



Cambrionix PowerSync Series

Terminal Command Reference for Cambrionix PowerSync Chargers

1 Introduction

The guide describes how to remotely control devices in the PowerSync Series (PS4) via their control interface. Remote control permits the PowerSync Series to be integrated into a larger system that is controlled by a host computer.

On the PowerSync series, command is achieved via the host USB type-C socket. The charger appears as a virtual serial port (also called a UART or VCP). On Microsoft Windows, the charger will appear as a COM port. On macOS, a device file is created in the `/dev/` directory. This is of the form `/dev/tty.usbserial-S` where `S` is an alpha-numeric serial string unique to each device in the PowerSync Series.

Devices in the PowerSync Series are hereinafter collectively referred to as “PS4”.

Commands that are issued to the serial port are referred to as *terminal commands*.

The settings modified by commands in this document are volatile – that is, the settings are lost when the PS4 is rebooted or powered off.

This document is subject to change. Any parsing of data should be tolerant of new features being added.

2 Prerequisites

FTDI Drivers

PS4 devices incorporate a FT230X USB to UART converter IC from FTDI International. On Windows 7 or later, a driver may automatically be installed (if Windows is configured to download drivers from the internet automatically). If this is not the case, the driver may be downloaded from www.ftdichip.com. The VCP drivers are required. For Linux or Macintosh computers the default OS drivers should be used.

UART Settings

The default communications settings should be set to 115200 baud, 8 data bits, no parity, and 1 stop bit. This is sometimes referred to as 115200,8,N,1. No flow control is used. ANSI terminal emulation should be selected. Lines sent by the PS4 are terminated with `<CR><LF>`. Lines received by the PS4 should be terminated with at least `<CR>`. `<LF>` is ignored.

The PS4 will accept back-to-back bytes, however, the host computer should wait for a fresh command prompt from the PS4 before issuing a new command string.

Boot text and command prompt

At boot, the PS4 will issue a string of ANSI escape sequences to reset an attached terminal emulator. This is followed by the title block, then a command prompt.

The command prompt is always `>>`, followed by `<CR><LF>`, except in boot mode where it is `boot>>` followed by `<CR><LF>`.

To reach a new command prompt, send CTRL-C. This cancels any partially entered command string.

3 Command Summary

Command	Description
mode <m> [p]	Set mode <m> for port [p] or all ports
mode c <p>	Set charge mode for a port <p>
qcmode <q> <p>	Set quick charge mode <q> for port <p>
state [p]	Show state for port [p] or all ports
system	Show hardware and firmware information
health	Show voltages, temperature, errors and boot flag
cef	Clear error flags
sef [flags]...	Set error flags
crf	Clear rebooted flag
limits	Show voltage and temperature limits
power [pwr [p]]	Show power totals or set board max to [pwr], or just port [p] in mW
detail [p]	Show detailed state for port [p] or all ports
host [auto off]	Show if USB host is present, and set mode change
id	Show id string
bd	Board description
logp <s>	Report V and mA for each port every <s> seconds
loge [s]	Report state for all ports every [s] seconds and events as they occur
l	Live view (periodically updated screen showing system state)
remote [exit kexit]	Enter or exit mode where console is controlled by terminal
ledb <p> <row> <ptn>	Set individual LED on row <row>, port <p> to flash bit pattern <ptn>
leds <row> [ptnstr]	Set flash pattern of a string of LEDs on row <row> to string <ptnstr>
cls	Clear terminal screen
reboot [watchdog]	Reboot (optionally by using watchdog timeout)

Notes

Throughout this text, compulsory parameters to terminal commands are shown in triangular brackets: < >. Optional parameters are shown in square brackets: [].

Text as it appears in the serial data stream is shown in the `Courier New` typeface.

4 id command

In order to easily identify the product that is providing the serial port being used for communication, there is the id command. The id command also provides some basic information about the firmware.

Syntax: `id`

Response:

A single line of text containing multiple name:value pairs that can be used to identify the product.

Name	Value
mfr	Manufacturer string eg cambrix
mode	A string to describe which operating mode the firmware is in eg main
hw	A short string naming the hardware eg PS4
hwid	A hexadecimal value used internally to identify the board eg 0x13
fw	A pseudo number representing the firmware revision eg 1.68
bl	A pseudo number representing the bootloader revision eg 0.15
sn	A serial number, currently not used, eg 000000
group	Used on some products to order firmware updates which is useful when updating boards that are daisy-chained together so that down-stream boards are updated and rebooted first.
fc	Firmware Code is used to denote which firmware the board accepts

Example

```
>> id
mfr:cambrix,mode:main,hw:PowerSync 4,hwid:0x00, fw:0.28,bl:0.20,sn:000000,group:-,fc:ps
```

5 bd command

The bd command provides a description of the internal connections to the ports of the board. This includes all the charging ports as well as the virtual serial port and any expansion ports that may be present. This is to allow software to be able to navigate the USB connection tree without needing to have in-built specific knowledge of how this board was designed.

Syntax: bd

Response:

Some name value pairs indicating the presence or not of the optional features of the board. This is followed by a description of each USB hub in turn, listing what is attached to each port of that hub. Each port of a hub will be attached to a charging port, an expansion port, a downstream hub, a USB serial device or is unused.

The optional features are indicated by these entries:

Name	Value
Ports	The number of charging ports the product provides
Sync	A '1' indicates the product provides sync capability
Temp	A '1' indicates the product can measure temperature
EXTPSU	A '1' indicates the product is supplied with a voltage that is greater than 5.2V, generally 12V. This is usually provided by a laptop style PSU.

The attachment section can have the following entries, all indices are 1 based:

Name	Value	Description
Nodes	<n>	A number indicating the number of nodes this description set includes. A node will be either a USB hub or a USB controller.
Node <i> Type	<type>	<i> is an index indicating which node this is. <type> is an entry from the Node Type table below.
Node <i> Ports	<n>	A number indicating how many ports this node has. USB hubs generally have 4 or 7 ports.
Hub <i> Port <p>	Hub <j>	The USB hub <i> has a down-stream hub <j> connected to its port <p>
Hub <i> Port <p>	Control Port	The USB hub <i> has the USB serial port attached to port <p>
Hub <i> Port <p>	Expansion Port <e>	The USB hub <i> has an expansion port attached to port <p>
Hub <i> Port <p>	Port <c>	The USB hub <i> has the charging port <c> attached to port <p>
Hub <i> Port <p>	Optional Hub <j>	The USB hub <i> may have a down-stream hub <j> connected to its port <p> but this is optional so may not be fitted
Hub <i> Port <p>	Turbo Hub <j>	The USB hub <i> has a USB hub capable of operating in Turbo mode attached to port <p>
Hub <i> Port <p>	USB3 Hub <j>	The USB hub <i> has a USB 3.x hub attached to port <p>
Hub <i> Port <p>	Unused Port	The USB hub <i> has nothing attached to its port <p>

Node type can be one of the following:

Node Type	Description
Hub <j>	A USB 2.0 hub index <j>
Optional Hub <j>	A USB hub that may be fitted, index <j>
Root <r>	A USB controller with a root hub which also means the USB bus number will change
Turbo Hub <j>	A USB hub capable of operating in Turbo mode with index <j>
USB3 Hub <j>	A USB 3.x hub with index <j>

Example

```
>> bd
Ports: 4
Sync: 1
Temp: 1
EXTPSU: 1
Nodes : 2
Node 1 Type : Hub 1
Node 1 Ports : 4
Hub 1 Port 1 : Hub 2
Hub 1 Port 2 : Expansion port 1
Hub 1 Port 3 : Control Port
Hub 1 Port 4 : Expansion port 2
Node 2 Type : Hub 2
Node 2 Ports : 4
Hub 2 Port 1 : Port 1
Hub 2 Port 2 : Port 2
Hub 2 Port 3 : Port 3
Hub 2 Port 4 : Port 4

>>
```

Parsing of the output should be flexible as new fields may be added in future.

6 mode command

Modes and states

Each port on the PS4 has different *modes* it can be placed in. When placed in a particular mode, the port will transition from one *state* to another according to whether a mobile device is attached, and whether that device is charging or not.

The mode command

Each port can be placed into one of four modes by using the `mode` command.

Syntax: `mode <m> [p]`

<m> is one of the following mode characters:

Mode character <m>	Mode	Description
c	Charge	The port is readied for charging a device, and can detect if a device is attached or detached.
o	Off	Power (VBUS) to the port is removed. No charging occurs. No device attach or detach detection is possible.

The port parameter, [p], is optional. It can be used to specify the port number. If left blank, all ports are affected by the command.

Response:

None (new command prompt appears)

Examples

To turn off all ports:

```
>> mode o
```

To put just port 2 in charge mode:

```
>> mode c 2
```

7 state command

After a port is placed into a particular mode (e.g. charge mode) it can transition into a number of states. The `state` command is used to list the state of each port. It also shows the voltage and current being delivered to the mobile device and any error flags.

Syntax: `state [p]`

Where [p] is the port number.

Response:

Comma separated fields, one row per port.

Row format: p, voltage 10mV, current mA, flags, time charging, time charged, energy

Field	Description	
p	The port number pertaining to the row	
voltage 10mV	Voltage being delivered to the mobile device, in 10mV	
current mA	Current being delivered to the mobile device, in mA (milliamperes)	
flags	List of case-sensitive flag characters, separated by spaces. Flags may mean different things in different columns	
	1st column	Description
	"A"	Device is ATTACHED to this port
	"D"	No device is attached to this port. Port is DETACHED
	"P"	Port has established a PD contract with device
	"C"	Cable has non-type-C connector at far end, no device detected
	2nd column	
	"I"	Port is IDLE
	"S"	Port is the host port and is connected
	"C"	Port is CHARGING
	"F"	Port has FINISHED charging
	"O"	Port is in OFF mode
	"c"	Power is enabled on port but no device is detected
	3rd column	
	"_"	Quick charge mode is disallowed
	"+"	Quick charge mode is allowed but not enabled
"q"	Quick charge mode is enabled but not in use	
"Q"	Quick charge mode is in use	
time charging	Time in seconds the port has been charging for	
time charged	Time in seconds that the port has been charged for (x means not valid yet).	
energy	Energy the device has consumed in watthours (calculated every second)	

Note : Current measurement is typically has a resolution of 9.76mA or 11.18mA depending on the product.

Examples

Default state of PS4, with no attached devices (D) and all ports in charge mode, but idle (I):

```
>> state
0, 0507, 0000, A S -, 0, x, 0.00
1, 0000, 0000, D I -, 0, x, 0.00
2, 0000, 0000, D I -, 0, x, 0.00
3, 0000, 0000, D I -, 0, x, 0.00
4, 0000, 0000, D I -, 0, x, 0.00
```

A PD capable tablet connected to port 2, which is charging at 19.99V, 1466mA.

```
>> state
0, 0506, 0000, A S -, 0, x, 0.00
1, 0000, 0000, D I -, 0, x, 0.00
2, 1999, 1466, P C -, 14, x, 0.06
3, 0000, 0000, D I -, 0, x, 0.00
4, 0000, 0000, D I -, 0, x, 0.00
```

Whilst the above tablet was charging, another mobile device was attached to port 3 which is not PD capable.

```
>> state
0, 0507, 0000, A S -, 0, x, 0.00
1, 0000, 0000, D I -, 0, x, 0.00
2, 2002, 1412, P C -, 77, x, 0.56
3, 0503, 0306, A I -, 12, x, 0.01
4, 0000, 0000, D I -, 0, x, 0.00
```

The device on port 3 has had quick charge mode enabled but is not using it

```
>> state
0, 0506, 0000, A S -, 0, x, 0.00
1, 0000, 0000, D I -, 0, x, 0.00
2, 2000, 0940, P C -, 750, x, 4.82
3, 0512, 1458, A I q, 685, x, 1.73
4, 0000, 0000, D I -, 0, x, 0.00
```

The device on port 3 is now using quick charge mode

```
>> state
0, 0507, 0000, A S -, 0, x, 0.00
1, 0000, 0000, D I -, 0, x, 0.00
2, 1999, 0939, P C -, 758, x, 4.86
3, 0894, 0601, A I Q, 693, x, 1.95
4, 0000, 0000, D I -, 0, x, 0.00
```

Showing just the state of port 3:

```
>> state 3
3, 0501, 0957, A I -, 33, x, 0.03
```


8 Viewing system parameters

To view system-wide parameters, issue the `system` command.

Syntax:

```
system
```

Response:

First row: system title text, beginning with “Cambrionix”

Second and subsequent rows: parameter:value pairs, one pair per row

Parameter	Description	Possible values
Hardware	Name of PowerSync device. More names may be added in future.	PowerSync 4 4 Port Charger, 24V supply
Firmware	Firmware version string	Typically in “n.nn” format, where is decimal 0..9 Strings are permitted (e.g. “1.09-Alpha”) for some versions
Compiled	Date and Time of the Firmware	
Group	Group letter read from PCB jumpers	1 character, 16 values: “-”, “A” .. “O” A dash (“-”) means no group jumper is fitted
Panel ID	Panel ID number of front panel board	“None” if no panel was detected Otherwise “0” .. “15”

Example

```
>> system
cambrionix PowerSync 4 4 Port USB Charge+Sync
Hardware: PowerSync 4
Firmware: 0.28
Compiled: Jan 21 2019 14:24:06
Group: -
Panel ID: Absent
```

Notes

- The system title text may change across firmware releases.
- The “Group” parameter is updated every time the `system` command is run.
- The “Panel ID” is updated at power-up or reboot.

9 System health

The health command displays the supply rail voltages, PCB temperature, error flags and the rebooted flag.

Syntax:

```
health
```

Response:

parameter:value pairs, one pair per row.

Parameter	Description	Possible values	
Input Now	Present input supply voltage		
Input Min	Lowest input supply voltage seen		
Input Max	Highest input supply voltage seen		
Input Flags	List of input supply rail error flags, separated by spaces		No flags: voltage is acceptable
		"UV"	Under-voltage event occurred
		"OV"	Over-voltage event occurred
Temperature Now	Present PCB temperature, °C	">100 C"	Temperature is above 100°C
		"<0.0 C"	Temperature is below 0°C
		"tt.t C"	Temperature, e.g. "32.2 C"
Temperature Max	Highest PCB temperature seen, °C	">100 C"	Temperature is above 100°C
		"<0.0 C"	Temperature is below 0°C
		"tt.t C"	Temperature, e.g. "32.2 C"
Temperature Flags	Temperature error flags		No flags: temperature is acceptable
		"OT"	Over-temperature (over-heat) event occurred
Rebooted Flag	Used to detect if system has booted or rebooted	"R"	System has booted or rebooted
			Flag cleared using <code>crf</code> command

Examples

A normal, healthy system. The rebooted flag has been cleared using the `crf` command:

```
>> health
System up for:      1631 seconds
Input Now: 25.54
Input Min: 25.22
Input Max: 25.55
Input Flags:
Temperature Now (C): 25.1
Temperature Max (C): 30.9
Temperature Flags:
Rebooted flag:
```

A system with all error flags showing. In addition, the boot flag has not been cleared:

```
>> health
System up for:      135 seconds
Input Now: 23.84
Input Min: 9.42
Input Max: 27.75
Input Flags: UV OV
Temperature Now (C): 27.4
Temperature Max (C): 78.9
Temperature Flags: OT
Rebooted flag: R
```

10 Clearing the error flags

The OV, UV, and OT flags are latched 'on', and can be cleared using the `cef` command ("clear error flags"). If the error condition persists, the PS4 will set the flag again after it is cleared.

11 Setting the error flags

It can be useful to set the error flags to examine the system behaviour when an error occurs. To do this, use the `sef` command ("set error flags")

Syntax:

```
sef <flags>
```

where <flags> is a list of one or more flags:

Flag	Function
12UV	Input rail under-voltage
12OV	Input rail over-voltage
OT	PCB over-temperature

Example

To set the 12UV and OT flags:

```
>> sef 12UV OT
```

Notes

1. Calling `sef` without parameters is valid, and sets no error flags.

12 Clearing the rebooted (R) flag

The boot flag can be cleared using the `crf` command (“clear rebooted flag”). This is useful to observe if the PS4 has rebooted in between commands. To do this, clear the `crf` flag at the start of a controlling script. Then poll it along with the other flags using the `health` or `state` commands. If the R flag is found to be set, then previous settings (such as profile selection, or port modes) will have been lost, and will need to be re-sent to the PS4.

13 Displaying system limits

To show the limits (thresholds) at which the under-voltage, over-voltage and over-temperature errors are triggered, issue the `limits` command.

Example

```
>> limits
Input Min: 9.59
Input Max: 27.00
Temperature (C): 75.0
```

Notes

- The limits are hard-coded into the firmware and cannot be changed by a terminal command.
- The supply rails are sampled every 1ms. They must be over or under voltage for 20ms before a flag is raised.
- The temperature is measured every 10ms. Batches of 32 samples are averaged to give the result.
- If the 5v Rail is sampled twice in a row below 4.5v the ports are shutoff

14 Boot mode

Boot mode is used to modify the firmware within the PS4. It is not documented here. However, if you find the system in boot mode (where the command prompt is `boot>>`), you can return to normal operation by issuing the `reboot` command or by power-cycling the system.

15 Host detection

The PS4 looks at the HOST USB socket for a VBUS supply from an attached host computer.

The `host` command can be used to reveal if a host computer is attached or not. It can also be used to prevent the PS4 from automatically changing modes.

Syntax: `host [mode]`

Mode string [mode]	Description
<code>auto</code>	The ports will enable sync connectivity as the host comes and goes. Charging is always enabled unless the port is turned off.
<code>off</code>	If VBUS is removed from the host port, all charging ports will be turned off.

Response (if no parameter is supplied):

Present: `<yes|no>`

Mode change: `<auto|off>`

Response (if parameter is supplied)

None (new command prompt appears)

Examples

To set host mode to off:

```
>> host off
```

To see if a host is present, and the reveal the mode:

```
>> host  
Present: yes  
Mode change: auto
```

16 Logging port current and voltage

The logp command is used to display the current and voltage for all ports at a preset time interval.

Syntax:

```
logp [seconds]          (seconds is in range 1..32767)
```

Response:

CSV (comma separated values), which can be copied into a spreadsheet or grapher program.

Example

Here is a device being attached to port 2, left for 26 seconds, and then removed:

```
>> logp 1
Logging seconds, mA, V, degrees C with period (mins:secs): 00:01
Press CTRL-C to stop

000000, 0000, 0507, 0000, 0000, 0000, 0000, 0000, 0000, 0000, 0500, 25.1
000001, 0000, 0507, 0000, 0000, 0000, 0000, 0000, 0000, 0000, 0499, 24.3
000002, 0000, 0507, 0000, 0000, 0000, 0000, 0000, 0000, 0000, 0500, 24.7
000003, 0000, 0507, 0000, 0000, 0000, 0000, 0000, 0000, 0000, 0500, 24.6
000004, 0000, 0507, 0000, 0000, 0000, 0000, 0000, 0000, 0000, 0500, 24.8
000005, 0000, 0507, 0000, 0000, 0000, 0155, 0000, 0000, 0000, 0499, 24.7
000006, 0000, 0507, 0000, 0000, 0031, 1998, 0000, 0000, 0000, 0499, 25.0
000007, 0000, 0508, 0000, 0000, 0000, 2000, 0000, 0000, 0000, 0500, 24.9
000008, 0000, 0507, 0000, 0000, 0003, 2002, 0000, 0000, 0000, 0499, 25.1
000009, 0000, 0507, 0000, 0000, 0043, 1998, 0000, 0000, 0000, 0500, 24.6
000010, 0000, 0507, 0000, 0000, 0171, 2002, 0000, 0000, 0000, 0501, 24.8
000011, 0000, 0507, 0000, 0000, 0284, 2003, 0000, 0000, 0000, 0500, 25.0
000012, 0000, 0507, 0000, 0000, 0403, 2001, 0000, 0000, 0000, 0499, 24.8
000013, 0000, 0507, 0000, 0000, 0519, 2001, 0000, 0000, 0000, 0500, 24.8
000014, 0000, 0506, 0000, 0000, 0640, 2001, 0000, 0000, 0000, 0500, 24.6
000015, 0000, 0507, 0000, 0000, 0761, 2001, 0000, 0000, 0000, 0500, 24.7
000016, 0000, 0507, 0000, 0000, 0883, 2001, 0000, 0000, 0000, 0500, 25.1
000017, 0000, 0507, 0000, 0000, 1005, 2000, 0000, 0000, 0000, 0500, 24.6
000018, 0000, 0507, 0000, 0000, 1009, 2000, 0000, 0000, 0000, 0499, 24.7
000019, 0000, 0507, 0000, 0000, 1094, 2000, 0000, 0000, 0000, 0500, 24.6
000020, 0000, 0507, 0000, 0000, 1152, 2000, 0000, 0000, 0000, 0500, 24.6
000021, 0000, 0507, 0000, 0000, 1177, 2001, 0000, 0000, 0000, 0500, 24.3
000022, 0000, 0507, 0000, 0000, 1200, 2000, 0000, 0000, 0000, 0500, 24.7
000023, 0000, 0507, 0000, 0000, 1223, 2000, 0000, 0000, 0000, 0499, 24.9
000024, 0000, 0507, 0000, 0000, 1217, 2000, 0000, 0000, 0000, 0500, 25.1
000025, 0000, 0507, 0000, 0000, 1215, 2000, 0000, 0000, 0000, 0500, 24.6
000026, 0000, 0507, 0000, 0000, 1208, 2000, 0000, 0000, 0000, 0499, 25.0
000027, 0000, 0507, 0000, 0000, 1205, 2000, 0000, 0000, 0000, 0499, 25.0
000028, 0000, 0507, 0000, 0000, 1203, 2001, 0000, 0000, 0000, 0500, 24.7
000029, 0000, 0507, 0000, 0000, 1198, 2001, 0000, 0000, 0000, 0500, 24.4
000030, 0000, 0507, 0000, 0000, 1194, 2001, 0000, 0000, 0000, 0499, 24.7
000031, 0000, 0507, 0000, 0000, 1189, 2001, 0000, 0000, 0000, 0500, 25.1
000032, 0000, 0507, 0000, 0000, 0000, 0000, 0000, 0000, 0000, 0499, 24.7
000033, 0000, 0507, 0000, 0000, 0000, 0000, 0000, 0000, 0000, 0500, 24.9
000034, 0000, 0507, 0000, 0000, 0000, 0000, 0000, 0000, 0000, 0500, 24.8
000035, 0000, 0507, 0000, 0000, 0000, 0000, 0000, 0000, 0000, 0500, 25.0
```

Notes

- The parameter is specified in seconds, but is confirmed as minutes:seconds for convenience:

```
>> logc 600
Logging seconds, mA, V, degrees C with period (mins:secs): 10:00
```

- Some terminal emulators (e.g. PuTTY) can save the incoming serial data stream directly to a file. This is useful for subsequent plotting of the CSV output using a spreadsheet or graphing application.
- The current output is rounded to 1mA prior to display.

17 Logging events

The loge command is used to report port status change events and periodically report the state of all ports.

Syntax:

```
loge [seconds]          (seconds is in range 0..32767)
```

Response:

CSV (comma separated values), which can be copied into a spreadsheet or graphing program.

If a seconds value of '0' is specified then the periodic reporting is disabled and only port status change events will be reported.

If no seconds parameter is supplied a default value of 60s will be used.

A timestamp is output before each event or periodic report.

Example

Here is a device being attached to port 2, left for 9 seconds, and then removed:

```
>> loge 3
Logging events
Press Ctrl-C to stop

System up for 327
0, 0507, 0000, A S -, 0, x, 0.00
1, 0000, 0000, D I -, 0, x, 0.00
2, 0000, 0000, D I -, 0, x, 0.00
3, 0000, 0000, D I -, 0, x, 0.00
4, 0500, 0000, C c -, 0, x, 0.00
Event at 328
2, 0000, 0000, D c -, 0, x, 0.00
Event at 329
2, 1998, 0009, P C -, 1, x, 0.00
System up for 330
0, 0507, 0000, A S -, 0, x, 0.00
1, 0000, 0000, D I -, 0, x, 0.00
2, 1999, 0034, P C -, 2, x, 0.00
3, 0000, 0000, D I -, 0, x, 0.00
4, 0499, 0000, C c -, 0, x, 0.00
Event at 330
2, 1999, 0034, P C -, 2, x, 0.00
System up for 333
0, 0507, 0000, A S -, 0, x, 0.00
1, 0000, 0000, D I -, 0, x, 0.00
2, 2001, 0167, P C -, 5, x, 0.00
3, 0000, 0000, D I -, 0, x, 0.00
4, 0500, 0000, C c -, 0, x, 0.00
System up for 336
0, 0507, 0000, A S -, 0, x, 0.00
1, 0000, 0000, D I -, 0, x, 0.00
2, 2001, 0517, P C -, 8, x, 0.00
3, 0000, 0000, D I -, 0, x, 0.00
4, 0499, 0000, C c -, 0, x, 0.00
Event at 337
2, 0381, 0000, A C -, 9, x, 0.00
Event at 338
2, 0000, 0000, D I -, 0, x, 0.00
System up for 339
0, 0507, 0000, A S -, 0, x, 0.00
1, 0000, 0000, D I -, 0, x, 0.00
2, 0000, 0000, D I -, 0, x, 0.00
3, 0000, 0000, D I -, 0, x, 0.00
4, 0500, 0000, C c -, 0, x, 0.00
```

Notes

- Terminal commands are accepted while in this mode but commands are not echoed and the command prompt is not issued.

18 Reboot

Reboots the firmware.

Syntax:

```
reboot [watchdog]
```

Response:

None (new command prompt appears).

If the `watchdog` parameter is included (i.e. “`reboot watchdog`”) then the system will lock into an infinite, unresponsive loop whilst the watchdog timer expires. The expiration takes several seconds, after which the system will reboot.

If the `reboot` command is issued without a parameter, the `reboot` command is executed immediately.

After a reboot, the USB serial port connection to the host on U16 series devices is reset. This will result in a terminal emulator (e.g. PuTTY or ZTerm) shutting the connection. U8 series devices keep the serial connection to the host open during a reset.

At boot, a string of ANSI escape sequences is sent to the host to reset the terminal emulator. It is suggested that the host software discards everything up to and including the “>>” command prompt.

Although all the firmware is reset when a `reboot` command is issued, the USB port controllers are not *fully* reset, although they are returned to their default mode. To perform a full reset, power-cycle the PS4.

The `reboot` command is also present in boot mode. There it takes no parameter.

Rebooting sets the “R” (rebooted) flag, which is reported by the `health` and `state` commands.

19 Live view

Live view provides a self-refreshing view of the port states and flags. Ports can be commanded using single key presses. To enter live view, use the 'l' command.

Live view is designed for interactive use by humans, not computers. It makes extensive use of ANSI escape sequences to control the cursor position. Do not try to script the control of the live view - its commands and layout may change greatly between firmware versions.

The terminal size (rows, columns) must be large enough or the display will be corrupted. The PS4 attempts to set the number of rows and columns of the terminal when entering live view mode.

Example

```
cambrionix PowerSync 4 4 Port USB Charge+Sync (live view)
Port  Flags      V      mA  State          Start  End  Energy
   0  A-        5.07    0   Sync
>  1                0.00    0   Charge (idle)
>  2  P-       20.01   401   Charging        6                0.00Wh
>  3                0.00    0   Charge (idle)
>  4  C         5.00    0   Detecting

Host present: Yes
                        Input : 25.47V
Total Power :    8W
Seconds since power on: 577
                        Temperature: 46.7C

Flags:   A:Attached, E:System Error, e:Port Error
Commands: o)ff c)harge q)uit live view
          Select port with 1..4. / toggles all ports
```


20 Console remote control

The LEDs, switches and LCD on the console can be controlled via terminal commands. This allows the firmware control of the console to be disabled and for the console to be controlled by the user's software instead. This might be useful where a single board computer or suchlike is connected to the PS4.

Entering remote control mode

To disable the console control from the firmware, and allow it to be controlled via the terminal, issue the `remote` command without parameters:

```
>> remote
```

The LEDs will be turned off when entering remote control mode. The LCD will be unaffected, and previous text will remain. Use `clcd` to clear the LCD (described later).

Leaving remote control mode

To leave remote control mode, and allow the console to be controlled by the firmware once more, issue the `remote exit` command:

```
>> remote exit
```

The LEDs will be reset and the LCD cleared when leaving remote control mode.

Leaving remote control mode when a key is pressed

Sometimes it can be useful to exit remote control mode when a console key is pressed. For example, you may want to display text on the LCD, but return back to the normal charging status screen when a key is pressed. The `kexit` parameter to `remote` tells the PS4 to enter remote control mode, but exit automatically when a console key is pressed:

```
>> remote kexit
```

Notes

1. In `remote kexit` mode, the `keys` command will not return key events.
2. It *is* permitted to move from `remote` mode into `remote kexit` mode, and vice-versa.
3. The key click sound still operates regardless of the remote mode.
4. Charging, syncing and security still operate in remote mode. However, their status will not be reported to the console, and the user will need to poll the status flags (using the `state` and `health` commands) to determine the system state.
5. If the `keys`, `lcd`, or `clcd` commands are issued when not in `remote` or `remote kexit` mode, then an error message will be shown, and the command will not be executed.
6. If the `ledb` or `leds` commands are issued when not in `remote` or `remote kexit` mode, then lower case parameters are ignored.

Writing to the LEDs

There are two methods to write to the LEDs in remote control mode: `ledb` and `leds`. First however, the operation of the LEDs will be described.

There are three LEDs per port. Each LED has a separate flash pattern assigned to it. The flash pattern is an 8 bit byte. Each bit is repeatedly scanned in sequence from MSB to LSB (i.e. left to right). A '1' bit turns the LED on, and a '0' turns it off. For example, a bit pattern of decimal 128 (binary 10000000b) would pulse the LED briefly. A bit pattern of decimal 127 (binary 01111111b) would see the LED on for most of the time, only turning off briefly.

ledb command

The `ledb` command can be used to assign a flash bit pattern to an individual LED.

Syntax:

```
ledb <port> <row> <ptn> [H | R]
```

<port> is the port number, starting at 1

<row> is the LED row number, starting at 1. Typically these are arranged as follows:

Row	LED Function
1	Charged
2	Charging
3	Sync mode

<ptn> can be specified as decimal (range 0..255), hexadecimal (range 00h to ffh) or binary (range 00000000b to 11111111b). Hexadecimal number must end with 'h'. Binary numbers must end with 'b'. More significant digits can be omitted for all radices. For example, '0b' is the same as '00000000b'. Hexadecimal numbers are not case-sensitive.

[H | R] optional parameters 'H' takes over control of the LED without a remote command. 'R' releases control of the LED back to normal operation.

Example

To flash the charging LED on port 8 at 50/50 duty cycle, use:

```
>> ledb 8 2 11110000b
```

To turn on the port 1 charged LED continuously (i.e. no flashing):

```
>> ledb 1 1 ffh
```

To turn off the port 1 sync LED:

```
>> ledb 1 3 0
```

leds command

The `leds` command can be used to assign a string of flash patterns to one row of LEDs. This is much faster for controlling an entire row of LEDs. Just three calls of the `leds` command can set all the LEDs on the system.

Syntax:

```
leds <row> [ptnstr]
```

<row> is address as for `ledb` above.

[`ptnstr`] is a string of characters, one per port, starting at port 1. Each character represents a different flash pattern to be assigned to the port. A string of 8 characters will assign flash patterns to 8 ports. The valid pattern

characters are:

Pattern Character	LED function	Flash pattern
0 (zero)	Off	00000000
1 (one)	On continuously (not flashing)	11111111
f	Flash fast	10101010
m	Flash medium speed	11001100
s	Flash slowly	11110000
p	Single pulse	10000000
d	Double pulse	10100000
O (upper case O)	Off (no remote command needed)	00000000
C	On (no remote command needed)	11111111
F	Flash fast (no remote command needed)	10101010
M	Flash medium speed (no remote command needed)	11001100
S	Flash slowly (no remote command needed)	11110000
P	Single pulse (no remote command needed)	10000000
D	Double pulse (no remote command needed)	10100000
R	Release "no remote command needed " LEDs back to normal use	
x	Unchanged	unchanged

The uppercase commands are useful if the default charging and sync behaviour is required but the host requires one or more LEDs to behave differently e.g. attract user attention to a particular port.

Example

To set up the following flash pattern on the row containing the 'charged' LEDs:

Port	LED Function
1	Unchanged
2	On
3	Flash fast
4	Single pulse
5	Off
6	On continuously
7	On continuously
8	Unchanged

Issue the command:

```
>> leds 1 x1fp011
```

Note that the first LED (port 1) needed to be skipped using the x character. Port 8 was not altered as the pattern string only contained 7 characters.

Notes

- For both `ledb` and `leds` commands, the position in the flash pattern is synchronised across all LEDs. If no LEDs are lit, then the position in the flash pattern is reset and kept at the MSB. Therefore the first LED to be set to a flash pattern starts at the first bit (the MSB) in the flash pattern. Subsequent LEDs may start part-way through their flash patterns, depending on where the `ledb` or `leds` command was issued.

2. No harm or erroneous behaviour will result if the LEDs are written to, but do not physically exist.
3. The LED state is not re-established when `remote` mode is exited and then re-entered.

21 Error reporting

Commands that succeed will output their required results, ending with a new command prompt. Failed commands will output an error code of the form “*Ennn: Error explanation”, followed by a new command prompt. “nnn” is always a three digit decimal number.

Example

Specifying a non-existent port to the `mode` command:

```
>> mode c 17
*E410: Port number must be 1..8
```

22 Fatal errors and the command prompt

When the system encounters a fatal error, the error is reported to the terminal immediately in the following format:

```
*FATAL ERROR Ennn: Error text
```

Ennn is a three-digit decimal error reference number. “Error text” describes the error.

The console LCD will display the error number. The console LEDs will show the error code in binary on the 'charged' LEDs.

After a fatal error has been reported the terminal will only be receptive to CTRL-C (ASCII decimal 3) and ENTER (ASCII decimal 13). If either of these are received, then the system will enter bootloader mode. If CTRL-C or ENTER are not received within the watchdog timeout period (approximately 9 seconds) then the system will reboot.

Important

If a fatal error occurs whilst a controlling script is sending a CTRL-C or ENTER character to the PS4, then bootloader mode will be entered. It is vital that the controlling script recognises bootloader mode, and knows how to exit it.

Bootloader mode is indicated by the prompt `boot>>` (sent on a new line)

The normal command prompt is `>>` (sent on a new line).

In bootloader mode, non-bootloader commands will be responded to with:

```
*E900: Invalid bootloader command
```

To exit bootloader mode, use the `reboot` command, and wait for the normal command prompt to return.

For testing purposes, bootloader mode can be entered by using the `boot` command.

23 Terminology

Term	Explanation
PS4	Any device in the PowerSync Series. E.g. PowerSync 4.
UART	Universal Asynchronous Receiver Transmitter. The hardware that drives a serial port connection.
VCP	Virtual COM port
/dev/	Devices directory on Linux and macOS
IC	Integrated Circuit
PWM	Pulse width modulation. The duty cycle is the percent of time the PWM is in the high (active) state
<CR>	Carriage return character (ASCII decimal 13)
<LF>	Line feed character (ASCII decimal 10)
Console	The front-panel attached to the PS4. The Console provides the keys, piezo sounder and LCD
Terminal	Program that allows characters to be sent and received over a UART connection
Port	USB socket on the front of PS4 that is used to connect mobile devices.
MSB	Most significant bit
LSB	Least significant bit
<ESC>	Escape character (ASCII decimal 27)

24 Revision History

Revision	Date	Author	Comments
1.0	04-Feb-2019	AJG	Initial draft release, documented for firmware version 0.28
1.1	21-Feb-2019	AJG	Rebrand

All trademarks used throughout are acknowledged to be the property of their respective owners.