Yocto Quark

User's Guide 1st Ed

Last Updated: September 18, 2015

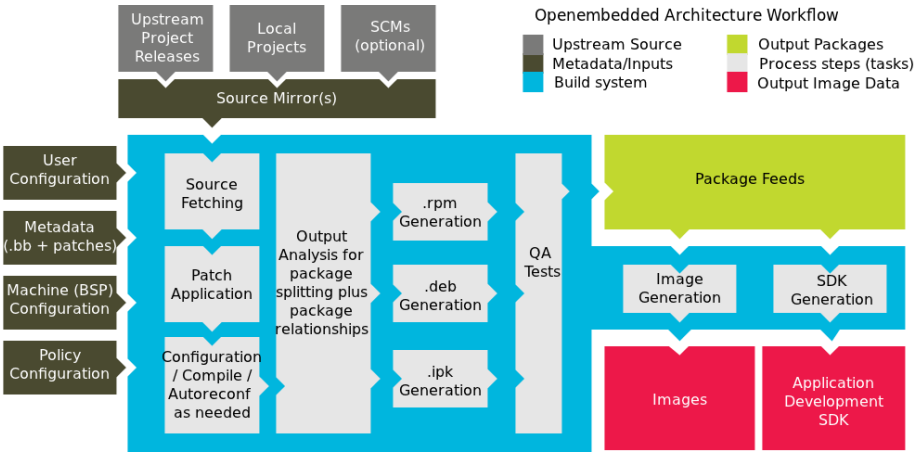
Table of Contents

Chapter 1

Introduction and Overview

1.1 Yocto Project Overview

The Yocto Project* is an open source collaboration project that provides templates, tools and methods to help you create custom Linux-based systems for embedded products regardless of hardware architecture.



1.2 Included in Yocto

The Yocto Project accomplishes the following:

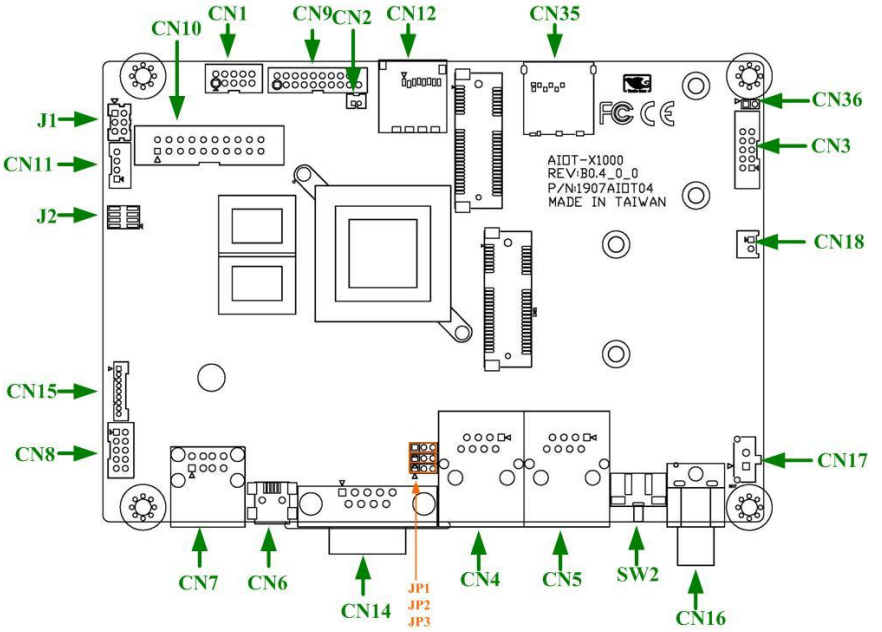
- Co-maintains and leverages Bitbake and OpenEmbedded-Core, and extends them by adding COTS BSPs, a reference distribution, documentation, etc.
- Provides a tested, pre-prepared combination of build system components
- Includes autobuilder sessions
- QA testings
- Eclipse Plugins
- Branding / Compatibility Program
- …etc…

This guide describes how to set up and run the AAEON AIOT Quark SoC X1000 Kit.

Chapter 2

Platform Setup

2.1 Platform Overview



2.2 List of Connectors

Label	Function	Connector Type
CN1	JTAG Programming Port	(TF)BOX HEADER.5*2P.180D(M).DIP.2.0mm
CN2	Batter	(TF)WAFER BOX.2P.180D.(M).1.25mm
CN3	ADC	(TF)BOX HEADER.5*2P.180D(M).DIP.2.0mm.
CN4	10/100 RJ45	(TF)RJ45.12P.90D(F).W/Transformer & LED.DIP
CN5	10/100 RJ45	(TF)RJ45.12P.90D(F).W/Transformer & LED.DIP
CN6	MINI USB	(TF)MINI USB CONNECTOR R/A 0.8.R/A 0.8mm.5P.90D(F)
CN7	DUAL USB	(TF)USB CONNECTOR DUAL PORT.8P.90D.(F).for USB2.0
CN8	DUAL USB	(TF)BOX HEADER.5*2P.180D.(M).2.00mm.Narrow Frame.DIP
CN9	GPIO	(TF)BOX HEADER.10*2P.180D(M).DIP.2.0mm.Narrow Frame
CN10	ZIGBEE / ENERGY SPI or UART MODULE	(TF)BOX HEADER.10*2P.180D.(M).2.54mm.
CN11	I2C	(TF)WAFER BOX.4P.180D.(M).2.0mm.W/LOCK DIP

CN12	Micro-SD Card	(AOH)(TF)Micro SD SKT.8P.90D(F).SMD.Push-Push type
CN14	Serial Port RS232/RS485/RS422	(TF)D-SUB CONNECTOR.9P.90D(M).DIP.Green.
CN15	Serial Port RS232/RS485/RS422	(TF)WAFER BOX.9P.180D(M).DIP.1.25mm.
CN16	DC Input	(TF)DC Power Jack.3P.90D(F).
CN17	DC Input	(TF)WAFER BOX.2*1P.180D(M).DIP.3.0mm.
CN18	Power LED	(TF)WAFER BOX.2P.180D.(M).2.0mm.W/LOCK DIP.
CN36	Micro-SDLED	(TF)PIN HEADER.2*1P.180D.(M).2.0mm.DIP
CN20	Full MiniPCIE	(TF)MiniCard SLOT.52P.90D.(F).SMD
CN21	HalfMini PCIE	(TF)MiniCard SLOT.52P.90D.(F).SMD
J1	RESET	(TF)WAFER BOX.6P.180D(M).2.0mm.W/LOCK DIP.
J2	SPI Flash	(TF)PIN HEADER.4*2P.180D.(M).1.27mm.SMD.W/Cap.

2.3 Connecting to Target System

The platform is designed as a headless device and does not support KVM (Keyboard, Video, Mouse). You must connect remotely by using serial interface (RS-232 or RS-485) see section 2.3.1.

Install the following packages on your Linux system:

For Ubuntu 64-bit (validated in this release):

sudo apt-get install build-essential gcc-multilib vim-common uuid-deviasl subversion autoconf

2.3.1 Serial Connection

To update the firmware on the target, it is necessary to connect to the target a terminal emulator using the provided serial cable. The example below assumes you are using Putty.

1. Connect the RS-232 debug console port connector on the platform to the host computer, using the provided 3.5 mm to DB-9 cable and optional DB-9 to USB adapter.
2. Turn on the platform. A device is created: /dev/ttyUSB0
3. Run the terminal emulator on the host computer using one of the following:

sudo putty &

or

gksudo putty to run Putty as root

or

sudochmod 666 /dev/ttyUSB0

Use the following settings:

a. Speed = 115,200

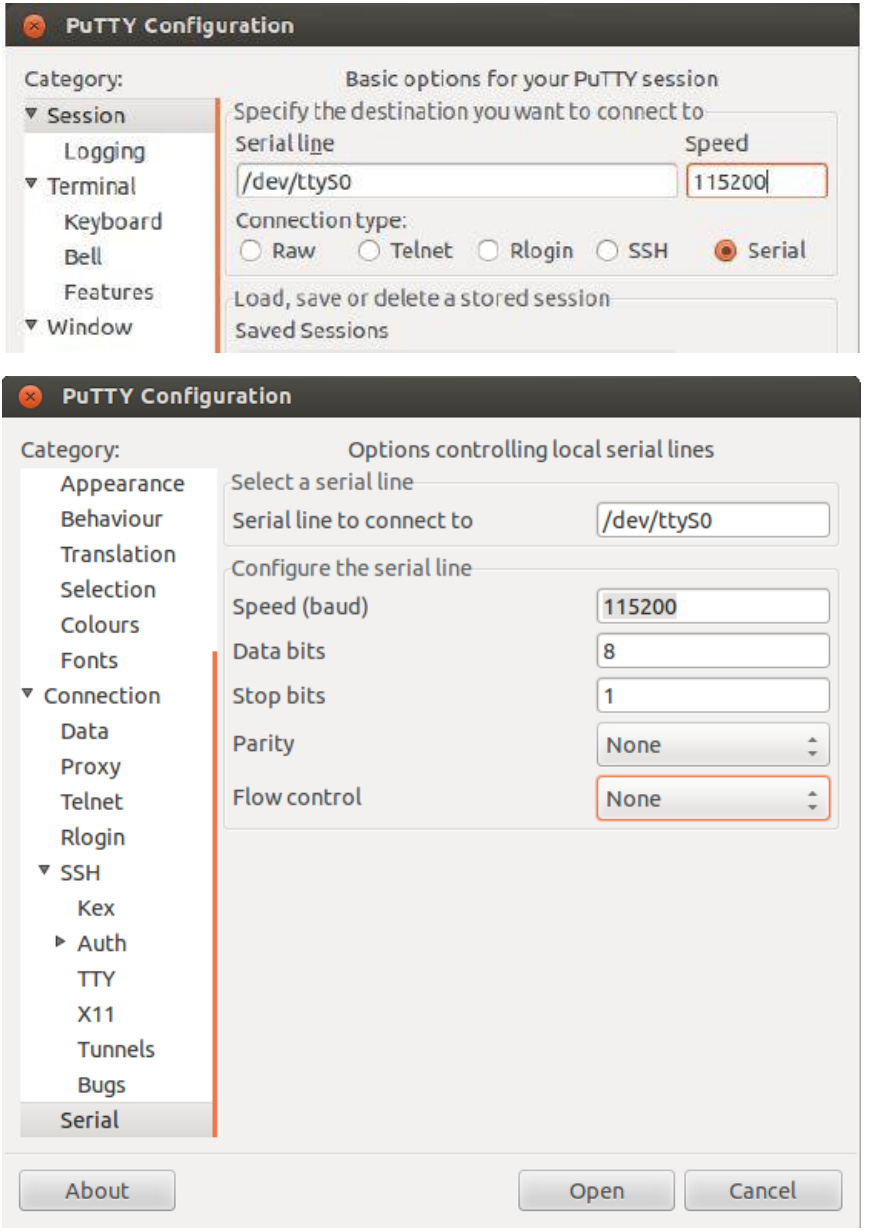
b. Data Bits = 8

c. Parity = None

d. Stop Bits = 1

e. Flow Control = None

f. Preferred emulation mode is ANSI


PuTTY Configuration
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Basic options for your PuTTY session
Specify the destination you want to connect to
Serial line
Speed
/dev/ttyS0
115200
Connection type:
Raw
Telnet
Rlogin
SSH
Serial
Load, save or delete a stored session
Saved Sessions
PuTTY Configuration
Category:
Appearance
Behaviour
Translation
Selection
Colours
Fonts
Connection
Data
Proxy
Telnet
Rlogin
SSH
Kex
Auth
TTY
X11
Tunnels
Bugs
Serial
Options controlling local serial lines
Select a serial line
Serial line to connect to
/dev/ttyS0
Configure the serial line
Speed (baud)
115200
Data bits
8
Stop bits
1
Parity
None
Flow control
None
About
Open
Cancel

4. Attach the proper AC adaptor plug to the external power supply.
5. Plug the 2.1mm circular connector on the power supply into the platform 5V DC input. On each of the LAN ports, one LED will be lit.
6. The platform will start the boot process. Progress can be observed on the host computer terminal emulator.

Continue with the procedures in this document to set up the software.

Chapter 3

Customizing Platform Image

3.1 Initializing Build Environment

From the root directory of your Source Directory, initialize your environment and provide a meaningful Build Directory name:

$ source oe-init-build-env mybuilds

For more about the BSP **(** Board support package **)** setting , please refer to the Intel® Quark™ SoC X1000 BSP Build and Software User Guide

3.2 Adding Layers

To add a layer, you basically need to add its path to the BBLAYERS variable, which is in the$BUILDDIR/conf/bblayers.conf file

example :

```
...
BBLAYERS ?= " \
  /home/User/QuarkBSP/meta-clanton_v1.1.0-dirty/meta \
  /home/User/QuarkBSP/meta-clanton_v1.1.0-dirty/meta-iot-devkit \
  /home/User/QuarkBSP/meta-clanton_v1.1.0-dirty/meta-yocto \
  /home/User/QuarkBSP/meta-clanton_v1.1.0-dirty/meta-quark-bsp \
  /home/User/QuarkBSP/meta-clanton_v1.1.0-dirty/meta-galileo \
```

3.3 Layers structure

–<layer dir>/classes

–<layer dir>/conf

–<layer dir>/recipes-<category1>

–<layer dir>/recipes-<category…>

where<layer dir> is usually the layer name (e.g.,*meta-quark-bsp*)

3.4 Add Package

To add a package, you basically need to add item to the "IMAGE_INSTALL_append" variable, which is in the$BUILDDIR/conf/local.conf file

example :

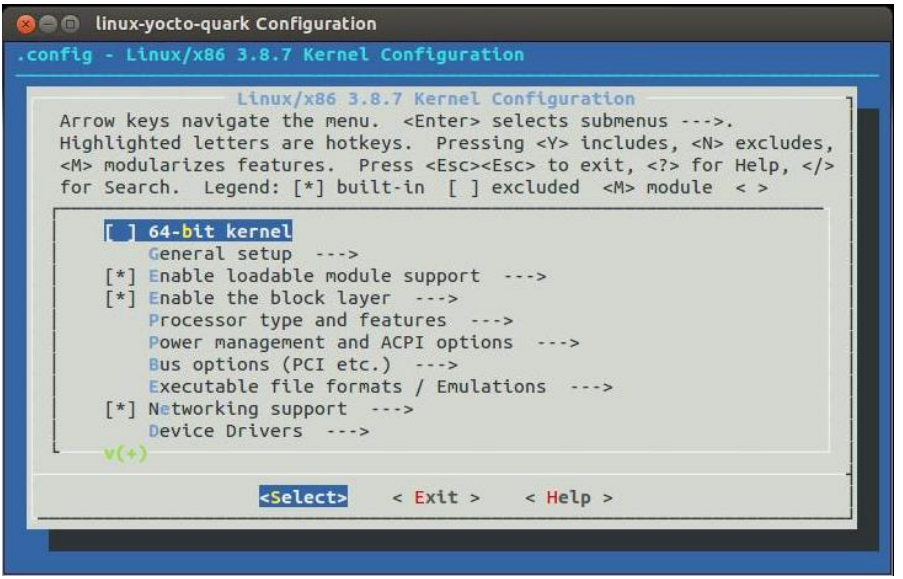
...

IMAGE_INSTALL_append = " minicom trousers packagegroup-tpmtpm-tools"

3.5 Modifying the Linux Kernel

bitbakelinux-yocto-quark -c kernel_configme –f

bitbakelinux-yocto-quark -c menuconfig



bitbakelinux-yocto-quark -c compile -f

bitbakelinux-yocto-quark -c deploy

3.6 Building the Image

bitbakelinux-yocto-quark

Once you have your image, you can take steps to load and boot it on the target hardware. You can learn about BitBake in general by reading the BitBake User Manual. Please refer to the Yocto Project Mega-Manual for details.

Chapter 4

Quark™ SoC X1000 Drivers

4.1 Overview

System on a Chip in the context of AAEON Quark™ SoC X1000 refers to peripheral hardware south of the host bridge interface. SoC software drivers bind the hardware interfaces into standard Linux* sub-systems. Linux* kernel baseline of 3.8.7 (or higher) is required to ensure proper integration and compatibility of upstream reused kernel drivers.

4.2 Hardware Interface and Drivers

The table below lists the hardware interface implemented on AAEON Quark™ SoC X1000 and identifies whether the associated driver is one of the following:

- Standard (unmodified), off-the-shelf driver
- Modified version of off-the-shelf driver, enhanced to enable AAEON Quark™ SoC X1000 specific features

Note: Refer to the software sources to determine the complete list of modified or added files as compared to the Linux* kernel baseline 3.8.7.

- Created to be AAEON Quark™ SoC X1000 specific

AAEON Quark™ SoC X1000 Hardware Interfaces and Drivers

Hardware Interface	Standard Linux* Driver	Modified Linux* Driver	AAEON Quark™ SoC X1000 Specific Driver
USB OHCI Controller Interface	X		
USB 2.0 EHCI Controller Interface	X		
USB Device Interface		X†	
SD/MMc Controller Interface	X		
UART + DMA Interface		X†	

SPI Master Interface		X	
I^2C Master Interface	X		
I^2C/GPIO Interface			X
Ethernet Interface		X	

† PCI vendor/device identifiers added for AAEONQuark™ SoC X1000.

NOTE :Refer to the Software Developer's Manual for Linux guide X1000 Drivers section for details.

4.3 Expansion Drivers

This section describes drivers that are included with the Intel® Quark™ SoC X1000 Software package to enable board-specific functionality.

- AD7298 Driver
- Bluetooth* Driver (requires mini-PCIe card)
- Wi-Fi* Driver (requires mini-PCIe card)
- 3G Modem Driver (requires mini-PCIe card)

4.3.1 AD7298 Driver

The Analog Devices* AD7298 is a 12-bit, low power, 8-channel, successive approximation ADC with an internal temperature sensor. The LS-ADC does not provide a user-space interface directly, it is provided by the IIO subsystem in the Linux* kernel.

The ADC registers with the IIO subsystem as an IIO ADC device driver. As such, it makes calls to functions on the IIO kernel API and provides callbacks which can be used by the IIO subsystem to invoke driver operations.

To load the drivers for the AD7298, perform the following sequence:

- Enable GPIO driver:
  modprobeintel_qrk_gip
  modprobegpio_sch
- Enable IIO support:
  modprobeindustrialio
- Enable SPI driver:
  modprobe spi-pxa2xx
- Enable AD7298 driver:
  modprobe ad7298

After the driver loading sequence is complete, the AD7298 driver enables the following data points via the Industrial I/O (IIO) kernel API directly read from the ADC chip. Refer to the Software Developer's Manual for Linux guide**AD7298 Driver** section for details.

4.3.2 Bluetooth* Driver

Bluetooth functionality is provided by a mini-PCIe card connected to the mini-PCIe slot on the platform. The following cards have been validated with the AAEON Quark™ SoC X1000 Software:

- Intel® Centrino® Wireless-N 135 card
- Intel® Centrino® Advanced-N 6205 Wi-Fi Radio Module (Dual Band Wi-Fi, 2.4 and5 GHz)

The following drivers must be loaded to enable USB-bluetooth components:

modprobeehci-hcd

modprobeohci-hcd

modprobeehci-pci

modprobebtusbl

Once loaded, the sysfsentry below should appear:

/sys/module/Bluetooth

The following user-space components are required:

bluetoothd

hciconfig

hcitool

Refer to the Software Developer's Manual for Linux guide **Bluetooth* Driver** section for details.

4.3.3 Wi-Fi* Driver

Wi-Fi functionality is provided by a mini-PCIe card connected to the mini-PCIe slot. The Intel® Centrino® Advanced-N 6205 Wi-Fi Radio Module (Dual Band Wi-Fi, 2.4 and 5 GHz) has been validated with the AAEON Quark™ SoC X1000 Software.

- To load a driver for the Intel® Centrino® Advanced-N 6205 Wi-Fi Radio Module, type the following command:

modprobeiwlwifi

After a successful load of this driver, the following sysfspath is available:

/sys/class/net/wlp2s0

- Enable/Disable WLAN Radio
- Get the index of the device

Rfkill list

- Disable Radio

Rfkill block 0

- Enable radio

Rfkill unblock 0

- Scan for Wi-Fi Networks

Iwlist wlp2s0 scan

- Edit wpa_supplicant.conf (*NOTE：depend on each Wi-FI AP parameter*)

vi /etc/wpa_supplicant.conf

network={ ssid="Wi-Fi AP ssid"

key_mgmt=WPA2-PSK

proto=WPA2

pairwise=CCMP

group=CCMP

psk="PASSWORD" }

- Apply the configuration

wpa_supplicant -B -i wlan0 -c /etc/wpa_supplicant.conf–B

- Set network parameter

ifconfig wlan0 192.168.11.198

route add default gw 192.168.11.1

- Check Network interface status

Ping 168.95.1.1

Refer to the Software Developer's Manual for Linux guide **Wi-Fi* Driver** section for details.

4.3.4 3G Modem Driver

GSM/3G communications functionality can be provided by a mini-PCIe card connected to the mini-PCIe slot. The Telit* HE910 mini-PCIe module (specifically, the functionality for GSM Voice and SMS communications, and HSPA+ data communications) has been validated with the Intel® Quark™ SoC X1000 Software.

Driver Requirements:

- Telit* HE910 requires USB2.0 support in kernel
- Telit* HE910 requires PPP (point-to-point protocol) support in kernel
- Use of active GPS antenna needs external circuit for powering antenna's amplifier

Software tool requirements:

- minicom- for running scripts
  Can be compiled as ipk package
- microcom- handy for executing simple AT commands
  Microcom is a part of busybox package.
  If it is not installed, it can be enabled in yocto using the command:
  bitbakebusybox -c menuconfig
  then re-installed as ipk package.
- pppd - Point-to-point protocol
  ppp is used for data packet connection. It can be enabled in yocto as an imagefeature "ppp"

To load the drivers, perform the following sequence:

- Enable USB controllers:
  modprobeehci-hcd

modprobeohci-hcd

modprobeehci-pci

- Enable Communication Device Class Abstract Control Model interface:

  modprobecdc-acm

- Mount cdc-acm module

  modprobecdc-acm.ko

- Test 3G-modem by AT command

  cat /dev/ttyACM0 &

  echo ?嫂n "ATE0\r" > /dev/ttyACM03

  echo ?嫂n "AT\r" > /dev/ttyACM0

  if 3G-modem work successfully, it will show "OK"

Refer to the Software Developer's Manual for Linux guide 3G Modem Driver section for details.