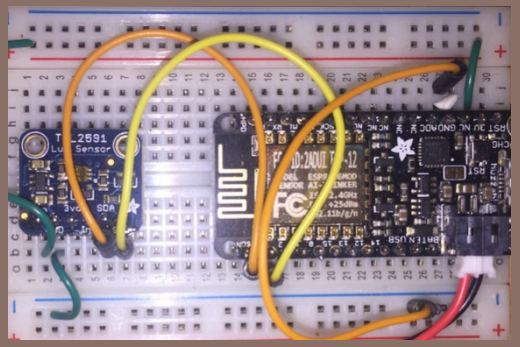
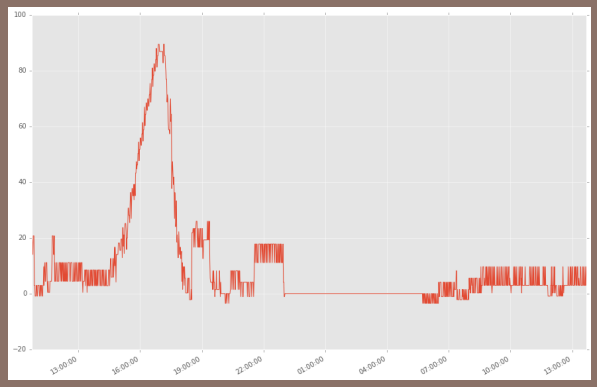
MICROPYTHON IOT HACKATHON



Featuring the ESP8266

Jeff Fischer
Data-Ken Research
jeff@data-ken.org
https://data-ken.org
Sunnyvale, California, USA



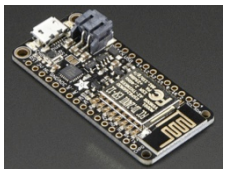
January 2017

Tonight's Agenda

- Overview lecture
- Build and test system (hardware and software)
- Additional projects (if time permits)

Why Python for IoT?

- High-level, easy to prototype ideas and explore options
- Runs on embedded devices



- Python data analysis ecosystem

NumPy

Array and matrix processing

pandas

High level data analysis tools

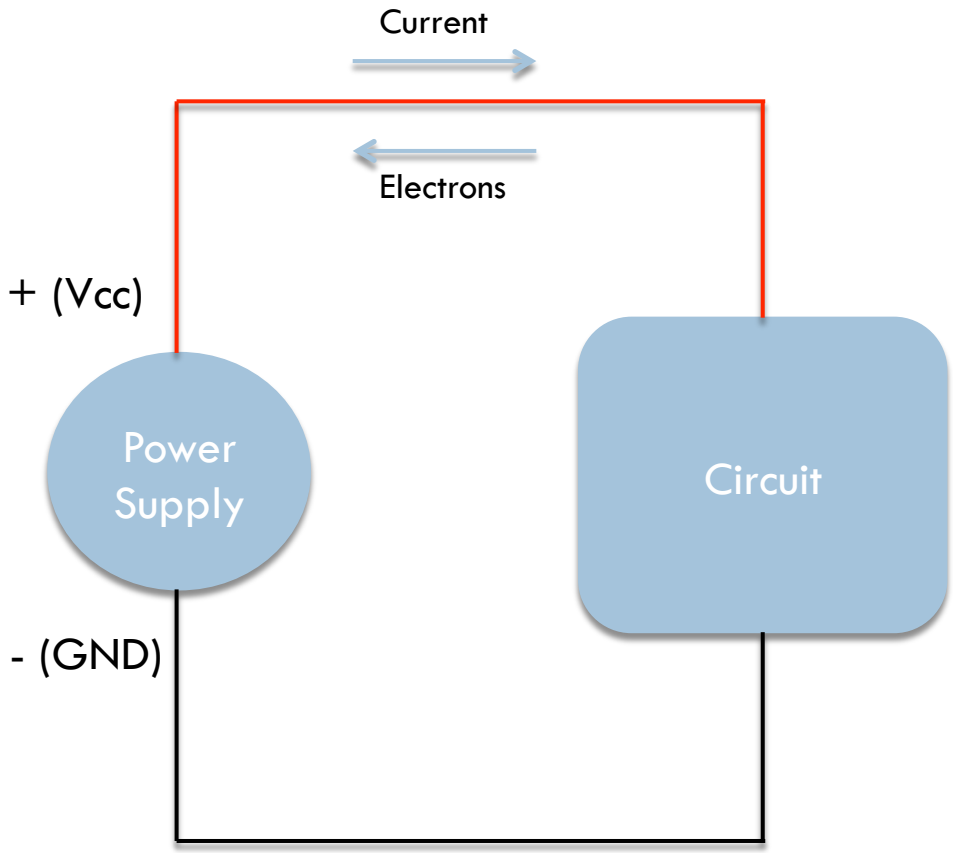
SciPy

Numerical analysis routines

scikit learn HMMLearn

Machine learning

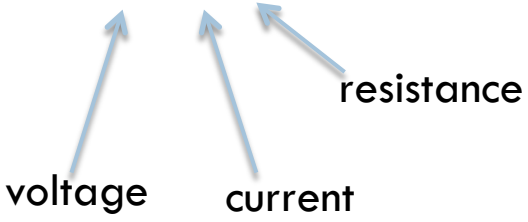
Basic Electronics



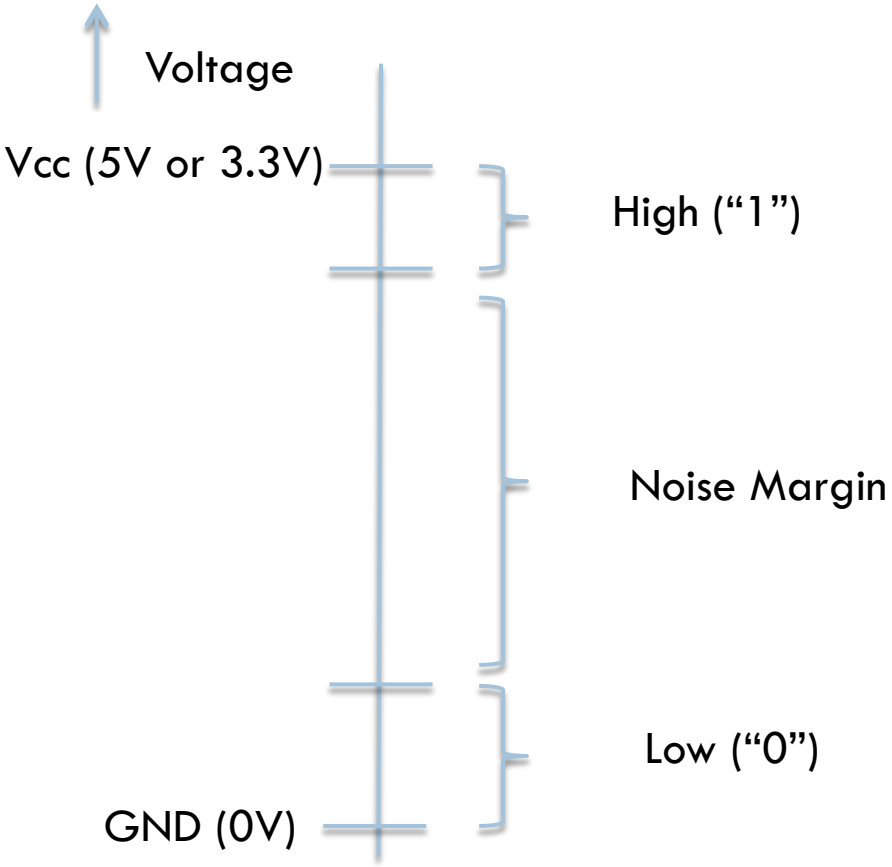
Voltage = Electrical Pressure
Current = Flow of electric charge
Resistance = Difficulty to pass electric charge

Ohm's Law

$V = I R$



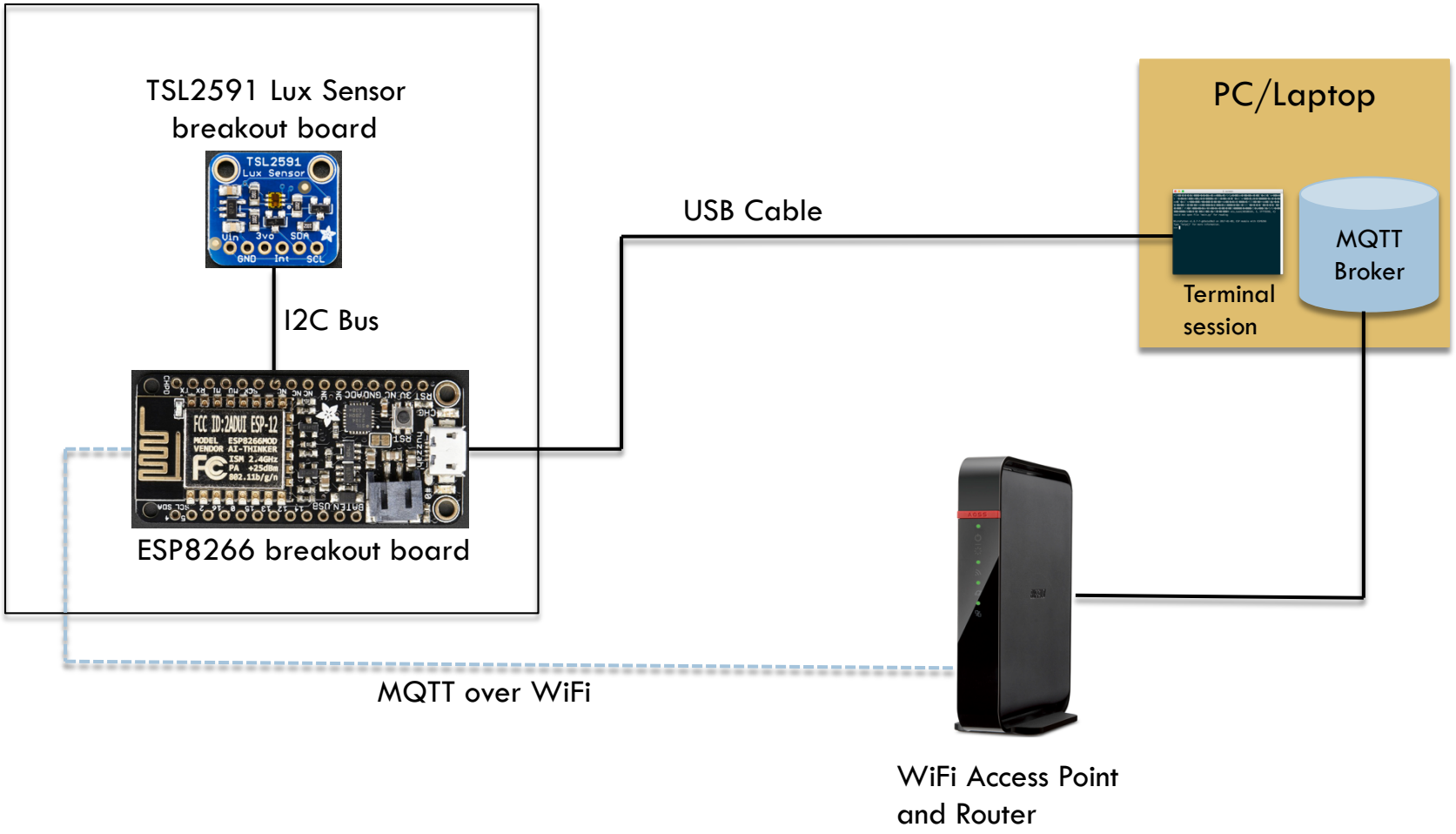
Digital Logic



Cautions

- Do not connect power and ground directly to each other ("shorting")
- Chips are sensitive to static discharge, be careful
  - You might touch some metal (e.g. your laptop chassis before handling the electronics)

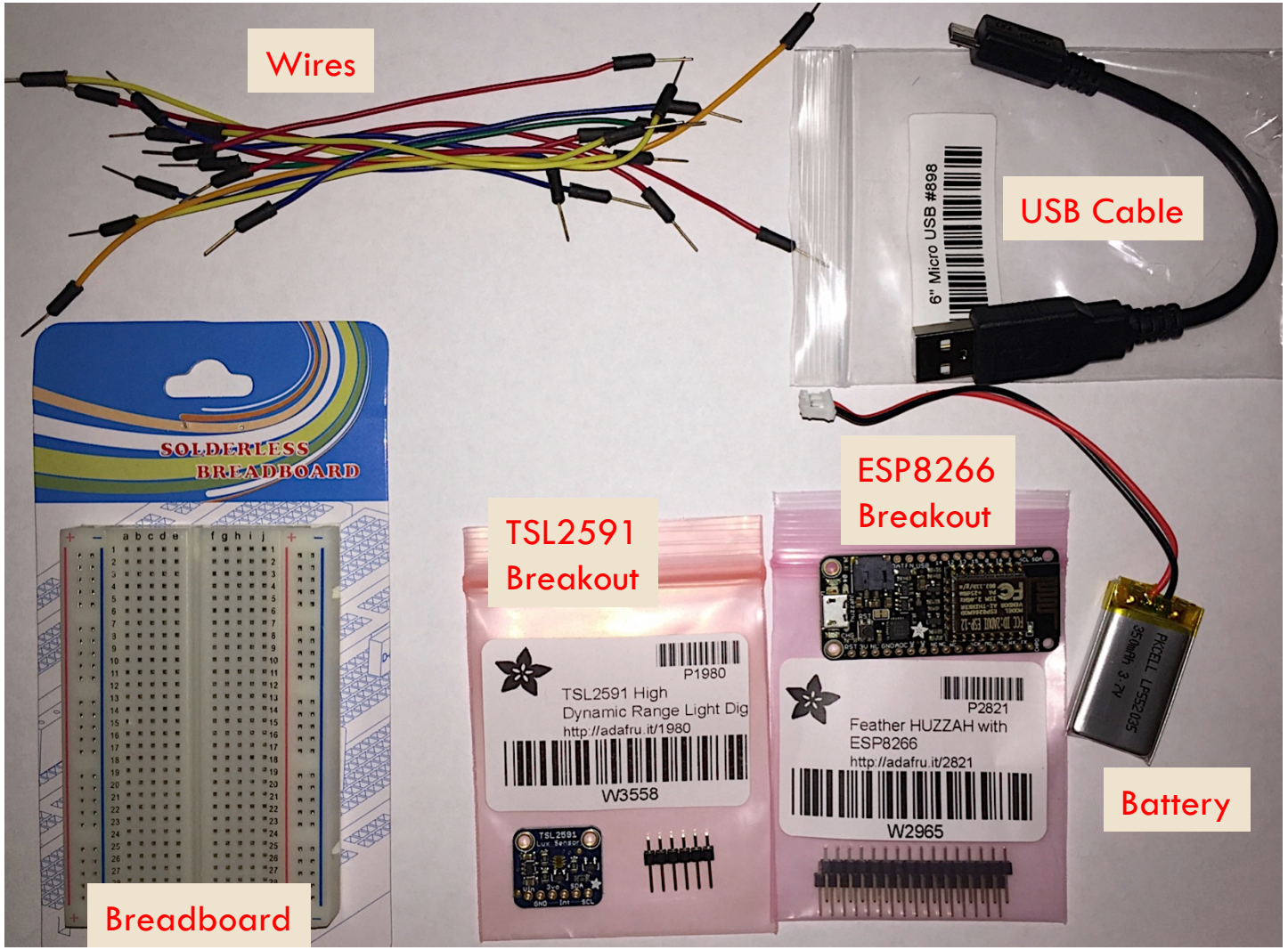
System Overview



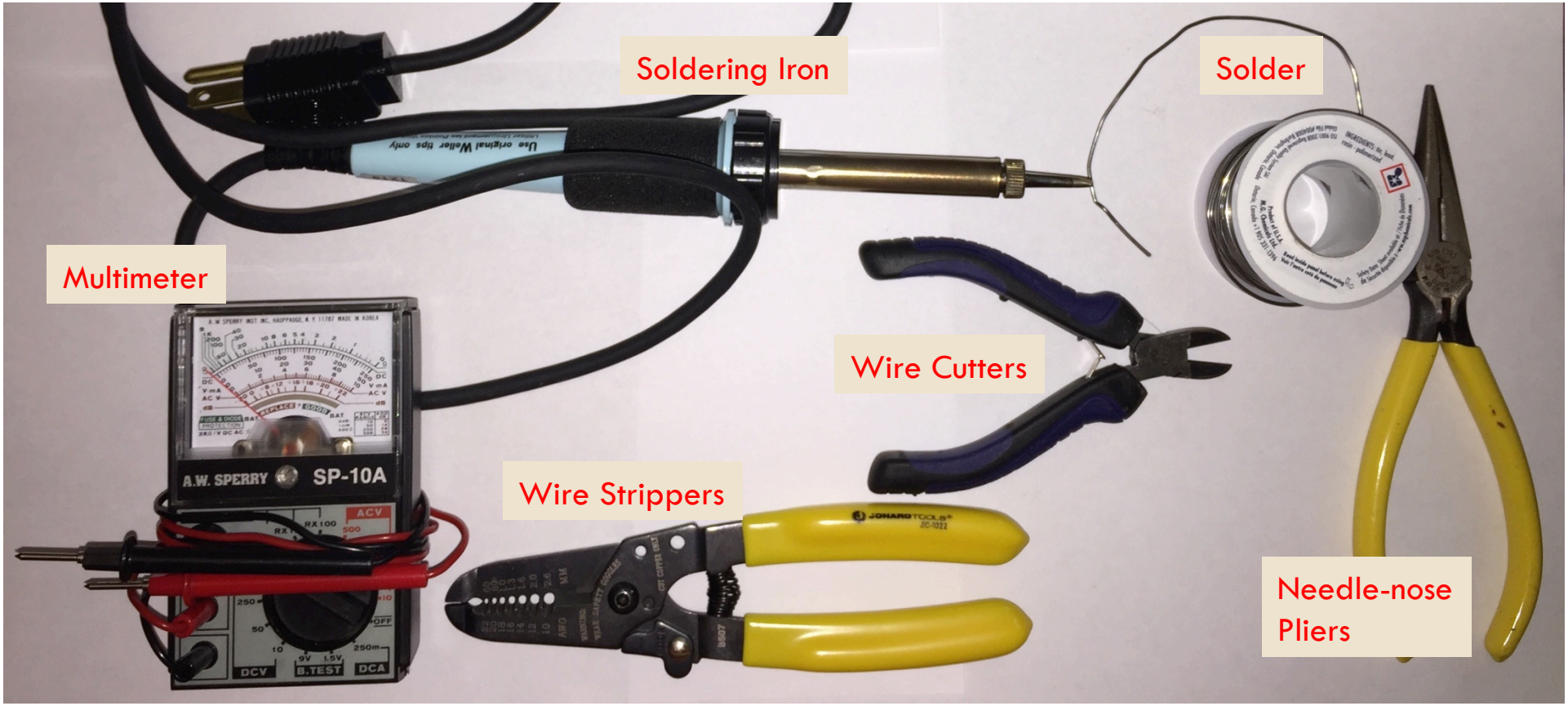
Steps

1. Hardware Assembly
2. Firmware and software install
3. Application to read the sensor
4. Messaging with MQTT

Parts

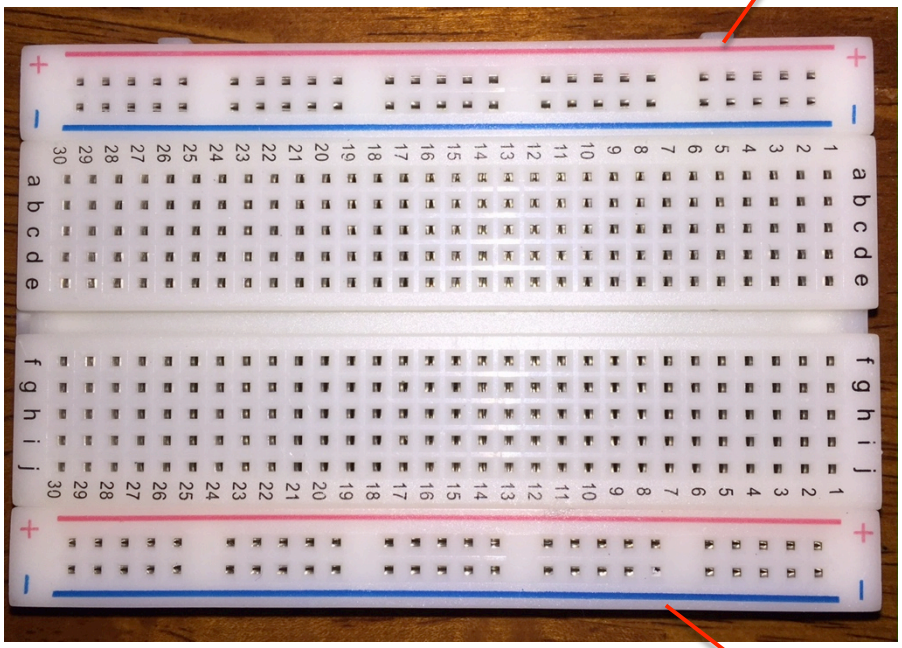


Recommended Tools



Breadboards

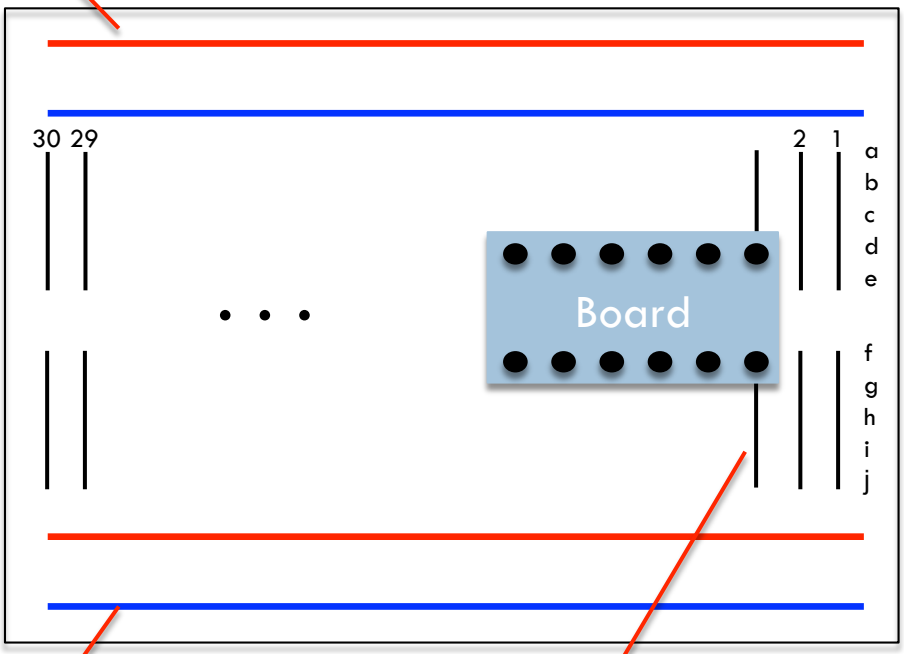
Photo



Use for Power

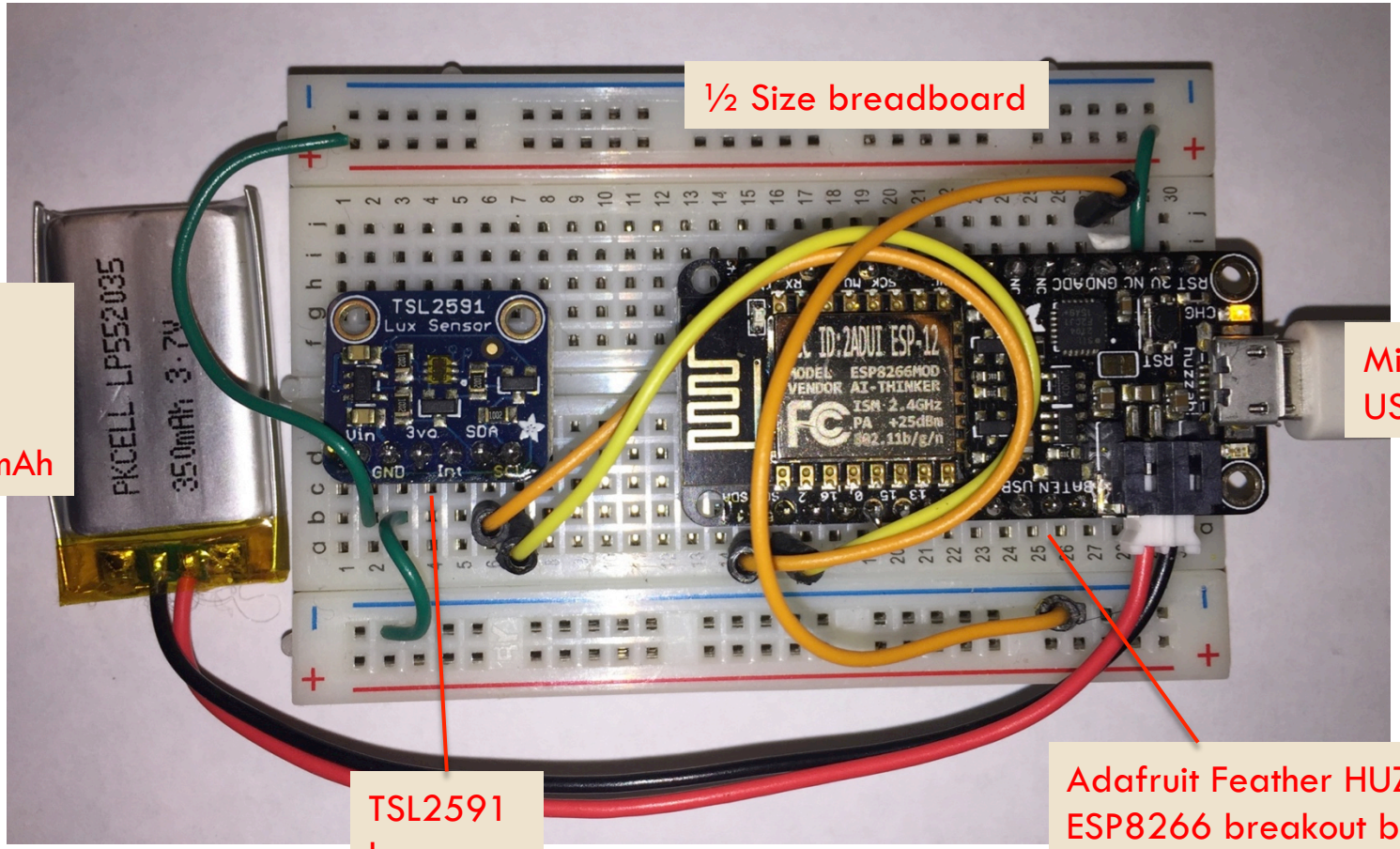
Use for GND

Electrical Connections

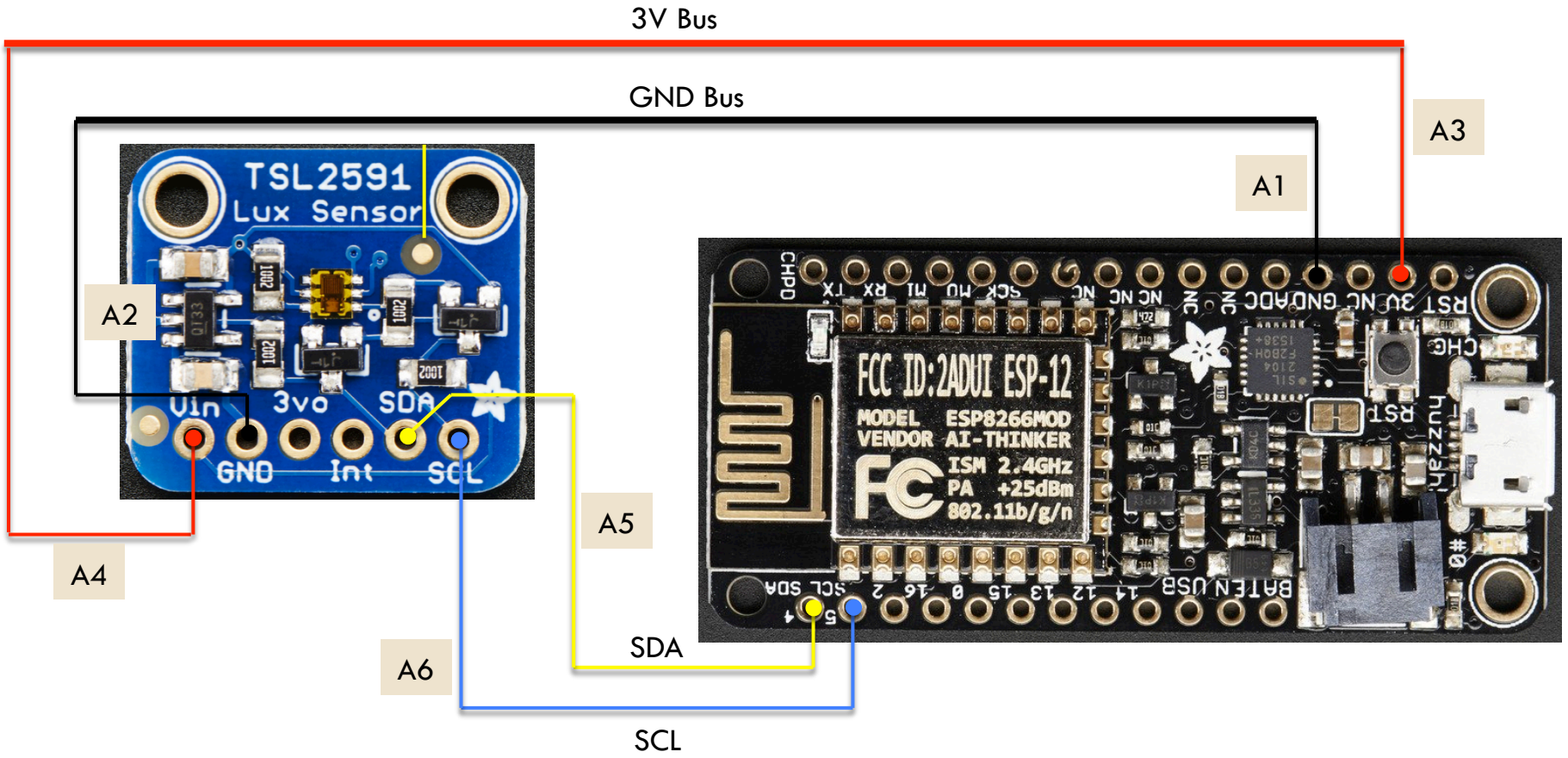


Use for pin connections

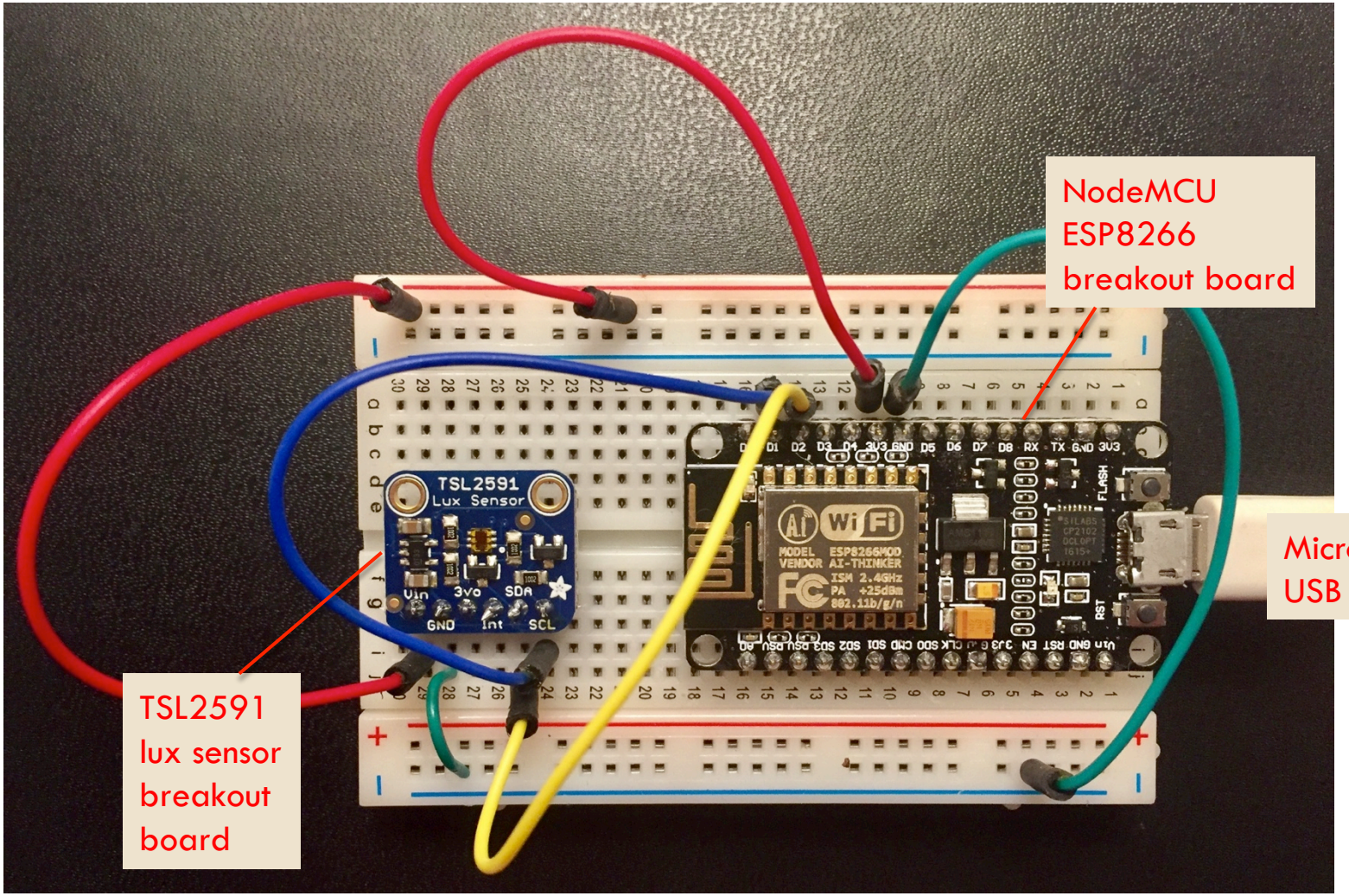
System with Adafruit Feather HUZZAH



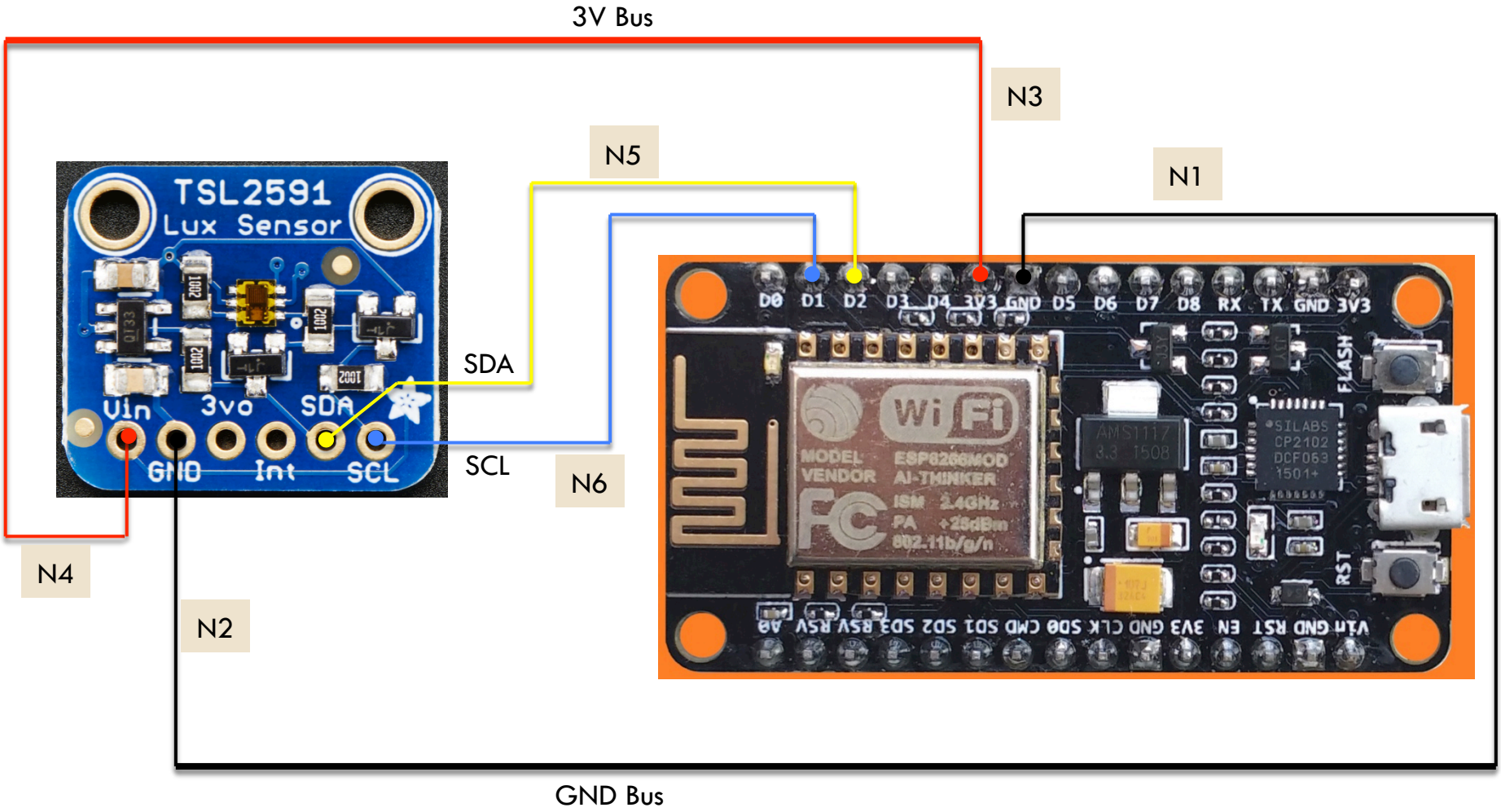
Feather HUZZAH: Wiring Diagram



System with NodeMCU

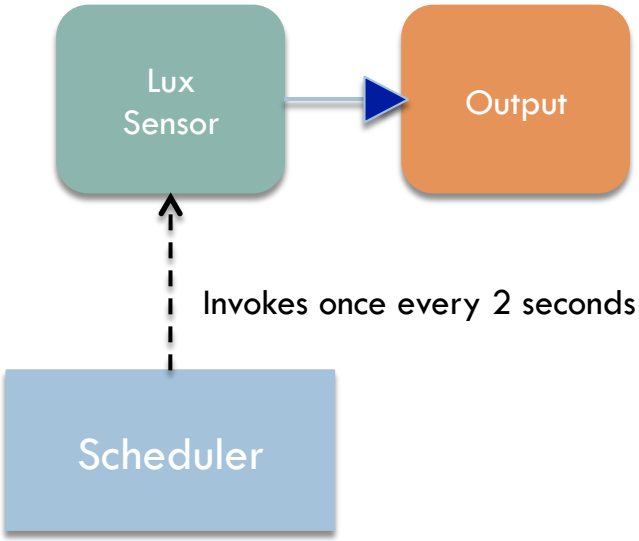


NodeMCU: Wiring Diagram

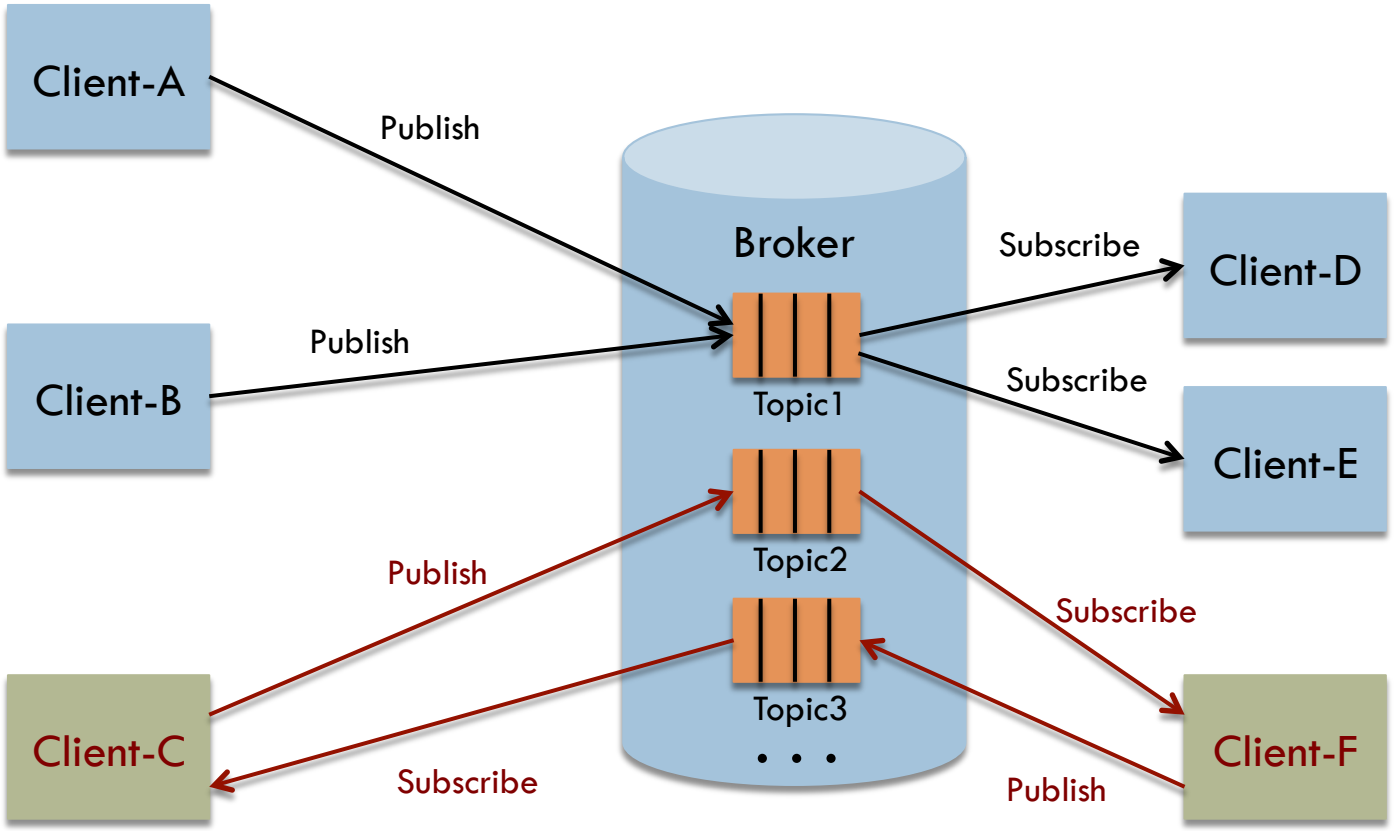


Reading the Light Sensor

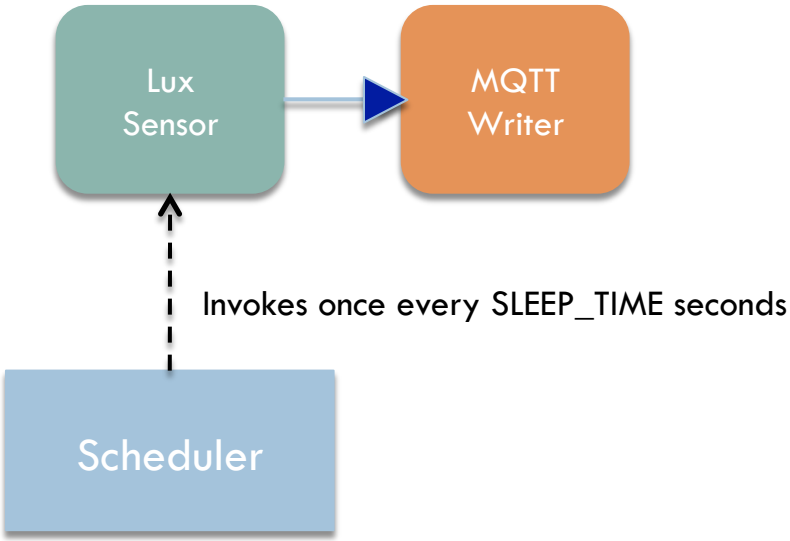

```
from antevents import *
from tsl2591 import Tsl2591
tsl = Tsl2591('lux-1')
sched = Scheduler()

sched.schedule_sensor(tsl, 2.0, Output())
sched.run_forever()
```



MQTT



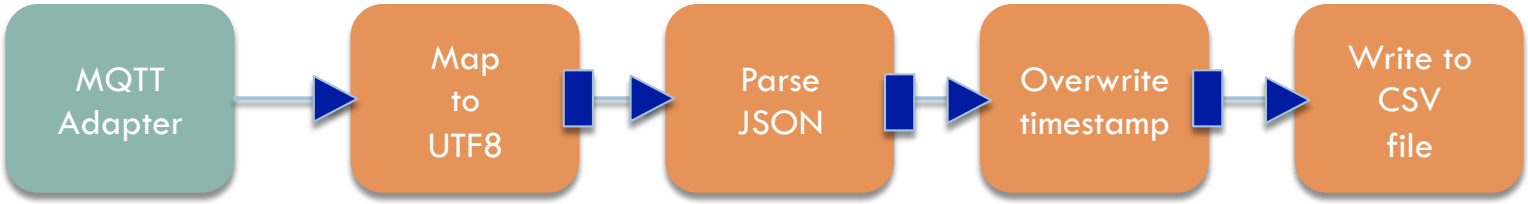
Client.py: Send events to MQTT



Client.py: Send events to MQTT

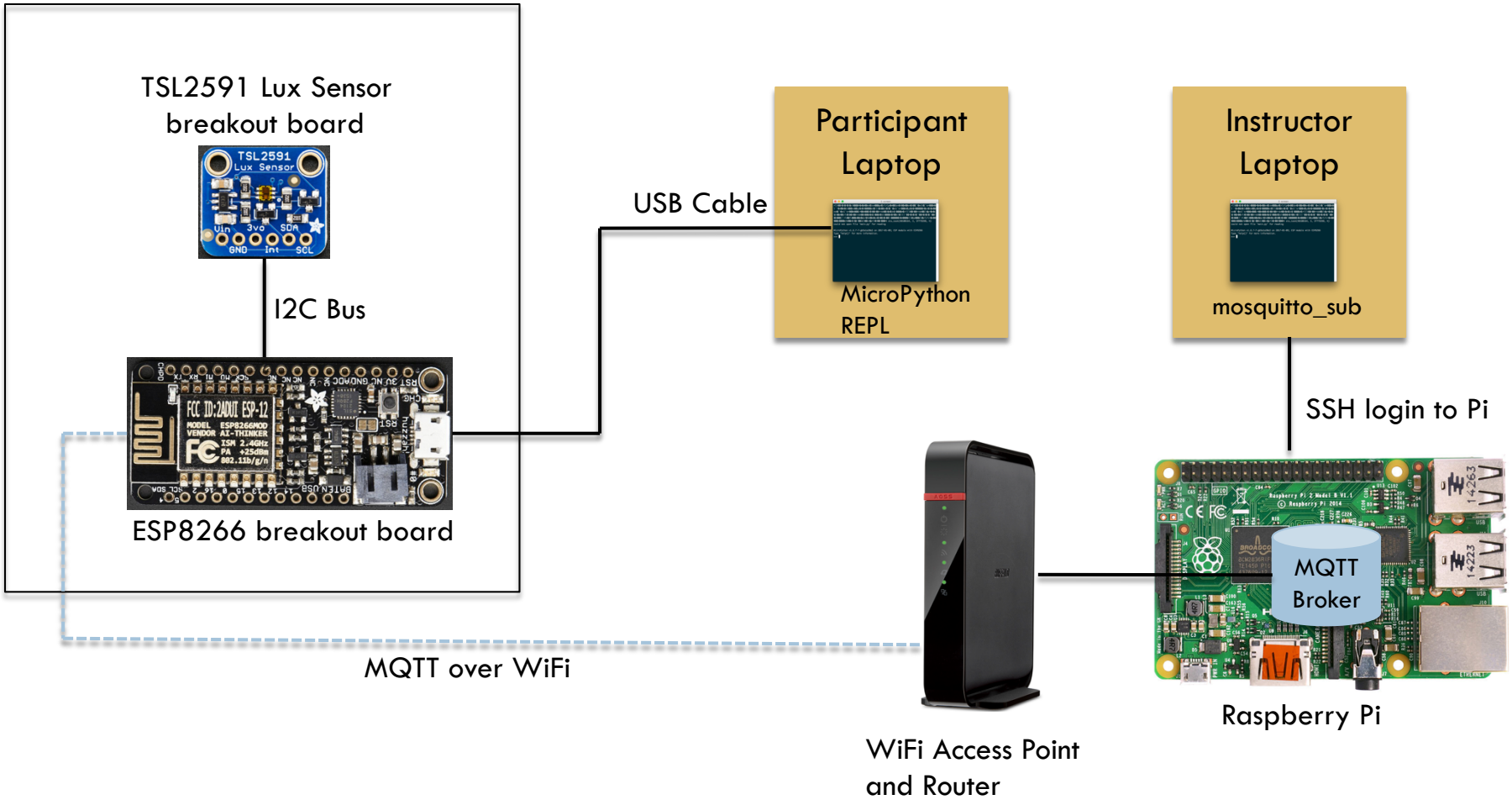

```
# Configuration is stored in a separate config.py
from config import SENSOR_ID, WIFI_ESSID, WIFI_PASSWORD, MQTT_HOST,\
                   MQTT_TOPIC, SLEEP_TIME

from wifi import wifi_connect, disable_wifi_ap
from antevents import *
from tsl2591 import Tsl2591
from mqtt_writer import MQTTWriter

# setup network
disable_wifi_ap()
wifi_connect(WIFI_ESSID, WIFI_PASSWORD)


# create objects
m =  MQTTWriter(SENSOR_ID, MQTT_HOST, 1883, MQTT_TOPIC)
tsl = Tsl2591(SENSOR_ID)
sched = Scheduler()

# schedule and run
sched.schedule_sensor(tsl, SLEEP_TIME, m)
print("Running main loop with sample every %s seconds..." % SLEEP_TIME)
sched.run_forever()
```

Server.py




```
mqtt.select(lambda m:(m.payload).decode('utf-8'))\
    .from_json(constructor=SensorEvent)\
    .select(lambda evt: SensorEvent(sensor_id=evt.sensor_id,
                                    ts=time.time(),
                                    val=evt.val))\
    .csv_writer(filename)
```

Today's MQTT Setup



Next Steps

- Follow the detailed instructions in the online documentation, starting with "Hardware Assembly"
- You can skip the section on the MQTT broker – you can connect to my Raspberry Pi
- If you get done early, take a look at the extra projects section. Some projects you might try:
  - Graph the light data in Jupyter
  - Turn on an LED
  - Read a temperature sensor
  - Read a door open/close sensor
- Feel free to ask for help!

Thank You

Questions?

More information

Email: jeff@data-ken.org

Hackathon Tutorial: http://micropython-iot-hackathon.readthedocs.io/en/latest/

Website and blog: https://data-ken.org

AntEvents: https://github.com/mpi-sws-rse/antevents-python