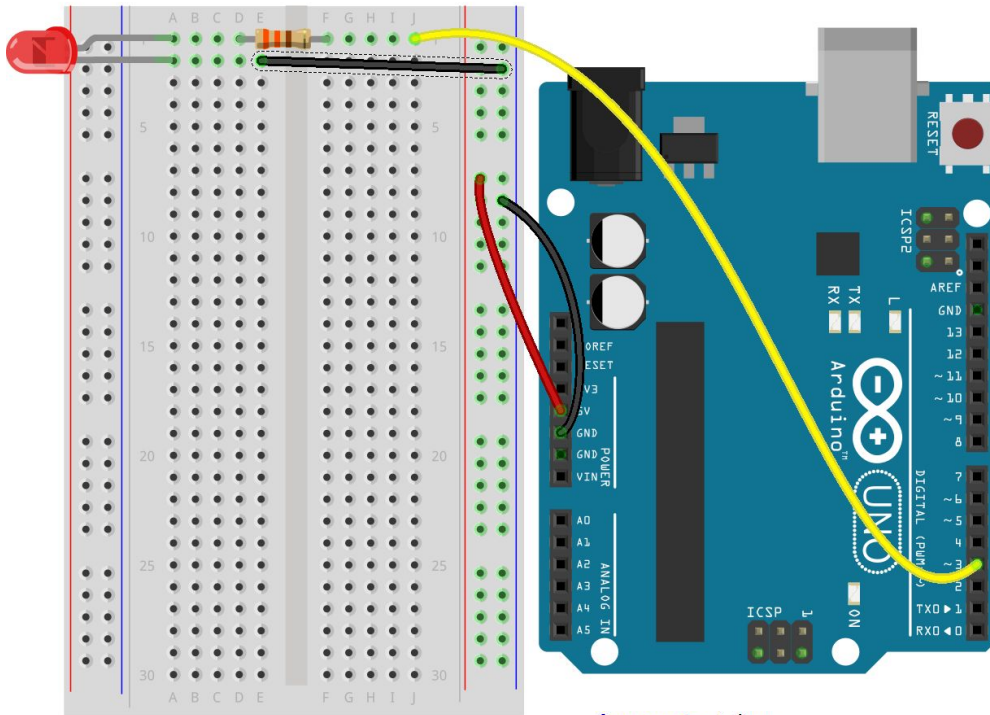# IoT with Python and Arduino

Workshop with Shawn Hymel

Pre-flight checklist:
- Download Arduino IDE from arduino.cc
- Upload *File > Examples > Firmate > StandardFirmata* to Arduino
- Install Python 3.7.x from python.org
- Install pyFirmata with `pip install pyfirmata`
- Create an account at thingspeak.com

1 Firmata Blinky

Arduino is running StandardFirmata

```python
import time

from pyfirmata import Arduino, util

# Parameters
COM_PORT = 'COM37'
SLEEP = 0.5                          # seconds

# Connect to firmata
print('Connecting to Arduino')
board = Arduino(COM_PORT)

# Define pins
d3 = board.get_pin('d:3:o')

# Do forever
while True:
    d3.write(1)
    time.sleep(SLEEP)
    d3.write(0)
    time.sleep(SLEEP)
```
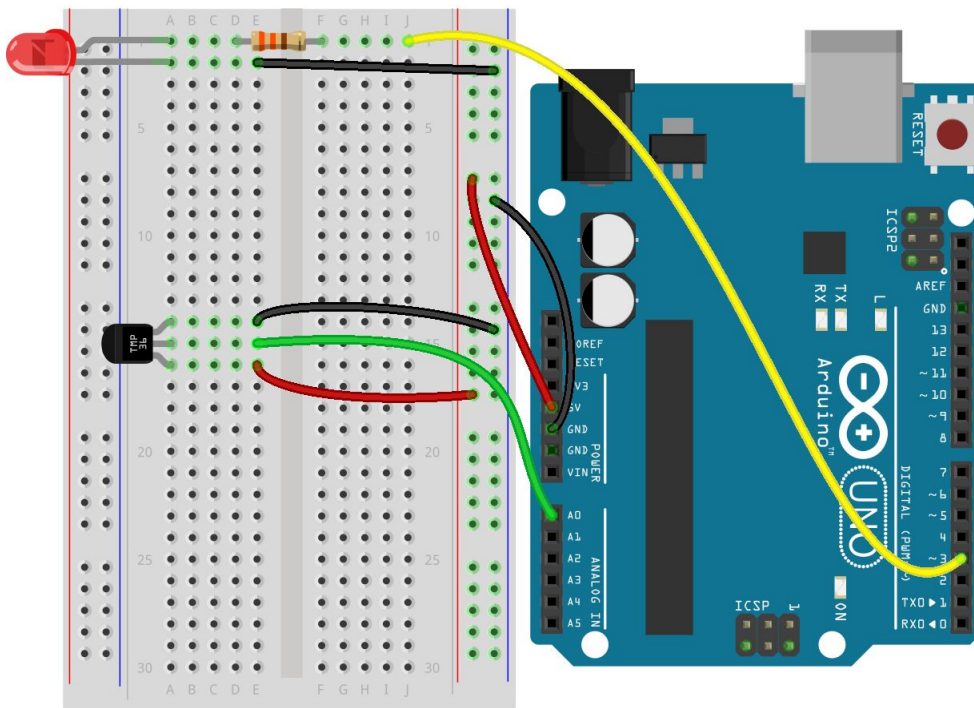
Change to your Arduino's COM port

Save this code as blink.py on your computer and run with `python blink.py`

Challenge: Make the LED fade in and out (hint: read the README at github.com/tino/pyFirmata)

2 Read Temperature

Arduino is running StandardFirmata

Change to your Arduino's COM port

Save this code as temp.py on your computer and run with `python temp.py`

```python
import time

from pyfirmata import Arduino, util

# Parameters
COM_PORT = 'COM37'
SLEEP = 1.0                              # seconds
AREF = 5.0

# Connect to firmata
print('Connecting to Arduino')
board = Arduino(COM_PORT)

# Define pins
a0 = board.get_pin('a:0:i')
d3 = board.get_pin('d:3:o')

# Start thread to sample analog pin values
it = util.Iterator(board)
it.start()
time.sleep(0.1)

# Do forever
while True:

    # Read temperature
    voltage = a0.read() * AREF
    temp_val = (voltage - 0.5) * 100
    temp_str = '%.1f'%temp_val
    print("Temperature: " + temp_str + " C")

    # Wait before next request
    time.sleep(SLEEP)
```

Challenge: Blink the LED each time you read the temperature sensor.

3 HTTPS GET

```python
import http.client

HOSTNAME = 'example.com'

# Construct GET request
get_str = '/index.html'
print(HOSTNAME + get_str)

# Send request (update channel)
conn = http.client.HTTPSConnection(HOSTNAME)
conn.request('GET', get_str)
resp = conn.getresponse()
print(resp.status, resp.reason)

# Print page
for line in resp.read().splitlines():
    print(line)
```

Challenge: Print out the contents of a different page

4 Post something to ThingSpeak (enter into a browser)

https://api.thingspeak.com/update?api_key=BMZ44FSBTE6JANSU&field1=4.2

Change to your own API key

Challenge: Add a second field and post something to it.

5 IoT Weather Station

Change to
your own
COM port
and API key →

```python
import http.client
import time

from pyfirmata import Arduino, util

# Parameters
COM_PORT = 'COM37'
API_KEY = 'BMZ44FSBTE6JANSU'
HOSTNAME = 'api.thingspeak.com'
AREF = 5.0
SLEEP = 10                                  # seconds

# Connect to firmata
print('Connecting to Arduino')
board = Arduino(COM_PORT)

# Define pins
a0 = board.get_pin('a:0:i')
d3 = board.get_pin('d:3:o')

# Start thread to sample analog pin values
it = util.Iterator(board)
it.start()
time.sleep(0.1)

# Do forever
while True:

    # Turn on LED to show that we are sampling
    d3.write(1)

    # Read temperature
    voltage = a0.read() * AREF
    temp_val = (voltage - 0.5) * 100
    temp_str = '%.1f'%temp_val
    print("Temperature: " + temp_str + " C")

    # Construct GET request
    get_str = '/update?api_key=' + API_KEY + '&field1=' + temp_str
    print(HOSTNAME + get_str)

    # Send request (update channel)
    conn = http.client.HTTPSConnection(HOSTNAME)
    conn.request('GET', get_str)
    resp = conn.getresponse()
    print(resp.status, resp.reason)

    # Turn off LED to show that we are done
    d3.write(0)

    # Wait before next request
    time.sleep(SLEEP)
```

Challenge: Add a photocell to your weather station and post light values (as a second field) in addition to your temperature values.