Overview

Designed for full control of power, SuperSpeed and Hi-Speed data per port in harsh industrial environments, the USBHub3+ is built inside a rugged metal housing with DIN rail mount points. The system is designed to withstand ESD strikes up to ±30kV air- and contact-discharge (IEC61000-4-2 level 4) and is fully operational up to 50°C ambient temperatures. The USBHub3+ device can even survive a developer's desk.

Features

- Seperately enable/disable power, SuperSpeed and Hi-Speed data for each downstream port
- Alternate Eurostyle terminal block input for V+, GND, earth
- Wide DC input power range (9V to 24V)
- Ability to change downstream USB device detection profile. Standard Downstream Port (SDP) [default] and Charging Downstream Port (CDP) are supported[1]
- Eight programmable device (downstream) USB3 type-A connectors
- One dedicated, always-on downstream USB3 type-A connector
- Two host (upstream) USB3 Standard B connectors
- One dedicated control port on a USB Mini-B connector
- Automatic or programmed selection of host input
- Host (upstream) USB port selection LED indicators
- Device (downstream) USB voltage and current measurements
- 8 device ports capable of delivering up to 5A
- Programmable current limiting per device port (1mA to 4.094A in 1mA resolution)
- Boost USB2 upstream and downstream data signal levels
- Full USB battery charge specification (BC1.2) support

Description

The USBHub3+ features a software-programmable USB hub supporting USB 3.1 gen1 SuperSpeed (5Gbps), USB 2.0 Hi-Speed (480Mbps), USB 2.0 full-speed (12Mbps) and USB 1.0/2.0 low-speed (1.5Mbps). It's 8 programmable downstream ports have software controllable SuperSpeed and Hi-Speed data lines, allowing the user to select the speed at which downstream devices may enumerate. The speed devices actually enumerate at is indicated by LEDs on the hub and can be read via the software interface. All data lines can be disabled to simulate device removal.

The V_{BUS} line (5V) for each downstream channel has a hardware current limit of 5A, per the USB BC1.2 safety-limit specification. Further, a lower current limit can be set from software from 1mA to 4.094A with 1mA resolution. The wide input range is limited to 60W total power consumption. USB power and data can be independently switched on or off for advanced USB testing applications. All interfaces are protected against reverse polarity and over-voltage, and the system is designed to operate from 0°C to 50°C ambient with no external cooling or fans.

The USBHub3+ can connect to a host PC by BrainStem™ link, as well as operate independently by running embedded, user-defined programs based on the BrainStem Reflex language.

[1] See http://www.usb.org/developers/docs/devclass_docs/ under the category Battery Charging for full details.

Absolute Maximum Ratings

Stresses beyond those listed under ABSOLUTE MAXIMUM RATINGS can cause permanent damage to the device. These are stress ratings only and functional operation of the device at these or any other conditions beyond those indicated under RECOMMENDED OPERATING CONDITIONS is not implied. Exposure to absolute-maximum rated conditions for extended periods affects device reliability and may permanently damage the device.

Parameter	Minimum	Maximum	Units
Input Voltage, V_{supply}	2	30.0	V
Input Current, I_{supply}		5	A
V_{supply} power		65	W
Voltage on any V_{bus} line, upstream and downstream	0.0	6.0	V
Voltage on any USB D+/D-, upstream and downstream	-0.3	5.3	V

Table 1: Absolute Maximum Ratings

Handling Ratings

Parameter	Conditions/Notes	Minimum	Typical	Maximum	Units
Ambient Operating Temperature, T_A	Non-Condensing	0.0	25.0	50.0	°C
Storage Temperature, T_{STG}		-10.0	-	85.0	°C
Electrostatic Discharge, V_{ESD}	Exceeds IEC 61000-4-2, level 4, air and contact discharge	0.0	-	±30	kV

Table 2: Handling Ratings

Recommended Operating Ratings

Values presented apply to the full operating temperature range.

Parameter	Conditions/Notes	Minimum	Typical	Maximum	Units
Input Voltage, V_{supply}		9.0	12.0	24.0	V
Voltage on V_{bus} inputs and outputs		4.5	5.0	5.5	V

Table 3: Recommended Operating Ratings

Block Diagram

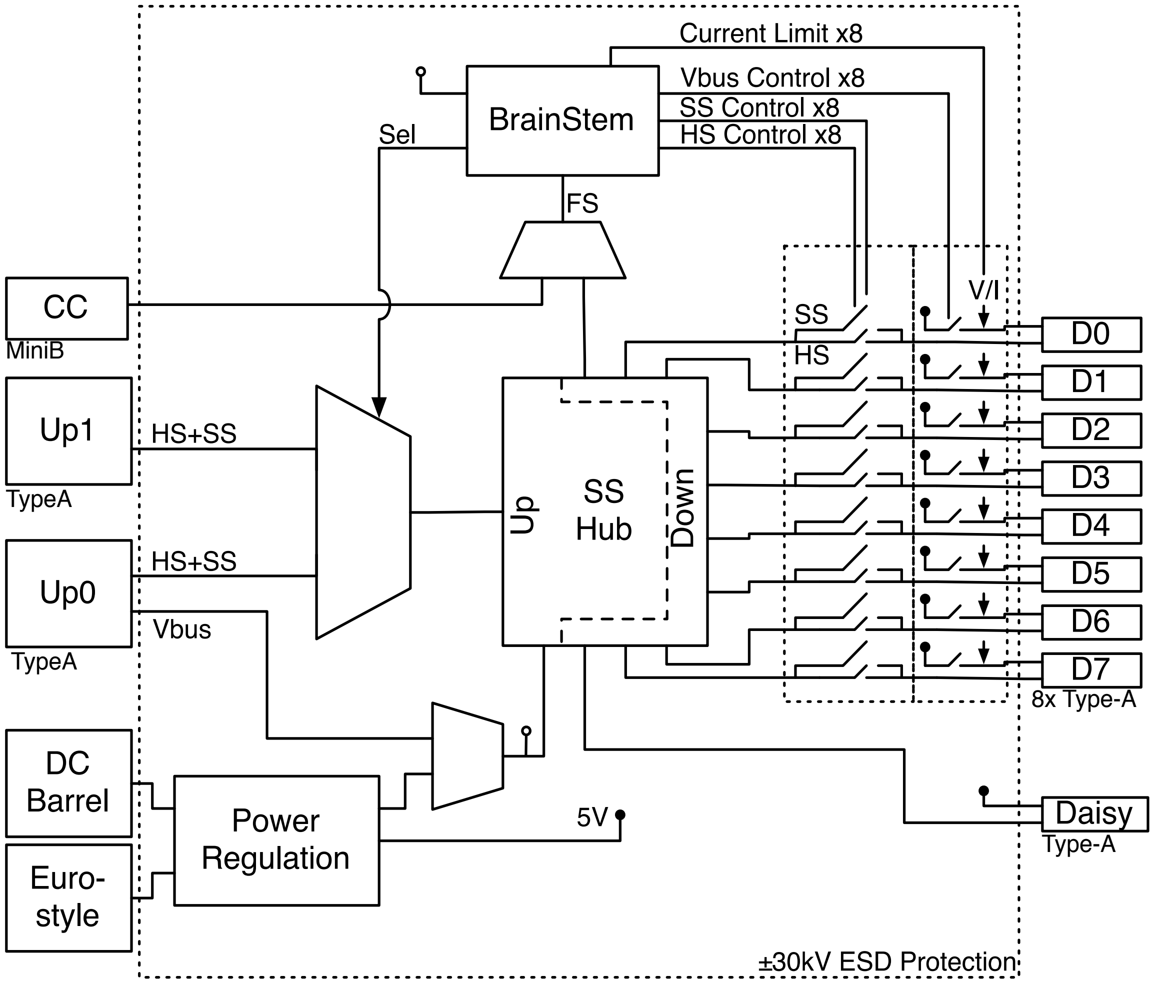


Figure 1: USBHub3+ Block Diagram

Typical Performance Characteristics

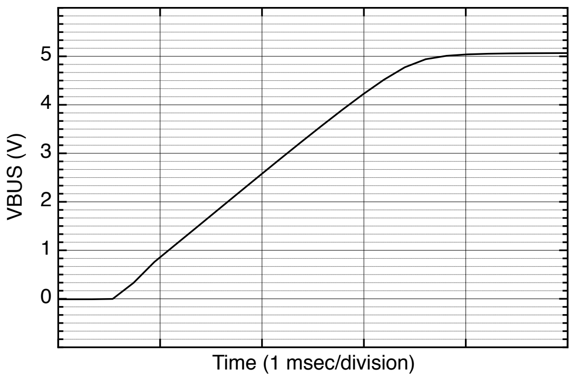
Values presented apply to the full operating temperature range.

Parameter	Conditions/Notes	Minimum	Typical	Maximum	Units
Input Power, W_{supply}, no downstream devices attached		-	2.2	-	W
Input Voltage Under Voltage Lockout, V_{supply}		7.96	8.065	8.17	V
Input Voltage Over Voltage Lockout, V_{supply}		26.14	26.82	27.5	V
Wide Input Range System Efficiency	@12.0V input, nominal 8A load[2]	84	-	92	%
USB Downstream (VBUS) Output Voltage	No load on downstream USB ports	4.947	5.10	5.25	V
USB Downstream (VBUS) Voltage Measurement Resolution		-	8	-	mV
USB Downstream Hardware Overcurrent Trip		4800	5000	5400	A
USB Downstream Current Trip Point Resolution		-	1	-	mA
USB Downstream Software Overcurrent Trip		0	-	4095	mA
Short Circuit Detection Time	Time from detection of short to port power switch disconnect.	-	-	5.0	ms
USB SuperSpeed Data Rate	May depend on host or devices			5	Gbps
USB Hi-speed Data Rate	May depend on host or devices			480	Mbps
USB Downstream VBUS Current Supply (SDP mode)	USB 2.0 data lines disabled or no USB host present	-	-	100	mA
USB Downstream VBUS Current Supply (SDP mode)	USB 2.0 data lines enabled and USB host must be present	-	-	500	mA
USB Downstream VBUS Current Supply (CDP mode)	USB 2.0 data lines must be enabled	-	-	1500	mA
Current consumption powered from USB2 cable into CH0 upstream port	No V_{supply} present	-	180	-	mA
Current consumption power from USB3 cable into CH0 upstream port	No V_{supply} present	-	425	-	mA

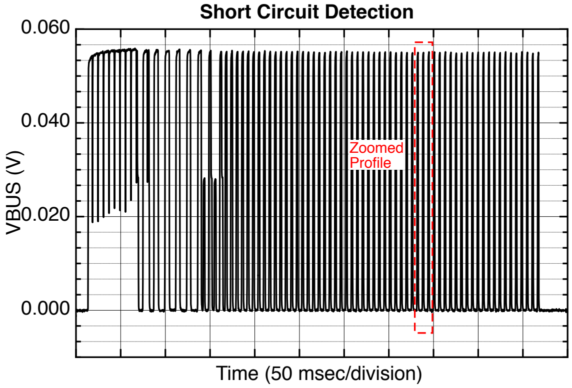
Table 4: Typical Performance Characteristics

[2] Representative 8A load based on 8 USB downstream devices running in CDP mode consuming approximately 1.0A each.

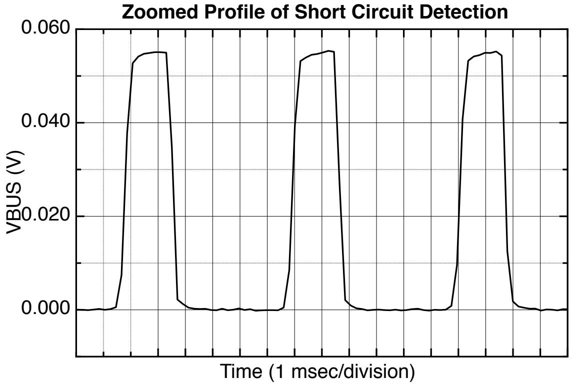
USB Downstream Enable into 100mA Load



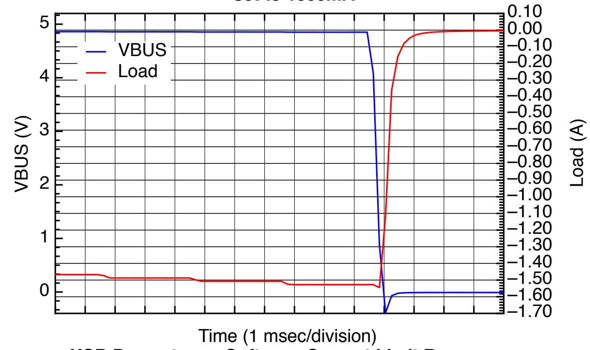
Short Circuit Detection



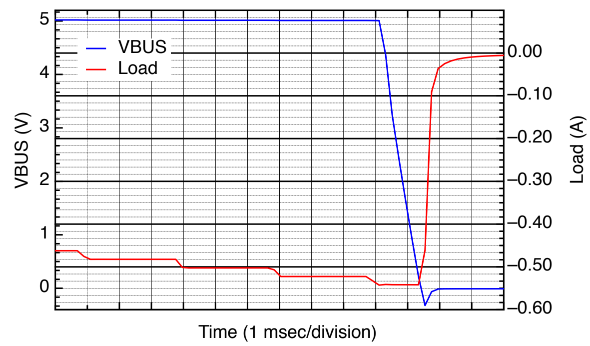
Zoomed Profile of Short Circuit Detection



USB Downstream Software Current Limit Response set to 1500mA



USB Downstream Software Current Limit Respone set to 500mA



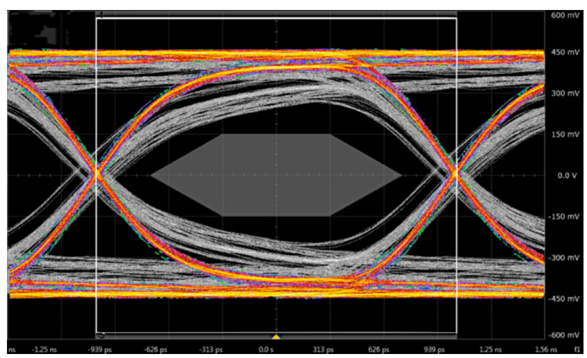


Figure 2: Downstream USB eye diagram USB A to device. 3.2m cable in greyscale; 0.3m cable in color.

Device Drivers

USBHub3+ leverages operating system user space interfaces that do not require custom drivers for operation on modern operating systems.

Outdated and unsupported operating systems may require the installation of a BrainStem USB driver. Installation details on installing USB drivers can be found within the BrainStem Development Kit under the "drivers" folder. Please note that Windows 7 will require looking at the supplied INF file found in the "drivers\brainstem_WinUSB" folder of the download.

Capabilities and Interfaces

The USBHub3+ is built on Acroname's BrainStem system which provides simple high level APIs, a real-time embedded runtime engine and modular expandability. Functionality details unique to the USBHub3+ are described in the following sections; refer to Table 10: Supported USBHub3+ BrainStem Entity API Methods for a complete list of all available API functionality. All shortened code snippets are loosely based on the C++ method calls and meant to be psuedocode – Python and Reflex are virtually the same. Please consult the BrainStem Reference for implementation details.[3]

System Entities

Every BrainStem module includes a single System Entity. The System Entity allows access to configuration settings such as the module address, input voltage, control over the user LED and many more.

Serial Number

Every USBHub3+ is assigned a unique serial number at the factory. This facilitates an arbitrary number of USBHub3+ devices attached to a host computer. The following method call can retrieve the unique serial number for each device.

```
stem.system.getSerialNumber(serialNumber)
```

Module Default Base Address

BrainStems are designed to be able to form a reactive, extensible network. All BrainStem modules come with a default network base address for identification on the BrainStem network bus. The default module base address for USBHub3+ is factory-set as 6, and can be accessed with:

```
stem.system.getModule(module)
```

Saving USB Entity Settings

Some entities can be configured and saved to non-volatile memory. This allows a user to modify the startup and operational behavior for the USBHub3+ away from the factory default settings. Saving system settings preserves the settings to become the new default. Most changes to system settings require a save and reboot before taking effect. The USB Port Enumeration Delay and USB Downstream Current Limit, can be changed with immediate effect. USB Boost mode settings (both upstream and downstream), for example, will not take effect unless a system save operation is completed, followed by a full power cycle. A firmware update will return all settings to default. Use the following command to save changes to system settings before reboot:

```
stem.system.save()
```

Saved Configurations	
Software Offset	I2C Rate
Router Address	Port Enumeration Delay
Boot Slot	Downstream Boost
Port State	Current Limit – per port
Upstream Boost	Port Mode (SDP, CDP) – each port

USB Entities

`usb` entities provide a mechanism to control all functionality for the upstream and downstream USB ports.

3 See BrainStem software API reference at https://acroname.com/reference/ for further details about all BrainStem API methods and information.

USB Downstream Channels

Downstream USB channels can be manipulated through the `usb` entity command to enable/disable USB data and VBUS lines, measure current, measure VBUS voltage, boost data line signals, and measure temperature.

Manipulating both data lines and VBUS lines for a single channel simultaneously can be done by calling the following method (channel 0-7):

```
stem.usb.setPortEnable(channel)
stem.usb.setPortDisable(channel)
```

Manipulating both data lines while not affecting the VBUS lines for a single channel simultaneously can be done by calling the following method (channel 0-7):

```
stem.usb.setDataEnable(channel)
stem.usb.setDataDisable(channel)
```

Manipulating just the USB 2.0 high-speed data lines for a single channel can be done by calling the following method (channel 0-7):

```
stem.usb.setHiSpeedDataEnable(channel)
stem.usb.setHiSpeedDataDisable(channel)
```

Manipulating just the USB 3.1 SuperSpeed data lines for a single channel can be done by calling the following method (channel 0-7):

```
stem.usb.setSuperSpeedDataEnable(channel)
stem.usb.setSuperSpeedDataDisable(channel)
```

Manipulating just the USB VBUS line for a single channel can be done by calling the following method (channel 0-7):

```
stem.usb.setPowerEnable(channel)
stem.usb.setPowerDisable(channel)
```

The USB VBUS voltage, as well as the current consumed on VBUS, can be read for each channel by calling the following methods (channel 0-7), where the second variable passed into the method is the write location of the result:

```
stem.usb.getPortVoltage(channel,μV)
stem.usb.getPortCurrent(channel,μA)
```

Current-limit settings can be accessed for each channel by calling the following methods (channel 0-7), where the second variable passed into the method is either the set value or the write location of the result:

```
stem.usb.getPortCurrentLimit(channel,μA)
stem.usb.setPortCurrentLimit(channel,μA)
```

The USB power regulation stage (5.0V-regulated for VBUS) can be monitored for temperature stability, using the following method, where the variable passed into the method is the write location of the result:

```
stem.usb.getSystemTemperature(μC)
```

USB Downstream Operational Mode

The mode setting defaults to Standard Downstream Port (SDP) mode and a current-limit of 500 milliamps – the device can alternately be set to Charging Downstream Port (CDP) mode for devices that require high port charge current above 500 milliamps.

```
stem.usb.setPortMode(channel,mode)
stem.usb.getPortMode(channel,mode)
```

Available options for Downstream Operational Mode are:

- 0 – Standard Downstream Port (SDP)
- 1 – Charging Downstream Port (CDP)

USB Downstream Enumeration Delay

Once a USB device is detected by the USBHub3+ it is possible to create a host computer connection (enumeration) delay. The enumeration delay can mitigate or eliminate host kernel instabilities by forcing devices to enumerate in succession, allowing a focus on validation of drivers and software. The enumeration delay is in milliseconds, and applied to each downstream port.

```
stem.usb.setEnumerationDelay(delay)
stem.usb.getEnumerationDelay(delay)
```

USB Boost Mode

Boost mode increases the drive strength of the USB 2.0 high-speed data signals (SuperSpeed data and power signals are not changed). Boosting the data signal strength may help to overcome connectivity issues when using long cables or connecting through relays, "pogo" pins or other adverse conditions. This setting is not applied until a system save call and power cycle of the hub. The system setting is then persistent until changed (followed by a save and reboot) or the hub is hard reset. After a hard reset, the default value of 0% boost is restored unless a new boost value has been applied and saved. A hard reset is done by pressing the “Reset” button on the back of the hub while the hub is powered.

Boost mode can be applied to both the upstream and downstream USB ports.

```
stem.usb.getDownstreamBoostMode(setting)
stem.usb.setDownstreamBoostMode(setting)
stem.usb.getUpstreamBoostMode(setting)
```

```
stem.usb.setUpstreamBoostMode(setting)
```

The *setting* parameter is an integer that correlates to the following:

- 0 – no boost
- 1 – 4% boost
- 2 – 8% boost
- 3 – 12% boost

USB Hub Upstream Channels

The USBHub3+ is perfect for environments where multiple devices need to be shared or switched between two host computers using two host (upstream) connections via USB Mini-B connectors. The upstream connection can be detected or specified using the following methods:

```
stem.usb.getUpstreamMode(mode)
stem.usb.setUpstreamMode(mode)
```

The *mode* parameter can be defined as the following (C++ macro name in parentheses):

- 2 (usbUpstreamModeAuto)
- 0 (usbUpstreamModePort0)
- 1 (usbUpstreamModePort1)

The default operational mode is to have the USB Upstream (to a host computer) be enabled through Channel 1 and auto detect which USB port to use. Automatic detection uses the absence/presence of a VBUS connection coming through the USB type-B connector on Channel 0; i.e. if a cable is connected in channel 0, then channel 0 will be used for upstream traffic.

BrainStem Control Channel

The USBHub3+ also has a dedicated control channel on the USB mini-B. This is a full-speed USB 2.0 connection for BrainStrem interface only. No USB hub traffic can flow on this connection. When a cable is connected to the mini-B connector, the BrainStem device will be controlled only through this port. The USB 3.0 type-B connectors are then used only for USB hub traffic to connect downstream USB devices.

USB Hub Operational Mode

In addition to targeting individual downstream USB ports, a bit-mapped hub state interface is also available. This interface allows the reading or setting of all USB downstream ports in one functional call.

```
stem.usb.getHubMode(state)
stem.usb.setHubMode(state)
```

The value *state* must be a 32-bit word, defined as the following:

Bit	Hub Operational Mode Result Bitwise Description
0	USB Channel 0 USB Hi Speed Data Enabled
1	USB Channel 0 USB VBUS Enabled
2	USB Channel 1 USB Hi Speed Data Enabled
3	USB Channel 1 USB VBUS Enabled
4	USB Channel 2 USB Hi Speed Data Enabled
5	USB Channel 2 USB VBUS Enabled
6	USB Channel 3 USB Hi Speed Data Enabled
7	USB Channel 3 USB VBUS Enabled
8	USB Channel 4 USB Hi Speed Data Enabled
9	USB Channel 4 USB VBUS Enabled
10	USB Channel 5 USB Hi Speed Data Enabled
11	USB Channel 5 USB VBUS Enabled
12	USB Channel 6 USB Hi Speed Data Enabled
13	USB Channel 6 USB VBUS Enabled
14	USB Channel 7 USB Hi Speed Data Enabled
15	USB Channel 7 USB VBUS Enabled
16	USB Channel 0 USB Super Speed Data Enabled
17	Reserved
18	USB Channel 1 USB Super Speed Data Enabled
19	Reserved
20	USB Channel 2 USB Super Speed Data Enabled
21	Reserved
22	USB Channel 3 USB Super Speed Data Enabled
23	Reserved

24	USB Channel 4 USB Super Speed Data Enabled
25	Reserved
26	USB Channel 5 USB Super Speed Data Enabled
27	Reserved
28	USB Channel 6 USB Super Speed Data Enabled
29	Reserved
30	USB Channel 7 USB Super Speed Data Enabled
31	Reserved

Table 5: Hub Operational Mode Result Bitwise Description

USB Hub State

In addition to targeting individual downstream USB ports, overall hub state information can be represented in a bit packed result showing every port's state:

```
stem.usb.getHubState(bank,status)
```

The value *status* must be a 32-bit word, defined as the following:

Bit	Hub State (Bank = 0) Result Bitwise Description
0	USB CH0 device is attached
1:2	Reserved
3	USB CH0 error. See USB Hub Error Status Mapping
4	USB CH0 Hi Speed Operation
5	USB CH0 Super Speed Operation
6:7	Reserved
8	USB CH1 device is attached
9:10	Reserved
11	USB CH1 error. See USB Hub Error Status Mapping
12	USB CH1 Speed Hi Speed Operation
13	USB CH1 Super Speed Operation
14:15	Reserved
16	USB CH2 device is attached
17:18	Reserved
19	USB CH2 error. See USB Hub Error Status Mapping
20	USB CH2 Speed Hi Speed Operation
21	USB CH2 Super Speed Operation
22:23	Reserved
24	USB CH3 device is attached
25:26	Reserved
27	USB CH3 error. See USB Hub Error Status Mapping
28	USB CH3 Hi Speed Operation
29	USB CH3 Super Speed Operation
30:31	Reserved

Table 6: Hub State (Bank 0) Result Bitwise Description

Bit	Hub State (Bank = 1) Result Bitwise Description
0	USB CH4 device is attached
1:2	Reserved
3	USB CH4 error. See USB Hub Error Status Mapping
4	USB CH4 Hi Speed Operation
5	USB CH4 Super Speed Operation
6:7	Reserved
8	USB CH5 device is attached
9:10	Reserved
11	USB CH5 error. See USB Hub Error Status Mapping
12	USB CH5 Speed Hi Speed Operation
13	USB CH5 Super Speed Operation
14:15	Reserved
16	USB CH6 device is attached
17:18	Reserved
19	USB CH6 error. See USB Hub Error Status Mapping
20	USB CH6 Speed Hi Speed Operation
21	USB CH6 Super Speed Operation
22:23	Reserved
24	USB CH7 device is attached
25:26	Reserved
27	USB CH7 error. See USB Hub Error Status Mapping

28	USB CH7 Hi Speed Operation
29	USB CH37 Super Speed Operation
30:31	Reserved

Table 7: Hub State (Bank 1) Result Bitwise Description

USB Hub Error Status Mapping

It is possible to retrieve current error states for all downstream ports in a single 32-bit word. Since 8 downstream USB ports are available, the *bank* parameter should be set to 0 for downstream ports 0-3 and the *bank* parameter should be set to 1 for downstream ports 4-7.

```
stem.usb.getHubErrorStatus(bank,status)
```

Errors can be cleared on each individual channel (0, 1, 2 or 3) by calling the following method:

```
stem.usb.clearPortErrorStatus(channel)
```

```
Calling this command clears the error bit
flag (see Table 6) in the hub state as well
as any errors indicated in the error state
bit mask (Table 7).
```

Details about the hub error status 32-bit word are as follows:

Bit	Hub Error Status (Bank = 0) Result Bitwise Description
0	USB CH0 overcurrent limit exceeded
1:7	Reserved
8	USB CH1 overcurrent limit exceeded
9:15	Reserved
16	USB CH2 overcurrent limit exceeded
17:23	Reserved
24	USB CH3 overcurrent limit exceeded
25:31	Reserved

Table 8: Hub Error Status (Bank 0) Result Bitwise Description

Bit	Hub Error Status (Bank = 1) Result Bitwise Description
0	USB CH4 overcurrent limit exceeded
1:7	Reserved
8	USB CH5 overcurrent limit exceeded
9:15	Reserved
16	USB CH6 overcurrent limit exceeded
17:23	Reserved
24	USB CH7 overcurrent limit exceeded
25:31	Reserved

Table 9: Hub Error Status (Bank 0) Result Bitwise Description

USBHub3+ Supported Entity Methods Summary

Detailed entity class descriptions can be found in the BrainStem Reference (https://acroname.com/reference/entities/index.html). A summary of USBHub3+ class options are shown below. Note that when using Entity classes with a single index (aka, 0), the index parameter can be dropped. For example:

`stem.system[0].setLED(1)` → `stem.system.setLED(1)`

Entity Class	Entity Option	Variable(s) Notes
store[0-1]	getSlotState	
	loadSlot	
	unloadSlot	
	slotEnable	
	slotDisable	
	slotCapacity	
	slotSize	
system[0]	save	
	reset	
	setLED	
	getLED	
	setSleep	
	setBootSlot	
	getBootSlot	
	getInputVoltage	
	getInputCurrent	
	getVersion	
	getModuleBaseAddress	
	getModuleSoftwareOffset	
	setModuleSoftwareOffset	
	setHBInterval	
	getHBInterval	
	getRouterAddressSetting	
	getModule	
	getSerialNumber	
	setRouter	
	getRouter	
	getModel	
timer[0-8]	getExpiration	
	setExpiration	
	getMode	

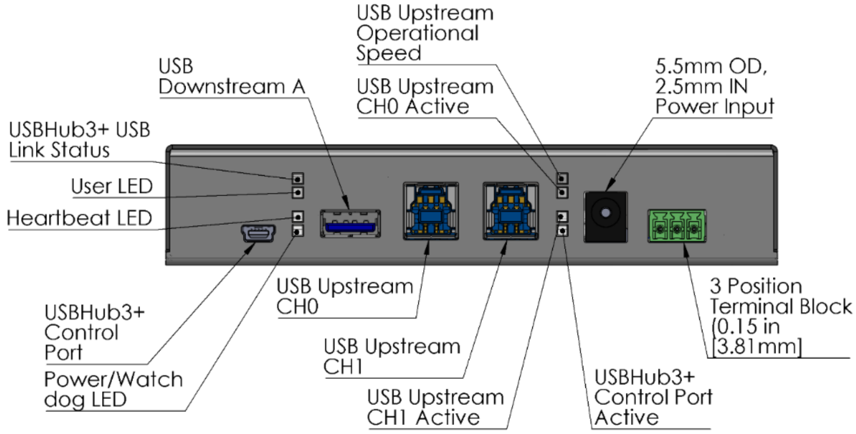
	setMode	
usb[0]	setPortEnable	channel can be 0-7
	setPortDisable	channel can be 0-7
	setDataEnable	channel can be 0-7
	setDataDisable	channel can be 0-7
	setHiSpeedDataEnable	channel can be 0-7
	setHiSpeedDataDisable	channel can be 0-7
	setSuperSpeedDataEnable	channel can be 0-7
	setSuperSpeedDataDisable	channel can be 0-7
	setPowerEnable	channel can be 0-7
	setPowerDisable	channel can be 0-7
	getPortVoltage	channel can be 0-7
	getPortCurrent	channel can be 0-7
	getPortCurrentLimit	channel can be 0-7
	setPortCurrentLimit	channel can be 0-7
	setPortMode	channel can be 0-7. mode can be 0 (SDP) or 1 (CDP)
	getPortMode	channel can be 0-7
	getHubMode	
	getHubState	
	setHubState	
	getHubErrorStatus	
	getSystemTemperature	
	setEnumerationDelay	
	getEnumerationDelay	
	clearPortErrorStatus	channel can be 0-7
	getUpstreamMode	
	getUpstreamState	
	setUpstreamBoostMode	
	setDownstreamBoostMode	
	getUpstreamBoostMode	
	getDownstreamBoostMode	

Table 10: Supported USBHub3+ BrainStem Entity API Methods[4]

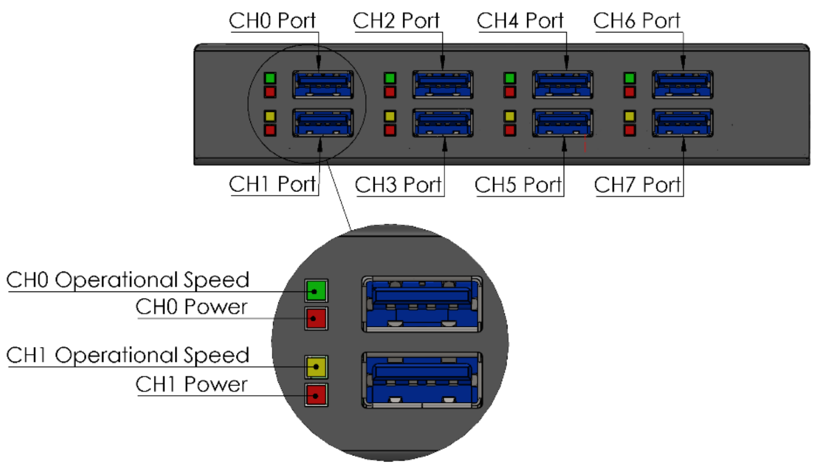
[4] See BrainStem software API reference at https://acroname.com/reference/ for further details about all BrainStem API methods and information.

LED Indicators

The back of the USBHub3+ has a set of indicators that show control information and connectivity status. The USBHub3+ has a USB Link Status that turns yellow once a host device has enumerated the BrainStem module. A user controllable LED (blue) can be manipulated through user code with the API. When a software connection is established to the USBHub3+, the Heartbeat LED (green when on) pulses on and off. Upstream host speed to the internal USB hubs is indicated with the USB upstream Operational Speed – when a Super speed connection is created the LED will turn green. When the Operational Speed is yellow it indicates a host's connection is operating at Hi speed (Full speed). When no host connection is present, the USB Upstream Operational Speed, USB Upstream CH0 Active, and the USB Upstream CH1 Active LEDs will be turned off. In the event a USB connection to the USBHub3+ is connected through the USBHub3+ Control Port, the USBHub3+ Control Port Active will turn green.



Each port shows the connected device's enumeration speed with a green LED indicating SuperSpeed and a yellow LED for Hi-speed or lower speeds. When no downstream device is connected the Operational Speed indicator will be turned off. Each port's power state is indicated by a red LED.



Host Port Control Application Notes

The two upstream ports can be connected to two different host computers. Because of how USB architecture works, only one of the host computers can be selected to have access to the downstream USB ports. Through the BrainStem APIs, you can change which of the two host computers has access to the downstream ports. Also, a dedicated USB control port is available that provides a constant connection to the BrainStem module independent of which USB host port is connected. Additionally this provides the capacity to disconnect both USB upstream host connections completely.

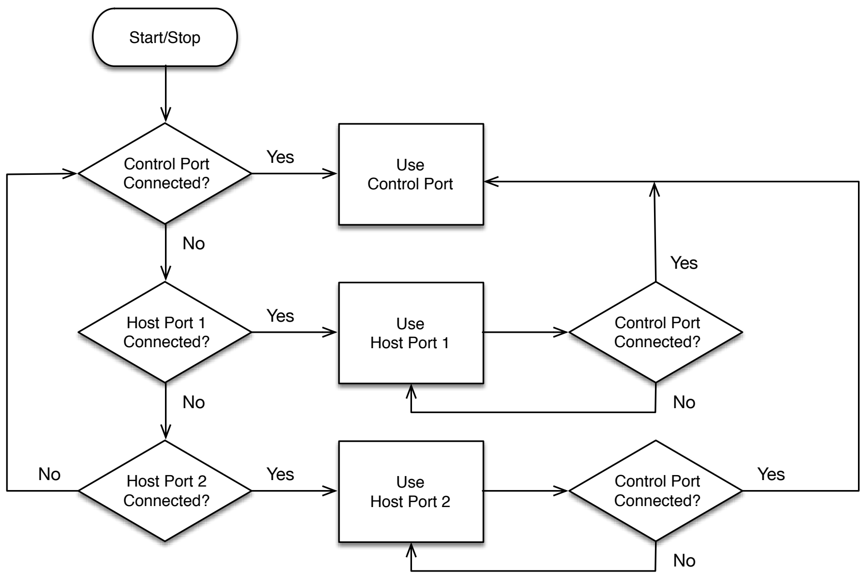


Figure 3: Host Port Control Flow

Mechanical

Dimensions are shown in inches [mm]. 3D CAD models are available through the USBHub3+ product page's Downloads section.

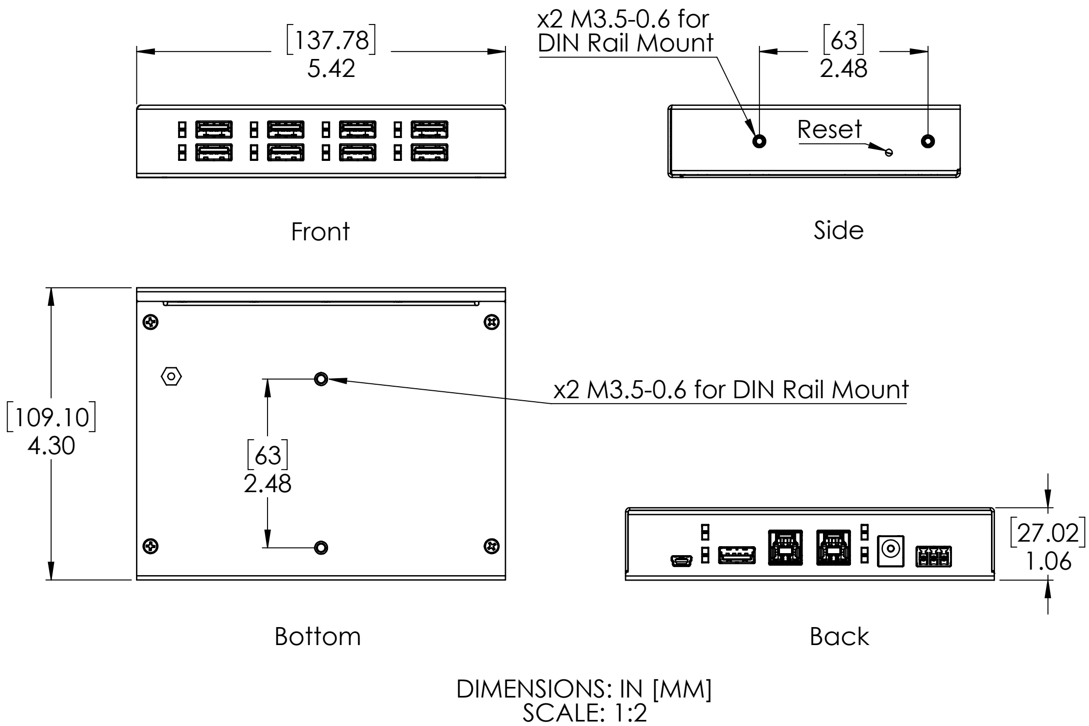


Figure 4: USBHub3+ Mechanical

DIN Rail Mounting

DIN rail mounts have been designed into the USBHub3+ case with an appropriate clip as often used for industrial control equipment. Mounting clip hardware is not included with the USBHub3+. The mounting holes are compatible with many widely available “small” DIN rail mounting clips, and Acroname part number C31-DINM-1. The USBHub3+ can be mounted in two positions:

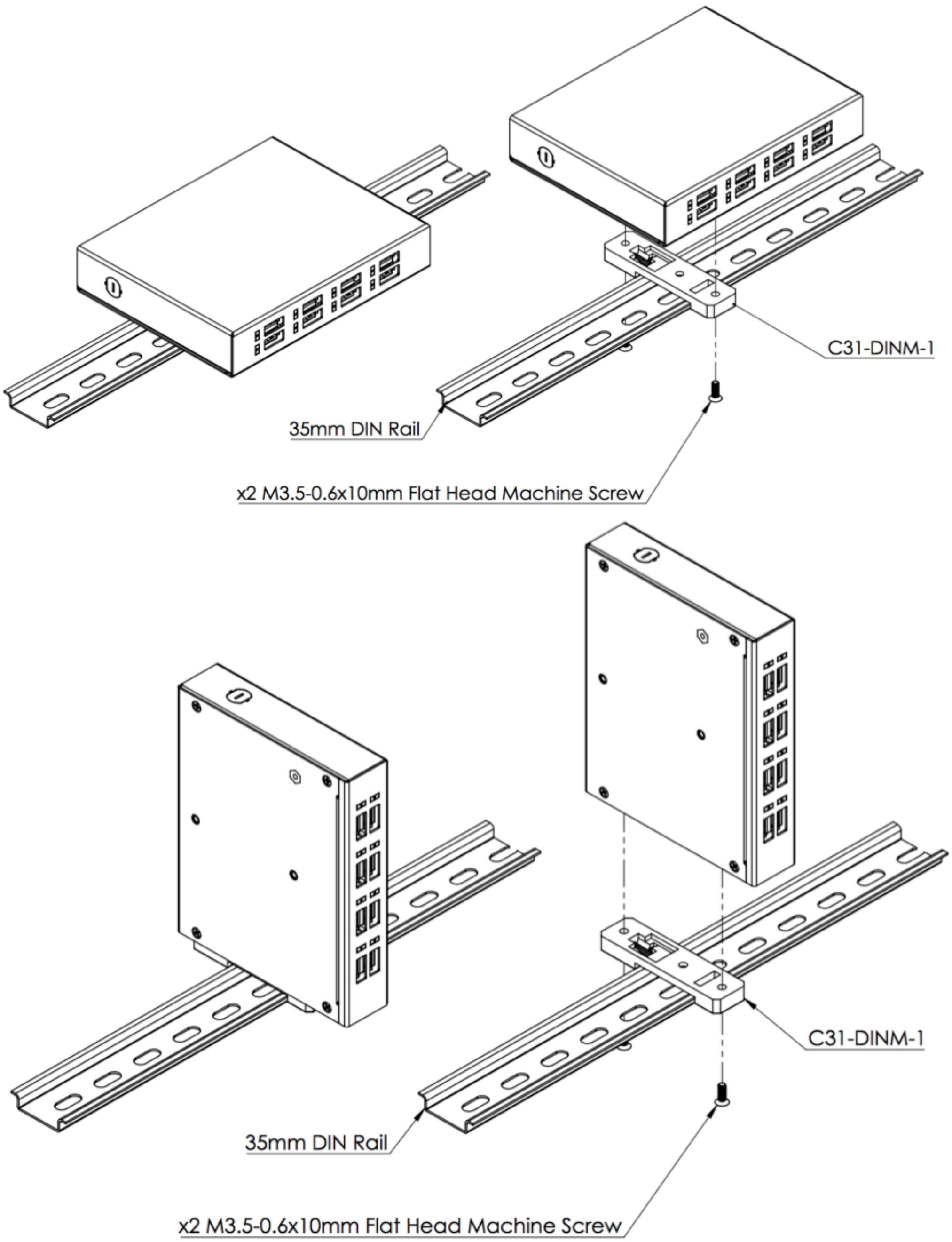
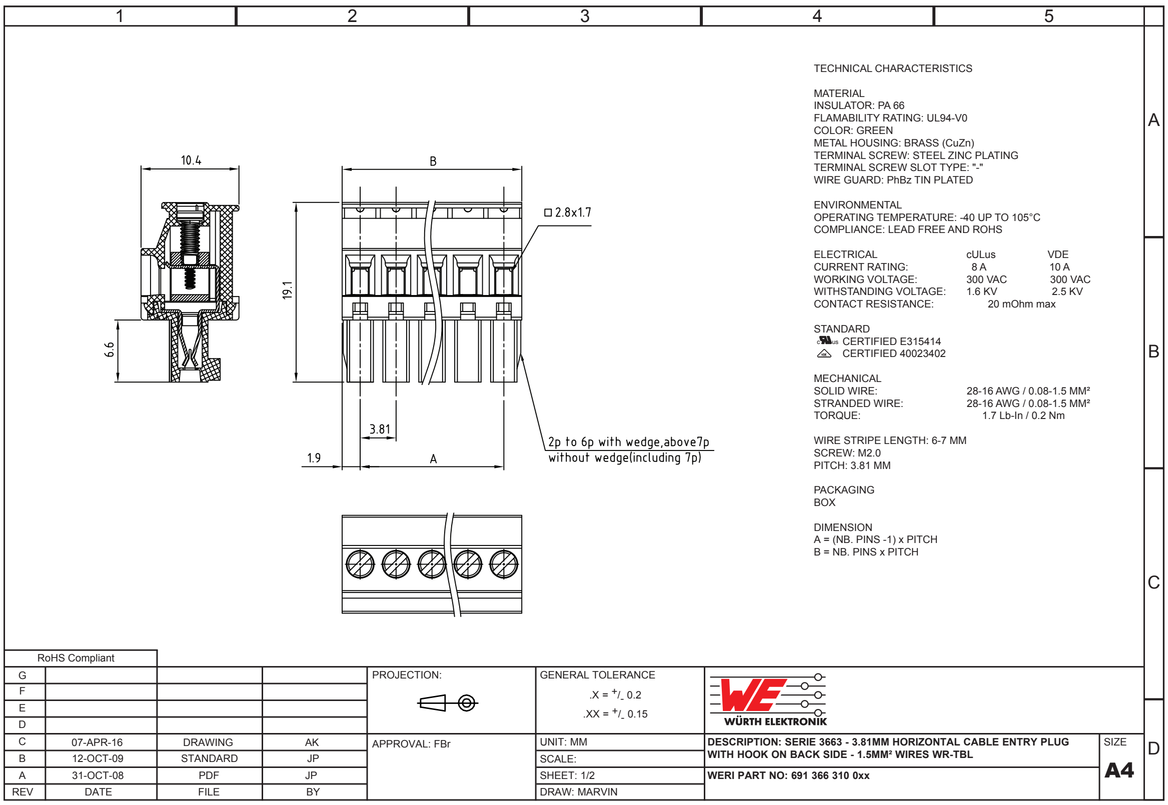


Figure 5: USBHub3+ DIN Rail Mount

Euro-style 3-pin Terminal Block

In addition to the standard barrel jack power input an alternative power input, a Euro-style 3-pin terminal block (“Euroblock”), is also provided. This connector provides the additional benefits of a dedicated earth connection and a higher current rating (8 A) than the barrel connector (5 A). Many manufacturers make compatible connectors; one example is the Wurth Elektronik Serie 3663- 3.81mm Horizontal with Hook on Back Side WR-TBL, part number 691366310003 (see diagram below).



Source: http://katalog.we-online.com/en/em/TBL_3_81_3663_HORIZONTAL_W_HOOK_ON_BACK_SIDE_69136631000X

FCC Compliance Statement

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

This device complies with part 15 of FCC Rules. Operation is subject to the following two conditions; (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Document Revision History

All major documentation changes will be marked with a dated revision code

Revision	Date	Engineer	Description
1.0	September 26, 2016	JTD	Initial Release