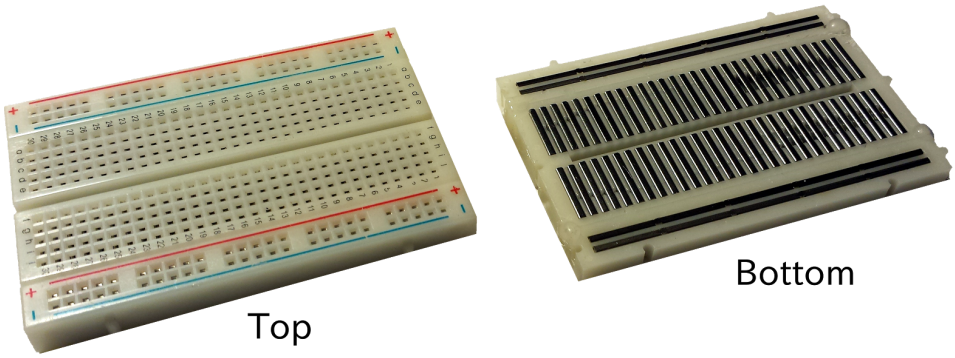
Program-Your-Own Helicopter Kit by Stage One Education

You're about to build a circuit that will give your computer control over your S107G helicopter! These instructions provide a step-by-step workshop for learning about circuits. If you're already familiar with circuits and all that, just skip down to the **IR LED Array** section to get right into flying.

Breakout, Beginnings & Breadboards

---

1. How breadboards work:



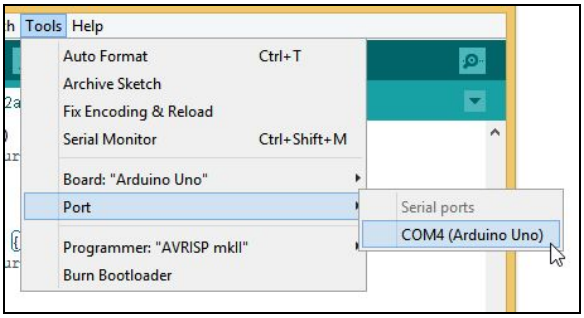
The upsidedown breadboard on the right shows the connection pattern of a breadboard. This pattern is pretty standard - it makes connecting wires together easy and convenient!

2. Your kit (14 items):
    a. Arduino microprocessor
    b. USB cable
    c. Helicopter
    d. Yellow Helicopter charging cable
    e. Breadboard
    f. IR LED array
    g. Power Transistor
    h. Two LED lights and two resistors
    i. AA Battery holder
    j. 9V battery snap
    k. Seven Batteries
    l. Small button
    m. Fan
    n. A few wires

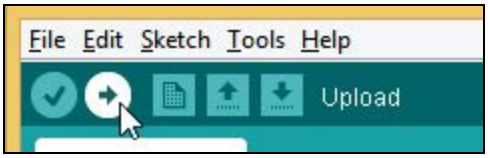
3. **Be sure to start charging your helicopter!**
    a. You need to turn the helicopter off for it to charge.
    b. If the light on the USB connector is **off**, it is charging.

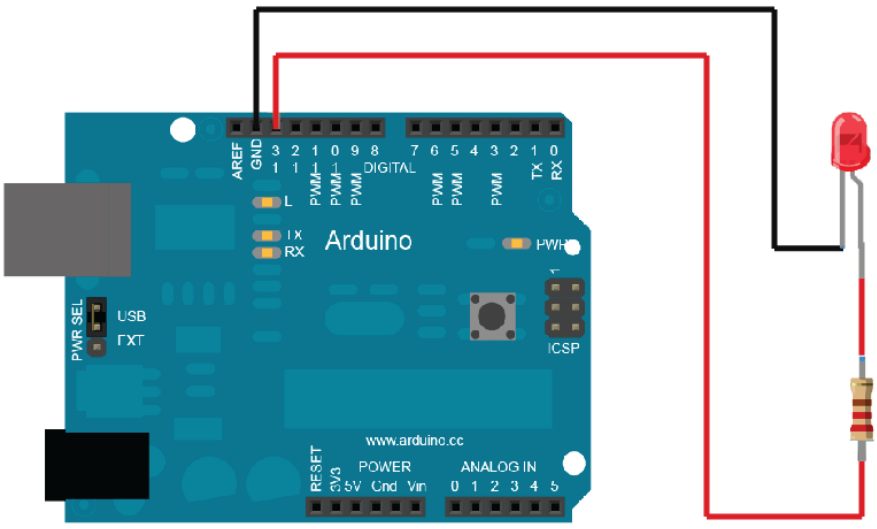
Blinking LED

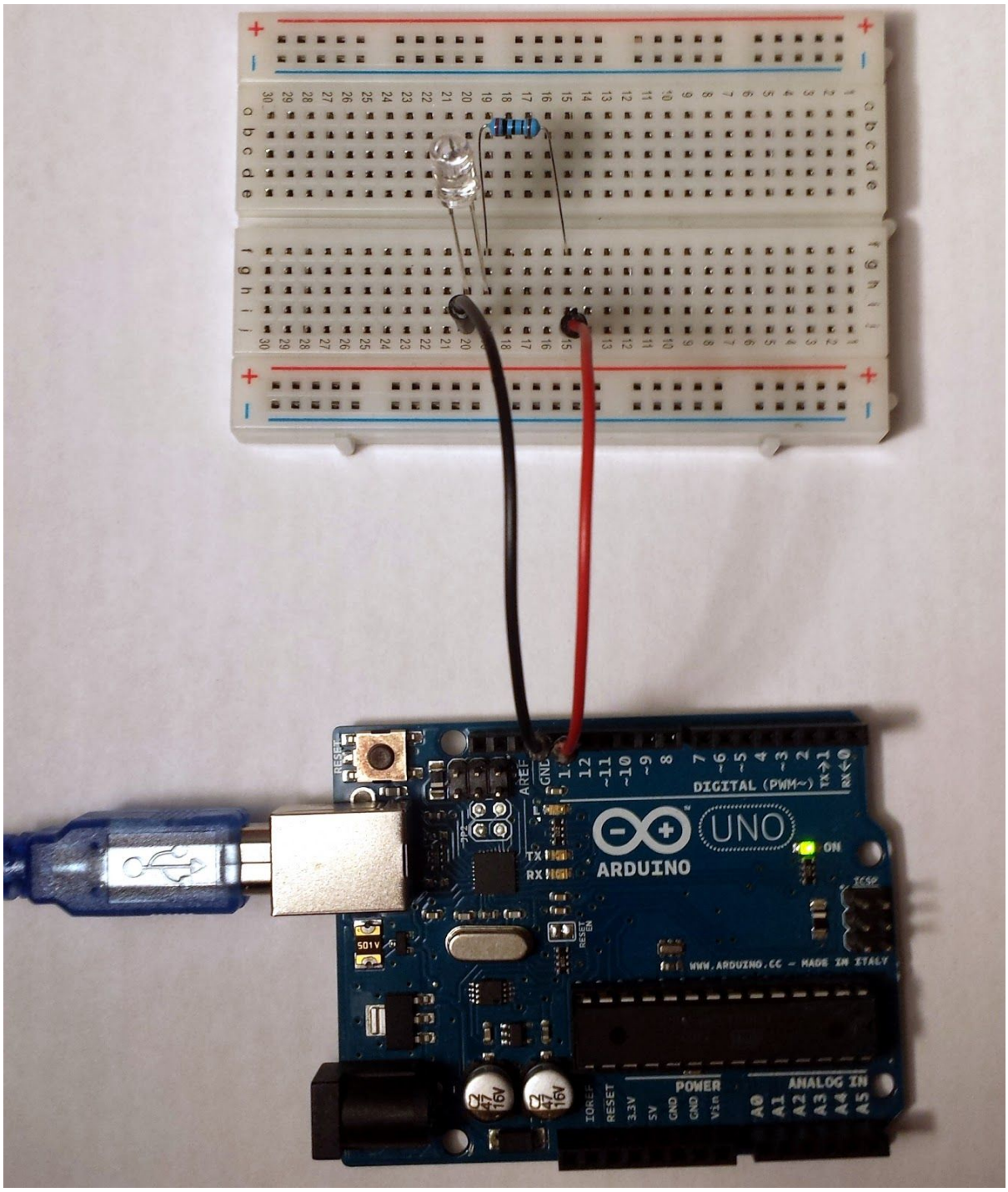
1. Install the Arduino software from:
    - https://www.arduino.cc/en/Main/Software (under *Download the Arduino IDE*)
    - If it asks, say 'yes' when the installer asks about installing the drivers.
2. Plug the Arduino into the computer USB port
3. Open Arduino software
4. From the Arduino menu, select the serial port for the Arduino Uno:
    - Windows: **Tools > Port > COM1 (or 2,3,…)**
    - Mac: **Tools > Port > /dev/cu.usbmodem<xxxx>**
    - Not working? Check:
        - Arduino plugged in (try disconnecting and reconnecting)
        - Try a different COM port



3. Set up the circuit on the breadboard (figure below).
4. **How do you know which side of the LED is plus (+) or minus (-) ?**
    a. The longer lead is (+), connected to pin 13 in the diagram below.
5. Load the default blink code:
    a. File > Examples > 01.Basics > Blink
6. Check serial port. Upload. The button looks like this:
7. (optional) Change the LED's blinking period

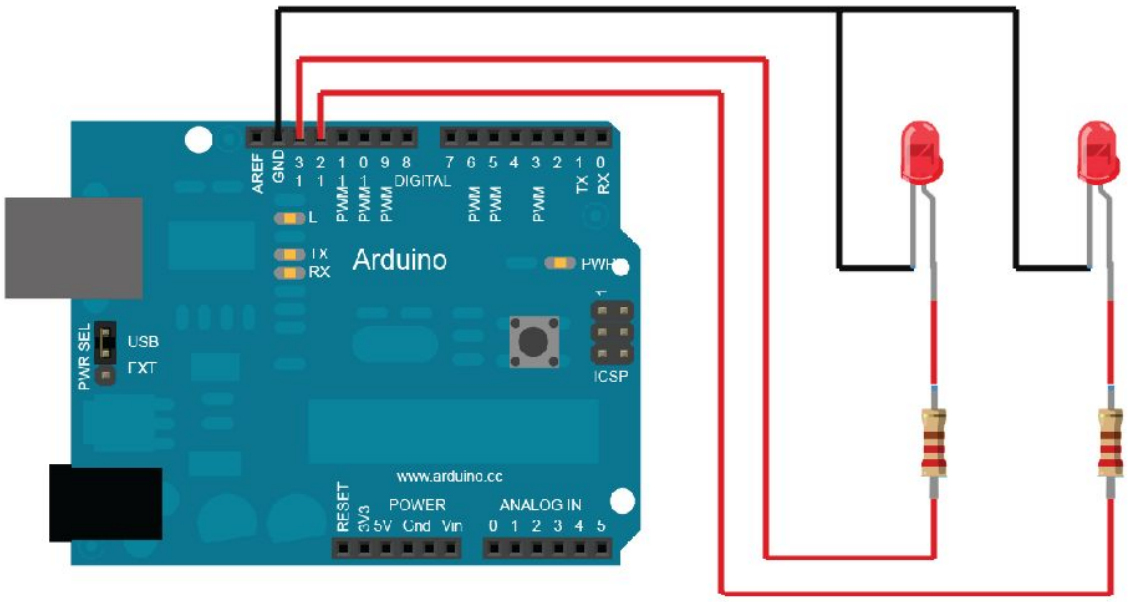





UNO
ARDUINO
DIGITAL (PWM~)
POWER
ANALOG IN
WWW.ARDUINO.CC – MADE IN ITALY

Second Blinking LED

1. Set up a second LED on the circuit board
2. Program the software to setup pin 12 in `OUTPUT` mode
3. Make the software blink the second light ON while the first one is OFF, and vice-versa



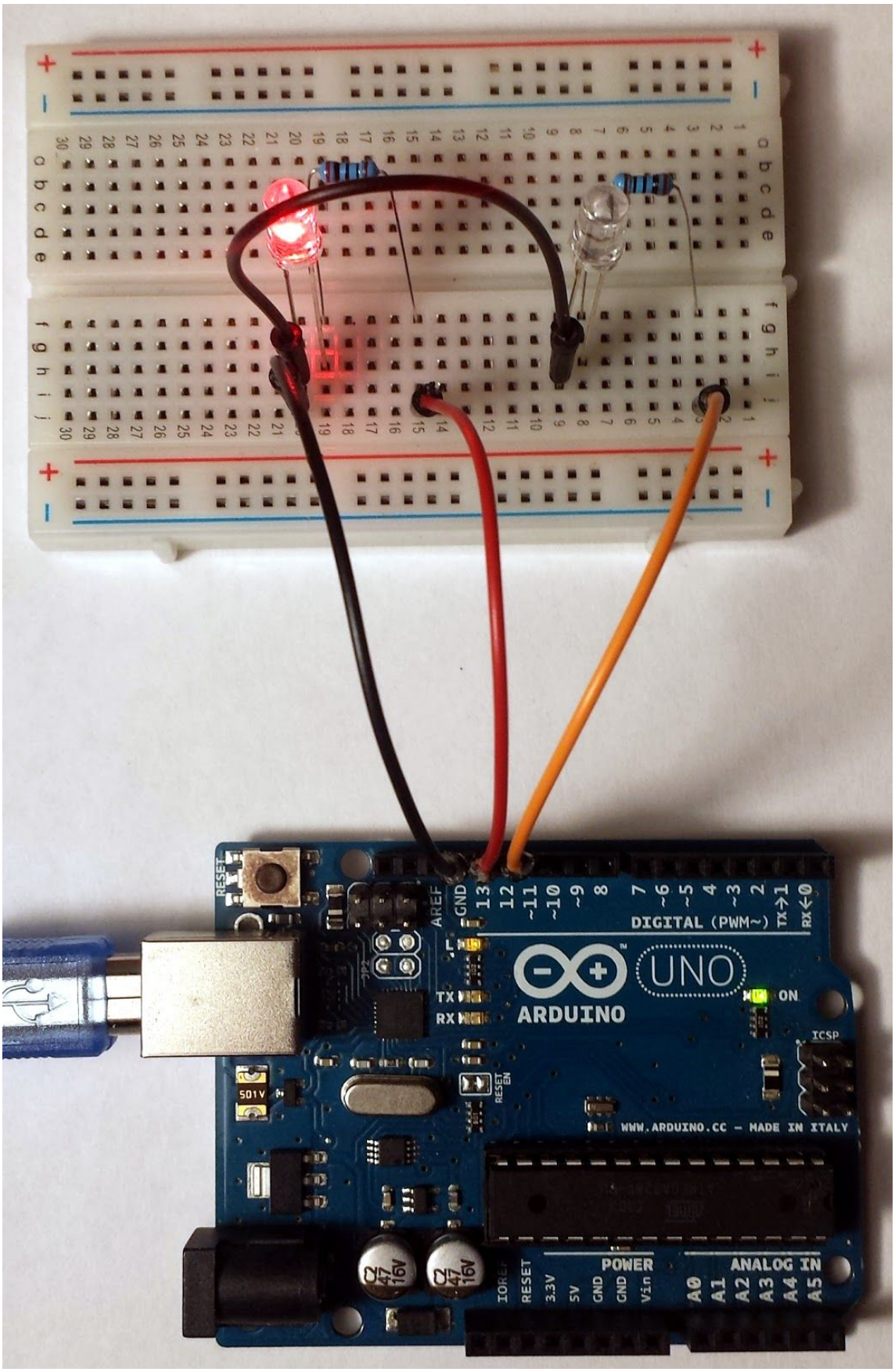
Changing the code (some hints):

```
// this code is ignored!
```

- The computer **ignores** anything with a "`//`" in front of it. Those lines often includes helpful hints to the programmer (you).
- Other lines of code (like `pinMode`) are colored. Capitalization matters.

```
pinMode(13, OUTPUT);
```

- The brackets "`{`" and "`}`" organize the code into chunks.
- The code in `void setup() {...code...}` is run **once** when you hit the upload button.
- The code in `void loop() {...code...}` is run again and again endlessly.
- All lines of code need to have a semicolon "`;`" at the end
- You can do this exercise by copy+pasting lines of code, and making small changes
- You need the two LED's to have <u>different</u> positive pins but the <u>same</u> ground (negative)


RESET
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
UNO
ARDUINO
ON
TX
RX
ICSP
WWW.ARDUINO.CC – MADE IN ITALY
POWER
ANALOG IN
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

Flying the Helicopter with the Remote

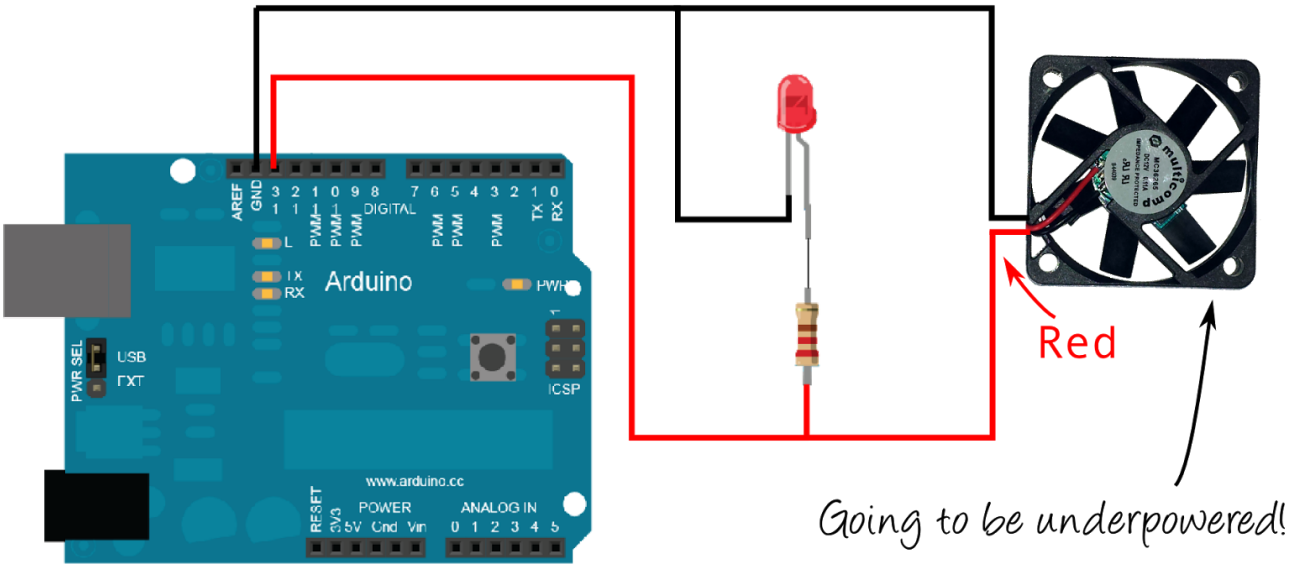
Time to fly the helicopters with the remote control!
We'll take a quick break from the circuit, then get back to it.

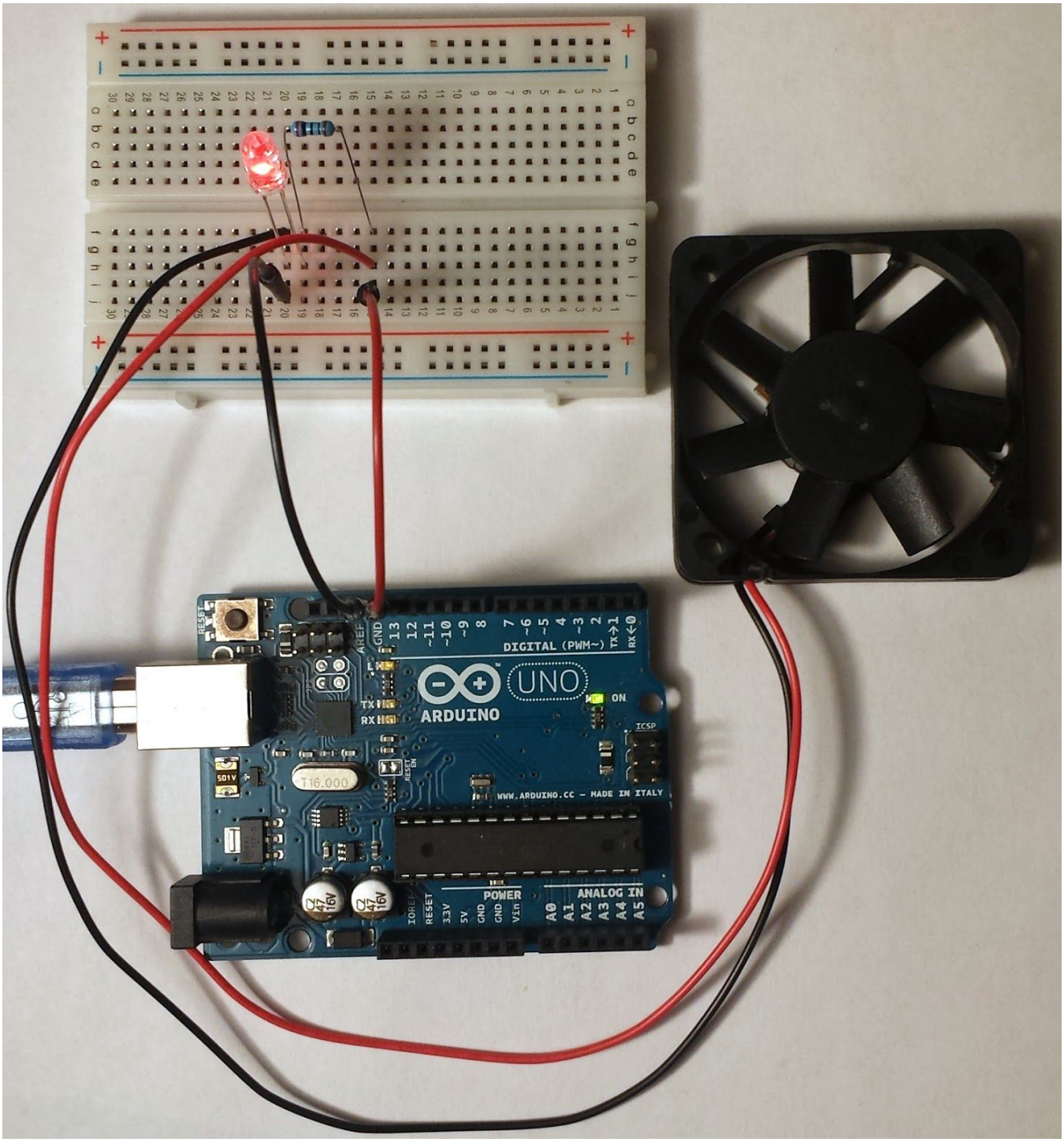
- **You only have about 5 minutes of battery life.**
- Helicopter must be level and still when turned on to calibrate its gyroscope.
- You may need to do some turning on and off.

Fan & Transistor

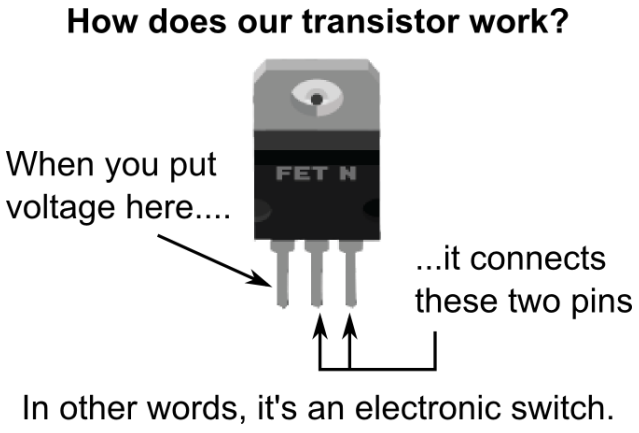
- **Recharge your helicopter.**

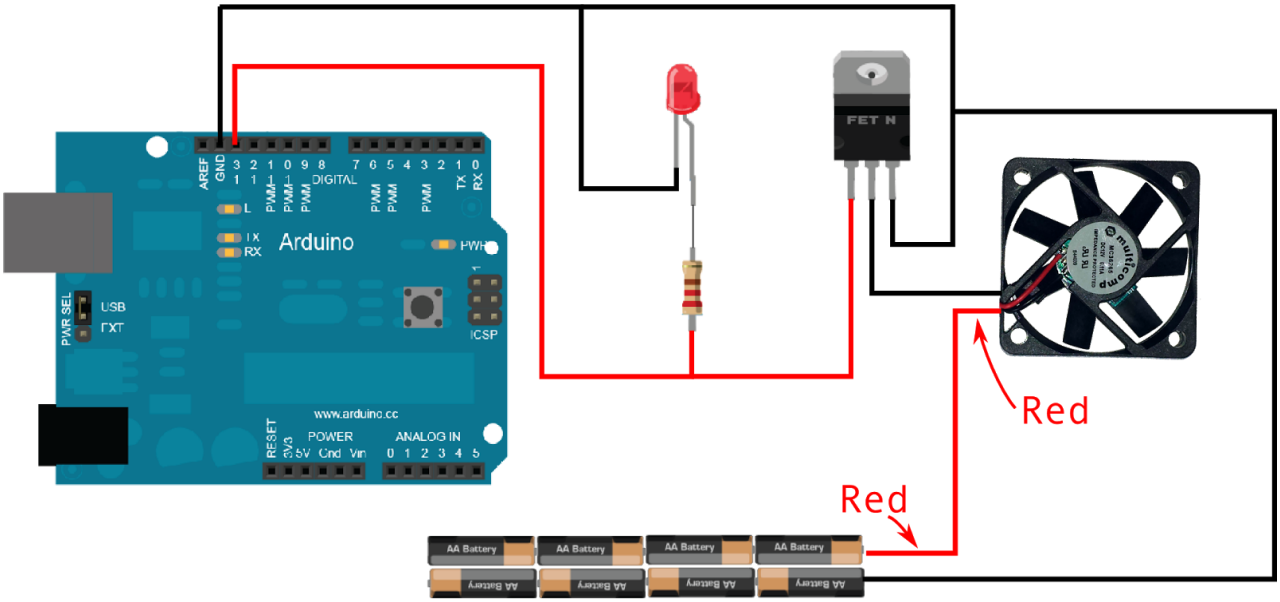
1. Hook up the fan to the Arduino, as shown below:




DIGITAL (PWM~)
UNO
ARDUINO
ON
TX
RX
WWW.ARDUINO.CC – MADE IN ITALY
POWER
ANALOG IN

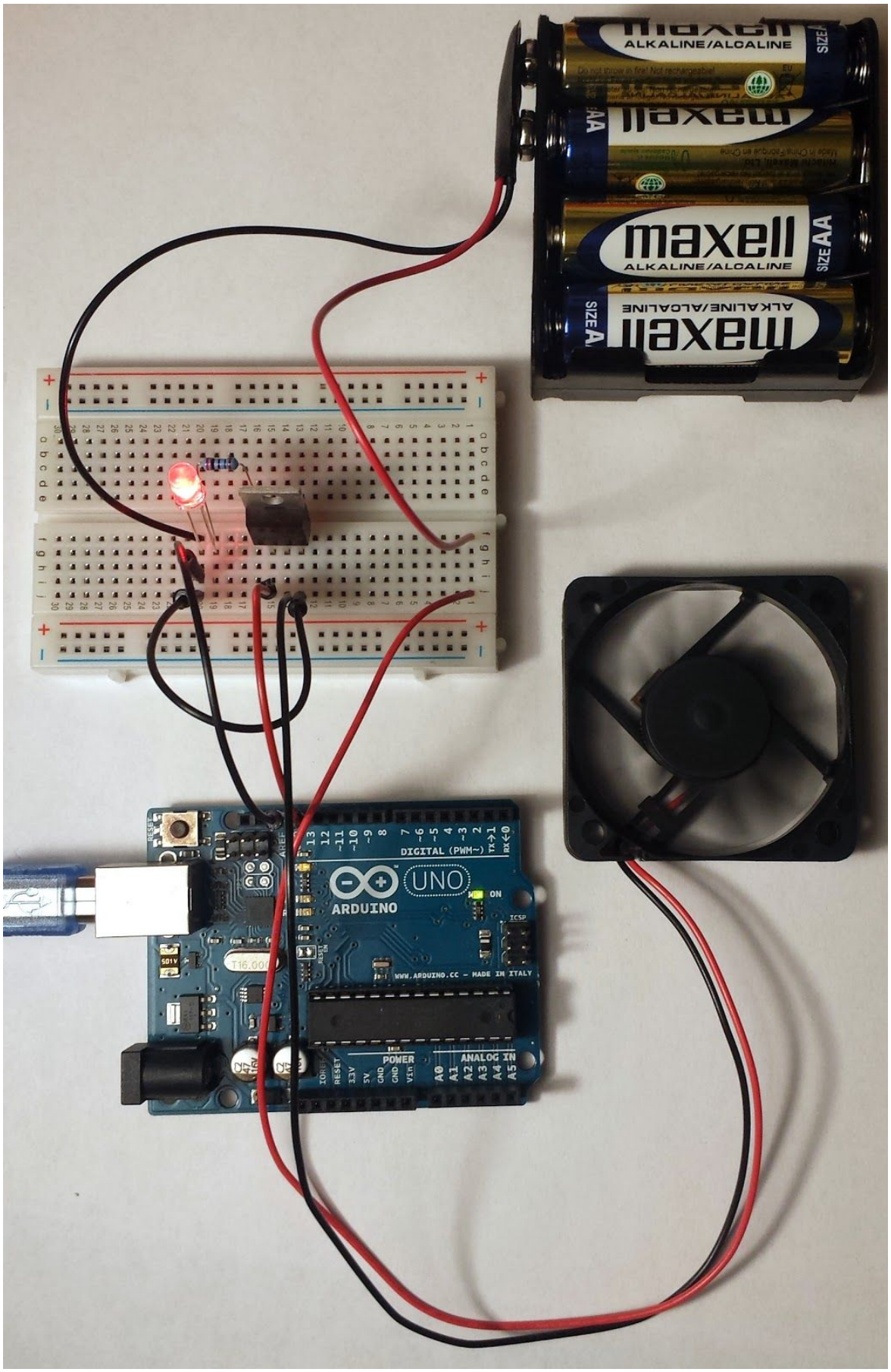
2. The Arduino doesn't have enough power to make the fan go fast. Let's add more power…
3. Hook up the power transistor to the fan in the circuit below.
4. **Be sure to check the circuit. Don't connect the battery positive to pin #13!**
5. Modify the code to make the fan completely stop and completely start again.
6. Modify the code to make the fan run continuously at 50% power. (You might need to look up "pulse width modulation" or "PWM")
7. Run the fan at 75% power.





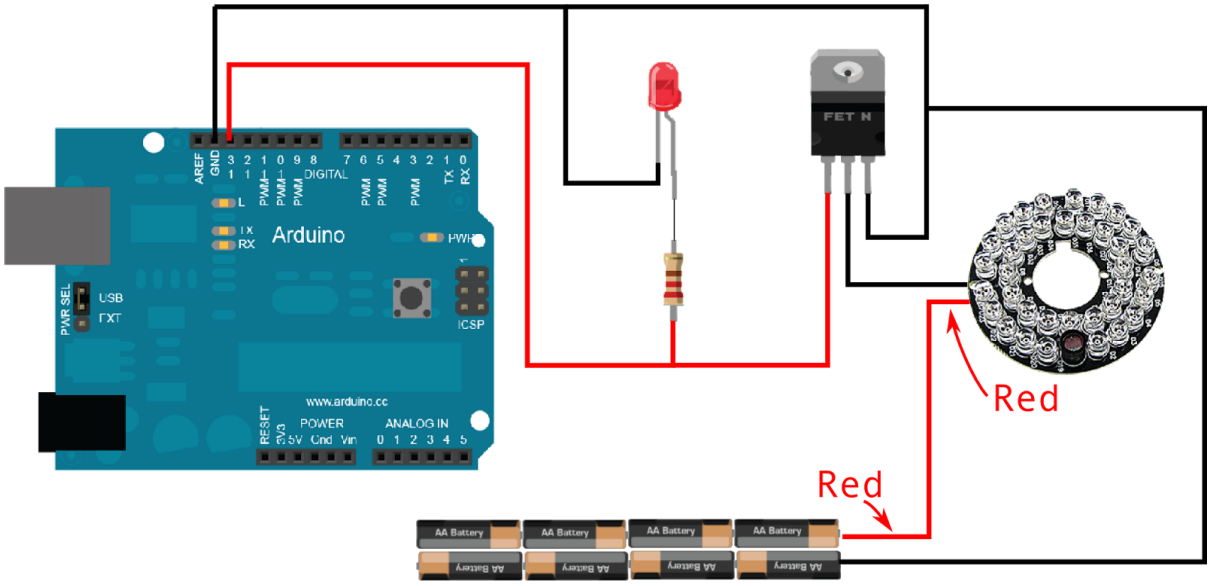
Common Problems:

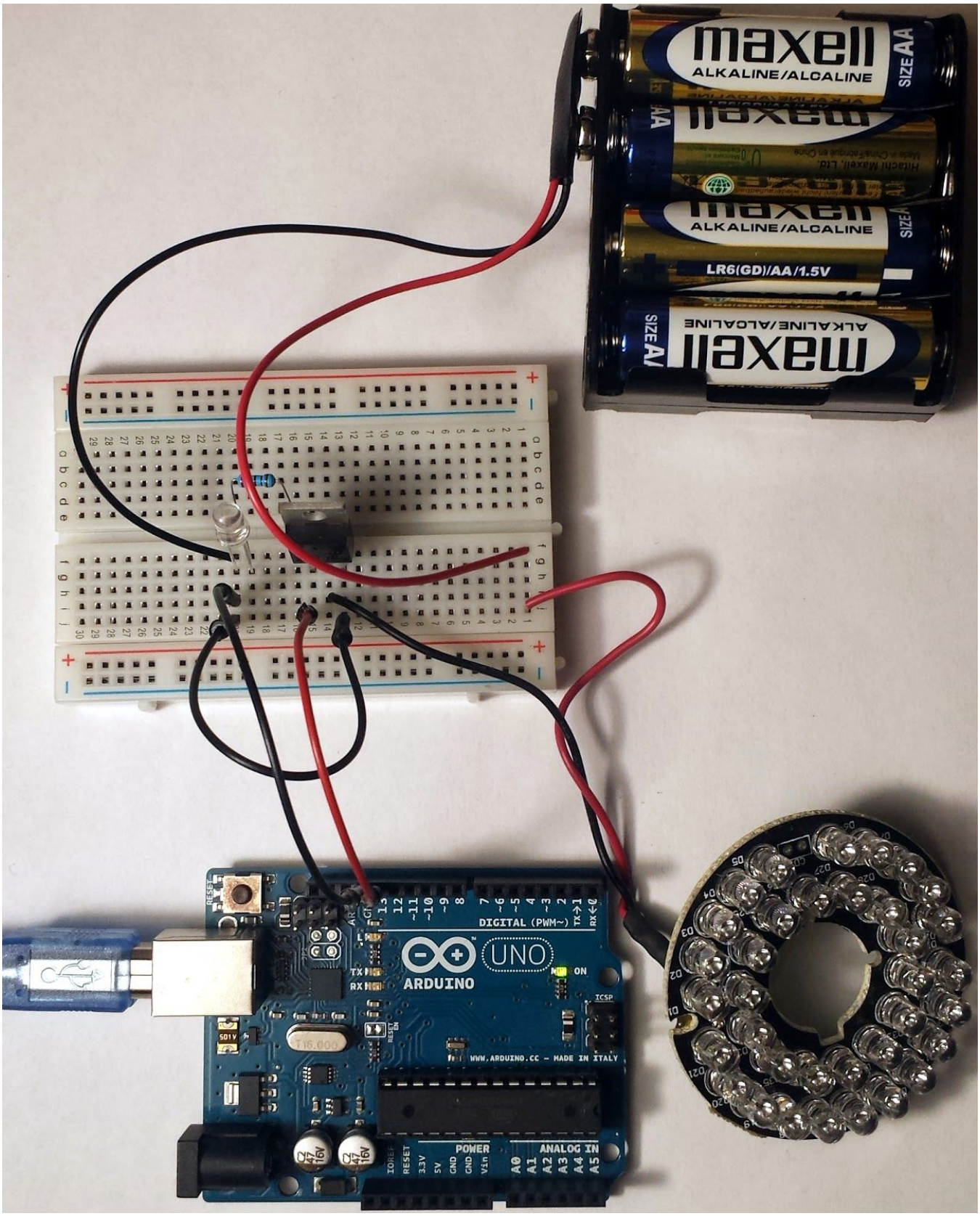
- You have to push hard to get the transistor into the breadboard.
- Make sure the left pin of the transistor is connected directly to pin 13, as in the diagram, *not* behind the resistor.
- Ground (black wire) on the battery should be connected to ground on the Arduino


maxell
ALKALINE/ALCALINE
SIZE AA
DIGITAL (PWM~)
UNO
ARDUINO
ON
ICSP
WWW.ARDUINO.CC – MADE IN ITALY
POWER
ANALOG IN

IR LED Array

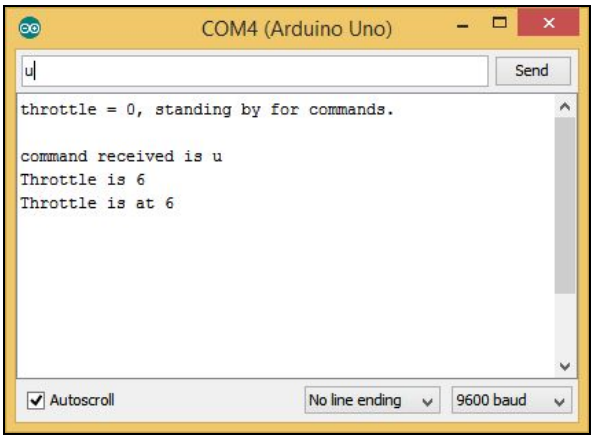
1. Replace the fan with the light as shown in below.
2. How can you tell if it's on? You can't see infrared, but these also leak a little into visible light...
3. (optional) Blink the IR LED at different rates. How fast can you make it blink?




maxell
ALKALINE/ALCALINE
SIZE AA
LR6(GD)/AA/1.5V
ARDUINO
UNO
DIGITAL (PWM~)
POWER
ANALOG IN
WWW.ARDUINO.CC – MADE IN ITALY

Basic Flight Control

1. Now we're going to control the helicopter via code!
2. Download the source code: http://www.stageoneeducation.com/helicopters.html
3. Open the helicopter code file: **helicopter_2_channel.ino**
4. Upload.
5. Open **Tools > Serial Monitor**
6. Turn on the helicopter and get it to sync with the correct arduino station.
7. To fly, enter into the serial monitor:
    a. Only one letter at a time. Must push "Enter" each time.
    b. **When in doubt, type 0 then Enter**
    c. You might need to power cycle helicopter to sync to the right channel.
    d. Commands:



u	increase throttle	**j**	decrease throttle
w	forwards	**s**	backwards
a	rotate left	**d**	rotate right
0	shutdown	**r**	recenter

- Hold the helicopter in your hand and watch the serial monitor.

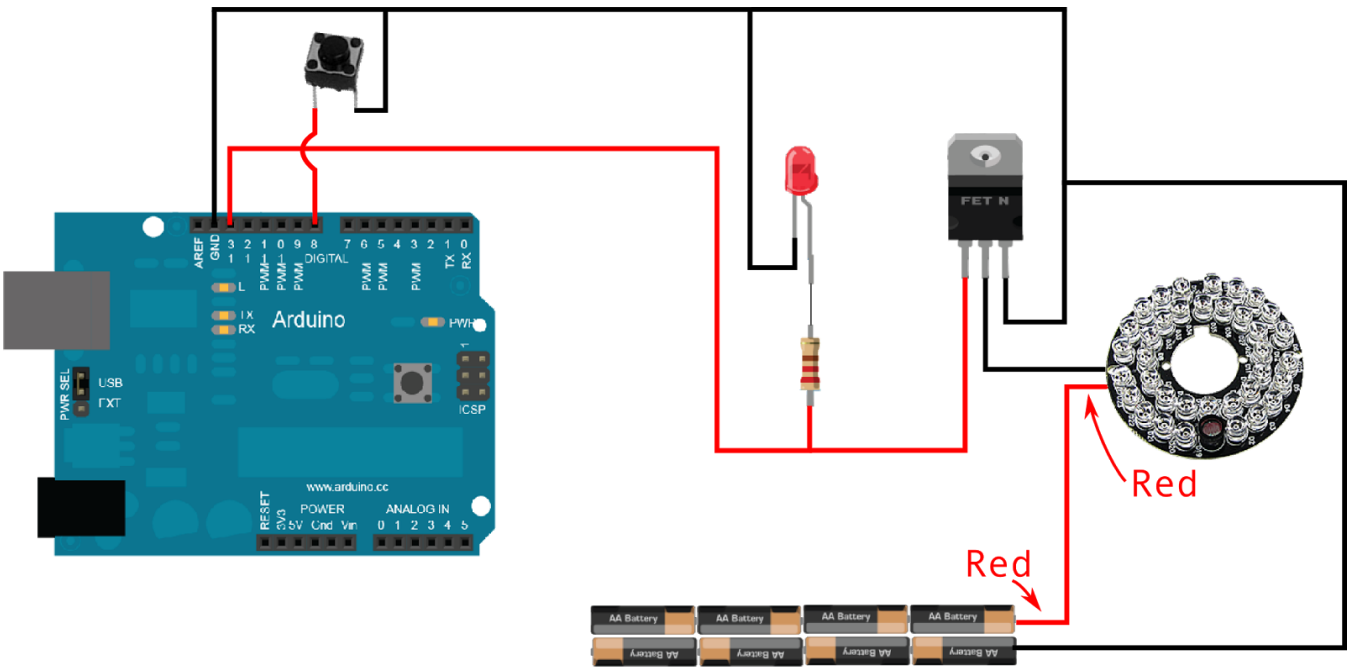
Adding a Button

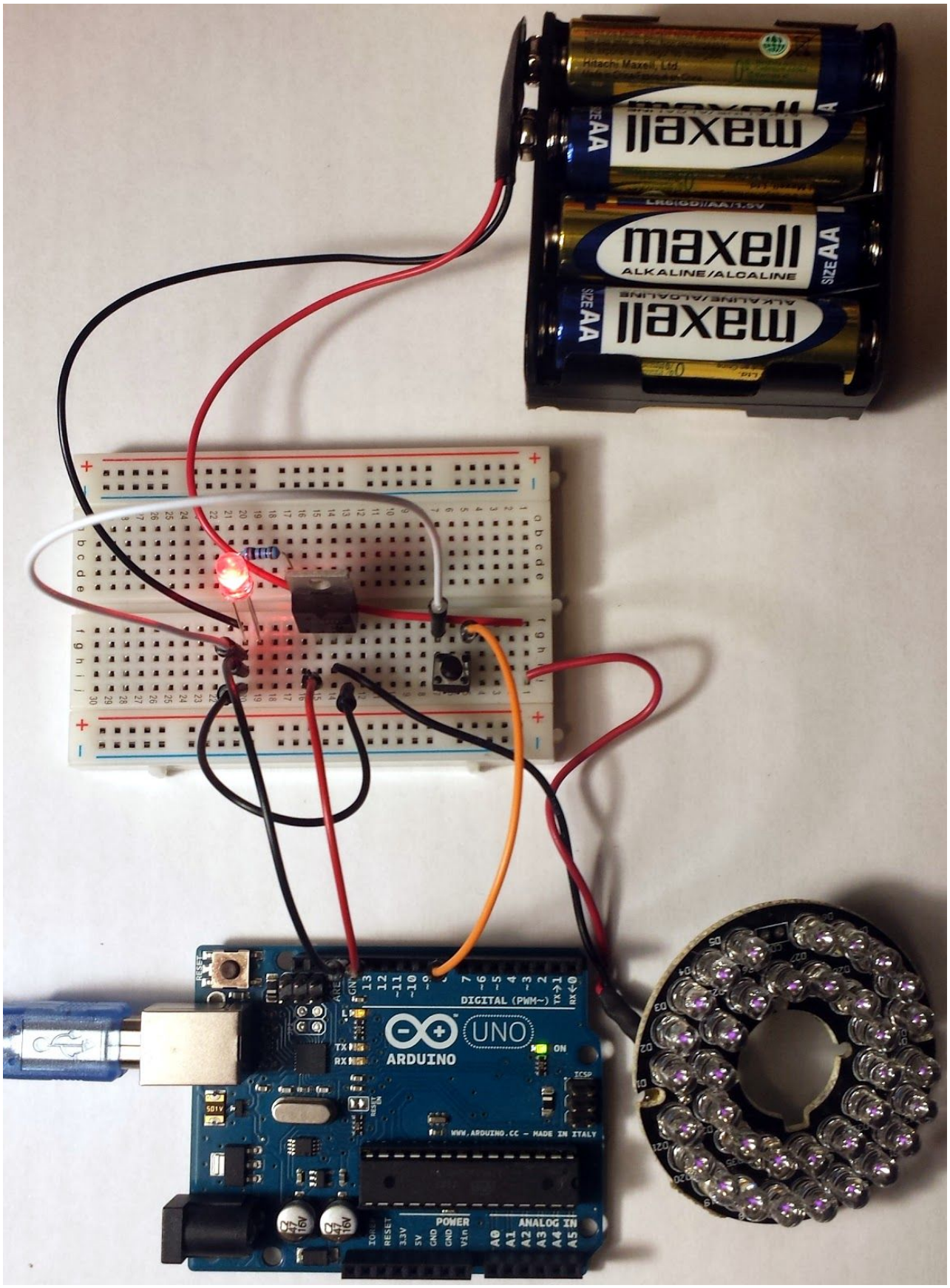
- **Charge your helicopter while you modify the circuit**

1. Modify your circuit to include a button. This is the "autonomous flight" button.
2. Make sure your button connects to pin **#8** on the Arduino.
3. When the code is running, check that when you hit the button, the **Serial Monitor** says "You hit the button."

Common Problems:

- If it says "You hit the button" even when you aren't pushing the button, you might have the button turned 90° the wrong way (connecting both of its pins together) in the breadboard.

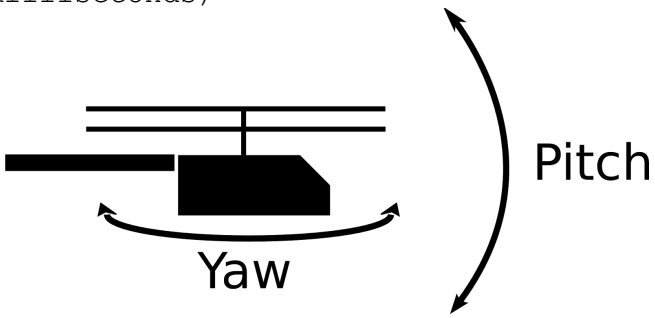

Arduino
FET N
Red
Red
AA Battery
AA Battery
AA Battery
AA Battery


maxell
ALKALINE/ALCALINE
SIZE AA
DIGITAL (PWM~)
UNO
ARDUINO
ON
POWER
ANALOG IN
WWW.ARDUINO.CC – MADE IN ITALY

Autonomous Takeoff

`HoldCommand(pitch, yaw, throttle, time in milliseconds)`

Pitch: `0 - 127,` **`63`** is neutral
Yaw: `0 - 127,` **`63`** is neutral
Throttle: `0 - 127,` **`127`** is slam into ceiling speed



1. Uncomment the example lines:

Get rid of these "//"

```
//HoldCommand(63, 63, 110, 500); // example take-off
                                 // start with lots of throttle!

//HoldCommand(63, 63, 65, 1000); // hover for 1 second
```

and upload your changes

2. Press the button and your helicopter should automatically take off
   Not working? Make sure it works with the serial monitor, then hit the button

Autonomous Flight Patterns

As always with engineering, it is best to start simple. Tune `(pitch, yaw, throttle, time in milliseconds)` for:

- **Flying forward a few feet, while maintaining the same height**
- **Turning 90 degrees**

Variables are your friends!

- While tuning parameters, it's a mess to change each number each time...

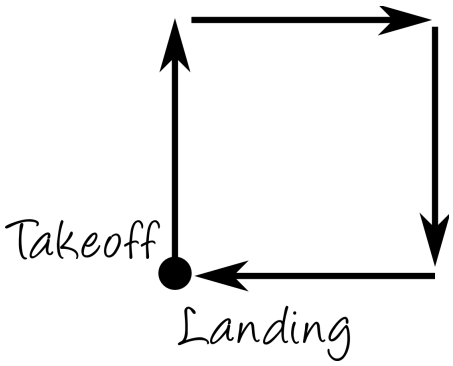
```
HoldCommand(63, 63, 60, 200);
HoldCommand(63, 63, 60, 200);
HoldCommand(63, 63, 60, 200);
HoldCommand(63, 63, 60, 200);
```

- Something like this is much easier!

```
int hoverThrottle = 60;
HoldCommand(63, 63, hoverThrottle, 200);
HoldCommand(63, 63, hoverThrottle, 200);
HoldCommand(63, 63, hoverThrottle, 200);
HoldCommand(63, 63, hoverThrottle, 200);
```

Now try an autonomous box

- Take off, fly forward, turn right, repeat 4 times, land.



Extra challenges:

1. Fly an autonomous Figure-8
2. Change the way the keyboard controls the helicopter (hint: look for `case 'u':` in the code and duplicate it)
3. Add a “Do a 360” button
4. (pretty hard, requires you to know a fair bit of C++): Make it so that, when the helicopter is flying, the button aborts the flight.

Feedback

Awesome job! We went through:

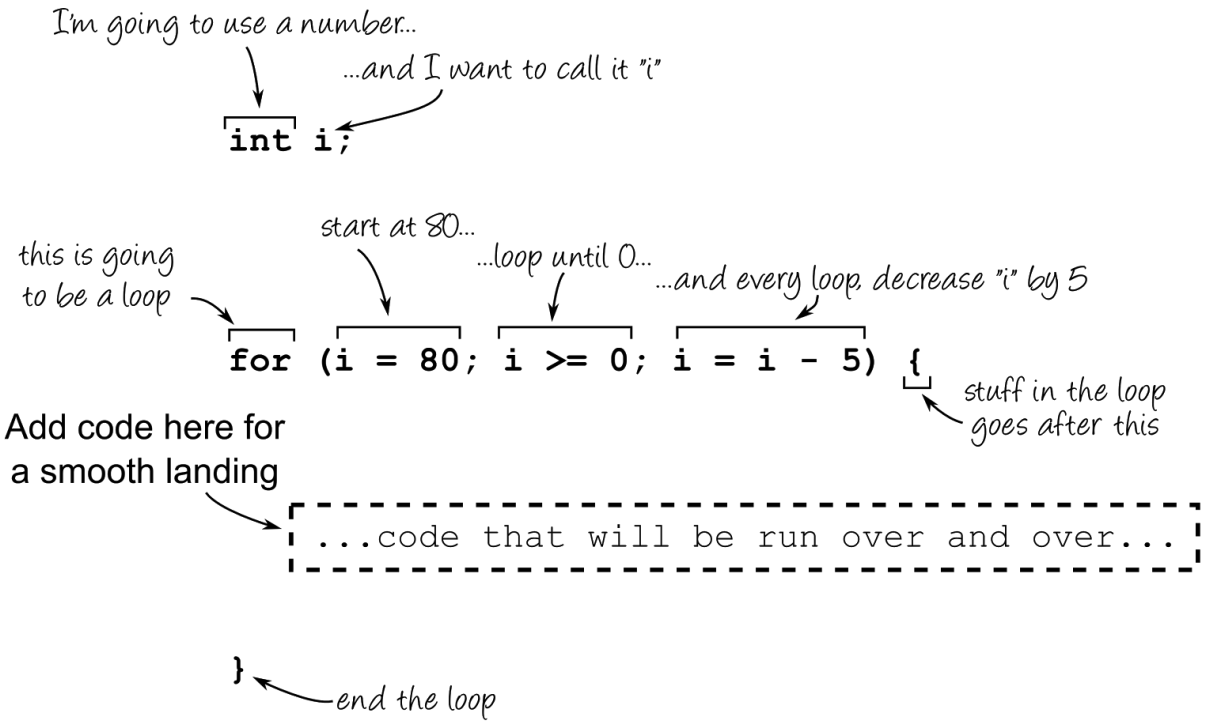
1. Blinking an LED
2. Manual helicopter flight
3. Controlling a motor with a transistor
    - (if you can run a motor, you can make a robotic car!)
4. Keyboard-driven helicopter flight
5. Autonomous flight

We’re constantly tweaking and improving! Help us:

- Head to http://tinyurl.com/heli-feedback to give us your feedback.
- Or use the QR Code to the right

Try a better landing with `for` loops.

A smooth landing with a `for` loop (real helicopters slowly spin down their propellers)
(type `the code` in, add the "`//`" back on the previous lines):



How to think about a for loop: