

# *Mawlana Bhashani Science and Technology University*



---

## *Department of Information and Communication Technology*

---

**Course Code** : *ICT-4101*  
**Course Name** : *Telecommunication Engineering*  
**Name of the Assignment** : *Zodiac Open Flow Switch (Configure)*  
**Assignment No.** : *01*

**Submitted to:**  
**Mr. Nazrul Islam**  
**Assistant Professor,**  
**Dept. of ICT, MBSTU**

**Submitted by:**  
**Md. Nazmul Hasan**  
**ID: IT-17005**  
**Session: 2016-17**  
**4th Year 1st Semester**

**Date of Submission:** 20-October-2020

## Zodiac OpenFlow Switch (Configure)

### Objective:

- Configure and interact with Zodiac FX OpenFlow Switch.
- Exploring the Zodiac FX context.

**Q: 01.** What is Zodiac FX? Briefly explain about Zodiac FX switch.

**Answer:** Zodiac FX is the first OpenFlow switch designed to sit in a desk, not in a datacenter. Until now the power of Software Defined Networking (SDN) was only available to the administrators of large corporate networks. Even though there are numerous free or open source SDN controllers the one thing that was missing was a small, affordable OpenFlow switch. In this demo, we present Zodiac FX the world's smallest OpenFlow Software Defined Network Switch.

The Zodiac FX is a 4 port network development board designed for hobbyists, students, researchers, embedded developers or anyone who requires a low cost network development platform. Even though it was initially designed to allow affordable access to OpenFlow enabled hardware it's open source firmware it can be used in any number of other applications. By providing the firmware source code users are free to not only create their own versions but also use it as a basis for a completely different type of device. Some such applications may include: Router, Bridge, Load Balancer, Web server, VPN concentrator and many more. The main communication peripherals of Zodiac FX are sketched in Fig.

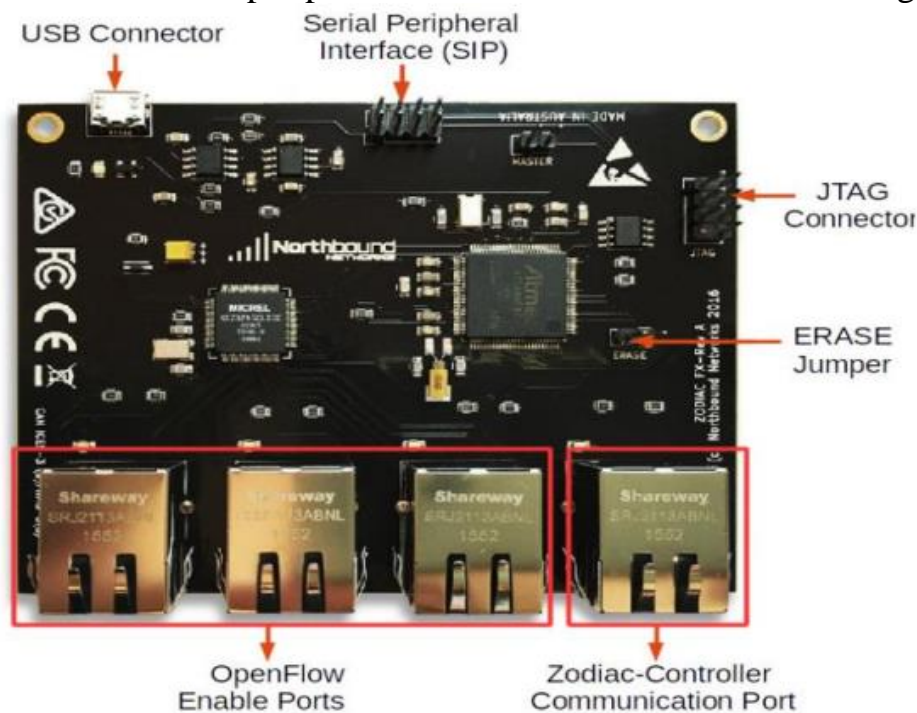


Fig. Communication peripherals of Zodiac FX.

**Q: 02.** How Command Line Interface (CLI) works through Zodiac FX switch?

**Answer:** The Command Line Interface provides the ability to configure settings and monitor the operation of the Zodiac FX. To simplify operations the CLI uses the concept of a “context”, limiting the available commands to the currently selected context. There are currently 3 available contexts:

- Base
- Config
- OpenFlow
- Debug

To enter the required context simply type the name of the context on the command line while at the base level. To return to the base level type “exit”. The current context is shown in brackets between the device name and the prompt. For example, the config context would show: Zodiac\_FX (config).



Zodiac FX Console

**Q: 03.** What are the throughput measurements from the Zodiac FX switch when using various Ryu applications?

**Answer:** The measurement tests compare the throughput of three controller applications:

Pass Through (PT): This is an application that instructs the switch to forward traffic from port 1 to port 2 and vice versa.

Simple Switch (SS): This is an application that implements a L2 learning switch using OpenFlow 1.3.

Simple Switch Re-imagined (SS2): This is an application created by inside-openflow.com to expand on SS by implementing a multi-table flow pipeline. Using multiple tables allows a controller application to implement more logic and state in the switch itself, reducing the load on the controller. This application uses four tables.

Faucet: This is a controller application that is designed for production networks. Faucet is deployed in various settings, including the Open Networking Foundation, as described here. The application employs a flow pipeline of seven tables.

In addition to the above applications, throughput measurements were also taken in a native mode. In this mode, the Zodiac FX switch works as a traditional L2 switch without a controller.

**Q: 04.** Give some examples of Zodiac FX CLI commands.

**Answer:**

```
Zodiac_FX# show ports
```

```
Port 1
```

```
Status: DOWN
```

```
VLAN type: OpenFlow
```

```
VLAN ID: 100
```

```
Port 2
```

```
Status: DOWN
```

```
VLAN type: OpenFlow
```

```
VLAN ID: 100
```

```
Port 3
```

```
Status: DOWN
```

```
VLAN type: OpenFlow
```

```
VLAN ID: 100
```

```
Port 4
```

```
Status: DOWN
```

```
VLAN type: Native
```

```
VLAN ID: 200
```

```
Zodiac_FX# show status
```

```
Device Status
```

```
Firmware Version: 0.57
```

```
CPU Temp: 37 C
```

```
Uptime: 00:00:01
```

```
Zodiac_FX# show version
```

```
Firmware version: 0.57
```

```
Zodiac_FX# config
```

```
Zodiac_FX(config)# show config
```

```
Configuration
```

```
Name: Zodiac_FX
```

```
MAC Address: 70:B3:D5:00:00:00
```

```
IP Address: 10.0.1.99
```

```
Netmask: 255.255.255.0
```

```
Gateway: 10.0.1.1
```

OpenFlow Controller: 10.0.1.8  
OpenFlow Port: 6633  
Openflow Status: Enabled  
Failstate: Secure  
Force OpenFlow version: Disabled  
Stacking Select: MASTER  
Stacking Status: Unavailable

Zodiac\_FX(config)# show vlans

VLAN ID	Name	Type
100	'Openflow'	OpenFlow
200	'Controller'	Native

Zodiac\_FX(config)# exit

Zodiac\_FX# openflow

Zodiac\_FX(openflow)# show status

OpenFlow Status Status: Disconnected

No tables: 1

No flows: 0

Table Lookups: 0

Table Matches: 0

Zodiac\_FX(openflow)# show flows

No Flows installed!

Zodiac\_FX(openflow)# exit

We want to use the controller address on our PC and connect eth0 on the PC to Port 4 of the switch (probably by plugging them both into the same local area network).

Zodiac\_FX# show ports

...

Port 4

Status: UP

VLAN type: Native

VLAN ID: 200

debian:~ \$ sudo ip addr add 10.0.1.8/24 label eth0:zodiacfx dev eth0

debian:~ \$ ip addr show label eth0:zodiacfx

inet 10.0.1.8/24 scope global eth0:zodiacfx

valid\_lft forever preferred\_lft forever

debian:~ \$ ping 10.0.1.99

```
PING 10.0.1.99 (10.0.1.99) 56(84) bytes of data.  
64 bytes from 10.0.1.99: icmp_seq=1 ttl=255 time=0.287 ms  
64 bytes from 10.0.1.99: icmp_seq=2 ttl=255 time=0.296 ms  
64 bytes from 10.0.1.99: icmp_seq=3 ttl=255 time=0.271 ms  
--- 10.0.1.99 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 1998ms rtt  
min/avg/max/mdev = 0.271/0.284/0.296/0.022 ms
```

Now to check the OpenFlow basics. We'll use the POX controller, which is a simple controller written in Python 2.7.

```
debian:~ $ git clone https://github.com/noxrepo/pox.git  
debian:~ $ cd pox  
debian:~ $ ./pox.py openflow.of_01 --address=10.0.1.8 --port=6633 --verbose  
POX 0.2.0 (carp) / Copyright 2011-2013 James McCauley, et al.  
DEBUG:core:POX 0.2.0 (carp) going up...  
DEBUG:core:Running on CPython (2.7.9/Mar 8 2015 00:52:26)  
DEBUG:core:Platform is Linux-4.1.19-v7+-armv7l-with-debian-8.0  
INFO:core:POX 0.2.0 (carp) is up.  
DEBUG:openflow.of_01:Listening on 10.0.1.8:6633  
INFO:openflow.of_01:[70-b3-d5-00-00-00 1] connected  
Zodiac_FX(openflow)# show status  
Status: Connected  
Version: 1.0 (0x01)  
No tables: 1  
No flows: 0  
Table Lookups: 0  
Table Matches: 0
```

**Conclusion:** I've completed this assignment from the help of Internet. The application of Zodiac FX open flow switch is applied by Glen Turner. I gathered this information from his blog. Finally I've learned about Zodiac FX OpenFlow switch and it's application.