Look Inside.™

Workshop – Intel® Galileo Board

Introduction and Basics of Intel® Galileo Board

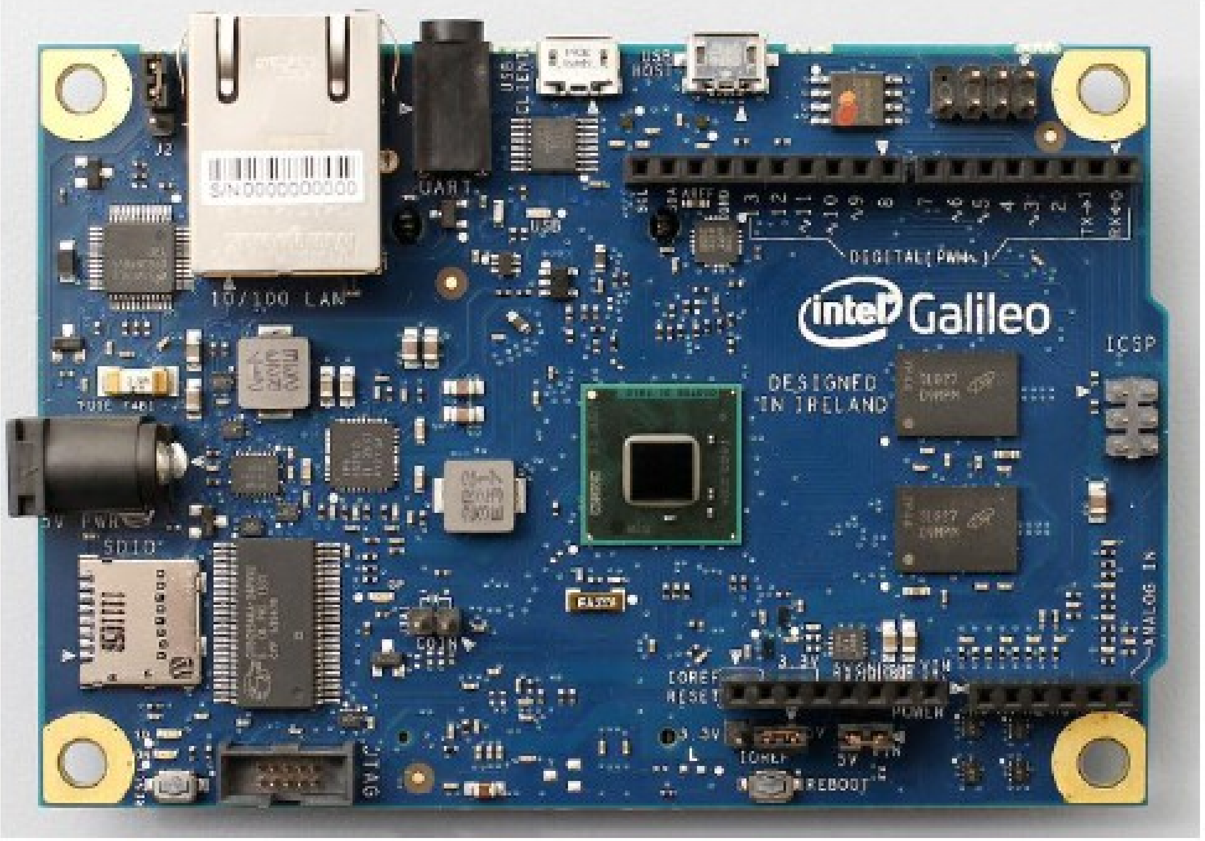
Walter Netto

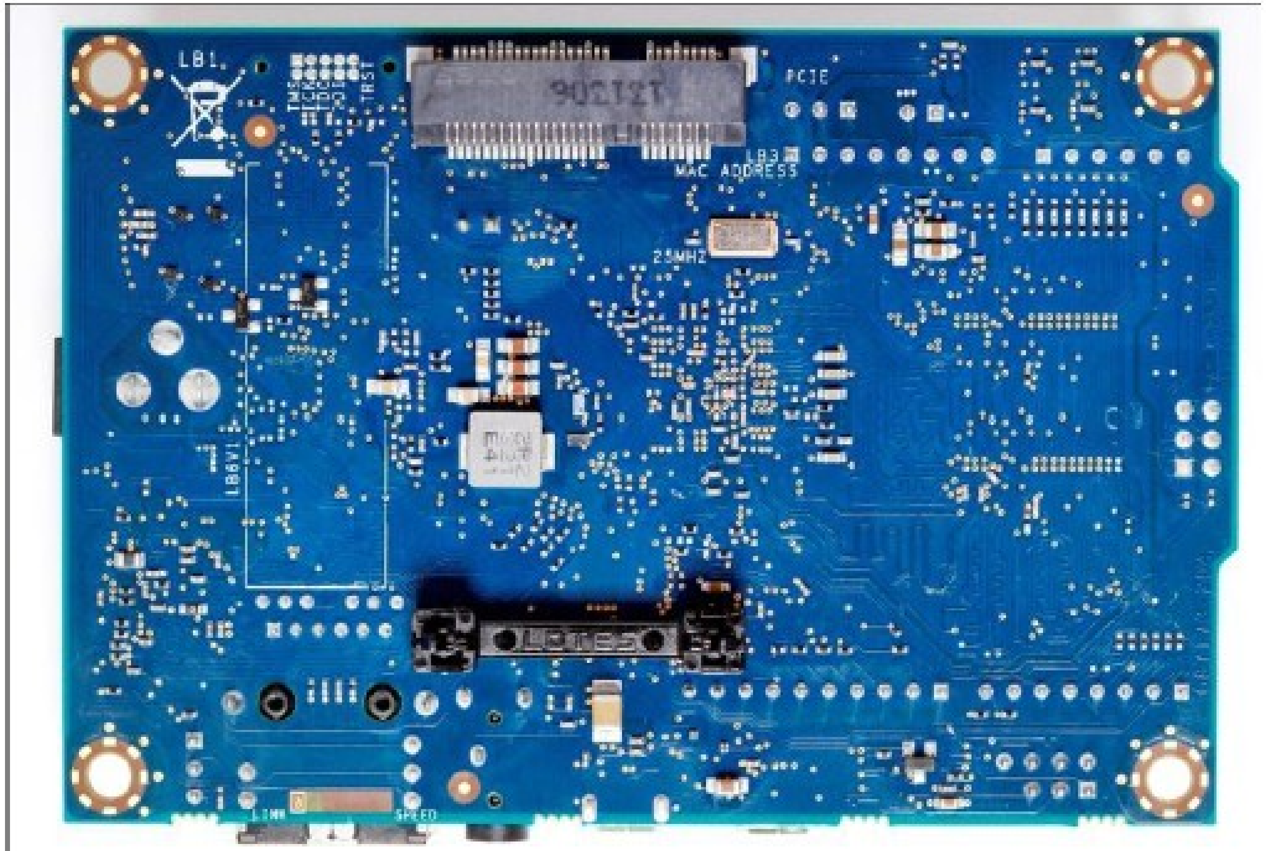
November 03th, 2014

Agenda

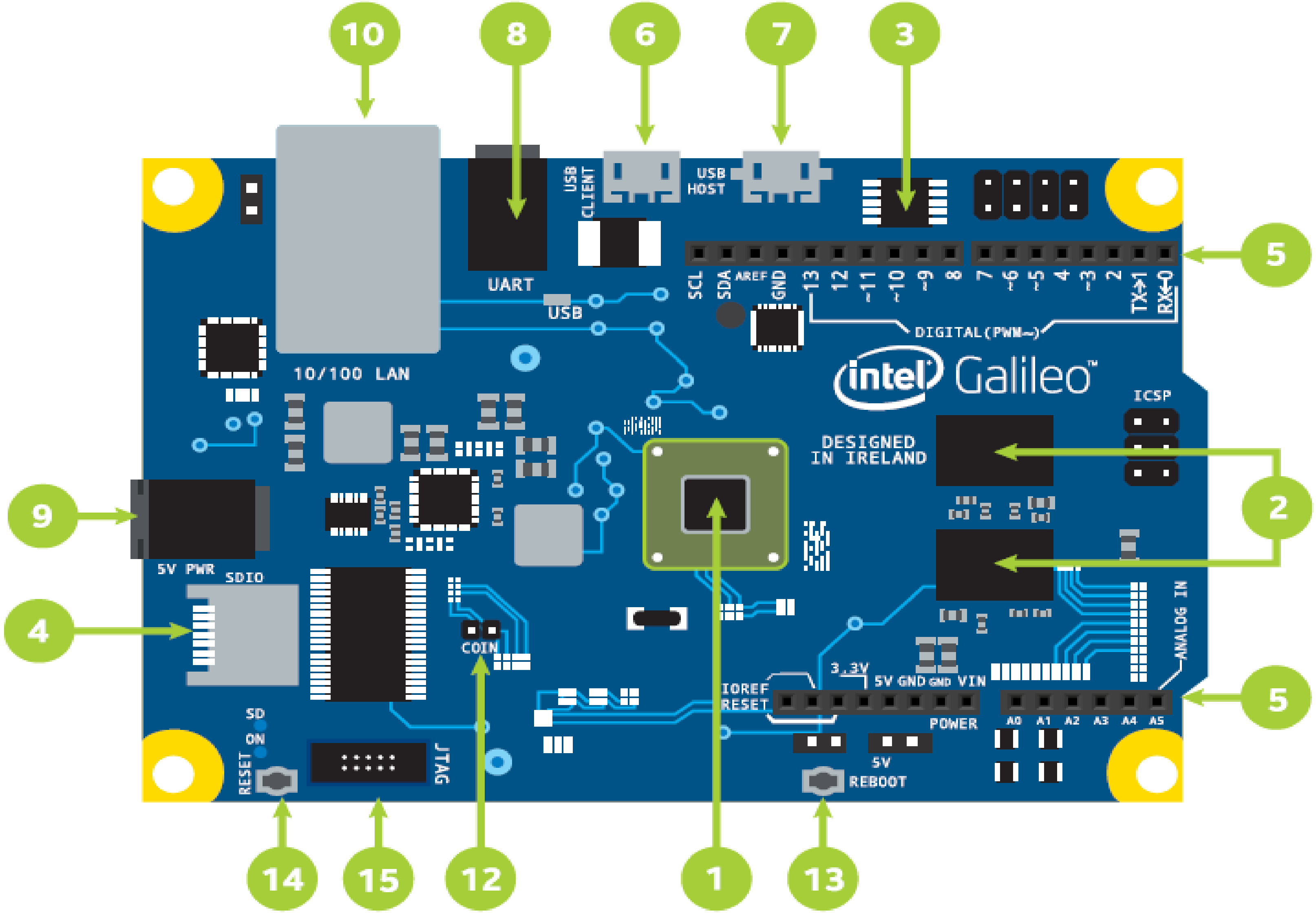
- Intel® Galileo Board
  - Overview
  - Physical Characteristics
  - Communication
  - Processor Features
  - Storage Options
  - Galileo Target Software
- Getting Started
- Sketches and Code Examples

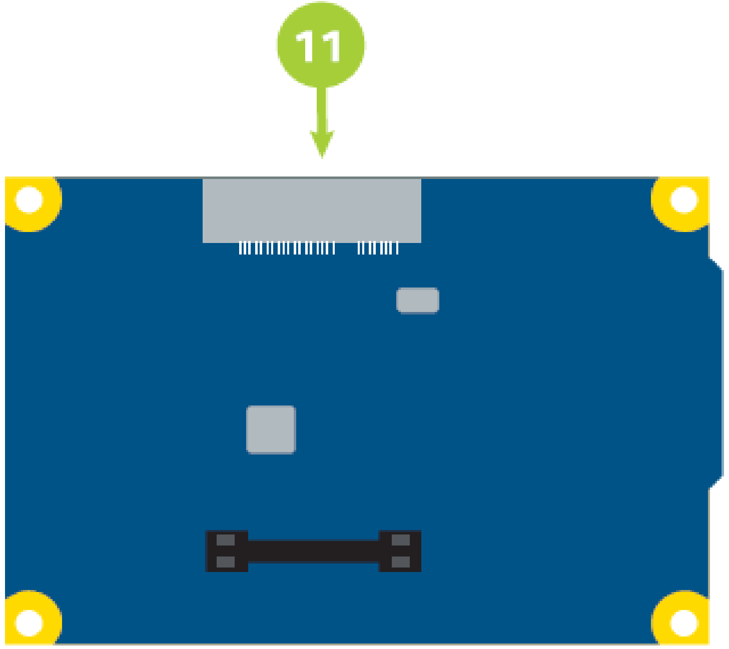
Intel® Galileo Board Overview





Intel® Galileo Board Overview





Back View

Intel® Galileo Board Overview

1. Processor
2. Random Acess Memory (RAM)
3. Flash Memory
4. Micro SD Card Slot
5. Arduino Expansion Clock Pins
6. USB Client Port
7. USB Host Port
8. Serial Port
9. Power Input
10. Ethernet Port
11. Mini PCIe Socket
12. Coin (RTC Support)
13. Reboot Button
14. Reset Button
15. JTAG Header

Intel® Galileo Board Overview

- The Intel® Galileo board is based on the Intel® Quark SoC X1000, a 32-bit Intel Pentium®-class system on a chip (SoC). It is the first board based on Intel® architecture designed to be hardware and software pin-compatible with shields designed for the Arduino Uno R3.

- The Galileo board is also software-compatible with the Arduino Software Development Environment.

- The Intel processor and surrounding native I/O capabilities of the SoC provides for a fully featured offering for both the maker community and students alike.

Intel® Galileo Board
Physical Characteristics

- 10 cm long and 7 cm wide with the USB connectors, UART jack, Ethernet connector, and power jack extending beyond the former dimension.
- Four screw holes allow the board to be attached to a surface or case.
- Standard 10-pin JTAG header for debugging
- Reset button to reset the sketch and any attached shields
- Galileo is powered via an AC-to-DC adapter, connected by plugging a 2.1mm center-positive plug into the board's power jack. The recommended output rating of the power adapter is 5V at up to 3A.

Intel® Galileo Board Communication

- 10/100 Mb Ethernet RJ45 port
- USB 2.0 Client port
- USB 2.0 Host port
- RS-232 UART port and 3.5mm jack
- Mini PCI Express (mPCIe) slot with USB2.0 Host support

Intel® Galileo Board
Processor Features

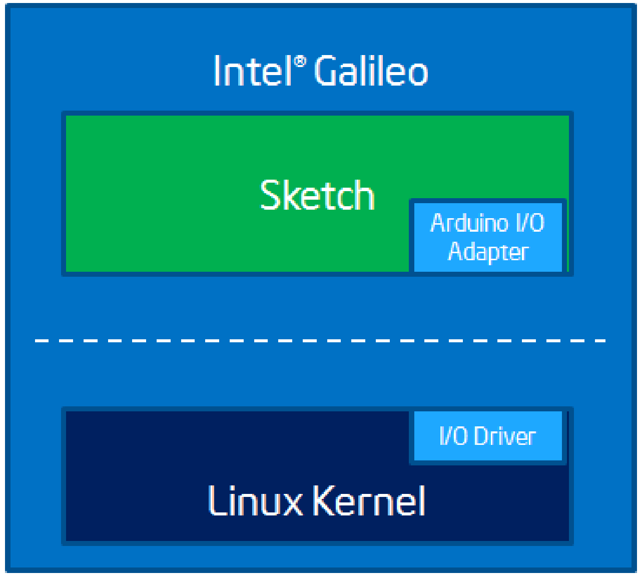
- 400 MHz 32-bit Intel® Pentium® instruction set architecture (ISA)-compatible processor
    - 16 KBytes L1 cache
    - 512 KBytes of on-die embedded SRAM
    - Simple to program: Single thread, single core, constant speed
    - ACPI compatible CPU sleep states supported
    - Integrated Real Time Clock (RTC), with optional 3V “coin cell” battery for operation between turn on cycles

Intel® Galileo Board Storage Options

- 8 MByte Legacy SPI Flash to store firmware (bootloader) and the latest sketch
  - Between 256 KByte and 512 KByte dedicated for sketch storage
- 512 KByte embedded SRAM
- 256 MByte DRAM
- Optional micro SD card offers up to 32 GBytes of storage
- USB storage works with any USB 2.0 compatible drive
- 11 KByte EEPROM programmed via the EEPROM library

Intel® Galileo Board
Galileo Target Software

- Use the Arduino IDE software to create programs for Galileo called “sketches.”
- To run a sketch on the board, simply connect a power supply, connect Galileo's USB Client port to your computer, and upload the sketch using the IDE interface.
- The sketch runs on the Galileo board and communicates with the Linux kernel in the board firmware using the Arduino I/O adapter



Getting Started

Steps

1. Download the Arduino environment and board firmware
2. Get a board and cables
3. Connect the board
4. Install the drivers and other software
5. **TRICK**
6. Launch the Arduino IDE application
7. Update your board firmware
8. Example
9. Bibliography

1. 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/).

1. Download the Arduino environment and board firmware

- 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.
    - LINUX_IMAGE_FOR_SD_Intel_Galileo_v1.0.0.7z (38.7 MB)

1. Download the Arduino environment and board firmware

- This release contains multiple zip files, including:

  - (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)

1. Download the Arduino environment and board firmware

- Windows:
  - Unzip the IDE file to the C:\ directory.
  - 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.

1. Download the Arduino environment and board firmware

- Linux:
  - 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*”

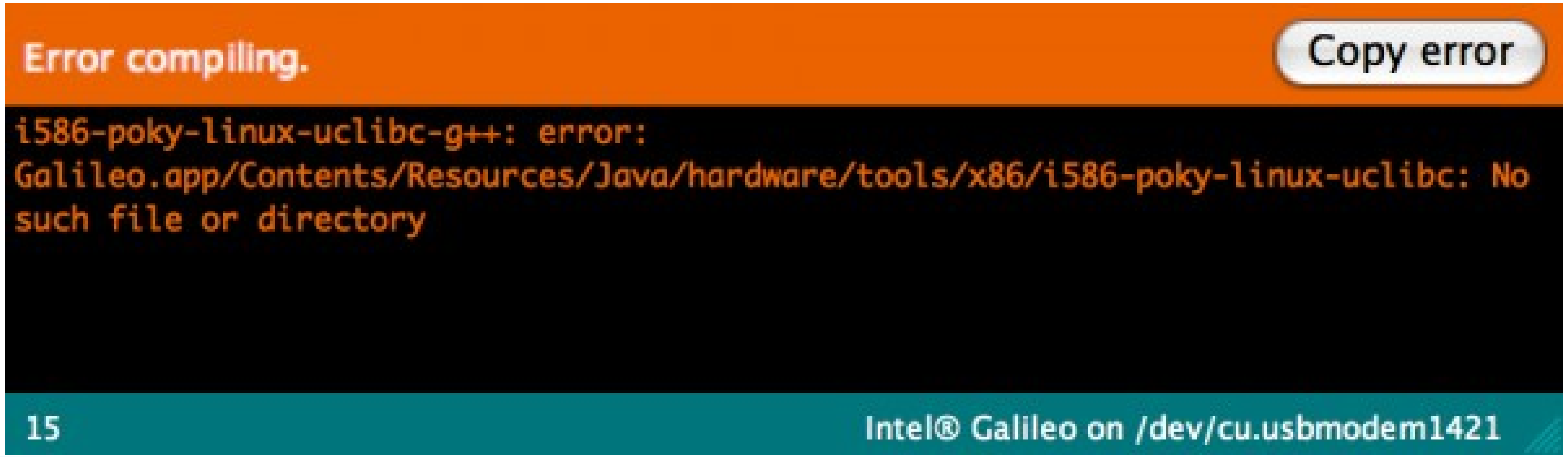
  - To avoid interference with the IDE, you **must** disable modem manager. The exact command will depend on your Linux distribution. For example, the command “sudo apt-get remove modemmanager” may work.
  - In the same directory where the package was downloaded, launch the Arduino IDE by executing “*./arduino*”

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.

1. Download the Arduino environment and board firmware

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

Troubleshooting: If an error like this shows up, There may be a problem with the name of your application, make sure there are no spaces (“Arduino Galileo” becomes “ArduinoGalileo”).



Error compiling.

```
i586-poky-linux-uclibc-g++: error:
Galileo.app/Contents/Resources/Java/hardware/tools/x86/i586-poky-linux-uclibc: No
such file or directory
```

2. Get a board and cables

The Kit contains:

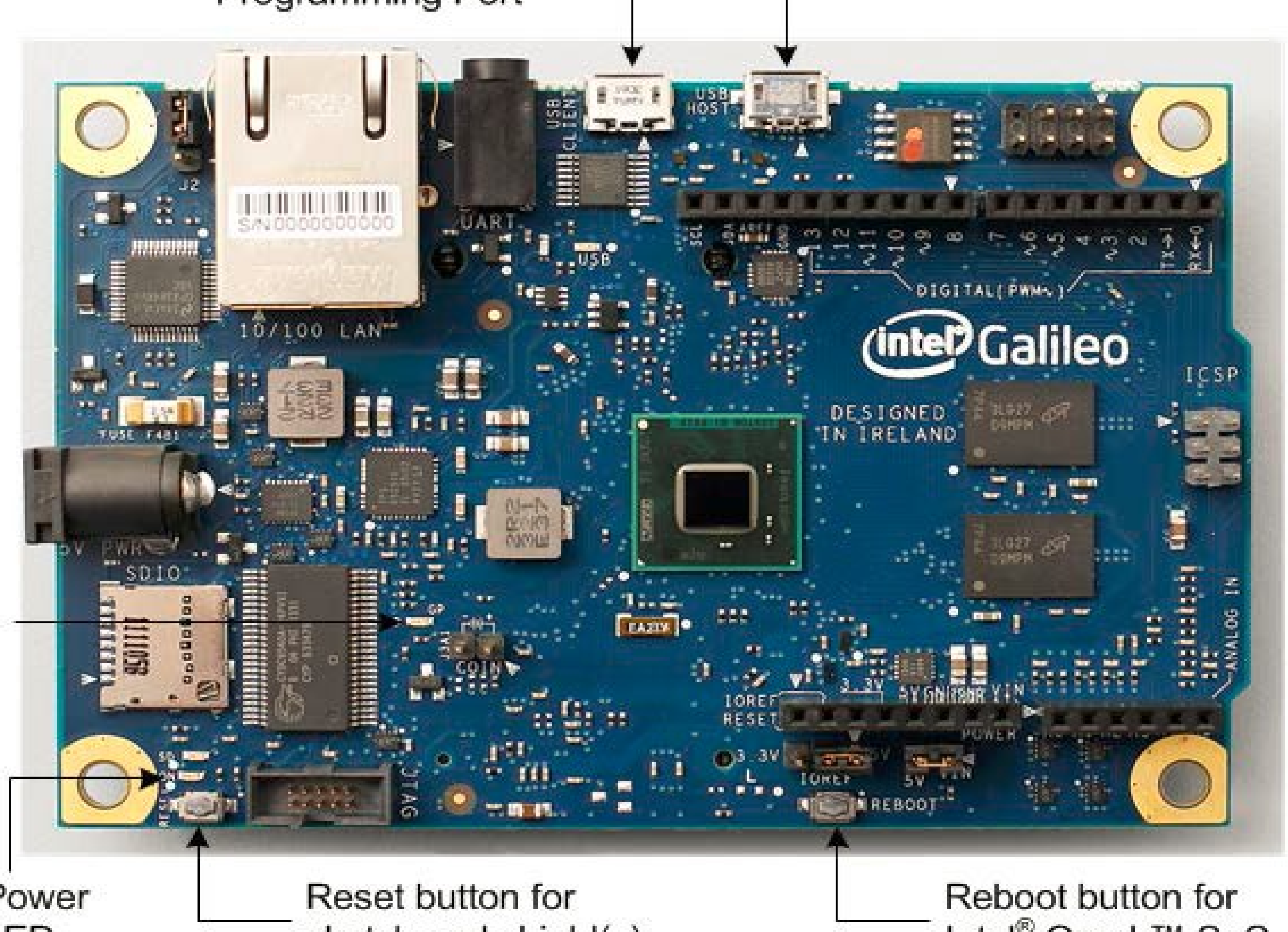
- 1x Intel® Galileo Customer Reference Board (CRB) (Fab D with blue PCB)
- 1x 5v power supply with cable

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.

2. Get a board and cables

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

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

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.

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

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

4. Install the drivers and other software

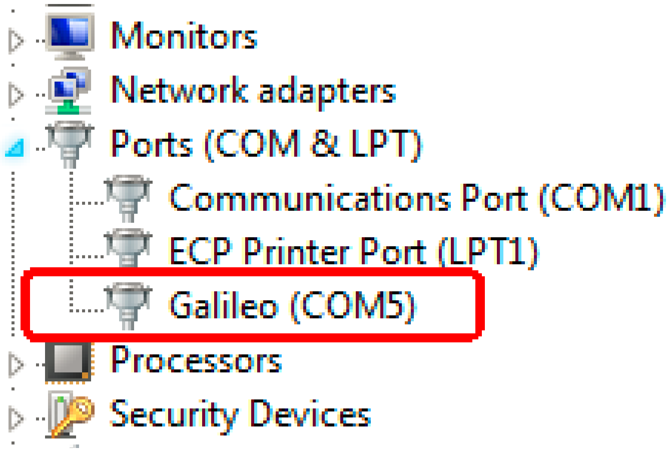
Windows:

- If not done already, connect the 5V power cable to the Galileo board and to a power outlet.
- 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.
- 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**.
- 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 in slide 24**.
- Right-click on the **Gadget Serial V2.4** port and choose the **Update Driver Software** option.

4. Install the drivers and other software

Windows:

- Choose the **Browse my computer for Driver software** option.
- Navigate to the *hardware/arduino/x86/tools* directory. This allows the proper driver file *linux-cdc-acm.inf* to be installed
- 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.



4. Install the drivers and other software

Troubleshooting: If your upload hangs, eventually producing the error:

- line 34: /dev/ttyS78: Read-only file system

Your COM port number may just be too high. To reassign the COM port:

1. On Device Manager right click your Galileo device and select Properties.
2. Go to Port Settings tab and click Advanced.
3. Click the COM Port Number drop-down, and assign the Galileo to a COM port that’s less than 10.
4. Reboot your computer and try uploading again.

4. Install the drivers and other software

Windows:

- **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 from where you stopped

4. Install the drivers and other software

Linux:

- If not done already, connect the 5V power cable to the board and to a power outlet. Wait for the board to boot
- Connect the USB cable to the USB Client Port (closest to the Ethernet) and to a PC
- Check that ACM port is available by typing in the terminal *ls /dev/ttyACM**
- 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. Install the drivers and other software

Linux:

- 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
- Select your board via **Tools > Board > Intel® Galileo**
- 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*

4. Install the drivers and other software

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.

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

When you try to open the “arduino.exe” it will not work. So you need to do this TRICK:

1. Go to “c:\arduino-windows-1.0.3\java\bin -> javaw.exe” and CREATE A SHORCUT on to your desktop.
2. Right-click in the shortcut and select “properties
3. On “target” change de path to: “***C:\arduino-windows-1.0.3\java\bin\javaw.exe -Duser.language=en -Duser.region=US -Xms128m -Xmx128m –classpath*** "***lib;lib\pde.jar;lib\core.jar;lib\jna.jar;lib\ecj.jar;lib\RXTXcomm.jar;lib\commons-exec-1.1.jar***" ***processing.app.Base***”
4. **Use this shortcut to run the Arduino Galileo IDE !!!**

6. Launch the Arduino IDE application

- **Windows:** In the desktop, double-click *the shortcut*

- **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

7. Update your board firmware

- Start the Arduino code editor for Galileo. (If you have an old Arduino editor installed, make sure you start the new custom one for Galileo: version 1.5.3)

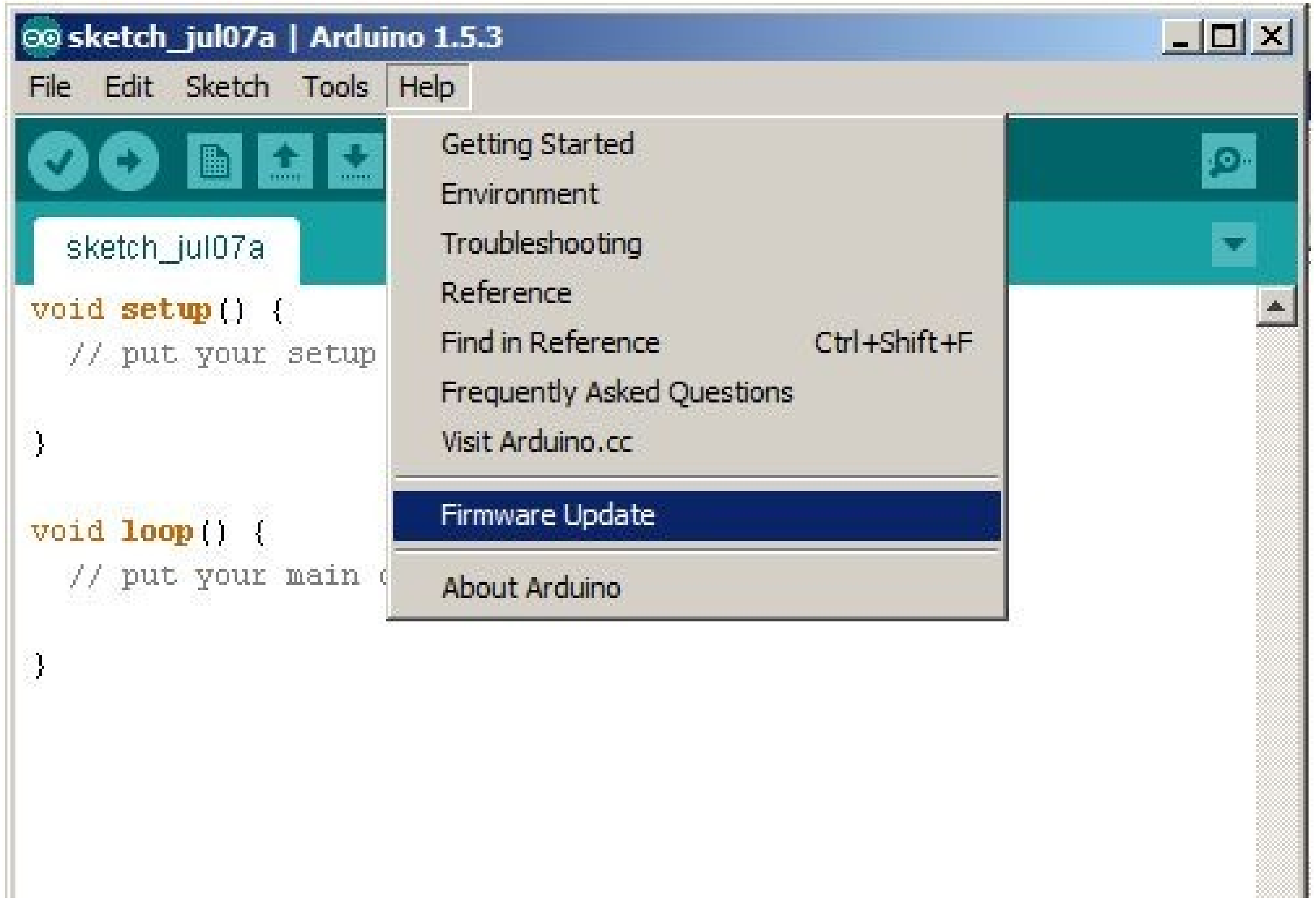
- Launch the IDE and select the board via **Tools > Board > Intel® Galileo** (If you have an old Arduino editor installed, make sure you start the new custom one for Galileo: version 1.5.3)

  Note: Make sure the micro-SD card _not_ installed (otherwise you might get an error message 'Target firmware version query failed')

- Select the correct serial port using **Tools > Serial Port**

7. Update your board firmware

- Launch the software upgrade using **Help > Firmware Upgrade** (be patient !)



```
void setup() {
  // put your setup

}

void loop() {
  // put your main c

}
```

7. Update your board firmware

- 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.

- The upgrade progress takes about **5 minutes** and is displayed in several popup messages. During the upgrade process, you will not have access to the IDE.
  - Make sure you don’t unplug either power or USB from Galileo.

- 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.

7. Update your board firmware

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.

8.1 Example

The Blink Example

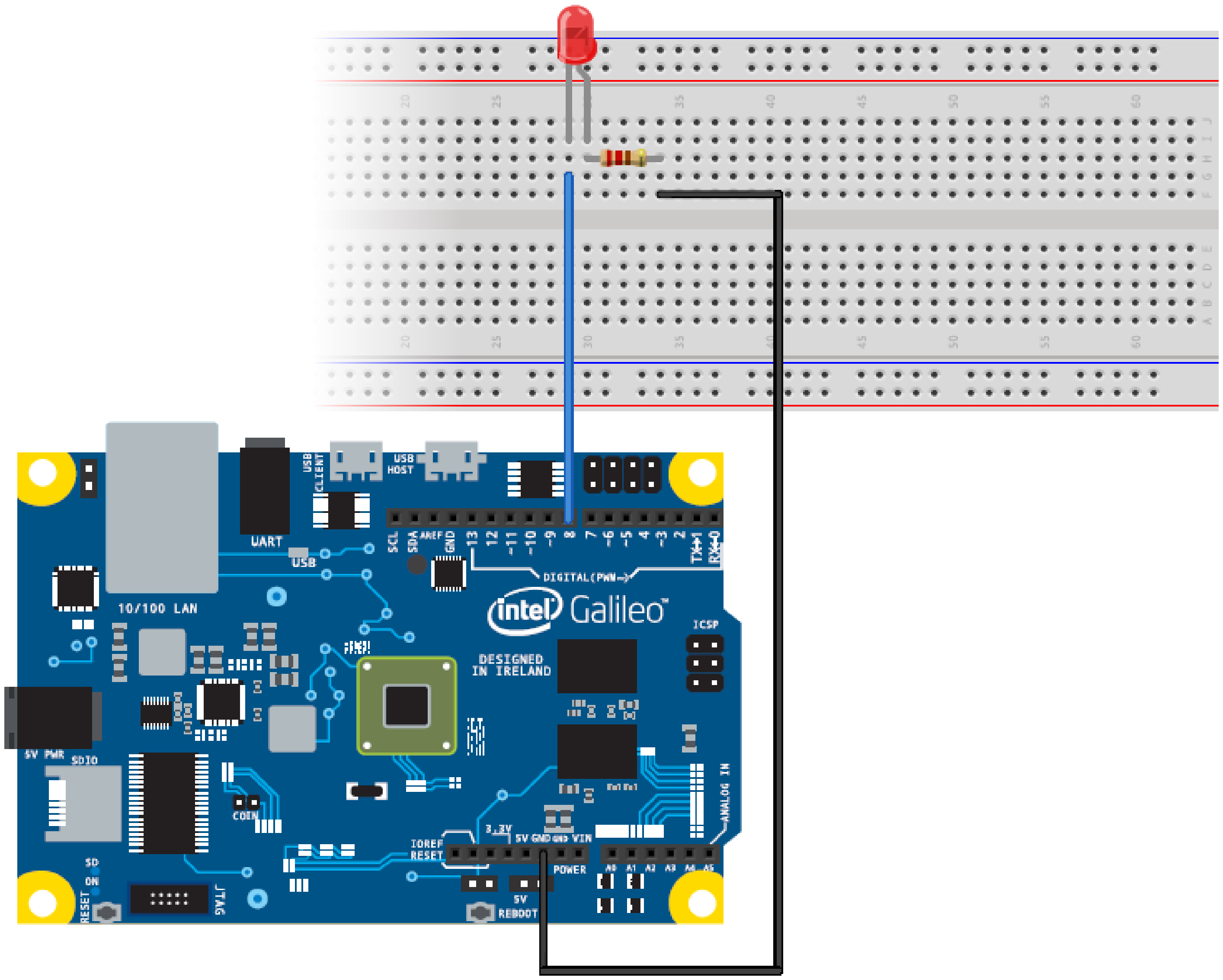
(this is a modification of the example available at: **File -> Examples -> 01.Basics -> Blink**)

- You will need:
  - Galileo Board
  - Breadboard
  - 1 x LED
  - 1 x 220-Ohm resistor
  - 2 x Male-to-male jumper wires

- Make the connections as follow in the schematic

8.1 Example

Schematic



8.1 Example

Copy the following code in to the Arduino Galileo IDE:

```
/* Blink
   Turns on an LED on for one second, then off for one second, repeatedly.
   This example code is in the public domain. */

// Pin 8 has an LED connected. Give it a name:
int led = 8;

// the setup routine runs once when you press reset:
void setup() {
  pinMode(led, OUTPUT);     // initialize the digital pin as an output.
}
```

8.1 Example

```
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(1000);                              // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                            // wait for a second
}
```

Click the Upload button in the toolbar and watch the LED blinking

9. Bibliography

https://software.intel.com/iot

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

https://communities.intel.com/community/makers/galileo

https://github.com/GalileoWorkshop/IESC

http://www.instructables.com/id/Galileo-getting-started/

https://learn.sparkfun.com/tutorials/galileo-getting-started-guide

https://communities.intel.com/thread/48869