TF-M Build System Update

TF-M Open Tech Forum

October 1st, 2020

Raef Coles

Interface changes

- Minor adjustments to command-line options
- Removal of configuration files
- Dependency handling

Command-line changes

- COMPILER -> CMAKE_TOOLCHAIN_FILE
  - -DCMAKE_TOOLCHAIN_FILE=<tfm root>/toolchain_ARMCLANG.cmake
  - Single centralized file for all compiler configuration
  - Available compilers == available toolchain files
  - Removal of most compiler-specific code from main codebase
  - Magic cmake variable
- TARGET_PLATFROM -> TFM_PLATFORM
  - -DTFM_PLATFORM=mps2/an521
  - Corresponds to directory -> no need to map platform name
  - Renamed due to argument change

Removal of configuration files

- Moved to composable options
  - ConfigRegression -> -DTEST_S=ON -DTEST_NS=ON
  - ConfigDefaultIPC -> -DTFM_PSA_API=ON
  - ConfigDefaultIPCTFMLVL2 -> -DTFM_PSA_API=ON –DTFM_ISOLATION_LVL=2
- Almost all configuration combinations are valid
- Can run NS and S regression tests separately
- Config/config_default.cmake is ultimate reference
- Works with ccmake / cmake-gui (in theory)
- -DTFM_EXTRA_CONFIG if you want to define your own config
- Config/profile and config/build_type

Dependency handling

- Automatic dependency downloading -> better first run UX
- Dependency path settings
  - -DMBEDCRYPO_PATH=<path to>/mbedtls
  - Detailed further at bottom of tfm_build_instructions.rst
- No more fixed dependency paths
  - Generated files are no longer checked into source
- Tradeoff in some cases
  - Problems with clean builds
  - Download deps into source tree?
  - Download deps into trusted_firmware_m/..

Under the hood changes

- Full dependency tree modelling -> incremental builds
- Automatic generation of generated files
- Full ninja support
- Using upstream cmake compiler support
  - Bumped cmake required version to 3.15
- Better integration support
- Modular dual-core support

Automatic generation of generated files

- Generated files removed from source tree
  - Secure scatter files / veneers etc.
  - <build_dir>/generated
- Generated when cmake is run
- Supports incremental rebuilding (patch in review)
- Makes it more obvious that you should be editing the template
- Has a couple of downsides
  - Need python and yaml to build TF-M
  - Template files sometime not that readable

Better integration support

- -DNS=OFF
- Builds secure world + psa api as static library
- Can link psa_api_ns from cmake
- Still some linkages with CMSIS
  - Only used when TFM_NS_CLIENT_IDENTIFICATION is on
  - Could (should?) be abstracted out
  - Because of this tf-m-tests repo is still required

Modular dual-cpu support

- Re-uses the preload.cmake files / _compiler_reload()
- Changes the CPU compilation flags in **directory** scope
  - Includes subdirectories
  - Partitions directory tree into Secure and NS code
- Changes applied to targets declared in directories
  - platform_ns declared in interface/CMakeLists.txt
- Not always ideal
  - test is NS but test/test_services is Secure again

Known issues

- A lot of documentation still needs updating (patch in progress)
  - Core test integration guide
- Issue with UART on NXP platform (being looked at)
- Install/export directory not generated (PR being reviewed)
- Core tests / IRQ tests not being run (patch in progress)
- Support for binary psa-arch-tests (design in progress)
- Some issue with sometimes lacking debug symbols
  - Possible workaround – MBEDCRYPTO_BUILD_TYPE=debug

Report an issue

- Dedicated ticket in Phabricator
  - https://developer.trustedfirmware.org/T834
- TF-M mailing list
- Create a patch

Big thanks

- Xinyu Zhang and Karl Zhang for CI
- Balint Matyi and Mate Toth-Pal for platform porting
- Thomas Törnblom for IAR
- Partners for platform porting and reviewing
- Anton Komlev for all the rest

The End

Thank you