Quarch Technology Ltd

AN-004

Application Note

For use with:

**Original Power Modules  
XLC Power Modules**



# Change History

|  |  |  |
| --- | --- | --- |
| -01 |  | Initial Release |
| -02 |  | Improved installation and library structure |
| -03 |  | Bug fix where commands could fail to run |
| -04 |  | Support added for QTL1658 modules |
| -05 |  | Fixed bug with setting a custom save path |
| -06 |  | Fixed bug with device identity |
| -07 | Feb 2017 | Improved documentation and examples |
| -08 | May 2017 | Fixed USB duplicate module issue |
| -09 | Feb 2019 | Updated for Python 3.x |

Contents

[Change History 2](#_Toc475533244)

[Introduction 4](#_Toc475533245)

[Modules Supported 4](#_Toc475533246)

[Installation and setup 5](#_Toc475533247)

[Linux setup 5](#_Toc475533248)

[Windows setup 8](#_Toc475533249)

[Provided Examples 10](#_Toc475533250)

[Basic Commands 10](#_Toc475533251)

[RAM capture example 11](#_Toc475533252)

[Streaming capture example 12](#_Toc475533253)

[Customer support from Quarch 13](#_Toc475533254)

[Access support from the Quarch website 13](#_Toc475533255)

[Find discussion topics, support information and testing ideas. 13](#_Toc475533256)

# Introduction

Quarch Programmable Power Modules (PPMs) can be controlled using Python and a USB connection. This document explains how to setup Python to interface with a Quarch module using USB.

The examples make use of the libusb USB library, libusb1 Python package, and QuarchUSB Python library to simplify communicating over USB.

This method of control is now considered ‘legacy’, new implementations are suggested to use the QIS (Quarch Instrumentation Server) examples in AN-012.

This examples can be implemented on both Windows and Linux. It has been tested on Win 7, Win 10 and Ubuntu. Minor changes may be required for other systems.

# Modules Supported

Original power modules:

* QTL1455
* QTL1658
* QTL1730
* QTL1727

XLC power modules

* QTL1824 (QTL1824-02A modules do NOT support streaming, but can be upgraded by Quarch)
* QTL1847

HD power modules

* None currently supported, see AN-012

# Installation and setup

## Linux setup

This setup was based on Ubuntu Linux and a clean install. Changes may be needed for other systems. You will require admin access for the install.

* 1. Copy the application note files to a folder on your Linux system
  2. Check python 2.x is installed:

Open a terminal window

>python

The response must be a v2.x version (3.x is not supported)

You can install python with

>sudo apt-get install python

* 1. Run the basic test script. No module is needed at this point

>sudo python BasicFunctionTests.py

If a list of USB devices is displayed, the test has passed, skip to step 7. At this point you are likely to see errors about loading usb libraries (indicating your system needs to install them).

* 1. Install libUSB 1.0 for linux (later versions are not supported)

>sudo apt-get install libusb-1.0.0-dev

You can also go to <http://www.libusb.info> for downloads and help

* 1. Install python libUSB1 wrapper

This wrapper is provided in the lib folder

Go to [YourFolder]/lib/libusb1-1.x.x

>sudo python setup.py install

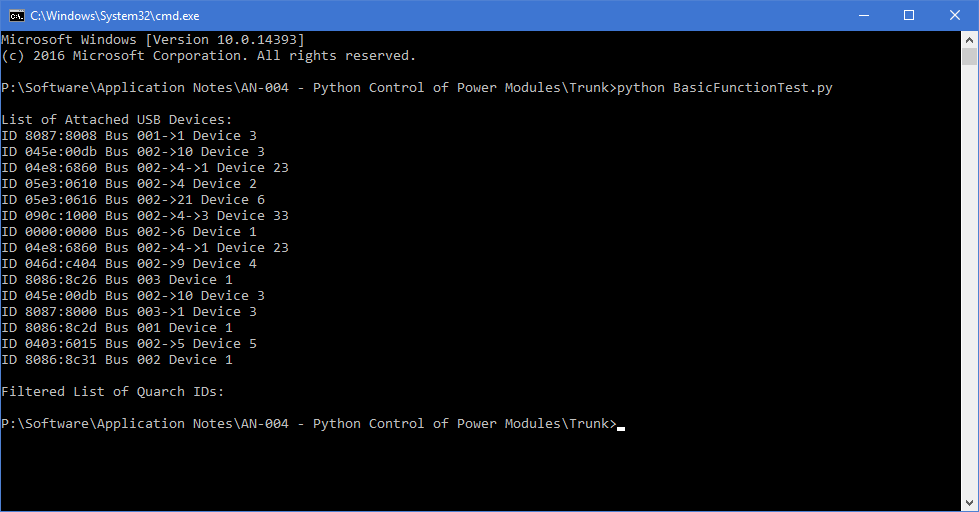
If you do not have Python ‘setup tools’ installed, you will get an error, to fix this, install the tools:

>sudo apt-get install python-setuptools

You can also obtain and install this package via instructions at <https://pypi.python.org/pypi/libusb1>

* 1. Run the basic test script again, and verify that USB devices are now seen

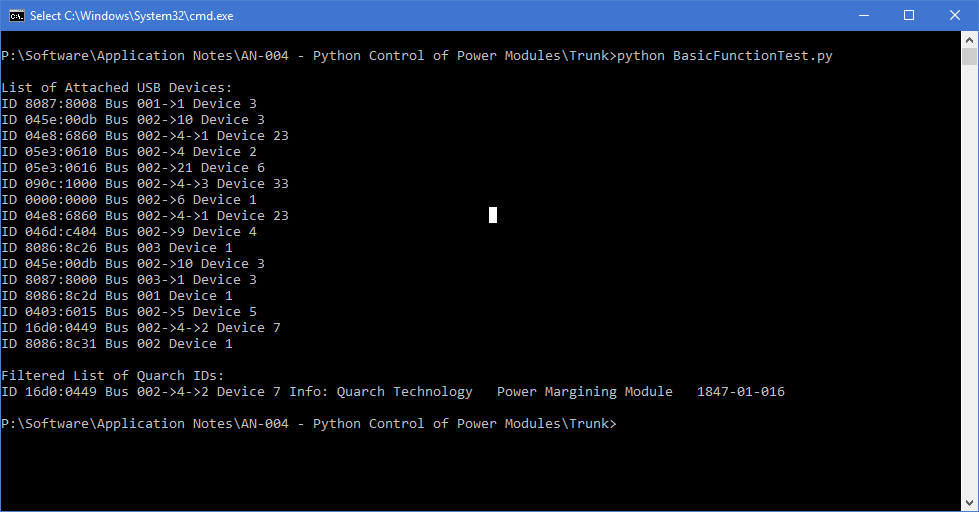
>sudo python BasicFunctionTests.py



* 1. Attach the Quarch power module and run the test again

>sudo python BasicFunctionTests.py

You should now see a list of all USB modules, followed by a list of Quarch devices attached



* 1. You are now fully ready to run the main examples

Default USB permissions on Ubuntu require ‘sudo’ to run the script. This can be avoided by setting explicit permissions for Quarch USB devices.

Create a text file called Quarch-Permissions-USB.rules

Into the file, enter the text:

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

Save the file and close it

On the command line, copy it into the rules folder:

**>sudo cp** Quarch-Permissions-USB.rules /etc/udev/rules.d

You will need to reset for this to take effect

## Windows setup

This setup was based on Windows 10 and a clean install. Changes may be needed for other systems. You may require admin access for the install.

* 1. Copy the application note files to a folder on your system
  2. Download and install the Quarch USB driver from: <http://quarch.com/downloads?field_file_type_tid=113>
  3. Check python 2.x is installed

If not, download and install it from:   
<https://www.python.org/downloads>

* 1. Run the basic test script from the copied folder. No module is needed at this point

>python BasicFunctionTests.py

If a list of USB devices is displayed, the test has passed, skip to step 7. At this point you are likely to see errors about loading usb libraries (indicating your system needs to install them).

* 1. libUSB 1.0 for windows

This is provided in the \lib folder of the application note, and automatically linked in at run time. You should not have to install anything.

* 1. Install python libUSB1 wrapper

This wrapper is provided in the lib folder

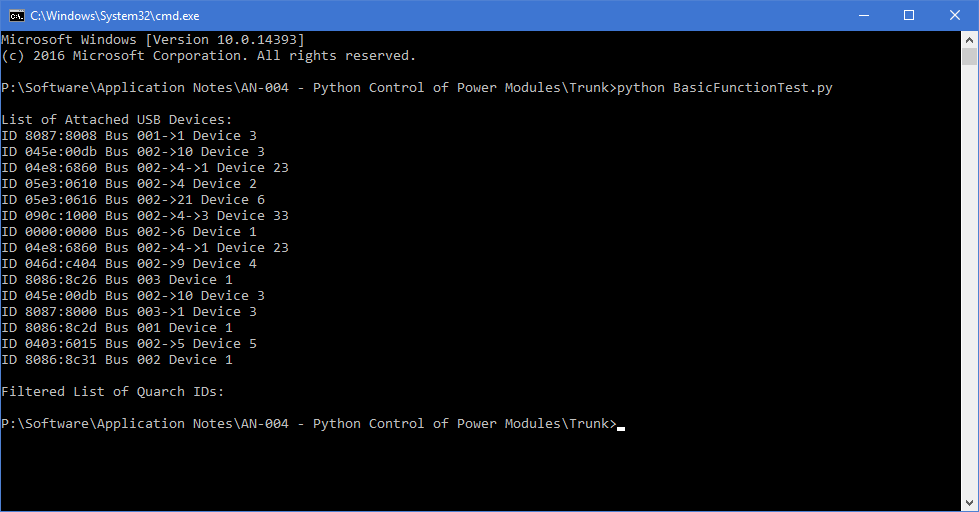
Go to [YourFolder]/lib/libusb1-1.x.x

>python setup.py install

You can also obtain and install this package via instructions at <https://pypi.python.org/pypi/libusb1>

* 1. Run the basic test script again, and verify that USB devices are now seen

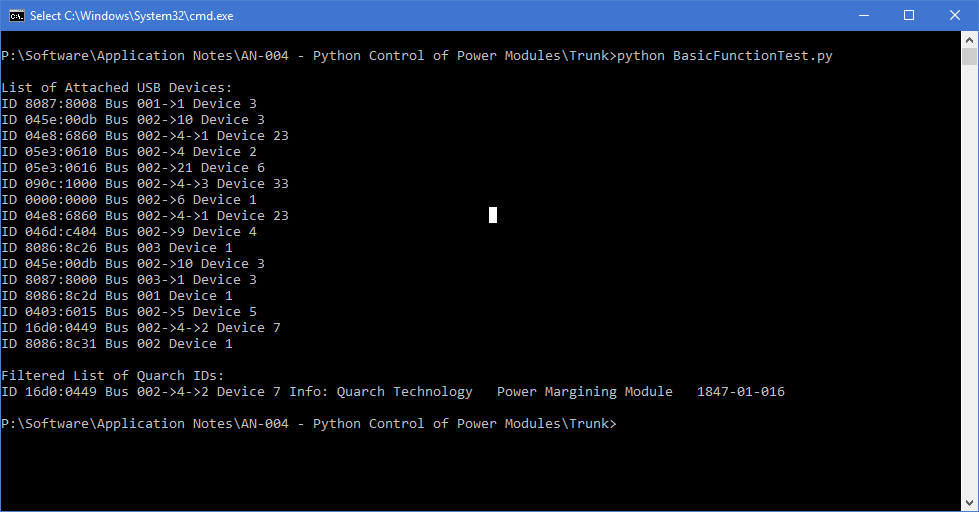
>python BasicFunctionTests.py



* 1. Attach the Quarch power module and run the test again

>python BasicFunctionTests.py

You should now see a list of all USB modules, followed by a list of Quarch devices attached



* 1. You are now fully ready to run the main examples

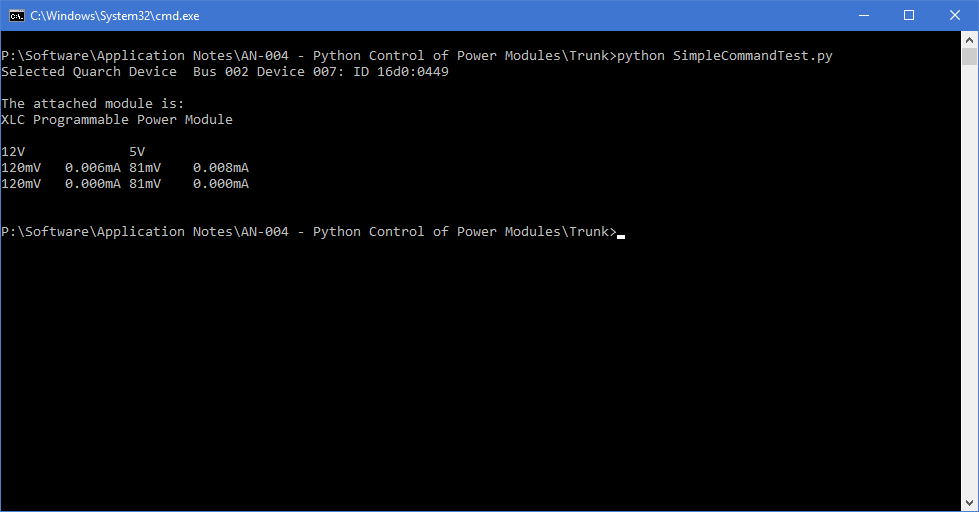
# Provided Examples

## Basic Commands

The file SimpleCommandTest.py contains a simple example, connecting to the module over USB, running a series of basic commands and printing the measurements retrieved from the device.

These methods let you see how to set the module up and how to take measurements. All the commands you can use are found in the main technical manual for the product.

>python SimpleCommandTest.py



The output includes the name of the attached module and measurements for both output channels.

## RAM capture example

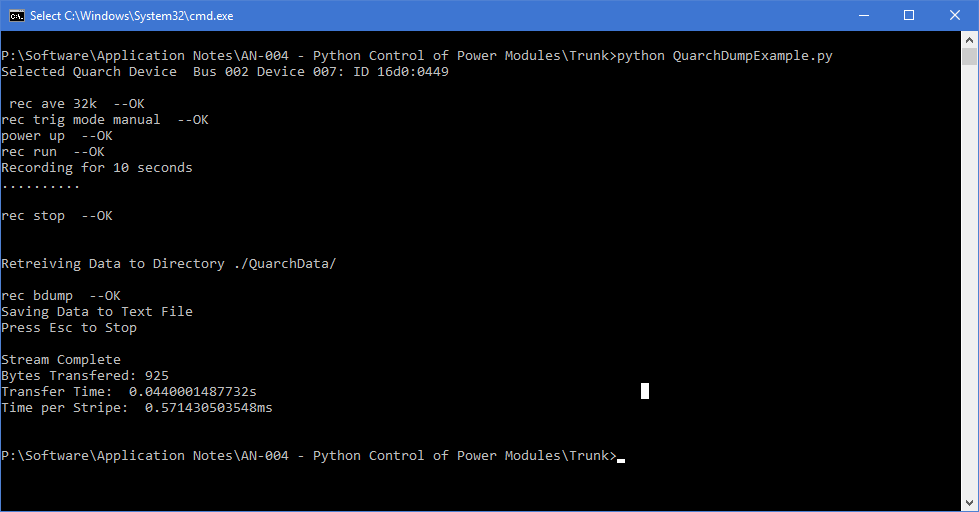
The file QuarchDumpExample.py connects to the module over USB and runs a number of commands to prepare for a capture to the module’s internal RAM.

It then triggers the capture operation and waits while for a period of time. This time could be used to run traffic to the drive under test, or to send additional commands to the Quarch module (for example: to margin the voltage output).

Finally, we send a command to stop the recording, then download the data to a CVS file, for later viewing and processing.

This example uses a manual trigger to start capture, but any form of trigger (such as threshold or external trigger could also be used).

>python QuarchDumpExample.py



The output the commands that we ran (and their responses). It also shows the statistics on the data that we pulled back from the module.

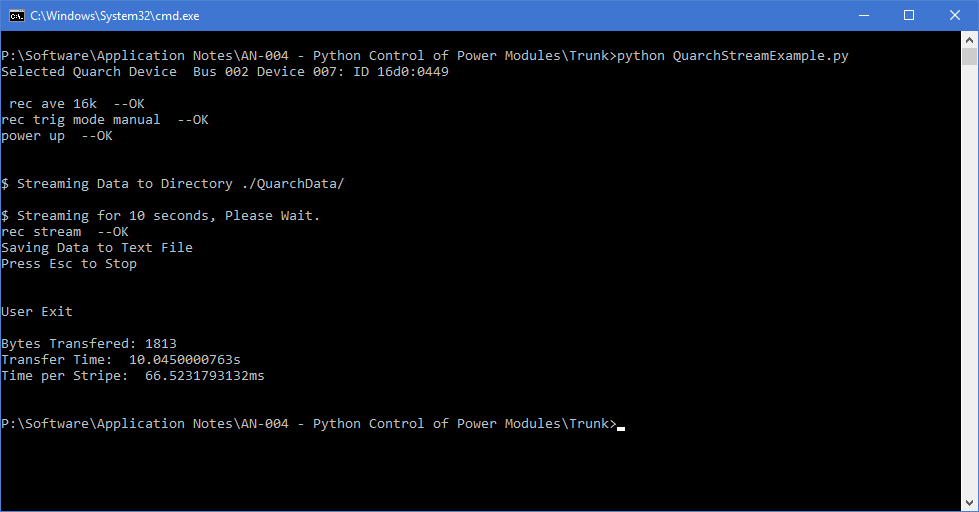
## Streaming capture example

The file QuarchStreamExample.py connects to the module over USB and runs a number of commands to prepare for streaming capture of data.

‘Streaming’ involves the PC client software constantly puling data back from the Quarch module. This allows for much longer recording times that when using the internal RAM on the module. The lower the averaging rate, the faster that data must be read back to keep up. XLC modules can generally stream at 16 sample averaging with a fast PC. This example is set to a far higher value initially so as to produce a small and simple CSV file.

The script sets up the module and then begins streaming for a set period of time. When complete a CVS file will have been created, for later viewing and processing.

>python QuarchStreamExample.py



The output the commands that we ran (and their responses). It also shows the statistics on the data that we pulled back from the module.

# Customer support from Quarch

There are multiple ways to access the support you need. You can contact us directly or access an extensive range of valuable support materials from <http://quarch.com/support>.

* Contact us direct
* Get going quickly and easily, with help direct from the engineers:
* Call +44 1343 508 140 or email [support@quarch.com](mailto:support@quarch.com) during UK office hours.
* Our international partners are well trained in the use of our products and can deal with many basic technical queries from within your time zone, if you prefer. Check <http://quarch.com/resellers> for the contact details of your regional supplier.

## Access support from the Quarch website

You can download up-to-date software and drivers, technical manuals, datasheets and more from our website. To help you get started quickly we provide additional documents, such as examples in Perl, Python and C# and Telnet and Serial instructions.

* Key **places to visit on** the Quarch website
* Register your Quarch product to confirm your international warranty: <http://quarch.com/product-registration>
* Download a wide range of documentation, free applications and drivers to help you make the best possible use of your Quarch tools: <http://quarch.com/content/downloads>
* Access the Quarch support forum (<http://quarch.com/forum>):

## Find discussion topics, support information and testing ideas.

Browse existing topics or login to your user account to ask for information and advice.

Sign up for Quarch Technical Updates to get the most out of your Quarch products. Updates are published approximately once a quarter and include news about the latest features, tools, application notes and software updates. See <http://quarch.com/content/sign-quarch-technical-updates>.