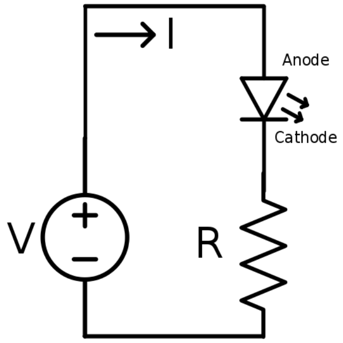
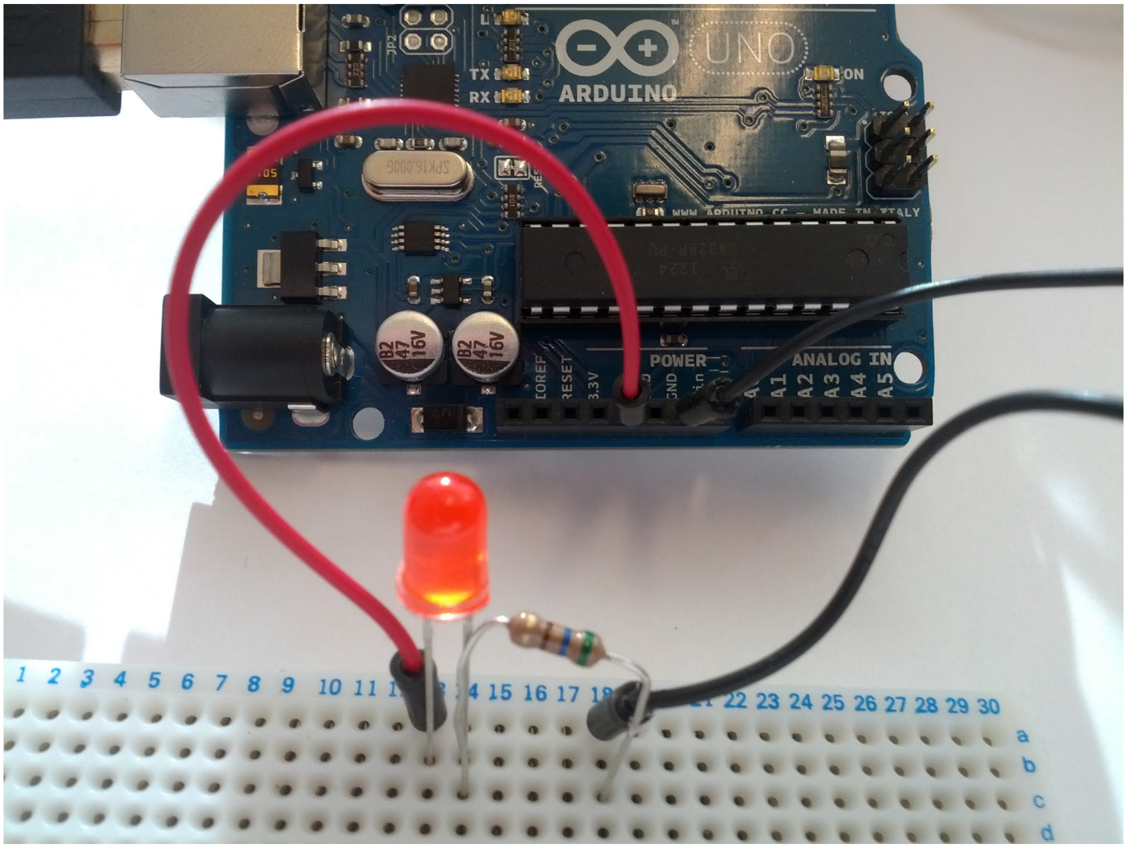
Telegraph 1: Sending

In the next assignment, we will start to explore telegraphs (i.e. writing from afar) in both analog and digital forms. You should submit your morse code program with a comment at the top answering the below questions called: **telegraph_lastname.py**

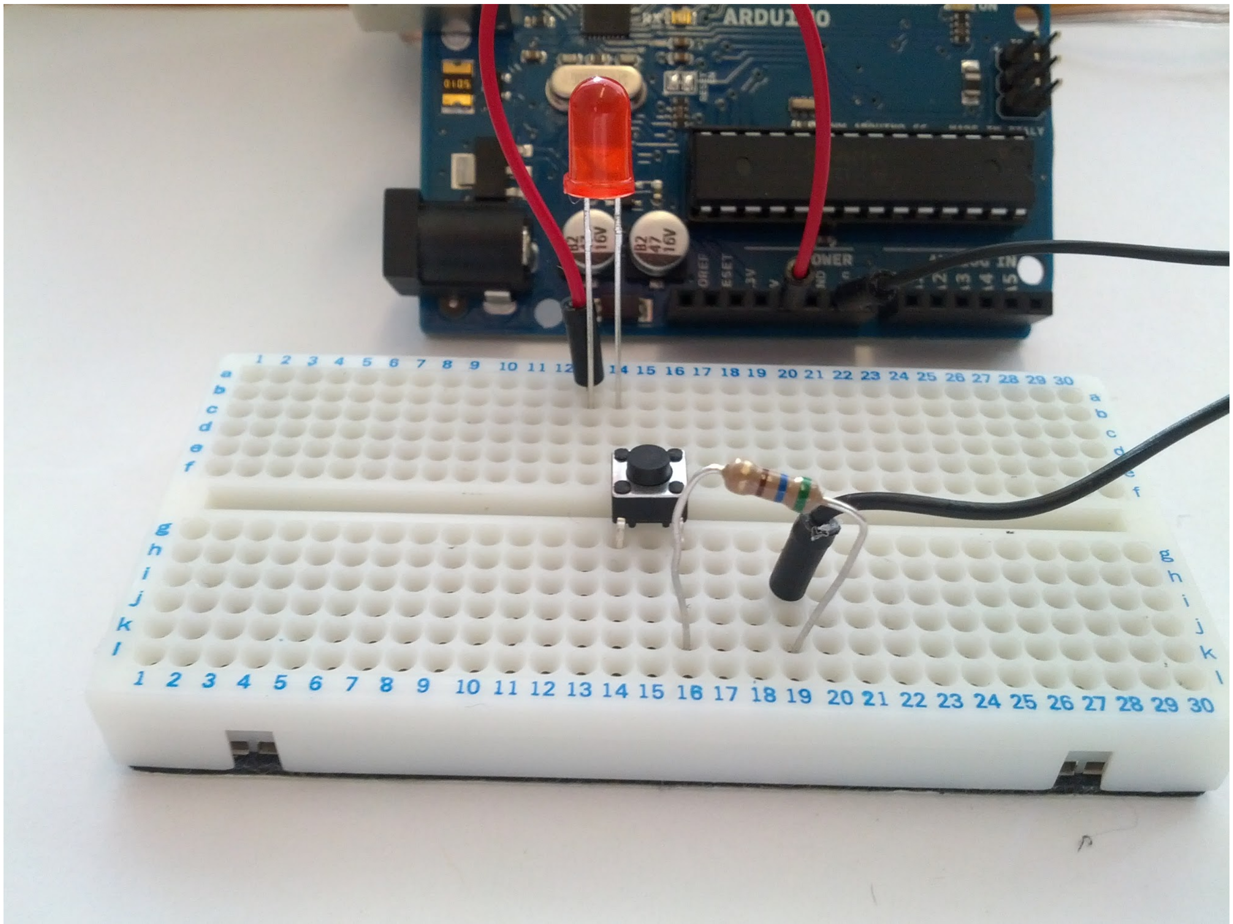
Analog 0: Powering on an LED

Wire up an LED using a resistor from the kit to the 5V and ground pins from your Arduino kit. The long end of the LED (anode) should be connected to the 5V port. And the short end c the LED (cathode) to GND. The order (polarity) matters for the LED, but not the resistor. What happens when you connect the red wire 50 the 3.3V port? http://upload.wikimedia.org/wikipedia/commons/thumb/c/c9/LED_circuit.svg/500px-LED_circuit.svg.png	

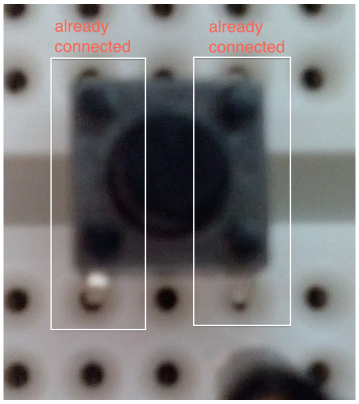


Analog 1: Controlling an LED

Put a switch in series with the LED. When you press the switch the LED should flash.

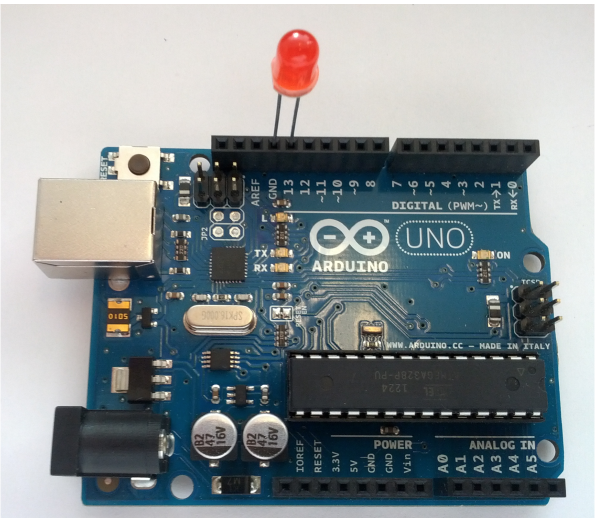


The switch

The four-legged switch is a little strange. Two pairs of the pins are already connected before the switch is ever pressed. When the switch is pressed all four pins become connected.	

Digital 0: Turning on an LED

Plug the long leg of the LED (the anode) into pin #13 and the short leg (the cathode) into ground (GND) port. Once you connect the Arduino to your computer using the USB cord, you can interact with the board using Calico. The following code will turn on the LED. Myro, like Processing, is a module that includes many functions for interacting with robots, devices, etc.



```
from Myro import *

makeRobot("Arduino",

”/dev/tty.usbmodem1411")

makeDigitalOutput(13)
digitalWrite(13, 1)
```

Note: In order to connect to your arduino you need to find the name of the USB connection. In the Calico shell type:

```
    from os import listdir
    listdir(“/dev”)
```

Your port should be named “/dev/tty.usbmodem” followed by a number.

Digital 1: Controlling the Morse Code

Instead of wiring the LED to the 5V power through the switch, connect the anode of the LED (the long leg) to Pin 13.   Using the keyPressed() event handler, blink out the morse code for the letter pressed.

```
from Processing import *

window(300, 300)
textSize(128)                              # font size
fill(24)
noStroke()
textAlign(CENTER, CENTER)                  # text alignment

def keyHandler():
    background(196)
    if key() == 'A' or key() == 'a':       # check if key is an ‘a’
        text("A", 150, 150)                # draw “A” at location (150, 150)
    if key() == 'B' or key() == 'b':       # check if key is a ‘b’
        text("B", 150, 150)                # draw “B” at location (150, 150)

onKeyPressed += keyHandler                 # call keyHandler when key is pressed
loop()                                     # start the sketch
```

Here is an alternate form:

```
from Processing import *

window(300, 300)
textSize(128)                                  # font size
fill(24)
noStroke()
textAlign(CENTER, CENTER)                      # text alignment

while True:                                    # forever
    background(196)
    if isKeyPressed():                         # if the key is currently pressed
        if key() == 'A' or key() == 'a':       # check if key is an ‘a’
            text("A", 150, 150)                # draw “A” at location (150, 150)
        if key() == 'B' or key() == 'b':       # check if key is a ‘b’
            text("B", 150, 150)                # draw “B” at location (150, 150)
```

`delay(10)`	#delay 10 milliseconds