Big Dog Power API

Updated 2/20/24:

This API Document Applies to PDU's that have been updated to Firmware Version 2.07 or newer.

Default Username and Password:
UN: admin
PW: admin

Command Structure:

Username and Password @ IP Address of PDU / Specific Command = Outlet Number if applicable

For Example:
"admin:admin@192.168.1.40/turnOn?outlet=5"

Turn on outlet:

URL	Type	Variables	Notes	Response
/turnOn?outlet=x	GET	X is the outlet number (1-3)		OK

Turn off outlet:

URL	Type	Variables	Notes	Response
/turnOff?outlet=x	GET	X is the outlet number (1-3)		OK

Reboot outlet:

URL	Type	Variables	Notes	Response
/rebootOutlet?outlet=x	GET	X is the outlet number (1-3)		OK

Get device status:

URL	Type	Variables	Notes	Response
/updateOutlets	GET	none	The response is a JSON array with 4 items. The first item is the overall system status. The next 3 items are the outlet.	[{"chipID":"246F28DB09D8", "model":3,"upTime":0, "mov1":1, "mov2":2}, {"num":1,"isOn":true,"a":0, "v":115.99,"w":0, "upTime":14932}, {"num":2,"isOn":true,"a":0.14, "v":115.99, "w":12.3,"upTime":14946},

			Num = outlet number
			isOn = is the outlet on
			a = amps
			v = voltage
			w = watts
			uptime = time outlet has been on in seconds.
			
			13 outlet – data[0] will also include t1, t2, t3, t4 values to represent temperatures in F.
			Please note meter data can take up to 60 seconds to appear.
				{"num":3,"isOn":true,"a":0, "v":115.99,"w":0, "upTime":1846}]

Get ESP Firmware:

URL	Type	Variables	Notes	Response
/getFirmwareESP	GET	none		1.20

Get ESP MCU:

URL	Type	Variables	Notes	Response
/getFirmwareMCU	GET	none		0.3

Factory Reset ESP:

URL	Type	Variables	Notes	Response
/factoryReset?password=x	GET	X is the mac address of the ESP32		OK

Get Button Last Event:

URL	Type	Variables	Notes	Response
/getButtonLastEvent	GET	None	0 is no event 1 is pressed 2 is released 3 is long press	1

UPS Serial Test:

URL	Type	Variables	Notes	Response
/upsTest?text=x	GET	X is any text you want to send.	This is a loopback test. The text you send from the API will be sent over	There are 2 responses, success and failure: Success:

			Serial. It will wait 100ms and then read any data on the serial line. If that data matches what was sent the test will pass.	Pass: textReceived Fail: Fail: Nothing Received

Get Input Trigger State:

URL	Type	Variables	Notes	Response
/ getLastInputTrigger	GET	None	0 is no event/open 1 is pressed	1

Turn on Output:

URL	Type	Variables	Notes	Response
/ turnOnOutput	GET	None	Turns on the IO output	OK

Turn off Output:

URL	Type	Variables	Notes	Response
/ turnOffOutput	GET	None	Turns off the IO output	OK

Get Output State

URL	Type	Variables	Notes	Response
/ getLastOutputState	GET	None		0 is Off 1 is On

Set LED Backlights

URL	Type	Variables	Notes	Response
/ setBackight?item=X&state=Y	GET	X a string of the led you want to control: LCD Up Down Right Left Enter Y is the state of the LED 0 = Off 1 = On		OK

Write Text to LCD

URL	Type	Variables	Notes	Response
/ writeLCD?text=X	GET	X can be any string you like.	The ESP32 will write the text to both lines on the LCD. The LCD will quickly revert to the text that should be displayed.	OK

Get Last Navigation Button

URL	Type	Variables	Notes	Response
/ getLastNavButton	GET	None		0 is Off 1 is On

Update All Metering:

URL	Type	Variables	Notes	Response
/updateAllMetering	GET	none	This will force the MCU to check all outlets on/off state, then the metering data for all outlets. A JSON message with the current system state will be returned.	[{"chipID":"246F28DB09D8", "model":3,"upTime":0, "mov1":1, "mov2":2}, {"num":1,"isOn":true,"a":0, "v":115.99,"w":0, "upTime":14932}, {"num":2,"isOn":true,"a":0.14, "v":115.99, "w":12.3,"upTime":14946}, {"num":3,"isOn":true,"a":0, "v":115.99,"w":0, "upTime":1846}]