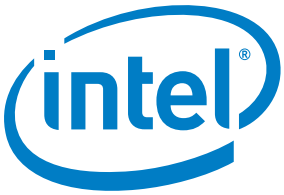

Intel Atom® Processor E3900 Series/ Intel® Celeron® Processor N3350/ Intel® Pentium® Processor N4200 Board Support Package for Yocto Project*

Release Notes

MR3.1 Release

August 2017

Document Number: 333732-006US

Contents

Figures

Tables

Revision History

Date	Revision	Description
August 2017	006	Maintenance Release 3.1 (MR3.1)
June 2017	005	Maintenance Release 3 (MR3)
April 2017	004	Maintenance Release 2.2 (MR2.2)
February 2017	003	Maintenance Release 2 (MR2)
December 2016	002	Maintenance Release 1 (MR1)
August 2016	001	Initial Release (Gold)

§

1.0 Introduction

Intel's Board Support Packages (BSPs) provide some tune options and generic hardware support to cover most current Intel CPUs and devices. Intel's Yocto Project*-based BSP for the Intel Atom® processor E3900 Series/Intel® Celeron® processor N3350/Intel® Pentium® processor N4200 is available for testing/evaluation and project-based software development.

1.1 Terminology

Table 1. Terminology

Term	Description
API	Application Programming Interface
BKC	Best-Known Configuration
BSP	Board Support Package
DDX	Device Dependent X
DMA	Direct Memory Access
DPMS	Display Power Management System
DRI	Direct Rendering Infrastructure
DRM	Direct Rendering Manager
DRRS	Display Refresh Rate Switching
DSI	Display Serial Interface
ECC	Error Checking and Correction
eDP*	Embedded DisplayPort
eMMC*	Embedded Multi-Media Card
FDK	Firmware Development Kit
GPIO	General-purpose input/output
GUI	Graphical User Interface
HDCP*	High-bandwidth Digital Content Protection
HDMI*	High-Definition Multimedia Interface
HID	Human Interface Device
HPET	High-Performance Event Timer
I^2C	Inter-Integrated Circuit
IIO	Industrial Input/ Output
IOCTL	Input/ Output Control

Term	Description
IoT	Internet of Things
LPC	Low Pin Count
LPE	Low-Power Engine
LPSS	Low-Power subsystem
LTS	Long-Term Support
OS	Operating System
PCIe*	Peripheral Component Interconnect Express
PIO	Programmed Input/ Output
PSR	Panel Self-Refresh
PWM	Pulse Width Modulation
RAM	Random Access Memory
RTC	Real Time Clock
SDIO*	Secure Digital Input Output
SMBus	System Management Bus
SoC	System-on-Chip
SPI	Serial Peripheral Interface
SSH	Secure Shell
UART	Universal asynchronous receiver/transmitter
USB	Universal Serial Bus
VPP	Video Post Processing

1.2 Terminology Update

Intel is phasing out the terms “master” and “slave.” Therefore, in this document, these terms are replaced with the words “source” and “sink” in descriptive text only.

These changes do not apply to the register or signal names themselves. A register or signal name that in the past included either the word “master” or a variation of the word “master,” or the term “slave,” or a variation of the word “slave,” has not changed.

For example,

- Before:
  - Master Mode Mono, and Stereo Playback with TLV320AIC3107 Codec
  - Slave Mode Mono, and Stereo Playback with TLV320AIC3107 Codec
- After:
  - Source Mode Mono, and Stereo Playback with TLV320AIC3107 Codec
  - Sink Mode Mono, and Stereo Playback with TLV320AIC3107 Codec

1.3 Intended Audience

This release note is for users of the BSP for Yocto Project* for Intel Atom® processor E3900 Series/Intel® Celeron® processor N3350/Intel® Pentium® processor N4200.

1.4 Customer Support

Contact your Intel representative for support or submit an issue to http://premiersupport.intel.com.

1.5 Reference Documents

Table 2. Reference Documents

Document	Document No./Location
Overrun and Underrun Issue in USB2.0	570645
Intel Atom® Processor E3900 Series, Intel® Celeron® Processor N3350, and Intel® Pentium® Processor N4200, Formerly Apollo Lake RDC Website	https://www.intel.com/content/www/us/en/embedded/products/apollo-lake/overview.html

Document	Document No./Location
Intel® Platform Security Discovery for Intel Atom® Processor E3900 Series Intel® Celeron® Processor N3350 and Intel® Pentium® Processor N4200 Release Notes	573486

§

2.0 BSP Release Notes

2.1 Hardware and Software Compatibility

Hardware

Intel Atom® processor E3900 Series/ Intel® Celeron® processor N3350/ Intel® Pentium® processor N4200.

Software

BSP MR3.1 Release, Kernel v4.1.42

Jethro* v2.0.3

2.2 Release Content

Refer to the Best known Configuration (BKC) for Intel Atom® processor E3900 Series/ Intel® Celeron® processor N3350/ Intel® Pentium® processor N4200 on the Apollo Lake RDC website for details.

§

3.0 Component Release Notes

3.1 I/O and Kernel

3.1.1 Introduction

This section contains general release information for I/O and kernel components for the Yocto Project*.

3.1.2 New Features

None

3.1.3 Product Features

Supported I/O and Kernel features.

- Storage: Serial Peripheral Interface (SPI) NOR Flash, Embedded Multi-Media Card (eMMC*), SD* card, SATA*, Universal Serial Bus (USB) 2/3 host, USB device
- System: Real Time Clock (RTC), thermal, High Performance Event Timer (HPET), 8253 timer, watchdog
- Low Power Sub-System (LPSS): Universal asynchronous receiver/transmitter (UART)/High Speed UART (HSUART), Inter-Integrated Circuit (I^2C), SPI
- Memory: Error Checking and Correction (ECC)
- Power Management: S3, S4, S5, Intel® P-state driver, S0ix
- Connectivity: Gigabit Ethernet
- Miscellaneous: Low Pin Count (LPC), Peripheral Component Interconnect Express (PCIe*), System Management Bus (SMBus), general-purpose input/output (GPIO), SDIO*, Pulse Width Modulation (PWM), IOSF-SB
- S0ix power management – Enabled the sleep model s0ix (Refer to Section 3.1.6, Known Issues).
- S0ix telemetry driver – This driver provide insight to IOSS and PSS IPs power status and s0ix residency for debug purpose.
- Universal Serial Bus (USB) dual role hardware detection – With platform hardware support, the detection of USB host or device connected is switched automatically.
- USB dual role default mode – Driver module parameter to enable default to host or device configuration during boot up.

- LPSS SPI Programed Input/ Output (PIO)/Direct Memory Access (DMA) transfer threshold configuration – The threshold to use PIO or DMA can be configured through board file.
- LPSS I²C* timeout setting configuration - Added new Input/ Output Control (IOCTL) interface for I²C controller timeout configuration.
- LPSS I²C speed mode configuration – The speed mode configuration is now done through BIOS. This timing parameter for different speed mode is also control by BIOS, which was previously hard-coded in driver.
- LPSS HSUART full-duplex support – Verified driver full-duplex support.

3.1.4 Changes to Existing Features

None

3.1.5 Unsupported Features

None

3.1.6 Known Issues

Table 3. I/O and Kernel – Known Issues

Reference Number	Issue
1504458971†	Fail to change speed and Duplex of network using 6–33 firmware
1504472452†	Unable to enter autonomous s0ix

† Issues will be fixed in the next kernel version.

3.1.7 Fixed Issues

Table 4. I/O and Kernel – Fixed Issues

Reference Number	Issue
1405644921	SATA drive constantly spins up/down
1504394551	LEDs on the board still on for few minutes after execute sleep and freeze commands in the background process
1504483098	PWM Power Management shows Active all the time
1504484339	Failed to build image with TFM Governor enabled

3.1.8 Workarounds

Table 5. I/O and Kernel – Workarounds

Reference Number	Observation	Workaround
1504339522	Unable to go into S0ix state when PCIE Ethernet card is connected to the SUT	The driver for PCIE Ethernet card must support S0ix. Check with the PCIE Ethernet card vendor for more details if S0ix feature is required.
1405644921	SATA drive constant spin up and down	Change the autosuspend delay ms in /etc/udev/rules.d/pci_pm.rules to a higher value.

3.2 Graphics

3.2.1 Introduction

This section contains general release information for the Internet of Things (IoT) Graphics and Media Driver on Intel Atom® processor E3900 Series/Intel® Celeron® processor N3350/Intel® Pentium® processor N4200 for Yocto Project*. Graphics are derived from open-source i915 Direct Rendering Manager (DRM) and i965 DRM together with some proprietary components in the RPM (tar ball) format, for example, Intel® Unified 3D Library, Intel® Media SDK.

3.2.2 New Features

- HEVC Macroblock and QP control: The video driver will have the ability to control Macroblock and QP setting.

3.2.3 Product Features

- OpenGL ES 3.1 through Intel® Unified 3D Library Dynamic Range Increase (DRI). With this, MESA library usage is not on POR for broad market.
- i915 atomic interface support.
- i915 atomic: Non-blocking nuclear pageflip.
- Fastboot modeset timing. For example, i915 driver shall complete full mode set within 20ms in single display, 35ms within dual displays and 55ms or less for triple displays configuration.
- Gamma through i915 driver DRM plane property support.
- DisplayPort* brightness Application Programming Interface (API) interface.
- Splash Screen support with capabilities; for example, image data, image quality, scaling, multiple displays, config displays, config destination size, minimize time to

display the splash screen, `request_firmware` interface, and no format conversion on splash screen image.

- Hibernation and resume including during 3D and Video.
- Standby and resume including during 3D and Video.
- Prioritized GPU task scheduler.
- Media/Video:
  a. Intel® Media SDK support. Refer to Intel® Media SDK release note for more detail.
     i. HEVC/H.265 8-bit encode (Intel® Media SDK)

  Note: Refer to Section 3.2.6 Known Issues for limitation details.

     ii. VP8 2160p decode (Intel® Media SDK)
     iii. X11 DRI3/Present Extension (Intel® Media SDK)
     iv. Advance deinterlacing (Intel® Media SDK)
     v. GStreamer decode plug-in (Intel® Media SDK)
     vi. GStreamer sink plug-in (Intel® Media SDK)
     vii. GStreamer Video Post Processing (VPP) plug-in (Intel® Media SDK)
  b. Open-source technology video acceleration:
     i. HEVC/H.265 10-bit decode
     ii. Skin tone detection
     iii. Sharpening
     iv. Up/Down scaling
     v. Denoise
     vi. PRIME buffer sharing
     vii. Advance Deinterlacing

- Updated i915, i965, DRM, and Device Dependent X (DDX) drivers
- Display – single High-Definition Multimedia Interface (HDMI*), DisplayPort*, Embedded DisplayPort (eDP*)
- Display - Multi-displays, rotation, scaling, centering
- Display – plane color key, blending
- Display Power Management System (DPMS), ACPI
- 3D – OpenGL 3.3, OpenGL ES 3.0
- Decode - H264, MPEG2, VC1, JPEG2, VP8, HEVC 8bit, [M]JPEG
- Encode – H264, [M]JPEG
- Video processing, color conversion
- RC6*, Turbo, Display Refresh Rate Switching (DRRS), Panel Self-Refresh (PSR)
- High-bandwidth Digital Content Protection (HDCP*) 1.4

- GStreamer plug-in (decode and sink)
- eDP1.3
- HDCP 1.4
- Several DDX features
- GPU-based XVideo adapters
- Frame Packing Stereoscopic 3D
- Frame Sequential Stereoscopic 3D
- Top-bottom Stereoscopic 3D
- HDCP* daemon support for multiple clients
- Nano libva
- ProcAmp - i965 video driver provides VPP method to adjust ProcAmp values for example, Brightness, Contrast, Saturation, and Hue.
- Patch for libva stolen memory
- DRM forklift from 4.9+
- HEVC 8-bit encode
- Intel® Media SDK composition supports more than 64 Channel inputs
- Intel Media SDK VPP support 8K resolution
- SFC (Scaler and Format Converter) Enabling
- AVC Macroblock and QP control
- AVC Region of Interest (ROI) control
- HEVC Region of Interest (ROI) control
- EGL_MESA_image_dma_buf_export: This extension provides entry points for integrating EGLImage with the dma-buf infrastructure. The extension allows creating a Linux dma_buf file descriptor or multiple file descriptors, in the case of multi-plane YUV image, from an EGLImage
- DRI Image Extension
- Multi-stream: The i915 drivers shall support the DisplayPort 1.2 multi-stream function when used with display devices that also support the DP 1.2 multi-stream capability.
- P-state to Frequency: The i915 driver shall export the following function that can be used by other kernel drivers to convert a P-state value to a frequency.
- int i915_gpu_pstate2freq(int pstate):
- Max P-state Support.

- Enable turbo: The i915 driver shall export the enable turbo function that can be called by other kernel drivers to disable the GPU frequency limit that was enabled by the disable turbo function.
- Disable turbo: The i915 driver shall export the following function that can be called by other kernel drivers to limit the internal maximum GPU frequency to RPe.
- Register callback: The i915 driver shall export the following function that can be used by other kernel drivers to register a frequency notification callback function.
- Unregister callback: The i915 driver shall export the following function that can be used by other kernel drivers to remove a registered frequency notification callback function
- Frequency notification callback: The i915 driver shall notify registered clients whenever the GPU frequency changes.
- Extend P-state and i915 driver to provide callbacks enabling kernel space CPU/GPU governor implementation

3.2.4 Changes to Existing Features

None

3.2.5 Unsupported or Discontinued Features

- Text Tuning
- MPEG2 encode
- Shared Virtual Memory
- Display Configuration - genlock
- Genlock multi-pipe
- Display detection override
- CRTC list
- GStreamer VA API multi-stream decode with H.264
- GStreamer VA API file based Raw YUV as input encode

3.2.6 Known Issues

Table 6. Graphics – Known Issues

Reference Number	Description
1504340500	Rendercheck triangles test fails

Reference Number	Description
1504340562	Rendercheck gradients test fails
1504229313	Gstreamer VAAPI sharpen element error with minimum and maximum range of value
1504229597	Video upscaling seen in X11 matchbox when Gstreamer VAAPI rotation 90 and 270 with force aspect ratio
1504290865	CPU pipe a FIFO underrun messages observed irregularly
1504393332	MIPI JDI boots up with no display on it.
1504396649	IGT kms_planeblend subtest fails.
1504397843	Hardware acceleration is disabled when perform CSC and scaling for 8K YV12 with MSDK
1504438443	Gstreamer fails to transcode h264 and h265 file on certain resolution.
1504451506	Output of transcoding with mfxh264enc and qtmux is unable to play.
1504464451	20% exceeding expected bitrate for HEVC in BRC
1504493262	64Channel SMT performance regression between ww16 UMD+MR2 MSDK and MR2
1504513901	IGT Subtests in kms_frontbuffer_tracking fails
1504513965	H264 Encode including B-frame will gets corrupted output
1504514747	IGT Subtests in pm_rpm fails
1504515139	System Hangs while running rendercheck
1504515274	Libva version is not shown in vainfo
1504527090	out-of-bounds access in i915
1504542143	128CH video wall on dual 4k display kernel warning when decoding is done
1805325009	Samples could not be built on a non-X11 environment

3.2.7 Fixed Issues

Table 7. Graphics – Fixed Issues

ID	Title
1804338136	DRM does not expose all universal planes - It only exposes cursor plane instead
1504410661	IGT Subtest - stolen-hibernate of gem_stolen failed
1504438279	IGT subtest- kms_plane_scaling fails
1504478931	Encoding perfomance reuces to 20% compared with previous release.
1504495516	20xH264 Video Decode Render-less CPU Usage spike up to ~70% with flag sync=false
1504496316	Fixed for HEVC decoder with 32 aligment issue

3.3 Audio

3.3.1 Introduction

This section contains general release information for audio. Refer to the Audio User Guide for installation information on HD Audio and I2S* Audio.

3.3.2 Product Features

Table 8. Audio – Product Features

I/O Component	Summary of Feature	Feature Availability
HD Audio	48kHz, Stereo HD Audio playback through onboard HD Audio Codec	Yes
	48kHz, Stereo HD Audio capture through onboard HD Audio Codec	Yes
	HDMI Audio Playback	Yes
	HTML5 Audio Playback	Yes
	DisplayPort Audio Playback	Yes
	1 HDMI and 1 DisplayPort Audio Playback	Yes
	2 HDMI Audio Playback	Yes
	2 DisplayPort Audio Playback	Yes
	Power management for HDMI Audio, DisplayPort Audio, HDA Codec	Yes
I^2S* Audio	I2S, 48kHz, Source Mode Stereo Playback with Dummy Codec (Refer to Section 1.2 for Terminology Update)	Yes

I/O Component	Summary of Feature	Feature Availability
	I2S, 48kHz, Source Mode Stereo Capture with Dummy Codec (Refer to Section 1.2 for Terminology Update)	Yes
	I2S, 48kHz, Source Mode Stereo Playback with WM8731 Codec (Refer to Section 1.2 for Terminology Update)	Yes
	I2S, 48kHz, Source Mode Stereo Capture with WM8731 Codec (Refer to Section 1.2 for Terminology Update)	Yes
	I2S, 48kHz, Source Mode Mono, and Stereo Playback with TLV320AIC3107 Codec (Refer to Section 1.2 for Terminology Update)	Yes
	I2S, 48kHz, Source Mode Stereo Capture with TLV320AIC3107 Codec (Refer to Section 1.2 for Terminology Update)	Yes
	I2S, 48kHz, Sink Mode Mono, and Stereo Playback with TLV320AIC3107 Codec (Refer to Section 1.2 for Terminology Update)	Yes
	I2S, 48kHz, Sink Mode Stereo Capture with TLV320AIC3107 Codec (Refer to Section 1.2 for Terminology Update)	Yes
	Power management	Yes
	ACPI NHLT Table	Yes
	Ease use for customer feature	Yes
FDK tools	Topology binary customization across SSP1-6 with following parameters: Sampling rate: 8khz, 16khz, 24khz, 32khz, 48khz, 96khz, 192khz Channel number: 1ch, 2ch, 4ch, 8ch Bit format: 16/16, 16/32, 24/24, 24/32, 32/32	Yes
	Blob file customization for various sampling rate, channel number, bit format	Yes

3.3.3 New Features

FDK (Firmware Development Kit) tools:

- Topology binary customization across SSP1-6 with following parameters:
- Sampling rate: 8khz, 16khz, 24khz, 32khz, 48khz, 96khz, 192khz
- Channel number: 1ch, 2ch, 4ch, 8ch
- Bit format: 16/16, 16/32, 24/24, 24/32, 32/32

3.3.4 Mandatory BIOS Settings

1. **Mandatory BIOS settings for HD Audio**

   `DEVICE MANAGER > SYSTEM SETUP > SOUTH CLUSTER CONFIGURATION > HD AUDIO CONFIGURATION > HD-AUDIO I/O BUFFER OWNERSHIP=` HD Audio Link owns all the I/O buffers

2. **Mandatory BIOS settings for I²S**

   `DEVICE MANAGER > SYSTEM SETUP > SOUTH CLUSTER CONFIGURATION > HD AUDIO CONFIGURATION > HD-AUDIO I/O BUFFER OWNERSHIP=`I²S port owns all the I/O buffers

3.3.5 Known Issues

Table 9. Audio – Known Issues

Reference Number	Issue
1504169268	[HDA] Audio record overrun. Refer to Section 3.3.7, Limitation
1504484785	Intel FDK tool - fail to support 44.1khz, 22.05khz, and 6 channel related audio. Refer to Section 3.3.7, Limitation

3.3.6 Fixed Issues

Table 10. Audio –Fixed Issues

Reference Number	Issue
1504485532	Intel FDK tool - blob generator fail to support audio configuration with BCLK with 50% duty cycle

3.3.7 Limitation

- [1504484785] FDK no to support 44.1khz mono channel and also 6 channel configuration across 24kHz, 48kHz and 96kHz.
- For TLV320AIC3107 codec to work in Low-Power Engine (LPE) Audio, rework is needed. Refer to the audio user guide for details.
- HD Audio and SSP cannot co-exist in one bzimage.
- [1504169268] It is not recommended to run audio play/record on the same USB2.0 USB flash drive that boots up the board. Refer to the application note titled *Overrun and Underrun Issues in USB2.0*.

3.4 Intel® ISP Firmware and Driver

3.4.1 Introduction

This section contains general release information for Intel® ISP Firmware and Driver. For more information, Refer Intel Atom® Processor E3900 Series, Intel® Celeron® Processor N3350, and Intel® Pentium® Processor N4200, Formerly Apollo Lake RDC Website or contact your intel representative for more details.

3.4.2 New Features

- Add 2M camera OV2740 for general market enabling:
- OV2740 - Preview Enhanced ISP Settings
- OV2740 - Preview AE Speed Control (HAL)
- OV2740 - Preview AE Speed Control (AIQ)
- OV2740 - Preview AWB Speed Control (HAL)
- OV2740 - Preview AWB Speed Control (AIQ)
- OV2740 - Preview vULL

3.5 Intel® Integrated Sensor Solution Utility

3.5.1 Introduction

This document contains general release information for the Intel® Integrated Sensor Solution Utility. Refer to Section 5.2.5 Install Image into On-Board eMMC* for installation information of Integrated Sensor Solution Utility.

3.5.2 New Features

None

3.5.3 Product Features

Table 11. Intel® Integrated Sensor Solution Utility– Product Features

I/O Component	Summary of Feature
Intel® Integrated Sensor Solution Utility	1. Supports accelerometer 3D sensor for Bosch* BMC150 accelerometer & BMA255 acceleration sensors, barometer sensor for Bosch BMP280 barometric pressure sensor, ambient light sensor for Lite-On* AL3010 digital ambient light sensor, and gyrometer sensor for Bosch BMG160 gyroscopes through the Industrial Input/ Output (IIO) interface:

I/O Component	Summary of Feature

	a. Read raw
	b. IIO triggered buffer
	c. Supports polling mode
	2. Supports interrupt mode through IIO interface for accelerometer 3D sensor for Bosch BMC150 accelerometer and ambient light sensor for Lite-On AL3010 digital ambient light sensor.
	3. ISH drivers consist of intel-ish-ipc.ko, intel-ishtp.ko, and intel-ishtp-hid.ko. Guide to unload and reload ISH modules are as below:
	a. To unload all three ISH drivers at once:
	`$ modprobe -r intel-ish-ipc`
	b. To reload all three ISH drivers at once:
	`$ modprobe intel-ish-ipc`
	c. To unload ISH drivers one by one using rmmod:
	`$ rmmod intel-ishtp-hid`
	`$ rmmod intel-ish-ipc`
	`$ rmmod intel-ishtp`
	OR
	`$ rmmod intel-ish-ipc`
	`$ rmmod intel-ishtp-hid`
	`$ rmmod intel-ishtp`
	d. To reload ISH drivers one by one using insmod:
	`$ cd /lib/modules/4.1.27apollolake/kernel/drivers/hid/intel-ish-hid/`
	`$ insmod intel-ishtp.ko`
	`$ insmod intel-ish-ipc.ko`
	OR
	`$ insmod intel-ishtp.ko`
	`$ insmod intel-ishtp-hid.ko`
	`$ insmod intel-ish-ipc.ko`
	4. Supports Soletta* Project framework and sample applications for accelerometer 3D sensor, barometer sensor, ambient light sensor, and gyroscope 3D sensor.
	5. ISH drivers support S0iX and S3 state:
	a. Able to enter S0iX and S3 suspend mode
	b. Able to resume after being suspended
	c. Sensor functionalities are restored after resuming from suspend
	6. Soletta Project sensor sample applications:
	a. Accelerometer sensor sample application using flow-based programming (FBP)
	b. Ambient light sensor sample application using high-level C API programming
	c. Barometer sensor sample application using flow-based programming with MQTT
	d. Gyroscope sensor sample application using traditional C programming with MQTT
	7. Supports Soletta-dev-app.

I/O Component	Summary of Feature
	8. Supports ISH Calibration Tool, ISH Debug Tool, and ISH Manufacturing Tool.

3.5.4 Known Issues

None

3.5.5 Fixed Issues

None

3.5.6 Limitation

3.5.6.1 Sampling Frequency

The IIO Linux* sysfs interface allows users to read and write the sampling frequency of each IIO device.

The unit used for IIO device sampling frequency is Hz. In the Intel® Integrated Sensor Solution Firmware, the sampling frequency is equivalent to the Human Interface Device (HID) Report Interval property. From the HID specifications, the Report Interval value is a 32-bit unsigned integer represented in milliseconds. Hence, during the conversion from Hertz to milliseconds in the IIO driver, the precision of the value is up to milliseconds.

Example 1: User writes 11 Hertz for sampling frequency

User inputs: **11 Hertz** → 90.9090 milliseconds = 90 milliseconds

User reads back: 90 milliseconds → 11.11111 Hertz = **11.1 Hertz** (not 11 Hertz) with the ***precision of 1 decimal number***.

Figure 1. Sampling Frequency Example 1

```
root@intel-corei7-64:~# echo 11 > /sys/bus/iio/devices/iio:device4/in_accel_sampling_frequency
root@intel-corei7-64:~# cat /sys/bus/iio/devices/iio:device4/in_accel_sampling_frequency
11.100000
```

Example 2: User writes 48 Hertz for sampling frequency

User inputs: **48 Hertz** → 20.833 milliseconds = 20 milliseconds

User reads back: 20 milliseconds → **50 Hertz**

Figure 2. Sampling Frequency Example 2

```
root@intel-corei7-64:/sys/bus/iio/devices/iio:device0# echo 48 > in_intensity_sampling_frequency
root@intel-corei7-64:/sys/bus/iio/devices/iio:device0# cat in_intensity_sampling_frequency
50.000000
```

Hence, due to this precision limitation, the sampling frequency value entered by user may not be accurate.

3.5.7 Related Documentation

1. HID Specification
   - USB HID: http://www.usb.org/developers/hidpage/Hut1_12v2.pdf
   - HID for Windows* OS: https://msdn.microsoft.com/en-us/library/windows/hardware/dn613934(v=vs.85).aspx
2. HID Sensor Custom
   - http://lxr.free-electrons.com/source/Documentation/hid/hid-sensor.txt
   - http://lxr.free-electrons.com/source/drivers/staging/iio/Documentation/
3. IIO Sensor
   - http://lxr.free-electrons.com/source/drivers/staging/iio/Documentation/
   - http://lxr.free-electrons.com/source/tools/iio/
4. IIO Generic Buffer Application for Accelerometer
   - http://lxr.free-electrons.com/source/tools/iio/
5. Soletta Project at GitHub:
   - https://github.com/solettaproject/soletta/tree/v1
6. Soletta Project Presentation Slides:
   - https://github.com/solettaproject/soletta/wiki/Presentations
7. Soletta Project Documentations:
   - https://github.com/solettaproject/soletta/wiki/Documentation
8. Soletta Project `c-api`:
   - http://solettaproject.github.io/docs/c-api/
9. Soletta Project Flow Node description:
   - http://solettaproject.github.io/docs/nodetypes/

3.6 Software Development Tools

3.6.1 Introduction of Intel® System Studio

Intel® System Studio provides key tools for embedded and system development, including a C++ compiler, optimization libraries, power and performance analyzers (Intel® SoC Watch and Intel® VTune™ Amplifier),debug and trace capabilities, and more. Visit the following registration page to get the product: https://registrationcenter.intel.com/en/forms/?productid=3006 and refer to the Intel® System Studio product release notes for more details**.**

3.6.2 New Feature

Added the new recipes on adding the Intel® IPP component into the Yocto* target.

The Intel® SoC Watch is updated to v2.3.1. The Intel® VTune version is updated to 2017 update 3. Both are available from Intel® System Studio 2017 update 3 release.

3.7 Intel® Platform Security Discovery

3.7.1 Introduction

Intel® Platform Security Discovery is designed specifically to allow other applications at run time to know what hardware security capabilities are available on the platform. For more information, refer to the PSD release note (CDI #573486) for more details. Alternatively, refer APL RDC Website or contact your Intel representative for more details.

§

4.0 Where to Find the Release

A copy of this release note is available at the GitHub repository for Intel Atom® processor E3900 Series/Intel® Celeron™ processor N3350/Intel® Pentium™ processor N4200.

§

5.0 Getting Started with Board Support Package

5.1 Setting up the Host Machine

The following are the minimum host system configurations to build BSP for the Yocto Project*:

- Intel® Core™ i7 processor (4 cores)
- Linux* OS of choice for Yocto Project* build is Ubuntu* 14.04 LTS OS
- 4 GB Random Access Memory (RAM) and 500 GB disk space
- High-speed network connectivity

Note: To enable the BSP build for Yocto Project*, set up and enable Secure Shell (SSH) keys on your host machine. Refer to the Setup Guide for details.

5.2 Getting Started with BSP for Yocto Project*

Download the BSP for Yocto Project* from GitHub to your host machine

- HTTPS directly from https://github.com/01org/iotg-yocto-bsp-public/tree/e3900/master by selecting the appropriate tag version, for example, E3900-MR3.1, from the top left menu or
- SSH using the following command: git clone https://github.com/01org/iotg-yocto-bsp-public.git -b e3900/master

This git tree is maintained as single product branch. To get code base from the previous release, for example the MR3.1 release, check out to the specific tag `git checkout E3900 MR3.1`.

5.2.1 Default Configuration Set for core-image-sato Image in this BSP

- Meta-intel contains an i915 graphics driver that depends on GStreamer plug-ins. These plug-ins require license flags set to "commercial" to be included in the build. LICENSE_FLAGS_WHITELIST = "commercial" already set by the template in the local.conf for your build.
- To enable full graphics video and display in the image, Intel includes a package group tailored to showcase the graphics capability on this platform. The packagegroup-core-graphics-essential is in meta-intel-middleware. This packagegroup is set to build into core-image-sato by default in this BSP.
- To execute 64-bit standalone applications, you need to enable a multilib environment in your image. The following lines in local.conf are commented out by default. To enable multilib support, remove the “#” in front of these lines.

```
require conf/multilib.conf
DEFAULTTUNE = "corei7-64"
MULTILIBS = "multilib:lib32"
DEFAULTTUNE_virtclass-multilib-lib32 = "corei7-32"
```

- To enable 32-bit libraries into final bootable image, you need to add the following settings in local.conf.

```
IMAGE_INSTALL_append = `` lib32-glib-2.0 lib32-gcc''
```

- The BSP supports FreeGLUT library. However, it is not enabled by default. To enable FreeGLUT library support, you need to add the following line in build/conf/local.conf.

```
IMAGE_INSTALL_append = `` freeglut''
```

5.2.2 Your First Build

1. If this is your first build, run the setup.sh script from your bsp-apollolake-i/ directory:

   $./setup.sh

2. The setup.sh script prompts you with a menu for choice of audio machine driver. The BSP meta layer of the Yocto Project* support 3 MACHINE types. The machine settings in local.conf are updated whenever you select features in setup.sh.

Figure 3. Machine Driver Options

```
Select an option:

1. Build kernel image with CAVS HD Audio driver (Default)

2. Build kernel image with CAVS SSP Audio driver

3. Build kernel image with legacy HD Audio driver

Default option is build kernel image with CAVS HD Audio
driver. If no input is received within 20 secs, default will
be used.
```

3. Once the machine driver has been selected, the script prompts you with another menu for choice of build. By default, the core-image-sato-sdk is selected. Otherwise, you may key in the numerical selection for core-image-sato or linux-kernel as the bzImage, or set up a custom build.

Figure 4. Build Options

```
Select an option:

1. core-image-sato-sdk (Default)
```

```
2. core-image-sato

3. linux-kernel

4. custom

Default build target is core-image-sato-sdk. If no input is
received within 20 secs, default target is built.
```

4. Setup.sh performs the following tasks before building the BSP image for Yocto Project*:
   a. Checks the host machine build environment for the following:
      - Linux* distribution on host machine
      - Required software dependencies (this is only performed for Ubuntu* 14.04 OS)
      - Version of installed Python* programming language, network connectivity, git config settings, and gitproxy settings
   b. Prepares the sources:
      - Downloads Linux Kernel v4.1.42 from Yocto Project.org
      - Applies IOTG Intel Atom® Processor E3900 Series kernel patches
      - Combo layer downloads poky Jethro* v2.0.3 and other meta layers based on setup/combolayer.conf
      - Applies patches to BSP recipes for Yocto Project
      - Sets up the path to local kernel source in Linux kernel recipe
      - Sets up bblayers for BitBake build
      - Sets up local.conf for BitBake build
      - Prepares the environment for BitBake build
      - Starts the BitBake image, builds automatically based on selection

5. This process creates a build folder named "yocto_build" at the same level as your `bsp-apollolake-i/` directory. The BitBake component of Yocto Project is running at this directory: `/yocto_build/build/`. For setup.sh to run completely, this process may take up to 5 hours depending on the performance of your build machine. When the build process is completed, you may browse for the image from the following path:
   `<path>/yocto_build/build/tmp/deploy/images/intel-corei7-64-<machine-drivers>/`
   - HDDIMG image file name: core-image-sato-sdk-intel-corei7-64-<machine-drivers>-<build-date-time>.hddimg
   - ISO image file name: core-image-sato-sdk-intel-corei7-64-<machine-drivers>-<build-date-time>.iso

5.2.3 For Subsequent Build

1. If you need to modify the recipes or configurations, make your customization in the yocto_build folder after running the/setup script on your host machine.
2. The machine settings in local.conf are updated whenever you select features in setup.sh. However, you may change it for your own build testing.

   Default settings:
   MACHINE ??= "intel-corei7-64-cavs-hda"
   - The Linux* kernel source code is patched with the kernel tarball that contains audio code base with CAVS Audio support.
   - Kernel configurations are set to compile CAVS HD-Audio.

   Additional settings:
   MACHINE ??= "intel-corei7-64-cavs-ssp"
   - The Linux* kernel source code is patched with the kernel tarball that contains audio code base with CAVS Audio support.
   - Kernel configurations are set to compile LPE Audio (SSP).

   MACHINE ??= "intel-corei7-64"
   - The Linux* kernel source code is patched with kernel tarball that contains audio code base with legacy audio support.
   - Kernel configurations are set to compile legacy HD-Audio.

     In yocto_build/build/conf/local.conf, can set the MACHINE type and AUDIO_FEATURES to build.
3. When you are ready to rebuild, go to the yocto_build folder to run the following command:

```
$ cd <path to directory>/yocto_build

# When you source in your yocto_build directory, you will be
automatically be routed to the build/ directory
$ source oe-init-build-env

# For core-image-sato
$ bitbake core-image-sato

# For core-image-sato-sdk
$ bitbake core-image-sato-sdk

# For linux-kernel bzIMage only
$ bitbake linux-yocto
```

5.2.4 Integrate and Build Image with ISH Debug and Calibration Tools

1. Download ISH tools RPM compressed as zip file and meta-intel-proprietary layer file from the link mentioned in BKC and put into a staging location.

2. Put RPM zip file to one directory (such as /home/user/rpm-binary) and meta-intel ISH proprietary layer to other path (such as /home/user/proprietary)

3. Follow the steps in Sections 5.2, 5.2.1 and 5.2.2 to build the default image

4. Once image build is successfully completed, go to the yocto_build/ folder.

   $ `cd /home/user/development/yocto_build/`

5. Go to build/conf/ folder.

   `$ cd build/conf/`

6. Edit the bblayers.conf to include meta-intel-proprietary layer.path

   $ `vim bblayers.conf`

   e.g: BBLAYERS = `"/home/user/proprietary/meta-intel-ish \"`

7. Edit the local.conf to include the path where you have placed your downloaded RPM binaries.

   $ `vim local.conf`

8. Add the following line at the top of local.conf file.

   ```
   export RPM_PATH="<full_path_to_where_you_have_placed_the_
   downloaded_rpm>"
   ```

   For example, if you placed your RPM binaries in `/home/user/rpm-binary/`, then in your `local.conf`, insert:-

   ```
   export RPM_PATH="/home/user/rpm-binary"
   ```

9. We have separated the tools into different functional groups, that is, trace tools, update PDT tool, ISSU, and calibration tool. To install the desired tools, add the package name of the tool into `IMAGE_INSTALL_append` list. Edit the `local.conf` to let the compiler know the ISH tool binaries you wanted to install.

   `$ vim local.conf`

10. To try out calibration tool, do the following:

    ```
    IMAGE_INSTALL_append = " ish-tools ish-tools-calibration"
    ```

11. You can further add on the packages to install by customizing in `local.conf`. Always add ish-tools `in IMAGE_INSTALL_append` list to avoid build issue. A list of software packages available for customization based on user preference:

    - **`ish-tools-calibration`**
      - This installs LinuxCalibrationTool and TestingConfig.xml into /usr/bin/.
    - **`ish-tools-issu`**
      - This installs ISSU into /usr/bin/.
    - **`ish-tools-tracetools`**
      - This installs ISSTraceCollector and TraceConfigTool into /usr/bin/.
    - **`ish-tools-updatepdttool`**
      - This installs UpdatePDTTool into /usr/bin/.

12. You are now ready to build your image with ISH tool binaries
    Go up one level above to build/.

    ```
    $ cd ..
    ```

    Prepare environment to run `bitbake` command.

    ```
    $ source ../oe-init-build-env .
    ```

    Start the image compilation.

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

5.2.5 Install Image into On-Board eMMC*

Note: You need a live bootable USB flash drive or hard disk to install the image into the onboard eMMC*. These instructions assume installation into a USB flash drive.

1. Copy the image into the USB flash drive using the "**dd**" command.
2. Assuming the USB flash drive is mounted as `/dev/sdc` on the Linux* host machine, change to the directory where the image is stored and type the following command in the terminal:

   ```
   $ dd if=core-image-sato-intel-corei7-64.hddimg of=/dev/sdc && sync
   ```

3. Plug the USB flash drive into the Intel Atom® Processor E3900 Series platform and choose to boot off the USB flash drive.
4. Choose the "**Install**" option in the Grub menu. Then, choose the correct partition to install your image from the command line interface.

Note: eMMC* should be detected as /dev/mmcblk0.

5. After the installation is complete, remove your USB flash drive and press "**ENTER**" to reboot.

5.2.6 Optional Configuration

For this released image, log on as root without password on the command line interface.

If you want to use the Graphical User Interface (GUI), follow these steps:

 # change directory to /home/root

```
$ cd /home/root
```

 # edit the .xinitrc file as follows:

```
$ vi .xinitrc
```

Comment out the "exec xterm" line and uncomment the "exec matchbox-session" line as follows:

#exec xterm

exec matchbox-session

Save and close. Type the “startx” command in the command line interface.

$ startx

5.2.7 Known Issues (General and BSP for Yocto Project*)

- The HDDIMG image file checksum (MD5SUM) changes after being installed in the USB flash drive using mkefidisk.sh.

  Background:
  The change in the image file checksum is expected because when the image file was mounted and unmounted, some filesystem-related (ext4 in this case) information (for example, the number of times the image was referenced and the last date and time the image was mounted) were updated into the image, which resulted in a different MD5SUM checksum after the image was flashed.

  Solution:

  No fix is required. There is no functional change in the image. The image file checksum is just for reference to ensure the image is not corrupted during the download process.

- [1504212818] Media player unable to play sound files.

  Background:

  The media player was based on the gst-player* that inherits from open source. This issue has a lower priority to be fixed.

  Workaround:

  Invoke command: aplay or gst-play <sound_file> to play a sound file from the terminal.

§