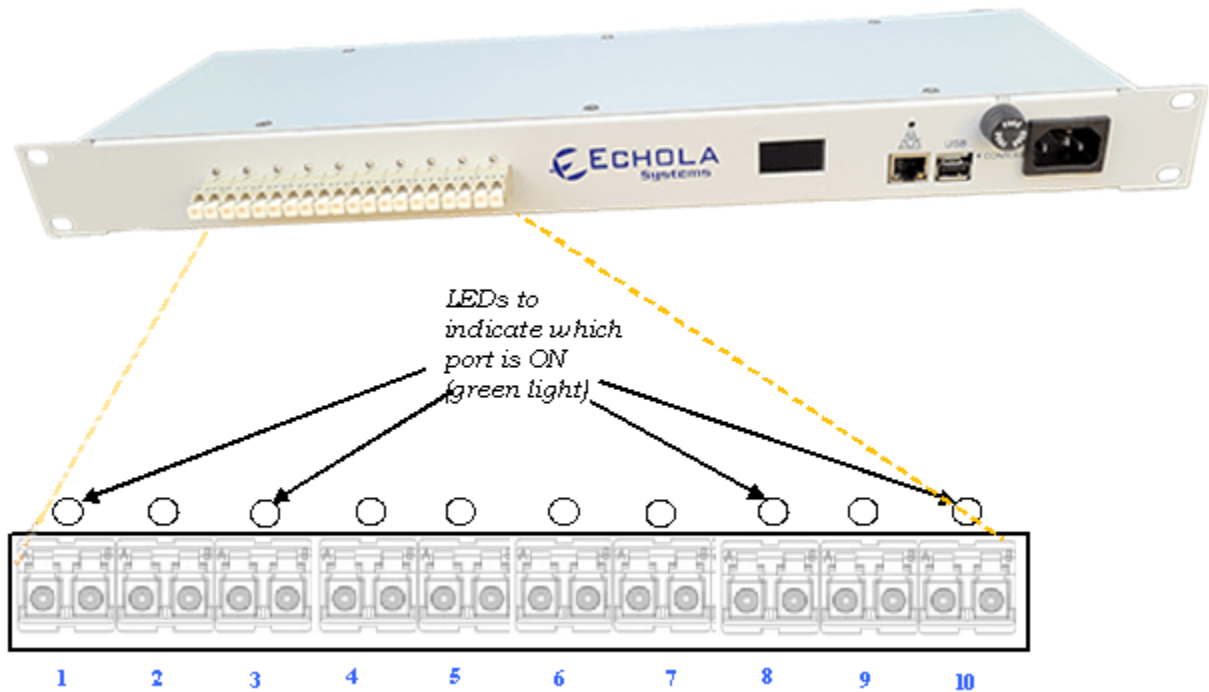




Echola Systems L.L.C
1161 Ringwood Ct, Ste 100
San Jose, CA-95131
Phone: 408 321 9663
Fax: 408 321 9663
<http://www.echola.com/optical>

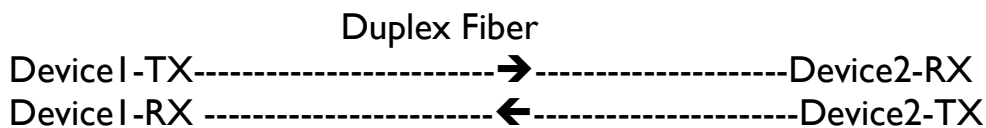
Installation

FC1011-MM is a multi-mode fiber-cut switch that uses 1x1 optical switch for each port. It supports up to ten ports in one compact enclosure (1U, 5 inches deep) that can be controlled or monitored individually. Some of the examples in this guide are related to the VFC series switch, which is not a 1x1 switch based but VOA based. So, if there is a mention of attenuation, it is specific to the VOA-based fiber-cut switch, not related to the FC1011-MM.

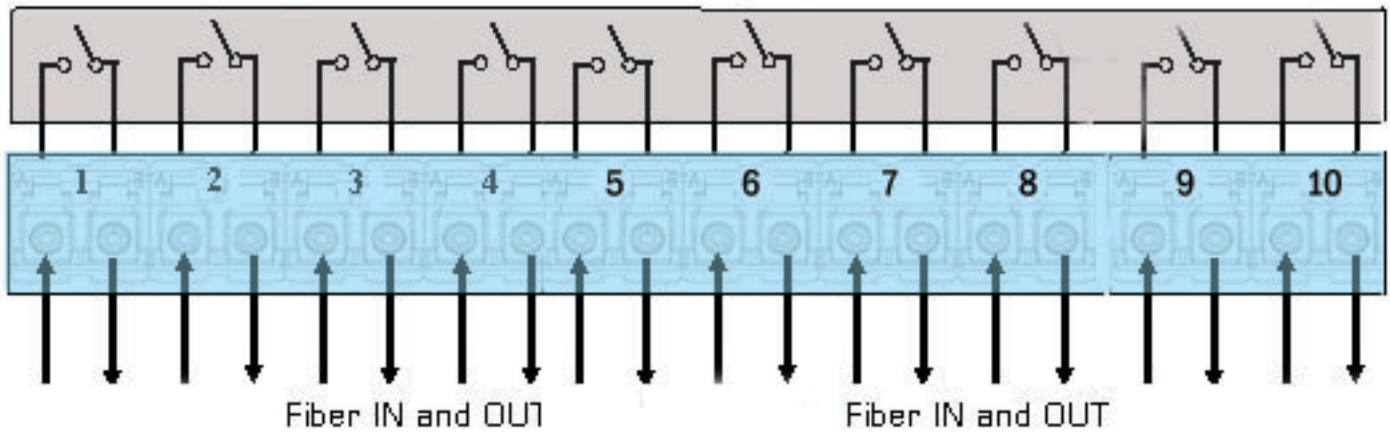


Connecting test equipment to FC1011

For example, if you want to simulate a fiber cut scenario between 2 optical devices using one fiber/one direction.



Each LC connector of FC1011 has two input/outputs marked as B and A.



Then you need to disconnect one of the fibers connecting these two devices and connect them through FC1011 port, between A and B like the following:

Device1-TX → [FC1011-Port1B→FC1011-Port1A] → Device2-RX

If you want to simulate a break between both TX and RX fibers of these devices under test (which may be the case in many scenarios), then you will have to use two ports of FC1011, like the following

Device1-TX → [FC1011-Port1B→FC1011-Port1A] → Device2-RX

Device1-RX ← [FC1011-Port2B←FC1011-Port2A] ← Device2-TX

UPSR, 2F-BLSR protection ring testing may require 2 FC1011 ports, while 4F-BLSR might need four ports. To test 100Gbps ethernet devices, you will require a total of 20 ports (2x FC1011s) for ten lanes in each direction. Similarly, for 40Gbps, you will need eight ports for four lanes in each direction.

Configuring FC1011

FC1011 provides a micro-USB console port and an Ethernet (10/100) port for management. USB console port is generally used in special situations such as to debug network connectivity if FC1011 is not reachable through Ethernet.

If you use USB console port, then you will need a USB-A to micro-USB cable. Use either putty (windows) or screen (Linux) or equivalent terminal login software. The settings for the console port need to be **115200 baud rate with 8-N-1**. Also, make sure you are using the correct COM port number (For instance, you can find the correct COM port using Device Manager on Windows).

If you use Ethernet, you will have to configure the PC or Laptop's IP to match FC1011's default network. The default network configuration is DHCP so you will see the IP address of the unit on OLED display if Ethernet is connected to a network.

Default Account settings

Login: **admin**
Password: **admin**

Login: **osctl**
Password: **osctl**

The default root password is "osctl," but be careful while using root; we recommend using "sudo" instead to gain root access. Also, make sure to change the root password after the initial setup.

Configuring Static IP

Use "osctl" command to configure a static IP address after login as "root" user. "osctl -?" shows detailed osctl command options with examples.

```
# osctl -i 192.168.1.10 -m 255.255.255.0 -g 192.168.1.1
```

The above command configures the IP address of FC1011 as 192.168.1.10 with mask 255.255.255.0 and gateway & DNS as 192.168.1. Once the IP is configured from a PC or a Laptop using Ethernet or serial port, you can then connect FC1011 to your network and access it using "telnet" or "ssh."

Configuring Dynamic IP

A DHCP server should be running on your network in order to configure dynamic IP. The DHCP client on the device then gets the IP from the server.

```
# osctl -D
```

Make sure you know the assigned IP address to log in using "telnet" or "ssh."

Configuring Hostname

To change the hostname of FC1011 switch use '-h' option.

```
# osctl -h FC1011-SW-1
```

The above command changes the hostname of FC1011 to FC1011-SW-1.

Configuring DNS/Nameserver

To configure a DNS or Nameserver, use "-n" option of osctl.

```
# osctl -n 192.168.1.11
```

Configuring Syslog Server

To send system-generated events to an external Syslog server, use the "-S" option. In addition, you have to specify the address of the Syslog server which will receive these event logs. (Note: this option is only available in software version 2.0 and above).

```
# osctl -S 192.168.1.200
```

Version Info

The following command shows hardware and software versions and the serial number of the switch.

```
# osctl -V
```

Other Administrative commands

Most of the other administrative functions can be achieved using standard Debian Linux commands. For example, to change the password, use the "passwd" command from the Linux prompt, and to add a new user use the "useradd" command. You have to be a "root" user to add a new user.

Operation

To simplify the operation, all functions are provided in a single command called "osctl." The osctl command provides three primary functions

1. Network & Host configuration (allowed only in "root")
2. Switch Control & Status
3. Port group Management

Network & Host configuration

The following are the commands used to configure network.

1. To configure static IP
osctl -i <ip> -m <mask> -g <gw>
2. To use DHCP (dynamic IP)
osctl -D
3. To configure hostname alone
osctl -h <hostname>
4. To configure a DNS or Nameserver, use "-n" option
osctl -n <dns-server-ip>
5. To configure syslog server to receive FC1011's events
osctl -S <syslog-server-ip>

For details, refer to the installation section as these are explained in that section.

In order to switch a particular port or ports or a group ON or OFF, you can use following command:

```
$ osctl -p {<port#/s> | <port-range> | <port-group>} <on|off> [-t <secs>]
```

Note that the "{}" (braces) groups options, and "|" is equivalent to "or." If the options are in square brackets "[]" then it is optional. Wherever you see port# or in_port#, they are all the same and represent one of 8 ports of FC1011.

For example:

```
$ osctl -p 4 on
```

→ Switch port 4 to ON. LED on port 4 of FC1011 lids when this command is issued confirms that the port is switched ON.

```
$ osctl -p "1 2 3" on
```

→ switch ports 1, 2, and 3 to ON. Note the double quote; without that, it won't work.

```
$ osctl -p "1-4" off
```

→ switch ports 1, 2, 3, and 4 to OFF (range 1 to 4). This comes in handy when you want to switch a bunch of ports together.

```
$ osctl -p gpl off
```

→ switch all ports in group "gpl" to OFF. You will have to create group before using it with "-c" option as explained in "Group Management" section.

```
$ osctl -p 2 off -t 120
```

→ switch port 2 to OFF after 120 **secs** (delayed switching)

```
$ osctl -p 2 off -T 120
```

→ switch port 2 to OFF gradually within 120 **milliseconds**. This option is only available with VFC series switch. It is different from the above delay, where the switching happens almost immediately *after* 120 secs of a wait, but switching takes 120 milliseconds to come to a completely OFF state to simulate real-world fiber cuts. Default is ten *milliseconds*.

Switch Status

The following command shows the status of a port or ports in a pre-defined group, whether it is on/off.

```
$ osctl -s [<port_group> | <in_port#>]
$ osctl -A [<port_group> | <in_port#>]
$ osctl -B [<port_group> | <in_port#>]
```

For example:

```
$ osctl -s
```

→ shows all ports' & groups' status as follows. It shows first all ports and tells you which ports are part of a group.

```

      All Ports Status
=====
Port   | Port Status
=====
  1    |    off
  2    |    off
  3    |    off
  4    |    off
  5    |     on
  6    |     on
  7    |     on
  8    |     on
-----
      Group Status
=====
Group Name | Port Status
=====
    gp1    |    off
-----
-----
      Group's Port Details
=====
GroupName: <gp1>
Port Status: off
Ports in the group: 1 2 3 4
-----
```


\$ osctl -s gpI

→ shows group "gpI" status only. You will have to create group before using it with "-c" option as explained in "Group Management" section.

\$ osctl -s 3

→ shows port 3 status only

\$ osctl -A 2

→ displays port 2 attenuation in 1-256 range

\$ osctl -B 4

→ displays port 4 attenuation in dB

Port group Management

Port group management commands provide the convenience of switching many ports together identified by a name. The group name can be any alphanumeric name but cannot be a number alone or start with a number. For example, the group name cannot be "10" or 10gp1.

1. To create a new group
\$ osctl -c <port-group> {<in-port#/s> | <inport-range>}
2. To update existing group
\$ osctl -u <port-group> {-a | -r <in-port#/s>}
3. To delete a group
\$ osctl -d <port-group>
4. To delete all groups
\$ osctl -R
5. To list ports in a group
\$ osctl -l [<port-group> | <in-port#>]

For example:

```
$ osctl -c gp1 "1 3 4"
```

→ Creates group named "gp1" with ports 1,3 and 4. The double quotes around space-separated port numbers are required. Note that when you make a group, all the ports in the gp1 will be in the OFF state by default.

```
$ osctl -c gp2 "5-8"
```

→ creates a group named "gp2" with ports 5,6,7 and 8 (5 to 8 range)

```
$ osctl -u gp1 -a "5 6"
```

→ adds ports 5 & 6 to existing group gp1

```
$ osctl -u gp1 -a 7
```

→ adds port 7 to existing group gp1

```
$ osctl -u gp1 -r "2 4"
```

→ removes ports 2 & 4 from group gp1

```
$ osctl -d gp1
```

→ deletes group gp1 and release all ports which were part of the group.

\$ osctl -l

→ lists ports in group, like

GroupName: <gp l>

Ports in the group: 1 2 3 4

\$ osctl -R

→ delete all groups in the database.

Setting Attenuation (VFC series only)

-A option

The "-A" option is used to set the attenuation of any particular output port or group to a fraction of input power. It takes a number from 1 to 256 as a divisor. Then, it attenuates the power output of the port by that number. The more the divisor, the more the attenuation is (256 is fully off & 1 is on).

The attenuation setting is nonlinear. So, if you set the attenuation to 128, it doesn't mean you are setting the attenuation to half of the source power. Please refer to the graph at the end of the manual, which shows the relationship between applied voltage and attenuation. The applied voltage is linear, but the VOA response is nonlinear. Currently, the attenuation setting is not saved; once you reset or reboot the switch attenuation setting is lost; you will have to configure it again.

```
$ osctl -A {<port_group> | <in_port#>} [ <divisor [1-256]> | [<divisor range [1-256]> -T <msec>]]
```

For example:

```
$ osctl -A I 128
```

-> set attenuation of port#I to 128 (256 is fully off & 1 is on)

```
$ osctl -A I
```

-> displays current attenuation of port#I

```
$ osctl -A gpI 64
```

-> set the attenuation of all ports in the group gpI to 64

```
$ osctl -A 3-8 32
```

-> set the attenuation of ports 3,4,5,6,7 & 8 to 32

```
$ osctl -A I 100-160 -T 5000 (requires sw. ver 3.0 and above)
```

-> set the attenuation of port#I to 100 first and ramp-up to 160 in 5000 milliseconds period. This option helps create **packet errors** as opposed to the clean fiber-cut scenario.

```
$ osctl -A I 160-100 -T 5000 (requires sw. ver 3.0 and above)
```

-> set the attenuation of port#l to 160 first and ramp down to 100 in 5000 milliseconds period.

The osctl version 5.3 and above includes a default calibration setting, which allows you to set attenuation in **dB**, but it may not be accurate. Therefore, you need to calibrate all VOAs to get better accuracy. Please refer to the calibration section in this document for the details.

-B option

The -B option is used to set the attenuation in dB.

```
$ osctl -B {<port-group> | <in-port#>} [ <dB [1-25]>]
```

Examples:

```
$ osctl -B l 5
```

→ set attenuation of port#l to 5 dB

```
$ osctl -B l
```

→ displays current attenuation of port#l

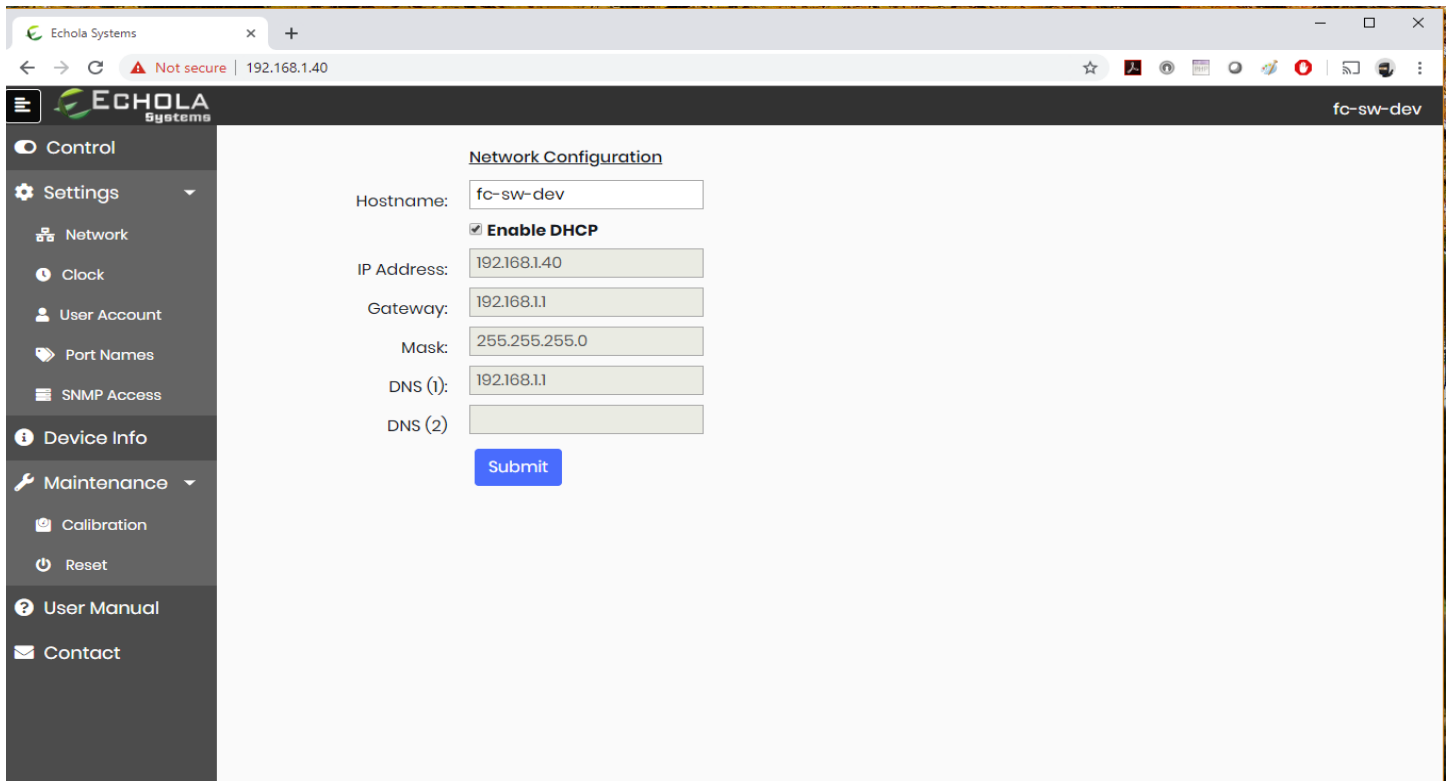
Web interface

Control page

The default page for the Web Application is the Control page. You can switch on/off each port using the dropdown menu or the radio button.

Setup page

The setup page allows you to change Hostname, Static IP address configuration and Dynamic IP (DHCP). Initially, it shows the current set values. Then, you can change them by clicking on the corresponding field. Note that changing the IP, Gateway, Mask, or DHCP setting will make the switch **reboot!**



Device Info page

The device Info page shows hardware and software information such as model and version.

Automating Echola's Optical Switches

Using RESTful API

Our REST APIs accept JSON request bodies, returns JSON-encoded responses, and use standard HTTP response codes, authentication (basic), and verbs (GET and PUT only). The following table summarizes the available APIs.

Method	URI	Action	Request (body) data
GET	https://[host_ip]/api/ports	Retrieve all ports state	
GET	https://[host_ip]/api/ports/[port_#]	Retrieve specific port's state	
PUT	https://[host_ip]/api/ports/[port_#]	Switch On Off a specific port	{"state":<on off>}
GET	https://[host_ip]/api/voa	Retrieve all ports attenuation	
GET	https://[host_ip]/api/voa/[port_#]	Retrieve specific port's attenuation	
PUT	https://[host_ip]/api/voa/[port_#]	Set attenuation of a specific port	{"attenuation":<data>}
GET	https://[host_ip]/api/voadb	Retrieve all ports attenuation in dB	
GET	https://[host_ip]/api/voadb/[port_#]	Retrieve specific port attenuation in dB	
PUT	https://[host_ip]/api/voadb/[port_#]	Set attenuation of a specific port in dB	{"attenuation":<data>}

Examples:

1. The following example shows how the attenuation of port 1 can be set using a 'curl' command-line script. **Please note, it is not a good practice to use the password on the command line directly, as shown below. Alternatively, you could use curl's netrc option "-n <user_password_file>" instead of "-u" option and protect the password file with the appropriate permissions.**

```
curl -X PUT \  
-u admin:admin \  
-H "Content-Type: application/json" \  
-d '{"attenuation": "159"}' \  
-k https://192.168.1.120/api/voa/1
```

2. Following command returns the state of port 2 (whether it is ON or OFF)

```
curl -u admin:admin \  
-H "Content-Type: application/json" \  
-k https://192.168.1.120/api/ports/2
```

responds with

```
{"state":"on"}
```

3. Following command returns the attenuation of port 1

```
curl -u admin:admin \  
  -H "Content-Type: application/json" \  
  -k https://192.168.1.120/api/voa/1
```

responds with

```
{"attenuation":"159"}
```

4. Following command switches port 2 to OFF

```
curl -u admin:admin -X PUT \  
  -H "Content-Type: application/json" \  
  -d '{"state": "off"}' \  
  -k https://192.168.1.120/api/ports/2
```

5. Following command sets, the attenuation of port 1 with an endpoint which accepts input in dB

```
curl -X PUT \  
  -u admin:admin \  
  -H "Content-Type: application/json" \  
  -d '{"attenuation": "10"}' \  
  -k https://192.168.1.120/api/voadb/1
```

Using osctl CLI

FCI011 configuration or monitoring can be automated using osctl command available on the switch using external terminal automation tools such as Tcl/Expect. Unlike RESTful API, which supports limited configurations, the osctl command supports all configurations, including network.

Running scripts from Unix/Linux systems

If you want to run the script from a Unix/Linux-based machine, then there is a possibility that you may have these tools on your system. First, check if it's already been installed by typing "expect" from the Unix/Linux prompt. If it is not installed, you will have to install it using the package install tool for that particular flavor of Unix/Linux. For instance, on Fedora core Linux, you can use "yum install tcl expect" to install Tcl and Expect.

Running scripts from Windows

For Windows-based systems, you can install the Windows free community version of ActiveTcl from Activestate, then make sure to install "expect" using the command "teacup install Expect."

Also, you need to enable the "telnet" client on Windows before running any scripts. To enable telnet on Windows, follow these steps

1. Start
2. Control Panel
3. Programs and Features
4. Turn Windows features on or off
5. Check Telnet Client Hit OK

After that, you can start Telnet via Command Prompt to check if it works.

The following sample script logs into FCI011 switch, issues a switch command, checks whether the switch command was successful, and returns the result before terminating the telnet session. This script takes an argument (port number and state of the port (on/off)) from the command line argument. Cut and Paste the following script onto any editor and save as "rosctl". Then you can run the script by issuing "rosctl -p <port#> on|off". For instance, to switch port 2 to ON, you can call the script rosctl -p 2 on". Make sure you have a proper path set for Expect on the first line "#!/usr/bin/expect" for Unix/Linux-based systems. For Windows, you will have to uncomment 'exec' and 'package' commands, as the script mentions. All comments inside '#' provide more info on what the script is doing.

Sample Tcl/Expect script Using 'osctl' CLI

```
#!/usr/bin/expect
#####
#####
# This script switches the given port and verifies if the port is switched from a remote
machine
#      Command Usage: rosctl -p <port#> on|off
#####
#####
# For windows uncomment following
#  exec tclsh "$0" ${1+"$@"}
#  package require Expect

# Check number of arguments passed to this command if < 3 then spit out error & exit

    if { $argc < 3 } {
        puts "Usage: rosctl -p <port#> on|off\n"
        exit 1
    }

# Set telnet host, username, password and other parameters, modify these to reflect your
setup

    set hostname "192.168.2.20"
    set username "osctl"
    set password "osctl"
    set prompt "osctl@.*\$"
    set port [lindex $argv 1]
    set status [lindex $argv 2]
    set commandcontrol "osctl -p $port $status"
    set commandstatus "osctl -s $port"

# Display info.

    puts "Connecting to $hostname."

# Connect to the telnet server using the "spawn" command.

    spawn telnet $hostname
    #spawn C:\Putty\putty.exe -telnet $hostname
```

Wait for a login prompt.

```
expect -re "(Name|login|Login|Username).*:.*" {
```

```
    # Login prompt received. Send user name to FC1011/FC1011.
```

```
    send "$username\r"
```

```
} eof {
```

```
    # No login prompt received. Display an error.
```

```
    puts "could not connect\n"
```

```
}
```

Wait for a password prompt from the Unix server.

```
expect "Password:" {
```

```
    # Password prompt received. Send the password.
```

```
    send "$password\r"
```

```
}
```

Wait for the switch prompt.

```
expect -re $prompt {
```

```
    # Issue osctl command to switch given port
```

```
    send "$commandcontrol\r"
```

```
}
```

Wait for the switch prompt again to check status.

```
expect -re $prompt {
```

```
    # Issue osctl command to check status
```

```
    send "$commandstatus\r"
```

```
}
```

```
# Discard echoed command - we need only the status
    expect "$commandstatus\r"

# Discard unwanted prompt as well

    expect -re "(.*)$prompt"

#Debug
#puts "\nGOT*****$expect_out(buffer)*****\n"
#puts "\n GOTS #####$expect_out(1,string)#####\n"

# Save remaining to buffer 'data'

    set data $expect_out(1,string)

# Check return status and display result accordingly
    switch -re $data {
        "off" { puts "Port $port is OFF" }
        "on" { puts "Port $port is ON" }
        default { puts "Port $port status is unknown" }
    }

# Terminate telnet

send "exit\r"
```

Hardware Specifications

Electrical	
Input Power	100-240 AC
Total Power Consumption	< 7 Watts
Serial Port	1x micro-USB
Networking	1x 10/100 Ethernet
Optical Specification for FC Series	
Connectivity	10 duplex LC connectors
Data rate	Any data rate - Physical Layer Switch, no limitation
Wavelength (nm)	Multi-mode version: 850/1310 nm Single-mode version: 1280-1625 nm
Optical Technology	VOA
Insertion Loss (dB)	0.5 Typical, 0.8 Max. (without connectors)
Switching Time (ms)	≤ 10
Crosstalk (dB)	≤ -80
Repeatability (dB)	≤ 0.1
Optical Specification for VFC Series (SM)	
Connectivity	10 duplex LC connectors
Data rate	Any data rate - Physical Layer Switch, no limitation
Display	OLED (1 inch)
Wavelength (nm)	1310±50nm & 1550±50nm
Optical Technology	MEMS VOA
Insertion Loss (dB)	< 0.8 dB (without connectors)
Switching Time (ms)	≤ 5
Crosstalk (dB)	≥ 65
Attenuation Dynamic Range (dB)	> 30
Attenuation Resolution (steps)	256 steps
Optical Power Handling (mW)	≤ 250mW/channel
Environmental	
Operating Temperature (°C)	-5 ~ +75
Storage Temperature (°C)	-40 ~ +85
Relative Humidity Range (%)	0 ~ 85

General Software Info

- This device runs the Debian distribution of the Linux OS. Most of the Debian tools can be used as-is for debugging purposes.
- All the group information is stored in a hidden XML database file. If this file is corrupt, the system will recover from this error by trying to copy the backup database file. If this happens, it will throw a warning message, but it is not guaranteed that all the group information will be restored correctly. In that case, you may need to recreate missing groups.
- For adding additional users, you can use Linux's "useradd" command. You have to be "root" to add a new user to the system.

→ Contact Info

If you have any technical questions and need help, you can send an email to support@echola.com. You can also download the latest documents and software from our website www.echola.com.