



Quadcopter Spray Can Mod

Created by Becky Stern



<https://learn.adafruit.com/quadcopter-spray-can-mod>

Last updated on 2024-06-03 01:43:36 PM EDT

Table of Contents

Overview	3
Materials & Tools	4
Sprayer device	5
Power/servo circuit	11
Control servo with IRIS+	13
Settings for indoor flight	17
Sonar sensor cables	17
Fly it!	20
Prior attempt at standalone sprayer	21

Overview



This guide follows an attempt to make a spray painting quadcopter for indoor drone painting.

It is a series of experiments and observations, not quite a perfect step-by-step tutorial!
=D

Use this guide on your own property only! We do not endorse the use of this technology for any illegal painting.

This guide was developed with support from [NEWINC \(https://adafru.it/fkv\)](https://adafru.it/fkv), [Deep Lab \(https://adafru.it/fkw\)](https://adafru.it/fkw), and [3D Robotics \(https://adafru.it/cP3\)](https://adafru.it/cP3), and through collaboration with [Dan Moore \(https://adafru.it/fkT\)](https://adafru.it/fkT) and KATSU.

Materials & Tools



This project has many parts and optional components.

For the sprayer device:

- [Can Gun \(https://adafru.it/fkm\)](https://adafru.it/fkm) or [Spray Extender \(https://adafru.it/fkn\)](https://adafru.it/fkn)
- [Micro servo \(http://adafru.it/169\)](http://adafru.it/169) or [high-torque micro servo \(http://adafru.it/1143\)](http://adafru.it/1143)
- Paper clip or other small piece of wire
- Rotary tool for drilling mounting holes
- Zipties and/or E6000 adhesive

For the plug-and-play power circuit:

- Piece of [Perma-Proto \(http://adafru.it/589\)](http://adafru.it/589) or other perfboard
- [5V regulator \(http://adafru.it/2164\)](http://adafru.it/2164) or [UBEC Step-Down \(Buck\) Converter \(http://adafru.it/1385\)](http://adafru.it/1385)
- [Premium jumper wires \(http://adafru.it/760\)](http://adafru.it/760)
- Header pins
- Velcro tape
- Soldering station and supplies

General quadcopter stuff:

- [3D printed prop guards \(https://adafru.it/fku\)](https://adafru.it/fku) (now updated for IRIS+)
- extra propellers

- spray paint with straw from cleaning spray bottle
- stiff wire for mounting sprayer - we found that tomato cages work great

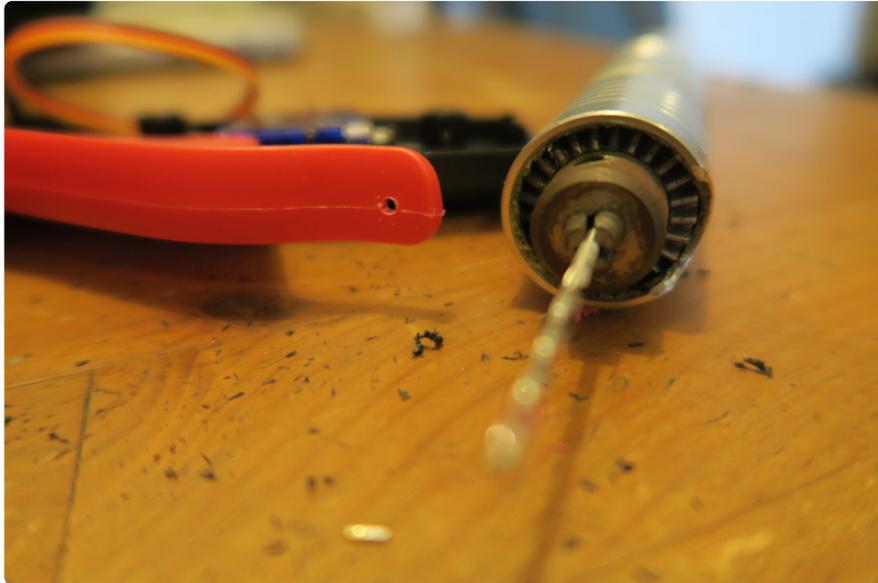
Sprayer device



This sprayer worked well for us but there are others available online or in your local hardware store that will work just as well! Open it up to reveal the mechanism inside...



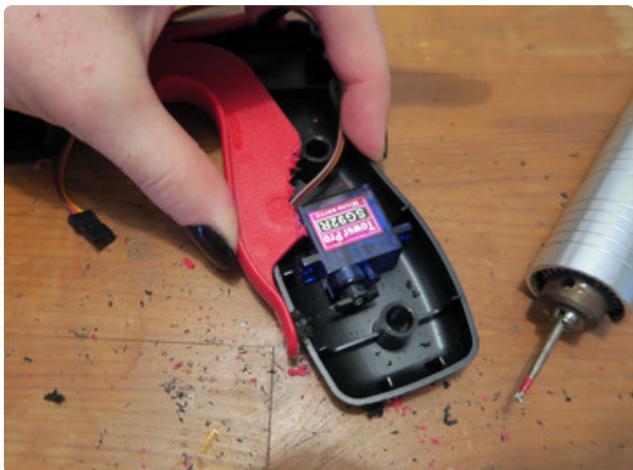
Test fit your micro servo in a location where the movement can pull the can trigger. Affix with E6000 glue or epoxy and allow to dry (quadcopter foot being used in above photo as a spacer only while the glue dries).

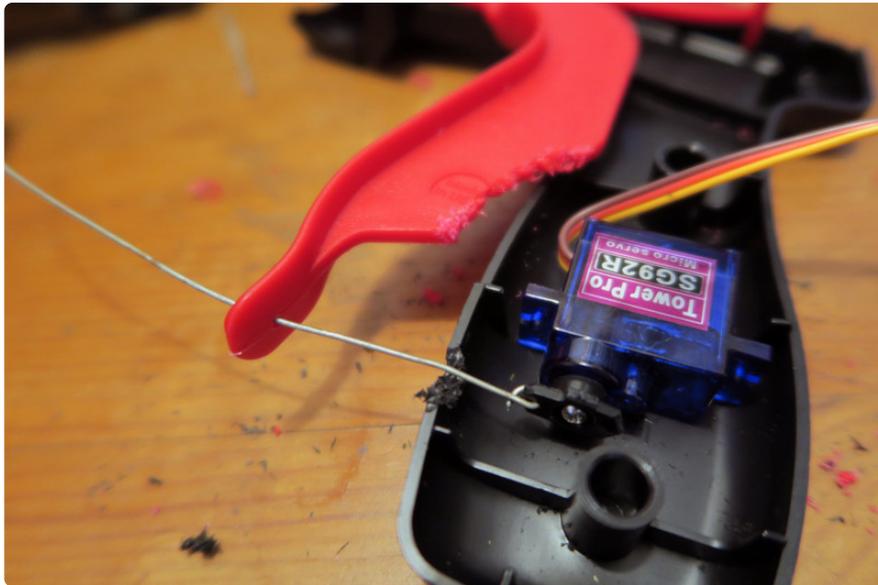


Drill a small hole in the trigger for affixing a wire later.



Cut the red trigger mechanism so it fits around the motor and doesn't intersect.





Attach a wire to the horn of the motor and run it through the hole in the trigger. It's also useful to route out a bit of clearance in the handle itself for the wire to move.



Route the servo wire out the top of the can gun and leave slack on the trigger wire for calibrating later.



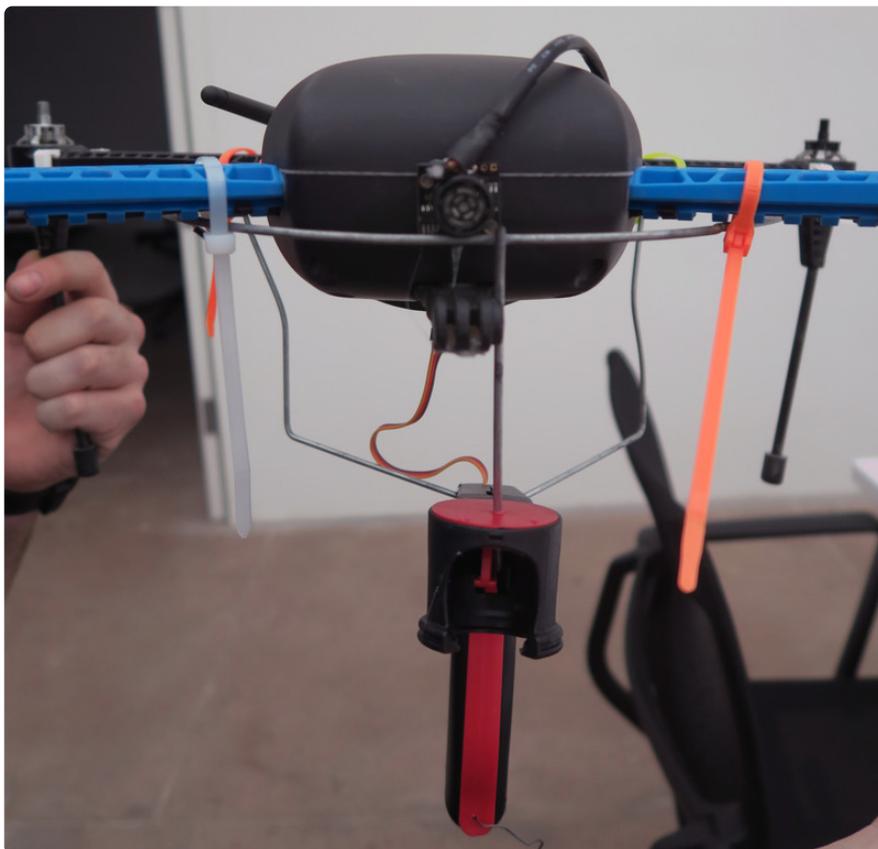
When prototyping, it's always helpful to have two of each part in case one breaks or needs troubleshooting! The can gun on the right suffered a small break while trimming the red trigger, which was fixed with pieces of a chopstick and copious glue. Learn from our mistake and only trim a small amount from the red trigger mechanism. =]



We also had success with [this type of spray trigger \(https://adafru.it/fgU\)](https://adafru.it/fgU), which was designed to attach to the end of a painting extension pole for spraying wasps' nests, etc. This type did not require any drilling, as the motor sits nicely at the pole junction, held in place with zip ties.

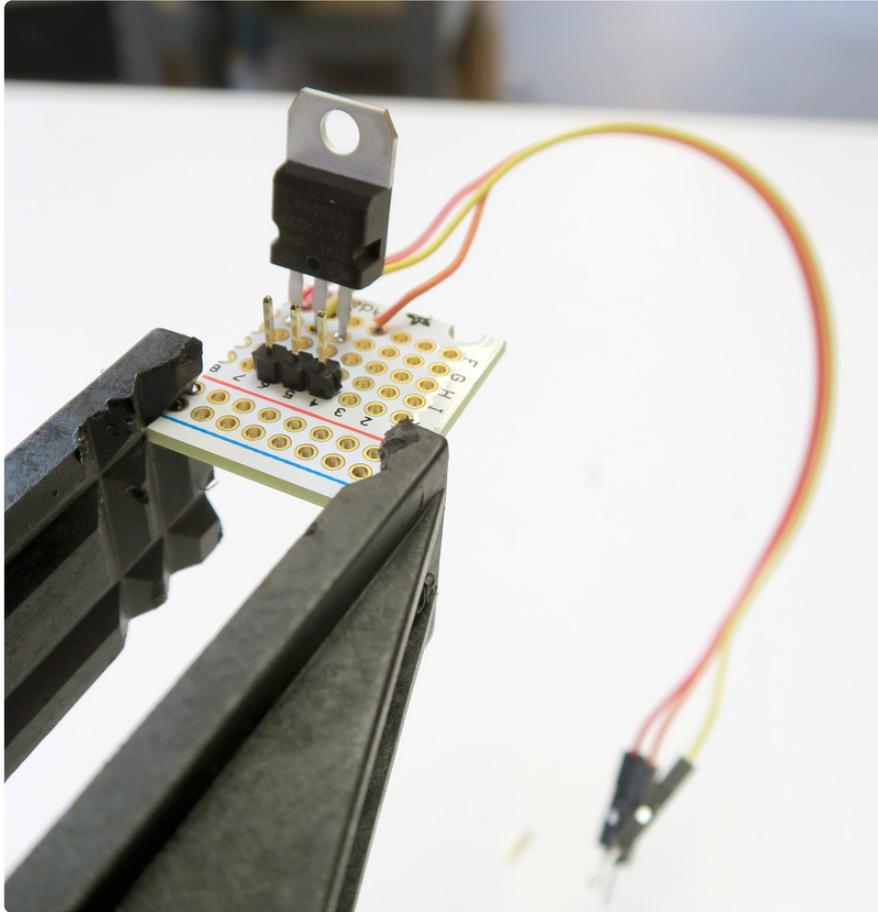


Holes do need to be drilled for mounting to the quadcopter, however. =]



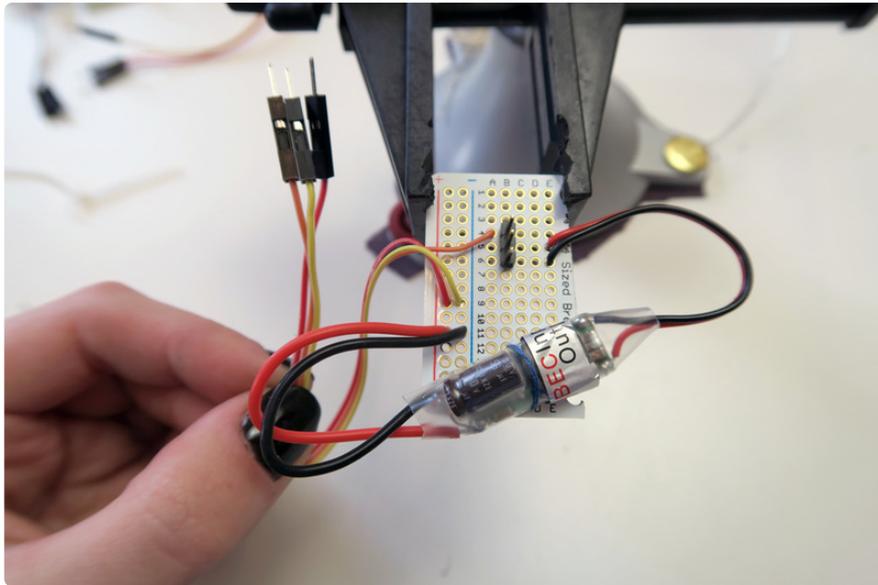
To mount to IRIS+, we used a tomato cage (available at most big box home/garden stores), cut down to size. The round shape lets it fit against the bot's arms and attach with zip ties, and the three straight supports come down and stick into holes on the plastic sprayer.

Power/servo circuit



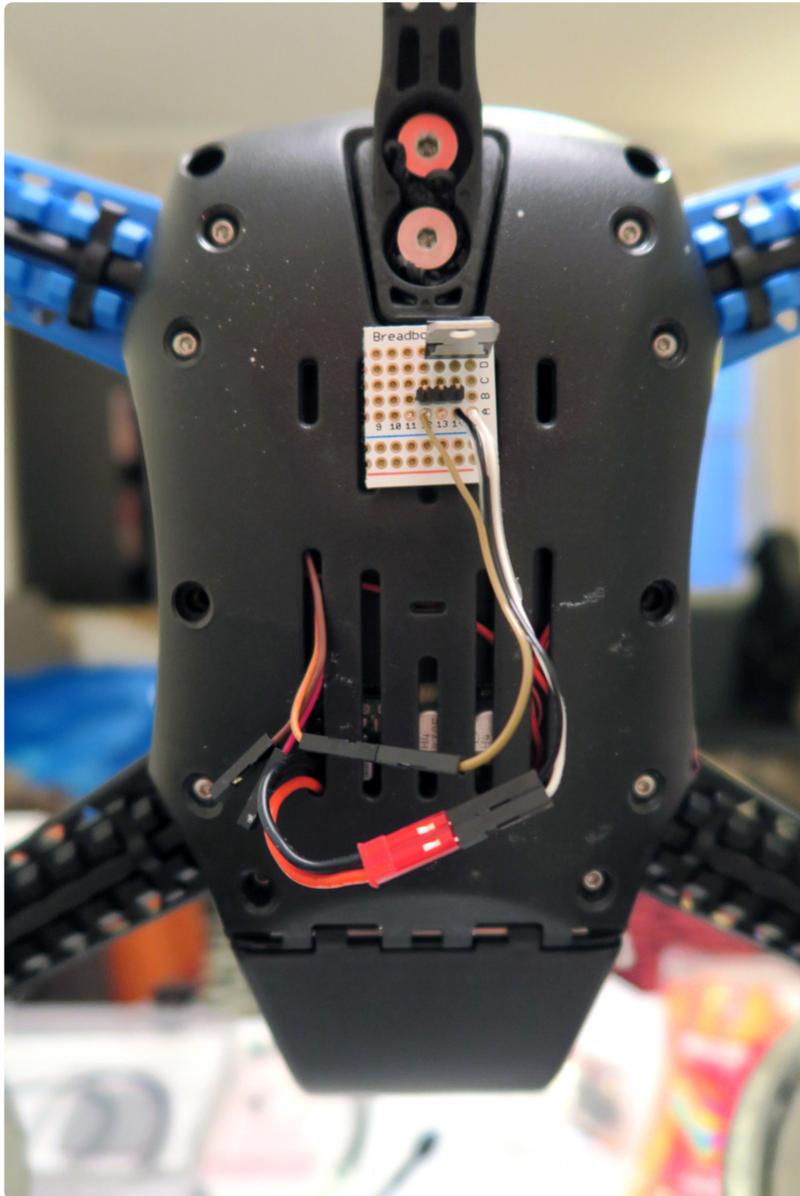
On a small piece of perfboard, solder a 5V regulator. Attach three wires with female headers to the perfboard: one to the regulator input (far left pin), one to the regulator ground (center pin), and one to a free row on the perfboard (servo control).

Solder three header pins to the perfboard in a row: to the voltage regulator ground, output, and the servo control wire.



If you want to control more than just a servo, use a buck converter! It can deliver more current without heating up. It is wired similarly to the voltage regulator where the 12V battery source and ground hooks up to the input, and the header pins wire up to the output + and -, as well as the servo control as above.

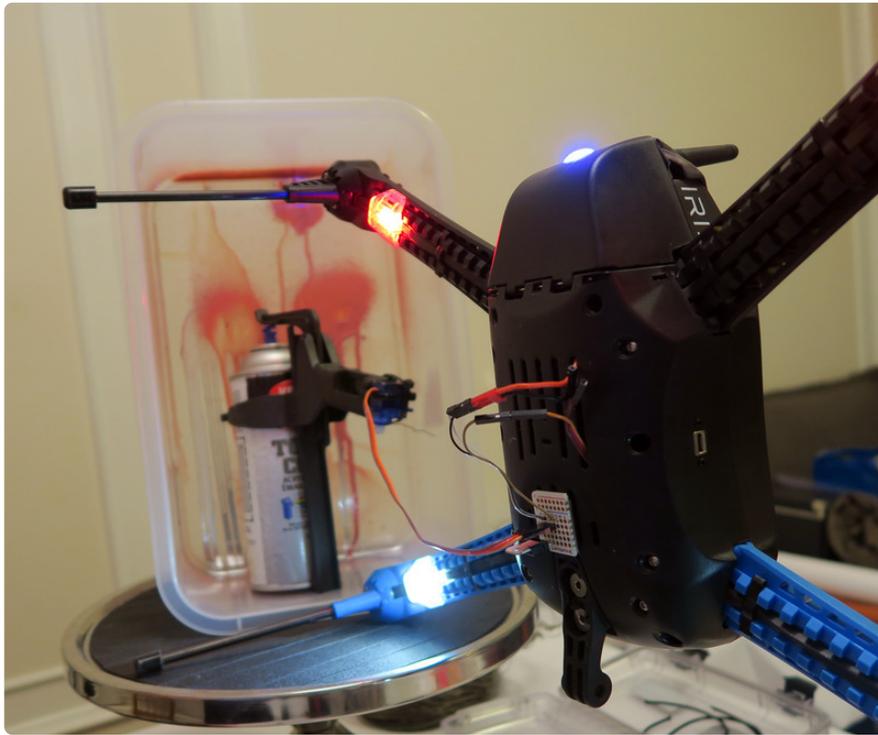
Control servo with IRIS+



Use Velcro tape to affix the PCB to the underside of IRIS+.

Connect the voltage regulator input wire to IRIS+'s red power wire, and likewise connect the ground wire to the black IRIS+ ground wire.

Connect the servo signal wire to the white wire on IRIS+ (shown above is original IRIS with different colored wires). The black wire remains not connected.



Connect the servo to the header pins on the circuit, being sure to connect it in the correct orientation-- the brown wire is ground and should align with the voltage regulator center pin (ground), the red wire is positive and should align with the voltage regulator output, and the yellow wire should align with the servo signal coming from IRIS+.



We'll configure IRIS+ to activate the servo as if it were one of the servos in a gimbal.



At first we triggered the servo with channel 6, the knob on the IRIS+ controller as shown above. On IRIS classic, this "just worked" when the servo control wire was plugged into the orange gimbal control wire because the knob is already configured to adjust the angle of a camera this way. This approach didn't work out of the box on IRIS+, however, and requires adjusting the parameters to link up the control channel to the gimbal control.

Another option is to link the imaginary gimbal's 'camera trigger' servo to channel 7 (the switch to the left of the knob on the controller). The system expects this servo to press for a configurable length of time, then release.

Either trigger mode might be preferable, depending on whether you want long stretches or short bursts of spray! Below is a parameter file you can load into your bot for the camera shutter option:

[paramter_iris_shutter.param.zip](#)

<https://adafru.it/jCk>

Click the "LOAD" button on the right in your mission planning software, and select your recently downloaded param file to load.


```
351 RC6_TRIM,992
352 RC7_DZ,0
353 RC7_FUNCTION,0
354 RC7_MAX,2017
355 RC7_MIN,991
356 RC7_REV,1
357 RC7_TRIM,992
358 RC8_DZ,0
```

Adjust your CAM_* parameters for spray duration, etc.:

```
50 BRD_SER1_RTSCCTS,2
51 BRD_SER2_RTSCCTS,2
52 CAM_DURATION,20
53 CAM_SERVO_OFF,1000
54 CAM_SERVO_ON,2000
55 CAM_TRIGG_DIST,0.00000000
56 CAM_TRIGG_TYPE,0
57 CH10_OPT,0
58 CH11_OPT,0
```

Settings for indoor flight

We followed the [ArduPilot tutorial on indoor flying \(https://adafru.it/fkU\)](https://adafru.it/fkU), so be sure to read through it thoroughly. We've created a settings profile you can load onto your copter but you should also be familiar with these basic ideas:

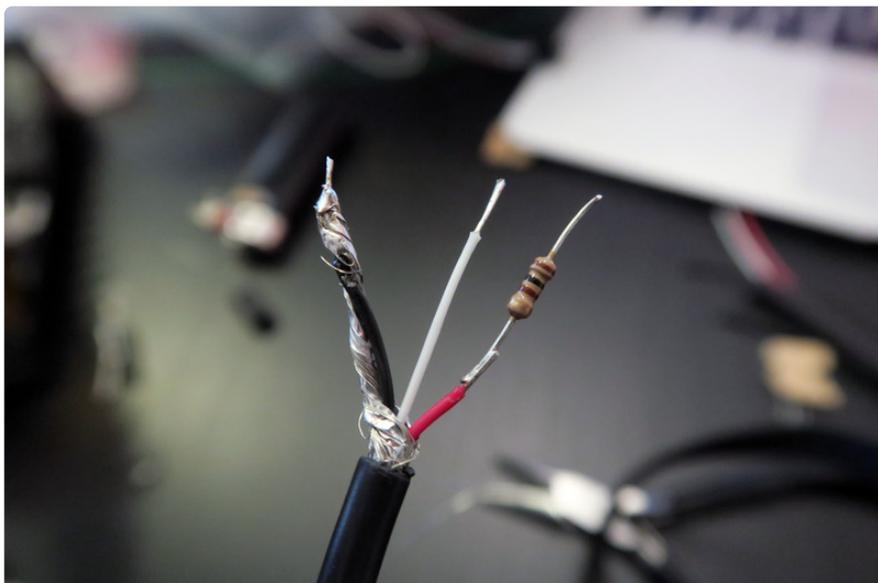
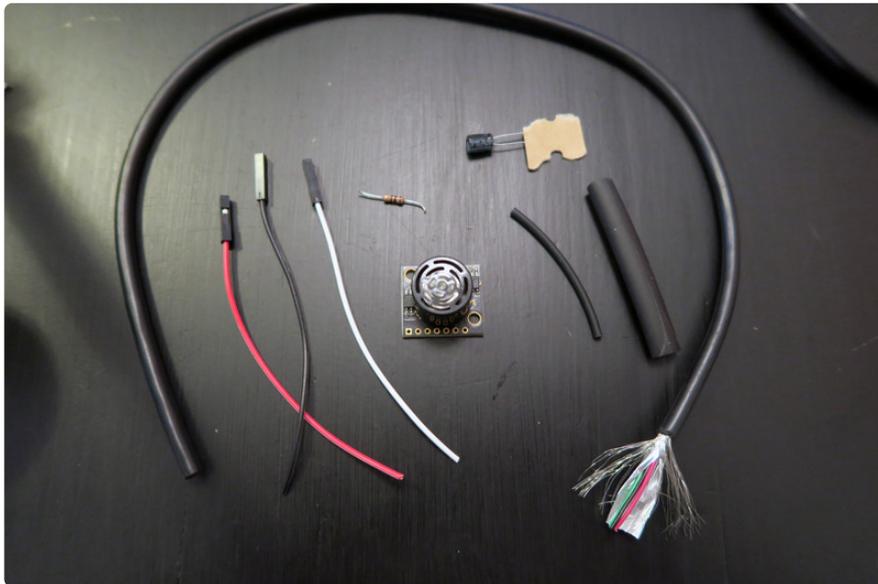
- many auto modes use GPS, and GPS doesn't work inside. GPS must be disabled and no auto modes can't be used.
- Adding a downward-facing sonar sensor can help make an indoor altitude hold mode

Sonar sensor cables

The Pixhawk can receive info from two analog sonar sensors, which you can use to aid in an indoor "altitude hold" mode or for checking distance from a wall. Check out [the ArduPilot page on wiring and configuring sonar sensors \(https://adafru.it/fkQ\)](https://adafru.it/fkQ) to the Pixhawk and Mission Planner.

These sensors are sensitive to interference caused by the quadcopter's motors. To cut down on interference, a special cable must be made! In addition to one of the MaxBotix sensors suggested in the ArduPilot guide, you will need:

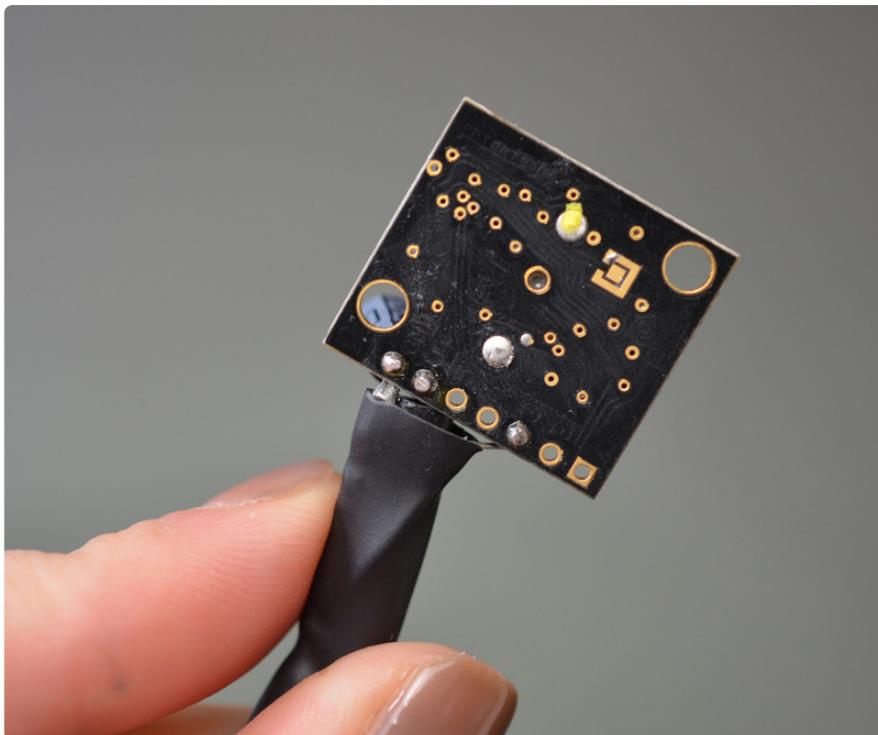
- shielded cable with at least three wires (recycled USB cable works)
- [100uF capacitor](http://adafru.it/2193) (<http://adafru.it/2193>)
- 10 ohm resistor
- [heat shrink tubing](http://adafru.it/1649) (<http://adafru.it/1649>)
- [5-position connector](https://adafru.it/fkS) (<https://adafru.it/fkS>) for Pixhawk



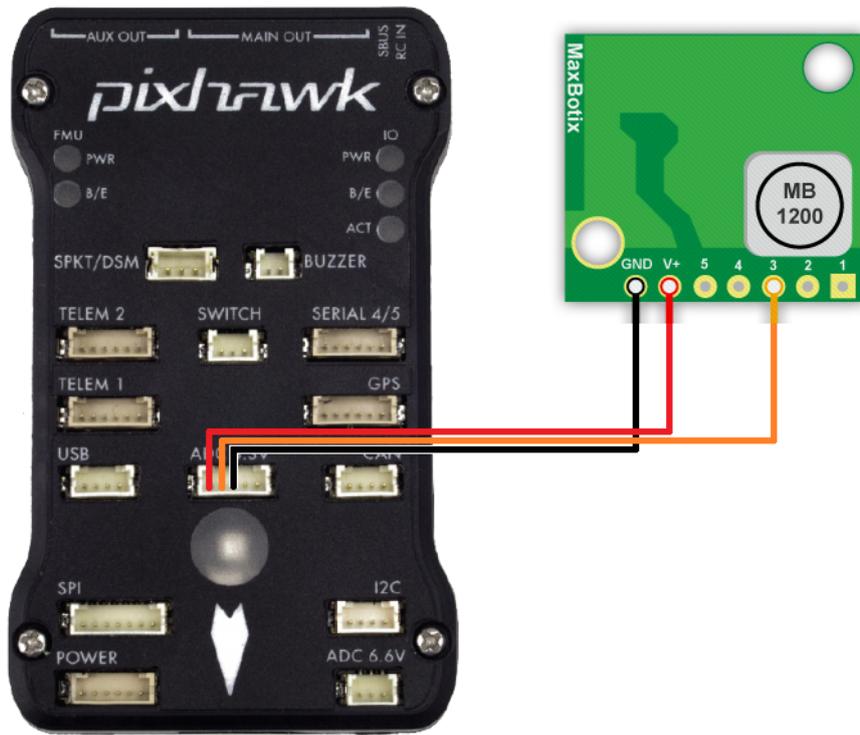
Strip and tin the three wires, joining the shielding to the black ground wire and attaching the resistor at the end of the red power lead. Slide a small piece of heat shrink over the resistor and its wire connection.

Insert black to ground, the power-line resistor to V+, and the white signal wire to the sensor's analog out.

Before soldering, also insert the capacitor between power and ground, making sure the long leg goes to ground.



Solder the other ends of the wires to correspond with the pins on the 5-connector plug on the Pixhawk and of the cable according to this diagram from the [ardupilot site \(https://adafru.it/fkQ\)](https://adafru.it/fkQ):



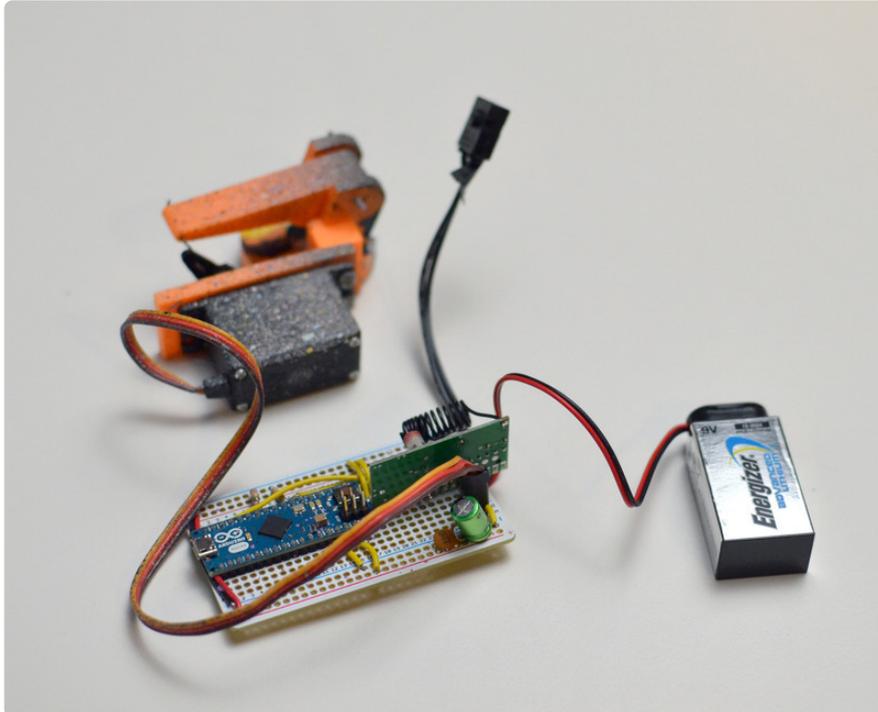
Although we did get sonar signals coming into the Pixhawk, we did not get so far as to use them to successfully control the autopilot for IRIS+ for this project-- perhaps you can pick up where we left off!

Fly it!



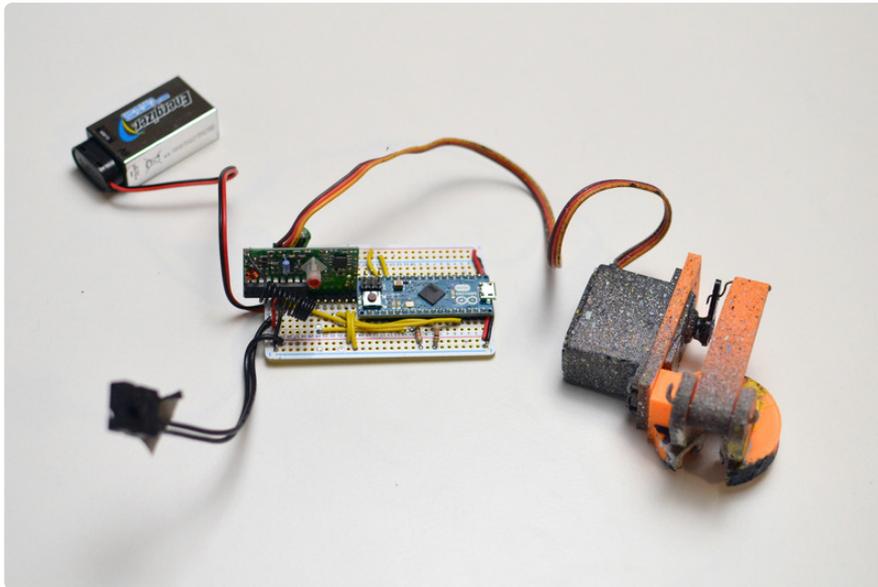
Easier said than done! This rig is very challenging to fly, especially indoors. Make sure you have plenty of practice before affixing the spray can, which changes the center of gravity of the bot.

Prior attempt at standalone sprayer



If your drone isn't so forthcoming about providing power connectors or gimbal control documentation, you may wish to create a standalone parasitic device to trigger your spray can while it's mounted on the quadcopter. Pictured above is [the "icarus one" version of the device used in KATSU's drone paintings \(https://adafru.it/frj\)](https://adafru.it/frj), which uses a 3D printed servo can mount based off [sliptonic's silly-string trigger on Thingiverse \(https://adafru.it/fkM\)](https://adafru.it/fkM).

The store-bought injection-molded plastic spray mounts are really much better suited for this project than 3D printing, but for some reason it just hadn't occurred to me to look for such a product at the time.



The electronics consist of the following:

- [standard servo](http://adafru.it/155) (<http://adafru.it/155>)
- [Arduino Micro](http://adafru.it/1086) (<http://adafru.it/1086>)
- [on/off switch](http://adafru.it/1092) (<http://adafru.it/1092>)
- [9v battery clip](http://adafru.it/80) (<http://adafru.it/80>)
- [315MHz RF receiver](http://adafru.it/1096) (<http://adafru.it/1096>) and [remote](http://adafru.it/1095) (<http://adafru.it/1095>)
- [half-sized Perma-Proto](http://adafru.it/1609) (<http://adafru.it/1609>)

Sure, the range on the transmitter/receiver pair isn't close to the range on the quadcopter's radios, but for painting you want to be close to the action anyway so this was considered an acceptable limitation.

The code simply took two digital inputs from the RF receiver and used their states to turn on the servo and press the lever as well as release it:

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo
                // a maximum of eight servo objects can be created
int triggerpin = 5;
int offpin = 2;

int buttonState; // the current reading from the input pin
int lastButtonState = LOW; // the previous reading from the input pin
int button2State; // the current reading from the input pin
int lastButton2State = LOW; // the previous reading from the input pin

void setup()
{
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
  pinMode(triggerpin, INPUT);
  pinMode(offpin, INPUT);
}
```

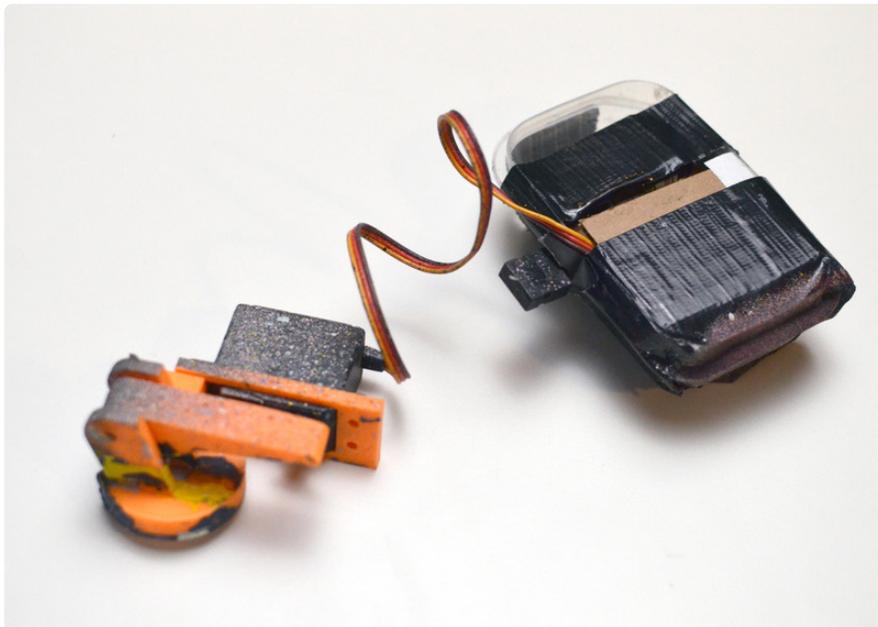
```

}

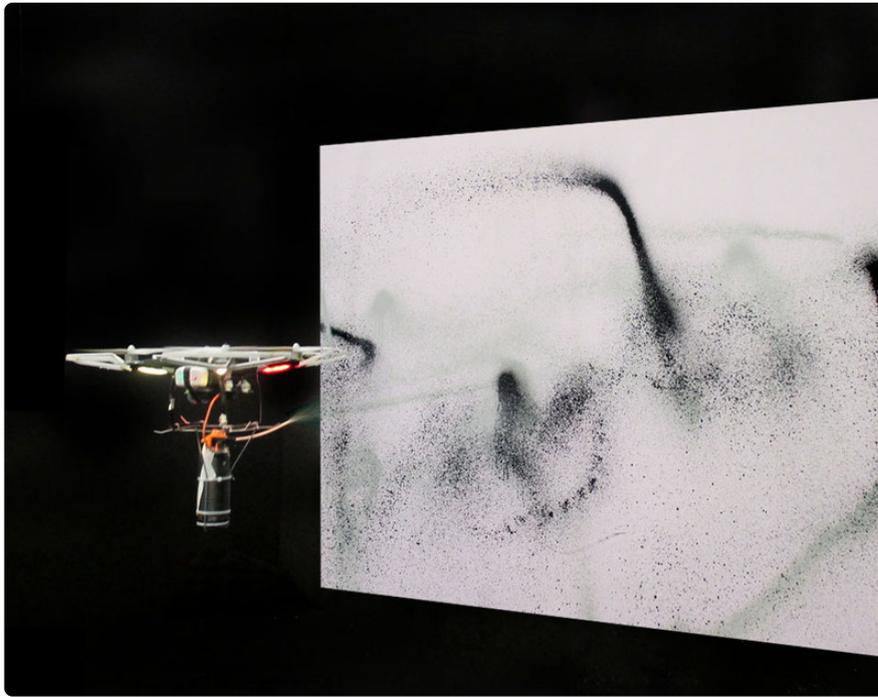
void loop()
{
  buttonState = digitalRead(triggerpin);
  button2State = digitalRead(offpin);

  // compare the buttonState to its previous state
  if (buttonState != lastButtonState || button2State != lastButton2State) {
    // if the state has changed, increment the counter
    if (buttonState == HIGH) {
      // if the current state is HIGH then the button
      // went from off to on:
      //buttonPushCounter++;
      myservo.write(90);
      delay(15);
      //Serial.println("on");
      //Serial.print("number of button pushes: ");
      //Serial.println(buttonPushCounter);
    }
    else if (button2State == HIGH) {
      // if the current state is LOW then the button
      // went from on to off:
      //Serial.println("off");
      myservo.write(0);
      delay(15);
    }
  }
  // save the current state as the last state,
  //for next time through the loop
  lastButtonState = buttonState;
}
}

```



It is this standalone device that was attached to a DJI Phantom quadcopter to create [KATSU's drone paintings \(https://adafru.it/fkP\)](https://adafru.it/fkP). A [circuit diagram is available at the icarus one site \(https://adafru.it/frj\)](https://adafru.it/frj).



The known issues with this circuit are mainly related to range and responsiveness once the servo is engaged, depressing the spray can.

The current draw on the circuit seems to decrease the sensitivity of the RF receiver, resulting in a spraying can that can't be turned off remotely, so it sprays until it runs out of paint. This was also considered an acceptable limitation for experimentation under the circumstances.