|  |
| --- |
| Vooji Designs |
| Open Source Sound-Signal Device |
| Voojibox |

|  |
| --- |
| Ed Wojtaszek dba Vooji Designs  3/9/2014  Rev - |

Voojibox

Open Source Sound-Signal Device

This document is provided for your personal use or for reuse under [Creative Commons BY-NC-SA 3.0](http://creativecommons.org/licenses/by-nc-sa/3.0/).

# Introduction

I designed this device as an inexpensive way for a sailboat racing club to enjoy the benefits of an automated sound signal. It uses the signaling prescribed by U.S. Sailing, *The Racing Rules of Sailing 2013 – 2016* (RRS). The parts are commonly available electrical and mechanical components. The open source Arduino controller is a version designed by Sparkfun Electronics®. The open source Arduino software is provided by me under the provisions of the GNU General Public License, version 3 (GPL-3.0) as given in Appendix A. If you use the design as-is, the parts cost is between $200 and $250 (2013 dollars). The hardware and documentation are provided under [Creative Commons BY-NC-SA 3.0](http://creativecommons.org/licenses/by-nc-sa/3.0/). The software and documentation source can be obtained on the [GitHub repository](https://github.com/votsek/voojibox.git).

I would say that this is not a beginner project. Soldering skills are required to assemble the circuit board and you will need a good soldering station. Use of a drill press is recommended for making holes in the plastic and aluminum parts because better control of the drill bit and drill feed speed can be achieved. If something doesn’t work, debug of the assemblies may be needed. If you can’t use the same tooling or techniques that I use, you may need to improvise. Some knowledge of the c programming language and real time programming are needed. Overall the design isn’t complicated, but there are still many things that can go wrong and you don’t want to waste the money.

# Design

The device is designed to be portable and to operate on a stand-alone basis. It contains a 12v sealed lead acid battery that is capable of many race starts before recharging is necessary. The standard design incorporates an air horn on a base with a 15 foot cord. The cord allows placement of the horn away from race management personnel. An optional design uses the committee boat power and horn. This adds an external power cable and changes the box internal wiring slightly. This option is not covered in detail in this document. This cost of this box option is not significantly different cost from the standard box. The option eliminates the cost of the horn system.

The user controls on the signal box are the power toggle, Horn button, Mode button, and the Start button. The only other external user interfaces are the coaxial power jack used to connect the battery charger and the horn connector. When the power toggle is switched to the “ON” position, the Arduino is powered-up, waiting for either the Mode button interrupt of the Start button interrupt. A green LED indicates that the Arduino has power. The momentary Mode button is used to select the specific RRS start signal sequence. All three are programmed into the device: Appendix S, Sound-Signal Starting System; Rule 26; and Rule 29.2. Rolling start variants of each are also programmed. The red LED shows the selection as one, two, or three blinks, respectively. The rolling variants are indicated by the yellow LED, which is illuminated every other button push. For example, a Rule 26 rolling start would be indicated by the yellow light and two blinks of the red LED. The most recently used mode is “remembered” when the device is powered off. When the device is powered on, the previously used mode is indicated by the appropriate number of red LED blinks as well as the yellow LED rolling start indicator.

The unit has a special signal sequence that is used for handicap racing. This is the pursuit race signal, indicated by the yellow light and four red flashes. The first three minutes use RRS Appendix S. Each subsequent minute repeats the final minute of the Appendix S sequence. This is used to start boats in order of their handicap, slowest boats to the fastest. Each fleet is assigned a time slot. The race committee can use fleet flags to indicate the next fleet to start. There can be start signals where there is no starting fleet so that gaps in the handicap can be accommodated. This method permits the race committee to score handicapped boats in the finishing order rather than recording start and finish times.

The standard programmed start sequences are summarized in Table 1.

The momentary Start button starts the signal sequence. The Mode button is disabled until the power is cycled off and on. Before the signal sequence begins, the internal piezo buzzer will beep five times in one second intervals, the unit gives five short blasts on the horn to warn the racers that the starting sequence is about to begin, and the piezo buzzer will beep five more times in one second intervals. The red LED will be illuminated during the final one minute interval before the start. The rolling starting signal sequence can only be halted by switching the device “OFF” using the power toggle.

If the rolling start is not being used, another start sequence can be initiated using the momentary Start switch after the Starting signal. It is necessary to turn the device “OFF” and back “ON” to change the mode. It is not necessary to turn the device “OFF” and back “ON” to start a new signal sequence. The device will not charge unless the power switch is “OFF”.

During the signaling, the red LED is illuminated during the final minute of the sequence as an aid for the race committee. Also, for modes 2, 3, 4, and 5 there are three internal beeps during the final fifteen seconds prior to each signal event. This is a race committee aid as well to be used as an audible cue for procedural race committee actions such as flag raising or lowering.

The Horn switch bypasses the control logic and the Arduino. It is designed to be used to hail sailors for recalls and other purposes determined by the race committee. It is active as long as the power is “ON”.

The heart of the device is the Arduino, supported by a simple shield circuit that interfaces with the device active components: horn, Horn button, Mode button, Start button, and the three LEDs. Power is provided by a 12v lead acid battery, which should stay charged through as many as 20 start cycles using a 15 ampere horn. The power is routed using a 6-position terminal block.

|  |  |  |  |
| --- | --- | --- | --- |
| ***Table 1 – Voojibox Modes and LED Indicators*** | | | |
| **Mode** | **Selected Sequence** | **Red**  **Blinks** | **Yellow** |
| 0 | The starting sequence is three minutes long. The sequence is as specified in Appendix S, Sound-Signal Starting System, The Racing Rules of Sailing for 2013 - 2016, U.S. Sailing. A long signal is made at (t-0) to signal the start of the race. | 1 | OFF |
| 1 | Same as Mode 0, but the sequence repeats. The start of the three minute interval for fleets subsequent to the first is the long start signal at (t-0). | 1 | ON |
| 2 | Uses the Racing Rules of Sailing Rule 26 signaling. Sound signals are made at (t-5), (t-4), (t-1), and at the start (t-0). | 2 | OFF |
| 3 | The Mode 2 sequence repeats with the start of the five minute interval (t-5) being the start signal of the previous fleet at (t-0). | 2 | ON |
| 4 | Uses the Racing Rules of Sailing Rule 29.2. The first sound signal is made at (t-6), consisting of two long blasts. At (t-5), signaling reverts to Rule 26. | 3 | OFF |
| 5 | This rolling variant of Mode 4 uses the signaling of Mode 4 for the first start. After the first start signal at (t-0), the signaling reverts to Mode 3. That is, the start of the five minute interval (t-5) is the start signal of the previous fleet at (t-0). | 3 | ON |
| 6 | This is a special rolling mode called 3 x 1 Appendix S signaling. Appendix S signaling is used to start the first fleet as in Mode 0. Subsequent starts are signaled using the final one minute sequence of Appendix S. This is useful for pursuit races, which is an alternative to handicap races. | 4 | ON |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Table 2 – Voojibox Parts List*** | | | | | | |
| **Source** | **Qty** | **Name** | **Part Number** | | **Notes** | |
| [www.batteryspace.com](http://www.batteryspace.com/) | 1 | SLA Battery | LA-12V3.4 | | Use only battery chargers designed for use with sealed lead acid batteries | |
| [www.batteryspace.com](http://www.batteryspace.com/) | 1 | Lead Acid Smart Charger (0.8A) for 12v Lead Acid | CH-LA1208 | |  | |
| [www.westmarine.com](http://www.westmarine.com/) | 1 | Plano 1460 Guide Series Waterproof Case | 11520012 | |  | |
| [www.sparkfun.com](http://www.sparkfun.com/) | 1 | Arduino Pro328 - 5v/16MHz | DEV-10915 | |  | |
| [www.sparkfun.com](http://www.sparkfun.com/) | 1 | Proto Shield | DEV-07914 | |  | |
| [www.sparkfun.com](http://www.sparkfun.com/) | 1 | Diode Rectifier - 1A 50V | COM-08589 | | High surge capacity diode - MOSFET reverse voltage protection | |
| [www.sparkfun.com](http://www.sparkfun.com/) | 1 | N-Channel MOSFET 60V 30A | COM-10213 | | Claxon switching | |
| [www.sparkfun.com](http://www.sparkfun.com/) | 1 | Jumper Wire - JST Black Red | PRT-09914 | | Arduino power (JST = Japan Solderless Terminals) | |
| [www.sparkfun.com](http://www.sparkfun.com/) | 1 | Standoffs Plastic - 3/8" (10 pack) | PRT-10461 | | Mount Arduino to project box | |
| [www.sparkfun.com](http://www.sparkfun.com/) | 2 | 10K resistor | COM-08374 | |  | |
| [www.sparkfun.com](http://www.sparkfun.com/) | 1 | Screw - Phillips Head (1/4", 4-40, 10 pack) | PRT-10453 | | Mount Arduino to standoffs | |
| <www.digikey.com> | 1 | Power toggle switch - DPDT | EG2403-ND | | Switch enables charge circuit when off | |
| <www.digikey.com> | 1 | 5.5 x 2.5 mm DC Power Jack | SC1048-ND | |  | |
| <www.digikey.com> | 1 | 5.5 x.2.5 mm DC Power Plug | SC1051-ND | |  | |
| <www.digikey.com> | 15 | #22 snap socket | 94838-01-ND | | Connect low power and signal wire | |
| <www.digikey.com> | 15 | #22 snap plug | 94840-01-ND | | Connect low power and signal wire | |
| <www.digikey.com> | 13 | 6 stud #22 spade connector | 94769-01-ND | | Connect wire to terminal block | |
| <www.digikey.com> | 6 | 6 stud #14 spade connector | 94775-01-ND | | Connect wire to terminal block | |
| <www.digikey.com> | 4 | #14 x .25 quick connect | 94820-01-ND | | Connect wire to battery and relay | |
| <www.digikey.com> | 2 | #22 x .25 quick connect | 94803-01-ND | | Connect wire to relay | |
| [www.digikey.com](http://www.digikey.com/) | 1 | 6-Pole Panel Mount Terminal Block | WM5794-ND | | Power connections | |
| [www.digikey.com](http://www.digikey.com/) | 2 | Red momentary switch | GH1368-ND | | Mode and Horn switches | |
| [www.digikey.com](http://www.digikey.com/) | 1 | Black momentary switch | GH1367-ND | | Start switch | |
| [www.digikey.com](http://www.digikey.com/) | 1 | Black ABS box | HM229-ND | | Electronics box | |
| [www.digikey.com](http://www.digikey.com/) | 1 | Horn power receptacle | SC1230-ND | | Case mounted horn cord connector | |
| [www.digikey.com](http://www.digikey.com/) | 1 | Horn power plug | SC1215-ND | | Horn cord plug | |
| [www.digikey.com](http://www.digikey.com/) | 1 | Green panel mount LED | 350-2120-ND | | Power light | |
| [www.digikey.com](http://www.digikey.com/) | 1 | Yellow panel mount LED | 350-2122-ND | | Rolling Start light | |
| [www.digikey.com](http://www.digikey.com/) | 1 | Red panel mount LED | 350-2118-ND | | Mode light | |
| [www.digikey.com](http://www.digikey.com/) | 1 | Piezo buzzer | 458-1252-ND | |  | |
| [www.digikey.com](http://www.digikey.com/) | 2 | P-clamps | RP327-ND | | Horn plate cable clamps | |
| [www.boltdepot.com](http://www.boltdepot.com/) | 3 | 1/4-20 x 1/2" SS pan head machine screw | 1376 | | Horn plate leg fasteners | |
| [www.boltdepot.com](http://www.boltdepot.com/) | 3 | 1/4-20 SS stop nuts | 2554 | | Horn plate leg fasteners | |
| [www.boltdepot.com](http://www.boltdepot.com/) | 2 | 8-32 x 3/8" SS pan head machine screws | 1345 | | Horn plate cable clamp fastener | |
| [www.boltdepot.com](http://www.boltdepot.com/) | 2 | 8-32 SS nuts | 2559 | | Horn plate cable clamp fastener | |
| [www.boltdepot.com](http://www.boltdepot.com/) | 2 | 4-40 x 1/4" RH machine screws | 1701 | | Mount piezzo buzzer to electronics box lid | |
| [www.boltdepot.com](http://www.boltdepot.com/) | 4 | 8-32 x 1/2" RH machine screws | 1725 | | Mount terminal block to electronics box | |
| [www.boltdepot.com](http://www.boltdepot.com/) | 4 | 5/16" x 4/40 FH machine screws | 9332 | | Mount Arduino on standoffs to electronics box | |
| [www.admiralmetals.com](http://www.admiralmetals.com/) | 1 | 1/8" x 8" x 10" 6061 aluminum plate |  | | Horn plate (price includes 3 legs) | |
| [www.admiralmetals.com](http://www.admiralmetals.com/) | 3 | 1/8" x 1" x 6" 6061 aluminum bar |  | | Horn plate legs | |
| [www.amazon.com](http://www.amazon.com/) | 1 | Wolo Model 415 Airsplitter Horn |  | |  | |
| [www.homedepot.com](http://www.homedepot.com/) | 1 | Command Brand Picture & Frame Hanging | 17206 | | Fasten electronics box and battery to the case | |
| [www.homedepot.com](http://www.homedepot.com/) | 1 | Command Brand Picture & Frame Hanging | 17202 | | Fasten horn relay to the case | |
| **External Power Option** | | | | | | |
| [www.digikey.com](http://www.digikey.com/) | 2 | 15A Power Plug - Auto | | APP-001-15AMP-ND | | One on each end of the 14-2 power cable |
| [www.digikey.com](http://www.digikey.com/) | 1 | 15A power socket - Auto | | AS212-ND | | Mount on the end of the case |
| **Bulk Items** | | | | | | |
| hardware store |  | Black 22 gauge solid wire | |  | |  |
| hardware store |  | Red 22 gauge solid wire | |  | |  |
| hardware store |  | Green 22 gauge solid wire | |  | |  |
| hardware store |  | 16' 14-2 appliance cord | |  | | Flexible, rubber sleeve |
| **Tooling** | | | | | | |
| [www.sparkfun.com](http://www.sparkfun.com/) |  | FTDI Cable 5v | | DEV-09718 | | Used only to upload to the board |
| **My Soldering Setup** | | | | | | |
| [www.sra-solder.com](http://www.sra-solder.com/) |  | Xuron 485 Long Nose Plier | |  | |  |
| [www.sra-solder.com](http://www.sra-solder.com/) |  | Xuron 170-II Micro-Shear Flush Cutter | |  | |  |
| [www.sra-solder.com](http://www.sra-solder.com/) |  | Aoyue Basic Soldering Station 936 | |  | |  |
| [www.sra-solder.com](http://www.sra-solder.com/) |  | Soldering Iron Tip T-S4 | |  | |  |
| [www.sra-solder.com](http://www.sra-solder.com/) |  | ESD 2 Layer Smooth Rubber Work Surface Mat with Wrist Strap | |  | |  |
| [www.sra-solder.com](http://www.sra-solder.com/) |  | MG4885-227G, Wire 63/37 Rosin Core .032 | |  | |  |
| [www.sra-solder.com](http://www.sra-solder.com/) |  | Double Clamp Stand with Magnifier | |  | |  |

The software is written in “c” with many calls to Arduino functions. The horn signal generation is straightforward since it is sequential logic. Signal generation begins with the Start button interrupt. When the starting signal sequence is finished, the Start button interrupt is enabled so that the user can begin another sequence. The software constants are biased to account for the timing accuracy of the Arduino clock. The bias may vary with each board. My plan is to calibrate each board over a two hour period to determine the adequacy of the bias. The objective is to achieve adequate accuracy over a maximum 60 minute period. This means that there will be no noticeable drift from the timeline when measured with a stopwatch. This covers what I consider the worst case of 12 successive rolling starts. The software is compiled using a Mac or PC and downloaded into the Arduino non-volatile memory prior to final assembly of the device. The Arduino Interactive Development Environment needed to edit, compile and download the software can be obtained at [http://arduino.cc/.](http://arduino.cc/)

As with the base hardware, the software is also easy to modify to meet additional requirements. There are a few conventions used in the software to assure timing accuracy that should be maintained when adding or modifying the signal sequences. One of these is the measurement of the duration of function calls and other software code during a signal interval. These measurements are subtracted from the interval durations to account for the time that it takes to execute that software. The corrected durations are used as the wait times. For example, the time that it takes to sound the horn is subtracted from the time interval to give the remaining time to wait in the interval before the next signal event.

All of this is housed in a polycarbonate case. The battery and controller box are fastened to the inside of the case using adhesive strips. The controller box protects the circuit boards should the heavy battery come loose. The case is moisture resistant. Water can damage the switches and coaxial charging jack.

Table 2 lists all of the parts needed to complete the following steps. The Digi-Key Corporation ® bill of materials (BOM) is available as a comma separated variable file in the repository. This CSV file can be uploaded to Digi-Key using their BOM Manager to order the parts. The SparkFun Electronics ® parts can be found on the SparkFun web site as the Voojibox Wishlist, which is a public wishlist.

# Step One – Assemble the Shield

The shield assembly consists of the board, headers, components, jumpers, and off board leads. Solder the headers to the board first. I use a little drop of super glue on each end to hold each header in place and square to the board for soldering. Solder all components, onboard jumpers, and leads to the shield. The recommended routing and placement of top side jumper wires and components is shown in the Shield Wiring illustration Figure 1. Figure 2 lists the Bridge Connections on the reverse side of the board. You need to understand the shield circuits that you are wiring, shown in Figure 3.

As you solder the components and jumper wire to the top side of the board, do not trim the bottom side wire. A minimum protrusion of ½” will be needed. These will be used to make wire bridge connections for components and off board leads.

After all of the components and wire are soldered, solder the bridges required to make the connections shown in the Bridge Connections table. Each bridge is made by bending the “From” wire and making a single wrap around the “To” wire. Figure 4 illustrates the technique. Before soldering the connection at the wrap, cut the excess wrap wire away from the joint. Do not bend the MOSFET leads. After soldering, all of the excess wire and the MOSFET leads must be trimmed away.

I used many wire colors to code their function. There are other ways to achieve this. For example, a label maker can be used to print number or letter coded labels that can be wrapped on the wire. As a minimum, I recommend using green (ground), black (-), and red (+) 22 gauge wire for the leads and on-board jumpers.

# Step Two – Assemble and Upload the Arduino

Before assembling the Arduino, use the board to make a drilling template. This template will be used to mark the location of holes that need to be drilled into the controller box (Figure 5). The holes are used to mount the Arduino / shield assembly into the inside of the controller box.

Solder the headers onto the Arduino Pro. I use a small drop of super glue on each end of the headers to hold them upright and square for soldering. Fasten four plastic spacers to the Arduino using the round 4-40 x ½” round head screws (Figure 6). You are ready to upload the software.

The upload process uses the FTDI cable and the Arduino Interactive Development Environment (IDE). Connect the FTDI cable to the Arduino with the black wire aligned with the pin marked “GND” on the board. If you are using Windows, the drivers are installed when the cable is initially plugged into a USB port. After the driver is installed, you can make one configuration change to enable the auto-reset. Open the Device Manager (in Control Panel > System > Hardware), and find the USB Serial Port under Ports. Right-click and select Properties. Go to Port Settings > Advanced and check Set RTS on Close under Miscellaneous Options.

Start the IDE and open the sketch for the Sound Signal Device. Go to Tools>Serial Port and check the port being used by your Arduino. Go to Tools>Board and check “Duemilanove w/ATmega 328”. Compile the sketch by going to Sketch>Verify / Compile. Now you are ready to click on File>Upload. This starts the upload, which takes a few minutes. A small message will be displayed in the lower left to indicate that the upload is complete. If the status area at the bottom of the IDE window has any red error messages, then you have some debugging to do.

Plug in the two-wire JST power cable. Insert the shield pins into the Arduino headers and press them together firmly. There are28 pins so pressing them together requires some force applied judiciously with a very slight rocking motion. The Arduino and shield assembly is ready to calibrate and install into the controller box.

# Step Three – Connect, Test, and Calibrate the Arduino and Shield

Make temporary connections and test using a breadboard. This should be done before any further assembly is done. Use small momentary switches as the mode switch, start and horn switches. I use the 12v sealed lead acid battery as the power source. You should see the LEDs cycling and hear the horn relay being activated. If not, check the wiring and solder joints. The most common causes for failures are a missing jumper, improperly connected jumper, or a poor crimp connection on a connector.

After confirming correct operation of the assembly, calibration is necessary. The Arduino Pro uses a resonator as the clock source. Neither the resonator nor a crystal oscillator is accurate enough for timing over a long period of time.

Calibration is probably not needed for RRS Appendix S three-minute rolling starts because competitors use the signals rather than a stopwatch to time their start. Competitors will rely on stopwatches more for the other signal sequences because there is a long interval between signals. I calibrate because in any scenario I want competitors to be able to rely on the timing accuracy and the length of the start intervals.

The initial assumption is that this timer does not require accuracy over a long period of time. This accuracy would be needed for rolling starts that continue for a long period such as two hours. That would be about 40 three-minute starts and this is not a practical scenario. My approach insures that there is no noticeable (stopwatch measured) drift in the signals after an hour of rolling starts.

Set the mode for rolling three minute starts. Press the start switch and listen for the horn relay clicks. Have a chronometer ready. I use my Timex wristwatch CHRONO mode. Start the chronometer when you hear the start signal click at the three minute mark. It is easy to recognize because of the five second countdown clicks. Count “5, 4, 3, 2, 1, 0” and on zero start the chronometer.

Let the timer run for about two hours. Stop the chronometer on the next start signal that you hear. Count the relay clicks “5, 4, 3, 2, 1, 0” and on zero stop the chronometer. Write down the chronometer reading, which should have the format HH:MM:SS where HH is the number of hours, MM is the number of minutes, and SS is the number of seconds. The number of three minute intervals that were timed can be calculated from this reading.

Calculate the total number of seconds elapsed as follows:

SecondsTotal = HH x 60 x 60 +MM x 60 + SS

Calculate the number of three minute periods elapsed.

IntervalsInteger = Integer [ ( SecondsTotal / 180) + .5 ] where *Integer [X]* is the whole number in *X*.

The timing error is given by

ErrorPercent = [ 180 x IntervalsInteger – SecondsTotal ] / SecondsTotal

If ErrorPercent is negative, the Arduino clock is running ahead of real time and the number of clock ticks per interval need to be biased downward. It ErrorPercent is positive, the Arduino clock is running behind real time and the number of clock ticks per interval need to be biased upward.

Calculate the bias for each of the software parameters and enter into the software code where appropriate. The software is well commented and the constants that require the bias are all in one place in the source code. Be sure to use the ErrorPercent ratio rather than the percent value. That is, .024% is a factor of .00024, which is the factor calculated by the formula above.

This approach works because the error is not large. It is large enough that variations in signal timing will be noticeable after several minutes if not calibrated. Racers will notice the variation during later starts of a rolling sequence. The calibration only affects the rolling sequences that can continue for a significant length of time.

Note: The Arduino Pro resonator is stable through a range of temperatures. This is important because the timer will be used through both warm and cold seasons. It does not require calibration for each season, only one time at assembly to account for variations in the resonator and electronics.

# Step Four – Mount the Arduino and Shield in the Electronics Box

The small plastic box is used to protect the small boards from mayhem should the heavy lead acid battery break loose. I chose this case from many available so that the wall thickness would allow countersinking for the standoff screws on the bottom and tapping for the barrier strip and buzzer mounting screws. The countersinking keeps the bottom flat so that adhesive strips can be used to fasten the box to the case. Tapping for the barrier strip and buzzer eliminates the need for nuts on the inside of the box. Before installing the Arduino and shield into the box, drill and tap the bottom for the barrier strip and drill a hole for the leads that exit the from inside the box into the case. The hole for the leads should be at the end of the box opposite the Arduino/Shield assembly (Figure 7). Drill and tap the holes in the box top for the piezo buzzer. A small hole is drilled near the buzzer for the buzzer leads to go into the box. Drill holes for the LEDs. There is no exact location for the box top holes, but they should be at the end of the box that is opposite the Arduino/Shield assembly (Figure 8).

The buzzer and LEDs are mounted on the electronics box to avoid wire clutter inside the case. The clear case provides visibility to the LEDs. The buzzer is audible when the case is closed.

When marking the holes, be sure to allow adequate clearance to mount the Arduino assembly inside the box. Get an estimate of the position of the holes by placing the Arduino assembly inside the box. The assembly should be positioned at one end of the box to allow for wiring that will be stuffed into the open end. The hole for the leads as well as the LEDs and the buzzer mounted on the box cover are at the open end of the box. The Arduino should be oriented so that the power cable connector faces the open end. Slide it toward the end of the box until it stops and slide it back about ¼”. Note the location of the spacers on the bottom of the box.

It is important to mark the Arduino mounting holes so that the Arduino is centered at one end of the box. The reason is to allow adequate room for the screws and to assure that the Arduino will be mounted flush with the interior bottom of the box. The marking and drilling must be done accurately since there is very little play in the mounting standoffs to account for misplaced mounting holes.

Figure 9 summarizes all of the electronics box hole locations and sizes.

It is best to run all of the leads to the short end of the box and under the Arduino between the mounting spacers (Figure 10). This reduces the strain on the solder connections on the board. The leads for the LEDs and the piezoelectric buzzer are connected to the devices with snap connectors. The four ground leads for the buzzer, green, yellow, and red LEDs are twisted together into one snap connector. The shield has two ground wires and one of them remains inside the box connected to the buzzer and LEDs. The buzzer and LED signal wires must be doubled over prior to insertion into the crimp fitting on the connector. The wires are very small gauge and doubling them is necessary to get a good crimp connection.

Run the external wires through the hole drilled for the external leads. Snap all of the connectors together inside the box and coil the wires to fit inside the open end of the box (Figure 11). The coils may be held together with a double wrap of 22 gauge wire. I don’t like using cable ties here because they are difficult to remove and can stress the small wires. Use care to avoid closing the box lid on wires because the lid will cut the wires when screws are tightened.

# Step Five – Install Everything into the Case

Throw away the rubbery liner that comes with the case. Drill holes in the case lid for the power switch and the pushbuttons (Figure 12). The switch and button holes are ¼”. Drill the holes for the charging socket and the horn connector in the base of the case opposite the handle (Figure 13). The charging socket is 5/16”. The horn connector is 13/16” and it is best to use a hole saw. There is no exact location for these holes, but all of them should be near the center of the case to allow clearance for the wires, battery, and electronics box.

Use care when drilling. The polycarbonate of the case can shatter. I chose guide-point drill bits made by DeWalt. My opinion is that almost the entire cutting edge comes into contact with the plastic at the start of the drilling process. As a result, there is less chance of the flutes biting the edge, twisting into the thin plastic, and snapping it. I also use a drill press for better control, feeding the bit and backing out successively until the hole is through.

Leads should be soldered to the switches, charging socket, and claxon connector before installation of the battery, relay, and electronics box (Figure 14).

The small electronics box, the battery, and the relay are installed into the box using 3M™ Command™ Strips (Figure 15). I recommend using the two-part strips that have a Velcro backing. This makes it easy to remove the parts for maintenance and repair.

Attach the strips to the bottom of the electronics box, battery and relay using the adhesive backing. Position them so that the tabs will be toward the inside of the case when installed. The leads on the electronics box should face the center of the case. Press the second part of each strip together using the Velcro sides. Do a practice install of each part so that you get an idea how to achieve alignment to get everything square with the sides of the case. The method that seems to work the best is to slide the box or battery along the end of the case as you lower them into place. The fit is close and they almost self-align as you slide them in. When you are ready, remove the backing from the strips to expose the adhesive and slide them into the case, pressing them to the bottom to get the adhesive to grab.

All of the leads are connected with crimp type connectors. Connections to the terminal barrier block are done with spade connectors. The suggested connections are shown in Figure 16. Connections to the battery and other high current connections such as the relay are made with ¼” quick connect fittings. The exception is that the low current 12v and drain connections to control the relay are also made with quick connects. All other wire to wire connections are made with round snap connectors. The connectors simplify maintenance and repair in the future. Care should be taken to crimp the wire, not the insulation. If the fully built device fails to operate, these crimp connections are one place to look for continuity issues.

Polarity is important for the LEDs, the relay, and the battery. The LEDs that I selected use colored wire to indicate polarity. The relay “+” terminal should be connected to 12v and the “-“ terminal to the MOSFET drain. Care must be taken when connecting to the battery to be sure that the “+” and “-“ leads go to the correct terminal on the battery.

The overall wiring of the case is shown in Figure 17.

# Step Six – Power Up and Try All Options

The first thing to try after power up is to cycle through the modes. If that works, select a mode and push the start switch. You don’t need to have the horn connected to do this. You can listen for the switching sound of the relay instead. If something doesn’t work, you will need to debug the device. This includes checking voltage levels and inspecting connectors. It may be necessary to trace wires to be certain that the polarity is correct for the LEDs, the relay control circuit, and the battery. If the polarity was reversed, check for damage to the connected circuit or component.

That’s the best advice that I can give. If you have taken on the build of this device, you should be skilled at any further diagnostics or debug that may be required. It is a reasonable simple device, so it shouldn’t be difficult to get it to work.

# Step Seven – Build the Horn Mount

The legs and the plate are 1/8” aluminum (Figure 18). Mark the locations of the holes and center punch for drilling. Scribe the radii at each corner.

Aluminum is soft, but not so easy to cut. To cut the plate corner radii, I use a jigsaw with a fine metal blade at a low speed. The blade should be a new one so that it is as sharp as possible to start. When you are finished, the blade will probably be ready for the trash. The aluminum heats up and binds to the cutting blade as well as wearing it down very quickly.

Once cut, the corners must be completed with a file. Keep the file card handy and use it often to clear the aluminum from the file serrations. Use the file to remove all of the burrs from the edges and to round over the edges for a smooth feel. Use emery cloth to smooth and burnish the edges. The edges need to be rounded to prevent injury to users from sharp cutting edges.

Drill the mounting holes in the plate. I use a 9/64” bit to make pilot holes before drilling to the final size. Remove the burrs from the holes using a large countersink bit, careful to just remove the burrs and avoid making the holes larger.

The legs are cut to length and burrs removed. The corners must be rounded with a file to give a smooth edge to the entire piece. Mark and drill the hole in each leg. Scribe the bend lines in each leg and use a bending brake to make 90o angles. A bending brake is recommended because I don’t know of another method that will result in consistent and sharp 90o bends. I have not tried, but clamping in a vise at the bend lines and using a mallet to bend may work. Details are shown in Figure 19.

The plate is assembled using the horn parts, cable clamps and specified screws and bolts.

# Step Eight – Fabricate the Horn Cable

Use 14-2 copper appliance cable. The diameter of 14-3 is too large for the connector and the ground is not necessary. The maximum length of the cable is 15’ to handle the peak power of the horn compressor which is 15A.

The tricky end is the one with the case connector. The pins must be removed prior to soldering to avoid melting the connector housing. Use fine point pliers to grab the solder cup by the edge and pull each pin from the connector base. Do not clamp the pliers onto the outside of the pin since you may crush the solder cup. Put the screw cap on the cable before soldering. After soldering, reassemble the connector and screw the cap to the connector base.

The horn compressor end of the cable uses crimp quick connect fittings sized for the compressor connectors.

# Notes on Parts Selection

MOSFET – This part has a logic level gate with a threshold gate voltage of 2v. This is important because it requires only the Arduino 5v digital signal to switch it. MOSFETs without the logic level gate require more circuitry to provide higher voltage to the gate to make it switch.

Claxon (Horn) – The claxon should not exceed 15A. The 14-2 gauge extension cable is sized to support this load. The length of the cable should not exceed 15’ for the specified load.

Battery – The lead acid battery was chosen for price, performance and size. Other 12v battery chemistry will work as well. You will pay a much higher price for some of the newer battery technologies, but the performance advantages will not be incremental for this application.

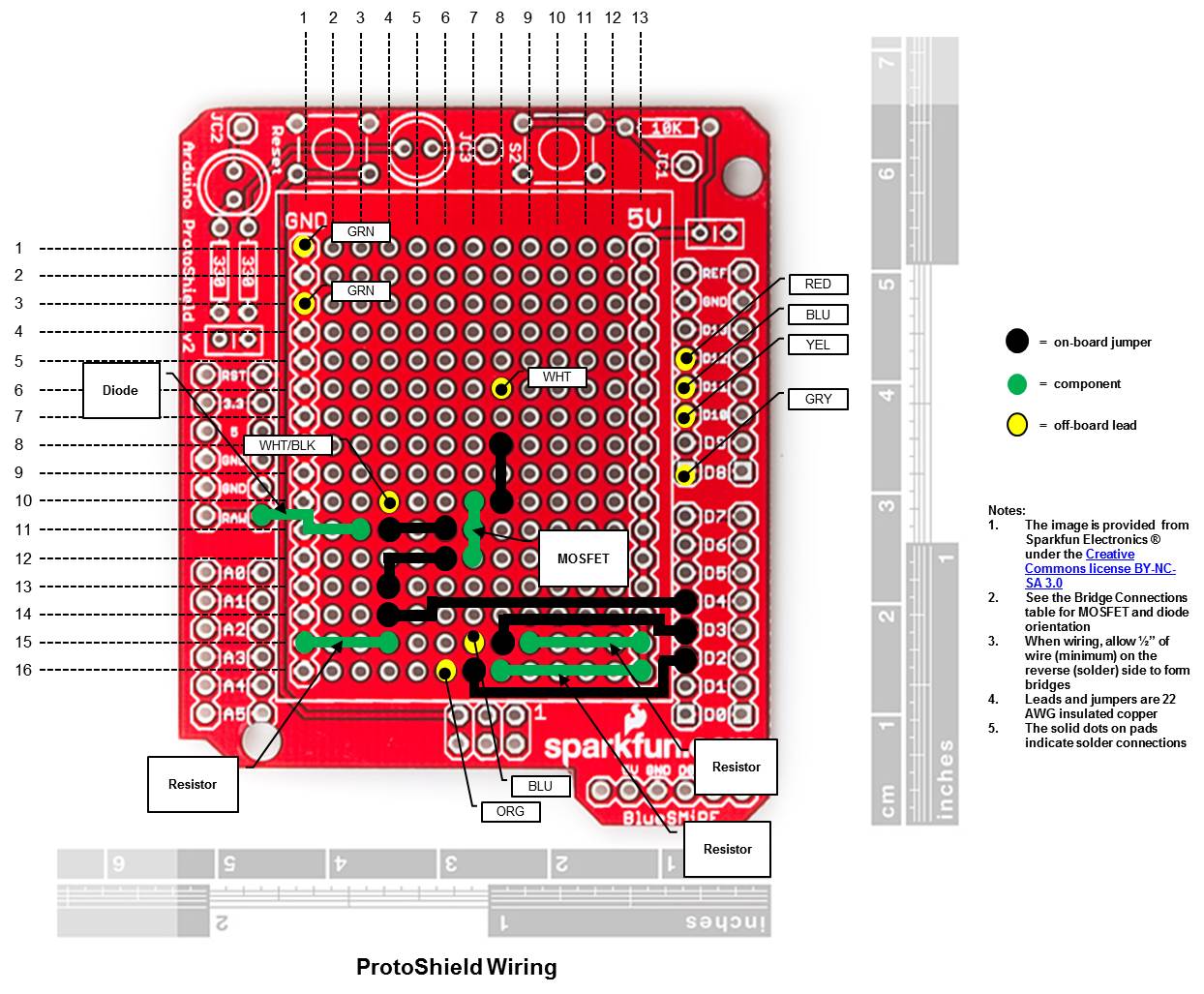
DPDT Power Switch – This part cuts off the power to the Arduino when “OFF” and connects the jack to the battery for charging.

Arduino Board – I use the SparkFun Electronics® board because it and all of the parts inside the electronics box can be sourced from SparkFun Electronics®. It is a good board and I like minimizing the number of sources for parts. I built the original design prototype using the SparkFun Inventor’s Kit.

Proto Shield – This part is available from Sparkfun Electronics® and simplifies making connections with the Arduino. Shield headers plug into the mating headers on the Arduino, making all of the pins available on the shield. The board has a convenient ground bus and 5v bus. All of the unique signal device circuitry is soldered onto the shield.

# Appendix A – Assembly Figures

Figure 1 - Shield Wiring

****

|  |  |  |  |
| --- | --- | --- | --- |
| **Figure 2 - Bridge Connections** | | | |
| **“From” Node** | **From** | **To** | **“To” Node** |
| White Battery Minus Lead | (6,8) | (8,8) |  |
|  | (10,8) | (10,7) | MOSFET Source |
|  | | | |
| Diode Anode | (11,3) | (11,4) |  |
| White/Black Load Lead (Load Minus) | (11,4) | (10,4) |  |
|  | | | |
|  | (11,6) | (11,7) | MOSFET Drain |
|  | | | |
|  | (15,4) | (14,4) | MOSFET Gate Switch Sink |
|  | (14,4) | (13,4) | D4 MOSFET Gate Switch |
|  | | | |
|  | (12,6) | (12,7) | MOSFET Gate |
|  | | | |
|  | (15,9) | (15,8) | D3 5V (Mode) Pull-Up Resistor |
| Blue Mode Switch Lead | (15,8) | (15,7) |  |
|  | | | |
|  | (16,8) | (16,7) | D2 5V (Start) Pull-Up Resistor |
| Orange Start Switch Lead | (16,7) | (16,6) |  |
| Note:   1. The “From” and “To” coordinates reference the component side of the ProtoShield. The solder joints for the bridges are made on the reverse side. 2. The coordinates are ( row , column ) according to the row and column numbering on the ProtoShield Wiring illustration. | | | |

Figure 3 - Shield Circuits

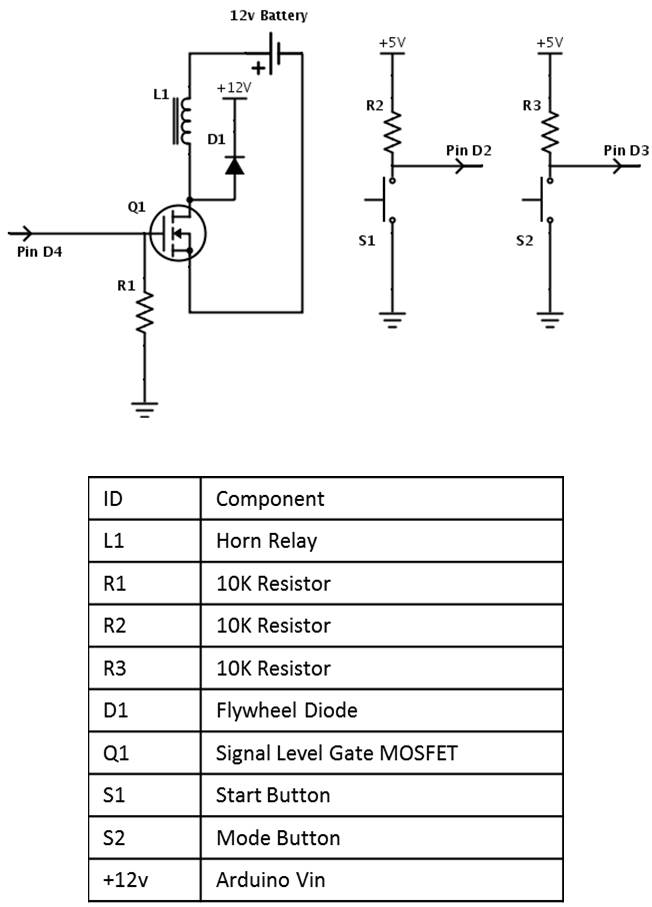
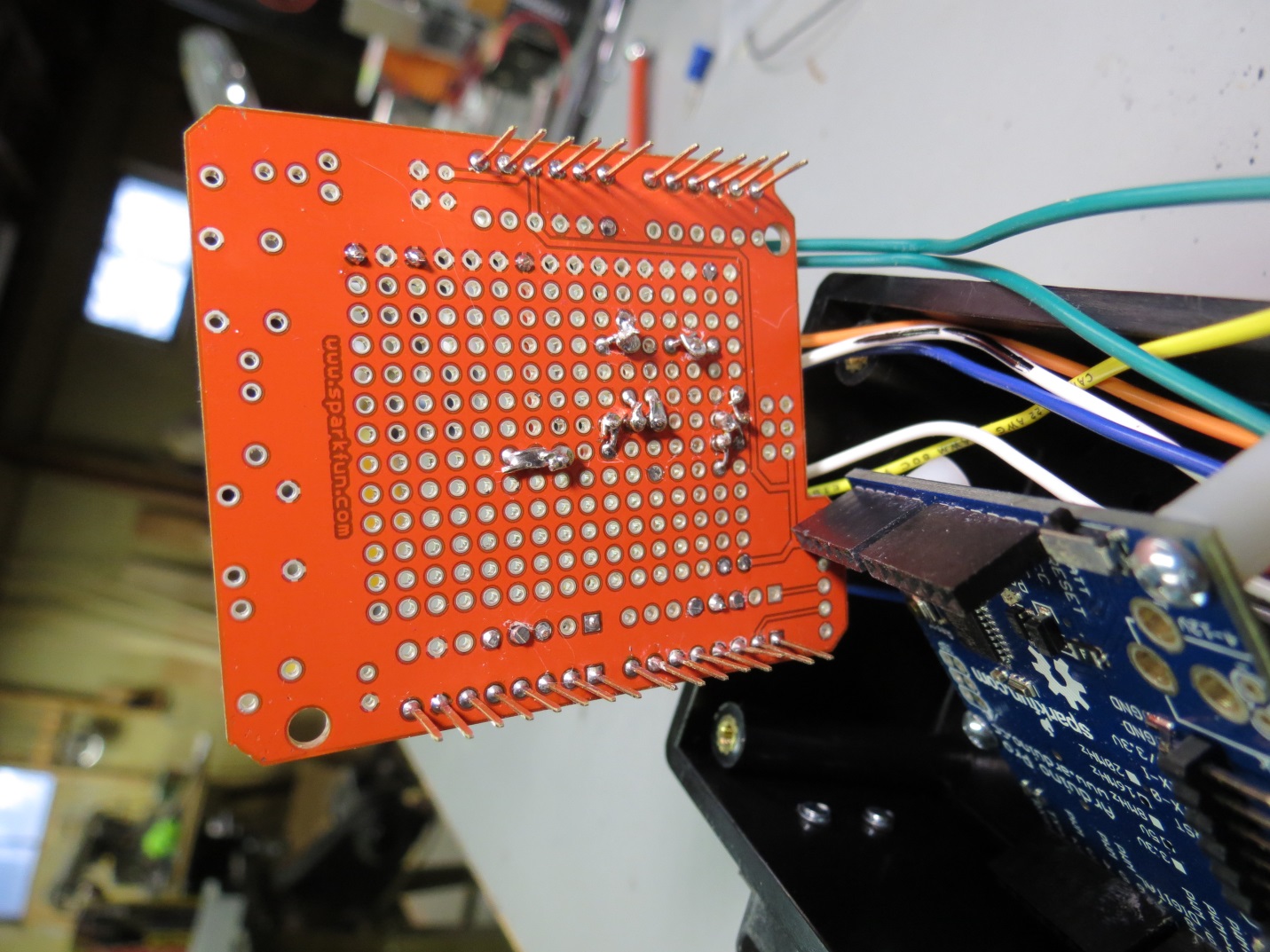


Figure 4 - Shield Bridge Solder Joints

Figure 5 - Arduino Mounting Holes

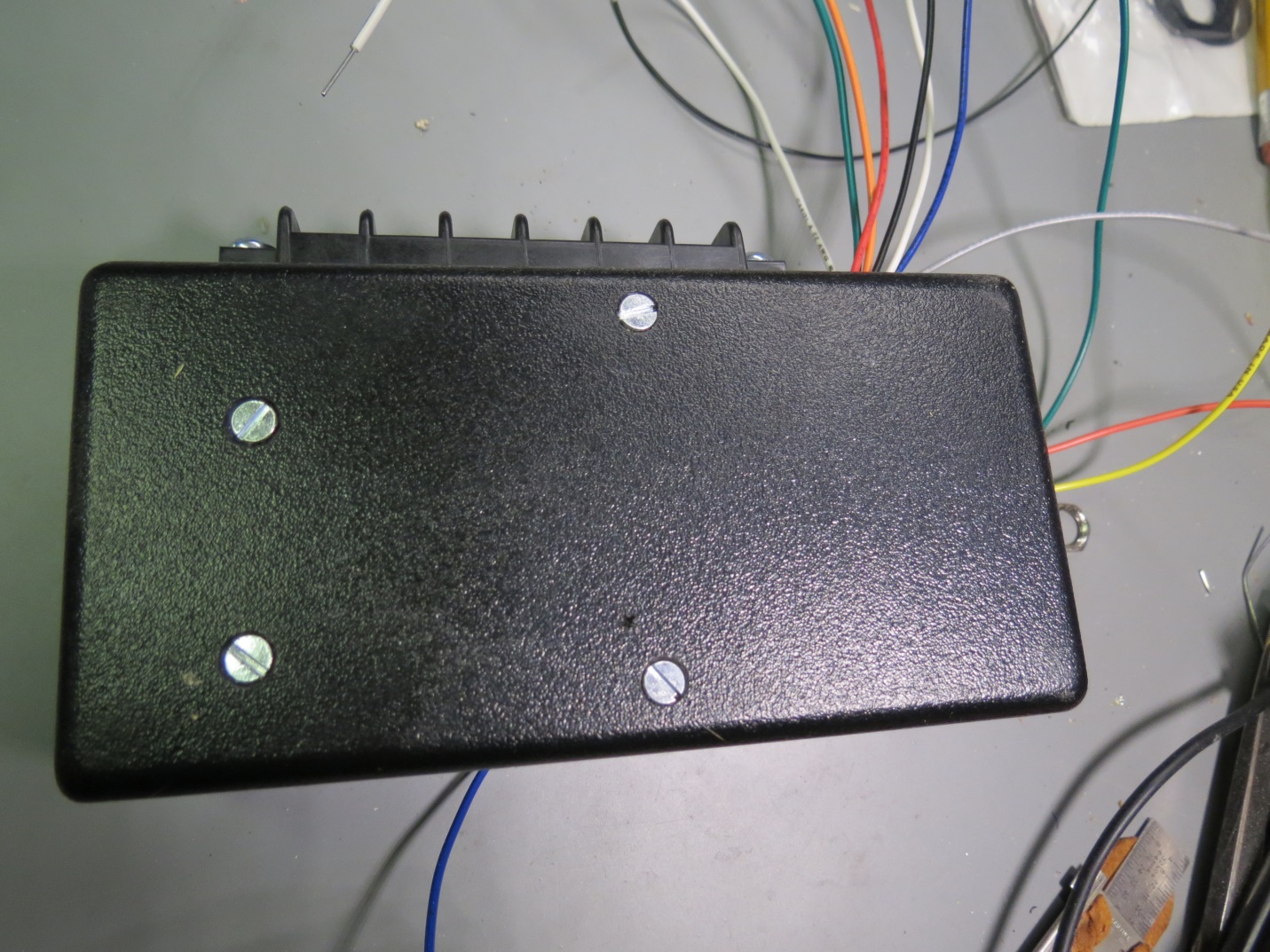


Figure 6 - Arduino Board with Headers and Spacers

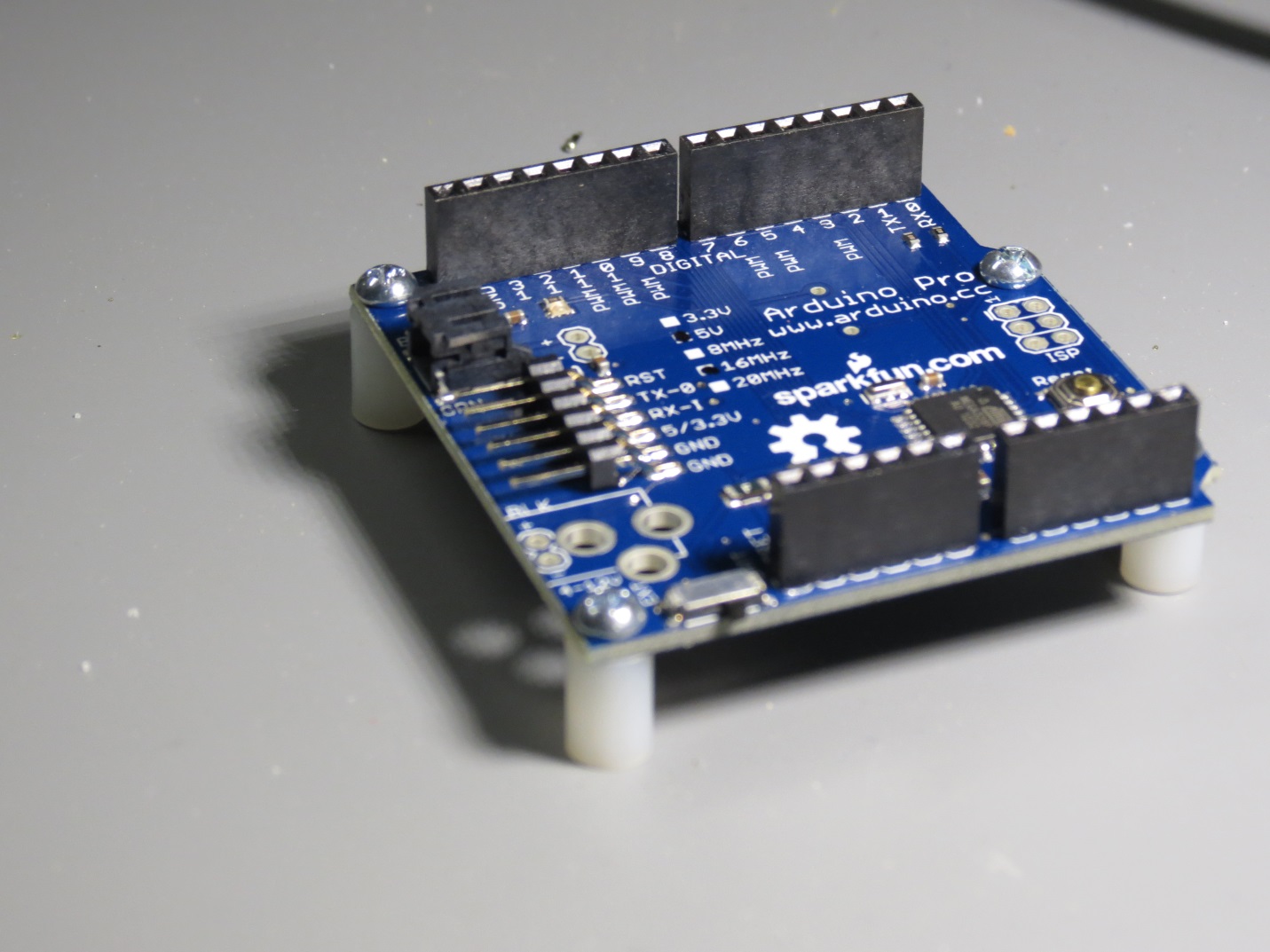


Figure 7 - Box Bottom Hole Locations

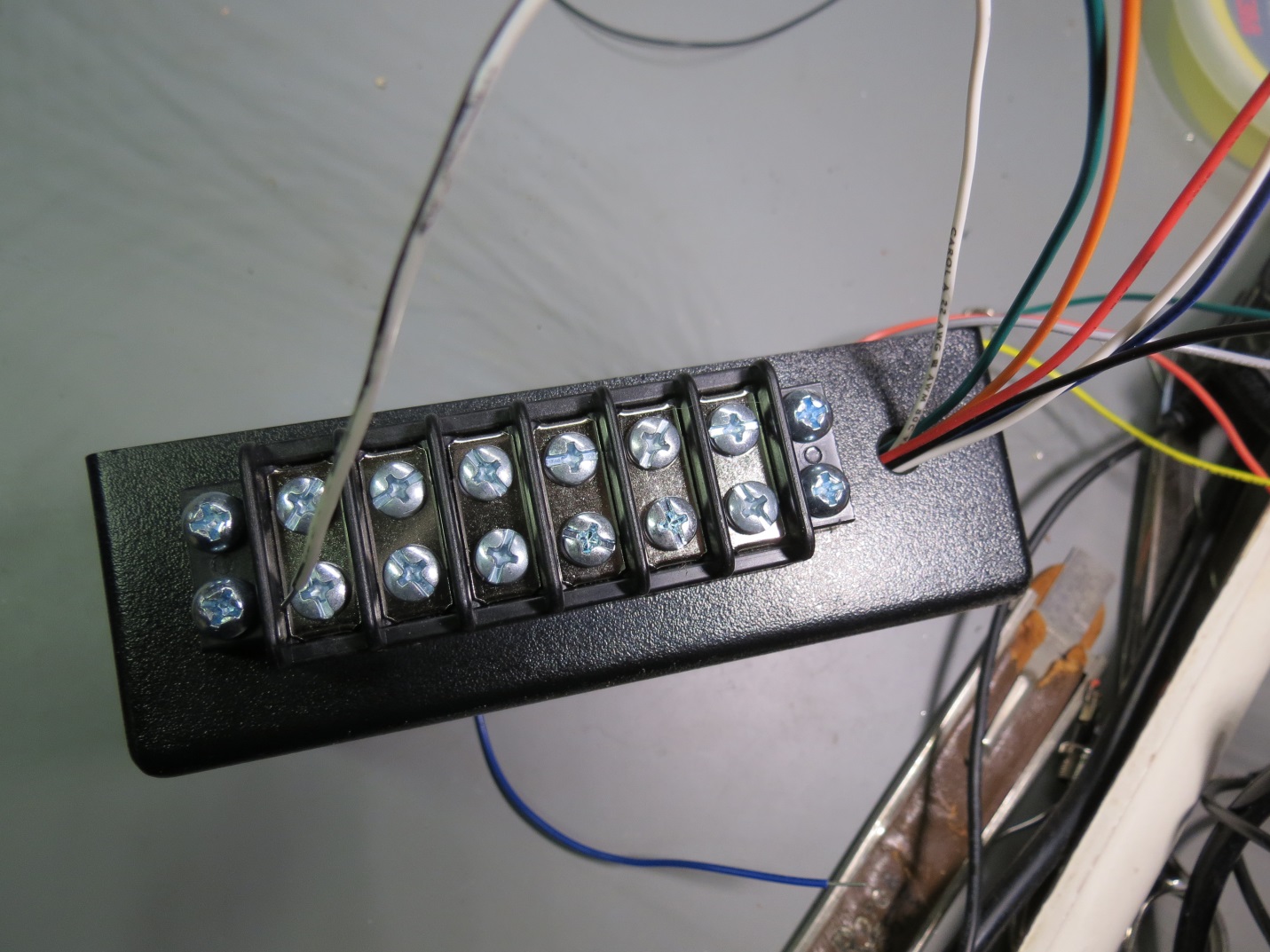
****

Figure 8 - Box Top Hole Locations

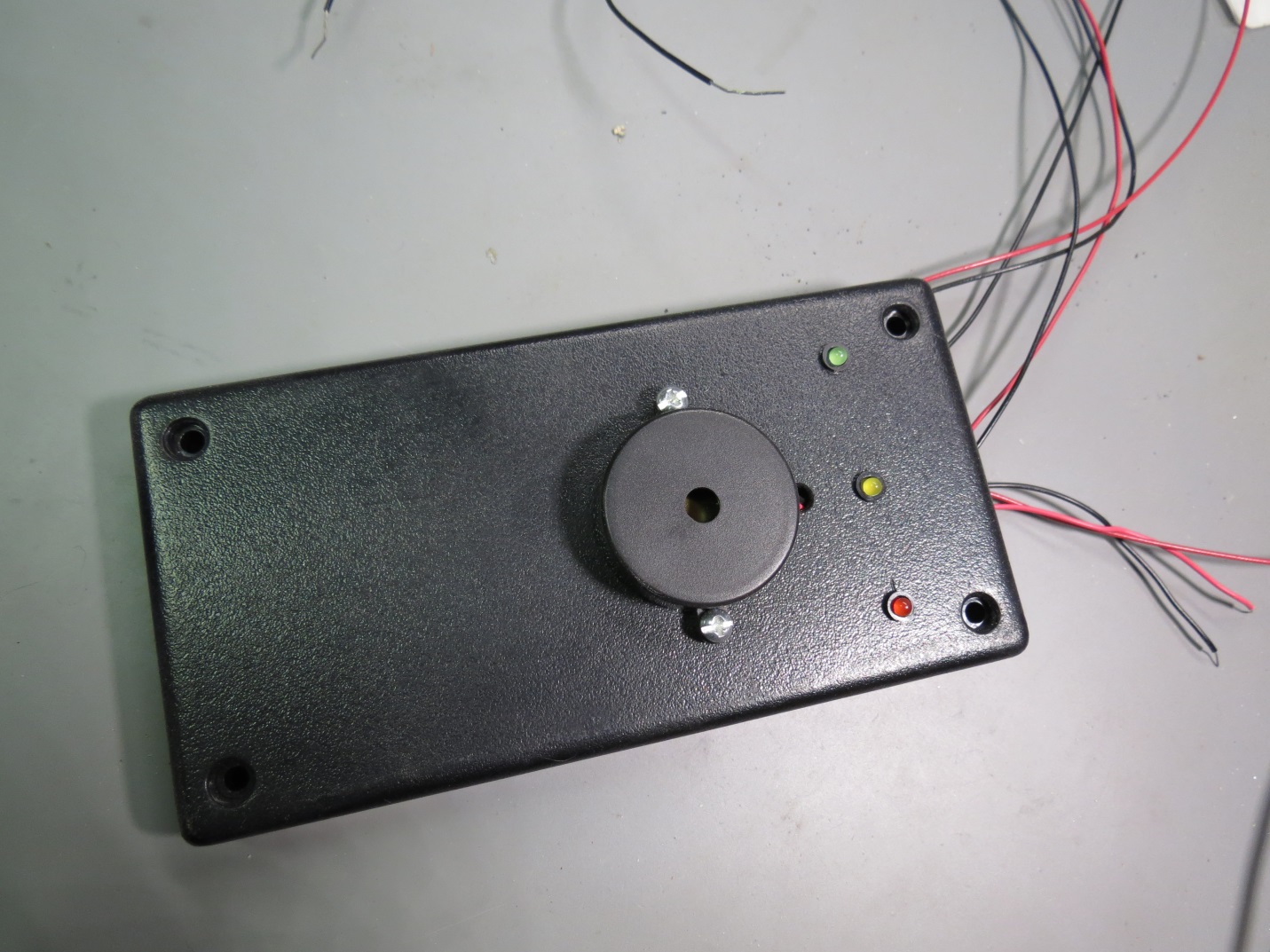
****

Figure 9 - Electronics Box Holes

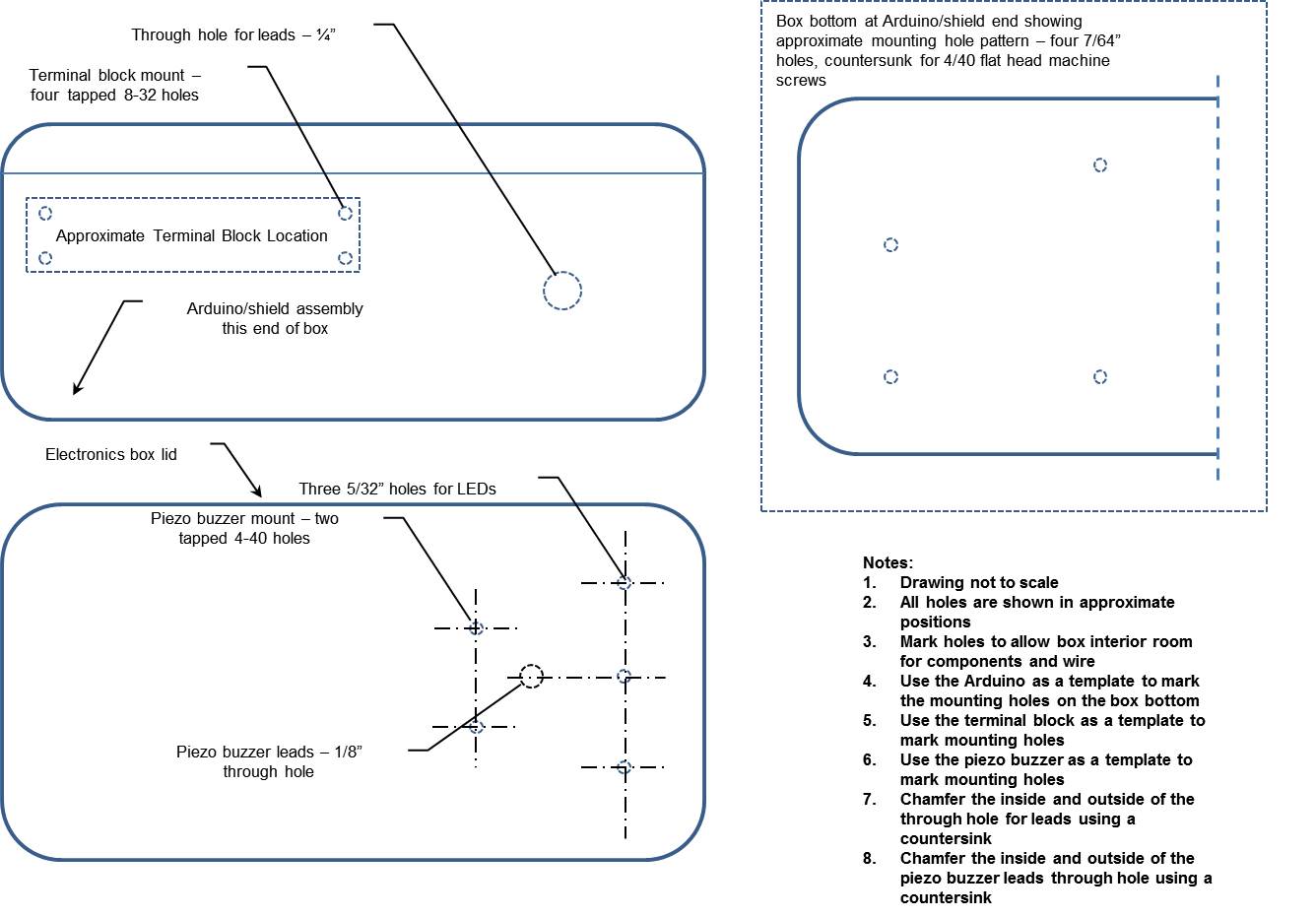
****

Figure 10 - Arduino/Shield and Lead Placement

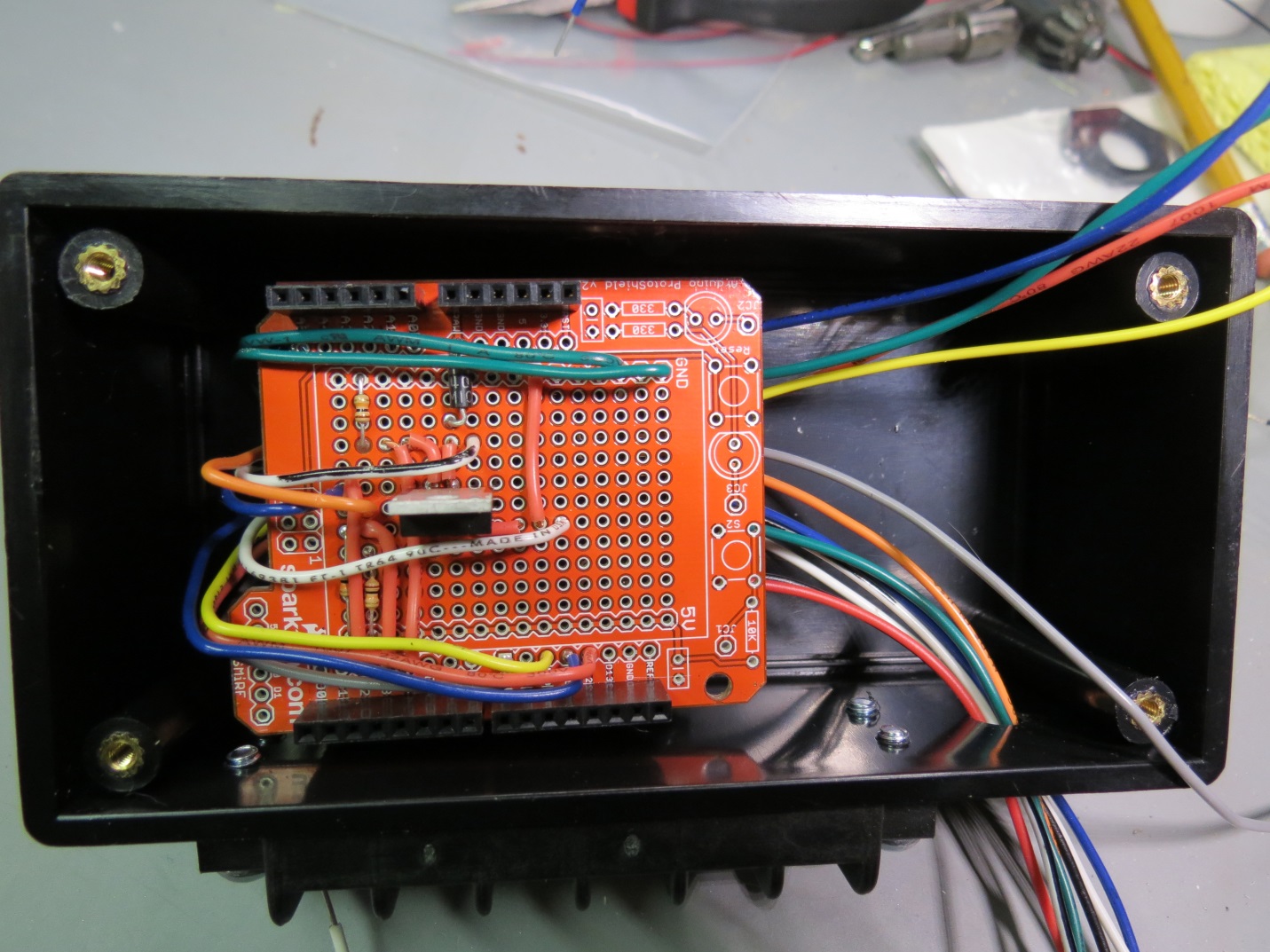
****

Figure 11 - Wire Bundle Inside the Box

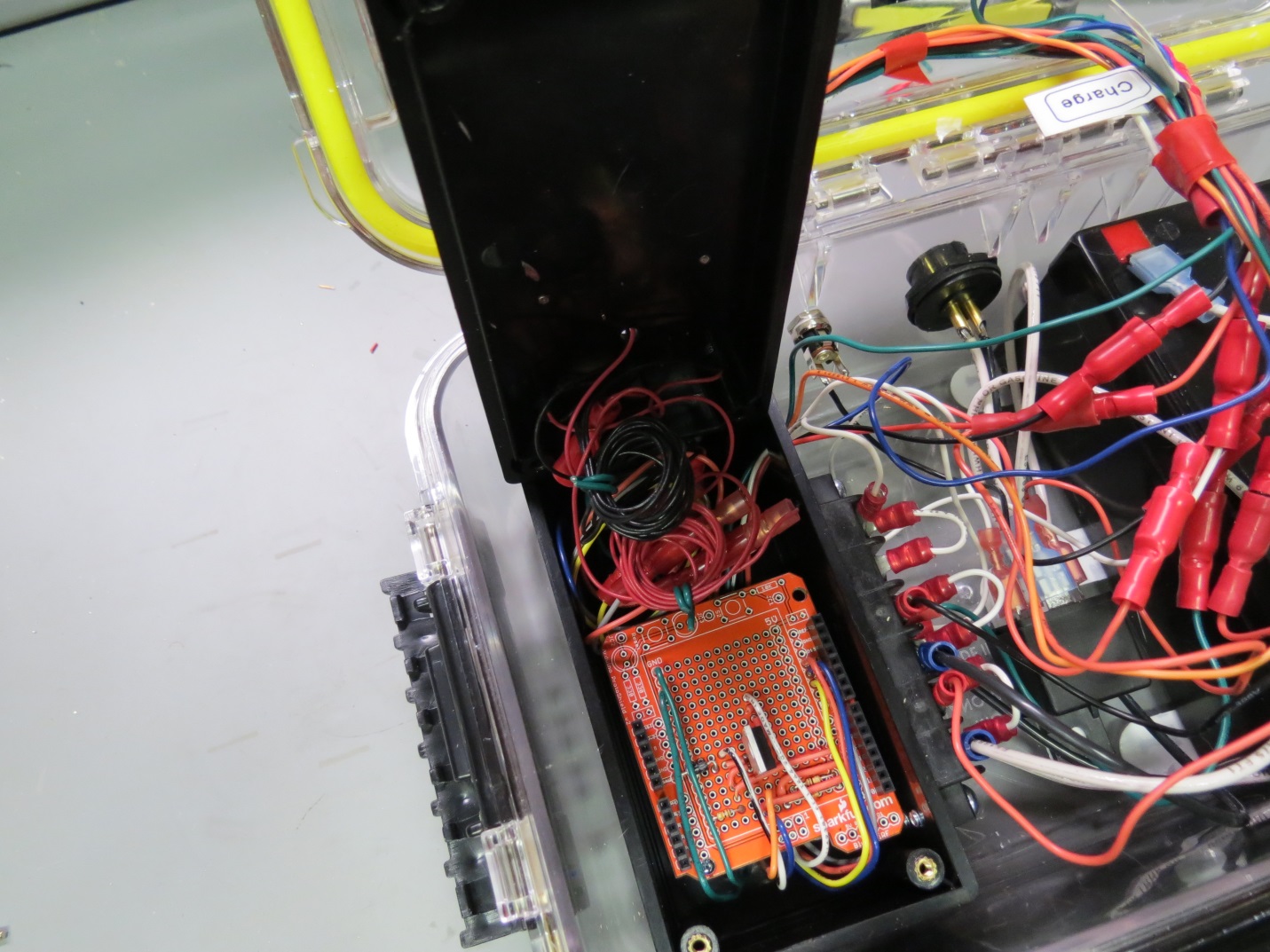
****

Figure 12 - Case Lid Hole Locations

****

Figure 13 - Case Base Hole Locations

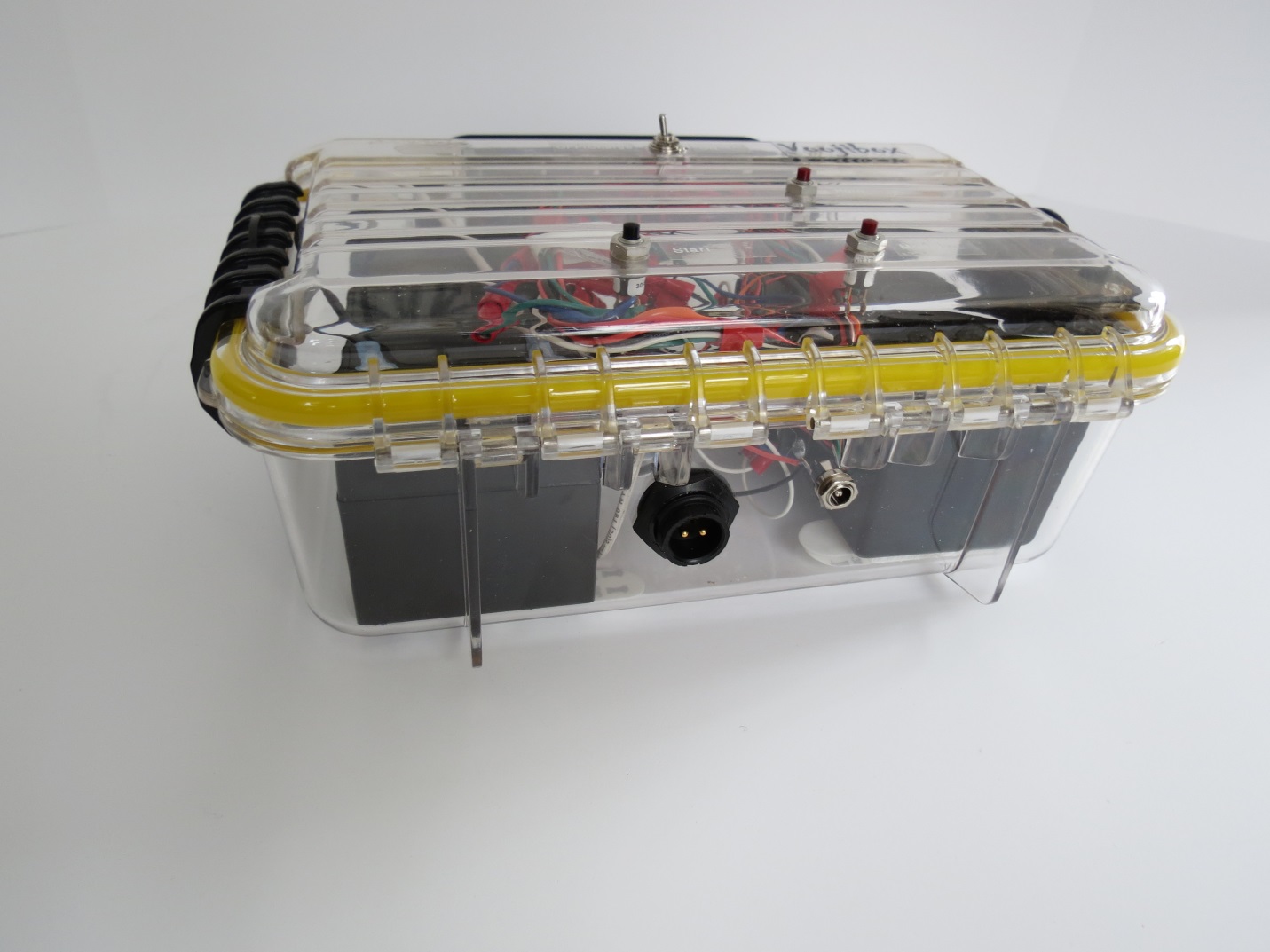
****

Figure 14 - Case Wiring

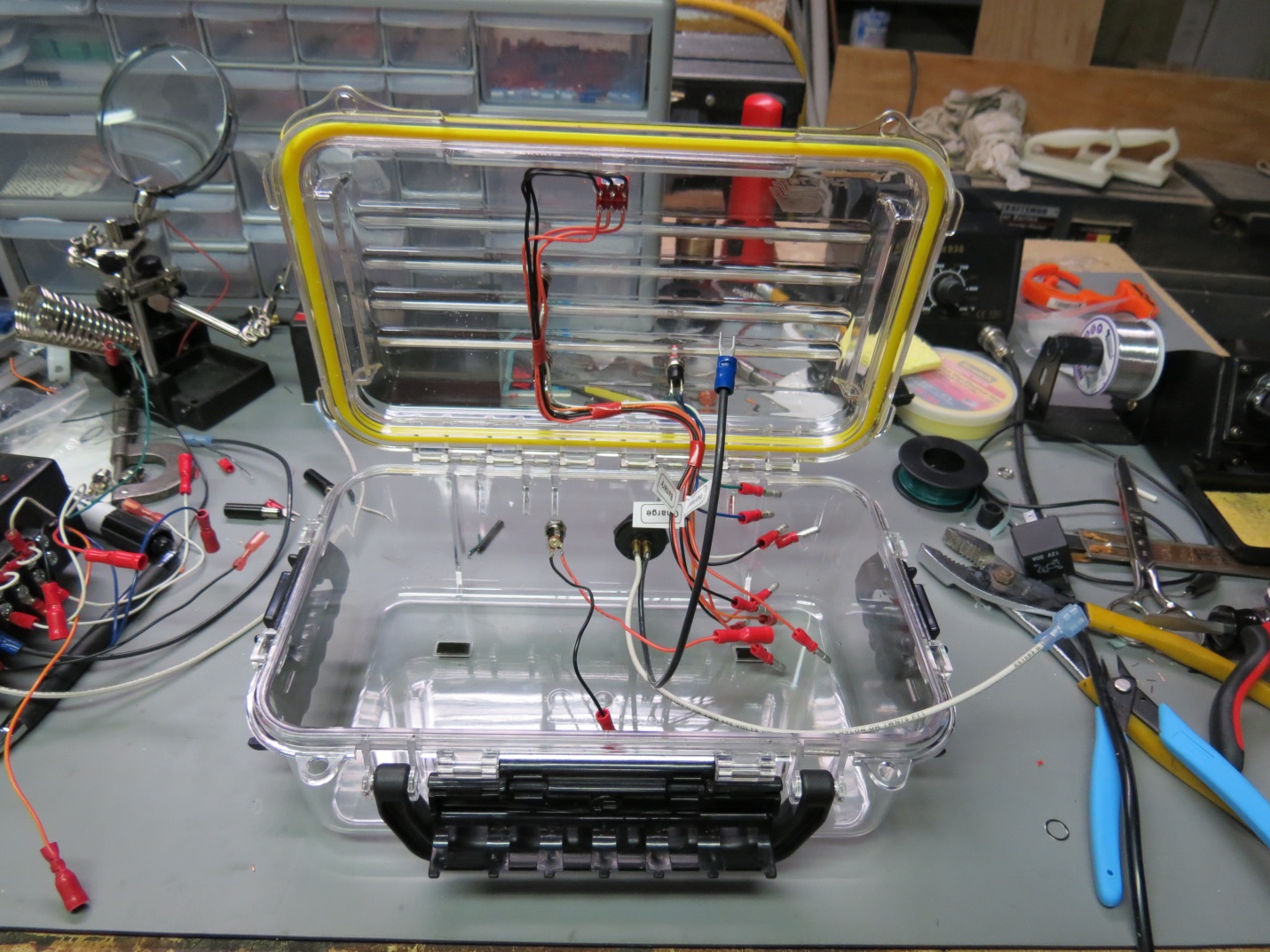
****

Figure 15 - Use of Command Brand Strips

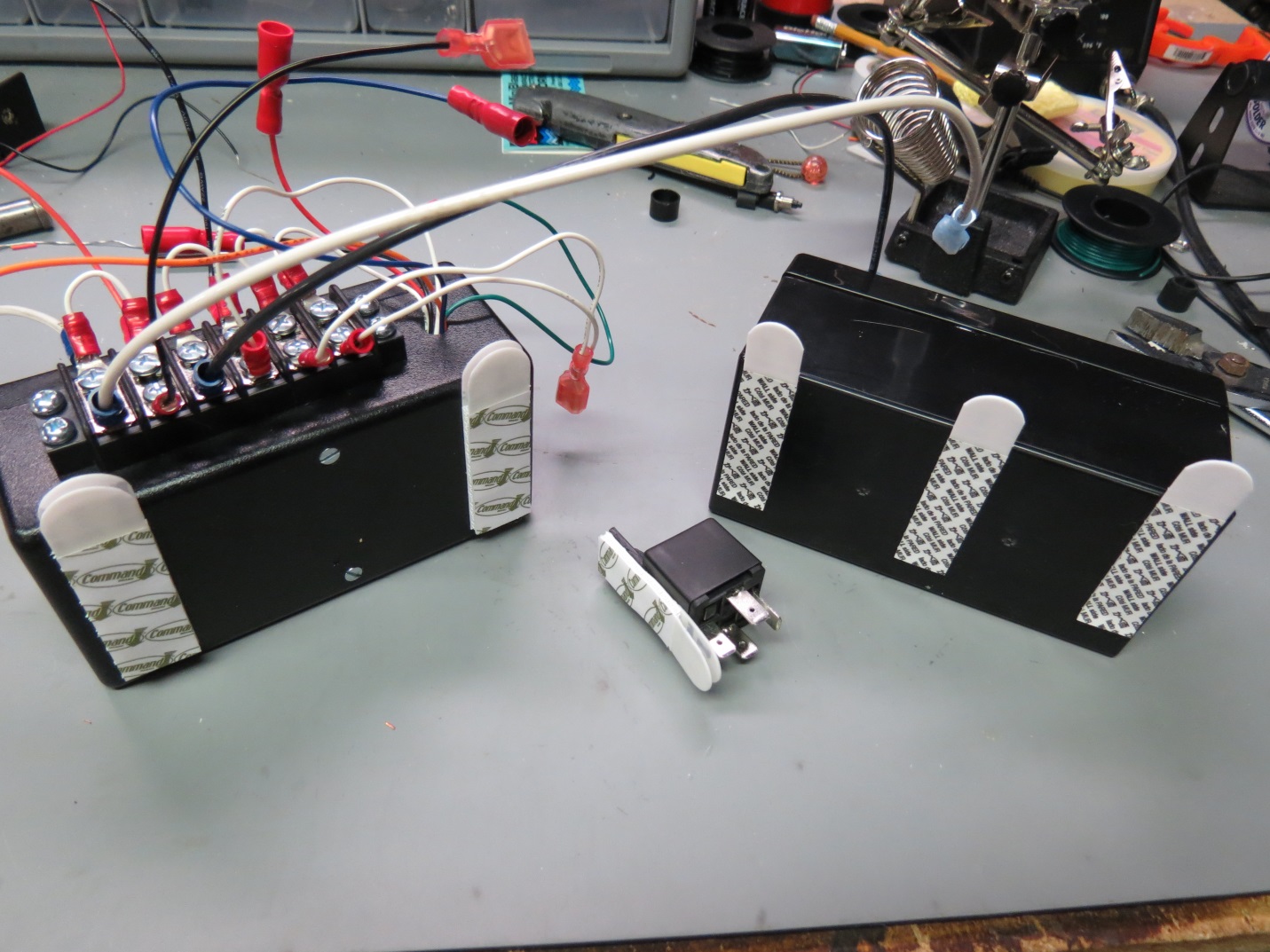
****

Figure 16 - Terminal Strip Wiring

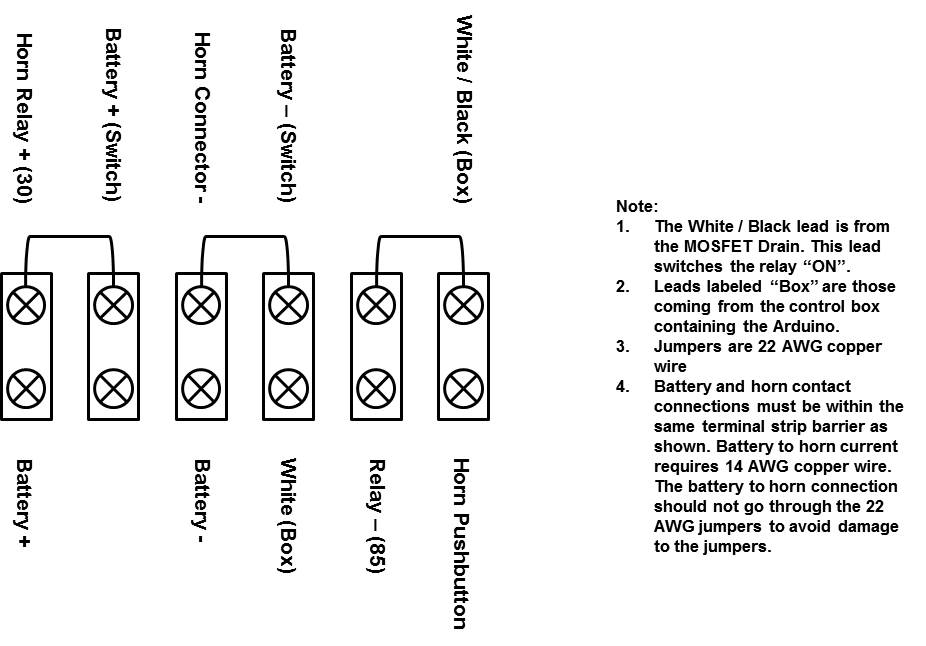
****

Figure 17 - Case Wiring Summary

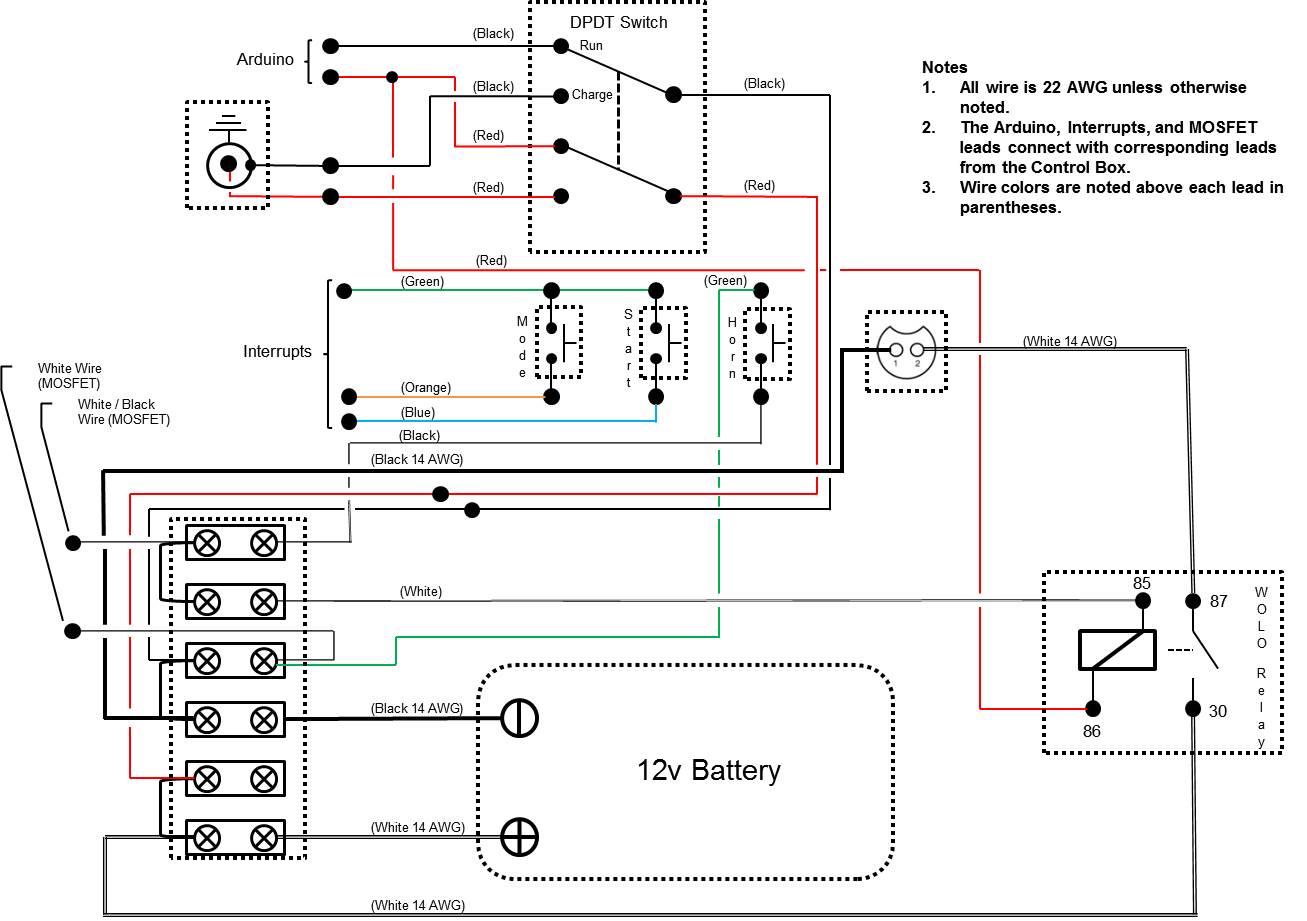
****

Figure 18 - Horn Plate

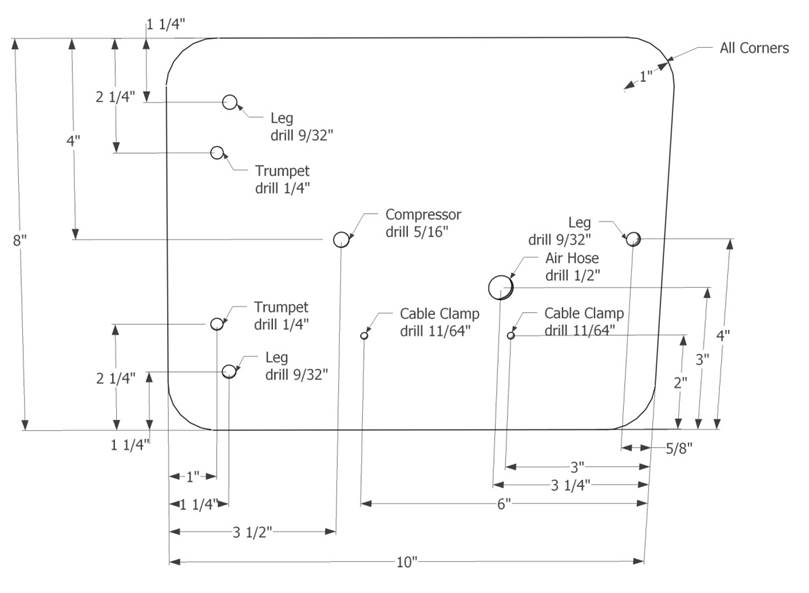
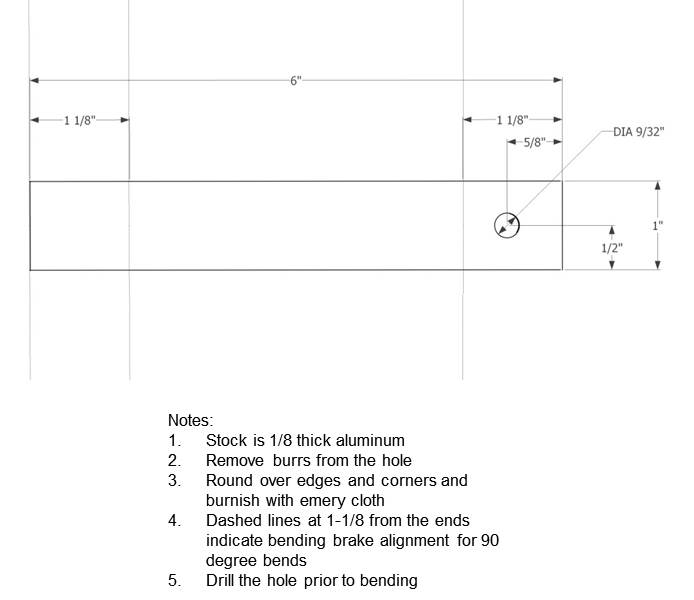
****

Figure 19 - Horn Plate Legs

****

# Appendix B - GNU General Public License, version 3 (GPL-3.0)

0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

* a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
* b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.
* c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
* d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

* a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
* b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
* c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
* d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
* e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

* a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
* b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
* c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
* d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
* e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
* f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's “contributor version”.

A contributor's “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.