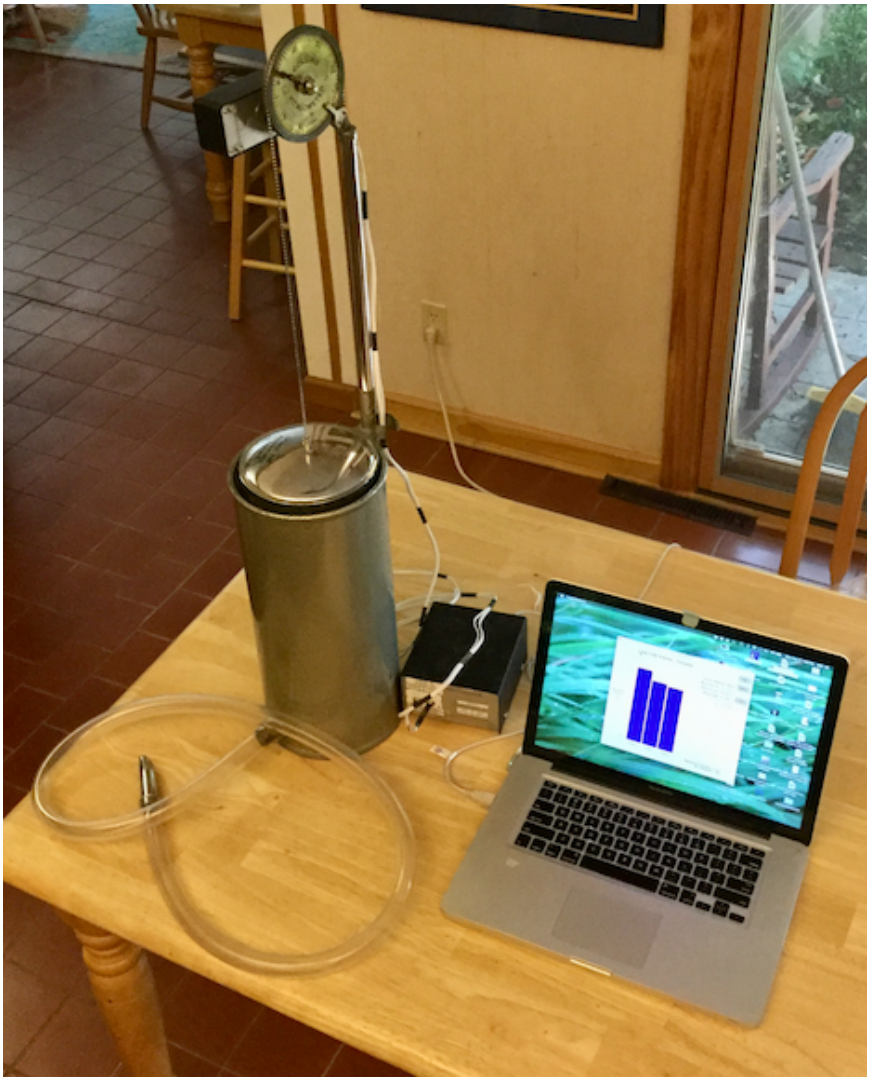
LungTidal (SONIC)

Instructions for using the LungTidalSonic program for measuring lung tidal volume with the spirometer and setting up the spirometer interface device for OSX.



Updated May 2018 (from May 2009)

LUNGTIDAL (SONIC)

Manual

Updated September 2018 from May 2009

Introduction

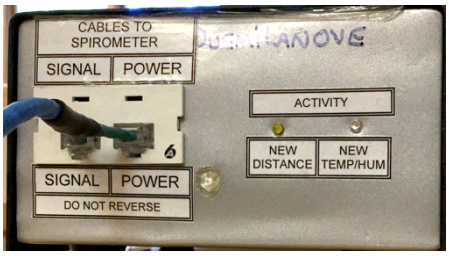
The following are the instructions for using the LungTidalSonic program and for setting up the LungTidalSonic, Arduino based, computer interface device that measures the spirometer volume with a sonic distance measurement sensor.

The LungTidalSonic program is a standalone Java application created in Processing that reads and graphs USB port serial data coming in a from an Arduino device connected to the computer's USB port.

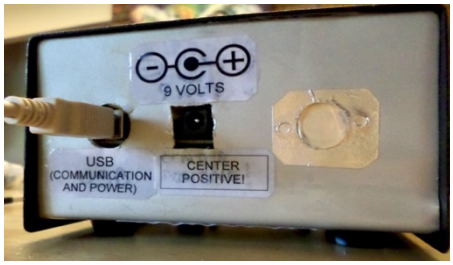
Setup/Checkout

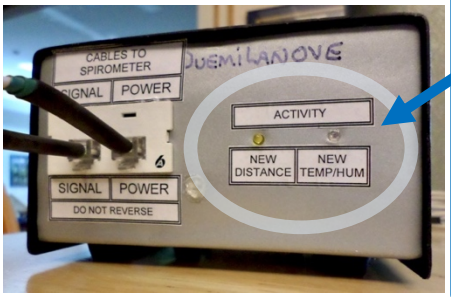
0) Install the FTDIUSBSerialDriver_v2_4_2 serial port driver and then reboot the computer. This step is not necessary if the serial port driver has already been installed, but that is often unlikely. This serial driver is on the LungTidalSonic CD. It can also be downloaded from a Google search.

1) Plug the spirometer's cable phone plugs into the LungTidalSonic phone jacks. Make sure the phone plug labeled 'P' is plugged into the Power labeled phone jack. The plug labeled 'S' must be plugged into the Signal labeled phone jack. Do not reverse the connections.



2) Connect the Arduino box to the computer via a USB cable connecting at the Arduino box backside. Ignore the power connection that is also at the box backside.





Two LEDs on the Arduino box front side indicate activity. The "NEW TEMP/HUM" LED flashes dimly on a regular basis every few seconds. The "NEW DISTANCE" LED flashes when there is a new different distance measured. These LED flashes indicate the Arduino box program is powered and running its program. Waving your hand underneath the sensor should result in LED activity.

3) Make sure the connection cable is tucked into the metal hose clip on the spirometer. Otherwise the cable might drape down where the sonic sensor might sense it.

4) Run the LungTidalSonic program. LungTidalSonic will inform you if it does not see the Arduino or if it does not find the calibration file.

5) Adjust the chain and pulley arraignment such that the indicator on the pulley reads zero when the dome is at its bottom, empty state. The water reservoir must have been properly filled prior to this step.

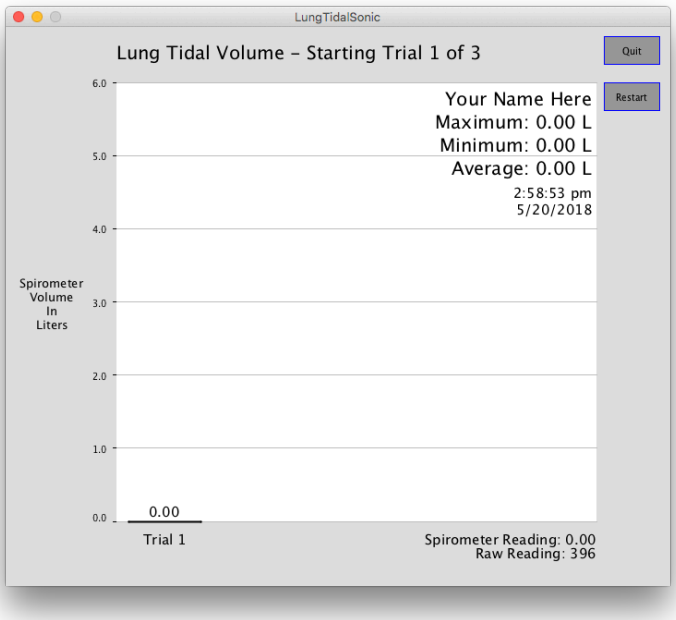
6) Once the pulley is properly zeroed, with LungTidalSonic running, slowly move the spirometer up and down to verify that the dial readout matches the LungTidalSonic Spirometer Reading values.

7) When ready "press" the Restart button on the LungTidalSonic screen.

Operation

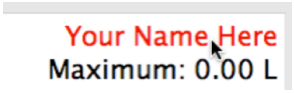
Running the Trial Series

The LungTidalSonic program is ready to measure three lung tidal trials after the Reset button is pressed. A Print button for printing appears after the third lung tidal measurement. There is a location for the tester's name above the Maximum reading result. You can enter the tester's name at any point before, during or after the trial series.



Entering Your Name (Is Optional)

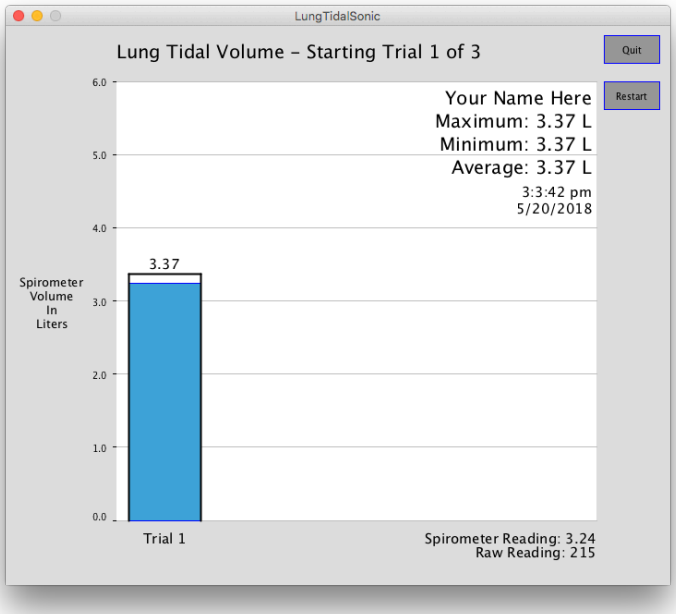
1) Entering your name is optional. Its intent is when sessions are printed or saved. Move the mouse pointer to the "Your Name Here" location. This is the name entry location. The text should turn red when the mouse is over the text.



2) Anything you type on the keyboard becomes part of the name entry when the text is red. Press the delete key to delete one character at a time whatever text is already in the name entry. Then type your name and finish by moving the mouse off the name entry.

TRIAL SERIES

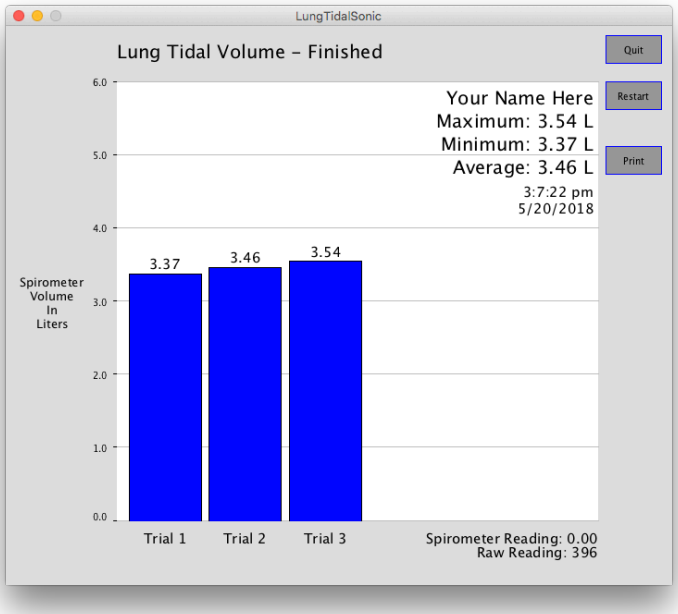
The LungTidalSonic program is ready to measure three lung tidal trials after the Reset button is pressed. LungTidalSonic shows increasing tidal lung measures in a bar graph form. The bar graph moves up and down according to the current measurement. The bar graph's outline is the maximum value measured. The bar graph's filled inside is the current measured value.



LungTidalSonic waits until the reading value is above a small amount before considering the trial as having started. Therefore, it may seem like there is no response when starting to blow each trial.

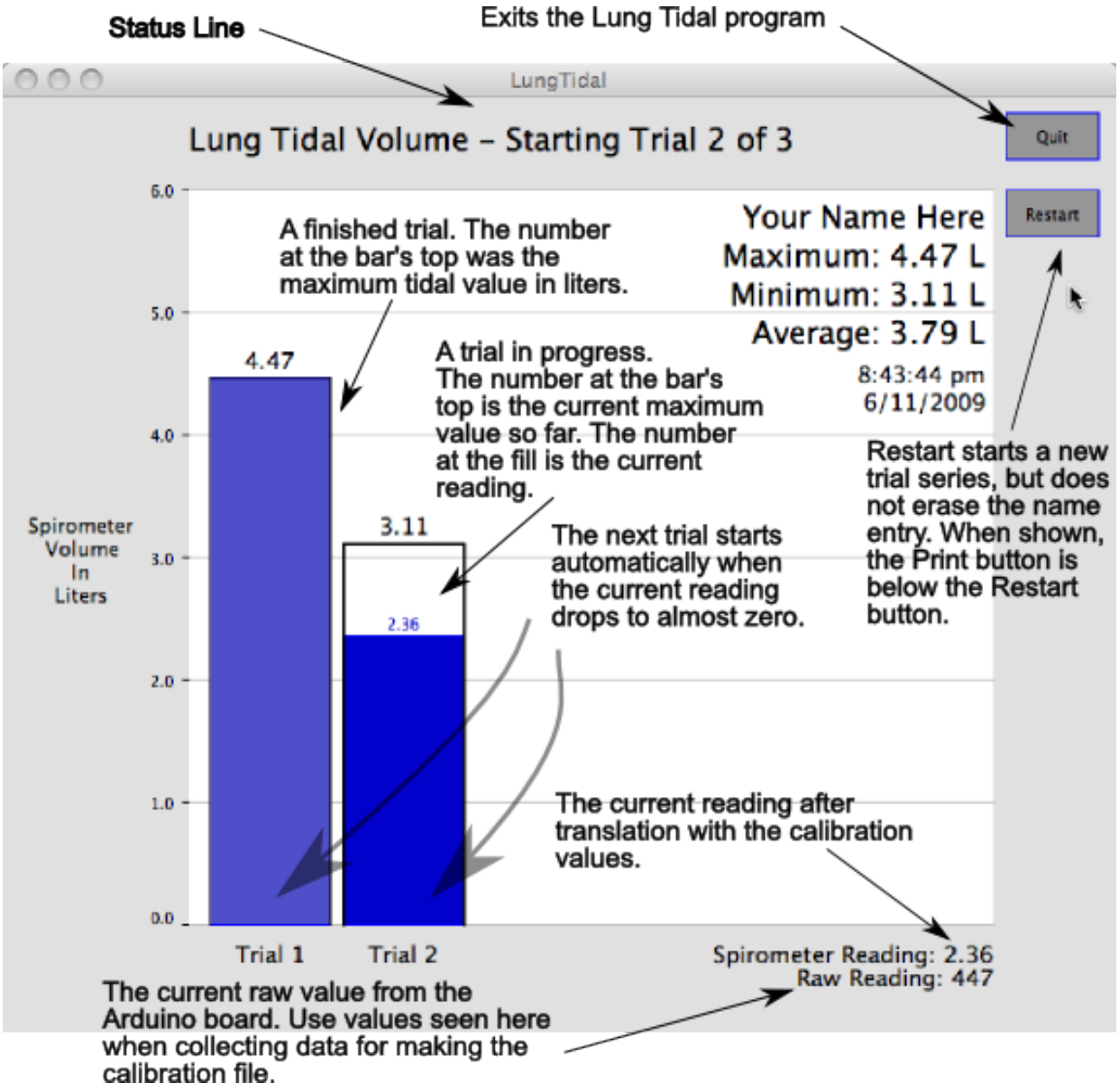
LungTidalSonic ends the trial when measurements reach zero after it sees enough measurement to consider the trial as having started. It immediately starts the next trial.

The lung tidal measurement session ends after three trials. The Print button appears. The print button creates a PDF of the current LungTidalSonic screen. This PDF then opens in Apple's Preview application. It will take some time before Preview opens the PDF if Preview is not already running. Use Preview to print the PDF to your printer of choice. It is recommended that you close the PDF document after printing but leave the Preview application running. Doing so speeds up the next printing.



The Restart button may be used at any time to restart the trial series. Restarting does not change the name entry text, so remember to change the name entry if you are printing the trial series screens.

THE LUNGTIDAL SCREEN



TECHNICAL BACKGROUND

The Arduino device is an “Arduino Duemilanove” programmable micro-controller chip (PIC) circuit board designed for connecting to a standard USB port. The “Duemilanove” Arduino model was current in 2009 when this LungTidal system was created but is now considered obsolete in 2018. This obsolescence results in awkwardness regarding the USB connection because a serial port driver must be installed at the connecting computer.

The black box mounted on the spirometer at the pulley contains an ultrasonic distance sensor (an HC-SR04) and a temperature/humidity sensor (a VMA311, which is a DHT11 but with a different pin order). The Arduino PIC has input and output connections connected to these sensors. A program placed on the Arduino PIC continuously transmits out its USB port numbers corresponding to the distance measured for the spirometer dome height. That distance is calculated based upon timing sound bursts bounced off the dome top by the HC-SR04. Distance is calculated using the speed of sound in air. A correct sound speed in air requires knowing the air temperature and air humidity measured by the VMA311(DHT11) sensor.

The program performing distance calculations is on the Arduino board and will remain loaded until intentionally replaced. As previously mentioned, this program continuously sends distance measurement values out the Arduino’s USB port to whatever is connected to that USB port. This is a one-way communication scheme.

The LungTidalSonic program on the connected computer needs to see the Arduino device connected to the USB port. LungTidalSonic will see the Arduino device only if a USB driver for the Arduino board is installed on the computer and if the Arduino is also connected via a USB cable. When this happens the serial driver name “/dev/tty.usbserial-A7006Tac” is in the list of serial ports available to a Java program. LungTidalSonic is a java program. The USB driver name is hardcoded in the LungTidalSonic java program. The particular USB driver installed to see this older Arduino provides that name. LungTidalSonic must coordinate with that name.

The LungTidalSonic program uses a “calibration_data.tsv” text file that provides translation information needed to map the incoming raw numbers to the actual spirometer dial reading. Raw values reported by the LungTidalSonic program corresponding to each 0.5-liter dial reading from 0 to 6 are used to create the calibration file. The “calibration_data.tsv” calibration file needs to be in a folder named “Data” within the folder that contains the LungTidalSonic program. When Processing “exports” the LungTidalSonic application as an OSX application bundle the export process creates and populated the Data folder location within the application bundle using the Data folder present in the source code folder.

Instructions on how to install the USB driver and how to manage the calibration file are in the Appendix. These instructions assume that all the necessary components are in place, functioning and the spirometer is properly filled with water with its chain positioned for zero on the spirometer dial.

The program running on the Arduino is written in C using the Arduino integrated development environment (IDE). LungTidalSonic is written in Processing using the Processing 3 integrated development environment (IDE). The Processing language compiles into Java, so the LungTidalSonic application in its finished form is actually a Java program. The LungTidalSonic application is made with a Java runtime embedded in the LungTidalSonic application bundle.

Note: The look and feel similarity to the Arduino IDE and the Processing IDE is confusing when first encountered. The Arduino IDE is actually a version of the Processing IDE meant for programming Arduino boards using C. Both use the same underlying platform and therefore look and feel much the same.

APPENDIX - INSTALLATION

The LungTidalSonic CD contains the LungTidalSonic application and USB driver in addition to all the source files used to create and maintain LungTidalSonic.

Installing the application only

Use only the **LungTidalSonic** application bundle and the **FTDUSBSerialDriver** installer in the **"LungTidalSonicInstallationFiles"** folder. The **FTDUSBSerialDriver** is a .dmg that requires running the installer. Copy these files to the computer.

Find the **FTDUSBSerialDriver** installer .dmg file and run it to install the driver. The disk image will mount as a disk that contains the installer application. Double click on the installer application and follow the instructions. These are standard installation programs. The computer's administrative password is required and a restart is required.

LungTidalSonic is ready to run once the FTDI USB driver is installed and properly connected. The only required connections are the computer to Arduino USB connector and the Arduino to spirometer connector. The Arduino power connection is not necessary in this application. This external power connector is provided for future use of the Arduino device that might require more power than what is available from the computer via the computer's USB connection.

If you copy the entire LungTidalSonic folder to the computer.

It can be confusing if you copy the entire LungTidalSonic folder from the CD to the computer because the folder contains all the supporting files in addition to the files mentioned above.

APPENDIX - CALIBRATION

The original LungTidal spirometer setup used a potentiometer to sense the spirometer's dial movement. That sensor was not linear. A calibration table was needed to map readings to liters. The current setup instead senses the reservoir height using sound. This ultrasonic sensor is linear. Keeping the calibration idea in the overall scheme has benefit for a number of reasons.

The original calibration table is a simple text file present in the /Data folder within the Processing LungTidalSonic project folder. Processing the development program (IDE) that is used to create LungTidalSonic. Processing is also the name for the program language used. The Processing language is very similar to the Java language.

The end product LungTidalSonic application is a Java program application bundle for OSX. Processing places a copy of the calibration file within that application bundle at /Java/data. One can see and edit this test file with the "Show package Contents" feature of the OSX Finder. Editing the file that is within the

application bundle changes the file for that particular bundle only. There is essentially no record of your change when performed in this manner.

A more methodical way to edit this file is to edit the file in the /Data folder within the Processing source files as was done originally and then "exporting" a new application bundle. This means you will be using the Processing IDE (You'll need to download and install it.). Before exporting the new bundle, you will be able to run LungTidalSonic within the Processing IDE to see how your calibration file edit is working. Then have Processing re-export the OSX application. Copy the newly exported application to where you want it. Note when exporting, you do want to embed the Java runtime.

The calibration_data.tsv is a tab separated values table of data pairs that LungTidalSonic uses to interpolate dial readings between data pairs. The table has enough data pairs to interpolate dial readings for each 0.5 liter. Going beyond 6.0 liters is accomplished by extrapolation using the last two values in the table. The values in this table are the averages from four sets of readings.

The current file contains the following:

```
# Spirometer calibration table
# 10/2018 modified for sonic readings
#
# Separate the data with tabs
# Use a text editor that writes raw text.
# <volume on dial><tab><raw analog data>
0    400
0.5  363.75
1.0  338.75
1.5  303.75
2.0  276.25
2.5  247.5
3.0  222.25
3.5  198.75
4.0  171.75
4.5  153.25
5.0  128.5
5.5  97.5
6.0  77.25
# calibration data end
#
# An alternate USB driver name can be placed here.
# LungTidal will try to use the USB driver name you
# place here if it cannot use the one it was compiled
# for. The format is USB:/the driver name
#
# example - USB:/dev/tty.usbserial-A800eGv4
#
# LungTidal shows the USB driver names it finds when it
# cannot find the one for which it is compiled or set here.
# Thus you can set the name for the USB driver it needs to
```

```
# look for in this file if you know for sure one of the
# drivers it lists as detected is in fact the one for the
# Arduino.
#
USB:/dev/tty.usbserial-A800eGv4
```

Use only a raw text editor such as TextEdit to edit or create the calibration_data.tsv file.

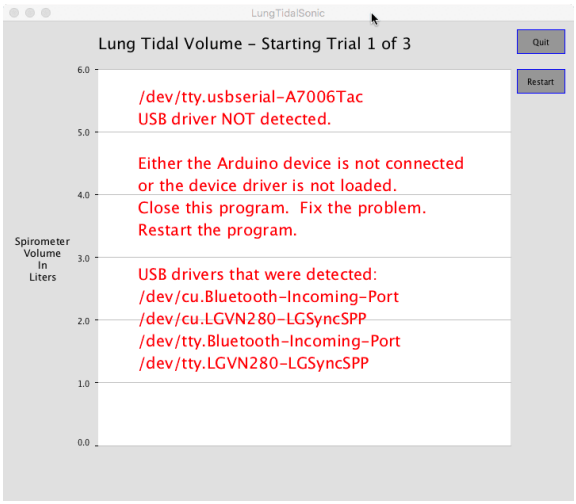
Appendix - Error Screens

The following images show the two LungTidalSonic error screens that might occur. The instructions on these screens are intended to be self-explanatory.

"No USB driver ..."

The "No USB driver ..." screen results from an error trying to read the Arduino through the USB connection. This will happen if the FTDI USB driver is not installed. How to install the driver is mentioned above in the Installation section.

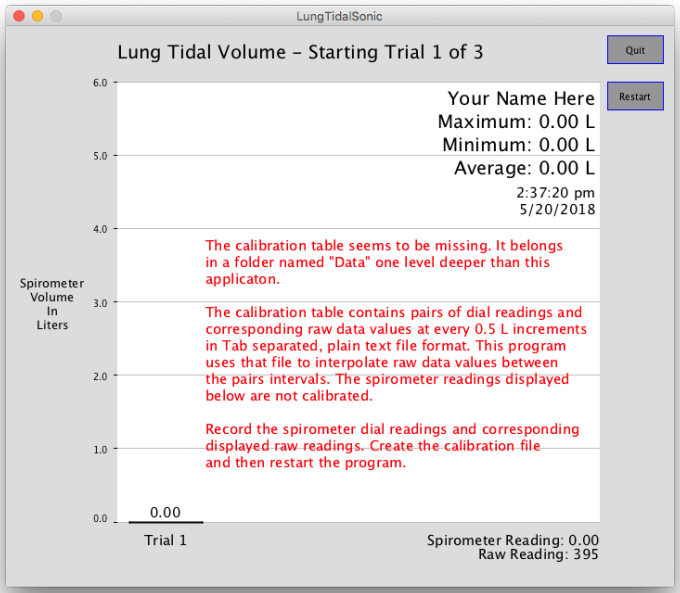
A simple reason for the error condition is that the USB cable for the computer to Arduino connection is not plugged in. If so then quit the LungTidalSonic application, plug in the cable and then run the LungTidalSonic application again.



If the "No USB driver ..." screen still occurs with the USB driver installed and the cable connected then more in-depth trouble shooting is required. You should see the green power LED lit on the Arduino board by peering into the Arduino box. Look into the slots on both sides of the box. The green LED means the Arduino is receiving power from the USB cable and that the cable may be good.

When running within the Processing IDE the LungTidalSonic application outputs messages to the console as it goes through its routines looking for the correct USB connection and the calibration log. You will be able to see what is happening and this will help in determining what to do next.

"THE CALIBRATION FILE IS MISSING ..."

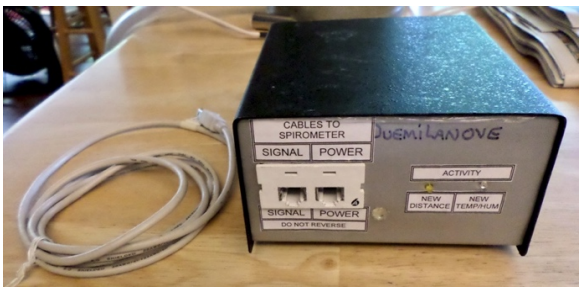


The "The calibration file is missing ..." screen results when LungTidal does not find the calibration_data.tsv file.

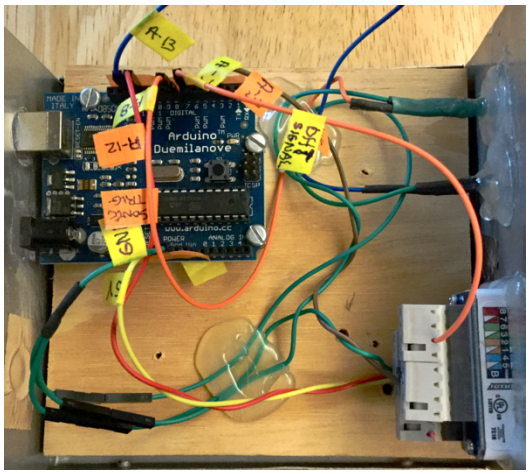
When running the LungTidalSonic ***exported application*** the calibration_data.tsv file is present within the \Java\data\ folder ***inside*** the LungTidalSonic application bundle. It is placed there automatically when LungTidalSonic is "Exported" with Processing. Therefore, the most likely reason for this error is making a mistake when editing the calibration_data.tsv file under the hood so to say.

LungTidalSonic still functions without the calibration file. While in this condition LungTidalSonic functions in a raw data mode so that you can use the raw date for manually creating the correct values for the calibration_data.tsv file.

Appendix - The Arduino Interface



The Arduino Duemilanove is mounted to a wood board that is in turn mounted within a steel project enclosure.



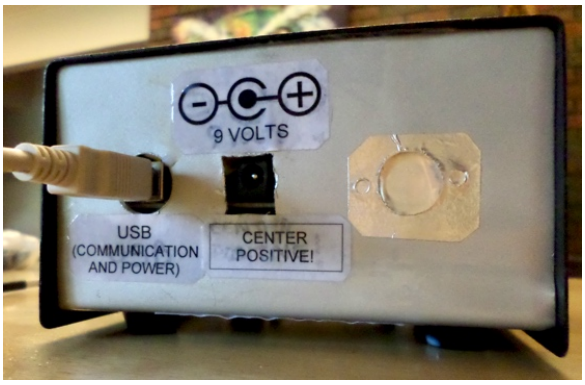
Wiring connections to the Arduino are with jumper wires. These are labeled so that they can be reseated if for some reason they disconnect. Make sure the Arduino is disconnected from power before attempting to reseat any wires.

Wire Color Inside the Box from Arduino to Phone Jacks

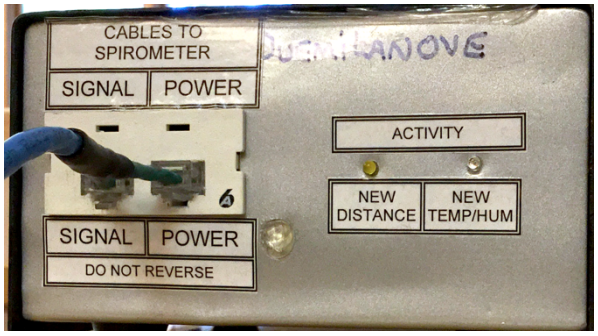
(This color scheme applies only to wires within the Arduino box.)

Purpose	Color	Arduino Location
5 volts	Yellow	Arduino 5v
Ground	Green	Arduino GND
DHT Signal	Orange	Arduino 7
Sonic Trigger (HC-SR04)	Red	Arduino 12
Sonic Echo (HC-SR04)	Brown	Arduino 3

The enclosure back has a USB connection jack and a center positive external power jack. External power is not required for the Arduino to read the spirometer. The USB cable connection powers the Arduino in addition to handling the serial data transport.



The enclosure front is where the cable to the spirometer connect. The setup uses two standard phone plugs labeled “Signal” and “Power”. The two jacks are ethernet jacks modified with epoxy so that only phone jacks can fit the jacks. The phone size only jacks avoid using an ethernet jack that someone might connect to a real ethernet device. Doing that might damage the Arduino and/or the ethernet device.



The spirometer connections to the Arduino are five wires. Two wires are 5 volts power and ground. One wire is the signal from the temperature/humidity sensor. The last two wires are for the sonic sensor’s trigger (TRIG) and echo (ECHO) signals.

A single ethernet jack has more than five wires. For safety reasons the LungTidalSonic’s cable scheme spreads the five required wires across two four wire phone jacks. The jack labeled “POWER” carries the 5V, TRIG, ECHO and GND wires. The TRIG and ECHO are wired in separate twists in the single ethernet cable used to run from the box up to the sensor housing. TRIG is twisted with the 5V wire and ECHO is twisted with the GND wire.

The jack labeled “SIGNAL” carries only the DHT sensor signal. If a single ethernet jack were used then someone might plug a live ethernet cable into the Arduino box or directly plug the ethernet cable into a computer or live ethernet jack. Doing so could easily fry the LungTidalSonic equipment.

Wire Color for the Cable from the LungTidalSonic Black Box to the Arduino Box

(This color scheme applies only to the ethernet cable wires.)

Purpose	Color	Plug Label	Ethernet Number
Ground	Blue	Power	3
Sonic Trigger (HC-SR04)	White-Blue Stripe	Power	4
Sonic Echo (HC-SR04)	White-Green Stripe	Power	5
5 Volts	Green	Power	6
Not used in this cable	Orange	Signal	3
Not used in this cable	White-Orange Stripe	Signal	4
Not used in this cable	White-Brown Stripe	Signal	5
DHT signal	Brown	Signal	6

Appendix - Other Computer Systems and Changing LungTidalSonic

Being a Java application created in Processing, LungTidalSonic can be made to run on computers running Unix and Windows operating systems. To do this one "exports" the desired system type application while having the LungTidalSonic "sketch" sources loaded in Processing. The Processing source files are the files ending with ".pde". The Processing development is available by download.

LungTidalSonic may be easily altered and this is done with the Processing development application.