Developer Environment Configure

2022年2月25日 2:15

STM32MP1 Developer Environment Configure

来自 <https://wiki.stmicroelectronics.cn/stm32mpu/wiki/STM32MP1_Developer_Package>

1. **PC Prerequisites**

   来自 <https://wiki.stmicroelectronics.cn/stm32mpu/wiki/PC_prerequisites#Installing_extra_packages>

   - **Packages required by OpenEmbedded/Yocto**

   ```
   $> sudo apt-get update
   $> sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib build-essential
   chrpath socat cpio python3 python3-pip python3-pexpect xz-utils debianutils iputils-ping python3-git
   python3-jinja2 libegl1-mesa libsdl1.2-dev pylint3 pylint xterm
   $> sudo apt-get install make xsltproc docbook-utils fop dblatex xmlto
   $> sudo apt-get install libmpc-dev libgmp-dev
   ```

   - **Packages needed for some "Developer Package" use cases**

   ```
   $> sudo apt-get install libncurses5 libncurses5-dev libncursesw5-dev libssl-dev linux-headers-generic
   u-boot-tools device-tree-compiler bison flex g++ libyaml-dev libmpc-dev libgmp-dev
   ```

2. **Installing the Starter Package**

   来自 <https://wiki.stmicroelectronics.cn/stm32mpu/wiki/STM32MP15_Discovery_kits_-_Starter_Package>

   - **Downloading the image**

   en.FLASH-stm32mp1-openstlinux-5-10-dunfell-mp1-21-11-17_tar_v3.1.0.xz

   ```
   $> tar -xvf en.FLASH-stm32mp1-openstlinux-5-10-dunfell-mp1-21-11-17_tar_v3.1.0.xz
   ```

   - **Create the Image**

   ```
   $> cd stm32mp1-openstlinux-5.10-dunfell-mp1-21-11-17/images/stm32mp1/scripts/
   $> ./create_sdcard_from_flashlayout.sh ../flashlayout_st-image-
   weston/optee/FlashLayout_sdcard_stm32mp157c-dk2-optee.tsv
   ```

   - **Image flashing**

   ```
   $> sudo dd if=../flashlayout_st-image-weston/extensible/../../FlashLayout_sdcard_stm32mp157c-dk2-
   optee.raw of=/dev/sdb bs=8M conv=fdatasync status=progress
   ```

3. **Installing the SDK**

   - **Download and install the STM32MP1 SDK**

   en.SDK-x86_64-stm32mp1-openstlinux-5.10-dunfell-mp1-21-11-17.tar_v3.1.0.xz

   ```
   $> tar -xvf en.SDK-x86_64-stm32mp1-openstlinux-5.10-dunfell-mp1-21-11-17.tar_v3.1.0.xz
   $> ./stm32mp1-openstlinux-5.10-dunfell-mp1-21-11-17/sdk/st-image-weston-openstlinux-weston-stm32mp1-
   x86_64-toolchain-3.1.11-openstlinux-5.10-dunfell-mp1-21-11-17.sh -d <STM32MP1 SDK PATH>
   ```

   - **Starting up the STM32MP1 SDK**

   ```
   $> source <STM32MP1 SDK PATH>/environment-setup-cortexa7t2hf-neon-vfpv4-ostl-linux-gnueabi
   ```

4. **Installing the Linux kernel**

   - **Downloading the Linux kernel**

   en.SOURCES-kernel-stm32mp1-openstlinux-5.10-dunfell-mp1-21-11-17.tar.xz

   ```
   $> tar -xvf en.SOURCES-kernel-stm32mp1-openstlinux-5.10-dunfell-mp1-21-11-17.tar.xz
   $> cd stm32mp1-openstlinux-5.10-dunfell-mp1-21-11-17/sources/arm-ostl-linux-gnueabi/linux-
   stm32mp-5.10.61-stm32mp-r2-r0/
   $> tar -xvf linux-5.10.61.tar.xz
   $> cd linux-5.10.61/
   $> for p in `ls -1 ../*.patch`; do patch -p1 < $p; done
   ```

   - **Configure the Linux kernel**

   ```
   <Linux kernel installation directory>/README.HOW_TO.txt helper file tells the instructions.
   $> mkdir -p ../build
   $> make ARCH=arm O="$PWD/../build" multi_v7_defconfig fragment*.config
   $> for f in `ls -1 ../fragment*.config`; do scripts/kconfig/merge_config.sh -m -r -O $PWD/../build
   $PWD/../build/.config $f; done
   $> yes '' | make ARCH=arm oldconfig O="$PWD/../build"
   ```

   - **Building the Linux kernel**

   ```
   $> cd <build directory>
   $> source <STM32MP1 SDK PATH>/environment-setup-cortexa7t2hf-neon-vfpv4-ostl-linux-gnueabi
   $> make ARCH=arm uImage vmlinux dtbs LOADADDR=0xC2000040
   $> make ARCH=arm modules
   $> make ARCH=arm INSTALL_MOD_PATH="$PWD/../build/install_artifact" modules_install
   ```

- **Deploying the Linux kernel**

```
Replace the kernel + devicetree in SDCARD.
$> sudo cp arch/arm/boot/uImage /media/$USER/bootfs
$> sudo cp arch/arm/boot/dts/stm32mp157*.dtb /media/$USER/bootfs
Replace the kernel modules in SDCARD.
$> rm install_artifact/lib/modules/5.10.61/build install_artifact/lib/modules/5.10.61/source
$> find install_artifact/ -name "*.ko" | xargs $STRIP --strip-debug --remove-section=.comment --
remove-section=.note --preserve-dates
$> sudo cp -r install_artifact/lib/modules/* /media/$USER/rootfs/lib/modules/
Regenerate the list of module dependencies and the list of symbols provided by modules.
Board $> depmod -a
Board $> sync
Board $> reboot
```

5. Installing the OP-TEE

- **Downloading the OP-TEE**

```
en.SOURCES-optee-stm32mp1-openstlinux-5.10-dunfell-mp1-21-11-17.tar.xz
$> tar -xvf en.SOURCES-optee-stm32mp1-openstlinux-5.10-dunfell-mp1-21-11-17.tar.xz
$> cd stm32mp1-openstlinux-5.10-dunfell-mp1-21-11-17/sources/arm-ostl-linux-gnueabi/optee-os-
stm32mp-3.12.0-stm32mp-r2-r0/
$> tar -xvf optee-os-stm32mp-3.12.0-stm32mp-r2-r0.tar.gz
$> cd optee-os-stm32mp-3.12.0-stm32mp-r2/
$> for p in `ls -1 ../*.patch`; do patch -p1 < $p; done
```

- **Building the OP-TEE**

```
<OP-TEE installation directory>/README.HOW_TO.txt helper file tells the instructions.
$> export FIP_DEPLOYDIR_ROOT=$PWD/../../FIP_artifacts
$> source <STM32MP1 SDK PATH>/environment-setup-cortexa7t2hf-neon-vfpv4-ostl-linux-gnueabi
$> make -f $PWD/../Makefile.sdk CFG_EMBED_DTB_SOURCE_FILE=stm32mp157c-dk2 all
The generated FIP images are available in $FIP_DEPLOYDIR_ROOT/fip
```

- **Updating the SDK**

```
$> cp -r $PWD/../build/stm32mp157c-dk2/export-ta_arm32/* <STM32MP1 SDK PATH>/sysroots/cortexa7t2hf-
neon-vfpv4-ostl-linux-gnueabi/usr/include/optee/export-user_ta
```

- **Deploying the OP-TEE**

```
Replace the fip-stm32mp157c-dk2-optee.bin and recreate the image.
$> cp $FIP_DEPLOYDIR_ROOT/fip/fip-stm32mp157c-dk2-optee.bin <STM32MP1 IMAGE PATH>/stm32mp1/fip
```

6. Installing the OPTEE-CLIENT

- **Downloading and Building the OPTEE-CLIENT**

```
Find optee_client-3.12.0.tar.gz of OPTEE-version 3.12.0 in optee_os/CHANGELOG.md.
$> tar -xvf optee_client-3.12.0.tar.gz
$> cd optee_client-3.12.0/
$> source <STM32MP1 SDK PATH>/environment-setup-cortexa7t2hf-neon-vfpv4-ostl-linux-gnueabi
$> make
```

- **Deploying the OPTEE-CLIENT**

```
Replace the tee-supplicant in board.
$> scp out/tee-supplicant/tee-supplicant root@<ip of board>:/usr/bin
```

OPTEE CA & TA Developer Environment Configure

- **CA Developer Environment Configure**

```
$> mkdir workspace
$> cd worksapce
$> mkdir frank_ca
$> vim frank_ca/Makefile
```

```
CC      ?= $(CROSS_COMPILE)gcc
LD      ?= $(CROSS_COMPILE)ld
AR      ?= $(CROSS_COMPILE)ar
NM      ?= $(CROSS_COMPILE)nm
OBJCOPY ?= $(CROSS_COMPILE)objcopy
OBJDUMP ?= $(CROSS_COMPILE)objdump
READELF ?= $(CROSS_COMPILE)readelf
```

```
OBJS = main.o

CFLAGS += -Wall -I../ta/include -I./include
#Add/link other required libraries here
LDADD += -lteec

BINARY = optee_example_frank_ca

.PHONY: all
all: $(BINARY)

$(BINARY): $(OBJS)
        $(CC) -o $@ $< $(LDADD)

.PHONY: clean
clean:
        rm -f $(OBJS) $(BINARY)

%.o: %.c
        $(CC) $(CFLAGS) -c $< -o $@
```

```
$> source <STM32MP1 SDK PATH>/environment-setup-cortexa7t2hf-neon-vfpv4-ostl-linux-gnueabi
$> make
```

- **TA Developer Environment Configure**

```
$> mkdir workspace
$> cd worksapce
$> mkdir frank_ta
$> vim frank_ta/Makefile
```

```
CFG_TEE_TA_LOG_LEVEL ?= 4


TA_DEV_KIT_DIR=$(shell echo $$SDKTARGETSYSROOT)/usr/include/optee/export-user_ta


# The UUID for the Trusted Application
BINARY=8aaaf200-2450-11e4-abe2-0002a5d5c51b


-include $(TA_DEV_KIT_DIR)/mk/ta_dev_kit.mk


ifeq ($(wildcard $(TA_DEV_KIT_DIR)/mk/ta_dev_kit.mk), )
clean:
        @echo 'Note: $$(TA_DEV_KIT_DIR)/mk/ta_dev_kit.mk not found, cannot clean TA'
        @echo 'Note: TA_DEV_KIT_DIR=$(TA_DEV_KIT_DIR)'
endif
```

```
$> source <STM32MP1 SDK PATH>/environment-setup-cortexa7t2hf-neon-vfpv4-ostl-linux-gnueabi
$> make
```