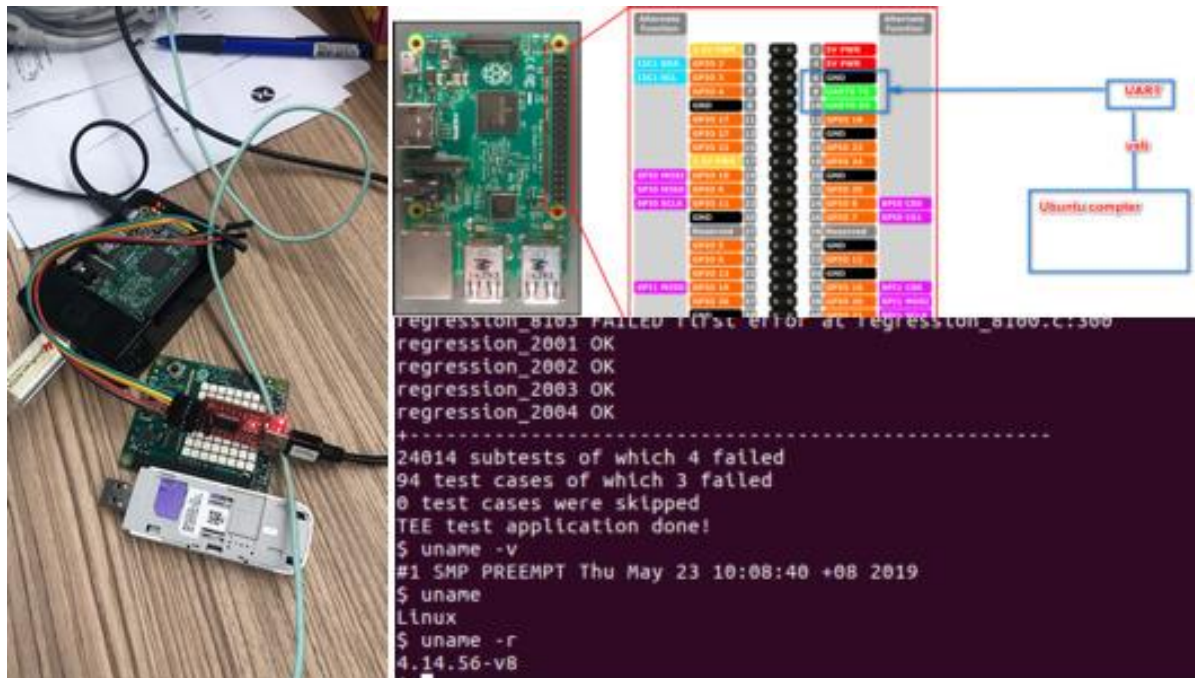


Steps to Set up Raspbain with OPTEE Raspberry PI mode3

Edit by LiuYuancheng 29/05/2019

As the OS OPTEE official web only provide the light Linux for Raspberry PI. If we follow the instruction in the <https://optee.readthedocs.io/building/devices/rpi3.html> (As shown below) and get to the step6, we can run the "xtest" but the Linux OS can only provide few function for further usage. This document will provide the detail steps about how to set the Raspberry PI mode 3's Raspbain system with OPTEE function.



Step 1: Prepare the Raspberry PI mode 3 with Raspbain system installed:

(If you have set the PI already, then skip the step1)

Dev Env: Windows 10/7

1.1 Insert the 16GB SD card in the windows machine and use "SD Memory card formatter" to format the SD card. Download the SD memory card formatter from <https://www.sdcard.org/downloads/formatter/> and follow all the default setting.

1.2 Down load the Raspberry PI Raspbain OS(32-bit) from <https://www.raspberrypi.org/downloads/raspbian/>

1.3 Download the FlashFlawless from <https://www.balena.io/etcher/> and flash the Raspbain image in to the SD card, put the SD card in Raspberry PI to double confirm the it works normally.

Step 2: Build the Raspbain with OPTEE enable

Dev Env: Ubuntu 16.04

2.1 Install the packages that need to be installed to start with to make OPTEE:

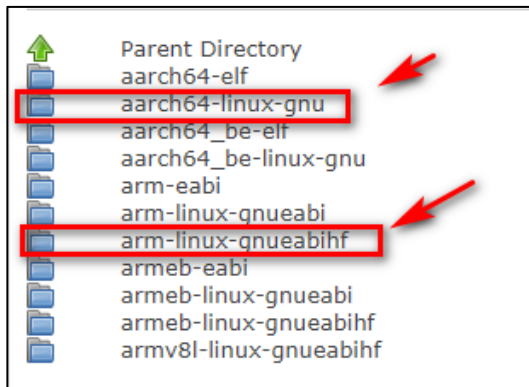
```
$ sudo apt-get install android-tools-adb android-tools-fastboot autoconf \
  automake bc bison build-essential cscope curl device-tree-compiler \
  expect flex ftp-upload gdisk iasl libattr1-dev libc6:i386 libcap-dev \
  libfdt-dev libftdi-dev libglb2.0-dev libhidapi-dev libncurses5-dev \
  libpixman-1-dev libssl-dev libstdc++6:i386 libtool libz1:i386 make \
  mtools netcat python-crypto python-serial python-wand unzip uuid-dev \
  xdg-utils xterm xz-utils zlib1g-dev
```

2.2 Download the Cross Build Toolchain:

AARCH64 & AARCH32 both needed, and AARCH32 must > 6.0 from linaro:

<https://releases.linaro.org/components/toolchain/binaries/>

Download these 2: "arm-linux-gnueabi" and "aarch64-linux-gnu" and set the path in the cofig.mk file:



Config.mk:

```
export CROSS_COMPILE :=  
/path/to/your/linaro/aarch32/bin/arm-linux-gnueabi-  
  
export CROSS_COMPILE_AARCH64 :=  
/path/to/your/linaro/aarch64/bin/aarch64-linux-gnu-
```

Install the build package: **sudo apt-install u-boot-tools**

Build and check the result:

```
$ ./prepare-env.sh # if you had download all packages, skip this.  
$ make patch # this will patch linux kernel & ATF, if you have done before,  
skip this.  
$ make
```

```
optee@optee-VirtualBox: ~/raspbain-tee/raspbian-tee-master/out  
optee@optee-VirtualBox:~/raspbain-tee/raspbian-tee-master$ cd out/  
optee@optee-VirtualBox:~/raspbain-tee/raspbian-tee-master/out$ ls  
boot  out.zip  rootfs  
optee@optee-VirtualBox:~/raspbain-tee/raspbian-tee-master/out$ tree -L 4  
.  
├── boot  
│   ├── armstub8.bin  
│   ├── bcm2710-rpi-3-b.dtb  
│   ├── bcm2710-rpi-3-b-plus.dtb  
│   ├── config.txt  
│   ├── uboot.env  
│   └── uImage  
├── out.zip  
└── rootfs  
    ├── bin  
    │   ├── optee_example_acipher  
    │   ├── optee_example_aes  
    │   ├── optee_example_hello_world  
    │   ├── optee_example_hotp  
    │   ├── optee_example_random  
    │   ├── optee_example_secure_storage  
    │   └── tee-supPLICANT  
    ├── lib  
    │   ├── libteec.so -> libteec.so.1  
    │   ├── libteec.so.1 -> libteec.so.1.0  
    │   ├── libteec.so.1.0  
    │   ├── modules  
    │   │   ├── 4.14.98-v7  
    │   │   └── optee_armtz  
    │   │       ├── 484d4143-2d53-4841-3120-4a6f636b6542.ta  
    │   │       ├── 5dbac793-f574-4871-8ad3-04331ec17f24.ta  
    │   │       ├── 8aaaf200-2450-11e4-abe2-0002a5d5c51b.ta  
    │   │       ├── a734eed9-d6a1-4244-aa50-7c99719e7b7b.ta  
    │   │       ├── b6c53aba-9669-4668-a7f2-205629d00f86.ta  
    │   │       └── f4e750bb-1437-4fbf-8785-8d3580c34994.ta
```

Step3: Install and test

3.1 Copy the file in the out to the SD card:

```
$ cp ./out/boot/* /media/user/boot  
$ sudo cp -r ./out/rootfs/* /media/user/rootfs
```

3.1 Put the SD card into the Raspberry PI and boot up. When you login, then:

```
$ ls /dev/tee*
/dev/tee0 /dev/teepriv0 # this prove tee driver & optee-os works.
$ sudo tee-supplciant &
$ sudo optee_example_hello_world
```

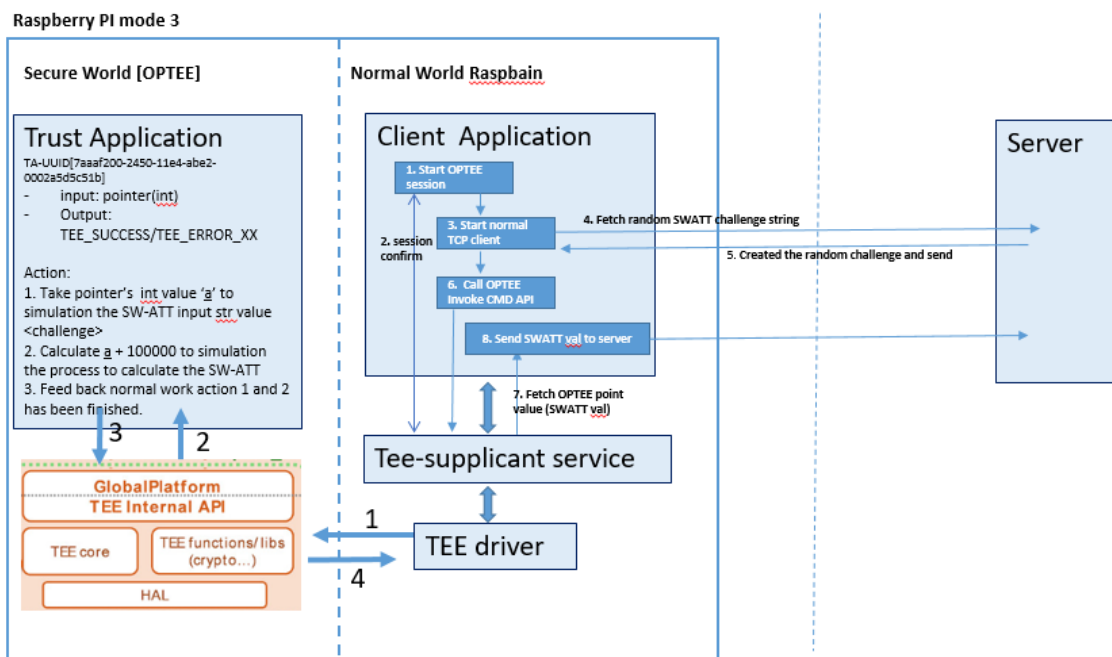
The result shows like this, which means the OPTEE has been set successfully:

```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo optee_example_hello_world
Invoking TA to increment 42
TA incremented value to 43
pi@raspberrypi:~ $ sudo optee_example_hello_world
Invoking TA to increment 42
TA incremented value to 43
pi@raspberrypi:~ $
```

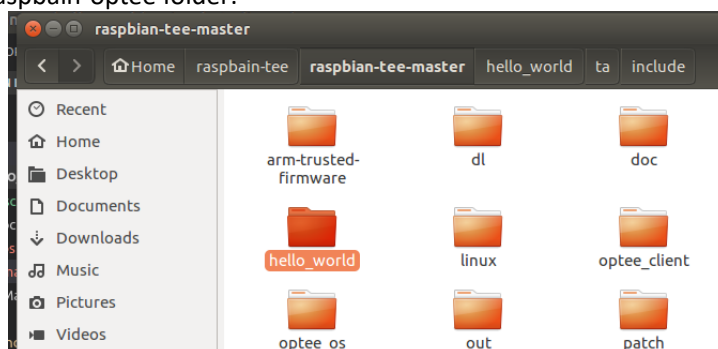
Step 4: Create a new Trust Application and run in the Raspbain with OPTEE

This is the trust client we are going to make:

OPTEE trust application [client <-> server] design



4.1 Down load the OPTEE trust example from https://github.com/linaro-swg/hello_world and put the folder in the raspbain-optee folder:



[optional]

After change the host\main.c and ta\hello_world_ta.c program We need to set the UUID in the file to make ta not conflict with the existed ta:

1. ta\include\hallo_world_ta.h
2. ta\Android.mk
3. ta\Makefile

4.2 Define the toolchains and environment variables with all 32bit setting and make:

```
export TEEC_EXPORT=$PWD/../optee_client/out/export
```

```
export HOST_CROSS_COMPILE=${The arm-linux-gnueabi position in <2.2> }/aarch32/bin/arm-linux-gnueabi-
```

```
export TA_CROSS_COMPILE=${The arm-linux-gnueabi position in <2.2> }/aarch32/bin/arm-linux-gnueabi-
```

```
export TA_DEV_KIT_DIR=$PWD/../optee_os/out/arm/export-ta_arm32
```

```
make
```

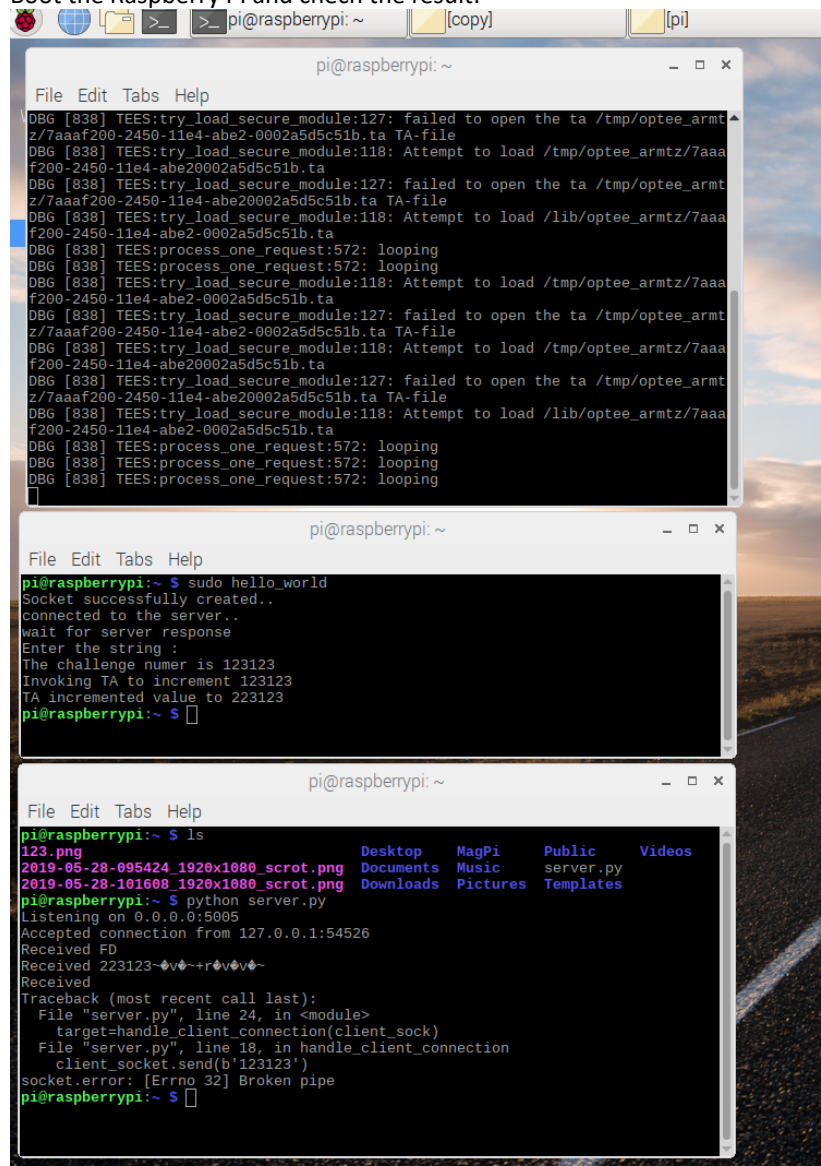
4.3 Copy the file to the system and test:

Copy the file:

If the make steps finished and success, insert the Raspbian SD card to the Ubuntu machine and

1. Copy the **host\hello_world** to **\media\user\rootfs\bin** folder
2. Copy the **ta\7aaaf200-2450-11e4-abe2-0002a5d5c51b.ta** to **\media\user\rootfs\lib\optee_armtz**

Boot the Raspberry PI and check the result:



Reference

<https://github.com/benhaz1024/raspbian-tee>

https://github.com/linaro-swg/hello_world