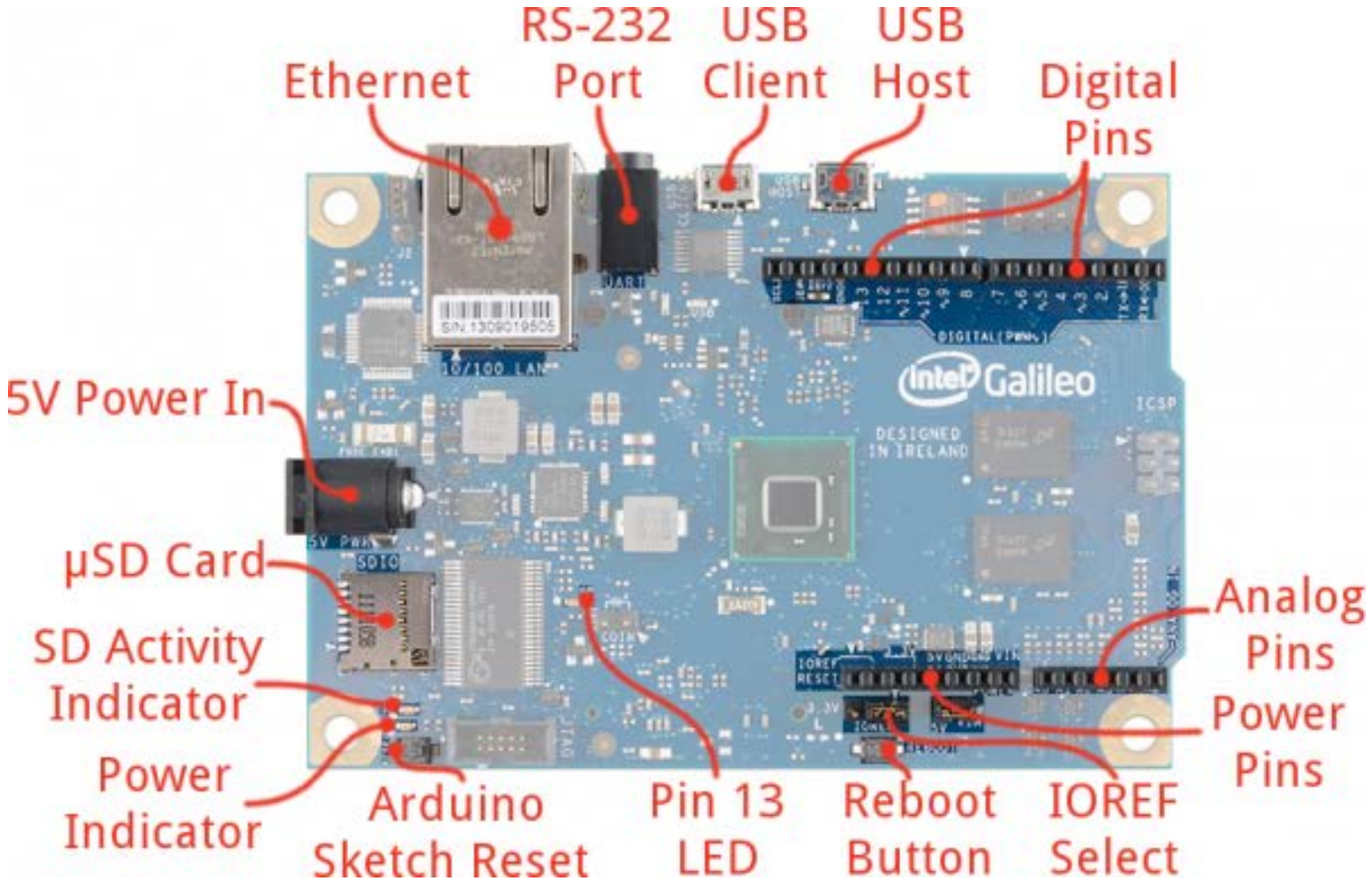
Set-Up the Galileo

Introduction and Getting Ready

What is the Galileo?

- A microprocesser – a fully functioning computer
- Compatible with Arduino software and hardware
- Runs Linux onboard for more advanced functions

Board Overview



What We'll be Doing with It

- Using sensors to control onboard functions
- Controlling sensors and LEDs based on Internet data
- Sending sensor data to the cloud
- Interfacing between Arduino and Linux (Python)
- Using Python packages

Get the Workshop Materials

- Clone https://github.com/aloverso/IntelGalileoWorkshop onto your computer
- Here you'll find:
  - Setup – materials for setting up the Galileo
  - Presentations – all powerpoints used for this class
  - Python and Linux Files – files we will use later, to be copied to the Linux shell
  - Arduino Files
- Open the Arduino Files folder and copy all folders into your personal Arduino folder (usually located in **Documents**)

Get the Arduino Software

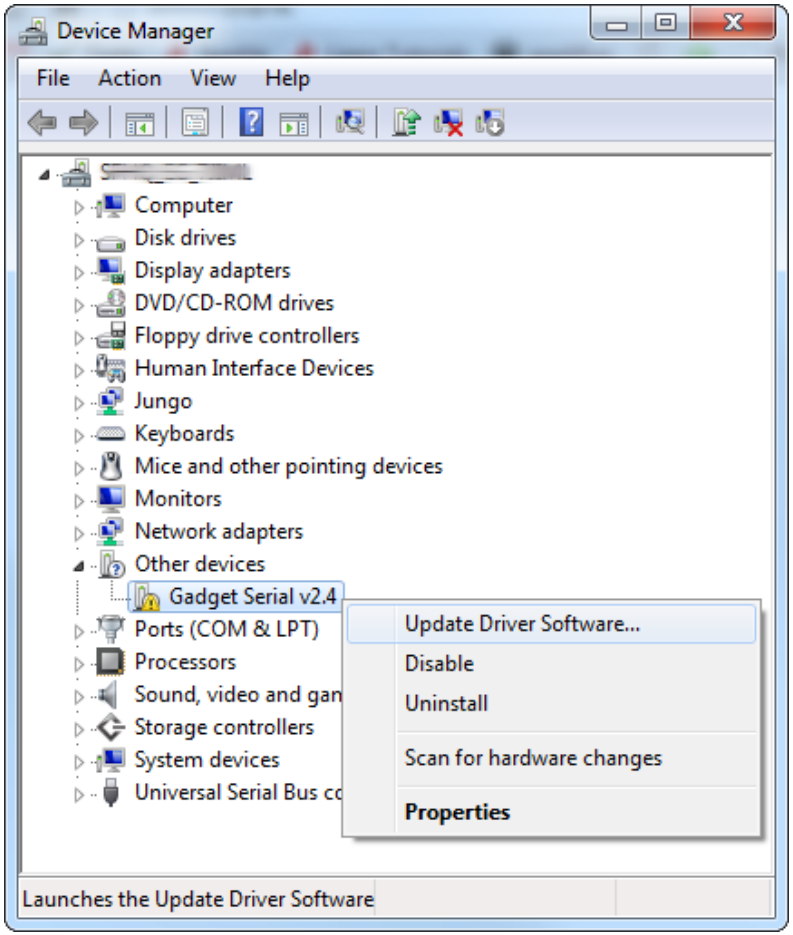
- Download the software for your operating system here: http://www.intel.com/support/galileo/sb/CS-035101.htm
- Unzip it and move the folder (*arduino-1.6.0*) to your top-level directory C:
- Rename it Galileo to make it easy to find, or any name you want as long as it **has no spaces**
- Double-click the *arduino.exe* application to run it and make sure it starts up

Install the Device Drivers

- Connect the 5V power supply to the board, and into a wall plug
- Connect a micro-USB cable from the board to one of your computer's USB ports
- Upon connecting the board, Windows will automatically attempt to install the driver and, unsurprisingly, it will fail. We'll have to manually install the driver.
- Open up the **Device Manager**. (Either *Start* > *Run* > *devmgmt.msc*, or go to the Control Panel, select *System and Security* > *System* and click Device Manager.)

Install the Device Drivers

- Locate the **Gadget Serial v2.4** device, under the Other devices tree. Right-click that and select **Update Driver Software**...
- On the first window that pops up, click **Browse my computer for driver software**. And on the next page select **Browse...** and navigate to the *hardware\arduino\x86\tools* folder within your Arduino Galileo software installation.
- Then click **Next**.



Install the Device Drivers

- **Click Install** on the next *Windows Security* window that pops up. And, after a number of loading-bar-scrolls, the installation should complete and you should be greeted with a *Windows has successfully updated your driver software* window.
- Look back at the **Device Manager**, under the *Ports* tree now. There should be an entry for **Galileo (COM #)**. Remember which COM # your Galileo is assigned, it'll be important for Arduino sketch uploading and the next step.

Updating Firmware

- Reboot the Galileo by **FIRST unplugging the micro-USB cable** and NEXT **unplugging the 5V power**
- Reboot by **FIRST plugging in 5V power** and **NEXT plugging in the micro-USB cable**
- Run the *arduino.exe* application in the software you downloaded in the first step
- Go to **Tools** and select **Serial Port** (this might take a few moments to be un-greyed out while the board starts up). Select the **COM port** that you saw earlier in *Device Manager*

Updating Firmware

- Go to **Tools > Board** and select Intel Galileo Gen 2
- Go to **Help > Firmware Update**. Click *yes* to proceed, and the process will begin. It will take a few minutes. **Be sure not to unplug the power or USB while it is updating**
- When it is done it will pop up with a *Target Firmware updated successfully* message

Uploading your First Sketch

- Go to **File > Examples > 01.Basics > Blink**
- Make sure the **Serial Port** and **Board** are still correct
- Click the **Upload** button

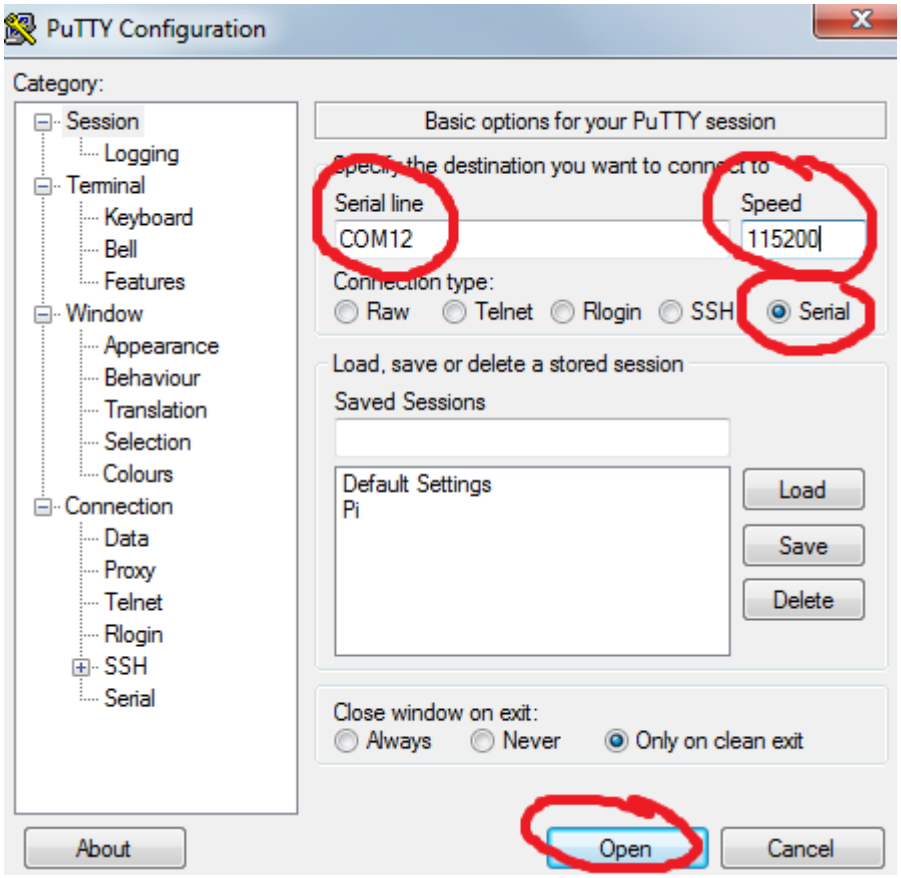
- After the upload completes, you should see a tiny green LED blinking once per second. The LED is connected to Pin 13 and is labelled “L” directly next to the USB port on the board

Connecting to the Linux Shell

- Plug in the **6-pin serial-USB** cable from the Galileo into your USB drive. Pay attention - it should be marked on your board which end is black and which is green
- Reopen **Device Manager** to find the COM number of the serial USB port. It will also be listed in the *Tools > Serial Port* menu in Arduino
- Open a **Serial terminal program**. I've included my favorite (PuTTY) in the Github. Another common choice is Tera Term, which you can download if you choose

Connecting to the Linux Shell

- Open your Serial terminal program and set the **COM port** to the same as your serial cable and set the **baud rate (speed) to 115200**



Connecting to the Linux Shell

- Connect, and it should open a text terminal window. Hit enter, and you will be prompted for a login. Enter **`root`** and hit enter (no password)
- You are now in the shell! Make sure it works by running a command like **`pwd`**, which should return **`/home/root`**

Testing the Shell (Learn a few Linux commands)

- Type **`pwd`**. This stands for **print working directory** and will print the *path* to your current location in the directory tree (in Linux, folders are called directories). This path might be the **root directory** (**/**) or a deeper directory like **/home/root/**

Testing the Shell

- Type **`cd`**. Without any arguments, this returns you to the root directory.
- Type **`ls`**. This lists the folders and files in your current directory. If you are in the root directory, you will see a bunch of folders. If you are in another directory, you might see files and folders or even nothing at all.

Testing the Shell

- Type **`cd [directory name]`** (pick any one of the directories that was listed when you typed **`ls`**). For example, type **`cd home`** or **`cd usr`**. cd means **change directory** and will move you into the directory name that you've chosen. Type **`pwd`** to show that it worked.
- Type **`ls`** see what's in your new directory, and the **`cd`** to return to the root directory

Testing the Shell

- Try making a file! Type **`touch [filename].txt`** to create a new file. Choose any filename you want. For example, I might say **`touch test.txt`**
- Add content to your file by typing **`echo "Hello, World!" > test.txt`**. Substitute with any string of text you want and the name of your file.
- Check that it worked by using the **`cat`** command to read the contents of your file. For example, **`cat test.txt`** and the terminal will print *Hello, World!* to the terminal.
- Remove your file by typing **`rm test.txt`**

The Bigger Linux Image

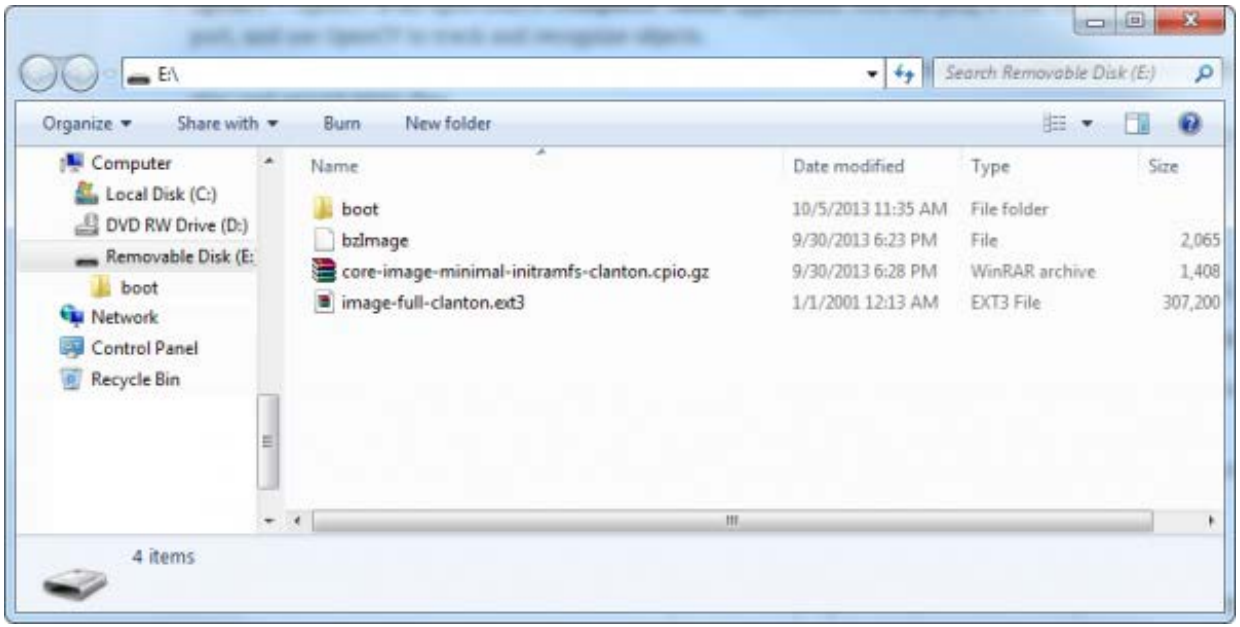
- Download the image from https://communities.intel.com/docs/DOC-22226 (It's called SD-Card Linux Image) and unzip it
- Get your micro-SD card and adapter and insert it into your computer

The Bigger Linux Image

- **Format** the SD card by opening a Windows Explorer and finding the SD card (in My Computer > Devices likely)
- Right-click the card and select **Format** from the menu
- Make sure the File System setting is set to **FAT32**
- Click **Start** and **OK** (*this erases all SD card content!!)*

The Bigger Linux Image

- When it's complete, move the **contents** of the ***image-full-galileo*** folder onto the card. Make sure they are in the **top level**:



The Bigger Linux Image

- Remove the SD card from the computer
- Power down the Galileo by unplugging first the USB and then the power, and then insert the SD card into the slot, and reboot the Galileo
- Connect to the Linux shell and type **`python`** into the command window. It should open a Python interpreter from the SD card image

Moving Files to the Galileo (mini Python project)

- Open the *record_time.py* file from the Python Scripts folder in the Github. This is a quick Python file that records the date and time
- You need to open it with a text editor like Notepad. I recommend downloading Sublime Text, it's my favorite text editor:

  http://www.sublimetext.com/2

Moving Files to the Galileo

- Connect an ethernet cable from your computer to the Galileo
- We use **scp** to transfer files. This already exists on Mac. For Windows, I've included a **pscp** program in the same folder to use for file transfer

Moving Files to the Galileo

- Open a **Windows Command Prompt** and navigate to the directory where *record_time.py* is by using **`cd`** to change directories and **`dir`** to list the contents of a directory (because this is Windows, not Unix/Linux where you use ls)
- In your **Linux shell**, type the command **`ifconfig`** to see internet connection settings. Under the **`eth0`** (ethernet) output, on the second line, look for the phrase **`inet addr`** followed by an IP address. Note or copy this address.

Moving Files to the Galileo

- In the Windows Command Prompt, type:
  - **`pscp -scp ./record_time.py root@[YOUR IP ADDRESS HERE]:/home/root`**
- for example, mine might look like
  - **`pscp -scp ./record_time.py root@192.168.137.228:/home/root`**
  - but your IP will be different
- When prompted, type **`y`** to store the key in cache

Moving Files to the Galileo

- This saved the file into the `/home/root` directory on the Galileo. In your Galileo command prompt, type **`cd /home/root`** to access this directory and **`ls`** to list the contents. *record_time.py* should be listed in the contents
- Run the file by typing **`python record_time.py`**. The program will access the current date and time and write it into a text file. Feel free to run the program a couple times.

Moving Files to the Galileo

- Use the command **`ls`** again to list the contents of the directory. Now that you have run the program, you should see a file called *button_log.txt*
- View the contents of the text file with **`cat button_log.txt`**. It will show the timestamps of when you ran the *record_time* program