

# **hanabee**

## **Bluetooth Low Energy Beacon**

**Android SDK v 1.0.8**

**IoT Biz Team**  
**HANA Micron. Inc.**

### **Documentation for Android Development**

HANA Micron. Inc.  
B-903, Pangyo siliconpark, 613  
Sampyeong-dong, Bundang-gu, Seongnam-si  
Gyeonggi-do, South Korea  
+82) 31 698 9022  
[hanabee@hanamicron.co.kr](mailto:hanabee@hanamicron.co.kr)

This document is available at: [https://github.com/HanaBEE/HanaBEE\\_Android\\_SDK](https://github.com/HanaBEE/HanaBEE_Android_SDK)

2015. 09. 11  
HANA Micron. Inc. All rights reserved.

## Table of Contents

TABLE OF CONTENTS .....	2
<b>START WITH HANABEE BLE BEACON .....</b>	<b>3</b>
HANABEE 의 특징.....	3
<b>빠른 시작 .....</b>	<b>4</b>
샘플소스를 이용하여 곧바로 HANABEE 를 이용해보세요. (이클립스) .....	4
샘플소스를 ECLIPSE 에서 IMPORT 하기 .....	5
<b>HANABEE SDK API DETAILS .....</b>	<b>5</b>
HANABEE.....	8
<b>이벤트 가이드.....</b>	<b>16</b>
<b>TROUBLE SHOOTING &amp; FAQ.....</b>	<b>18</b>
HANABEE SDK 없이 HANABEE 비콘 개발을 할 수 있나요? .....	18
INITBLE() 에서 에러발생(1).....	18
INITBLE() 에서 에러 발생(2).....	19
HANABEE 비콘이 없는데 샘플소스를 실행시키고 싶습니다. ....	19
안드로이드 단말을 가상의 HANABEE 비콘으로 이용할 수 있나요? .....	19
POINTDOUBLE 클래스를 ANDROID INTENT 의 PARCELABLE 로 전달할 수 있나요? .....	20
POSITIONNING 에서 결과 값이 계속 (0,0) 으로 나옵니다. ....	20
SETPOSITIONNODEINFO 로 좌표 입력 시 음수를 입력해도 되나요?.....	21
NOCLASSDEFFOUNDERROR 가 발생합니다. 어떻게 해결하나요? .....	21
HANABEE 비콘을 주문하고 싶습니다. 쇼핑몰 같은 것이 있나요?.....	22
HANABEE SDK 가 잘 동작하지 않습니다. 뭐가 문제인가요? .....	23
전파인증, KC 인증을 받으셨나요?.....	23
다른 질문이 있으면 어디로 물어보면 되나요?.....	23
<b>LICENSE .....</b>	<b>24</b>

## Start with hanabee BLE Beacon

### hanabee 의 특징

hanabee 는:

블루투스 4.0, Low Energy (BLE) 를 이용한 비콘시스템으로, 아래와 같은 특징을 가지고 있습니다.

- 보안강화(Anti-Spoofing): 128 bit 암호화
  - 배터리 관리(Battery Management): 추가 접속 없이 배터리 정보확인.
  - 근접감지(Proximity Sensing): 가능
  - 삼각 측위를 통한 위치계산(Trilateration): 가능
  - 배터리 타입:
    - hanabee Slim: 교체 가능한 AA 건전지 2 개 (2800 mAh)
    - hanabee Mini: 교체 가능한 CR2450 코인 배터리 1 개 (800 mAh)
- 또는
- USB 전원을 이용한 상시 전원공급

SDK 이용을 위해 필수적인 사항

- Android 4.3 또는 이후 버전 OS (Bluetooth le 지원)
- BLE 를 지원하는 Android device(일부 기기는 OS 가 Android 4.3 이상이지만 하드웨어에서 지원하지 못하는 경우가 있을 수 있습니다.)

## 빠른 시작

샘플소스를 이용하여 곧바로 hanabee 를 이용해보세요. (이클립스)

아래와 같은 방법으로 샘플소스를 다운로드 받을 수 있습니다.

1.깃허브 클라이언트를 인스톨 하세요.

<https://windows.github.com/> 는 윈도우 이용자용이며,

<https://mac.github.com/> 는 매킨토시 이용자 용입니다.

2.샘플소스 페이지로 이동하세요.

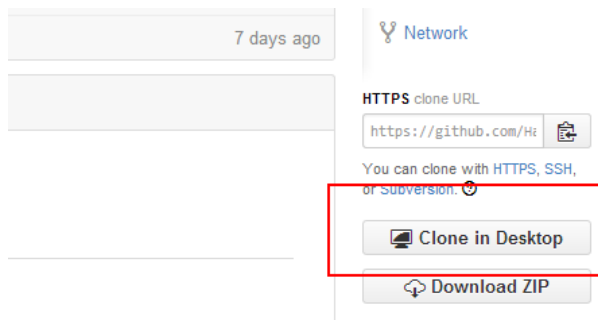
비콘 스캔 + 이벤트:

[https://github.com/HanaBEE/HanaBEE\\_Android\\_SDK](https://github.com/HanaBEE/HanaBEE_Android_SDK)

비콘 스캔 + 이벤트 + Indoor positioning (Trial version)

[https://github.com/HanaBEE/HanaBEE\\_Android\\_Indoor\\_SDK](https://github.com/HanaBEE/HanaBEE_Android_Indoor_SDK)

3.**Clone in Desktop** 을 누르시면 다운로드가 시작됩니다.



4.샘플소스는 깃허브 클라이언트에서 지정한 폴더로 다운로드 됩니다.

5.위 과정 대신, 직접 커맨드 라인에서 git clone 커맨드를 이용해서 다운로드 받아도 됩니다.

6.또는, 3 번 과정에서 Download ZIP 을 클릭하시면 Zip 파일로 전체 파일을 다운로드 받을 수 있습니다. 이 방법을 이용할 경우, Git 관련 프로그램을 설치할 필요가 없습니다.

## 샘플소스를 Eclipse 에서 import 하기

1. Project explorer – 우 클릭 – Import 선택
2. General – Existing Projects into Workspace 선택  
(안드로이드 선택 시 Import 되지 않는 환경이 있어 General 로 import 하는 것을 권장합니다)
3. 다운로드 받은 샘플소스 경로를 클릭 후 확인을 누릅니다.

## hanabee SDK API Details

hanabee 의 API 의 상세한 내용입니다.

**준비:** hanabee SDK 를 이용하려면 AndroidManifest.xml 에 다음 블루투스 권한이 반드시 설정되어야 합니다.

android.permission.BLUETOOTH  
android.permission.BLUETOOTH\_ADMIN

Sample code:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    .....
    .....

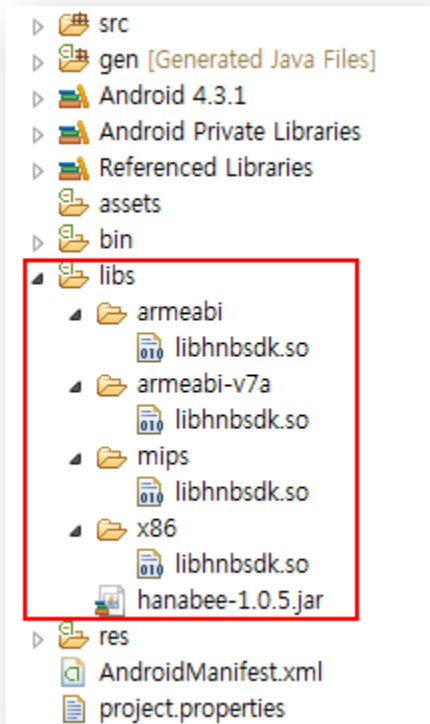
<!-- Add permissions below -->
    <uses-permission android:name="android.permission.BLUETOOTH" />
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

아래 서비스를 Application 내에 등록해 주시기 바랍니다.

```
<application
    .....
    <activity .....
    </activity>

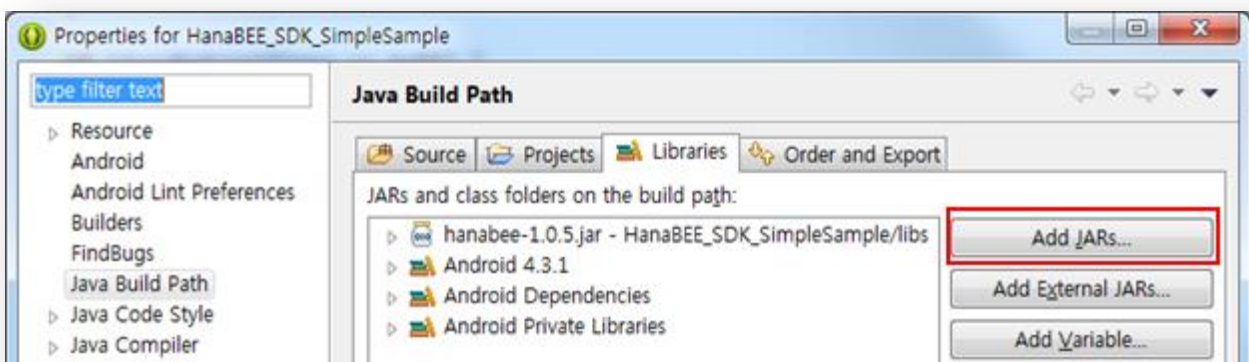
<!-- Add this service -->
    <service android:name="com.hanamicon.beacon.bluetooth.BLEScanService"
        android:exported="false" />
```

hanabee SDK 를 이용하여 새로운 프로젝트를 생성하시려면, 아래 파일이 반드시 필요합니다.



libs 내의 모든 파일. armeabi / armeabi-v7a / mips / x86 폴더는 대상 단말의 CPU 형태를 의미합니다. 만약 제작하시는 앱의 대상 단말이 특정 CPU 를 지원하지 않는다면, 해당 CPU 에 해당되는 폴더 및 하위 파일은 삭제하셔도 됩니다.

이후, 좌측 Project navigator 우클릭 – properties – Java build path – Add JARs 를 선택하여 hanabee-X.X.X.jar (X 는 버전번호)를 선택하시면 됩니다.



**주의:** hanabee 1.0.7 버전부터 Indoor positioning 기능이 있는 SDK와 그렇지 않은 SDK 두 종류로 분리되었습니다.

**hanabee SDK:**

[https://github.com/HanaBEE/HanaBEE\\_Android\\_SDK](https://github.com/HanaBEE/HanaBEE_Android_SDK)

비콘 스캔 기능과 이벤트 기능 탑재. 시간제한 없음.

**hanabee Indoor SDK (Trial):**

[https://github.com/HanaBEE/HanaBEE\\_Android\\_Indoor\\_SDK\\_trial](https://github.com/HanaBEE/HanaBEE_Android_Indoor_SDK_trial)

비콘 스캔 기능과 이벤트 기능 탑재. 추가적으로 Indoor positioning 기능이 탑재되어 있으며, 시간제한이 있음. (1.0.8 버전: 2015 년 12 월 31 일까지)

시간 제한이 없는 정식버전 이용을 원하시면 [hanabee@hanamicron.co.kr](mailto:hanabee@hanamicron.co.kr)으로 문의 주시기 바랍니다.

아래 매뉴얼에서 **[Indoor only]** 말머리가 있는 항목은 Indoor SDK에만 적용되는 내용입니다.

### Hanabee

hanabee 의 메인 클래스입니다

**초기화:** Hanabee 를 이용하기 위해서는 instance 를 얻어야 합니다. Hanabee 는 Singleton 으로, getInstance 를 통해 인스턴스를 생성해야 합니다.

getInstance(Context context)

**Method:**

**public static** Hanabee getInstance(Context context)

**Return type:**

Hanabee

**Description:**

Singleton Instance를 반환.

getInstance(Context context, boolean isForceBluetoothOn)

**Method:**

**public static** Hanabee getInstance(Context context, **boolean** isForceBluetoothOn)

**Return type:**

Hanabee

**Description:**

Singleton Instance를 반환. isForceBluetoothOn가 true이면 Bluetooth 팝업을 띄우지 않고 강제로 Bluetooth를 시작.

Sample code:

```
private Hanabee mHanabee;
// Turn on bluetooth by your own code
mHanabee = Hanabee.getInstance(this);

// or, force start bluetooth.
// mHanabee = Hanabee.getInstance(this, true);
```



**[Indoor only] 포지션 스캔 설정:** 4 개 이상의 비콘과 해당 비콘의 좌표를 미리 입력합니다. 이후 스캔 동작 시, 설정된 좌표에 해당되는 현재 위치를 얻을 수 있습니다.

**[Indoor only] setPositionNodeList(ArrayList<BeaconInfo> nodeList)**

**Method:**

**public void** setPositionNodeList(ArrayList<BeaconInfo> nodeList)

**Return type:**

void

**Description:**

Positioning 에 이용할 비콘리스트를 입력. 비콘의 개수는 4 개 이상이어야 하며, 각 비콘은 MAC 과 X,Y 좌표(미터 단위)를 가지고 있어야 함. startHanabee 실행 이전에 설정해야 함.

**[Indoor only] getPositionNodeList ()**

**Method:**

**public** ArrayList<BeaconInfo> getPositionNodeList()

**Return type:**

ArrayList<BeaconInfo>

**Description:**

이전에 setPositionNodeList 를 통해 입력했던 비콘 데이터를 반환

Sample code:

```
private Hanabee mHanabee;
mHanabee = Hanabee.getInstance(this);

// Position
ArrayList<BeaconInfo> positionNodeList = new ArrayList<BeaconInfo>();
positionNodeList.add(new BeaconInfo(0, 0, "90:59:AF:2A:AD:3F"));
positionNodeList.add(new BeaconInfo(0, 8, "90:59:AF:2A:94:68"));
positionNodeList.add(new BeaconInfo(8, 8, "90:59:AF:2A:C4:04"));
positionNodeList.add(new BeaconInfo(8, 0, "90:59:AF:2A:AD:4C"));
mHanabee.setPositionNodeList(positionNodeList);
```

Sample code description:

Position scan 을 하기 위해서는 사전에 4 개 이상의 비콘의 정보를 입력해 두어야 합니다. 이것은 Trilateration 계산을 위한 필수사항입니다. 이를 위해 `ArrayList<BeaconInfo>` 객체를 하나 생성하여, Beacon 의 목록을 만든 뒤, `setPositionNodeInfo` 를 통해 등록하면 됩니다.

등록되는 Beacon 정보는 X 좌표, Y 좌표, MAC Address 입니다.

이 과정 이후 `startHanabee` 를 호출할 수 있습니다. 결과값은 `onPosition(String floorID, double x, double y)` 콜백을 통해 받을 수 있으며, `floorID` 는 해당 좌표목록의 ID (현재는 첫 번째 입력된 비콘의 MAC Address) 이며, `x, y` 가 계산된 스마트폰의 좌표입니다. 이 `x, y` 좌표값을 이용하여 원하는 작업을 수행하시면 됩니다.

**근접 정보 설정:** 근접정보를 얻고싶은 비콘의 MAC Address, 비콘과의 거리(즉 반지름 `radius`, 단위: m), 오차 허용범위(`marginOfError`, 단위: m) 를 설정합니다. 이후 `HanabeeResultCallback` 콜백의 메소드인 `onProximity` 를 통해 설정한 비콘의 정보를 수신할 수 있습니다.

`setProximityList(ProximityList proximityList)`

**Method:**

```
public void setProximityList(ProximityList proximityList) {
```

**Return type:**

void

**Description:**

Proximity region 계산에 이용할 beacon 목록 입력. 이 목록에 있는 Beacon 에 대해, RSSI 와 거리 값을 얻을 수 있음.

`proximityList` 에는 `addProximityRegion` 를 통해 Beacon 의 Mac Address, 반지름(Radius, 단위: m), 오차 허용범위(`marginOfError`, 단위: m)를 입력해야 함.

### getProximityList ()

**Method:**

```
public ProximityList getProximityList()
```

**Return type:**

```
ProximityList
```

**Description:**

이전에 setProximityList 를 통해 입력했던 비콘 데이터를 반환

### Sample code:

```
private Hanabee mHanabee;  
mHanabee = Hanabee.getInstance(this);  
  
ProximityList proximityList = new ProximityList();  
proximityList.addProximityRegion("90:59:AF:2A:95:90", 50, 1);  
proximityList.addProximityRegion("78:C5:E5:6C:0D:F7", 50, 1);  
mHanabee.setProximityList(proximityList);
```

### 비콘 스캔 시작하기:

setPositionNodeList, setProximityList 를 통하여 미리 스캔 할 비콘의 정보를 입력한 다음, startHanabee 를 호출하여 실제 스캔을 시작합니다.

검색된 결과는 HanabeeResultCallback 콜백 함수를 통해 수신 받을 수 있습니다.

주의: setPositionNodeList, setProximityList 를 통해 미리 비콘의 정보를 설정하지 않으면 해당 비콘에 대한 정보를 수신 받을 수 없습니다.

```
mHanabee.startHanabee(new HanabeeResultCallback() {  
  
    @Override  
    public void onPosition(String floorID, double x, double y) {  
        // TODO Auto-generated method stub  
    }  
  
    @Override  
    public void onProximity(String macAddress, int radius, ProximityEvent
```

```
event, ProximityState state, double range, double rssi) {  
    // TODO Auto-generated method stub  
}  
  
@Override  
public void onHanabee(BeaconHanabee hanabee) {  
    // TODO Auto-generated method stub  
}  
  
@Override  
public void oniBeacon(BeaconiBeacon iBeacon) {  
    // TODO Auto-generated method stub  
}  
  
@Override  
public void onError(ErrorCode error, String detailedReason) {  
    // TODO Auto-generated method stub  
}  
});
```

HanabeeResultCallback 상세설명:

### [Indoor only] onPosition

#### Method:

```
public void onPosition(String floorID, double x, double y)
```

#### Description:

setPositionNodeList를 통해 입력한 비콘을 이용하여, 스캐너(스마트폰)의 현재 x, y좌표를 계산하여 리턴.

String floorID: 해당 목록의 ID. 현재는 PositionNodeList에 입력된 첫 번째 beacon의 MAC Address가 반환됨.

double x: 스캐너(스마트폰)의 현재 x 좌표

double y: 스캐너(스마트폰)의 현재 y 좌표

### onProximity

#### Method:

```
public void onProximity(String macAddress, int radius, ProximityEvent event, ProximityState state, double range, double rssi)
```

#### Description:

setProximityList로 설정한 비콘에 대한 정보를 지속적으로 수신.

String macAddress: 현재 수신 된 Beacon의 MAC Address.

int radius: 이전에 설정한 해당 Beacon의 반지름.

ProximityEvent event: 현재 발생한 이벤트를 반환

ProximityState state: 현재 상태를 반환

double range: 계산된 비콘과의 거리를 반환

double rssi: 수신된 RSSI를 반환

### onHanabee

#### Method:

```
public void onHanabee(BeaconHanabee hanabee)
```

#### Description:

수신 된 BLE Beacon 중 Hanabee beacon의 정보를 반환. 미리 정보를 지정하지 않아도 스캔결과값을 얻을 수 있음. 결과 값인 BeaconHanabee에서는 아래의 정보를 얻을 수 있음.

public int getBattery(): 현재 배터리 잔량

public String getUuid8(): 해당 Hanabee 비콘의 UUID8 을 반환 (8바이트 UUID, iBeacon의 16바이트 UUID와는 다름)

public int getMajor(): Major를 반환

public int getMinor(): Minor를 반환

public int getMeasuredPower(): MeasuredPower, 즉 1미터 거리에서의 RSSI값을 반환.

### oniBeacon

**Method:**

```
public void oniBeacon(BeaconiBeacon iBeacon)
```

**Description:**

수신 된 BLE Beacon 중 iBeacon의 정보를 반환. 미리 정보를 지정하지 않아도 스캔결과값을 얻을 수 있음. 결과 값인 BeaconiBeacon 에서는 아래의 정보를 얻을 수 있음.

```
public String getUUID(): 해당 비콘의 UUID를 반환 (iBeacon의 16바이트 UUID)
```

```
public int getMajor(): Major를 반환
```

```
public int getMinor(): Minor를 반환
```

```
public int getMeasuredPower(): MeasuredPower, 즉 1미터 거리에서의 RSSI값을 반환.
```

### 비콘 스캔 정지 / 재 시작하기:

아래 메소드를 통해 비콘 스캔을 정지 / 재 시작할 수 있습니다. Bluetooth LE Scan 은 배터리를 소모하는 기능이기 때문에, 동작을 원치 않을 때는 중지시켜 주는 것이 좋습니다.

### stopHanabee ()

**Method:**

```
public void stopHanabee()
```

**Return type:**

```
void
```

**Description:**

비콘 스캔을 정지.

restartHanabee ()

**Method:**

```
public void restartHanabee()
```

**Return type:**

```
void
```

**Description:**

startHanabee 을 한 번 이상 실행한 적 있을 때, 스캔이 정지되어 있다면, 비콘 스캔을 재 시작.

그 외: Hanabee 설정 등 기타 API 입니다.

Method	Return value	Description
isPossibleBLE(Context)	boolean	현재 단말의 BLE 지원여부 반환
setScanInterval(int)	void	스캐닝 간격을 설정, 단위는 ms. 기본값은 1 초(1000)이며 이보다 작은 값이 들어올 경우 1 초로 설정됨.

Scan interval description:

아래와 같이 설정하시면 됩니다.

```
mHanabee = Hanabee.getInstance(this);  
// Interval should be less than 1000 (default).  
mHanabee.setScanInterval(10000);
```

- 1초 이상으로 설정 시, 위치 / 거리 계산 정밀도는 매우 떨어지게 됩니다.
- 3초 이상 스캔 시, 3초 스캔 후 대기하게 됩니다.
- 즉 10초로 설정하면, 3초 스캔 - 7초 대기 - 3초 스캔 이 반복됩니다.  
만약 20초로 설정했다면, 3초 스캔 - 17초 대기가 반복됩니다. 다시 말해 최소 3초 이상은 스캔을 지속하게 됩니다.
- 1초 이상, 3초 이하로 설정 시, 스캔 자체는 지속적으로 수행되게 되며, 다만 설정한 interval 값에 해당하는 시점에 위치/거리 측정 알고리즘을 계산하게 됩니다.

## 이벤트 가이드

hanabee 1.0.4 부터 지원하는 Event / State 에 대한 가이드입니다.

Event는 특정 비콘과 일정 이하의 거리가 되었을 때, 이를 알려주는 신호입니다.  
Event 를 이용하기 위해서는 먼저 해당 비콘의 Mac address와 해당 비콘과의 거리 기준 값, 오차 범위한계를 설정 해 주어야 합니다.

이용 예:

```
proximityList.addProximityRegion("90:59:AF:2A:95:90", 50, 1);
```

이 경우 Mac address가 “90:59:AF:2A:95:90” 인 비콘에 대하여, 반지름 50미터, 오차 범위한계 1미터를 설정했습니다.

Event는 콜백 함수의 형태로 전달되며, 아래와 같은 종류가 있습니다.

Approach: 비콘의 BLE 전파를 처음 탐지

Enter: 설정한 영역 이내에 진입

Leave: 설정한 영역 이내에서 영역 밖으로 이동

Vanished: 비콘에서 멀어져서 BLE 전파가 더 이상 수신되지 않는 영역까지 떨어짐

Appear: 멀리서 다가오는 것이 아니라, 설정한 영역 내에서 스마트 폰이 등장

Expired: Enter 이후 일정시간이 지난 뒤, 신호가 사라짐

StayIn: Enter 이후 해당 위치에서 계속 머무름

StayOut: Approach 이후 해당 위치에서 계속 머무름



**오차 극복을 위한 추가 설정:** 비콘의 설치 지역에 따라 어느 정도의 신호 오차가 발생할 수 있습니다. 이 때, 정확한 이벤트 발생을 위해 각 이벤트 별로 일정 시간 이상 일정 횟수 이상의 이벤트가 중첩되어야 해당 이벤트가 발생한 것으로 설정할 수 있습니다.

설정은 아래 해당 변수에 대하여 hanabee 인스턴스의 getter / setter 를 이용하여 설정하시면 됩니다.

```
public class EventRules {  
  
    public static int APPROACH_EVENT_COUNT = 0;  
    public static int APPROACH_EVENT_TIME = 0;  
  
    public static int STAYOUT_EVENT_COUNT = 0;  
    public static int STAYOUT_EVENT_TIME = 20000;  
    public static int STAYOUT_EVENT_INTERVAL = 60000;  
  
    public static int ENTER_EVENT_COUNT = 3;  
    public static int ENTER_EVENT_TIME = 0;  
  
    public static int APPEAR_EVENT_COUNT = 0;  
    public static int APPEAR_EVENT_TIME = 0;  
  
    public static int STAYIN_EVENT_COUNT = 0;  
    public static int STAYIN_EVENT_TIME = 20000;  
    public static int STAYIN_EVENT_INTERVAL = 60000;  
  
    public static int LEAVE_EVENT_COUNT = 3;  
    public static int LEAVE_EVENT_TIME = 0;  
  
    public static int VANISH_EVENT_COUNT = 0;  
    public static int VANISH_EVENT_TIME = 20000;  
  
    public static int EXPIRED_EVENT_COUNT = 0;  
    public static int EXPIRED_EVENT_TIME = 20000;  
  
}
```

예: `APPROACH_EVENT_COUNT` 가 10 인 경우, APPROACH 이벤트가 10 회 이상 발생해야 해당 이벤트가 발생한 것으로 인식.

`APPROACH_EVENT_TIME` 이 1000 인 경우, 이전의 상태변화 이후 1000ms 가 지난 다음, 새로운 이벤트가 발생가능하며 1000ms 이전에 발생한 이벤트는 무시.

## Trouble shooting & FAQ

개발 중 발생할 수 있는 에러와 자주 질문하시는 내용에 대한 설명입니다.

### hanabee SDK 없이 hanabee 비콘 개발을 할 수 있나요?

질문: 안드로이드 기본 샘플 예제에도 Android BLE 샘플이 들어있는 것으로 알고 있습니다. 예제를 읽어보니 정말 쉽더군요. 저는 hanabee SDK 없이 모든 부분을 직접 구현하고 싶은데, 어떻게 하면 되나요?

답변: 부분적으로는, hanabee SDK 없이도 hanabee 비콘을 응용한 개발이 가능합니다. 제품 설명서에 포함되어있는 Mode 설정을 참고하여, hanabee Beacon 을 iBeacon 호환모드로 설정하면 다른 iBeacon 과 동일하게 동작하게 되며, 이를 이용한 개발이 가능합니다.

단, iBeacon 모드에서는 hanabee 의 Advertisement packet encryption 이 동작하지 않습니다. 즉, 누구나 해당 Advertisement 의 내용을 decryption 없이 확인할 수도 있습니다.

### initBLE() 에서 에러발생(1)

질문: java.lang.SecurityException: Need BLUETOOTH permission: Neither user ????? nor current process has android.permission.BLUETOOTH.Hanabee 라는 에러가 나오는데 어떻게 해야 하나요?

답변: Android.xml 에서 BLUETOOTH\_PERMISSION 을 설정하지 않아 발생하는 에러입니다. Android.xml 에 아래와 같은 코드를 추가하세요.

```
<uses-permission android:name="android.permission.BLUETOOTH" />
```

추가적으로, 아래 코드도 추가되어야 합니다.

```
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

### initBLE() 에서 에러 발생(2)

질문: java.lang.SecurityException: Need BLUETOOTH ADMIN permission: either user ????? nor current process has android.permission.BLUETOOTH\_ADMIN. 이라는 에러가 나오고 있습니다. 어떻게 수정해야 하나요?

답변: BLUETOOTH\_ADMIN 권한을 설정하지 않아 발생한 문제입니다. 아래와 같이 권한을 설정하세요. 두 가지 권한이 모두 설정되어 있어야 합니다.

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

### hanabee 비콘이 없는데 샘플소스를 실행시키고 싶습니다.

질문: 깃 허브에서 소스를 봤는데 집에서 한 번 실행시켜보고 싶습니다. 그런데 hanabee 비콘을 아직 구매할 계획은 없습니다. 테스트할 수 있는 방법이 있을까요?

답변: 현재로서는 비콘 없이 테스트하는 것은 불가능합니다. 판교 하나마이크론 SI 사업팀에 방문하시면 비콘 시연 및 테스트가 가능합니다. 조만간 비콘을 상시 테스트 할 수 있는 환경을 제공해드릴 예정입니다.

### 안드로이드 단말을 가상의 hanabee 비콘으로 이용할 수 있나요?

질문: Android 단말이 Scanner 역할을 하고 있는 것 같습니다. Scanner 가 가능하다면 Advertiser 로도 동작할 수 있는 게 아닌가요? 그렇다면 제 Android 단말을 가상의 hanabee 비콘으로 이용할 수 있지 않을까요?

답변: 결론부터 말씀 드리자면, 불가능합니다. 2015 년 1 월 현재 Android 최신버전인 5.0 Lolipopt 에서 BLE Advertiser 기능을 제공하고 있습니다. 이를 통해 Android 단말을 비콘으로 만드는 것은 가능합니다. 그러나, 이 방식으로는 hanabee 의 암호화 모드를 구현할 수 없어 hanabee 비콘으로는 이용할 수 없습니다.

### PointDouble 클래스를 Android intent 의 parcelable 로 전달할 수 있나요?

질문: startPositionScan 의 콜백함수에서 값을 PointDouble 이라는 클래스로 반환하고 있는데, 이 값을 Android intent 의 parcelable 형태로 전달하고 싶습니다. 가능한가요?

답변: PointDouble 클래스는 Parcelable 클래스를 상속받았고, 필요한 기능이 구현되어 있어 Parcelable 로 전달하셔도 됩니다. 단순히 Double 변수 두 개를 리턴하기 위한 목적으로 쓰고 있는 것이고, 그 이외엔 별 특별한 이유는 없기 때문에 Parcelable 로 전달하셔도 되고, x, y 값을 각각 따로 이용하셔도 됩니다.

### Positionning 에서 결과 값이 계속 (0,0) 으로 나옵니다.

질문: startPositionScan 을 실행했는데, 콜백 함수로 전달받은 PointDouble trackingResult 값의 x, y 가 모두 계속 0,0 으로 나오고 있습니다. 정상적인 값을 받으려면 어떻게 해야 하나요?

답변: startPositionScan 메소드를 수행하기 전에 반드시 setPositionNodeInfo 에서 비콘 4 개의 위치를 설정해 주어야 합니다. 아래 코드를 참고하시면 됩니다. 순서는 (1) BeaconInfo 의 ArrayList 를 하나 만듭니다. (2) 만들어진 ArrayList 에 BeaconInfo 를 추가합니다. X 좌표, Y 좌표, Hanabee Beacon 의 MAC Address 를 설정해주면 됩니다. (3) 이 ArrayList 를 setPositionNodeInfo 메소드를 이용하여 Hanabee 인스턴스에 입력해줍니다. (4) startPositionScan 을 실행하면 콜백 함수로 적절한 현재 좌표 값을 받을 수 있습니다.

```
private Hanabee mHanabee;  
mHanabee = Hanabee.getInstance(this);
```

```
ArrayList<BeaconInfo> positionNodeList = new ArrayList<BeaconInfo>();
positionNodeList.add(new BeaconInfo(0, 0, "90:59:AF:2A:AD:3F"));
positionNodeList.add(new BeaconInfo(0, 8, "90:59:AF:2A:94:68"));
positionNodeList.add(new BeaconInfo(8, 8, "90:59:AF:2A:C4:04"));
positionNodeList.add(new BeaconInfo(8, 0, "90:59:AF:2A:AD:4C"));

mHanabee.setPositionNodeInfo(positionNodeList);

mHanabee.startPositionScan(new Hanabee.PositionScanCallback() {
    @Override
    public void onPositionScan(PointDouble trackingResult) {
        // TODO trackingResult.x 와 trackingResult.y를 이용하여 코드 작성
    }
});
```

### setPositionNodeInfo 로 좌표 입력 시 음수를 입력해도 되나요?

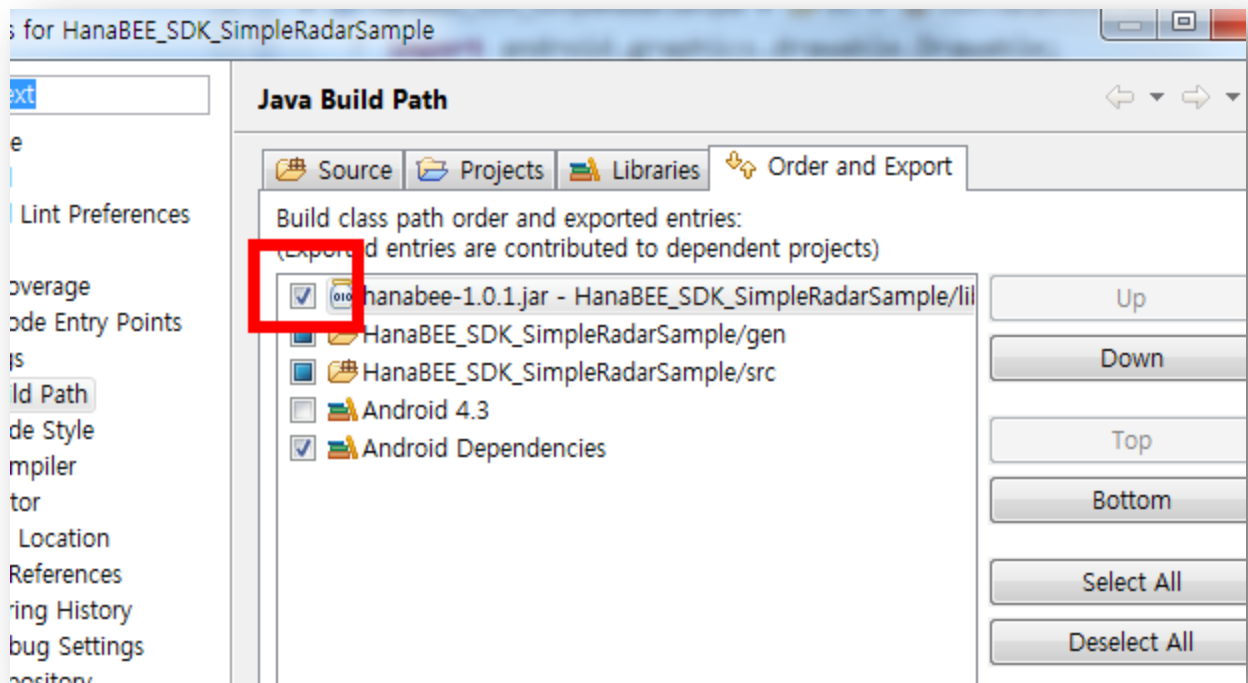
질문: setPositionNodeInfo 에 좌표 값으로 굳이, 꼭, 별로 필요하지는 않지만 음수를 입력하고 싶습니다. 불가능하지는 않은 일이겠지요?

답변: 어... 음... 지금은 중학교 때 배운 좌표의 1 사분면만을 생각해주시면 감사하겠습니다. 설정하는 hanabee Beacon 의 좌표 값 X, Y, 둘 다 양수일 경우에만 정상적인 값을 리턴되는 것을 보장할 수 있습니다.

### NoClassDefFoundError 가 발생합니다. 어떻게 해결하나요?

질문: lib 에 있는 jar 파일을 제 프로젝트로 카피한 뒤, NoClassDefFoundError 가 지속적으로 발생합니다. 어떻게 해결해야 하나요?

답변: Java build path 가 적절하게 설정되지 않았을 때 위와 같은 에러가 발생할 수 있습니다. 프로젝트 – 우클릭 – Java build path 를 선택하시고, Order and Export 화면에서 hanabee.jar 파일을 체크 해주시면 이 문제가 해결 됩니다.



**hanabee 비콘을 주문하고 싶습니다. 쇼핑몰 같은 것이 있나요?**

질문: BLE 비콘, iBeacon 으로 개발을 해 보고 싶습니다. 혹시 별도의 판매창구가 있나요? 쇼핑몰 링크가 있으면 알려주시면 감사하겠습니다.

답변: 네이버 스토어 팜을 통해 저희 제품을 구매하실 수 있습니다.

네이버 스토어팜:

[http://shopping.naver.com/search/all\\_search.nhn?query=hanabee%20beacon](http://shopping.naver.com/search/all_search.nhn?query=hanabee%20beacon)

홈페이지: <http://www.hanabee.io/>

블로그: <http://hanabeebeacon.tumblr.com/>

### hanabee SDK 가 잘 동작하지 않습니다. 뭐가 문제인가요?

질문: 모든 걸 다 세팅 했다고 생각했는데, 잘 동작을 하지 않습니다. 어떻게 해야 할까요?

답변: 1)먼저 Android 단말의 블루투스 기능을 한 번 껐다가 켜세요. (특정 단말의 경우에는, 재부팅이 필요할 수 있습니다).

2) 그리고 비콘에 전원이 정상적으로 들어와 있는지 확인 하세요. hanabee

Manager 를 통해 해당 비콘의 상태를 확인할 수 있습니다.

( <https://play.google.com/store/apps/details?id=com.hanamicron.hanabeemanager> )

3) Android.xml 에서 Bluetooth 권한이 잘 설정되어 있는지를 먼저 확인하세요.

그리고 BLEScanService 가 등록되어 있는지 확인하세요.

4)그 다음, hanabee 에 필요한 jar 파일과 각 폴더가 정상적인 위치에 있는지 확인하시고, jar 가 build path 에 적절하게 들어가 있는지 확인하세요.

이 내용은 이 문서 처음 부분의 hanabee SDK API Details 에서도 확인하실 수 있습니다.

5)이것도 안 된다면, [hanabee@hanamicron.co.kr](mailto:hanabee@hanamicron.co.kr) 로 증상에 대한 메일을 보내주세요.

### 전파인증, KC 인증을 받으셨나요?

질문: 한국에서는 전파인증(KC 인증)을 받아야 전자제품을 팔 수 있는 것으로 알고 있습니다. 혹시 전파인증을 받으셨나요?

답변: hanabee Slim / Mini 모두 KC 인증을 받았으며, 승인번호는 아래와 같습니다.

hanabee Slim : MSIP-CRM-HSD-HBS-900 (2014/10/28)

hanabee Mini : MSIP-CRM-HSD-HBM-900 (2014/12/07)

### 다른 질문이 있으면 어디로 물어보면 되나요?

기타 문의사항이 있으시면, [hanabee@hanamicron.co.kr](mailto:hanabee@hanamicron.co.kr) 로 문의 주시기 바랍니다.

## License

2015. 하나마이크론(주)  
2015. HANA Micron. Inc.

**hanabee** 는 하나마이크론의 등록상표이며, 모든 권리는 하나마이크론에 귀속되어 있습니다.

**hanabee** is registered Trade Mark of HANA Micron. Inc. All rights reserved.