1) Connect to WiFi Network.

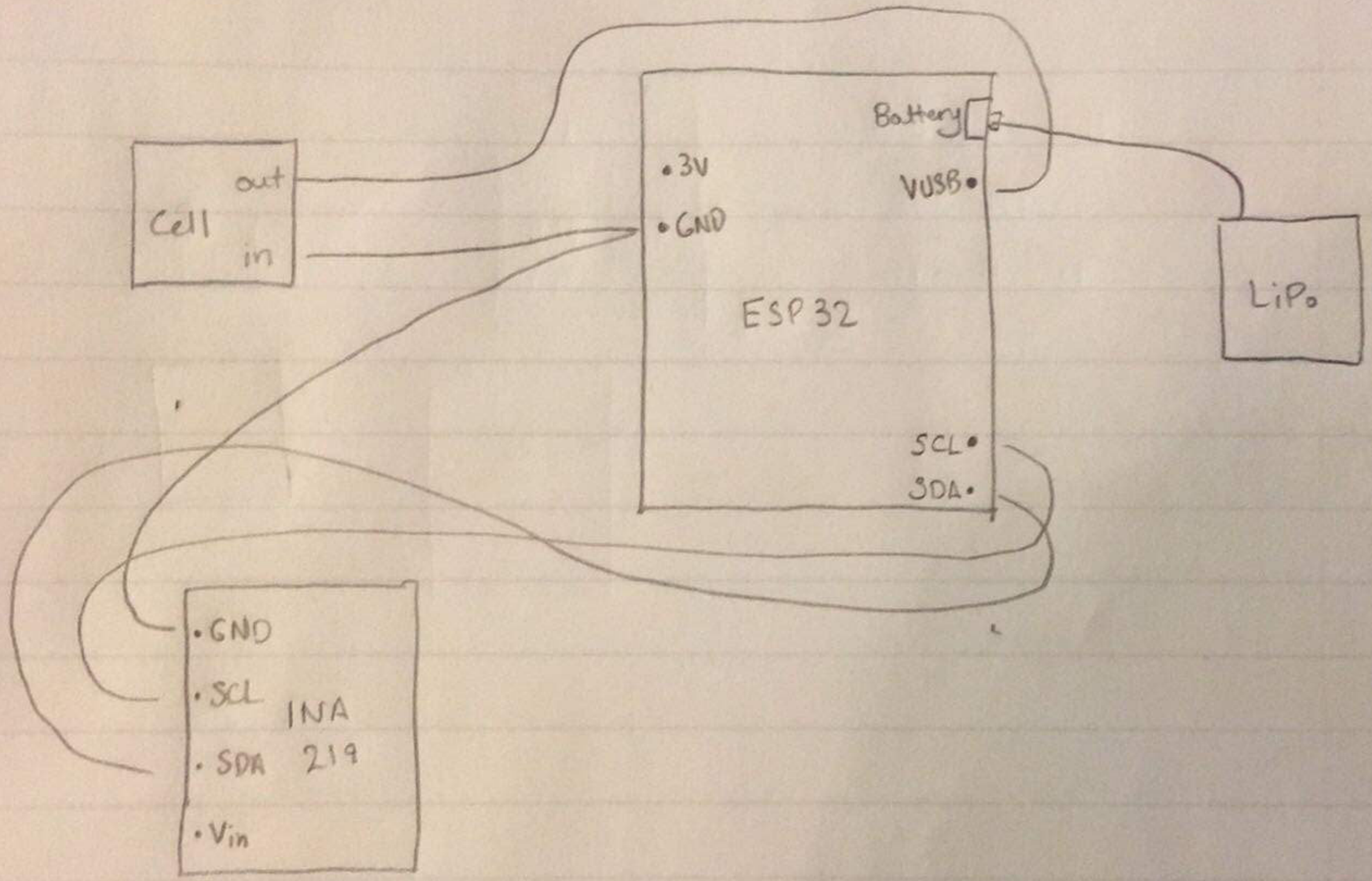
IP Address: 10.0.0.228

Date and Time: Mon Feb 18 19:13:23 2019

```
import network, time, machine

wlan = network.WLAN(network.STA_IF)
wlan.active(True)
if wlan.isconnected():
    print("No WiFi Connection")
    # code to handle the problem ...
print("Connecting to Wifi ... ")
wlan.connect('Next time for sure though', 'Dontworryaboutit', 5000)

for _ in range(1000):
    if wlan.isconnected():
        print("Connected to WiFi!")
        break
    time.sleep_ms(100)
if not wlan.isconnected():
    print("Unable to connect to WiFi")
    wlan.disconnect()

machine.WDT(False)

rtc = machine.RTC()
rtc.ntp_sync(server="pool.ntp.org")
for _ in range(100):
    if rtc.synced(): break
    time.sleep_ms(100)
if rtc.synced():
    print(time.strftime("%c", time.localtime()))
else:
    print("Unable to get ntp time")
```

```python
from mqttclient import MQTTClient
import network
import math

# Important: change the line below to a unique string.
session = "rimuru"
BROKER = "iot.eclipse.org"

# check wifi connection
wlan = network.WLAN(network.STA_IF)
wlan.active(True)
ip = wlan.ifconfig()[0]
if ip == '0.0.0.0':
    print("no wifi connection")
    # code to handle the problem ...
else:
    print("connected to WiFi at IP", ip)

# connect to MQTT broker
print("Connecting to MQTT broker", BROKER, "...", end="")
mqtt = MQTTClient(BROKER)
print("Connected!")

# send data
# In this sample, we send "fake" data. Replace this code to send useful data,
# e.g. measurement results.
for t in range(100):
    s = math.sin(t/10)
    # add additional values as required by application
    topic = "{}/data".format(session)
    data = "{},{}".format(t, s)
    print("send topic='{}' data='{}'".format(topic, data))
    mqtt.publish(topic, data)

# do the plotting (on host)
print("tell host to do the plotting ...")
mqtt.publish("{}/plot".format(session), "create the plot")

# free up resources
# alternatively reset the microphyton board before executing this program again
mqtt.disconnect()
```

```python
import paho.mqtt.client as paho
import matplotlib.pyplot as plt

# Important: change the line below to a unique string,
# e.g. your name & make corresponding change in mqtt_plot_mpy.py
session = "rimuru"
BROKER = "iot.eclipse.org"
qos = 0

# connect to MQTT broker
print("Connecting to MQTT broker", BROKER, "...", end="")
mqtt = paho.Client()
mqtt.connect(BROKER, 1883)
print("Connected!")

# initialize data vectors
# in this example we plot only 1 value, add more as needed
t = []
s = []

# mqtt callbacks
def data(c, u, message):
    # extract data from MQTT message
    msg = message.payload.decode('ascii')
    # convert to vector of floats
    f = [ float(x) for x in msg.split(',') ]
    print("received", f)
    # append to data vectors, add more as needed
    t.append(f[0])
    s.append(f[1])

def plot(client, userdata, message):
    # customize this to match your data
    print("plotting ...")
    plt.plot(t, s, 'rs')
    plt.xlabel('Time')
    plt.ylabel('Sinusoid')
    print("show plot ...")
    # show plot on screen
    plt.show()

# subscribe to topics
data_topic = "{}/data".format(session, qos)
plot_topic = "{}/plot".format(session, qos)
mqtt.subscribe(data_topic)
mqtt.subscribe(plot_topic)
mqtt.message_callback_add(data_topic, data)
mqtt.message_callback_add(plot_topic, plot)

# wait for MQTT messages
# this function never returns
print("waiting for data ...")
```


Cell
out
in
3V
GND
Battery
VUSB
ESP32
LiPo
SCL
SDA
GND
SCL
SDA
Vin
INA
219