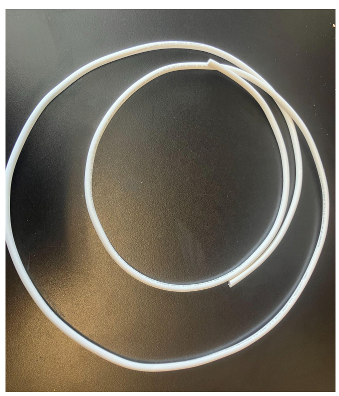
Building J-Sensors

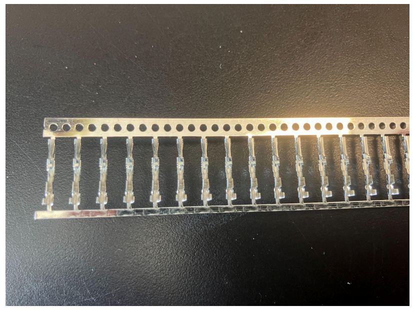
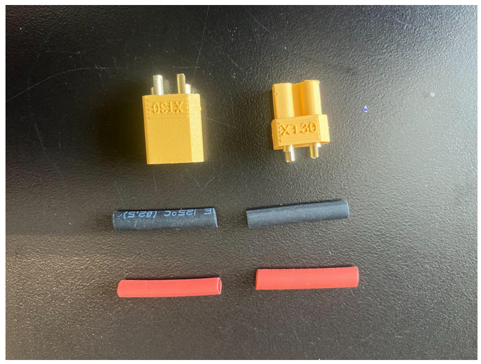
Links

- D. Bastviken original paper building these sensors: https://bg.copernicus.org/articles/17/3659/2020/
- Jonas's recent paper using these sensors: https://www.sciencedirect.com/science/article/pii/S0048969723015115?via%3Dihub
- Jonas's GitHub with information about how to build sensors: https://github.com/JonasStage/Methane-and-CO2-sensor
- Parts to Order: Parts List - JSensors

1. Assembling Power Cables

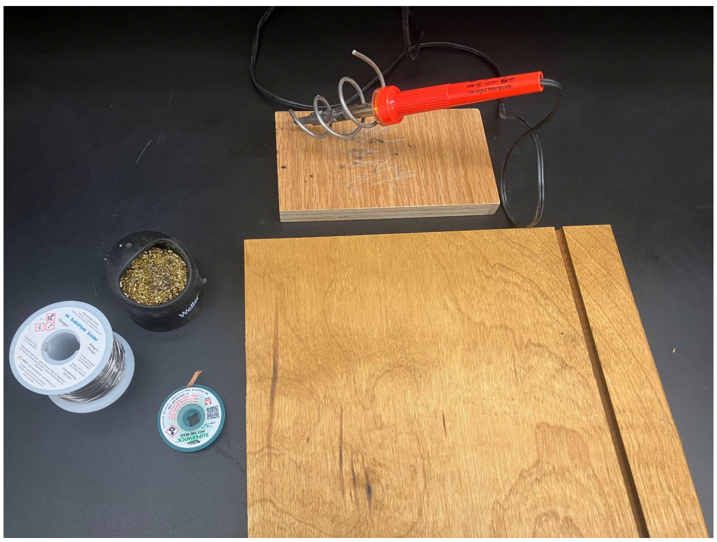
Parts:

Part Name	Picture	Description
Multi Conductor cable		This is the big white cable that come cut into 1 m strands
Housing Receptacle		These connect the power cable to the sensor

Crimp Tin Connection Socket		These attached to the ends of the multicable to connect to the housing receptacle
XT30 connectors		These connect the multicable of the power cable to the multicable of the battery cable (female on the left; male on the right)
Battery cable lugs		These connect the battery cable to the battery

You will also need:

1. For stripping wires and attaching crimp sockets:
    a. Wire cutters
    b. Scalpel
    c. Cutting board
2. For soldering
    a. Soldering iron (with thin tips, something like [this](#))
    b. Solder wire
    c. Desolder “tape”(to fix mistakes)
    d. Cutting board

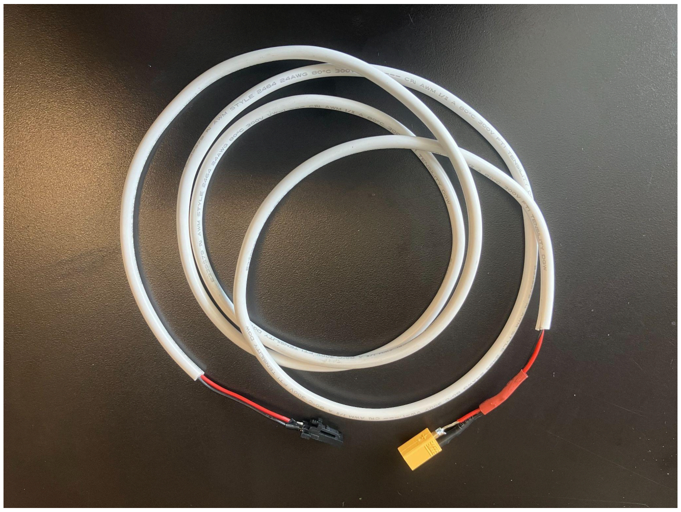


Assemble Power Cables

This cable is one of the two cables used to connect the sensor to the battery, and this cord plugs directly into the sensor. You will need at least one power cable for each sensor, but it is good to have backups since this cable is delicate. On each power cable, one end is set up to connect to the sensor (sensor end) using a housing receptacle (black T shaped), and the other end of the cable is set up to connect to the battery cable (battery end) using a XT30 connector (yellow).

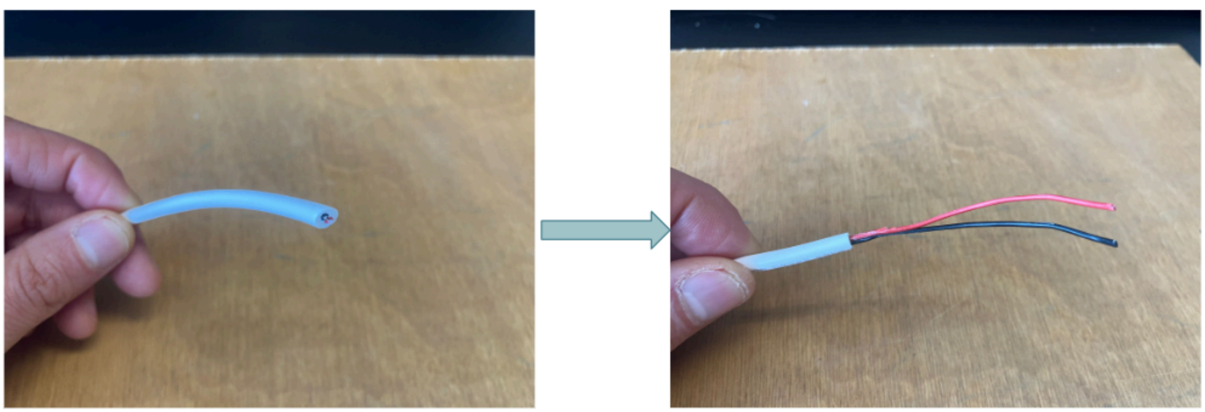
The multi cable should arrive cut into sections that are about a meter and a half long. If not, use the wire cutters to cut sections (1.25 - 1.5 meters) of the multi cable. Each section will be one power cable.

A completed power cable looks like this:

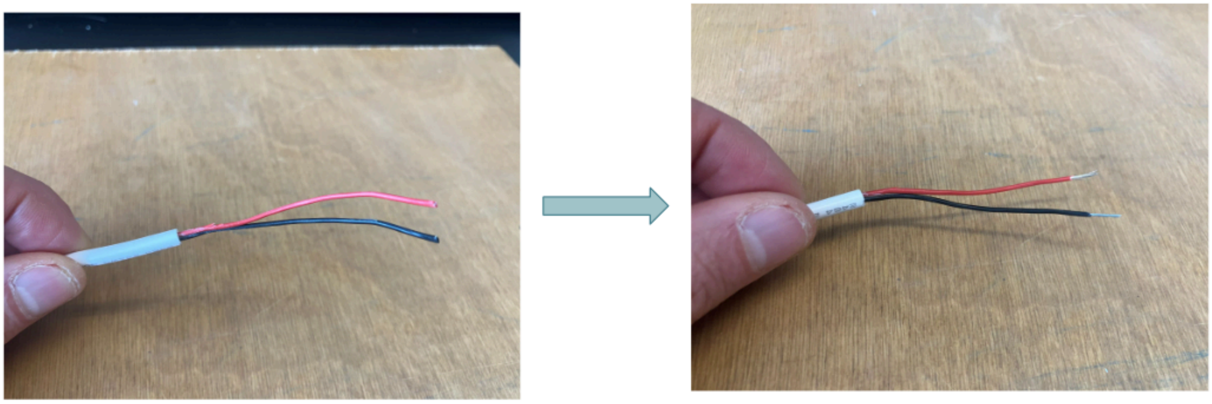


Preparing the Sensor End

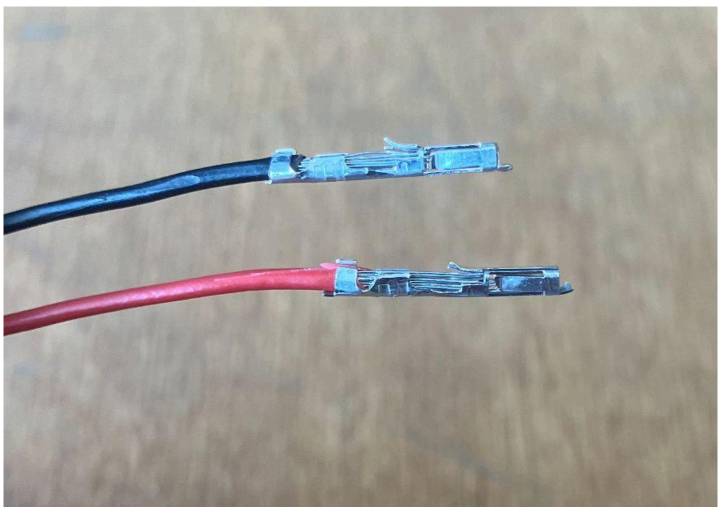
1. At one end of the power cable, strip (remove the white coating from) the end so that you have 5-7 cm of exposed black and red wire.
    a. Be careful not to damage the black and red plastic casing on the cables inside. If you do, cover the exposed wire using electrical tape (Alternatively: cut until the point of the damaged part and start over)
    b. Make sure the the black and red cables are the same length so that they don’t cause additional strain



2. At the end of the black cable and the end of the red cable, strip (remove the red or black plastic) to expose about 1 cm of the metal wire



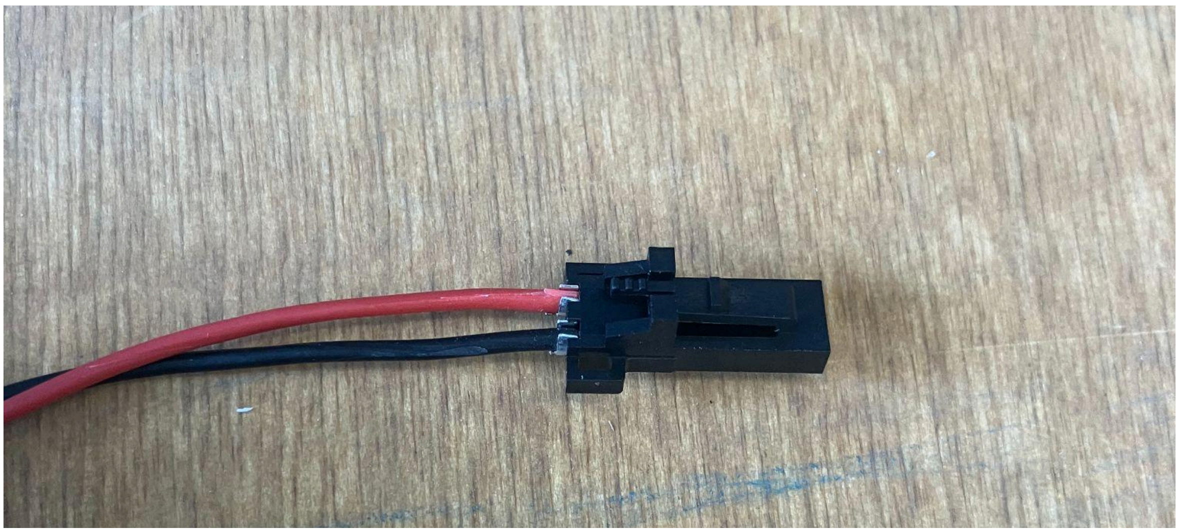
3. Crip one tin connection socket on the end of the red cable using the pliers to squeeze it in place. Repeat the same process with another socket on the end of the black cable.
    a. only squeeze the part at the end of the connection socket & the part in the middle



4. Insert the connection sockets (the silver part) into the black housing receptacle.
    a. NOTE: We have two different Versions of J sensors in the lab (Version 2 and Version 3) because Jonas updated the code and the circuit board and changed the temp/humidity sensor in the last year since he was hear. The Version 2 sensors are the ones that he made while he was here last spring (2023) and they have a blue PCB (top layer). The version 3 sensors are the ones that we have made in the last few weeks (2024) and then have a green PCB (top layer). The power cables for these two versions are NOT compatible. We have power cables that work for each version, and they are color coded with labeling tape: power cords with blue tape work with the Version 2 Sensors with a Blue PCB; and power cords with green tape work with the Version 3 Sensors with green PCBs. The reason that the cords are not interchangeable is because the polarity of one of the connection points is reversed on the newer version (the new orientation makes more sense).
    b. If the receptacle is sitting flat the “top” of the crimp connection socket (the side that you crimped closed) should be pointed up

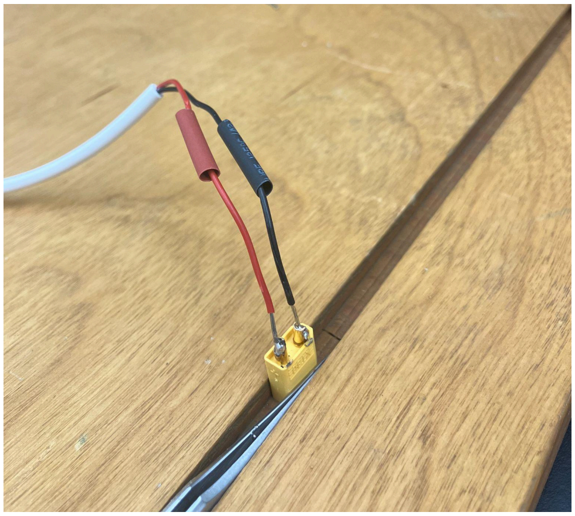
i. For **BLUE** Power Cables: The red cable should be going into the receptacle on left side and the black cable should be going into the receptacle on the right
    1. Black cable goes into the side with “flash” triangle denotation on the bottom of the black receptacle

ii. For **GREEN** power cables: The red cable should be going into the receptacle on the right side and the black cable should be on the left
    1. Red cable goes into the side with “flash” triangle denotation on the bottom of the black receptacle



Preparing the Battery End

1. On the other end of the power cable strip (remove the white coating) the cable to expose 8-10 cm of the red and black cable (See Step 1 of the Sensor End directions for pictures and additional notes)
    a. *Note*: for this end of the cable you want the exposed red and black cables to be slightly longer
2. At the end of the red and black cables strip the colored plastic to expose 1-2 cm of the metal wire beneath. (See Step 2 of the Sensor End directions for pictures and additional notes)
3. Put a red 30XT connector sleeve on the red cable and a black XT30 connector sleeve on the black cable
4. Solder the exposed wire ends onto a Female XT30 connector. Make sure to solder the red cable to the positive (flat) side of the connector and the black cable to the negative (rounded) side of the connector (see plus and minus signs in connector)
   Note: placing the connector in a fixed place facilitates the soldering process



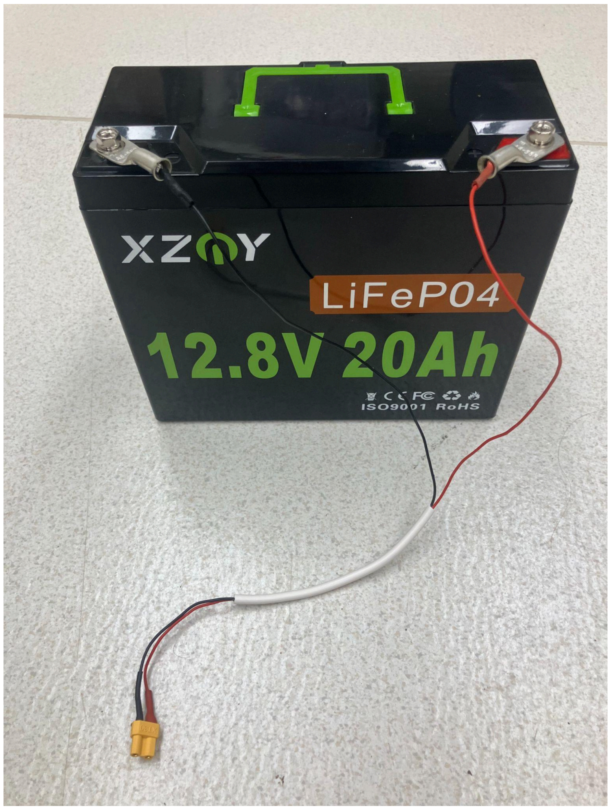
5. Push the sleeves into place close to the XT30 and use the flat of the soldering iron to heat the slaves until they shrink to fit (use back part to not put tin on the sleeves and to not burn the sleeves)

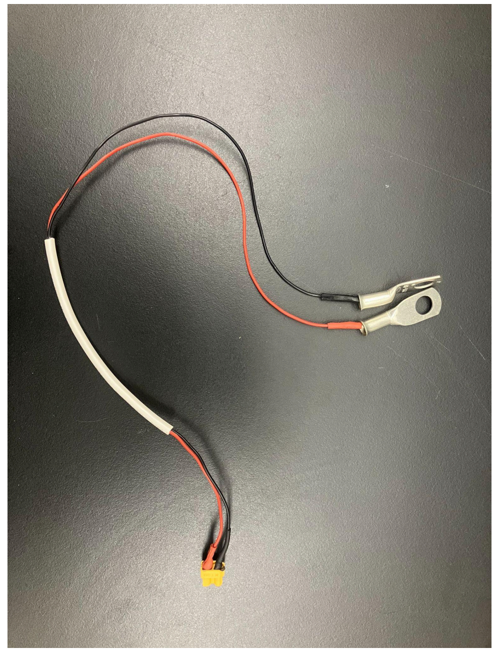
Assemble Battery Cables

This cable is one of the two cables used to connect the sensor to the battery, and this cord connects directly to the terminals of the battery. The purpose of this short cable is to create a connection point where the power cable can easily “talk to” the battery without needing to screw directly into the terminal every time – it is just to save time and logistics when you have to connect and disconnect the sensor from the battery in the field. Each sensor (each battery) needs one battery cable (but again it is good to have extras). On each battery cable, one end is set up to connect to the terminals of the battery (Battery End) using battery lugs (silver), and the other end of the cable is set up to connect to the sensor cable (Sensor end) using an XT30 connector (yellow).

To start, cut one of the 1.25 m sections of multi conductor cable into three roughly equal pieces (each around 40 cm). Each of these sections will be a battery cable.

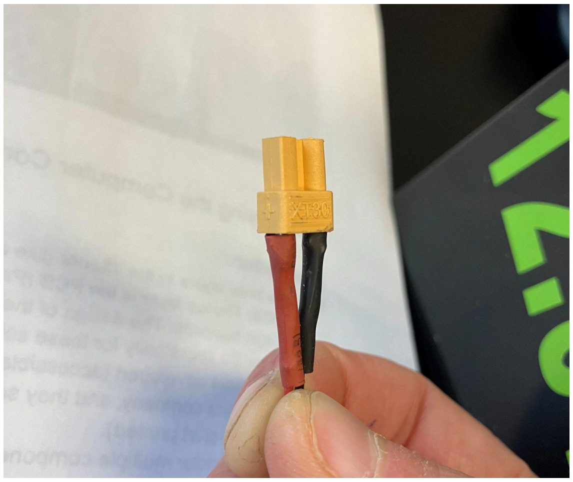
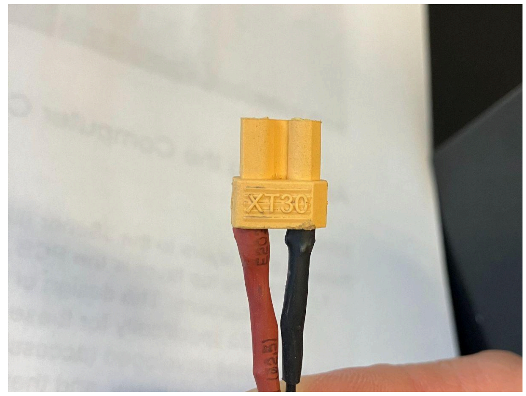
A completed battery cable looks like this:





Prepare the Sensor End

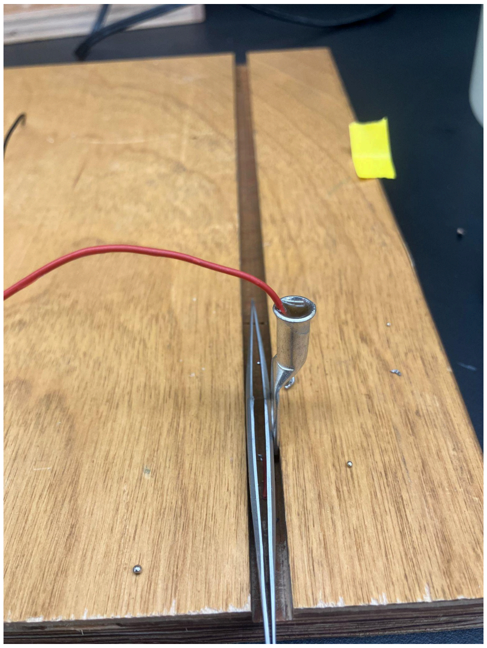
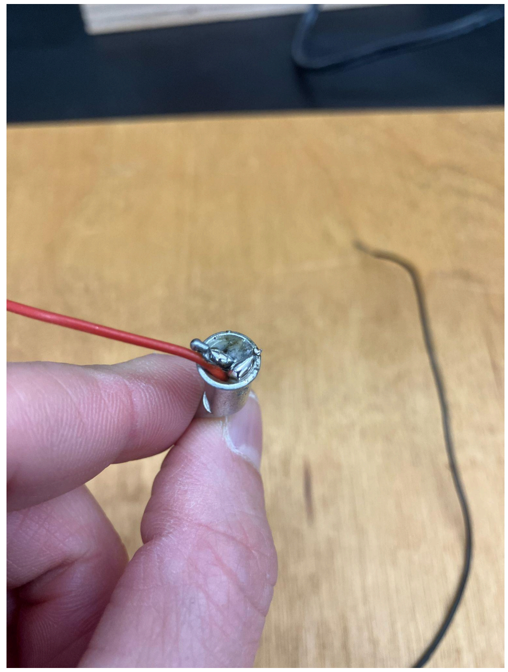
1. Strip one end of the 8 cm multi conductor to expose roughly 15 cm of the internal red and black cables (See Power Cables, Sensor End, step 1 for photos and more details)
2. At the end of the red and black cables strip the colored plastic to expose 1-2 cm of the metal wire beneath. (See Step 2 of the Sensor End directions for pictures and additional notes)
3. Put a red 30XT connector sleeve on the red cable and a black XT30 connector sleeve on the black cable
4. Solder the exposed wire ends onto a Male XT30 connector. Make sure to solder the red cable to the positive (flat) side of the connector and the black cable to the negative (rounded) side of the connector (See Power Cables Battery End Steps 3-5 for photos and more details)

5.  

6. Push the sleeves into place close to the Male XT30 and use the flat of the soldering iron to heat the slaves until they shrink to fit

Prepare the Battery Ends

1. Strip one end of the 40 cm multi conductor to expose roughly 15 cm of the internal red and black cables (See Power Cables, Sensor End, step 1 for photos and more details)
2. At the end of the red and black cables strip the colored plastic to expose 1-2 cm of the metal wire beneath. (See Step 2 of the Sensor End directions for pictures and additional notes)
3. Solder each of the internal cables (red and black) into their own cable lug.
    a. Note: This is a funky step because the battery lugs we have are way to big for these cables (we have not been able to find a better size fit system to go from the big battery to the little power cable). The goal is just to make sure there is a solid solder connection between the multi cable and the metal of the battery lug so that it can connect to the battery. We have found that the best way to do this is to first fill the battery lug with some solder, melt it, repeat, and then stick the cable into the melted metal. Then add some more solder on top.

2. Assembling the Computer Components

Introduction:

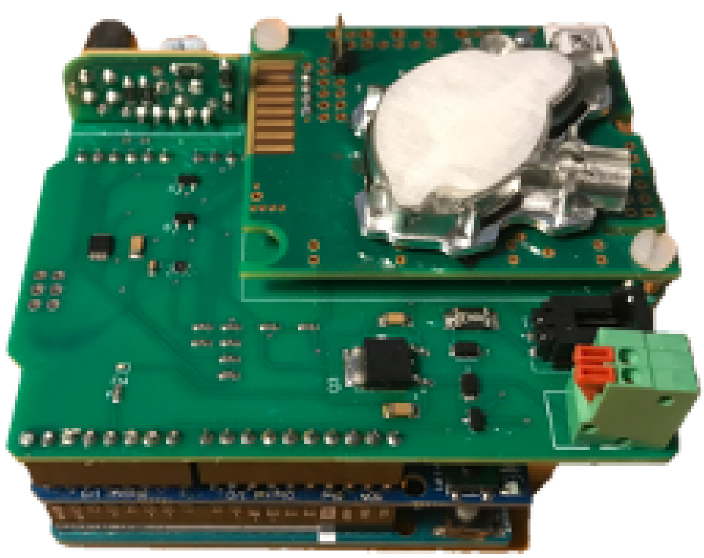
There are three layers to the J-Senor (like a cake).

1. **PCB:** The top layer is the PCB (Printed Circuit Board) and this is where most of the magic happens.
    a. The design of the PCB that we are using was developed by Jonas Stage Sø specifically for these sensors. To get these PCBs we take the (GERBER) files that Jonas developed (accessible on his GitHub linked above), we send those files to an electronics company, and they send us back the custom printed boards (like having a custom t-shirt printed). We also have the PCB company assemble the surface mounted components (this is cheaper and easier than doing it ourselves).
        i. When you order the PCBs: (https://cart.jlcpcb.com/quote) upload the GERBER file (from Jonas's GitHub), select your color, leave everything else as default, then at the bottom of the page select "PCB assembly." As you continue through the checkout process the site will ask you for the BOM and CLP. These are csv files that can be found on Jonas's GitHub with the GERBER file under Version3 > SHT high bit > PCB
            1. The website will give you a warning that "JP1 and JP2 are missing" and will not be assembled. That is okay! Those are spots on the board that you can use to bypass the linear voltage converter. You don't need anything installed here.

b. As part of the sensor assembly, you will need to solder multiple components onto the PCB: (these are not necessarily in the order you will add them to the sensor)
    i. Walrus Connector Header (black receptacle used to connect to the battery cable)
    ii. Vertical terminal block (green and orange box used to connect to the fan)
    iii. CH_4 sensor
    iv. Connections to the Data Logging Shield and the Arduino Uno
    v. CO_2 sensor
c. *Note*: There are multiple versions of this because Jonas has made updates/adjustments. The layout of Version 3 (**GREEN** PCBs purchased March 2024 that are a fun shape) is different from the layout that Jonas made while he was in Ithaca in March 2023 (**BLUE** PCBs). The main differences are:
    i. A new higher accuracy temperature and humidity sensor
    ii. There are surface mounted components (we have the PCB manufacturer add these): LED, linear voltage regulator, temperature/humidity sensor
    iii. The polarity of the connection to the power cable is reversed (the power cables are not transferable between the two versions)

2. **Data Logging Shield:** The middle layer is the data logging shield. This is where the SD card lives (where our data is stored and where we offload) and the logging shield is powered by its own little battery.
3. **Arduino**: The bottom layer is the Arduino Uno (which is technically a microcontroller board). We do not mess with this layer much. We essentially just attach it onto the bottom of our cake.

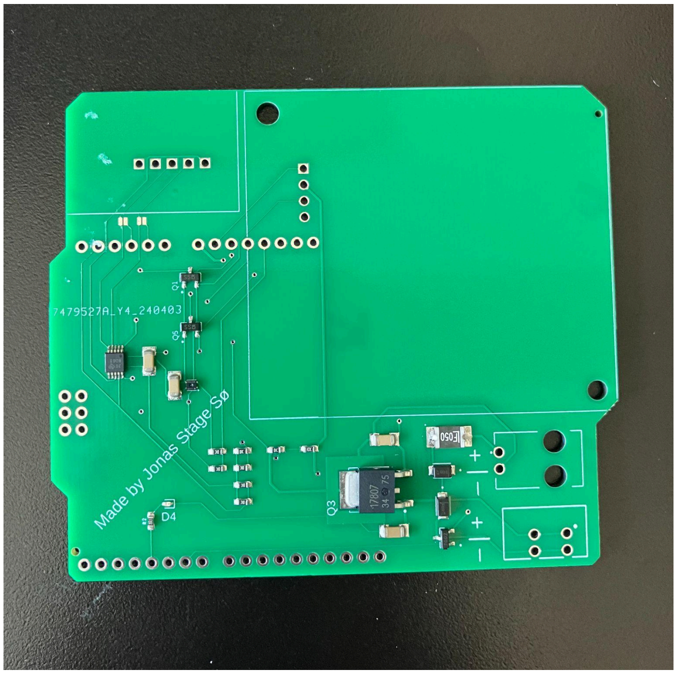
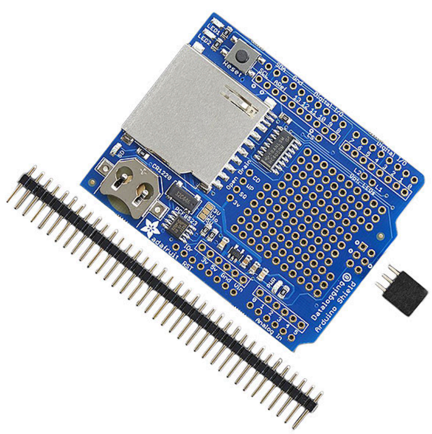
The completely assembled sensor (layer cake) looks like this: (note that this is Version 3)

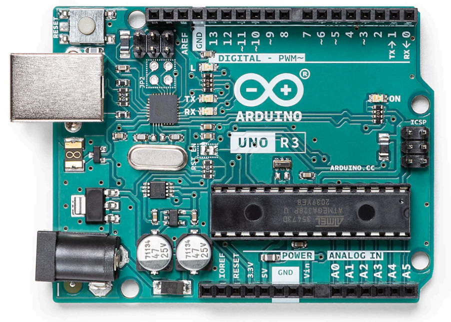


ADD PHOTO (of completed sensor with arrows for what everything is)

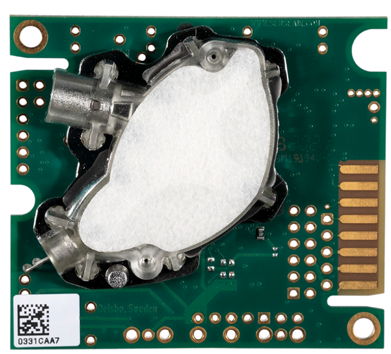
Parts

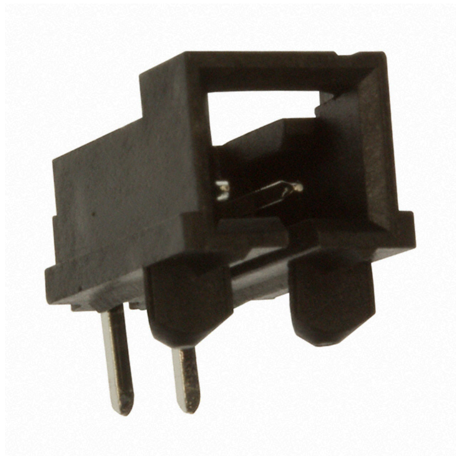
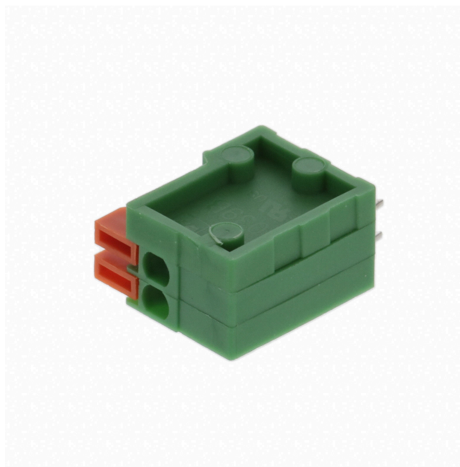
Layers

Part Name	Picture
PCB (Top) **CUSTOM**	
Data Logging Shield (Middle)	

Arduino Uno (Bottom)	

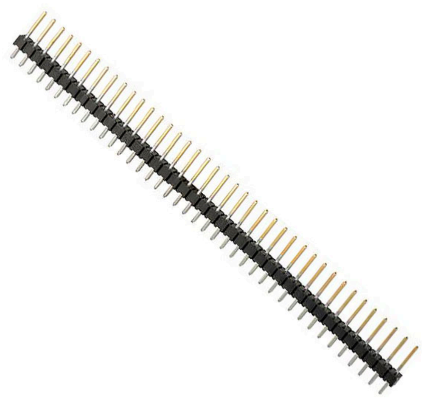
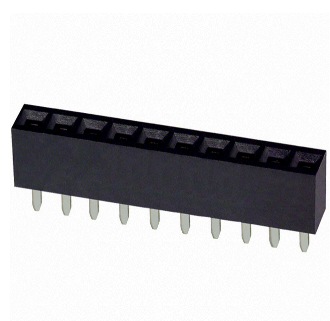
Components

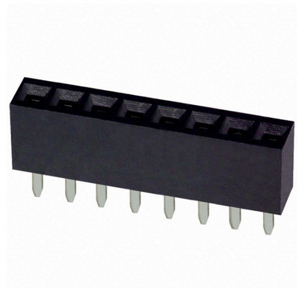
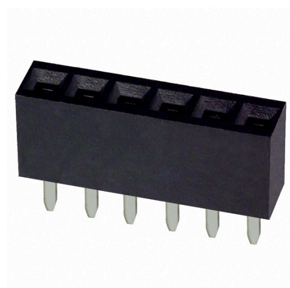
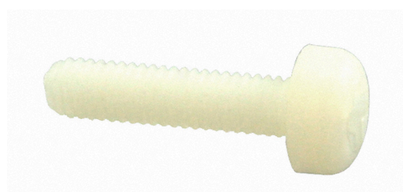
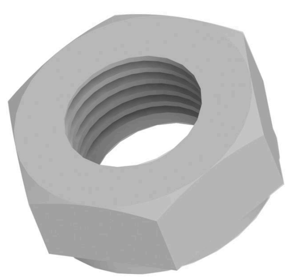
Part Name	**Picture**	**Description**
CH_4 Sensor		From company Figaro
CO_2 Sensor		From company Sensair

Walrus Connector Header		This is where the power cable plugs into the sensor
Vertical Terminal Block		This is where the wires to the fan connect to the sensor
Small Coin Battery		Little 3 volt battery to be installed on the Data Logging Shield to power the RTC
SD Card		SD card to be installed on the Data Logging Shield where the data is stored (offload the SD card to offload the data from the sensors)

Connector Pieces

(*affectionately called Legos*)

Part Name	**Picture**	**Description**
Silver Vertical Con Header		Long pointy silver connectors
Gold Vertical Con Header		Shorter gold connects
Con header Lego 10 piece		Black legos with 10 ports

Con header Lego 8 piece		Black legos with 8 ports
Con header Lego 6 piece		Black legos with 6 port
Plastic Screw		Screws for connecting CO2 sensors
Aluminum spacer		Spacers for connecting CO2 sensors
Plastic nut		Nut for connecting CO2 Sensors

Data and Code Cords

Part Name	**Picture**	**Description**

Cable A to B Plug		This is the cord used to load arduino code onto the sensor, should have a few but don't need one per sensor; CABLE A PLUG TO B PLUG 4.92', make sure this is a power and data cable
SD Card Reader		This is the cord we use to offload data from the SD card

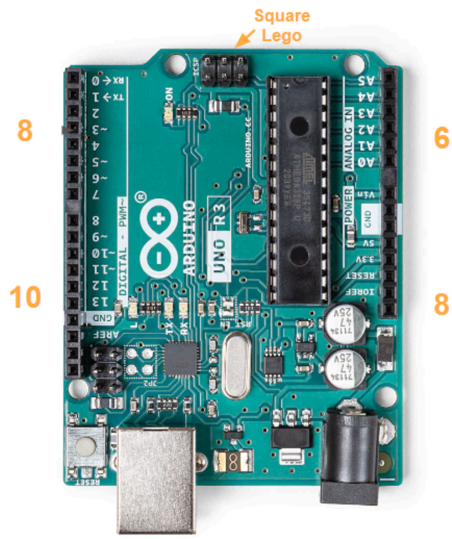
Soldering Tips and Tricks

1. When you are soldering, heat the metal ring on the board, not the solder wire or the connector pins directly
2. When you are soldering, hold the solder wire like a pen and you can use its stiffness to help hold things in place
3. For a successful solder event, there will be a moment when you see the solder liquidize and flow into the hole. The solder also changes color here (unicorn blood)
4. Don't breathe in the smoke from the solder! It has lead in it and it is not good for you

Steps to Assemble

1. Connect the Arduino and the data logging shield
    a. Use the wire cutters to cut sections of the Silver Vertical Con Header (long and pointy). For each sensor you need:
        i. 1 section that is 10 points long
        ii. 2 sections that are 8 points long
        iii. 1 section that is 6 points long

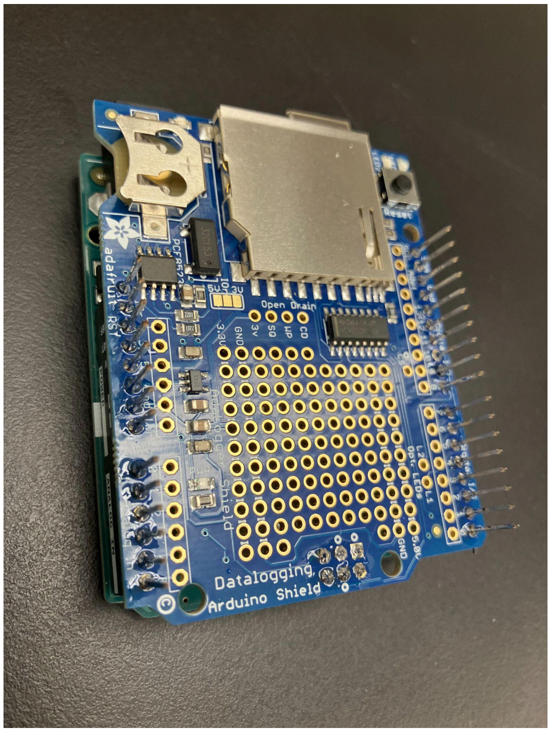
b. Place each section into the arduino so that the long points are sticking up and in the orientation shown here:

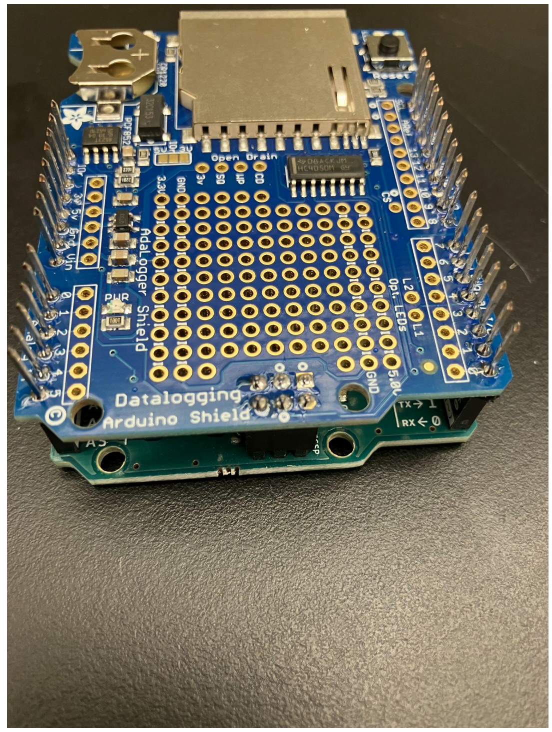


c. Place the small black square box lego on the Arduino Uno as shown in the diagram above so that the points are facing up
   i. Note: this square box lego connector comes with the data logging shield
   ii. Orientation on the Arduino: the square black lego goes on the short side between the 8 connector and the 6 connector (opposite the code cable inlet)

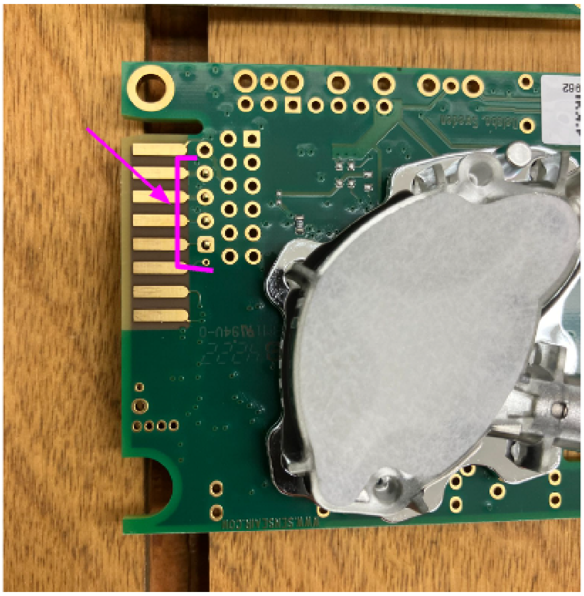
d. Place the data logging shield on top of the arduino so that the points of the Silver Vertical Con Header and the Square Black Lego poke through the corresponding holes

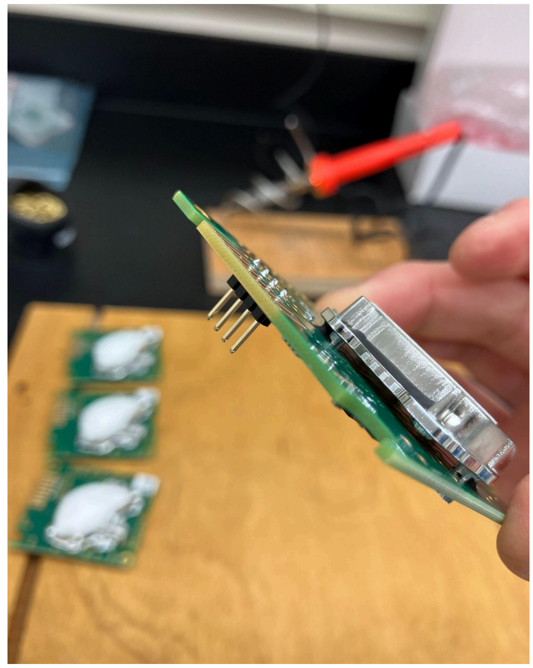
e. Solder all of the points of the Silver Vertical Con Header and the Square Black Lego into place



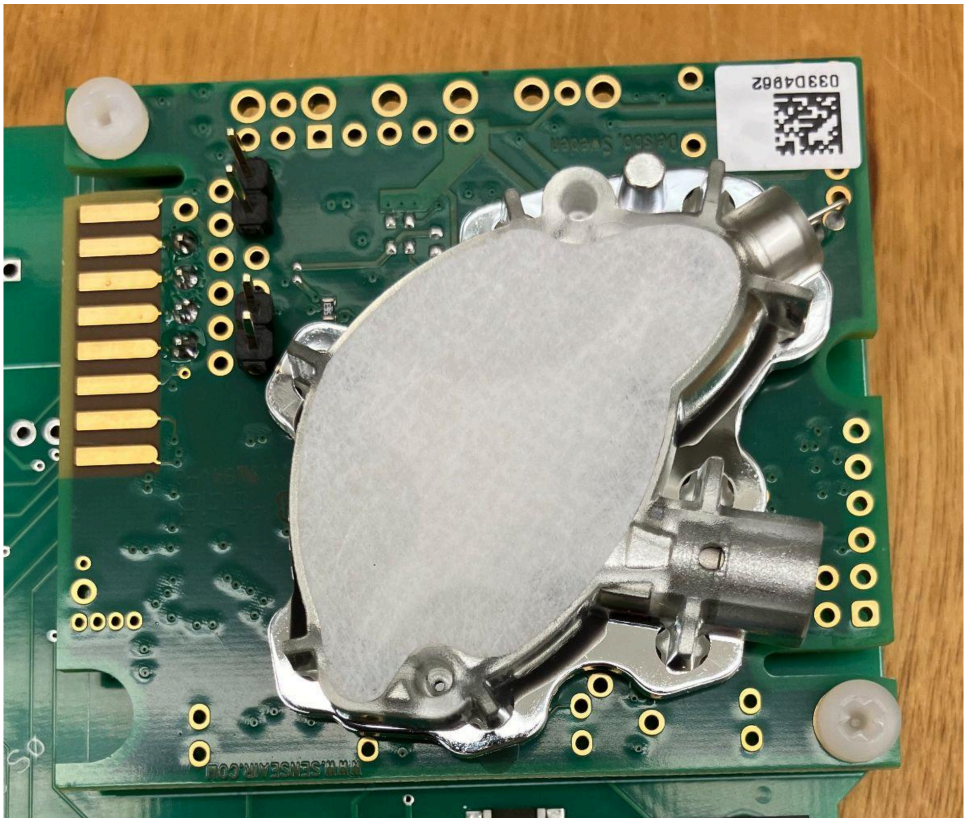


2. Assemble the CO2 sensor and connect it to PCB
    a. For each sensor, cut a section of Gold Vertical Conn Header with 4 points. Balance the Gold Vertical Conn Header under the CO2 sensor (so that the black plastic part is on the bottom (opposite of the white sensor). Make sure to line up the points of the Gold Vertical Conn Header in the correct holes in the CO2 sensor! (see image below). Solder a section of Gold Vertical Con Header with 4 points from the bottom (make sure in the correct holes)
        i. [ADD PHOTOS]

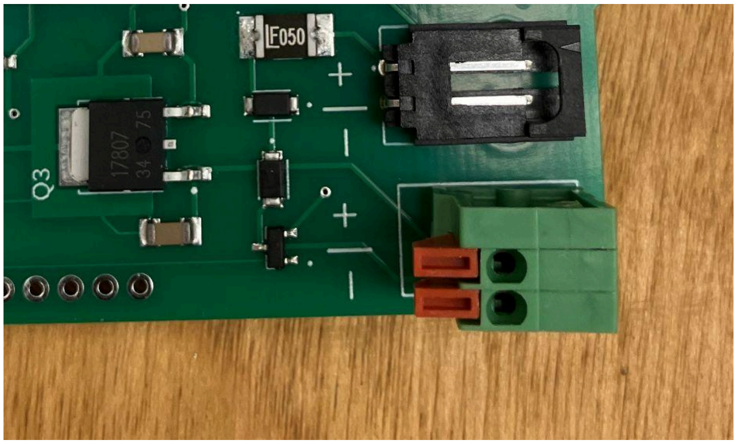


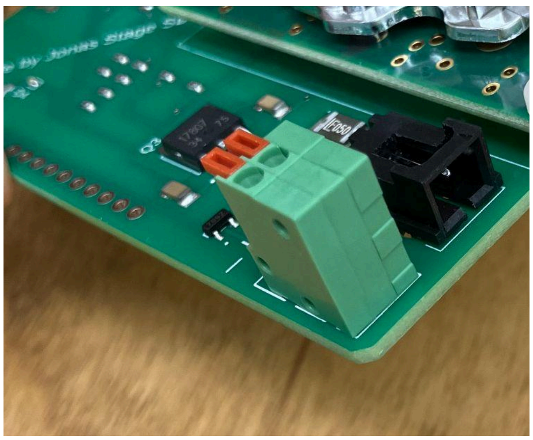


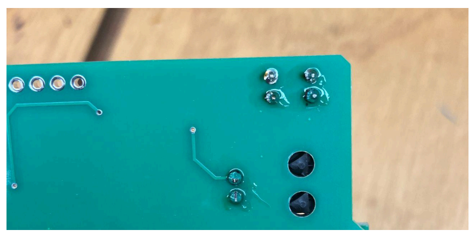
b. For each sensor, cut two sections of Gold Vertical Con Header with 2 points each. Orient these connectors so that the black plastic part is on top of the CO2 Sensor (the same side as the big white piece) and then solder them in place from the bottom. These are the two “short cut” pins that can be used for zero and atmospheric calibration of these sensors.
    i. Note that this photo was taken of a fully assembled sensor, but at this point your CO2 sensor should NOT be mounted to the PCB



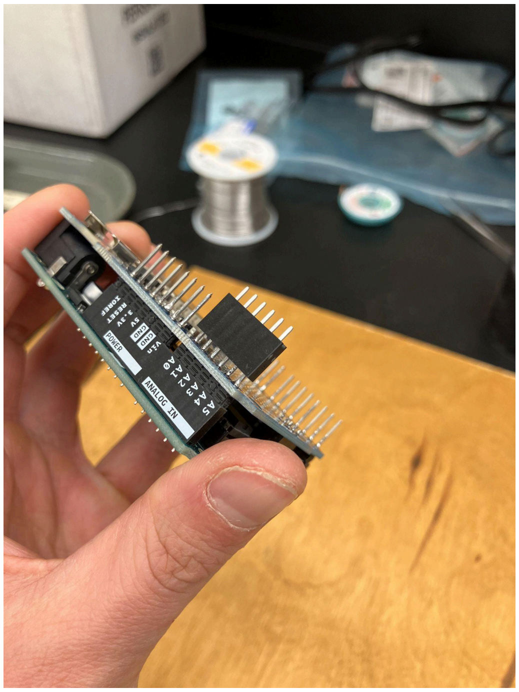
3. Solder the Vertical Terminal Block (green and orange connector to the fan) onto the PCB







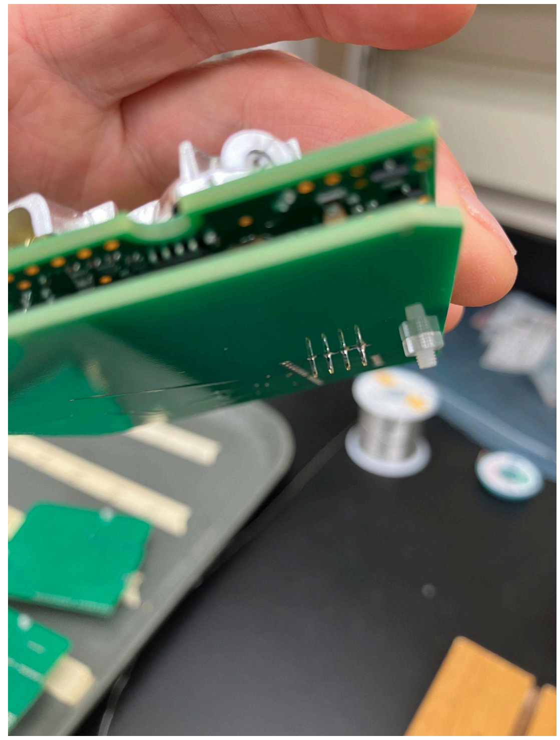
4. Solder the Walrus Connection Header (black receptacle where the power cable plugs into to the PCB) onto the PCB
    a. See photos above with the vertical terminal block
5. Solder the CH4 sensor to the PCB
    a. [ADD PHOTOS]
6. Connect the PCB to the Arduino and data logging shield
    a. Place the Con Header Legos (10, 8, and 6 pieces) on top of the data logging shield. Each Con Header Lego should be placed over the corresponding number of points coming up from the data logging shield such that the point of the Silver Vertical Con Header goes into the bottom (flat part) of the Con Header Lego.
        i. You will need one Con Header Lego-10; one Con Header Lego-6; and two Con Header Lego-8s



        ii. [ADD PHOTOS]

    b. Place the PCB on top of the Data Logging Shield so that the points of the Con Header Legos poke up through the corresponding holes
    c. Solder all of the points of the Con Header Legos into place on the PCB

7. Screw the CO2 Sensor into place on PCB with the spacers between the sensor and the PCB and use the nuts to secure in place
    a. [ADD PHOTOS]
8. Solder the CO2 sensor to the PCB

9. Insert the small coin battery on Data Logging Shield Layer to power the real time clock (RTC)
10. Insert SD card into the slot on the Data Logging Shield

3. Load Arduino Software Onto the Sensor

Background/Intro:

These sensors will continuously be running the last code that you "pushed" onto them (they are like a wind up car, they keep going in the same direction you pointed them until you pick them up and change direction). There are three different sketches (code files or sets of instructions) that you will need push onto these sensors under different circumstances:

1. Time Sync Sketch– "RTC_set.ino" This sketch tells the sensor to sync its "internal" time to the time on your computer. The purpose is to set the RTC (real time clock) on the sensor to the correct time so that you when you write down in the field that the sensor was deployed (for example at 10:23 am) that time matches the time that the sensor records
    a. It is a good idea to check that the time is correct on all of your sensors about once a month (the clock will be slowly become more off)
2. Measurement Sketch – "DEC_High_bit_methane_CO2_sensor.ino" (or the "Sensor" Sketch from Jonas's GitHub)This sketch tells the sensor to "measure" for 15 minutes and then to "flush" for 15 minutes. This is the core that you want pushed to the sensor when you take it out into the field. Remember that the measurement time window is when the

sensor is measuring the partial pressure of CH4 and CO2, and the flushing time window is when the fan is running to circulate and flush the air out of the chamber ("reset").

   a. Note that you can change these time window (Measure for 30 and flush for 20 ect.) you would just need to reach out Jonas and figure out how to update the instructions code
   b. For this code, an empty csv file is created and the data (measurements) are stored there on the SD ccard

3. Calibration Sketch – "CAL_High_bit_methane_CO2_sensor.ino" this sketch tells the sensor to continuously measure CH4 and CO2 partial pressure. This is the code that you will push to the sensor before you put it into the calibration chamber (because when calibrating we don't want the sensors to stop measuring for 15 minutes while flushing)
   a. Again, for this code, an empty csv file is created and the data (measurements) are stored there on the SD card

Setting up your Computer

Before you can push code to the sensors, you need to set up your computer so that it can "talk" to the sensor (with arduino software and the relevant libraries) and so that your computer has the right instructions to give (has the code files). The prerequisites to set up the code on the sensor, you first need to:

1. Have Arduino IDE software installed on your computer. You can download it from the Arduino website.
2.
3. Install all of the required libraries
   a. Required libraries for Sensors from Jonas https://github.com/JonasStage/Methane-and-CO2-sensor/tree/main/Arduino%20libraries
   b. To Load/install a library in the arduino software (you must have them zipped; .zip): Arduino IDE → Sketch → Include library → add zip library
   c. Additionally needed library besides the ones on GDrive: Adafruit BusIO
4. Download the Arduino sketch for the sensor and the RTC. Make sure to download and install the needed libraries first. Have time sync code file, calibration code file & measurement code file setting file ready; emptied SD card
   a. Sketch for the sensor: (Measurement) https://github.com/JonasStage/Methane-and-CO2-sensor/blob/main/Version%203/SHT_highbit_CH4_CO2/SHT_highbit_methane_CO2_sensor/SHT_highbit_methane_CO2_sensor.ino
   b. Sketch for setting the RTC: https://github.com/JonasStage/Methane-and-CO2-sensor/tree/main/RTC
   c. Sketches specific to DEC project in the Holgerson Lab can be found on teh Google Drive Here: Arduino_Code

The First Time Connecting New Sensors

The first time you connect brand new (just assembled) sensors:

1. Run the sync time sketch to set the RTC (make sure the time looks correct) (Details bellow)
2. Push the Calibration Code and then calibrate the sensors
3. Push the Measurement Code when you are ready to take them out and measure

Pushing Code to the Sensors

Once your computer is set up and you are ready to load the code on the sensors:

1. Connect the sensor to the laptop using the USB-A to USB-B cable (make sure it is a power & data cable)
2. Open Arduino IDE, and connect to the sensor by choosing the correct port in the scroll-down menu at the top; if connected correctly, there should be a port stating Arduino Uno (COMX)

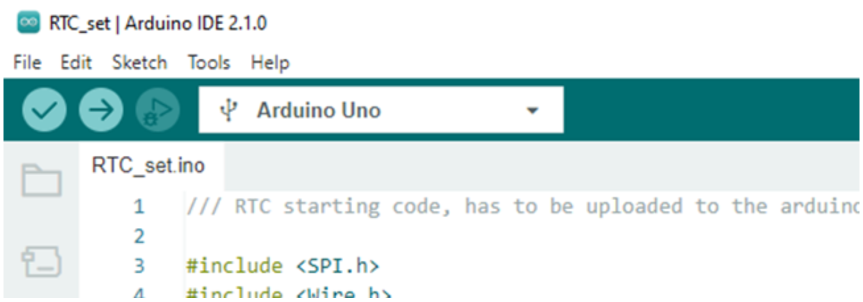
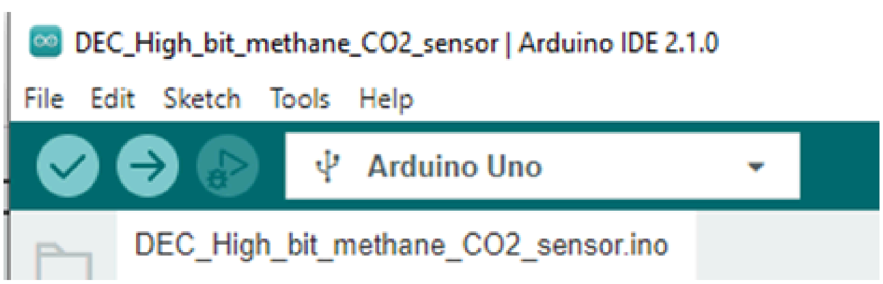


Figure 1: Connect the sensor to the software

3. Sync sensor time & laptop time:
   a. Open the Time Sync Sketch (RTC) in the Arduino Uno Software: Click File > Open in the upper left corner of the screen then navigate to where you have the Sketch saved on your computer. Sketch is named “RTC_set.ino”,
   b. Push the Time Sync Code to the sensor by pushing the arrow in the top left corner



   c. To check that it is working, click on the magnifying glass in the top right corner; this will open a tab in the bottom part of the window named ´Serial Monitor´; here, the sensor readings should display

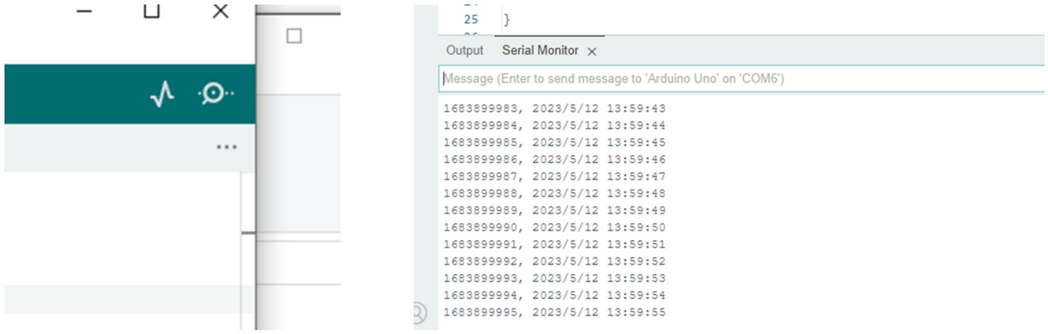
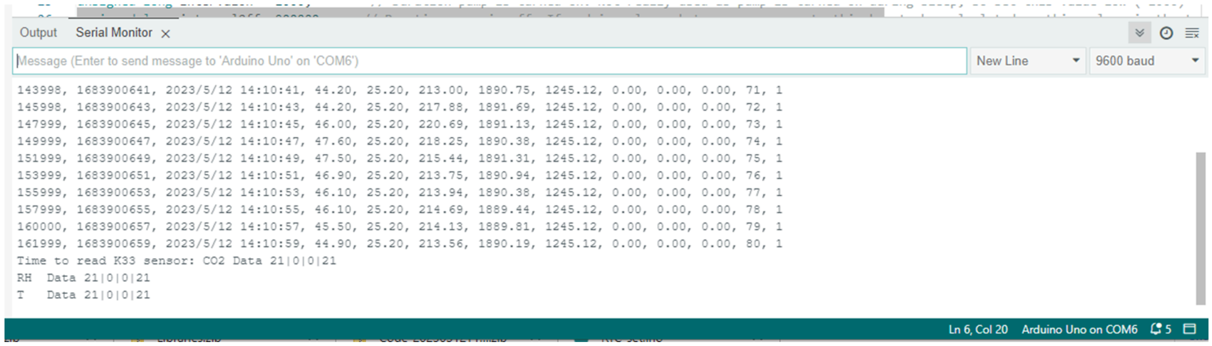


Figure 3: Magnifying glass button to click on to display outputs (left) and example of output with RTC script (right)

4. Set Sensor to measure or calibrate
    a. Open the Measurement Sketch in the Arduino Uno Software: (Same process as opening the RTC Sketch) Click File > Open in the upper left corner of the screen then navigate to where you have the Sketch saved on your computer. Sketch is named "DEC_High_bit_methane_CO2_sensor.ino"
        i. Note that you can also use the "Sensor" Sketch from Jonas's GitHub but I believe that the time windows are different (not 15 min measure and 15 min flush)
    b. Push/Upload the Measurement Sketch to the sensor by pushing the arrow in the top left corner (again, this is the same process as pushing the RTC sketch)
        i. Note that this takes a couple of seconds to load
    c. To check that it is working, click on the magnifying glass in the top right corner; this will open a tab in the bottom part of the window named ´Serial Monitor´; here, the sensor readings should display
        i. Note that you might not get values from the K33 ELG CO2 sensor if you only power the sensor through USB as it needs at least 9 V.



5. When you have verified that the sensor is running the code that you want it to be, simply disconnect the sensor from the USB-A to USB-B cable (There no “preparation” in the software needed before this step). The next time that the sensor receives power (when you plug it in to the battery in the field or in the calibration chamber) it will keep running the same code that you pushed to it.
6. Take the SD card off, connect the SD card to the laptop, and erase all the data on it (you don’t need the data of the CH4 and CO2 in the office or the field station where you were running this code)

4. Building Chambers

Parts:

For each chamber you will need:

- Bucket (1)
- Foil tape (~½ roll)
- Tupperware (2)
- screws (2)
- Nuts (2)
- Duct tape (⅓ of a roll)
- Insulation foam (~ 1m by 1m)
- Vinyl tubing (3 m)
- Fan (1)
- Pump nozzle (1)

Steps:

1. Wrap the outside of the bucket in aluminum foil tape to increase the reflectance.
    a. Note that if you want to use a bucket of a different size/dimensions it’s important to test and make sure that it wont put the sensor too close to the water’s surface as this might cause the sensor to short (break) due to water damage. Here a bucket with a volume of 13.5 l and surface area of 0.0615 m2 is used.
2. Remove the handles from the buckets (these will just get in the way)

3. Drill Tube holes in buckets: Drill two holes on each side of the bucket (9/16th in), approximately 3/4 of the distance from the top of the bucket to the bottom.These will serve as the two ports for the two different vinyl tubes (one connected to the fan to flush the chamber, and the other open to equilibrate pressure.
    a. Use a 9/16 in drill bit so that the tubing will fit snugly inside without leaking

4. Drill Ziptie Holes in the bucket: Using a smaller drill bit, drill small holes in the outer lip of the bucket for the zip ties to go through to attach to the life jacket
    a. Note when you are choosing your bucket that you want this lip that you can connect to. You do not want these connection holes to be drilled into the main body of the bucket.

5. Swiss cheese tupperware: Using the large (9/16 in) drill bit, drill holes in the sides of the tupperware (storage box) to allow airflow (I did 5 holes on each side of the tupperware). Do not drill big holes on the lid or on the bottom of the tupperware. Ensure that
    a. You will need two tupperware for each chamber
    b. The tupperware will need to be able to fit inside the buckets and it is nice to have a bit of room on the sides. We have had success with the tupperware that have flaps that snap down over the sides.
    c. On each tupperware pair, make sure that at least one of them has a hole large enough for the vinyl tubing to go through

6. Drill holes in the tupperware for the screws and the fan cable hole
    a. Stack two tupperware bottom to bottom (facing away from each other). This is how they will sit when assembled on the chamber so that you can open the tops and access the sensor and the fan.
    b. Using the 9/16 drill bit, drill matching holes through the bottom of both tupperware in three corners
    c. The two opposite corners will be to screw the tupperware in place on the bucket and the third hole in the other corner will be used to allow the fan cables to pass through

7. Drill holes in the bucket for the tupperware and fan
    a. Place one of the tupperware (with the three holes already drilled) flat on the bottom of the outside of the bucket. Drill through the holes in the tupperware to make matching holes in the bucket.
8. Assemble screws and tupperware and put on bucket
    a. Place one tupperware inside the bucket (bottom of the tupperware against the bottom of the bucket on the inside).
        i. Make sure this is not the one with the big hole for the vinyl tube!
    b. Thread screws through the matching holes in the bucket and the tupperware
    c. Add an O-ring to ensure no air is exchanged through the drilled holes.
    d. Place the other tupperware on the bottom of the bucket on the outside and line up the holes so that the screws poke through the holes in the second tupperware
    e. Use the nuts to secure the two tupperware together (making a bucket sandwich)
    f. Replace the lids on the tupperware (don’t lose these!)
    g.

9. Connect the pump nozzle to the outlet of the fan
    a. [ADD PICTURE]
10. Put the fan and the connected pump nozzle into the tupperware on the outside of the bucket and thread the cables through the third hole (that doesn't have a screw in it) down into the bucket (where it can connect to the sensor). Seal the hole with some silicone.
    a. [ADD PICTURE]
11. Cut tube
    a. Cut vinyl tubing into 1.5 m sections.
    b. You will need two sections of vinyl tubing (each 1.5 m) for each chamber
    c. When cutting the tubing cut one end flat (to either fit on the nozzle or be open to the air) and cut the other end at an angle to fit through the holes in the bucket
    d. Insert one section of tubing into each of the holes (this will be a tight squeeze because you want to get an airtight seal)
    e. [ADD PICTURE]
12. Thread the vinyl tubing through one of the large holes in the outside tupperware and connect to the nozzle on the fan
    a. Both sections of tubing should be coming out of holes on the sides of the bucket. One tube should be attached to the fan and the other should be free

    b. You can also do this step our in the field to make them easier to transport but threading the tubing in and out is kind of clunky
    c. [ADD PICTURE]

13. Cut out life jacket
    a. Cut out an octagon of styrofoam with a circle inside it big enough for the widest part of the bucket to sit in (not that you want a few cm of extra room around the bucket so that it can move a bit and does not get pulled above the surface of the water
    b. Duct Tape the rough edges of the pink styrofoam life jacket so that you are not leaving little bits of foam everywhere in the lake

14. Zip tie the whole contraption together: thread the heavy duty zip ties through the holes in the bucket and around the life jacket (you can also do this step out in the field to make them easier to transport)
    a. [ADD PICTURE]

5. Calibrating J Sensors

Background:

This protocol is for calibrating the CH_4 sensor on the J sensors. This procedure will need to be repeated if we change either the CH_4 sensor or the humidity sensor on the J sensor. This calibration procedure needs to be carried out for each sensor. Multiple sensors can be calibrated at once, but *each* sensor needs its own calibration coefficients (they are not the same across sensors). Note that the CO_2 sensor arrives calibrated from the manufacturer. The CO_2 sensor's output is already in CO_2 ppm and does not need to be calibrated in our lab.

The CH_4 sensors on the J Sensors work by measuring the resistance in a special anode. The resistance in the anode changes based on the concentration of CH_4 present and the concentration of H_2O present (humidity). The sensor reads and outputs those changes in resistance as readings in millivolts (mV), meaning that when we offload the sensor the numbers we get for CH_4 are a bunch of millivolt readings. In order for that information to be useful to us, we need a way to "convert" those millivolt readings to ppm of CH_4. This procedure is how we gather the data to create the *conversion coefficients* we can then use to convert from millivolts to ppm of CH_4.

The resistance of the anode responds to changes in two things: 1) CH_4 and 2) humidity. As a result, in order to get our calibration coefficients we need to have a range of CH_4 concentrations (2 ppm to 115 ppm) at multiple humidity levels (High >= 20,000 ppm H2O; medium ~15,000 ppm H2O; and low <= 10,000 ppm H2O)

Protocol:

1. Power on J sensor(s) to be calibrated and LGR/Picarro to warm up.
2. Inside the calibration chamber (home depot bucket with lid) set up:
    a. Sensor(s) to be calibrated
    b. Battery connected to sensors (make sure to cover the battery with a towel or something to separate it from the sensors and prevent a short in the circuit)
    c. Large computer fan connected to the battery
    d. I also taped a p-cup to the side of the bucket and put the end of the tube coming from the humidity chamber in the p-cup to catch any excess water coming in

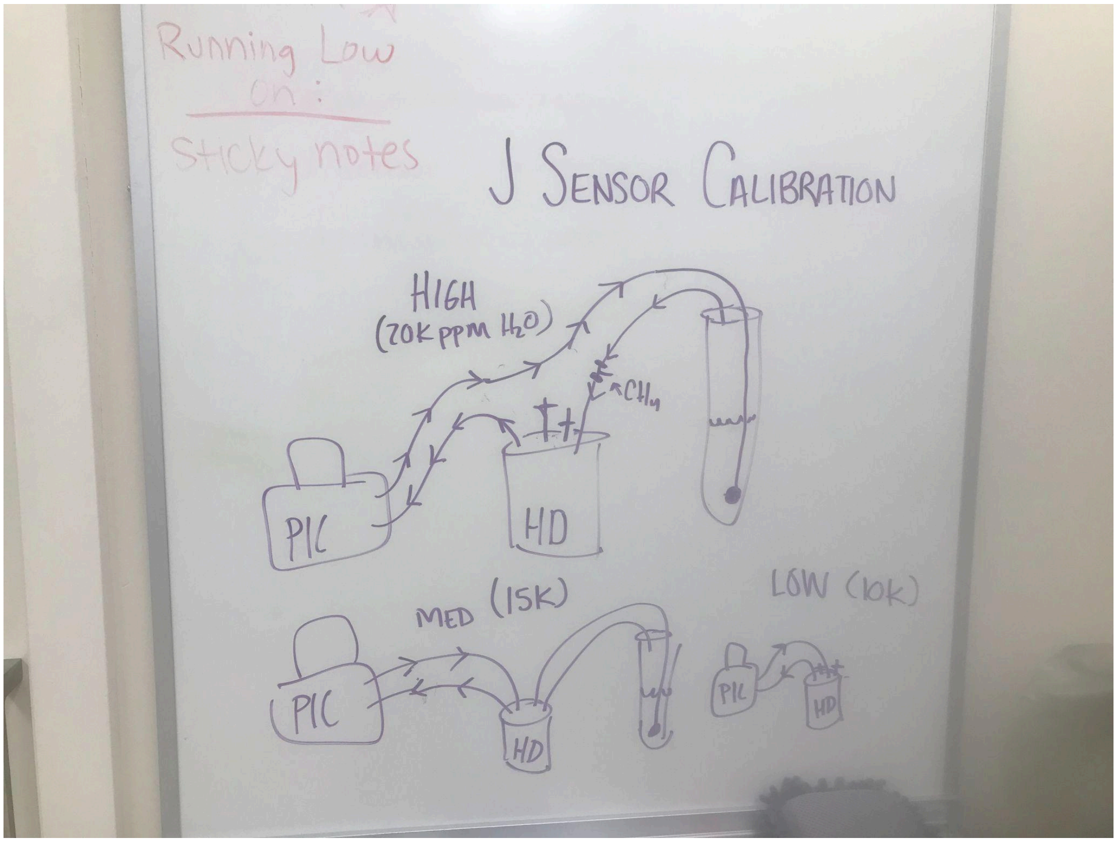
3. Set up tubing based on diagrams for high humidity.
    a. Outflow from Picarro or LGR is connected via 3 way stop cock to the tubing of the aerator.
    b. The Aerator is submerged in the humidity tube.
    c. Humidity tube is ¾ full of hot water.
    d. Tubing leaving the humidity tube goes directly into the calibration chamber (Home Depot Bucket).
    e. Tube leaving the calibration chamber goes into the inflow of the LGR
        i. Note: Use the shortest/least amount of tubing possible so that the LGR is reading the concentrations in the bucket (that the J sensors are reading) rather than in the tubing. Using shorter tubing also helps you change the humidity in the chamber and helps control leaks.
    f. The other two ports in the calibration chamber have short sections of tubing sticking out that are attached to closed luer locks.
4. Close the system with the hot water in the humidity tube and the aerator running with the outflow air from the LGR.
5. Using the LGR interface, monitor the humidity levels in the calibration chamber and wait until the humidity reaches above 20k ppm of H2O. (this can take a little bit, you might have to leave it for 20 minutes to a half hour).
6. Once the humidity levels are high enough, inject methane through one of the short tubes with a leur lock. Inject enough methane to get to above 100 ppm (This usually takes about 300 mL if you are using the 1% methane that we use for standard curves on the GC).
7. Because the system is not perfectly sealed, the methane will slowly diffuse out of the calibration chamber over time allowing us to get measurements continuously from 100

ppm down to 2. Leave the calibration system alone and check on it periodically (every 45 minutes or so). You should see the methane concentration gradually decreasing over time.

   a. Alternatively, you can add methane to increase the concentration in the calibration chamber in steps to monitor and make sure you cover each gradation equally. I have been increasing my 20 ppm every 20 minutes and letting the concentration decrease

8. Once the concentration of methane gets down to about 2 ppm (this can take a couple of hours) change the tubing configuration to that for medium humidity. Try to do this quickly to allow minimal escape of humidity.
   a. Outflow tube from the LGR connects to one of the inflow ports in the calibration chamber with the leur lock
   b. Tube going into the humidity chamber (through the aerator) connects directly into the other inflow port in the calibration chamber with the leur lock.
   c. Tube out of humidity chamber does not change (still goes directly into the calibration chamber) and tube out of calibration chamber into LGR does not change.
9. Note: The humidity levels also should have slowly decreased and should be around 15,000. Allowing the system to still be connected to the humidity chamber but not actively pumping air through the water should allow the system to stay at medium humidity.
10. Again, inject enough methane to get to above 100 ppm.
11. Leave the calibration system alone and check on it periodically as the methane concentration gradually decreases over time.
12. Just as before, once the concentration of methane gets down to about 2 ppm (this can take a couple of hours) change the tubing configuration. This time you are going to set it for low humidity by disconnecting the calibration chamber from the humidity tube completely. Humidity should decrease to about 10,000 ppm
13. Again, inject enough methane to get to above 100 ppm. Leave the calibration system alone and check on it periodically as the methane concentration gradually decreases over time.
14. Once concentrations get down to approximately 2 ppm, open the system and offload the J sensor SD card.
15. Use that data to run the J sensor calibration code: (01_J_Sensor_Calibration.R) https://github.com/meredithholgerson/JSensorCalibration (this is a private repository so you will need to ask MH for access)

Figures:


Running Low
on:
sticky notes
J Sensor Calibration
HIGH
(20K PPM H_2O)
CH_4
PIC
HD
MED (15K)
PIC
HD
LOW (10K)
PIC
HD


THE HOME DEPOT


ETHAN


ASSEMBLY

Testing Sensors

Data Analysis

- The package that Jonas wrote for dealing with sensor data: https://github.com/JonasStage/FluxSeparator
- Jonas noted paper about low R2 values at fluxes near zero: https://onlinelibrary.wiley.com/doi/full/10.1002/jpln.201600499?utm_sq=gvl258wtvi

Offloading Data from the Sensors