How to Control a DIY Mini Drone with an Arduino™

https://www.sciencebuddies.org/science-fair-projects/project-ideas/Robotics_p038/robotics/drone-programming-arduino (https://www.sciencebuddies.org/science-fair-projects/project-ideas/Robotics_p038/robotics/drone-programming-arduino)

Procedure PDF Date: 2023-10-25

Experimental Procedure

Note: if you purchased the Mini Drone Kit from Home Science Tools, some of these steps (like stripping the wires and drilling holes in a wood base) are already done for you.

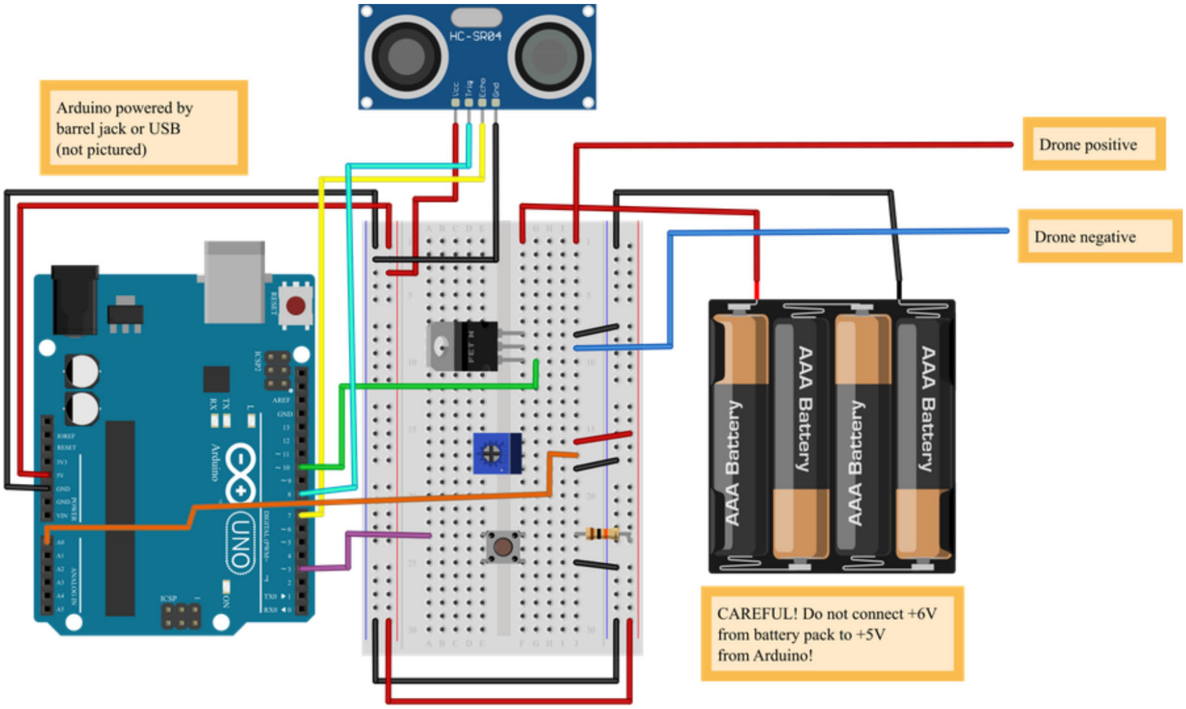
Build Your Drone

1. Watch the following video and read the directions in the Build a Simple Drone and Test How Much It Can Lift (http://www.sciencebuddies.org/science-fair-projects/project-ideas/Aero_p058/aerodynamics-hydrodynamics/diy-drone-how-much-weight#procedure) project to build your drone. You should use two guide poles for your drone instead of one (as shown in the video). This will prevent the wires from getting twisted around the pole.

https://www.youtube.com/watch?v=Q-BluEJBHLw (https://www.youtube.com/watch?v=Q-BluEJBHLw)

Connect the Arduino

1. Orient the breadboard so it is facing you with the writing upright. Then connect all of the following, as shown in Figure 5 (click for a bigger version of the diagram (https://www.sciencebuddies.org/cdn/Files/17050/10/drone-arduino-altitude-breadboard-circuit-1.png)). If you know how to read a circuit schematic (https://www.sciencebuddies.org/cdn/Files/17051/12/arduino-drone-control-circuit-diagram-1.png), you can use that instead (Figure 6).
    a. Arduino 5V pin to breadboard positive bus.
    b. Arduino GND pin to breadboard ground bus.
    c. Breadboard left positive bus to breadboard right positive bus.
    d. Breadboard left ground bus to breadboard right ground bus.
    e. Ultrasonic sensor GND pin to breadboard ground bus (use male-female jumper wires to connect to the sensor pins).
    f. Ultrasonic sensor 5V pin to breadboard positive bus.
    g. Ultrasonic sensor Echo pin to Arduino digital pin 7.
    h. Ultrasonic sensor Trig pin to Arduino digital pin 8.
    i. MOSFET in holes F8, F9, and F10, with the large metal tab facing to the left.
    j. Hole J8 to ground bus.
    k. Hole G10 to Arduino digital pin 10.
    l. Potentiometer in holes F16, F17, F18.
    m. Hole J16 to positive bus.
    n. Hole J17 to Arduino analog input A0.
    o. Hole J18 to ground bus.
    p. Push button straddling the middle gap of the breadboard, with pins in holes E23, E25, F23, and F25.
    q. Hole A23 to Arduino digital pin 3.
    r. 10 kΩ resistor from hole J23 to positive bus.
    s. Hole J25 to ground bus.
    t. Drone positive wire to J1. The drone's wires are too flexible to push into the breadboard on their own. Twist or crimp them to a short piece of jumper wire to make the connection.
    u. Drone negative wire to J9.
    v. 4xAA battery pack negative (black) wire to ground bus.
    w. 4xAA battery pack positive (red) wire to F1. **Important**: Do not connect the battery pack's positive wire to the breadboard's positive bus! This will create a short circuit between the Arduino's 5 V power and the battery pack's 6 V power.



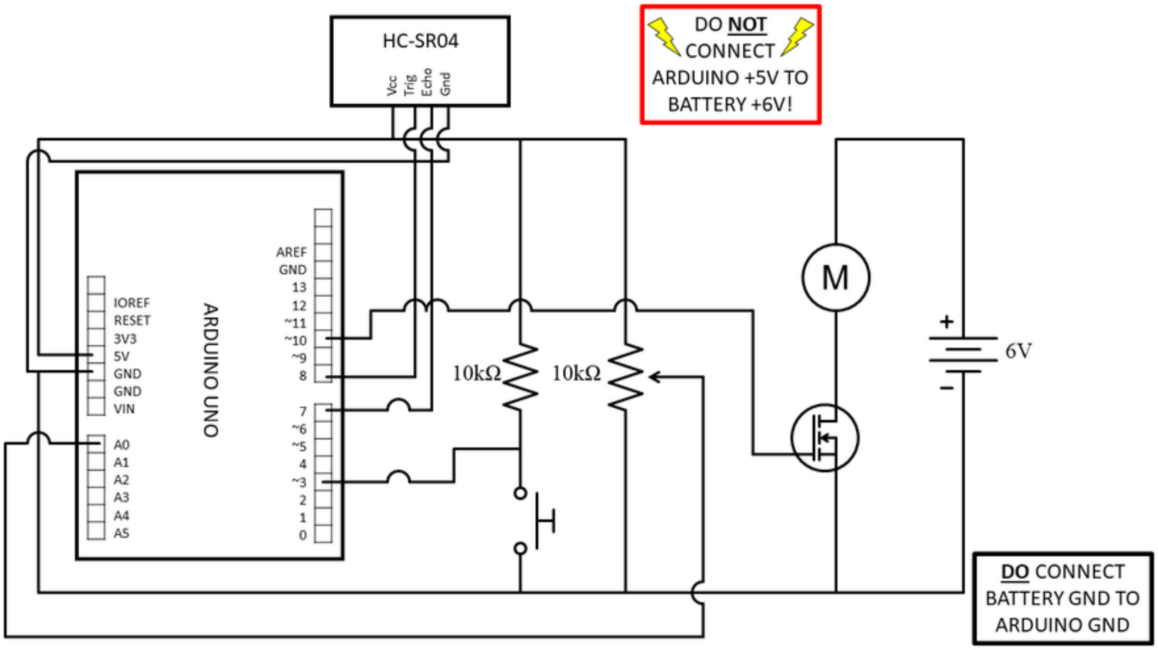
Arduino powered by barrel jack or USB (not pictured)

Drone positive

Drone negative

CAREFUL! Do not connect +6V from battery pack to +5V from Arduino!

Figure 5. Breadboard diagram for Arduino drone control circuit.



DO **NOT** CONNECT ARDUINO +5V TO BATTERY +6V!

DO CONNECT BATTERY GND TO ARDUINO GND

Figure 6. Circuit schematic for Arduino drone control circuit.

2. Upload the code to your Arduino.
   a. Download drone_altitude_control_code.ino (http://www.sciencebuddies.org/cdn/Files/17046/11/drone_altitude_control_code.ino) to your computer.
   b. Temporarily disconnect one of the drone's wires from the breadboard. This will prevent the drone from accidentally taking off when you power up your Arduino (this can happen depending on the last program that ran on the Arduino).
   c. Connect the Arduino to your computer with the USB cable.
   d. Open drone_altitude_control_code.ino in the Arduino IDE.
   e. Press the "Upload" button to send the code to your Arduino.
3. Test your drone.
   a. Make sure the drone is threaded onto the guide poles.
   b. Place the ultrasonic sensor directly under the drone, facing up (Figure 7).
   c. Re-connect the drone's wire to the breadboard.
   d. Press the push button once to make the drone lift off.

e. Adjust the potentiometer to change the drone's height.
f. Gently tap the drone up or down with your finger or a pencil (watch out for the propellers!). What happens?
g. Press the push button again to make the drone land. You can re-press the button to make it lift off again.

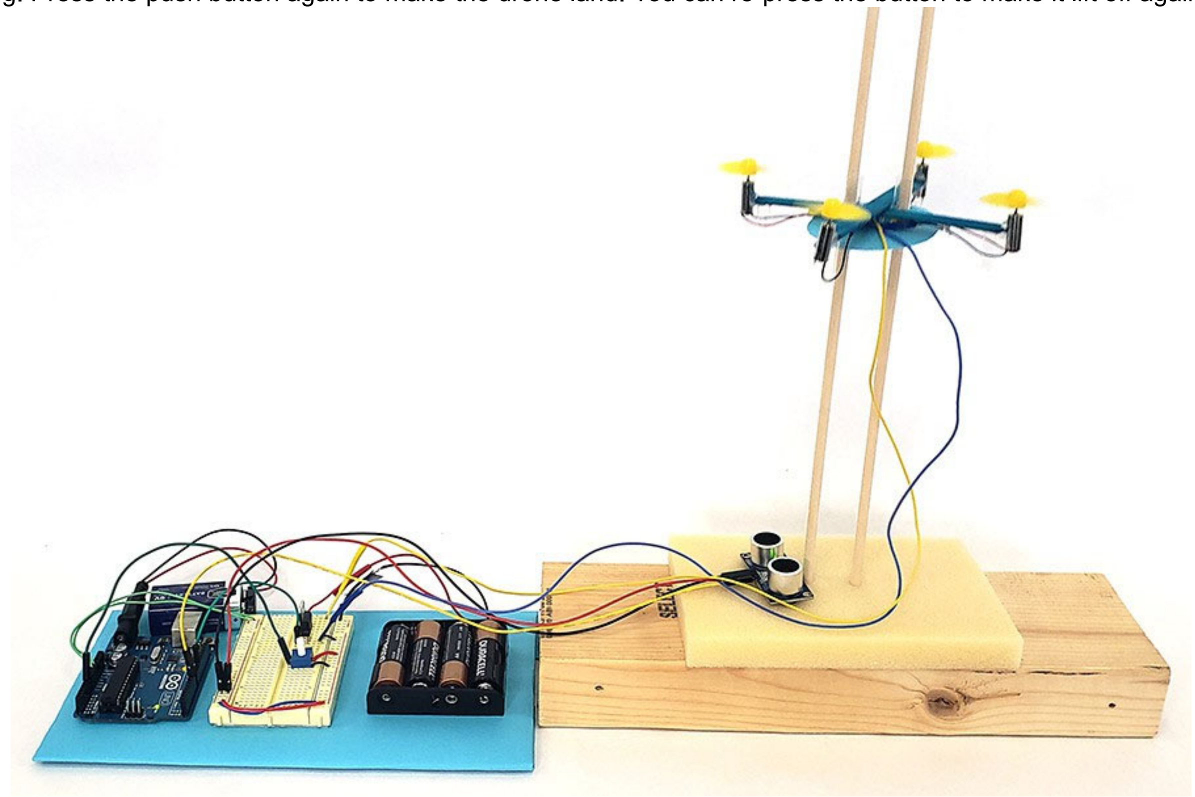


Figure 7. Experimental setup, with the drone on guide poles and ultrasonic sensor placed under the drone facing upward.

Using Your Drone for a Science or Engineering Project

To use your drone for a science or engineering fair project, it is not sufficient to simply present what you have built so far based on the Science Buddies instructions and example code. You will need to conduct your own experiment (for a science project) or design/program something new (for an engineering project). Before you begin either type of project, you should carefully read through the comments in the example code and make sure you understand how it works. Here are a few suggestions for a project:

- How does the drone respond to disturbances of different amounts? For example, try holding the drone at fixed distances from the target height (1 cm, 2 cm, 3 cm...) and then releasing it.
- The example code uses a proportional controller. It increases (or decreases) the motor speed by an amount *proportional* to the difference between the target height and actual height. What happens if you change the code to use basic on/off control instead (i.e. either turning the motors on full speed or turning them completely off, nothing in between)? How does the drone's behavior compare to the proportional controller?
- What is the effect of changing the gain (the variable *K* in the code) of the proportional controller?
- Can you program a proportional-integral-derivative (PID) controller (http://www.sciencebuddies.org/science-fair-projects/project-ideas/Robotics_p036/robotics/drone-pid-controller) instead of a basic proportional controller?
- Can you program the drone to follow a pre-programmed path? For example, continuously flying up and down in an oscillating (sinusoidal) pattern? What about quickly flying to different fixed heights and hovering for a few seconds before moving to the next height? How can you get the drone to closely follow the path with minimal error?
- Can you design and build a handheld controller for your drone? What if you use an analog joystick (http://www.avantlink.com/click.php?tt=cl&mi=10609&pw=182414&ctc=drone-programming-arduino&url=https%3a%2f%2fwww.jameco.com%2fz%2f27800-Parallax-2-Axis-Joystick_2082855.html), just like the type found in video game controllers and many commercial drone controllers, instead of a potentiometer?

In any of the above projects, there are several ways you can collect data about your drone's movement. The easiest way is to copy data from the Arduino serial monitor into a spreadsheet program so you can graph the data*. This will allow you to plot the drone's position vs. time as you change variables or use different control methods. However, you can also film your drone and analyze its motion using a program like Tracker (https://physlets.org/tracker/). You could even compare the two different methods and turn that into a project of its own!

* Copying data from the Arduino serial monitor can be a bit tricky due to an apparent bug. If you press CTRL+A then CTRL+C to select and copy all the data in the serial monitor window, it does not work. Instead, when you are done collecting data, unplug the USB cable to disconnect the Arduino from the computer and stop printing to the serial monitor. Then, select all of the data in the serial monitor *except for the first line*. Do this by clicking at the beginning of the second line, scrolling all the way to the bottom, holding SHIFT, then clicking at the end of the last line. Then press CTRL+C to copy the data, and you can paste it into another program like Notepad or Excel®. You may need to use the Text to Columns feature in Excel (https://support.microsoft.com/en-us/office/split-text-into-different-columns-with-the-convert-text-to-columns-wizard-30b14928-5550-41f5-97ca-7a3e9c363ed7), or the equivalent in another spreadsheet program, to get the data into separate columns.

Frequently Asked Questions (FAQ)

FAQ for this Project Idea available online at https://www.sciencebuddies.org/science-fair-projects/project-ideas/Robotics_p038/robotics/drone-programming-arduino#help.