

# Quick Installation & Operating Guide

OS214-MM/SM - 5 PORT (1x4 DUPLEX ) OPTICAL switch



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

OS214 is a Linux based 5 port (1x4 duplex) Layer 1 optical switch used in patch panel automation applications to share one expensive device with four other optical devices. It provides a serial (RS232) and an Ethernet (10/100) port connectivity for management. Serial port is normally used in special situations such as to debug network connectivity if OS214 is not reachable through Ethernet.

You might require a Laptop or a PC to configure OS214 with an IP address before connecting to your network so that you would be able to access OS214 from remote. You could either use Serial or Ethernet port to configure IP. If you use serial port using supplied null modem cable then you would need to configure serial for **38400 baud rate with 8-N-1** to access OS214. If your PC doesn't have a serial Port then you need to use a serial to USB converter, which is not supplied with the product, to connect to PC's USB port.

If you use Ethernet then you would need to configure the PC or Laptop's IP to match OS214's default network. The default network configuration is as follows

**IP Address: 10.1.1.100**

**Mask: 255.255.255.0**

**Gateway: 10.1.1.1**

**DNS: 10.1.1.1**

If you have successfully configured Serial or Ethernet then you would see the os214 login prompt; login as "root" to setup network.

root's password: *osctl*

There is one non-root default user available on OS214 which can be used once the switch is setup (for doing switching of port).

username: *osctl*

password: *osctl*

## 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 IP address of OS214 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 OS214 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 want to dynamically assign an available IP address to OS214 then use following option.

```
# osctl -D
```

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

## Configuring Hostname

You can also change the hostname of OS214 switch using "osctl".

```
# osctl -h OS214-SW-1
```

The above command changes hostname of OS214 to OS214-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.

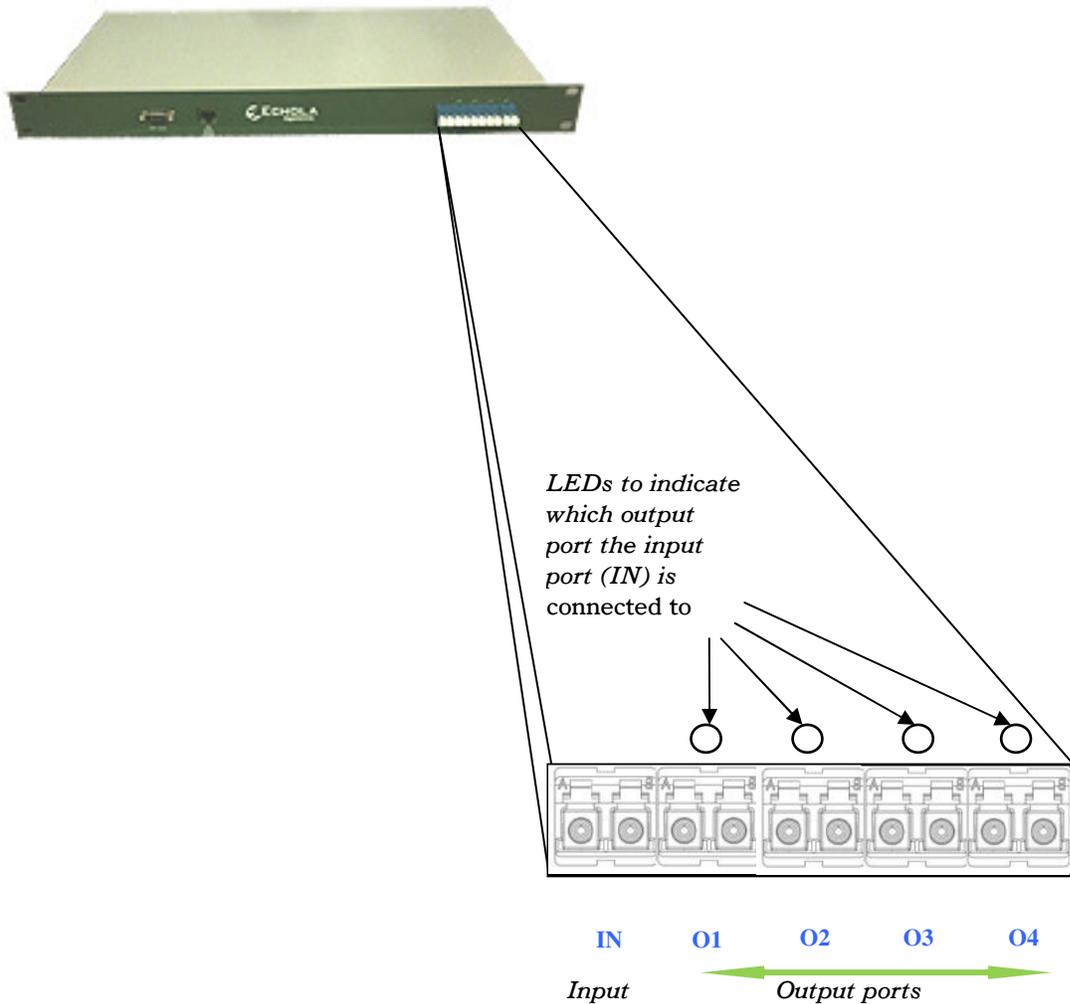
```
# osctl -v
```

### **Other Administrative commands**

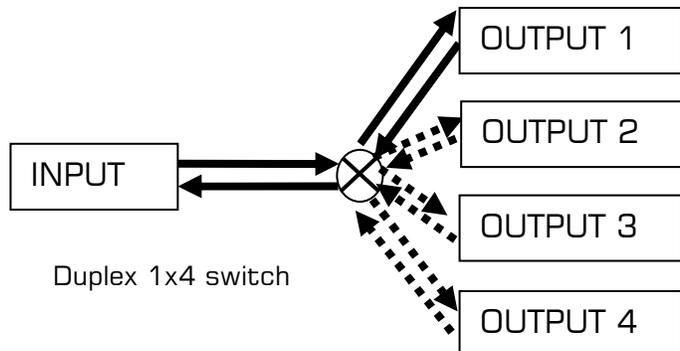
Most of other administrative functions can be done using standard Linux commands. For example, to change the password use "*passwd*" command from the Linux prompt and to add new user use "*useradd*" command. You have to be "root" user in order to add a new user.

# Operation

OS214's front panel has 5 ports with LC duplex connectors numbered left to right IN and 1-4. The input port is labeled as "IN" and 4 output ports labeled as "O1", "O2", "O3" and "O4". *Even though we refer ports as input and output ports they are bi-directional (duplex) so you can use them as either input or output. This means switch can act as 1x4 or 4x1 switch.*



The input port's Tx and Rx are internally *connected* to corresponding output ports' Tx and Rx. This way you can share one device with 4 other devices and switch between.



The switching of light from input port to any of the four output ports can be achieved using a command line interface. To simplify the operation, all functions are provided in a single command called "osctl". Osctl provides three major functionalities.

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

Osctl will show correct syntax usage if you make mistakes in typing command options. It will also provide some examples on how to use the command when you make mistakes. Also you may use "osctl -?" which shows all syntaxes with all examples.

## **Network & Host configuration**

Network setup has 4 different syntaxes.

1. To configure static IP  
**# osctl -i <ip> -m <mask> -g <gw>**
2. To use DHCP  
**# osctl -D**
3. To configure hostname alone  
**# osctl -h <hostname>**
4. To configure DNS/NameServer  
**# osctl -n <dns ip addr>**
5. To configure Syslog server for event logging  
**# osctl -S <syslog server ip addr>**

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

## Switch Control

In order to switch input to output port 1 to 4 you have to use `osctl` in following way.

```
$ osctl -p <1|2|3|4> [-t <secs>]
```

Note that the "`|`" is equivalent to "or". If the options are in square brackets "`[]`" then it is optional.

*For example:*

```
$ osctl -p 4
```

➔ switch input port to output port 4. Now port 'IN' and port O4 are connected; the LED on output port 4 lights up when this command is issued which confirms that the input port is switched to out port 4.

```
$ osctl -p 1
```

➔ switch input port to output port 1. Now port 'IN' and port 'O1' are connected; the LED on output port 1 lights up when this command is issued which confirms that the input port is switched to out port 1.

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

➔ switch/connect input port 'IN' to output port 2 after 120 secs (delayed switching).

# Switch Status

Following command shows the status of the switch.

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

*For example:*

```
$ osctl -s
```

➔ shows port status as follows. Note that first column shows Port 1 which refers to input port. The second column is the one you should look at which shows what port is connected to the input port currently. In this case it is 4<sup>th</sup> output port.

```
          All Ports Status
=====
  Port   | Port Status
=====
    1     |    4
-----
```

```
$ osctl -s 2
```

➔ shows port 2 status only

# Automating OS214 Optical Switch

In order to automate OS214 switch you would need to write **tcl/expect** based script. The tcl and expect scripting languages are easy to learn. We have given an example script at the end which you can modify to suit your need. There are tons of online sources for learning tcl & expect.

The following provides quick high level overview of tcl and expect <http://cplug.org/t/uploads/2009/02/tcl-expect.pdf>. There is a good book from O'Reilly which provides great insight into expect language itself: "Exploring Expect: A Tcl-based Toolkit for Automating Interactive Programs (Nutshell Handbooks)".

## Running scripts from Unix/Linux systems

If you want to run the script from a Unix/Linux based machines then there is possibility that you may be already having these tools on your system. Check if it's already been installed by typing "expect" from Unix/Linux prompt. If it is not then you will have to install it using package install tool for that particular flavor of Unix/Linux. For instance, on Fedore 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 windows free community version of ActiveTcl from Activestate <http://www.activestate.com/activetcl/downloads>. The expect is not available yet for 64bit version of Windows 7/Vista. So you will need to download 32bit version for ActiveTcl first and then make sure to install "expect" using command "teacup install Expect".

Also you need to enable "telnet" client on Windows before running any scripts. In order to enable telnet on Windows follow these steps

- Start
- Control Panel
- Programs And Features
- Turn Windows features on or off
- Check Telnet Client
- Hit OK

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

The following sample script actually login into OS214 switch and issue a switch command then check whether the switch command was successful and return the result before terminating the telnet session.

This script takes argument (port number and state of the port (on/off)) from commands line argument. Cut and Paste the following script on to any editor and save as "rosctl". Then you can run the script from any machine which supports tcl/expect by issuing **rosctl -p <port#>**. For instance, to switch input port to output port 2, you can call script as **rosctl -p 2**.

Make sure you have proper path set for expect on first line "`#!/usr/bin/expect`" for Unix/Linux based systems. For windows you will have to uncomment 'exec' and 'package' commands as mentioned in the script. All comments inside '#' provide more info on what the script is doing.

## The Sample Tcl/Expect script

```
#!/usr/bin/expect
#####
# Script switches to given port and verifies if the port is switched from a remote machine
# Command Usage: rosctl -p <port#>
#####
# If you run this from a Windows machine, 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 < 2 } {
    puts "Usage: rosctl -p <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 port [lindex $argv 1]
set commandcontrol "osctl -p $port"
set commandstatus "osctl -s $port"

# Display info.

puts "Connecting to $hostname."

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

spawn telnet $hostname

# Wait for a login prompt.

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

    # Login prompt received. Send user name to OS214.

    send "$username\r"

} eof {

    # No login prompt received. Display an error.

    puts "could not connect\n"
}

# Wait for a password prompt from the OS214.
```

```

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 "\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 {
    "1" { puts "Input port is now connected to $port" }
    "2" { puts "Input port is now connected to $port" }
    "3" { puts "Input port is now connected to $port" }
    "4" { puts "Input port is now connected to $port" }
    default { puts "Input port is now connected to unknown port!" }
}

# Terminate telnet
send "exit\r"

```

## Troubleshooting

- ➔ If you use telnet and is very slow, it could be due to name server / DNS issue. Make sure you have name server configured correctly (-n option). Try "ping <name\_server\_ip>" to see if name server configured can be reached from OS214. If you don't have dns on your network then you can just remove file "/etc/resolv.conf" and see if it works normal.
- ➔ All the group information is stored in a hidden XML database file. If for some reason this file is corrupted, 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.

# Hardware Specifications

<b>Electrical</b>	
Input Power	100-240 AC
Total Power Consumption	< 7 Watts
Serial Port	1x RS232
Networking	1x 10/100 Ethernet
<b>Optical</b>	
Connectivity	5 Duplex LC ports
Data rate	No limitation - Physical Layer Switch
Wavelength (nm)	<u><b>OS414-SM</b></u> 1280-1340 nm 1520-1625 nm <u><b>OS414-MM</b></u> 850/1310 nm
Optical Technology	1x4 Electro-mechanical-optical switch
Insertion Loss (dB)	0.5 Typical, 0.8 Max. (without connectors)
Switching Time (ms)	≤ 10
Crosstalk (dB)	≤ -80
Repeatability (dB)	≤ 0.1
<b>Mechanical &amp; Environmental</b>	
Each Switch's Durability (Cycles)	> 10 <sup>7</sup>
Chassis (RU)	1 RU
Dimension (inch)	19 (W) x 9(D) x 1.75 (H)
Operating Temperature range (°C)	-20 ~ 75
Storage Temperature range (°C)	-40 ~ 70
Humidity	5 – 85 % RH

## Contact Info

If you have any technical questions and need help you can send email to [support@echola.com](mailto:support@echola.com) or call 408-321-9663. You can also download latest documents and software from our website [www.echola.com](http://www.echola.com).