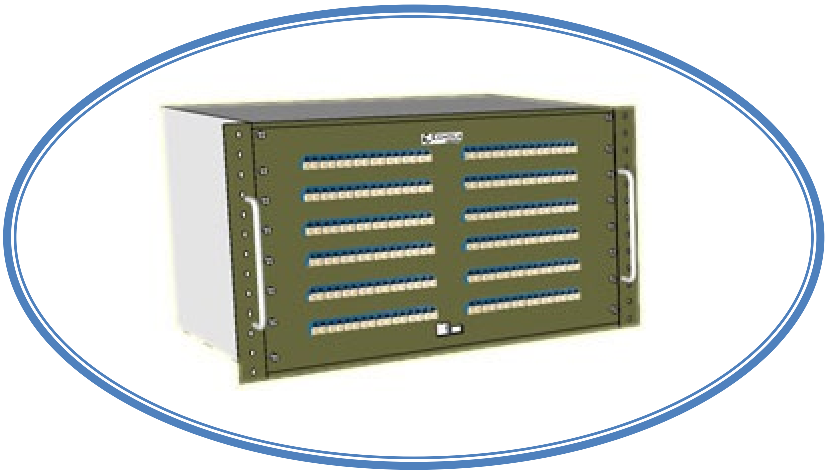
Installation & Operating Guide

OMS Series Matrix Switch

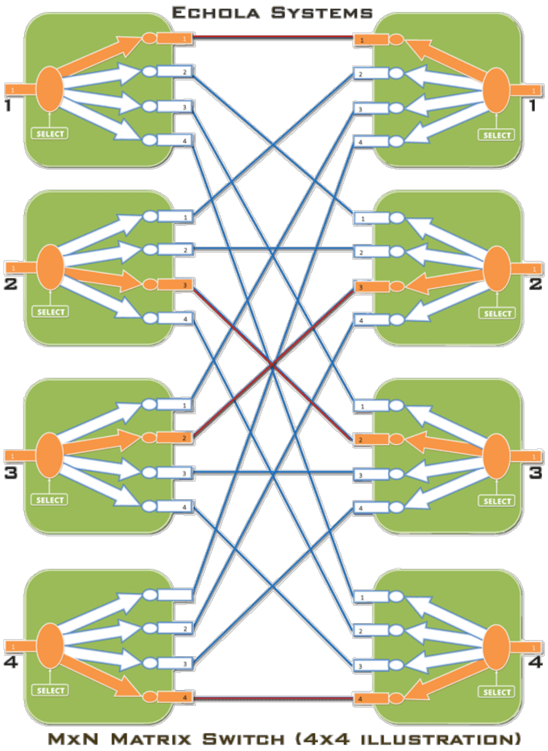


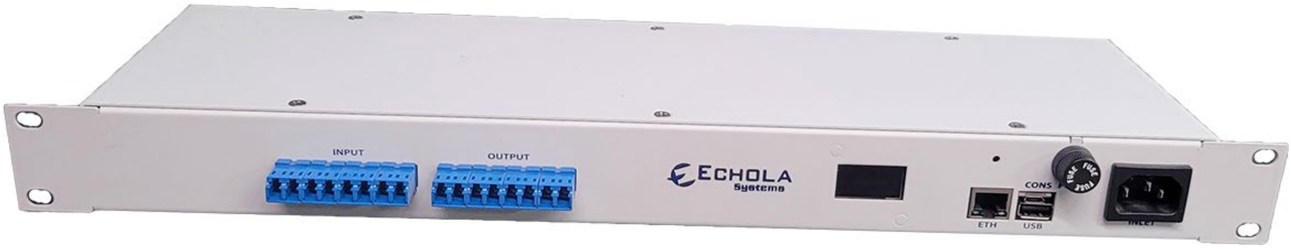
ECHOLA Systems

Installation

The OMS series switch is a non-blocking MEMS based MxN all-optical cross-connect switch that enables light to be switched seamlessly from any input port to any output port.

The figure below illustrates the operation of a 4x4 matrix switch. This matrix comprises eight 1x4 switches connected in the configuration shown. It demonstrates a simple setup where the switch is programmed to connect input port 1 to output port 1, input port 2 to output port 3, input port 3 to output port 2, and input port 4 to output port 4. However, the switch can be programmed to connect any input port to any output port using the GUI or CLI commands. The switching time is less than 20ms. Typical applications include patch panel automation, Optical Signal Routing and Network Protection.





OMS8 8x8 Non-blocking Matrix Switch

The above is an 8x8 matrix switch housed in a 1RU unit, featuring 8 simplex input ports and 8 simplex output ports. Any input port can be connected to any output port using the GUI or CLI or using REST APIs. Its low insertion loss, broadband capability, compact design, and low power consumption make it ideal not only for lab automation but also for managing large optical networks.

Configuring OMS series switch

OMS series switch provides a serial UART port (micro-USB) and an Ethernet (RJ45) port connectivity for management. Serial port is only used in special situations such as to debug network connectivity if OMS series switch 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).

Default Account settings

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

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

The default root password is "osctl." However, we recommend using "sudo" to gain root access instead of directly using the root account. Additionally, ensure you change the root password immediately after the initial setup.

Configuring Static IP

> Before configuring IP please make sure the date and time settings are correct. You can use Linux CLI *timedatectl* or our GUI/Webpage 'clock' page to set the time. If you use the GUI/Webpage you will have to uncheck NTP and set the date and time.

The management module runs an embedded version of Linux, so anyone familiar with Linux will find it easy to use. But we provide a CLI command called 'osctl' which can be used to configure and use most of the features. So, to configure a static IP address you can use following syntax.

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

'`osctl -?`' shows detailed osctl command options with examples.

The above command configures IP address of OMS series switch as 192.168.1.10 with network mask 255.255.255.0 and gateway & DNS as 192.168.1.1. Once the IP is configured from a PC or a Laptop using Ethernet or serial port, you can then connect OMS series switch to your network and access it using "telnet" or "ssh".

Configuring Dynamic IP

If you have a DHCP server running on your network and you may want to dynamically assign an available IP address to OMS series switch by using following option.

```
# osctl -D
```

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

Configuring Hostname

You can also change the hostname of OMS series switch switch using "osctl".

```
 # osctl –h OMS8-SW-1
```

The above command changes hostname of OMS series switch to OMS series switch-SW-1.

Configuring DNS/Nameserver

To configure a DNS or Nameserver, you can 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 "–S" option of osctl. You have to specify the address of the syslog server which will receive these event logs. (Note: this option 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 serial number of the switch.

```
# osctl –V
```

Other Administrative commands

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

Operation

To simply the operation, all functions are provided in a single command called “osctl”. osctl provides 3 major functions

1. Network & Host configuration (allowed only as “root” user)
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, you can use “-n” option of osctl.
   ```
   # osctl -n <dns_server_ip>
   ```
5. To configure syslog server to receive OMS series switch’s events
   ```
   # osctl -S <syslog_server_ip>
   ```

For details on above commands refer to above installation section.

Switch Control

In order to cross connect an input port to an output port or a group of ports, the following command can be used.

```
$ osctl -p <input_port#> <output_port#> [-t <secs>]
```

Note that the “{}” (braces) groups options and “|” is equivalent to “or”. If the options are in square brackets “[]” then it is optional.

For example:

```
$ osctl -p 1 4
```

- switch input port 1 to output port 4. Note, there should be space between p,1 and 4.

```
$ osctl -p 2 1 -t 120
```

- switch input port 2 to output port 1 after 120 secs (delayed switching)

Switch Status

Following command shows the status of ports, which input port is mapped to which output port or status of ports in a pre-defined group.

```
$ osctl -s [<input_port#>]
```

For example:

```
$ osctl -s
```

- shows all ports status as follows. It shows which input port connected to which output and also shows the active group.

```
        All Ports Status
=========================
    In Port  |  Out Port
=========================
    1        |    4
    2        |    1
    3        |    5
    4        |    12
    5        |    14
    6        |    2
    7        |    6
    8        |    7
    9        |    9
    10       |    11
    11       |    10
    12       |    8
    13       |    13
    14       |    3
    15       |    16
    16       |    15
----------------------------------
   Active Group: config1
----------------------------------
```

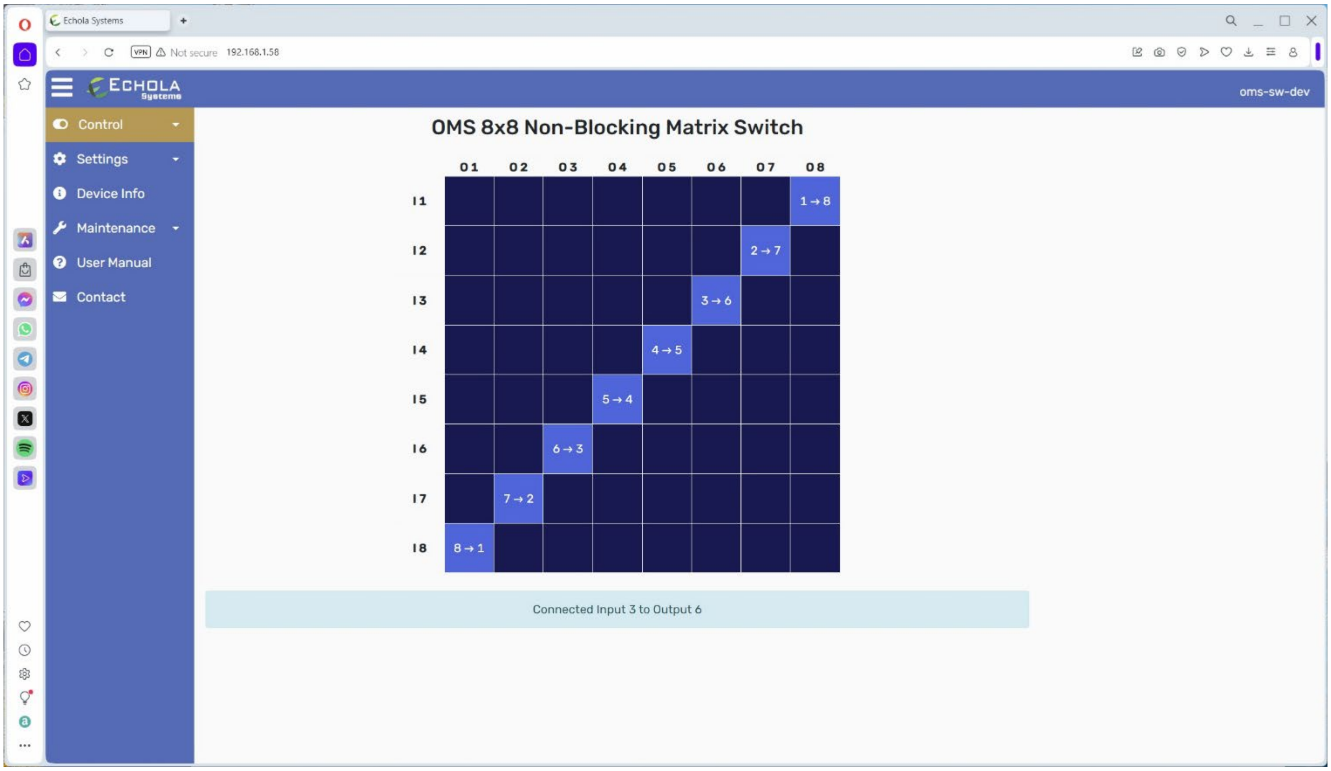
```
$ osctl -s 2
```

- shows which output port the port2 is cross connected to.

```
$ 1
```

Web interface

The OMS series Non-Blocking Matrix Switch front-end GUI provides an intuitive and interactive interface for managing input-output connections seamlessly. The main display consists of a dynamic matrix table, where rows represent input ports, and columns represent output ports. To establish a connection, simply click on the cell corresponding to the desired input-output pair. The system ensures non-blocking operation by automatically swapping or reassigning affected connections to maintain the integrity of the matrix configuration. Connection states are visually represented with distinct color codes for clarity. Any changes made are immediately reflected in the table and synchronized with the backend system. Additionally, the GUI offers real-time feedback messages centered below the matrix to confirm actions or notify users of potential issues, ensuring a user-friendly and efficient operation.



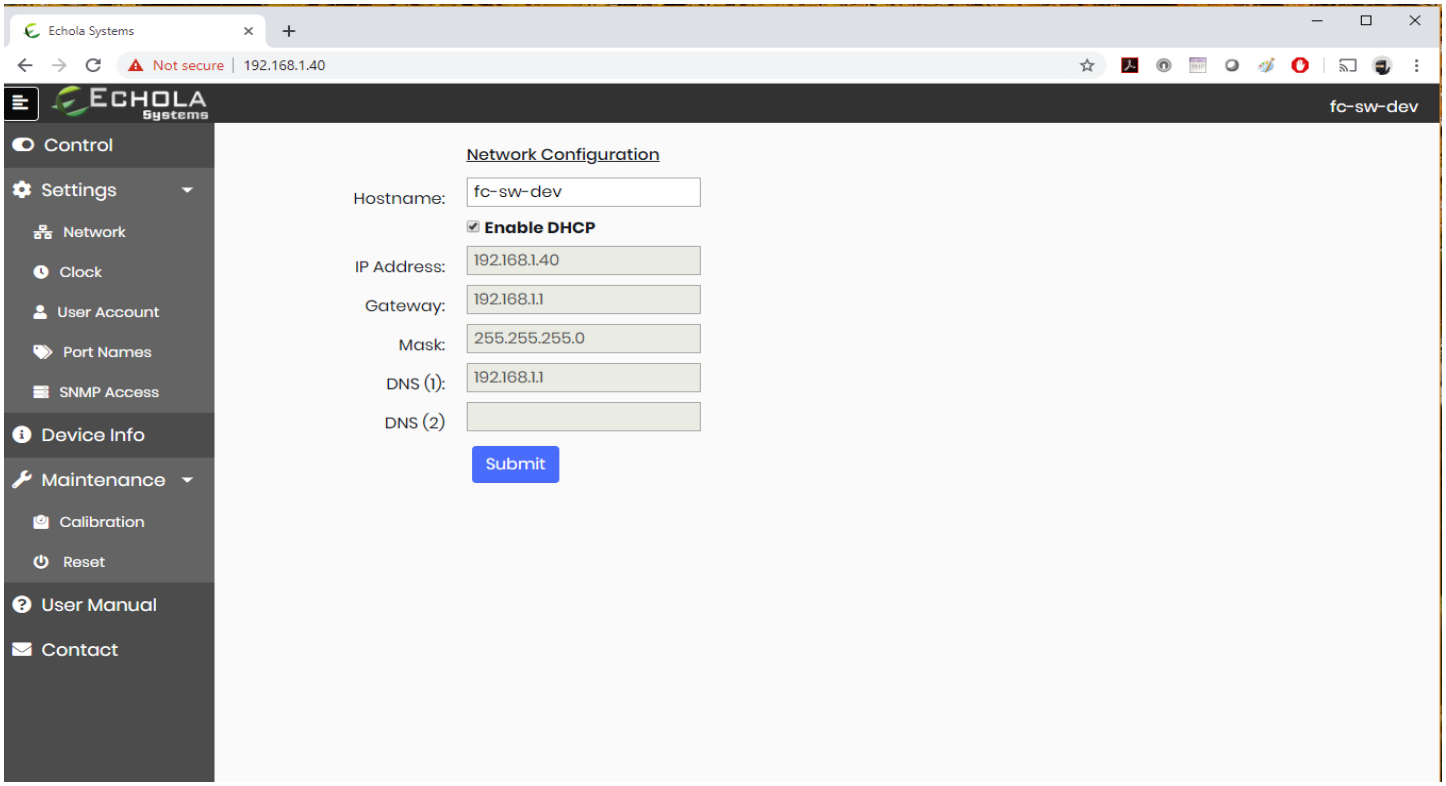
OMS 8x8 Non-Blocking Matrix Switch

	O1	O2	O3	O4	O5	O6	O7	O8
I1								1→8
I2							2→7	
I3						3→6		
I4					4→5			
I5				5→4				
I6			6→3					
I7		7→2						
I8	8→1							

Connected Input 3 to Output 6

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!



Network Configuration

Hostname:	fc-sw-dev
	☑ Enable DHCP
IP Address:	192.168.1.40
Gateway:	192.168.1.1
Mask:	255.255.255.0
DNS (1):	192.168.1.1
DNS (2)	

Device Info page

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

Maintenance page

This page can be used to "Reset/Reboot" the system or calibrate the VOA. The Calibration of the VOA requires a laser source and an optical meter. Please get in touch with support@echola.com for calibration details.

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 or output port	
PUT	https://[host_ip]/api/ports/[port_#]	Switch On|Off a specific port (FC/VFC Series)	{"state":<on | off>}
PUT	https://[host_ip]/api/ports/[port_#]	Connect an input to an output (OMS Series)	{"output_port":<port#>}
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 connection between input port 1 and output port 3 is made using a 'curl' command.

```
$ curl -X PUT \
    -u admin:admin \
    -H "Content-Type: application/json" \
    -d '{"output_port": "3"}' \
    -k https://192.168.1.120/api/ports/1
```

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.

2. Following command returns the state of port 2 (state here is ‘output_port#’)

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

responds with {"state":"5"}
```

Using the osctl CLI

The VFC1011 configuration and monitoring can be automated using the osctl command, which is available on the switch and can be used with external terminal automation tools such as Tcl/Expect. Unlike the RESTful API, which supports limited configurations, the osctl command provides access to all configurations, including network settings.

Running Scripts from Unix/Linux Systems

If you want to run scripts from a Unix/Linux-based machine, you may already have the necessary tools installed. To check, type expect at the Unix/Linux prompt. If it is not installed, use the package installation tool specific to your system to install it. For example, on Fedora-based systems, you can run:

```
$ yum install tcl expect
```

on Debain/Ubuntu

```
$ sudo apt update
$ sudo apt install tcl expect
```

This will install both Tcl and Expect.

Running Scripts from Windows

For Windows-based systems, you can install the free community version of ActiveTcl from ActiveState. After installation, ensure you install "Expect" using the following command:

```
$ teacup install Expect
```

Additionally, you need to enable the "Telnet" client on Windows before running any scripts. To enable Telnet, follow these steps:

1. Go to **Start**.
2. Open the **Control Panel**.
3. Navigate to **Programs and Features**.
4. Click on **Turn Windows features on or off**.
5. Check the box for **Telnet Client** and click **OK**.

After enabling Telnet, you can verify it by starting Telnet via Command Prompt.

Sample Tcl/Expect script Using 'osctl' CLI

The following sample script logs into the VFC1011 switch, issues a switch command, verifies if the command was successful, and then terminates the Telnet session. The script takes the port number and the desired state (on or off) as command-line arguments.

To use this script:

1. Copy and paste the script into any text editor.
2. Save it as rosctl.
3. Run the script using the following syntax:

```
$ rosctl -p <input_port#> <output_port#>
```

For instance, to connect input port 2 to output port 6:

```
$ rosctl -p 2 6
```

Important:

- Ensure the first line of the script specifies the correct path to Expect (#!/usr/bin/expect) for Unix/Linux systems.
- For Windows, uncomment the exec and package commands as instructed in the script.
- All comments beginning with # provide detailed explanations of the script's actions.

This approach ensures efficient and automated configuration of the VFC1011 switch.

```
#!/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 <input_port#> <output_port#>\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 input_port [lindex $argv 1]
      set output_port [lindex $argv 2]
      set commandcontrol "osctl -p $input_port $output_port"
      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 VFC1011/VFC1011.

          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)
      puts "Output Port is set to $data"


# Terminate telnet

send "exit\r"
```

OMS Series Switch Specifications

Parameters	Min	Typical	Max	Unit
Operation Wavelength	750	1270~1630	2400	nm
Insertion Loss (SM)		1.5	2.2	dB
Insertion Loss (MM)		1.8	2.8	dB
Crosstalk, On/Off Ratio	45		70	dB
Extinction Ratio (PM Fiber)	18		23	dB
Switch Speed		5	20	ms
Durability	10^9			cycle
Polarization Dependent Loss		0.04	0.2	dB
Wavelength Dependence Loss [3]		0.1	0.3	dB
Return Loss	50			dB
Repeatability		0.05	0.1	dB
Operating Temperature	-5		65	°C
Storage Temperature	-40		85	°C
Electrical Power Consumption			50	W
Enclosure Dimension	1RU/2RU/4RU			

Troubleshooting

- ✓ This device runs the Debian distribution of the Linux OS. Most Debian tools can be used as-is for debugging purposes.

- ✓ All group information is stored in a hidden XML database file. If this file becomes corrupt, the system will attempt to recover by copying the backup database file. If this occurs, a warning message will be displayed. However, it is not guaranteed that all group information will be restored correctly. In such cases, you may need to recreate any missing groups manually after restoration. You can also manually initiate the recovery process by using the command osctl -R.

- ✓ Some customers have encountered issues with running out of space due to error messages accumulating over time in the /var/log directory. You can recover this space by using the command osctl -R and entering y when prompted.

- ✓ To add additional users, you can use Linux's useradd command. You must use sudo command to add a new user to the system.

Contact Info

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