

---

## The FUnicorn Arduino-Compatible Desk Enhancer

---

### FEATURES

- ◆ Majestic gold unicorn
- ◆ Beautiful cursive lettering
- ◆ Unusual cyan-colored horn tip LED
- ◆ Cardboard stand
- ◆ Smoked acrylic LED obscurer for maximum surprisability
- ◆ Gorgeous rainbow USB cable
- ◆ Capacitive touch activated
- ◆ Arduino-compatible
- ◆ Power via USB, wall adapter, or battery
- ◆ Connector for Big Red Button add-on
- ◆ Shield pattern for easy expansion

### APPLICATIONS

- ◆ Telling off your boss
- ◆ Telling off your coworkers
- ◆ Telling off 2020
- ◆ Telling off your family
- ◆ Telling off your friends

### DESCRIPTION

Ever had one of those days where you just have one thing to say to the world? Well, we at Alpenglow Industries have the solution. Let the Unicorn say it for you.

The FUnicorn (we pronounce it "eff - unicorn") is an Arduino-compatible board designed both for laughs, and to teach a



beginner a little bit about electronics and programming. 120 LEDs spell out its special message in glorious cursive, making it incognito for little ones because no one's teaching that shit in school anymore. The FUnicorn ships with a cardboard stand, an acrylic shield to hide the LEDs until lit, a USB cable, and capacitive touch code that cycles through 5 blinking patterns. Just touch the Unicorn to light up the message!

You can add on a Big Red Button, because who doesn't love mashing one of those? This enables the Unicorn to be battery-powered as it can sleep between button presses, while the capacitive touch code needs to be continuously powered.

Add sensors and other types of activation to your heart's content via the Arduino shield pattern on the back!

## BACKGROUND:

We made the original prototype as a gag gift for a white elephant gift exchange. We loved it so much that we decided to revise it a few times, make it Arduino-compatible, and release our potty-mouthed unicorns out into the wild.

### Base Kit:

- FUnicorn Board
- Cardboard stand
- Smoked acrylic LED shield
- Beautiful rainbow USB cable
- Stickers

### Full Kit:

- All of the above
- Big Red Button (60mm arcade)
- Cable for button
- Big Red Button Box
- Sticker sheet with cutout templates and History Eraser decals
- Screws and a grommet in case you want to 3D print our Big Red Button Enclosure
- 9V battery
- 9V battery cable
- Arduino shield and ISP headers
- FUnicorn magnet and more stickers

### Software:

- available through the Arduino IDE
- on GitHub under Alpenglow Industries

## WHAT ELSE CAN YOU DO WITH IT?

Well, you can't change the message, those LEDs are all soldered in place. But you can change the blinking pattern to your heart's content. Make more patterns, make the sequence random, make a pattern trigger off a certain number of button presses or by holding the button down for a certain length of time. There's also a resistive touch "button" option if you short the unicorn's leg to its hoof, check out our GitHub repo for the demo code.

With the Arduino shield pattern on the back, you can also make it trigger off of just about anything imaginable:

- add an ultrasonic distance sensor and make a Social Distance Unicorn that goes off if anything comes within 6 feet
- add a Raspberry Pi and Google AI Voice kit and make it voice-activated. Because nothing is as satisfying as saying "Hey Google, F@#! You!"
- add a microphone input so that ANY noise will set it off
- add a motion sensor and put it in a window near your front door to tell off visitors
- add a thermistor to let people know how much you hate working in an unairconditioned office in the summer



## ELECTRICAL CHARACTERISTICS

RefDes	Parameter	Typical Value	Units
J4	Input Voltage ( <i>note 1</i> )	3.3 – 17	V
J5	USB, Data & Input Voltage	5	V
J3	ISP & Input Voltage ( <i>note 2</i> )	3.3 – 5	V
J2	Illuminated Button Connector	Pinout (pin 1 is at top): 1 – 3.3V (LED power) 2 – Switched GND (for turning LED on/off) 3 – Button signal (pulled high) 4 – Button GND	
S1	Reset Button	Open (pulled high) = Normal Operation Pushed (low) = Reset	
U2	ATMega328P I/O Level	3.3	V
U1, U3, U4	RT9193-25GB Regulator Output	2.5	V

*Note 1: It can be powered off of a Li-Ion battery but you may not be able to use all the energy capacity. A 9V battery works quite well, so does 3 AAA batteries in series.*

*Note 2: Only use the ISP connector if you know what you're doing. You can use a 5V or a 3.3V programmer and you can power the board via the programmer, or use a programmer that polls the board voltage. Yes, we thought of everything here.*

For more detailed characteristics, please refer to the ATMega328P datasheet.

## SCHEMATIC & BILL OF MATERIALS

See separate documents



## SET UP THE UNICORN:

### The Base Kit:

1. Unpack the components from the box. You'll have the FUnicorn board, 3 cardboard stand pieces, an LED shield, and a rainbow USB cable.



2. Slide the 3 cardboard pieces together to make the stand

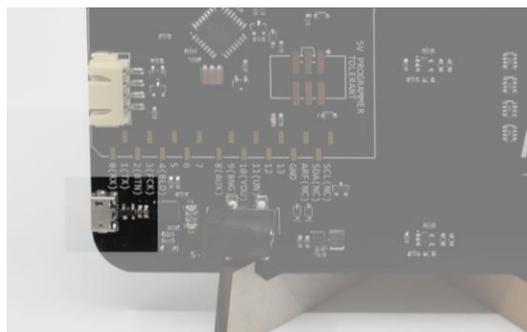


3. Set the FUnicorn board on the stand
4. Peel the protective cover from the LED shield

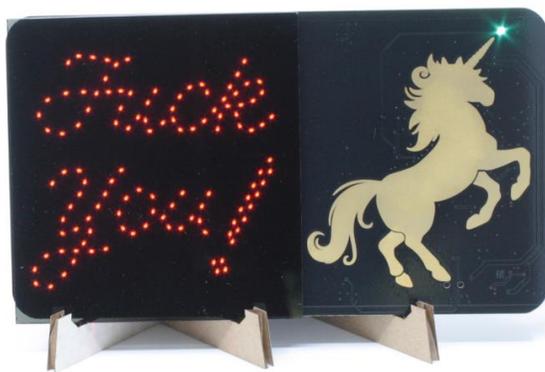
5. Set the LED shield on the stand, over the LEDs



6. Plug in the USB cable to a charger or computer port



7. Wait for the unicorn's horn to pulse
8. Touch the unicorn!

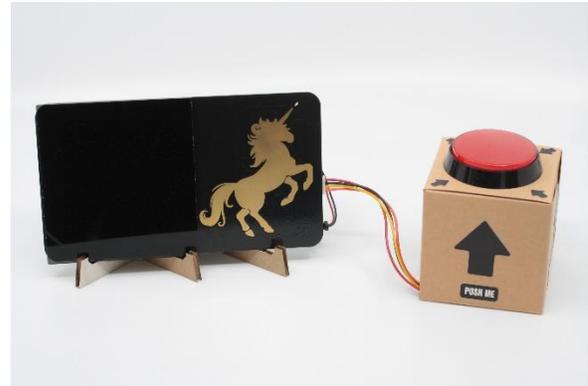


**The Full Kit:**

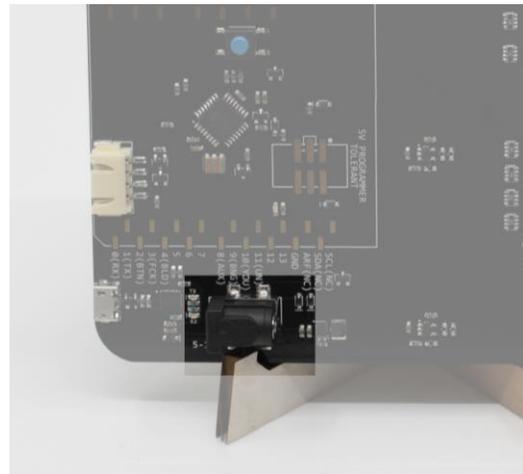
1. Unpack the components from the box. You'll have everything above plus a Big Red Button Kit, a 9V battery and cable, and Arduino Shield and ISP headers.



2. Set up the FUnicorn as in the Base Kit instructions
3. Assemble the Big Red Button Kit per its instructions (separate pdf)
4. Plug the Big Red Button into the FUnicorn board



5. Now you can activate the message by either touching the Unicorn or pressing the Big Red Button.
6. If you want to run the Unicorn off of battery power, unplug the USB cable and plug in the 9V battery (cable included).



Then press and hold the button for the duration of a blinking pattern. The horn will blink twice, indicating that it's now in low power mode. Note that capacitive touch on the unicorn is no longer enabled. To re-enable capacitive touch, power cycle the FUnicorn or press the small blue reset button on the back.

## HOW IT WORKS:

### Turning on the LEDs:

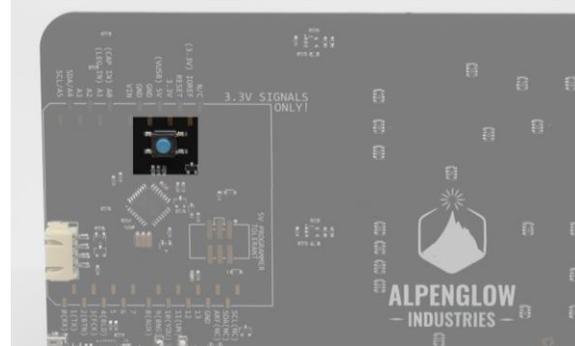
The red LEDs are broken into 3 banks: FUCK, YOU, and ! (called “bang” in the software). Each bank is hooked up to the output of a linear low-dropout voltage regulator, which supplies them with 2.5V. The enable pin on each regulator is hooked up to the processor, and this is how the words are turned on and off. Additionally, the processor pins are timer outputs, so you can PWM the enable line and make them fade in and out, or “glow”. The horn LED is separate and driven off of 3.3V because of its larger forward voltage drop. It’s also hooked up to a timer output so it can be PWM’d. See the schematic for more details.

Each time the unicorn is activated, it will light up the LEDs. There are 5 built-in blinking sequences that it will cycle through.

### Activation methods:

There are 3 built-in methods to activate the LEDs. The code the FUnicorn is shipped with activates the message by either touching the unicorn, or by attaching a button and pushing it. The capacitive touch requires a continuous power source and will drain a battery quickly. The FUnicorn can be put into a more battery-friendly low-power mode by pressing and holding the button for the duration of a blink sequence. The horn LED will then blink twice, indicating it’s now in low power mode and that capacitive touch is turned off. To turn capacitive touch back on, either press the small blue reset

button on the back of the unicorn, or cycle power.



Reset Button

### Capacitive Touch Code:

The FUnicorn uses the ADCTouch library to continuously charge, discharge, and read the voltage on the pin attached to the unicorn. It serves as a very basic single-pin capacitive touch “button”. Because of the layout of this board, it has low sensitivity, so unlike other capacitive touch buttons that can potentially activate when your finger is still some distance away from the board, this one won’t activate before the unicorn is actually touched. The activation threshold can be changed in software, but the default should work in most environments. If you have trouble activating it, try touching closer to the nape of the unicorn’s neck, or try touching the unicorn with your thumb, and touching the back of the board with your other fingers on the same hand.

### External Button:

This is the most straightforward way of activating the message, and our Big Red Button kits make it even more fun. It’s the best for battery operation as the button is hooked up to a hardware interrupt pin

which can wake the FUnicorn up from sleep. For our low-power button mode (or the Button-only Code), after the blink sequence is done, the FUnicorn goes to sleep until the button is pressed again.

### Resistive Touch:

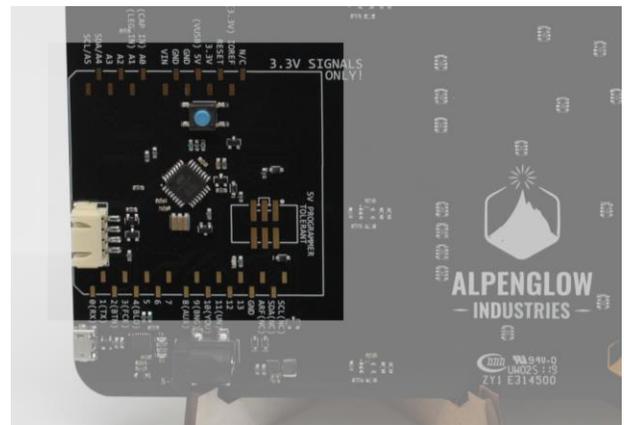
Yes, this is also an option! The unicorn's lower leg is attached to a processor pin and pulled high. The unicorn's hoof is attached to GND. When you bridge between the two of them with a slightly moist finger, your finger will conduct and pull the leg to GND, activating the LEDs. Note that this is separate code that you'll have to load. It tends to be more dependent on each individual's skin, and less reliable than the capacitive touch. But it's fun to play with and learn about! You can also change the activation threshold in software.

### Modifying the blink patterns or triggers:

You'll first need to follow the instructions below for loading the Arduino IDE, FUnicorn Board, and FUnicorn library. The main code and blinking patterns are located in FUnicorn.cpp, with supporting info and nomenclature in FUnicorn.h.

In general, you'll always want to place a call to `initFUnicorn()` in `setup()`. To have an external button wake the FUnicorn from sleep, also place a call to `initButt()` and `sei()` in `setup`, and copy and paste the `ISR` and `checkButt()` functions from `FUnicorn_ShipCode.ino`.

To add different triggers, you can solder on the Arduino shield headers and plug in a shield or wire up different sensors. You'll need to add your own code to set up the pins or communications, and there's sample arduino code for practically every sensor out there, just google around and look on [arduino.cc](http://arduino.cc).

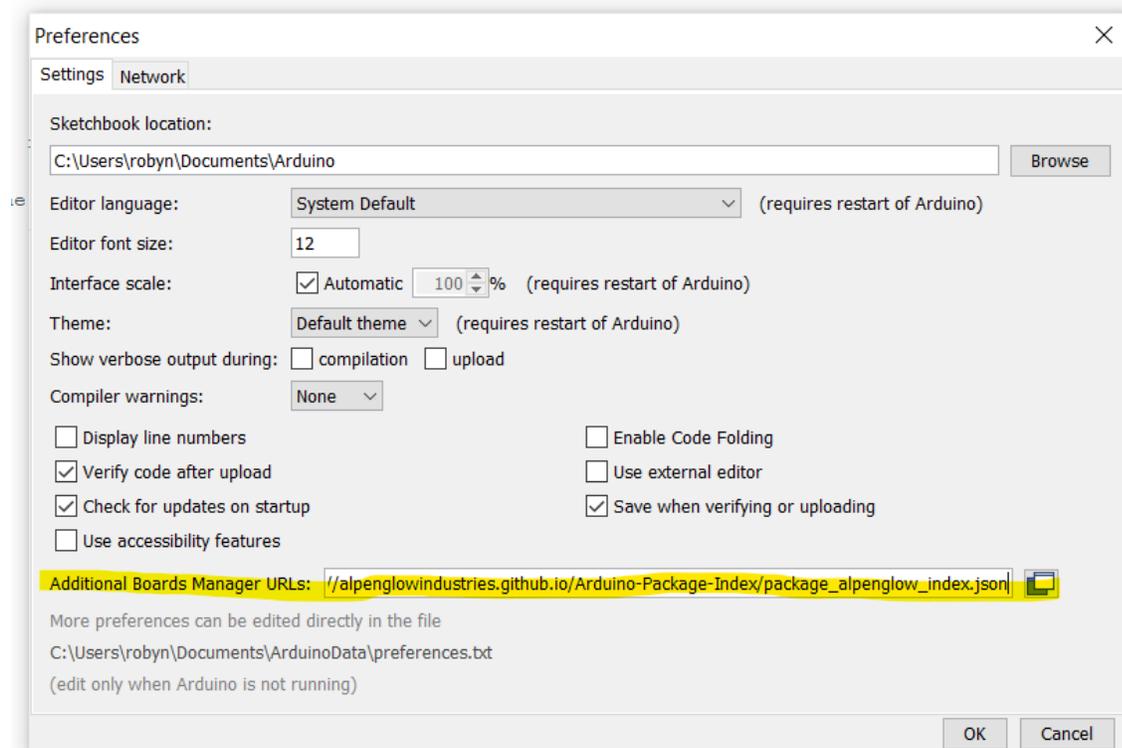


Arduino Shield Pattern

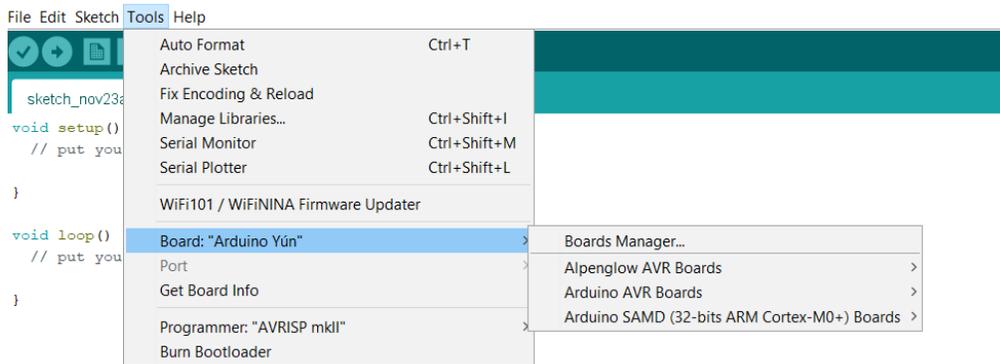
## USING THE ARDUINO IDE TO REPROGRAM THE FUNICORN:

There are several steps to get the Arduino IDE downloaded and set up to work with your new FUnicorn board the first time. You won't have to repeat them all every time!

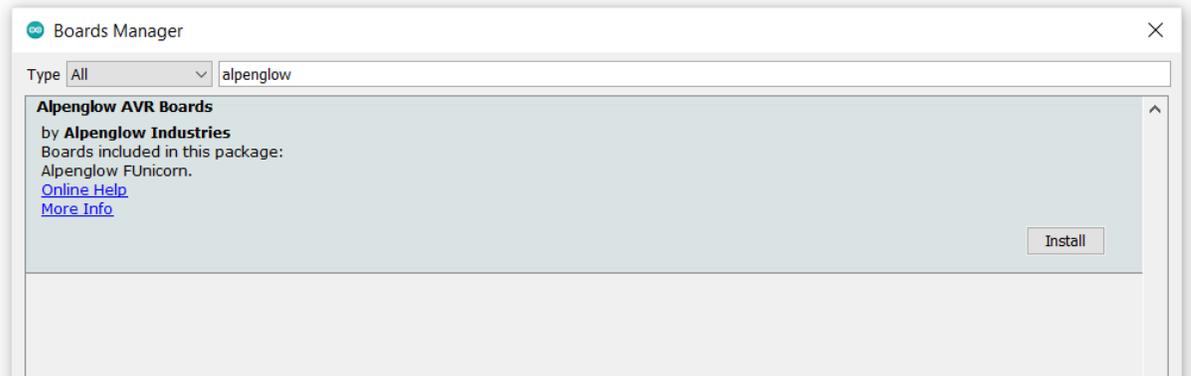
1. Download the Arduino IDE software from [www.arduino.cc](http://www.arduino.cc) and install it on your computer.
2. Open the Arduino IDE
3. Add the Alpenglow Industries Boards Manager url to Preferences:
  - a. Go to File --> Preferences
  - b. Add the following url to "Additional Board Manager URLs":  
[https://alpenglowindustries.github.io/Arduino-Package-Index/package\\_alpenglow\\_index.json](https://alpenglowindustries.github.io/Arduino-Package-Index/package_alpenglow_index.json)
  - c. Click OK, OK to exit



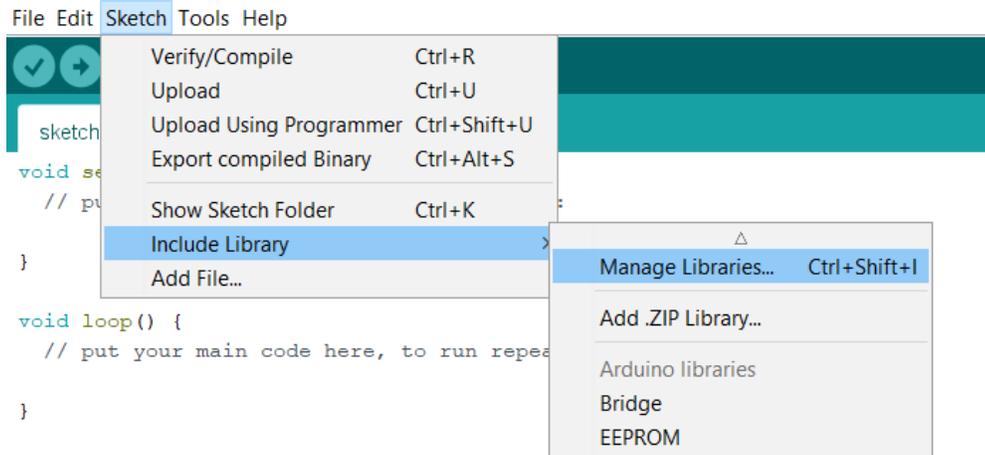
4. Add Alpenglow Industries boards to Arduino
  - a. Go to Tools → Board → Boards Manager  
(It does not matter what board is currently selected under Board: "BoardName")



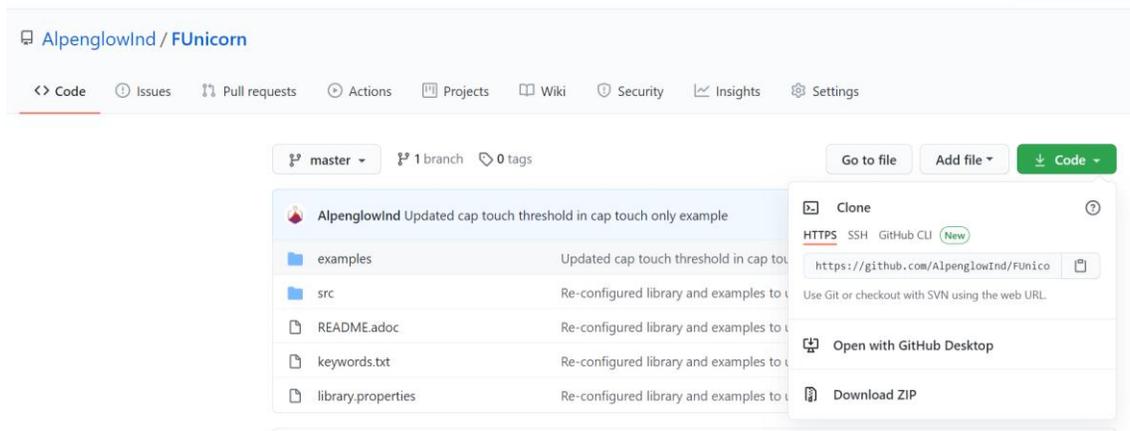
- b. Search for "alpenglow" in the search bar
- c. Click Install for Alpenglow AVR Boards



5. Add the FUnicorn library to Arduino
  - a. (coming soon!) Add through **Library Manager**
    - i. This is the preferred method since you'll automatically receive updates
    - ii. Go to Sketch → Include Library → Manage Libraries



- iii. Type "alpenglow" into the search bar
  - iv. Install Alpenglow FUnicorn
- b. Add through **Add .ZIP Library**
  - i. Go to <https://github.com/AlpenglowInd/FUnicorn>
  - ii. Click Code → Download ZIP

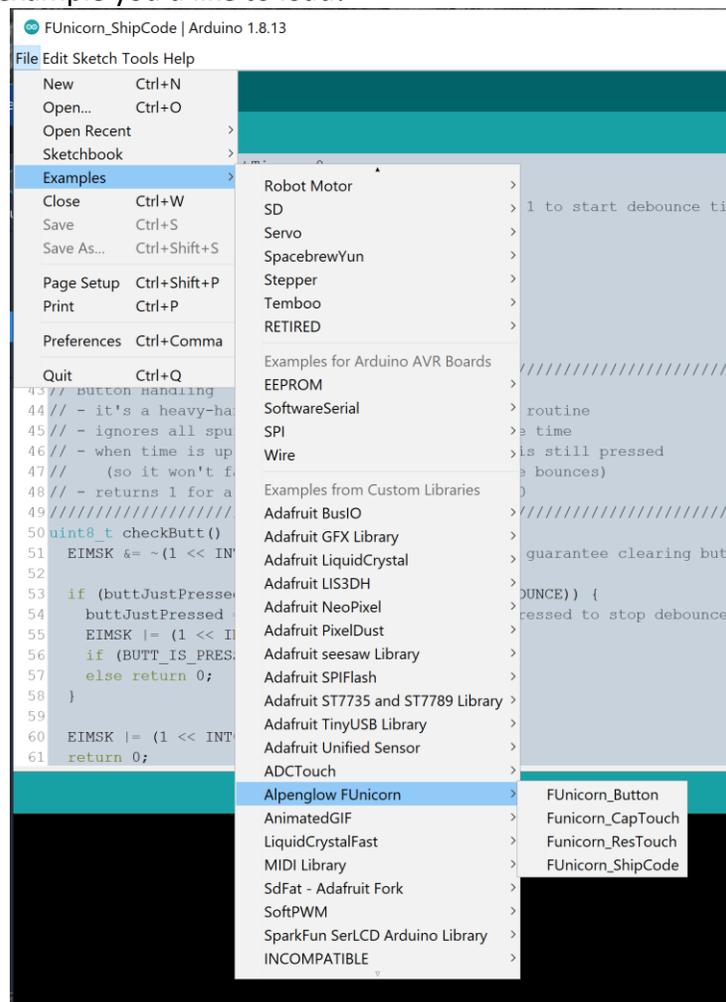


- iii. Go to Sketch → Include Library → Add .ZIP Library.
  - iv. Select the .zip folder you just downloaded

You're done with the once-only steps!

The following steps you'll need to either do or confirm whenever you want to load new code on your FUnicorn.

6. Loading the Code:
  - a. Plug in the FUnicorn to a USB port on your computer. USB drivers should automatically load if it's the first time you've plugged it into that USB port.
  - b. Go to File --> Examples
  - c. Under "Examples from Custom Libraries" select Alpenglow FUnicorn
  - d. Choose the example you'd like to load!

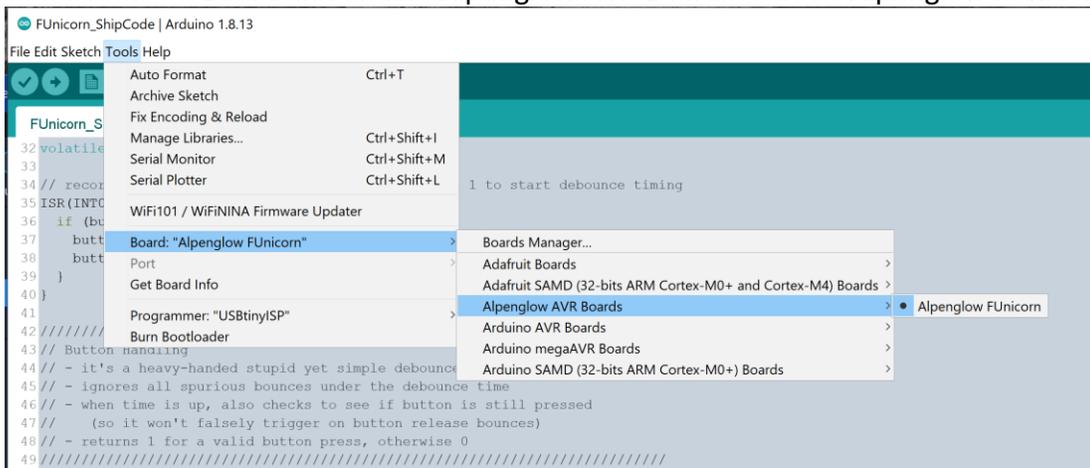


7. Code Examples:
  - FUnicorn\_ShipCode = the default code shipped with the unicorn, the message is activated via either capacitive touch or by pressing a button. Pressing and holding the button for the duration of a blinking cycle puts it into low-power button-only mode.

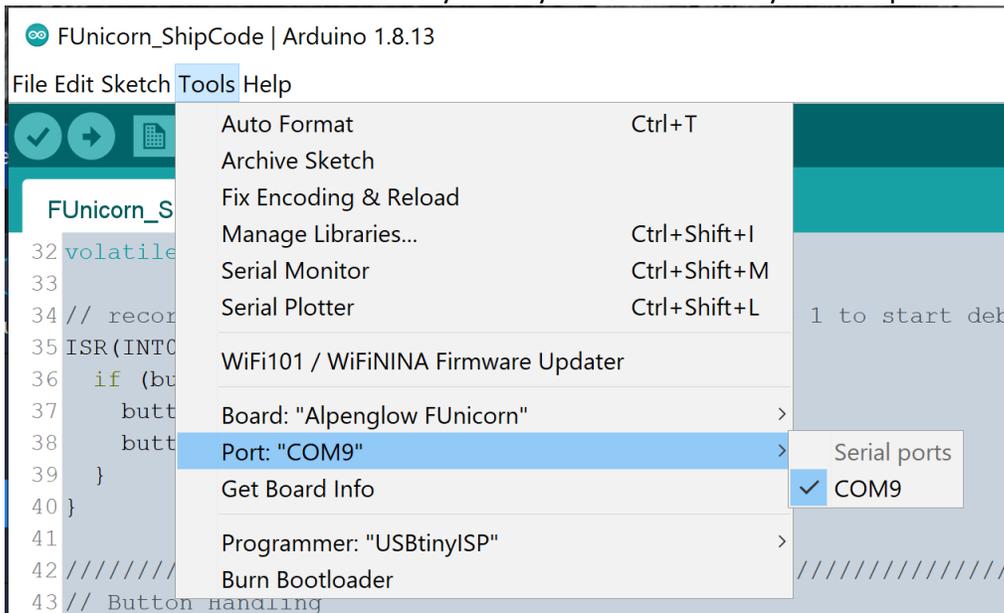
- FUnicorn\_CapTouch = capacitive touch only code.
- FUnicorn\_Button = low power button-only mode, good for battery power.
- FUnicorn\_ResTouch = sets up the leg and hoof as a resistive touch "button" if you touch them both together. Generally requires a moist finger. Maybe not the best for COVID times.

8. Send the code to the FUnicorn

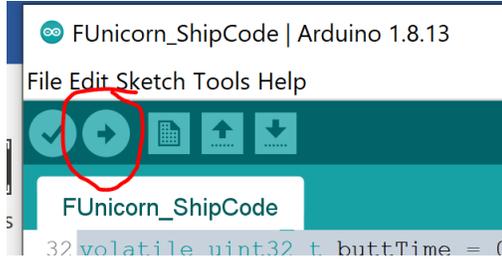
- a. Go to Tools --> Board and under "Alpenglow AVR Boards" select "Alpenglow FUnicorn"



- b. Go to Tools --> Port and select the COM port number of your Alpenglow FUnicorn. Make sure the correct one is selected – you may have several on your computer!



- c. Click the "-->" button in the toolbar

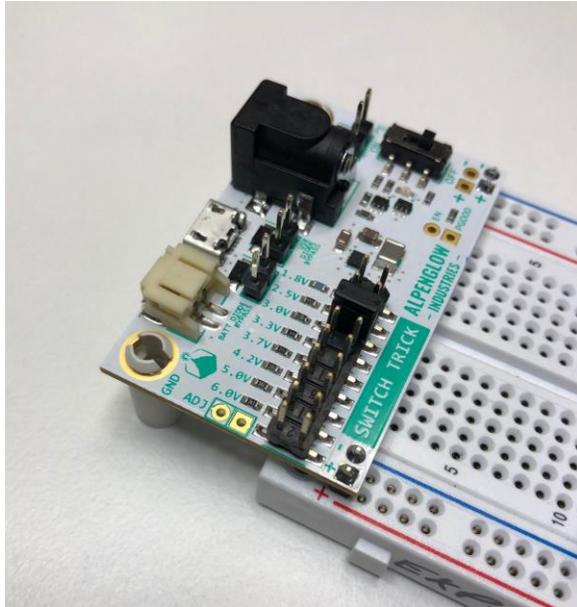


- d. When a "Done Uploading" message appears, the code is loaded!
- e. COM port drivers can sometimes crash. If your COM port isn't showing up, unplug the FUnicorn, restart Arduino, look at the list of COM ports under Tools → Port and write down the numbers. Click out of Tools → Port, then plug in the FUnicorn. Go back to Tools → Port, and the new FUnicorn COM port should be on the list. Select it. Click the → button on the toolbar again.
- f. Also note that your FUnicorn may show up as different COM port numbers when plugged into different USB ports on your computer.
9. If you're new to Arduino, there are many tutorials on [www.arduino.cc](http://www.arduino.cc) that take you through writing and modifying code, what Sketches are, what .ino files are, and what the setup() and loop() sections are, how to use libraries, and what basic C syntax looks like.



## YOU MAY ALSO LIKE

SWT-0001  
The SwitchTrick  
Breadboard-compatible switching power supply



IVT-0002  
I Voted!  
Pre-assembled PCB Blinky Badge



## CONNECT WITH US!

We love to see what you do with our products!  
Tag us in your posts and use #frivolouscircuits or #alpenglowindustries.

Instagram: @frivolous.circuits  
Twitter: @frivolous\_circs  
YouTube: Alpenglow Industries  
Hackaday: Alpenglow Industries  
Email: info@alpenglowindustries.com