Fog Horn Control

Bluetooth Enabled

Harry Keith

Copyright 2025

Fundamental problem statement: I have a sailboat without any horn. I have a VHF with loudhailer capability that can make fog horns, but none are loud enough to meet COLREGS, the VHF isn’t easy to figure out, I don’t want a loudhailer, and I’d like a “horn button” at the helm station. I will mount a basic horn under the radome on the mast, and I need wiring/controls.

Concept: Control is by a simple ESP32 microcomputer (a “baby Arduino” that uses the Arduino building blocks but is less powerful, smaller, and cheaper). It will output to a high side MOSFET, that is turned “on” by taking the GATE low. This allows the helm station to send a very low power ground signal to activate the control, requiring only a single wire of any size (actual power required to control a FET isn’t measurable). Initial plan was a 4-way switch (Off/Power /Sail/Anchor), but mounting the switch was a bit of a conundrum. Then I had the idea of a cell phone app to control by Bluetooth. For me, a fog horn is an exceptionally unusual event. The complication of using a cell phone is trivial compared to finding a nice place for a switch. Total cost for me was under $10, because I had the miscellaneous bits. Total cost including bits is under $30 if you can get buddy to 3D print the case.

The control circuit is built using MIT App Inventor, which generates a \*.apk. Send it to your Android phone, open it (and approve installation from unknown sources). It works well, but it is not great on error handling. If the Bluetooth is disconnected, reconnect it. If you try and turn it on when the ESP32 isn’t near, it fails – when you get close, reconnect to Bluetooth.

The timing for the fog signals includes up to a 15 second random length plus/minus variation, to ensure you don’t mimic someone else.

Construction – in extreme detail to help as many beginners as possible.

* You will need to 3D print the box. This is quick and easy, and can be printed in any material. If you don’t have a printer, you may have a friend. Worst case is print services on line. The filament (AKA, “ink”) is inexpensive – this project uses about $0.50.
* If you use a TO-92 format transistor, hot glue the flat side of the transistor to the underside of the ESP32, with the leads pointing toward the USB connector. For these instructions, orient the ESP32 with the USB connector “UP.”
* Take the right transistor lead and solder it to the second pin down on the right. This is labeled (on the other side) “Gnd.”
* Take the middle transistor lead and solder one end of a resistor to it. Slip some heat shrink over to protect wires.
* The other end of the resistor is soldered to third pin down on left. This is labeled “D13.”
* Take the left most pin, solder a 6” piece of thin stranded wire to it, and cover the joint with heat shrink.
* Spread hot melt glue liberally. It provides moisture protection (?), guards against short circuits, and holds all the parts in place.
* Take the other end of the thin wire, strip about an inch, form a small loop in the end, and solder it to provide some rigidity.
* Slide a 3/8 #6 screw through the hole labeled “SW” (Switch), lay the loop of wire on it, add a nut, and draw the nut into the plastic. Heating the nut with a soldering iron will help. Note that there is a slot in the case for the wire to lay into.
* Using a 3/8 #6 in the hole labeled “+” and a ¼” #6 in the hole labeled “HORN”, seat the other two nuts.
* Add a dab of hotmelt glue, or a drop of super glue, to encourage all nuts to stay in place.
* Take the FET, and using a small wire cutter, cut off the center pin. Take the tips of the remaining ends, and bend them in a short 90-degree toward the outside (test fit for location). Solder the ends of R2 to these two leads (I made loops in the R2 leads to go around the screws, then soldered – it helps hold it all in place).
* Set the ESP32 on the pins, and snap the lid in place.
* Connect the control to the computer by a USB-C cord (or Micro, as appropriate).
* Install the Arduino Integrated Development Environment (“IDE”) and the ESP32 support. Lots of guides on the ‘net, but this is one:
* <https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>
* Open the IDE, select the ESP32 board you bought (I don’t think it matters much for our purposes, but I selected the “ESP 32 DEV MODULE”).
* Connect the IDE to the ESP32. This may be tricky. You may have to install a driver, and determine which COM port you are using. If you are a complete beginner, this may be a challenge. But anyone at all comfortable with Windows configuration can get it working.
* Open the sketch (the \*.ino file).
* Select Sketch/upload to compile and install on the ESP
* If you use the switch driven version, a 1P4T switch is needed. Common is connected to a GND pin, Motor is connect to Pin 25, Sail to Pin 26, and Anchor to Pin 27. Details are left to the intrepid builder.
* Download the \*.apk file to your Android, and install it. You may be asked for confirmation, as it is not a Google approved app. There is no support for Apple products

Note, I don’t have an iPhone and don’t care, but the MIT App Inventor doesn’t work on Apple and from what I can tell there are mandatory fees/headaches to develop for an iPhone. If anyone wants to make an app, it’s super simple. Connect to the ESP32 bluetooth instance, and push “0” for off, “1” for Power, “2” for Sail, and “3” for Anchored.

Final hookup:

* Power to the “+” terminal (the top terminal, labeled “+”). Fuse according to the horn.
* Heat Sink (the left terminal, labled “Horn”) goes to the horn power
* If you wish to use a manual horn button, it goes to the bottom screw, labled “SW” (switch). The switch connects this terminal to ground.
* You will need a USB-C power supply. It MUST share the same ground as the horn circuit (this is normal for most boats).

Supplies (none of the specs are critical– use what you have/can find):

* ESP32. I got mine on Ali for $3.50. I used the Type-C, it also comes with Micro-USB input. https://www.aliexpress.us/item/3256805517875317.html
* TO-220AB (this is a “shape” code, not a product code) enhancement mode P-FET. I used what I had on hand. Here is a 10-pack for $8. https://a.co/d/9upsfjG
* NPN transistor. Any small NPN is fine, I used what I had. The TO-92 is the right form factor. Here’s a 25-pack for $6.45.
* Two 10k Ohm resistors. Any wattage, 1/8W fits nicely (actual power is about .01W). Any resistance between 3k and 20k Ohm is fine, it’s not critical. I had these on hand. Here’s a lifetime assortment for $8. https://a.co/d/7BrVdYQ
* USB power supply with USB-C connector, $3 on Ali: htttps://www.aliexpress.us/item/3256806725083769.html
* If you want the physical switch version, this is the switch I bought for that. You will need a knob of some sort. $2 https://www.aliexpress.us/item/3256803842053691.html
* Solder, a bit of light wire, two ¼” and one 3/8” #6 screws with nuts, and some wiring terminals. Fusing to support your horn(5-10A?)

Theory of Operation.

First, my degree is mechanical engineering. I took courses in EE, but that is NOT my degree or my career. So this is a functional design, not a “clean” design. I’m sure there are those that will take exception to it. But, that aside.

* The ESP program controls the system by Pin 13, set high for “ON” and low for “OFF.”
* At idle, the Gate of Q2 is held high by 12V through R2, ensuring Q2 is OFF
* When Pin 13 is high, 3.3V is sent through R1 to the Base (labeled “B” on the schematic) of Q1, and Q1 conducts (connects Collector “C” to Emitter “E”). The 10k resistor (R1) serves to limit current through Q1 (“B” to “E”) to about .15mA, protecting Q1. When Q1 conducts, the Gate (labeled “G” on the schematic) of Q2 is taken low. R2 limits current through Q1 (“C” to “E”) to .15mA.
* When Gate of Q2 (“G”) is low, Q2 conducts (connects Source “S” to Drain “D”). FETs are virtually switches – unlike transistors, they are OFF or ON (resistance measured in small milliohms). (This is technically not true, but for our purposes it is true).
* When Pin 13 goes low again, Q1 turns off, “G” is pulled back to B+ by R2, Q2 turns off.
* Optionally, a small gauge wire can be run from Q2 “G” (Labeled “SW” on the box) to a switch at the helm, and then to ground. When this switch is closed, it pulls “G” low, and turns on Q2, activating the horn.
* As an aside, these TO-220AB FETs are cheap, super powerful, and super simple. This is the 3rd time I’ve used one on the boat. I have one turning on the fans on my heating system (driven by the heater computer), one driven by my BMS to disconnect my alternator from my LFP bank, and now one driving my horn. The application is easy and reliable for small Amp circuits (larger circuits may require heat sinks).



