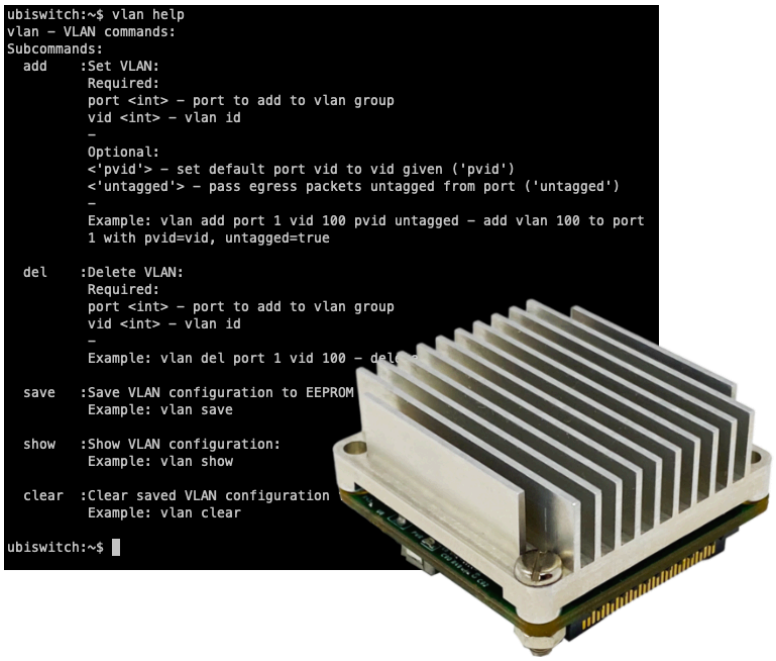
BotBlox



BloxOSLite Software Guide

Using the BloxOSLite CLI with BotBlox Hardware

Software Guide

Applies to:

MPN: BB-UBS-B-1 (Module)
MPN: BB-UBS-B-1-NDAA (Module)
MPN: BB-UD1-B-1 (Baseboard)
MPN: BB-UD1-B-1-NDAA (Baseboard)
MPN: BB-UDM-A-1 (Baseboard Mini)
MPN: BB-UDM-A-1-NDAA (Baseboard Mini)
MPN: BB-UPK-B-1 (Puck Mini)
MPN: BB-PMX-B-1 (Puck Maxi)

January 2025

BotBlox

Contents

Contents **2**
1 Introduction **3**
2 Features of BloxOSLite **3**
3 Hardware Setup **4**
3.1 Requirements 4
3.1 Physical UART Connection 4
3.2 Using the BotBlox baseboards 5
3.3 Step by step guide 8
4 Command Guide **9**
4.1 Port PHY status/control 9
4.1.1 Enable/disable link 10
4.1.2 Change PHY settings 10
4.1.3 Saving Port MAC/PHY configuration 11
4.1.4 Clearing Port MAC/PHY configuration 12
4.1.5 Show Port MAC/PHY configuration 12
4.1.6 Read Port Temperature 12
4.2 VLAN 13
4.2.1 Adding a VLAN member 14
4.2.2 Deleting a VLAN member 15
4.2.3 Saving VLAN configuration 15
4.2.4 Clearing VLAN configuration 16
4.2.6 Show VLAN configuration 16
4.3 Link Aggregation / LAG 17
4.3.1 Create an LAG group 18
4.3.2 Delete a LAG group 18
4.3.3 Add a port to a LAG group 19
4.3.4 Remove a port from a LAG group 20
4.3.5 Saving LAG configuration 20
4.3.6 Clearing LAG configuration 21
4.3.7 Clearing LAG configuration 21
4.4 PCS mode on SERDES ports (Ports 0, 9 and 10) 22
4.4.1 Change PCS Mode 22
4.5 Change EEE mode on PHY/MAC 23
4.5.1 Change EEE Mode 23
5 Datasheet Changelog **24**
6 Contact **24**

BotBlox

1 Introduction

UbiSwitch (BB-UBS-B-1) is a rugged, 8 port gigabit + 3 x 10G SFP ethernet switch module designed for harsh environments, verified by MIL-STD-810H testing.

UbiSwitch is a stackable module that places all ethernet, power and control signals onto stackable headers. This allows UbiSwitch to be used with any type of connector breakout.

BotBlox currently supplies two connector breakout systems, UbiSwitch BaseBoard (BB-UD1-B-1) and UbiSwitch BaseBoard Mini (BB-UDM-A-1).

UbiSwitch contains an onboard microcontroller which runs custom built software. This software is called BloxOSLite, and it allows configuration of switch over a 3.3V UART port using a command line interface (CLI).

This CLI is only accessible via the 3.3V UART, and is not accessible via any of the ethernet ports. Because of this UbiSwitch itself does not have an IP address.

Note that while this document exclusively references “UbiSwitch”, the software commands also apply to BB-PMX-A-1 (Puck Maxi) and BB-PUK-B-1 (Puck Mini). This is because, at their core, both Puck Maxi and Puck Mini contain an UbiSwitch Module.

2 Features of BloxOSLite

BloxOSLite is constantly under development with new updates being released to add features. The latest version can be found at the link below.

https://github.com/botblox/bloxoslite-releases/releases

The latest version as of the release of this document is v0.4.0, which was released on 30 January 2025. This version supports the features below.

Feature Name	Description
Port PHY status/control	Allows the following to be controlled on an individual port basis ● Enable/disable port ● Autonegation enable/disable ● Speed 10/100/1000/10000Mbps ● Duplex full/half ● Port speed advertise ● Read port temperature

VLAN	Allows ports to be assigned as Trunk or Access ports
Link Aggregation	Link Aggregation groups of ports
Device software update over serial	Allows the device software to be updated over the 3.3V UART serial port.

Please be sure to check the latest release on github, as the table above will become out of date when the next version of BloxOSLite is released. As of May 2024, all UbiSwitch modules are shipped with the latest version of the firmware preloaded.

3 Hardware Setup

3.1 Requirements

To access the CLI you will need the following hardware and software:

- UbiSwitch Module
- A compatible baseboard and cabling to access the UART port
- Any USB to UART converter (if using UART console)
- Any PC with serial software installed (eg PuTTY)

3.1 Physical UART Connection

The CLI is accessed using the 3.3V UART port on the UbiSwitch Module. This UART port is designed to be connected to another UART port on a management PC, or to a USB port on a management PC through a UART to USB converter.

The ground signal on UbiSwitch must also be connected to the connected UART device. The UART connection is a 3.3V TTL, single ended signal.

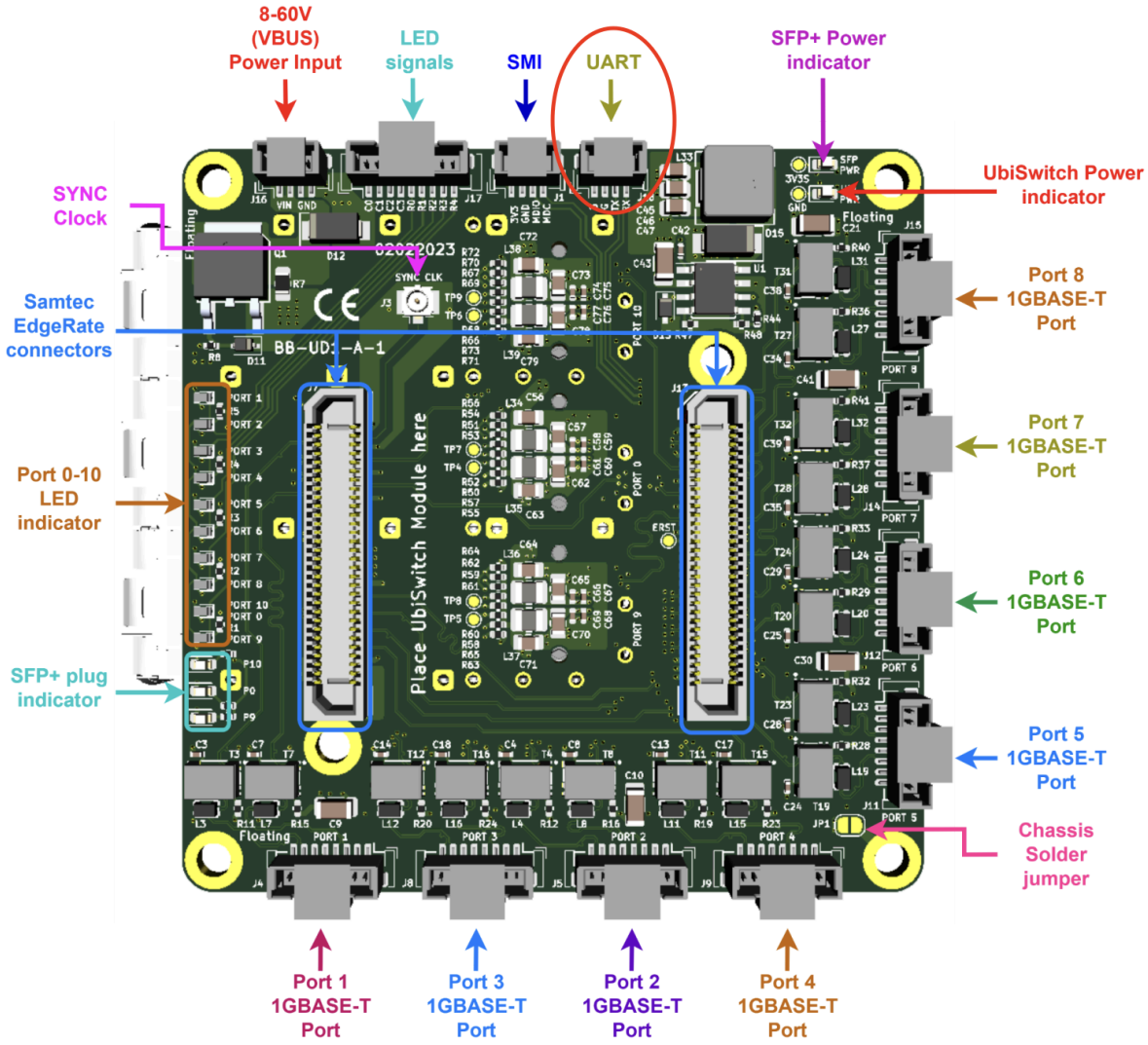
This TTL Serial (UART) is NOT 5V Tolerant. You must only connect a 3.3V level UART device here. Connecting a 5V device will permanently damage UbiSwitch.

Note that the UART connection is not isolated from the main ground on UbiSwitch. Therefore you must make sure UbiSwitch and the connected UART device share the same ground.

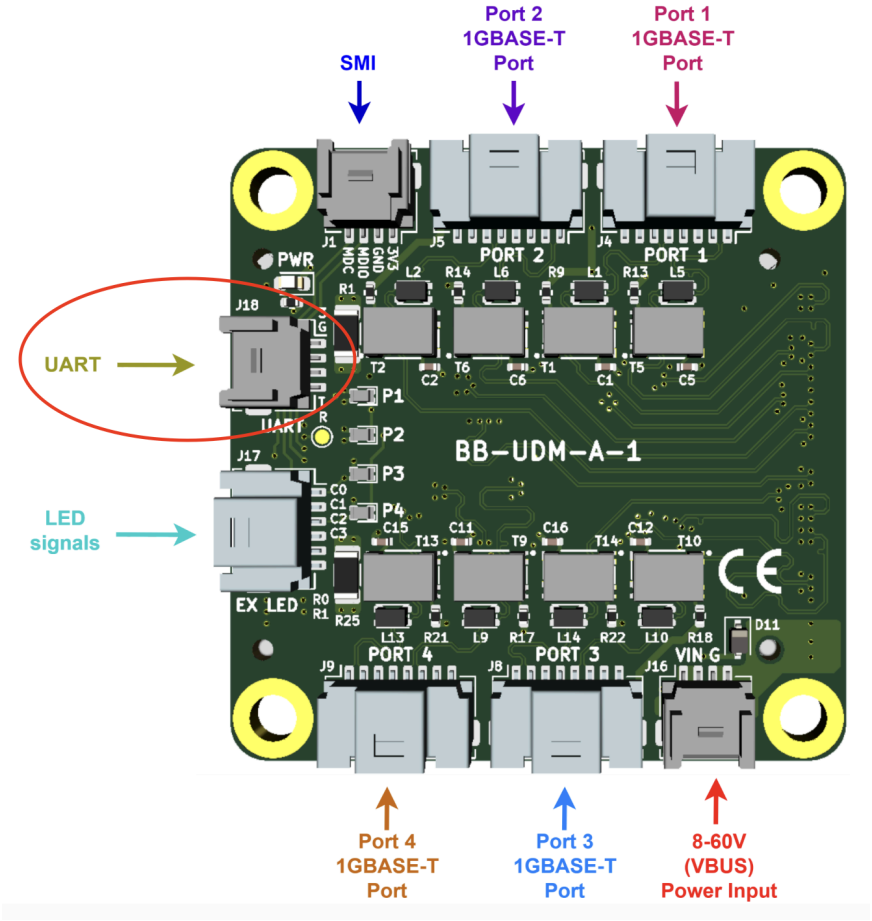
You can get around this limitation by using an isolated line driver or isolated UART to USB converter.

3.2 Using the BotBlox baseboards

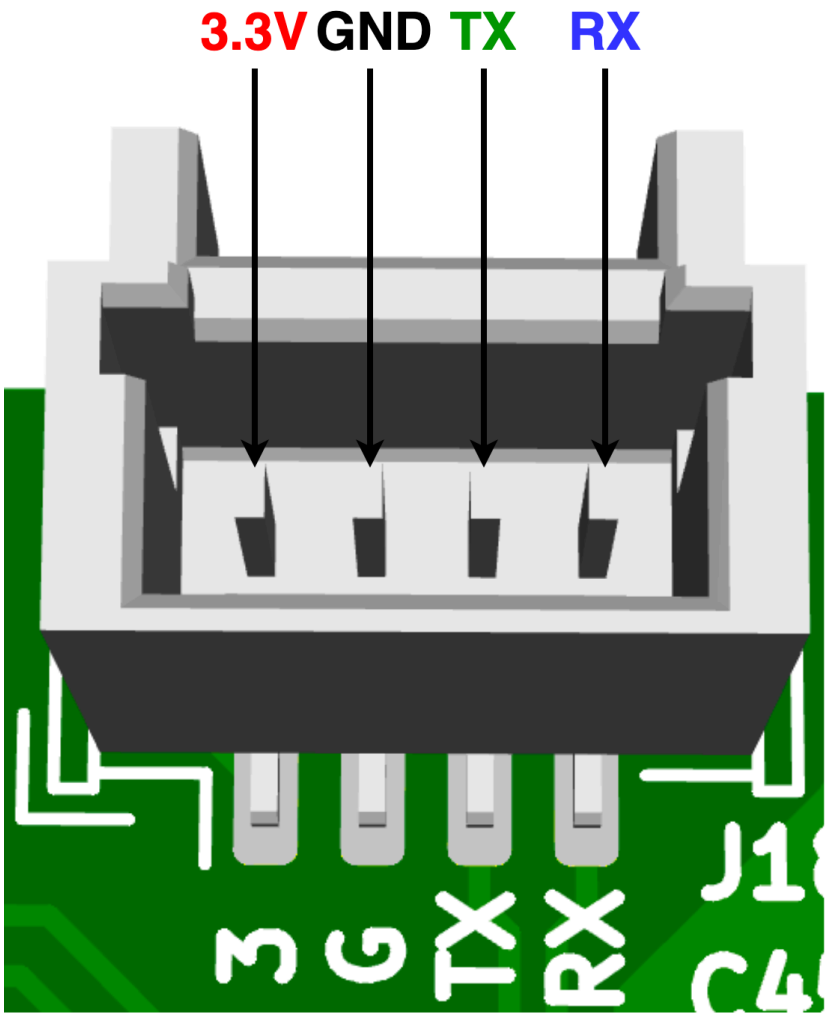
The image below shows where the UART port is located on the UbiSwitch BaseBoard.



The image below shows where the UART port is located on UbiSwitch BaseBoard Mini.



The pinout of both UART connectors is identical on both boards, and is shown in the image below.



TX is an output from UbiSwitch, connect it to the RX of the UART device.
RX is an input to UbiSwitch, connect it to the TX of the UART device.
GND on UbiSwitch should be connected to the ground on your UART device.
3.3V is an output from UbiSwitch, whether you need to connect this to your UART depends on your UART device.

- Some UART device outputs a 3.3V to power a target, then do not connect this to the 3.3V signal on UbiSwitch.

- Some UART devices require a 3.3V reference to be input to set the level of TX and RX. In this case, do connect UbiSwitch's 3.3V output to the UART device's 3.3V. Be aware that the UART device must not draw more than 200mA from this 3.3V.
- Most UART devices output a 3.3V signal, thus do not need the 3.3V from UbiSwitch to be connected.
- If you are in doubt then start by not connecting the 3.3V signal and check communications.

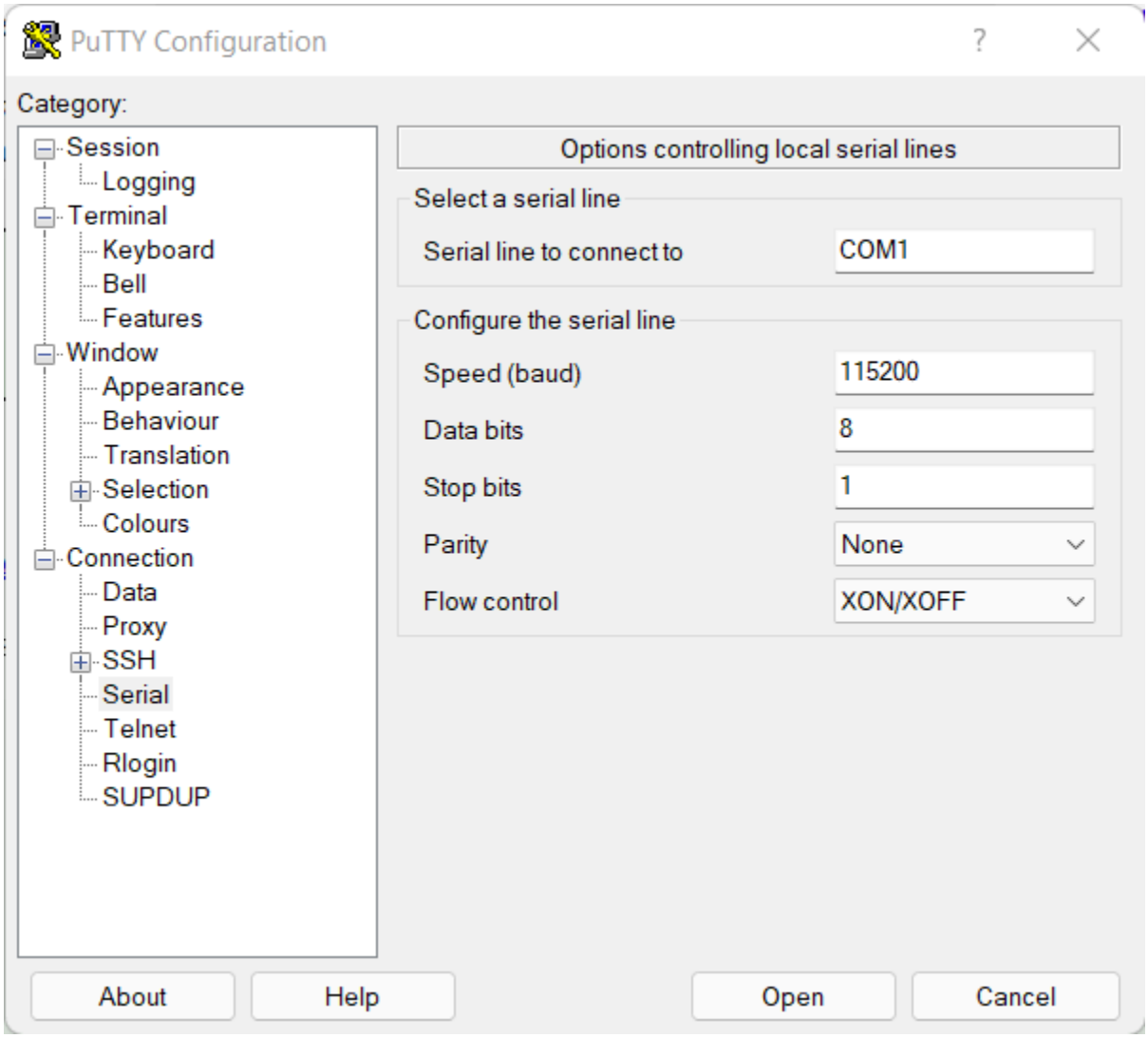
This TTL Serial (UART) is NOT 5V Tolerant. You must only connect a 3.3V level UART device here. Connecting a 5V device will permanently damage UbiSwitch.

3.3 Step by step guide

Step 1: With UbiSwitch powered off, connect the UART port of UbiSwitch to your PC.

Step 2: Open PuTTY, and open the serial port associated with the UART port on the PC. Use the following settings.

- Type: Serial
- Baud Rate: 115200bps
- Parameters: 8 data bits, no parity, 1 stop bit (8-n-1)



Step 3: Open a terminal on PuTTY using these settings

Step 4: Power on UbiSwitch. At power on, it will output text on the terminal.

4 Command Guide

4.1 Port PHY status/control

For a quick reference of how to use these commands, type `port 1 help` into the command line interface.

```
ubiswitch:~$ port 1 help
1 - Argument corresponds to port number
Subcommands:
  link     :Set port state:
            Required:
            port <int> - port
            link <str> - toggle port (on, off)
            Example:
            port 1 link off - set port 1 off, link will not be formed with link
            partner

  autoneg  :Set port autonegotiation:
            Required:
            port <int> - port
            autoneg <str> - toggle autoneg (on, off)
            -
            Optional:
            speed <int> - port speed (10, 100, 200, 1000, 2500, 5000, 10000)
            duplex <str> - port duplex (full, half)
            -
            Example:
            port 1 autoneg off speed 1000 duplex full - set port 1 phy autoneg
            off with speed 1000Mbps at full duplex
            port 1 autoneg on - set port 1 phy autoneg on

ubiswitch:~$
```

Listed below are the commands that allow you to control port speed, duplex and auto-negotiation settings.

4.1.1 Enable/disable link

```
[port <int>] link [{"off", "on"}]
```

Parameters

Property	Type	Description	Required?
`port`	`int`	Switch port PHY number	✅
`link`	`string`	String literal describing whether the PHY should have link down or up	✅

Example

```
port 1 link off
port 1 link on
```

Disable and then re-enable port 1 link

Available on `v0.1.0` or higher.

4.1.2 Change PHY settings

```
[port <int>] change [autoneg {"off", "on"}] [speed <int>] [duplex
{"full", "half"}] [advertise <int> <int> ...]
```

Parameters

Property	Type	Description	Required?
`port`	`int`	Switch port PHY number	✅
`autoneg`	`string`	String literal describing whether the port's PHY should enable auto-negotiation.	❌
`speed`	`int`	PHY speed if auto-negotiation disabled. Must be (10, 100, 200, 1000, 2500, 5000, 10000). Required if auto-negotiation is disabled.	❌
`duplex`	`string`	PHY duplex if auto-negotiation disabled. String literal must be `"full"` or `"half"`. Required if auto-negotiation is disabled.	❌

`advertise`	`List[int]`	The advertised link modes supported by the PHY. You can add support for multiple link modes by providing a list of ints. `0x01` - 10BaseT half `0x02` - 10BaseT full `0x04` - 100BaseT half `0x08` - 100BaseT full `0x10` - 1000BaseT half `0x20` - 1000BaseT full	❌

Example

Disable auto-negotiation

```
port 1 change autoneg off speed 100 duplex half
```

Disable auto-negotiation and set link speed to 100Mbps, half-duplex.

Re-enable auto-negotiation

```
port 1 change autoneg on
```

Change link mode advertisement

```
port 1 change advertise 0x01 0x02
```

Changes link mode advertisement on port 1 to only support 10BaseT half and full modes

Available on v0.2.0 or higher.

4.1.3 Saving Port MAC/PHY configuration

This allows you to save port MAC and PHY configurations stored so that the configuration will be written on reset. This command works on all ports simultaneously, it does not work on a port-by-port basis.

```
port save
```

Available on v0.2.0 or higher.

4.1.4 Clearing Port MAC/PHY configuration

This allows you to clear port MAC and PHY configuration from memory so that on reset, no port configuration will be loaded on reboot. This command works on all ports simultaneously, it does not work on a port-by-port basis.

```
port clear
```

Available on v0.3.1 or higher.

4.1.5 Show Port MAC/PHY configuration

Shows the configuration of a specific port.

```
[port <int>] show conf
```

Available on v0.2.0 or higher.

4.1.6 Read Port Temperature

```
[port <int>] temperature
```

4.2 VLAN

Allows VLAN control on the switch's ports. For a quick reference of how to use these commands, type `vlan help` into the command line interface.

```
ubiswitch:~$ vlan help
vlan - VLAN commands:
Subcommands:
  add     :Set VLAN:
           Required:
           port <int> - port to add to vlan group
           vid <int> - vlan id
           -
           Optional:
           <'pvid'> - set default port vid to vid given ('pvid')
           <'untagged'> - pass egress packets untagged from port ('untagged')
           -
           Example: vlan add port 1 vid 100 pvid untagged - add vlan 100 to port
           1 with pvid=vid, untagged=true

  del     :Delete VLAN:
           Required:
           port <int> - port to add to vlan group
           vid <int> - vlan id
           -
           Example: vlan del port 1 vid 100 - delete vlan 100 from port 1

  save    :Save VLAN configuration to EEPROM
           Example: vlan save

  show    :Show VLAN configuration:
           Example: vlan show

  clear   :Clear saved VLAN configuration from EEPROM:
           Example: vlan clear

ubiswitch:~$
```

4.2.1 Adding a VLAN member

```
vlan add [port <int>] [vid <int>] [{"pvid"}] [{"untagged"}]
```

Parameters

Property	Type	Description	Required?
`port`	`int`	Port number. Must indicate a port which actually exists	✅
`vid`	`int`	VLAN identifier for VLAN group. Must be between `1` and `4096`.	✅
`pvid`	`string`	String literal for whether the VLAN identifier will be added as a the port's default VLAN information	❌
`untagged`	`string`	String literal for whether the packets egressing from the port should be untagged, as is the case for access ports	❌

Example

Setting trunk ports

```
vlan add port 1 vid 5
vlan add port 1 vid 4
```

Adds port 1 to be a member of both VLAN groups with ID 4 & 5. Used for trunk ports for inter-switch connections.

Setting access ports

```
vlan add port 2 vid 4 pvid untagged
```

Adds port 2 to be a member of a single VLAN group with ID 4. VID=4 is added as the port's default VLAN information and egressing packets are untagged. Used for access ports to end devices.

Available on `v0.1.0` or higher.

4.2.2 Deleting a VLAN member

```
vlan del [port <int>] [vid <int>]
```

Parameters

Property	Type	Description	Required?
`port`	`int`	Port number. Must indicate a port which actually exists	✅
`vid`	`int`	VLAN identifier for VLAN group. Must be between 1 and 4096.	✅

Example

Deleting a port from a VLAN

```
vlan add port 1 vid 3
vlan del port 1 vid 3
```

Adds port 1 and then removes port 1 from VLAN with ID=3.

Available on v0.1.0 or higher.

4.2.3 Saving VLAN configuration

This allows you to save the current VLAN configuration stored in the switch into memory so that the configuration will be retained after reset.

```
vlan save
```

Available on v0.1.0 or higher.

4.2.4 Clearing VLAN configuration

This allows you to clear the saved VLAN configuration from memory so that on reset, no VLAN configuration will be loaded.

```
vlan clear
```

Available on v0.1.0 or higher.

4.2.6 Show VLAN configuration

Shows all saved VLAN assignments.

```
vlan show
```

Available on v0.1.0 or higher.

4.3 Link Aggregation / LAG

Allows you multiple ports to be combined into a single logical link for added redundancy and bandwidth. For a quick reference of how to use these commands, type `lag help` into the command line interface.

```
ubiswitch:~$ lag help
lag - LAG/LACP commands:
      Use these commands to create and delete Link Aggregation groups. This
      provides a way to aggregate different ports into a single logical bonded
      port.
Subcommands:
  create  :Create LAG Group:
           Required:
           <str> - 4 character LAG name
           -
           Example: lag create bnd0 - create LAG with name 'bnd0'

  add     :Add port to LAG Group
           Required:
           id <str> - LAG name
           port <int> - port number
           -
           Example: lag add id bnd0 port 1 - add port 1 to LAG 'bnd0'

  leave   :Remove port from LAG Group:
           Required:
           id <str> - LAG name
           port <int> -
           -
           Example: lag remove id bnd0 port 1 - remove port 1 from LAG 'bnd0'

  del     :Delete LAG Group
           Required:
           <str> - 4 character LAG name
           -
           Example: lag del bnd0 - delete LAG with name 'bnd0'

  save    :Save LAG configuration to EEPROM. The configuration will be loaded on
           reset
           Example: lag save

  show    :Show LAG groups currently set on switch.
           Example: lag show

  clear   :Clear saved LAG configuration from EEPROM
           Example: lag clear

ubiswitch:~$
```

4.3.1 Create an LAG group

```
lag create [<string>]
```

Create an LAG interface which you can add/remove ports from.

Parameters

Property	Type	Description	Required?
`id`	`string`	Name given to the LAG group. Must be 4 characters or less.	✅

Example

Create an LAG with name bnd0

```
lag create bnd0
```

Available on v0.1.0 or higher.

4.3.2 Delete a LAG group

```
lag del [<string>]
```

Parameters

Property	Type	Description	Required?
`id`	`string`	Name given to the LAG group. Must be 4 characters or less. The LAG must not have any ports added and its name must exist.	✅

Example

```
lag del bnd0
```

Available on v0.1.0 or higher.

4.3.3 Add a port to a LAG group

```
lag add [id <str>] [port <int>]
```

Parameters

Property	Type	Description	Required?
`id`	`string`	Name given to the LAG group. Must be 4 characters or less. The LAG must not have any ports added and it's name must exist.	✅
`port`	`int`	Switch port to add to LAG	✅

Example

```
lag create bnd0
lag create bnd1
lag add port 1 id bnd0
lag add port 2 id bnd0
lag add port 3 id bnd1
lag add port 4 id bnd1
```

Create two LAGs labelled bnd0 and bnd1 and add ports 1/2 and 3/4 to those LAG groups respectively.

Available on v0.1.0 or higher.

4.3.4 Remove a port from a LAG group

```
lag leave [id <str>] [port <int>]
```

Parameters

Property	Type	Description	Required?
`id`	`string`	Name given to the LAG group. Must be 4 characters or less. The LAG must not have any ports added and it's name must exist.	✅
`port`	`int`	Switch port to remove from LAG	✅

Example

```
lag create bnd0
lag add port 1 id bnd0
lag del port 1 id bnd0
```

Create an LAG labelled bnd0 and add, then immediately delete, port 1 from being part of that LAG.

Available on v0.1.0 or higher.

4.3.5 Saving LAG configuration

Save the current LAG configuration stored in the switch into memory so that the configuration will be retained between resets.

```
lag save
```

Available on v0.1.0 or higher.

4.3.6 Clearing LAG configuration

Clears the saved LAG configuration from memory so that on reset, no LAG configuration will be loaded.

```
lag clear
```

Available on v0.1.0 or higher.

4.3.7 Clearing LAG configuration

Shows the current LAG configuration

```
lag show
```

Available on v0.1.0 or higher.

4.4 PCS mode on SERDES ports (Ports 0, 9 and 10)

Allows you to change the mode of the SERDES ports on UbiSwitch, ports 0, 9 and 10.

4.4.1 Change PCS Mode

```
[port <int>] mac [mode {"1000basex", "sgmii", "2500basex",
"5gbaser", "10gbaser", "usxgmii"}]
```

Property	Type	Description	Required ?
`port`	`int`	Switch port PHY number	✅
`mode`	`string`	Possible PCS modes for SERDES ports supported: 1000basex, sgmii, 2500basex, 5gbaser, 10gbaser, usxgmii	✅

Available on v0.4.0 or higher.

4.5 Change EEE mode on PHY/MAC

Allows you to turn on or turn off Energy Efficient Ethernet (EEE) on any of the ports on UbiSwitch.

4.5.1 Change EEE Mode

```
[port <int>] eee [{"on", "off"}]
```

Property	Type	Description	Required ?
`port`	`int`	Switch port PHY number	✅
`eee`	`string`	**on** for advertise EEE to link partner, **off** for no advertisement EEE to link partner	✅

Available on v0.4.0 or higher.

5 Datasheet Changelog

Date	Datasheet Version	Author	Notes
5/12/2024	A_A	Andrei Tsugui, Josh Elijah	Initial release
30/01/2025	A_B	Josh Elijah	• Added extra guidance on 3.3V TTL serial port protection • Added changing PCS mode feature using BloxOSLite • Added toggling EEE feature
27/02/2025	A_C	Josh Elijah	• Added notes to explain that the 3.3V signal on UbiSwitch is an output.

6 Contact

If you have any questions regarding this product, please contact us:

info@botblox.org
4 Pavilion Court 600 Pavilion Drive,
Northampton Business Park,
Northampton,
England
NN4 7SL