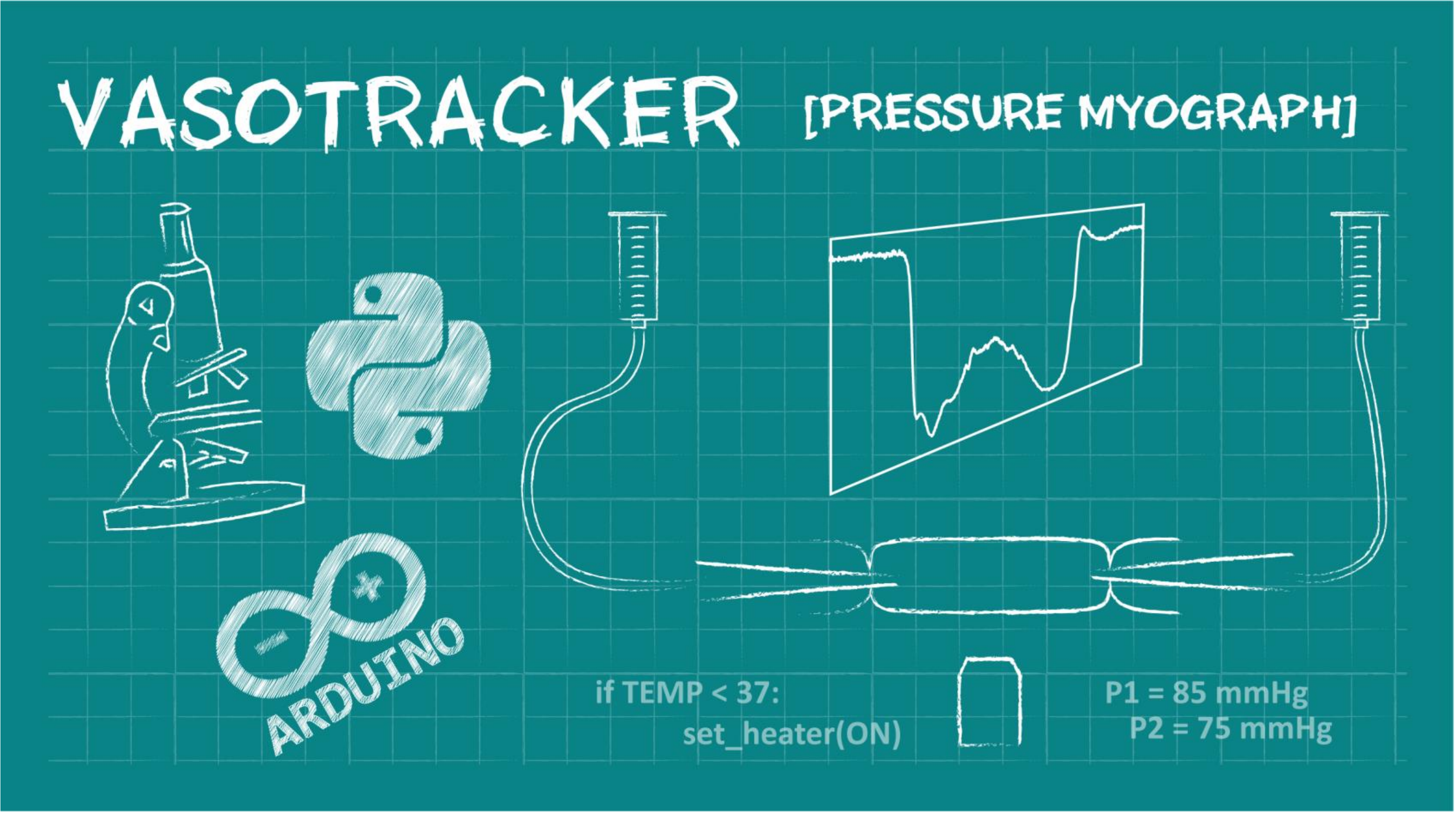

Pressure Monitor User Manual

06/07/2021

Developed by Calum Wilson at the University of Strathclyde and Nathan Tykocki at Michigan State University

Email vasotracker@gmail.com if you need help.

Table of Contents

Overview

The VasoTracker Pressure Monitor

The VasoTracker Pressure Monitor is an open-source, low-cost data acquisition system for monitoring two inline (flow-through) pressure sensors. The system is part of the VasoTracker pressure myograph system (www.vasotracker.com). It features an LCD for real-time visualization of intraluminal pressure as well as VasoTracker software integration for data recording.

General specifications

The main goal of the VasoTracker pressure monitor is to read the pressure on either side of a cannulated blood vessel that has been mounted in a pressure myograph. To do this, we use two in-line pressure transducers that each provide a small voltage signal that changes with pressure. To convert the voltage signal from each of the two inline pressure transducers into pressure (in mmHg) and to display the information to the user, we take advantage of the Arduino open-source microcontroller boards and associated stackable “shields”. The system can measure pressure in the range of 0 – 200 mmHg.

Citing VasoTracker

We hope that VasoTracker is of use to you or anyone else wishing to study blood vessel function. Our work developing and releasing VasoTracker as an open source platform was made possible by grant funding from the Wellcome Trust and British Heart Foundation. Our ability to support, develop and maintain VasoTracker will depend on more grants, so whether you use the VasoTracker software and hardware as a complete system or only certain bits, please cite VasoTracker in your scientific publications.

> Penelope F. Lawton, Matthew D. Lee, Christopher D. Saunter, John M. Girkin, John G. McCarron and Calum Wilson (2019). VasoTracker, a low-cost and open source pressure myograph system for vascular physiology. Frontiers in Physiology 10:99.
> DOI: https://doi.org/10.3389/fphys.2019.00099

Developers

Calum Wilson - University of Strathclyde:
Nathan Tykocki - Michigan State University:

Parts List

VasoTracker Pressure Monitor

Component	Supplier	Product #	Qty	Price	Total	Supplier Link
Arduino Uno**	Arduino	RB-Ard-34	1	£20.00	£20.00	RobotShop
Wheatstone bridge amplifier shield	RobotShop	RB-Onl-38	1	£19.01	£19.01	RobotShop
LCD Shield	RobotShop	RB-Cyt-73	1	£8.16	£8.16	RobotShop
Inline pressure transducers	Honeywell	26PCDFG5G	2	£47.20	£94.40	Mouser
12” 4-pin jumper wire	RobotShop	RB-Spa-1107	2	£1.41	£2.82	RobotShop
Arduino stackable headers	RobotShop	RB-Spa-928	1	£1.31	£1.31	RobotShop
Female to female jumper	RobotShop	RB-Gog-65	1	£3.90	£3.90	RobotShop
Heatshrink tubing	RobotShop	RB-Gog-47	1	£2.48	£2.48	RobotShop
3D printed Enclosure	VasoTracker	-	1	-	-	
Prices correct 06/07/2021		**Total**			**£151.68**	

The software and instructions provided only work for Arduino Uno boards with the ATMega microcontroller chip. Other variants (e.g., SAMD51 or ESP-32 chips) cannot be used.

Connections and Wiring

Pressure Sensor Wiring

The 26PCDFG5G pressure transducer has four pins. This version of the sensor comes with a handy cable. There are four wires:

- Pin 1 (Red wire): Vs
- Pin 2 (White wire): Signal +
- Pin 3 (Black wire): Ground
- Pin 4 (Green wire): Signal -

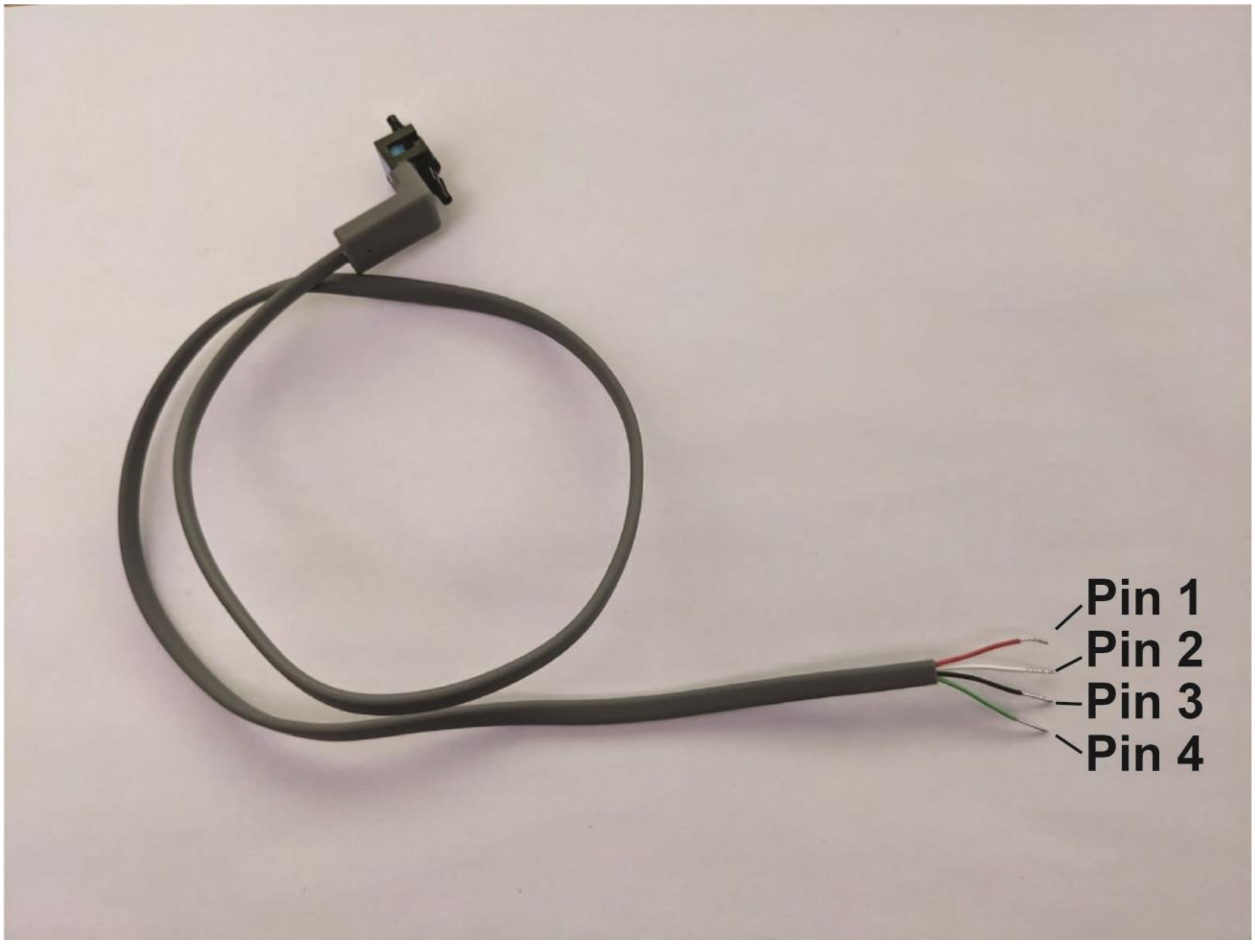

Figure 1 – Pressure sensor wiring

Wheatstone Bridge Shield connections

The following instructions are for use with the supplied Honeywell pressure transducer cable. There is a crossover within the cable header. If you make your own cable then the Pin order is different.

The Wheatstone shield uses the following Arduino pins:

- A0: Sensor 1 input
- A1: Sensor 2 input
- 5V, 3.3V, GND: to power the shield

Each pressure sensor needs to be connected to one of the 4-pin Molex connectors on the Wheatstone bridge shield. The Molex pins are as follow:

- Molex Pin 1: Vs
- Molex Pin 2: Signal +
- Molex Pin 3: Signal -
- Molex Pin 4: Ground

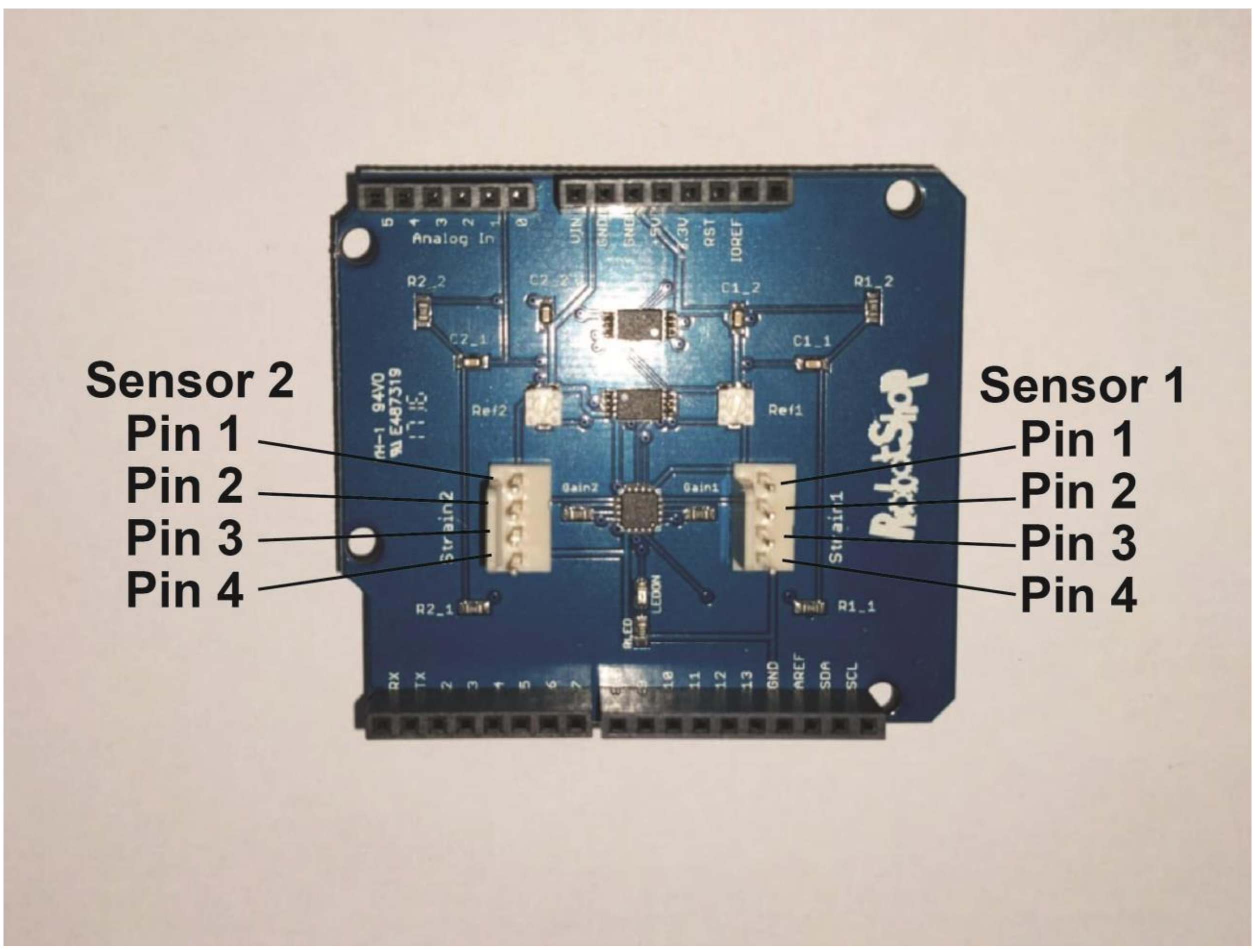

Figure 2 – Wheatstone bridge pinout.

Wheatstone Bridge Shield Setup

Each channel on the Wheatstone bridge shield has its own reference voltage that is set using specific channel potentiometers. Calibrating these is necessary to ensure that the shield works as intended.

Documentation for the Wheatstone Bridge Shield can be found on the RobotShop website.

Users need to erform the calibration procedure detailed in the “Test Procedure” pdf file (found here).

If you are too lazy to read this document, then please turn both potentiometers fully clockwise, then turn back 1/4 - 1/8 of a turn so that the potentiometer screw is parallel to the Arduino headers.

https://www.robotshop.com/en/strain-gauge-load-cell-amplifier-shield-2ch.html

The Wheatstone bridge shield contains two potentiometers

Building the VasoTracker Pressure Monitor

Mapping the pressure transducer connections to the Wheatstone bridge

The table below maps the pressure transducer connections to the Molex connector pins on the Wheatstone bridge shield:

Connection	Pressure sensor pin	Wheatstone bridge pin
Vss	Pin 1 (Red wire)	Molex Pin 1
Signal +	Pin 2 (White wire)	Molex Pin 2
Ground	Pin 3 (Black wire)	Molex Pin 4
Signal -	Pin 4 (Green wire)	Molex Pin 3

Table 1 - Mapping of pressure transducer connections to Wheatstone bridge shield

Step 1: Connect the 4-pin jumper wires to the Molex connectors on the Wheatstone bridge shield.

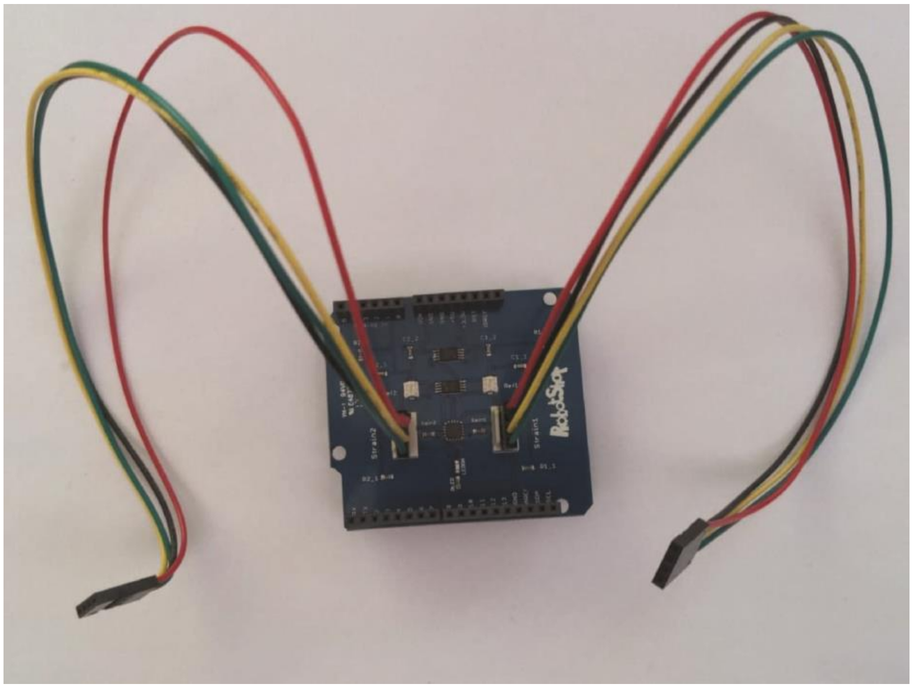

Figure 3 – Wheatstone bridge with jumper wires attached (step 1)

Step 2: Slide a piece of heatshrink tubing over the jumper wires

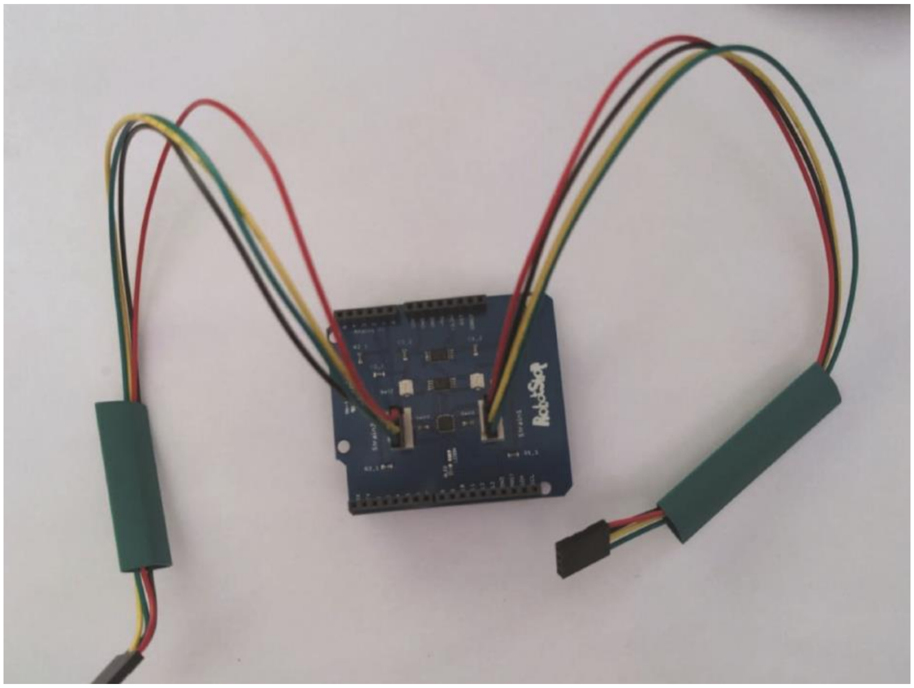

Figure 4 – Heat shrink tubing placement (step 2).

Step 3: Connect the pressure transducer cable to the jumper wire as per Table 1.
Step 4: Slide a piece of heat shrink tubing over the jumper wires. Heat gently (with heat gun or soldering iron) to shrink. Be careful not to burn the heat shrink tubing or damage the wiring insulation.

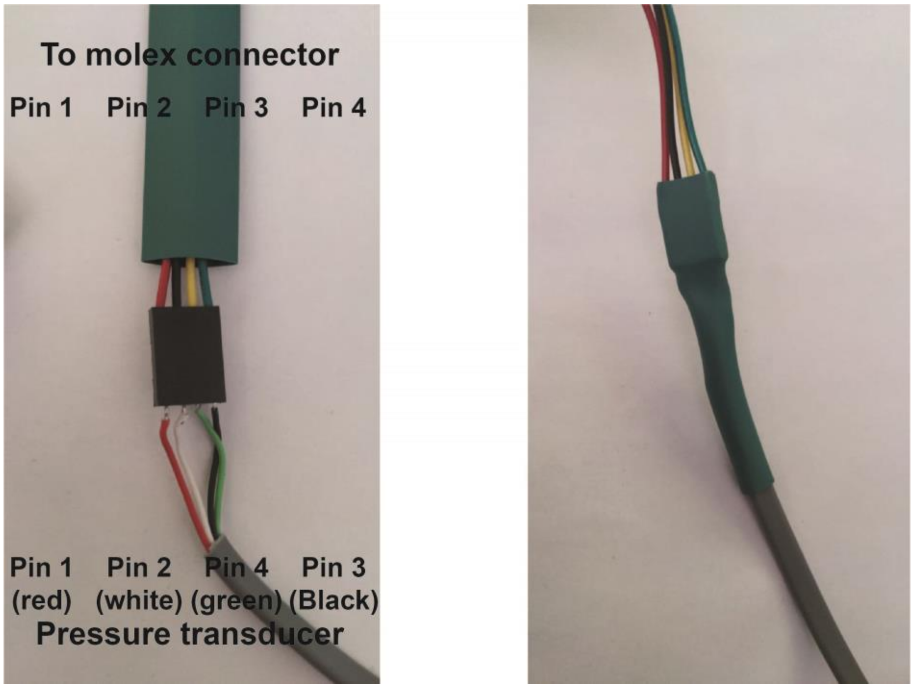

Figure 5 – Pressure sensor connections before (Step 3) and after (step 4) being secured with heat shrink tubing.

Stacking the Arduino Shields

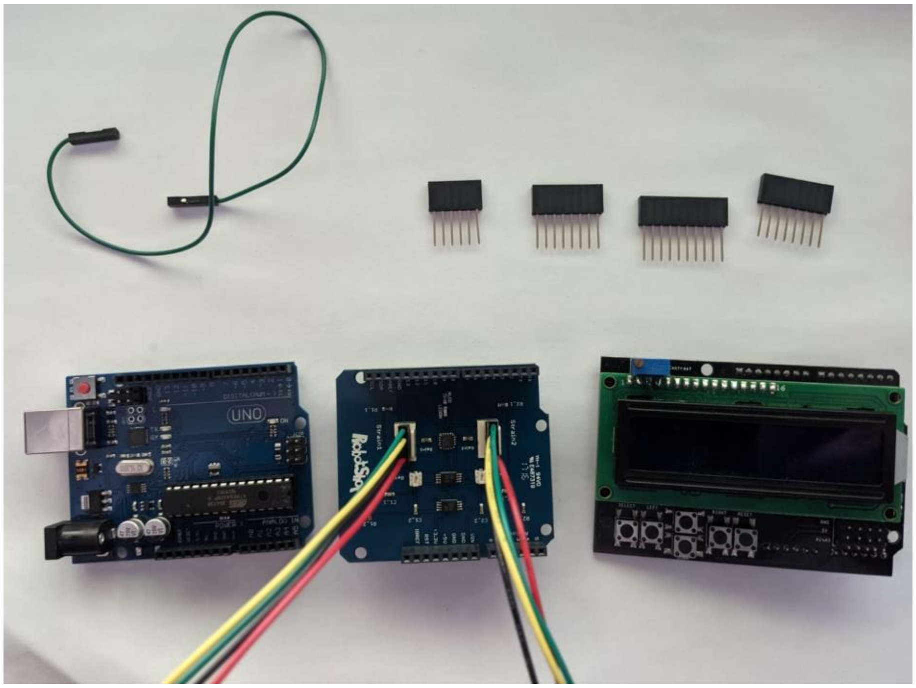

Figure 6 – VasoTracker pressure monitor components

Step 1: Stack the wheatstone bridge shield on top of the Arduino Uno.

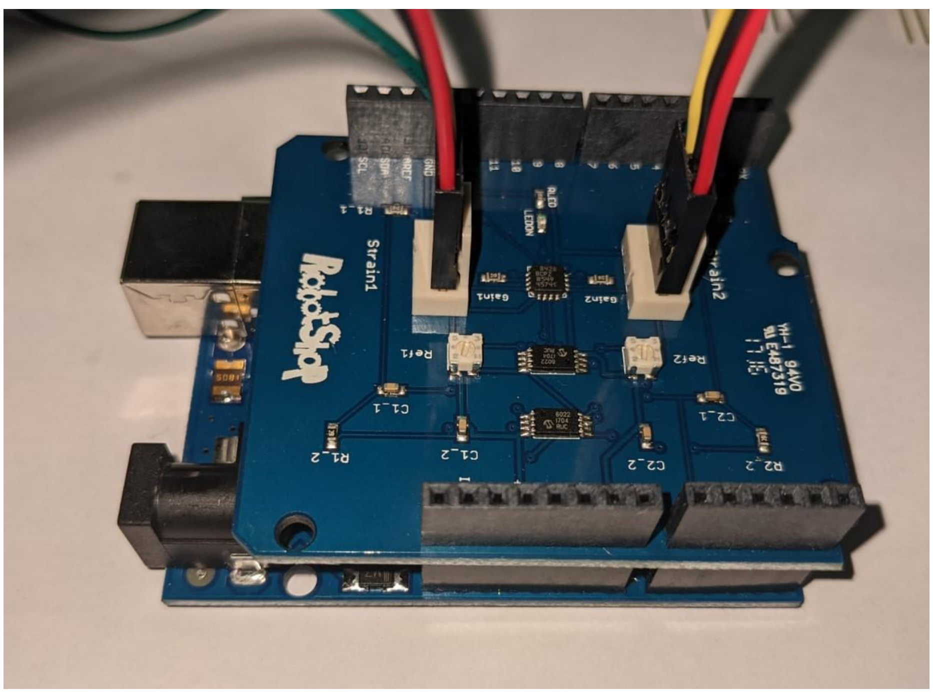

Figure 7 – Stacked Wheatstone bridge shield and Arduino Uno

Step 2: Place stackable headers on top of the Wheatstone Amplifier Shield (these are required to allow clearance for the pressure transducer connections). Bend the A0 pin on the stackable header so that it does not connect to the Arduino Uno board (see Figure 8).

The LCD shield normally uses the A0 pin for the button commands. However, the strain guage also uses the A0 (and A1) inputs on the Arduino. To get around this problem, we route the LCD A0 pin to A5.

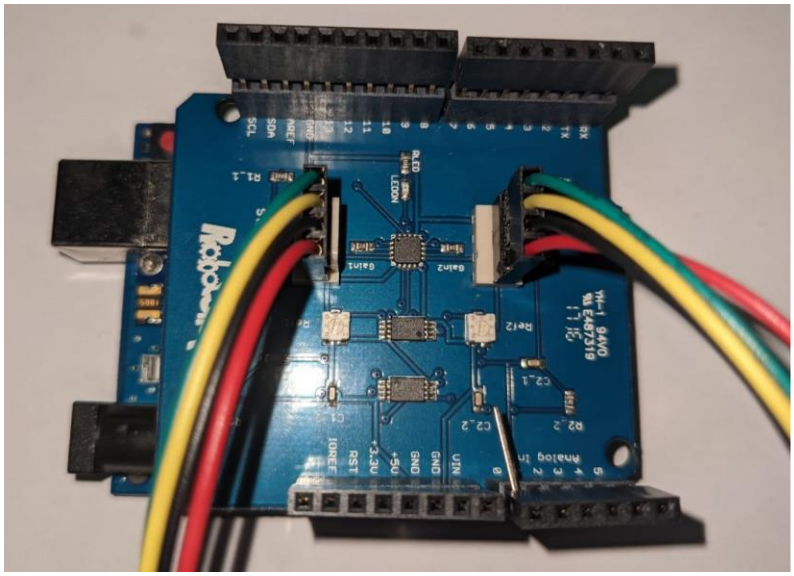

Figure 8 – Placement of stackable headers.

Step 2: Connect the jumper cable to the bent A0 pin.

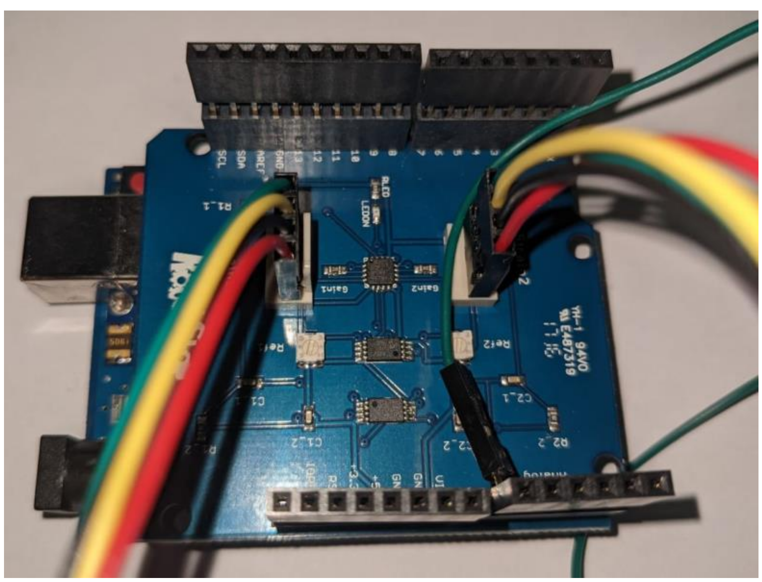

Figure 9 – Placement of LCD jumper cable.

Step 4: Stack the LCD shield on top of the headers. Route the jumper cable to the A5 pin.

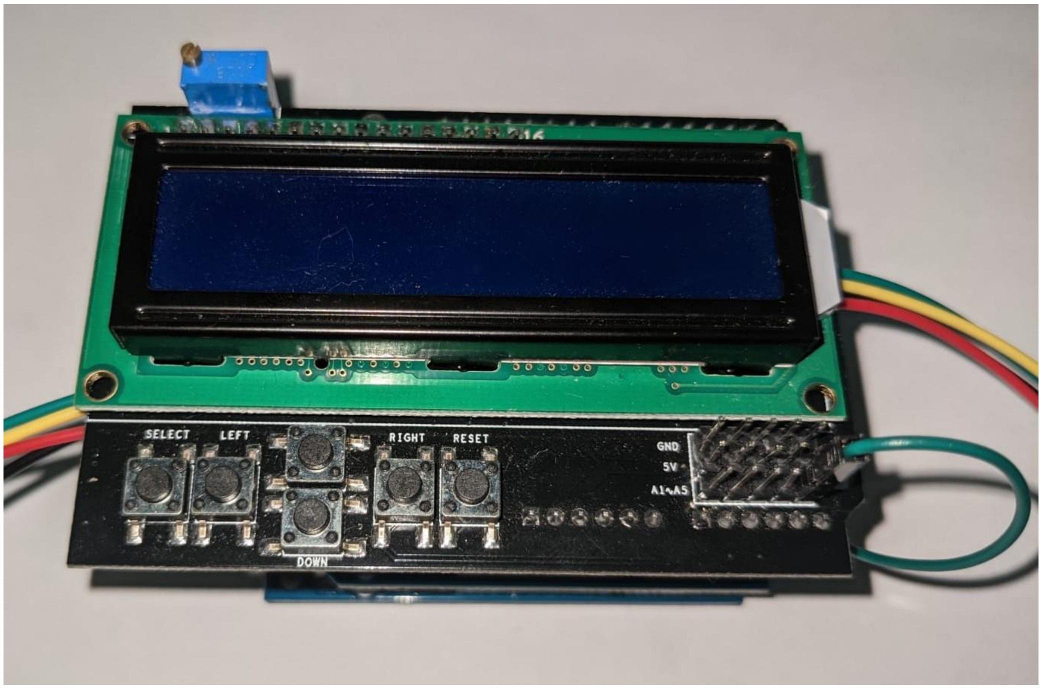

Figure 10 – Stacked LCD shield.

Alternative method for rerouting LCD pins: Remove the A0 pin from the LCD shield (heat it up with a soldering iron and pull it out using pliers). Solder a jumper wire from A0 to A5 on the board. This is a better way to do the above.

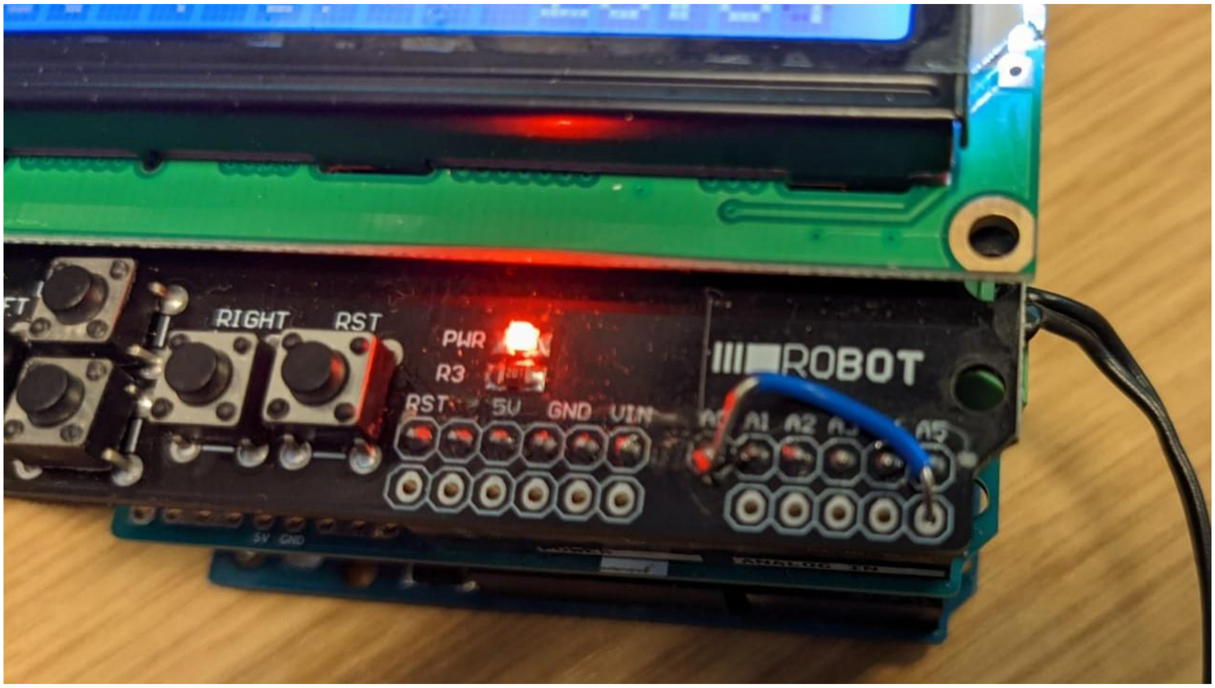

Figure 11 – Alternative way to reroute the pins on the LCD shield. You should do a neater job than we did on this one.

Step 4: If you want, 3D print an enclosure to protect your VasoTracker pressure monitor. The STL files for the enclosure below can be found on the VasoTracker Pressure Monitor webpage.

Figure 12 – VasoTracker Pressure Monitor

Programming the VasoTracker pressure monitor

The software and instructions provided only work for Arduino Uno boards with the ATMega microcontroller chip. Other variants (e.g., SAMD51 or ESP-32 chips) cannot be used.

Step 1: Download and install the Arduino IDE. This can be done using either Windows or Mac OSX; however, the VasoTracker software installer only runs in Windows.

If you want to run VasoTracker on iOS then you must install from the source code. Instructions for doing so can be found in the VasoTracker Software User Manual.

Step 2: Download the latest VasoTracker Pressure Monitor Arduino code.

Step 3: Unzip the “Pressure_Monitor_V*x*” folder, where ‘*x*’ represents the version number. You must keep all the files in this folder together, and the folder name must not be changed.

Step 4: Connect the Arduino to your computer using the Arduino USB cable.

Step 5: Within the “Pressure_Monitor_V*x*” folder, you will find three files: (1) “pressure.cpp”; (2) “pressure.h”; and (3) “VasoTracker_Pressure_Monitor_V*x*.ino”. Using the Arduino IDE software, open “VasoTracker_Pressure_Monitor_V*x*.ino”.

Step 6: Configure the Arduino IDE for your Arduino Uno. If you purchased an Arduino Uno clone board (e.g., Adafuit Metro, please see the instructions for configuring the board at the supplier’s website).

Tools → Board → Arduino Uno

Tools → Port → COM *X* (Arduino Uno). *“X” will vary by computer and system, depending on how many USB peripherals are attached.*

Step 7: Upload the code to the Arduino by clicking the upload button. No modifications to the codee are required if you are doing a standard build as outlined above.

Figure 12 – Arduino IDE with upload button highlighted

Figure 13 – A programmed pressure monitor.

Setting up the VasoTracker pressure monitor

On first use, the VasoTracker pressure monitor needs to be calibrated. Once this has been done, the calibration values are saved into memory on the Arduino Uno and can be reloaded each time the pressure monitor is used – even after disconnecting the power. We recommend that you recalibrate the sensors once a week. We use a Big Ben manometer, but any calibrated manometer will suffice, such as a blood pressure cuff (with the cuff removed).

Figure 14 – Big Ben manometer.

Step 1: Use your tubing and connectors of choice to connect the first transducer to the manometer.

Step 2: On the recalibrate screen (Figure 15), use the up/down buttons until the screen reads "YES" and press the "ENTER" button (left most on LCD screen).

Step 3: The next screen is used to set the low pressure used for calibration (Figure 16). Use the UP/DOWN (10 mmHg increments) or left/right buttons (1 mmHg increments) to set the low calibration pressure to 0 mmHg. Save this value by pressing "ENTER".

Step 4: The next screen shows the reading from the pressure sensor (Figure 17). Ensure the manometer is set to the pressure set in the previous step (0 mmHg). Press "ENTER" to save this value.

Step 5: The next screen is used to set the high pressure used for calibration (Figure 18). Use the UP/DOWN (10 mmHg increments) or left/right buttons (1 mmHg increments) to set the low calibration pressure to 160 mmHg. Save this value by pressing "ENTER".

Step 6: The next screen shows the reading from the pressure sensor (Figure 19). Ensure the manometer is set to the pressure set in the previous step (160 mmHg). Press "ENTER" to save this value.

Steps 7-10: Repeat Steps 3-6 for the second pressure transducer.

Step 11: Connect the pressure sensors to your pressure myograph system and use with VasoTracker. If used with VasoTracker pressure myograph software, the pressure values with be saved along with blood vessel diameter data.

Figure 15 – Recalibration selection screen.

Figure 16 – Low pressure calibration selection screen.

Figure 17 – Low pressure reading screen.

Figure 18 – High pressure calibration selection screen.

Figure 19 – High pressure reading screen.

Figure 20 – Calibrated VasoTracker pressure monitor.

www.vasotracker.com