

M.2 PQU – Usage Manual

Table of Contents

1. Scope.....	2
1.1. Overview	2
2. Referenced Documents	2
3. Document History.....	2
4. PQU – QMA BOARD Setup Steps	3
4.1. Overview	3
4.2. Validation of Entropy Being Generated	5
4.3. Quality Tests	5

1. Scope

1.1. Overview

Guidance for how to install the M.2 Board and get random numbers from it by EigenQ, Inc.

2. Referenced Documents

None

3. Document History

DATE	VERSION	AUTHOR	DESCRIPTION
18-Sep-25	1.1	Pedro	Initial Draft
19-Sep-25	1.2	Andre	Reviewed Content and Design

4. PQU – QMA BOARD Setup Steps

Guidance for a technician to setup, install and use the M.2 PQU Board. To complete this task the following is required:

- Ubuntu Server 24.04.3
- Linux Kernel 6.8.X
- M.2 PQU Board
- Ubuntu 24.04.4 EigenQ Package containing:
 - Linux-Kernel Drivers for our M.2 PQU Board
 - EigenQ Accelerator Binary
 - EigenQ Systemd compatible start/stop script.

4.1. Overview

This section guides you through setting up and validating the EigenQ M.2 PQU Board on Ubuntu 24.04.x: confirm prerequisites, install the hardware, hold kernel updates, deploy EigenQ services, build/load the driver, and verify service health and entropy generation. Complete these steps in order before proceeding to the quality tests.

1. Flash Ubuntu Server 24 on the desired machine.
2. Confirm the kernel version is compatible
3. Install the M.2 PQU into the machine (turn off the machine, insert the device, and turn it on).
4. Confirm the M.2 PQU was recognized by the O.S, execute this on the terminal:

```
lspci|grep accelerators
```

Expected Result:

```
01:00.0 Processing accelerators: Xilinx Corporation Device...
```

5. Hold kernel updates, otherwise whenever the ubuntu kernel updates, you may need to recompile the driver.

```
sudo apt-mark hold linux-image-$(uname -r)
```

```
sudo apt-mark hold linux-image-generic linux-generic
```

6. Open the EigenQ Package and move the gq-qrng-daemon file to its location.

```
mv gq-qrng-daemon /usr/bin/gq-qrng-daemon
```

7. Fix its permissions.

```
chmod 755 /usr/bin/gq-qrng-daemon
```

8. Still on the EigenQ Package directory, move it to its location.

```
mv gq-accel-qrng.service /etc/systemd/system/
```

9. Fix the file permissions.

```
chmod 644 /etc/systemd/system/gq-accel-qrng.service
```

10. Now, unpack the linux-kernel-driver.tar file, and access it.

```
tar -xf linux-kernel.tar
```

```
cd linux-kernel
```

11. Install the driver dependencies, so we can compile it.

```
apt-get update && apt-get install make gcc g++
```

12. Access the xdma folder, compile and install the driver.

```
cd xdma && make install
```

13. Load the driver

```
modprobe xdma
```

14. Confirm the driver was properly loaded.

```
ls /dev/xd*
```

The command above should return, if nothing is returned, please reboot the machine, and check it again.

15. Now if the driver is loaded, let's start the service.

```
systemctl start gq-accel-qrng
```

16. Check if the service is working.

```
systemctl status gq-accel-qrng
```

4.2. Validation of Entropy Being Generated

We can check if the board is generating numbers by executing the following command and checking its throughput.

```
apt-get install pv
```

```
cat /dev/qrandom|pv -r -a -b -p > /tmp/rand_output
```

Expected output:

```
205MiB [18.0MiB/s] [17.1MiB/s] [
```

You can then stop the command by pressing control + c.

Delete the temporary file:

```
rm /tmp/rand_output
```

4.3. Quality Tests

We can verify the quality of the entropy being generated. For that we can use FIPS statistical tests or Dieharder.

Installation:

```
sudo apt-get update && apt-get install dieharder rng-tools5
```

Execution:

Generate some random data into a file that will come from EigenQ's Qrandom device:

```
dd if=/dev/qrandom of=/tmp/random_data bs=1G count=1 iflag=fullblock
```

FIPS statistical tests:

```
cat /tmp/random_data | rngtest
```

Dieharder statistical tests:

```
dieharder -a -g 201 -f /tmp/random_data > /tmp/random_data_dieharder_output &  
tail -f /tmp/random_data_dieharder_output
```