Intel® Galileo and Intel® Galileo Gen 2

Getting Started Guide

March 2015

Revision 008

Document Number: 329685-008

Revision History

Revision	Description	Date
001	First external release: Package version: 0.7.5 for Arduino* IDE v1.5.3	October 2, 2013
002	Added links to maker community and support pages. Updated board picture with additional callouts.	October 4, 2013
003	Updated file names and sizes.	October 10, 2013
004	Updated download URL and other minor text updates.	October 16, 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.	December 23, 2013
006	Updated for package version 1.0.0 for Arduino* IDE v1.5.3. Added Section 12, Serial console (optional). Removed obsolete instructions in Section 13. Added Section 14, Troubleshooting.	May 22, 2014
007	Updated for package version 1.0.2 for Arduino* IDE v1.5.3. Added support for the Intel® Galileo Gen 2 board.	June 23, 2014
008	Updated for package version 1.6.0+Intel. Added support for Arduino* IDE v1.6.0.	March 13, 2015

§

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. You must download it from the Intel website.

Release 1.0.2 adds support for the second generation Intel® Galileo board, also called Intel® Galileo Gen 2. In this document, for convenience:

- *Software* and *software package* are used as generic terms for the IDE software that runs on both Intel® Galileo and Intel® Galileo Gen 2 boards.
- *Board* is used as a generic term when either Intel® Galileo or Intel® Galileo Gen 2 boards can be used. If the instructions are board-specific, the exact model is identified.

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 Wi-Fi).
12. Serial console (optional).
13. Manually update your board firmware (optional).
14. Troubleshooting

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 Intel® Galileo kit contains:

- 1x Intel® Galileo Customer Reference Board (CRB) (Fab D with blue PCB)
- 1x 5 V power supply with cable

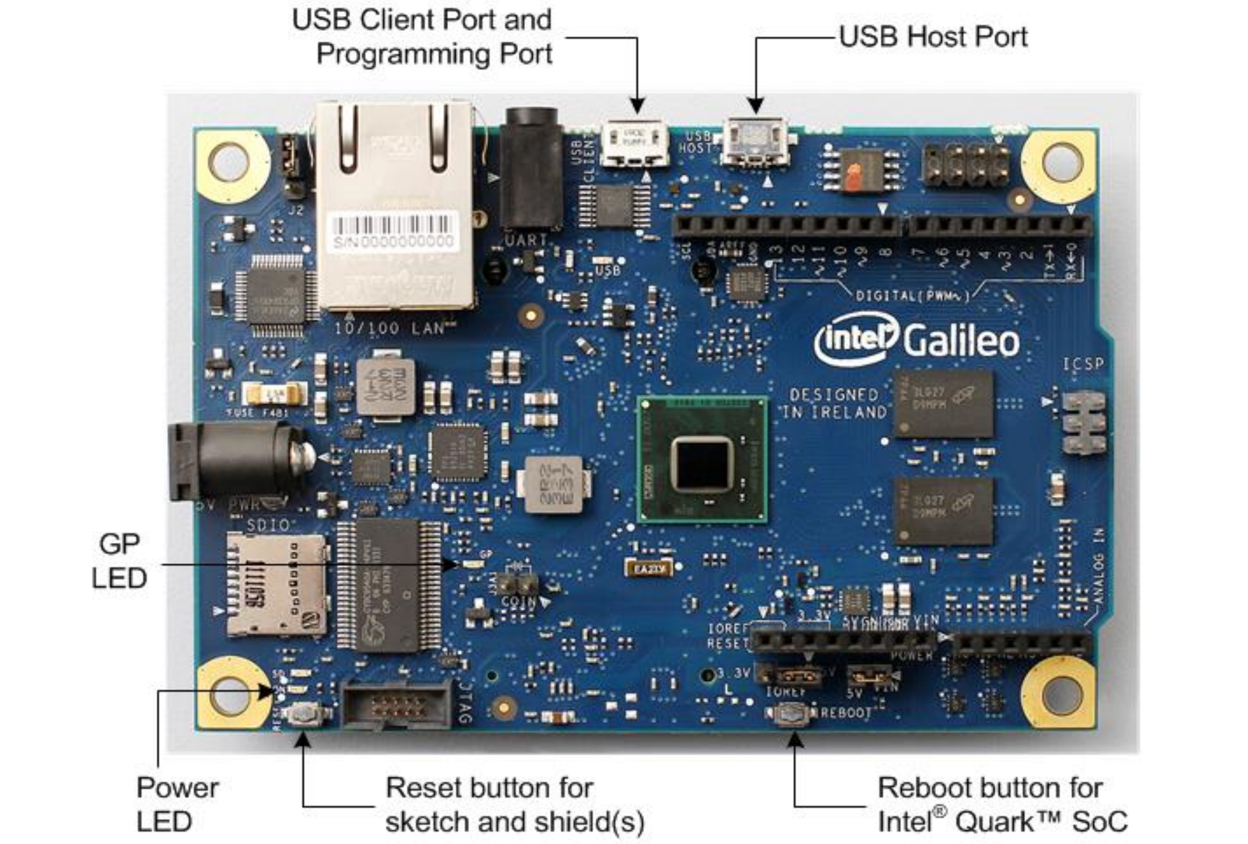
Warning: You must use a power supply to power the board! You will damage the board if you power it with a USB cable.

Warning: The power supplies are NOT interchangeable between Intel® Galileo and Intel® Galileo Gen 2 boards. The Intel® Galileo Gen 2 power supply is 12 V and it will permanently damage first generation Galileo boards.

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

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

Figure 1 Intel® Galileo board


USB Client Port and Programming Port
USB Host Port
GP LED
Power LED
Reset button for sketch and shield(s)
Reboot button for Intel® Quark™ SoC

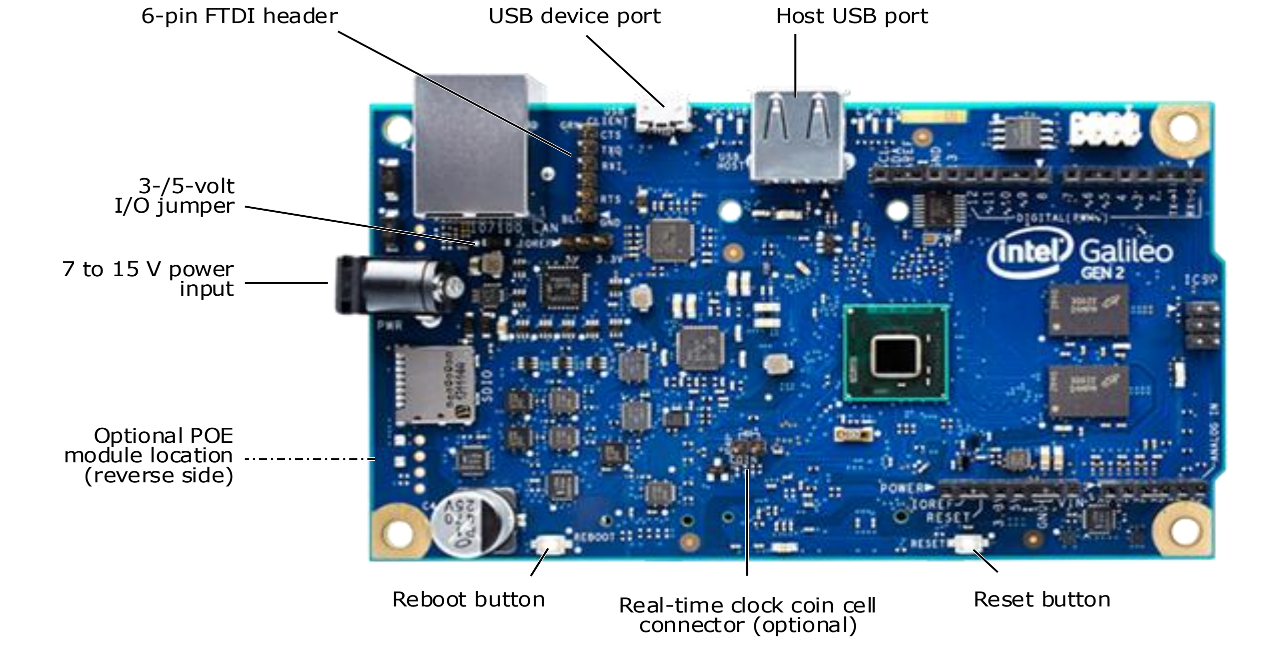
The **Intel® Galileo Gen 2** kit contains:

- 1x Intel® Galileo Gen 2 Customer Reference Board (CRB) (Gen 2 marking)
- 1x 12 V power supply with cable

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

Warning: The power supplies are *NOT* interchangeable between Intel® Galileo and Intel® Galileo Gen 2 boards. The Intel® Galileo Gen 2 power supply is 12 V, and **it will permanently damage first generation boards**.

Figure 2 Intel® Galileo Gen 2 board



2. Download the Arduino environment and board firmware

Download the latest IDE and firmware files from the Intel® Download Center:

https://downloadcenter.intel.com/download/24748/Intel-Galileo-Firmware-and-Drivers-1-0-4

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

Note: When unzipping the packages, extract the contents 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:
  - `IntelArduino-1.6.0+Intel-Linux32.txz (141 MB)`
  - `IntelArduino-1.6.0+Intel-Linux64.txz (146 MB)`
  - `IntelArduino-1.6.0+Intel-OSX.zip (228 MB)`
  - `IntelArduino-1.6.0+Intel-Windows.7z (304 MB)`
  - `IntelArduino-1.6.0+Intel-Windows.exe (139 MB)`
- (Mandatory for Wi-Fi): Files for booting board from SD card. See Section 11. (Download *SDCard.1.0.4.tar.bz2* from *https://downloadcenter.intel.com/download/24355/Intel-Galileo-Software-Package-1-0-4*.)
- (Optional) Linux* BSP (Board Support Package) sources for Intel® Quark™ SoC-based systems. (Download BSP-Patches-and-Build_Instructions.1.0.4.txt: *https://downloadcenter.intel.com/downloads/24355/Intel-Galileo-Software-Package-1-0-4*.
- (Optional) Board Support Package (BSP) sources including Yocto archive: *https://downloadcenter.intel.com/download/24702/Intel-Galileo-GPL-Compliance-files-1-0-4*. The BSP build process requires additional 30 GB of free disk space. (This is a legal compliance package. You cannot build BSP with this package.)

If you are working with the BSP sources, see the *Intel® Quark™ SoC X1000 BSP Build Guide*:

- *http://www.intel.com/support/processors/quark/sb/CS-035175.htm*

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.

Mac 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.

Linux*:

1. In your home directory, extract the package with the appropriate command:

   ```
   tar -xf IntelArduino-1.6.0+Intel-Linux32.txz
   ```

   ...or:

   ```
   tar -xf IntelArduino-1.6.0+Intel-Linux64.txz
   ```

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 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.

3. Connect the board

Intel® Galileo 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.

1. Connect the 5 V power cable to the Galileo board and to a power outlet.
2. Always connect the power before any other connection. The Power LED shown above (board label = ON) will turn on.

Intel® Galileo Gen 2 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.

1. Connect the 12 V power cable to the Intel® Galileo Gen 2 board and to a power outlet.
2. Always connect the power before any other connection. The power LED shown above (board label = ON) will turn on.

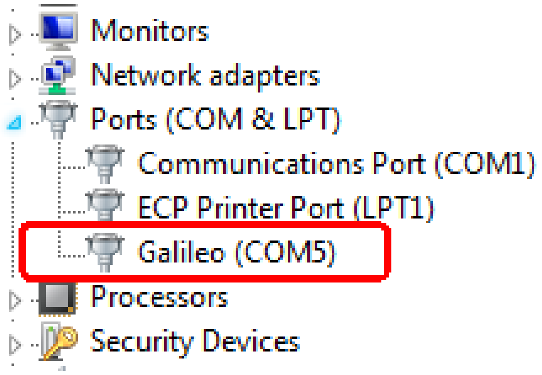
4. *Install the drivers and other software*

Windows:

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

   Note: *Always* connect the 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 through 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 so, 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 4 above.

Mac OS X:

The 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 power cable to the board and to a power outlet. Wait for the board to boot.

Note: *Always* connect the 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.usbmodem*nnnnn where *nnnnn* is a number such as *fd121*. Do not select the */dev/tty* port.

Linux:

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

   Note: Always connect the 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*`. If the `/dev/ttyACM0` port is not present, do the following:
   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.
   - For Intel® Galileo boards, select *Tools* > *Board* > *Intel® Galileo*.
   - For Intel® Galileo Gen 2 boards, select *Tools* > *Board* > *Intel® Galileo (Gen 2)*.
6. Select the port *Tools* > *Serial Port* > */dev/ttyAMC0*

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`

5. Launch the Arduino IDE application

Windows: In the folder `arduino-1.6.0`, double-click *arduino.exe*

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

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.

6. Update your board firmware

We have created a firmware updater tool that makes updating Intel® Galileo boards easy. Download the *Intel® Galileo Firmware Updater Tool User Guide* and the tool itself from our download center:

- *Intel® Galileo Firmware Updater Tool User Guide* (332076)
- *http://downloadcenter.intel.com/download/24748*

7. Open the blink example

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

8. Select your board

For Intel® Galileo boards, select *Tools* > *Board* > *Intel® Galileo*.

For Intel® Galileo Gen 2 boards, select *Tools* > *Board* > *Intel® Galileo (Gen 2)*.

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 your board.

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*.

Mac OS X: Use /dev/cu.usbmodem*nnnnn*

Linux: Use `/dev/ttyAMC0`

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 Wi-Fi)

Follow these steps to boot your board from an SD card. This setup also allows you to save your sketch to the 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 Wi-Fi, you *must* follow these steps. The Wi-Fi 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_vn.n.n.7z* zip file (*n.n.n* indicates the version; see the download link in Section 0).

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

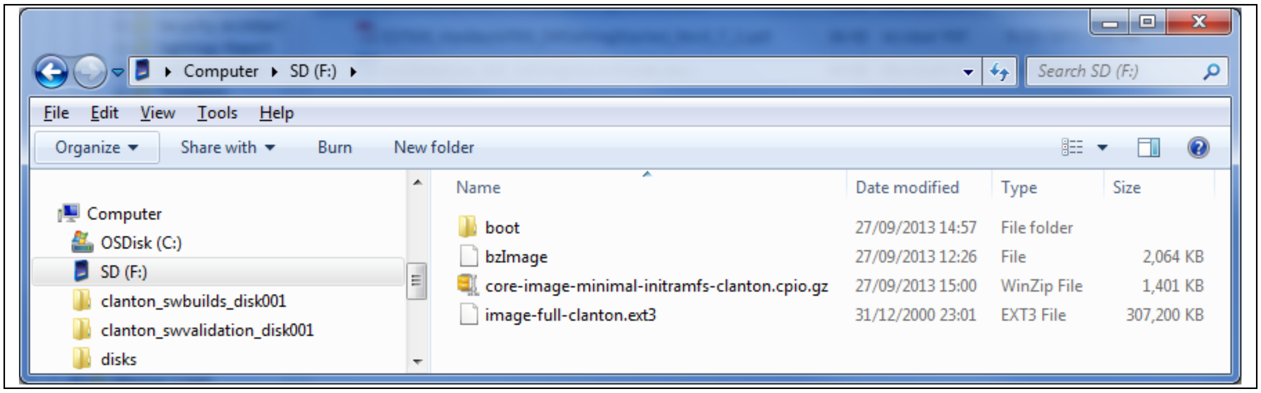
– SD card must be formatted as FAT or FAT32.

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

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

   Note: 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 Figure 3 for a sample view.

Figure 3 SD card contents


Computer ▸ SD (F:) ▸
Search SD (F:)
File Edit View Tools Help
Organize ▾ Share with ▾ Burn New folder
Computer
OSDisk (C:)
SD (F:)
clanton_swbuilds_disk001
clanton_swvalidation_disk001
disks

Name	Date modified	Type	Size
boot	27/09/2013 14:57	File folder	
bzImage	27/09/2013 12:26	File	2,064 KB
core-image-minimal-initramfs-clanton.cpio.gz	27/09/2013 15:00	WinZip File	1,401 KB
image-full-clanton.ext3	31/12/2000 23:01	EXT3 File	307,200 KB

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 console (optional)

If you are using the Intel® Galileo Gen 2 board, the console serial connector has changed to a 6-pin header.

The new serial console connector on the Intel® Galileo Gen 2 board is similar to established types used on the Arduino board and other development boards. It is directly compatible with the FTDI cable connector for Arduino TTL serial cable products. The connector slips right on, and the Intel® Galileo Gen 2 board has the identifiers *GRN* and *BLK* silkscreened onto the surface, to help you can align the cable with the wire colors on the FTDI connector hood. The 6-pin header operates at 3.3 V TTL levels.

- The new connector mates with FTDI part number **TTL-232R-3V3**.
- The new connector is compatible with many FTDI breakout boards when configured for 3.3 V operation.

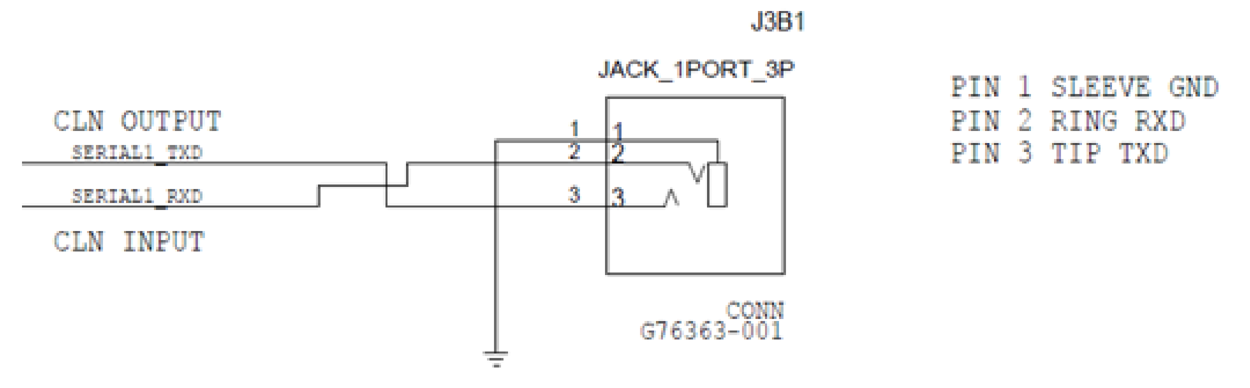
Note: For details, see the *Technical Note: Intel® Galileo board (Gen 2) Console Serial Connector Changes* (#330738).

You can access the embedded Linux using a serial cable connected to the 3.5 mm jack port on the 1st generation 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 jack, the connections are shown below:

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

Figure 4 3-pin jack connections



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

Problems with COM ports

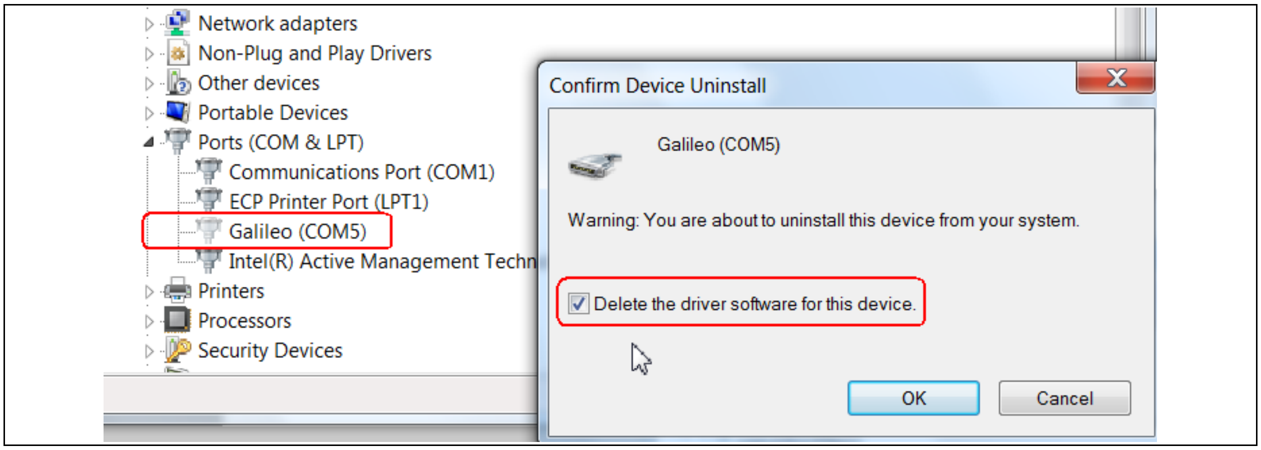
There is a known issue with Windows that it creates a new COM port every time a device is connected. In cases when there are many unused COM ports, the board may stop working and/or the IDE may return the error below:

```
/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 board.
2. On your Windows host machine, open a command prompt window and run it as administrator as follows:
   a. Click the *Start* Menu, and type *Command*.
   b. 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.

Figure 5 Confirm uninstall



Confirm Device Uninstall

Galileo (COM5)

Warning: You are about to uninstall this device from your system.

☑ Delete the driver software for this device.

6. Connect the USB cable to the 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.

§