# Keembay flashtool

## 1. Software Overview

Keembay flashtool enables user to recover Keembay(KMB) device. A KMB device boots to Firmware Recovery Boot path when:

* A platform Recovery jumper is ON
* No valid Capsule or boot FIPs found

It is a CLI program that runs on remotehost(host) and downloads the firmware image package(FIP) to the KMB device when the user issues the download command. More details on the download command and other usage can be found in the Quick Start section below.

The tool supports KMB device(s) directly connected to the host via one of the following interfaces:

* USB
* PCIe

## 2. Features

### 2.1. Supports USB and PCIE interfaces

The flashtool detects Keembay devices connected to the USB of the host. Then it downloads FIP to KMB device via USB using the standard FastBoot protocol. It uses the following open source [FastBoot](https://android.googlesource.com/platform/system/core/+/master/fastboot/) tool to provide this support.

The tool detects Keembay devices connected to the PCIE slot of the host. It delivers firmware recovery images to KMB device(s) via PCIe EP interface using XLink Userspace API. For details on XLink APIs, please refer to SAS\_xLink\_Communication\_1.0.pdf.

### 2.2. Checks Firmware Image Package

The flashtool checks the FIP layout is as per 'ARM Trusted Firmware' guideline before starting the download process. The checking is done using the open source fiptool which can be found [here.](https://github.com/ARM-software/arm-trusted-firmware/blob/v2.1/docs/firmware-design.rst#firmware-image-package-creation-tool)

Note: We will use this tool to check the layout of the package only. No signature verification is done.

However, user can turn off this feature using the --force option. See Quickstart for more details

### 2.3. User can specify retries and timeout

All the errors during the device detection and download process will be captured and handled properly.

This enables the user to specify the number of retries after any error using the --retries=<int> option. The default is 0(no retry). In the event of device is not found after the error, the tool will not retry despite the user specifying the number of retries.

User can also set a timeout duration(in seconds). This will kill the device download process after the timeout duration. The default is None.

### 2.4. Logging is built-in

The tool has logging feature enabled by default. User can customize it using config.json file. Please see Configuration section below for more details.

### 2.5. Supports configuration file

Apart from customizing logging, user can store default values for options like timeout. Please see Configuration section below for more details.

## 3. Product Scope

The flashtool is intended to recover KMB device that is unable to boot due to missing/corrupted FIP or has recovery jumper ON. This tool does not support Firmware Over-The-Air, or "FOTA" which requires a fully working and booting system first.

Supports locally connected KMB devices through USB and PCIE only. So, no network interface like ethernet/WiFi/other supported. Low level access to the flash(eMMC) using SPI or JTAG is not supported as well.

Supports firmware recovery only. OS recovery is not supported now. The term firmware here refers to a collection of BL2U, BL31, BL32 and BL33(Uboot) which is packaged into FIP.

## 4. System Requirement

* Linux Ubuntu 16.04 or greater with Sudo privilage
* Python 3.5 or greater with Pip
* PCIE recovery: System that supports more than 4GB of PCIe BAR

4.1. Below are the steps to increase BAR size in Uzel:

4.1.1 Power On Uzel without the device. Press Del to enter BIOS

4.1.2 Go to Chipset-->System Agent(SA) Configuration-->Above 4GB MMIO BIOS-->Enabled

4.1.3 Save and Exit. System will continue to boot. Power off.

4.1.4 Set Keembay EVM to PCIe recovery mode. Plug in and power On Uzel.

4.1.5 If Uzel boots, then the new settings are in place. We can continue with recovering the device.

## 5. Installation

5.1. Extract the release package and install the package using pip.

$ unzip -d flashtool\_YYYYmmdd/ flashtool\_YYYYmmdd.zip  
$ cd flashtool\_YYYYmmdd/  
$ pip3 install --user .

5.2. Once the installation has completed successfully, create a symbolic link to the launcher in /usr/bin/ so the launcher can be used with sudo.

$ sudo ln -s $(which flashtool) /usr/bin/flashtool

5.3. Once the installation has completed successfully, you can check the tool working by running

$ flashtool --help

5.4. For PCIe, driver component need to be loaded. To do that, inside the release package:

$ cd drivers  
$ sudo dpkg -i kmb-pcie-host-driver.deb  
$ sudo insmod /lib/modules/$(uname -r)/updates/dkms/mxlk.ko

## 6. Usage Guide

6.1. The device needs to be placed in 'recovery' mode.

6.1.1. Two ways the device will go into 'recovery' mode: - Auto recovery: There is no valid fip in flash. - Forced recovery: By using 'recovery jumper'

6.1.2. For Keembay EVM - USB: Set the BOOT DIP SW to eMMC Recovery boot and connect KMB via USB to a Linux host:

------------------------------  
BOOT DIP SW #: 1 2 3 4 5 6 7 8  
------------------------------  
 Value : 1 1 1 1 0 0 0 1  
------------------------------

* PCIE: Set the BOOT DIP SW to eMMC PCIe Recovery boot and connect KMB via PCIE to a Linux host and power on the host:
* ------------------------------  
  BOOT DIP SW #: 1 2 3 4 5 6 7 8  
  ------------------------------  
   Value : 1 1 1 1 0 1 0 1  
  ------------------------------

6.2. Keembay flashtool needs to be run with Sudo privilage. The commands are below.

$ flashtool --help  
Usage: flashtool [OPTIONS] COMMAND [ARGS]...  
  
Options:  
 --version Show the version and exit.  
 -h, --help Show this message and exit.  
  
Commands:  
 devices List available devices. Type "flashtool devices -h" for more  
 download Download binaries into device. Type "flashtool download -h" for more.

### 6.1. download

$ flashtool download --help  
Usage: flashtool download [OPTIONS] FIP  
  
 Download FIP into device  
  
Options:  
 -v, --verbose  
 -t, --timeout INTEGER Timeout in seconds.  
 -d, --device-type [usb|pcie] Filter devices based on type: USB or PCIe  
 -f, --force Skip FIP checking.  
 -h, --help Show this message and exit.

### 6.2. devices

$ flashtool devices --help  
Usage: flashtool devices [OPTIONS]  
  
 List available devices  
  
Options:  
 -v, --verbose  
 -d, --device-type [usb|pcie] Filter devices based on type: USB or PCIe  
 -h, --help Show this message and exit.

## 7. Configuration

The flashtool can be configured using a config.json file located in the <user home directory>/flashtool/ or current working directory. The config file can be used to configure:

### 7.1 App Configuration

Timeout, force and retries can be configured here so the user will not need to specify these options repeatedly. Example

"app": {  
 "timeout": 10,  
 "force": false,  
 "retries": 0  
 }

### 7.2 Logging configuration

flashtool's logging features can be configured here. For details on log configuration, please see [here](https://docs.python.org/2/howto/logging-cookbook.html#an-example-dictionary-based-configuration).

For frequent use case, two config file examples have been provided in the examples/ folder. In order to use them, you need to rename and copy them to ~/flashtool/.

$ mkdir ~/flashtool  
$ cp examples/config\_default.json ~/flashtool/config.json

The config\_full.json will provide both console and file logging. Additionally, console output will be formatted with color(success=green, error, critical, exception=red, warning=orange). The traceback of exception will not be printed to console but to the log file.

The config\_default.json will give you similar behavior as python default logger i.e. no log file and full error message(including traceback) printed to console.