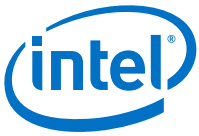
Intel® Quark SoC X1000 Software

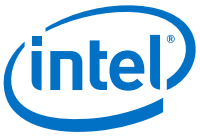
Package Version: 1.0

Release Notes

04 March 2014

Order Number: 330232-001US





Contents

Tables

Revision History

Date	Revision	Description
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: 1.0 (Gold)

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

- Customer Reference Boards:
  - Kips Bay (Fab C, green PCB)
  - Galileo (Fab D, blue PCB)
- 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).

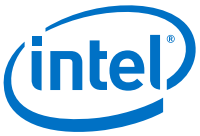
These release notes also include known issues with third-party or reference platform components that affect the operation of the software.

1.1 Features

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

1.1.1 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
- UEFI 2.3.1 compliant
- ACPI 5.0

1.1.2 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
- U-Boot memory tests ported and included

1.1.3 Linux* Operating System (OS)

- lsADC 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)
- I^2C* 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

1.1.4 OpenOCD

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

1.2 Limitations

Caution: This release includes a known Linux boot failure caused by failure to remap the PCIe MMIO region (256MB) from physical to virtual addressing. For details including a workaround, see Errata 2.25.

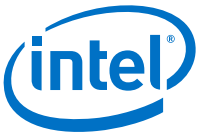
The software package has the following limitations:

- S3 support is implemented but not validated. It is not recommended for use in this release.
- Legacy SPI flash recovery not implemented.
- 1588 time-stamping protocol not supported in this release.
- DMA UART driver not supported in this release.
- Watchdog timer not enabled.
- Automatic version number updating during the update/recovery process is not implemented. Rollback protection (preventing downgrading to a previous software version) requires the version number of a software module to be greater or equal to the corresponding version number stored in the SPI flash. Support to update the version number stored in SPI flash if the corresponding software module is being updated, has not been added.
- UEFI 2.3.1 Secure Boot support is not implemented.
- eADC firmware update utility is not included in this release.
- Support for multiple keys is not included in this release.
- Backup image support is not included in this release.

1.3 Unplanned Functionality

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

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



1.4 Component Versions

1.4.1 Packages

grub-legacy_5775f32a+v1.0.0.tar.gz
meta-clanton_v1.0.0.tar.gz
Quark_EDKII_v1.0.0.tar.gz
quark_linux_v3.8.7+v1.0.0.tar.gz
sha1sum.txt
spi-flash-tools_v1.0.0.tar.gz
sysimage_v1.0.0.tar.gz

1.4.2 BIOS/Firmware Version

Development Platform	Version
Clanton Peak SVP	1.0

1.5 Related Documentation

The documents in Table 1 provide more information about the software in this release.

Table 1. Related Documentation

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 Linux* Programmer's Reference Manual	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
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

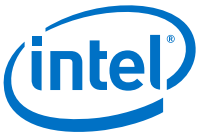
2.0 Known Issues

The known issues in the current release are listed below.

Table 2. Known Issue Summary

2.1 38292 - Cannot force MMC into 4-bit mode due to kernel bug (Sheet 1 of 2)

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



2.1 38292 - Cannot force MMC into 4-bit mode due to kernel bug (Sheet 2 of 2)

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 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	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: 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. What actually happens: All assets will be duplicates of the last asset listed in layout.conf, including RSA signatures and any other variables such as SVN indices. 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.
Workaround	If the same binary asset must be duplicated within an image but with a different SVN index from the original, you can make a symbolic link to the asset. Alternatively, you can make a copy of the asset first and reference that copy in the asset descriptor block for the duplicate.

2.3 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.

2.4 46834 - UART interrupt handler not restored after resume from S3

Title	UART interrupt handler not restored after resume from S3
Id	46834
Implication	Suspected race condition between 8250 restore code and interrupt handler. Following resume from S3, 8250 will be in polled mode, not interrupt mode.
Workaround	Do not enter into S3 (unsupported).

2.5 48226 - eSRAM driver cannot map code required to do mapping

Title	eSRAM driver cannot map code required to do mapping
Id	48226
Implication	eSRAM driver depends on code internally and externally in order to map things into eSRAM. During the mapping process, over-layed sections of DRAM become NULL for a time. It is not possible to eSRAM overlay code to itself be overlayed.
Workaround	Do not try to overlay any of the following kernel symbols: intel_cln_esram_* intel_cln_sb_* memcpy spin_lock spin_unlock spin_lock_irqsave spin_unlock_irqrestore pci_read_config_dword pci_write_config_dword

2.6 53887 - Deadlock in bluetooth stack - inherited from upstream kernel

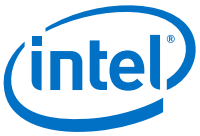
Title	Deadlock in bluetooth stack - inherited from upstream kernel
Id	53887
Implication	When using the bluetooth software stack, a potential deadlock message can be found in /var/log/messages. Could potentially cause a lock-up but this has yet to be shown.
Workaround	None.

2.7 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.

2.8 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.



2.9 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.

2.10 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.

2.11 60147 - Quark enumerates incorrect device class as a USB CDC ACM device

Title	Quark enumerates incorrect device class as a USB CDC ACM device
Id	60147
Implication	As a USB CDC ACM device, the Quark SoC enumerates a USB Device descriptor with Class, SubClass and DeviceProtocal 02, 00, and 00 respectively. This is incorrect given that the Quark CDC ACM setup uses Interface Association Descriptors. The USB specification recommends different values in the device descriptor when using IADs, consequently, Windows may generate errors. The values in the device descriptor should be EFh, 02h, 11h, respectively.
Workaround	None.

2.12 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.

2.13 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

2.13 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

2.14 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.

2.15 64225 - SPI flash tool does not detect duplicate sections or duplicate options in a section in layout.conf

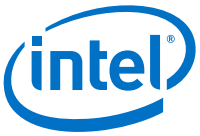
Title	SPI flash tool does not detect duplicate sections or duplicate options in a section in layout.conf
Id	64225
Implication	The python module ConfigParser used by the SPI flash tools cannot process multiple sections in the layout.conf file that do not have unique names. Within each section, it cannot 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.

2.16 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.

2.17 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.



2.18 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.

2.19 66053 - Poor USB write performance caused by automounter

Title	Poor USB write performance caused by automounter
Id	66053
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.

2.20 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.

2.21 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.

2.22 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 0xFFFFFFE4. As there is no valid exception handler at this address, system behavior is undefined.
Workaround	None.

2.23 70897 - SPI flash corruption can cause a system built without the SECURE_LD build option to become un-recoverable

Title	SPI flash corruption can cause a system built without the SECURE_LD build option to become un-recoverable
Id	70897
Implication	The RMU.bin area of SPI flash is not currently protected by the Protected BIOS Range Registers (RCBA + 3080h -> RCBA + 308Bh). The RMU.bin is a critical component that is required for the Recovery boot path. If the RMU.bin image in SPI flash gets corrupted during SPI flash updates, then the system will be unrecoverable and unable to boot.
Workaround	Avoid updating this area of SPI flash on unsecure systems.

2.24 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.

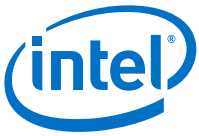
2.25 71061 - Linux boot failure on failure to remap PCIe MMIO region (256MB) from physical to virtual addressing

Title	Linux boot failure on failure to remap PCIe MMIO region (256MB) from physical to virtual addressing
Id	71061
Implication	V1.0.0 firmware requires the operating system to map PCI express MMIO space from physical to virtual address. However, in the current release, the kernel calls UEFI runtime service SetVirtualAddressMap() without PCI express MMIO space being mapped to virtual addresses. The impact is the system will reboot in Recovery mode earlier in kernel boot.
Workaround	Add the text “vmalloc=384M” to all Kernel command lines in all grub .conf files resident in SPI flash and external media. Example kernel command line is shown below: kernel /bzImage root=/dev/ram0 console=ttyS1,115200n8 earlycon=uart8250,mmio32,$EARLY_CON_ADDR_REPLACE,115200n8 vmalloc=384M reboot=efi,warm apic=debug rw

2.26 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	Segfault may occur for specific instruction sequences when using the “lock” prefix instruction on Linux. Not all usages of lock prefixed instructions cause the issue. Specific circumstances are under investigation by Intel.

Workaround	Inclusion of a 'nop' instruction immediately prior to a lock prefix instruction has been shown to resolve the segfault in cases seen so far by Intel.
	Example of failing code sequence using a user space application:
	.code32
	.globl main
	.type main, @function
	main:
	pushl %ebp
	movl %esp, %ebp
	subl $0xfff88, %esp
	movl %ebp, %eax
	sub $0x100000, %eax
	lock addl $0x1,(%eax)
	leave
	ret
	
	Example of suggested fix:
	.code32
	.globl main
	.type main, @function
	main:
	pushl %ebp
	movl %esp, %ebp
	subl $0xfff88, %esp
	movl %ebp, %eax
	sub $0x100000, %eax
	nop
	lock addl $0x1,(%eax)
	leave
	ret



3.0 Resolved Issues

None; first production release of the software.

§ §