Lab-Distribution Package

* + **Work with distribution package**

* + **Purpose of the Lab**
    - **Step1 : First full build distribution package**
    - **Step2 : Add simple Linux application (led on user button) with devtool**

**Create recipe**

**Add new layer to distribution with simple Linux application**

**Rebuild weston image and flash the image**

**Test the new image with Linux application**

* + - **Step3 How to modify a component in Yocto ((optional)**
      * **How to modify an application**
      * **How to modify kernel**
      * **How to modify M4 fw**
      * **Test co-processing example**
  + **Overview of STM32mp1 embedded software distribution environment**
    - **Document links**
    - **Bitbake commands**
    - **Directory structures**
  + **Wiki user guide reference**

[**STM32MP1 Distribution Package**](https://wiki.st.com/stm32mpu/index.php/STM32MP1_Distribution_Package)

[STM32MP1 Distribution Package - OpenSTLinux distribution](https://wiki.st.com/stm32mpu/index.php/STM32MP1_Distribution_Package_-_OpenSTLinux_distribution)

[How\_to\_create\_a\_new\_open\_embedded\_layer](https://wiki.st.com/stm32mpu/index.php/How_to_create_a_new_open_embedded_layer)

OpenEmbedded[\_-\_](https://wiki.st.com/stm32mpu/index.php/OpenEmbedded_-_devtool)devtool

[How\_to\_customize\_the\_Linux\_kernel](https://wiki.st.com/stm32mpu/index.php/How_to_customize_the_Linux_kernel)

[How\_to\_create\_your\_own\_machine](https://wiki.st.com/stm32mpu/index.php/How_to_create_your_own_machine)

How to Integrate OpenEmbbded layer to ST Distribution example:

[How\_to\_install\_JAVA\_JDK](https://wiki.st.com/stm32mpu/index.php/How_to_install_JAVA_JDK)

(internal http://intranet.lme.st.com:8000/php-bin/ug\_mcdmpu/index.php/How\_to\_install\_JAVA\_JDK)

[BitBake\_cheat\_sheet](https://wiki.st.com/stm32mpu/index.php/BitBake_cheat_sheet)

* + **Step1 : STM32MP1 Distribution Package First Build**

* + **Set the environment variable for purpose of this hands'on**

|  |
| --- |
| gedit /home/osboxes/.bashrc &  add at end of file  **export SDK\_ROOT=/local/STM32MP15-Ecosystem-v1.0.0/Developer-Package**  **export DISTRI\_ROOT=/local/STM32MP15-Ecosystem-v1.0.0/Distribution-Package**  **export DISTRI\_RELEASE=openstlinux-4.19-thud-mp1-19-02-20**  **export WORK\_DIR=$DISTRI\_ROOT/$DISTRI\_RELEASE/workdir**  **export DISTRI\_BUILD\_PATH=$DISTRI\_ROOT/$DISTRI\_RELEASE/build-openstlinuxweston-stm32mp1**  **export ST\_HANDS\_ON\_APPLI=st-hands-on/openstlinux-distribution/openstlinux-hands-appli**  **export ST\_RECIPES=meta-open-hands-layer/recipes-openstlinux-hands/openstlinux-hands/**  **export WORK\_RECIPES=$DISTRI\_BUILD\_PATH/workspace/recipes/**  **export FLASH\_IMAGE\_TRUSTED=flashlayout\_st-image-weston/FlashLayout\_sdcard\_stm32mp157c-dk2-trusted.tsv**  before  export STENV\_NOGERRIT=yes  source $HOME/bin/stenv.sh |
| Open new terminal window  cd $DISTRI\_ROOT/openstlinux-4.19-thud-mp1-19-02-20 |
|  |

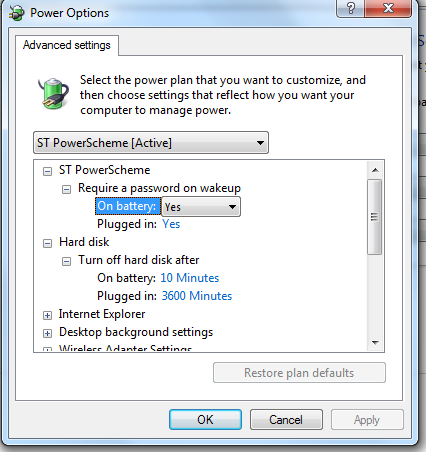
* + **Get the environment from manifest**

|  |
| --- |
| **repo init -u** [**https://github.com/STMicroelectronics/oe-manifest.git -b refs/tags/openstlinux-4.19-thud-mp1-19-02-20**](https://github.com/STMicroelectronics/oe-manifest.git%20-b%20refs/tags/openstlinux-4.19-thud-mp1-19-02-20) |
| **repo sync** |

* + **Build environment with bitbake**

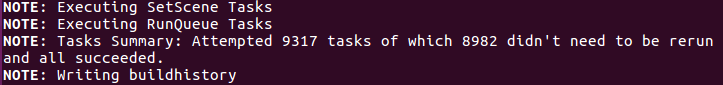
 Pre-Requisite before building using Virtual Machine:

check power options for Hard disk : **Turn off hard disk** **after** more than 20 hours as in below example:



|  |
| --- |
| DISTRO=openstlinux-weston MACHINE=stm32mp1 source layers/meta-st/scripts/envsetup.sh |
| bitbake st-image-weston |

You can check you have following log with “all suceeded” at end of build:



You can also go in directory

$DISTRI\_BUILD\_PATH/tmp-glibc/deploy/images/stm32mp1

And check images have been generated :

Example :

tf-a-stm32mp157c-dk2-trusted.stm32

u-boot-stm32mp157c-dk2-trusted.elf

st-image-bootfs-openstlinux-weston-stm32mp1.ext4

st-image-userfs-openstlinux-weston-stm32mp1.ext4

st-image-vendorfs-openstlinux-weston-stm32mp1.ext4

st-image-weston-openstlinux-weston-stm32mp1.ext4

* + **Step2 : STM32MP1 Distribution Package Rebuild application**

**Step 2.1 Create recipe**

* + **Copy hands-on to working directory**

|  |
| --- |
| **cd $DISTRI\_ROOT/$DISTRI\_RELEASE**  **DISTRO=openstlinux-weston MACHINE=stm32mp1 source layers/meta-st/scripts/envsetup.sh** |
| **mkdir $WORK\_DIR**  **cd $WORK\_DIR** |
| **cp -a /mnt/hgfs/shared/st-hands-on/ .** |

* + **Add recipe to openstlinux-hands-appli example**

|  |
| --- |
| **devtool add openstlinux-hands $ST\_HANDS\_ON\_APPLI** |

* + **Build new recipe for openstlinux-hands-appli**

|  |
| --- |
| **devtool build openstlinux-hands** |

|  |
| --- |
| Binaries are built in  $**DISTRI\_BUILD\_PATH/tmp-glibc/**work/cortexa7hf-neon-vfpv4-openstlinux\_weston-linux-gnueabi/**openstlinux-hands**/  Build logs are in  $**DISTRI\_BUILD\_PATH/tmp-glibc/**work/cortexa7hf-neon-vfpv4-openstlinux\_weston-linux-gnueabi/**openstlinux-hands**/\*/temp |

* + **Observe the change after building the recipe**

|  |
| --- |
| **cat $DISTRI\_BUILD\_PATH/conf/bblayers.conf**    You should see new line with  /…./workspace \    **ls $DISTRI\_BUILD\_PATH/workspace/recipes/**  openstlinux-hands  **ls $DISTRI\_BUILD\_PATH/workspace/appends/**  openstlinux-hands.bbappend |

* + **Can Deploy to target and test (skipped here)**

|  |
| --- |
| **devtool deploy-target openstlinux-hands root@<board ip@>**    Example If ethernet usb activated  **devtool deploy-target openstlinux-hands** [**root@192.168.7.2**](mailto:root@192.168.7.2) |

**Step 2.2 Create layer**

* + **Create openstliunx-hands layer**

|  |
| --- |
| **cd $DISTRI\_ROOT/$DISTRI\_RELEASE/layers/meta-st**    **bitbake-layers create-layer --priority 6 -e openstlinux-hands meta-open-hands-layer** |

* + **Add openstlinux-hands layer**

|  |
| --- |
| **cd $DISTRI\_BUILD\_PATH**    **bitbake-layers add-layer ../layers/meta-st/meta-open-hands-layer/** |

* + **Observe the changes after adding the layer**

|  |
| --- |
| **cd $DISTRI\_ROOT/$DISTRI\_RELEASE/layers/meta-st**    **ls meta-open-hands-layer** |

* + **Copy recipe and source**

in our case source will be local but could be a git repository

|  |
| --- |
| **cd $DISTRI\_ROOT/$DISTRI\_RELEASE/layers/meta-st**    **cp $WORK\_RECIPES/openstlinux-hands/openstlinux-hands.bb $ST\_RECIPES/openstlinux-hands.bb**    **mkdir ./$ST\_RECIPES/**openstlinux-hands    **cp $WORK\_DIR/$ST\_HANDS\_ON\_APPLI/Makefile ./$ST\_RECIPES/**openstlinux-hands    **cp $WORK\_DIR/$ST\_HANDS\_ON\_APPLI/openstlinux-hands.c ./$ST\_RECIPES/**openstlinux-hands |

* + **Modify SRC\_URI with local sources**

in our case source will be local but could be a git repository

|  |
| --- |
| No information for SRC\_URI yet present (only an external source tree was specified)  Please add in **$ST\_RECIPES/openstlinux-hands.bb**    SRC\_URI = "file://openstlinux-hands.c  \  <file://Makefile>"  S = "${WORKDIR}" |

* + **Add the component to image**
* create **st-image-weston.bbappend with following contents**

|  |
| --- |
| **cd $DISTRI\_ROOT/$DISTRI\_RELEASE/layers/meta-st/meta-st-openstlinux/recipes-st/images**  touch **st-image-weston.bbappend**  **gedit st-image-weston.bbappend &**  Please add  IMAGE\_INSTALL\_append = "openstlinux-hands" |

* + **Add the component to distribution**

|  |
| --- |
| cd $DISTRI\_ROOT/$DISTRI\_RELEASE/layers/meta-st/meta-st-openstlinux/conf/template/    Add layer to ADDONSLAYERS in bblayers.conf.sample  ADDONSLAYERS += [${@'${OEROOT}/meta-st/meta-open-hands-layer' if os.path.isfile('${OEROOT}/meta-st/meta-open-hands-layer/conf/layer.conf') else ''}](mailto:$%7b@'$%7bOEROOT%7d/meta-st/meta-open-hands-layer'%20if%20os.path.isfile('$%7bOEROOT%7d/meta-st/meta-open-hands-layer/conf/layer.conf')%20else%20''%7d) |

* + **The recipe layer is integrated so could reset the temporary working directory**

|  |
| --- |
| **devtool reset openstlinux-hands** |

* + **Check recipe in bitbake**

|  |
| --- |
| cd $DISTRI\_BUILD\_PATH  bitbake -s | grep openstlinux-hands  can build with individual commands  bitbake -c do\_clean openstlinux-hands  bitbake -c do\_compile openstlinux-hands  bitbake -c do\_install openstlinux-hands |

* **Step 2.3 Build and test the image**
  + **Can rebuild the build image**

|  |
| --- |
| **devtool build-image -p openstlinux-hands st-image-weston** |

**or**

|  |
| --- |
| **bitbake st-image-weston** |

* + **Flash the new image**

|  |
| --- |
| Flash images are in  $**DISTRI\_BUILD\_PATH/tmp-glibc/deploy/images/stm32mp1** |

To generate raw image

|  |
| --- |
| cd $**DISTRI\_BUILD\_PATH/tmp-glibc/deploy/images/stm32mp1** |
| **./scripts/create\_sdcard\_from\_flashlayout.sh $FLASH\_IMAGE\_TRUSTED** |

* + **Test the new image**

 on target

|  |
| --- |
| /usr/local/openstlinux-hands |

* **Step 3 How to modify a component in Yocto (optional)**
  + **Step3.1 How to modify an application**

Wiki reference

[**STM32MP1 Distribution Package**](https://wiki.st.com/stm32mpu/index.php/STM32MP1_Distribution_Package)

Section 9: Modifying software running on Arm Cortex-A

Please refer to previous step2 and use devtool modify

|  |
| --- |
| **devtool modify openstlinux-hands $WORK\_DIR/my\_openstlinux-hands** |

 Modify source code

|  |
| --- |
| **devtool build openstlinux-hands**  **or**  bitbake -c do\_clean openstlinux-hands  bitbake -c do\_compile openstlinux-hands  bitbake -c do\_install openstlinux-hands  **bitbake st-image-weston** |

* + **Step3.2 How to modify kernel (kernel config, built-in driver, dts file)**

How to customize the linux kernel in yocto

The previous method can not be used with linux kernel as

all component of yocto image have dependency on the kernel

So when doing **bitbake st-image-weston it would take a very long time**

So modification must be applied as a patch added in recipe on the kernel

Wiki reference

[**STM32MP1 Distribution Package**](https://wiki.st.com/stm32mpu/index.php/STM32MP1_Distribution_Package)

Section 9: Modifying software running on Arm Cortex-A

[How\_to\_customize\_the\_Linux\_kernel](https://wiki.st.com/stm32mpu/index.php/How_to_customize_the_Linux_kernel)

|  |
| --- |
| **devtool extract linux-stm32mp $WORK\_DIR/my\_linux-stm32mp**  cd **$WORK\_DIR/my\_linux-stm32mp/**  **git status** |

Apply modifications and create patch

|  |
| --- |
| git format-patch -2  cp \*.patch **$DISTRI\_ROOT/$DISTRI\_RELEASE/layers/meta-st/meta-st-stm32mp/recipes-kernel/linux/linux-stm32mp/4.19/4.19.9/** |

 Create **$DISTRI\_ROOT/$DISTRI\_RELEASE/layers/meta-st/meta-st-stm32mp/recipes-kernel/linux/** linux-stm32mp\_4.19.bbappend and add patch

|  |
| --- |
| At end of  SRC\_URI\_append = " \  ….  file://${LINUX\_VERSION}/4.19.9/<patch\_name>.patch \  ...  " |

* + **Step3.3 How to modify M4 firmware**

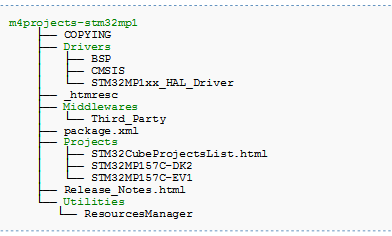
[**STM32MP1 Distribution Package**](https://wiki.st.com/stm32mpu/index.php/STM32MP1_Distribution_Package)

Section 10 : Modifying software running on Arm Cortex-M

[STM32CubeMP1\_Package](https://wiki.st.com/stm32mpu/index.php/STM32CubeMP1_Package)

Section 5.6 : Distribution Package for STM32CubeMP1

|  |
| --- |
| **devtool modify** m4projects-stm32mp1  **$WORK\_DIR/my\_** m4projects-stm32mp1  **cd $WORK\_DIR/my\_** m4projects-stm32mp1 |



Add new fw source code MyFirmware to new directory in STM32MP157C-DK2 for instance

Add new fw entry in M4 project

|  |
| --- |
| **cd $DISTRI\_ROOT/$DISTRI\_RELEASE/layers/meta-st/meta-st-stm32mp/recipes-extended/m4projects**  **add** MyFirmware in **m4projects-stm32mp1.bb**  **follow example:**  'STM32MP157C-DK2/Applications/OpenAMP/OpenAMP\_TTY\_echo' \ |

Modify default fw :

|  |
| --- |
| **cd $DISTRI\_ROOT/$DISTRI\_RELEASE/layers/meta-st/meta-st-stm32mp/conf/machine/include/**  **Modify** DEFAULT\_COPRO\_FIRMWARE with new fw name **st-machine-common-stm32mp.inc** |

Rebuild all M4 fw (including new one)

|  |
| --- |
| **cd** $**DISTRI\_BUILD\_PATH**  bitbake m4projects-stm32mp1 -f -c cleanall  bitbake m4projects-stm32mp  bitbake st-image-weston |

* + **Step3.4 Test co-processing example in yocto**

Apply previous steps 3.2 and 3.3 to example from Co-Processing example rebuild st-image-weston,

reflash the image and test it.

* + **Overview of STM32mp1 embedded software distribution environment**

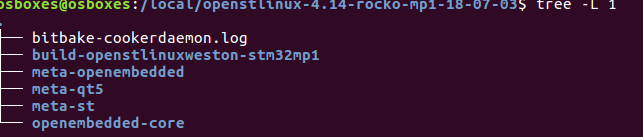
**Bitbake commands**

https://wiki.st.com/stm32mpu/index.php/BitBake\_cheat\_sheet

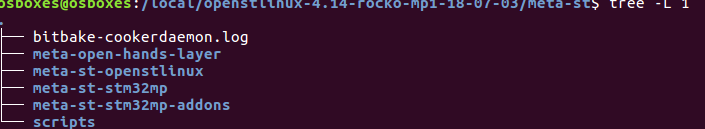
|  |
| --- |
| Reminder of unitary commands  bitbake -h  bitbake –s  List all recipes  bitbake -s | grep <recipe\_name>  bitbake –c listtasks <recipe\_name>  List all tasks possible from recipes  bitbake -c do\_clean <recipe\_name>  bitbake -c do\_compile <recipe\_name>  bitbake -c do\_install <recipe\_name> |
|  |

**STM32mp1 embedded software main directories**

**Recipe and meta-layer directories (from repo manifest)**



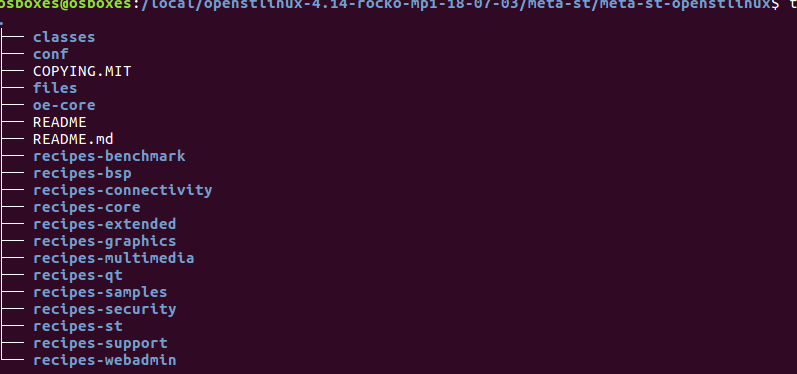
**ST layers**



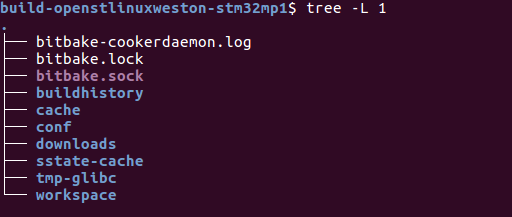
**OpenSTLinux BSP layer**



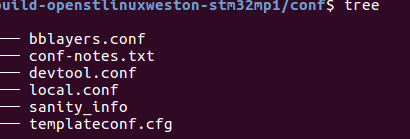
**OpenSTLinux application framework layer**



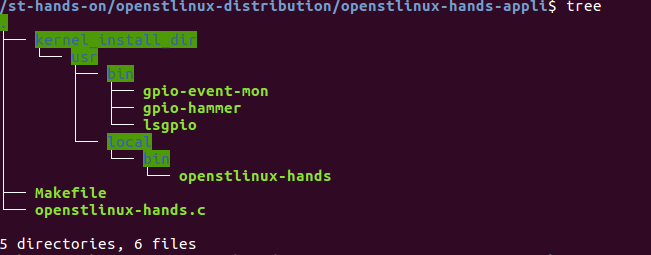
**Build directory**



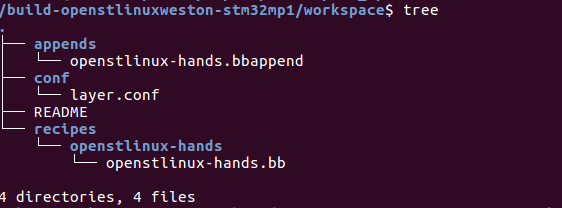
**Conf directory**



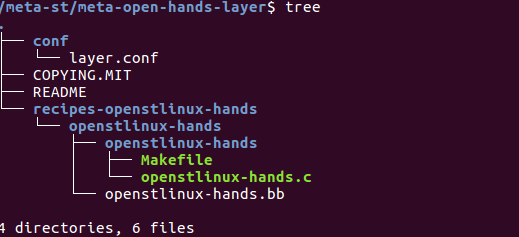
**Hands on work source**



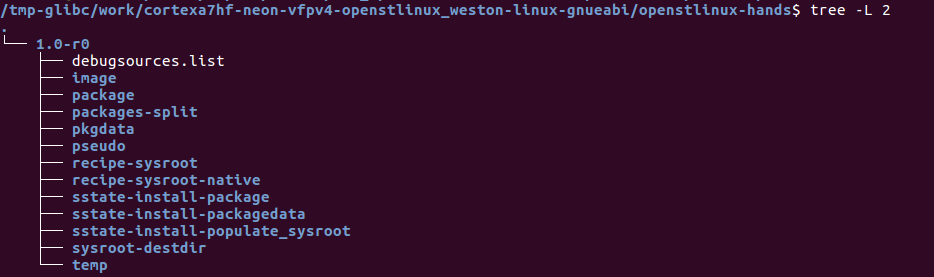
**Hands on recipe (temporary directory after devtool add)**



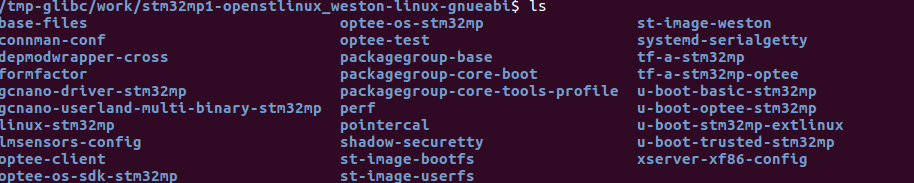
**Hands on layer (after updating source and recipe)**



**Hands on build directory log files**



**Kernel build directory**



**How to add/modify bsp layer reminder**

|  |
| --- |
| Configuration files for the machine are in  **conf/bblayers.conf**   Add or replace in the **BBLAYERS** variable from  **BBLAYERS** += " \  ${BASELAYERS} \  **${BSPLAYER} \**  ${ADDONSLAYERS} \  …  **conf/local.conf**  Configure **MACHINE** to the one you want to build  Configure **BB\_NUMBER\_THREADS** and **PARALLEL\_MAKE** |
| Our machine is in  **meta-st/meta-st-stm32mp/conf/machine/stm32mp1.conf** |