

Android SDK

버전	수정 날짜	수정 내용
V.3.0.0	2026.02.09	<ul style="list-style-type: none"> • interface DotDeviceMessage <ul style="list-style-type: none"> - receivedErrorCallback 추가 (내부 에러 전송 callback) - ErrorCodes 내부 에러 코드 제공 • connect() 함수에서 DotDevice를 리턴하지 않도록 변경, DotDevice는 디바이스 연결 성공 시 제공 (DataCodes.Connected) • 모든 기기 연결 해제 실행 시 연결 된 기기와 연결 시도 중 인 기기 모두 연결 해제 되도록 변경
	2026.01.28	<ul style="list-style-type: none"> • USB 기기(300셀, 20셀) 연결 시 App에서 Type 체크 → SD에서 자동 Type 체크 후 연결 • DataCodes 추가 <ul style="list-style-type: none"> - <i>CommandSendFail : CommandSend 전송 실패 시</i> - <i>Reconnecting : 재연결 시도 시</i> • BLE 연결 및 재연결 안정화 (DotPadProcess) <ul style="list-style-type: none"> - reconnectAttempts : 최대 재 연결 시도 횟수(10번) - reconnectTimeoutMs : 재 연결 시간 (3000)
	2026.01.07	<ul style="list-style-type: none"> • Dot 점역 라이브러리에서 메모리 간섭으로 인한 앱 비 정상 오류 인한 문제 해결
	2025.11.28	<ul style="list-style-type: none"> • 초기 문서 작성

목차

1. 기본 동작 설명
2. SDK 버전
3. 권한
4. Class 설명
5. 리스너
6. Function

1. 기본 동작

1. DotPadScanner init 시 Application을 넘겨줌
 - * DotPadScanner 을 사용하지 않으려면 **DotPadProcess application Setting**은 필수
2. DotPadScanner Init 후 DotPadSearchListener 연결
3. **startBleScan or startUsbScan 호출 후** onBLEScan or onUSBScan 리스너 연결 할 Device 받기

4. **DotPadProcess fun setCallBack(dotDeviceMessage: DotDeviceMessage) 리스너 연결**

5. **DotPadProcess.connect 호출 하여 기기 연결**

6. 기기 연결 및 정보, 프로토콜 전송 시 **DotDeviceMessage** 리스너를 통해 전달

* **DotPad320A 모델은 USB연결시 A-to-C 케이블만 지원가능하며 C타입 젠더를 통해 연결하여 사용가능**

* **DotPad320X 모델은 C-to-C 케이블 사용이 가능합니다.**

2. Version 정보

2.1 현재 SDK 빌드 환경

Library 명	버전
com.android.application	8.13.0
org.jetbrains.kotlin.android	2.0.21
org.jetbrains.kotlin.plugin.compose	2.0.21
com.android.library	8.13.0

Dependency	버전
androidx.core:core-ktx	1.10.1
androidx.appcompat:appcompat	1.6.1
com.google.android.material:material	1.10.0
com.squareup.okhttp3:okhttp	4.12.0

Gradle Wrapper:

```
distributionUrl=https\://services.gradle.org/distributions/gradle-8.13-bin.zip
```

2.2 SRT 1.8.x 호환 버전

Library 명	버전
com.android.application	8.4.2
org.jetbrains.kotlin.android	1.8.0
com.android.library	8.4.2

Gradle Wrapper:

```
distributionUrl=https\://services.gradle.org/distributions/gradle-8.7-all.zip
```

3. 권한

1. BLE 권한

```
<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
<uses-permission android:name="android.permission.BLUETOOTH_SCAN"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADVERTISE"/>
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

2. USB

```
<uses-feature android:name="android.hardware.usb.host" />
```

3. 파일 Open (DTMS OPEN)

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

4. Class 설명

Class 명	설명
DotPadScanner	<ul style="list-style-type: none">• Bluetooth / USB 기기를 조회하는 클래스• DotPadScanner 객체 생성시 DotPadProcess.application 자동 세팅
DotPadProcess	<ul style="list-style-type: none">• Bluetooth / USB 연결을 통합 관리하는 프로세스 클래스• 여러 기기를 동시에 관리(내부 리스트 보유)• 중복 연결 방지: 동일 식별자(USB: vendorId:productId:deviceName, BLE: address)로 이미 연결된 기기가 있으면 그 인스턴스를 재사용• DotPadProcess 사용시 DotPadProcess.application 세팅 필수!• KM2-300A, KM2-20 기기로 동시 연결 테스트 완료

5. 리스너

1. DotPadScanner - interface DotPadSearchListener

Function	설명
onBLEScan(device: BluetoothDevice)	BLE 기기 조회

onUSBScan(devices: List<UsbDevice>)	USB 기기 조회
isBLEScanning(isScanning: Boolean)	BLE 조회 여부
onError(msg:String)	오류 메시지

2. DotPadProcess

Function	설명
fun setCallBack(dotDeviceMessage: DotDeviceMessage)	콜백 인터페이스 등록

Class	설명	Function 및 변수
interface DotDeviceMessage	<ul style="list-style-type: none"> • 장치 비식별 함수(기기 구분이 필요 없는 간단한 처리). • receivedMessageCallBack : 상기 DataCodes 기반 이벤트 수신. • sendMessageCallback : 송신한 프로토콜 원문(로깅/디버깅용) • receivedErrorCallback : ErrorCodes 기반 이벤트 수신. 	<pre>fun receivedMessageCallBack(dataCode: DataCodes, msg: String) fun sendMessageCallback(protocol: String) fun receivedErrorCallback(errorCode: ErrorCodes, msg: String)</pre>
	<ul style="list-style-type: none"> • 장치 식별 포함 함수(여러 기기 동시 연결 시 권장). • device 로 어떤 기기에서 온 이벤트인지 식별 가능. • receivedKeyCallBack: 키 이벤트 수신. 	<pre>fun receivedMessageCallBackWithDevice(device: DotDevice, dataCode: DataCodes, msg: String) fun receivedKeyCallBack(device: DotDevice?, keyCode: KeyCodes, msg: String)</pre>
enum class DataCodes	디바이스와 통신 중 발생하는 이벤트/응답의 종류 를 나타냅니다.	<ul style="list-style-type: none"> • Connected : 기기 정보 세팅까지 완료된 실사용 가능 상태. • ConnectedFail : 연결 시도 실패. • Disconnected : 연결 종료. • BoardInfo : 보드 레이아웃/지원 정보 수신(그래픽/텍스트 지원, 행/열 수 등). • BleMacAddress : BLE MAC 주소 정보 수신. • DeviceName : 기기명 수신. • DeviceFWVersion : 펌웨어 버전 수신. • DeviceHWVersion : 하드웨어 버전 수신. • ResponseDisplayLineAck : 해당 라인 수신 • ResponseDisplayLineNonAck : 해당 라인 미수신 • ResponseDisplayLineComplete : 해당 라인 출력 완료 • CommandError : Command 에러 • CommandNone : 정의되지 않은 Command

		<ul style="list-style-type: none"> • <i>CommandSendFail</i> : Command 전송 실패 • <i>Reconnecting</i> : 재연결 시도
enum class KeyCodes	디바이스에서 보낸 키 이벤트 종류를 나타냅니다.	<ul style="list-style-type: none"> • <i>KeyFunction1</i> : Function 1번 키 입력 • <i>KeyFunction2</i> : Function 2번 키 입력 • <i>KeyFunction3</i> : Function 3번 키 입력 • <i>KeyFunction4</i> : Function 4번 키 입력 • <i>KeyFunction12</i> : Function 1번 키와 2번 키 입력 • <i>KeyFunction13</i> : Function 1번 키와 3번 키 입력 • <i>KeyFunction23</i> : Function 2번 키와 3번 키 입력 • <i>KeyFunction24</i> : Function 2번 키와 4번 키 입력 • <i>KeyFunction34</i> : Function 3번 키와 4번 키 입력 • <i>KeyElse</i> : KeyCodes에 정의되어 있지 않은 조합 키 입력 • <i>PanningAll</i> : Panning Left와 Right 키 입력 • <i>PanningLeft</i> : Panning Left 키 입력 • <i>PanningRight</i> : Panning Right 키 입력 • <i>LPF1</i> : Panning Left 키와 Function 1번 키 입력 • <i>RPF4</i> : Panning Right 키와 Function 4번 키 입력
enum class ErrorCodes	프로세스에서 발생하는 오류의 종류를 나타냅니다.	<ul style="list-style-type: none"> • <i>ApplicationNotSet</i>: application 세팅이 되지 않음 • <i>UsbPermissionDenied</i>: USB 권한 없음 • <i>AlreadyConnected</i>: 이미 연결된 기기 • <i>AlreadyConnecting</i>: 이미 연결 시도 중 • <i>ConnectFailed</i>: 연결 실패

6. Function

- Function

1. DotPadScanner

Function 명	설명
fun startBleScan()	BluetoothDevice를 중복없이 조회하여 리스너로 전송
fun stopBleScan()	Ble 스캔을 종료
fun startUsbScan()	USB 기기를 조회하여 List 형태로 리스너로 전송

2. DotPadProcess

2.1 기기 연결 및 해제

Function 명	설명
fun connect(device: UsbDevice)	<ul style="list-style-type: none"> • USB 연결 • 이미 동일 기기가 연결돼 있으면 null 반환

	<ul style="list-style-type: none"> • 연결시도중으로 내부 등록 후 연결이 완료되면 콜백으로 DataCodes.Connected와 DotDevice 객체 반환
fun connect(bluetoothDevice: BluetoothDevice)	<ul style="list-style-type: none"> • BLE 연결 • BLE 주소로 중복 여부 판단, 중복 시 null 반환 • 연결시도중으로 내부 등록 후 연결이 완료되면 콜백으로 DataCodes.Connected와 DotDevice 객체 반환
fun disconnect(dotDevice: DotDevice? = null)	<ul style="list-style-type: none"> • 연결 해제 • dotDevice == null 이면 모든 연결 해제 및 내부 목록 초기화 • 특정 dotDevice가 지정되면 해당 기기만 해제 후 목록에서 제거 <p>모두 해제: disconnect() 특정 기기만 해제: disconnect(targetDevice)</p>

2.2 기기 프로토콜 전송

Function 명	설명
fun requestDeviceInfo(dotDevice: DotDevice, type:DeviceInfo)	<ul style="list-style-type: none"> • 기기 정보 조회 <pre>enum class DeviceInfo {*DeviceName*, *FirmwareVersion*, *HardwareVersion*}</pre> <ul style="list-style-type: none"> • *DeviceName * 기기 이름 요청 프로토콜 전송 • *FirmwareVersion * 펌웨어 버전 요청 전송 • *HardwareVersion * 하드웨어 버전 요청 전송 <p>응답은 콜백 (DotDeviceMessage / DotDeviceMessageWithDevice) 으로 수신됩니다.</p>
fun displayGraphicData(hexData: String, dotDevice: DotDevice? = null, displayMode: DisplayMode = DisplayMode.GraphicMode)	<ul style="list-style-type: none"> • 그래픽데이터 전송 • 그래픽 데이터(hex 문자열) 표시 • dotDevice == null 이면 모든 연결된 기기에 전송 • 기본 displayMode는 GraphicMode전체 기기에 표시: displayGraphicData("FFFFFFFF") 특정 기기에 표시: displayGraphicData("FFFFFFFF", targetDevice)
fun displayTextData(inputData: String, dotDevice: DotDevice? = null, needsTranslation: Boolean = true, displayMode: DisplayMode = DisplayMode.TextMode, callback: ((dotDevice:DotDevice, result: String) → Unit)? = null)	<ul style="list-style-type: none"> • 텍스트데이터 전송 • 텍스트 데이터 표시 • needsTranslation == true 이면 내부 점역 (BrailleUtil.translateToString) 후 표시 • callback 반환값: 기기별 실제 표시에 사용한 점자 문자열(점역 결과)점역 후 표시: displayTextData("Hello", targetDevice) hex 문자열로 바로 표시: displayTextData("FFFFFF", null, needsTranslation = false)
fun displayGraphicData(dotDataBoolean: List<List<Boolean>>, dotDevice:DotDevice? = null)	<ul style="list-style-type: none"> • 2차 배열 데이터 전송 • 2차 배열을 hex사로 변환 후 그래픽 영역에 표시 • dotDevice == null 이면 모든 연결된 기기에 전송

fun displayTextData(dotDataBoolean: List<List<Boolean>>. dotDevice:DotDevice? = null)	<ul style="list-style-type: none"> • 2차 배열 데이터 전송 • 2차 배열을 hex로 변환 후 텍스트 영역에 표시 • dotDevice == null 이면 모든 연결된 기기에 전송
fun displayLineData(lineId:Int, startCellIndex:Int, hexData:String, displayMode:DisplayMode, dotDevice:DotDevice? = null)	<ul style="list-style-type: none"> • 특정 라인 데이터 전송 • 특정 라인 데이터(hex 문자열) 표시 • dotDevice == null 이면 모든 연결된 기기에 전송

2.3 DTMS 열기

Function 명	설명
fun getDTMStoURI(jsonString: String): DTMS	<ul style="list-style-type: none"> • DTMS JSON 문자열을 파싱하여 DTMS 객체 생성 • JSON 예시 구조: {"items": "[{ \"graphic\": {\"data\": \"...\"}, \"text\": {\"plain\": \"...\"} }, ...]\"} • assets 폴더의 샘플 DTMS 파일 참조

2.4 점역

Function 명	설명
fun getBrailleLanguages() : Map<String, List<String>>	<ul style="list-style-type: none"> • 점역 언어 및 Grade
fun setBrailleLanguages(selectLanguage: String, selectGrade:String?)	<ul style="list-style-type: none"> • selectLanguage : 점역 언어 설정 (getBrailleLanguages에 Key) • selectGrade : 점역 Grade (getBrailleLanguages에 List)
fun translateText(inputText:String, callback: (String) → Unit)	<ul style="list-style-type: none"> • 언어 점역 하기 • inputText : 평문 언어 • callback : 점역 된 언어 callback

*** 일본어 점역은 네트워크가 필요하여 Gradle에 *libs.okhttp3* 추가 필요, 일본어 점역을 사용하지 않을 경우 필요 없음**

2.5 블루투스 재연결

- 연결된 기기의 블루투스가 끊겼을 경우 재연결 시도

변수 명	설명
reconnectAttempts : Int	최대 재연결 시도 횟수
reconnectTimeoutMs:Long	재연결 타임아웃

*** 디폴트로 reconnectAttempts = 10, reconnectTimeoutMs = 3000으로 세팅되어 있음**