reduced-power mode (minimal power inputs, battery disconnected or disabled, no loads on outputs) when reflashing the SupMCU.

Note Some reflashing may erase and re-initialize the NVM parameters of the SupMCU. Before reflashing, make a note of the NVM parameters (`cd ..; cd NVM; list` in SupMCU's CLI).

FPGA

The FPGA can be reprogrammed at any time when the EPSM1 is powered using a Pumpkin FPGA Programming Adapter 1 and a MicroSemi FlashPro4, with MicroSemi FPExpress software.

Note FPGA reprogramming is normally only done at the factory, and not by end users.

Prior to (re-)programming the FPGA, disconnect the battery and all loads from the EPSM, and limit aggregate SAI current to 800mA.

Tip While reprogramming the FPGA, the SupMCU will throw a range of errors -- these can be ignored.

On the Pumpkin FPGA Programming Adapter 1, be sure to populate jumper **`JP1`**. Do not use **`J155`**, and leave **`H2`** unconnected.

[1] ds30 Loader GUI, special version for Pumpkin.

Connections

Overvoltage Protection Shunt

The default / integrated Overvoltage Protection (OVP) shunt is located on the underside of the EPSM1's top card.

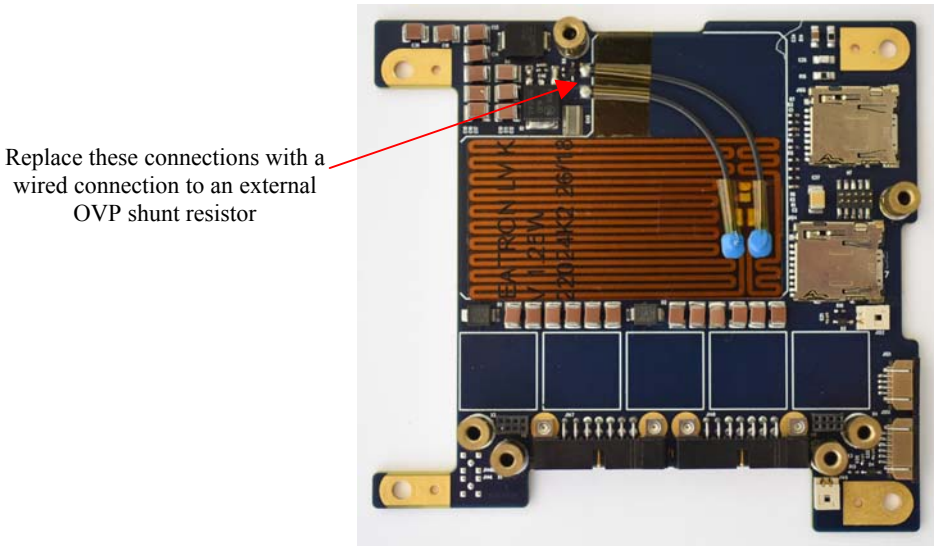

Figure 1: Underside of lid card, showing default / integrated OVP shunt resistor

The OVP shunt resistor has limited ability to absorb energy associated with overvoltage conditions on the EPSM1's ring. For applications that require a greater energy absorption than what the default shunt resistor can safely provide, the two wires connecting the resistor to the shunt circuit should be disconnected, and an external resistor should be connected to those two pads via discrete wires.

Tip The upper of the two OVP shunt resistor solder connections shown in Figure 1 is connected to power ground (**PGND**). If a shunt resistor cannot be isolated from **PGND**, ensure that this is the node that is connected to system ground.

SAIs

The EPSM1's Solar Panel Inputs (SAIs) are distributed across four 10-pin connectors. SAI1-SAI4 are on the first two, and SAI5 and

SAI6 are on the last two. Because of this and the per-pin current ratings of the Hirose DF13-series connectors, SAI5 and SAI6 each have twice the current rating of SAI1-SAI4. The voltage ratings of all six SAIs are identical.

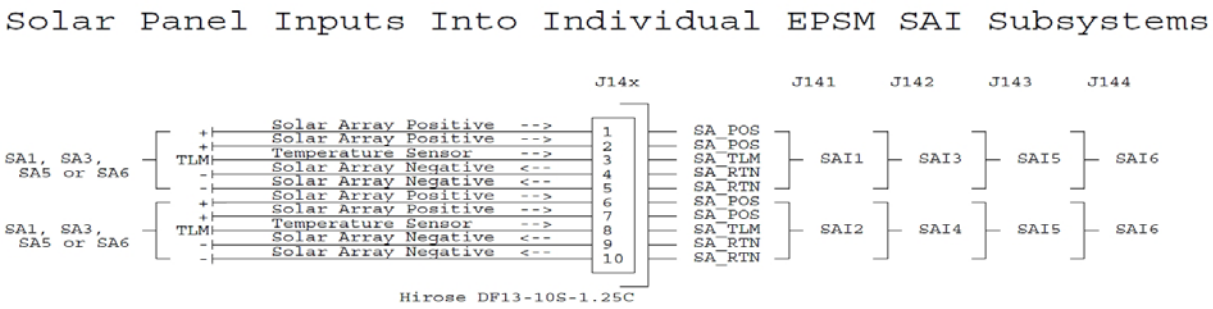

Figure 2: SAI 10-pin connector detail

The 10-pin SAI connectors are split into two 5-pin subsections with identical functionality; SAI1-SAI4 each are mapped to 5 pins, shared on connectors **`J141`** and **`J142`**; SAI5 and SAI6 are each mapped to two pairs of 5 pins on individual connectors **`J143`** and **`J144`**.

Each SAI accepts the positive and negative nodes of a solar panel string, as **`SA_POS`** and **`SA_RTN`**. Additionally, an analog telemetry input is accepted as **`SA_TLM`**.[2]

Warning Each SAI **`SA_TLM`** input is connected to its own +5Vdc pull-up resistor internal to the EPSM1. The voltage on all eight **`SA_TLM`** signals must remain below +3.3Vdc; failure to heed this will result in gross errors in the analog I/O functions (and some additional ones) of the EPSM1.

Tip Unused **`SA_TLM`** inputs must be connected to **`SA_RTN`** on the same SAI.

The mating harness for each SAI connector is to be terminated with a Hirose DF13 10-pin socket, Hirose P/N DF13-10S-1.25C.

Note All of the EPSM1 **`SA_RTN`** nodes are connected to the EPSM's power grounds (**`PGND`**).

[2] SAI5 and SAI6 each have two telemetry inputs (A & B).

Simulating Solar Panels

For each SAI, a solar panels can be simulated via an external power supply and a serial resistor.

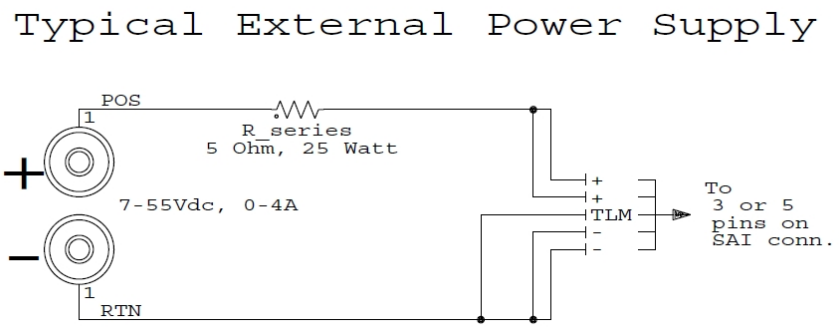

Figure 3: External power supply configured as SAI input

Warning Do not reverse polarity on SAI inputs -- this can lead to immediate damage to the EPSM1.

The series resistor serves to enable confirmation of the EPSM1's MPPT operation.

Tip A single external power supply can feed all six SAIs, so long as a series resistor is used with every connected SAI input.

Alternatively, Pumpkin's Load Board B can be used with a suitable harness. The Load Board B has a built-in trimpots to simulate LM335 temperature sensors.

Connecting Solar Panels

Each SAI accepts a connection from one or more strings of solar cells, and from an optional temperature sensor. Each SAI input has its own MPPT function, that maximizes available power from its connected solar cells.

Note When multiple strings of solar cells are connected to a single SAI, they must all be of the same nominal voltage, and each string must have a blocking diode.

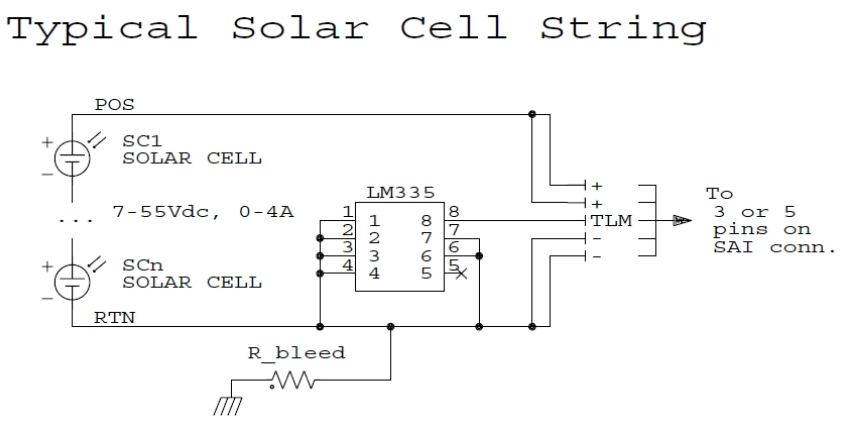

Figure 4: Example solar cell string and temperature sensor, for SAI input

Warning Do not reverse polarity on SAI inputs -- this can lead to immediate damage to the EPSM1.

Example Connection

In Figure 5 below, all six SAIs are being fed by solar panels simulated via an external power supply. In each case, three wires are present at each 5-pin SAI input: **`SA_POS`**, **`SA_TLM`** and **`SA_RTN`**. **`J14x`** pins 2/3/4 and 7/8/9 are used in this example.

Tip In this example, the maximum current into the SAIs is 1A for SAI1-SAI4 (each), 2A for SAI5 and 2A for SAI6.

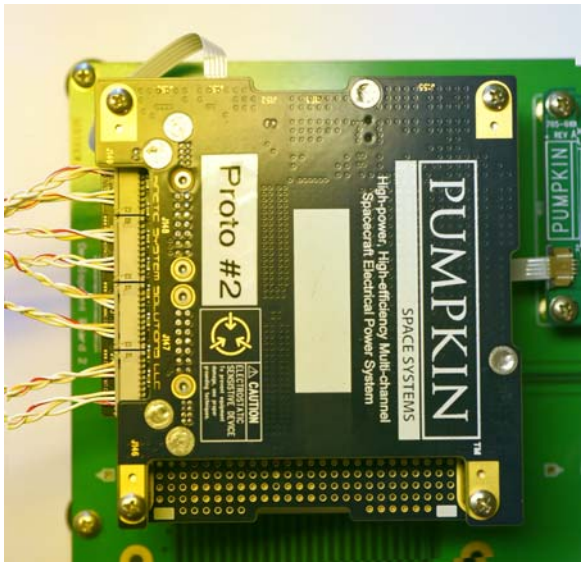

Figure 5: SAIs being fed by external power supply

Power Sequencing

SAIs

Since the SAIs are solely inputs, input power to any of the SAIs can be applied at any time, in any order. From none to all of the SAIs can be powered at any time. In a system that is powered only via the SAIs, removing (all) SAI power will immediately shut down the EPSM1.

Tip Given their (relatively) low current limits and the existing of non-zero source impedances when real solar cells or solar cells simulators are employed, the currents into the EPSM1 via the SAIs are relatively modest, compared to battery and load currents.

Batteries

Warning The batteries can source or sink substantial currents into and out of the EPSM1. Novice EPSM users may wish to current-limit attached batteries with series resistors or the like.

The EPSM1 can be started from batteries alone. If/when there are no active SAI inputs present, all power will be sourced from the batteries, and no battery charging will occur.

When active SAI inputs are present, the batteries will be charged when there is adequate power to power the EPSM1 internals (on the 3V3, 5V0 and 12V buses), and the loads attached to the 3V3, 5V0, 12V and AUX buses, with power left over for battery charging.

Generally speaking, batteries should be the "Last On, First Off" (LIFO) system attached to the EPSM, but this is not required.

Warning Batteries must always be disabled while they are connected to or disconnected from the EPSM1.

Loads

Wherever possible, loads should only be applied to the outputs of the EPSM1 after the EPSM1 has booted and is operational.

EPSM1 Subsystems

Indicators

The EPSM1 has four green status LEDs on the FPGA card, and one green status LED connected to the SupMCU.

The FPGA LEDs are assigned to these functions:

- Innermost: blinks regularly to indicate FPGA reads by the SupMCU
- Others: not implemented

The SupMCU LED follows the usual SupMCU status LED functions.

RBF Inhibit

The RBF inhibit (3-pin DF13 connector) completely disables the entire EPSM1.

Tip This inhibit is ideally suited for use as a "panic switch."

Sep Inhibit

The Separation inhibit (2-pin DF13 connector) disables the FPGA and thus, the inputs and outputs of the EPSM1. The SupMCU continues to operate if/when there are active power sources at the EPSM1's inputs.

SupMCU Terminal

The SupMCU is provided with a terminal that accepts SCPI and other commands via its command-line interface (CLI) and displays lots of useful information. To use the terminal, connect a Pumpkin USB Debug Adapter at 115200,N,8,1.

SupMCU Firmware Version

The SupMCU reports the firmware versions of its bootloader and its main firmware, via its splash screen at startup.

```
Pumpkin(TM) EPSM Rev C bl v1.3.2r at 11.06MHz.
Built on Dec  7 2018 at 09:54:38 with XC16 v1.25 for PIC24EP256MC206.
Listens 5s at 9600 to 230400baud; send 0x55('U') as sync character.

Pumpkin(TM) EPSM Rev C fw v1.5.0d & SupMCU Core v1.2.9a.
Built on Sep 10 2019 at 14:52:03 with XC16 v1.25 for PIC24EP256MC206.
```

Figure 6: Firmware versions in SupMCU splash screen

The firmware versions and build dates are useful to know, when upgrading firmware in the field via Pumpkin's bootloader GUI software.

Tip The **`!splash`** command can be used from any folder location within the CLI, to view the splash screen.

Resets

The EPSM1 normally ignores its external **`-RESET`** signal. This is normal for EPS operation, because the EPSM1 must be able to operate without interference from external reset signals, and because **`-RESET`** is an active-low signal whose value is ill-defined as power rails come up in a distributed system.

The EPSM1's source of reset can be several things, including Power-On Restart (POR), a commanded restart, or a WDT restart. The external behavior of the EPSM1 (in terms of the various output rails becoming active, etc.) is identical in all of these restart conditions.

Restarting

The **`!restart`** command can be used to restart the EPSM1. No additional actions (e.g., activating a separation inhibit) are required -- a restart can be commanded at any time, in any situation. Restarting is a controlled, multi-step process managed by the SupMCU.

The **`sup:res now`** command has the same effect as **`!restart`**, and can be issued over I2C or via the CLI.[3]

[3] SCPI commands are only processed from within the SCPI folder when in the CLI.

FPGA Version and Temperature

The EPSM1 has two temperature sensors ("FPGA" and "System") that are monitored by the EPSM for overtemperature conditions. Their current values are shown in the SupMCU status screen, along with the FPGA version.[4]

```
0)00:00:04:06.43 task_eps_mon_stat: ============================================================
0)00:00:04:06.44 task_eps_mon_stat: FPGA: 15A9/v1.1.3  ON(0)  SYSCON0-3: 0x001C|0000|0000|0007
0)00:00:04:06.47 disp_data_vref:    SR: 0xEB4C  FPGA: +26.9C   System: +35.3C   SupMCU: +27.2C
0)00:00:04:06.52 disp_data_bcrio:     -89.3C +29.5C +11.3C +15.4C-111.3C +18.7C +24.1C +34.2C
0)00:00:04:06.56 task_eps_mon_stat: ============================================================
```

Figure 7: Temperatures in SupMCU status display

OT1 and OT2

The EPSM1 has a user-configurable overtemperature (OT) monitoring system that seeks to protect components by reducing overtemperature conditions should they arise, by shedding loads. The inputs to the OT system are only the FPGA and System temperatures.

As soon as the OT1 setpoint is exceeded, the AUX output is disabled. It remains disabled until the FPGA and System temperatures both drop below the OT1 setpoint by the OT1 hysteresis of 1.5ºC, whereupon it is automatically re-enabled. Messages to this effect will be visible in the CLI when these actions occur.

Tip The assumption associated with OT1 operation is that the major sources of heat generation in the system are the load on the AUX output, and perhaps also the battery blocks. Heat is generated by the end-to-end flow of high currents in the EPSM1 and spreads throughout the EPSM1, where it is measured via the FPGA and System temperature sensors.

When the OT2 setpoint is exceeded, the EPSM1 reboots in a manner identical to what happens when a WDT or other reset occurs. There is no hysteresis associated with OT2.

Note A fan can be used to accelerate cooling of the EPSM1 when in a laboratory test configuration, when investigating OT1 and OT2 settings.

Tip The OT1 and OT2 setpoint values are stored in NVM and can be changed (within certain bounds) by the user.

[4] "SR" stands for the FPGA's scratchpad register.

WDT

The EPSM1 has a user-configurable watchdog timer (WDT). If the WDT is not kicked via an external command or via the CLI within the specified number of seconds, the EPSM1 will restart automatically.

The WDT is normally kicked via an I2C external command from an OBC-type host. The purpose of the EPSM1's WDT is to restart the entire system via an EPSM1 restart (and hence, a full-system power cycling) whenever the system's main computer / OBC is "in the weeds."

Tip For a WDT period of 180s, kicking the WDT every 30s is a good kick rate that is neither too fast, nor too slow.

The WDT cannot be disabled. The WDT value (in seconds) can be set by the user, within certain bounds. A relatively short value (say, 180 seconds) is typically recommended for flight applications. The value should not be so short that it compromises the overall system's ability to successfully boot, nor should it be so long that the system is in danger of not operating properly for a long time.

The remaining number of seconds before the EPSM1 restarts due to a WDT timeout is displayed in the status screen:

```
0)00:00:04:06.90 task_eps_mon_stat: ============================================================
0)00:00:04:06.90 task_eps_mon_stat: Rail   ref (mV)   V (mV)  dV (%)  dose (mGy)   WDT: 64753s
```

Figure 8: Remaining seconds before WDT restart

Tip The WDT value (in seconds) is stored in NVM.

Tip In a configuration where no external device is kicking the EPSM1's WDT, the EPSM1 will always restart once the WDT period expires. To keep the EPSM1 from restarting in these conditions, it may be useful to temporarily set the WDT value to its maximum value of 65535s. This can be done via these commands:

```
epsm:nvm wdt,65535
epsm:wdt kick
```

This will redefine the WDT value to 65,535s and restart the WDT countdown. The EPSM1 will now restart after approximately 18 hours after the most recent kick of the WDT.

At the next EPSM1 (re)-start, the WDT will be redefined to be the NVM value. commanded above, the new WDT value will not be written to NVM unless an NVM write operation is performed by the user.

Solar Arrays

The solar array inputs SAI1-SAI6 are provided on Hirose DF13 connectors.

For ground testing, Pumpkin suggests using a Pumpkin Load Board B configured with 5Ω series resistance, and with the potentiometers simulating connected LM335 temperature sensors. See the EPSM1 datasheet for connector pinouts. With a Pumpkin Load Board B between the EPSM1's Solar Array Inputs (SAIs) and a single- or multi-channel power source, a single dc power supply can feed the entire EPSM1.

The status of the SAIs is shown in the SupMCU status display:

```
0)00:00:59:46.66 task_eps_dummy:   ============================================================
0)00:00:59:46.67 task_eps_dummy:   Block     P (mW)  V (mV)  dV (%)   I (mA)  Imax (mA)  stat
0)00:00:59:46.68 task_eps_dummy:   ------------------------------------------------------------
0)00:00:59:46.70 disp_data_conv:   SAI1:     +4482    23968             +187       +475  0x00
0)00:00:59:46.72 disp_data_conv:   SAI2:     +4855    23920             +203       +475  0x00
0)00:00:59:46.74 disp_data_conv:   SAI3:     +5336    23824             +224       +475  0x00
0)00:00:59:46.75 disp_data_conv:   SAI4:     +4866    23856             +204       +475  0x00
0)00:00:59:46.77 disp_data_conv:   SAI5:     +9204    22896             +402       +474  0x00
0)00:00:59:46.79 disp_data_conv:   SAI6:     +9439    22800             +414       +474  0x00
```

Figure 9: SAI blocks in SupMCU status display

In Figure 9, we see that all six SAIs are being fed from a supply that is in excess of 22Vdc.[5] Additionally, SAI1-SAI4 are drawing ca. 200mA, and SAI5-SAI6 are drawing twice that amount, at

[5] The external supply is set to 25Vdc. The loss is due to 5Ω series resistors between the external power supply and each SAI input.

400mA. But all are under the maximum setting of 475 mA. Power into the EPSM1 as shown as positive power, and in aggregate, the six SAIs are converting approximately 39W of simulated solar power into ring bus energy to be used inside the EPSM1.

Tip Since the SAIs are only sources of power, their currents are always shown as plus/'+'.

Tip Feeding the EPSM1 from a single supply when in development has the advantage of a single place where SAI power into the EPSM1 can be interrupted, and quickly (if required).

Tip Eliminating input power to the EPSM is easy in the case of the SAIs -- just turn off the external power source. This is because the SAIs are strictly inputs to the EPSM1.

Tip The MPPT action of the SAIs seeks to equalize power across all active SAIs. As power through the SAIs climbs, they will eventually match each other.

Tip Total power into the system via the SAIs is limited by / via the individual SAI current limits, and the input voltages at the SAI inputs.

Batteries

Two battery connections are supported as BAT1 and BAT2. They connect to the EPSM1 via 14-pin Harwin M80-series connectors.

Each BAT block is capable of discharging (DSG) or charging (CHG) the connected battery, as conditions require.

Note Only a 4S combination of Li-Ion cells is currently supported (16400mV max).

The status of the BATs is shown in the SupMCU status display:

```
0)00:00:04:06.57 task_eps_mon_stat: Block     P (mW)  V (mV)  dV (%)   I (mA) Is/Im (mA)  stat
[SNIP]
0)00:00:04:06.69 task_eps_mon_stat: - - - - - - - - -
0)00:00:04:06.75 disp_data_conv:    BAT1:    +68574    15728      -4    +4360      +8000  0x00
0)00:00:04:06.76 disp_data_conv:    BAT2:   +114243    15376      -6    +7430      +8000  0x00
```

Figure 10: BAT blocks in SupMCU status display

Tip Current from the EPSM1 into BATs (i.e., CHG mode) is shown as minus/'-'; current out of BATs into the EPSM1 (i.e., DSG mode) is shown as plus/'+'.

The EPSM supports the pass-thru of I2C traffic to each of the two connected batteries. Whatever I2C bus the EPSM is connected to, also provides access to both batteries.

Note EPSM I2C passthrough functions only for I2C masters that connect to system **SCL_SYS** and **SDA_SYS** on the EPSM's 104-pin connector. These passthroughs work only in one direction. An I2C master connected directly to a battery that is connected to the EPSM1 will not be able to read or write to the EPSM1 over I2C.

Output Blocks

Four Output Blocks are provided: 3V3, 5V0, 12V & AUX. They connect to the 104-pin bus connector.

The status of the output blocks is shown in the SupMCU status display:

```
0)00:00:04:06.57 task_eps_mon_stat: Block     P (mW)  V (mV)  dV (%)   I (mA) Is/Im (mA)  stat
[SNIP]
0)00:00:04:06.57 task_eps_mon_stat: ------------------------------------------------------------
0)00:00:04:06.79 disp_data_conv:    3V3:       -445     3300      +0     -135      -5000  0x00
0)00:00:04:06.80 disp_data_conv:    5V0:       -524     4998      +0     -240      -5000  0x00
0)00:00:04:06.82 disp_data_conv:    12V:       -299    11996      +0     -265      -8000  0x00
0)00:00:04:06.84 disp_data_conv:    AUX:    -200670    27968      +0    -7175      -8000  0x00
```

Figure 11: Output blocks in SupMCU status display

Note The 12V block supplies the 5V0 and 3V3 blocks; therefore the total current of the 12V, 5V0 and 3V3 blocks cannot exceed 8A.

Computed Instantaneous Power

The SupMCU status display computes the instantaneous power of the system with each refresh. Four wattages are shown:

- The (source) power into the EPSM1 at the SAIs
- The power into or out of the EPSM1 at the BATs
- The (sink) power out of the EPSM1 at the output blocks
- The sink power subtracted from the source power

The computed instantaneous power is shown in the SupMCU status display:

```
0)00:00:04:06.85 task_eps_mon_stat: - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
0)00:00:04:06.86 task_eps_mon_stat: Total:   +37661  +182817 -201940              =   18538mW
```

Figure 12: Computed instantaneous power in SupMCU status display

In Figure 12, the solar arrays are providing 37.6W of power, 182.8W is being drawn from the batteries, and 201.9W is being consumed by the 3V3, 5V0, 12V and AUX buses. This leaves a computed waste heat of 18.5W in the system, and an overall efficiency of 91%.

FPGA Rails

The FPGA rails are shown in the SupMCU status display:

```
0)00:00:59:46.96 task_eps_dummy:   ============================================================
0)00:00:59:46.97 task_eps_dummy:   Rail   ref (mV)   V (mV)  dV (%)
0)00:00:59:46.98 task_eps_dummy:   ------------------------------------------------------------
0)00:00:59:46.99 disp_data_vref:   VREF1:     1250     1249      +0
0)00:00:59:46.99 disp_data_vref:   VREF2:     1250     1249      +0
0)00:00:59:47.00 disp_data_vref:   VREF3:     1250     1249      +0
0)00:00:59:47.01 disp_data_vref:   VREF4:     1250     1249      +0
0)00:00:59:47.02 disp_data_vref:   VREF5:     1250     1248      +0
0)00:00:59:47.02 disp_data_vref:   h N5V:     5000     5000      +0
0)00:00:59:47.03 disp_data_vref:   c 1V5:     1500     1490      +0
0)00:00:59:47.04 disp_data_vref:   h 3V3:     3300     3263      -1
0)00:00:59:47.05 disp_data_vref:   h 5V0:     5000     4982      +0
0)00:00:59:47.05 disp_data_vref:   h 10V:    10000     9812      -1
0)00:00:59:47.06 disp_data_vref:   Ring1:    32000    35200      +9
0)00:00:59:47.07 disp_data_vref:   Ring2:    32000    35248      +9
0)00:00:59:47.08 disp_data_vref:   Ring3:    32000    35040      +8
```

Figure 13: FPGA rails in SupMCU status display

In Figure 13, the measured value and the desired setpoint value for each rail is shown, along with the percentage deviation.

Tip This region of the SupMCU status display can be used as an "at-a-glance" confirmation that the rails are all operating properly.

Tip The ring bus (rails) will range based on operating conditions. For a highest output of 28Vdc, the ring bus will stabilize at a minimum of 28Vdc+4Vdc once some appreciable current is drawn at the outputs.

Analog Inputs

The analog inputs are shown in the SupMCU status display:

```
0)00:00:59:47.08 task_eps_dummy:   ============================================================
0)00:00:59:47.09 task_eps_dummy:   Function / ANx:    T (C)   [min    V (mV)    max]      ADC
0)00:00:59:47.10 task_eps_dummy:   ------------------------------------------------------------
0)00:00:59:47.12 disp_data_analog: SAI1_IO  / AN0:   +39.37   3122     3125     3173    0xF3C
0)00:00:59:47.13 disp_data_analog: SAI2_IO  / AN1:   +24.71   2971     2978     3025    0xE85
0)00:00:59:47.15 disp_data_analog: SAI3_IO  / AN2:   +36.01   3081     3091     3145    0xF12
0)00:00:59:47.17 disp_data_analog: SAI4_IO  / AN3:   +25.03   2965     2981     3018    0xE89
0)00:00:59:47.18 disp_data_analog: SAI5A_IO / AN6:   +17.34   2865     2904     2911    0xE29
0)00:00:59:47.20 disp_data_analog: SAI5B_IO / AN7:   +17.18   2865     2903     2913    0xE27
0)00:00:59:47.22 disp_data_analog: SAI6A_IO / AN8:   -11.11   2606     2620     2631    0xCC6
0)00:00:59:47.23 disp_data_analog: SAI6B_IO / AN9:   -11.27   2604     2618     2631    0xCC4
0)00:00:59:47.24 disp_data_analog: SNS_VUSB / AN10:          1734     1770     1799    0x2E0
0)00:00:59:47.25 disp_data_analog: -RESET   / AN11:          3271     3277     3289    0xFFA
0)00:00:59:47.26 disp_data_analog: SNS_4096 / AN12:          4096     4096     4097    0x9FC
```

Figure 14: Analog inputs in SupMCU status display

The analog inputs are handled by the SupMCU, and are typically used for temperature sensors on the solar panels, as well as a sanity check on some other reference voltages.

In Figure 14, the analog inputs are shown as either temperatures and/or voltages, with their historic minimum and maximum values.

Warning Any voltage over 3.3Vdc injected into the analog inputs will corrupt all of the analog input values.

EPSM Telecommand Interface

The EPSM normally runs preconfigured and automatically provides managed input power, battery charging, regulated outputs and overcurrent protection functions. Additionally, telemetry can be read from the EPSM, and commands can be written to it.

Telemetry

The EPSM1's various voltages and currents and other telemetry items can be read through the usual telemetry reads supported by the SupMCU. EPSM status values are ready by the SupMCU at around 4Hz; the most recently read data is then returned when the SupMCU is queried for EPSM1 telemetry.

See the SupMCU Firmware Reference Manual for more information.

Commands

Most of the supported commands are listed in the splash screen viewable in the CLI.

```
CLI: "!restart" to restart.
CLI: "cd ..; cd SCPI" to enter command directory.
CLI: "cd ..; cd NVM; list" to display NVM values.
CLI: "cd ..; cd NVM; unlock; write" to write new values to NVM.
CLI: "cd ..; cd NVM; unlock; erase; !restart" to clear NVM values for new firmware release.
CLI: "epsm:bat   {1|2},{on|off|i_max_chg,<value16>}" to control battery block.
CLI: "epsm:bus   {3V3|5V|12V|aux|bat1|bat2},{on|off}" to enable or disable selected block or outputs.
CLI: "epsm:deb   {en|dis},<mask16>" to enable debug strings (0x00FC is most useful mask).
CLI: "epsm:fpga  {flash|init|off|on|lock|unlock}" to program, disable, enable, re-initialize, lock or
                  unlock the FPGA.
CLI: "epsm:fpga  read,<addr16>,XX" to read FPGA register @ <addr> in 0xFFFF format.
CLI: "epsm:fpga  write,<addr16>,D,<value16>" to write FPGA register @ <addr> with <value> (use
                  decimal or 0xFFFF format).
CLI: "epsm:nvm   {3V3|5V|12V|aux},v_set,<value16>" to set bus voltage setpoint.
CLI: "epsm:nvm   {bat1|bat2},{i_max_chg|v_limit},<value16>" to set NVM battery CHG current limits.
CLI: "epsm:nvm   debug,<mask16>" to write debug mask to NVM. Follow with NVM unlock & write!
CLI: "epsm:nvm   {ot1_max|ot2_max},<value16>" to set overtemperature (OT) trip thresholds.
CLI: "epsm:nvm   {sai1|sai2|sai3|sai4|sai5|sai6},{i_max|i_min|v_min},<value16>" to set NVM SAI MPPT
                  current and voltage limits.
CLI: "epsm:nvm   wdt,<value16>" to set NVM WDT timeout.
CLI: "epsm:sai   {1|2|3|4|5|6},{on|off|i_max},<value16>}" to control SAI block. N.B. SAI on requires
                  FPGA unlock.
```

```
CLI: "epsm:wdt   kick" to kick the watchdog timer (WDT).
```

Figure 15: A section of the splash screen listing available commands

---

Note Always test the commands you intend to use, against the firmware release you are running on your EPSM1. The suite of valid commands and telemetry may change from one release to another.

Bus

Each of the six buses (3V3, 5V0, 12V, AUX, BAT1 and BAT2) can be selectively enabled or disabled via the **`epsm:bus {bus_name},{ON|OFF}`** command.

Disabling a bus effectively disconnects the relevant block from the EPSM1's ring bus. The voltage of a disabled bus will fall to 2-3Vdc when there is nothing connected to it. Even a very small load -- on the order of 5mA or less -- can pull that disabled bus voltage to 0Vdc.

Tip To ensure that disabled buses collapse to 0Vdc when disabled, add a small load resistor to the bus in question.

Tip The 3V3, AUX, BAT1 and BAT2 buses can all be individually enabled or disabled. However, disabling the 5V0 bus also disables the 3V3 bus, and disabling the 12V bus disables the 5V0 and 3V3 buses.

Disabling the 12V, 5V0 or 3V3 buses is not recommended and may result in unpredictable behavior.

Understanding EPSM Operations

Ring Bus

The ring bus consists of bulk capacitance that is used as energy storage. The ring bus voltage varies over the operation of the EPSM1, based on the maximum input voltage(s) and minimum output voltage(s). No direct control of the ring bus is afforded the EPSM1 user.

Warning Never draw power from the ring bus. The ring bus is internal to the EPSM1's operation, and is not available to any external users.

An overvoltage protection (OVP) shunt attached to the ring bus prevents excessively high voltages from appearing on the ring bus.

Warning It's important to note that there are leakage paths present between inputs and the ring bus, even when those input blocks are disabled. Therefore do not assume that the ring bus is denergized, even when input blocks are disabled.

Block Operation

Each block also has bulk capacitance -- the capacitance aids in smoothing ripple at the desired voltage. Input blocks boost their input voltage up to the ring bus voltage. Output blocks buck the ring bus voltage down to the desired voltage. Battery blocks boost to or buck from the ring bus based on the battery's requirements.

There is a minimum and a maximum duty cycle that a block can run at. At minimum duty cycle, the block's voltage is near 0Vdc. At maximum, it's near 60Vdc. When a block is disabled, leakage currents will dictate the voltage of the associated bulk capacitance.

Internal Rails

The EPSM1's FPGA and SupMCU operate on a variety of rails that are sourced from the ring bus. They can operate even when all of the individual blocks are disabled. The FPGA and SupMCU switch over from their internal converters to using the rails of the 3V3 and 5V0 blocks when they are available.

Power Deficit Conditions

When the total power drawn by/from/out of the EPSM exceeds the total power into the EPSM, the EPSM is unable to operate properly, and voltage rails will collapse. The loads on the EPSM include loads associated with charging batteries, and loads on the output blocks (3V3, 5V0, 12V and AUX). The internal power required by the EPSM is relatively low (around 3W max).

In a typical power deficit condition, the ring bus voltage collapses to a few volts. Depending on the state of the ring bus, the EPSM1's

SupMCU and FPGA may or may not be operational; hence operation in this condition may not be predictable. For example, preset current limits may not be observed in a power deficit condition, because the deficit is preventing certain EPSM1 circuits (like the ADC converters that sense current) from functioning correctly.

The SupMCU will attempt to detect a power deficit condition, and will automatically attempt to restart the system after a power deficit condition.

Warning Make every effort to avoid operating the EPSM1 in power deficit conditions; prolonged operation may damage the EPSM1.

Dynamics & Limits

A simple way to consider the EPSM1 is "power in equals power out." This holds true and is easily verified as long as none of the current limits are reached.

For example, let's assume six SAI inputs at 25V with 5 Ohm inline resistors simulating six 10S1P strings of triple-junction cells, each limited to 475mA. The maximum power that can be drawn from these six inputs is:

```
(25V - (0.475A x 5Ohms)) x 0.475A = 10.75W per SAI
```

So, with a single power supply set to 25V and 5A, the EPSM1 will never draw more than 2.85A from this supply, and it can draw up to 64.5W from the solar array that the power supply is simulating (the remaining power is lost in the 5Ohm series resistors).

At low currents with a non-zero load, the SAIs will distribute their currents equally between SAI1-SAI4, and SAI5-SAI6 at twice that number. Once the current approaches the preprogrammed current limits of SAI5-SAI6, the currents drawn by SAI1-SAI4 will rise to match the currents in SAI5-SAI6 (within the **`i_max`** preprogrammed limits). For a given, constant power draw of the EPSM1, you can disable an SAI and the other five SAIs will increase their currents (up to the preprogrammed current limit) to maintain a constant power draw through the SAIs.

Now, let's assume you wish to draw power from the AUX output. In this example -- neglecting losses -- you could draw no more than 64.5W (or 2.3A at 28Vdc) from the AUX output, before the

ring bus would be unable to deliver the desired current. If, in this case, you set an AUX current limit to, say, 1.5A, then you could gracefully deny the AUX bus any excess power, yet keep everything else up and running, assuming the availability of 64.5W from the SAIs.

Similarly, if we have 64.5W available and we attach a battery that needs a lot of charging, those 64.5W -- neglecting losses -- will be routed to the battery in an attempt to charge the battery. If a load (e.g., on AUX) is also attached, then the EPSM1 will attempt to deliver to the load, and any remaining current will be used to charge the battery.

Example 1

The EPSM1 is idling, with the attached battery disabled, and no load on AUX:

```
0)00:00:02:16.93 task_eps_dummy:   ============================================================
0)00:00:02:16.95 task_eps_dummy:   FPGA v0.0.7 is ON   SYSCON0-3: 0x0000 0x0000 0x0000 0x0007
0)00:00:02:16.97 disp_data_vref:   T(C)        FPGA: +29.95   System: +31.85
0)00:00:02:16.98 task_eps_dummy:   ============================================================
0)00:00:02:16.99 task_eps_dummy:   Block     P (mW)  V (mV)  dV (%)   I (mA)  Imax (mA)  stat
0)00:00:02:17.00 task_eps_dummy:   ------------------------------------------------------------
0)00:00:02:17.02 disp_data_conv:   SAI1:      +549    24960              +22       +475  0x00
0)00:00:02:17.04 disp_data_conv:   SAI2:      +624    24976              +25       +475  0x00
0)00:00:02:17.06 disp_data_conv:   SAI3:      +548    24944              +22       +475  0x00
0)00:00:02:17.07 disp_data_conv:   SAI4:      +574    24960              +23       +475  0x00
0)00:00:02:17.09 disp_data_conv:   SAI5:     +1044    24864              +42       +474  0x00
0)00:00:02:17.11 disp_data_conv:   SAI6:      +993    24832              +40       +474  0x00
0)00:00:02:17.12 task_eps_dummy:   - - - - - - - -
0)00:00:02:17.14 disp_data_conv:   BAT1:      -589    16832      +0      -35      -2000  0x00
0)00:00:02:17.17 disp_data_conv:   BAT2:      -504    16816      +0      -30      -2000  0x00
0)00:00:02:17.18 task_eps_dummy:   - - - - - - - -
0)00:00:02:17.20 disp_data_conv:   3V3:       -511     3299      +0     -155      -5000  0x00
0)00:00:02:17.21 disp_data_conv:   5V0:       -574     4994      +0     -115      -5000  0x00
0)00:00:02:17.23 disp_data_conv:   12V:       +361    12044      +0      +30      -8000  0x00
0)00:00:02:17.25 disp_data_conv:   AUX:       -420    28032      +0      -15      -3000  0x00
0)00:00:02:17.26 task_eps_dummy:   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
0)00:00:02:17.27 task_eps_dummy:   Total:    +4333    -1093   -1144               =    2096mW
0)00:00:02:17.28 task_eps_dummy:   ============================================================
0)00:00:02:17.29 task_eps_dummy:   Rail   ref (mV)   V (mV)  dV (%)
0)00:00:02:17.30 task_eps_dummy:   ------------------------------------------------------------
0)00:00:02:17.31 disp_data_vref:   VREF1:     1250     1249      +0
0)00:00:02:17.31 disp_data_vref:   VREF2:     1250     1249      +0
0)00:00:02:17.32 disp_data_vref:   VREF3:     1250     1249      +0
0)00:00:02:17.33 disp_data_vref:   VREF4:     1250     1249      +0
0)00:00:02:17.33 disp_data_vref:   VREF5:     1250     1249      +0
0)00:00:02:17.34 disp_data_vref:   h N5V:     5000     4980      +0
0)00:00:02:17.35 disp_data_vref:   c 1V5:     1500     1492      +0
0)00:00:02:17.36 disp_data_vref:   h 3V3:     3300     3261      -1
0)00:00:02:17.37 disp_data_vref:   h 5V0:     5000     4978      +0
0)00:00:02:17.37 disp_data_vref:   h 10V:    10000     9816      -1
0)00:00:02:17.38 disp_data_vref:   Ring1:    32000    40160     +20
0)00:00:02:17.39 disp_data_vref:   Ring2:    32000    40176     +20
0)00:00:02:17.40 disp_data_vref:   Ring3:    32000    40160     +20
0)00:00:02:17.40 task_eps_dummy:   ============================================================
0)00:00:02:17.41 task_eps_dummy:   Function / ANx:    T (C)   [min    V (mV)    max]      ADC
0)00:00:02:17.42 task_eps_dummy:   ------------------------------------------------------------
0)00:00:02:17.44 disp_data_analog: SAI1_IO  / AN0:   +39.18   3077     3123     3123    0xF38
0)00:00:02:17.45 disp_data_analog: SAI2_IO  / AN1:   +25.23   2927     2983     2984    0xE8A
0)00:00:02:17.47 disp_data_analog: SAI3_IO  / AN2:   +35.17   3034     3083     3084    0xF06
0)00:00:02:17.49 disp_data_analog: SAI4_IO  / AN3:   +25.07   2939     2982     2983    0xE88
0)00:00:02:17.50 disp_data_analog: SAI5A_IO / AN6:   +16.97   2821     2901     2901    0xE23
0)00:00:02:17.52 disp_data_analog: SAI5B_IO / AN7:   +17.05   2821     2902     2902    0xE24
0)00:00:02:17.54 disp_data_analog: SAI6A_IO / AN8:   -13.32   2548     2598     2598    0xCA9
0)00:00:02:17.55 disp_data_analog: SAI6B_IO / AN9:   -13.24   2549     2599     2599    0xCAA
0)00:00:02:17.56 disp_data_analog: SNS_VUSB / AN10:           1843     1843     1898    0x2FE
0)00:00:02:17.57 disp_data_analog: -RESET   / AN11:           3276     3282     3286    0xFFE
0)00:00:02:17.58 disp_data_analog: SNS_4096 / AN12:           4096     4097     4097    0x9FB
```

Figure 16: EPSM1 Idling, no batteries, no loads

In Figure 16 we see that the system is idling, with very little power being drawn in from the SAIs, no power from/to the batteries, and

no power to the AUX bus.[6] SAI currents are low (and SAI5-SAI6 are double what SAI1-SAI4 are), there's little waste heat (2W), and the ring bus is at 40V.

This condition will persist indefinitely, until a change in the EPSM1 sources or sinks occurs.

Now, let's add a load to the AUX bus. Since a load is an unambiguous sink of current, the EPSM1 will react by drawing more current from its sources to provide power to the AUX bus. Since the SAIs are the only sources connected to the system, the current drawn through the SAIs will increase by the amount required to balance the new load on the AUX bus.

```
0)00:00:02:05.95 task_eps_dummy:   ============================================================
0)00:00:02:05.97 task_eps_dummy:   FPGA v0.0.7 is ON   SYSCON0-3: 0x0000 0x0000 0x0000 0x0007
0)00:00:02:05.99 disp_data_vref:   T(C)        FPGA: +29.55   System: +31.85
0)00:00:02:06.00 task_eps_dummy:   ============================================================
0)00:00:02:06.01 task_eps_dummy:   Block     P (mW)  V (mV)  dV (%)   I (mA)  Imax (mA)  stat
0)00:00:02:06.02 task_eps_dummy:   ------------------------------------------------------------
0)00:00:02:06.04 disp_data_conv:   SAI1:     +8697    23008             +378       +475  0x00
0)00:00:02:06.05 disp_data_conv:   SAI2:     +8404    23088             +364       +475  0x00
0)00:00:02:06.07 disp_data_conv:   SAI3:     +8798    22912             +384       +475  0x00
0)00:00:02:06.09 disp_data_conv:   SAI4:     +7932    22992             +345       +475  0x00
0)00:00:02:06.11 disp_data_conv:   SAI5:    +10076    22592             +446       +474  0x00
0)00:00:02:06.13 disp_data_conv:   SAI6:    +10265    22512             +456       +474  0x00
0)00:00:02:06.14 task_eps_dummy:   - - - - - - - -
0)00:00:02:06.16 disp_data_conv:   BAT1:      -588    16816      +0      -35      -2000  0x00
0)00:00:02:06.19 disp_data_conv:   BAT2:      -503    16784      +0      -30      -2000  0x00
0)00:00:02:06.20 task_eps_dummy:   - - - - - - - -
0)00:00:02:06.21 disp_data_conv:   3V3:       -528     3300      +0     -160      -5000  0x00
0)00:00:02:06.23 disp_data_conv:   5V0:       -575     5004      +0     -115      -5000  0x00
0)00:00:02:06.25 disp_data_conv:   12V:        +359    11984      +0      +30      -8000  0x00
0)00:00:02:06.27 disp_data_conv:   AUX:     -48578    28080      +0    -1730      -3000  0x00
0)00:00:02:06.27 task_eps_dummy:   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
0)00:00:02:06.28 task_eps_dummy:   Total:   +54173    -1092  -49322              =    3759mW
0)00:00:02:06.30 task_eps_dummy:   ============================================================
0)00:00:02:06.31 task_eps_dummy:   Rail   ref (mV)   V (mV)  dV (%)
0)00:00:02:06.31 task_eps_dummy:   ------------------------------------------------------------
0)00:00:02:06.32 disp_data_vref:   VREF1:     1250     1249      +0
0)00:00:02:06.33 disp_data_vref:   VREF2:     1250     1249      +0
0)00:00:02:06.34 disp_data_vref:   VREF3:     1250     1249      +0
0)00:00:02:06.34 disp_data_vref:   VREF4:     1250     1249      +0
0)00:00:02:06.35 disp_data_vref:   VREF5:     1250     1249      +0
0)00:00:02:06.36 disp_data_vref:   h N5V:     5000     4988      +0
0)00:00:02:06.37 disp_data_vref:   c 1V5:     1500     1492      +0
0)00:00:02:06.37 disp_data_vref:   h 3V3:     3300     3263      -1
0)00:00:02:06.38 disp_data_vref:   h 5V0:     5000     4984      +0
0)00:00:02:06.39 disp_data_vref:   h 10V:    10000     9824      -1
0)00:00:02:06.40 disp_data_vref:   Ring1:    32000    35168      +9
0)00:00:02:06.40 disp_data_vref:   Ring2:    32000    35232      +9
0)00:00:02:06.41 disp_data_vref:   Ring3:    32000    35184      +9
0)00:00:02:06.42 task_eps_dummy:   ============================================================
0)00:00:02:06.43 task_eps_dummy:   Function / ANx:    T (C)   [min    V (mV)    max]      ADC
0)00:00:02:06.44 task_eps_dummy:   ------------------------------------------------------------
0)00:00:02:06.45 disp_data_analog: SAI1_IO  / AN0:   +36.77   3077     3099     3110    0xF1D
0)00:00:02:06.47 disp_data_analog: SAI2_IO  / AN1:   +21.87   2927     2950     2971    0xE63
0)00:00:02:06.49 disp_data_analog: SAI3_IO  / AN2:   +33.17   3034     3063     3069    0xEF0
0)00:00:02:06.50 disp_data_analog: SAI4_IO  / AN3:   +22.99   2939     2961     2969    0xE71
0)00:00:02:06.52 disp_data_analog: SAI5A_IO / AN6:   +12.66   2821     2858     2879    0xDF0
0)00:00:02:06.54 disp_data_analog: SAI5B_IO / AN7:   +12.74   2821     2858     2888    0xDF1
0)00:00:02:06.55 disp_data_analog: SAI6A_IO / AN8:   -15.38   2548     2577     2579    0xC92
0)00:00:02:06.57 disp_data_analog: SAI6B_IO / AN9:   -15.46   2549     2576     2580    0xC91
0)00:00:02:06.58 disp_data_analog: SNS_VUSB / AN10:          1849     1849     1898    0x301
0)00:00:02:06.59 disp_data_analog: -RESET   / AN11:          3276     3277     3286    0xFFC
0)00:00:02:06.60 disp_data_analog: SNS_4096 / AN12:          4096     4096     4097    0x9FD
```

Figure 17: EPSM1 powering a load, no batteries

In Figure 17 we see that the power drawn from the SAIs has jumped from 4.3W when idling, to 54.2W now. There has been no change with the batteries, and the AUX bus is now drawing 1.7A for 49.3W. The losses have increased from 2.1W to 3.6W, for an increase of 1.5W for an increase of power through the system of 51W.

6 This particular EPSM 1 has not been calibrated, hence some currents will report greater than zero but are effectively zero.

Now that the output bus(ses) are loaded, the ring bus voltage has fallen to 35Vdc. This makes for more efficient conversion.

There are no other substantial changes in the system; power is simply flowing from the SAIs, into the ring bus, and out the AUX bus. But, at higher currents than we saw before.

Now, let's enable a battery on BAT1 while leaving the load on the AUX bus. Since there is already another load on the EPSM1, we'll either see not much of a change (if the battery is fully charged at 16800mV), or we'll see the EPSM1 start charging the battery while also supplying power to the AUX bus.

```
0)00:00:01:58.62 task_eps_dummy:   ============================================================
0)00:00:01:58.64 task_eps_dummy:   FPGA v0.0.7 is ON   SYSCON0-3: 0x0002 0x0000 0x0000 0x0007
0)00:00:01:58.67 disp_data_vref:   T(C)        FPGA: +29.35   System: +31.55
0)00:00:01:58.68 task_eps_dummy:   ============================================================
0)00:00:01:58.68 task_eps_dummy:   Block     P (mW)  V (mV)  dV (%)   I (mA)  Imax (mA)  stat
0)00:00:01:58.69 task_eps_dummy:   ------------------------------------------------------------
0)00:00:01:58.71 disp_data_conv:   SAI1:    +10580    22368             +473       +475  0x00
0)00:00:01:58.73 disp_data_conv:   SAI2:    +10744    22384             +480       +475  0x00
0)00:00:01:58.75 disp_data_conv:   SAI3:    +10594    22352             +474       +475  0x00
0)00:00:01:58.77 disp_data_conv:   SAI4:    +10587    22384             +473       +475  0x00
0)00:00:01:58.78 disp_data_conv:   SAI5:    +10767    22432             +480       +474  0x00
0)00:00:01:58.80 disp_data_conv:   SAI6:    +10363    22432             +462       +474  0x00
0)00:00:01:58.81 task_eps_dummy:   - - - - - - - -
0)00:00:01:58.84 disp_data_conv:   BAT1:     -9704    16448      -2     -590      -2000  0x00
0)00:00:01:58.86 disp_data_conv:   BAT2:      -492    16400      -2      -30      -2000  0x00
0)00:00:01:58.87 task_eps_dummy:   - - - - - - - -
0)00:00:01:58.89 disp_data_conv:   3V3:       -528     3300      +0     -160      -5000  0x00
0)00:00:01:58.91 disp_data_conv:   5V0:       -575     5000      +0     -115      -5000  0x00
0)00:00:01:58.92 disp_data_conv:   12V:        +359    11992      +0      +30      -8000  0x00
0)00:00:01:58.94 disp_data_conv:   AUX:     -48382    28048      +0    -1725      -3000  0x00
0)00:00:01:58.95 task_eps_dummy:   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
0)00:00:01:58.96 task_eps_dummy:   Total:   +63637   -10196  -49126               =    4315mW
0)00:00:01:58.97 task_eps_dummy:   ============================================================
0)00:00:01:58.98 task_eps_dummy:   Rail   ref (mV)   V (mV)  dV (%)
0)00:00:01:58.99 task_eps_dummy:   ------------------------------------------------------------
0)00:00:01:59.00 disp_data_vref:   VREF1:     1250     1249      +0
0)00:00:01:59.01 disp_data_vref:   VREF2:     1250     1249      +0
0)00:00:01:59.01 disp_data_vref:   VREF3:     1250     1249      +0
0)00:00:01:59.02 disp_data_vref:   VREF4:     1250     1249      +0
0)00:00:01:59.03 disp_data_vref:   VREF5:     1250     1249      +0
0)00:00:01:59.04 disp_data_vref:   h N5V:     5000     4988      +0
0)00:00:01:59.04 disp_data_vref:   c 1V5:     1500     1492      +0
0)00:00:01:59.05 disp_data_vref:   h 3V3:     3300     3264      -1
0)00:00:01:59.06 disp_data_vref:   h 5V0:     5000     4986      +0
0)00:00:01:59.06 disp_data_vref:   h 10V:    10000     9824      -1
0)00:00:01:59.07 disp_data_vref:   Ring1:    32000    34224      +6
0)00:00:01:59.08 disp_data_vref:   Ring2:    32000    31680      -1
0)00:00:01:59.09 disp_data_vref:   Ring3:    32000    32816      +2
0)00:00:01:59.10 task_eps_dummy:   ============================================================
0)00:00:01:59.10 task_eps_dummy:   Function / ANx:    T (C)   [min    V (mV)    max]      ADC
0)00:00:01:59.11 task_eps_dummy:   ------------------------------------------------------------
0)00:00:01:59.13 disp_data_analog: SAI1_IO  / AN0:   +36.93   3077     3100     3110    0xF19
0)00:00:01:59.15 disp_data_analog: SAI2_IO  / AN1:   +22.34   2927     2954     2971    0xE63
0)00:00:01:59.16 disp_data_analog: SAI3_IO  / AN2:   +33.32   3034     3064     3069    0xEEC
0)00:00:01:59.18 disp_data_analog: SAI4_IO  / AN3:   +23.13   2939     2962     2967    0xE6D
0)00:00:01:59.20 disp_data_analog: SAI5A_IO / AN6:   +11.66   2821     2848     2879    0xDDE
0)00:00:01:59.21 disp_data_analog: SAI5B_IO / AN7:   +11.90   2821     2850     2888    0xDE1
0)00:00:01:59.23 disp_data_analog: SAI6A_IO / AN8:   -15.61   2548     2575     2579    0xC8A
0)00:00:01:59.25 disp_data_analog: SAI6B_IO / AN9:   -15.85   2549     2573     2580    0xC87
0)00:00:01:59.25 disp_data_analog: SNS_VUSB / AN10:          1851     1853     1898    0x302
0)00:00:01:59.26 disp_data_analog: -RESET   / AN11:          3276     3281     3286    0xFFA
0)00:00:01:59.27 disp_data_analog: SNS_4096 / AN12:          4096     4097     4097    0x9F9
```

Figure 18: EPSM powering a load with battery

In Figure 18 shows that in addition to powering the 1.7A load on AUX, the EPSM1 is also charging the battery to the tune of a 600mA charge current (denoted as -590, a current that is being sourced from the EPS). This additional 10W of charging power had to come from somewhere; we see that the power from the SAIs has increased by 10W.

Additionally, we see that all six SAIs are running at their specified maximum current of 475mA. While SAI5 and SAI6 were close to

their maximums in Figure 17, SAI1-SAI4 were not. The additional load required by battery charging has been spread equally across all six SAIs in this case.

Note also that the ring bus voltage has fallen further, in its asymptotic approach towards 32Vdc.

Example 2

We begin this example with the EPSM1 idling, powered solely from BAT1, and with no load on AUX:

```
0)00:00:06:19.56 task_eps_dummy:   ============================================================
0)00:00:06:19.58 task_eps_dummy:   FPGA v0.0.7 is ON   SYSCON0-3: 0x0010 0x0100 0x0000 0x0007
0)00:00:06:19.60 disp_data_vref:   SP: 0xEB4C  FPGA: +24.7C   System: +26.3C
0)00:00:06:19.61 task_eps_dummy:   ============================================================
0)00:00:06:19.62 task_eps_dummy:   Block     P (mW)  V (mV)  dV (%)   I (mA)  Imax (mA)  stat
0)00:00:06:19.62 task_eps_dummy:   ------------------------------------------------------------
0)00:00:06:19.64 disp_data_conv:   SAI1:         +0       16              -35       +475  0x00
0)00:00:06:19.66 disp_data_conv:   SAI2:         +0       16               -6       +475  0x00
0)00:00:06:19.68 disp_data_conv:   SAI3:         +0       16               -5       +475  0x00
0)00:00:06:19.70 disp_data_conv:   SAI4:         +0       32               -1       +475  0x00
0)00:00:06:19.71 disp_data_conv:   SAI5:         +0       32               -2       +474  0x00
0)00:00:06:19.73 disp_data_conv:   SAI6:         +0       16               +0       +474  0x00
0)00:00:06:19.74 task_eps_dummy:   - - - - - - - -
0)00:00:06:19.76 disp_data_conv:   BAT1:     +1923    16032      -4     +120      +2000  0x00
0)00:00:06:19.79 disp_data_conv:   BAT2:       -45     1504   -1017      -30      -2000  0x00
0)00:00:06:19.79 task_eps_dummy:   - - - - - - - -
0)00:00:06:19.81 disp_data_conv:   3V3:       -495     3302      +0     -150      -5000  0x00
0)00:00:06:19.83 disp_data_conv:   5V0:       -475     5000      +0      -95      -5000  0x00
0)00:00:06:19.85 disp_data_conv:   12V:         +59    11988      +0       +5      -8000  0x00
0)00:00:06:19.86 disp_data_conv:   AUX:       -116     3328    -744      -35      -2000  0x01
0)00:00:06:19.87 task_eps_dummy:   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
0)00:00:06:19.88 task_eps_dummy:   Total:        +0    +1878   -1026               =     852mW
0)00:00:06:19.89 task_eps_dummy:   ============================================================
0)00:00:06:19.90 task_eps_dummy:   Rail   ref (mV)   V (mV)  dV (%)
0)00:00:06:19.91 task_eps_dummy:   ------------------------------------------------------------
0)00:00:06:19.92 disp_data_vref:   VREF1:     1250     1250      +0
0)00:00:06:19.93 disp_data_vref:   VREF2:     1250     1250      +0
0)00:00:06:19.93 disp_data_vref:   VREF3:     1250     1250      +0
0)00:00:06:19.94 disp_data_vref:   VREF4:     1250     1250      +0
0)00:00:06:19.95 disp_data_vref:   VREF5:     1250     1249      +0
0)00:00:06:19.96 disp_data_vref:   h N5V:     5000     4980      +0
0)00:00:06:19.96 disp_data_vref:   c 1V5:     1500     1499      +0
0)00:00:06:19.97 disp_data_vref:   h 3V3:     3300     3266      -1
0)00:00:06:19.98 disp_data_vref:   h 5V0:     5000     4974      +0
0)00:00:06:19.99 disp_data_vref:   h 10V:    10000     9804      -1
0)00:00:06:19.99 disp_data_vref:   Ring1:    32000    33696      +5
0)00:00:06:20.00 disp_data_vref:   Ring2:    32000    32976      +2
0)00:00:06:20.01 disp_data_vref:   Ring3:    32000    34736      +7
0)00:00:06:20.02 task_eps_dummy:   ============================================================
0)00:00:06:20.03 task_eps_dummy:   Function / ANx:    T (C)   [min    V (mV)    max]      ADC
0)00:00:06:20.03 task_eps_dummy:   ------------------------------------------------------------
0)00:00:06:20.05 disp_data_analog: SAI1_IO  / AN0:    +31.7   3005     3048     3058    0xED9
0)00:00:06:20.06 disp_data_analog: SAI2_IO  / AN1:    +17.8   2839     2909     2923    0xE2C
0)00:00:06:20.08 disp_data_analog: SAI3_IO  / AN2:    +27.4   2986     3005     3021    0xEA4
0)00:00:06:20.09 disp_data_analog: SAI4_IO  / AN3:    +17.8   2895     2909     2924    0xE2C
0)00:00:06:20.11 disp_data_analog: SAI5A_IO / AN6:     +9.7   2798     2828     2840    0xDC7
0)00:00:06:20.12 disp_data_analog: SAI5B_IO / AN7:     +9.8   2794     2829     2841    0xDC8
0)00:00:06:20.13 disp_data_analog: SAI6A_IO / AN8:    -19.7   2510     2534     2546    0xC58
0)00:00:06:20.15 disp_data_analog: SAI6B_IO / AN9:    -20.0   2513     2531     2545    0xC55
0)00:00:06:20.16 disp_data_analog: SNS_VUSB / AN10:            280      284     1869    0x076
0)00:00:06:20.17 disp_data_analog: -RESET   / AN11:           3276     3280     3287    0xFFB
0)00:00:06:20.18 disp_data_analog: SNS_4096 / AN12:           4096     4096     4097    0x9FA
```

Figure 19: EPSM1 Idling, BAT1, no SAIs, no loads

In Figure 19 we see that the system is idling, with 120mA drawn from BAT1, no power from the SAIs, and no power to the AUX bus. There's little waste heat (852mW), and the ring bus is at 34V.

Now, let's add a 500mA load to the AUX bus, as shown in Figure 20. Current from BAT1 (the only source of power currently available) jumps from 120mA to 1050mA to service the new 15W load on AUX. The battery voltage has dropped by about 1V compared to its fully-charged voltage of 16.8V. Waste heat has barely changed, and the ring bus voltage has fallen by around 1V.

```
0)00:00:13:17.45 task_eps_dummy:   ============================================================
0)00:00:13:17.47 task_eps_dummy:   FPGA v0.0.7 is ON   SYSCON0-3: 0x0012 0x0000 0x0000 0x0007
0)00:00:13:17.49 disp_data_vref:   SP: 0xEB4C  FPGA: +25.3C   System: +28.0C
0)00:00:13:17.50 task_eps_dummy:   ============================================================
0)00:00:13:17.51 task_eps_dummy:   Block     P (mW)  V (mV)  dV (%)   I (mA)  Imax (mA)  stat
0)00:00:13:17.52 task_eps_dummy:   ------------------------------------------------------------
0)00:00:13:17.54 disp_data_conv:   SAI1:         +0       16                -3       +475  0x00
0)00:00:13:17.55 disp_data_conv:   SAI2:         +0       16                -6       +475  0x00
0)00:00:13:17.57 disp_data_conv:   SAI3:         +0       32                -5       +475  0x00
0)00:00:13:17.59 disp_data_conv:   SAI4:         +0       32                -1       +475  0x00
0)00:00:13:17.61 disp_data_conv:   SAI5:         +0       32                -2       +474  0x00
0)00:00:13:17.63 disp_data_conv:   SAI6:         +0       32                +0       +474  0x00
0)00:00:13:17.64 task_eps_dummy:   - - - - - - - -
0)00:00:13:17.65 disp_data_conv:   BAT1:    +16632    15840      -6    +1050      +2000  0x00
0)00:00:13:17.68 disp_data_conv:   BAT2:        -45     1504   -1017       -30      -2000  0x00
0)00:00:13:17.69 task_eps_dummy:   - - - - - - - -
0)00:00:13:17.70 disp_data_conv:   3V3:        -494     3298      +0     -150      -5000  0x00
0)00:00:13:17.72 disp_data_conv:   5V0:        -474     4998      +0      -95      -5000  0x00
0)00:00:13:17.74 disp_data_conv:   12V:         +59    11984      +0       +5      -8000  0x00
0)00:00:13:17.76 disp_data_conv:   AUX:     -14882    28080      +0     -530      -2000  0x00
0)00:00:13:17.77 task_eps_dummy:   - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
0)00:00:13:17.78 task_eps_dummy:   Total:        +0   +16586  -15791                =     795mW
0)00:00:13:17.79 task_eps_dummy:   ============================================================
0)00:00:13:17.80 task_eps_dummy:   Rail   ref (mV)   V (mV)  dV (%)
0)00:00:13:17.81 task_eps_dummy:   ------------------------------------------------------------
0)00:00:13:17.82 disp_data_vref:   VREF1:     1250     1250      +0
0)00:00:13:17.82 disp_data_vref:   VREF2:     1250     1250      +0
0)00:00:13:17.83 disp_data_vref:   VREF3:     1250     1250      +0
0)00:00:13:17.84 disp_data_vref:   VREF4:     1250     1250      +0
0)00:00:13:17.84 disp_data_vref:   VREF5:     1250     1249      +0
0)00:00:13:17.85 disp_data_vref:   h N5V:     5000     4980      +0
0)00:00:13:17.86 disp_data_vref:   c 1V5:     1500     1499      +0
0)00:00:13:17.87 disp_data_vref:   h 3V3:     3300     3264      -1
0)00:00:13:17.87 disp_data_vref:   h 5V0:     5000     4972      +0
0)00:00:13:17.88 disp_data_vref:   h 10V:    10000     9796      -2
0)00:00:13:17.89 disp_data_vref:   Ring1:    32000    32048      +0
0)00:00:13:17.90 disp_data_vref:   Ring2:    32000    32096      +0
0)00:00:13:17.90 disp_data_vref:   Ring3:    32000    31552      -1
0)00:00:13:17.91 task_eps_dummy:   ============================================================
0)00:00:13:17.92 task_eps_dummy:   Function / ANx:    T (C)   [min    V (mV)    max]      ADC
0)00:00:13:17.93 task_eps_dummy:   ------------------------------------------------------------
0)00:00:13:17.94 disp_data_analog: SAI1_IO  / AN0:    +31.7   3005     3048     3058    0xED8
0)00:00:13:17.96 disp_data_analog: SAI2_IO  / AN1:    +18.1   2839     2912     2923    0xE2E
0)00:00:13:17.97 disp_data_analog: SAI3_IO  / AN2:    +27.7   2986     3008     3021    0xEA6
0)00:00:13:17.99 disp_data_analog: SAI4_IO  / AN3:    +18.2   2895     2913     2924    0xE30
0)00:00:13:18.00 disp_data_analog: SAI5A_IO / AN6:     +9.7   2798     2828     2840    0xDC6
0)00:00:13:18.01 disp_data_analog: SAI5B_IO / AN7:     +9.9   2794     2830     2841    0xDC8
0)00:00:13:18.03 disp_data_analog: SAI6A_IO / AN8:    -19.4   2510     2537     2546    0xC5B
0)00:00:13:18.04 disp_data_analog: SAI6B_IO / AN9:    -19.8   2513     2533     2545    0xC56
0)00:00:13:18.05 disp_data_analog: SNS_VUSB / AN10:            280      284     1869    0x076
0)00:00:13:18.06 disp_data_analog: -RESET   / AN11:           3272     3277     3287    0xFF6
0)00:00:13:18.07 disp_data_analog: SNS_4096 / AN12:           4096     4097     4097    0x9F9
```

Figure 20: EPSM1 Idling, BAT1, no SAIs, 500mA AUX load

Tip In this example, the maximum discharge current from the 16.8V BAT1 battery is set at 2000mA. In Figure 20, by inspecting the power values in the **`P (mW)`** column, we see that most of the power associated with BAT1's 1050mA current is flowing out of the AUX output at 28V. In this condition, the current drawn by the load(s) connected to AUX can roughly double to 1000mA before we encounter a power deficit condition.

Now we'll simulate coming out of eclipse, by providing power via the SAIs, at 18V:

```
0)00:00:21:09.93 task_eps_dummy:   ============================================================
0)00:00:21:09.95 task_eps_dummy:   FPGA v0.0.7 is ON   SYSCON0-3: 0x0012 0x0000 0x0000 0x0007
0)00:00:21:09.97 disp_data_vref:   SP: 0xEB4C  FPGA: +26.1C   System: +29.3C
0)00:00:21:09.98 task_eps_dummy:   ============================================================
0)00:00:21:09.99 task_eps_dummy:   Block     P (mW)  V (mV)  dV (%)   I (mA)  Imax (mA)  stat
0)00:00:21:10.00 task_eps_dummy:   ------------------------------------------------------------
0)00:00:21:10.02 disp_data_conv:   SAI1:     +7028    15312             +459       +475  0x00
0)00:00:21:10.04 disp_data_conv:   SAI2:     +7141    15392             +464       +475  0x00
0)00:00:21:10.06 disp_data_conv:   SAI3:     +7326    15200             +482       +475  0x00
0)00:00:21:10.07 disp_data_conv:   SAI4:     +6934    15376             +451       +475  0x00
0)00:00:21:10.09 disp_data_conv:   SAI5:     +7387    15200             +486       +474  0x00
0)00:00:21:10.11 disp_data_conv:   SAI6:     +7394    15216             +486       +474  0x00
0)00:00:21:10.12 task_eps_dummy:   - - - - - - - -
0)00:00:21:10.15 disp_data_conv:   BAT1:    -23500    16096      -4    -1460      -6000  0x00
0)00:00:21:10.17 disp_data_conv:   BAT2:      -659    16496      -1      -40      -2000  0x00
0)00:00:21:10.18 task_eps_dummy:   - - - - - - - -
0)00:00:21:10.20 disp_data_conv:   3V3:        -494     3299      +0     -150      -5000  0x00
0)00:00:21:10.21 disp_data_conv:   5V0:        -375     5002      +0      -75      -5000  0x00
0)00:00:21:10.23 disp_data_conv:   12V:         +59    11992      +0       +5      -8000  0x00
0)00:00:21:10.25 disp_data_conv:   AUX:     -14890    28096      +0     -530      -2000  0x00
0)00:00:21:10.26 task_eps_dummy:   - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
0)00:00:21:10.27 task_eps_dummy:   Total:   +43213   -24160  -15700                =    3353mW
0)00:00:21:10.28 task_eps_dummy:   ============================================================
0)00:00:21:10.29 task_eps_dummy:   Rail   ref (mV)   V (mV)  dV (%)
```

```
0)00:00:21:10.30 task_eps_dummy:   ------------------------------------------------------------
0)00:00:21:10.31 disp_data_vref:   VREF1:     1250     1250      +0
0)00:00:21:10.32 disp_data_vref:   VREF2:     1250     1250      +0
0)00:00:21:10.32 disp_data_vref:   VREF3:     1250     1250      +0
0)00:00:21:10.33 disp_data_vref:   VREF4:     1250     1250      +0
0)00:00:21:10.34 disp_data_vref:   VREF5:     1250     1249      +0
0)00:00:21:10.35 disp_data_vref:   h N5V:     5000     4984      +0
0)00:00:21:10.35 disp_data_vref:   c 1V5:     1500     1499      +0
0)00:00:21:10.36 disp_data_vref:   h 3V3:     3300     3264      -1
0)00:00:21:10.37 disp_data_vref:   h 5V0:     5000     4978      +0
0)00:00:21:10.37 disp_data_vref:   h 10V:    10000     9808      -1
0)00:00:21:10.38 disp_data_vref:   Ring1:    32000    32672      +2
0)00:00:21:10.39 disp_data_vref:   Ring2:    32000    32128      +0
0)00:00:21:10.40 disp_data_vref:   Ring3:    32000    32624      +1
0)00:00:21:10.41 task_eps_dummy:   ============================================================
0)00:00:21:10.41 task_eps_dummy:   Function / ANx:    T (C)   [min    V (mV)    max]      ADC
0)00:00:21:10.42 task_eps_dummy:   ------------------------------------------------------------
0)00:00:21:10.44 disp_data_analog: SAI1_IO  / AN0:    +27.5   3005     3006     3058    0xEA2
0)00:00:21:10.45 disp_data_analog: SAI2_IO  / AN1:    +11.2   2839     2843     2923    0xDD7
0)00:00:21:10.47 disp_data_analog: SAI3_IO  / AN2:    +26.0   2986     2991     3021    0xE8F
0)00:00:21:10.48 disp_data_analog: SAI4_IO  / AN3:    +16.9   2895     2900     2924    0xE1E
0)00:00:21:10.49 disp_data_analog: SAI5A_IO / AN6:     +7.3   2798     2804     2840    0xDA6
0)00:00:21:10.51 disp_data_analog: SAI5B_IO / AN7:     +7.1   2794     2802     2841    0xDA4
0)00:00:21:10.52 disp_data_analog: SAI6A_IO / AN8:    -21.6   2510     2515     2549    0xC3E
0)00:00:21:10.54 disp_data_analog: SAI6B_IO / AN9:    -21.3   2511     2518     2547    0xC42
0)00:00:21:10.55 disp_data_analog: SNS_VUSB / AN10:            280     1864     1869    0x306
0)00:00:21:10.56 disp_data_analog: -RESET   / AN11:           3272     3284     3287    0xFFC
0)00:00:21:10.57 disp_data_analog: SNS_4096 / AN12:           4096     4096     4097    0x9F8
0)00:00:21:13.58 task_eps_dummy:   ============================================================
```

Figure 21: EPSM1 charging BAT1 with small AUX load, SAI voltage = 18V

In Figure 21 we see that the power delivered to AUX has not changed, but now BAT1 is being charged, and is drawing 1460mA. All six of the SAIs are running at or near their maximum currents, and SAI power is shared equally across all six SAIs. With all of the SAIs running and the battery charging, waste heat has increased substantially. The ring bus voltage has not changed much.

In Figure 21, the SAI input voltage is 18V.[7] Let's increase that voltage to 28V, and observe what happens:

```
0)00:00:21:24.51 task_eps_dummy:   ============================================================
0)00:00:21:24.53 task_eps_dummy:   FPGA v0.0.7 is ON   SYSCON0-3: 0x0010 0x0000 0x0000 0x0007
0)00:00:21:24.55 disp_data_vref:   SP: 0xEB4C  FPGA: +26.6C   System: +30.8C
0)00:00:21:24.56 task_eps_dummy:   ============================================================
0)00:00:21:24.57 task_eps_dummy:   Block     P (mW)  V (mV)  dV (%)   I (mA)  Imax (mA)  stat
0)00:00:21:24.58 task_eps_dummy:   ------------------------------------------------------------
0)00:00:21:24.59 disp_data_conv:   SAI1:    +12136    25232             +481       +475  0x00
0)00:00:21:24.61 disp_data_conv:   SAI2:    +11487    25472             +451       +475  0x00
0)00:00:21:24.63 disp_data_conv:   SAI3:    +12136    25232             +481       +475  0x00
0)00:00:21:24.65 disp_data_conv:   SAI4:    +12187    25232             +483       +475  0x00
0)00:00:21:24.67 disp_data_conv:   SAI5:    +11982    25280             +474       +474  0x00
0)00:00:21:24.69 disp_data_conv:   SAI6:    +12040    25296             +476       +474  0x00
0)00:00:21:24.69 task_eps_dummy:   - - - - - - - -
0)00:00:21:24.72 disp_data_conv:   BAT1:    -51131    16336      -2    -3130      -6000  0x00
0)00:00:21:24.75 disp_data_conv:   BAT2:      -572    16352      -2      -35      -2000  0x00
0)00:00:21:24.76 task_eps_dummy:   - - - - - - - -
0)00:00:21:24.77 disp_data_conv:   3V3:       -494     3299      +0     -150      -5000  0x00
0)00:00:21:24.79 disp_data_conv:   5V0:       -375     5000      +0      -75      -5000  0x00
0)00:00:21:24.81 disp_data_conv:   12V:        +60    12008      +0       +5      -8000  0x00
0)00:00:21:24.83 disp_data_conv:   AUX:     -14873    28064      +0     -530      -2000  0x00
0)00:00:21:24.83 task_eps_dummy:   - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
0)00:00:21:24.84 task_eps_dummy:   Total:   +71971   -51704  -15683               =    4584mW
0)00:00:21:24.86 task_eps_dummy:   ============================================================
0)00:00:21:24.86 task_eps_dummy:   Rail   ref (mV)   V (mV)  dV (%)
0)00:00:21:24.87 task_eps_dummy:   ------------------------------------------------------------
0)00:00:21:24.88 disp_data_vref:   VREF1:     1250     1250      +0
0)00:00:21:24.89 disp_data_vref:   VREF2:     1250     1250      +0
0)00:00:21:24.90 disp_data_vref:   VREF3:     1250     1250      +0
0)00:00:21:24.90 disp_data_vref:   VREF4:     1250     1250      +0
0)00:00:21:24.91 disp_data_vref:   VREF5:     1250     1249      +0
0)00:00:21:24.92 disp_data_vref:   h N5V:     5000     4988      +0
0)00:00:21:24.93 disp_data_vref:   c 1V5:     1500     1499      +0
0)00:00:21:24.93 disp_data_vref:   h 3V3:     3300     3266      -1
0)00:00:21:24.94 disp_data_vref:   h 5V0:     5000     4978      +0
0)00:00:21:24.95 disp_data_vref:   h 10V:    10000     9808      -1
0)00:00:21:24.96 disp_data_vref:   Ring1:    32000    33344      +4
0)00:00:21:24.96 disp_data_vref:   Ring2:    32000    33344      +4
0)00:00:21:24.97 disp_data_vref:   Ring3:    32000    32832      +2
0)00:00:21:24.98 task_eps_dummy:   ============================================================
0)00:00:21:24.99 task_eps_dummy:   Function / ANx:    T (C)   [min    V (mV)    max]      ADC
0)00:00:21:25.00 task_eps_dummy:   ------------------------------------------------------------
0)00:00:21:25.01 disp_data_analog: SAI1_IO  / AN0:    +28.2   3005     3013     3058    0xEAC
```

[7] In this setup, the SAIs are all being fed from a single power supply, each through an inline 5Ω resistor. 475mA x 5Ω = 2.375V, roughly the difference between 18V and the voltages reported by the SAIs.

```
0)00:00:21:25.03 disp_data_analog: SAI2_IO  / AN1:    +11.4   2839     2845     2923    0xDDB
0)00:00:21:25.04 disp_data_analog: SAI3_IO  / AN2:    +26.1   2985     2992     3021    0xE92
0)00:00:21:25.06 disp_data_analog: SAI4_IO  / AN3:    +17.0   2892     2901     2924    0xE20
0)00:00:21:25.07 disp_data_analog: SAI5A_IO / AN6:     +7.3   2798     2804     2840    0xDA7
0)00:00:21:25.08 disp_data_analog: SAI5B_IO / AN7:     +7.6   2794     2807     2841    0xDAB
0)00:00:21:25.10 disp_data_analog: SAI6A_IO / AN8:    -21.2   2510     2519     2549    0xC44
0)00:00:21:25.11 disp_data_analog: SAI6B_IO / AN9:    -21.1   2511     2520     2547    0xC46
0)00:00:21:25.12 disp_data_analog: SNS_VUSB / AN10:            280     1856     1869    0x303
0)00:00:21:25.13 disp_data_analog: -RESET   / AN11:           3272     3282     3287    0xFFC
0)00:00:21:25.14 disp_data_analog: SNS_4096 / AN12:           4096     4097     4097    0x9F9
```

Figure 22: EPSM1 charging BAT1 with small AUX load, SAI voltage = 28V

In Figure 22 we see a substantial increase in BAT1 charging current, due to greater voltages (and hence, power) in the SAIs. With the SAI voltage increased from 18V to 28V, the BAT1 charging current doubled, though it was still substantially below the maximum allowed (6000mA). Apart from an increase in waste heat (due to more power flowing through the SAIs), there are no other substantive changes to accompany this increase in SAI voltage.

Tip In this example, the charging current doubled despite an increase of only 56% of SAI voltage. How did that happen? There are six SAIs -- not one -- and battery charging current is a function of SAI power, not SAI voltages or currents alone. In fact, the charging current in this example is still limited by the (relatively low) programmed SAI current limits.

This example illustrates how battery charging is dependent on available SAI power. Both SAIs and BATs power the EPSM1 itself, and any attached loads. The EPSM will balance overall power flow such that any available power left over after powering the 3V3, 5V, 12V and AUX buses, will be used for battery charging.

Output Block Fuses

Each output block (3V3, 5V0, 12V and AUX) has a dynamic current-vs-voltage curve to ensure a graceful response to overcurrent conditions. When the preset trip current is exceeded, the voltage falls gracefully in a roughly constant-power situation. In this condition, a fuse bit is set (and can be queried). Once the overcurrent condition removed from the output, the output voltage will revert to its preset value.

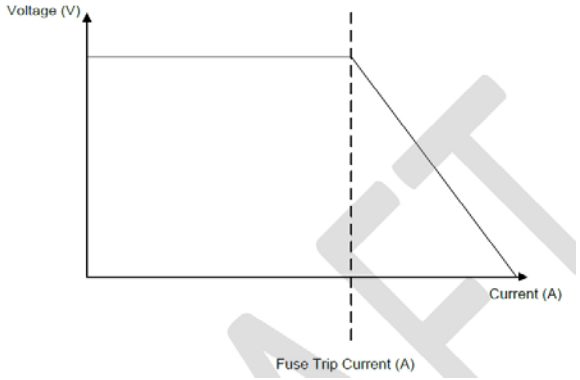

Figure 23: IV curve for output blocks

In the EPSM status screens below, this can be demonstrated.

In Figure 24, a current of 1035mA is being drawn from the AUX output (configured for 28Vdc at 1000mA). Because the trip current has been exceeded, the AUX output has dropped to 1.3V. Note also that the AUX block is still enabled (**`stat:b0`** is 0), and its fuse bit is set (**`stat:b1`** is 1). This condition will persist until the load on the AUX output drops below the trip current of 1000mA.

```
0)00:00:06:59.15 task_eps_dummy:   ============================================================
0)00:00:06:59.17 task_eps_dummy:   FPGA v0.0.7 is ON   SYSCON0-3: 0x0000 0x0000 0x0001 0x0007
0)00:00:06:59.19 disp_data_vref:   SP: 0xEB4C  FPGA: +30.45C  System: +32.35C
0)00:00:06:59.20 task_eps_dummy:   ============================================================
0)00:00:06:59.21 task_eps_dummy:   Block     P (mW)  V (mV)  dV (%)   I (mA)  Imax (mA)  stat
0)00:00:06:59.22 task_eps_dummy:   ------------------------------------------------------------
0)00:00:06:59.24 disp_data_conv:   SAI1:      +642    19456              +33       +475  0x00
0)00:00:06:59.26 disp_data_conv:   SAI2:      +739    19456              +38       +475  0x00
0)00:00:06:59.27 disp_data_conv:   SAI3:      +680    19440              +35       +475  0x00
0)00:00:06:59.29 disp_data_conv:   SAI4:      +623    19488              +32       +475  0x00
0)00:00:06:59.31 disp_data_conv:   SAI5:     +1387    19264              +72       +474  0x00
0)00:00:06:59.33 disp_data_conv:   SAI6:     +1389    19296              +72       +474  0x00
0)00:00:06:59.34 task_eps_dummy:   - - - - - - - -
0)00:00:06:59.36 disp_data_conv:   BAT1:      -420    16800      +0      -25      -2000  0x00
0)00:00:06:59.39 disp_data_conv:   BAT2:      -670    16768      +0      -40      -2000  0x00
0)00:00:06:59.40 task_eps_dummy:   - - - - - - - -
0)00:00:06:59.41 disp_data_conv:   3V3:       -511     3300      +0     -155      -5000  0x00
0)00:00:06:59.43 disp_data_conv:   5V0:       -399     4996      +0      -80      -5000  0x00
0)00:00:06:59.45 disp_data_conv:   12V:       +119    11992      +0      +10      -8000  0x00
0)00:00:06:59.47 disp_data_conv:   AUX:      -1374     1328   -2015    -1035      -1000  0x02
0)00:00:06:59.48 task_eps_dummy:   - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
0)00:00:06:59.49 task_eps_dummy:   Total:    +5461    -1090   -2165              =    2206mW
```

Figure 24: AUX output in overcurrent condition

In Figure 25, the load on AUX was reduced to 950mA, the AUX output has (automatically) recovered from the trip condition, and the fuse has been reset. The AUX output remained enabled through all of this example.

```
0)00:00:10:38.98 task_eps_dummy:   ============================================================
0)00:00:10:39.00 task_eps_dummy:   FPGA v0.0.7 is ON   SYSCON0-3: 0x0000 0x0000 0x0000 0x0007
0)00:00:10:39.03 disp_data_vref:   SP: 0xEB4C  FPGA: +31.45C  System: +33.05C
0)00:00:10:39.04 task_eps_dummy:   ============================================================
0)00:00:10:39.05 task_eps_dummy:   Block     P (mW)  V (mV)  dV (%)   I (mA)  Imax (mA)  stat
0)00:00:10:39.06 task_eps_dummy:   ------------------------------------------------------------
0)00:00:10:39.07 disp_data_conv:   SAI1:     +3875    18368             +211       +475  0x00
0)00:00:10:39.09 disp_data_conv:   SAI2:     +3634    18448             +197       +475  0x00
0)00:00:10:39.11 disp_data_conv:   SAI3:     +3927    18352             +214       +475  0x00
0)00:00:10:39.13 disp_data_conv:   SAI4:     +3890    18352             +212       +475  0x00
0)00:00:10:39.15 disp_data_conv:   SAI5:     +7806    17120             +456       +474  0x00
0)00:00:10:39.16 disp_data_conv:   SAI6:     +7904    17184             +460       +474  0x00
0)00:00:10:39.17 task_eps_dummy:   - - - - - - - -
0)00:00:10:39.20 disp_data_conv:   BAT1:      -420    16816      +0      -25      -2000  0x00
0)00:00:10:39.23 disp_data_conv:   BAT2:      -671    16784      +0      -40      -2000  0x00
0)00:00:10:39.24 task_eps_dummy:   - - - - - - - -
0)00:00:10:39.25 disp_data_conv:   3V3:       -495     3300      +0     -150      -5000  0x00
0)00:00:10:39.27 disp_data_conv:   5V0:       -374     4998      +0      -75      -5000  0x00
0)00:00:10:39.29 disp_data_conv:   12V:       +119    11988      +0      +10      -8000  0x00
0)00:00:10:39.30 disp_data_conv:   AUX:     -26535    28080      +0     -945      -1000  0x00
0)00:00:10:39.31 task_eps_dummy:   - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
0)00:00:10:39.32 task_eps_dummy:   Total:   +31039    -1091  -27285              =    2663mW
```

Figure 25: AUX output after overcurrent condition removed

Of note is the behavior in these conditions if/when the AUX block is disabled. In Figure 26, with the load on the AUX output and the AUX block disabled (**`stat:b0`** is 1), the AUX output has dropped to 0V (due to the continued presence of the load), and the fuse is (still) set (**`stat:b1`** is 1). The fuse will automatically be cleared when the AUX block is (re-)enabled AND the current through it is less than the trip current.

```
0)00:00:07:28.46 task_eps_dummy:   ============================================================
0)00:00:07:28.48 task_eps_dummy:   FPGA v0.0.7 is ON   SYSCON0-3: 0x0002 0x0100 0x0001 0x0007
0)00:00:07:28.50 disp_data_vref:   SP: 0xEB4C  FPGA: +30.35C  System: +31.85C
0)00:00:07:28.51 task_eps_dummy:   ============================================================
0)00:00:07:28.52 task_eps_dummy:   Block     P (mW)  V (mV)  dV (%)   I (mA)  Imax (mA)  stat
0)00:00:07:28.53 task_eps_dummy:   ------------------------------------------------------------
0)00:00:07:28.55 disp_data_conv:   SAI1:      +448    19520             +23       +475  0x00
0)00:00:07:28.57 disp_data_conv:   SAI2:      +410    19536             +21       +475  0x00
0)00:00:07:28.58 disp_data_conv:   SAI3:      +429    19536             +22       +475  0x00
0)00:00:07:28.60 disp_data_conv:   SAI4:      +449    19552             +23       +475  0x00
0)00:00:07:28.62 disp_data_conv:   SAI5:      +777    19440             +40       +474  0x00
0)00:00:07:28.64 disp_data_conv:   SAI6:      +855    19440             +44       +474  0x00
0)00:00:07:28.65 task_eps_dummy:   - - - - - - - -
0)00:00:07:28.67 disp_data_conv:   BAT1:      -420    16832      +0      -25      -2000  0x00
0)00:00:07:28.70 disp_data_conv:   BAT2:      -671    16784      +0      -40      -2000  0x00
0)00:00:07:28.71 task_eps_dummy:   - - - - - - - -
0)00:00:07:28.72 disp_data_conv:   3V3:       -494     3299      +0     -150      -5000  0x00
0)00:00:07:28.74 disp_data_conv:   5V0:       -374     4996      +0      -75      -5000  0x00
0)00:00:07:28.76 disp_data_conv:   12V:       +119    11980      +0      +10      -8000  0x00
0)00:00:07:28.78 disp_data_conv:   AUX:         +0        0      +1      -40      -1000  0x03
0)00:00:07:28.79 task_eps_dummy:   - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
0)00:00:07:28.80 task_eps_dummy:   Total:    +3371    -1092    -749              =    1530mW
```

Figure 26: AUX block disabled, after fuse trip event