
비콘 송수신을 이용한 통신 연구 보고서

2021.05.10

작성자 조은희
cho108@krri.re.kr

목 차

[illegible]

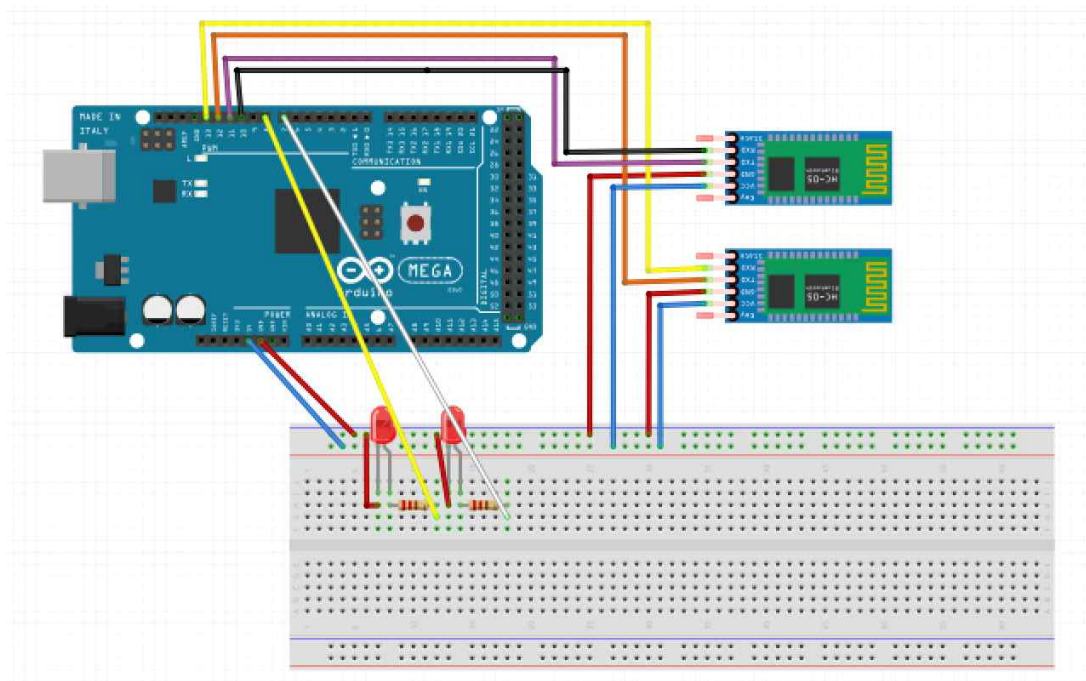
1. 실험 보고

1-1. 실험환경

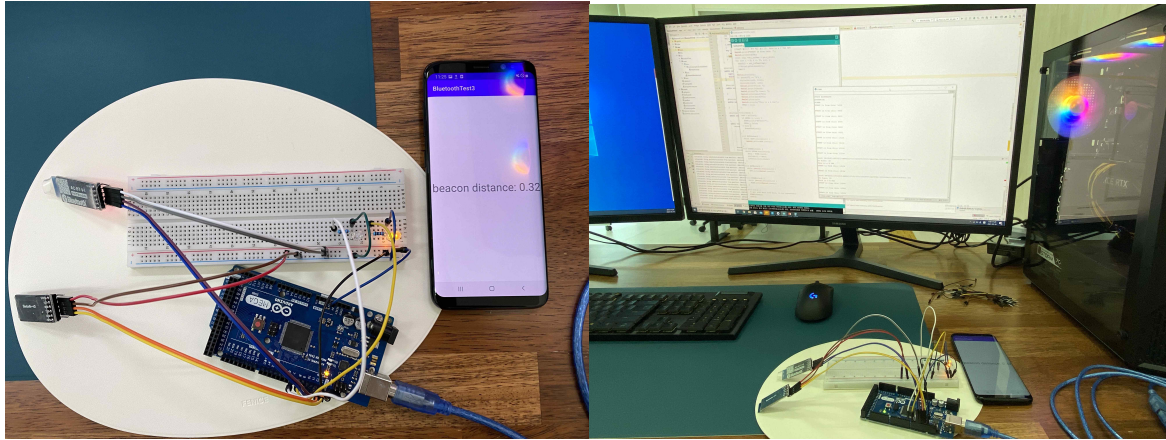
분류	종류	비고
블루투스	HM-10	2개
아두이노	아두이노 2560 MEGA	
스마트폰	삼성 갤럭시 S8	안드로이드 버전: 9.0

[표1 - 실험환경]

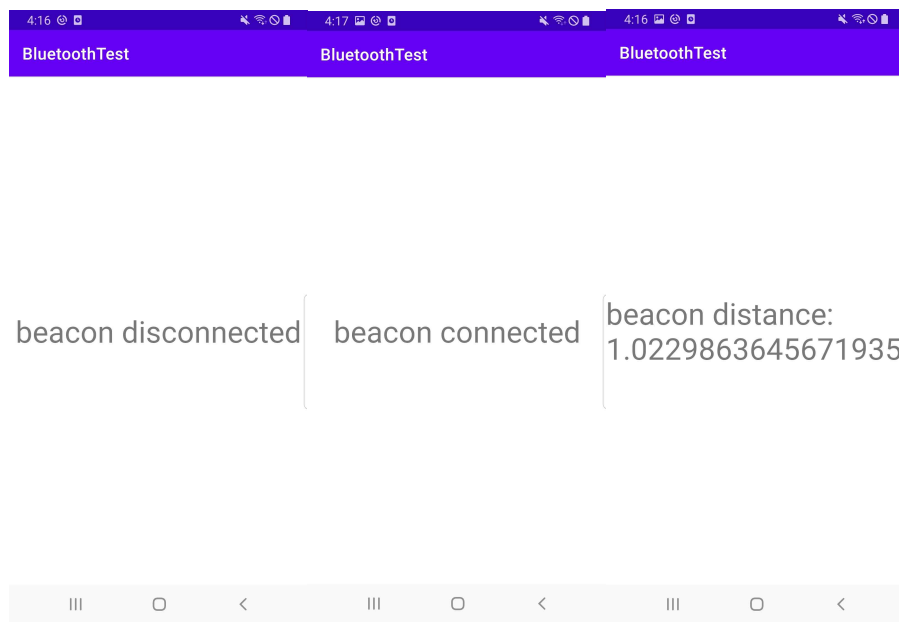
아두이노에서 HM-10 모듈 두 개를 사용하여 회로를 구성하였으며, 비콘의 송수신을 확인하기 위한 안드로이드 어플리케이션을 간단히 제작하였다. 블루투스 모듈은 각각 송신 작업과 스캔 작업으로 나뉘어 송수신 모드 변경 없이 진행하였다.



[그림 1 - HM-10 2개를 사용하여 구성한 회로]



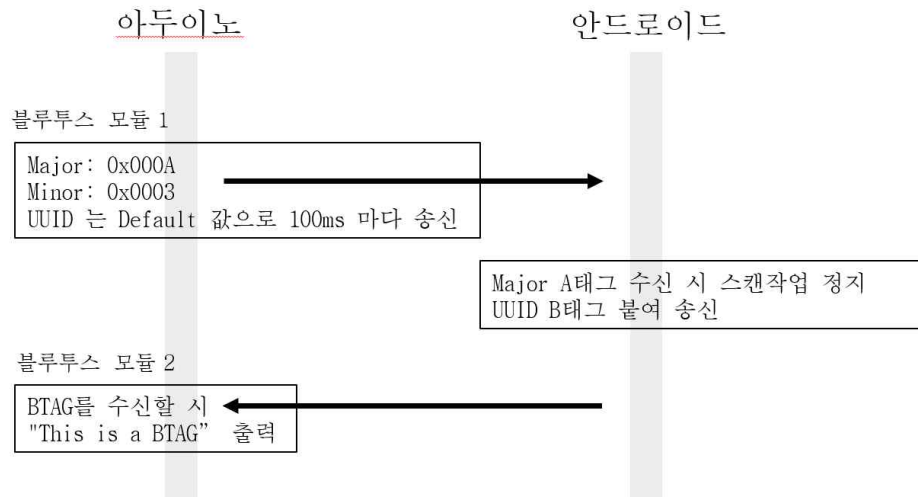
[그림 2 - 실제 구성한 아두이노, 안드로이드]



[그림 3 - 비콘 통신 안드로이드 테스트 (비콘이 인식되지 않을 때/ 비콘이 인식 될 때 /인식된 비콘의 Major 값이 A 일 때)]

1-2. 실험과정

비콘 통신



[그림 4 - 아두이노와 안드로이드간 통신 구성도]

아두이노에 부착되어 있는 블루투스 모듈 두 개는 각각 송신 기능과 수신기능을 맡는다. 따라서 더 이상의 송수신 모드 변경으로 인한 지연시간은 없다.

안드로이드는 지속적으로 비콘을 스캔한다. 안드로이드는 major 값에 A 태그가 있는 비콘을 수신하게 되면 해당 비콘과의 거리를 출력하고 그 즉시 스캔 작업을 정지한다. 이후, UUID에 B 태그를 붙여 비콘 신호를 송신한다. 아두이노의 블루투스 모듈2는 B 태그 비콘을 수신하면 “This is a BTAG”를 출력한다.

```

void advertisingA() {
  Serial.println("ADVERTISE");
  digitalWrite(7, HIGH);
  ADV.write("AT+RENEW"); //공장초기화
  delay(delayTime);
  ADV.write("AT+MAJ0x000A"); //비콘의 major 설정
  delay(delayTime);
  ADV.write("AT+MIN0x0003"); //비콘의 minor 설정
  delay(delayTime);
  ADV.write("AT+ADVI0"); //신호송출주기 100ms 설정
  delay(delayTime);
  ADV.write("AT+ADTY2"); //advertising type: allow advertising and scan response
  delay(delayTime);
  ADV.write("AT+IBEA1"); // iBeacon 활성화
  delay(delayTime);
  ADV.write("AT+DELO1"); // allow to broadcast and scanning
  delay(delayTime);
  ADV.write("AT+RESET"); //리셋
  delay(delayTime);
}
  
```

[그림 5 - 아두이노 비콘 송신 코드]

```

if(distance>0) {
    if (MAJOR == 0x000A) {
        handler.sendMessage( what: 2);
        try {
            beaconManager.stopMonitoringBeaconsInRegion(region);
        } catch (RemoteException e) {
            e.printStackTrace();
        }
        try {
            beaconManager.stopRangingBeaconsInRegion(region);
        } catch (RemoteException e) {
            e.printStackTrace();
        }
        Log.d(TAG2, msg: "*****STOP*****");
    } else {
        Log.d(TAG2, msg: "THIS IS REAL VALUE: " + MINOR);
        handler.sendMessage( what: 0);
    }
}

```

[그림 6 - 안드로이드 비콘 스캔 중단 코드]

[그림 6]는 수신한 비콘의 major 값이 A 태그 일 때, 스캔하는 작업을 중지하는 안드로이드 코드이다.

```

bool is_Start(int tmp, String po) {
    //TAG가 붙었는지 계속 확인 붙었으면, This is a B TAG 출력
    Serial.print("START is from this: ");
    Serial.println(now);
    const char *str_toChar = po.c_str();
    for (int i = 0; i <= 77; i++) {
        uuid[i] = str_toChar[tmp];
        //Serial.print(uuid[i]);
        tmp++;
    }

    Serial.println();
    if (uuid[17] == 'B') {
        digitalWrite(8, HIGH);
        digitalWrite(7, LOW);
        Serial.print("uuid: ");
        Serial.println(uuid);
        Serial.print("Tx Power: ");
        Serial.print(uuid[76]);
        Serial.print(uuid[77]);
        Serial.println();
        Serial.println("This is a B TAG");
        BTAG = true;
    }
}

```

[그림 7 - 아두이노 B 태그 수신확인 코드]

1-3. 실험 결과

```
START BLUETOOTH
ADVERTISE
SCANB
START is from this: 9455
START is from this: 9455
START is from this: 9455
START is from this: 9455
START is from this: 9455
START is from this: 9455
START is from this: 13106
START is from this: 13106
START is from this: 13106
START is from this: 13106
uuid: OK+DISC:18010215:BB00C127101122223333F4911BA9FFA6:0005000AC5:55AB990E476C:-050
Tx Power: 50
This is a B TAG
START is from this: 13106
START is from this: 13106
```

[그림 8 - 아두이노 스캔 주기 / B 태그를 발견했을 때]

안드로이드와 달리 아두이노는 스캔을 지속적으로 하지 못한다. 따라서 스캔을 하기 위해서는 “AT+DISI?” 라는 스캔시작 AT 명령어를 아두이노 블루투스 모듈2에 반복해서 호출하여야 한다.

[그림 8]은 “AT+DISI?” 명령어가 호출되어 그 스캔응답을 받을 때의 시간을 출력한 것이다. ms 단위로 출력되어 있으며, 다음 “AT+DISI?” 명령어가 실행되기까지의 시간 간격은 약 3.5초에서 4초 사이이다. 블루투스 모듈 1개로 송수신 모드를 변경하며 실험했던 지난 측정시간보다는 약 9초가량 감소된 것을 알 수 있다.

비콘 송신송출주기의 경우 100ms 로 매우 빠른 편이지만 비콘 신호 수신주기의 경우 비콘 수신용 블루투스 모듈을 사용하더라도 4초 내로 속도가 느린편에 속한다. 이를 해결하기 위해 아두이노의 스캔 모드를 없애고 안드로이드를 이용하여 서버에서 아두이노에 데이터를 전달할 계획이다.

참고문헌

- KS X 6924
- <https://people.ece.cornell.edu/land/courses/ece4760/PIC32/uart/HM10/DSD%20TECH%20HM-10%20datasheet.pdf> (For AT Command)