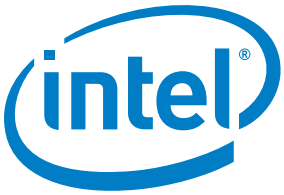

Building Intel® Atom™ E3800 Processor Development Kit Yocto Project* Board Support Package (BSP)

User Guide

June 2014

Document Number: 330693-001US

Contents

Figures

Tables

Revision History

Date	Revision	Description
June 2014	001	Initial release

§

1 Introduction

This document provides instructions for building the Yocto Project* Reference OS for the Intel® Atom™ E3800 Processor Development Kit (formerly known as Valley Island) and contains the following information:

- Section 1: Introduction
- Section 2: Supported Platforms
- Section 3: Building the Image for Valley Island Yocto Project* Board Support Package (BSP)
- Section 4: BIOS and Board Configuration
- Section 5: Product features
- Section 6: Known Issues for LPSS I/O Devices

For more information about Baytrail product info, refer to the EDC webpage at:

http://www.intel.com/content/www/us/en/intelligent-systems/bay-trail/atom-processor-e3800-family-overview.html

1.1 Terminology

Table 1. Terminology

Term	Description
ACPI	Advanced Configuration and Power Interface
APIC	Advanced Programmable Interrupt Controller
BSP	Board Support Package
EDC	Embedded Design Center
FPGA	Field Programmable Gate Array
KSC	Keyboard System Controller
LPSS	Low Power Sub-System
PCI	Peripheral Component Interconnect
PWM	Pulse Width modulation
RTC	Real Time Clock
SCC	Storage Connecting Circuit
SUT	System Under Test

1.2 Reference Documents

Table 2. Reference Documents

Document	Document No./Location
Yocto Project Setup Getting Started Guide*	330696

§

2 Supported Platforms

The Yocto Project* Reference OS for Valley Island supports the following platforms:

- Bayley Bay Fab C (Green)
- Bakersport Fab A (Red)
- Valley Island Development Kit*

Note: * Minimal test coverage (Boot test).

§

3 Building the Image for Valley Island Yocto Project* Board Support Package (BSP)

3.1 Prerequisites

Prior to building the image for Valley Island Yocto Project* Intel BSP, make sure that the environment setup is complete based on the steps in the *Yocto Project* Setup Getting Started Guide* (Document Number: 330696).

3.2 Steps for Building the Image

Now that the Yocto Project* poky repo and meta-intel repo are ready in your build machine, follow the steps in this section closely.

Intel recommends that you place your build folder at the same level as the `poky` directory. This shortens the `grep` and `find` command execution time when you explore the Yocto Project* recipe files.

3.2.1 Step 1: Set Up the Build Environment for the Target BSP

Enter the following commands:

```
$ cd ~/development/
$ mkdir build
$ cd build
$ source ~/development/poky/oe-init-build-env <bsp_name>
```

The environment is set up and your shell prompt is moved to `~/development/build/<bsp-name>`.

Example:

```
$ source ~/development/poky/oe-init-build-env valleyisland
```

3.2.2 Step 2: Setup the Build Configuration Files – bblayers.conf

Now you must link your build environment to the correct recipe folder. To do this, perform the following steps:

1. Modify the ~/development/build/<bsp-name>/conf/bblayers.conf file.

   ```
   $ vim ~/development/build/valleyisland/conf/bblayers.conf
   ```

2. Modify the `bblayers.conf` as shown Figure 1.

Figure 1. Modifications to bblayers.conf

```
1  # LAYER_CONF_VERSION is increased each time build/conf/bblayers.conf
2  # changes incompatibly
3  LCONF_VERSION = "6"
4
5  BBPATH = "${TOPDIR}"
6  BBFILES ?= ""
7
8  BBLAYERS ?= " \
9    /home/ilab/poky-daisy/meta \
10   /home/ilab/poky-daisy/meta-yocto \
11   /home/ilab/poky-daisy/meta-yocto-bsp \
12   /home/ilab/poky-daisy/meta-intel \
13   /home/ilab/poky-daisy/meta-intel/meta-isg/meta-valleyisland \
14   /home/ilab/poky-daisy/meta-intel/meta-tlk \
15   "
16 BBLAYERS_NON_REMOVABLE ?= " \
17   /home/ilab/poky-daisy/meta \
18   /home/ilab/poky-daisy/meta-yocto \
19   "
```

Note: The path to meta-tlk ensures that the OS image you are building carries the time-limited-kernel feature. This means that the OS image will reboot itself every 10 days. The intention is to encourage customers to build their own production OS.

If you do not want this feature, simply remove line-14 (see the following note).

Note: `/home/ilab/poky-daisy` should be replaced with your build machine user's home directory, e.g. `/home/<user>/development/poky`.

3.2.3 Step 3: Set Up the Build Configuration Files – local.conf

Next, configure your local build configuration file (local.conf) so that it matches your target platform machine type.

```
$ vim ~/development/build/valleyisland/conf/local.conf
```

Specify the machine type to match your target platform. You can search for the offered machine type in the following meta-intel BSP directory structure:

The machine type configuration file is at the following local within a BSP recipe: meta-intel/meta-<bsp>/conf/machine/<machine-type>.conf.

For example: meta-intel/meta-isg/meta-valleyisland/machine/ has the following machine type:

- valleyisland-32.conf
- valleyisland-64.conf

Thus, to set the machine type for your local build configuration file, simply make the following change:

Figure 2. Modification to Local Build Configuration File for Machine Type

```
#
# Machine Selection
#
# You need to select a specific machine to target the build with. There are a selection
# of emulated machines available which can boot and run in the QEMU emulator:
#
#MACHINE ?= "qemuarm"
#MACHINE ?= "qemumips"
#MACHINE ?= "qemuppc"
#MACHINE ?= "qemux86"
#MACHINE ?= "qemux86-64"
#
# There are also the following hardware board target machines included for
# demonstration purposes:
#
#MACHINE ?= "atom-pc"
#MACHINE ?= "beagleboard"
#MACHINE ?= "mpc8315e-rdb"
#MACHINE ?= "routerstationpro"
#
# This sets the default machine to be qemux86 if no other machine is selected:
MACHINE ??= "valleyisland-32"
```

There are two types of “MACHINE” options for Valley Island BSP.

- For 32-bit system: valleyisland-32
- For 64-bit system: valleyisland-64

The meta-valleyisland contains support Intel® High Definition Audio (Intel® HD Audio). However, Intel® HD Audio support is not turned on by default. The Intel® HD Audio driver is dependent on `gstreamer` plugins and `ffmpeg` plugins to work properly. These `gstreamer` plugins require additional license flag settings to be included in the build.

For Intel® HD Audio support, add the following line in local.conf.

```
LICENSE_FLAGS_WHITELIST = “commercial”
```

Warning: By performing this step, you acknowledge that you are including a software package with a commercial license requirement. Yocto Project* does not grant any commercial licenses and it is the responsibility of the user to obtain the proper license.

3.2.4 Step 4: Build an Image

There are a few images that you can build with an Intel BSP recipe and poky layers. These image types are located under the `poky/meta/recipes-*/images/` folder.

Additionally, some BSP recipes may have a specific image type defined under `meta-intel/meta-<bsp>/images/`.

The most frequently used image types are the following:

- core-image-sato = Image contains Sato mobile-style desktop
- core-image-sato-sdk = core-image-sato + SDK + development tool chain
- core-image-minimal = basic Linux* kernel with shell terminal only

To build your choice of image, do the following:

```
$ cd ~/development/build/valleyisland
# Command: bitbake <image-type>
$ bitbake core-image-sato
```

OR

```
$ bitbake core-image-sato-sdk
```

OR

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

The above command initiates the image-build processes; where you see `bitbake`, the build system spawns multiple threads to fetch, configure, compile, install, and package each software ingredients.

At the end of the build, the bootable image with the file name `<image-type>-<machine-type>.hddimg` will be located at the following location for Yocto Project* version 1.5 and later:

```
~/development/build/<bsp-name>/tmp/deploy/images/<machine-type>/<image-type>-<machine-type>.hddimg
```

Example:

```
~/development/build/crystalforest/tmp/deploy/images/valleyisland-32/core-image-sato-valleyisland-32.hddimg
```

3.2.5 Step 5: Prepare the Bootable HDDIMG Image on a USB Mass Storage Device

Note: The following steps are performed on the development machine used to build the Yocto Project* Intel BSP recipe as described in Section 3.2.4.

This BSP recipe contains (or builds) live images that must be converted to a partitioned image format to boot them on the supported platforms.

To deploy the hddimg image on a USB or SATA device (boot-medium), you must know the device name on your host machine and the device name that is intended on the target platform.

Warning: Be careful with this step as using the wrong host device path can result in overwriting data on your host machine.

In Linux* distros, USB and SATA devices typically appear as `/dev/sdb`, `/dev/sdc`, etc. Watch your system messages as you connect the device to determine exactly which device name is assigned to the device. You can also use `df` to detect which is the device that is associated with the USB or SATA boot-medium.

On the Valley Island platform, when the prepared boot-medium is attached, assuming only one storage device, typically your USB or SATA boot medium is `/dev/sda`.

There is a script under poky/scripts/contrib that can help install the HDDIMG into the USB stick.

1. Enter the following command lines:

```
# Identify the device file for your USB thumb drive.
$ df
# To write the bootable image to USB thumb drive, we assume it
is at /dev/sdc.
# Unmount the the USB thumb drive first.
$ umount /dev/sdc

# Go to the scripts directory in poky.
$ cd ~/development/poky/scripts/contrib/
# Command: ./mkefidisk.sh <USB> <path/to/image>
<target/rootfs>
```

Example:

```
$ sudo ./mkefidisk.sh /dev/sdc core-image-sato-valleyisland-
32.hddimg /dev/sda
```

2. Finally, insert the USB thumb drive in your target platform USB slot and boot up the system. The BIOS or boot-loader should boot the image to the Sato desktop.

Note: If your BIOS indicates that it cannot understand the bootable USB, it may mean that your USB thumb drive is corrupt. In this case, you must reformat the USB thumb drive as follow and redo the commands in this section:

```
# This assumes the USB thumb drive is at /dev/sdc
$ sudo dd if=/dev/zero of=/dev/sdc bs=1M count=512
```

§

4 BIOS and Board Configuration

Table 3. BIOS and Board Configuration

BIOS/FPGA/KSC Configuration		
	Bayley Bay Fab C	**Bakersport Fab A**
PU Stepping	B0	
BIOS version	BYTICRB_IA32_R_SPI_0080_21_SeC_Enable.bin BYTICRB_X64_R_SPI_0080_21_SeC_Enable.bin	
KSC Version	ksc_v3_12.bin	
General Settings	• Turn off Secure Boot • Device Manager > System Setup > Boot > Security Boot > Disable	
LPSS and SCC configuration	**IMPORTANT** The LPSS & SCC mode is PCI Mode by default in the BIOS setting. • To enable PCI/ACPI mode: — Device Manager > System Setup > South Cluster Configuration > LPSS & SCC Configuration > LPSS & SCC Devices Mode (select ACPI Mode or PCI Mode) • The eMMC controller v4.5 is enabled by default. To switch to eMMC controller v4.41: — Device Manager > System Setup > South Cluster Configuration > LPSS & SCC Configuration > SCC eMMC Boot Controller > eMMC4.41	
Intel® HD Audio Configuration	• Set Device Manager > System Setup > South Cluster Configuration > Audio Configuration > LPE Audio Support>Disable • Set Device Manager > System Setup > South Cluster Configuration > Audio Configuration > Audio Controller >Enable	
USB Configuration	• To enable XHCI controller: — Device Manager > System Setup > South Cluster Configuration > USB Configuration > EHCI Controller > Disable — Device Manager > System Setup > South Cluster Configuration > USB Configuration > XHCI Controller > Enable — Device Manager > System Setup > South Cluster Configuration > USB Configuration > XHCI Mode > Enable • To enable USB Device (Bakersport ONLY): — Device Manager > System Setup > South Cluster Configuration > USB Configuration >USB OTG Support > PCI Mode — Device Manager > System Setup > South Cluster Configuration > USB Configuration >USB VBUS > OFF	
SATA Configuration	• To enable AHCI mode: — Device Manager > System Setup > South Cluster Configuration > SATA Drives > Chipset SATA Mode > AHCI	

LPSS I/O Device Note:

The I2C controller driver supports fast mode by default. To enable standard mode, append the arguments to the kernel command line.

When the image is booted into a GRUB menu, press **e** to enable kernel command-line editing. Append the line at the end of linux/vmlinuz.

"i2c-designware-pci.force_std_mode=1" (PCI mode)

"i2c-designware-platform.force_std_mode=1" (ACPI mode)

§

5 Product features

This implementation supports the following features:

- Linux* Kernel version 3.10.40
- Intel's Open Source Integrated Graphics Driver for Linux* (i915)
- SATA (IDE, AHCI), USB Host Controller 2.0 and 3.0
- USB device mode (Bakersport ONLY)
- eMMC and SD Host Controller
- Intel® HD Audio, GbE (Ethernet), APIC, RTC
- LPSS I/O devices - I2C Designware, SPI-PXA2XX, HSUART, SMBus i801, PWM (ACPI mode only)
- LPSS device enumeration in PCI mode and ACPI mode

§

6 Known Issues for LPSS I/O Devices

Issue: HSUART

When running PCI mode HSUART at baud rate 2M and above, you may receive a kernel message "serial8250: too much work for irq...". Generally, it will not disrupt the transfer and the data transfer will complete without data corruption. However, occasionally data transfer may halt when this kernel message appears. In this case:

1. Reopen the HSUART port. To do this, stop the current data transfer and close the terminal.
2. Then, open a new terminal and execute the data transfer again.

Issue: SD Cards that supports DDR50 mode

Due to the limitations in BIOS, not all SD cards that support DDR50 mode are supported by this BSP.

§