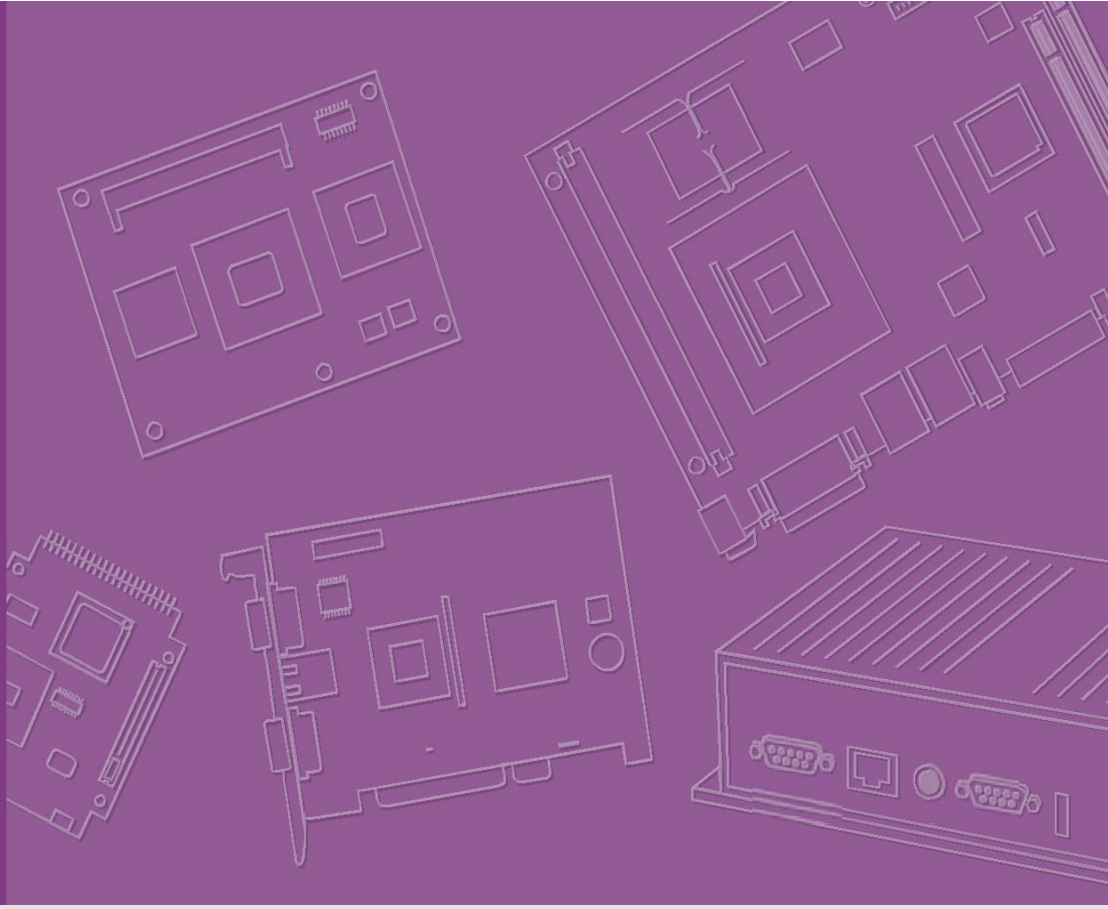

User Guide

Yocto Linux

Board Support Package Ver.2
For Quark series

ADVANTECH

Enabling an Intelligent Planet

Table of Contents

Chapter 1

Getting Started

1. Getting Started

1.1 Prerequisites

All operations in this guide are based on Ubuntu 12.04 LTS 64bit only. First please install Ubuntu 12.04 LTS 64bit* with minimum 2GB memory.

*** ubuntu-12.04.1-desktop-amd64.iso**

1.1.1 To install required packages

Please login and perform the following commands:

```
sudo apt-get install ssh
sudo apt-get install ia32-libs libx11-dev:i386 libreadline6-dev:i386 \
   libgl1-mesa-glx:i386 zlib1g-dev:i386 uuid-dev:i386 liblzo2-dev:i386 \
   libncurses5-dev:i386
sudo apt-get install \
   bison build-essential ccache dpkg flex gcc g++ gettext intltool \
   libarchive-zip-perl libfreetype6-dev libdbus-glib-1-dev liborbit2-dev \
   libxml2-dev libx11-dev libgtk2.0-dev liblzo2-2 libtool m4 \
   patch rpm tcl uboot-mkimage uuid zlib1g zlib1g-dev \
   git gnupg flex bison gperf build-essential  zip \
   curl libc6-dev libncurses5-dev x11proto-core-dev libx11-dev:i386 \
   libreadline6-dev:i386 libgl1-mesa-glx:i386 libgl1-mesa-dev g++-multilib \
   mingw32 tofrodos python-markdown libxml2-utils xsltproc zlib1g-dev:i386 \
   gcc-4.6 g++-4.6 cpp-4.6 gcc-4.6-multilib uuid-dev liblzo2-dev \
   uboot-mkimage libarchive-zip-perl \
   wget git-core unzip texinfo gawk diffstat build-essential chrpath \
   sed cvs subversion coreutils texi2html \
   docbook-utils python-pysqlite2 help2man make gcc g++ \
   desktop-file-utils libgl1-mesa-dev libglu1-mesa-dev mercurial \
   autoconf automake groff curl lzop asciidoc xterm
sudo apt-get install libncurses5-dev:i386 liblzo2-dev:i386 uuid-dev:i386
sudo ln -s /usr/lib/i386-linux-gnu/mesa/libGL.so.1 /usr/lib/i386-linux-gnu/libGL.so
tar zcvf ~/usr_lib_i386-linux-gnu_for_Building_Android_KK.tar.gz \
     /usr/lib/i386-linux-gnu/{libuuid.a,libuuid.so,liblzo2.so,liblzo2.a}
sudo apt-get install uuid-dev liblzo2-dev
sudo tar zxvf ~/usr_lib_i386-linux-gnu_for_Building_Android_KK.tar.gz -C /
```

1.1.2 To install JDK

Please download "jdk-6u45-linux-x64.bin" manually, put it to directory ~/FILES/ and perform the following commands:

```
cd /usr/lib
sudo ~/FILES/jdk-6u45-linux-x64.bin
sudo mkdir jvm; cd jvm
sudo mv ../jdk1.6.0_45 .
cd jdk1.6.0_45/
sudo update-alternatives --install /usr/bin/java    java    /usr/lib/jvm/jdk1.6.0_45/jre/bin/java    2
sudo update-alternatives --install /usr/bin/javac   javac   /usr/lib/jvm/jdk1.6.0_45/bin/javac   2
sudo update-alternatives --install /usr/bin/jar     jar     /usr/lib/jvm/jdk1.6.0_45/bin/jar     2
sudo update-alternatives --install /usr/bin/javap   javap   /usr/lib/jvm/jdk1.6.0_45/bin/javap   2
sudo update-alternatives --install /usr/bin/javadoc javadoc /usr/lib/jvm/jdk1.6.0_45/bin/javadoc 2
sudo update-alternatives --config javap
sudo update-alternatives --config javadoc
sudo update-alternatives --config java
sudo update-alternatives --config javac
sudo update-alternatives --config jar
cd ~/
sudo sh -c "echo "JAVA_HOME=/usr/lib/jvm/jdk1.6.0_45" >> /etc/environment"
```

1.2 Conventions

${BOARD_ID} : board id
e.g. ubc222 or ubc221

${IMAGE_PACK} : prebuilt image pack
e.g. `U222LIV2011_quark_2016-01-29.zip`

${IMAGE_DIR} : the directory prebuilt image pack extracted to
e.g. `~/U222LIV2011_quark_2016-01-29`

${BSP_PACK} : BSP pack
e.g. `U222LBV2011_2016-01-29.zip`

${BSP_HOME} : the directory BSP pack extracted to
e.g. `~/LBV2011/meta-clanton_v1.2.0`

${BDIR} : build directory
e.g. `yocto_build`

${SD_MOUNT} : mount point of SD card in Ubuntu
e.g. `/media/sdf1`

${POKY} : Yocto poky version
e.g. `1.7.2`

debug console / serial console
serial terminal program (e.g. minicom, putty, teraterm ...) that serial port is configured to 115200 8N1

terminal console
terminal program (e.g. gnome-terminal, xfce4-terminal ...)

1.3 Introducing BSP

The BSP is based on Yocto Project with Intel enhanced features for Quark, plus specific target board features from Advantech Inc..

1.3.1 Naming Rule

The BSP/prebuilt image pack name is consist of the model name followed by "LB" or "LI" plus version number and released date.

For example, U222LBV2011_2016-01-29.zip which "U222" stands for **U**BC-**222**, "LB" is acronym of **L**inux **B**SP, "V2011" stands for **V**ersion **2.011**.

For example, U222LIV2011_quark_2016-01-29.zip which "LI" is acronym for prebuilt **L**inux **I**mage.

1.3.2 BSP pack

Unpack BSP pack to home directory by performing the following command:

```
$ unzip ${BSP_PACK} -d ~/
```

The description of some important folders list below:

${BSP_HOME}/

meta-advantech/ : meta layer by Advantech

meta-intel-*/ : meta layer by Intel

setup.sh : to setup one new build environment

oe-init-build-env : to initiate build environment

1.3.3 Prebuilt image pack

Perform the following command to unpack prebuilt-image-pack to home directory.

```
$ unzip ${PREBUILT_IMAGE_PACK} -d ~/
```

Prepare one FAT32 formatted SD card, and mount it to mount point.

```
$ cp -a ${PREBUILT_IMAGE_DIR}/sdcard/* ${SD_MOUNT}/
```

1.4 Build Instructions

1.4.1 To create one new build environment

Perform the following commands in terminal console

```
$ cd ${BSP_HOME}/
$ ./setup.sh
$ BOARD=${PRODUCT} source ./oe-init-build-env ${BDIR}
```

1.4.2 To continue an exist build environment

Perform the following commands in terminal console

```
$ cd ${BSP_HOME}
$ BOARD=${BOARD_ID} source ./oe-init-build-env ${BDIR}
```

1.4.3 To build all image files

1) To create/continue one build environment
2) Perform the following command in terminal console

```
$ bitbake image-full
```

3) The following files will be located in directory "./tmp/deploy/images/quark" while building process finished successfully.

```
boot/grub/grub.conf
bzImage
core-image-minimal-initramfs-quark.cpio.gz
grub.efi
image-full-quark.ext3
```

1.4.4 To build toolchain installer

1) To create/continue a build environment
2) Perform the following command in terminal console

```
$ bitbake image-full -c populate_sdk
```

3) The installer, iot-devkit-glibc-x86_64-image-full-i586-toolchain-**${POKY}**.sh, will be located in the directory "./tmp/deploy/sdk".

1.4.5 To build grub individually

1) To create/continue a build environment
2) Perform the following command in terminal console

```
$ bitbake grub
```

3) The file, grub.efi, will be located in directory, ./tmp/deploy/images/quark.

1.4.6 To build linux kernel individually

1) To create/continue a build environment
2) Perform the following command in terminal console
   A. to show up menuconfig

      ```
      $ bitbake linux-yocto-quark -c menuconfig
      ```

   B. to do build

      ```
      $ bitbake linux-yocto-quark
      ```

3) The files, bzImage, will be located in directory, ./tmp/deploy/images/quark.

1.4.7 To build initramfs individually

1) To create/continue a build environment
2) Perform the following command in terminal console

   ```
   $ bitbake core-image-minimal-initramfs
   ```

3) The file, `core-image-minimal-initramfs-quark.cpio.gz`, will be located in directory, ./tmp/deploy/images/quark.

Chapter 2

Customization

2. Customization

2.1 Setting up SDK

1) Please follow 1.4.4 to build one toolchain installer
2) Perform the following command in terminal console

```
$ cd ${BSP_HOME}/${BDIR}/tmp/deploy/sdk
$ sudo ./iot-devkit-glibc-x86_64-image-full-i586-toolchain-${POKY}.sh
```

3) Enter directory or press Enter while following question shows up:

```
Enter target directory for SDK (default: /opt/iot-devkit/1.7.2):
```

4) Just press Enter while following question shows up:

```
You are about to install the SDK to "/opt/iot-devkit/1.7.2". Proceed[Y/n]?
```

5) While following message shows up means the SDK is ready.

```
Extracting SDK...done
Setting it up...done
SDK has been successfully set up and is ready to be used.
```

2.2 Setting up cross compiling environment

1) SDK has been set up. (ref. 2.2)
2) Perform the following command in terminal console

```
$ source /opt/iot-devkit/${POKY}/environment-setup-i586-poky-linux
```