

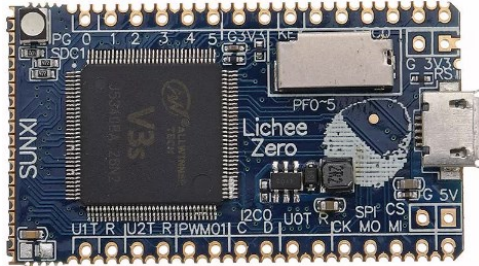
How to compile Linux with Qt5 option using Yocto for LICHEE PI ZERO

- Post author: [Michał Wołowik](#)
- Post published: 2020-04-23
- Post category: [Blog](#) / [Linux](#)
- Post comments: [0 Comments](#)

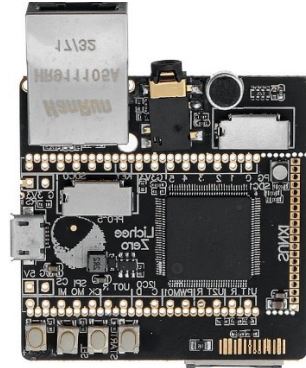
Instruction how to build an image for Lichee Pi Zero and Lichee Pi Zero Dock in Yocto

Products:

Lichee Pi Zero Version



Lichee Pi Zero Dock Version



General Note:

Assumed that Linux Ubuntu is installed

List of tested elements

WiFi
Ethernet
Lcd
Touchscreen
Led
Backlight for Lcd

List of not tested elements

Bluetooth – appears during system boot up
Microphone
Headphone

How to build images

1. First, make sure the following packages are installed in the system

```
sudo apt-get install gawk wget diffstat unzip texinfo gcc-multilib build-essential chrpath socat libssl1.2-dev xterm libgmp3-dev libmpc-dev
```

Note: More information can be found on the Yocto reference manual.

2. Download necessary Yocto packages listed below. Be sure to be in the root of the home folder.

```
mkdir yocto
cd yocto
mkdir build
git clone git://git.yoctoproject.org/poky --depth 1 -b dunfell
cd poky
git clone git://git.openembedded.org/meta-openembedded --depth 1 -b dunfell
git clone https://github.com/meta-qt5/meta-qt5.git --depth 1 -b dunfell
git clone https://github.com/voloviq/meta-licheepizero --depth 1 -b dunfell
```

if you have problem with download from

```
git clone git://git.openembedded.org/meta-openembedded --depth 1 -b dunfell
```

you can use following command:

```
sudo apt-get install tor
```

then

```
torify git clone git://git.openembedded.org/meta-openembedded --depth 1 -b dunfell
```

3. Select directory to build Linux

Zero version

```
source oe-init-build-env ~/yocto/build/licheepizero
```

Zero Dock version

```
source oe-init-build-env ~/yocto/build/licheepizero-dock
```

4. Modify bblayers.conf (located in ~/yocto/build/licheepizero/conf (or lichEEPizero-dock/conf))

```
BBLAYERS ?= " \
${HOME}/yocto/poky/meta \
${HOME}/yocto/poky/meta-poky \
${HOME}/yocto/poky/meta-openembedded/meta-oe \
${HOME}/yocto/poky/meta-openembedded/meta-networking \
${HOME}/yocto/poky/meta-openembedded/meta-python \
${HOME}/yocto/poky/meta-qt5 \
${HOME}/yocto/poky/meta-licheepizero \
"
```

Note: Please adapt PATH of conf/bblayers.conf if necessary.

5. Modify or align following elements in local.conf (located in ~/yocto/build/licheepizero/conf (or lichEEPizero-dock/conf)) file

```
MACHINE ??= "licheepizero-dock"
or
MACHINE ??= "licheepizero"
DL_DIR = "${HOME}/yocto/downloads"
SSTATE_DIR = "${HOME}/yocto/sstate-cache"
TMPDIR = "${HOME}/yocto/tmp"
at the end add some option if necessary
RM_OLD_IMAGE = "1"
INHERIT += "rm_work"
```

Note: Please adapt rest of conf/local.conf parameters if necessary.

if you want to use some package (it's for our QT project not necessary) more add following command to the end of local.conf file

```
IMAGE_INSTALL_append = "qtserialport minicom packagegroup-core-buildessential nano qtsvg qtsvg-plugins qtdeclarative
qtdeclarative-qmlplugins qtdeclarative-plugins qtquickcontrols-qmlplugins qtquickcontrols2 qtquickcontrols2-qmlplugins
qtgraphicaleffects-qmlplugins"
```

```
CORE_IMAGE_EXTRA_INSTALL += "sqlite3"
```

```
#IMAGE_INSTALL_append = "qtbase-plugins qtbase-tools qtimageformats-plugins qtsystems qtsystems-tools qtsystems-qmlplugins qtscrip
qtlocation-plugins qtlocation-qmlplugins procps"
```

```
IMAGE_INSTALL_append = " minicom packagegroup-core-buildessential"
```

```
IMAGE_INSTALL_append = " qtdeclarative qtdeclarative-plugins qtdeclarative-qmlplugins qtgraphicaleffects-qmlplugins qtsvg qtsvg-plugins
qtquickcontrols2 qtquickcontrols-qmlplugins qtquickcontrols2-qmlplugins"
```

```
#CORE_IMAGE_EXTRA_INSTALL += "qtsvg"
```

```
#IMAGE_INSTALL_append = " qtsvg"
```

```
MACHINE_EXTRA_RRECOMMENDS = " kernel-modules"
```

Example of this file exist in [repository](#) (local.conf)

6. Build objects

Issue from console one of the following option

console image

bitbake console-image

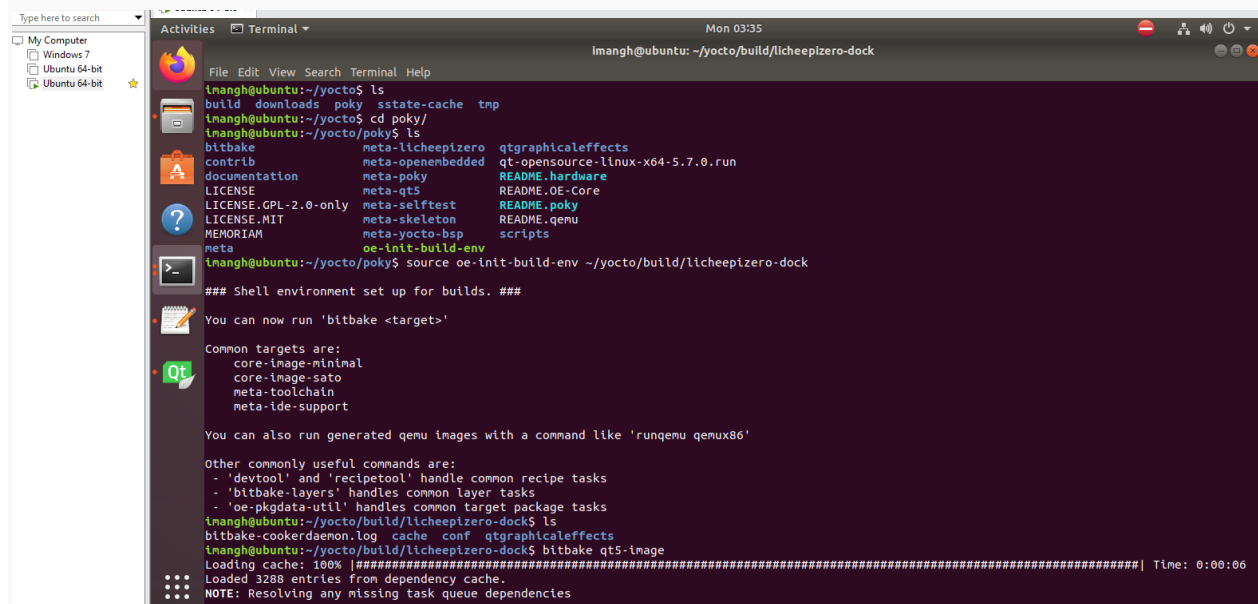
sato (X11) image

bitbake sato-core-image

qt5 toolchain

bitbake meta-toolchain-qt5

qt5 toolchain sdk
bitbake meta-toolchain-qt5



```
lnangh@ubuntu:~/yocto$ ls
build downloads poky sstate-cache tmp
lnangh@ubuntu:~/yocto$ cd poky/
lnangh@ubuntu:~/yocto/poky$ ls
bitbake      meta-licheepzero  qtgraphicaleffects
contrib      meta-openembedded qt-opensource-linux-x64-5.7.0.run
documentation meta-poky          README.hardware
LICENSE      meta-qt5           README.OE-Core
LICENSE.GPL-2.0-only meta-selftest      README.poky
LICENSE.MIT  meta-skeleton      README.qemu
MEMORIAM    meta-yocto-bsp     scripts
meta        oe-init-build-env
lnangh@ubuntu:~/yocto/poky$ source oe-init-build-env ~/yocto/build/licheepzero-dock
## Shell environment set up for builds. ##
You can now run 'bitbake <target>'
Common targets are:
  core-image-minimal
  core-image-sato
  meta-toolchain
  meta-ide-support
You can also run generated qemu images with a command like 'runqemu qemu86'
Other commonly useful commands are:
  - 'devtool' and 'recipetool' handle common recipe tasks
  - 'bitbake-layers' handles common layer tasks
  - 'oe-pkgdata-util' handles common target package tasks
lnangh@ubuntu:~/yocto/build/licheepzero-dock$ ls
bitbake-cookerdgenon.log cache conf qtgraphicaleffects
lnangh@ubuntu:~/yocto/build/licheepzero-dock$ bitbake qt5-image
Loading cache: 100% |#####| Time: 0:00:06
Loaded 3288 entries from dependency cache.
NOTE: Resolving any missing task queue dependencies
```

This step takes one day!

7. After compilation images appear in

Zero version

~/yocto/tmp/deploy/images/licheepzero

Zero Dock version

~/yocto/tmp/deploy/images/licheepizero-dock

8. Insert SD CARD into dedicated CARD slot and issue the following command to write an image

Note:

Be 100% sure to provide a valid device name (of=/dev/sde). Wrong name "/dev/sde" damage Your system file !

Zero version

```
sudo dd if=~/yocto/tmp/deploy/images/licheepizero/qt5-image-licheepizero.sunxi-sdimg of=/dev/sde bs=1024
```

Zero Dock version

```
sudo dd if=~/yocto/tmp/deploy/images/licheepizero-dock/qt5-image-licheepizero-dock.sunxi-sdimg of=/dev/sde bs=1024
```

To get ip address of board you must :

first: go to serial debug port and type:

ifconfig

To test the program written and built in QT software, it (eg. Standby) should be in the folder /home/root/

for transfer this program to Lichee Pi Zero Dock Version board you can use

SSH @root(ip address of board eg. 192.168.1.1)

or use 3rd party program like putty or winscp.

and then run the following commands.

chmod 777 Standby

./ Standby



Limitation

rootfs-resize not [working](#) (SD CARD size can be resized manually)

no wiringpi or similar library to control GPIO in C code

discover problem when WiFi connected to access [point](#) (probably some drivers issues), nevertheless WiFi works

Source link: <https://www.emsyslabs.com/how-to-compile-linux-with-qt5-option-using-yocto-for-lichee-pi-zero/>