



Intel® Galileo Software

Package Version: 1.0.2 for Arduino IDE v1.5.3
Release Notes

23 June 2014



1 Introduction

This document describes extensions and deviations from the functionality described in the *Intel® Galileo Board Getting Started Guide*, available at: www.intel.com/support/go/galileo

This software release supports the following hardware and software:

- Intel® Galileo Customer Reference Board (CRB), Fab D with blue PCB
- Intel® Galileo (Gen 2) Customer Reference Board (CRB), Gen 2 marking
- Intel® Galileo software v1.0.2 for Arduino Integrated Development Environment (IDE) v1.5.3

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.

Features in this release are described in [Section 1.4](#).

Release 1.0.2 adds support for the 2nd 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 Galileo (Gen 2) boards.
- *Board* is used as a generic term when either Intel® Galileo or Galileo (Gen 2) boards can be used. If the instructions are board-specific, the exact model is identified.

1.1 Downloading the software release

Download the latest Arduino IDE and firmware files here:

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

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_v1.0.2.tgz (73.9 MB)
 - Intel_Galileo_Arduino_SW_1.5.3_on_Linux64bit_v1.0.2.tgz (75.2 MB)
 - Intel_Galileo_Arduino_SW_1.5.3_on_MacOSX_v1.0.2.zip (56.0 MB)
 - Intel_Galileo_Arduino_SW_1.5.3_on_Windows_v1.0.2.zip (106.9 MB)
- (Mandatory for WiFi) Files for booting board from SD card. See [Section 11](#).
LINUX_IMAGE_FOR_SD_Intel_Galileo_v1.0.2.7z (51.2 MB)
- (Optional) Linux* BSP (Board Support Package) sources for Intel® Quark™ SoC-based systems.
- Board_Support_Package_Sources_for_Intel_Quark_v1.0.2.7z (2.53 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_v1.0.2_full_yocto_archive.tar.gz (2.8 GB)

If you are running the IDE software, see the *Intel® Galileo Board Getting Started Guide*. [<https://communities.intel.com/docs/DOC-21838>]

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

1.2 Supported operating systems

The software release has been tested on the following operating systems.

Operating System	Version
Windows*	Windows* 7 (32-bit & 64-bit), Windows* 8, Windows* 8.1
Linux*	Ubuntu 12.04 (32-bit and 64-bit)
OS X*	Built on: OS X version 10.8.5 Tested: OS X 10.6.8, 10.7.5, and 10.9.2 developer preview

1.3 Supported Arduino libraries

The software release supports the following Arduino libraries:

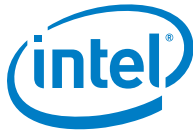
- SPI
- EEPROM
- UART
- GPIO
- Wi-Fi
- Servo
- USB Host

1.4 Features in Release 1.0.2

- Added support for the Intel® Galileo (Gen 2) Customer Reference Board (CRB).
- The Intel® Quark BSP package version 1.0.1 is patched to update the build to OpenSSL 1.0.1h. This addresses security issues announced previously. We recommend updating both the host IDE software and the SD card runtime to address OpenSSL fixes and updating the firmware on the Galileo for consistency.
- Several fixes were incorporated into the Arduino core libraries.
- Described library support for `tone()` and `pulseIn()`, see below for details.

1.4.1 Galileo-specific implementation of `tone()`

The standard `tone()` implementation is described at:



<http://arduino.cc/en/reference/tone>

There are several Galileo-specific implementation differences to be noted:

- Intel® Galileo and Galileo (Gen 2) board differences:
 - On the first generation Galileo board (Fab D), `tone()` produces tones, but most pins will not be accurate because of the IO speed limitation of the Cypress expander chip. Pins 2 and 3 do generate accurate tones.
 - On the Galileo Gen 2 board, `tone()` generates accurate tone frequencies with the exception of pins 7 and 8.
- The method of `tone(pin, frequency, duration)` blocking call.
 - `toneNonBlocking(pin, frequency, duration)` is non-blocking.
 - `tone(pin, frequency)` is also non-blocking and generates a tone at that pin until a `noTone(pin)` is called for that pin.
 - Since each non-blocking call of the Tone API is a thread, it only generates accurate tones for the when the thread has the CPU.
- The Galileo Tone implementation supports generating multiple tones with different frequencies, if non-blocking calls are used. However, this also means that each thread gets less CPU time and each pin generates the tone only when their corresponding thread has the CPU.

1.4.2 Galileo-specific implementation of `pulseIn()`

The standard `pulseIn()` implementation is described at:

<http://arduino.cc/en/Reference/pulseIn>

There are several Galileo-specific implementation differences to be noted:

- Intel® Galileo or Galileo (Gen 2) board differences:
 - On the first generation Galileo board (Fab D), `pulseIn()` is only accurate for pins 2 and 3.
 - On the Galileo Gen 2 board, `pulseIn()` is accurate for all digital pins, except for pins 7 and 8.
- Works reliably on pulses from 3 microseconds to 2^{32} microseconds (71.58) minutes.
- Accuracy of +/- 2 microseconds.
- Does not have the accuracy limitation for longer pulses.
- Occasionally an inaccurate longer pulse value(+ 40µs) is returned due to the fact that the sketch is not the only running process under Linux.

1.5 Features in previous releases

1.5.1 Features in Release 1.0.0:

- Firmware update (*.cap file) now installs v1.0.0 capsule image, and includes support for installing firmware on boards with firmware released after 0.7.5.
- Improved compatibility of Arduino libraries with standard C++ math and string classes.



- Added `min()` and `max()` to `Arduino.h`
- Null definitions for `interrupts()` and `noInterrupts()` macros in `OSAbstract.h`
- Fixed issues with pin 0 and pin 1.
- Added support to the library for interacting with the ADC logic on the board.
- Added `UtilTime` module to library.
- Added library support for `OneWire` and the Dallas Temperature ICs.
- Added examples for the Ethernet and Wi-Fi modules to the library.

1.5.1.1 Fixed Heartbleed vulnerability in Release 1.0.0

The 1.0.0 release included updates for an OpenSSL security issue, CVE-2014-0160. This issue has been addressed by updating OpenSSL 1.0.1e with patches to close the TLS issue. In order to mitigate the risk of exposed certificate keys and user data, users should remove existing versions of the Galileo Arduino IDE for Linux, install this update, and use the 1.0.0 updated version of the SD Card runtime for Wi-Fi support.

Galileo components from October 2013 release and vulnerability to CVE-2014-0160:

- Galileo SD Card runtime 0.7.5, released October, 2013 (downloadable) Possibly vulnerable to CVE-2014-0160.
- Galileo Arduino IDE for 64-bit Linux 0.7.5, released October, 2013 (downloadable) Possibly vulnerable to CVE-2014-0160.
- Galileo Arduino IDE for 32-bit Linux 0.7.5, released October, 2013 (downloadable) Possibly vulnerable to CVE-2014-0160.

The vulnerability has been corrected in this release. No other versions or products are affected, including the on-board Linux built into the Galileo board.

1.5.2 Features in Release 0.7.5:

- Board firmware (*.cap file) installation using the IDE. See the *Getting Started Guide* for details.
- Wi-Fi support (see [Section 1.5.2.1](#))
- Sketch persistence (SD only, see [Section 1.5.2.2](#))
- USB input device support (see [Section 1.5.2.3](#))
- Fast GPIO support
- I2C enhancements for PWM configuration
- Ethernet
- Linux SD additions

1.5.2.1 Wi-Fi support

Note: Integrated Wi-Fi functionality (**not** Wi-Fi shield) is supported in this release.



Validated on the Intel® Centrino® Wireless-N 135 adapter described at:
<http://www.intel.com/content/www/us/en/wireless-products/centrino-wireless-n-135.html>

To use the Intel® Centrino® Wireless-N 135 adapter, run the Linux OS from mass storage (microSD) as described in the *Getting Started Guide, Booting your board from an SD card*.

You can use the existing Wi-Fi library and sketches that are in the IDE. Details are available here: <http://arduino.cc/en/Reference/WiFi>

The library uses standard Linux wireless tools such as `iwconfig`, `iwgetid`, `ifconfig`, and `wpa_passphrase`.

Wi-Fi has been tested with the Intel® Centrino® Wireless-N 135 adapter but may work on any Linux Wi-Fi hardware used by the standard Linux tools referenced above.

Runs out of the box with this software release (Linux) and `udhcpc` client.

Not tested:

- WEP encryption
- Using Intel® Galileo as an Access Point (AP)
- Heavy traffic usage
- DNS
- Many other extended use scenarios

1.5.2.2 SD Library

Note: SD Library is supported on SD card only (**not** SD interfaces on shields) in this release.

All of the functions listed below from the Arduino library reference have been implemented.

```
class File
  File(FILE *f, char *name);
  File(void);
  ~File(void);
  virtual size_t write(uint8_t);
  virtual size_t write(const uint8_t *buf, size_t size);
  virtual int read();
  virtual int peek();
  virtual int available();
  virtual void flush();
  int read(void *buf, uint16_t nbyte);
  boolean seek(uint32_t pos);
  uint32_t position();
  uint32_t size();
  void close();
  operator bool();
  char * name();
  boolean isDirectory(void);
  File openNextFile();
  void rewindDirectory(void);
```

```
class SDClass
```



```
boolean begin(uint8_t csPin = 0);
File open(const char *filepath, uint8_t mode = FILE_READ);
boolean exists(char *filepath);
boolean mkdir(char *filepath);
boolean remove(char *filepath);
boolean rmdir(char *filepath);
```

Status of example sketches:

- Card Info – missing functionality from `SDFFile` library which has not been implemented.
- Data logger – uses `File.println()` function which is inherited from `SDFFile` and is not advertised on the Arduino website.
- Dump file – works correctly
- Read write – works correctly

1.6 Supported shields

For a list of the shields that were tested, see the Intel® Galileo Shields List, which is posted here: <https://communities.intel.com/docs/DOC-21855>

1.7 Supported sketches

The following Arduino-based sketches have been tested on at least one of the three supported OSes (Windows, Linux, OS X):

Basic Arduino Examples		
AnalogInOutSerial	AnalogInput	AnalogReadSerial
Arrays	ASCIITable	BareMinimum
Blink	BlinkWithoutDelay	Button
Calibration	CharacterAnalysis	Debounce
DigitalInputPullup	DigitalReadSerial	Fade
Fading	ForLoopIteration	Graph
IfStatementConditional	PhysicalPixel	ReadAnalogVoltage
SerialCallResponse	SerialCallResponseASCII	SerialEvent
StateChangeDetection	StringAppendOperator	StringCaseChanges
StringLength	StringLengthTrim	StringStartsWithEndsWith
StringSubstring	StringToInt	StringToIntRGB
switchCase	switchCase2	VirtualColorMixer
WhileStatementConditional		

Sketches from Arduino Starter Kit		
p02_SpaceShipInterface	p03_LoveOMeter	p04_ColorMixingLamp,



Sketches from Arduino Starter Kit		
p08_DigitalHourglass	p09_MotorizedPinwheel	p10_Zoetrope
p14_TweakTheArduinoLogo	p15_HackingButtons	

Arduino Library Sketches		
Autoscroll	Blink	ConnectNoEncryption
ConnectWithWPA	Cursor	Display
DumpFile	EEPROM_clear	EEPROM_read
EEPROM_write	HelloWorld	listfiles
ScanNetworks	Scroll	SerialDisplay
setCursor	SimpleWebServerWiFi	TextDirection
WiFiWebClient	WiFiWebServer	

1.8 4.20g security disclaimer

This product is designed and configured as a developer device. As such it includes only basic functionality and requires you as the developer to add the capabilities necessary for your particular uses. The device does not include any specific or enhanced security functionality and is configured by default to be openly accessible to aid your development. Specifically the device does not include or support any limitations or controls on what software can be executed or booted by the hardware. There is no secure boot, secure update or other firmware control mechanisms. This means that anyone with physical access to the device can change the software (firmware) running on the device. By default the administrative access to the device, including the supplied Linux software images, is also not access restricted. Anyone with physical access to the device can access administrative privileges (i.e. "root" access) to inspect or modify the device without a password or other authentication. If your use requires these interfaces be secured, you are responsible for adding or configuring capabilities to do so.

1.9 Known issues in the release

Issue #	Section	Description
54396	1.9.1	Windows 7 IDE - COM port stops working
54857	1.9.2	Unzipping packages with long file paths
54863	1.9.3	Timeout errors cause sketch download failure or firmware upgrade failure
54935	1.9.4	Using Serial.* without serial console will block sketch
55278	1.9.5	COM port disappears and IDE needs to be closed
55303	1.9.6	attachInterrupt HIGH/LOW triggering mode not supported



Issue #	Section	Description
55458	1.9.7	I2C adapter fails with "controller timed out" error message
55503	1.9.8	When power is disconnected and reconnected "USB Device not detected" error occurs.
55527	1.9.9	OS X IDE zip file must be unzipped on Mac
55631	1.9.10	SPI LSB-first mode not supported
55634	1.9.11	UART doesn't support non printable characters
77426 [GAL-21]	1.9.12	Cannot boot from SD card when a USB device is in the USB host port
77507 [GAL-75]	1.9.13	Galileo Gen 2 only: IRQs missed if pulses too close together on GPIO expanders
78738 [GAL-84]:	1.9.14	I2C/GPIO level-triggered interrupts cause system hang
N/A	1.9.15	Arduino IDE menus are greyed out on Mac after upgrade

1.9.1 54396: Windows 7 IDE - COM port stops working

On Windows 7, you may see an error where the Galileo board's COM port stops working.

The IDE will display this error text (details are **highlighted**):

```
#mv the downloaded file to /sketch/sketch.elf
target_download_name="${host_file_name##*/}"
echo "Moving downloaded file to /sketch/sketch.elf on target"
Moving downloaded file to /sketch/sketch.elf on target
#$fixed_path/lsz.exe --escape -c "cp sketch /sketch/sketch.elf" <>
$TTY_PORT_ID 1>&0
$fixed_path/lsz.exe --escape -c "mv $target_download_name
/sketch/sketch.elf; chmod +x /sketch/sketch.elf" <> $TTY_PORT_ID 1>&0
C:\Users\enyquist\Downloads\arduino-
1.5.3\hardware\arduino\x86\tools\izmir\clupload_win.sh: line 39:
/dev/ttyS40: No such file or directory
```

Workaround: Reboot your Windows host PC. The USB serial port should then be present and the IDE will be able to upload sketches to the Galileo board again.

1.9.2 54857: Unzipping packages with long file paths

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/>).



1.9.3 **54863: Timeout errors cause sketch download failure or firmware upgrade failure**

Both issues described below are related to timeout errors:

- Sketch downloads to the board may fail after multiple sketches have been downloaded. If this happens, reset the board.
- If the firmware upgrade is stuck for more than 10 minutes or if you get any upgrade error, unplug the cables, and retry the firmware upgrade procedure again.

1.9.4 **54935: Using Serial.* without serial console will block sketch**

If a sketch uses Serial.* to output to the IDE via USB, the IDE serial console should be open. If the console is not open, then the sketch loop may block and the loop will stop. If using persistent sketches with no IDE connected, the same blocking behavior will occur.

Workaround: do not output to the IDE Serial Console port unless connected to the IDE with Serial Console monitor open.

1.9.5 **55278: COM port disappears and IDE needs to be closed**

USB CDC sometimes doesn't enumerate COM port on full board reset (power jack or REBOOT button).

Workaround: Close the IDE, reboot the board, and restart IDE.

1.9.6 **55303: attachInterrupt HIGH/LOW triggering mode not supported**

High and low level triggered interrupts are not available in this release, therefore, only edge-triggered interrupts can be used.

Workaround: Do not use level-triggered interrupts.

1.9.7 **55458: I2C adapter fails with "controller timed out" error message**

Very rarely, the I2C driver has been seen failing with the following error string:
`intel_cln_gip 0000:00:15.2: controller timed out`

Implication: When the error occurs, GPIOs are unusable.

Workaround: Cold reset the board by unplugging and replugging the power supply.



1.9.8 55503: When power is disconnected and reconnected “USB Device not detected” error occurs

On Windows and Linux, it has been reported that a device already enumerated will become not detected after the power is disconnected and reconnected.

This is planned to be fixed in a future release.

1.9.9 55527: OS X IDE zip file must be unzipped on Mac

You must unzip the OS X IDE on your Mac's normal hard disk due to symbolic links within the IDE. Copying or transferring the application must be done using the OS X Finder. Any other methods for copying or moving the application may damage it.

Workaround: Download the zip file on your Mac's hard drive and then unzip it.

1.9.10 55631: SPI LSB-first mode not supported

SPI LSB-first mode is not supported, therefore, sketches using SPI in LSB-first mode will fail.

Workaround: switch to MSB-first mode if your device supports it.

1.9.11 55634: UART doesn't support non printable characters

`Serial.print()` does not support non-ASCII characters, therefore, sketches that use non-printable characters will malfunction.

1.9.12 77426 [GAL-21]: Cannot boot from SD card when a USB device is in the USB host port

When a USB device is connected to the host port, the Galileo board cannot boot from an SD card. The following grub error is returned:

```
No "/boot/grub/grub.conf" file found on 1st USB or SD device
```

Workaround: Remove the USB device first, then boot from the SD card.

1.9.13 77507 [GAL-75]: Galileo Gen 2 only: IRQs missed if pulses too close together on GPIO expanders

On the Intel® Galileo Gen 2 boards, there is a PCAL9555A GPIO expander that provides interrupt support for some of the digital I/O header pins. Those pins are IO2-3 and IO14-9, and also the shield reset button.

If those pins are configured to generate interrupt notifications, and if the rate of interrupt trigger events (e.g. falling/rising edge signals on the pin) exceeds a combined rate of approximately 1000 interrupt events per second, the interrupt



notifications from the PCAL9555A may stop working. Notifications for subsequent interrupt events will not be received by software.

Possible implications:

- “Change-mode” interrupts (where an interrupt is generated on either a rising or a falling edge input signal) should be used on IO2-3 only if the rate of interrupts is likely to exceed 1000 per second. For other interrupt modes (falling edge only, rising edge only, low level, high level), it is possible to use SoC GPIO pins instead which are also connected to IO2-3.
- Interrupts should be used on IO14-19 only if the rate of interrupts is likely to exceed 1000 per second. However, due to the presence of 1uF capacitors on these pins and their effect on signal rise/fall times, it is unlikely that these pins would be used for high-rate interrupt signaling.
- The shield reset input is intended for use with a manually-pressed reset button on an Arduino shield. In that scenario, the rate of button presses is unlikely to exceed 1000 per second. However, there is a chance that signal bounce from the mechanical switch could conceivably trigger this scenario.

Workaround: It is possible to restore interrupt functionality by reading the current input values from any GPIO pin(s) on the PCAL9555A that are configured to generate interrupts. This will effectively “clear” the outstanding interrupts and allow new interrupt notifications to be detected by software.

1.9.14 78738 [GAL-84]: I2C/GPIO level-triggered interrupts cause system hang

System hangs during testing level-triggered interrupt handling in the I2C/GPIO driver (intel_qrk_gip). After loading the driver, the GPIO pin level goes low (verified with multimeter) and stays low. The interrupt fires and the system hangs forever (no response on shell via serial or ssh).

Workaround: Reboot the board.

1.9.15 N/A: Arduino IDE menus are grayed out on Mac after upgrade

After doing a firmware upgrade using a Mac, the menus in the Arduino IDE may be grayed out.

Workaround: restart the IDE.

1.10 Resolved issues

Issue #	Section	Description
54858	1.10.1	Driver installation in Windows gives a warning
55516	1.10.2	Pins have pullups enabled at reset time
55564	1.10.3	IO Expander address is hardcoded



Issue #	Section	Description
55593	1.10.4	Reboot button fails to reboot the board
55603	1.10.5	Workaround for servo library
55813	1.10.6	SD Library cannot create new files
56375	1.10.7	Arduino app on OS X does not work when Mac auto renames app
GAL-48	1.10.8	Sketch downloading sometimes fails on Linux hosts
N/A	1.10.9	IDE version 0.7.5 cannot update firmware if Galileo has firmware version 0.8.0 or higher

1.10.1 54858: Driver installation in Windows gives a warning

The previous Release Notes version stated this item was Open; this was incorrect. This issue has been resolved.

During driver installation, you may see this error: **Windows cannot verify the publisher of this software**. This error will be fixed in a future release and can be ignored. Select **Install the driver anyway**.

1.10.2 55516: Pins have pullups enabled at reset time

In release 0.7.5, when the Intel® Galileo board is powered on, and before the sketch is executed, pins have pullups enabled by default.

Resolution: In release 1.0.0, the default configuration of Arduino pins on Galileo is now input, with pullups disabled.

1.10.3 55564: I/O Expander address is hardcoded

The I2C address of the I/O expander is hardcoded to 0x20, regardless of the configuration set on the J2 header. The address of the IO Expander cannot be changed to avoid address clash with another device on the bus.

Resolution: The Cypress driver now automatically detects its own I2C address based on J2 header configuration.

1.10.4 55593: Reboot button fails to reboot the board

In release 0.7.5, when the reboot button is pressed, the board resets; however, it doesn't wake up unless power is removed and reattached.

Resolution: In release 1.0.0, this has been fixed.

1.10.5 55603: Workaround for servo library

In release 0.7.5, Servo library was not supported.

Resolution: This has been fixed in release 1.0.0.



The workaround for the previous release is described below.

An example of a sketch is also below. The sketch configures the PWM at 125 Hz frequency which gives you a pulse width granularity of 31 μ sec. You are free to move the frequency up or down as you wish, keeping in mind that granularity will increase and decrease respectively.

Datasheet for the IO expander is at: <http://www.cypress.com/?docID=31413> page 13 is most relevant.

Mapping between Arduino pins and Cypress PWM ports:

- Pin3: GPORT0_BIT4_PWM7
- Pin5: GPORT0_BIT1_PWM5
- Pin6: GPORT1_BIT0_PWM6
- Pin9: GPORT0_BIT3_PWM1
- Pin10: GPORT0_BIT0_PWM7
- Pin11: GPORT1_BIT1_PWM4

Example Sketch

```
#include "Wire.h"

int PIN = 9;
void setup() {
    // put your setup code here, to run once:

    Wire.begin();
    pinMode(PIN, OUTPUT);
    analogWrite(PIN,1);

    // Set divider to get 125Hz freq.
    Wire.beginTransaction(0x20);
    Wire.write(0x2C);
    Wire.write(0x03);
    Wire.endTransmission();

    // Select programmable PWM CLK source
    Wire.beginTransaction(0x20);
    Wire.write(0x29);
    Wire.write(0x04);
    Wire.endTransmission();

    // Set period register
    Wire.beginTransaction(0x20);
    Wire.write(0x2a);
    Wire.write(0xff);
    Wire.endTransmission();

    // Set minimum duty cycle (31us @ 125Hz)
    Wire.beginTransaction(0x20);
    Wire.write(0x2b);
    Wire.write(0x01);
    Wire.endTransmission();
}

void loop() {
}
```



1.10.6 55813: SD Library cannot create new files

The SD library should check if a valid file exists when passed a path to a file to be opened. As this check is not done, the library is not able to handle files that do not exist.

File.write() will crash when called on a File which was created with a new file. This might be true of other File methods.

Resolution: This has been resolved in release 1.0.0.

1.10.7 56375: Arduino application on OS X does not work when application is automatically renamed

If you already have the Arduino IDE installed on your Mac when you install the Galileo software, the Galileo IDE application will be renamed with a number and a space. The space causes the Galileo application to be unable to compile.

Resolution: This has been resolved in release 1.0.0.

1.10.8 GAL-48: Sketch downloading sometimes fails on Linux 64-bit hosts

Sketch downloading sometimes fails on Linux hosts. Behavior has been confirmed on Ubuntu 12.04.04 and Linux Mint 16 64-bit (based on Ubuntu 13.10).

Resolution: Identified as Will Not Fix in release 1.0.2. The workaround is to remove modem manager and reboot the Linux host using the following commands:

```
sudo apt-get remove modemmanager  
sudo reboot
```

1.10.9 N/A: IDE version 0.7.5 cannot update firmware if Galileo has firmware version 0.8.0 or higher

The IDE in release 0.7.5 contained an issue that affected firmware updates. If the Galileo board was updated with firmware version 0.8.0 or higher, the IDE was unable to perform firmware updates.

Resolution: This has been resolved in release 1.0.0.



Revision History

Release Date	Document Revision	Description
23 June 2014	006	Updates for Package Version: 1.0.2 for Arduino IDE v1.5.3 Added support for the 2 nd generation Intel® Galileo board. Updated Known Issues in the Release and Resolved Issues .
22 May 2014	005	Updates for Package Version: 1.0.0 for Arduino IDE v1.5.3 Added Fixed Heartbleed vulnerability in Release 1.0.0 . Added 4.20q Security Disclaimer . Updated Known Issues in the Release and Resolved Issues .
16 October 2013	004	Updated download URL and other minor text corrections.
10 October 2013	003	Updated file names and sizes. Added Known Issue 56375. Removed list of supported shields (now a separate document).
04 October 2013	002	Updated Resolved Issues.
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, Centrino, Quark, and the Intel logo 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.