Lab-Developer package

* + **Work with "developer package"**
  + **Purpose of the Lab**

* + **Step1 : Extract and install the SDK**
  + **Step2 : Hello World application**
  + **Step3 : Extract and build Linux kernel**
  + **Step4 : Extract and build TF-A kernel**
  + **Step5 : Extract and build U-Boot kernel**
  + **Step6 : Extract and build OP-TEE kernel**
  + **Step7 : Add simple Linux application (led control on user button)**
  + **Step8 : Add simple Linux kernel module (dummy example : green led blinking)**
  + **Step9 : How to debug simple Linux application with gdb (optional)**
  + **Step10 : How to debug simple Linux kernel module with printk (optional)**

**Color convention**

***Blue rectangle: commands on Linux Host***

***Open Linux Terminal Window and copy/paste Linux Host command lines:***

|  |
| --- |
| ***Linux host command lines*** |

***Pink rectangle: commands on the board***

***Open Hyperterminal with following command line from Linux host***

|  |
| --- |
| ***minicom –D /dev/ttyACM0*** |

***Then copy/paste board command lines in Hyperterminal:***

|  |
| --- |
| ***Board command lines*** |

 You may need following copy after copy from linux host to board to force real copy

|  |
| --- |
| ***sync*** |

**Wiki user guide reference**

Getting Started

[Develop on Arm® Cortex®-A7](https://wiki.st.com/stm32mpu/index.php/Getting_started/STM32MP1_boards/STM32MP157C-DK2/Develop_on_Arm%C2%AE_Cortex%C2%AE-A7/Install_the_SDK)

Development zone

<https://wiki.st.com/stm32mpu/index.php/STM32MP1_Developer_Package>

* + **Step1 :STM32MP1 Developer Package: *Installing the SDK***
  + **wiki user guide**

Getting Started

[Install SDK](https://wiki.st.com/stm32mpu/index.php/Getting_started/STM32MP1_boards/STM32MP157C-DK2/Develop_on_Arm%C2%AE_Cortex%C2%AE-A7/Install_the_SDK)

Development zone

[**STM32MP1 Developer Package**](https://wiki.st.com/stm32mpu/index.php/STM32MP1_Developer_Package)

[STM32MP1 Developer Package - SDK](https://wiki.st.com/stm32mpu/index.php/STM32MP1_Developer_Package_-_SDK)

[Example of directory structure for Packages](https://wiki.st.com/stm32mpu/index.php/Example_of_directory_structure_for_Packages)

* + **Get developer package zip from wiki user guide**

Get [en.SDK-x86\_64-stm32mp1-openstlinux-4.19-thud-mp1-19-02-20.tar.xz](https://www.st.com/content/ccc/resource/technical/software/sw_development_suite/group0/32/5e/0d/c9/05/87/40/c0/stm32mp1dev_yocto_sdk/files/SDK-x86_64-stm32mp1-openstlinux-4.19-thud-mp1-19-02-20.tar.xz/jcr:content/translations/en.SDK-x86_64-stm32mp1-openstl)

Already copied in shared VM

* + **Extract the developer package**

|  |
| --- |
| gedit /home/osboxes/.bashrc &  add at end of file  **export SDK\_ROOT=/local/STM32MP15-Ecosystem-v1.0.0/Developer-Package**  before  “ export STENV\_NOGERRIT=yes  source $HOME/bin/stenv.sh” |

* + **Install the SDK (4min)**

|  |
| --- |
| Open a **new** terminal window to get SDK\_ROOT initialized  cd $SDK\_ROOT  tar xvf en.SDK-x86\_64-stm32mp1-openstlinux-4.19-thud-mp1-19-02-20.tar.xz  chmod +x stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sdk/st-image-weston-openstlinux-weston-stm32mp1-x86\_64-toolchain-2.6-openstlinux-4.19-thud-mp1-19-02-20.sh  ./stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sdk/st-image-weston-openstlinux-weston-stm32mp1-x86\_64-toolchain-2.6-openstlinux-4.19-thud-mp1-19-02-20.sh -d $SDK\_ROOT/SDK |

* + **Starting up the SDK environment variables**

|  |
| --- |
| source $SDK\_ROOT/SDK/environment-setup-cortexa7t2hf-neon-vfpv4-openstlinux\_weston-linux-gnueabi |

* + **Check environment variables**

|  |
| --- |
| echo $ARCH  arm  echo $CROSS\_COMPILE  arm-openstlinux\_weston-linux-gnueabi-  $CC --version  arm-openstlinux\_weston-linux-gnueabi-gcc (GCC) 8.2.0  echo $OECORE\_SDK\_VERSION  2.6-openstlinux-4.19-thud-mp1-19-02-20 |

**Step2 :STM32MP1 Developer Package: *Simple Hello World application***

* + - **wiki user guide**

Getting Started

[Create a simple hello world application](https://wiki.st.com/stm32mpu/index.php/Getting_started/STM32MP1_boards/STM32MP157C-DK2/Develop_on_Arm%C2%AE_Cortex%C2%AE-A7/Create_a_simple_hello-world_application)

* + - **header files for programming application**

The header files for programming applications are in the SDK in

**$SDK\_ROOT/SDK/sysroots/cortexa7t2hf-neon-vfpv4-openstlinux\_weston-linux-gnueabi/usr/include**

* + - **Application source code**

|  |
| --- |
| cd $SDK\_ROOT  tar xvf DeveloperPackageLab.tar |

* + - **Build source code**

|  |
| --- |
| cd $SDK\_ROOT/stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sources/gtk\_hello\_world\_example  make |

* + - **Test application**

|  |
| --- |
| scp gtk\_hello\_world [root@192.168.7.2:/usr/local](mailto:root@192.168.7.2:/usr/local) |

|  |
| --- |
| cd /usr/local/  ./gtk\_hello\_world |

**Step3 :STM32MP1 Developer Package: *Installing the Linux kernel***

* + - **wiki user guide**

Getting Started

[Modify, rebuild and reload the Linux® kernel](https://wiki.st.com/stm32mpu/index.php/Getting_started/STM32MP1_boards/STM32MP157C-DK2/Develop_on_Arm%C2%AE_Cortex%C2%AE-A7/Modify,_rebuild_and_reload_the_Linux%C2%AE_kernel)

Development Zone

[STM32MP1 Developer Package - Linux kernel](https://wiki.st.com/stm32mpu/index.php/STM32MP1_Developer_Package_-_Linux_kernel)

[File:Linux.README.HOW TO.txt](https://wiki.st.com/stm32mpu/index.php/File:Linux.README.HOW_TO.txt)

* + **Get the Linux kernel software package**

Get en.SOURCES-kernel-stm32mp1-openstlinux-4.19-thud-mp1-19-02-20.tar.xz

Already copied in shared VM

* + **Extract the kernel source**

|  |
| --- |
| cd $SDK\_ROOT/  tar xvf en.SOURCES-kernel-stm32mp1-openstlinux-4.19-thud-mp1-19-02-20.tar.xz  cd stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sources/arm-openstlinux\_weston-linux-gnueabi/linux-stm32mp-4.19-r0  tar xvf linux-4.19.9.tar.xz |

* + **Prepare and configure kernel source from** [File:Linux.README.HOW TO.txt](https://wiki.st.com/stm32mpu/index.php/File:Linux.README.HOW_TO.txt)

|  |
| --- |
| source $SDK\_ROOT/SDK/environment-setup-cortexa7t2hf-neon-vfpv4-openstlinux\_weston-linux-gnueabi  cd linux-4.19.9/  for p in `ls -1 ../\*.patch`; do patch -p1 < $p; done  make multi\_v7\_defconfig fragment\*.config  for f in `ls -1 ../fragment\*.config`; do scripts/kconfig/merge\_config.sh -m -r .config $f; done  yes '' | make oldconfig |

* + **Compile kernel source from README and kernel modules (25min)**

|  |
| --- |
| make uImage vmlinux dtbs LOADADDR=0xC2000040  (or  make –j 2 uImage vmlinux dtbs LOADADDR=0xC2000040)  make modules  mkdir -p $PWD/install\_artifact/  make INSTALL\_MOD\_PATH="$PWD/install\_artifact" modules\_install |

* + Install on target uImage (kernel image) and device tree

|  |
| --- |
| scp arch/arm/boot/uImage [root@192.168.7.2:/boot](mailto:root@192.168.7.2:/boot)  scp arch/arm/boot/dts/stm32mp157\*.dtb root@192.168.7.2:/boot |

* + Install on target modules

|  |
| --- |
| rm install\_artifact/lib/modules/4.19.9/build install\_artifact/lib/modules/4.19.9/source  scp -r install\_artifact/lib/modules/\* root@192.168.7.2:/lib/modules |
| /sbin/depmod –a  sync  reboot |

**Step4 : STM32MP1 Developer Package *Installing TF-A***

* + - **wiki user guide**

Development Zone

[STM32MP1 Developer Package - TF-A](https://wiki.st.com/stm32mpu/index.php/STM32MP1_Developer_Package_-_TF-A)

[File:TF-A.README.HOW TO.txt](https://wiki.st.com/stm32mpu/index.php/File:TF-A.README.HOW_TO.txt)

* + **Get the TF-A software package**

Get en.SOURCES-tf-a-stm32mp1-openstlinux-4.19-thud-mp1-19-02-20.tar.xz

Already done in shared VM

* + **Extract the source**

|  |
| --- |
| cd $SDK\_ROOT/  tar xvf en.SOURCES-tf-a-stm32mp1-openstlinux-4.19-thud-mp1-19-02-20.tar.xz  cd stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sources/arm-openstlinux\_weston-linux-gnueabi/tf-a-stm32mp-2.0-r0/  tar xvf v2.0.tar.gz |

* + **Prepare and configure** TF-A **source from** [File:TF-A.README.HOW TO.txt](https://wiki.st.com/stm32mpu/index.php/File:TF-A.README.HOW_TO.txt)

|  |
| --- |
| cd arm-trusted-firmware-2.0/    for p in `ls -1 ../\*.patch`; do patch -p1 < $p; done |

* + **Compile TF-A source from README**

|  |
| --- |
| source $SDK\_ROOT/SDK/environment-setup-cortexa7t2hf-neon-vfpv4-openstlinux\_weston-linux-gnueabi  make -f $PWD/../Makefile.sdk all |

* + **Install TF-A binarie (trusted boot chain only) from README**

**Plug usb card reader or use usb mass storage (ums 0 mmc 0 in uboot)**

|  |
| --- |
| ls -l /dev/disk/by-partlabel/  example :  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 bootfs -> ../../sdb4**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 fsbl1 -> ../../sdb1**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 fsbl2 -> ../../sdb2**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 rootfs -> ../../sdb6**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 ssbl -> ../../sdb3**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 userfs -> ../../sdb7**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 vendorfs -> ../../sdb5**  **dd if=<tf-a binary> of=/dev/<device partition> bs=1M conv=fdatasync**  where  <tf-a binary> is tf-a-\*.stm32  <device partition> is fsbl1 partition |

**Step5 : STM32MP1 Developer Package *Installing U-Boot***

* + - **wiki user guide**

Development Zone

[STM32MP1 Developer Package - U-Boot](https://wiki.st.com/stm32mpu/index.php/STM32MP1_Developer_Package_-_U-Boot)

[File:U-Boot.README.HOW TO.txt](https://wiki.st.com/stm32mpu/index.php/File:U-Boot.README.HOW_TO.txt)

* + **Get the U-Boot software package**

Get en.SOURCES-u-boot-stm32mp1-openstlinux-4.19-thud-mp1-19-02-20.tar.xz

Already done in shared VM

* + **Extract the source**

|  |
| --- |
| cd $SDK\_ROOT/  tar xvf en.SOURCES-u-boot-stm32mp1-openstlinux-4.19-thud-mp1-19-02-20.tar.xz  cd stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sources/arm-openstlinux\_weston-linux-gnueabi/u-boot-stm32mp-2018.11-r0/  tar xvf v2018.11.tar.gz |

* + **Prepare and configure** U-Boot **source from** [File:U-Boot.README.HOW TO.txt](https://wiki.st.com/stm32mpu/index.php/File:U-Boot.README.HOW_TO.txt)

|  |
| --- |
| cd u-boot-2018.11/    for p in `ls -1 ../\*.patch`; do patch -p1 < $p; done |

* + **Compile U-Boot source from README**

|  |
| --- |
| source $SDK\_ROOT/SDK/environment-setup-cortexa7t2hf-neon-vfpv4-openstlinux\_weston-linux-gnueabi  make -f $PWD/../Makefile.sdk all |

* + **Install Uboot basic boot chain rom README**

**Plug usb card reader or use usb mass storage (ums 0 mmc 0 in uboot)**

|  |
| --- |
| ls -l /dev/disk/by-partlabel/  example :  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 bootfs -> ../../sdb4**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 fsbl1 -> ../../sdb1**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 fsbl2 -> ../../sdb2**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 rootfs -> ../../sdb6**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 ssbl -> ../../sdb3**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 userfs -> ../../sdb7**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 vendorfs -> ../../sdb5**  **dd if=<uboot\_spl binary> of=/dev/<device partition> bs=1M conv=fdatasync**  where  <uboot\_spl binary> is u-boot-spl.stm32-\*  <device partition> is fsbl1 partition for instance sdb1  **dd if=<uboot binary> of=/dev/<device partition> bs=1M conv=fdatasync**  where  <uboot binary> is u-boot\*.img  <device partition> is ssbl partition for example **sdb3** |

* + **Install Uboot trusted boot chain rom README**

**Plug usb card reader or use usb mass storage (ums 0 mmc 0 in uboot)**

|  |
| --- |
| ls -l /dev/disk/by-partlabel/  example :  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 bootfs -> ../../sdb4**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 fsbl1 -> ../../sdb1**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 fsbl2 -> ../../sdb2**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 rootfs -> ../../sdb6**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 ssbl -> ../../sdb3**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 userfs -> ../../sdb7**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 vendorfs -> ../../sdb5**  fsbl1 partition is for tfa    **dd if=<uboot binary> of=/dev/<device partition> bs=1M conv=fdatasync**  where  <uboot binary> is u-boot\*.img  <device partition> is ssbl partition for example **sdb3** |

* + **Step6 : STM32MP1 Developer Package *Installing OP-TEE***
    - **wiki user guide**

Development Zone

[STM32MP1 Developer Package - OP-TEE](https://wiki.st.com/stm32mpu/index.php/STM32MP1_Developer_Package_-_OP-TEE)

[File:OP-TEE.README.HOW TO.txt](https://wiki.st.com/stm32mpu/index.php/File:OP-TEE.README.HOW_TO.txt)

* + **Get the OP-TEE software package**

Get en.SOURCES-optee-stm32mp1-openstlinux-4.19-thud-mp1-19-02-20.tar.xz

Already done in shared VM

* + **Extract the source**

|  |
| --- |
| cd $SDK\_ROOT/  tar xvf en.SOURCES-optee-stm32mp1-openstlinux-4.19-thud-mp1-19-02-20.tar.xz  cd stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sources/arm-openstlinux\_weston-linux-gnueabi/optee-os-stm32mp-3.3.0-r0/  tar xvf 3.3.0.tar.gz |

* + **Prepare and configure** OP-TEE **source from** [File:OP-TEE.README.HOW TO.txt](https://wiki.st.com/stm32mpu/index.php/File:OP-TEE.README.HOW_TO.txt)

|  |
| --- |
| cd optee\_os-3.3.0/    for p in `ls -1 ../\*.patch`; do patch -p1 < $p; done |

* + **Compile OP-TEE source from README**

|  |
| --- |
| source $SDK\_ROOT/SDK/environment-setup-cortexa7t2hf-neon-vfpv4-openstlinux\_weston-linux-gnueabi  chmod 755 scripts/bin\_to\_c.py  make -f $PWD/../Makefile.sdk all |

* + **Install opTee binaries from README**

**Plug usb card reader or use usb mass storage (ums 0 mmc 0 in uboot)**

|  |
| --- |
| ls -l /dev/disk/by-partlabel/  example :  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 bootfs -> ../../sdb4**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 fsbl1 -> ../../sdb1**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 fsbl2 -> ../../sdb2**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 rootfs -> ../../sdb6**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 ssbl -> ../../sdb3**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 userfs -> ../../sdb7**  **lrwxrwxrwx 1 root root 10 Mar 15 07:34 vendorfs -> ../../sdb5**  **dd if=<teeh binary> of=/dev/<device partition> bs=1M conv=fdatasync**  where  <**teeh bina**ry> is tee-header-\*-optee.stm32  <device partition> is **bootfs** partition example **sdb4**  **dd if=<teed binary> of=/dev/<device partition> bs=1M conv=fdatasync**  where  <**teed** binary> is tee-pageable-\*-optee.stm32  <device partition> is **rootfs** partition example **sdb6**  **dd if=<teex binary> of=/dev/<device partition> bs=1M conv=fdatasync**  where  <teex binary> is tee-pager-\*-optee.stm32  <device partition> is **userfs** partition **example sdb7** |

* + **Step7 : STM32MP1 Developer Package *Build Linux application***

* + **Get Linux hands on sources from DeveloperPackageLab.tar (if not already done)**

**(more details in STM32MP1 hands on OpenSTLinux-v1.2.pdf)**

|  |
| --- |
| cd $SDK\_ROOT  tar xvf DeveloperPackageLab.tar |

* + - **Build source code**

|  |
| --- |
| cd $SDK\_ROOT/stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sources/st-hands-on/openstlinux-distribution/openstlinux-hands-appli  source $SDK\_ROOT/SDK/environment-setup-cortexa7t2hf-neon-vfpv4-openstlinux\_weston-linux-gnueabi  make  clean  make  make install |

* + **Copy binary to target**

|  |
| --- |
| scp -r ./kernel\_install\_dir/\* [root@192.168.7.2:/](mailto:root@192.168.7.2:/) |

* + **Test application**

on target

|  |
| --- |
| cd /usr/local/bin    ./openstlinux-hands |

check led is working on user button

* + **Step8 : STM32MP1 Developer Package *Build Linux kernel module***

* + - **Wiki article : How\_to\_control\_a\_GPIO\_in\_kernel\_space**

<https://wiki.st.com/stm32mpu/index.php/How_to_control_a_GPIO_in_kernel_space>

* + **Build kernel module**

|  |
| --- |
| cd $SDK\_ROOT/stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sources/st-hands-on/openstlinux-distribution/openstlinux-hands-driver  source $SDK\_ROOT/SDK/environment-setup-cortexa7t2hf-neon-vfpv4-openstlinux\_weston-linux-gnueabi    export KERNEL\_SRC\_PATH=$SDK\_ROOT/stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sources/arm-openstlinux\_weston-linux-gnueabi/linux-stm32mp-4.19-r0/linux-4.19.9  make clean  make |

* + **Build device tree (dtb)**

Copy first new dts file for board (observe changes)

|  |
| --- |
| cd $SDK\_ROOT/stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sources/st-hands-on/openstlinux-distribution/openstlinux-hands-driver  diff stm32mp157c-dk2.dts.st-hands-on archive/stm32mp157c-dk2.dts.origin  cp $SDK\_ROOT/stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sources/st-hands-on/openstlinux-distribution/openstlinux-hands-driver/stm32mp157c-dk2.dts.st-hands-on $SDK\_ROOT/stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sources/arm-openstlinux\_weston-linux-gnueabi/linux-stm32mp-4.19-r0/linux-4.19.9/arch/arm/boot/dts/stm32mp157c-dk2.dts    cd $SDK\_ROOT/stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sources/arm-openstlinux\_weston-linux-gnueabi/linux-stm32mp-4.19-r0/linux-4.19.9/  source $SDK\_ROOT/SDK/environment-setup-cortexa7t2hf-neon-vfpv4-openstlinux\_weston-linux-gnueabi    make ARCH=arm dtbs |

**Copy binaries to target**

|  |
| --- |
| cd $SDK\_ROOT/stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sources/arm-openstlinux\_weston-linux-gnueabi/linux-stm32mp-4.19-r0/linux-4.19.9/arch/arm/boot/dts/  scp stm32mp157c-dk2.dtb [root@192.168.7.2:/boot/](mailto:root@192.168.7.2:/boot/)  cd $SDK\_ROOT/stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sources/st-hands-on/openstlinux-distribution/openstlinux-hands-driver  scp openstlinux-hands-driver.ko root@192.168.7.2:/lib/modules/ |

* + **Test kernel driver**

reset the board (to use updated device tree and kernel module )

on target

|  |
| --- |
| **cd /lib/modules**    **insmod openstlinux-hands-driver.ko**   check the module in installed  **lsmod** |

Check led is blinking during 10 seconds

|  |
| --- |
| **rmmod openstlinux-hands-driver.ko** |

* + **Step9 : STM32MP1 Developer Package *Debug Linux application with gdb (optional)***

* + **Build application with debug option**

|  |
| --- |
| cd $SDK\_ROOT/stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sources/st-hands-on/openstlinux-distribution/openstlinux-hands-appli    source $SDK\_ROOT/SDK/environment-setup-cortexa7t2hf-neon-vfpv4-openstlinux\_weston-linux-gnueabi    **In Makefile**  # Add / change option in CFLAGS if needed  **CFLAGS += -g**  **make clean**  **make**  **make install** |

* + **Copy binary to target**

|  |
| --- |
| scp -r ./kernel\_install\_dir/\* root@192.168.7.2:/ |

* + **Setup gdb**

**Establish usb ethernet connection**

on target

|  |
| --- |
| cd /usr/local/bin    gdbserver host:1234 openstlinux-hands |

On linux PC

|  |
| --- |
| arm-openstlinux\_weston-linux-gnueabi-gdb  (gdb)  **(gdb) target remote 192.168.7.2:1234** |

* + **Step10 : STM32MP1 Developer Package *Debug Linux kernel module with printk (optional)***

* + **Build kernel module adding prink message**

|  |
| --- |
| cd $SDK\_ROOT/stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sources/st-hands-on/openstlinux-distribution/openstlinux-hands-driver  source $SDK\_ROOT/SDK/environment-setup-cortexa7t2hf-neon-vfpv4-openstlinux\_weston-linux-gnueabi    export KERNEL\_SRC\_PATH=$SDK\_ROOT/stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sources/arm-openstlinux\_weston-linux-gnueabi/linux-stm32mp-4.19-r0/linux-4.19.9    Modify in openstlinux-hands-driver.c printk message  printk("GPIO example exit myName \n");  make clean  make |

* + **Copy binary to target**

|  |
| --- |
| cd $SDK\_ROOT/stm32mp1-openstlinux-4.19-thud-mp1-19-02-20/sources/st-hands-on/openstlinux-distribution/openstlinux-hands-driver  scp -r openstlinux-hands-driver.ko root@192.168.7.2:/lib/modules/ |

* + **Test new binary**

on target

|  |
| --- |
| **cd /lib/modules/**    **insmod openstlinux-hands-driver.ko** |

C:\E998E085\D2515B86-A96D-4A4C-97E9-F3917523806C_files\image001.pngC:\E998E085\D2515B86-A96D-4A4C-97E9-F3917523806C_files\image002.png

|  |
| --- |
| check new message is displayed  some kernel logs are also available in  **dmesg** |

C:\E998E085\D2515B86-A96D-4A4C-97E9-F3917523806C_files\image003.png