1. Software-Anforderung für Trusted_Firmware_M über DS5 :

- **Betriebssystems:**
    - Ubuntu 16.04 x64
    - Ubuntu 18.04 x64
    - Windows 10 x64 + git-bash (MinGW)
    - Windows 10 x64 + Cygwin x64

- **Unterstützte C-Compiler:**
    - Arm Compiler v6.10
    - Arm Compiler v6.11
    - Arm Compiler v6.12
    - Arm Compiler v6.13
    - GNU Arm compiler v6.3.1
    - GNU Arm compiler v7.3

- **Unterstützte Cmake-Versionen oder GNUmake-versionen:**
    - 3.7
    - 3.10
    - 3.11
    - 3.12
    - 3.13
    - 3.14
    - GNU make v4.2.1 executable from Cygwin
    - GNU make v4.2.1 executable from msys2
    - GNU make v4.2 executable from DS5 v5.29.1 (see <DS5 directory>/bin)

- **Ubuntu Setup:**
    - DS-5 v5.29.1.(über arm-Webseite runterladen und mit „chmod +x <datei>“ und „./<datei>“ installieren)
        - https://developer.arm.com/tools-and-software/embedded/legacy-tools/ds-5-development-studio/downloads
    - Git tools v2.10.0 (sudo apt install git)
    - Git Large File Storage (git lfs install)
    - CMake (cmaek 3.14 runterladen und installieren)
    - GNU Make
    - Python3 and the pip package manager (sudo apt install python3 und sudo apt install python3-pip)
    - Python3 packages: *cryptography, pyasn1, yaml, jinja2 v2.10, cbor v1.0.0:*

```
(pip3 install --user cryptography pyasn1 pyyaml jinja2 cbor)
```

- **Eine Source-Datei erstellen wie folgt(mit source ./<Datei.sh> ausführen):**

Ein Konto bei der Developer.arm.com-webseite erstellen und bei DS5 bei License-manager eine 30tägige Lizenz erstellen

```
#set ARM Path and License
export PATH=/usr/local/DS-5_v5.29.1/sw/ARMCompiler6.10.1/bin:$PATH
export ARM_TOOL_VARIANT=ult
export ARM_PRODUCT_PATH=/usr/local/DS-5_v5.29.1/sw/mappings
export ARMLMD_LICENSE_FILE=/home/arsalan/.flexlmrc


#set cmake
export PATH=/home/arsalan/Downloads/cmake-3.17.3-Linux-x86_64/bin:$PATH


#set GNU make
export PATH=/usr/local/DS-5_v5.29.1/sw/gcc/bin:$PATH
```

2. Build Instructions :

- **Die Firmware herunterladen:**

```
cd <TF-M base folder>
git clone https://git.trustedfirmware.org/trusted-firmware-m.git
git clone https://github.com/ARMmbed/mbed-crypto.git -b mbedcrypto-3.0.1
git clone https://github.com/ARM-software/CMSIS_5.git -b 5.5.0
```

- **Build steps for the AN521 target platform:**

```
cd <TF-M base folder>
cd trusted-firmware-m
mkdir cmake_build
cd cmake_build
cmake ../ -G"Unix Makefiles" -DTARGET_PLATFORM=AN521 -DCOMPILER=ARMCLANG
cmake --build ./ -- install
```

- **Regression Test for the AN521 target platform:**

```
cd <TF-M base folder>
cd trusted-firmware-m
mkdir cmake_test
cd cmake_test
cmake -G"Unix Makefiles" -DPROJ_CONFIG=`readlink -f ../configs/ConfigRegression.cmake` -
DTARGET_PLATFORM=AN521 -DCOMPILER=ARMCLANG ../
cmake --build ./ -- install
```

- **Build for PSA Developer API Compliance tests:**

```
cd <TF-M base folder>
cd trusted-firmware-m
mkdir cmake_psa_test
cd cmake_psa_test
cmake -G"Unix Makefiles" -DPROJ_CONFIG=`readlink -f ../configs/ConfigPsaApiTest.cmake` -
DPSA_API_TEST_CRYPTO=ON -DTARGET_PLATFORM=AN521 -DCOMPILER=ARMCLANG ../
cmake --build ./ -- install
```

- **Build for PSA FF(IPC) Compliance tests:**

```
cd <TF-M base folder>
cd trusted-firmware-m
python3 tools/tfm_parse_manifest_list.py -m tools/tfm_psa_ff_test_manifest_list.yaml append
```

- **PSA FF Aktivieren für** ConfigPsaApiTestIPCTfmLevel2.cmake config**:**

```
cd <TF-M base folder>
cd trusted-firmware-m
mkdir cmake_psa_test
cd cmake_psa_test
```

```
cmake -G"Unix Makefiles" -DPROJ_CONFIG=`readlink -f
../configs/ConfigPsaApiTestIPCTfmLevel2.cmake` -DPSA_API_TEST_IPC=ON -DTARGET_PLATFORM=AN521 -
DCOMPILER=ARMCLANG ../
cmake --build ./ -- install
```

Der Rest wird in einer anderen Datei erklärt. Es handelt sich um DS5-DIE.

Quelle:
https://ci.trustedfirmware.org/job/tf-m-build-test-nightly/lastSuccessfulBuild/artifact/build-docs/tf-m_documents/install/doc/user_guide/html/index.html