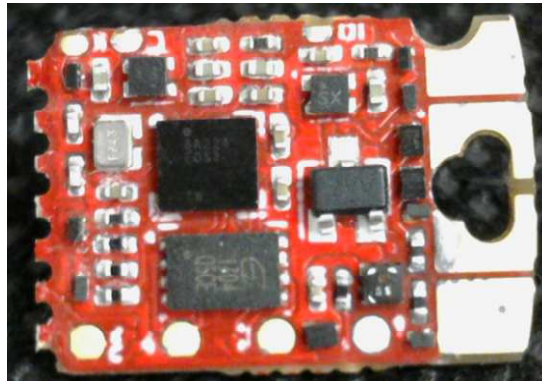


# ***Qomu Low Power MCU Open Source Hardware Development Kit***

## ***User Guide (Rev 1.0)***

This document serves as a guide for the user to getting started with the Qomu board.

This guide provides users with functional descriptions, configuration options for the Qomu Low Power MCU Open Source Hardware Development Kit. It also serves as a “Getting Started” and “How To” guide.



*Figure 1: Qomu Low Power MCU Open Source Hardware Development Kit*

## Qomu Low Power MCU Open Source Hardware Development Kit Overview

The Qomu Low Power MCU Open Source Hardware Development Kit (HDK) is a small form factor system ideal for exploring MCU + eFPGA development. Unlike other development kits which are based on proprietary hardware and software tools, Qomu is based on 100% open source hardware and is built around 100% open source software (including the Symbiflow FPGA Tools).

The Qomu HDK is powered by QuickLogic’s **EOS™ S3** (<https://www.quicklogic.com/products/eos-s3/>), the first eFPGA-enabled Arm Cortex®-M4F MCU to be fully supported with Zephyr RTOS and FreeRTOS. Other functionality includes:

- QuickLogic EOS S3 MCU Platform
- 16Mbit of on-board flash memory
- Touchpads (4) and RGB LED
- Powered from USB
- USB data signals tied to programmable logic

## Benefits

- Qomu HDK is small, fitted in a PC USB slot, inexpensive, and is 100% supported by open source tools.

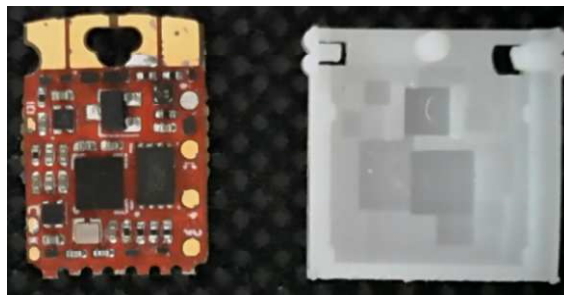
- With a Cortex M4F MCU and integrated eFPGA, the [EOS S3](#) lets you innovate with 100% open source hardware and software.

## Applications

- Tiny ML applications (such as with SensiML's AI Software Platform and Google's TensorFlow Lite) with eFPGA as accelerator engine
- General purpose MCU applications

## Getting Started

### What You Need



*Figure 2: Qomu & Plastic housing*

- 1 × Qomu HDK development board
- 1x Qomu USB plastic housing
- 1 x Micro-USB 2.0 extension cable<sup>1</sup>, Type A (male) to Type A (female) for Power and easy of connecting to PC
- Laptop (Win10 OS<sup>2</sup> or Linux OS) running terminal program (such as PuTTY<sup>3</sup> or any UART console application)

Note<sup>1, 2, 3</sup>:

<sup>1</sup> you will need to provide your own USB cable

<sup>2</sup> Windows 7 and Windows 8 are not supported

<sup>3</sup> PuTTY download & installation: <https://www.putty.org/>

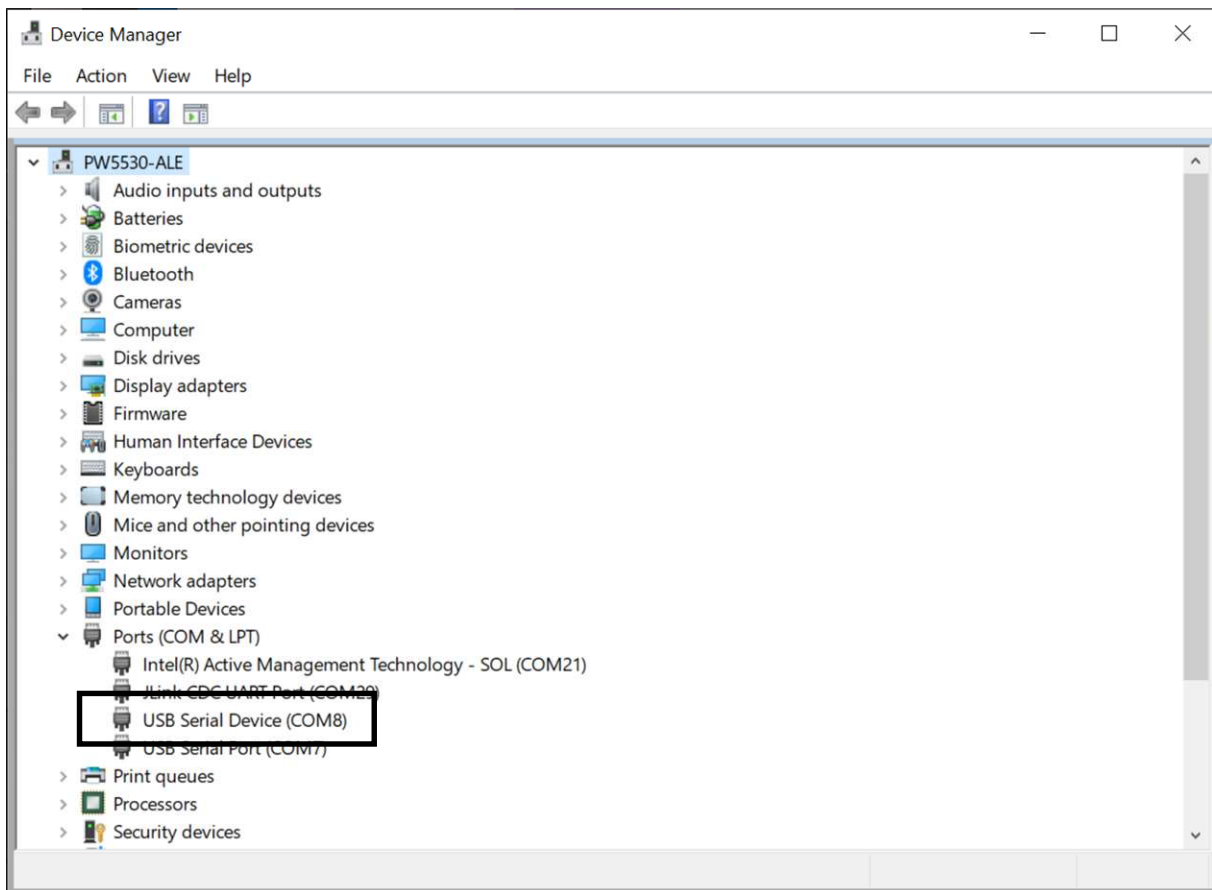
## Running pre-loaded program from Flash

The purpose of the pre-loaded program is to make it very fast and straightforward for a new user to verify the board is functioning correctly. While we do production testing of every Qomu HDK before we ship, sometimes things happen during shipping or storage. Running this test takes less than a minute to do, requires no knowledge of the HDK, and will give you peace of mind the HDK is ready for you to start innovating freely.



*Figure 3: Qomu insert into Plastic housing*

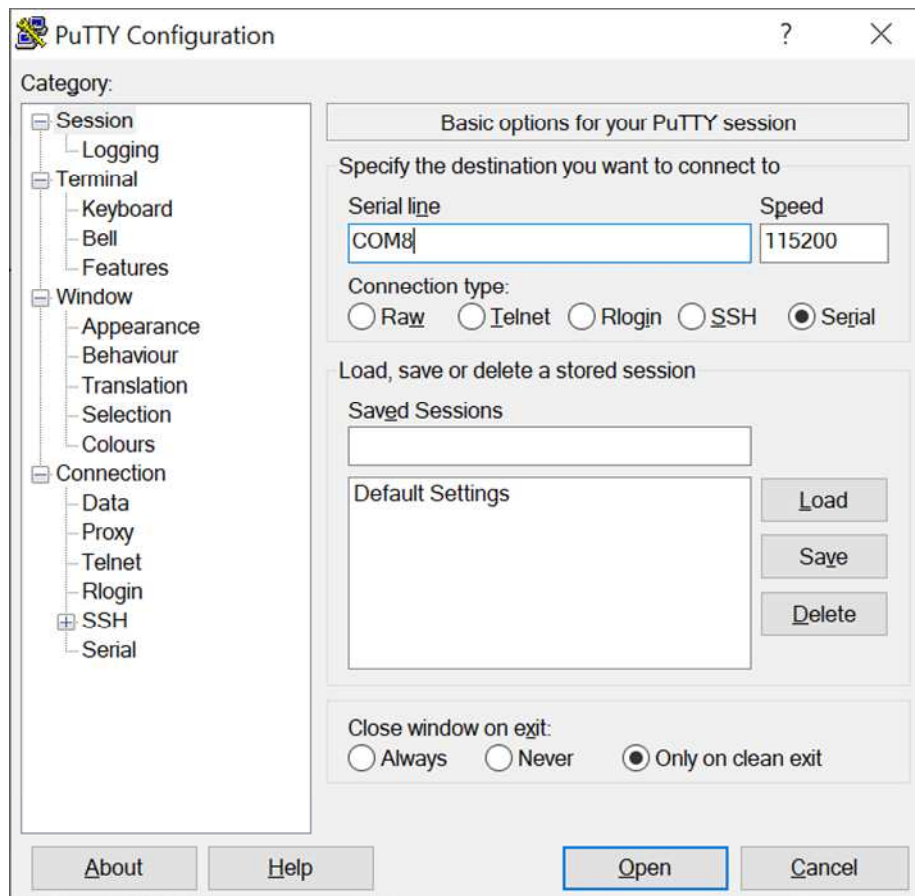
1. Place the Qomu board into the plastic housing.
2. Insert the housing into USB extension cable or PC USB port; recommend using USB extension cable for easy to connect and disconnect to PC. We also recommend USB hub with on/off button.
3. Upon power applied to the board, the LED flashes blue color for 5 seconds and then turns off.
4. Wait for 5 seconds.
5. On Windows 10 machine, open Device Manager to check for the assign COM port.



*Figure 4: Checking COM Port with Device Manager*

Note: for Win10 system, the system device manager may not fully configure Qomu as COM port when install for the first time, disconnect the power from the module and repeat step (3) to (6).

6. Launch PuTTY application and configure for Serial access and select "Open".



*Figure 5: Putty setup for COM port access*

7. Perform the followings in the PuTTY terminal:
  - a) Type: “diag” to bring up Qomu diagnostic menu
  - b) Type: “help” to bring up the menu of commands
  - c) Type: “red”; the command shows red color for LED
  - d) Type: “red”; the LED turns off the LED
  - e) Try “green” and “blue”

```
COM35 - PuTTY

#####
Quicklogic Qomu LED Test
SW Version: qorc-sdk/qf_apps/qf_helloworldsw
Mar 22 2021 10:18:54
#####

Hello world!!

*****
Command Line Interface
App SW Version: qorc-sdk/qf_apps/qf_helloworldsw
*****
[0] > █
```

*Figure 6: Example of Hello World application menu*

```
COM35 - PuTTY

#####
Quicklogic Qomu LED Test
SW Version: qorc-sdk/qf_apps/qf_helloworldsw
Mar 22 2021 10:18:54
#####

Hello world!!

*****
Command Line Interface
App SW Version: qorc-sdk/qf_apps/qf_helloworldsw
*****
[0] > diag
[1] diag > red
[1] diag > red
[1] diag > green
[1] diag > green
[1] diag > blue
[1] diag > blue
[1] diag > █
```

*Figure 7: Example of diagnostic menu*

## Board Layout Overview

### Components on the Board

- QuickLogic EOS S3 MCU Platform
- 16Mbit of on-board flash memory
- RGB LED
- 4 User Touch Pads
- IO signals break-routed into test points

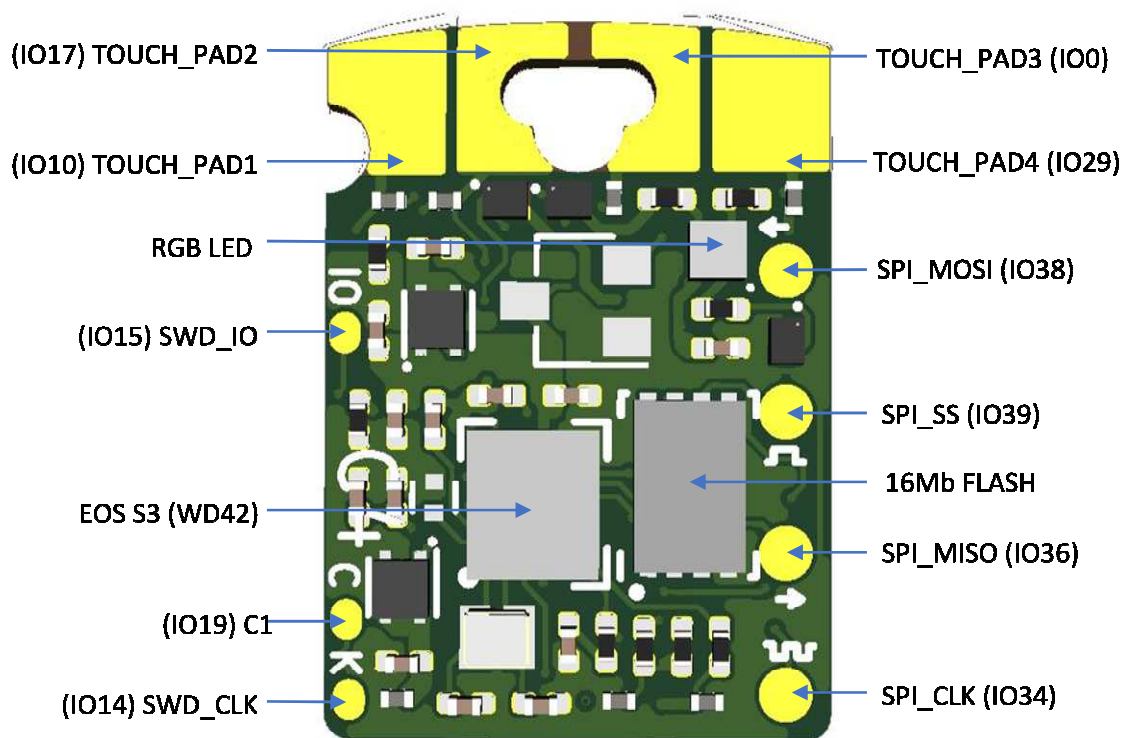


Figure 8: Qomu HDK Board Overview

## Qomu HDK power supplies

Qomu HDK power is supplied via USB pads

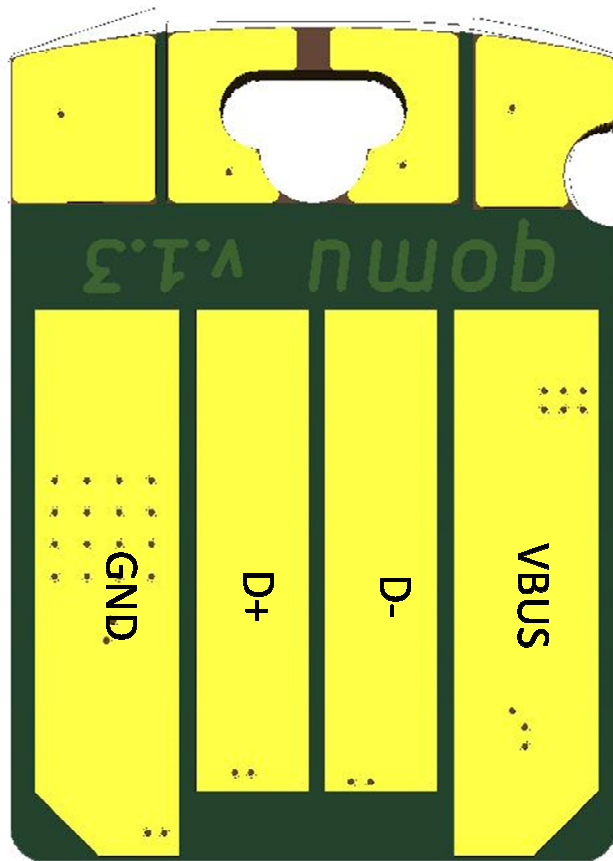


Figure 9: Qomu HDK Power input



## Flashing Binaries to Qomu HDK

Qomu supports loading and testing stand-alone eFPGA design or eFPGA + M4 MCU design, using the pre-programmed bootloader and flash-loading binaries. Recommend to only update standalone M4 application, M4+FPGA application or standalone FPGA application.

In each case, the procedure is the same.

**IMPORTANT: do not overwrite bootloader area and the bootfpga area. Overwrite these areas may make the Qomu HDK fail to boot properly and can't be recovered without specialize hardware.**

### Procedure

1. Remove USB power from the Qomu HDK.
2. Apply USB Power.
3. While the blue LED is blinking (for 5 seconds), touch the touch-pads with finger; if success, the green led will start flashing.
4. Check the COM port assigned COM value; see figure 3 for example of finding COM port in Windows 10 OS.
5. Use TinyFpgaProgrammer application to load the target application. Refer to <https://github.com/QuickLogic-Corp/TinyFPGA-Programmer-Application> for detail information on installation and program usage.

### Example (for Windows 10 OS)

- Flashing Qomu with standalone FPGA example, execute and reflash Qomu with the original HelloworldSW application. All required binaries are available under QORC-SDK: [https://github.com/QuickLogic-Corp/qorc-sdk/tree/master/qomu\\_apps/qomu-initial-binaries](https://github.com/QuickLogic-Corp/qorc-sdk/tree/master/qomu_apps/qomu-initial-binaries).
- Checkout TinyFPGA-Programmer-Application to local driver and follow instructions from the readme.rst.

### Flashing Qomu with standalone FPGA application

- Open CMD console.
- Change directory to location of TinyFPGA-Programmer-Application.
- Execute Procedure step (1) to (4).
- For step 5, type: " python.exe tinyfpga-programmer-gui.py --mode fpga --port [PC COMPORT] -appfpga [location of binaries]\qomu\_breath.bin --reset ".  
Note: tiny application writes the fpga breath application to flash; the application issues the system reset upon completion; the system blinks blue led for 5 seconds and load the breath application; after few seconds, the RGB led will shows breathing of various color. For more information with regard to the application operation, see [https://github.com/coolbreeze413/qorc-onion-apps/tree/master/qorc\\_fpga\\_breathe](https://github.com/coolbreeze413/qorc-onion-apps/tree/master/qorc_fpga_breathe).

### Flashing Qomu with HelloworldSW application

- Open CMD console.
- Change directory to location of TinyFPGA-Programmer-Application.
- Execute Procedure step (1) to (4).
- For step 5, type: " python.exe tinyfpga-programmer-gui.py --mode m4 --port [PC COMPORT] -m4app [location of binaries]\qomu\_helloworldsw.bin --reset ".

Note: see Running Pre-load Application section for details.

## Revision

Version	Date	Revision
1.0	March 2021	First release