**HW7 Report**

**IOT 11조**

RSSI is a measure of the power present in a received wireless signal.

It is mainly expressed as a negative dBm value, and the closer it is to 0 dBm, the stronger the signal.

N is the path loss index (varies from 2 in free space to 4 in indoor environment)텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

(Usually -30 is assumed to be a strong signal, and -90 is assumed to be a weak signal)

**Actual RSSI formula**

RSSI=−10nlog10(d)+α

**a**-> **txPowerAt**(tx-> transmitter, Rx->receiver)

**n-> PATH\_LOSS\_EXPONENT**

**Setting formula**

**distance = 10 ^ ((txPower - RSSI) / (10 \* n))**

**->float rawDistance = calculateDistance(rssi, txPowerAt1m);**

**float calibratedDistance = rawDistance \* CALIBRATION\_FACTOR;**

1.int **txPowerAt1m** = -59; (Assuming the default RSSI value at 1m is 59)

2.#define **PATH\_LOSS\_EXPONENT** 2.0(The higher n = 2~4 is set, the better it is for handling spaces with many obstacles. Currently, n = 2 is set to free space.)

3. #define **CALIBRATION\_FACTOR** 0.4(0.4 on the assumption that 2.5m will come out as rawDistance when measuring 1m)

The situation assumed in the code was created in a completely basic form.

Free space without obstacles, txPowerAt is -59 based on 1m

Result

**average**

텍스트, 스크린샷, 도표, 그래프이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

텍스트, 라인, 도표, 그래프이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.라인, 텍스트, 도표, 그래프이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.텍스트, 라인, 도표, 그래프이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.라인, 텍스트, 도표, 그래프이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

라인, 도표, 그래프, 텍스트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

**Result anlayzation**

error detection

If Rssi==0 or rssi>=txPower, -1.0 is returned.

Determined to be an error case

if (rssi == 0 || rssi >= txPower) {

    return -1.0; // Error cases

}

rssi>=txPower is set when txPower=-59(1m) is standard.

So measurements at 0.5m are not expected.

However, in the actual measurement, the RSSI at 0.5M was measured as rssi<-59 due to interference.

**0.5M**

In the current code, txPower=-59 is set to 1m, but in actual measurements, the standard for 0.5M is determined to be -59dBm.

The reason for this is believed to be due to severe interference from the surroundings. Based on this judgment,

I think it is better to set the N value (PATH\_LOSS\_EXPONENT) higher the closer the environment is.

In the current case, good results were seen in the 0.5M judgment.

**1M**

However, in the calculation, in the case of 1M, it was overfitted compared to the original value.

When measuring 1M, there were many obstacles around, such as laptops and people, and there was a lot of path loss, which resulted in a lot of noise, so it is believed that the actual measured value was high.

**2M**

In the case of 2M, there is a bouncing value, but it was measured in an environment with few obstacles, and compared to 1M, the noise is relatively less, so it is thought that an appropriate value was obtained, but the bouncing value still exists.

**3M**

In the case of 3M, it was measured in a space with few obstacles,

but it was judged to have a lot of variation because it received interference from other 2.4GHz devices (router and other BLE devices) and PATH\_LOSS\_EXPONENT was set to 2.0 (free space).

**4M**

In the case of 4M, there were often values ​​that jumped above 4M, but many values ​​smaller than the original distance were observed.

PATH\_LOSS\_EXPONENT was processed in a space without obstacles, so it was judged to be no problem.

However, due to the nature of BLE, 4M was measured at a long distance, so the signal was weak (about -90dBm), and noise was involved during transmission, so it was judged to be a short distance.

**Improvements**

After receiving a larger amount of RSSI, calculate the Mean and change to sending the value.

Depending on the incoming RSSI, it is expected that better performance will be achieved by changing the N value (PATH\_LOSS\_EXPONENT) by increasing it when it is close and decreasing it when it is far away. (For locations with few obstacles)