Quarch Technology Ltd

Application Note – AN-006

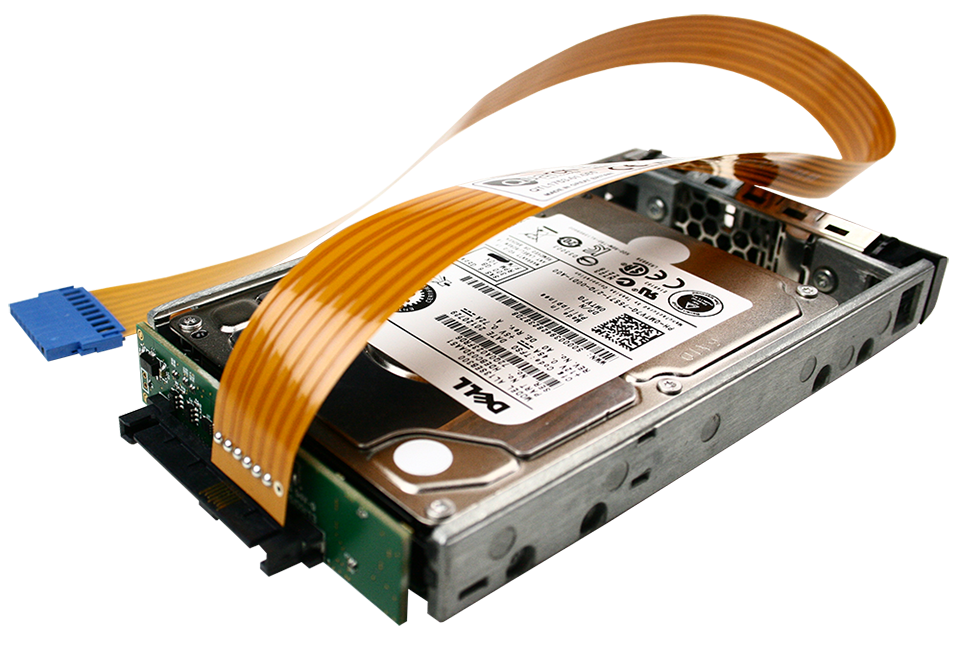
Python Control of Quarch Modules

For use with:

**Any Quarch module**

**QuarchPy Python Libraries**

Wednesday, 20 February 2019



# Change History

|  |  |  |
| --- | --- | --- |
| 1.0 |  | Initial Release |
| 1.1 |  | Added ReST control option |
| 1.2 |  | Improved ReST control (escaping special characters)  Added Array Controller example |
| 1.3 |  | Added details for CentOS setup |
| 1.4 | April 2018 | Major changes, now uses PIP installed library  Greatly simplifies install and version control |
| 1.5 | Feb 2019 | Updated for Python 3.x |
| 1.6 | May 2019 | Updated for quarchpy 2.x  New device and test selection in console instead of code |

Contents

[Change History 2](#_Toc512349223)

[Introduction 4](#_Toc512349224)

[Modules Supported 4](#_Toc512349225)

[Requirements 4](#_Toc512349226)

[Application Note Example Files 4](#_Toc512349227)

[Install Components 5](#_Toc512349228)

[Quarch USB Driver install 5](#_Toc512349229)

[Python install 5](#_Toc512349230)

[QuarchPy library install 5](#_Toc512349231)

[Setting up to run the examples 7](#_Toc512349232)

[Linux USB Permissions 7](#_Toc512349233)

[Hardware Setup 8](#_Toc512349234)

[Selecting the connection type 8](#_Toc512349235)

[Selecting the example to run 9](#_Toc512349236)

[Test description 10](#_Toc512349237)

# Introduction

Quarch modules can be easily controlled using Python and a USB, Serial or LAN connections.

This application note provides a number of simple examples, showing how to connect to and automate control of our products.

This application note (as of 1.4) makes use of the QuarchPy python library. This is a break from the previous versions, but is far easier to install and upgrade than previous versions.

## Modules Supported

All standard Quarch modules, which support LAN, Serial or USB interfaces

## Requirements

* 1. Install Quarch Torridon USB Drivers (if using Windows)
  2. Install Python 3.x
  3. Install QuarchPy Library

## Application Note Example Files

The **AN-006 - Python Control of Quarch Modules.zip** should be extracted to your preferred location.

**Python Control Examples.py**This file contains a number of simple python examples, which can be selected through the console window.

**QuarchPy Library**  
This library is required to run all Quarch Python examples and download instructions for it are later in this document

# Install Components

## Quarch USB Driver install

For windows, if you want to use USB control of modules, install the Torridon USB driver:

<https://quarch.com/file/torridon-driver-win8>

Download the file, extract and run the correct installer for your system.

## Python install

If you do not already have Python 3.x installed, download and install it from:

<https://www.python.org/downloads/>

Under Windows it is helpful to make sure the Python installation directory and PythonXX\Scripts are included in the PATH environment variable. See:

<https://docs.python.org/3/using/windows.html#excursus-setting-environment-variables>

## QuarchPy library install

The Quarch Python package can be installed from the Python web repository (assuming you have internet access) or via the download from our website.

### Web Install

From the command line:

**>pip install quarchpy**

If this fails, your path to “pip” may not be set, you can instead run:

**>python –m pip install quarchpy**

### Local Install

If you want to install from a downloaded folder, ensure the folder is unzipped to a local disk, navigate to the folder containing the setup.py file and run (noting the ‘.’ on the end):

**>pip install quarchpy .**

If this fails, your path to ‘pip’ may not be set, you can instead run:

**>python –m pip install quarchpy .**

### Upgrade

If you already have QuarchPy installed, you will get a failure message. If you want to upgrade to a new version, you need to add the ‘--upgrade’ command:

**>pip install --upgrade quarchpy**

The --upgrade command can similarly be used in any of the other examples, to load from a local install folder.

# Setting up to run the examples

## Linux USB Permissions

Linux systems require administrative rights to run python scripts for modules connected via USB. You can do that by running your script as root (sudo command) or changing the default USB permissions. This is done by creating a text file called **Quarch-permissions-usb.rules**

For ubuntu systems, you need to enter into that file:

SUBSYSTEM == “usb”, ATTRS{idVendor}==”16d0”, MODE=”0666”

SUBSYSTEM == “usb\_device”, ATTRS{idVendor}==”16d0”, MODE=”0666”

For Centos systems, you need:

SUBSYSTEM == “usb”, ATTRS{idVendor}==”16d0”, GROUP=”users”, MODE=”0666”

SUBSYSTEM == “usb\_device”, ATTRS{idVendor}==”16d0”, GROUP=”users”, MODE=”0666”

This file needs to be placed in /etc/udev/rules.d

Finally, the system either needs to be restarted or run the command:

**>sudo udevadm control -reload**

Then reconnect the USB device.

## Hardware Setup

Serial, USB or Ethernet

Host PC

Quarch Module

## Selecting the connection type

Our script now askes the user to select the device by inputting the index of the device displayed in the console. The script currently displays the preferred connection to a device in the order USB , TCP ,SERIAL, REST, TELNET. If you would like to list all the connections to all the devices available then you can add favouriteOnly = False to the passed parameters of scanDevices().

## Selecting the example to run

Depending on the module connected, select the correct test when prompted from the console.

* QuarchSimpleIdentify – Works with any module
* QuarchHotPlugExample – Works with all standard hot-plug modules
* QuarchSwitchExample – Works with SAS switch modules
* QuarchPowerMarginingExample – Works with Power Modules
* QuarchArrayExample – Works with array controllers

These are represented by these lines of code executing the functions mentioned above

QuarchHotPlugExample(myDevice)

QuarchSwitchExample(myDevice)

…

# Test description

Each test function will:

* 1. Demonstrate the sending of commands to the module
  2. Show how to read back and use a response to a command

Each example is different (depending on the functions available on the module) and tries to show a genuine, practical test that you can perform.