Labrador Documentation

(Work in Progress)

Chris Esposito <[admin@espotek.com](mailto:admin@espotek.com)>

[Licence](#_nvtsa7l657xx)

[Introduction](#_h5l82z3yf8k7)

[Interface Installation](#_5d3u6xny64di)

[Using The Hardware](#_ud5gi9tmz2wv)

[Pinout](#_s2in9xq0a87y)

[Connecting Your Labrador to a Breadboard](#_mh6ao4kf6do2)

[Connecting Your Circuits to Labrador (WIP)](#_qjxc40pfrd3d)

[A Simple Circuit Example](#_dycg915l400t)

[Firmware and Bootloader Mode](#_l7n6121mw9ba)

[Using the Interface](#_olaruya39jb5)

[Calibration](#_zhfl1cpty8ef)

[Shortcuts](#_i3eo97ftlkma)

[Buffers](#_kt6apid830a8)

[Oscilloscope](#_3k6d90zqof4)

[Signal Generator](#_u543n9k12xhy)

[Logic Analyzer](#_vyw65k1ta16v)

[Power Supply](#_pl8a7t94igar)

[Multimeter](#_hdf0v5hocu2l)

[Troubleshooting](#_ly9w38810t3m)

[The software interface is greyed out and says “Device Not Detected” (Windows).](#_jjph6ds08als)

[USB connection issues (Other platforms).](#_j6stnwx06z1s)

[All other problems.](#_rr5cu3nnc2aw)

# Licence

This document is licensed under the Creative Commons 4.0 BY-SA licence. You are free to download it, share it, and use it for any purpose (including commercial purposes), as long as any edits you make are shared back to the community, and that proper credit is given (a link back to espotek.com will suffice).

<http://creativecommons.org/licenses/by-sa/4.0>

# Introduction

Welcome to the initial documentation for the EspoTek Labrador!

Please note that it’s currently in a WIP state. If you have any suggestions, please right-click and leave a comment! It’s harder than you’d think to write a guide for something you invented yourself, so any questions or suggestions help!

Note that the “Using the Hardware” section is mainly based on confused emails I got from software people. If you’re familiar with electronics, you can safely skip it.

If you’re familiar with lab equipment, you can also skip most of the “Using the Interface” section, but I do recommend reading the section on “buffers” and everything in bold!

Apart from that, enjoy! Let’s move onto the setup stage.

# Interface Installation

Binary releases for all 3 platforms are available here:

<https://drive.google.com/open?id=0B7U0ulRLHf8cRVBkeFc2SHpUOGs>

The Windows .exe installer contains both 32 and 64 bit versions.

There are separate .deb files for 32-bit and 64-bit Linux.

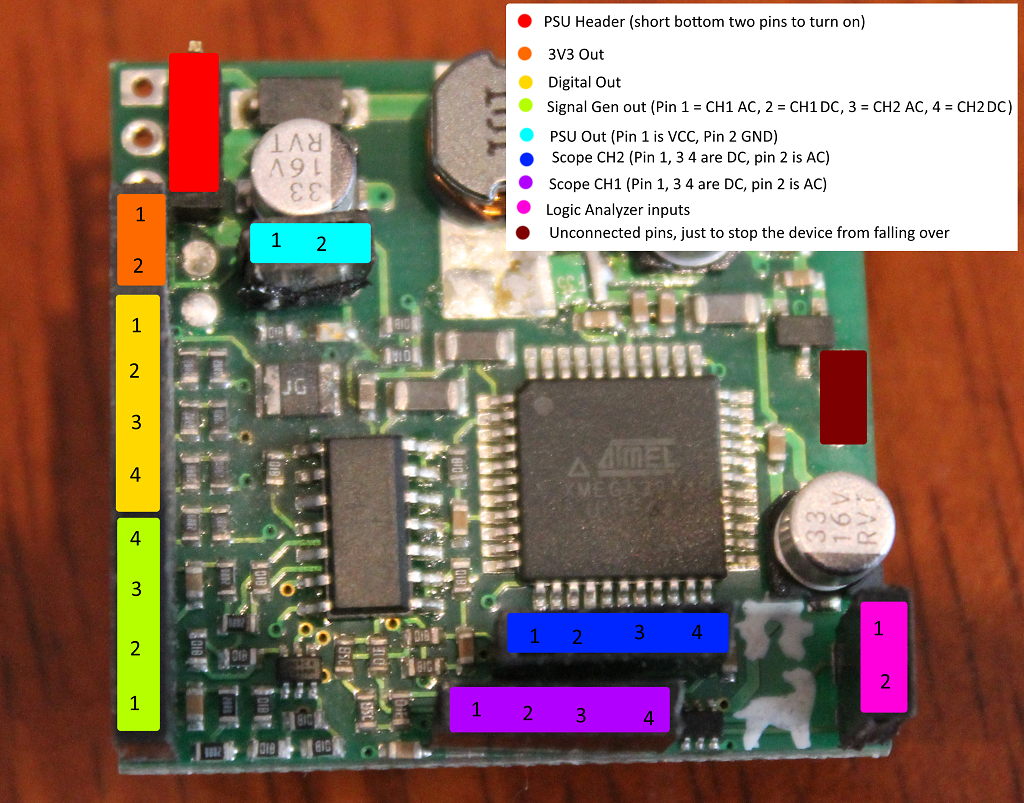
The OSX software is 64-bit only.

It’s recommended that you install the Linux software using the install script (not by double-clicking the .deb). If you do not use the install script, you will need to restart after installing the .deb.

The software on all platforms is in a stable state (as far as I’m aware), but there may be some minor bugs present!

# Using The Hardware

## Pinout

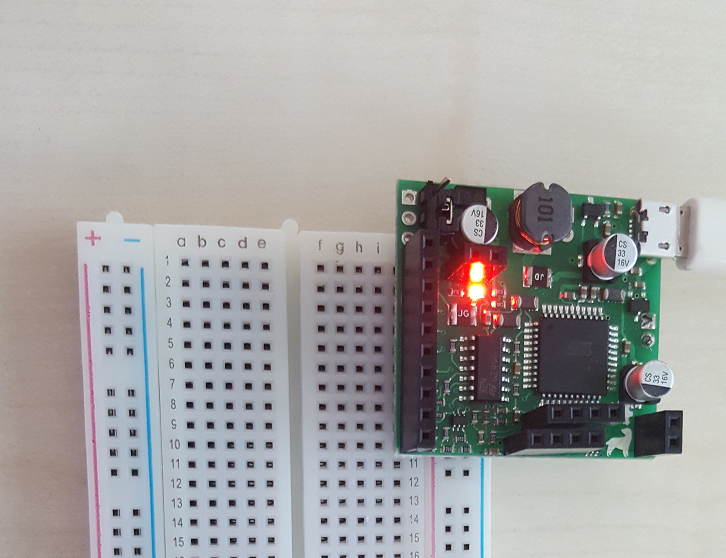


Yeah, it’s not very neat. Sorry.

If you don’t know what AC coupling is, please don’t use the AC-coupled pins. Use the DC-coupled pins instead as they don’t alter your signals.

Currently asking some folks on Fiverr to make a nicer diagram for me. Will upload when it comes through.

## Connecting Your Labrador to a Breadboard



(image credit: Gianpaolo Macario)

You can connect your Labrador board as shown above. Power supply pins should go in the side rails to prevent any shorts.

Of course, if you don’t have a breadboard handy, the board can also be used without one. But where’s the fun in that?

## Connecting Your Circuits to Labrador (WIP)

There are a few important things to keep in mind when using your Labrador board:

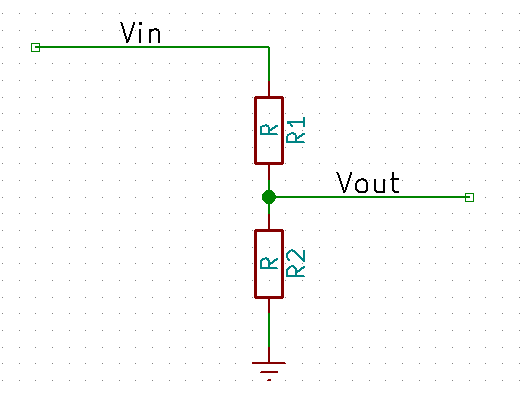
* Labrador is a *collection of independent tools* that happen to share a single connection and software interface. Apart from small amounts of unintended crosstalk, there are no internal connections between Labrador’s different instruments!
* All voltages are GND-referenced, with the exception of the multimeter which measures CH1 referenced against CH2. If your oscilloscope measures 5V, it means that the voltage at the node your oscilloscope pin is connected to is 5V higher than the Labrador’s GND pin[[1]](#footnote-0). Make sure you connect the GND point of your circuit under test to Labrador’s GND pin!
* If you don’t know what AC coupling is, don’t use the AC-coupled pins. Use the DC-coupled pins instead as they don’t alter your signals.
* Labrador is idiot-proof. You should be able to short any two pins together without damaging your board, and even if you do something like connecting a 12V external power supply directly to one of your digital output pins, they’ll nobly sacrifice themselves to ensure the rest of the board will still function (and you can fix it by replacing a single resistor). Experiment a bit! You can’t do too much wrong.
* Labrador is idiot-proof, but not complete-idiot-proof! If you start touching live wires to random components on your board, I cannot guarantee what will happen. Please make sure you **only connect your circuits to the external header pins**, and don’t dangle live wires near the microcontroller!

## 

## 

## A Simple Circuit Example

The simplest circuit I can think of is the voltage divider:

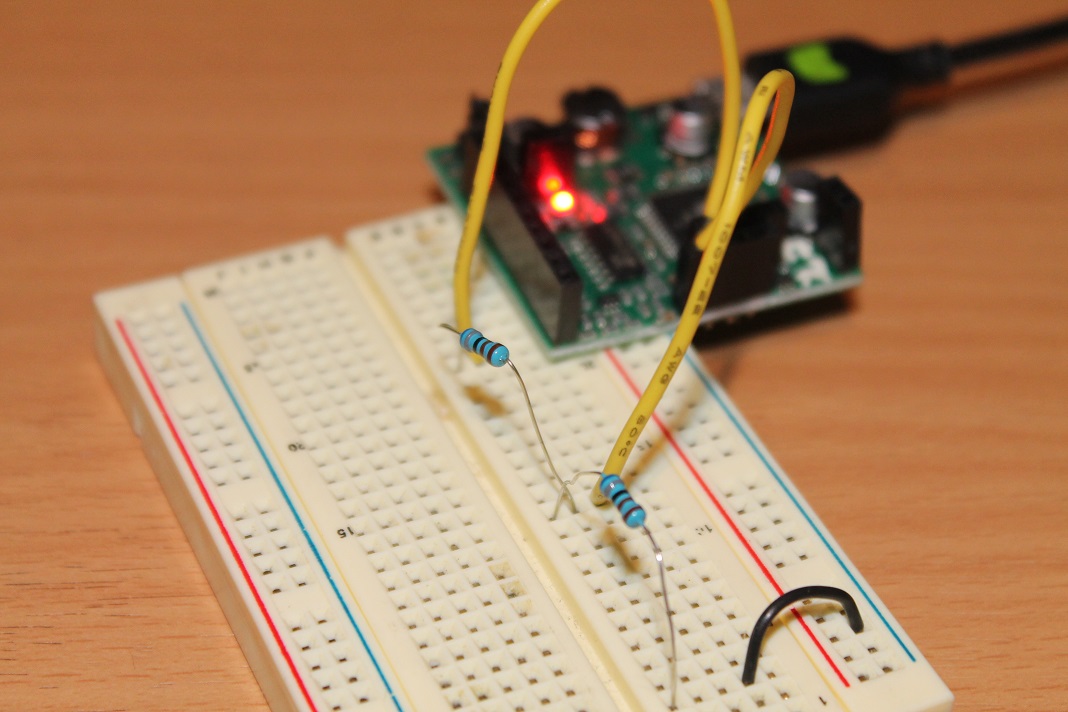


It transforms the input voltage Vin to the output voltage Vout according to the equation:

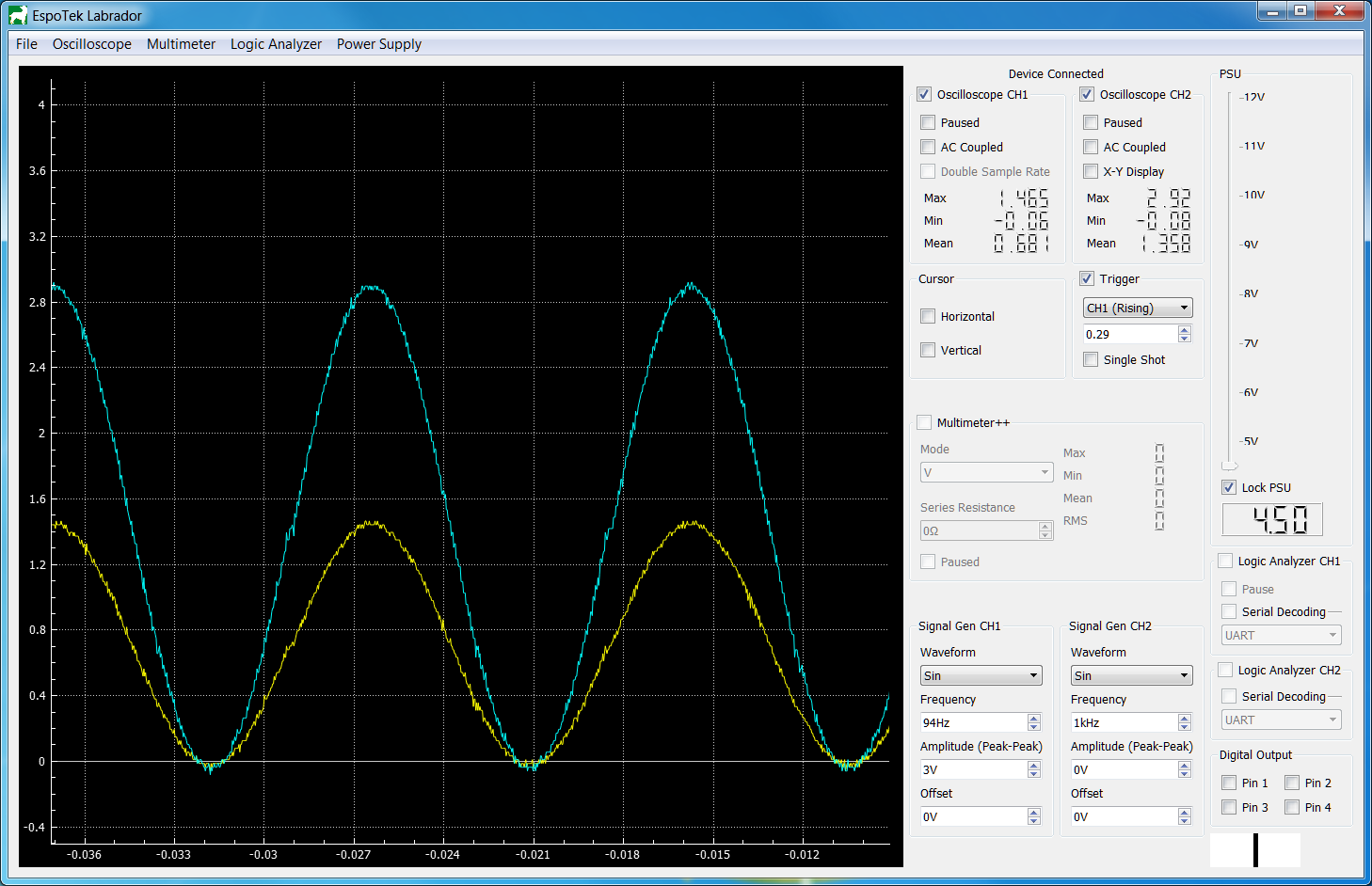
For example, if you used a 1k resistor for R1 and a 10k resistor for R2, the voltage at Vout would be equal to 10/11ths of the voltage at Vin.

I tested this theory in real life by building the voltage divider and connecting it to Labrador!

You can do the same. If you’ve not dealt with oscilloscopes or signal generators before, it’s the best place to start.



Above is a picture of my breadboard. The oscilloscope CH2 pin is connected directly to Vin with a standard solid-core jumper wire. The oscilloscope CH1 pin is connected to Vout. The GND node of the voltage divider is connected to the GND pin of the Labrador[[2]](#footnote-1). Vin is connected to the signal generator CH1 DC output. The resistors used were both 6.8k, meaning that the voltage at Vout should be approximately half[[3]](#footnote-2) the voltage at Vin. You can do the maths if you don’t believe me.



Above is a picture of my Labrador’s software interface. I used the signal generator to output a 3V sin wave. As you can see, the yellow trace (oscilloscope CH1), is half the height of the blue trace (oscilloscope CH2). Thus, the voltage at Vout is half of the voltage at Vin!

# 

# 

# Firmware and Bootloader Mode

Each version of the software interface is compatible with a specific version version of the board firmware. If there is a mismatch, then the software will automatically flash the required firmware and continue.

In case this goes wrong, you can enter bootloader mode and flash manually.

Bootloader mode is accessed by shorting Pin 4 from the digital out to GND. While in bootloader mode, you can update the board’s firmware and recover from potentially bad firmware updates.

On Windows, the simplest way to achieve this is to run flash.bat from your /firmware folder in your Labrador directory (C:/Program Files/EspoTek/EspoTek Labrador).

For Linux and Mac, you’ll need to download [dfu-programmer](https://github.com/dfu-programmer/dfu-programmer) and run the commands from the .bat file manually in terminal. Note that on Mac and Linux, unlike Windows you’ll want to flash the firmware ending in \_0002.hex.

# Using the Interface

Video here:

<https://player.vimeo.com/video/180909597>

It’s for an old version of the software. I need to make a new one.

## Calibration

I recommend calibrating your device before doing anything else. Select Oscilloscope->Calibration and follow the on-screen instructions. During step 1, you’ll want to ensure nothing is connected. During step 2, both oscilloscope channels should be shorted to the USB connector on the board.

## 

## 

## Shortcuts

Some shortcuts may be different on Mac OS X.

Try pressing Command instead of Ctrl.

|  |  |
| --- | --- |
| **KEY** | **ACTION** |
| Ctrl-B | Cycles logic analyzer CH1 baudrate forward |
| Ctrl-Shift-B | Cycles logic analyzer CH1 baudrate backward |
| Ctrl-Alt-B | Cycles logic analyzer CH2 baudrate forward |
| Ctrl-Alt-Shift-B | Cycles logic analyzer CH2 baudrate backward |
| Z | Snaps display to vertical/horizontal cursors |
| M | Brings up window to directly type in oscilloscope range |
| C | Takes snapshot of CH1 (CSV file) |
| V | Takes snapshot of CH2 (CSV file) |
| Up Arrow | Scroll in (useful for laptop users w/o mouse) |
| Down Arrow | Scroll out (useful for laptop users w/o mouse) |
| Left Arrow | Shift trace left by 10% (add delay) |
| Right Arrow | Shift trace right by 10% (reduce delay) |
| Ctrl-Left Arrow | Shift trace left by 50% (add delay) |
| Ctrl-Right Arrow | Shift trace right by 50% (reduce delay) |
| W | Alias of up arrow key |
| A | Alias of left arrow key |
| S | Alias of down arrow key |
| D | Alias of right arrow key |

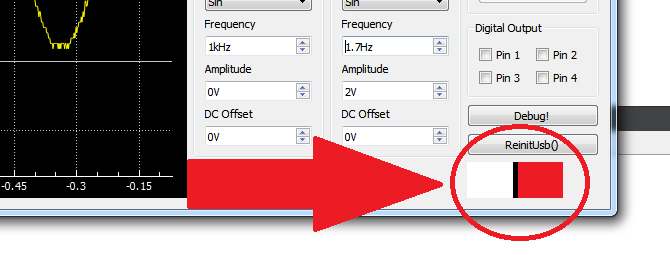
## Buffers

Due to RAM and USB bandwidth limitations, each board has only two buffer resources available. Each device requires a certain amount of these buffer resources to function.

|  |  |
| --- | --- |
| **Device** | **Buffers needed** |
| Oscilloscope (375ksps, per channel) | 1 |
| Oscilloscope (750ksps) | 2 |
| Logic Analyzer (3Msps, per channel) | 1 |
| Multimeter++ | 2 |
| All other devices | 0 |

This limits which devices can be turned on at any one point in time. For example, the Multimeter and Oscilloscope cannot be turned on at the same time since this would require 3 buffers. One would need to be switched off.

The number of remaining buffers is displayed graphically in the bottom right hand corner of the screen. To free up buffer resources, devices can be switched off by unticking the box next to their name.



## Oscilloscope

The oscilloscope was designed to be beginner friendly. For this reason, settings such as volts/div and attenuation have been replaced by a whole new interface.

**To zoom in and out**: point the mouse cursor at the area of interest and scroll. If no key is pressed, it will affect the voltage axis. If the Ctrl key is pressed, it will affect the time axis. If you’re on a laptop, you can use the up/down arrows in place of the mouse wheel.

**To add cursors**: click and drag. Left-click drops horizontal cursors and right-click drops vertical cursors. Relevant stats are displayed automatically, and ***you can focus on the area inside the cursors*** by selecting Oscilloscope>Range>Snap to Cursors or pressing Z.

**To measure stats such as the waveform max, min, mean and RMS:** enable them from the Oscilloscope>CHn Stats menu at the top. They are disabled by default because they cause display issues at low resolutions.

**To AC couple**: ensure that the input is connected to the AC pin on the board, and that the “AC Coupled” option is selected in the interface.

**To enable 750ksps:** on channel 1, select “double sample rate”. The default is 375ksps.

All references shown are relative to USB GND (Pin 2 of the power supply).

## Signal Generator

This is fairly self-explanatory. Select your wave shape, frequency, amplitude, offset and you’re done. One thing that should be noted to engineers is that **the offset refers to the difference between the lowest point of the wave and zero, and \_NOT\_ the DC component from the Fourier series**. This is to avoid confusing new users, since the positioning of this component changes depending on wave shape.

*A waveform of type sin with frequency, amplitude and offset all set to 1 will produce a 1V sine wave oscillating between 1V and 2V at a frequency of 1Hz.*

## Logic Analyzer

The logic analyzer uses the oscilloscope display to show a waveform, and can serial decode too. At the moment, only UART (8-N-1) is supported. Baudrate for serial decoding can be set by selecting Logic Analyzer > UART CHn in the top menu. SPI and I2C don’t work yet.

## Power Supply

Unlock, drag the slider, re-lock. Bob’s your Uncle.

## Multimeter

The multimeter measures the difference between the oscilloscope CH1 and CH2 pins, and uses 12-bit samples at 375ksps. The normal oscilloscope uses 8-bit samples only, **so sometimes the multimeter will function as a better oscilloscope than the oscilloscope will**!

**To measure voltage:** set the mode to “V” and connect the load in parallel to Pin 1 of Oscilloscope CH1 and Oscilloscope CH2.

**To measure current:** set the mode to “I” and connect the load in series to Pin 1 of Oscilloscope CH1 and Oscilloscope CH2. Add a reference resistor across Pin 4 of the two oscilloscope channels that:

* Is at most 10% of the load resistance.
* Will have a drop of at least 50mV.

Then type in the reference resistance used into the box.

As an example, if you are measuring the current through a circuit that you suspect draws 50mA from a 3.3V source, you can guesstimate its load resistance to be 66Ω (V=IR). 10% of this is 6.6Ω. To drop 50mV, you’ll need to use at least a 1Ω resistor. So the reference resistor you’ll need for this circuit should be in the range of 1Ω-6.6Ω Ohms. A 2.8Ω or 3.3Ω resistor will be ideal in this case.

**To measure resistance/capacitance:** connect your load to Pin 4 of oscilloscope CH1 and CH2. Connect Pin 1 of Oscilloscope CH1 to Pin 3 of the signal generator. Connect your reference resistor between Pin 1 of Oscilloscope CH2 and GND.

For resistance, you’ll want to aim for a reference resistor as close to the load resistance as possible. To be more accurate, you may want to take multiple measurements.

For capacitance, you’ll want to choose a reference resistor that gives you an RC time constant of about 1ms.

# 

# 

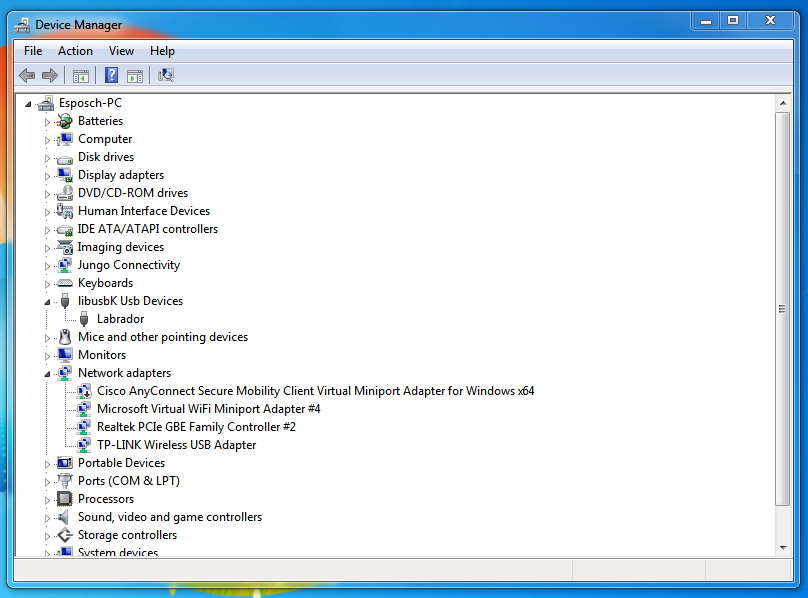
# Troubleshooting

### The software interface is greyed out and says “Device Not Detected” (Windows).

First, try the steps outlined here:

<https://www.youtube.com/watch?v=t2F1rFmyQmY>

If that doesn’t solve the problem, open the Device Manager. You should be greeted with a screen like this:



Look for *Labrador*.

**If it’s under the heading of “libusbK USB Devices”** but has a little yellow triangle on it, then the driver is correctly installed and configured - however there is not enough bandwidth.

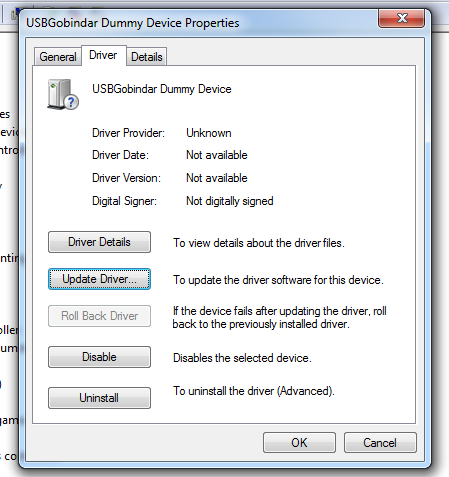
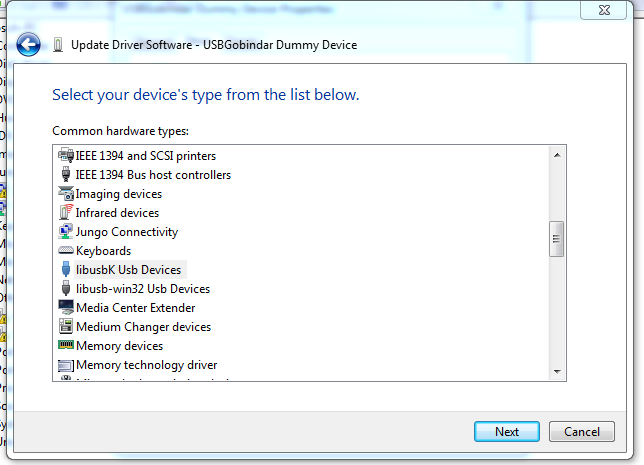
Move Labrador to a different USB port that does not share bandwidth with devices that stream constantly (in proper terminology: isochronous/interrupt devices). This includes things like keyboards, mice and speakers.

Some suggestions:

* If you’re using a hub, connect your other devices to the hub and give Labrador its own dedicated port.
* Plug your other devices into your USB2.0 ports, and Labrador into the USB3.0 or eSata ports (or vice versa).
* Devices that stream constantly are the most likely to compete with Labrador for bandwidth. If you are forced to share a port with Labrador, share it with something like a USB flash drive or card reader that does not stream constantly.
* If need be, disconnect unneeded USB devices.

Your Labrador board should start working.

**If it’s under the heading of “Other Devices”**, then the driver is not correctly installed and configured. To configure manually:

* Double click on the device name to open the Properties window.
* Select “Update Driver”, from the Driver tab.  
  
* Select “Browse my Computer for Driver Software” (Locate and install driver software manually).
* Select “Let me pick from a list of devices on my computer”.
* Then select “libusbK Usb Devices” from the list and the device will connect.  
  
* Note: if you cannot see “libusbK Usb Devices”, do not try to be smart and select libusb-win32, libusb0 or anything similar. Instead, re-run the installer and make sure that both driver boxes are ticked and installation completes successfully.

**If the device is not found at all.**

* Ensure you’re using a cable that has data lines. Some micro-USB phone charging cables are power-only!
* Remove all wires from your Labrador board. And unplug and re-plug it in, twice if necessary. This will get it out of bootloader mode.

### USB connection issues (Other platforms).

On non-Windows platforms, the Labrador reserves the entire bandwidth of a USB hub in order to ensure it gets a solid connection. Move Labrador to a different USB port that does not share bandwidth with devices that stream constantly (in proper terminology: isochronous/interrupt devices). This includes things like keyboards, mice and speakers.

Some suggestions:

* If you’re using a hub, connect your other devices to the hub and give Labrador its own dedicated port.
* Plug your other devices into your USB2.0 ports, and Labrador into the USB3.0 or eSata ports (or vice versa).
* Devices that stream constantly are the most likely to compete with Labrador for bandwidth. If you are forced to share a port with Labrador, share it with something like a USB flash drive or card reader that does not stream constantly.
* If need be, disconnect unneeded USB devices.

Your Labrador board should start working.

Some Mac OSX users have reported driver-related issues that prevent the Software Interface from communicating with the Labrador board. To fix it, install [homebrew](https://brew.sh/) and then run the following two commands from the terminal:

brew update

brew install libusb

After resetting your computer, it should work.

### All other problems.

If you’re having any other issues with your board, I’m happy to help.

The best thing to do is to post an [issue on the GitHub page](https://github.com/EspoTek/Labrador/issues), that way any solutions will be posted publicly.

The second best thing to do is to send me an email. I can be found at [admin@espotek.com](mailto:admin@espotek.com).

1. In Electrical Engineering there is no such thing as an “absolute voltage” at all; only how much higher or lower the voltage at one particular node is compared to another. A 1.5V battery doesn’t “have” 1.5V; it just means that the positive terminal is 1.5V higher than the negative terminal. [↑](#footnote-ref-0)
2. If you’re not familiar with how the connections in a breadboard work, I recommend checking out [Adafruit’s tutorial](https://learn.adafruit.com/breadboards-for-beginners/breadboards). I also recommend tearing the backing off of a breadboard at least once in your life. Just make sure you have duct tape handy to put it back together again. [↑](#footnote-ref-1)
3. I say “approximately half” because no component is 100% accurate. The resistors I used have a tolerance of +/- 1%, meaning that the actual resistance could be anywhere between 6.73k and 6.87k. [↑](#footnote-ref-2)