



Intel[®] Quark[™] SoC X1000 Software

Package Version: 1.1

Release Notes

February 2015



You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein
No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

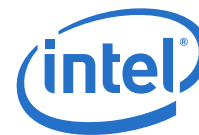
The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting: <http://www.intel.com/design/literature.htm>

Intel, Intel® Quark™ and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2015, Intel Corporation. All rights reserved.

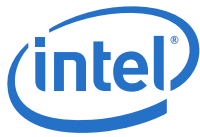


Contents

1.0	Description of Release	6
1.1	New Features in Release 1.1	6
1.1.1	BIOS/Firmware	6
1.1.2	Linux* Operating System (OS)	7
1.1.3	Tools	7
1.2	Features	7
1.2.1	BIOS/Firmware	8
1.2.2	Bootloader	8
1.2.3	Linux* Operating System (OS)	8
1.2.4	OpenOCD	9
1.3	Limitations	9
1.4	Unplanned Functionality	10
1.5	Component Versions	10
1.5.1	Packages	10
1.6	Related Documentation	10
1.7	Licensing	11
2.0	Known Issues	12
3.0	Resolved Issues	22

Tables

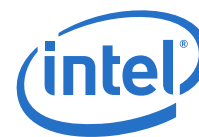
1	Related Documentation	10
2	License Files	11
3	Known Issue Summary	12
4	Resolved Issue Summary	22



Revision History

Date	Revision	Description
February 2015	003	General updates for software release 1.1.0 in the following sections: <ul style="list-style-type: none">• Section 1.1, "New Features in Release 1.1" on page 6.• Section 1.2.2, "Bootloader" on page 8• Section 1.3, "Limitations" on page 9.• Section 1.7, "Licensing" on page 11.• Section 2.0, "Known Issues" on page 12.• Section 3.0, "Resolved Issues" on page 22.
22 May 2014	002	General updates for software release 1.0.1 in the following sections: <ul style="list-style-type: none">• Section 1.1, "New Features in Release 1.1" on page 6.• Section 1.3, "Limitations" on page 9.• Section 1.7, "Licensing" on page 11.• Section 2.0, "Known Issues" on page 12.• Section 3.0, "Resolved Issues" on page 22.
04 March 2014	001	First public release of document.

§ §





1.0 Description of Release

This document describes extensions and deviations from the release functionality described in the documentation for the Intel® Quark™ SoC X1000 (formerly codenamed Clanton).

This release is called: Package Version: 1.1.0

Intel® Quark™ SoC X1000 Software supports the following Form Factor Reference Design boards (FFRDs):

- Customer Reference Boards:
 - Kips Bay (Fab C, green PCB)
 - Galileo Customer Reference Board (CRB), Fab D with blue PCB
 - Galileo (Gen 2) Customer Reference Board (CRB), Gen 2 marking
- Intel® Quark™ SoC X1000 Industrial/Energy Reference Design, “Cross Hill”
- Intel® Quark™ SoC X1000 Transportation Reference Design, “Clanton Hill”
- Intel-only System Validation Platform (SVP), “Clanton Peak”

For instructions on building and running the release software, see the Intel® Quark™ SoC X1000 Board Support Package (BSP) Build and Software User Guide (see [Table 1](#)).

These release notes also include known issues with third-party or reference platform components that affect the operation of the software.

1.1 New Features in Release 1.1

1.1.1 BIOS/Firmware

- Cross Hill boards: TPM support in BIOS
- Adds support for Galileo Gen2 board.
- BIOS changed to boot from external media before SPI flash payload for conformance with classic BIOS boot order
- BIOS will endeavour to recover SPI flash contents without the need for user input via the system console if the platform jumper/strap to force firmware recovery is applied.
- Support for the EDKII VS2008x86 (Microsoft* Visual Studio*) tool chain option has been added.
- BIOS instantiates ACPI Device objects for on-board I²C and SPI devices (a new BIOS build requirement for the ACPI5.0 compatible iasl compiler).
- Galileo (Fab D, blue PCB) specific: BIOS instantiates the ACPI GPIO Client Device object for Intel® Quark™ SoC GPIOs and Cypress IO Expander GPIOs (a new BIOS build requirement for the ACPI5.0 compatible iasl compiler).
- Galileo (Fab D, blue PCB) specific: BIOS routes out Intel® Quark™ SoC Uart0 signals to IO header pins by default.



- Legacy SPI Flash Recovery is implemented and tested. [Note: In the Intel® Quark™ SoC X1000 Software Release Notes for package release 1.0.0 and 1.0.1, this is incorrectly documented as “not implemented”]
- Galileo Gen2 specific: BIOS routes out Intel® Quark™ SoC Uart0 signals to IO header pins by default
- Aligned to UDK2014.SP1
- UEFI 2.4 Compliant EDKII core.
- UEFI 2.4 Secure Boot
- Quark Secure Boot Hardware Root of trust rollback protection.
- S3 Boot.
- BIOS changed to always follow secure lockdown boot policies (-DSECURE_LD) if booting on SecureSku Quark Soc parts.
- I²C access added to UEFI PI-PEI stage.
- C2 State added to ACPI tables.
- RMU Binary and MFH SPI Flash area is hardware write protected on open SKU Quark SoC parts.
- Added Legacy GPIO device to ACPI tables.
- Reserve ranges for Intel Platform Data Items and Intel Platform Types in SPI Flash Platform Data Area.
- BIOS changed to enter a HALT loop instead of warm resetting into recover mode if critical error detected during boot. This conforms to standard EDKII core behavior.

1.1.2 Linux* Operating System (OS)

- UART driver enables DMA capability (transmit only) in Intel® Quark™ SoC X1000 which increases performance for CAN over UART and RS-485 operation
- Adds ACPI S3 support
- Adds support for Galileo Gen2 board
- Adds XSLCAN (channel multiplexing over single Serial Line CAN)
- Clanton Hill: adds J1708 protocol support
- Clanton Hill: adds J1939 protocol support
- Clanton Hill: adds support for dual channel SLCAN (1 high-speed, 1 low-speed channel)
- Cross Hill: adds interrupt support to Maxim 78M6610+LMU Energy Measurement Processor
- Adds driver for Texas Instruments ADC1x8S102 ADC device
- Several USB host/device improvements and bug fixes

1.1.3 Tools

- Clanton Hill: CAN firmware upgrade tool
- The Platform data tool now has the ability to produce an independent platform data binary

1.2 Features

Features supported in this release are listed in the following subsections.



1.2.1 BIOS/Firmware

- Recovery:
 - Force recovery support (jumper/strap to force the system into recovery mode)
 - Secure recovery support (recovery capsules must be validly signed for Secure SKUs)
- Update:
 - Secure update support (update capsules must be validly signed for Secure SKUs)
- Secure Lock Down build support for secure SKUs. Includes `-DSECURE_LD` build option for creating image for secure SKUs. This restricts the boot options from EDKII (USB/SD/UEFI Shell boot are not allowed).
- Security features:
 - Protected BIOS Range registers, thus protecting more SPI flash regions.
 - SMI protection of SPI flash (secure SKUs only). Prevents non-EDKII code from updating SPI flash.
- ECC scrubbing (memory patrol scrubbing) disabled regardless of fuse setting
- Switch from SPI flash mapped platform data to ACPI objects for platform ID, MAC addresses, and serial number
- Secure boot using Root Of Trust ROM when using a secure SKU Intel® Quark™ SoC X1000
- Boot device selection:
 - SD boot
 - USB (OHCI/EHCI) boot
 - Payload boot (application in legacy SPI flash)
 - EFI Shell
- ACPI 5.0

1.2.2 Bootloader

- Secure boot Root of Trust when using a secure SKU Intel® Quark™ SoC X1000
- Isolated Memory Region (IMR) protection of compressed Linux* kernel before executing kernel
- Bootloader executed as payload from SPI flash
- Ability to load kernel and root-filesystem from SPI flash

1.2.3 Linux* Operating System (OS)

- IsADC and eADC (including calibration) optional plug-in for timer-based sampling trigger
- Drivers for Transportation Reference Design (Clanton Hill)
 - STMicroelectronics* LIS331DLH Accelerometer Driver
 - Audio Subsystem Driver
 - Analog AD7298 ADC Driver
- Thermal Driver
- HE910 3G Driver



- WiFi Driver:
 - Intel® Centrino® Wireless-N 135 (also provides Bluetooth via USB)
 - Intel® Centrino® Advanced-N 6205 (Dual Band WiFi, 2.4 and 5 GHz)
- I²C* interface
- IMR protection of kernel, text, and data sections
- Kernel logic to parse platform data specific to Clanton Peak, Industrial/Energy Reference Design (Cross Hill), and Transportation Reference Design (Clanton Hill)
- Ethernet
 - Two Ethernet interfaces: Clanton Peak, Industrial/Energy Reference Design (Cross Hill), and Transportation Reference Design (Clanton Hill)
 - One Ethernet interface: Kips Bay and Intel® Galileo board
- GPIOs fully programmable as input or output from kernel gpiolib
- SPI master interface x 2
- USB OHCI/EHCI port x 2
- USB device
- SD master interface
- ECC updates configurable at runtime through /sysfs interface
- Small embedded user-space busybox based system < 2 megabytes compressed

1.2.4 OpenOCD

- OpenOCD support is available with OpenOCD source
- GDB* server and Telnet* server support
- Halt/Step/Resume CPU
- CPU register access
- Memory access
- I/O Access (via OpenOCD command tool, not via GDB)

1.3 Limitations

The software package has the following limitations:

- Support for multiple keys is not included in this release.
- TPM measured boot is not included in this release.
- eADC firmware update utility is not included in this release.
- 1588 time-stamping protocol not supported in this release
- Watchdog timer not enabled
- UART limitations:
 - UART DMA driver supports transmit only
 - Bitrate limitation: Maximum reliable UART baud rate is limited to 921600bps
- Versioning information is not supported for individual SPI Flash assets.
- MFH SPI Flash version record is inaccurate if Manufacture SPI Flash Binary is built with EDKII standalone build and no existent if firmware update capsule is built using EDKII standalone build environment



- CAN limitations:
 - Bitrate limitation: 500 kbps is the highest speed supported on the current CAN channel. 10 kbps is not supported due to a hardware limitation of the MB91520 Fujitsu CAN controller.
 - Timestamps are not currently added by the IOC firmware prior to sending CAN messages over the UART to the Intel® Quark™ SoC X1000.

1.4 Unplanned Functionality

Support for the following items is not plan of record (POR):

- Network boot
- Legacy OS boot
- ECC scrubbing (also called memory patrolling)

1.5 Component Versions

1.5.1 Packages

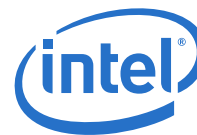
```
Board_Support_Package_Sources_for_Intel_Quark_v1.1.0:
grub-legacy_5775f32a+v1.1.0.tar.gz
meta-clanton_v1.1.0.tar.gz
Quark_EDKII_v1.1.0.tar.gz
quark_linux_v3.8.7+v1.1.0.tar.gz
sha1sum.txt
spi-flash-tools_v1.1.0.tar.gz
sysimage_v1.1.0.tar.gz
xSLCAN_Firmware_for_Intel_Quark_v1.1.0.zip (selected customers only)
```

1.6 Related Documentation

The documents in [Table 1](#) provide more information about the software in this release.

Table 1. Related Documentation (Sheet 1 of 2)

Document Name	Reference Number
Intel® Quark™ SoC X1000 Software Release Notes (this document)	330232
Intel® Quark™ SoC X1000 Board Support Package (BSP) Build and Software User Guide	329687
Intel® Quark™ SoC X1000 Software Developer's Manual for Linux*	330235
Intel® Quark™ SoC X1000 Secure Boot Programmer's Reference Manual	330234
Intel® Quark™ SoC X1000 UEFI Firmware Writer's Guide	330236
Intel® Galileo Board User Guide	330237
Source Level Debug using OpenOCD/GDB/Eclipse on Intel® Quark SoC X1000 Application Note https://communities.intel.com/docs/DOC-22203	330015

**Table 1. Related Documentation (Sheet 2 of 2)**

Document Name	Reference Number
Intel® Quark™ SoC X1000 Datasheet https://communities.intel.com/docs/DOC-21828	329676
Intel® Quark™ SoC X1000 Core Developer's Manual https://communities.intel.com/docs/DOC-21826	329679
Intel® Quark™ SoC X1000 Core Hardware Reference Manual https://communities.intel.com/docs/DOC-21825	329678
Clanton Hill and CAN Getting Started Guide This document is provided to selected customers only; contact your Intel representative.	545350

1.7 Licensing

This package contains source code licensed under one or more open source licenses. Consult the COPYING, README, or LICENSE files in the appropriate subdirectory. Intel does not make any representations or warranties, express or implied, including without limitation, any warranty of fitness for any purpose, merchantability or non-infringement.

The package also includes executable binaries provided under Intel Proprietary License (IPL) as listed in Table 2. The IPL license file is in the same directories as the binaries in the package.

Table 2. License Files

Location	Description
...\Quark_EDKII_v1.1.0\QuarkSocPkg\QuarkNorthCluster\Binary\QuarkMicrocode\RMU.bin	Microcode for the Intel® Quark™ SoC X1000. (RMU: Remote Management Unit)
...\Quark_EDKII_v1.1.0\QuarkSocPkg\QuarkNorthCluster\Binary\Quark2Microcode\RMU.bin	Microcode for a future generation Quark SoC.
xSLCAN_Firmware_for_Intel_Quark_v1.1.0\ioc_combined_image_clanton.mhx	xSLCAN firmware to be programmed to the Fujitsu xLSCAN controller on a Clanton Hill FFRD.



2.0 Known Issues

Known issues in the current release are listed in the following table.

Table 3. Known Issue Summary

38292 - Cannot force MMC into 4-bit mode due to kernel bug	13
45539 - SDMediaDevice.efi is setting older 25 MHz cards to 50 MHz	13
53037 - Linux kernel has some debug settings enabled by default.....	13
53887 - Deadlock in bluetooth stack - inherited from upstream kernel	13
57071 - Galileo board is unavailable after host computer sleeps.....	14
58381 - Attempting to unload a Linux driver which is in use causes console to freeze	14
60003 - Legacy RTC 'Valid' time bit is set even though RTC contains invalid time	14
60803 - BIOS error when using 2G MMC card.....	14
61236 - Real Time Clock update issue (sh: %4Y%2m%2d%2H%2M: bad number).....	14
63520 - SMBIOS fields are currently incorrect for the Quark reference platforms	15
64263 - Error detecting Western Digital USB 3.0 hard drive.....	15
65706 - Hot plug of USB key intermittently fails.....	15
65952 - USB Errors seen with Sandisk Cruzer 4GB Flash Drive.....	15
66053 - Poor USB write performance caused by automounter	15
66218 - Nonfunctional USB key may break the detection for other functional USB keys on Clanton Hill.....	16
69965 - Quark EDKII default exception handler entry point is not valid	16
70961 - Clanton Hill: If ETH0 is disconnected, ETH1 will not automatically pick up an address from DHCP ..	16
73848 - Spurious 'unmounting /media/realroot' error message	16
75161 - Boot log error: memory range cannot be reserved	17
75172 - Clanton Hill: USB Error messages reported when booting debug build of EDKII	17
78550 - Some USB keys not recognized by Quark EDKII recovery on Galileo and Galileo Gen2.....	18
80408 - Boot stalls and needs UART serial console activity to continue.....	18
83572 - I2C driver can lock up the system	18
89664 - The SD card may be unavailable on S3 resume.....	18
95013 - USB does not automount when using SD image but does when using SPI image	18
96194 - A suspend/resume cycle may result in very small memory leak.....	18
96268 - System fails to boot in 8259 (PIC) mode.....	19
96922 - UART software flow control not working when UART operates in DMA mode	19
98436 - Inconsistent DMA numbering on UART devices.....	20
98616 - J1939 message errors	20
99141 - xSLCAN message error.....	20
99257 - Gadget serial data loss at end of transfer	20
99874 - Cannot unload Legacy GPIO driver module	21
103693 - Galileo services running on non-Galileo platforms when board is booted from SD card.....	21



2.1 38292 - Cannot force MMC into 4-bit mode due to kernel bug

Title	Cannot force MMC into 4-bit mode due to kernel bug
Id	38292
Implication	There is a kernel bug that is seen when forcing MMC into 4-bit mode. If you use the command: modprobe sdhci debug_quirks=0x400000 Only one bit is set: SDHCI_QUIRK_FORCE_1_BIT_DATA, bit 22 The board fails to initialize; returning these errors: - 110 timeout - 5 I/O error
Workaround	Use the command: modprobe sdhci debug_quirks=0x8400000 This sets: SDHCI_QUIRK_FORCE_1_BIT_DATA, bit 22 SDHCI_QUIRK_MISSING_CAPS, bit 27

2.2 45539 - SDMediaDevice.efi is setting older 25 MHz cards to 50 MHz

Title	SDMediaDevice.efi is setting older 25 MHz cards to 50 MHz
Id	45539
Implication	25MHz SD cards will not be recognized or usable.
Workaround	Use 'Fast' 50MHz capable SD cards.

2.3 53037 - Linux kernel has some debug settings enabled by default

Title	Linux kernel has some debug settings enabled by default
Id	53037
Implication	The Intel® Quark™ SoC X1000 kernel default configuration has some debug configuration settings enabled (e.g. in SPI subsystem and memory management). This results in an increased kernel footprint and may have a performance impact.
Workaround	Disable the unwanted debug settings and rebuild the kernel. Note the default Quark kernel configuration file is at meta/cfg/kernel-cache/bsp/quark/quark.cfg

2.4 53887 - Deadlock in bluetooth stack - inherited from upstream kernel

Title	Deadlock in bluetooth stack - inherited from upstream kernel
Id	53887
Implication	When using the bluetooth software stack, a potential deadlock message can be found in /var/log/messages. Could potentially cause a lock-up but this has yet to be shown.
Workaround	None.



2.5 57071 - Galileo board is unavailable after host computer sleeps

Title	Galileo board is unavailable after host computer sleeps
Id	57071
Implication	When the Galileo board is connected to a host computer that enters sleep mode, and the host is woken, the Galileo board will be unavailable on USB. This behavior is caused by the Gadget Serial driver and is seen on all OSes (Linux, Windows, Mac OS).
Workaround	There is no workaround, you must reboot the Galileo board.

2.6 58381 - Attempting to unload a Linux driver which is in use causes console to freeze

Title	Attempting to unload a Linux driver which is in use causes console to freeze
Id	58381
Implication	When a driver is in use (like for instance SD/MMC mass storage device when an SD card is mounted) and user tries to remove it using 'modprobe -r mmc_block' then existing console hangs. Existing console is not usable until board rebooted or mass storage device unmounted from other console.
Workaround	Make sure the driver is not use before trying to unload. For instance unmount mass storage device first, then unload mmc_block driver.

2.7 60003 - Legacy RTC 'Valid' time bit is set even though RTC contains invalid time

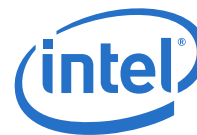
Title	Legacy RTC 'Valid' time bit is set even though RTC contains invalid time
Id	60003
Implication	Legacy RTC 'Valid' time bit is set even though RTC contains invalid time. Any software that trusts the 'Valid' bit without any sanity checks on the time/date may be using a corrupt date/time.
Workaround	None.

2.8 60803 - BIOS error when using 2G MMC card

Title	BIOS error when using 2G MMC card
Id	60803
Implication	2G Transcend MMC card (TS2GMMC4) is not recognised or is unusable.
Workaround	Use alternative MMC card.

2.9 61236 - Real Time Clock update issue (sh: %4Y%2m%2d%2H%2M: bad number) (Sheet 1 of 2)

Title	Real Time Clock update issue (sh: %4Y%2m%2d%2H%2M: bad number)
Id	61236



2.9 **61236 - Real Time Clock update issue (sh: %4Y%2m%2d%2H%2M: bad number) (Sheet 2 of 2)**

Implication	The initscripts provided in poky release 1.4 do not support the simplified date program used by busybox. This shows an error in the boot log and may prevent Linux from reading time from the RTC clock and from saving time to it.
Workaround	Go to the /etc/init.d/ directory on the target system. In both the bootmisc.sh and save-rtc.sh scripts there, search for: date -u +%4Y%2m%2d%2H%2M and replace with: date -u +%Y%m%d%H%M

2.10 **63520 - SMBIOS fields are currently incorrect for the Quark reference platforms**

Title	SMBIOS fields are currently incorrect for the Quark reference platforms
Id	63520
Implication	Only SMBIOS Type0 and Type2 fields have been validated to be correct. Software using any other SMBIOS entries may be using incorrect information.
Workaround	Only use validated SMIOS table entries.

2.11 **64263 - Error detecting Western Digital USB 3.0 hard drive**

Title	Error detecting Western Digital USB 3.0 hard drive
Id	64263
Implication	Western Digital USB3.0 HDD not recognized or usable.
Workaround	Use alternative USB HDD.

2.12 **65706 - Hot plug of USB key intermittently fails**

Title	Hot plug of USB key intermittently fails
Id	65706
Implication	USB key is not recognized or is unusable.
Workaround	Disconnect and reconnect the USB key.

2.13 **65952 - USB Errors seen with Sandisk Cruzer 4GB Flash Drive**

Title	USB Errors seen with Sandisk Cruzer 4GB Flash Drive
Id	65952
Implication	USB Key 'Sandisk Cruzer 4GB' is not recognized or is unusable in BIOS.
Workaround	Use alternative USB key.

2.14 **66053 - Poor USB write performance caused by automounter (Sheet 1 of 2)**

Title	Poor USB write performance caused by automounter
Id	66053



2.14 66053 - Poor USB write performance caused by automounter (Sheet 2 of 2)

Implication	Automounting of USB memory is done with the '-o sync' flag by default. For VFAT filesystems (the default on USB and SD memory), there is a performance degradation which causes a typical write to take about 5 minutes.
Workaround	One workaround is to search and replace '-o sync' with '-o flush' in the /usr/bin/automount.sh file. However, the copy command will return before the write is complete. If the USB memory device is removed before the write is complete, the board may be in an unbootable state.

2.15 66218 - Nonfunctional USB key may break the detection for other functional USB keys on Clanton Hill

Title	Nonfunctional USB key may break the detection for other functional USB keys on Clanton Hill
Id	66218
Implication	This issue is seen only when non-functional USB key is connected to J1. Note that J12 (USB port0) and functional USB key connected to J10 (USB port1 via hub). Issue is not seen when positions are swapped.
Workaround	Only connect functional USB devices (USB devices that EDKII can function with without errors) to the system.

2.16 69965 - Quark EDKII default exception handler entry point is not valid

Title	Quark EDKII default exception handler entry point is not valid
Id	69965
Implication	If the system hits an exception (divide by zero for example) during Quark EDKII boot then the system will vector to the default exception handler at address 0xFFFFFE4. As there is no valid exception handler at this address, system behavior is undefined.
Workaround	None.

2.17 70961 - Clanton Hill: If ETH0 is disconnected, ETH1 will not automatically pick up an address from DHCP

Title	Clanton Hill: If ETH0 is disconnected, ETH1 will not automatically pick up an address from DHCP
Id	70961
Implication	There are two PHYs on the Clanton Hill board. If ETH0 is disconnected and ETH1 is connected to the network with DHCP available, an address for ETH1 is not retrieved automatically.
Workaround	Enter the command 'ifup ETH1' to manually retrieve an address.

2.18 73848 - Spurious 'unmounting /media/realroot' error message

Title	Spurious 'unmounting /media/realroot' error message
Id	73848
Implication	When booting from mass storage the following error is returned in the boot log: umount: can't umount /media/realroot: Device or resource busy This occurs for images booted from mass storage devices.
Workaround	None; this error can be ignored.



2.19 75161 - Boot log error: memory range cannot be reserved

Title	Boot log error: memory range cannot be reserved
Id	75161
Implication	When booting, the following error is displayed in boot logs: [0.996963] pnp: PnP ACPI init [0.996963] ACPI: bus type pnp registered [1.003633] system 00:00: [mem 0xe0000000-0xe1ffffff] has been reserved [1.011283] system 00:00: [mem 0xfed1c000-0xfed1ffff] has been reserved [1.018649] system 00:00: [mem 0x000c0000-0x000dffff] has been reserved [1.026093] system 00:00: [mem 0x000e0000-0x000ffffff] could not be reserved
Workaround	This error message will not affect board operation and can be ignored.

2.20 75172 - Clanton Hill: USB Error messages reported when booting debug build of EDKII

Title	Clanton Hill: USB Error messages reported when booting debug build of EDKII
Id	75172
Implication	The following error messages are reported during boot on Clanton Hill with a debug build of EDKII: Error Count : 3 EhcControlTransfer: error - Device Error, transfer - 2 However, no functional USB issues are observed and USB is working as expected. Issue is currently under investigation.
Workaround	None.



2.21 78550 - Some USB keys not recognized by Quark EDKII recovery on Galileo and Galileo Gen2

Title	Some USB keys not recognised by Quark EDKII recovery on Galileo and Galileo Gen2
Id	78550
Implication	Recovery process will fail on Galileo and Galileo Gen2 with these USB keys. Currently the following USB keys have been seen to fail: 1) Sandisk cruzer 4GB 2) Transend 4GB
Workaround	Two potential workarounds have been identified: (1) Connect a USB hub to the Galileo Gen2 USB port and then connect the failing USB key(s) to the USB hub. The USB keys have been observed to pass in this configuration (2) Select a different USB key

2.22 80408 - Boot stalls and needs UART serial console activity to continue

Title	Boot stalls and needs UART serial console activity to continue
Id	80408
Implication	Problem appears rarely and when it does boot can be progressed by hitting any key. Normal prompts to select recovery are not compromised by this workaround.
Workaround	If problem does occur, hit return for boot to continue.

2.23 83572 - I²C driver can lock up the system

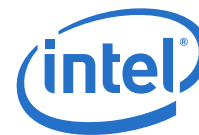
Title	Galileo USB device not recovering on reinsert
Id	83572
Implication	This is a very rare occurrence
Workaround	None

2.24 89664 - The SD card may be unavailable on S3 resume

Title	The SD card may be unavailable on S3 resume
Id	89664
Implication	The following error is reported when the error occurs "mmc0: error -110 (SD card was removed?) during S3 resume"
Workaround	None

2.25 95013 - USB does not automount when using SD image but does when using SPI image

Title	USB does not automount when using SD image but does when using SPI image
Id	95013
Implication	SPI image running Yocto 1.6 uses initd which provides a hotplug mechanism. Yocto using initd uses /etc/init.d/mdev.sh to echo mdev to /proc/sys/kernel/hotplug. The SD image running Yocto 1.6 uses systemd (YP1.6.1 SD) has no /etc/init.d/mdev.sh so USB is not automounted as a result.
Workaround	Manually mount USB device using mount command. For example, mount /dev/sda1 /media/mountpoint



2.26 96194 - A suspend/resume cycle may result in very small memory leak

Title	A suspend/resume cycle may result in very small memory leak
Id	96194
Implication	Each suspend/resume cycle may result in very small <100byte memory leak
Workaround	None

2.27 96268 - System fails to boot in 8259 (PIC) mode

Title	System fails to boot in 8259 (PIC) mode
Id	96268
Implication	If Linux is booted in legacy 8259 mode (by passing 'noapic nolapic' string in the kernel command line) the system will fail to boot, or result in unpredictable run-time behaviour. This is due to a driver being able to handle only APIC/MSI interrupt routing scheme.
Workaround	<p>In drivers/mfd/intel_qrk_gip_core.c, function intel_qrk_gip_probe(), identify the following block:</p> <pre> if (enable_msi) { pci_set_master(pdev); retval = pci_enable_msi(pdev); if (retval) goto err_release_drvdata; } </pre> <p>Delete the statements:</p> <pre> if (retval) goto err_release_drvdata; </pre> <p>Rebuild the kernel.</p>

2.28 96922 - UART software flow control not working when UART operates in DMA mode

Title	UART software flow control not working when UART operates in DMA mode
Id	96922
Implication	When UART is in DMA mode, software-based flow control will not work. Note hardware flow control will work in both PIO and DMA mode.
Workaround	To use software flow control, switch the UART to PIO mode by using the intel_quark_huart_dma.uartX_dma module parameter. Refer to the Quark Linux Software Developers' Manual for more details

**2.29 98436 - Inconsistent DMA numbering on UART devices**

Title	Inconsistent DMA numbering on UART devices
Id	98436
Implication	<p>The Quark HSUART driver enumerates DMA controllers starting from ID 1. This is inconsistent with the UART numbering, which starts from 0.</p> <p>If UARTs are used in DMA mode (default), there is no implication. On the other hand, in order to switch off DMA on a UART instance, the user needs to bear this off-by-one issue in mind.</p>
Workaround	<p>In order to disable DMA on a UART port, the following parameter must be passed to the kernel command line:</p> <ul style="list-style-type: none">- UART0: intel_quark_hsuart_dma.uart1_dma=0- UART1: intel_quark_hsuart_dma.uart2_dma=0

2.30 98616 - J1939 message errors

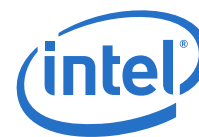
Title	J1939 message errors
Id	98616
Implication	Data errors can occur on J1939 message transfers
Workaround	None

2.31 99141 - xSLCAN message error

Title	xSLCAN message error
Id	99141
Implication	Data errors can occur message on xSLCAN message transfers
Workaround	None

2.32 99257 - Gadget serial data loss at end of transfer

Title	Gadget serial data loss at end of transfer
Id	99257
Implication	Raw data transfer over gadget serial can result in data loss
Workaround	Use cloader/zmodem for Galileo Arduino Sketch Upload



2.33 99874 - Cannot unload Legacy GPIO driver module

Title	Cannot unload Legacy GPIO driver module
Id	99874
Implication	The gpio-sch (legacy GPIO) driver has a bug in the module unloading routine which causes an invalid IRQ range to be freed up. Therefore if the module is unloaded, other drivers may lose interrupt delivery capability.
Workaround	Do not unload the gpio-sch module.

2.34 103693 - Galileo services running on non-Galileo platforms when board is booted from SD card

Title	Galileo services running on non-Galileo platforms when board is booted from SD card
Id	103693
Implication	When a Quark board is booted from mass storage device, Galileo-specific services get executed regardless of the board type. The Galileo services reserve the USB gadget serial device which is then not accessible by applications. Note this only happens under mass storage boot, which uses systemd, as opposed to minimal SPI boot, which uses init.
Workaround	Kill the launcher.sh and then kill the clloader process. Note the order is important. <pre> root@quark:~# ps grep clloader 250 root 1980 S /opt/cln/galileo/clloader --escape --binary --zmodem 252 root 2488 S grep clloader root@quark:~# ps grep launcher.sh 222 root 3088 S /bin/sh /opt/cln/galileo/launcher.sh 254 root 2488 S grep launcher.sh root@quark:~# kill -9 250 root@quark:~# kill -9 222 </pre>



3.0 Resolved Issues

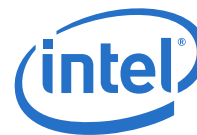
This section contains issues resolved since package version 1.0.0.

Table 4. Resolved Issue Summary

38542 - SPI flash tool / signing tools does not support multiple inclusions of same binary at different addresses.....	22
58453 - pch_ude driver crash on reload	23
64225 - SPI flash tool does not detect duplicate sections or duplicate options in a section in layout.conf....	23
71061 - Linux Boot Failure on Failure to Remap PCIe MMIO Region (256MB) from Physical to Virtual Addressing	24
71061 - Linux Boot Failure on Failure to Remap PCIe MMIO Region (256MB) from Physical to Virtual Addressing	24
71538 - Linux Segfault when Using Lock Prefix Instruction under Specific Circumstances	24
75904 - OpenSSL version affected by 'heartbleed' defect	24

3.1 38542 - SPI flash tool / signing tools does not support multiple inclusions of same binary at different addresses

Title	SPI flash tool / signing tools does not support multiple inclusions of same binary at different addresses
Id	38542
Implication	<p>When building an image using a layout.conf file that uses the same 'item_file' source in two (or more) asset descriptor blocks, the expected behavior is as follows:</p> <ol style="list-style-type: none">1. Image is generated.2. Two assets exist at different locations, with identical body data.3. Even though the bodies of both assets contain identical data, the RSA signature section of each asset should contain different signatures, due to the intentionally non-deterministic nature of the signing process. <p>What actually happens:</p> <p>All assets will be duplicates of the last asset listed in layout.conf, including RSA signatures and any other variables such as SVN indices.</p> <p>If, for example, 3 assets use the same 'item_file' source, and have SVN indices of 1, 2, and 3 respectively in layout.conf, and the one with SVN index 3 is the last one listed in layout.conf, then the other two assets that use this same 'item_file' source will also have an SVN index of 3, as well as identical RSA signatures.</p>
Resolution	Resolved in release 1.0.1. This use case is detected and a meaningful error message explains it is not supported.



3.2 58453 - pch_udc driver crash on reload

Title	pch_udc driver crash on reload
Id	58453
Implication	When ehci_pci, ehci_hcd, pch_udc, g_serial drivers are loaded and user executes: modprobe -r g_serial modprobe -r pch_udc modprobe pch_udc then pch_udc driver crashes. Problem seen on Galileo board. Driver is unusable until board rebooted.
Workaround	Unload first ehci-pci driver to revert to USB1.1, then g_serial and pch_udc drivers can be unloaded or reloaded.

3.3 64225 - SPI flash tool does not detect duplicate sections or duplicate options in a section in layout.conf

Title	SPI flash tool does not detect duplicate sections or duplicate options in a section in layout.conf
Id	64225
Implication	The python module ConfigParser used by the SPI flash tools cannot process multiple sections in the layout.conf file that do not have unique names. Within each section, it cannot process options that are not unique. This behavior may cause unexpected results. For example, if the layout.conf file has two sections called [Main], the second section will be used. If the layout.conf file has two options both called Size in the [Main] section, the second Size option will be used.
Workaround	Do not duplicate section names or option names within the same section in layout.conf.

3.4 66803 - Recovery boot intermittently stalls during PCI enumeration

Title	Recovery boot intermittently stalls during PCI enumeration
Id	66803
Implication	An intermittent system hang has been observed when booting a recovery image. This hang occurs during PCI enumeration. This hang has only been observed on a Cross Hill platform and happened 4 times out of 10 attempts.
Workaround	Retry the recovery process.



3.5 71061 - Linux Boot Failure on Failure to Remap PCIe MMIO Region (256MB) from Physical to Virtual Addressing

Title	Linux boot failure on failure to remap PCIe MMIO region (256MB) from physical to virtual addressing
Id	71061
Implication	V1.0.0 firmware required the operating system to map PCI express MMIO space from physical to virtual address. However, in the 1.0.0 release, the kernel called UEFI runtime service SetVirtualAddressMap() without PCI express MMIO space being mapped to virtual addresses. The impact was the system would reboot in Recovery mode earlier in kernel boot.
Workaround	Resolved in release 1.0.1.

3.6 71538 - Linux Segfault when Using Lock Prefix Instruction under Specific Circumstances

Title	Linux segfault when using lock prefix instruction under specific circumstances
Id	71538
Implication	When a memory instruction with LOCK prefix executes and if it encounters a page fault (#PF), the state of the CPU could potentially get corrupted. Software should avoid using the LOCK prefix for instructions that may cause page fault (#PF).
Workaround	<p>Resolved in release 1.0.1. Due to the LOCK prefix core silicon errata, the Yocto software release has patched the GNU assembler to remove LOCK instructions from code generated by the GNU toolchain. The workaround is enabled by default and no option has to be specified. All code is compiled with the workaround applied, so no binaries or libraries will include the LOCK prefix.</p> <p>The toolchain workaround can be verified to be in the toolchain by issuing the GNU assembler command: > as --help</p> <p>The help text will show the option: -mquark-strip-lock=[yes no] strip all lock prefixes; default is yes</p> <p>The workaround can be explicitly set/cleared from gcc compiler using the command: gcc -Xassembler -mquark-strip-lock=[yes no]</p>

3.7 75904 - OpenSSL version affected by 'heartbleed' defect

Title	OpenSSL version affected by 'heartbleed' defect
Id	75904
Implication	A missing bounds check in the handling of the TLS heartbeat extension can be used to reveal up to 64k of memory to a connected client or server. Documented here: https://www.openssl.org/news/secadv_20140407.txt
Resolution	Resolved in release 1.0.1. The OpenSSL recipe was updated to download and build a fixed version of OpenSSL.

