



## DLI LPC9 User's Guide

20170809T111854Z

---

## Contents

<b>1</b>	<b>Overview</b>	<b>2</b>
<b>2</b>	<b>Product features</b>	<b>3</b>
<b>3</b>	<b>Package contents</b>	<b>4</b>
<b>4</b>	<b>Basic setup and use</b>	<b>5</b>
<b>5</b>	<b>LCD and keypad</b>	<b>10</b>
<b>6</b>	<b>Setup page</b>	<b>12</b>
<b>7</b>	<b>Scripting</b>	<b>19</b>
<b>8</b>	<b>Event notification</b>	<b>26</b>
<b>9</b>	<b>Customization page</b>	<b>33</b>
<b>10</b>	<b>External APIs</b>	<b>34</b>
<b>11</b>	<b>Backing up settings</b>	<b>40</b>
<b>12</b>	<b>Firmware upgrade</b>	<b>41</b>
<b>13</b>	<b>Date/time</b>	<b>43</b>
<b>14</b>	<b>AutoPing</b>	<b>44</b>
<b>15</b>	<b>System log</b>	<b>48</b>
<b>16</b>	<b>Locking down the controller</b>	<b>49</b>
<b>17</b>	<b>Resetting settings to defaults</b>	<b>51</b>
<b>18</b>	<b>Specifications</b>	<b>52</b>
<b>19</b>	<b>Open source code</b>	<b>53</b>
<b>20</b>	<b>Technical support</b>	<b>54</b>
<b>21</b>	<b>Limited five year warranty</b>	<b>55</b>

## 1 Overview

Congratulations on selecting the DLI Web Power Switch, a surge-suppressed, AC power switch with automatic reboot and programmable power control.

Check out the main product [features](#). Some more technical [specifications](#) are available as well.

Be sure to check the [package contents](#) when unpacking the unit, then follow the instructions for [basic setup and use](#) if you are a first-time user. The device is controllable using LCD and keypad, which have several [modes](#). This should allow you to access the basic device features. More advanced settings are located on the [Setup page](#).

Web Power Switch Pro can be:

- extended with [user scripts](#);
- [customized](#) to add branding;
- accessed via a growing number of [external APIs](#);
- [upgraded](#) to newer firmware versions.

It can be configured to:

- [ping](#) other devices and take action if they don't respond;
- send [notifications](#) when certain events occur;

Web Power Switch Pro utility functions include:

- reading the [system log](#)
- [setting date/time](#);
- [backing up](#) settings;
- [locking down](#) security-sensitive functionality;
- [resetting](#) settings to defaults.

The firmware is based on [open-source code](#) which is provided to give you the option to build totally custom firmware.

Please contact [technical support](#) in case of any problems.

We offer a [limited five-year warranty](#) on these units.

## 2 Product features

Congratulations on selecting the DLI Web Power Switch, a surge-suppressed, AC power switch with automatic reboot and programmable power control. Its features include:

- 8 Switched Outlets + 2 Unswitched Outlets

Eight individual switch control circuits are provided with single outlets. Outlets are spaced for plugs and adapters. Un-switched outlets are provided for "always on" devices.

- Simple Web Interface

The internal web server is accessible from any browser. Simply enter an IP. Configuration and control are web-based.

- AutoPing™ Reboot

AutoPing continuously monitors an IP address. If a server, router, or other peripheral goes down, AutoPing can automatically reboot it without user intervention. Several devices can be monitored simultaneously.

- Programmable LCD Display

A 2x16 LCD displays status for each outlet.

Custom messages can be displayed via user scripts.

- Multi-User Password Security

Multi-user authentication limits access to the power controller. The administrator selects which outlets each user can control.

- Sequenced "On Timer"

A programmable delay timer allows outlets to be switched on in sequence, rather than simultaneously. Most devices draw a surge of power when initially switched on. Using this timer, more equipment can share a single circuit without overloads. Programmable scripts can be used to create customized power-up and shut-down sequences with variable timing.

- MOV Surge Suppression

Dual 3600W metal oxide varistors clamp power surges and spikes, protecting attached devices.

- Scripting Language, Syslog, and Utilities

Lua scripting can be used to create custom control and reboot sequences, schedule periodic reboots, etc. Internal and external event logs are provided.

- New Features

New features include HTTPS, WiFi support and Lua scripting.

- Field Upgradeable Firmware

Firmware is field upgradeable via Ethernet or WiFi.

### 3 Package contents

- Web Power Switch Pro.
- Power Cord.
- RP-SMA WiFi Antenna.

Please contact the freight carrier immediately if your package appears opened or damaged in transit. Call DLI at (408) 330-5599 for tech support, service, and hardware upgrades.

## 4 Basic setup and use

### 4.1 Factory defaults

The factory default network configuration is as follows:

- wired network: fixed IP address 192.168.0.100, netmask 255.255.255.0;
- wireless network: fixed IP address 192.168.254.1, netmask 255.255.255.0.

You can log in with username `admin` (lower case) and password `1234`. It is recommended that you change the password. You will be reminded to do so by a big red banner on the top of each page.

To reset to factory defaults, gently press the reset-to-defaults button below the LCD to enter the reset menu, then select a reset option.

### 4.2 Initial setup

Use these shortcuts if you are an experienced installer. We recommend reading the entire manual for first-time installation.

- Unpack. Save the carton.
- Attach the power cord.
- Attach an Ethernet cable from the controller to your LAN. Switch power on. If you are attached through a switch, you may need to cycle switch power to establish a connection.
- Ping the default address 192.168.0.100 to confirm that a network connection is established. If you don't receive a response, proceed to the [IP setup](#) section below.
- Log in to the power controller using the default user name `admin` and the password `1234`. Note: "admin" must be entered in lower case.
- Click the Settings link to reach the configuration page. Select the safest power-loss configuration for your installation: all OFF, all sequential ON or pre-powerloss state.
- Configure the power switch as described below. After each change, click Submit and wait for the page to refresh before continuing.

Tip: A three-bulb electrical safety-tester is handy for configuring the controller before attaching your equipment.

### 4.3 IP setup

If your network settings won't access the default IP, use a direct cable connection (temporarily bypass any switch or router) and follow these steps to add a compatible static IP, such as 192.168.0.50.

#### 4.3.1 Windows IP setup

Before adding an IP, close all programs and browsers. After the link is established, you can enable DHCP.

#### 4.3.1.1 Locating IP settings

In Windows, the first step is locating the network adapter TCP/IP properties. The procedure differs for each Windows version:

Windows XP, 2000, 2003:

- Open Start / Control Panel / Network Connections.
- In "classic view", select Start / Settings / Control Panel / Network Connections.
- Right-click on Local Area Network Connection and select Properties.
- Proceed to step 2.

Windows Vista:

- Open Start, right click on Network, then on Properties.
- Double click Network and Sharing Center.
- Click Manage Network Connections. A Network Connections window appears.
- Right click on the network connection to the switch, i.e. Local Area Network.
- Proceed to step 2. Windows 7:
- Open the Start orb, click on Control Panel.
- Click View Network Status and Tasks, then Change Adapter Settings.
- Proceed to step 2

Windows 8:

- Mouse or swipe to the bottom right corner and select Settings.
- Select Control Panel.
- Select Network and Sharing Center.
- Change Adapter Settings.
- Right click on your connected network and select Properties.
- Proceed to step 2

#### 4.3.1.2 Configuring static IP

The second step is adding an IP such as 192.168.0.50. Temporarily disable DHCP while configuring the switch.

- Select Internet Protocol TCP/IP V4 Properties and click Properties.
- Enter a compatible static IP such as 192.168.0.50.
- Click Apply and close windows.
- Ping the power switch to confirm the connection.
- Point your browser to 192.168.0.100
- Log in.

Detailed instructions are at [http://digital-loggers.com/ip\\_setup.html](http://digital-loggers.com/ip_setup.html)

### 4.3.2 Mac OS X IP setup

- Turn AirPort off temporarily.
- Click the Apple logo, then System Preferences, then Network.
- Select Built-In Ethernet and then Configure.
- Under the TCP/IP tab, select Manually
- Enter an IP address such as 192.168.0.1 as shown:
- Make changes shown.
- Point a browser to 192.168.0.100 Log in.

Find Mac setup details at [http://digital-loggers.com/mac\\_ip\\_setup.html](http://digital-loggers.com/mac_ip_setup.html)

## 4.4 Windows IP configuration (2000, 2003, XP, Vista)

If your default Windows settings won't access the controller, use a crossover cable and follow these steps to reach the controller's IP:

1. Close network programs and browsers
2. Go to Network Settings -> Local Area Network.
3. Use the keyboard shortcut - type "ncpa.cpl" and click OK.
4. Right click on your LAN connection and choose "Properties".
5. Highlight "Internet Protocol" and click the "Properties" button.
6. Click the "Advanced" button.
7. Under the IP Address settings, click the "Add" button.
8. Enter a new IP, such as 192.168.0.10, and a subnet mask of 255.255.255.0.
9. Press the "Add" button; this new IP is added the list.
10. Close all windows for the configuration to take effect.
11. Start your Browser and type 192.168.0.100 in the URL field.


The default user name and password are "admin" (lower case) and "1234".



## 4.5 Basic switch operation

After power-up, the controller performs a sequence of self-tests to ensure reliability. The controller may then be operated via a web browser. To access the controller, simply enter the IP address in the URL field of your web browser, then log in. You will be presented with a screen similar to this:

**DLI DIGITAL LOGGERS, INC.** Web Power Switch Pro

**Controller: DLI Controller** 

Wed Feb 24 05:18:39 2016 Session expires in 00:29:48

**Individual Control**

#	Name	State	Action
1	Outlet 1	OFF	<a href="#">Switch ON</a>
2	Outlet 2	OFF	<a href="#">Switch ON</a>
3	Outlet 3	OFF	<a href="#">Switch ON</a>
4	Outlet 4	OFF	<a href="#">Switch ON</a>
5	Outlet 5	OFF	<a href="#">Switch ON</a>
6	Outlet 6	OFF	<a href="#">Switch ON</a>
7	Outlet 7	OFF	<a href="#">Switch ON</a>
8	Outlet 8	OFF	<a href="#">Switch ON</a>

**Master Control**

[All outlets OFF](#)

[All outlets ON](#)

[Cycle all outlets](#)



Sequence delay: 3 sec.

**Navigation Menu:**

- [Outlet Control](#)
- [Setup](#)
- [Scripting](#)
- [Event Notification](#)
- [Customization](#)
- [External APIs](#)
- [Backup/Restore](#)
- [Firmware Upload](#)
- [Date/Time](#)
- [AutoPing](#)
- [System Log](#)
- [Logout](#)
- [Support](#)
- [Help](#)
- [Manual](#)
- [FAQ](#)
- [Product Information](#)
- [Digital Loggers, Inc.](#)
- [Source code](#)

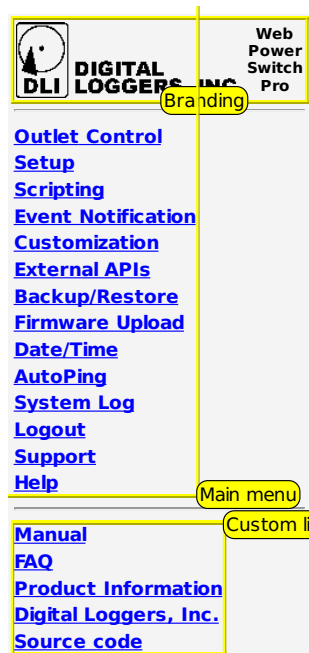
Version 1.4.1.0 / 1.4.1.0  
S/N:LPC92203123456

### Main power controller page

Note that in this and many other pages, logical blocks can be collapsed by clicking  in their right-hand corner, and later expanded by clicking .

## 4.6 Common page layout

You can navigate the menu links to access the controller's features:



Power controller menu

Custom links can be configured on the [Admin page](#). The branding block can be customized on the [Customization page](#).

## 4.7 Switching outlets on and off

The outlet control page lets you control outlets. The sequence in which outlets will be switched on is determined by settings on the [Admin page](#). To switch an outlet on or off, simply click to the right of the outlet name or number. Switching an outlet off is immediate. Switching an outlet on may be delayed if a different (or possibly even the same) outlet was recently turned on. The delay acts to protect the device from simultaneous inrush currents and limit cycling rate. You may also "cycle" a device which is connected to the controller. This feature is useful for rebooting Ethernet devices which may interrupt the web link to the controller. Clicking "Cycle" switches power off, waits a few seconds, and then switches power back on. This resets the attached device. You may also "cycle" all outlets using the "Cycle all outlets" button on the bottom of the page. Depending on your web browser settings, you may need to click the "refresh" button to update the on-screen status display after changing settings. A screen refresh setting is provided on the [Setup page](#).






## 4.8 Logout

Browser logout is automatic when a session is closed or after a time-out period. You can use a menu link to log-out in advance.

## 5 LCD and keypad

The LCD has 2 lines, 16 character positions each. The displayed data depends on the mode, and possibly also on [user scripting](#).

The keypad has 5 keys:

-  (UP),
-  (DOWN),
-  (ON),
-  (CYCLE),
-  (OFF).

During normal operation, the LPC9 LCD and keypad interface can be in one of the following modes:

- Outlet mode, which displays status of and allows to manipulate the unit's outlets;
- Network mode, which displays and allows to configure network settings.

You can cycle through the modes by pressing  and  keys simultaneously.



If the keypad is locked via web UI, the keypad doesn't work, and if you press a key, a message about this is displayed and stays in place until the next update of the LCD data; no action is taken.

### 5.1 Outlet mode

At boot, LPC9 starts in Outlet mode.





Outlet mode



In Outlet mode, the  and  keys allow to choose the outlet to display. The first line shows states of all outlets.

The selected outlet is marked by a blinking cursor. Its name is displayed on the second line.



If the outlet is not locked (see below) and its physical state matches the expected state, it is displayed as:



- a plus sign  for outlets that are on, or
- a minus sign  for outlets that are off.

If the outlet's physical state doesn't match the expected state, (e.g. it will be switched on in sequence ), its state is marked by:

- a minus/plus sign  for outlets that are physically off, but should be on, or
- a plus/minus sign  for outlets that are physically on, but should be off (this should be rare).

The  button switches the selected outlet on, likewise the  button switches it off.

You can press and hold each of the buttons for 3 or more seconds to lock the outlet in the corresponding state. Locked outlets can't be manipulated from web UI or with scripting, and won't be switched by using the hardware  /  keys unless you hold the corresponding key for 3 or more seconds to unlock it. Locked outlets are displayed as:

- a zero sign  for outlets that are locked off, or
- an asterisk  for outlets that are locked on.



The  button cycles the selected outlet unless it's locked.

Managing outlet lock state can only be done using the LCD and keypad (unless you enable SSH). Locked outlets' states cannot be altered, and they are not affected by [power loss recovery mode](#).

## 5.2 Network mode






Network mode











In Network mode, the  and  keys allow to select the network parameter. The selected parameter name is displayed together with its current value.







Network configuration submenu

Most values can be changed. Pressing the  button allows to change the parameter. The leading character of the first line, previously a whitespace , becomes an asterisk  to indicate that.

Different kinds of values are changed in different ways:

- for selectable options, like IP address mode, WiFi mode and encryption, the  /  keys cycle through the available options;
- for netmasks, the  /  keys increase or decrease the mask;
- for IP and MAC addresses, a virtual cursor is used to allow modification of individual address bytes; the cursor is initially placed on the last byte, and can be advanced to the preceding byte by pressing ; the  /  keys increase or decrease the current byte's value;
- for the WiFi pre-shared key, any of the ,  or  buttons generates a new key;
- the WiFi SSID can't be changed from keypad.

Pressing  saves the current value and starts network reconfiguration if needed. The display isn't updated instantly, and you need to refresh it by pressing  / . You can cancel editing at any time by pressing .

## 6 Setup page

The setup page allows the administrator to configure the power controller. These settings are supported:

### 6.1 Controller and outlet names

Unit Names		Confirm
Controller name	<input type="text" value="DLI Controller"/>	
Outlet 1 name	<input type="text" value="Outlet 1"/>	<input type="checkbox"/>
Outlet 2 name	<input type="text" value="Outlet 2"/>	<input type="checkbox"/>
Outlet 3 name	<input type="text" value="Outlet 3"/>	<input type="checkbox"/>
Outlet 4 name	<input type="text" value="Outlet 4"/>	<input type="checkbox"/>
Outlet 5 name	<input type="text" value="Outlet 5"/>	<input type="checkbox"/>
Outlet 6 name	<input type="text" value="Outlet 6"/>	<input type="checkbox"/>
Outlet 7 name	<input type="text" value="Outlet 7"/>	<input type="checkbox"/>
Outlet 8 name	<input type="text" value="Outlet 8"/>	<input type="checkbox"/>

#### Controller and outlet names

Use the controller name fields to assign a Controller Name to the power controller itself. Examples are "Server Rack Power Strip" or "Plutonium Refinery Control". The Controller Name field appears on the top of the home page. Assign a separate name to each outlet, such as "Missile Launcher" or "Email Server" to make identification of each circuit simple.

### 6.2 Delays

Delay	
Wrong password lockout	<input type="text" value="60"/> minutes. (0-60)
ON sequence delay	<input type="text" value="3"/> seconds. (1-255)
Cycle delay	<input type="text" value="0"/> seconds. (1-255)
Brown-out re-latch delay	<input type="text" value="10"/> seconds. (1-255)
Refresh screen every	<input type="text" value="1"/> minutes. (1-255)
Enable screen refresh	<input checked="" type="checkbox"/>

#### Delays

When a time value is entered in the "All ON sequence delay" field, the power controller will pause for a period of time before switching each outlet on in sequence. This delay helps prevent the power surges and blown circuit breakers which can occur when multiple devices are switched on simultaneously. A delay of 60 seconds is suggested for server applications. You may also enter a screen refresh delay in this section. If "Enable screen refresh" is checked, and a delay value is entered, your browser should periodically update the status screen.

### 6.3 Power loss recovery modes

Power Loss Recovery Mode⌵

When recovering after power loss	<input checked="" type="radio"/> Turn all Outlets off <input type="radio"/> Turn all Outlets on <input type="radio"/> Return to pre-powerloss state
----------------------------------	---

#### Power loss recovery modes

The power loss recovery mode setting has three settings which take effect after a power failure:

1. You can turn all outlets off (all systems will be switched off until manually turned on later) by checking the first box.
2. You can automatically turn all outlets on using the "All ON sequence delay" described above. Check the second option to do this.
3. You can return to the same outlet settings that were used prior to the power loss. The "All ON sequence delay" will also be used in this instance. Click the third option to return to pre-powerless state.

### 6.4 User-defined links

User Defined Links⌵

#	URL	Description
1	<input type="text" value="http://www.digital-loggers.co"/>	<input type="text" value="Digital Loggers, Inc."/>
2	<input type="text" value="/src.tar.gz"/>	<input type="text" value="Source code"/>

#### User-defined links

You may link to other power controllers, your own web pages, or remote web sites by entering up to four URLs and descriptions in the Setup page. For example, enter "Site Two Power Controller" in the description field with a URL of "http://192.168.0.250/". These links appear on every page of the main web UI.

## 6.5 Network settings

General Network Settings	
Hostname	<input type="text" value="power"/>
Location	<input type="text" value="Rack #3"/>
Contact	<input type="text" value="Joe Random &lt;joe@random.example&gt;"/>
HTTP Port	<input type="text" value="80"/>
HTTPS Port	<input type="text" value="443"/>
Enable SSH Server	<input checked="" type="checkbox"/>
SSH Port	<input type="text" value="22"/>
Syslog Server	<input type="text" value="192.168.0.2"/>
Syslog severity threshold	<input type="text" value="Debug"/>
Same subnet access only	<input checked="" type="checkbox"/> <b>FROM 192.168.0.0-192.168.0.255 (wired), 192.168.254.0-192.168.254.255 (wireless) ONLY</b>
Allowed SSH public keys	<div style="border: 1px solid #ccc; height: 150px; width: 100%;"></div>
<input type="button" value="Submit"/>	

### General network settings


You can adjust the HTTP and HTTPS port bindings. If left empty, the corresponding service is not accessible. It may be a good idea to disable HTTP if HTTPS satisfies your needs. Disabling both for security is possible; you can use the [LCD and keypad](#), or SSH to re-enable them if needed.

Enabling SSH will allow full control over the device, possibly bypassing most of the restrictions, e.g. setting protection. The SSH port is customizable as well. The SSH server also accepts public key authentication for a configurable set of keys (the format is the same as in the `authorized_keys` file).

You can limit the severity of the locally recorded log messages by setting a minimal severity. Note that the local log is circular, with old messages being replaced by newer ones; messages aren't persisted across reboots. For persistent storage, you can configure the unit to send the system log to a syslog server. All messages, regardless of severity, are sent; the receiver is expected to do the filtering.

Same subnet restriction can be used to prevent remote access from outside. **ONLY MACHINES IN THE SAME SUBNET WILL CONNECT AFTER ENABLING THIS.** If connectivity is lost, use a local connection such as a laptop with a crossover cable to restore your original network settings; you can also use the [LCD and keypad](#) for that.

### 6.5.1 Wired network settings

LAN Configuration 	
MAC Address	7C:E1:FF:00:00:00
Protocol	Dynamic IP/DHCP ▾
IP address	192.168.0.100
Subnet mask	255.255.255.0
Gateway	192.168.0.1
DNS servers, comma-separated	192.168.0.1
<input type="button" value="Submit"/>	

#### Wired network settings

The device MAC address is provided for reference only and cannot be changed in this form. If you need to change the MAC address, you may do so via LCD+keypad, SSH or using the REST-like API. Be sure you know what you're doing, as e.g. assigning a conflicting MAC address, or a broadcast MAC address will make an interface unusable.

To configure the unit to use static IP assignment, a fixed IP address and network mask must be entered. If a default gateway is specified it must be on the same subnet as the IP address specified. A number of DNS server IP addresses can be supplied separated by commas, e.g. 192.168.0.1, 8.8.8.8. If DNS servers are available, some other configuration variables can accept hostnames instead of IP addresses.

If you wish to configure the unit to use DHCP IP assignment, you needn't change the IP, network mask, default gateway and DNS servers; rather, after the unit obtains a DHCP lease, the parameters will be displayed for reference.

It is recommended to configure the DHCP server to provide a static lease for the LPC9 using its MAC address (also displayed).

When changing IP addresses, you may need to restart the unit and your network switch to validate the new IP on an "auto-configuring" switch port.



### 6.5.2 Wireless network settings

WiFi Configuration <span style="float: right;">^</span>	
WiFi module enabled	<input checked="" type="checkbox"/>
MAC Address	7C:E1:FF:00:00:00
Protocol	Static IP
Channel	11 (2462 MHz)
Mode	Access Point
SSID	DLI_LPC956
Encryption	WPA PSK
Encryption Key	••••••••
IP address	192.168.254.1
Subnet mask	255.255.255.0
Gateway	
DNS servers, comma-separated	
<input type="button" value="Submit"/>	

#### Wireless network settings

The wireless network adapter has settings similar to those of the wired network adapter (see above), and adds WiFi-specific ones.

It's possible to disable the wireless module entirely by unchecking the "WiFi module enabled" checkbox. The wireless MAC address is configured to match the wired MAC address as the adapters will never be on the same subnet in a regular setup. Use LCD+keypad, SSH access or the REST-like API if you need to change that.

The WiFi module can operate either in Access Point ("server") or Station ("client") mode. Either way, the name of the wireless network to create/connect to must be specified as the SSID.

If the unit is configured to be an Access Point and have a static IP assignment, it starts a DHCP server on the wireless interface.

It is possible to use no encryption on the WiFi channel, or one of the WPA, WPA2 or WPA/WPA2 mixed mode with pre-shared key (the key has to be entered then). WEP encryption is considered insecure and is not supported. Other encryption modes are not supported.

### 6.5.3 Network settings protection

You may press the "protect" button to lock the network settings (this will also affect the external API settings). Once locked, the network settings cannot be changed except by pressing the physical reset button on the front of the unit.

### 6.6 Access control

The administrator's username and password can (and should) be changed from the default values. Note that you need to provide the current password for confirmation.

Administrator credentials	
Administrator login	<input type="text" value="admin"/>
Old administrator password	<input type="password"/>
New administrator password	<input type="password"/>
Confirm new administrator password	<input type="password"/>
<input type="button" value="Submit"/>	<input type="button" value="Protect"/>

#### Administrator credentials

In addition to the administrator, any number of users with individual passwords and outlet permissions may be configured on the setup page. Only the administrator can edit user names and passwords (users can only inspect and switch outlets).

Access control:										
User Name	Password	Controlled Outlets								Apply
		1	2	3	4	5	6	7	8	
<input type="text" value="tom"/>	<input type="password" value="....."/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="Change"/>
<input type="text" value="dick"/>	<input type="password"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="Change"/>
<input type="text" value="harry"/>	<input type="password" value="....."/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="Change"/>
<input type="text"/>	<input type="password"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="Change"/>

#### User access control

Checkboxes to the right of each user name outlet control access privileges. Users can only see and interact with the chosen outlets. For example, user 'harry' would see the following on login:

#	Name	State	OFF / ON	CYCLE
1	Outlet 4	<b>OFF</b>	<a href="#">Switch ON</a>	
3	Outlet 7	<b>OFF</b>	<a href="#">Switch ON</a>	
4	Outlet 8	<b>OFF</b>	<a href="#">Switch ON</a>	
<a href="#">Logout</a>		<a href="#">Help</a>	3 sec.	

#### User outlet control screen

Individual outlets can be manipulated as usual. The top links allow switching all accessible outlets on, off or cycling them.

The next group is comprised of miscellaneous settings for access control.

Access settings	
Allow legacy plaintext login methods	<input type="checkbox"/>
Hide user passwords	<input type="checkbox"/>
Hide WiFi password	<input type="checkbox"/>
Disable local keypad	<input type="checkbox"/>
Show device name on login page	<input type="checkbox"/>

### Access settings

If you need to access the controller with clients supporting Basic authentication, or by browsers without JavaScript, you may need to enable the "Allow legacy plaintext login methods" setting. Those methods transmit passwords over the network and are thus considered insecure. This includes Basic authentication over HTTPS, which is secure relative to Basic authentication over HTTP, but relies solely on TLS for security, which is considered risky by some experts. DLI Ethernet Sender should not require this setting.

The "Hide user passwords" and "Hide WiFi password" settings configure whether clients should be able to read back the relevant values; this may be a security issue if there are untrusted administrator users.

The "Disable local keypad" setting is designed for untrusted physical environments. When enabled, the LCD will briefly indicate that the [keypad](#) is disabled on each keypress and otherwise ignore it.

Note that this still leaves the [reset button](#) available to an attacker.

The "Show device name on login page" setting can be used to control if unauthenticated users can see the device name (it used to be controllable by a space character preceding the device name in previous controller models, but is now an individual setting). This may be a convenience, but also a possible security issue.

## 6.7 Miscellaneous settings

The following setting group controls aspects of data presentation of the unit.

Miscellaneous Settings	
Display LCD text in all CAPS	<input type="checkbox"/>
Meter and plot default image format:	<input type="text" value="PNG"/>

### Miscellaneous settings

You can force all the text displayed on the LCD to be in CAPS, which may be more legible. However, this won't affect the network settings, as they include the WiFi password which would be useless if capitalized.

## 7 Scripting

On its own, a power switch isn't very smart. Programmers can easily add custom functionality by using the built-in Lua-based scripting language in power controllers.

### 7.1 Hardware requirements

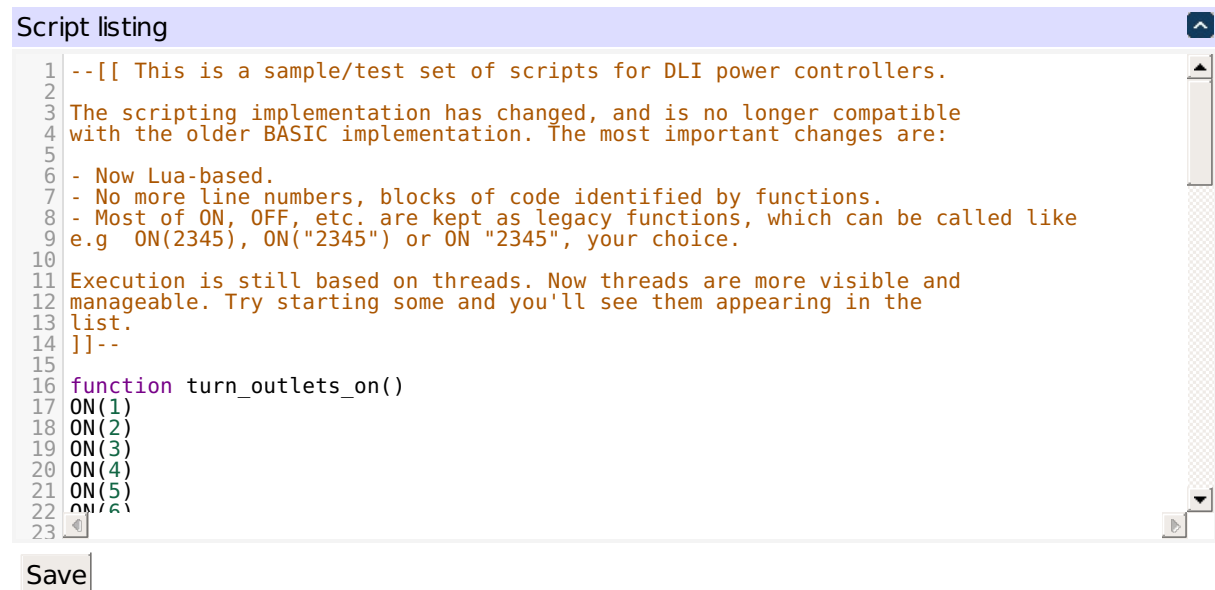
Lua-based scripting is available in all LPC9 controllers. Beeper, backlight, LCD, voltage and current monitoring functions are limited to products with appropriate hardware installed.

### 7.2 Entering scripts

First, for a quick overview of the script language visit the [sample scripts page](#) on the Digital Loggers web site. Log in as admin and use the [Scripting link](#) to access the programming page.

Scripting is based on the [Lua](#) programming language. A brief introduction is done below, but you may want to consult the general description, especially if you intend to write more complex scripts.

Script code is organized in functions. Configuration items which allow some scripting reaction to an event (reboot, autoping failure, etc.) will ask you for the name of the function to call (you will be offered a list of the functions defined in the script).



The screenshot shows a web-based script editor. At the top, there is a header 'Script listing' with a small upward arrow icon. Below the header is a text area containing script code. The code starts with a multi-line comment: '--[[ This is a sample/test set of scripts for DLI power controllers. The scripting implementation has changed, and is no longer compatible with the older BASIC implementation. The most important changes are: - Now Lua-based. - No more line numbers, blocks of code identified by functions. - Most of ON, OFF, etc. are kept as legacy functions, which can be called like e.g ON(2345), ON("2345") or ON "2345", your choice. Execution is still based on threads. Now threads are more visible and manageable. Try starting some and you'll see them appearing in the list. ]]' followed by a function definition: 'function turn\_outlets\_on()' and a list of 'ON' statements: 'ON(1)', 'ON(2)', 'ON(3)', 'ON(4)', 'ON(5)', 'ON(6)'. Below the text area is a 'Save' button.

```

1  --[[ This is a sample/test set of scripts for DLI power controllers.
2
3  The scripting implementation has changed, and is no longer compatible
4  with the older BASIC implementation. The most important changes are:
5
6  - Now Lua-based.
7  - No more line numbers, blocks of code identified by functions.
8  - Most of ON, OFF, etc. are kept as legacy functions, which can be called like
9  e.g ON(2345), ON("2345") or ON "2345", your choice.
10
11 Execution is still based on threads. Now threads are more visible and
12 manageable. Try starting some and you'll see them appearing in the
13 list.
14 ]]'--
15
16 function turn_outlets_on()
17 ON(1)
18 ON(2)
19 ON(3)
20 ON(4)
21 ON(5)
22 ON(6)
23

```

Save

#### User script source editor

You will need to define your functions to be able to use scripting. Simply putting calls to existing functions in the script will not work. Functions are defined like this:

```

function my_function()
    ... statements go here ...
end

```

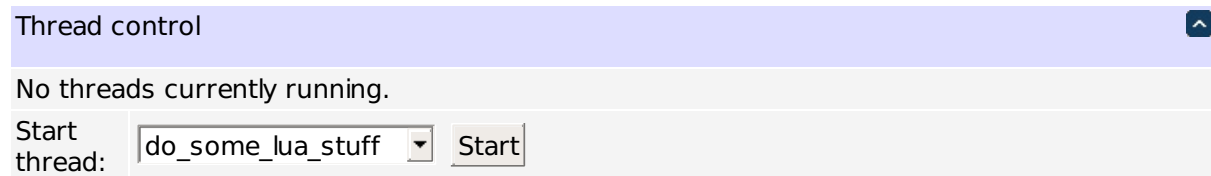
All functions defined this way will be usable from the web UI and callable externally. If you want to define a function for internal use, not to be called from outside, prefix it with `local`:

```
local function my_internal_function()
    ... statements go here ...
end
```

Functions in Lua are called with their arguments parenthesised, e.g. `func(arg1, arg2)`. Functions with no arguments are called with empty parentheses, like `func()`. However, according to Lua syntax, a single string function argument doesn't need parentheses, thus allowing the BASIC-like commands to avoid them if there's only one argument.

### 7.3 Threading

Multiple threads of execution can be running at the same time. Any number of threads may run concurrently.



#### User script thread list

Threads can be started from the web UI, via an HTTP request, by an [AutoPing](#) trigger, or from other threads using `thread.run`. They can be explicitly stopped using the web UI or by calling `thread.kill` or `thread.killall` from the script, or implicitly by calling `thread.limit`.

Every thread has an 'origin', which is usually a string identifying the function that started the thread. For instance, when you create a function like this:

```
function my_function()
    ... statements go here ...
end
```

and then start it with the web UI, its origin is the "my\_function" string. Threads created by other threads inherit their parent's origin, which can be useful when stopping a group of threads.

### 7.4 API levels

The scripting engine presents two sets of functions that you can use to write scripts:

- Legacy functions - functions which are designed to resemble the BASIC commands of the previous generations of EPCR/LPC controllers;
- Modern API - functions and objects which are designed to be easier to use.

You can use and even freely mix them as you wish, but only the modern API will receive further development attention. Some features are exposed only via the modern API because they had no corresponding legacy commands.

### 7.4.1 Legacy functions

Legacy functions (written in CAPS) are executed in sequence with a "step delay" after them. The legacy functions are designed so as to resemble the BASIC commands of the previous generations of EPCR/LPC controllers while remaining compatible with the Lua language.

Arguments to the legacy functions can be written as e.g. `ON(12345678)`, `ON "12345678"` or `ON ("12345678")`.

The supported legacy functions are:

- `ON`, `OFF`, `CYCLE`, `RESTORE` - perform the action on a list of outlets by numbers (as a number or a string);
- `BEEP (ON)` or `BEEP (on)` or `BEEP (true)` - turn beeper on;
- `BEEP (OFF)` or `BEEP (off)` or `BEEP (false)` or `BEEP (0)` - turn beeper off;
- `BEEP (number>0)` - turn beeper on for the specified number of seconds, then off;
- `SLEEP (number[, "unit"])` - suspend execution for the given amount of time (units default to "seconds", but can be "seconds", "minutes", "hours" or "days"; abbreviations like "sec", "h", "d" are also accepted);
- `LOG "String"` - write a message to the system log
- `DISPLAY "String"` - display a string on the LCD when it's in outlet mode. The following strings are expanded:
  - `%%` - literal "%";
  - `%o` - state of outlets, in the form "12456" (ON outlets are listed);
  - `%O` - state of outlets, in the form "+++++--";
  - `%n` - serial number;
  - `%f` - firmware version;
  - `%d` - system time/date;
  - `%M` - MAC address of the power controller;
  - `%i` - IP address of the power controller;
  - `%m` - IP network mask;
  - `%g` - IP gateway;
  - `\1` - move cursor to the beginning of line 1;
  - `\2` - move cursor to the beginning of line 2;
  - `\f` - clear screen;
  - `\v` - clear end of current line;
- `WOL "MAC address"` - attempt to wake device with specified MAC address up using Wake-on-Lan protocol (the device has to be in the same LAN segment);
- `TIME "server"` - synchronize time with server specified by IP address or hostname in quotes; you can use `TIME ()` without arguments to synchronize with "pool.ntp.org" if the DNS is configured correctly.

You still need to enclose the function contents in a `function name() ... end` as explained above.

## 7.4.2 Modern API

Modern API allows more object-oriented approach to scripting. You need to explicitly use the `delay()` function if you use the modern API and need a delay. Note that the outlet power-on sequence delay applies anyway.

Lua objects can have fields (data contained in the object) and methods (functions which affect the object's state). Object fields are accessed with a dot `.`, like `meter.reading`. However, different object implementations in Lua may use the colon `:` or the dot `.` to access the object's methods (`outlet:cycle()` or `outlet.cycle()`). In the modern API, all objects use the dot `.` to access their methods, to prevent confusion.

Modern API objects and functions are grouped into several categories for convenience.

### 7.4.2.1 Core Lua functions

To make scripting safer, only a limited subset of Lua features is supported by sandboxing. The following Lua standard library features are supported:

**Globals:** `_VERSION`, `assert`, `error`, `next`, `ipairs`, `pairs`, `pcall`, `xpcall`, `select`, `tonumber`, `tostring`, `type`, `unpack`.

**string library:** `string.byte`, `string.char`, `string.find`, `string.format`, `string.gmatch`, `string.gsub`, `string.len`, `string.lower`, `string.match`, `string.rep`, `string.reverse`, `string.sub`, `string.upper`.

**table library:** `table.insert`, `table.concat`, `table.maxn`, `table.remove`, `table.sort`.

**math library:** `math.abs`, `math.acos`, `math.asin`, `math.atan`, `math.atan2`, `math.ceil`, `math.cos`, `math.cosh`, `math.deg`, `math.exp`, `math.floor`, `math.fmod`, `math.frexp`, `math.huge`, `math.ldexp`, `math.log`, `math.log10`, `math.max`, `math.min`, `math.modf`, `math.pi`, `math.pow`, `math.rad`, `math.random`, `math.sin`, `math.sinh`, `math.sqrt`, `math.tan`, `math.tanh`.

**os library:** `os.clock`, `os.difftime`, `os.date`, `os.time`.

Additionally, `_G` points to the sandbox environment.

### 7.4.2.2 Delay functions

The `delay` function accepts the number of seconds to wait as an argument (it is assumed to be the script step delay if not specified). If a scripting thread doesn't call `delay()` or one of the legacy API functions every now and then, it can't be terminated by `thread.kill` and will eventually be shut down by the runtime.

### 7.4.2.3 Outlet management

The global variable `outlet` represents a Lua array of outlet objects, which have methods `on` (switch on), `off` (switch off), `cycle` (cycle), and two fields `state` (reflects the current state of the outlet) and `locked` (read-only field telling if the outlet is locked).

Global constants `on` and `off` are `true` and `false`, respectively, useful to make scripts more readable, like `outlet[1].state=on;`

#### 7.4.2.4 Thread management

Several threads can be executed simultaneously in a pseudo-parallel fashion. The global `thread` table contains these methods:

- `thread.run` can be used to start new threads; it accepts the thread function as argument, and returns the identifier of the resulting thread;
- `thread.kill` can be used to stop a thread; it accepts the identifier of the thread as an argument;
- `thread.killall` can be used to stop many threads; it accepts the origin of the threads to kill as an argument (without an argument, all threads are killed, including the calling one);
- `thread.limit` allows to ensure that no more than the specified number of threads with the same origin are present; its first argument is the maximum number of threads, and the second one is one of the strings "this", "earliest" or "latest", indicating which thread(s) should be killed if their count is above the limit (it's possible to specify an array of values, like {"this", "latest"}, instead).

#### 7.4.2.5 User interface

The global `ui` table provides means of configuring the LCD display, backlight and beeper.

Functions `ui.beep` and `ui.blink` configure the beeper and LCD backlight, respectively. Their first argument should be a string of "1"s and "0"s, which specifies the pattern, and their second argument should be the number of seconds after which the preceding behaviour is restored.

The `ui.line` table has two elements `ui.line[1]` and `ui.line[2]`, specifying the custom displayed strings for the LCD rows (or nil for regular operation of said row). This offers more fine-grained control than the `DISPLAY` command above.

#### 7.4.2.6 AutoPing integration

The global `autoping` table allows to query and configure AutoPing.

- `autoping.enabled`: boolean variable which allows enabling (`true`) or disabling (`false`) AutoPing;
- `autoping.items[N].enabled`: read-only boolean value indicating if the N'th AutoPing item is enabled;
- `autoping.items[N].enable`: function to call to attempt to enable (with the argument `true`) or disable (with the argument `false`) the N'th AutoPing;
- `autoping.items[N].addresses`: array of hostnames or IP addresses of the N'th AutoPing item's elements;
- `autoping.items[N].outlets`: array of outlets controlled by the N'th AutoPing item;
- `autoping.items[N].script`: the name of the scripting function run by the N'th AutoPing item when it's triggered (" " to cycle the controlled outlets);
- `autoping.items[N].status`: the run-time status of the N'th AutoPing item;
- `autoping.ping_interval`: ping interval;
- `autoping.ping_timeout`: ping timeout;
- `autoping.post_reboot_delay`: post-reboot delay;
- `autoping.max_reboot_count`: maximum reboot count;
- `autoping.pings_before_enabling`: pings before enabling.



### 7.4.2.7 Debugging

- `dump` - useful debugging function which outputs the argument to the system log, can be used to inspect state and even study the modern API itself (try `dump(_G)!`)
- `log` - contains methods `debug`, `info`, `notice`, etc. which accept a single string argument and log it at the corresponding severity level.

## 7.5 Starting scripts

There are a few ways to start scripts:

- On power up. This feature automatically starts a specified script function when power is first applied. The default is not to start any function, so pressing the "reset to defaults" button will disable this feature.
- By another thread. One thread can create another by using the `thread.run` function. For example, `thread.run(func1)` creates a new thread that starts executing the `func1` function. The execution of the parent thread continues.
- By issuing an HTTP request. Follow a link [http://Your\\_IP/script.cgi?run=func](http://Your_IP/script.cgi?run=func) to start execution from function `func`. This can be conveniently used by the end users by assigning the programmable web links on the left side of the page a target of the form `script.cgi?run=func`.
- Via AutoPing. The [AutoPing system](#) can be configured to automatically start execution when IP connectivity is lost. Enter the line number to execute in the Scripting box to the right of the corresponding IP on the AutoPing page.
- By manually clicking the Run button. Execution will start with the selected function.

## 7.6 Editing scripts

You don't need to disable scripting before editing scripts. If you make a syntactic error, the script won't be modified. Instead, you'll receive an explanatory message pointing to the error.

## 7.7 Stopping a thread

A thread terminates automatically when the end of its outermost function is reached. Click "Stop all running threads" to stop everything. You can also stop all scripts via HTTP using [http://Your\\_IP/script.cgi?stop](http://Your_IP/script.cgi?stop).

## 7.8 Relay debounce warning

Even with the scripting step delays, it is possible to create a script which will rapidly cycle a relay. This rapid cycling could result in a over current condition, tripped breaker, or stress to the power controller or attached equipment. Please be reasonable!

## 7.9 Configuration

The scripting server has the following configurable parameters:

Scripting settings	
Script step delay	<input type="text" value="1"/> seconds.
User message force display timeout	<input type="text" value="0"/> seconds.
User message timeout	<input type="text"/> seconds.
Start on reboot at	<input type="text" value="[Disabled]"/>
Trace script	<input type="checkbox"/>

### User script configuration

- Script step delay - the time in seconds to wait after execution of a legacy API function (e.g. ON, OFF). Modern API functions don't have internal delays; the `delay()` function should be used there.
- User message timeout - the time in seconds after which user messages (displayed e.g. with the DISPLAY command) disappear even if no keys are pressed on the LCD and no changes have been made to the outlet state (leave empty to have the messages displayed indefinitely).
- User message force display timeout - the time in seconds during which user messages are displayed even despite keys being pressed on the LCD or changes to the outlet state (leave empty to have the messages forcefully displayed indefinitely).
- Start on reboot at - the scripting function to start at cold boot.
- Trace script - enable diagnostic output about script progress to system log.

## 8 Event notification

Certain power controller events can trigger configurable notifications over a variety of media:

- email;
- XMPP (Jabber, Google Talk, etc.);
- SNMPv1, SNMPv2c and SNMPv3 TRAPs;
- SNMPv2c and SNMPv3 INFORMs;
- webhook.

Notification is based on the [Lua](#) programming language.

### 8.1 Notification context

When an event happens, a notification thread is started, with event's properties being copied to global variables (and thus constituting the context of further notification code snippets).

The most important of the rule action context (see below) is the `notify` function, which sends the current notification context to matching targets.

All kinds of events share the following properties:

- `id` — the event type identifier,
- `message` — a human-readable message,
- `severity` — the event severity level.

By manipulating the context, rules can check for and adjust the event's properties and prepare it for sending.

For example, you can have a rule with an empty condition and an action altering the 'message' variable like this:

```
message="Server room power: "..message
```

All the following rules will include the "Server room power:" prefix in the generated notifications.

Some properties, e.g. `message_short` or `message_long`, are supported by notification targets but aren't generated by any events; it's up to the action code to set them if necessary.

It's important to note that rules are processed linearly: adjustment of properties doesn't cause preceding rules to be re-examined.

Additionally, different event types have more specific properties. All event properties can be checked for in the or adjusted.

The following convenience severity level constants are defined:

- `EMERG, EMERGENCY` — "emergency" severity level,
- `ALERT` — "alert" severity level,
- `CRIT, CRITICAL` — "critical" severity level,
- `ERR, ERROR` — "error" severity level,
- `WARNING, WARN` — "warning" severity level,
- `NOTICE` — "notice" severity level,
- `INFO, INFORMATION, INFORMATIONAL` — "informational" severity level,
- `DEBUG` — "debug" severity level.

Their numeric values are defined so that a higher severity is larger, so condition like `severity>=CRITICAL` behaves like what you'd expect.

The [core Lua functions](#) are accessible from the context as well, should you need them.

## 8.2 Notification targets

The targets are a list of configurations which can deliver preprocessed messages to concrete recipients. Each target has a name, which needn't be unique. When a rule action calls `notify("target_name")`, all targets which have "target\_name" specified as name are triggered.

Notification targets				
#	Name	Type	Settings	Action
1	admin	Email	Recipient email address <input type="text"/> Sender email address <input type="text"/> Server <input type="text"/> Server port <input type="text"/> Username <input type="text"/> Password <input type="text"/> Transport encryption <input type="text" value="&lt;none&gt;"/>	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
2		SNMP v3 inform	Server address <input type="text"/> Authentication passphrase <input type="text"/> Authentication protocol <input type="text" value="&lt;none&gt;"/> Context engine ID <input type="text"/> Context name <input type="text"/> Privacy passphrase <input type="text"/> Privacy protocol <input type="text" value="&lt;none&gt;"/> Security engine ID <input type="text"/> Security name <input type="text"/> Default trap OID <input type="text"/>	<input type="checkbox"/>

### Notification target configuration

Other target properties depend on the kind of the target.

#### 8.2.1 Email notifications

Email notification targets have the following parameters:

- recipient email address (RFC 822);
- sender email address (RFC 822);
- server (hostname or IP address);
- server port (usually 465, 587 or 25 for unencrypted operations);
- username (for authenticating to the server);
- password (for authenticating to the server);
- transport encryption:
  - "smtps" for SMTP wrapped completely in TLS;
  - "starttls" for SMTP+STARTTLS command;

- "" (the empty string) for no encryption.

Some fields are autofilled for popular email services once you enter the sender's address.

If present, the `message_short` event property is used for the email subject; otherwise, `message` is used.

If present, the `message_long` event property is used for the email body; otherwise, `message` is used.

### 8.2.2 XMPP notifications

XMPP notification targets have the following parameters:

- recipient XMPP ID (RFC822-like);
- sender XMPP ID (RFC822-like);
- password (for authenticating to the server).

If present, the `message_short` event property is used for the message; otherwise, `message` is used.

### 8.2.3 SNMP notifications

The notification system supports sending:

- SNMPv1 TRAPs;
- SNMPv2c TRAPs and INFORMs;
- SNMPv3 TRAPs and INFORMs.

The difference between a TRAP and an INFORM is that an INFORM requires confirmation of receipt. Thus, the target test function can tell if the message has been delivered.

#### 8.2.3.1 Trap OIDs

The type of a TRAP or INFORM is indicated by its OID. SNMPv2c and SNMPv3 include the full OID in the message; SNMPv1 is different.

SNMPv1 traps are identified by the generic trap type (and correspond to the following trap OIDs):

- 0 — cold start (1.3.6.1.6.3.1.1.5.1),
- 1 — warm start (1.3.6.1.6.3.1.1.5.2),
- 2 — link down (1.3.6.1.6.3.1.1.5.3),
- 3 — link up (1.3.6.1.6.3.1.1.5.4),
- 4 — authentication failure (1.3.6.1.6.3.1.1.5.5),
- 5 — EGP neighbor loss (1.3.6.1.6.3.1.1.5.6).

If the generic trap type is 6, the trap is enterprise-specific (and is usually taken to correspond to OIDs using the template "1.3.6.1.4.1.ENTERPRISE-OID.0.SPECIFIC-TRAP-TYPE").

Unless you send only generic traps or have a OID tree registered to you, you may want to send 'user-configured' traps which carry no additional semantics other than the variable bindings. The `dlinGeneric` user-configured trap type is designed for that. Its OID is 1.3.6.1.4.1.45770.0.1; this can be specified directly for SNMPv2c or SNMPv3, or as an enterprise-specific (generic trap type=6) DLI ("enterprise OID"=45770) trap #1 (specific trap type=1).

### 8.2.3.2 Security settings

SNMPv1 and SNMPv2c use the 'community' security model which essentially identifies users by a shared secret which is sent over the network in plain text ('public' and 'private' being the most popular 'secrets'). That means that they are very insecure and shouldn't be deployed over an untrusted network. SNMPv3 has a more reasonable security model.

### 8.2.3.3 Variable bindings

All SNMP TRAPs and INFORMs accept the `snmp_values` event property to send extra values in the message. The property, if not nil, must be an array of the following shape:

```
{{oid1,value1,type1},{oid2,value2,type2},...}
```

The order may be important; you may want to consult the MIB. The following types are supported:

- "integer",
- "unsigned",
- "counter32",
- "string",
- "hex string",
- "decimal string",
- "nullobj",
- "objid" or "oid",
- "timeticks",
- "ipaddress" or "ip",
- "bits".

Type names are case-insensitive. A type may be omitted, in that case it will be inferred (nil values will be encoded as null objects, strings as octet strings, numbers as integers, "true" values as integer 1, and "false" values as integer 2 as per SMIv2).

### 8.2.3.4 SNMPv1 settings

SNMPv1 trap targets have the following parameters:

- server address (hostname or IP address of management station),
- community string (the shared secret for authentication to server),
- enterprise OID (number),
- default generic trap type (number 0..6),
- default specific trap type (number 0..2147483647).

SNMP v1 accepts `snmp_enterprise_oid`, `snmp_generic_trap_type` and `snmp_specific_trap_type` event properties to override the trap OID; they must be integers or string representations thereof. It also accepts the `snmp_values` event property as described [above](#).

### 8.2.3.5 SNMPv2c settings

SNMPv2c TRAP and INFORM targets have the following parameters:

- server address (hostname or IP address of management station),
- community string (the shared secret for authentication to server),
- trap OID (string)

SNMP v2c TRAPs and INFORMs accept an `snmp_trap_oid` event property to override the trap OID; it must be a string. They also accept the `snmp_values` event property as described [above](#).

### 8.2.3.6 SNMPv3 settings

SNMPv3 TRAP and INFORM targets have the following parameters:

- server address (hostname or IP address of management station),
- security name (username to authenticate as),
- security engine ID (leave empty for default),
- context name (leave empty for default),
- context engine ID (leave empty for default),
- authentication protocol ("MD5" and "SHA" are supported, leave empty "" for no authentication),
- privacy protocol ("DES" and "AES" are supported, leave empty "" for no encryption),
- authentication passphrase (should be at least 8 characters long, or empty if no authentication),
- privacy passphrase (should be at least 8 characters long, or empty if no encryption),
- trap OID (string)

You can either disable authentication and encryption, enable only authentication, or enable both. An encrypted, but not authenticated configuration is invalid.

SNMP v3 TRAPs and INFORMs accept an `snmp_trap_oid` event property to override the trap OID; it must be a string. They also accept the `snmp_values` event property as described [above](#).

## 8.2.4 WebHook notifications

Web hooks notifications send preconfigured POST HTTP requests to URLs.

- recipient URL — the URL to send a request to (must be an HTTP or HTTPS URL);
- content type — the type of content representation to use (must be "json" for JSON or "urlencoded" for URL-encoded).

The request payload (content) is a key-value map with the following keys:

- `type` — the `id` of the event type;
- `severity` — the severity label for the event (one of the strings "emergency", "alert", "critical", "error", "warning", "notice", "info" or "debug");
- `message` — the human-readable message of the event type;

If present, the `properties` event property is used to populate the payload with additional items; for instance, an action `properties={time=os.time() }` will result in a `time` key being added. Non-string values will be JSON-encoded (regardless of the content type).

If present, the `headers` event property is used to add HTTP headers to the request being sent.

If present, the `message_short` event property is used for the message; otherwise, `message` is used.

### 8.3 Notification rules

The rule system is centered around two kinds of entities:

- condition, which determines if a rule is matched, and
- action, which determines what happens if it matches.

Notification rules			
#	Condition	Action	Operations
1	outlet==1	message="The up escalator is down!" severity=CRITICAL	
2	severity>=CRITICAL	notify("admin")	
3			

Submit

#### Notification rule configuration

A condition is a Lua expression, while rules are Lua blocks (may contain several statements).

Roughly, the condition/action table is equivalent to:

```
if condition1 then action1 end
if condition2 then action2 end
if condition3 then action3 end
...
```

However, an important difference is that if a condition would cause an error, the condition is considered false instead; the corresponding action not taken, but the rule check goes on.

Additionally, an empty condition is equivalent to `true`, and the corresponding action is unconditionally taken. To disable an action without deleting it, you can use an explicitly false condition `false` or a condition that's not a valid Lua expression, e.g. `-`. To keep the condition text, you can wrap it with `false and (...)`.

Rules are applied from the first one to the last one, so order is important. The "Operations" column contains buttons which make manipulating rule order easier.

Though you can filter events by the type identifier, it's not necessary in many cases. For example, the condition `auth_allowed==false` will match only `dli.auth.login_denied` events, as `auth_allowed` is set to `false` for those events only. This is notably distinct from a `not auth_allowed` condition, which will match all sorts of messages which don't have an `auth_allowed` property.

Tricks which allow running code in the condition (as opposed to the action) are possible but discouraged.



## 8.4 Notification event types

Miscellaneous servers expose the types of events they can produce; they are presented on the notification configuration page in a compact form. Here's an example:

Authentication server events		
Severity	Message	Extra properties
NOTICE	login allowed for <u>auth_login@auth_ip</u> ( <u>auth_method</u> )	<u>id</u> ="dli.auth.login_allowed" <u>auth_allowed</u> =true
NOTICE	login denied for <u>auth_login@auth_ip</u> ( <u>auth_method</u> ): <u>auth_reason</u>	<u>id</u> ="dli.auth.login_denied" <u>auth_allowed</u> =false
NOTICE	protection violation attempt: <u>auth_reason</u>	<u>id</u> ="dli.auth.protection_violation"
INFO	<u>auth_login@auth_ip</u> session closed	<u>id</u> ="dli.auth.session_closed"

### Notification event types

Underlined items (both in the message and in the 'extra properties' column) specify properties which can be checked for. You can hover them for more detailed descriptions.

## 9 Customization page

The customization page allows the administrator to configure some user interface aspects of controller behavior.

### 9.1 Web page layout and branding

Customize page header	
Company name:	<input type="text"/>
Product name:	<input type="text"/>
Product URL:	<input type="text"/>
Logo (image URL):	<input type="text"/>
Logo width (default 195):	<input type="text"/>
Logo height (default 65):	<input type="text"/>
<input type="submit" value="Submit"/>	

#### Layout and branding settings

The branding block that appears on every web page can be customized:

- Product name: the displayed name of the product;
- Logo: the company logo image URL;
- Product URL: the URL that the image points to.

Custom logo dimensions may be supplied if needed. The logo URL may be absolute or relative (e.g. `/my_↔company.png`), in which case the related file should be placed in the `/www/static/` subdirectory of the unit's filesystem (probably via SSH).

Company name affects the alternative text for the logo image.

## 10 External APIs

The controller can be accessed programmatically using a number of protocols and APIs, including:

- the REST-like API (over HTTP),
- JSON-RPC (over HTTP),
- SNMP,
- UPnP

### 10.1 Common external API settings

External APIs	
Allow JSON-RPC:	<input checked="" type="checkbox"/>
Allow REST-style API:	<input checked="" type="checkbox"/>
Enable UPnP service:	<input type="checkbox"/>
Enable SNMP service:	<input type="checkbox"/>
Relax non-HTML method CSRF checks:	<input type="checkbox"/>
Relax non-HTML content type CSRF checks:	<input type="checkbox"/>

#### Common external API settings

Each of the external APIs can be enabled separately.

HTTP APIs perform cross-site request forgery checking to make sure they are not called by a misguided browser without JavaScript, bypassing browser security checks (a custom header needs to be present in the requests). Browsers can normally issue GET and POST requests with URL-encoded or multipart content types; you can tick the corresponding "relax ... checks" checkboxes to skip the checks in cases where the method or content type indicates that the request couldn't have been sent by a browser without JavaScript.

### 10.2 REST-like API

Name([view detailed description](#))

User-visible relay name

string

- represents a value in persistent storage
- write is denied if not [administrative user](#)
- no synchronization requirements

[Up to Relay object](#)

Allowed methods:

#### REST-like API demo

The REST-style API is based on the REST architectural style. It presents the state and configuration as an hierarchy of resources, and relies on HTTP to perform action signaling and content negotiation. Requests with different HTTP headers yield different representations of resources (e.g. plain text, HTML, JSON, etc.). A type description system is used to outline the object model.

Refer to the REST-style API reference for details.

### 10.3 JSON-RPC

JSON-RPC test (requires JavaScript) ⌵

Parameters	Request	Reply headers	Result										
<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid #ccc; padding: 2px;">URI:</td> <td style="border: 1px solid #ccc; padding: 2px;">/jsonrpc/ relay</td> </tr> <tr> <td style="border: 1px solid #ccc; padding: 2px;">Method:</td> <td style="border: 1px solid #ccc; padding: 2px;">get</td> </tr> <tr> <td style="border: 1px solid #ccc; padding: 2px;">"name"</td> <td style="border: 1px solid #ccc; padding: 2px;">Remove</td> </tr> <tr> <td style="border: 1px solid #ccc; padding: 2px;">Add new parameter:</td> <td style="border: 1px solid #ccc; padding: 2px;">Add</td> </tr> <tr> <td colspan="2" style="border: 1px solid #ccc; padding: 2px;">Perform RPC</td> </tr> </table>	URI:	/jsonrpc/ relay	Method:	get	"name"	Remove	Add new parameter:	Add	Perform RPC		<pre>POST /jsonrpc/relay HTTP/1.1 Content-Type: application/json-rpc Accept: application/json-rpc X-Requested-With: XMLHttpRequest Content-Length: 57  {"jsonrpc": "2.0", "id": 1, "method": "get", "params": ["name"]}</pre>	<pre>Connection: close Pragma: no-cache Allow: POST Expires: Mon, 01 Jan 1990 00:00:01 GMT Cache-Control: No-cache, no-store, must-revalidate, max-age=0 Transfer-Encoding: chunked</pre>	<pre>"DLI Controller"</pre>
URI:	/jsonrpc/ relay												
Method:	get												
"name"	Remove												
Add new parameter:	Add												
Perform RPC													

#### JSON-RPC demo

JSON-RPC allows to access an object model similar to the one of the REST-like API, but in a different manner which may be more suitable for some integration environments. All composite objects are visible using JSON-RPC, with their field values accessible using "get" (with the field name in the argument) and "set" methods (with the field name and value as arguments). Additionally, containers support "add", "remove" and "list" methods. The "describe" method can be used to output a type description for the object (similar to the REST API "description" relative URI).

### 10.4 UPnP settings

UPnP outlet binding configuration ⌵

#	Enable	Alternate name	Profile	Unique ID
1	<input checked="" type="checkbox"/>	Outlet 1	belkin_wemo_socket	b76cb5710
2	<input checked="" type="checkbox"/>	Outlet 2	belkin_wemo_socket	b76cb5711
3	<input checked="" type="checkbox"/>	Outlet 3	belkin_wemo_socket	b76cb5712
4	<input checked="" type="checkbox"/>	Outlet 4	belkin_wemo_socket	b76cb5713
5	<input checked="" type="checkbox"/>	Outlet 5	belkin_wemo_socket	b76cb5714
6	<input checked="" type="checkbox"/>	Outlet 6	belkin_wemo_socket	b76cb5715
7	<input checked="" type="checkbox"/>	Outlet 7	belkin_wemo_socket	b76cb5716
8	<input checked="" type="checkbox"/>	Outlet 8	belkin_wemo_socket	b76cb5717

Submit

#### UPnP settings

The unit's outlets can be exposed via UPnP as devices with different profiles. The currently supported profile is a Belkin WeMo socket.

### 10.5 SNMP settings

SNMP (simple network management protocol) exposes the control variables as a set of hierarchical resources identified by object identifiers (OIDs). An object identifier is roughly a sequence of non-negative integers (called arcs), separated by dots ('.'). A leading dot may be used to emphasize that it's an absolute OID; however, all of the OIDs configurable in LPC9 are absolute unless otherwise stated explicitly, and the leading dot is not needed, therefore, it's not supported.

SNMP OID subtrees				
ID	Description	Root OID	Read security level	Write security level
energyObject	energyObject MIB (RFC	1.3.6.1.2.1.229	Authenticated and encrypted	Authenticated and encrypted

#### SNMP OID subtree properties

SNMP v3 introduces a user-based security model, where a number of different users can exist whose requests can be signed, and possibly encrypted, and who can have different access rights to the OID tree.

The specified root OIDs and their children will be exposed over SNMP. All OIDs must be absolute but not preceded by a dot.

The root OIDs are actually treated as masks, indicating to set of roots to apply the permission to. In addition to the standard OID syntax, all but the first two arcs of an OID mask may contain:

- an asterisk "\*", which means that any value in this position will match, e.g. "1.2.\*.1" will match both "1.2.1.1" and "1.2.100.1";
- a dash-delimited range, e.g. "1.2.8.1-3" will match both "1.2.8.1" and "1.2.8.2";
- a comma-separated list of arcs, possibly including ranges, e.g. "1.2.8,9" will match both "1.2.8" and "1.2.9", and "1.2.1,6-8" will match both "1.2.1" and "1.2.7".

This can be used to implement fine-grained access to states of individual outlets (see below).

SNMP users							
Is allowed	Username	Authentication		Privacy		Access levels	
<input type="checkbox"/>	powerAdmin	SHA	Leave unchanged	AES	Leave unchanged	energyObject: Full	root: Forbidden
<input checked="" type="checkbox"/>	powerReader	SHA	Leave unchanged	AES	Leave unchanged	energyObject: Read-only	root: Forbidden
<input type="checkbox"/>	anotherUser	SHA	Leave unchanged	AES	Leave unchanged	energyObject: Read-only	root: Read-only
<input type="checkbox"/>		MD5	Leave unchanged	DES	Leave unchanged	energyObject: Forbidden	root: Forbidden

Submit

#### SNMP user table

The engine ID identifies the device, and plays an important role in SNMPv3, in particular in authentication and encryption. It will normally be autodetected by management software (SNMP clients), but you may save it for future reference (the default value is based on the device factory MAC address). You can even change it; however, if you do, all passwords for SNMPv3 users will be invalidated as they are stored in a localized form to improve security.

SNMP v1 and v2c do not have a notion of 'users'. Instead, a 'community string', acting as a shared secret, is transferred in requests in plain text. The following table allows to configure how community strings are mapped to the above users.

SNMP communities			
Community	IP address	Netmask	Mapped username
<input type="text" value="private"/>	<input type="text" value="192.168.0.0"/>	<input type="text" value="255.255.255.0"/>	<input type="text" value="powerAdmin"/>
<input type="text" value="public"/>	<input type="text" value="192.168.0.0"/>	<input type="text" value="255.255.255.0"/>	<input type="text" value="powerReader"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="powerAdmin"/>

### SNMP community-to-user mapping configuration

In this example, requests with the 'private' community string will be serviced as though they were made by the 'powerAdmin' user if they come from the 192.168.0.x subnet, and denied otherwise. Likewise, requests with the 'public' community string coming from the same subnet will be served as the 'powerReader' user.

## 10.6 SNMP energy object MIB support overview

The Net-SNMP agent included in LPC9 has built-in support for several well-known MIBs, but none of them deal with power control. The power-control-related ENERGY-OBJECT-MIB is described in RFC 7460, and is supported in the following manner:

- the root of the OID tree is at 1.3.6.1.2.1.229 as per RFC;
- the objects are outlets, with indices starting at 1 ;
- the current actual power states are indicated in the `eoPowerOperState` (1.3.6.1.2.1.229.1.2.1.9) table;
- the assigned (expected) power states can be manipulated in the `eoPowerAdminState` (1.3.6.1.2.1.229.1.2.1.8) table;
- supported power states are `ieee1621Off` (257) and `ieee1621On` (259) only

Additionally, among others, the following potentially useful parts of the above MIB are implemented:

- `eoPowerStateTotalTime`;
- `eoPowerStateEnterCount`.

These accumulate outlet state statistics. Note that those don't persist across device reboots.

The following parts of the above MIB are NOT implemented:

- `eoEnergyParametersTable`;
- `eoEnergyTable`;

- eoMeterCapabilitiesTable.

The following related MIBs are NOT supported:

- ENTITY-MIB;
- ENERGY-OBJECT-CONTEXT-MIB.

Additionally, modifying the user permissions via SNMP is NOT supported as they are generated from the configuration described above and the process is not easily reversible.

In the default configuration, the security level for accessing the energy object MIB subtree is high. You can set the access level to 'Minimal' to interact with the device using SNMPv2c and SNMPv1, or use SNMPv3 instead, which is the recommended and more secure alternative.

## 10.7 SNMP sample commands

These examples assume you have a at 192.168.0.100 with SNMPv3 user `powerAdmin` configured with SHA1 for authentication and AES for encryption, with password `powerAdminPassword` for both authentication and encryption. Requests with the `private` community string are assumed to be serviced as though they were made by the `powerAdmin` user.

You'll need Net-SNMP to run these samples; analogous commands should be available for other management software. The matching of requests vs SNMP protocol version is really arbitrary and is only used to demonstrate different ways of performing requests. Lines are broken using `\\` for readability. We use `-On` to force numeric OID output, and omit the leading '.' in output OIDs for simplicity.

An SNMPv2 SET to turn outlet #3 on:

```
$ snmpset -On -v 2c -c private 192.168.0.100 1.3.6.1.2.1.229.1.2.1.8.3 i 259
```

Output:

```
1.3.6.1.2.1.229.1.2.1.8.3 = INTEGER: 259
```

An SNMPv3 GET to get outlet #5 status:

```
$ snmpget -On -v 3 -u powerAdmin -l authPriv -a SHA -x AES \
-A powerAdminPassword -X powerAdminPassword 192.168.0.100 \
1.3.6.1.2.1.229.1.2.1.9.5
```

Output:

```
1.3.6.1.2.1.229.1.2.1.9.5 = INTEGER: 257
```

257 is `ieee1621Off`, so now you know the outlet is physically off.

An SNMPv3 SET to turn outlet #5 on:

```
$ snmpset -On -v 3 -u powerAdmin -l authPriv -a SHA -x AES \
-A powerAdminPassword -X powerAdminPassword 192.168.0.100 \
1.3.6.1.2.1.229.1.2.1.8.5 i 259
```

**Output:**

```
1.3.6.1.2.1.229.1.2.1.8.5 = INTEGER: 259
```

**Using SNMPv1 to enumerate the actual power states table:**

```
$ snmpwalk -On -v 1 \  
-c private \  
192.168.0.100 \  
1.3.6.1.2.1.229.1.2.1.9
```

**Output:**

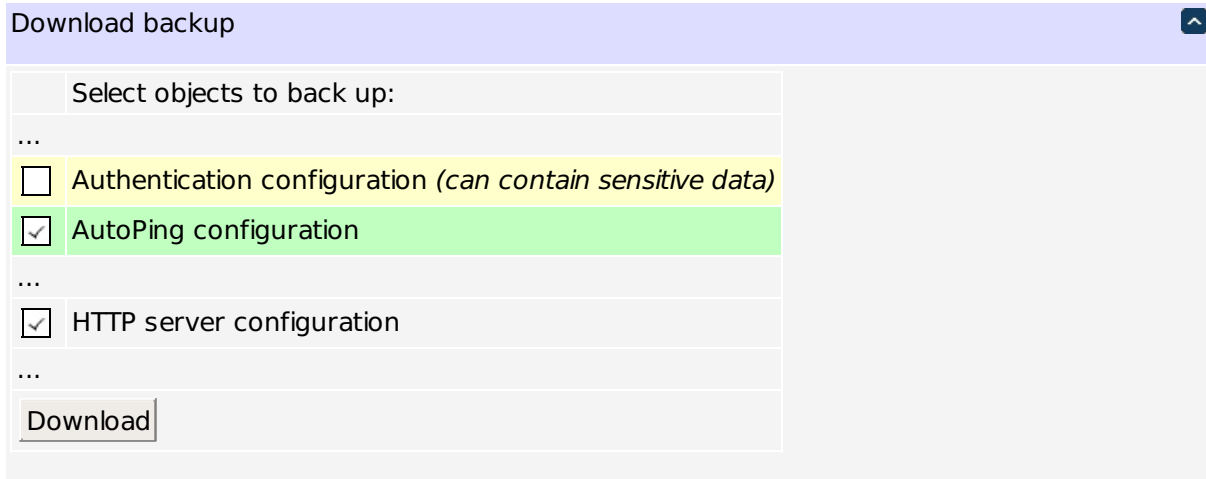
```
1.3.6.1.2.1.229.1.2.1.9.1 = INTEGER: 257  
1.3.6.1.2.1.229.1.2.1.9.2 = INTEGER: 257  
1.3.6.1.2.1.229.1.2.1.9.3 = INTEGER: 259  
1.3.6.1.2.1.229.1.2.1.9.4 = INTEGER: 257  
1.3.6.1.2.1.229.1.2.1.9.5 = INTEGER: 259  
1.3.6.1.2.1.229.1.2.1.9.6 = INTEGER: 257  
1.3.6.1.2.1.229.1.2.1.9.7 = INTEGER: 257  
1.3.6.1.2.1.229.1.2.1.9.8 = INTEGER: 257
```

You see that outlets 3 and 5 are on, and all others are off ; you can change outlet states as described above.



## 11 Backing up settings

The setting backup/restore system operates on a file level. It allows saving and restoring most configuration items, including those which have been done manually, e.g. via SSH. Settings can be backed up and restored selectively.



Download backup ⤴

Select objects to back up:

...

Authentication configuration (*can contain sensitive data*)

AutoPing configuration

...

HTTP server configuration

...

Download

### Backup setting selection (sample)

Modified files are highlighted in green. Choose the setting files you want to save (unknown files are shown as "File "+filename) and click "Download".

You may protect the security-sensitive parts of the configuration from being stored in a backup. Pressing the hardware reset button will be required to unlock.

## 12 Firmware upgrade

The controller's firmware can be upgraded to a newer version by first uploading it, and then committing the upgrade.

### 12.1 Uploading the firmware

Upload new firmware		
Firmware file:	<input type="button" value="Choose File"/> No file selected	
	OR	<input type="button" value="Upload"/>
Firmware URL:	<input type="text"/>	

Firmware upload page

You can upload a file or specify a URI where it can be downloaded from. Be sure to disable the [same subnet restriction](#) if you intend to download firmware from a server not in your local network.

By default, the unit beeps and blinks during update. You can disable this using the following form.

Configure update process	
<input type="checkbox"/>	Beep during upgrade
<input type="checkbox"/>	Blink during upgrade
<input type="button" value="Save"/>	

Firmware update configuration page

You may protect the current firmware from modification. Pressing the hardware reset button will be required to unlock.

### 12.2 Committing the firmware upgrade

After the firmware has been uploaded, you are presented with a form to perform the upgrade.

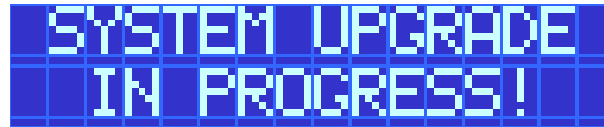
Firmware check OK  
Upgrading from 1.3.1.0 to 1.3.2.0.

Perform firmware update	
<input type="button" value="Update"/>	<input type="button" value="Cancel"/>


Firmware upgrade page

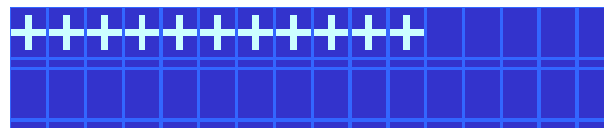
You can double-check the firmware version. If you wish to cancel the upgrade, it's best to explicitly press 'cancel' in the form so that the uploaded file could be cleaned up.

When you start an upgrade, a message about this briefly appears on the LCD.

A blue grid representing an LCD screen. The text "SYSTEM UPGRADE" is displayed on the top line, and "IN PROGRESS!" is displayed on the bottom line. The characters are white and spaced out within the grid cells.

**Firmware upgrade start indication**

Further activity is indicated by a cycling pattern of  signs.

A blue grid representing an LCD screen. The top row contains a series of white plus signs (+) that are cycling in and out of the grid cells from left to right, indicating progress. The rest of the grid is empty.

**Firmware upgrade progress indication**

After an upgrade has been completed, the unit is rebooted. The first boot may take longer than usual due to configuration updates.

A blue grid representing an LCD screen. The text "UPGRADE COMPLETE" is displayed on the top line, and "FIRST TIME INIT" is displayed on the bottom line. The characters are white and spaced out within the grid cells.

**Post-firmware upgrade initialization indication**

## 13 Date/time

The Date / Time page allows the administrator to set the internal clock and time zone. The clock may be set within the browser or synchronized with an NTP server.

Date and Time ⤴

Date  /  /  (MM/DD/YYYY)

Time  :  :  (HH:MM:SS)

### Date/time

The default OpenWrt configuration is to use the NTP servers `(0|1|2|3).openwrt.pool.ntp.org`. See [Scripting](#) scripting to perform manual synchronization.

The preferred time zone can be customized as well.

Time Zone ⤴

### Timezone

The selected timezone is used for header date/time display and formatting time in plots.

Internally, the time zone is stored in a format different from the display (it has a different meaning for + and -). You should take that into account when interpreting related REST API values and log output.

## 14 AutoPing

AutoPing can monitor a network device and perform a task if the device stops responding. It can also monitor a group of devices, the task will be executed if none of the group members respond. The task is either a list of outlets to reboot or a script to execute.

### 14.1 Common configuration

AutoPing Properties <span style="float: right;">⌵</span>		
Enable AutoPing:	<input checked="" type="checkbox"/>	
Time between pings:	<input type="text" value="30"/>	seconds. (2-3600)
Ping timeout to reboot:	<input type="text" value="150"/>	seconds. (2-3600)
Ping responses to enable autoping:	<input type="text" value="5"/>	pings. (0-100)
Times to attempt reboot:	<input type="text" value="5"/>	tries. (1-255)
Device reboot delay:	<input type="text" value="120"/>	seconds. (1-43200)
<input type="button" value="Apply"/>		


#### Common AutoPing settings

Be sure to enable AutoPing operation by ticking the "Enable AutoPing" checkbox. Certain [reset procedures](#) may turn it off automatically.




The following parameters are used for AutoPing operation:

- **Time between pings:** This is the time between each ping check of the IP address. 60 seconds should be useful for most applications. If multiple AutoPings are in use, each IP will be checked in sequence. For example, 2 autoping IPs and a setting of 60 seconds will cause each IP to be checked every 120 seconds.
- **Ping failures before reboot:** This sets the number of failed communication attempts that must be sequentially detected before a task is executed. For example, when set to 5, the target system must fail to respond 5 times in a row before it is rebooted. Since occasional network overloads and missed packets can occur during normal network operation, a number between 5 and 10 pings is recommended.
- **Ping responses to enable autoping:** To ensure a reliable connection, autoping will only be enabled after this many successful pings. We do not recommend changing this (10 is default) unless you must configure your controller before connecting it to the target devices.
- **Times to attempt reboot:** If you have an unreliable target device, limit the number of times it will be rebooted by entering that value here. For example, entering 5 will execute the task up to 5 times before giving up.
- **Device reboot delay:** After rebooting a device with a cold-boot power-off, a waiting period should occur before the IP address is re-checked by AutoPing. This delay allows the device to reboot. Windows and Linux servers can force automatic file system checks which may take several minutes to complete. Enter a safe value here, for example entering 600 would cause the power controller to start checking the server for normal operation 10 minutes after reboot. If a script is to be triggered, any delays contained in the code being executed should be considered in determining the delay setting here so that the thread completes before the delay elapses. This timer starts at the execution of the thread started.


## 14.2 Ping target configuration

To actually use AutoPing, add one or more AutoPing targets (IP addresses) to the list. The  button is used to remove a target from the list.

Below is an example autoping configuration with four targets:

AutoPing													
IP(s)	Reboot Outlets								Script	Action	Stats		
	1	2	3	4	5	6	7	8			TX	RX	HIT
<input type="checkbox"/> 74.125.87.103											790	578	
<input checked="" type="checkbox"/> 67.122.199.250	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	switch_off	<input checked="" type="checkbox"/> 	642	583	2
<input type="checkbox"/> 192.168.0.92	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	[Cycle]	<input checked="" type="checkbox"/> 	215	41	5
<input checked="" type="checkbox"/> 192.168.0.93	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	toggle_stuff_and_log	<input checked="" type="checkbox"/> 	823	822	0
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	[Cycle]	<input checked="" type="checkbox"/>	0	0	0

### Individual AutoPing settings

The checkbox to the left of the IP address is used to start/stop target monitoring. Confirm your action with  button. This button is also used to link a list of outlets or a script line to the autoping target.

You can select the outlets to perform trigger action on by ticking their respective checkboxes.

You can select a scripting action to perform when the AutoPing item triggers (by default the selected outlets are cycled). The action must be a function defined in the scripting server. It will receive a table of the 1-based indices of outlets selected (e.g. {1, 3, 6}). The order of outlets in the table is unspecified; use `table.sort` in the script function if you rely on a particular order.

The stats column shows some statistics:

- *TX* — the number of pings sent to the target IP address;
- *RX* — the number of pongs received back so far;
- *HIT* — the number of times the trigger action was executed.

On the sample image, three targets are being monitored (74.125.87.103, 67.122.199.250, and 192.168.0.93). 192.168.0.93 seems to be a very reliable/well-connected device: 823 pings were sent to it and 822 pongs received back. Chances are very good, the 823rd pong will arrive soon. The reboot task (script function `toggle_stuff_and_log`) was never executed.

Looks like 192.168.0.92 failed hard. The task (cycle outlets 3,5,6) was executed 5 times in a row but the target did not respond. Monitoring was automatically disabled.

74.125.87.103 and 67.122.199.250 form a group, the trigger task will be performed if they both lose 5 sequential packets simultaneously. This has happened 2 times so far. Monitoring a group of several external spatially separated reliable IP addresses (in this example they belong to Google and Digital Loggers respectively) may become very useful to detect a stuck ADSL modem or some other no-Internet condition.

### 14.3 Action on local network failures

AutoPing is designed to control operation of remote hosts. You usually don't want to e.g. cycle power to all servers if you turn on same subnet restriction. So AutoPing tries not to trigger if there might be a problem local to the unit itself. For example, if you detach the Ethernet cable from the unit, you'll see messages similar to the following:

```
kernel: eth0: link down
config.net: Interface "eth0" is down
autoping: ping x.y.z.t: no usable route to host, ..., not considered a failure
```

and no actions will be performed. A similar situation will occur if you reconfigure the controller to use a new IP network from which old addresses are unreachable.

### 14.4 Advanced ping targets

AutoPing targets don't have to be IP addresses. If you enter a hostname, it will be resolved before sending each request. If the name resolution fails, it is assumed to be a local error and, as described [above](#), no action is taken. If a name is resolved to multiple IP addresses, a random one is chosen.

AutoPing defaults to checking targets using the ICMP protocol by default. A variety of other ping target kinds can be used if you specify a URL instead of simply an IP address or hostname. Supported URL schemes include:

- `icmp` — this is explicit specification of the "regular" ping protocol, e.g. `icmp://192.168.0.1` is equivalent to `192.168.0.1` (note that no trailing slash is used);
- `tcp` — this causes AutoPing to try to establish a TCP connection to the given port, e.g. `tcp://192.168.0.1:22` can be used to check that there's a service listening on TCP port 22 (usually SSH) of `192.168.0.1` (note that no trailing slash is used);
- `http` and `https` — this causes AutoPing to perform a HTTP/HTTPS GET request for the given URL, e.g. `http://www.digital-loggers.com/index.html` can be used to check that the web server is responding and can serve its main page.

### 14.5 AutoPing events

The most often encountered AutoPing events are:

- `pinging ... (timeout)`
- `ping ... succeeded (time)`
- `ping ... failed (time)`

The time is request round-trip time, in seconds. Note that it's purely informative and can't be used as a measure of target response time unless it has order of hundreds of milliseconds and above.

Several failures in a row trigger AutoPing actions which are reported with corresponding events:

- `item ... (addresses...) failed [failures/max]`
- `item ... (addresses...) failed over (max) times in a row, disabling`

As described [above](#), local network failures don't count toward failure count, but generate these notifications instead:

- no usable route to host, possibly due to local network outage, not considered a failure (when a request isn't being sent)
- ping ... not received (time), possibly due to local network outage, not considered a failure (when an outage occurred after a request has been sent)

The events associated with item trial before enabling are self-explanatory:

- item ... (addresses...) enable approved
- item ... (addresses...) enable cancelled
- item ... (addresses...) trial restarted due to address list changes



## 15 System log

Most events occurring during controller operation are logged into the system log.

```
System Log:
Jan 1 00:00:50 power user.notice snmp[1095]: Initializing...
Jan 1 00:00:50 power user.notice snmp[1095]: Starting...
Jan 1 00:00:51 power user.notice upnp[1100]: Initializing...
Jan 1 00:00:51 power user.notice upnp[1100]: Starting...
Jan 1 00:00:53 power user.notice wwv[1106]: Initializing...
Jan 1 00:00:53 power user.notice wwv[1106]: Starting...
Jan 1 00:00:56 power user.notice dnsmasq: DNS rebinding protection is active, will discard upstream RFC1918 responses!
Jan 1 00:00:56 power user.notice dnsmasq: Allowing 127.0.0.0/8 responses
Jan 1 00:00:57 power daemon.info dnsmasq[1145]: started, version 2.66 cachesize 150
Jan 1 00:00:57 power daemon.info dnsmasq[1145]: compile time options: IPv6 GNU-getopt no-DBus no-IDN DHCP no-DHCPv6 no-Lua TFTP no-contrack no-ipset no-auth
Jan 1 00:00:57 power daemon.info dnsmasq-dhcp[1145]: DHCP, IP range 192.168.254.200 -- 192.168.254.248, lease time 12h
Jan 1 00:00:57 power daemon.info dnsmasq[1145]: using local addresses only for domain lan
Jan 1 00:00:57 power daemon.info dnsmasq[1145]: reading /tmp/resolv.conf.auto
Jan 1 00:00:57 power daemon.info dnsmasq[1145]: using nameserver 192.168.0.1#53
Jan 1 00:00:57 power daemon.info dnsmasq[1145]: using local addresses only for domain lan
Jan 1 00:00:57 power daemon.info dnsmasq[1145]: read /etc/hosts - 1 addresses
Jan 1 00:00:57 power daemon.info dnsmasq-dhcp[1145]: read /etc/ethers - 0 addresses
```

### System log

Note that the system log buffer has a fixed size, and old entries are removed automatically as new ones appear. The display is periodically updated.

You can use the `logread` command to read the same data in an SSH session. `logread -f` will display new entries in real time.

## 16 Locking down the controller

### 16.1 Intended locking use cases

In some cases it's required to grant administrative access to multiple, possibly not completely trusted, parties. The settings described below are designed to somewhat limit what an administrator can do to the device.




### 16.2 Protection bits

The following operations available to the administrator pose an increased risk and can be protected:

- changing administrator credentials;
- changing networking settings;
- changing notification settings;
- backing up private settings (passwords, keys, etc.);
- restoring settings from backup;
- upgrading firmware;
- entering maintenance mode.

### 16.3 Protection status indication

Overall protection status is indicated in the top right corner of [each page](#):

-  — no protection bits active;
-  — some protection bits active;
-  — all protection bits active.

Clicking on the icon gives more detailed info:

<b>Firmware protection is disabled</b>
<b>Notification settings protection is disabled</b>
<b>Private configuration protection is enabled but ineffective: SSH is enabled, which could be used to bypass protection Firmware upload is allowed, specially crafted firmware could be used to bypass protection</b>
<b>Administrator credentials protection is disabled</b>
<b>Maintenance mode lock is enabled but ineffective: SSH is enabled, which could be used to bypass protection Firmware upload is allowed, specially crafted firmware could be used to bypass protection</b>
<b>Protection from restore from backup is enabled but ineffective: SSH is enabled, which could be used to bypass protection Firmware upload is allowed, specially crafted firmware could be used to bypass protection</b>
<b>Network settings protection is disabled</b>

#### Protection status details




## 16.4 Unlocking protection



You should use [the reset button](#) and select the "Clear lock bits" reset mode to clear protection bits. This, of course, requires physical access to the unit.

## 17 Resetting settings to defaults

The device's settings can be reset to defaults by pressing the reset button to the left of the LCD.


You may want to take a [backup](#) of your settings first.

The display displays an overview of possible actions with a ticker on the second LCD line. You may interrupt it by pressing ,  or .

Use  and , or short presses of the reset button, to select a reset mode. A description of the currently selected mode is displayed on the second LCD line.

The following reset modes are available:

1. Clear lock bits: Clear protection bits only. Other settings are preserved.
2. Reset network and scripting: Clear protection bits, reset network settings and admin login, disable autoping and scripts. Other settings are preserved.
3. Reset network and scripting + enable Wifi: Clear protection bits, reset network settings and admin login, disable autoping and scripts, and enable open WiFi access. Other settings are preserved.
4. Complete wipe: Reset all settings to factory defaults and remove any user files. All settings will be lost!
5. Complete wipe + enable WiFi: Reset all settings to factory defaults and remove any user files, then enable open WiFi access. All settings are lost!

To activate the selected reset mode, press  or the reset button and hold it.

If all settings are reset (the two last 'wipe' reset modes), the Subnet Restriction will be enabled to prevent remote access using the default password. **ONLY MACHINES IN THE SAME SUBNET WILL BE ABLE TO CONNECT AFTER RESETTING TO DEFAULTS.** If connectivity is lost, use a local connection such as a laptop with a crossover cable to restore your original network settings.

## 18 Specifications

Alert Beeper	73dBa at 12". Programmable.
Applications	Commercial, industrial, and residential remote power control and reboot. Indoor use only.
Circuit Breakers	Manual reset, 15A Thermal, UL
Clock / RTC	15 year Li battery
Controls / Display	Reset-to-factory-default switch, 2x16 Backlit LCD w/ PowerSave, 5 button keypad
Dimensions	Outside 1.6x5.8x13" 12.3" between mtg. holes
Enclosure	Aluminum chassis, double grounded. Injection Molded high-temp thermoplastic base plate. Vented 2 sides. Fanless.
Ethernet Interface	10/100 autosensing, Static IP, TCP port selectable, 8 pin RJ-45 w/ internal FCC filtering
Humidity	8-80% RH Operating. Non-condensing. No liquid exposure is permissible.
Input Power Cord	Removable C19 14AWG 6' length standard
Inlet and Outlet Rating	UL, CSA 15A, 120VAC only
Input Frequency	Power supply - DC-400Hz
Operating Temperature	-30° to 170°F, -34° to 77°C
Options - User	Right-Angle 19" rack-mount bracket.
Power Supply Rating	90-240V, AC/DC Autosensing
Password Transmission	Encrypted, base 64 or HTTPS
Power Dissipation	3.9W Typ Max (all on) <3 W idle
Power Fail Hold-Over	350ms minimum (all relays on)
Power-Up Modes	Last used settings, all power on or off, sequential on or run user-script ~30s after power-up
Relay Contact Spec	15-25A AC/DC, 1/2HP
Surge Protection	3600W Metal Oxide Varistor
Software Controls	Individual outlets on/off , all on/off /cycle. Net settings, Web UI
Size (Single-Pack Carton)	2.4x7.7x15.8"
Weight (bare)	Single unit 3.6 lbs excl. cord & packaging.
WiFi	Atheros 9331 2.4G 802.11n RP-SMA

FCC Note:

The LPC9 may only be used with

- the manufacturer supplied antenna (Gain: 2.0dBi), or
- a 50 Ohm antenna of equal or lesser gain.

## 19 Open source code

In compliance with the spirit of the GNU Public License, source code is provided together with the firmware itself (accessible using one of the user-configurable links). Note that it is placed on the read/write firmware partition, so certain operations (like full factory reset) may remove it.

Purchasing a TLA and signing an NDA from Atheros are highly recommended before attempting any custom development; however, they aren't required to build the firmware (only the bootloader).

DLI cannot provide warranty or technical support for modified units; this includes units with custom firmware.

## 20 Technical support

Please register. Painless on-line registration gets you:

- free tech support,
- access to firmware updates,
- and information when updates and new features become available.

To save time, please have a look at the product FAQ page solutions. You may FAX questions to (408) 541-8459 or email: [support@digital-loggers.com](mailto:support@digital-loggers.com).

For phone support, call (408) 330-5599 with the following so we can better serve you:

- The firmware version level installed in the power switch. This information can be found on the lower left corner of the outlet control page.
- A description of the Ethernet devices connected to your unit, for example, a 10/100 PC and crossover cable.
- A description of the WiFi devices connected to your unit, i.e. their manufacturers and model numbers.

## 21 Limited five year warranty

The terms of this warranty may be legally binding. If you do not agree to the terms listed below, return the product immediately in original unopened condition for a full refund. The purchaser assumes the entire risk as to the results and performance of the unit.

DLI warrants this power controller to be free from major defects. No agency, country, or local certifications are included with this unit. It is the responsibility of the user to obtain such certifications if necessary for the customer's application. Buyer acknowledges and agrees that he is solely responsible for proper use, certification and safety testing of components supplied by DLI. DLI's entire liability and exclusive remedy as to defective hardware shall be, at DLI's option, either (a) return of the purchase price or (b) replacement or repair of the hardware that does not meet DLI's quality control standards and has been returned through proper RMA procedures. DLI's liability for repair or replacement is to DLI's customer ONLY.

WARRANTY SERVICE DOES NOT COVER DAMAGE TO SCREW TERMINALS FROM EXCESSIVE TORQUE OR DAMAGE DUE TO EXPOSURE TO WATER OR VIBRATION.

NO SUPPORT IS PROVIDED FOR MODIFIED FIRMWARE. MODIFICATION OF FIRMWARE VOIDS ALL WARRANTY.

Warranty service requires an original invoice from DLI and an RMA number provided by technical support. RMA material must be shipped prepaid to DLI. RMA numbers are valid for 15 days from date of issue. This warranty does not cover products which are modified (including firmware modifications), subjected to rough handling, or used in applications for which they were not originally intended. Batteries are not covered under warranty. Physical damage caused by customer or in transit to DLI is not covered under warranty. Please insure your shipments.

No oral advice or verbal warranties made by DLI's employees, dealers, or distributors shall in any way increase the scope of this warranty. DLI makes no warranty as to merchantability or fitness for any particular purpose. DLI assumes no liability for incidental or consequential damages arising from the use or inability to use this product. This warranty gives you specific legal rights. You may also have other rights that vary from state to state. Since some states do not allow the exclusion of liability for consequential damages, some of the above limitations may not apply to you. This product is not qualified or intended for mobile, airborne, medical or aerospace use or FDA Class III applications.



