The following design is based on the Environmental Microcontroller Units (EMUs) of James Mickley and colleagues at UCONN. See https://github.com/mickley/EMU and:

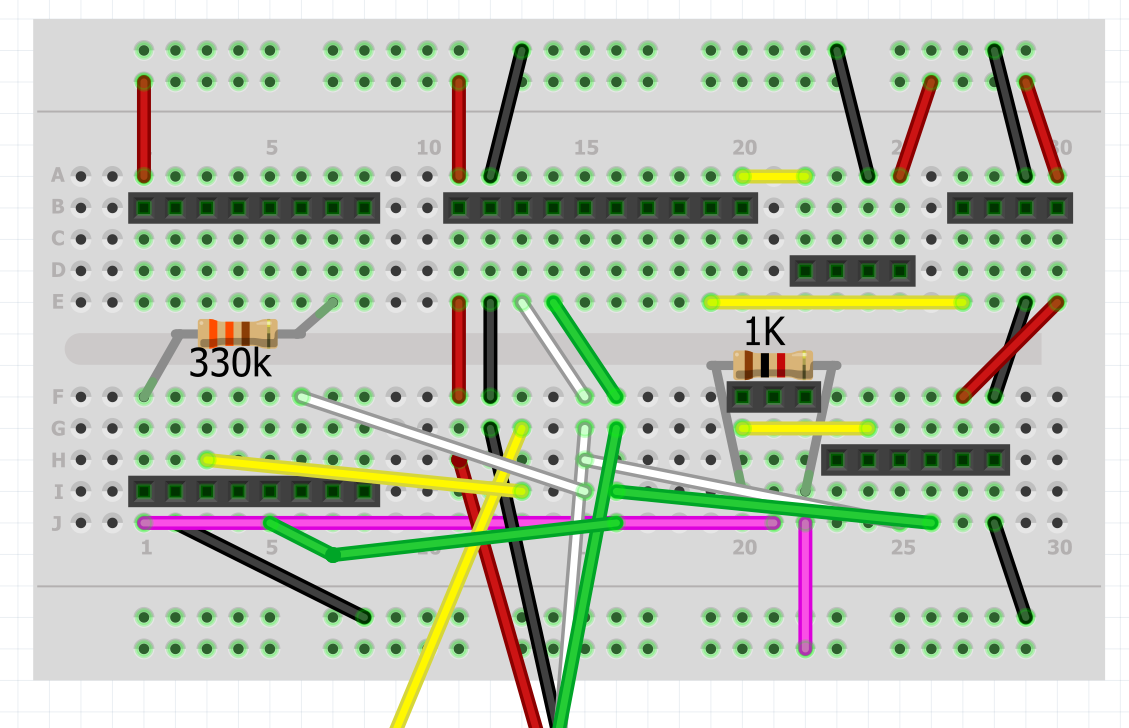
Mickley, J. G., T. E. Moore, C. D. Schlichting, A. L. DeRobertis, E. Mason, & R. Bagchi (2018). Measuring Microenvironments for Global Change: DIY Environmental Microcontroller Units (EMUs). Methods in Ecology and Evolution. 1-7. doi: 10.1111/2041-210X.13128

The below modifications include programming and data access via Arduino IDE rather than Lua; more robust external wiring and housing, soldered connections on a pre-printed circuit board in place of a breadboard; a soil temperature sensor; and two rather than one soil moisture sensors. Total per-unit cost is about $20.

This ‘B’ version of sensors was primarily designed to be completely belowground and not detected by animals in the field, but it contains several other improvements compared to the ‘A’ version including (1) pre-printed circuit boards which reduce construction time and improve reliability, (2) improved housings constructed from PVC for lower cost and increased water resistance, and (3) new code that allows sensor to enter ‘deep sleep’ if there is an error with the RTC and allows data download without uploading a different program. The below  
Arduino scripts are available at github.com/jdfridley/Sensors

The Arduino code was written with Arduino IDE version 1.8.9. The board manager was used to install ‘esp8266’ version 2.5.2 (note that more recent versions may be incompatible with the SPIFFS storage – I am working on a solution). The following Arduino libraries are also required: ‘RTClib’, ‘uRTClib’, ‘OneWire’, ‘DallasTemperature’, and ‘Adafruit\_1x15’.

1. **Wiring & testing components**
2. For testing, it is useful to have a copy of the printed circuit board on a breadboard—wired as in this diagram. A battery pack can be attached to the lower power rails to provide power. I generally test (1) microcontrollers-just make sure that they turn on; (2) RTCs, both to check that the time can be set and that the alarm function works; (3) MOSFETs, to make sure they work with the alarm. I have not had issues with other components but they could also be checked using this wiring.



microcontroller

microcontroller

ads 1115

vmc 2 (10cm)

vmc 3 (surface)

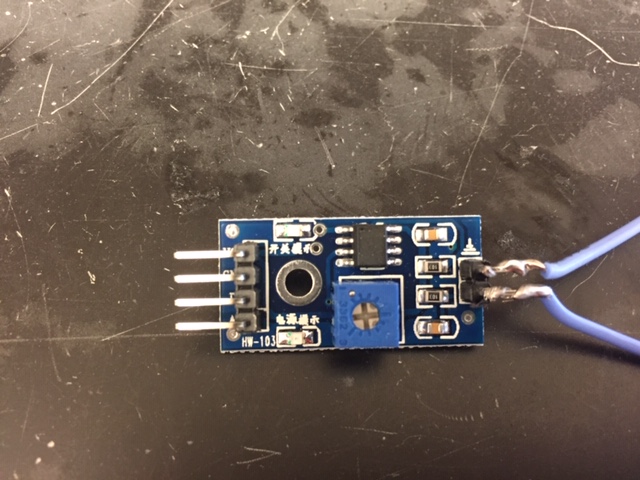
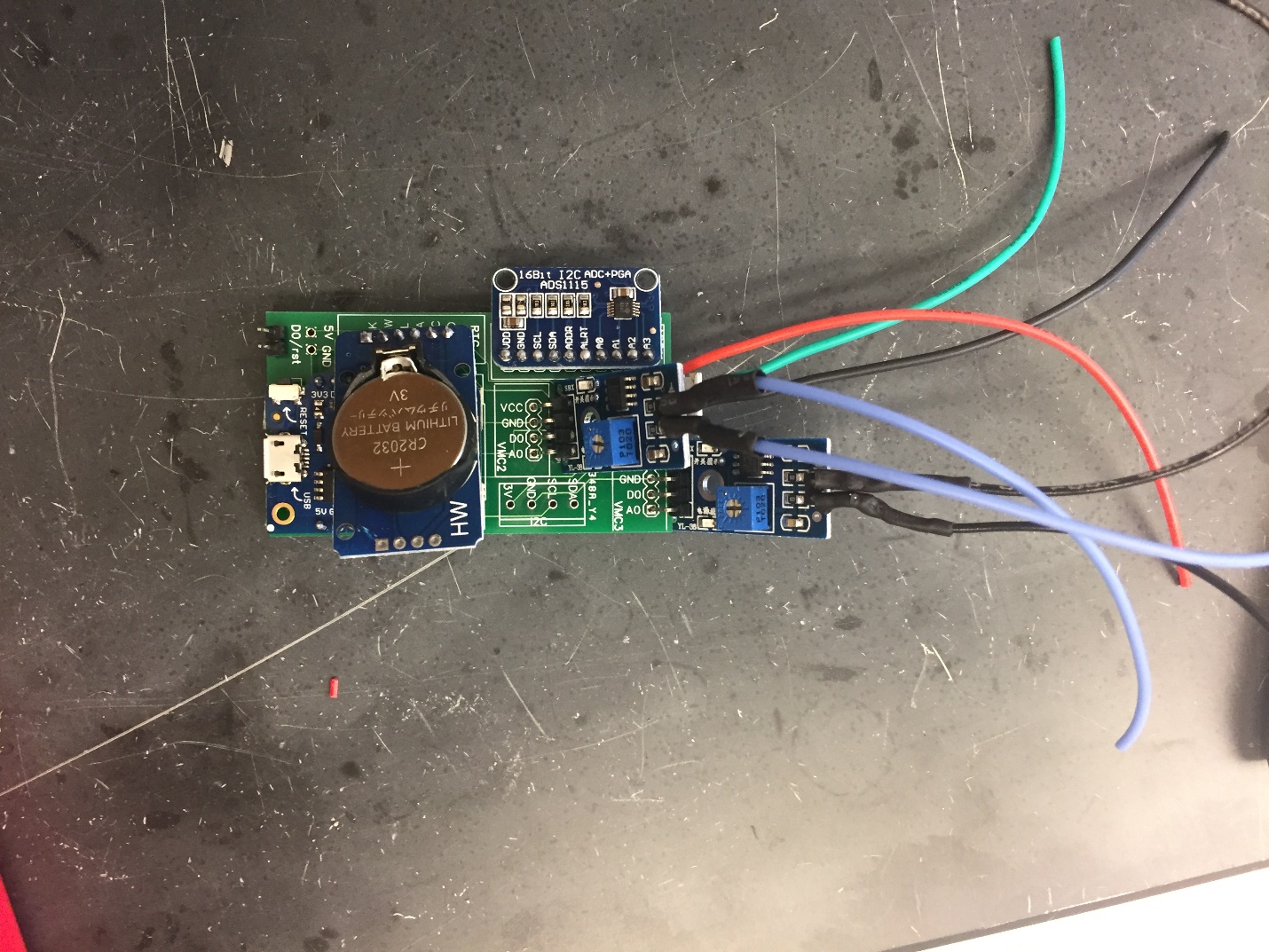
MOSFET

Clock

1. Circuit board – upgraded to a custom printed circuit board – design attached. We ordered these from JLCPCB for ~$0.50 per board--they decrease cost and time. If you are only making a few, you could solder these instead (see ‘A’ protocol).

*Gerber design files in ‘B sensor Gerber’ folder.*

i. Solder ~10 cm of color-coded wires to the “1wire” section of the board (red to 3v, black to gnd, yellow to OneWire). If using any I2C probes (not described in this protocol but see Mickley et al. or the ‘A’ sensor protocol) additional wires can be soldered to the I2C section

Custom printed circuit board with soldered components

3. Other components soldered to circuit board

*These components (pre-tested) can be wired into their correct locations on the circuit board above:*

i. microcontroller – lolin d1 mini

*Use short pins with black plastic part underneath. Test to make sure that the microcontroller can be programmed by uploading any sketch.*

ii. real-time clock – ds3231sn

*Remove upper-left resistor marked “102” and 4x472 resistor across from vcc. Remove LED lights to save power. MAKE SURE TO TEST THESE-about 25% do not work (either cannot accept time input or will not enable alarm). I am setting and checking the time with board\_config.ino and then running test\_clock\_blink.ino which should turn the microcontroller on once per minute and blink the onboard LED 3x.*

iii. analog-to-digital converter – ads1115

*Solder to pins. I haven’t had any issues with these.*

iv. P-mosfet

*Test these with test\_clock\_blink.ino sketch to make sure they work before soldering in! The front (where the black box sticks out) should point in the direction of the arrow on the circuit board.*

v. Battery pack

*Solder to ‘5V’ (red) and ‘GND’ (black) near the RTC and d1 mini.*

vi. Chip for soil moisture sensors (x2)

*Attach ~15 cm leads and cover each top pin with heat shrink before soldering to board. Do not insert completely into board—you want to be able to bend the chips back slightly to fit in the PVC housing. Remove the LEDs to save power.*

vii. Pins for jumper

*Use extra pins from microcontrollers or other components, cut to only 2 pins.*

viii. Resistors

*1k and 330k resistors-easiest to wire these on the back of the board and cover connections with a piece of electrical tape to avoid accidental shorts. Can wire on the front if careful…*

Soil moisture chip

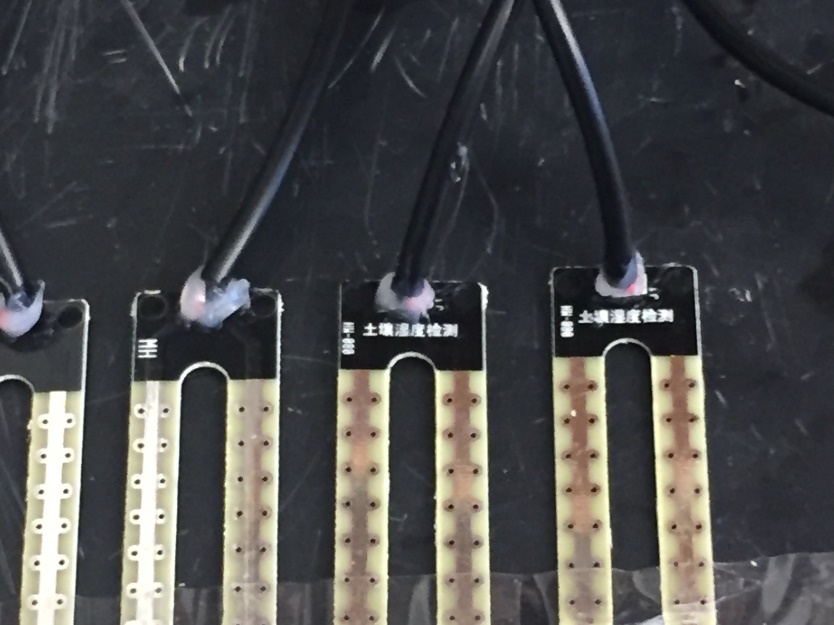
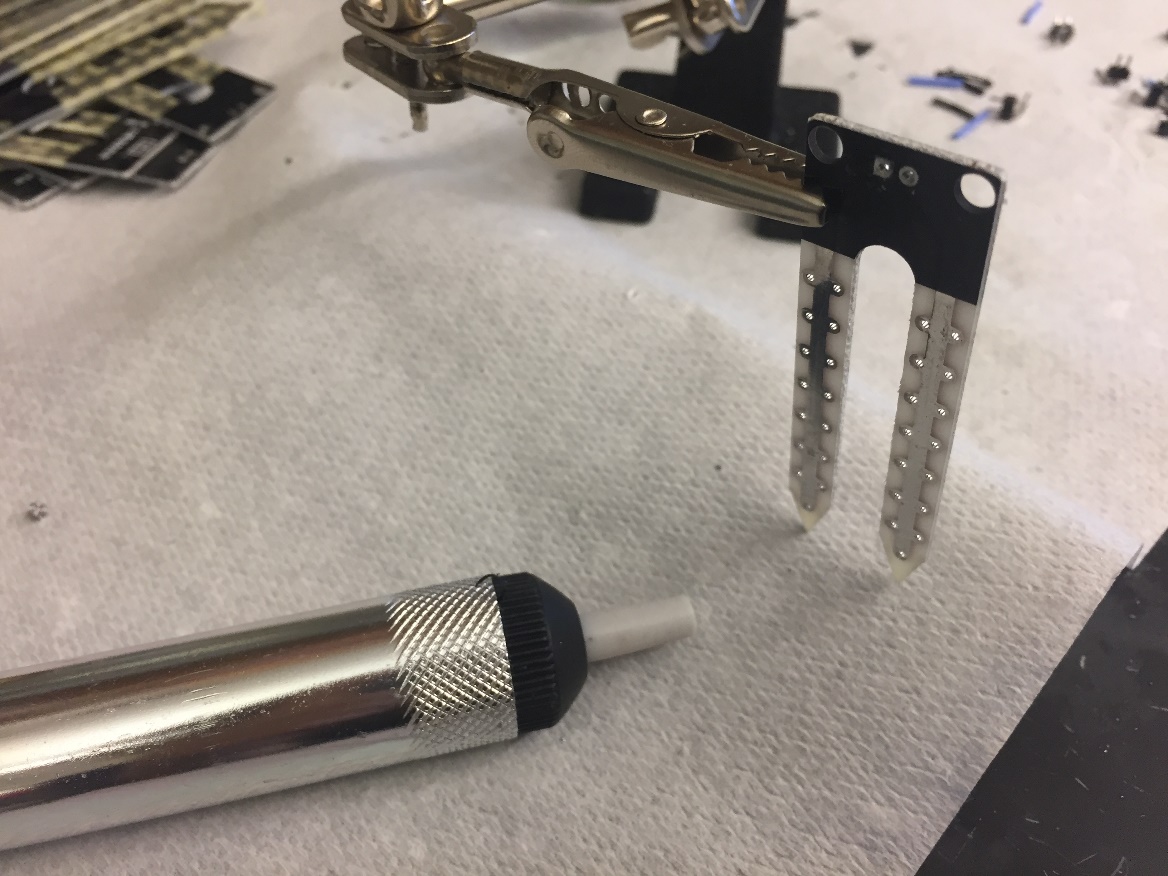
4. Sensors

iii. soil temp – ds18b20

*Cut wire to 50 cm length. Remove ~3cm of shielding and strip ~ 1cm of each wire.*

iv. soil moisture (x2)

*Remove pins, taking as much solder out of pin holes as possible with solder sucker. To make the shallow probe, cut excess wire from soil temp probe to 40 cm. Cut off excess yellow wire and solder red and black wires to probe; direction does not matter. Strip as little of the wire as possible. To make the deep probe, use 50cm of 2-strand 24awg wire. After soldering wires to probes, cover connection between wire and probe generously with 100% silicone and let dry at least overnight—this often requires touching up on the side the wire was resting on the next day. Paint modified silicone sealant over soldered connections on the other side. Remove ~2 cm of exterior shielding from the other end and strip 1-2cm.*



Prepping soil moisture sensor for wiring

Waterproofing soil moisture sensors

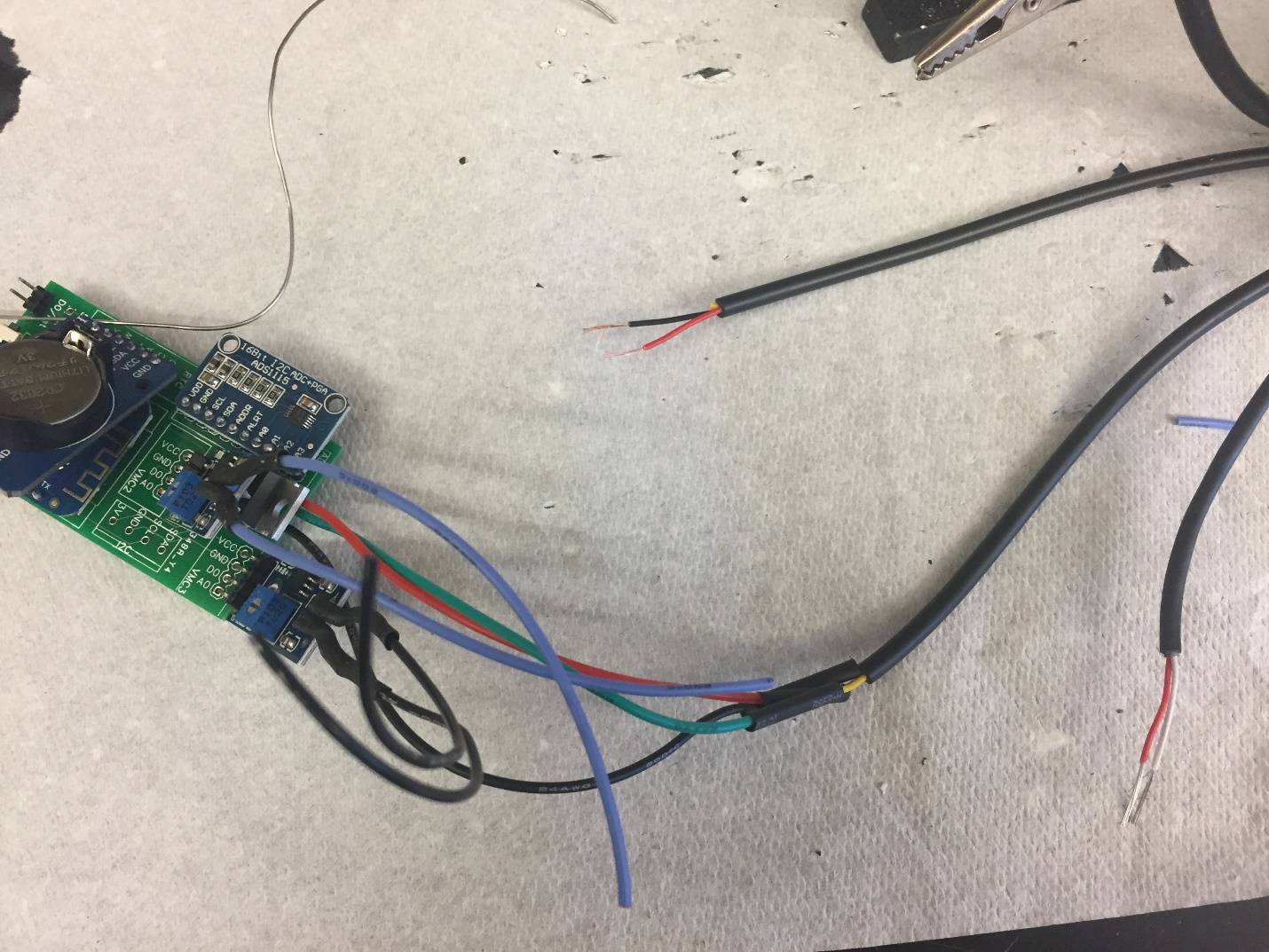
1. **Housing and assembly**

1. PVC and fittings – 2” FTP cap, 2” adapter, 2” x ½” bushing, PG13.5 cable gland

*Use PVC primer and cement to permanently attach adapter to bushing. When dry, coat cable gland threads with 100% silicone and tightly screw into bushing. Apply thin layer of 100% silicone around the connection between cable gland and PVC. Let dry again.*

1. Sensor attachment to circuit board

*Insert wires from soil moisture and temperature probes through cable gland. Solder the wires from the 50 cm long moisture probe to the chip in the ‘VMC2’ slot and the 40 cm long moisture probe to the chip in the ‘VMC3’ slot. Solder wires from temp probe to respective wires from ‘onewire’ slots (yellow- OneWire, red- 3V, black – gnd). Cover connections with heat shrink to avoid shorts. Solder wires for battery pack onto their connectors.*



Soldering sensors wires to prepared board

4. Waterproofing

*Position wires for soil sensors so that the stripped portion of wire is just inside of the PVC housing. Remove the outer screw cover of the cable gland and fill the inner ring around the wires with 100% silicone. Let dry at least 24h.*

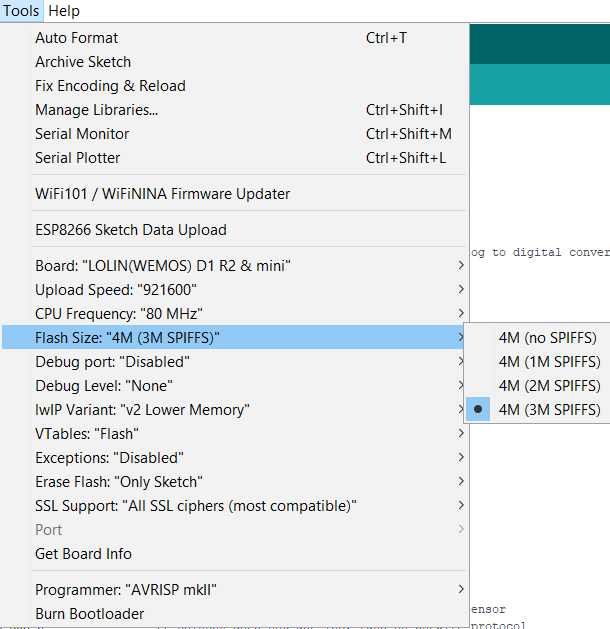


Nearly assembled sensors

1. **Testing, software and assembly**

1. Board and clock configuration: files at https://github.com/jdfridley/Sensors

*Run board\_config.ino with SPIFFS set to 3M to format SPIFFS and set the clock time.*

**

SPIFFS settings

2. Sensor testing

*Upload test\_sensor\_code.ino and check that all sensors read “good” and that the time is correct.*

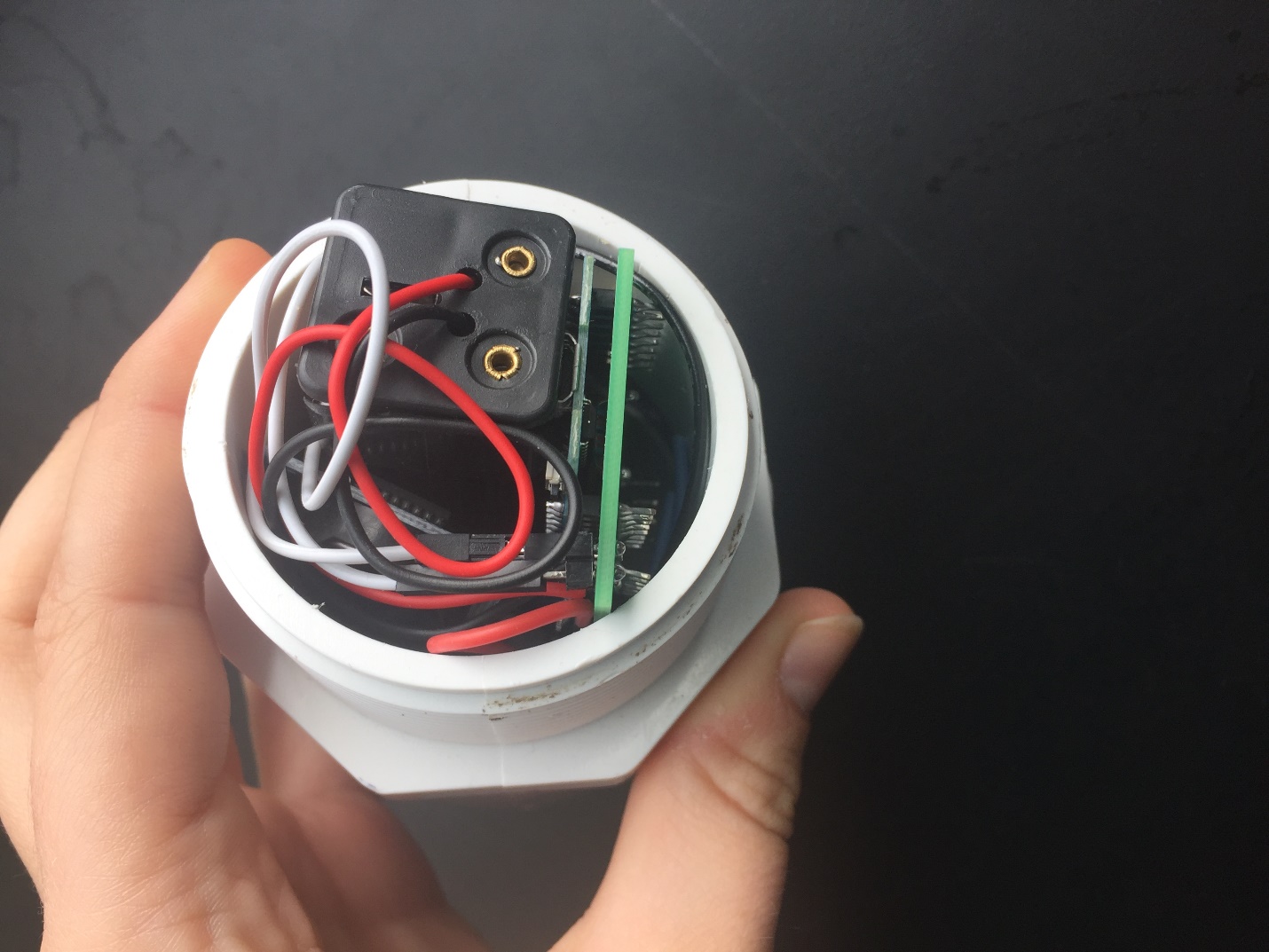
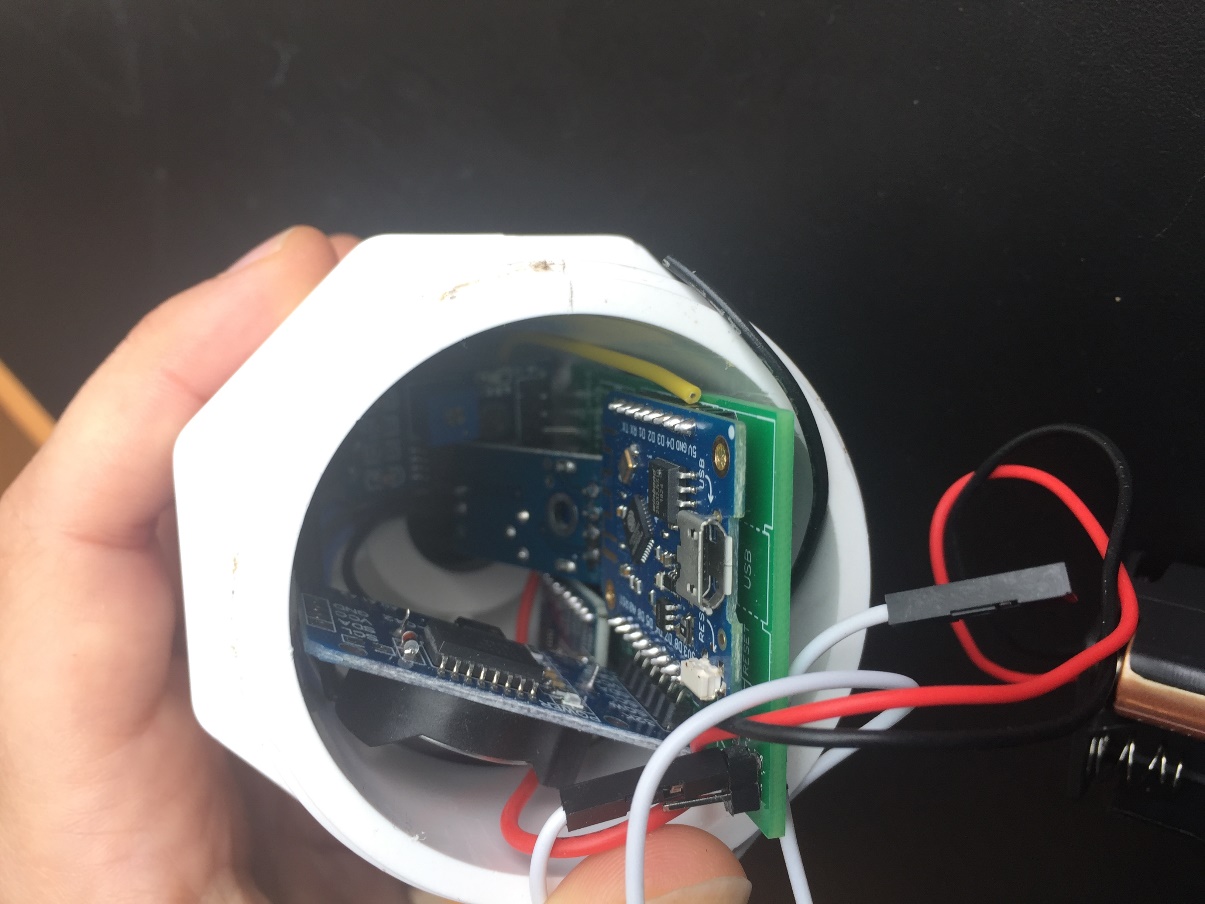
3. ESP8266 code

*Upload B\_sensor\_read\_dl.ino to microcontroller. Make sure to edit the sensor number! This code will write data from each sensor to SPIFFS every hour. Check back in several days to make sure all sensors are reading reasonable values (see ‘data download’ section below). For sensors with unusual values, check wiring/sensors/etc to determine cause. This program will continue reading after downloading data; to stop reading remove one or more batteries. DO NOT store using board\_config.ino program as this will drain batteries.*

1. Assembly

*Wrap battery pack in duct tape (electrical tape also works) to avoid damaging batteries by scraping against electrical components. Place battery pack between clock and D1 mini and insert all into PVC housing (it is a tight fit!). Place jumper over pins on circuit board. Wrap threads with PTFE tape. Place two silica gel packets in housing and seal tightly.*

Circuit board inside housing



Circuit board + battery pack inside housing

1. **Field procedure**

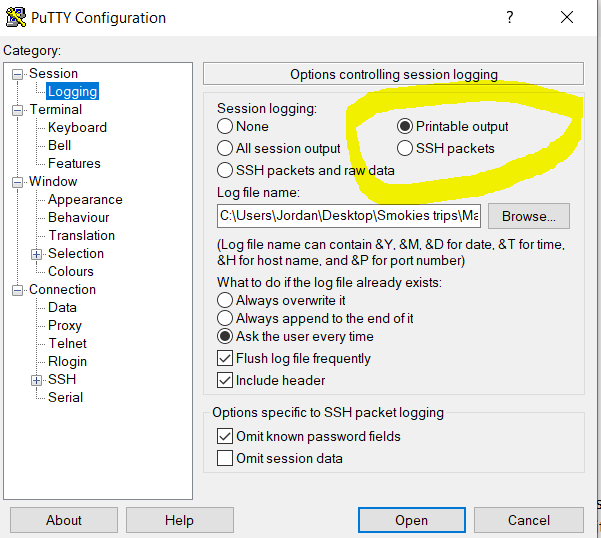
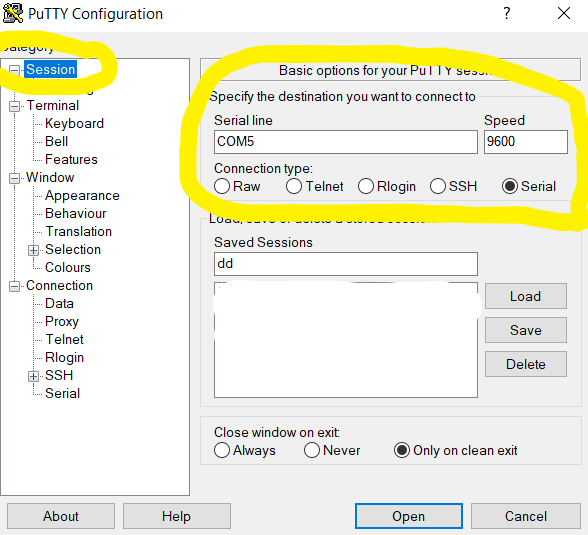
1. Sensor installation

*Use tulip bulb planter to create a hole slightly deeper than sensor housing and place sensor housing tightly in hole. Install soil moisture sensors upslope and to the left, with wires in a shallow trench leading back to housing. Pack soil back around housing and cover with removed leaves. Flag trees on either side of the sensor and note position between trees (ie center, or 3 feet from right tree looking upslope). Sensors are easily detected for data retrieval using a metal detector. Sensors installed in October 2019 were still reading and had battery levels >5.5V in March 2020.*

1. **Data download and processing**

1. Downloading data with PuTTY

*To download data, remove the jumper, plug microcontroller into computer and save serial input using PuTTY (input will start as soon as the microcontroller is plugged in). Files are saved as a raw text file (you can use the .txt extension to open in notepad or another text program).*

**

PuTTY configuration for data download (you may be using a different COM port)

2. Converting raw files to .csv (in R)

*For anything but a very small dataset, it is most convenient to process the files into a more standard format that can be read into R or other data analysis programs. To do this, place all of the raw data files in an otherwise empty folder. In a separate folder, save the calibration coefficients (calib\_coefs.csv) that convert raw readings from the soil moisture probes and soil temperature probe to volumetric soil moisture. You can then run the R script ‘ReadBSensors.R’; edit the ‘data\_path’ at the top to the folder with sensor files in it, the ‘Metadata\_path’ to the folder with calib\_coefs.csv in it, and the ‘Out\_path’ to the folder where you would like to save the finished .csv file with all of the data. The file will be saved as ‘sensordata.csv’ unless you edit the last line of code to include some other name. Note that this script assumes the temperature is the same for both of the soil moisture probes and the soil temperature probe; if they are installed at different depths you may want to calculate a temperature correction. Additional R scripts for removing dates when sensors were not deployed in the field and identifying malfunctioning sensors are available at https://github.com/jordanstark/Soil\_temp\_moisture\_EMU/DataCleaning*

**Useful links and resources**

The components and connections in this design are based largely on **Mickley, J. G., T. E. Moore, C. D. Schlichting, A. L. DeRobertis, E. Mason, & R. Bagchi (2018). Measuring Microenvironments for Global Change: DIY Environmental Microcontroller Units (EMUs). Methods in Ecology and Evolution. 1-7.** [**doi: 10.1111/2041-210X.13128**](http://dx.doi.org/10.1111/2041-210X.13128) **.**

**In addition to this paper, Mickley et al have a fantastic github resource for EMUs with troubleshooting and calibration information:** <https://github.com/mickley/EMU>

**Many thanks to Fernanda Santos at Oak Ridge National Lab for comments on this protocol.**