Pedestrian Tracking System Sensing Node Assembling Manual

Author: Cheyu Lin

Contact: cheyul@andrew.cmu.edu

Index

Introduction

Passive infrared (PIR) sensors are motion sensors that detect IR stimuli emitted by heating bodies and convert them into electrical outputs. They are used in a variety of applications, such as home automation, surveillance systems, and human trajectory analysis since they are low-cost, reliable and anonymous compared to other sensors like camera and Bluetooth tags, which are relatively expensive and privacy-invasive.

PIR sensors are categorized into digital and analog sensors, and they both have their advantages and disadvantages. Digital PIR sensors are the more common option that outputs 1 when motions are detected and 0 otherwise. A considerable number of applications leverage their binary nature since binary outputs require less effort to process and off-the-shelf digital PIR sensors are easy to access due to its prevalence. However, the digital characteristics also narrow the information availablility to users – digital PIR sensors can reflect nothing more than the presence of motions. Whereas its analog counterpart can not only sense motions, but also provide direction, velocity, and distance of pedestrians relative to the sensor by giving different outputs, which requires multiple digital PIR sensors to work synergistically to collect the same information. As informative as analog PIR sensors can be, they require additional amplification since the raw analog outputs are not prominent enough to be observed and processed directly by microcontrollers. Despite the implication of an additional amplifier, we still prefer analog PIR sensors since they are not only more informative, but they are also more energy-efficient in contrast to the digital counterpart when requiring the same capabilities.

To enable more people to benefit from analog PIR sensors, we designed a sensing node that pivots around an analog PIR sensor and included a GPS, SD card module, Arduino UNO, a divided cone, and a power supply system. In the sensing node, we have a plastic circuit board (PCB) that harbors the amplifier for the analog output, SD card module for data storage, and GPS capability to document information captured by the PIR sensor chronologically. The PCB also serves as a shield that sits atop an Arduino UNO so that users can expand the functionalities of the sensing node of their own accord. Additionally, we also devised a divided cone that increases the sensitivity of the sensor. All components reside in a water-proof enclosure and are powered by a battery, which is sustainably charged by a solar panel.

In this manual, we provide step-by-step instructions pictorially to construct the sensing node, which entails tools required to build the sensing node, the process of soldering components onto the PCB, assembling the cone, creating water-proof connections that are exposed, and putting everything together. To download design files for the PCB and the cone, code, and bill of materials, please consult the GitHub page.

Tools Required to Build the Sensing Node

Fig. 1 and 2 show all the tools needed to assemble the sensing node.

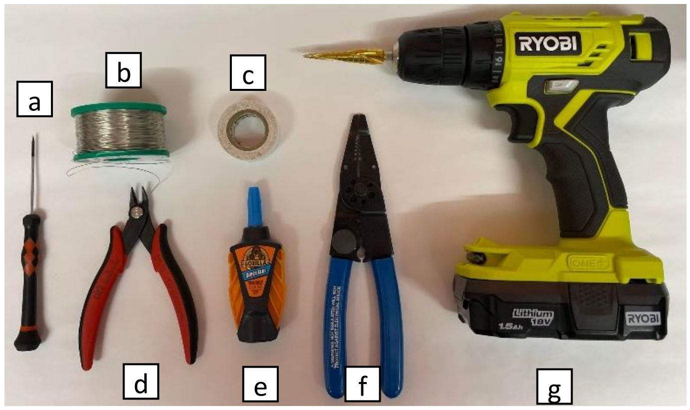


Fig. 1. (a) Flat-head screwdriver (b) solder wire (c) electrical tape (d) wire cutter (e) super glue (f) wire strip (g) driller

Fig. 2. Solder station and solder iron

PCB Schematic

In this section, we have the schematic of the Arduino shield that is comprised of external power supply, Arduino power supply, GPS and SD card module connections, and PIR signal amplifier in Fig. 2. The external power supply is connected to a 12-V battery to power Arduino, which subsequently powers the operation of the rest of the circuit through the 5-V pin in the Arduino power supply. GPS and SD card module are connected to the shield through "JP" labelled solder pads. As for the PIR signal amplifier, it processes the raw PIR signal by sending the signal through a non-inverting and an inverting band-pass filter, made up of operational amplifiers, resistors, and capacitors, and feeding the output to Arduino's analog pin A1. As a side note, "JP" labelled solder pads named "Arduino spare pins" are for adding external components according to user's needs. For instance, if an LED light is required, one can connect it to pin 5 in JP6, and control it through Arduino's digital pin 5.

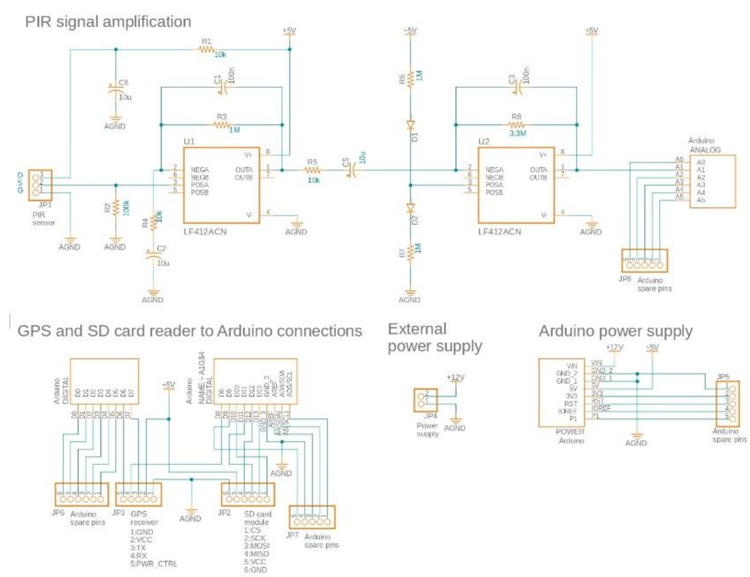
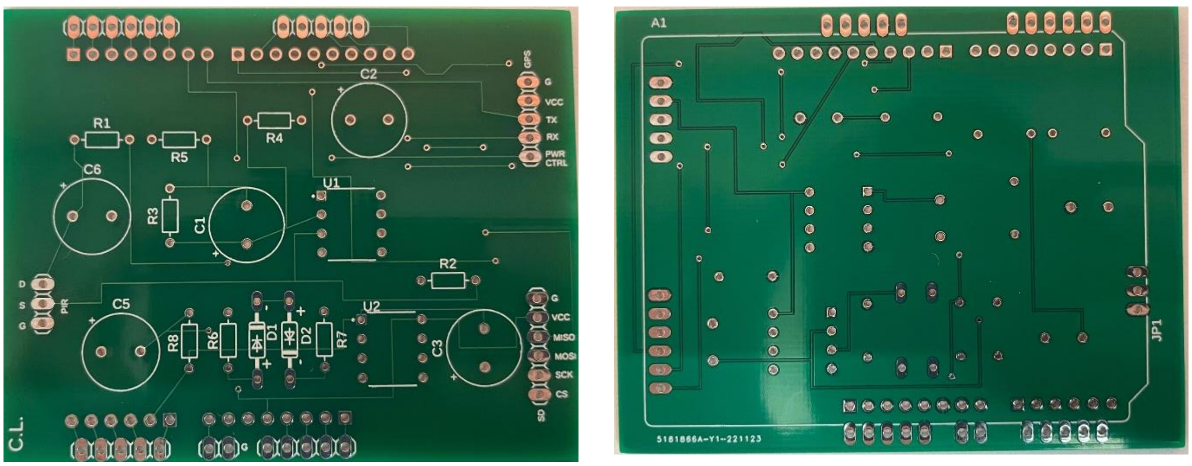


Fig. 2. Schematic of the Arduino shield that includes amplifier for raw PIR signal, GPS, and SD card module.

Soldering Arduino Shield

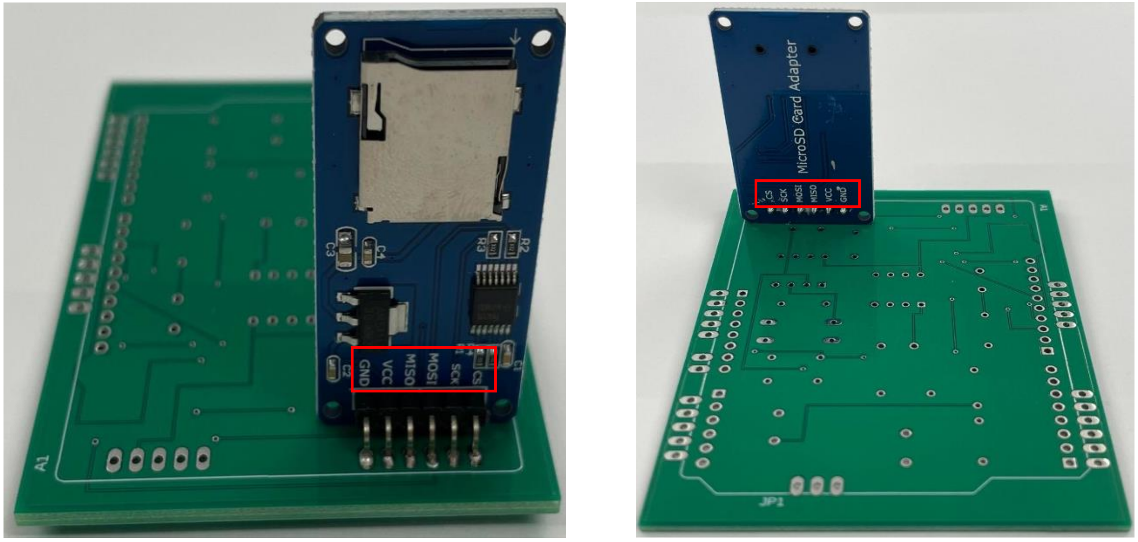
1. Download Gerber files from GitHub page under main -> Design files -> PCB Gerber files. Print the PCB from the PCB factory of user’s preference.
2. Purchase components listed in the bill of materials (also accessible in GitHub page). If user has your own preferences, please ensure that components bought line up with solder pads on the PCB.
3. Identify the front and back sides in Fig. 3 before soldering since some components belong to the front side and others belong the back side.



(a) (b)

Fig. 3. (a) The front side of the PCB has resistors, capacitors, diodes, and op-amps labelled as “R”, “C”, “D”, and “U” respectively. The “+” signs indicate positive polarity. (b) The back side of the PCB.

4. Solder the SD card module onto the back of the PCB by connecting module’s pins to their corresponding holes as shown in Fig. 4.
5. Solder the short end of male headers onto the front side of the PCB as in Fig. 5. One tip is to insert the long end of the headers into an Arduino and place the PCB on top to ensure headers are not lopsided when soldered. A caveat is that the long end of headers must be long enough to leave clearance when fitting PCB onto Arduino. See Fig 13 for visual explanation.
6. Solder op-amps onto the front side of the PCB with the dented pit on the op-amp aligning the white dot on the PCB as shown in Fig. 6.



(a) (b)

Fig. 4. SD card module soldered onto the PCB following the (a) module's front side facing outward and (b) back side facing inward.

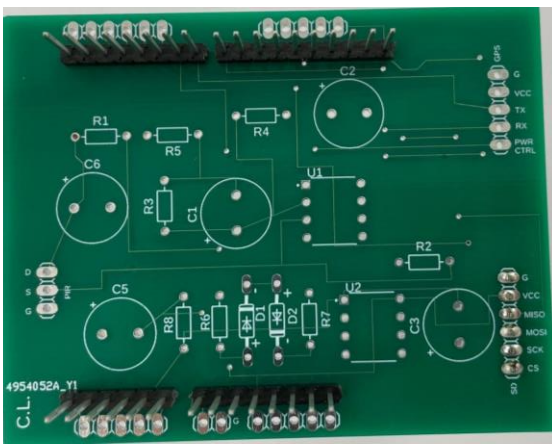


Fig. 5. Male headers soldered onto the PCB

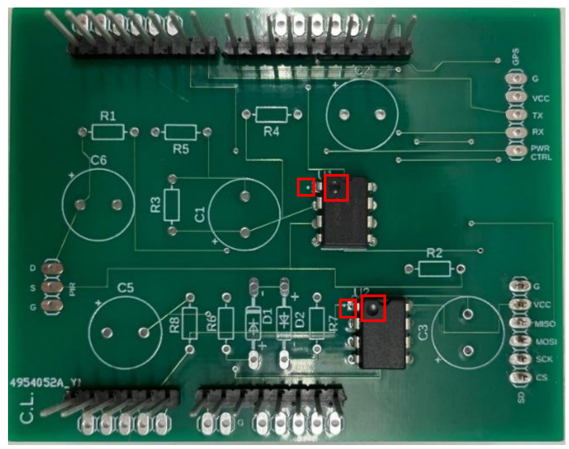


Fig. 6. Orient op-amps by matching framed white dots to pits.

7. Resistors are on the back side of the PCB. Solder 10k resistors to R1, R4, and R5; 100k to R2; 1M to R3, R6, and R7; 3.3M to R8. See Fig. 3 and 7 for reference.

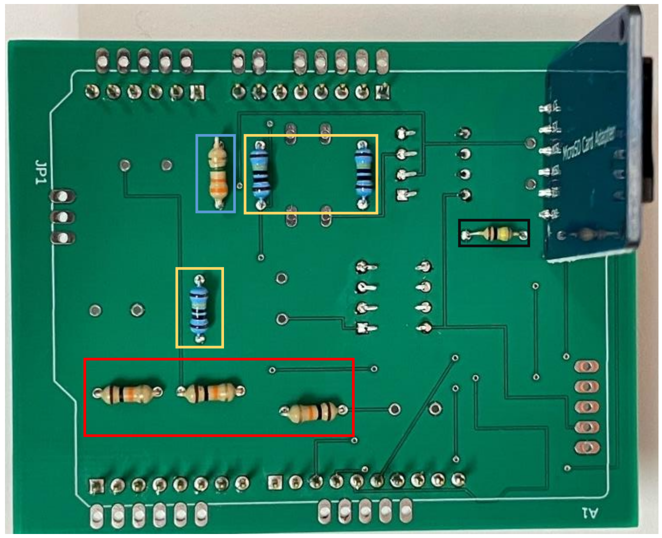


Fig. 7. Resistors are framed with different colors to indicate different resistances. 10k resistors are red. 100k resistor is black. 1M resistors are boxed yellow. 3.3M resistor is boxed blue.

8. Connect polarized capacitors on the back of the PCB with the white strip on the capacitor lining up with the negative sign on the PCB. Please refer to Fig. 3 and 8 to identify the polarities of capacitors and the corresponding solder pads on the PCB.

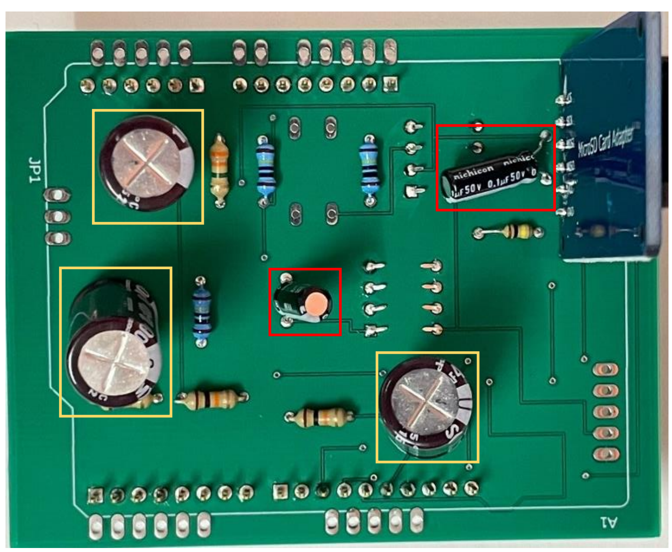


Fig. 8. 100n capacitors are boxed red, whereas 10u capacitors are boxed yellow.

9. Solder female headers onto the back of the PCB as shown in Fig. 9.

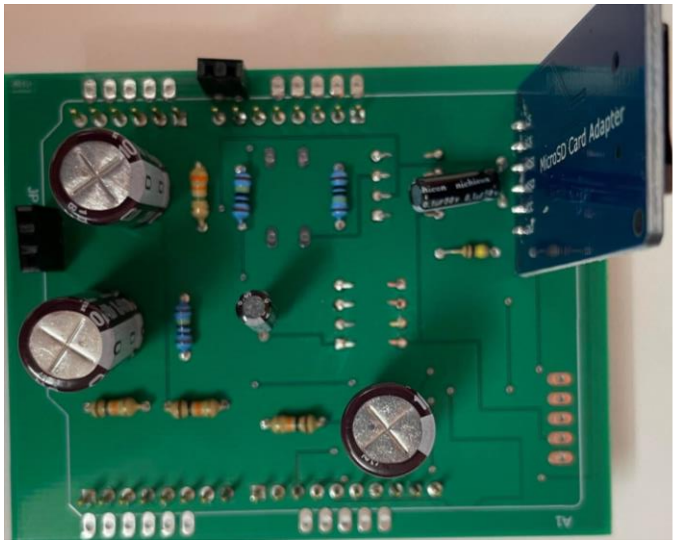


Fig. 9. The two-position header is for power supply, and the three-position header is for PIR sensor.

10. Solder diodes onto the back side of the PCB with the black-strip end connecting to the negative pin on the PCB. Please refer to Fig. 3 and 10 to identify the polarities of diodes and the PCB.

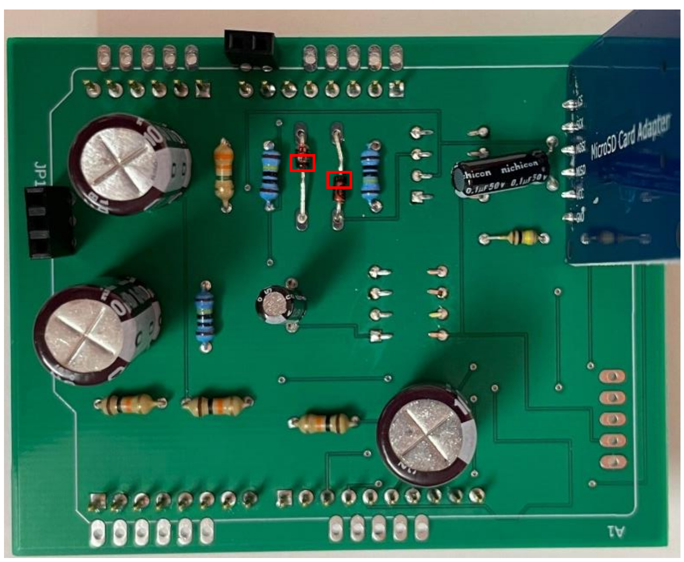


Fig. 10. The black strip that indicates negative polarity of diodes are framed in red boxes.

11. Solder terminal blocks on the back the PCB to connect GPS more easily as in Fig. 11.
12. In Fig. 12, excess leads are snipped with a wire cutter to prevent the circuit from malfunctioning by touching each other accidentally.
13. Connect the shield to Arduino following Fig. 13.

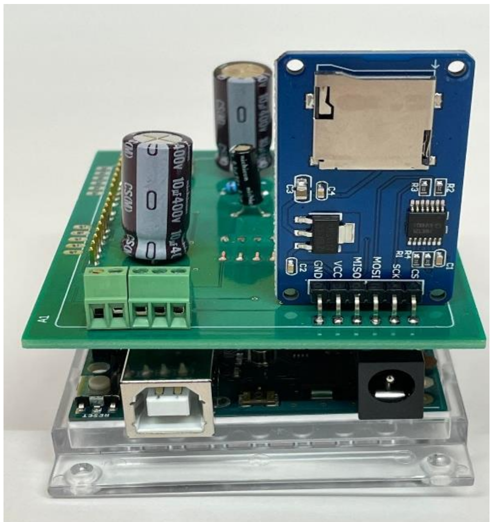


Fig. 11. Notice that the outlets face outward to connect GPS more easily.

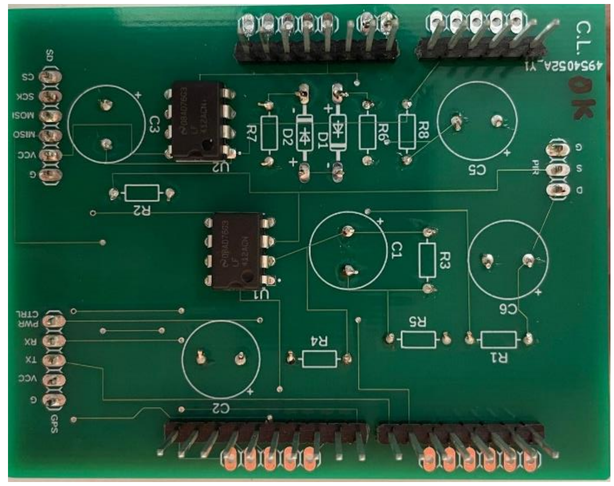
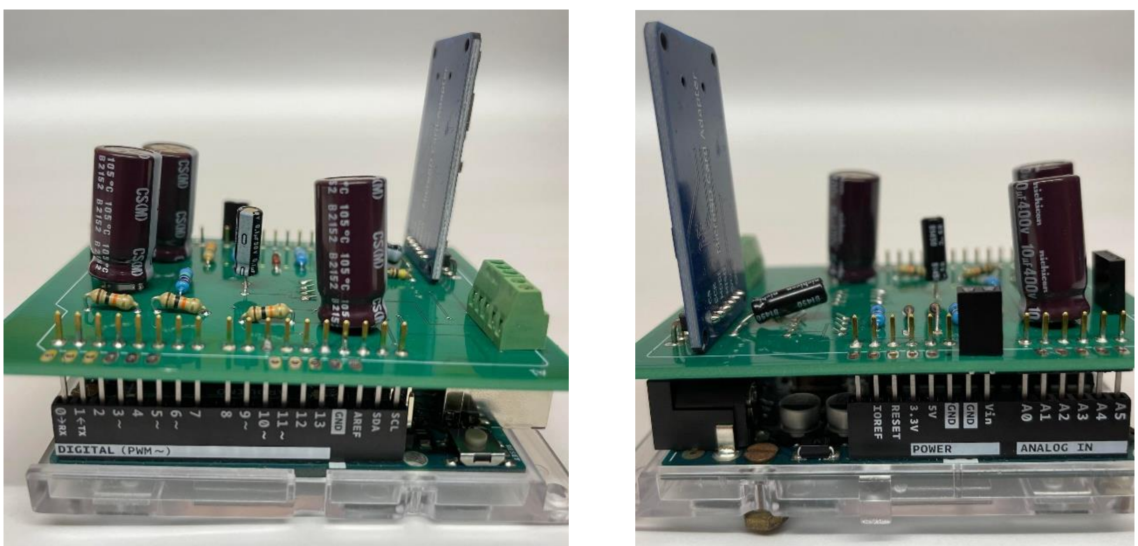


Fig. 12. Only diodes, resistors, and capacitors' leads are cut off.



(a) (b)

Fig. 13. There is space left between the shield and the Arduino to prevent them from touching each other.

14. Connect GPS to the terminal blocks and insert a micros SD card into the SD card module.
15. See Fig. 15 for the completed result.

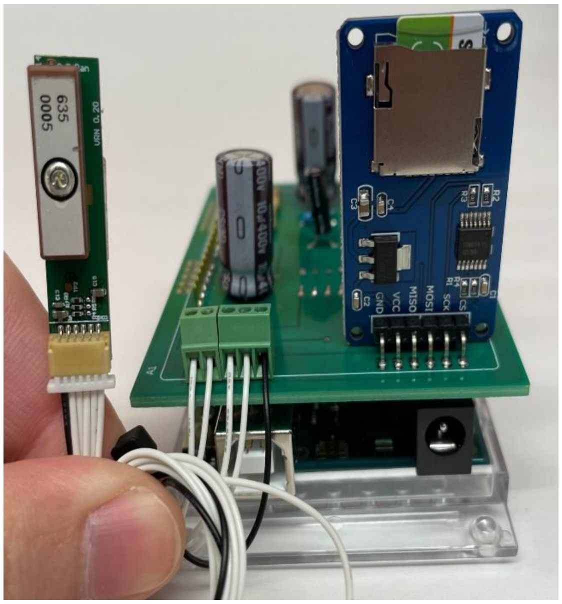


Fig. 14. Insert wires into the terminal blocks and tighten them to connect GPS.

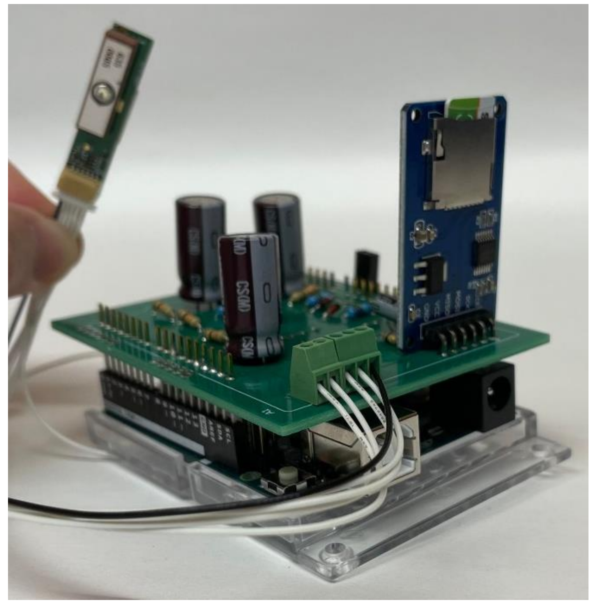
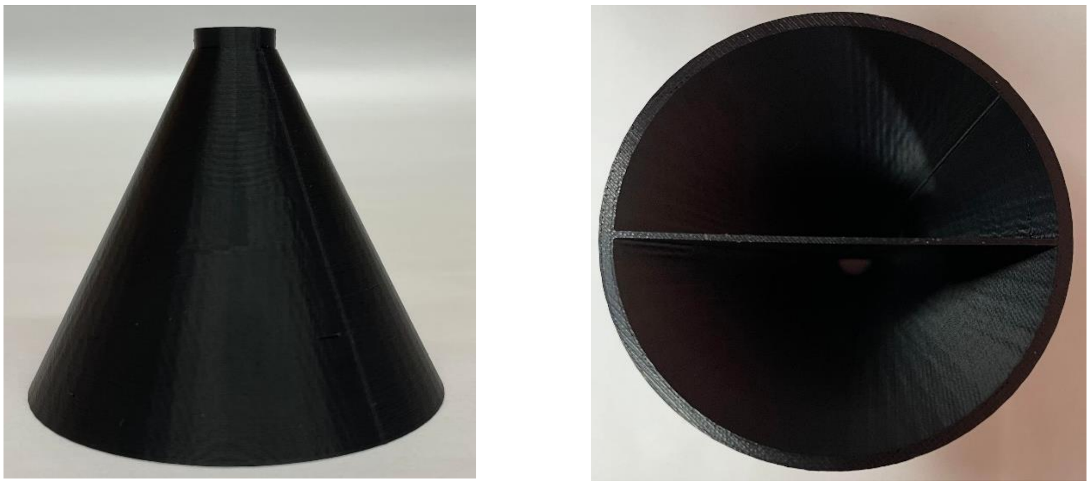


Fig. 15. Completed! Be proud!

Assembling PIR Cone and Tube

1. Download stl files from GitHub page under main -> Design files -> Cone and tube. Print the cone and tube from a 3D printing machine. The material we used is acrylonitrile butadiene styrene (ABS) filament.
2. Please refer to Fig. 16 for different perspectives of the PIR cone.



(a) (b)

Fig. 16. (a) Side view and (b) bottom view of the cone.

3. Insert the PIR sensor into the cone with the protuberance oriented vertically with the divider in the cone. Apply a layer of super glue over the PIR sensor. Connect jumper wires to the sensor and crimp the headers to tighten the connections. See Fig. 17.
4. Cover the tube over and glue it to the cone with super glue as in Fig. 18.

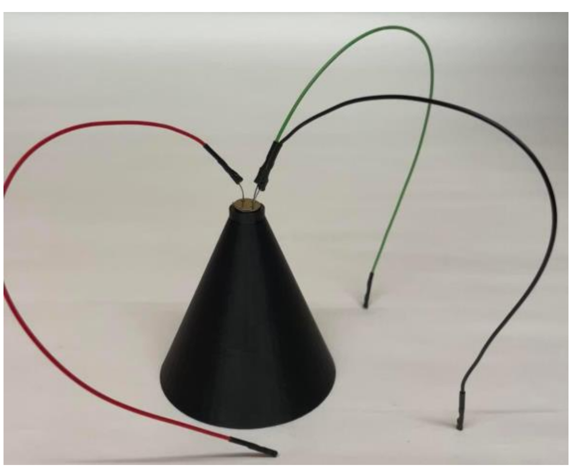


Fig. 17. The wires here are color-coded, where red represents drain, green connects to source, and black connects to ground.

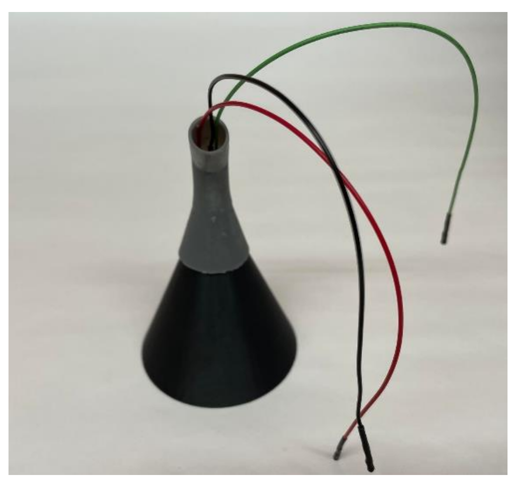
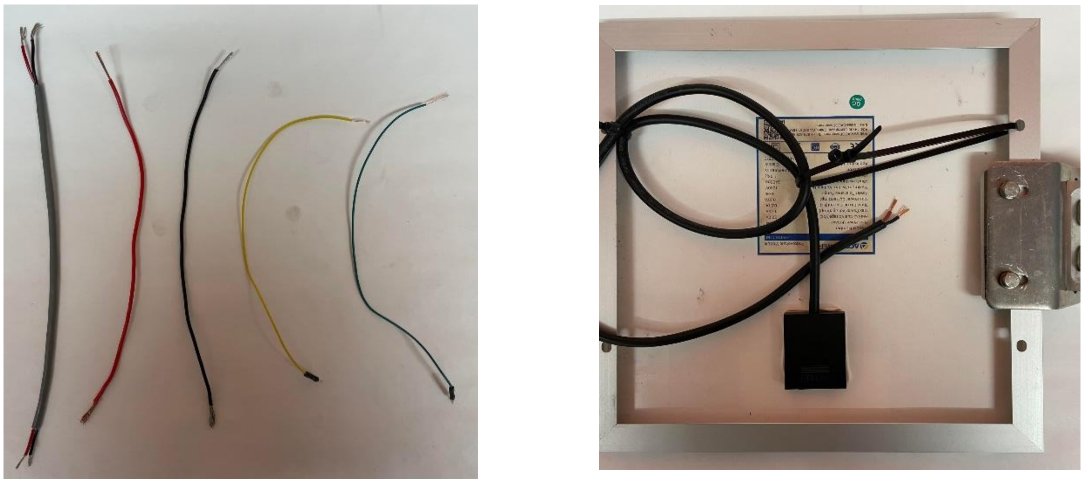


Fig. 18. Finished product!

Putting Everything Together

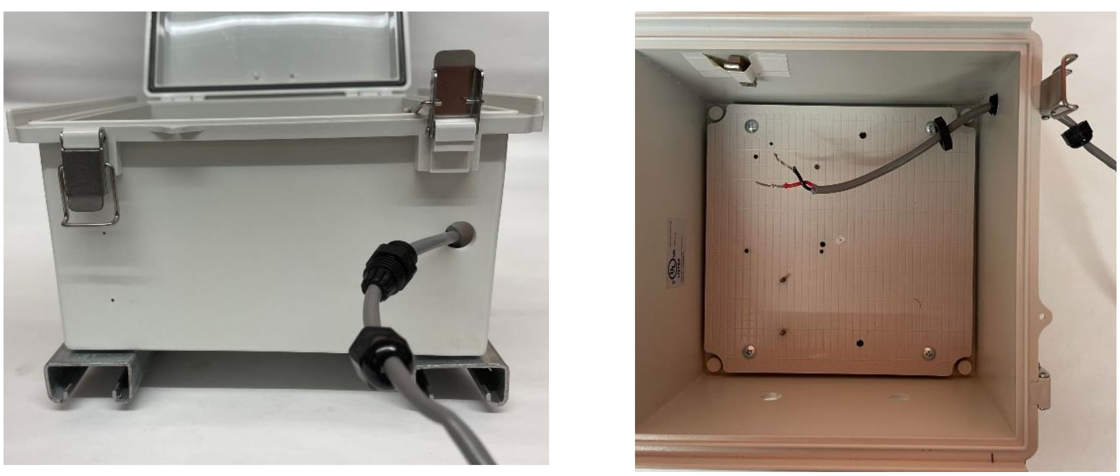
1. Peel and strip wires using wire cutter and wire strip as shown in Fig. 16 for later use.



(a) (b)

Fig. 16. (a) From left to right we have a gray cable to connect the solar controller to solar panel, two 12-AWG cable to connect solar controller to the battery, and two jumper wires to connect the solar controller to the shield. (b) Stripped wires behind solar panel.

2. Drill a hole of 0.5-inch diameter and pull the stripped gray cable in Fig. 19 (a) through the cable gland and feed it through the hole into the box as shown in Fig 20.



(a) (b)

Fig. 20. (a) Side view of the cable pulled through cable gland and into the box. (b) Top view of the cable. The cable gland is unscrewed into three pieces to demonstrate the relative position of each piece. Please tighten them after the cable is pulled into the box.

3. In Fig. 21, the two wires in the middle are fed into female connection receptacle and crimped. Whereas the two outer cables are clenched by nickel crimps and pushed into connection receptacle and connection plug until clicked.
4. Connect the wires to solar controller by wrapping the metal around the screws (as demonstrated with the left most cable in Fig. 22) and tighten them.

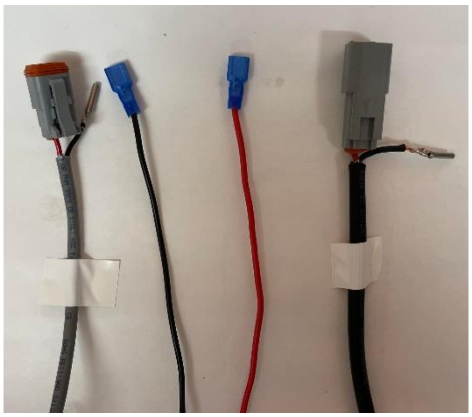


Fig. 21. Two outer cables should be held tight by nickel crimps that are left out for demonstration purposes before pushing into the receptacle and the plug.

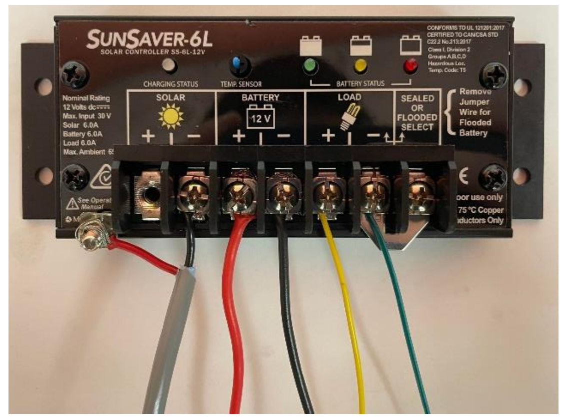


Fig. 22. Please remember to screw the left most wire into the solar controller.

5. Drill 0.5-inch diameter hole at the bottom of the box and insert the PIR cone made in the previous section as demonstrated in Fig. 23.

Fig. 23. The white label on the cone is used to mark orientation of the PIR sensor, which corresponds to the side of the PIR sensor without the protuberance.

6. Place the solar controller, battery, and Arduino into the box and connect everything!
7. Connect the solar controller with the solar panel!

Fig. 24. Components can be fixated using zip ties and screws to prevent wobbling during transportation.

Fig. 25. Great job!