How to configure TF-A BL2

Contents

1 Article purpose

This section details the TF-A BL2 stage (Trusted Firmware-A Boot Loader stage 2) used as FSBL (First Stage Boot Loader). It explains how to configure and build BL2 in STM32 MPU context, describes the build process from sources, and shows how to deploy it on your target.

2 Source code access and build process

2.1 Cross compilation

Cross compilation of TF-A BL2 is only required if it needs to be modified.
Refer to Setup Cross compile environment.

The build process creates an STM32 image. This binary is built in a single step during the build process.

2.2 Install sources

2.2.1 From the Developer package

The Developer Package contains OpenSTLinux and TF-A sources: TF-A Installation

2.2.2 Official source tree

Download the source code from the official Trusted Firmware-A git repository.

```
PC $>  git clone https://git.trustedfirmware.org/TF-A/trusted-firmware-a.git
```

Warning

The STM32MP1 platform is not yet fully upstreamed. Depending on the version used, some features may not be available.

For a full-featured software, go to STMicroelectronics github:

```
PC $>  git clone https://github.com/STMicroelectronics/arm-trusted-firmware.git
```

2.2.3 Distribution Package

It is possible to use the distribution package to download and rebuid TF-A BL2

2.3 Build process

2.3.1 TF-A Build flags

Here is the list of the mandatory flags that need to be specified to complete the TF-A BL2 build:

- ARM_ARCH_MAJOR = 7: the major version of Arm architecture to target (STM32MP1 is based on an Arm v7 architecture)
- ARCH = aarch32: specifies aarch32 architecture to be built
- PLAT = stm32mp1: builds an STM32MP1 platform
- DTB_FILE_NAME = <fdt file name>.dtb: this flag must be defined to build the proper target and include the correct DTB file into the final file
- The boot device(s) you use, one (or several) of:
  - STM32MP_EMMC = 1
  - STM32MP_SDMMC = 1
  - STM32MP_RAW_NAND = 1
  - STM32MP_SPI_NAND = 1
  - STM32MP_SPI_NOR = 1
- or a programming interface:
  - STM32MP_UART_PROGRAMMER = 1
  - STM32MP_USB_PROGRAMMER = 1

Optional flags:

- TRUSTED_BOARD_BOOT = 1: Enable Secure boot authentication

Warning

TRUSTED_BOARD_BOOT requires to have first downloaded The MBEDLTS[1] source code aligned version as specify in TF-A prerequisites [2].
MBEDTLS_DIR=<path_to_mbedtls_directory> must be used to compile.

- BUILD_PLAT = <folder>: custom output folder name (by default build/<debug/release>/)
- DEBUG = 1: adds debug information in all binaries
- V = 1: prints verbose compilation traces
- DYN_DISABLE_AUTH = 1: enables/disables authentication using device tree
- STM32MP_FORCE_MTD_START_OFFSET = <value>: overrides the default start offset to read FIP on MTD devices (need to be aligned with FlashLayout).

Information

Default offsets are STM32MP_NOR_FIP_OFFSET = 0x00080000 and STM32MP_NAND_FIP_OFFSET = 0x00200000

- STM32MP_USE_EXTERNAL_HEAP = 1: can be enabled to define MBEDTLS heap out of BL2 sources (defined in device tree).

For ecosystem release ≤ v3.0.0 compatibility

It is still possible to generate the a single BL2/BL32 file without FIP support, an option flag is available:

- STM32MP_USE_STM32IMAGE=1: Disable the FIP load and authentication management. Use the STM32 binary load and authentication used in ecosystem release ≤ v3.0.0 .

2.4 Build command

The BL2 generation depends on the selected storage device. By default, only one storage device is supported.

Warning

The DTB_FILE_NAME flag and the selected storage must be set to select the correct board configuration.

The device tree file for the target must be located in fdts folder (<board>.dts)

First add your own environment flags:

```
PC $>  unset LDFLAGS;
PC $>  unset CFLAGS;
```

Then compile the TF-A BL2.

The default build command for STM32MP15 is

```
PC $>  make ARM_ARCH_MAJOR=7 ARCH=aarch32 PLAT=stm32mp1 \
        <Selected storage> DTB_FILE_NAME=<board_name>.dtb
```

Here are build commands for the stm32mp157c-ev1 board (which supports different storage devices):

- Flash programming support

```
PC $>  make ARM_ARCH_MAJOR=7 ARCH=aarch32 PLAT=stm32mp1 STM32MP_USB_PROGRAMMER=1 \
        DTB_FILE_NAME=stm32mp157c-ev1.dtb
PC $>  make ARM_ARCH_MAJOR=7 ARCH=aarch32 PLAT=stm32mp1 STM32MP_UART_PROGRAMMER=1 \
        DTB_FILE_NAME=stm32mp157c-ev1.dtb
```

- Dedicated boot storage

```
PC $>  make ARM_ARCH_MAJOR=7 ARCH=aarch32 PLAT=stm32mp1 STM32MP_SDMMC=1 \
        DTB_FILE_NAME=stm32mp157c-ev1.dtb
```

```
PC $>  make ARM_ARCH_MAJOR=7 ARCH=aarch32 PLAT=stm32mp1 STM32MP_EMMC=1 \
        DTB_FILE_NAME=stm32mp157c-ev1.dtb
```

```
PC $>  make ARM_ARCH_MAJOR=7 ARCH=aarch32 PLAT=stm32mp1 STM32MP_RAW_NAND=1 \
        DTB_FILE_NAME=stm32mp157c-ev1.dtb
```

```
PC $>  make ARM_ARCH_MAJOR=7 ARCH=aarch32 PLAT=stm32mp1 STM32MP_SPI_NOR=1 \
        DTB_FILE_NAME=stm32mp157c-ev1.dtb
```

Not available on board

```
PC $>  make ARM_ARCH_MAJOR=7 ARCH=aarch32 PLAT=stm32mp1 STM32MP_SPI_NAND=1 \
        DTB_FILE_NAME=stm32mp157c-ev1.dtb
```

- Enable Secure boot on SDCard

```
PC $>  MBEDTLS_DIR=<path_to_mbedtls_directory> make ARM_ARCH_MAJOR=7 \
        ARCH=aarch32 PLAT=stm32mp1 TRUSTED_BOARD_BOOT=1 STM32MP_SDMMC=1 \
        DTB_FILE_NAME=stm32mp157c-ev1.dtb
```

2.5 Final image

The final image is available for Flash memory or SDCard update in the corresponding folder:

```
<BUILD_PLAT>/tf-a-<board>.stm32
Ex:
build/release/tf-a-stm32mp157c-ev1.stm32
```

3 Updating the software on board

3.1 Partitioning of binaries

The TF-A build provides a binary named tf-a-<board>.stm32 that MUST be copied to a dedicated partition named "fsblX" (X depends on the number of needed backups in the Flash).

Warning

TF-A must be located in the first partition of your boot device.

You can just update the first partition for a simple test, but all backup partitions must contain the same image at the end.

3.2 Updating via SDCard

If you use an SDCard, simply update TF-A using the dd command on your host.
Plug your SDCard into the computer and copy the binary to the dedicated partition; on an SDCard/USB disk the "fsbl1" partition is partition 1:

```
- SDCard: /dev/mmcblkXp1 (where X is the instance number)
- SDCard via USB reader: /dev/sdX1 (where X is the instance number)
```

- Under Linux®

```
PC $>  dd if=<tf-a file>.stm32 of=/dev/<device partition> bs=1M conv=fdatasync
```

Information

To find the partition associated to a specific label, just plug the SDCard/USB disk into your PC and call the following command:

```
 PC $>  ls -l /dev/disk/by-partlabel/
 total 0
 lrwxrwxrwx 1 root root 10 Jan 17 17:38 bootfs -> ../../mmcblk0p4
 lrwxrwxrwx 1 root root 10 Jan 17 17:38 fip -> ../../mmcblk0p3
 lrwxrwxrwx 1 root root 10 Jan 17 17:38 fsbl1 -> ../../mmcblk0p1          ? FSBL1 (T
F-A)
 lrwxrwxrwx 1 root root 10 Jan 17 17:38 fsbl2 -> ../../mmcblk0p2          ? FSBL2 (T
F-A backup ? same content as FSBL)
 lrwxrwxrwx 1 root root 10 Jan 17 17:38 rootfs -> ../../mmcblk0p5
 lrwxrwxrwx 1 root root 10 Jan 17 17:38 userfs -> ../../mmcblk0p6
```

- Under Windows®

CoreUtils [3] that includes the dd command is available for Windows.

3.3 Updating via USB mass storage on U-boot

See How to use USB mass storage in U-Boot.

Refer to the previous section to put tf-a-<board>.stm32 into SDCard/USB disk.

3.4 Updating your boot device via STM32CubeProgrammer

Refer to the STM32CubeProgrammer documentation to update your target.

4 Secure secret provisioning (SSP)

A specific TF-A BL2 build is required to manage SSP.

Warning

This STM32 output file must be signed to complete the SSP feature.

Dedicated files are delivered in the platform folder that contains the specific Makefile and bl2_plat_setup for the TF-A SSP.

The TF-A SSP is a subset of the standard TF-A BL2 that includes only:

- BL2 device tree
- BL2 image with limited support to the serial link device.

4.1 Additional flags

The mandatory flags to build the TF-A SSP are:

- **STM32MP_SSP = 1**

For the serial link (exclusive):

- STM32MP_UART_PROGRAMMER = 1
- STM32MP_USB_PROGRAMMER = 1

4.2 Build command

```
PC $>  make ARM_ARCH_MAJOR=7 ARCH=aarch32 PLAT=stm32mp1 \
        STM32MP_SSP=1 STM32MP_USB_PROGRAMMER=1 \
        DTB_FILE_NAME=<board>.dtb
```

or

```
PC $>  make ARM_ARCH_MAJOR=7 ARCH=aarch32 PLAT=stm32mp1 \
        STM32MP_SSP=1 STM32MP_UART_PROGRAMMER=1 \
        DTB_FILE_NAME=<board>.dtb
```

4.3 Final image

The final image is available in the corresponding folder:

```
tf-a-ssp-<board>.stm32
For example:
tf-a-ssp-stm32mp157c-ev1.stm32
```

- ↑ https://github.com/ARMmbed/mbedtls
- ↑ https://trustedfirmware-a.readthedocs.io/en/v2.4/getting_started/prerequisites.html
- ↑ http://gnuwin32.sourceforge.net/packages/coreutils.htm

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgment.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2023 STMicroelectronics – All rights reserved