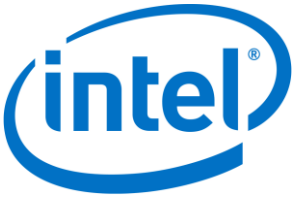

Intel® Galileo and Intel® Edison

Release Notes
Package Version: Arduino* IDE v1.6.0+Intel

March 2015

Revision 008

Document Number: 329686-008

Contents

Revision History

Revision	Description	Date
001	First external release: Package Version: 0.7.5 for Arduino IDE v1.5.3.	October 2, 2013
002	Updated Resolved Issues.	October 4, 2013
003	Updated file names and sizes. Added Known Issue 56375. Removed list of supported shields (now a separate document).	October 10, 2013
004	Updated download URL and other minor text corrections.	October 16, 2013
005	Updates for package version: 1.0.0 for Arduino IDE v1.5.3. Added Fixed Heartbleed vulnerability in Release 1.0.0. Added 4.20g security disclaimer. Updated Known issues and Resolved issues.	May 22, 2014
006	Updates for package version: 1.0.2 for Arduino IDE v1.5.3. Added support for the second generation Intel® Galileo board. Updated Known issues and Resolved issues.	June 23, 2014
007	Updates for package version: 1.0.3 and 1.0.4 for Arduino IDE v1.5.3. Added support for the Intel® Edison board. Updated Known issues and Resolved issues.	October 16, 2014
008	Updates for the package version 1.6.0+Intel. Added support for Arduino IDE v1.6.0. Updated Known issues and Resolved issues.	March 17, 2014

1 Introduction

This document describes the features, bug fixes, and known issues in the v1.6.0+Intel release of the Arduino* IDE software. 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® Edison for Arduino* Kit
- Arduino* Integrated Development Environment (IDE) v1.6.0+Intel

Note: This release uses a special version of the Arduino* IDE. You must download it from the Intel webpage listed below

In this document:

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

1.1 Downloading the software release

Download the latest Arduino* IDE files:

- *https://downloadcenter.intel.com/download/24355*

This release contains multiple compressed files, including:

- Operating system-specific IDE packages:
  - 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)

If you are running the IDE software, refer to the appropriate getting started guide for your platform and OS:

- *https://software.intel.com/en-us/iot/getting-started*

1.2 Supported operating systems

This 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)
Mac OS X*	Built on: OS X version 10.9.2 Tested: OS X 10.10

1.3 Supported Arduino* libraries

This software release supports the following Arduino libraries:

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

1.4 Supported shields

For a list of tested shields, see the *Intel® Galileo Board, Intel® Galileo Gen 2 Board, and Intel® Edison Board Shield Testing Report*: *http://www.intel.com/support/motherboards/desktop/sb/CS-035257.htm*.

1.5 Supported sketches

The following Arduino*-based sketches have been tested on at least one of the three supported OSes:

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

Sketches from Arduino* starter kit			
p02_SpaceshipInterface	p04_ColorMixingLamp,	p09_MotorizedPinwheel	p14_TweakTheArduinoLogo
p03_LoveOMeter	p08_DigitalHourglass	p10_Zoetrope	p15_HackingButtons

Arduino* library sketches			
Autoscroll	Display	HelloWorld	setCursor
Blink	DumpFile	listfiles	SimpleWebServerWiFi
ConnectNoEncryption	eeprom_clear	ScanNetworks	TextDirection
ConnectWithWPA	eeprom_read	Scroll	WiFiWebClient
Cursor	eeprom_write	SerialDisplay	WiFiWebServer

1.6 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.

1.7 Features in this release

SoftwareServo library. Software emulation of the Servo library. PWM servo signals are bit-banged instead of using the hardware PWM on the Intel® Edison device. The SoftwareServo library can be used in any digital pin (0-13). Use of SoftwareServo library is similar to that of the standard Servo library *http://arduino.cc/en/reference/servo*.

softAnalogWrite(). Software emulation for PWM functionality for Intel® Edison. PWM signals are bit-banged instead of being generated by dedicated hardware. Does not have the 8-bit limitation of the built PWM hardware of the Tangier SoC used for Intel® Edison. You may use *softAnalogWrite()* in any of the digital pins. Use is similar to that of the standard PWM, but instead of *analogWrite()*, use *softAnalogWrite()*.

Arduino v1.6.0. We have rebased the Arduino* IDE from v1.5.3 to v1.6.0. Consequently, this release offers the following features:

- Other (non-Intel boards) are now supported by the IDE by default.
- Platform files are unified to follow the Arduino* spec.
- Toolchain pathnames have been shortened.
- Windows* now has an installer, which installs the software at a default location and prevents long pathname issues.
- Linux* releases are now released with .txz file extension to match Arduino.cc releases.
- Libraries for Intel® Galileo and Intel® Edison now follow the v1.5.x format.

1.8 Known issues

Issue #	Section	Description
MAKER-200	1.8.1	Uploading Galileo sketches to Edison spawns zombie processes.
MAKER-210	1.8.1	Renbotics ServoShield2 does not work properly.
MAKER-211	1.8.2	pinMode() and analogRead() triggers interrupts on an attached pin.
MAKER-354	1.8.4	Serial2 for Edison does not properly terminate console input from the FTDI header

1.8.1 MAKER-210: Renbotics ServoShield2 does not work properly.

Edison only: Some shields that use I^2C may not work because they have pullup resistors that are too strong. Replacing these with weaker pullup resistors (10 kohm) or removing the resistors altogether should allow the shields to work.

1.8.2 MAKER-211: pinMode() and analogRead() triggers interrupts on an attached pin.

Edison only: When an interrupt is attached using *attachInterrupt()*, calling *pinMode()* and *analogRead()* for the first time triggers an interrupt. This is due to the muxing operations done when those functions are called. To avoid this issue, *pinMode()* should be called before *attachInterrupt()*, and *analogRead()* should also be called at least once for each pin used before calling *attachInterrupt()*.

1.8.3 MAKER-354: Serial2 for Intel® Edison does not properly terminate console input from the FTDI header.

When Serial2 is called for Intel® Edison it does not properly terminate console input from the FTDI header. As a result, when the user enters bytes to the FTDI header, half of the bytes go into the console and the other half are read by the serial buffer of the IDE.

§

2 Resolved issues

Table 1 Resolved in 1.6.0 Release

Issue #	Section	Description
MAKER 200	2.1	Uploading Intel® Galileo sketches to Intel® Edison platform spawns zombie processes.
MAKER 234	2.2	Servo API limits the amount of servo objects to 5 instead of 6.
MAKER 235	2.3	SPI.begin() breaks some GPIO functionality.
MAKER 286	2.4	WString.h missing #include avr/pgmspace.h.
MAKER 318	2.5	Reduced CPU usage of Serial.readBytes() and Serial.readBytesUntil().
MAKER 323	2.6	Merge Stream.cpp fix from Arduino*.
MAKER 324	2.7	Webserver.ino sketch only functional during first upload of sketch.
MAKER 330	2.8	Setting pwm to 0% duty cycle now functional on Intel® Edison.
MAKER 336	2.9	Support for all possible SPI speeds.
MAKER 337	2.10	Intel® Galileo timer interrupt missing first tick.

2.1 MAKER-200: Uploading Intel® Galileo sketches to Intel® Edison platform spawns zombie processes.

Uploading a sketch compiled for the Intel® Galileo platform into an Intel® Edison device (or vice versa), the upload process would succeed because the process was identical on both boards. However, when *clloader* attempted to start the binary file, it would fail and spawn a zombie process.

We fixed this by making the upload command unique for each board. This fix requires the boards to have the latest BSP for Intel® Galileo (1.0.x) and Intel® Edison (Rel2).

2.2 MAKER-234: Servo API limits the amount of servo objects to 5 instead of 6.

Fixed a bug where the servo class was limiting the amount of Servo objects to 5 instead of 6.

2.3 MAKER-235: SPI.begin() breaks some GPIO functionality.

Fixed a bug where calling *SPI.begin()* multiple times would break some GPIO functionality.

2.4 MAKER-286: WString.h missing #include avr/pgmspace.h.

Added missing *#include avr/pgmspace.h* to *WString.h*.

2.5 MAKER-318: Reduced CPU usage of Serial.readBytes() and Serial.readBytesUntil().

Reduced CPU usage when using *Serial.readBytes()* and *Serial.readBytesUntil()* by using *yield()*.

2.6 MAKER-323: Merge Stream.cpp fix from Arduino*.

Merged fixes to *Stream.cpp* from Arduino*.

2.7 MAKER-324: Webserver.ino sketch only functional during first upload of sketch.

Fixed a bug in the WebServer sketch where it only works on first upload.

2.8 MAKER-330: Setting pwm to 0% duty cycle now functional on Intel® Edison.

Setting the duty cycle to 0% on Intel® Edison is now functional. A workaround for the bug where the pin is paced on GPIO mode is used.

2.9 MAKER-336: Support for all possible SPI speeds.

We added support for higher SPI speeds (up to 20 MHz) for Edison. We also added a new function *SPI.setClockSpeed()* to the SPI class to allow the user to set the SPI clock speed to any of the supported speeds.

2.10 MAKER-337: Intel® Galileo timer interrupt missing first tick.

Fixed a bug with the Timer interrupt missing the first tick for ISR1.

§