Stovething

Prototype assembly instructions

Components & tools

The components included in the proof-of-concept system are listed in the document entitled *Stovething_prototype_components*.

In addition to the components, you'll need a soldering iron with a fine tip. I re uncomfortable soldering, perhaps you can borrow someone who can hel you.

You'll also need a PC or a laptop (but not a particularly powerful one).

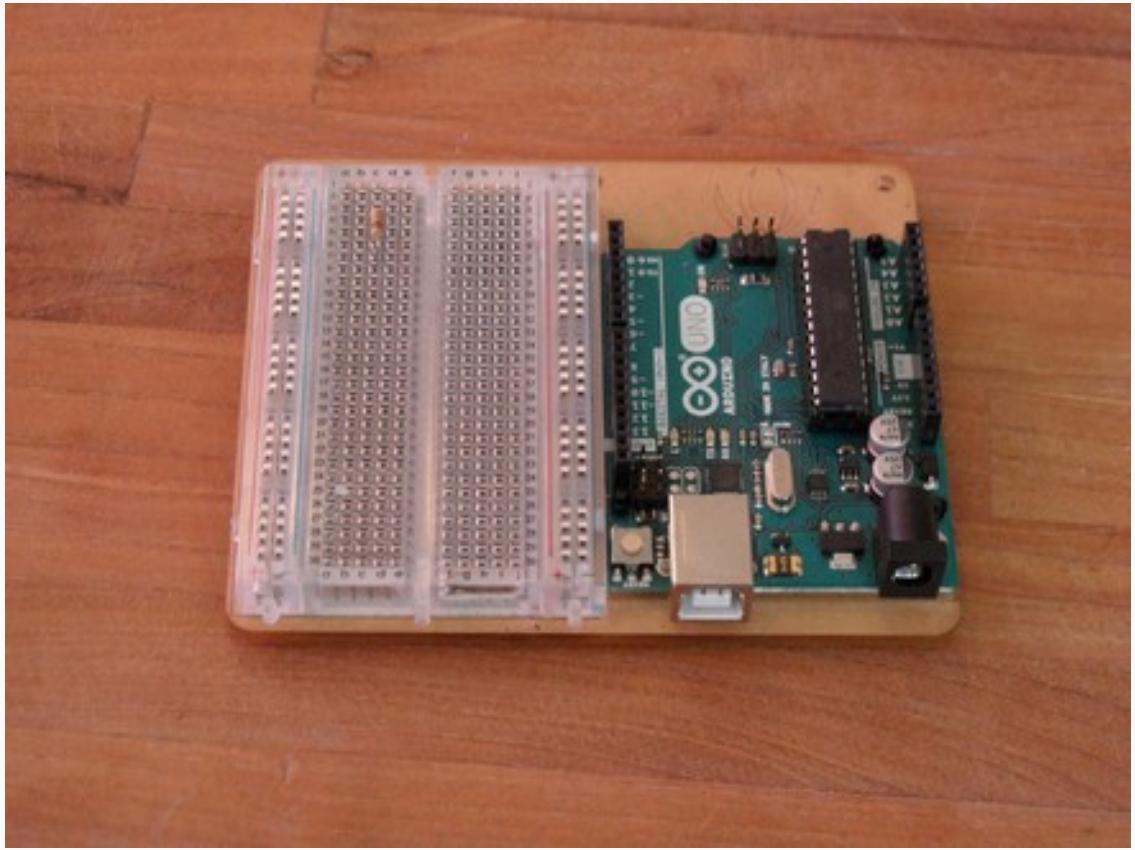
Setting up the Arduino

Step 1

(The following instructions are based on the Seeedstudio ARDX kit. Most kits contain more or less the same components, so you should be OK.)

Unpack the Arduino Uno R3 and the breadboard, and fix both to the base plate (assuming your kit comes with one - please check).

The kit includes a USB cable, which is not shown here.

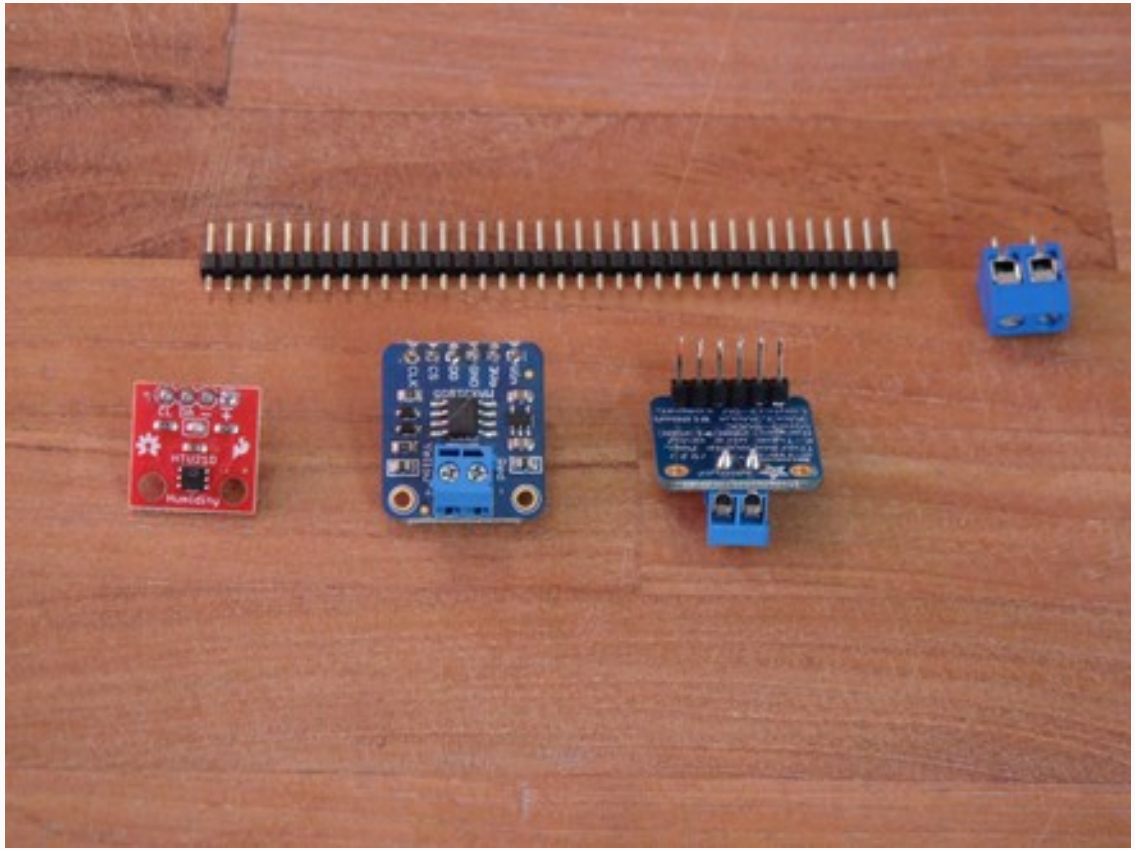


Step 2

Unpack the Adafruit MAX31855 thermocouple and the SparkFun temperature and humidity sensor.

Solder the header strip and the terminal block to the Adafruit board. Instructions are available here: https://learn.adafruit.com/adafruit-1-wire-thermocouple-amplifier-max31850k/assembly

Next, solder the header strip to the SparkFun sensor (which does not have a terminal block).



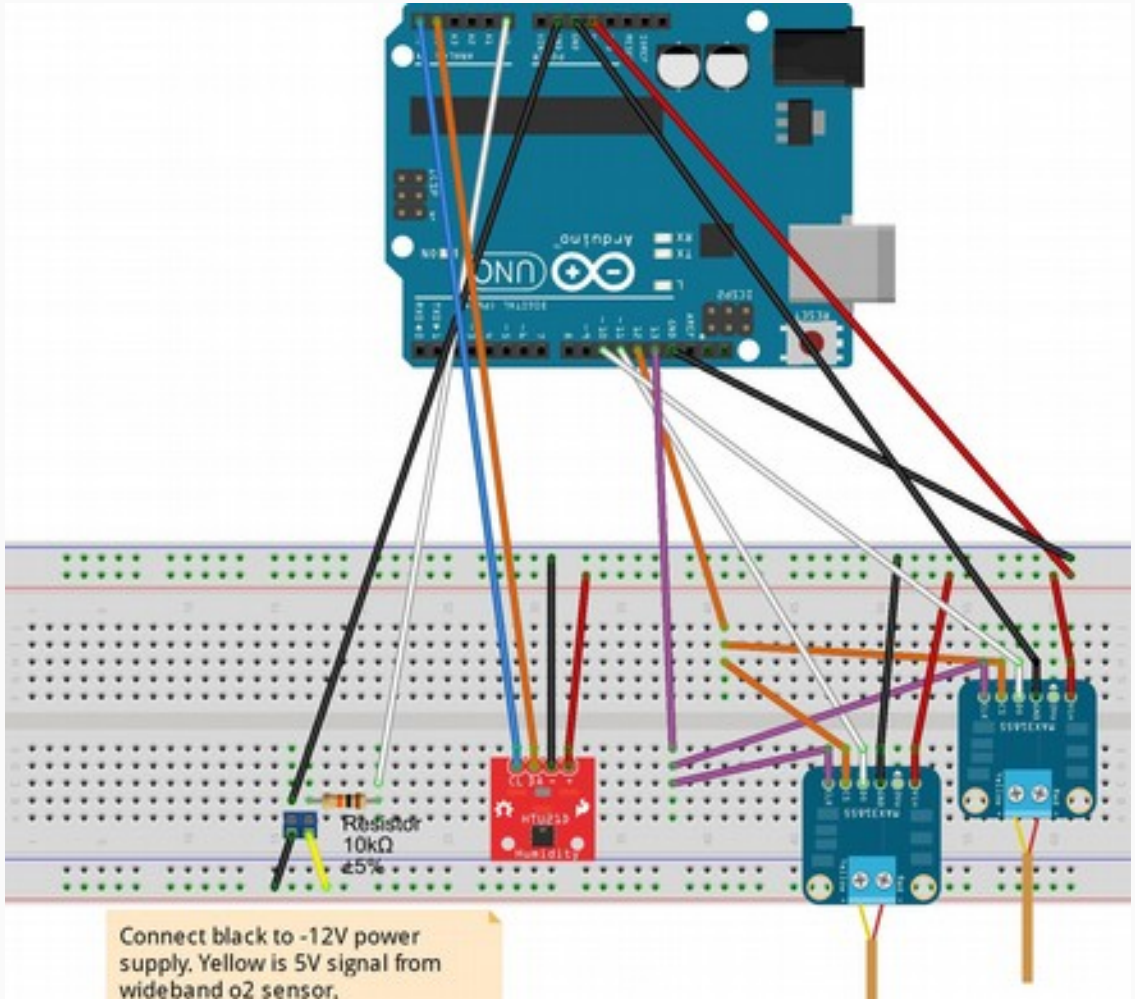
Step 3

Connect the Adafruit and Sparkfun sensors to the Arduino as shown on the wiring diagram (push the pins into the breadboard). The diagram is included in the following PDF file:

Stovething_wiring_diagram.pdf

The diagram was created in Fritzing. The original Fritzing file is also included, should you prefer to download the software and work with the actual file (allowing you to zoom in).

www.fritzing.org

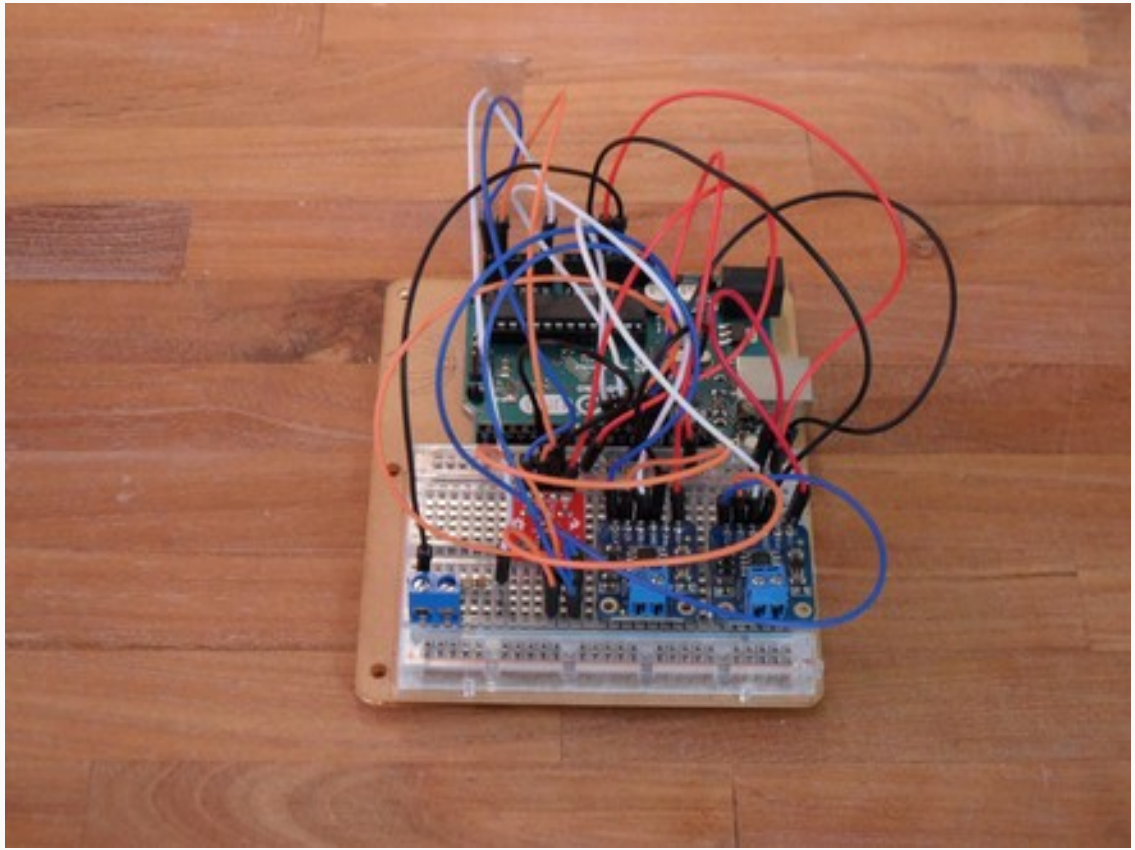


First bit done

If your Arduino looks like a coloured hedgehog, you've probably got it right.

Double check the connections, just to be sure.

If you haven't played with Arduino before, spend some time with it. The software is a bit off-putting, but there's lots on the internet to help out.



Installing the software

Step 1

Download and install the Arduino IDE (https://www.arduino.cc/en/Main/Software).

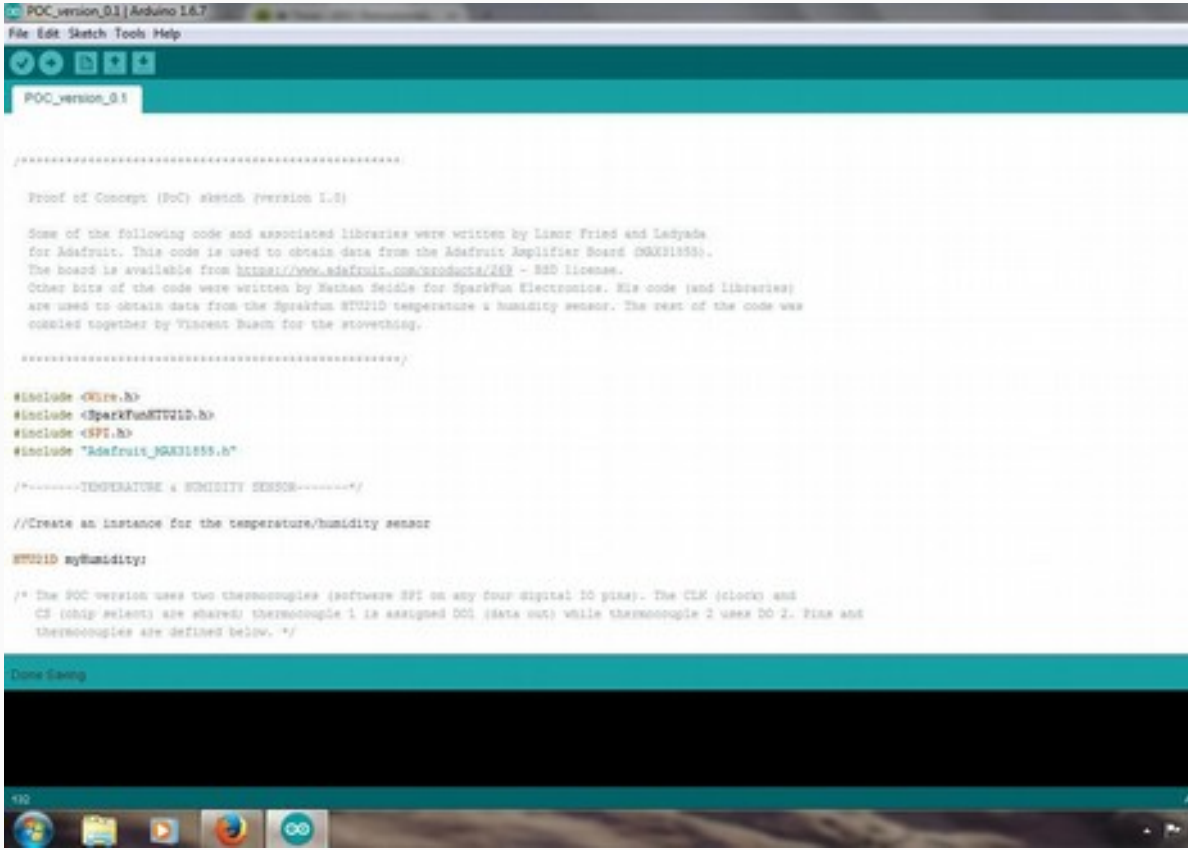
I use genuine Arduino products, so I feel less of an urge to donate, but please contribute to the project in some form.

Step 2

Start the Arduino software and open Stovething_prototype_sketch.ino from wherever you have saved it. I've added some comments in the code in an attempt to explain it.

Connect the Arduino to your computer using the USB cable. Make sure you're using the right port (Tools/Port). If you're not sure which port the Arduino is on, check via the Device Manager on your PC/laptop - under Ports (COM & LPT).

Once connected, upload the software (sketch) to the Arduino (the button with the arrow pointing to the right).



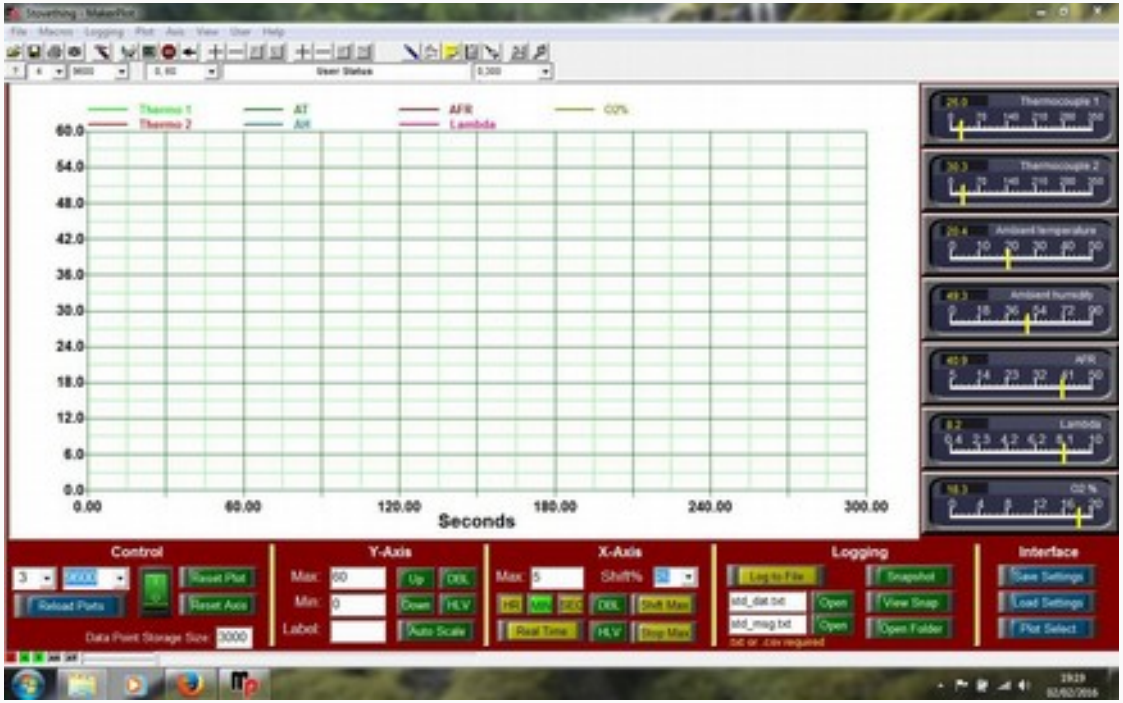
Step 3

The MakerPlot interface may not be the sexiest (especially compared with web-based solutions like freeboard.io) but the explanatory videos are excellent.

Once installed, go to Macros / Select Start-Up Macro and load the macro entitled Stovething_MakerPlot_interface.spm from wherever you have saved it.

The interface shown on the right should now automatically load each time you launch MakerPlot.

Excellent videos and explanations at http://www.makerplot.com/



Step 4

Download LogWorks 3 from Innovate Motorsports at:

http://www.innovatemotorsports.com/support.php

Logworks 3 includes LM Programme which is used to program the senso controller.

Connecting the sensors

Step 1

The thermocouples connect to the screw terminals on the Adafruit boards. Make sure you connect them the right way around, and note that you cannot use earthed thermocouples.

The thermocouples shown here are cheap (and may even be nasty). You can use any K type thermocouple.

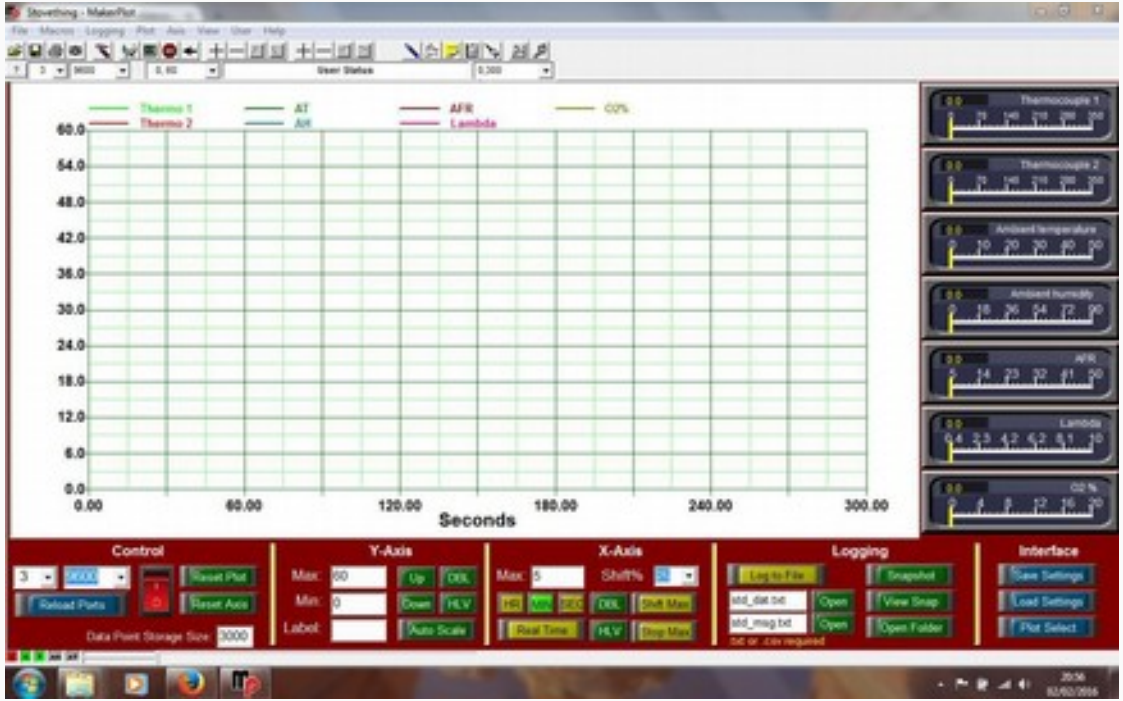
Test drive

Once the thermocouples are attached, connect the Arduino to your computer. Assuming the software sketch has been uploaded, launch MakerPlot, select Reload Ports and flick the Control switch.

You should, after a few seconds, see data appear for thermocouple 1, thermocouple 2, the ambient temperature and ambient humidity.

Heat the thermocouples with a lighter to see how the gauges and graphs respond.

Have a play with the buttons.



Step 2

The wideband oxygen sensor comes in a kit that contains a Bosch LSU 4.9 sensor, a controller (the rectangular box), a cable to connect the sensor to the controller (on the right of the manual) and a serial cable to connect the controlled to a PC/laptop (to the right of the sensor).

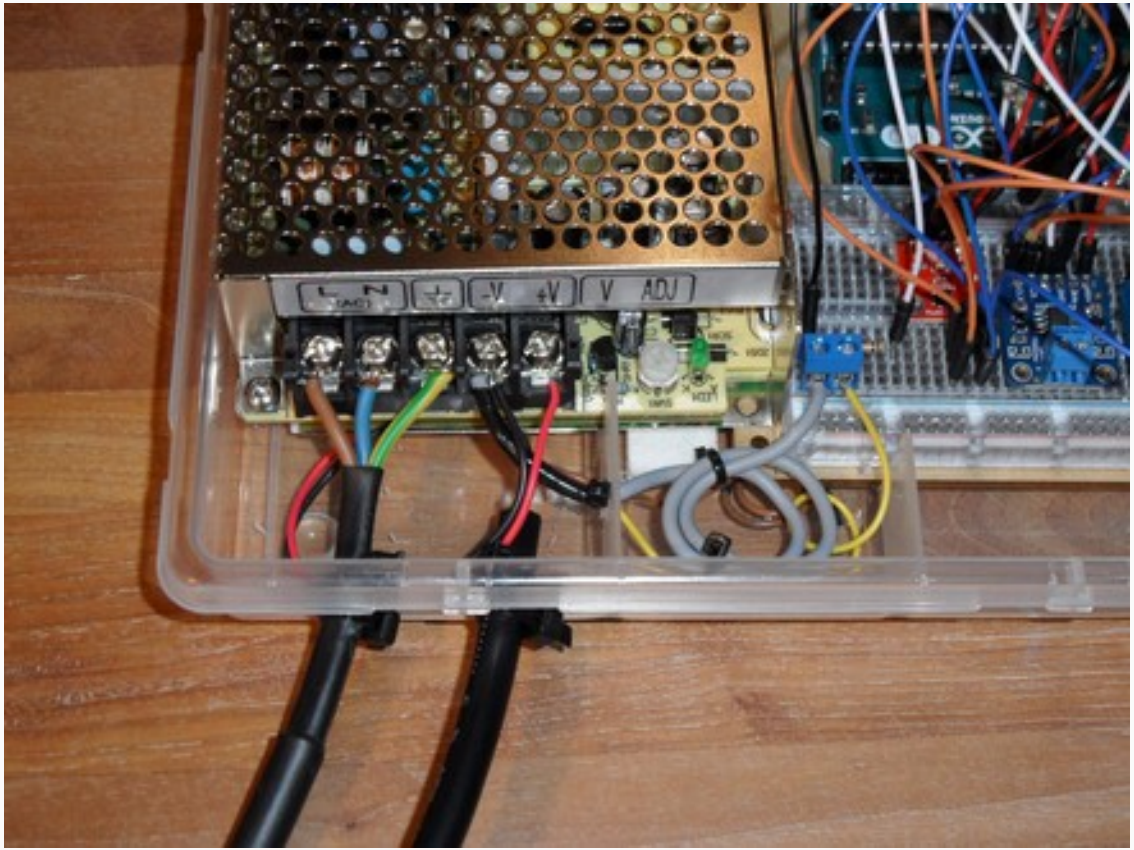
If your PC/laptop does not have a serial port, you'll need to steal a serial to USB converter (on top of the manual).

The manual explains how the sensor should be fitted, calibrated and programmed. Sounds complicated, is quite straightforward.

Step 3

Four coloured wires emerge from the controller (the cable on the right in the photo). Connect the red to +12 volts on the power supply, and the black to -12 volts. Tape off the brown wire, which is not used. The yellow wire should be connected to the screw terminal in keeping with the wiring diagram.

Run a cable between the -12 volts on the power supply and the ground on the screw terminal (see wiring diagram). The grey cable on the photo has a black end, which is a little confusing.



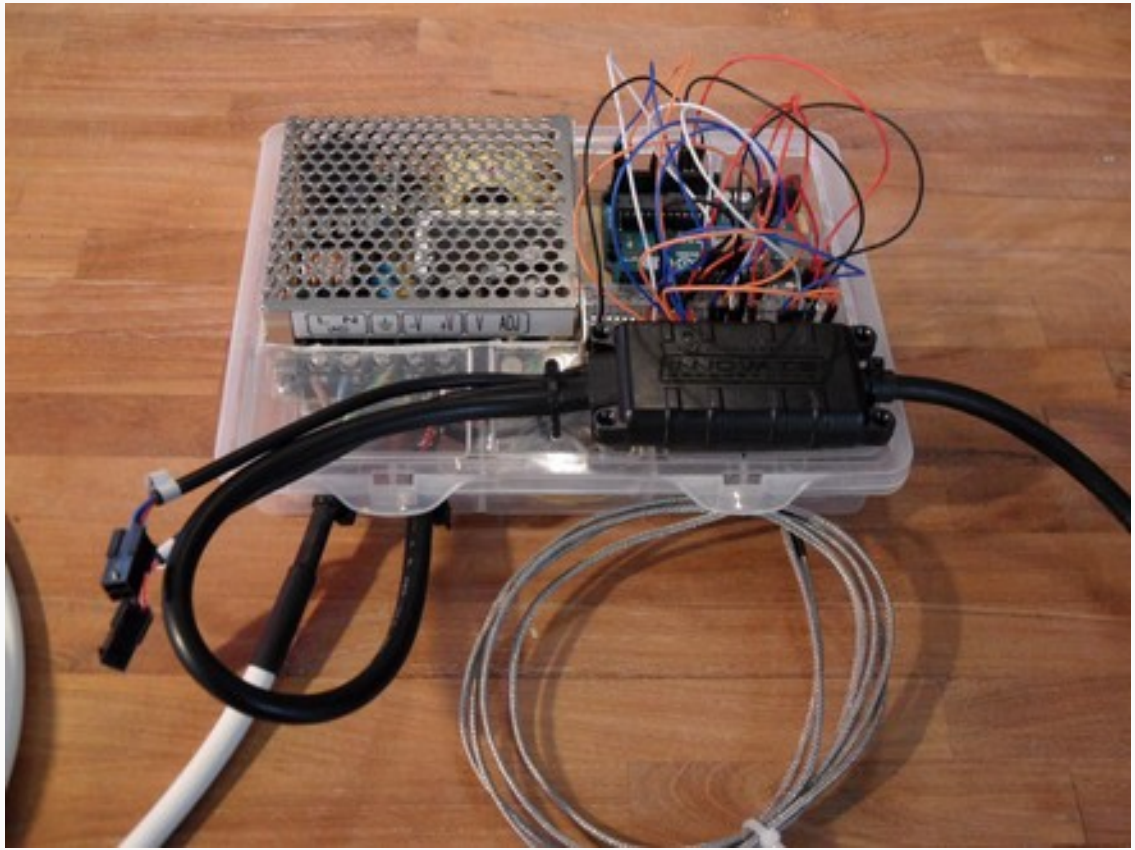
Casing

The cables got a bit much after a while so I bought a 70 cent plastic container to house the board, the power supply, and the controller.

The USB cable connects via a hole in the side, and the thermocouples emerge from the front.

The white cable is connected to (earthed) mains, the black goes to the sensor controller.

We’re nearly there...



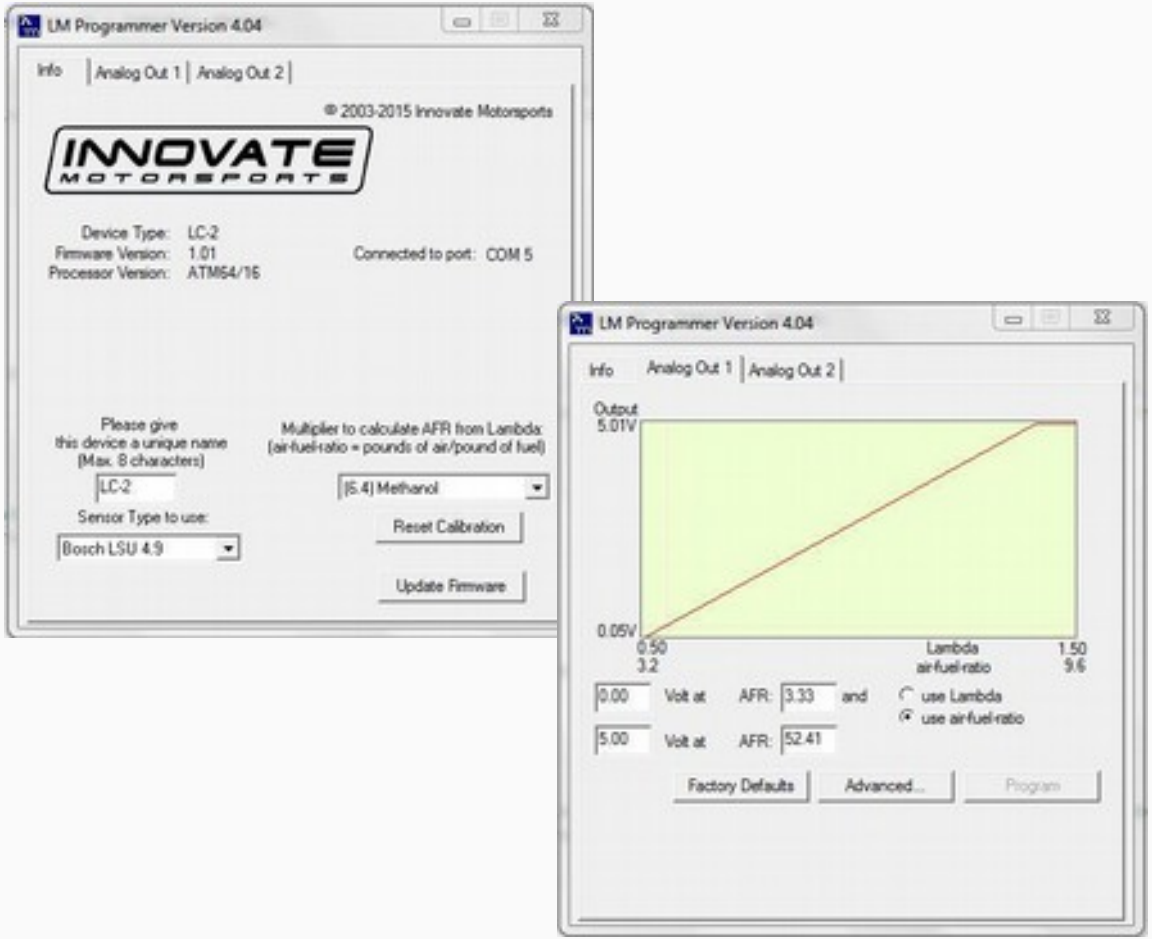
Step 4a

To program the controller, connect the serial cable to the OUT port/cable on the controller. Connect the other end to your computer, if need be via a serial to USB converter.

Power up the controller with the sensor disconnected, and launch the LM controller software.

Set the AFR to [6.4] Methanol, and select the Analog Out 1 tab. Click the ‘use air-fuel-ratio’ radio button and enter 3.33 for 0.00 volts and 52.41 for 5.00 volts.

Click yes if the program asks you whether you wish to load the new settings.



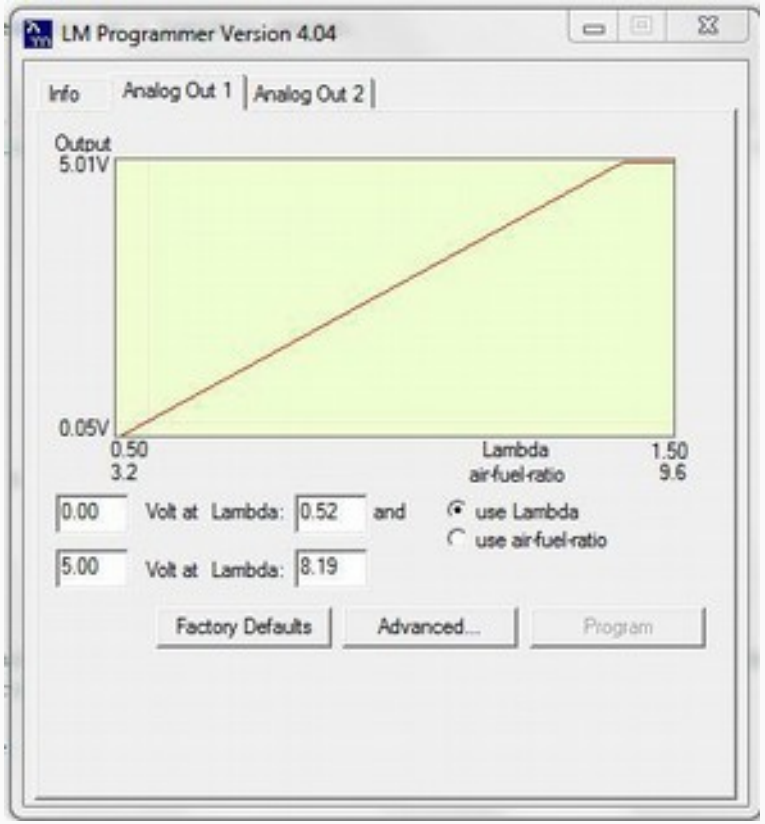
Step 4b

Click the ‘use Lambda’ radio button and enter 0.52 for 0.00 volts and 8.19 for 5.00 volts.

Next, select the Advanced button and set the timing to 6 seconds.

Again, click yes if the program asks you whether you want to implement the changes.

The controller is now programmed.



Step 5

Next step is to weld the bung, which is also included in the sensor kit, to the flue. I asked someone else to do this as I would have destroyed the flue completely. Needs a spray, but doesn’t leak any air. To minimise sensor contamination, I use an M18 1.5mm bolt while the stove is warming up (and whenever the system is not in use).

Step 6

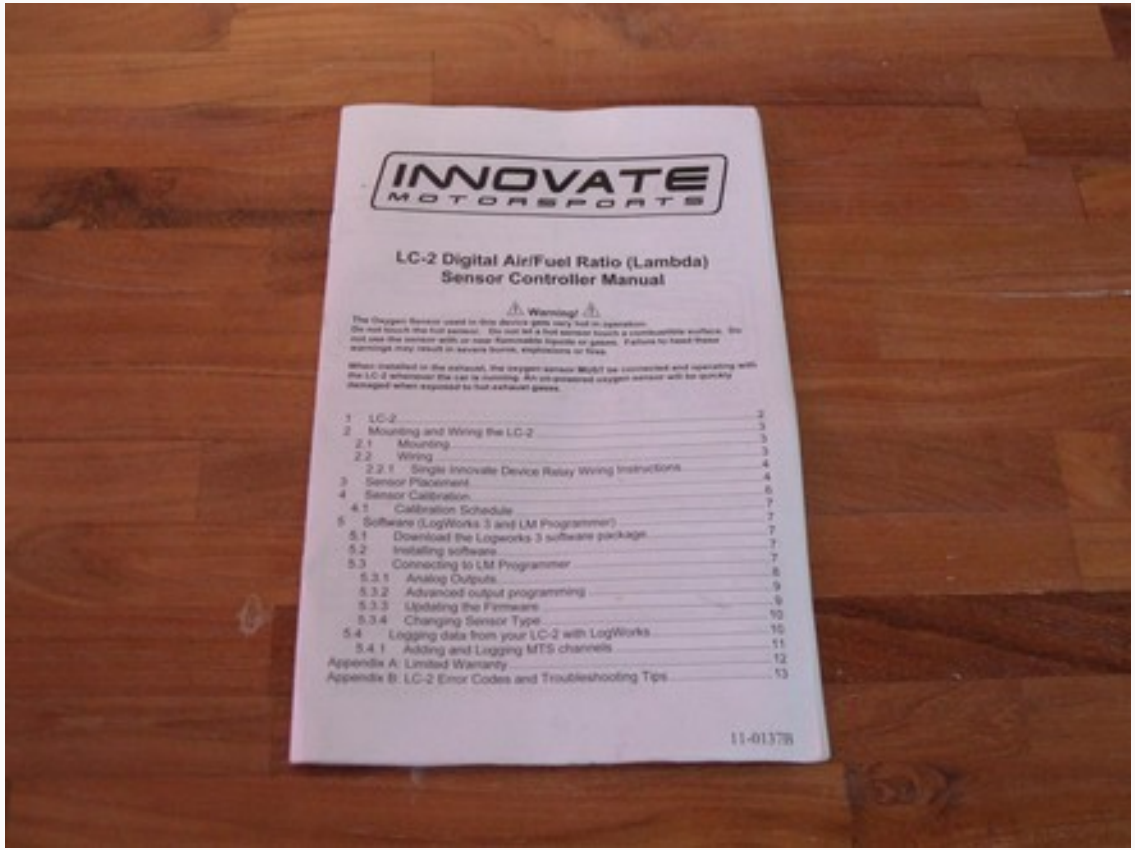
Last thing to do is to calibrate the sensor in accordance with the instructions set out in the manual. Again, very straightforward.

Once calibrated, power down the controller, fit the sensor to the flue, and start the controller up again.

The probe now outputs a signal to the Arduino. Start MakerPlot, and pray.

If you get stuck and can’t work it out yourself, I’d be happy to try and help out. Good luck!

buschbusch@xs4all.nl



Bugs & background

Wideband oxygen sensor

The system uses a programmable, self-contained and self-calibrating probe and controller that output an analog signal between 0 and 5 volts. The ability to program the sensor is important. Allow me to explain. Every fuel type has what is known as a stoichiometric AFR. This value denotes the air to fuel ratio at which all the fuel present in the (air/fuel) mixture is burnt. Petrol has a stoichiometric AFR of roughly 14.7, which means you need 14.7 grams of air to completely burn 1 gram of fuel.

Stoichiometric combustion occurs at lambda 1. Wood allegedly has an AFR of approximately 6.3. A wideband oxygen sensor set up for a car (read: petrol) might typically be programmed to monitor an lambda range between 0.5 and 2. Under this configuration, it wouldn't register a woodstove, which mostly burns richer than lambda 2. In other words, to monitor a woodstove, the sensor needs to monitor a much wider lambda range (between 0.52 and 8.19 on stovething).

Bugs

The signal from the oxygen sensor outputs to analog port A0 on the Arduino. When the oxygen sensor is switched on, and sending a signal to port A0, all is well. However, once the sensor is switched off, A0 will start to float after a while (sometimes 30 seconds, sometimes longer). This means the readings for AFR, lambda and O2% are no longer accurate when the oxygen sensor is switched off. Although this is what you might expect, it is something I still need to resolve.