TECHIN514: Lab 5

Name: Diyun Lu SID: 2430741 Net ID: ludiyun

1 Transmitting your HC-SR04 Data to Firebase

Take a screenshot on your Firebase RTDB's webpage to show that your data has been successfully uploaded.

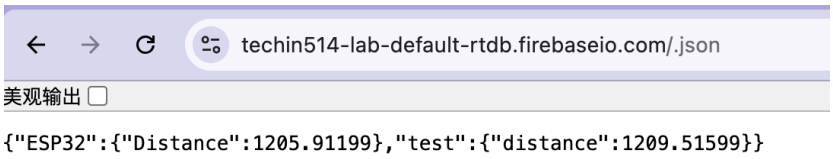

Figure 1: Data Received by Firebase

2 Power Consumption Measuring

Take a screenshot of the plot on your power profiler app to show the power consumption change during 1 minute with five different mode or stages of ESP32S3. Annotate what each stage stands for.

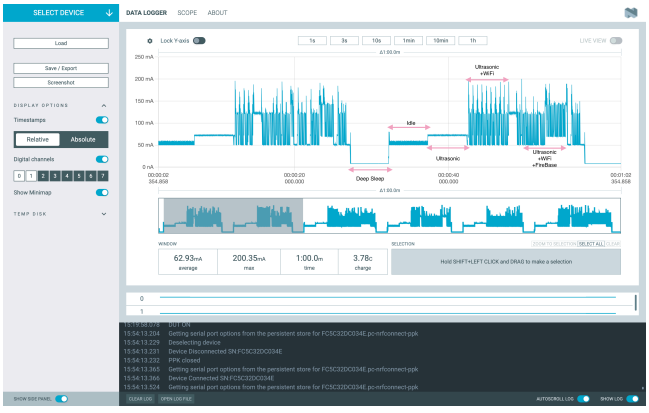

Figure 2: Overall Consumption

Set the time window from 1 minute to 10 seconds and take a screenshot for each power-using mode accordingly. Calculate how long a 500mAh battery would last when the ESP32S3 is in each mode.

- Deep Sleep Mode: average 17.89 mA during the five seconds. $time = 500mAh \div 17.89 \approx 27.95h$

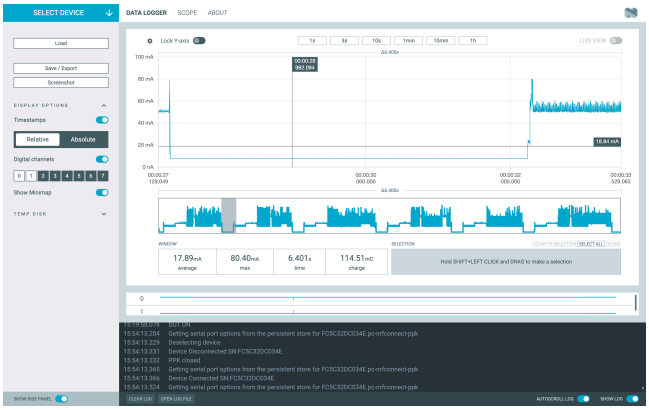

Figure 3: Deep Sleep Mode

- Idle ESP32 Mode: average of 52.17 mA during the five seconds. $time = 500mAh \div 52.17 \approx 9.58h$

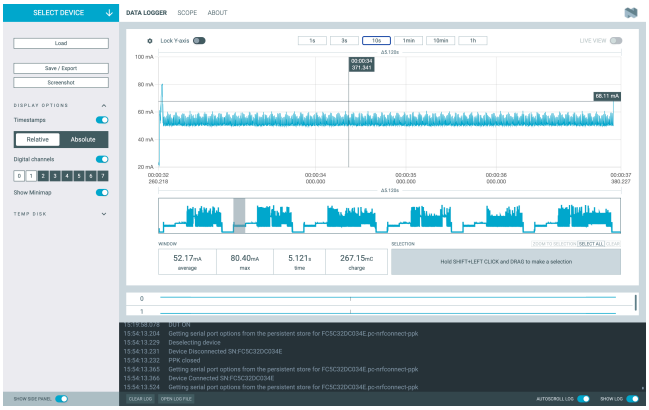

Figure 4: Idle Mode

- Ultrasonic Mode: average of 72.16 mA during the five seconds. $time = 500mAh \div 72.16 \approx 6.93h$

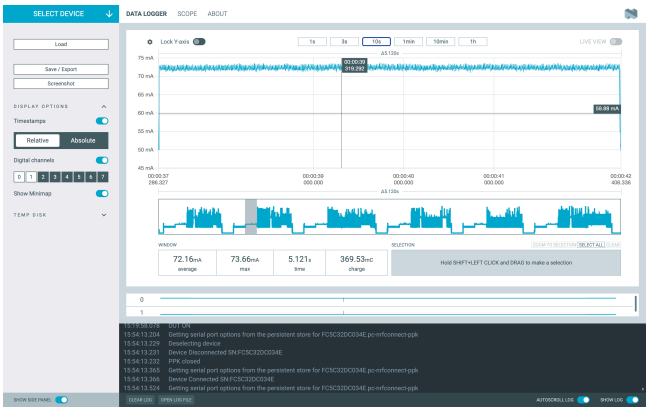

Figure 5: Ultrasonic Mode

- Ultrasonic + WiFi Mode: average 74.43 mA during the five seconds. $time = 500mAh \div 74.43 \approx 6.72h$

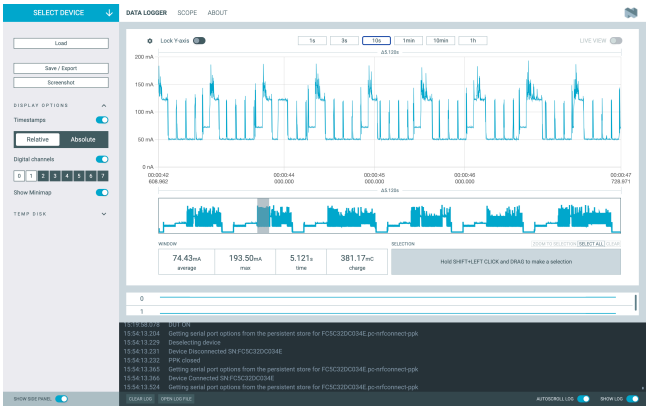

Figure 6: Ultrasonic + WiFi Mode

- Ultrasonic + WiFi + Firebasw Mode: average of 77.08 mA during the five seconds. $time = 500mAh \div 77.08 \approx 6.49h$

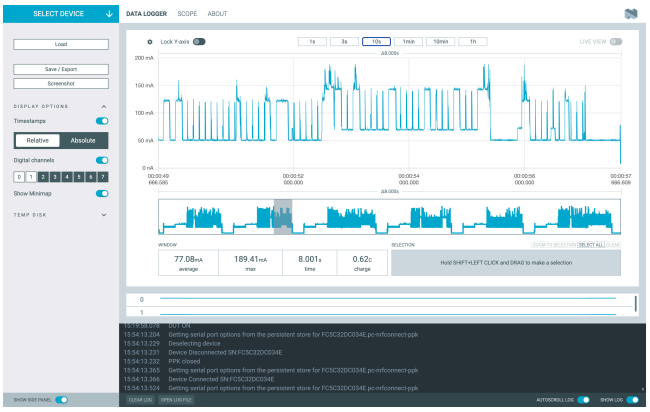

Figure 7: Ultrasonic + WiFi + Firebase Mode

Rewrite your code and measure the power consumption in different data transmitting rates. Take screenshots and do the calculations as above. Plot a figure to demonstrate the correlation between the data transferring rate and power consumption.

2.1. Different Transmission Rate

By checking the data uploaded to Firebase, we can confirm that the ESP32 functions as we want, transmitting data at a designated rate. As the figure shows, the timestamp is the current time in seconds, the transmission rate is in the unit of milliseconds, the last upload millis is the upload time in milliseconds.

In power consumption, no obvious drop can be seen. However, we can observe spikes in different densities according to different transmission rates; when sending data to Firebase, the power consumed is obviously higher.

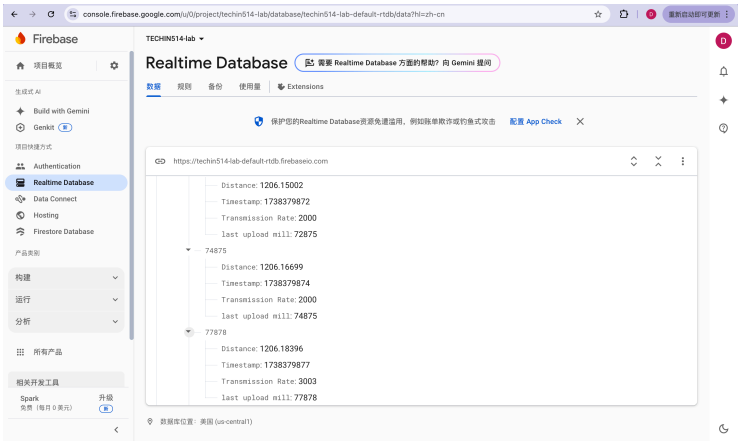

Figure 8: Firebase

Overall power consumption is shown below.

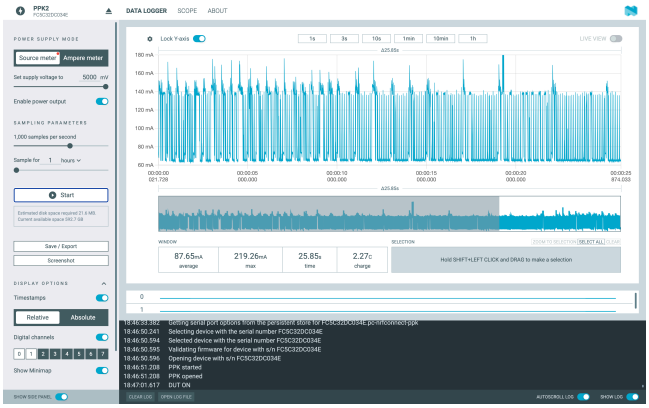

Figure 9: Overall View of Different Transmission Rates

The transmission rate is 2Hz:

Average current: 92.22mA, Time: $500mAh \div 92.22 \approx 5.42h$ The transmission

Figure 10: 2Hz

rate is 1Hz:

Average current: 91.34mA, Time: $500mAh \div 91.34 \approx 5.47h$

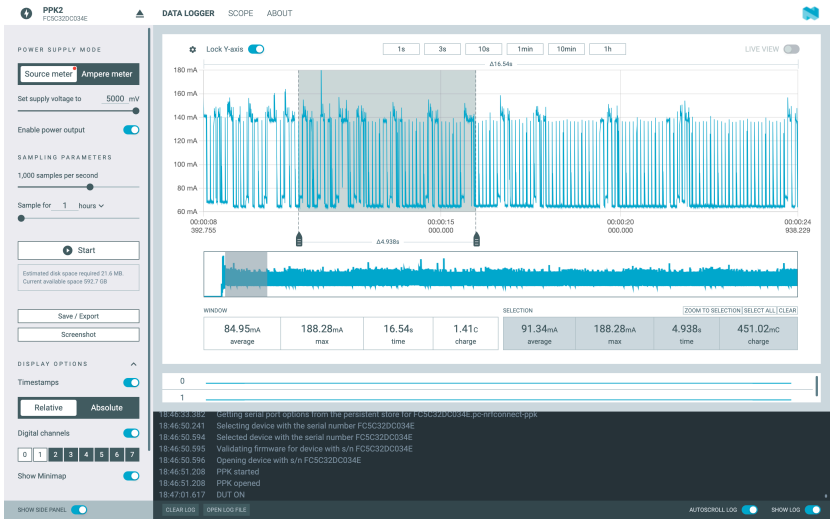

Figure 11: 1Hz

The transmission rate is 0.5Hz:

Average current: 77.87mA, Time: $500mAh \div 77.87 \approx 6.42h$ The transmission

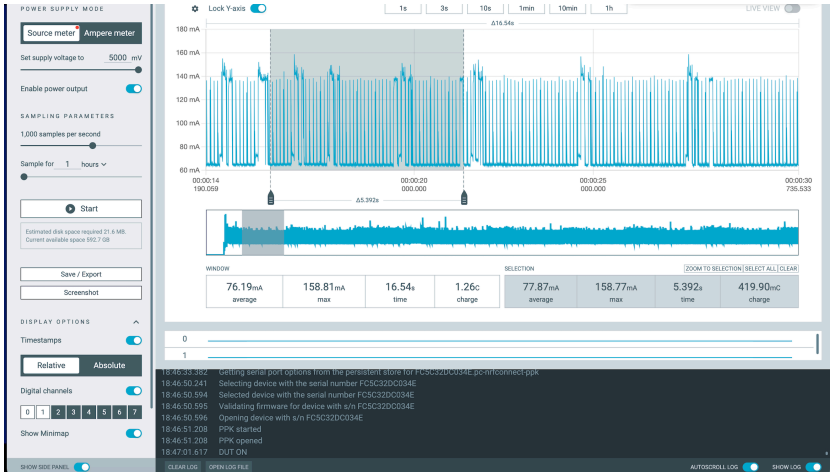

Figure 12: 0.5Hz

rate is 0.33Hz:

Average current: 76.19mA, Time: $500mAh \div 76.19 \approx 6.56h$

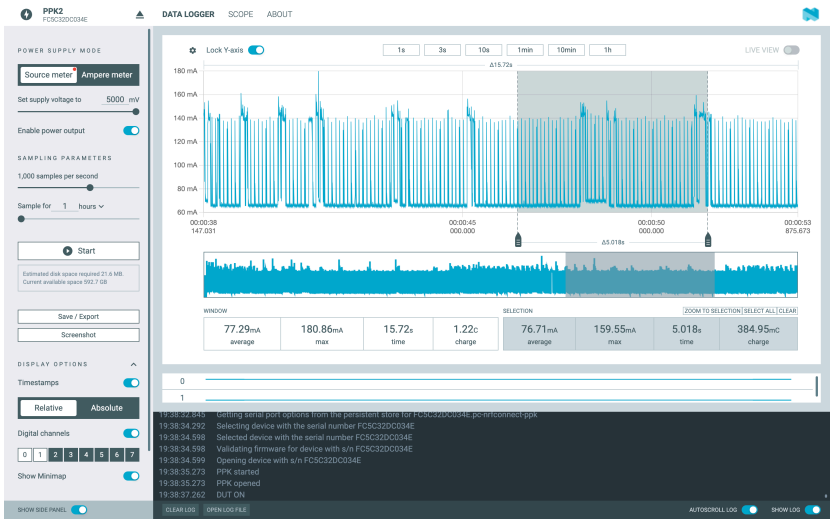

Figure 13: 0.33Hz

The transmission rate is 0.25Hz:

Average current: 75.28mA, Time: $500mAh \div 75.28 \approx 6.64h$

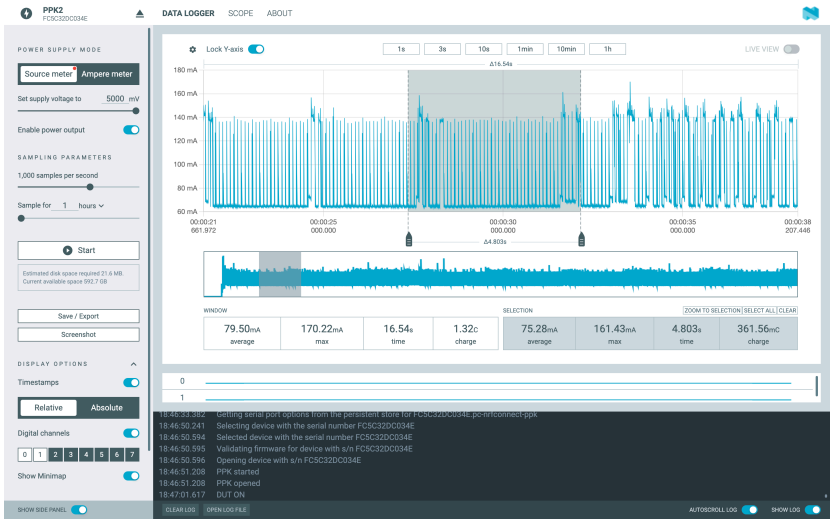

Figure 14: 0.25Hz

Correlation:

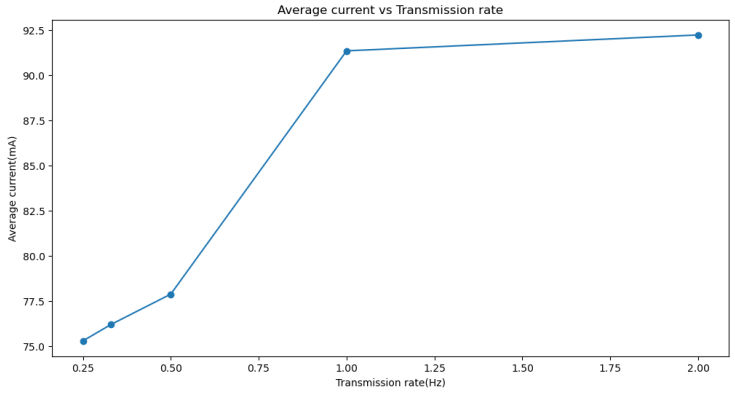

Figure 15: Correlation Between

3 Create your own power-saving strategy

The basic strategy is to put ESP32 to sleep when nobody's walking by. Detailed implementation is let the sensor detect every 8 seconds, if the distance

less or equal than 50 cm, then connects to WiFi and upload the data to Firebase. If no people is detected (all data higher than 50 cm) for 16 seconds, then sleep for 65 seconds. The overview of the power consumption when to-

Figure 16: Strategy Overview

tally idle, which means no detection after 20 seconds and sleep for 50 seconds:

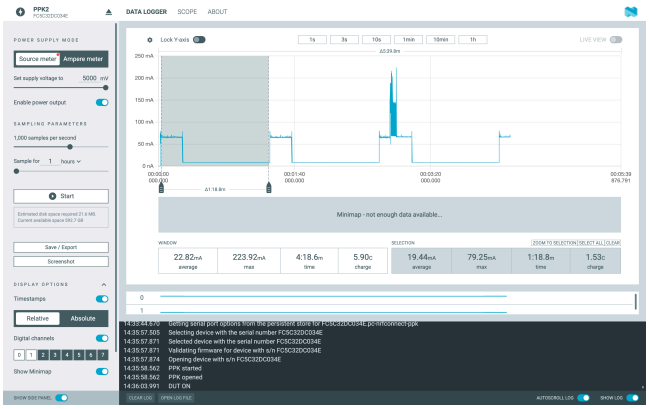

Figure 17: Idle Loop Consumption

The consumption for an idle loop is 19.44 mA. When data is transmitted, we can see spikes which is as below. With selected, we can see the average power consumption for transmitting is about 83 mA.

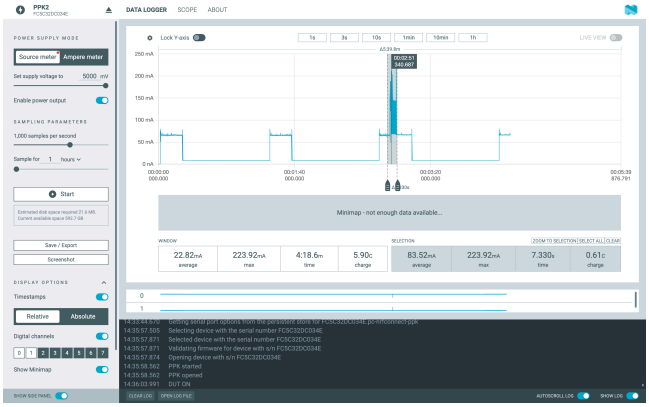

Figure 18: Caption

If we use a 500 mAh battery, and it is full of idle loops, it can last $500 \div 19.44 \approx 25.72h$. Which means it allows the device to run detection more than 24 hours, enabling sending data to Firebase for a few times.