



Intel® Galileo Board

Getting Started Guide

22 May 2014



Getting Started Guide

This document explains how to connect your Intel® Galileo board to the computer, install the software on your computer, and upload your first sketch.

You should be able to do this in under an hour (not counting your download time).

Note: This release uses a special version of the Arduino IDE. The first thing you **must** do is download it from the Intel website below and update the SPI flash on the board.

The basic steps you will follow are:

1. [Get a board and cables](#)
2. [Download the Arduino environment and board firmware](#)
3. [Connect the board](#)
4. [Install the drivers and other software](#)
5. [Launch the Arduino IDE application](#)
6. [Update your board firmware](#)
7. [Open the blink example](#)
8. [Select your board](#)
9. [Select your serial port](#)
10. [Upload the program](#)

See these sections for advanced tasks:

11. [Booting your board from an SD card \(mandatory for WiFi\)](#)
12. [Serial cable \(optional\)](#)
13. [Manually update your board firmware \(optional\)](#)
14. [Troubleshooting Tips](#)

For more information about Intel® Galileo, including online communities and support, try these links:

<http://www.intel.com/support/galileo>

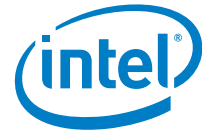
<http://communities.intel.com/community/makers>

1. Get a board and cables

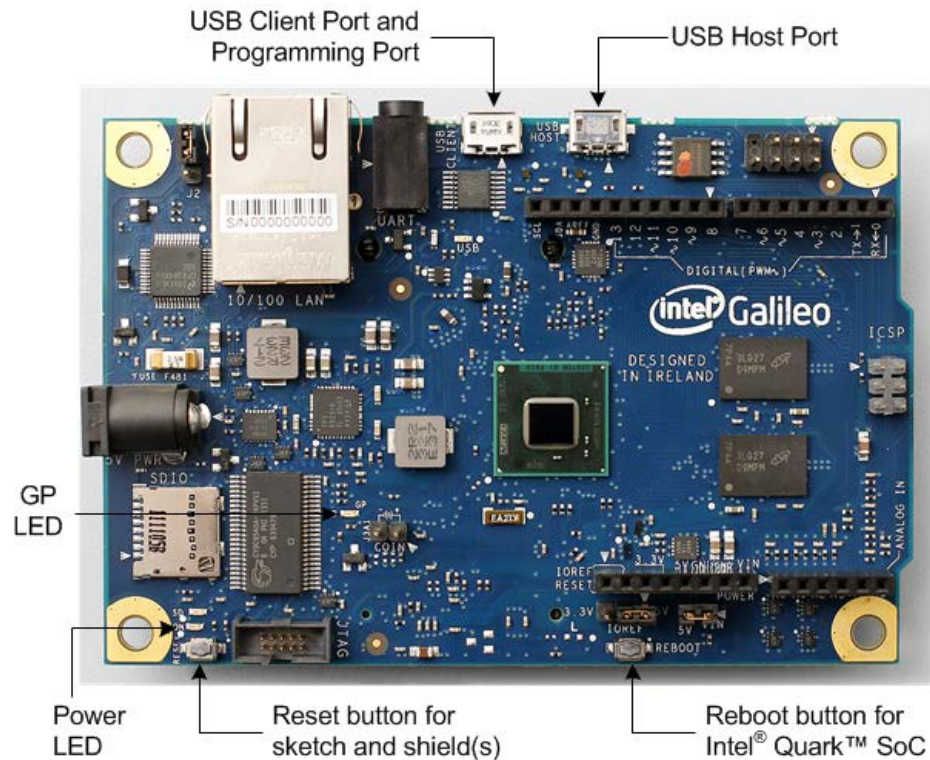
The kit contains:

- 1x Intel® Galileo Customer Reference Board (CRB) (Fab D with blue PCB)
 - 1x 5v power supply with cable
- Warning: You must use a power supply to power the board! You will damage the board if you power it with a USB cable.**

You also need a USB Type A Male to Micro-B Male cable (not included) to connect the board to your computer.



Intel® Galileo Board



Note: The I/O headers are identical to the Arduino Revision 1.0 pinout.

2. Download the Arduino environment and board firmware

Download the latest IDE and firmware files here:

<https://communities.intel.com/community/makers/drivers>

You will need up to 200 MB of free space, depending on which OS you are using.

Note: When unzipping the packages:

- Extract the package into the c:\ directory due to a known issue unzipping packages with long file paths.
- Use an unzip tool that supports an extended file path (for example, 7-zip from <http://www.7-zip.org/>).

This release contains multiple zip files, including:

- Operating system-specific IDE packages, contain automatic SPI flash update:
 - Intel_Galileo_Arduino_SW_1.5.3_on_Linux32bit_v1.0.0.tgz (73.9 MB)
 - Intel_Galileo_Arduino_SW_1.5.3_on_Linux64bit_v1.0.0.tgz (75.2 MB)
 - Intel_Galileo_Arduino_SW_1.5.3_on_MacOSX_v1.0.0.zip (55.9 MB)
 - Intel_Galileo_Arduino_SW_1.5.3_on_Windows_v1.0.0.zip (106.8 MB)
- (Mandatory for WiFi) Files for booting board from SD card. See [Section 11](#).
LINUX_IMAGE_FOR_SD_Intel_Galileo_v1.0.0.7z (38.7 MB)



- (Optional) Board Support Package (BSP) sources including Yocto archive:
The BSP build process requires additional 30 GB of free disk space.
`Board_Support_Package_Sources_for_Intel_Quark_v1.0.0_full_yocto_archive.tar.gz`
(1.84 GB)

If you are working with the BSP sources, see the *Intel® Quark™ SoC X1000 BSP Build Guide* [<https://communities.intel.com/docs/DOC-21882>]

Windows*:

1. Unzip the IDE file to the `c:\` directory.
 2. Make sure to preserve the folder structure. Double-click the folder to open it.
- If you are updating your IDE, be sure to uninstall the previous IDE version first.

Linux*:

1. In your home directory, extract the package with the appropriate command:
`tar -zxvf arduino-1.5.3-linux32.tar.gz`
or
`tar -zxvf arduino-1.5.3-linux64.tar.gz`
2. To avoid interference with the IDE, you **must** disable modem manager. The exact command will depend on your Linux distribution.
For example, the command `sudo apt-get remove modemmanager` may work.
3. In the same directory where the package was downloaded, launch the Arduino IDE by executing `./arduino`

Note: The first time you compile and install a sketch, the Galileo software will be installed automatically. This requires appropriate file and directory access permissions. You may be required to run the IDE as root or using `sudo arduino` depending on your Linux distribution.

OS X*:

1. Download the zip file on the Mac hard drive and unzip it.
2. In the Finder, double-click the zip file to uncompress the IDE application.
3. Drag and drop the Arduino application onto the Applications folder on your Mac.

3. Connect the board

WARNING: You must use a power supply to power the board! You will damage the board if you power it with a USB cable.

Connect the 5V power cable to the Galileo board and to a power outlet.

Note: Always connect the 5V power before any other connection.

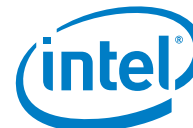
The Power LED shown above (board label = ON) will turn on.

4. Install the drivers and other software

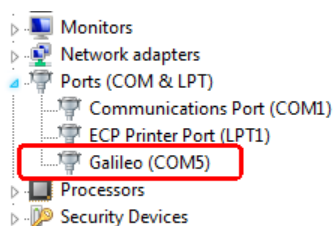
Windows:

1. If not done already, connect the 5V power cable to the Galileo board and to a power outlet.

Note: Always connect the 5V power before the USB connection.



2. Connect the USB cable to the USB Client Port (closest to the Ethernet) and to a PC. Wait for Windows to begin its driver installation process. After a few moments, the process will fail.
3. Click on the **Start** Menu, and open up the **Control Panel**. While in the Control Panel, navigate to **System**. Next, click on **System**. Once the System window is up, open the **Device Manager**.
4. Look under **Ports (COM & LPT)**. You should see an open port named **Gadget Serial V2.4**. If you **do not** see this open port, follow steps a-c in the **Note** below.
5. Right-click on the **Gadget Serial V2.4** port and choose the **Update Driver Software** option.
6. Choose the **Browse my computer for Driver software** option.
7. Navigate to the hardware/arduino/x86/tools directory. This allows the proper driver file linux-cdc-acm.inf to be installed.
8. Once the driver is successfully installed, Device Manager will show a **Galileo (COMx)** device under **Ports (COM & LPT)**. Note the **COMx** port number as it will be needed in the IDE later. The example below shows COM5.



Note: You may find **Gadget Serial 2.4** under **Other devices** in Device Manager. If this is the case, do the following:

- a. In Windows Explorer, go to `C:\Windows\System32\drivers` and look for `usbser.sys`. If it is present, you can skip these steps. If it is missing, copy `usbser.sys` **from** the archive location identified below:

Windows 7:

`C:\Windows\System32\DriverStore\FileRepository\mdmcpq.inf_amd64_neutral_fbc4a14a6a13d0c8\usbser.sys` (archive file)

Windows 8:

`C:\Windows\System32\DriverStore\FileRepository\mdmcpq.inf_amd64_d9e0b9c4fe044b4d\usbser.sys` (archive file)

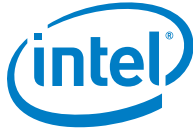
- b. In Windows Explorer, copy **to:** `C:\Windows\System32\drivers`
You may need to provide Administrator Permission to complete the copy.
- c. Once `usbser.sys` is copied, continue with [step 5](#) above.

Linux:

1. If not done already, connect the 5V power cable to the board and to a power outlet. Wait for the board to boot.

Note: **Always** connect the 5V power before the USB connection.

2. Connect the USB cable to the USB Client Port (closest to the Ethernet) and to a PC.
3. Check that ACM port is available by typing in the terminal `ls /dev/ttyACM*`



Note: If the `/dev/ttyACM0` port is not present, follow the procedure below:

- a. Create a file `etc/udev/rules.d/50-arduino.rules` and add the following:
`KERNEL=="ttyACM[0-9]*", MODE="0666"`
- b. Restart the udev with the following command: `sudo service udev restart`
If you are using a virtual machine (VM), you may need to reboot Linux within the VM.

4. Launch the IDE.

The first time you compile and install a sketch, the Galileo software will be installed automatically. This requires appropriate file and directory access permissions. You may be required to run the IDE as root or using `sudo arduino` depending on your Linux distribution.

5. Select your board via **Tools > Board > Intel® Galileo**
6. Select the port **Tools > Serial Port > /dev/ttyACM0**

Note: If the serial `ttyACM0` is not available, make sure you have the proper rights to access the ports. If the serial port is disabled on your IDE, exit the IDE and execute again using the command: `sudo arduino`

OS X:

The Galileo board is supported by the OS X built-in USB drivers, however, the board has to boot for it to show up because the port is driven by software on the board.

1. If not done already, connect the 5V power cable to the board and to a power outlet. Wait for the board to boot.

Note: **Always** connect the 5V power before the USB connection.

2. Connect the USB cable to the USB Client Port (closest to the Ethernet) and to your Mac. Check the **System Profiler > USB** setting to be sure that **Gadget Serial** is selected. If you are installing a new version of the IDE, you may need to re-select this setting.
3. In the Arduino IDE, the correct serial port shows in the **Tools > Serial Port** menu as `/dev/cu.usbmodemnnnnn` where `nnnnn` is a number such as `fd121`.

Do not select the `/dev/tty` port.

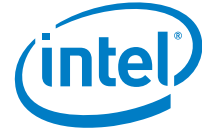
5. Launch the Arduino IDE application

Windows: In the folder `arduino-1.5.3`, double-click `arduino.exe`

Linux: In the folder `arduino-1.5.3`, launch Arduino by executing `./arduino`

Note: You may see an error stating Java is not present. If you are using Ubuntu, run the command `sudo apt-get install default-jre` to clear this error.

OS X: Open the Launchpad from the dock and click on the Arduino application icon.



6. Update your board firmware

The IDE contains the release-specific firmware for your board. Follow the steps below to update your board firmware using the IDE.

1. Remove all power from the board (USB and 5 V power cord). This makes sure that no sketch is running on the board.
2. Remove the SD card from the board (if it is inserted).
3. Power up the board by plugging in the 5V power supply.
4. Connect the USB cable to the USB Client Port (closest to the Ethernet). Note which COM port it is connected on.
5. Launch the IDE and select the board via **Tools > Board > Intel® Galileo**
6. Select the correct serial port using **Tools > Serial Port**

Note: Do not download any sketch to the board before you upgrade the firmware.

7. Launch the software upgrade using **Help > Firmware Upgrade**
8. A message is displayed asking you to confirm that 5V power cable is plugged in. Click **Yes** if it is connected. If no cable is plugged in, exit the upgrade process by selecting **No**, connect the power, and restart this process.
9. The board can be upgraded to newer software or downgraded to older software. The next message displays the current software version that is on the board and the software version that you are trying to flash onto the board. Select **Yes** to either Upgrade/Downgrade or flash the same software again.
10. The upgrade progress takes about 6 minutes and is displayed in several popup messages. During the upgrade process, you will not have access to the IDE.

Note: The power and USB cables **must** stay connected during the upgrade process.

11. When the upgrade completes, a message is displayed stating **Target Firmware upgraded successfully**. Click **OK** to close the message.

Note: On OS X, you must reboot the IDE before continuing.

If you want to flash a different firmware version that is not contained in the IDE, download the SPI flash image capsule files as described in [Section 2](#).

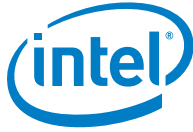
The downloaded capsule should be copied to the following locations, depending upon your operating system. Make sure that there is only one *.cap file in the location below. If you are downloading a new file, rename the previous *.cap to some other extension.

- Windows: Arduino-1.5.3/hardware/tools/x86/bin/
- Linux: Arduino-1.5.3/hardware/tools/
- OS X: Arduino.app/Contents/Resources/Java/hardware/tools/x86/bin/

Note: If you see a failure at any stage of the upgrade, refer to the *Release Notes* for known issues and possible workarounds.

7. Open the blink example

Open the LED blink example sketch: **File > Examples > 1.Basics > Blink**.



8. Select your board

Select **Tools > Board > Intel® Galileo**.

Note: If the IDE was closed and then reopened, it will default to a board that was previously selected. You may need to explicitly select **Intel® Galileo**.

9. Select your serial port

Select the serial device of your board from the **Tools > Serial Port** menu.

Windows: Use the *COMx* number assigned earlier. You can retrieve the port number by navigating to:

Start > Control Panel > System and Security > System > Device Manager
Look under **Ports (COM & LPT)** to see which *COMx* is assigned to **Gadget Serial**.

Linux: Use `/dev/ttyAMC0`

OS X: Use `/dev/cu.usbmodemnnnnnn`

10. Upload the program

Click the **Upload** button in the IDE and wait a few seconds. If the upload is successful, the message **Done uploading.** will appear in the status bar.

A few seconds after the upload finishes, you will see the green GP LED on the board (shown above) start to blink. Congratulations! You've gotten your board up and running.

For more information about Intel® Galileo, including online communities and support, try these links:

<http://www.intel.com/support/galileo>

<http://communities.intel.com/community/makers>

11. Booting your board from an SD card (mandatory for WiFi)

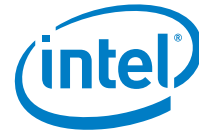
Follow these steps to boot your board from an SD card. This setup also allows you to save your sketch to the Galileo board, so it will be able to repeat sketches after board power-down. (These steps create a persistent `/sketch` folder and `rootfs`.)

Note: If you want to use WiFi, you **must** follow these steps. The WiFi driver is not present in the Linux image in SPI flash because it is too large. The Linux SD image also includes ALSA, V4L2, python, SSH, node.js, and openCV.

Required files are in the `LINUX_IMAGE_FOR_SD_Intel_Galileo_v1.0.0.7z` zip file (download link in [Section 2](#)).

Note: Your SD card **must** meet the following requirements:

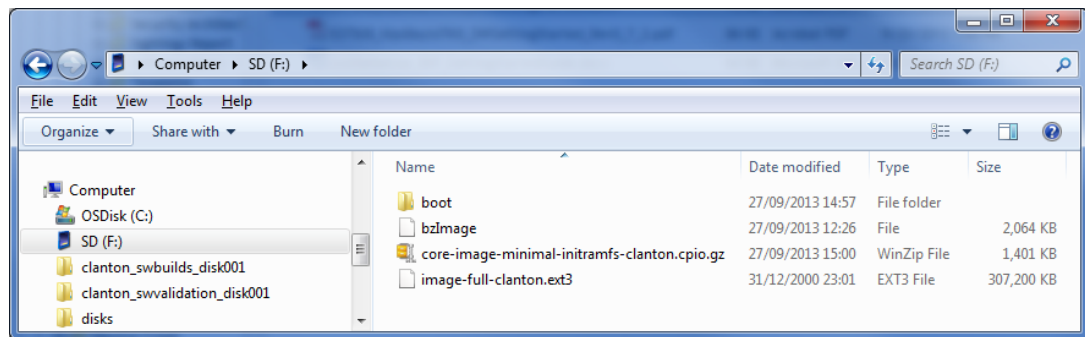
SD card must be formatted as FAT or FAT32.



SD card size must be 32GB (or smaller) and SDHC format. SDXC format is **not** supported.

1. Copy all files and directories from the zip file to your SD card.

You do not need to create a directory on the SD card. The zipfile contains all the necessary files and structure. Be sure it is extracted at the top level of the SD card. See the screenshot below for a sample view.



2. Insert the SD card, then power on the board.

Note: The first time you boot the board may take several minutes. This is expected behavior due to the SSH component creating cryptographic keys on the first boot.

Troubleshooting tips:

To boot from SD/USB, the grub instance embedded in the SPI flash is hardcoded to search for a `boot/grub/grub.conf` file in partition 1 on the SD/USB card. This is compatible with the factory formatting of most SD/USB devices. By default, the UEFI firmware does not try to boot from SD or USB, it is handled by grub.

If you use an SD or USB device that has been reformatted after manufacturing, you might experience problems booting from it. First, try to boot with a different memory device and see if the problem goes away. If you isolate the problem to a specific SD card, you can restore the factory formatting using this tool from the SD association: https://www.sdcard.org/downloads/formatter_4/

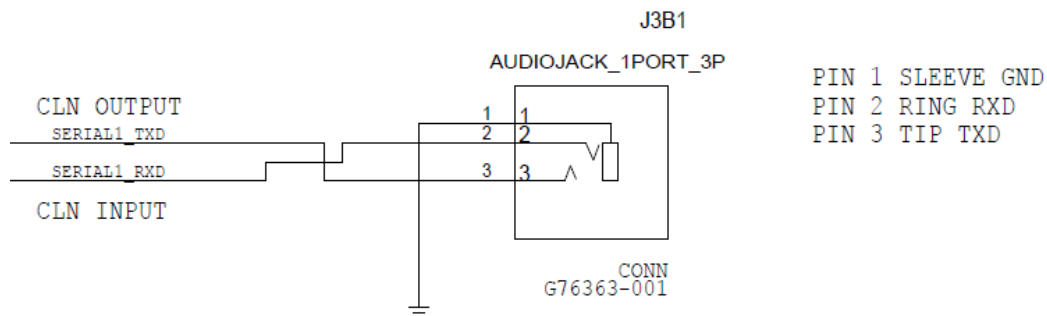
It is not recommended to use normal operating system tools to format flash memory devices.

12. Serial cable (optional)

You can access the embedded Linux using a serial cable connected to the 3.5 mm audiojack port on the Galileo board. Be sure that the cable is RS-232 format and not TTL format (details here: <https://www.sparkfun.com/tutorials/215>).

For a 3 pin audio jack, the connections are shown below:

- Pin 1 Sleeve = Ground
- Pin 2 Ring = RXD (input)
- Pin 3 Tip = TXD (output)



13. Manually update your board firmware (optional)

Note: This method has been replaced by functionality in the IDE.

You must update the firmware the first time you use the board to ensure the firmware on the board is in sync with the IDE.

The Intel® Quark™ BSP software provides a mechanism to update SPI flash contents based on EDKII capsules. These capsules contain a BIOS, bootloader, and compressed Linux run-time system sufficient to boot a Quark-based board, such as the Intel® Galileo board. For details, see the *Intel® Quark™ SoC X1000 BSP Build Guide* [<https://communities.intel.com/docs/DOC-21882>]

14. Troubleshooting Tips

Problems with COM ports

There is a known issue with Windows that it creates a new COM port every time a device is connected. In cases when there are many unused COM ports, the Galileo board may stop working and/or the IDE may return the error below:

```
/dev/ttyS24: No such file or directory
```

One workaround is to delete all unused and hidden COM ports, including the drivers.

1. Disconnect the USB cable from the Galileo board.
2. On your Windows host machine, open a command prompt window and run it as administrator as follows:

Click the **Start** Menu, and type **Command**. Right-click the command prompt icon and select **Run as administrator**.

3. In the command prompt window, type the commands below:

```
set devmgr_show_nonpresent_devices=1    to display hidden devices  
start devmgmt.msc                      to start the device manager
```

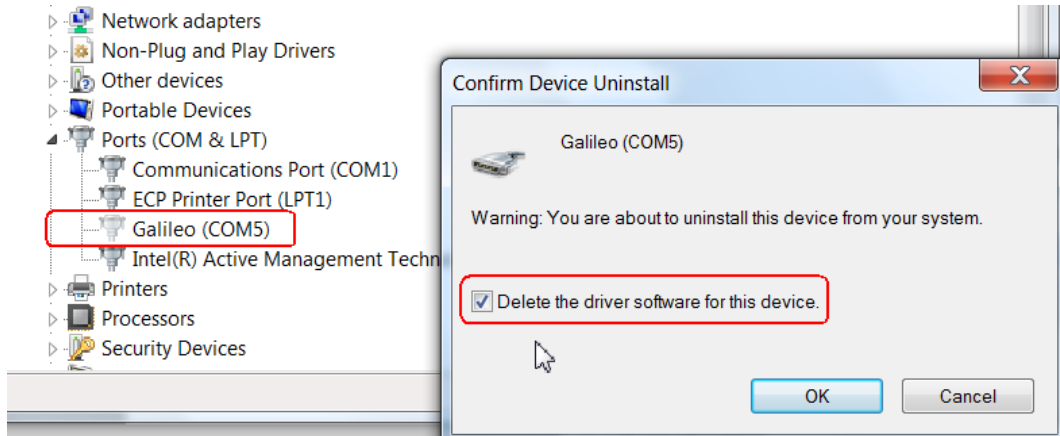
4. In the Device Manager window, select **View > Show Hidden Devices**

Unused COM ports will be grayed-out.



5. Select a COM port to delete by right-clicking it and selecting **Uninstall**. In the dialog, check the box "Delete the driver software for this device" and click **OK**.

Repeat this step for each unused COM port. Close Device Manager when complete.



6. Connect the USB cable to the Galileo board. You will be prompted to reinstall the driver as described in [Section 4](#).

Problems with Linux Virtual Machines

You may experience similar COM port issues if you are running a Linux Virtual Machine on a Windows host. Ensure that the COM port is completely functional using the Windows IDE and follow the steps in the previous tip. Alternatively, you can use a terminal emulator such as Teraterm or Putty to test serial port communication.



Revision History

Date	Document Revision	Description
22 May 2014	006	Updates for Package Version: 1.0.0 for Arduino IDE v1.5.3 Added Section 12, Serial cable (optional) . Removed obsolete instructions in Section 13 . Added Section 14, Troubleshooting Tips .
23 December 2013	005	Updated kit contents list. Updated filename/size of Board Support Package (BSP) sources including Yocto archive. Added Troubleshooting tip for Booting from SD card section.
16 October 2013	004	Updated download URL and other minor text updates.
10 October 2013	003	Updated file names and sizes.
04 October 2013	002	Added links to maker community and support pages. Updated board picture with additional callouts.
02 October 2013	001	First external release: Package Version: 0.7.5 for Arduino IDE v1.5.3

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document 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.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

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

Intel, the Intel logo, and Quark are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Copyright © 2014, Intel Corporation. All rights reserved.