

2022년 정보컴퓨터공학부 전기 졸업과제 중간 보고서

주제 : **UWB/BLE** 기반 **3D** 실내측위 및 동기화

분과 : **C**(네트워크/시스템)

팀이름 : 버퍼 리스트
팀원 : **201524498** 안현기
 201624458 김춘수
 201924553 이현진
지도교수 : 김원석
제출일 : **2022년 08월 01일**



부산대학교
PUSAN NATIONAL UNIVERSITY

목 차

목 차	2
1. 요구조건 및 제약 사항 분석에 대한 수정사항	3
1.1 요구조건	3
1.2 제약 사항 분석에 대한 수정사항	3
1.2.1 기존 제약사항에 대한 수정사항	3
1.2.2 추가 제약사항 및 대책	4
2. 설계 상세화 및 변경 내역	4
3. 갱신된 과제 추진 계획	5
4. 구성원별 진척도	6
5. 보고 시점까지의 과제 수행 내용 및 중간 결과	6
5.1 Beacon-Scanner 모듈 간 BLE 통신 구현 및 RSSI 측정	6
5.2 3D 위치좌표 계산을 위한 사변측량(Quadrilateration)	9
6. 참고문헌	11

1. 요구조건 및 제약 사항 분석에 대한 수정사항

1.1 요구조건

- UWB 또는 BLE를 기반으로 하여 실내측위 시스템을 개발하도록 한다.
- ToA 또는 TDoA를 활용한 위치 측정 모듈을 개발 한다.
- 여러 대의 UWB/BLE Anchor를 배치하여 3D 위치 측정 및 오차를 제어한다.

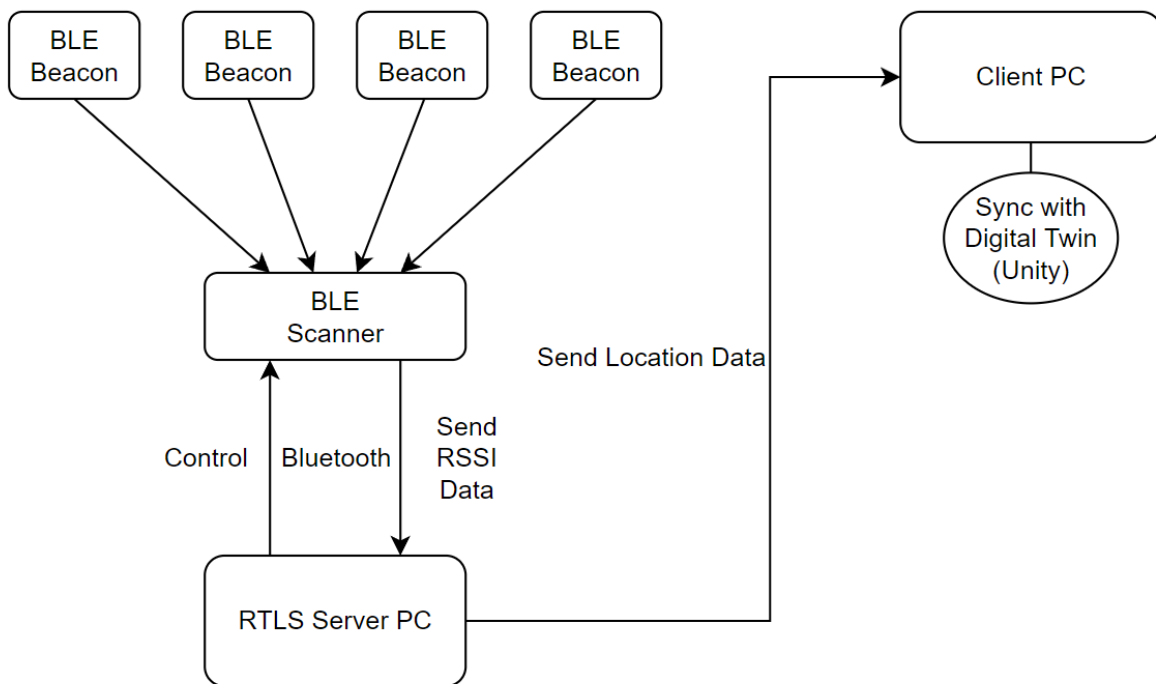


그림 1. 기존 RTLS 시스템 구성도

1.2 제약 사항 분석에 대한 수정사항

1.2.1 기존 제약사항에 대한 수정사항

- Beacon 및 Scanner로 활용할 모듈 변경
 - Arduino Nano 33 BLE 모듈로 실험한 결과 모듈간 통신거리 및 송수신감도가 RTLS를 구현하기에는 적합하지 못했음
 - Bluetooth 5가 이론상으로는 240m 정도까지 신호 전달이 가능하지만, 실제 모듈로 실험한 결과 3m 정도 떨어지면 통신이 불안정하였음
 - 이를 해결하기 위해 송신 출력을 높이거나, 수신 감도를 높이기 위한 자료 조사 및 실험을 수행하였으나 해결방안을 찾지 못함
 - 따라서 Adafruit itsybitsy nrf52840 모듈로 변경하여 과제를 진행할 예정. Adafruit itsybitsy nrf52840 모듈은 Arduino Nano 33 BLE 모듈과 마찬가지로 BLE 5.0 통신을 지원하며, 기존의 모듈에서 발생했던 통신 거리 문제가

발생했다는 사례를 찾을 수 없었고 오히려 더 먼 거리에서 통신이 가능함.
가격 또한 35,700원으로 기존과 큰 차이가 없어 해당 모듈을 선택함

1.2.2 추가 제약사항 및 대책

- 측정된 **RSSI**의 값을 바탕으로 거리를 계산할 때 한계점이 존재
 - **Anchor**에서 수신된 신호의 세기가 여러 장애물에 의해 약해졌을 경우에 이를 감별하기가 어려움
 - 따라서 **LOS** 환경에서 **NLOS** 환경으로 변화하는 등의 이유로 **RSSI** 값이 급격하게 변화할 시 거리 계산 과정에서 이를 보정할 예정
- **3D** 위치좌표 계산을 위한 삼변측량 구현의 어려움
 - 삼변측량은 **2차원** 좌표계에서 기준이 되는 **3개**의 모듈(**Anchor**)로부터 측정하고자 하는 모듈 사이의 거리를 측정하여 모듈의 위치를 특정하는 방법. 하지만 본 과제에서는 **3차원** 좌표계에서 모듈의 위치를 특정해야하므로 **4개**의 **Anchor**를 필요로 하며 오차에 더욱 민감해지므로 해를 구하기 힘들어짐

2. 설계 상세화 및 변경 내역

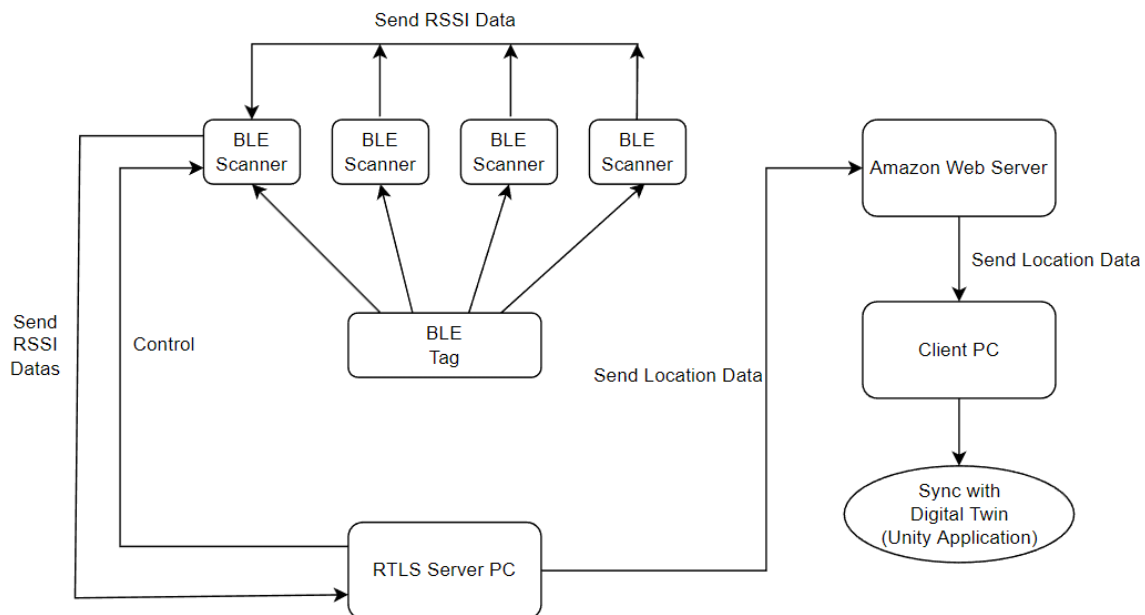


그림 2. 수정된 RTLS 시스템 구성도

- 설치된 **Anchor**에서 신호를 송신하지 않고, 상대적으로 저전력에서 동작하는 **Tag**에서 송신한 후 **Anchor**가 신호를 스캔하여 **RSSI**를 측정하도록 변경
- **Anchor** 중 하나는 일종의 **edge node**가 되어 동시간에 다른 **Anchor**에서 측정된 **RSSI** 값까지 모아 **Server PC**로 전송

- Server PC는 수신받은 RSSI 값을 사변측량을 이용해 위치 데이터로 변환
- Server PC와 Client PC가 직접 통신하는 대신 Server PC가 Cloud Server(AWS)에 위치 정보를 전송한 후 Client가 AWS에서 위치 정보를 받아오는 방법으로 변경

3. 갱신된 과제 추진 계획

	6월					7월				8월					9월	
주	1	2	3	4	5	1	2	3	4	1	2	3	4	5	1	2
안현기		RSSI 측정 및 거리 계산 구현														
										오차 제어 필터 개발						
										유니티 프로그램 개발						
김춘수	Beacon-Scanner간 통신 구현															
						과제에 적합한 모듈 재조사				edge node 구현						
										AWS 웹 서버 개발						
이현진	Beacon-Scanner간 통신 구현															
				좌표 계산 알고리즘 구현						유니티 프로그램 개발						
공통			기 말 고 사					중간 보고서 작성							최종 보고서 작성	

4. 구성원별 진척도

이름	진척도
안현기	<ul style="list-style-type: none"> Beacon-Scanner 간 RSSI 측정 프로그램 개발 및 거리 계산
김춘수	<ul style="list-style-type: none"> Beacon-Scanner 간 통신 구현 과제에 적합한 모듈 조사
이현진	<ul style="list-style-type: none"> 좌표 계산 알고리즘 구현 완료

5. 보고 시점까지의 과제 수행 내용 및 중간 결과

5.1 Beacon-Scanner 모듈 간 BLE 통신 구현 및 RSSI 측정

```

/*
  Scan

  The circuit:
  - Arduino MKR WiFi 1010, Arduino Uno WiFi Rev2 board, Arduino Nano 33
  IoT,
  Arduino Nano 33 BLE, or Arduino Nano 33 BLE Sense board.
*/

#include <ArduinoBLE.h>

long long setupTime = millis();

void setup() {
  Serial.begin(9600);
  while (!Serial);

  // begin initialization
  if (!BLE.begin()) {
    Serial.println("starting Bluetooth(r) Low Energy module failed!");

    while (1);
  }

  Serial.println("Bluetooth(r) Low Energy Central scan");
  BLE.setLocalName("RSSI Scanner");
  // start scanning for peripheral
  BLE.setEventHandler(BLEDiscovered, discovered);
  BLE.scanForName("RSSI TEST");
}

```

```
void logDevice(BLEDevice peripheral) {
    // print the local name and rssi value, if present
    if (peripheral.hasLocalName() && peripheral.localName() == "RSSI
TEST") {
        Serial.print("Local Name: ");
        Serial.println(peripheral.localName());
        Serial.println(peripheral.rssi());
        Serial.println();
    }
}

void discovered(BLEDevice peripheral) {
    BLE.stopScan();
    logDevice(peripheral);
    delay(250); // ms
    BLE.scanForName("RSSI TEST");
}

void loop() {
    BLE.poll(1000);
}
```

```
/*
    Broadcast

    The circuit:
    - Arduino MKR WiFi 1010, Arduino Uno WiFi Rev2 board, Arduino Nano 33
IoT,
    Arduino Nano 33 BLE, or Arduino Nano 33 BLE Sense board.
*/

#include <ArduinoBLE.h>

void setup() {
    if (!BLE.begin()) {
        while (1);
    }

    BLE.setLocalName("RSSI TEST");
    // start advertising
    BLE.advertise();
}
```

```
void loop() {
  // check if a central has been discovered
  BLEDevice central = BLE.central();
}
```

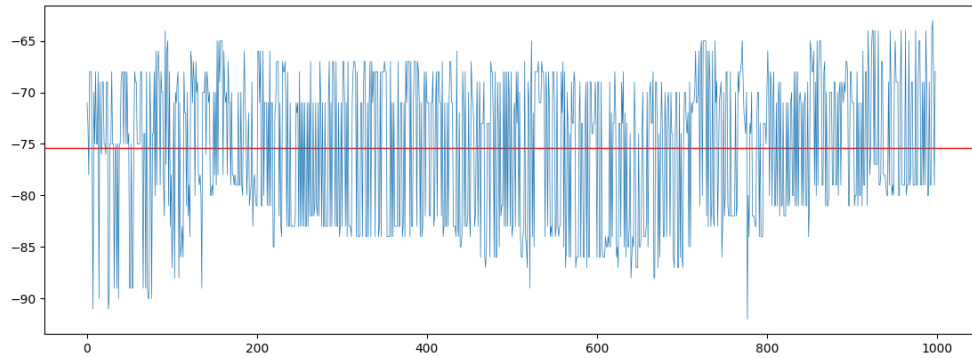


그림 3. 1m 거리에서의 Scanner-Beacon간 BLE 통신시 RSSI 값 측정 데이터

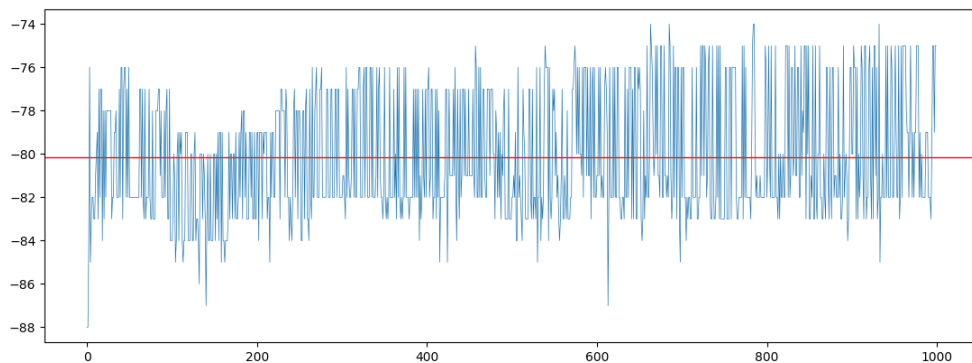


그림 4. 2m 거리에서의 Scanner-Beacon간 BLE 통신시 RSSI 값 측정 데이터

그림 3와 그림 4은 각각 LOS가 보장된 1, 2m 거리에서 Arduino nano 33 ble 모듈간 BLE 통신시 RSSI 값을 측정하여 그 변화를 시간순으로 나타낸 것이다. 각각 250ms 간격으로 1000회 측정하였으며 RSSI 값이 1m 거리에서는 평균 -73.876, 2m 거리에서는 평균 -80.136으로 나타났다. 평균값은 각 그림에서 붉은 실선으로 나타내었다. 오차가 지속적으로 발생하여 측정값이 어느 한 값에 수렴하지 않고 계속 진동하는 모습을 확인할 수 있다.

측정된 RSSI 값을 이용하여 두 모듈간 거리(단위: m)를 역계산하는 수식은 아래와 같다[1][2].

$$D = 10^{(A-RSSI)/10n}$$

여기서 A 는 기준 거리(1 m)에서 태그로부터 수신된 신호 세기를, n 은 신호 감쇠와 관련된 상수를 나타낸다. n 은 자유 공간에서 약 2의 값을 가지며 장애물이 많을 수록 커진다.

측정된 RSSI 값의 평균값을 사용하여 아래와 같은 결과를 얻을 수 있었다.

$$\begin{cases} A = -73.876db \\ n = 2.0797 \end{cases}$$

하지만 Arduino Nano 33 BLE 모듈의 송신세기가 매우 낮게 측정되어 신뢰성이 낮고, 실제로 3m 이상의 거리에서는 모듈간 통신이 불안정하여 오차와 관계없이 해당 모듈을 RTLS 시스템에 사용하기에는 무리가 있다고 판단된다.

Signal strength	Meaning
>-50 dB	very good
-50 ... -70 dB	good
-70 ... -80 dB	sufficient, but less reserves; repeater could be considered
-80 ... -90 dB	no reserves; don't use repeaters anymore
<-90 dB	limit of proper signal reception

그림 5. RSSI 값에 따른 신뢰성

5.2 3D 위치좌표 계산을 위한 사변측량(Quadrilateration)

먼저, 삼변측량을 이용해 2D상에서 위치를 특정하는 방법은 아래와 같다.

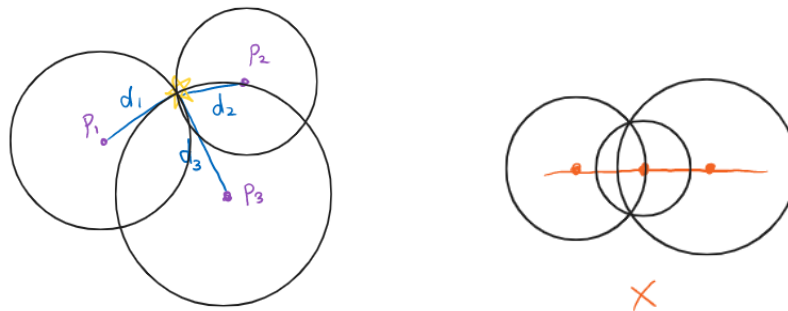


그림 6. 2D에서의 삼변측량을 이용한 좌표 계산

일단, 그림과 같이 Anchor들을 일직선으로 두지 않도록 한다.

$P_n(x_n, y_n)$ 까지의 거리가 d_n 일 때 $(x - x_n)^2 + (y - y_n)^2 = d_n^2$ 이다. ($n = 1, 2, 3$)

이 식들을 이용하여 2D 좌표를 계산할 수 있다.

$$\begin{cases} 2(x_2 - x_1)x + 2(y_2 - y_1)y = d_1^2 - d_2^2 + x_2^2 + y_2^2 - x_1^2 - y_1^2 \\ 2(x_3 - x_2)x + 2(y_3 - y_2)y = d_2^2 - d_3^2 + x_3^2 + y_3^2 - x_2^2 - y_2^2 \end{cases}$$

$P_1(1, 0)$, $P_2(0, 1)$, $P_3(-1, 1)$ 이고 구해야 할 좌표가 $(0, 0)$ 일 때 좌표 계산 예제 코드는 아래와 같다.

```
# 2D: fixed 3 points, target: (0, 0)
position_2d = np.array([[1, 0], [0, 1], [-1, 1]])
distance_2d = np.array([1, 1, 2]) # squared
A_2d = np.empty([2, 2])
b_2d = np.empty([2])
for i in range(2):
    x0 = position_2d[i][0]
    x1 = position_2d[i + 1][0]
    y0 = position_2d[i][1]
    y1 = position_2d[i + 1][1]
    A_2d[i][0] = 2 * (x1 - x0)
    A_2d[i][1] = 2 * (y1 - y0)
    b_2d[i] = distance_2d[i] - distance_2d[i + 1] - (x0 ** 2 + y0 ** 2)
+ (x1 ** 2 + y1 ** 2)
print(A_2d, b_2d)
location_2d = np.linalg.solve(A_2d, b_2d)
print(location_2d)

# visualization
xrange = np.arange(-2.5, 2.5, 0.02)
yrange = np.arange(-1.5, 2.5, 0.02)
x, y = np.meshgrid(xrange, yrange)
for i in range(3):
    eq = (x - position_2d[i][0]) ** 2 + (y - position_2d[i][1]) ** 2 -
distance_2d[i]
    plt.contour(x, y, eq, levels=[0])
plt.axis('equal')
plt.scatter(position_2d[:, 0], position_2d[:, 1])
plt.scatter(location_2d[0], location_2d[1])
plt.show()
```

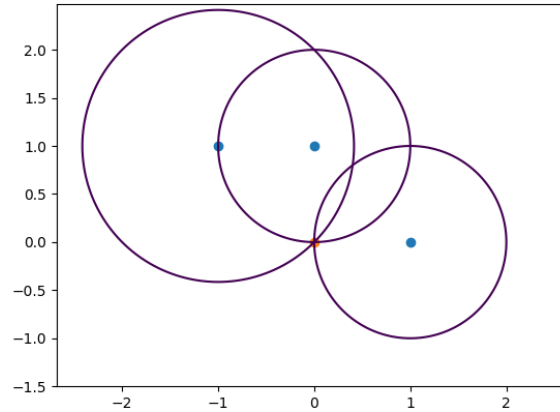


그림 7. python으로 직접 구현한 삼변측량 구현 결과 이미지

위의 과정을 아두이노 **BasicLinearAlgebra** 라이브러리로 계산할 수 있다.

3D 상에서 좌표를 계산하기 위해서는 해당 지점에서 고정된 네 점까지의 거리를 알아야 한다.

Intersection	Equivalency	Result
Intersection of 2 spheres	Circle	Circle
Intersection of 3 spheres	Circle \cap Sphere	Two points
Intersection of 4 spheres	Two points \cap Sphere	One point

그림 8. 3차원 공간상에서 구체가 접할 때 개수에 따른 결과

2D 상에서 삼변측량을 할 때의 연립방정식에 \approx 좌표와 식이 하나 추가된다.

$$\begin{cases} 2(x_2 - x_1)x + 2(y_2 - y_1)y + 2(z_2 - z_1)z = d_1^2 - d_2^2 + x_2^2 + y_2^2 + z_2^2 - x_1^2 - y_1^2 - z_1^2 \\ 2(x_3 - x_2)x + 2(y_3 - y_2)y + 2(z_3 - z_2)z = d_2^2 - d_3^2 + x_3^2 + y_3^2 + z_3^2 - x_2^2 - y_2^2 - z_2^2 \\ 2(x_4 - x_3)x + 2(y_4 - y_3)y + 2(z_4 - z_3)z = d_3^2 - d_4^2 + x_4^2 + y_4^2 + z_4^2 - x_3^2 - y_3^2 - z_3^2 \end{cases}$$

6. 참고문헌

- [1] Fengjun Shang, Wen Su, Qian Wang, 「A Location Estimation Algorithm Based on RSSI Vector Similarity Degree」, International Journal of Distributed Sensor Networks, Volume 2014, Article ID 371350, 22 pages, 2014. 8
- [2] Onkar Pathak, Pratik Palaskar, Rajesh Palkar, Mayur Tawari, 「Wi-Fi Indoor Positioning System Based on RSSI Measurements from Wi-Fi Access Points –A Tri-lateration Approach」, International Journal of Scientific & Engineering Research, Volume 5, Issue 4, 2014. 4