



ATID Co.,Ltd

AT288N API Reference Guide for Android Developers

Android Developer Guide

SW 팀


2023-06-12

개정 이력

| 버전 | 개정일자 | 개정사유 ¹ | 개정내역 ² | 작성자 |
|------|------------|-------------------|-------------------|------|
| v0.1 | 2016-05-25 | 초안 | 신규 생성 | 민경진 |
| v0.2 | 2017-03-13 | 수정 | 문서 보완 및 수정 | 조영진 |
| v0.3 | 2019-01-21 | 수정 | Intro 문구 수정. | 조영진 |
| v0.4 | 2023-06-12 | 수정 | Intro 문구 수정 | SW 팀 |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |


¹ 개정사유 : 제정 또는 개정 내용이 이전 문서에 대해 추가/수정/삭제인지 선택 기입

² 개정내역 : 개정이 발생하는 페이지 번호와 변경 내용을 기술

| | | | | | | | |
|---|--|---|------|----|------------|--------------|------|
|  | | AT288N API Reference Guide for Android Developers | | | | | |
| Android Developer Guide | | | | | 회사 | ATID Co.,Ltd | |
| 문서이름 | | 작성자 | SW 팀 | 날자 | 2023-06-12 | 버전 | v0.4 |

목차

| | |
|---|----|
| 목차 | 3 |
| 1. Intro | 4 |
| 2. Reference Library Guide | 5 |
| 2.1. Reader Class | 5 |
| 2.1.1. Constructor..... | 5 |
| 2.1.2. Event Synchronization Method..... | 5 |
| 2.1.3. Device Control Method..... | 8 |
| 2.1.4. Action Method..... | 10 |
| 2.1.5. Property Method..... | 15 |
| 2.1.6. Extended Property Method..... | 23 |
| 2.1.7. Static Method..... | 29 |
| 2.2. IReaderCallbackReceiver Interface..... | 30 |
| 2.2.1. Method..... | 30 |
| 2.3. Enumeration | 33 |
| 2.3.1. ModelType | 33 |
| 2.3.2. TagType | 33 |
| 2.3.3. BankType | 33 |
| 2.3.4. CommType..... | 33 |
| 2.3.5. InventoryFormatType..... | 33 |
| 2.3.6. SelectionActionType..... | 34 |
| 2.3.7. StoredModeType..... | 34 |
| 2.4. Constant | 35 |

| | | | | | | | |
|---|--|---|------|----|------------|--------------|------|
|  | | AT288N API Reference Guide for Android Developers | | | | | |
| Android Developer Guide | | | | | 회사 | ATID Co.,Ltd | |
| 문서이름 | | 작성자 | SW 팀 | 날자 | 2023-06-12 | 버전 | v0.4 |


1. Intro

본 문서는 AT288N SDK Library를 사용하여 응용 프로그램을 개발하고자 하는 Android개발자들을 위하여 SDK Library 사용 방법을 기술 하는데 그 목적이 있다.

본 문서에서 사용되는 개발 도구는 Android Studio 가 사용되었고 개발 대상 플랫폼은 Android 4.3 이상을 지원한다.

※ v1.10.2019012200 부터, Library, Demo, Sample 은 Eclipse가 아닌 Android Studio로 빌드된다.

| Library | 설명 |
|-------------------------|---|
| at288lib.jar | AT288N을 제어하기 위한 Android용 Library |
| At288lib.aar | V1.10.2019012200 부터, jar 파일이 aar 파일로 변경됨. |

| | | | | | | | |
|---|--|---|------|----|------------|----|--------------|
|  | | AT288N API Reference Guide for Android Developers | | | | | |
| All That Identification | | Android Developer Guide | | | 회사 | | ATID Co.,Ltd |
| 문서이름 | | 작성자 | SW 팀 | 날자 | 2023-06-12 | 버전 | v0.4 |

2. Reference Library Guide

2.1. Reader Class

Reader 클래스는 AT288 SDK Library를 사용하여 AT288N Device에 연결하거나 연결을 종료하고, 메시지를 호출하여 AT288N Device를 제어하고 속성을 설정하거나 반환하며, 이벤트로 AT288N Device로부터 반환되는 이벤트를 수신하는 클래스이다.

2.1.1. Constructor

Activity의 context와 이벤트를 수신할 Handler 개체를 가지고 Reader 클래스의 새 인스턴스를 초기화한다.

➤ Syntax

```
public Reader(Context context, IReaderCallbackReceiver receiver)
```

➤ Parameters

context: 사용자 Application의 Context

receiver: AT288N으로부터 수신된 이벤트를 처리하기 위해 Application에 구현된 interface

➤ Remarks

Context로부터 상속된 개체와 이벤트를 처리할 interface로 Reader 클래스의 새 인스턴스를 만든다.

2.1.2. Event Synchronization Method

Activity는 onCreate, onStart, onResume, onPause, onStop, onDestroy, onActivityResult등의 동기화 이벤트를 재정의하여 Activity의 Life Cycle를 관리한다. Reader 개체 역시 재정의된 함수에서 Life Cycle동기화 함수를 호출함으로써 Activity와 Life Cycle를 동기화 할 수 있다.

2.1.2.1. onCreate

Reader 개체의 소유 Activity가 오버라이드한 onCreate메서드에서 onCreate 이벤트를 동기화 하기 위해 호출한다.

➤ Syntax

```
public boolean onCreate()
```

➤ Return value

Bluetooth가 지원되지 않는 단말기이면 false를 반환한다.

➤ Remarks

소유 Activity의 onCreate 이벤트를 Reader 개체와 동기화 한다.

2.1.2.2. onStart

Reader 개체의 소유 Activity가 오버라이드한 onStart 메서드에서 onStart 이벤트를 동기화 하기 위해 호출한다.

➤ Syntax

```
public void onStart()
```

➤ Remarks

소유 Activity 의 onStart 이벤트를 Reader 개체와 동기화 한다.

2.1.2.3. onResume

Reader 개체의 소유 Activity가 오버라이드한 onResume 메서드에서 onResume 이벤트를 동기화 하기 위해 호출한다.

➤ Syntax

```
public void onResume()
```

➤ Remarks

소유 Activity 의 onResume 이벤트를 Reader 개체와 동기화 한다.

2.1.2.4. onPause

Reader 개체의 소유 Activity가 오버라이드한 onPause 메서드에서 onPause 이벤트를 동기화 하기 위해 호출한다.

➤ Syntax

```
public void onPause()
```

➤ Remarks

소유 Activity 의 onPause 이벤트를 Reader 개체와 동기화 한다.

2.1.2.5. onStop

Reader 개체의 소유 Activity가 오버라이드한 onStop 메서드에서 onStop 이벤트를 동기화 하기 위해 호출한다.

➤ Syntax

```
public void onStop()
```

➤ Remarks

소유 Activity 의 onStop 이벤트를 Reader 개체와 동기화 한다.

2.1.2.6. onDestroy

Reader 개체의 소유 Activity가 오버라이드한 onDestroy 메서드에서 onDestroy 이벤트를 동기화 하기 위해 호출한다.

➤ Syntax

```
public void onDestroy()
```

➤ Remarks

소유 Activity의 OnDestroy이벤트를 Reader 개체와 동기화 한다.

2.1.2.7. **onActivityResult**

Intent가 호출되는 메서드인 openDeviceListActivity메서드를 호출 후 Intent가 반환되는 값을 동기화 하기 위해 openDeviceListActivity를 호출한 Activity의 오버라이드된 onActivityResult메서드에서 호출한다.

➤ Syntax


```
public void onActivityResult (int requestCode, int resultCode, Intent data)
```

➤ Parameters

requestCode : App Activity의 오버라이드된 onActivityResult 메서드의 requestCode

resultCode : App Activity의 오버라이드된 onActivityResult 메서드의 resultCode

data : App Activity의 오버라이드된 onActivityResult 메서드의 data

| | | | | | | | |
|---|--|---|------|----|------------|----|--------------|
|  | | AT288N API Reference Guide for Android Developers | | | | | |
| All That Identification | | Android Developer Guide | | | 회사 | | ATID Co.,Ltd |
| 문서이름 | | 작성자 | SW 팀 | 날자 | 2023-06-12 | 버전 | v0.4 |

2.1.3. Device Control Method

Device Control 메서드들은 Reader 클래스의 인스턴스가 생성된 이후, 상황에 따라 AT288N에 연결 하는 방법을 제공한다.

2.1.3.1. openDeviceListActivity

주변 Bluetooth 장치를 검색하는 대화상자를 출력하고, 검색된 AT288N을 선택하여 Bluetooth 연결을 시도한다.

➤ Syntax

```
public void openDeviceListActivity()
```

➤ Remarks

Bluetooth 검색 Activity 가 생겨나며, Bluetooth 의 Inquiry 를 수행하여 Host 장비의 주변의 AT288N Device 를 검색한다. 검색 된 AT288N Device 를 선택하면 선택된 장비로 접속을 시도하고 접속 결과는 IReaderCallbackReceiver 인터페이스의 onReaderStateChange 메서드로 전달 된다. 2.4 Constants(Connection State) 참조.

2.1.3.2. connectMostRecentDevice

ConnectMostRecentDevice 메서드는 Reader와 최근에 연결되었던 AT288N Device에 접속을 시도하기 위한 메서드이다.

➤ Syntax

```
public void connectMostRecentDevice()
```

➤ Remarks

최근 Bluetooth 로 접속되었던 AT288N Device 로 접속을 시도하며, 접속 결과는 IReaderCallbackReceiver 인터페이스의 onReaderStateChange 메서드로 전달 된다. 2.4 Constants(Connection State) 참조.

2.1.3.3. ensureDiscoverable


AT288N Device가 Reader에 접속할 수 있도록 접속 대기 상태로 설정한다.

➤ Syntax

```
public void ensureDiscoverable()
```

➤ Remarks

AT288N Device 가 Bluetooth 모드에 있을 경우, 전원이 켜지면 최근 접속했던 Host 장치를 찾아 Bluetooth 접속을 시도한다. 이때 Reader 가 접속대기 상태에 있어야만 접속이 가능하다. AT288N Device 가 접속되면 IReaderCallbackReceiver 인터페이스의 onReaderStateChange 메서드로 결과가 전달 된다. 2.4 Constants(Connection State) 참조.

| | | | | | | | |
|--|---|-----|------|----|------------|--------------|------|
|  All That Identification | AT288N API Reference Guide for Android Developers | | | | | | |
| Android Developer Guide | | | | | 회사 | ATID Co.,Ltd | |
| 문서이름 | | 작성자 | SW 팀 | 날자 | 2023-06-12 | 버전 | v0.4 |

2.1.3.4. **activate**

Reader개체와 AT288N Device와의 상태를 동기화 한다.

➤ **Syntax**

```
public void activate()
```

➤ **Remarks**

접속을 시도하면 IReaderCallbackReceiver 인터페이스의 onReaderStateChange 메서드로 결과가 전달 된다. 전달 받은 state 가 STATE_CONNECTED 이면 activate 메서드를 호출하여 AT288N Device 와 응용프로그램을 동기화 한다. active 메서드를 호출하면, IReaderCallbackReceiver 인터페이스의 onReaderExtendedProperty 메서드로 속성 값을 전달 한다. 2.4 Constants(Extended Property Codes) 참조.

2.1.3.5. **start**

AT288N Device와 통신할 준비를 한다.

➤ **Syntax**

```
public void start()
```

➤ **Remarks**

이 메서드는 openDeviceListActivity 또는 connectMostRecentDevice 를 사용해서 연결을 시도하면 SDK 내부에서 호출되는 메서드 이므로 임의로 호출해서는 안 된다.

2.1.3.6. **stop**


즉시 AT288N과 Host와의 연결을 종료 한다.

➤ **Syntax**

```
public void stop()
```

➤ **Remarks**

이 메서드는 onDestroy 가 호출될 때 실행되는 메서드이고, Host 와의 연결을 종료 할 때 사용한다.

| | | | | | | | |
|---|--|---|------|----|------------|----|--------------|
|  | | AT288N API Reference Guide for Android Developers | | | | | |
| All That Identification | | Android Developer Guide | | | 회사 | | ATID Co.,Ltd |
| 문서이름 | | 작성자 | SW 팀 | 날자 | 2023-06-12 | 버전 | v0.4 |

2.1.4. Action Method

Tag operation 기능을 제공한다.

2.1.4.1. inventory

setTagType, setContinueModeEx 메서드로 설정한 내용에 기반한 Inventory 기능을 수행한다.

➤ Syntax

```
public void inventory(String mask)
```

➤ Parameters

mask : Tag 를 select 하기 위한 masking pattern. (16 진수 문자열)

➤ Remarks

Inventory 된 태그 데이터는 IReaderCallbackReceiver 인터페이스의 onReaderReadTag 메서드로 전달 된다.

Selection 기능을 사용하기 위해서는, SetSelectionBank, SetSelectionOffset, SetSelectionAction 등의 메서드로 masking pattern 이 적용될 위치를 미리 설정 해야만 한다.

2.1.4.2. inventory6b6cAnyone

ISO 18000-6B/6C를 구분하지 않고 Inventory 기능을 수행한다.

➤ Syntax

```
public void inventory6b6cAnyone()
```

➤ Remarks

IsAT288N_MA 메서드의 호출 결과가 true 인 경우에만 사용가능 하다.

Inventory된 태그 데이터는 IReaderCallbackReceiver 인터페이스의 onReaderReadTag 메서드로 전달 된다.

2.1.4.3. inventoryMemory

Inventory 동작을 수행하며 Tag의 특정 메모리 Bank의 값을 함께 읽는다.

➤ Syntax

```
public void inventoryMemory(BankType bank, int offset, int length, String mask)
```

➤ Parameters

bank : 접근하려는 Tag 의 Memory Bank

offset : 접근하려는 Tag 의 Memory Bank 의 시작 주소

length : 접근하려는 Memory 의 길이

mask : Tag 를 select 하기 위한 masking pattern. (16 진수 문자열)

➤ Remarks

Inventory된 태그 데이터는 IReaderCallbackReceiver 인터페이스의 onReaderReadTag 메서드로 전달 된다.

IsAT288N_MA 메서드의 호출 결과가 true 인 경우에는 사용할 수 없다.

2.1.4.4. inventoryMultiple

Multiple mode로 Inventory 기능을 수행한다.

➤ Syntax

```
public void inventoryMultiple()
```

➤ Remarks

Inventory된 태그 데이터는 IReaderCallbackReceiver 인터페이스의 onReaderReadTag 메서드로 전달 된다.

2.1.4.5. inventorySelection

특정 Tag들에 대해서 Inventory 기능을 수행한다.

➤ Syntax

```
public void inventorySelection(String mask)
```

➤ Parameters

mask : Tag 를 select 하기 위한 masking pattern. (16 진수 문자열)

➤ Remarks

mask로 입력된 pattern으로 시작하는 데이터를 가지는 태그에 대해서만 Inventory를 수행하며, Inventory된 태그 데이터는 IReaderCallbackReceiver 인터페이스의 onReaderReadTag 메서드로 전달 된다.

2.1.4.6. inventorySingle

Single mode로 Inventory 기능을 수행한다.

➤ Syntax

```
public void inventorySingle()
```

➤ Remarks

한번만 Inventory 를 수행하고 중단하며, Inventory 된 태그 데이터는 IReaderCallbackReceiver 인터페이스의 onReaderReadTag 메서드로 전달 된다.

2.1.4.7. inventoryFiltering

Inventory 기능을 수행한다.

➤ Syntax

```
public void inventoryFiltering(String action, String mask1, String mask2)
```

➤ Remarks

사용하지 않음

2.1.4.8. inventoryVLC

Inventory 기능을 수행한다.

➤ Syntax

```
public void inventoryVLC()
```

➤ Remarks

사용하지 않음

2.1.4.9. stopOperation

현재 수행중인 Action 계열의 동작을 중지 시킨다.

➤ Syntax

```
public void stopOperation()
```

➤ Remarks

수행중인 동작을 취소하고, IReaderCallbackReceiver 인터페이스의 onReaderActionChange 메서드로 Stop 되었음을 전달한다.

Action : ACTION_STOP

2.1.4.10. readMemory

Tag의 특정 메모리 Bank의 값을 읽는다.

➤ Syntax

```
public void readMemory(BankType bank, int location, int length)
public void readMemory(BankType bank, int offset, int length, String mask)
```

➤ Parameters

bank : 접근하려는 Tag의 Memory Bank

location, offset : 접근하려는 Tag의 Memory Bank의 시작 주소

length : 접근하려는 Memory의 길이

mask : Tag를 select하기 위한 masking pattern. (16진수 문자열)

➤ Remarks

읽은 태그 데이터는 IReaderCallbackReceiver 인터페이스의 onReaderReadTag 메서드로 전달 되며, 실패한 경우에는 onReaderResponse 메서드로 실패 코드를 전달한다.

2.2.1.3 onReaderResponse 참조.

IsAT288N_MA 메서드의 호출 결과가 true 이고 getTagType 의 결과가 ISO18000_6B 인 경우에는 사용할 수 없다.

2.1.4.11. writeMemory

Tag의 특정 메모리 Bank에 값을 기록한다.

➤ Syntax

```
public void writeMemory(BankType bank, int location, String data)
public void writeMemory(BankType bank, int offset, String data, String mask)
```

Parameters

bank : 접근하려는 Tag의 Memory Bank

location, offset : 접근하려는 Tag의 Memory Bank의 시작 주소

data : Tag에 기록하려고 하는 data(16진수 문자열)

mask : Tag를 select하기 위한 masking pattern. (16진수 문자열)

Remarks

수행 결과는 IReaderCallbackReceiver 인터페이스의 onReaderResponse 메서드로 전달된다. 2.2.1.3 onReaderResponse 참조.

IsAT288N_MA 메서드의 호출 결과가 true 이고 getTagType 의 결과가 ISO18000_6B 인 경우에는 사용할 수 없다.

2.1.4.12. lock

Tag에 접근 할 수 없도록 잠그거나 잠금 해제를 수행한다.

Syntax

```
public void lock(String bitMask, String action)
public void lock(String bitMask, String action, String mask)
```

Parameters

bitMask : action 으로 입력된 값을 적용할지 여부를 결정하는 값. (4자리 16진수 문자열)

action : 설정한 위치를 lock을 하기 위한 값. (4자리 16진수 문자열)

mask : Tag를 select하기 위한 masking pattern. (16진수 문자열)

Remarks


bitMask 와 action 값은 다음 테이블을 참고하여 계산한다.

Mask/Action State

| Offset | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Mask | Kill Pwd | | Access Pwd | | EPC | | TID | | User | |
| | skip/ write | skip/ write | skip/ write | skip/ write | skip/ write | skip/ write | skip/ write | skip/ write | skip/ write | skip/ write |

| Offset | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|-----------------------|---------------|-----------------------|---------------|--------------|---------------|--------------|---------------|--------------|---------------|
| Action | Kill Pwd | | Access Pwd | | EPC | | TID | | User | |
| | pwd read/ write | perma lock | pwd read/ write | perma lock | pwd write | perma lock | pwd write | perma lock | pwd write | perma lock |

수행 결과는 IReaderCallbackReceiver 인터페이스의 onReaderResponse 메서드로 전달된다. 2.2.1.3 onReaderResponse 참조.

| | | | | | | | |
|--|---|-----|------|----|------------|--------------|------|
|  All That Identification | AT288N API Reference Guide for Android Developers | | | | | | |
| Android Developer Guide | | | | | 회사 | ATID Co.,Ltd | |
| 문서이름 | | 작성자 | SW 팀 | 날자 | 2023-06-12 | 버전 | v0.4 |

IsAT288N_MA 메서드의 호출 결과가 true 이고 getTagType 의 결과가 ISO18000_6B 인 경우에는 사용할 수 없다.

2.1.4.13. kill

Tag를 더 이상 사용할 수 없게 한다.

➤ Syntax

```
public void kill(String password)
public void kill(String password, String mask)
```

➤ Parameters


password : kill을 수행할 Tag에 저장되어 있는 kill password

mask : Tag를 select하기 위한 masking pattern. (16진수 문자열)

➤ Remarks

수행 결과는 IReaderCallbackReceiver 인터페이스의 onReaderResponse 메서드로 전달되며, Kill 에 성공한 Tag 는 다시 사용할 수 있도록 되돌릴 수 없으므로 kill 메서드 사용에 주의해야 한다. 2.2.1.3 onReaderResponse 참조.

IsAT288N_MA 메서드의 호출 결과가 true 이고 getTagType 의 결과가 ISO18000_6B 인 경우에는 사용할 수 없다.

| | | | | | | | |
|--|---|-----|------|----|------------|--------------|------|
|  All That Identification | AT288N API Reference Guide for Android Developers | | | | | | |
| Android Developer Guide | | | | | 회사 | ATID Co.,Ltd | |
| 문서이름 | | 작성자 | SW 팀 | 날자 | 2023-06-12 | 버전 | v0.4 |

2.1.5. Property Method

Tag operation의 속성 및 모듈 속성 설정 기능을 제공한다.

2.1.5.1. getModelType

호스트와 연결된 AT288N의 모델 타입을 반환한다.

➤ Syntax

```
public int getModelType()
```

➤ Return value

AT288N의 타입 값. (ModelType 열거형의 값 참조.)

➤ Remarks

반드시 activate 메서드 호출 이후에 사용해야 한다.

2.1.5.2. getFirmwareVersion

AT288N의 UHF 모듈의 F/W 버전을 읽는다.

➤ Syntax

```
public void getFirmwareVersion()
```

➤ Remarks

결과는 IReaderCallbackReceiver의 onReaderProperty의 파라미터로 전달 된다.

Code : PROPERTY_VERSION

2.1.5.3. getGlobalBand

UHF 모듈의 국가 설정 값을 읽는다.

➤ Syntax

```
public void getGlobalBand()
```

➤ Remarks

결과는 IReaderCallbackReceiver의 onReaderProperty의 파라미터로 전달 된다.

Code : PROPERTY_GLOBAL_BAND

2.1.5.4. getState

AT288N과 Host와의 접속 상태를 반환한다.

➤ Syntax

```
public int getState()
```

➤ Return value

접속 상태를 나타내는 값. 2.4 Constants(Connection State)를 참조.

2.1.5.5. getPower

Antenna 송신 파워 레벨을 읽는다.

➤ Syntax

```
public void getPower()
```

➤ Remarks

결과는 IReaderCallbackReceiver의 onReaderProperty의 파라미터로 전달 된다.

Code : PROPERTY_POWER_GAIN

2.1.5.6. setPower

Antenna 송신 파워 레벨을 설정한다.

➤ Syntax

```
public void setPower(int value)
```

➤ Parameters

value : 송신 파워 레벨(0~19)

➤ Remarks

value가 0이면 최대 출력이고(30dBm) value를 증가 시키면 출력은 1dBm씩 감소한다.

예로, value를 5로 설정했다면 출력되는 파워 레벨은 25dBm이 된다.

2.1.5.7. getAccessPassword

AT288N에 설정되어 있는 Access password를 읽는다.

➤ Syntax

```
public void getAccessPassword()
```

➤ Remarks

결과는 IReaderCallbackReceiver의 onReaderProperty의 파라미터로 전달 된다.

Code : PROPERTY_ACCESS_PASSWORD

2.1.5.8. setAccessPassword

Access password를 AT288N에 설정한다.

➤ Syntax

```
public void setAccessPassword(String password)
```

➤ Parameters

password : access password(16진수 문자열)

2.1.5.9. getContinueMode


Inventory mode가 multiple인지를 읽는다.

➤ Syntax

```
public void getContinueMode()
```

➤ Remarks

결과는 IReaderCallbackReceiver의 onReaderProperty의 파라미터로 전달 된다.

| | | | | | | | |
|--|---|-----|------|----|------------|--------------|------|
|  All That Identification | AT288N API Reference Guide for Android Developers | | | | | | |
| Android Developer Guide | | | | | 회사 | ATID Co.,Ltd | |
| 문서이름 | | 작성자 | SW 팀 | 날자 | 2023-06-12 | 버전 | v0.4 |

Code : PROPERTY_CONTINUE_MODE

2.1.5.10. **setContinueMode**

Inventory mode를 multiple mode로 설정 또는 해제 한다.

➤ **Syntax**

```
public void setAccessPassword(boolean enabled)
```

➤ **Parameters**

enabled : multiple mode 설정 여부

➤ **Remarks**

enabled가 true이면 multiple mode, false이면 single mode

2.1.5.11. **getSelectionBank**

Select가 수행될 Tag의 Memory Bank를 읽는다.

➤ **Syntax**

```
public void getSelectionBank()
```

➤ **Remarks**

결과는 IReaderCallbackReceiver의 onReaderProperty의 파라미터로 전달 된다.

Code : PROPERTY_SELECTION_BANK

2.1.5.12. **setSelectionBank**

Select가 수행될 Tag의 Memory Bank를 설정한다.

➤ **Syntax**

```
public void setSelectionBank(BankType type)
```

➤ **Parameters**

type : Memory bank

➤ **Remarks**

type에 대해서는 BankType 열거형 참조

2.1.5.13. **getSelectionOffset**

Select가 수행될 때 masking pattern의 비교가 시작될 주소를 읽는다.

➤ **Syntax**

```
public void getSelectionOffset()
```

➤ **Remarks**

결과는 IReaderCallbackReceiver의 onReaderProperty의 파라미터로 전달 된다.

Code : PROPERTY_SELECTION_OFFSET

2.1.5.14. **setSelectionOffset**

Select가 수행될 때 masking pattern의 비교가 시작될 주소를 설정한다.

➤ **Syntax**

```
public void setSelectionOffset(int offset)
```

➤ **Parameters**

offset : 시작 주소

➤ **Remarks**

시작 주소는 bit 단위로 설정한다.

2.1.5.15. **getSelectionAction**

Select가 수행될 때 적용될 action을 읽는다.

➤ **Syntax**

```
public void getSelectionAction()
```

➤ **Remarks**

결과는 IReaderCallbackReceiver의 onReaderProperty의 파라미터로 전달 된다.

Code : PROPERTY_SELECTION_ACTION

2.1.5.16. **setSelectionAction**

Select가 수행될 때 적용될 action을 설정한다.

➤ **Syntax**

```
public void setSelectionAction(SelectionAction type)
```

➤ **Parameters**

type : 적용할 action

➤ **Remarks**

type에 대해서는 SelectionAction 열거형 참조

2.1.5.17. **getSelectionSession**

Select가 수행될 때 적용될 session을 읽는다.

➤ **Syntax**

```
public void getSelectionSession()
```

➤ **Remarks**

결과는 IReaderCallbackReceiver의 onReaderProperty의 파라미터로 전달 된다.

Code : PROPERTY_SELECTION_SESSION

2.1.5.18. **setSelectionSession**

Select가 수행될 때 적용될 session을 설정한다.

➤ **Syntax**

```
public void setSelectionSession(int value)
```

➤ **Parameters**

value : 적용할 session

➤ **Remarks**

Value : 0(S0), 1(S1), 2(S2), 3(S3)

2.1.5.19. **getBuzzer**

AT288N의 buzzer 사용 여부를 읽는다.

➤ **Syntax**

```
public void getBuzzer()
```

➤ **Remarks**

결과는 IReaderCallbackReceiver의 onReaderProperty의 파라미터로 전달 된다.

Code : PROPERTY_BUZZER

2.1.5.20. **setBuzzer**

AT288N의 buzzer On/Off 상태를 설정한다.

➤ **Syntax**

```
public void setBuzzer(boolean enabled)
```

➤ **Parameters**

enabled : on/off

➤ **Remarks**

enabled : true(on), false(off)

2.1.5.21. **getAutoPowerOffTime**

Host와 연결되어 있지 않고, 사용자로부터 아무런 입력도 받지 않았을 때, AT288N의 전원을 off 할 때까지의 대기 시간 값을 읽는다.

➤ **Syntax**

```
public void getAutoPowerOffTime()
```

➤ **Remarks**

결과는 IReaderCallbackReceiver의 onReaderProperty의 파라미터로 전달 된다.

Code : PROPERTY_AUTO_POWEROFF

2.1.5.22. **setAutoPowerOffTime**

Host와 연결되어 있지 않고, 사용자로부터 아무런 입력도 받지 않았을 때, AT288N의 전원을 off 할 때까지의 대기 시간 값을 설정한다.

➤ **Syntax**

```
public void setAutoPowerOffTime(int value)
```

➤ Parameters

value : 전원 off까지의 대기 시간.

➤ Remarks

단위는 분(minute)이며, 0으로 설정하면 배터리가 방전될 때 까지 off되지 않는다.

2.1.5.23. **getResponseTimeout**

명령 수행 시간을 반환한다.

➤ Syntax

```
public int getResponseTimeout()
```

➤ Return value

명령 수행 시간. (1000 ~ 10000)

➤ Remarks

단위는 밀리초(milliseconds)이다.

2.1.5.24. **setResponseTime**

명령 수행 시간을 설정한다.

➤ Syntax

```
public void setResponseTime(int nTimeout)
```

➤ Parameters

nTimeout : 명령 수행 시간.

➤ Remarks

단위는 밀리초(milliseconds)이다.

2.1.5.25. **getDeviceName**

Host와 연결된 AT288N의 friendly Bluetooth 이름을 반환한다.

➤ Syntax

```
public String getDeviceName()
```

➤ Return value

AT288N의 friendly Bluetooth 이름을 String형으로 반환한다.

2.1.5.26. **getPortInventoryTime**

Inventory Round 시간 중에서 antenna가 활성화 되어 있는 시간을 읽는다.

➤ Syntax

```
public void getPortInventoryTime()
```

➤ Remarks

결과는 IReaderCallbackReceiver의 onReaderProperty의 파라미터로 전달 된다.

Code : PROPERTY_ANTENNA_SWITCHING_TIME

2.1.5.27. **setPortInventoryTime**

Inventory Round 시간 중에서 antenna가 활성화 되어 있는 시간을 설정한다.

➤ **Syntax**

```
public void setPortInventoryTime(int value)
```

➤ **Parameters**

value : 활성화 시간

➤ **Remarks**

단위는 밀리초(milliseconds)이다.

2.1.5.28. **getPortIdleTime**

Inventory Round 시간 중에서 antenna의 유휴 시간을 읽는다.

➤ **Syntax**

```
public void getPortIdleTime()
```

➤ **Remarks**

결과는 IReaderCallbackReceiver의 onReaderProperty의 파라미터로 전달 된다.

Code : PROPERTY_POWER_IDLE_TIME

2.1.5.29. **setPortIdleTime**

Inventory Round 시간 중에서 antenna의 유휴 시간을 설정한다.

➤ **Syntax**

```
public void setPortIdleTime(int value)
```

➤ **Parameters**

value : 유휴 시간

➤ **Remarks**

단위는 밀리초(milliseconds)이다.

2.1.5.30. **getLBTChannel**

LBT Channel 정보를 읽는다.

➤ **Syntax**

```
public void getLBTChannel()
```

➤ **Remarks**

이 메서드는 getModelType 으로 수신된 값이 MT_AT288Japan 일 때만 유효하다.

결과는 IReaderCallbackReceiver의 onReaderProperty의 파라미터로 전달 된다.

Code : PROPERTY_LBT_CHANNEL

2.1.5.31. setLBTChannel

LBT Channel 정보를 설정한다.

➤ Syntax

```
public void setLBTChannel(int value)
```

➤ Parameters

value : LBT Channel 정보

➤ Remarks

다음의 테이블을 참조하여 value 값을 만든다. 아래 테이블에서 bit가 1이면 그에 해당하는 주파수를 사용하고 0이면 사용하지 않는다.

| Bit | 8Bit | 7Bit | 6Bit | 5Bit | 4Bit | 3Bit | 2Bit | 1Bit | 0Bit |
|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Ch No. | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| Freq.(MHz) | 923.4 | 922.8 | 922.2 | 921.6 | 921.0 | 920.4 | 919.2 | 918.0 | 916.8 |

이 메시드는 getModelType 으로 수신된 값이 MT_AT288Japan 일 때만 유효하다.

2.1.5.32. defaultSettings

설정 값을 기본값으로 되돌린다.

➤ Syntax

```
public void defaultSettings()
```

2.1.5.33. saveSettings


변경된 설정 값을 저장한다.

➤ Syntax

```
public void saveSettings()
```

➤ Remarks

AT288N의 전원이 꺼진 후 다시 켜지더라도 마지막 저장한 내용으로 적용되도록 설정한다.

| | | | | | | | |
|---|--|---|------|----|------------|----|--------------|
|  | | AT288N API Reference Guide for Android Developers | | | | | |
| All That Identification | | Android Developer Guide | | | 회사 | | ATID Co.,Ltd |
| 문서이름 | | 작성자 | SW 팀 | 날자 | 2023-06-12 | 버전 | v0.4 |

2.1.6. Extended Property Method

확장된 Tag operation의 속성 및 모듈 속성 설정 기능을 제공한다.

2.1.6.1. getAllExtendedProperty

확장 속성들을 읽는다.

➤ Syntax

```
public void getAllExtendedProperty()
```

➤ Remarks

이 메서드를 호출하면, 다음의 속성 값들이 IReaderCallbackReceiver의 onReaderExtendedProperty의 파라미터로 차례로 전달 된다.

- PROPERTY_EX_POWER_GAIN
- PROPERTY_EX_TAG_TYPE
- PROPERTY_EX_COMMUNICATION_TYPE
- PROPERTY_EX_CONTINUE_MODE
- PROPERTY_EX_STORED_MODE
- PROPERTY_EX_USE_SERIAL_NO

2.1.6.2. getSerialNo

AT288N의 Serial Number를 읽는다.

➤ Syntax

```
public void getSerialNo()
```

➤ Remarks

결과는 IReaderCallbackReceiver의 onReaderExtendedProperty의 파라미터로 전달 된다.
Code : PROPERTY_EX_SERIAL_NO

2.1.6.3. getPowerState

AT288N의 전원 On/Off 상태를 읽는다.

➤ Syntax

```
public void getPowerState()
```

➤ Remarks

결과는 IReaderCallbackReceiver의 onReaderExtendedProperty의 파라미터로 전달 된다.
Code : PROPERTY_EX_POWER_STATE


2.1.6.4. getPowerEx

Antenna 송신 파워 레벨을 읽는다.

➤ Syntax

```
public boolean getPowerEx()
```

➤ Remarks

| | | | | | | | |
|---|--|---|------|----|------------|----|--------------|
|  | | AT288N API Reference Guide for Android Developers | | | | | |
| All That Identification | | Android Developer Guide | | | 회사 | | ATID Co.,Ltd |
| 문서이름 | | 작성자 | SW 팀 | 날자 | 2023-06-12 | 버전 | v0.4 |

결과는 IReaderCallbackReceiver의 onReaderExtendedProperty의 파라미터로 전달 된다.

Code : PROPERTY_EX_POWER_GAIN

2.1.6.5. **setPowerEx**

Antenna 송신 파워 레벨을 설정한다.

➤ **Syntax**

```
public void setPowerEx(int value)
```

➤ **Parameters**

value : 송신 파워 레벨(0~19)

➤ **Remarks**

value가 0이면 최대 출력이고(30dBm) value를 증가 시키면 출력은 1dBm씩 감소한다.

예로, value를 5로 설정했다면 출력되는 파워 레벨은 25dBm이 된다.

2.1.6.6. **getTagType**

AT288N이 인식 하도록 설정된 Tag Type을 읽는다.

➤ **Syntax**

```
public void getTagType()
```

➤ **Remarks**

결과는 IReaderCallbackReceiver의 onReaderExtendedProperty의 파라미터로 전달 된다.

Code : PROPERTY_EX_TAG_TYPE

2.1.6.7. **setTagType**

AT288N이 인식할 Tag Type을 설정한다.

➤ **Syntax**

```
public void setTagType(TagType type)
```

➤ **Parameters**

type : Tag type 설정

➤ **Remarks**

type에 대해서는 2.3.2 TagType 열거형 참조.

ISO18000_6B는 IsAT288N_MA의 반환 값이 true 일 때만 유효하며,

ISO18000_6B로 설정된 후에는, read, write, lock, kill을 사용할 수 없다.


2.1.6.8. **getCommunicationType**

현재 설정되어 있는 Host와 AT288N과의 연결 방식을 읽는다.

➤ **Syntax**

```
public void getCommunicationType()
```

➤ **Remarks**

| | | | | | | | |
|--|---|-----|------|----|------------|--------------|------|
|  All That Identification | AT288N API Reference Guide for Android Developers | | | | | | |
| Android Developer Guide | | | | | 회사 | ATID Co.,Ltd | |
| 문서이름 | | 작성자 | SW 팀 | 날자 | 2023-06-12 | 버전 | v0.4 |

결과는 IReaderCallbackReceiver의 onReaderExtendedProperty의 파라미터로 전달 된다.

Code : PROPERTY_EX_COMMUNICATION_TYPE

2.1.6.9. **setCommunicationType**

Host와 AT288N과의 연결 방식을 설정한다.

➤ **Syntax**

```
public void setCommunicationType(CommType type)
```

➤ **Parameters**

type : 연결 방식

➤ **Remarks**

type에 대해서는 2.3.4 CommType 열거형 참조.

2.1.6.10. **getContinueModeEx**

Inventory type을 읽는다.

➤ **Syntax**

```
public void getContinueModeEx()
```

➤ **Remarks**

결과는 IReaderCallbackReceiver의 onReaderExtendedProperty의 파라미터로 전달 된다.

Code : PROPERTY_EX_CONTINUE_MODE

2.1.6.11. **setContinueModeEx**

Inventory type을 설정한다.

➤ **Syntax**

```
public void setContinueModeEx(boolean enabled)
```

➤ **Parameters**

enabled : Inventory type

➤ **Remarks**

Enabled가 true이면 multiple, false이면 single

2.1.6.12. **setContinueModeExN**

Inventory type을 설정한다.

➤ **Syntax**

```
public void setContinueModeExN(int value)
```

➤ **Parameters**

value : inventory type

➤ **Remarks**

value에 대해서는 2.4 Constants(Continue Mode Type)를 참조.

2.1.6.13. **getStoredMode**

저장 모드의 enable/disable 상태를 읽는다.

➤ **Syntax**

```
public void getStoredMode()
```

➤ **Remarks**

결과는 IReaderCallbackReceiver의 onReaderExtendedProperty의 파라미터로 전달 된다.

Code : PROPERTY_EX_STORED_MODE

2.1.6.14. **setStoredMode**

저장 모드의 enable/disable 상태를 설정한다.

➤ **Syntax**

```
public void setStoredMode(StoredModeType mode)
```

➤ **Parameters**

mode : 저장 모드

➤ **Remarks**

Mode에 대해서는 2.3.7 StoredModeType 열거형 참조.

2.1.6.15. **getStoredData**

AT288N에 저장되어 있는 Tag 리스트를 읽는다.

➤ **Syntax**

```
public void getStoredData()
```

➤ **Remarks**

IReaderCallbackReceiver의 onReaderReadTag의 파라미터로 Tag 데이터가 전달 되고, 데이터 전달이 종료되면 onReaderExtendedProperty 메서드로 종료 되었음을 전달 한다.

Code : PROPERTY_EX_GET_STORED_DATA

2.1.6.16. **deleteStoredData**

AT288N에 저장되어 있는 Tag 리스트를 삭제한다.

➤ **Syntax**

```
public void deleteStoredData()
```

➤ **Remarks**

저장된 데이터 삭제를 시작하고 삭제가 종료 되면 onReaderExtendedProperty 메서드로 종료 되었음을 전달 한다.

Code : PROPERTY_EX_DELETE_STORED_DATA

2.1.6.17. getVersionEx

AT288N의 main board F/W version을 읽는다.

➤ Syntax

```
public void getVersionEx()
```

➤ Remarks

결과는 IReaderCallbackReceiver의 onReaderExtendedProperty의 파라미터로 전달 된다.

Code : PROPERTY_EX_VERSION

2.1.6.18. getInventoryFormat

Inventory의 결과로 수신되는 데이터의 포맷을 읽는다.

➤ Syntax

```
public void getInventoryFormat()
```

➤ Remarks

결과는 IReaderCallbackReceiver의 onReaderExtendedProperty의 파라미터로 전달 된다.

Code : PROPERTY_EX_INVENTORY_FORMAT

2.1.6.19. setInventoryFormat

Inventory의 결과로 수신되는 데이터의 포맷을 설정한다.

➤ Syntax

```
public void setInventoryFormat(InventoryFormatType type)
```

➤ Parameters

type : inventory format type

➤ Remarks

type에 대해서는 2.3.5 InventoryFormatType 열거형 참조.

2.1.6.20. getUseSerialNo

Inventory의 결과로 수신되는 데이터에 AT288N의 Serial number 포함 여부를 읽는다.

➤ Syntax

```
public void getUseSerialNo()
```

➤ Remarks

결과는 IReaderCallbackReceiver의 onReaderExtendedProperty의 파라미터로 전달 된다.


Code : PROPERTY_EX_USE_SERIAL_NO

2.1.6.21. setUseSerialNo

Inventory의 결과로 수신되는 데이터에 AT288N의 Serial number 포함 여부를 설정한다.

➤ Syntax

```
public void setUseSerialNo(boolean enabled)
```

| | | | | | | | |
|--|---|-----|------|----|------------|--------------|------|
|  All That Identification | AT288N API Reference Guide for Android Developers | | | | | | |
| Android Developer Guide | | | | | 회사 | ATID Co.,Ltd | |
| 문서이름 | | 작성자 | SW 팀 | 날자 | 2023-06-12 | 버전 | v0.4 |

➤ **Parameters**

enabled : true(serial number 포함), false(serial number 포함하지 않음)

.

2.1.6.22. **getBatteryState**

AT288N의 배터리 잔량을 읽는다.

➤ **Syntax**

```
public void getBatteryState()
```

➤ **Remarks**

결과는 IReaderCallbackReceiver의 onReaderExtendedProperty의 파라미터로 전달 된다.

Code : PROPERTY_EX_BATTERY_STATE

2.1.6.23. **IsAT288N**

Host에 연결된 AT288이 New 버전인지를 반환한다.

➤ **Syntax**

```
public boolean IsAT288N()
```

➤ **Return value**

AT288N이면 true, AT288이면 false

2.1.6.24. **IsAT288N_MA**


Host에 연결된 AT288이 New 버전 MA 타입인지를 반환한다.

➤ **Syntax**

```
public boolean IsAT288N_MA()
```

➤ **Return value**

AT288N MA 버전이면 true, 그렇지 않으면 false

| | | | | | | | |
|--|--|---|------|----|------------|--------------|------|
|  All That Identification | | AT288N API Reference Guide for Android Developers | | | | | |
| Android Developer Guide | | | | | 회사 | ATID Co.,Ltd | |
| 문서이름 | | 작성자 | SW 팀 | 날자 | 2023-06-12 | 버전 | v0.4 |

2.1.7. Static Method

2.1.7.1. **getResponses**

이벤트 핸들러로 수신된 code를 문자열로 변환한다.

➤ **Syntax**

```
public static String getResponses(String code)
```

➤ **Parameters**

code : 이벤트 핸들러로 수신된 code

➤ **Return value**

성공하면 code에 대응 하는 결과 문자열을 반환하고 실패하면 공백 문자열을 반환한다.

2.1.7.2. **isBluetoothSupported**


Reader class 객체가 실행중인 Host의 bluetooth 지원 여부를 반환한다.

➤ **Syntax**

```
public static boolean isBluetoothSupported()
```

➤ **Return value**

Bluetooth가 지원되는 Host이면 true, 그렇지 않으면 false를 반환한다.

| | | | | | | | |
|---|--|---|------|----|------------|----|--------------|
|  | | AT288N API Reference Guide for Android Developers | | | | | |
| All That Identification | | Android Developer Guide | | | 회사 | | ATID Co.,Ltd |
| 문서이름 | | 작성자 | SW 팀 | 날자 | 2023-06-12 | 버전 | v0.4 |

2.2. IReaderCallbackReceiver Interface

AT288N에서 발생한 이벤트를 수신하기 위해서 App에서는 IReaderCallbackReceiver를 구현한 후, Reader 클래스의 생성자로 전달해야 한다.

2.2.1. Method

2.2.1.1. onReaderStateChanged

AT288N과 Host와의 접속 상태가 변경되었을 때 실행 된다.

➤ Syntax

```
void onReaderStateChanged(int state);
```

➤ Parameters

state : 접속 상태

➤ Remarks

state에 대해서는 2.4 Constants(Connection State)를 참조.

2.2.1.2. onReaderReadTag

Tag의 데이터를 읽었을 때 실행 된다.

➤ Syntax

```
void onReaderReadTag(int event, String tag);
```

➤ Parameters

event : event 종류

tag : Tag 데이터

➤ Remarks

event에 대해서는 2.4 Constants(Event Type)를 참조.

2.2.1.3. onReaderResponse

Access 메서드들이 수행 완료 또는 중단 되었을 때 실행 된다.

➤ Syntax

```
void onReaderResponse(int event, String code);
```

➤ Parameters

event : event 종류

code : 수행 결과.

➤ Remarks

event에 대해서는 2.4 Constants(Event Type)를 참조.

code로 수신되는 값은 16진수 문자열이지만, 는 getResponses 메서드를 사용해서 string 형태의 응답 문자열로 변환할 수 있다. 아래 테이블 참조.

| code | string | Description |
|------|--------|-------------|
|------|--------|-------------|

| | | |
|------|-------------------|--------------------|
| "00" | OtherError | 알 수 없는 이유로 인한 에러 |
| "01" | Success | 결과 성공 |
| "02" | Undefined | 정의 되지 않은 에러 |
| "03" | MemoryOverrun | 접근 가능한 메모리 범위를 벗어남 |
| "04" | MemoryLocked | 메모리가 잠겨 있음 |
| "05" | Timeout | 접근 허용 시간 초과 |
| "0B" | InsufficientPower | 전원 부족 |
| "0F" | NonSpecificError | 정의되지 않은 에러. |
| "FF" | AckHeartbeat | 치명적인 통신 에러 |

2.2.1.4. onReaderProperty

속성 값 반환 요청을 받았을 때 실행 된다.

➤ Syntax

```
void onReaderResponse(char code, String value);
```

➤ Parameters

code : 속성 값 타입

value : 속성 값

➤ Remarks

code에 대해서는 2.4 Constants(Property Codes)를 참조.

2.2.1.5. onReaderExtendedProperty

확장 속성 값 반환 요청을 받았을 때 실행 된다.

➤ Syntax

```
void onReaderExtendedProperty(char code, String value);
```

➤ Parameters

code : 확장 속성 값 타입

value : 속성 값

➤ Remarks

code에 대해서는 2.4 Constants(Extended Property Codes)를 참조.

2.2.1.6. onReaderActionCode

Action 메서드를 호출 할 때 실행되며, 호출된 메서드에 해당하는 action type을 받는다.

➤ Syntax

```
void onReaderActionCode(char action);
```

➤ Parameters

action : action type

➤ **Remarks**

action에 대해서는 2.4 Constants(Action Codes)를 참조.

2.2.1.7. **onReaderTimeout**

Access 메시드가 setResponseTime 으로 설정된 시간 내에 실행 완료 되지 못했을 때 실행 된다.

➤ **Syntax**

```
void onReaderTimeout();
```

➤ **Remarks**

code에 대해서는 2.4 Constants(Extended Property Codes)를 참조.

2.2.1.8. **onReaderMessage**


Bluetooth 연결 실패 또는 연결이 끊어졌을 때 실행된다.

➤ **Syntax**

```
void onReaderMessage(String msg);
```

➤ **Parameters**

msg : 에러 문자열

| | | | | | | | |
|---|--|---|------|----|------------|----|--------------|
|  | | AT288N API Reference Guide for Android Developers | | | | | |
| All That Identification | | Android Developer Guide | | | | 회사 | ATID Co.,Ltd |
| 문서이름 | | 작성자 | SW 팀 | 날자 | 2023-06-12 | 버전 | v0.4 |

2.3. Enumeration

2.3.1. ModelType

AT288N Device 타입을 정의 한다.

| 플래그 | 값 | 설명 |
|------------|---|----------|
| None | 0 | Unknown |
| AT288 | 1 | Standard |
| AT288Japan | 2 | Japanese |

2.3.2. TagType

AT288N으로 인식할 Tag 타입을 정의한다.

| 플래그 | 값 | 설명 |
|------------------|---|-----------------------|
| ISO18000_6B | 0 | ISO 18000-6B Tag |
| ISO18000_6C_GEN1 | 1 | 사용하지 않음 |
| ISO18000_6C_GEN2 | 2 | ISO 18000-6C GEN2 Tag |

2.3.3. BankType

Tag의 Memory Bank를 정의한다.

| 플래그 | 값 | 설명 |
|---------|---|-----------------|
| Reserve | 0 | Reserved memory |
| EPC | 1 | EPC memory |
| TID | 2 | TID memory |
| User | 3 | User memory |

2.3.4. CommType


AT288N과 Host 간의 연결 방법을 정의한다.

| 플래그 | 값 | 설명 |
|-----------|---|-------------------|
| Bluetooth | 0 | Bluetooth를 사용한 연결 |
| USB | 1 | USB를 사용한 연결 |

2.3.5. InventoryFormatType

Inventory 결과의 Format을 정의한다.

| 플래그 | 값 | 설명 |
|---------------|---|--------------------------|
| PC_EPC | 0 | PC + EPC |
| SERIAL_PC_EPC | 1 | Serial Number + PC + EPC |

| | | | | | | | |
|---|--|---|------|----|------------|----|--------------|
|  | | AT288N API Reference Guide for Android Developers | | | | | |
| All That Identification | | Android Developer Guide | | | 회사 | | ATID Co.,Ltd |
| 문서이름 | | 작성자 | SW 팀 | 날자 | 2023-06-12 | 버전 | v0.4 |

| | | |
|------------|---|---------------------|
| ONLY_EPC | 2 | EPC |
| SERIAL_EPC | 3 | Serial Number + EPC |

2.3.6. SelectionActionType


Action 메서드에서 mask 값의 비교 방법을 정의한다.

| 플래그 | 값 | 설명 |
|-------------|---|----------------------------|
| Matching | 0 | Mask가 Tag와 일치하는 경우 동작 함 |
| NonMatching | 1 | Mask가 Tag와 일치하지 않는 경우 동작 함 |

2.3.7. StoredModeType

Stored mode의 동작 방식을 정의 한다.

| 플래그 | 값 | 설명 |
|----------------|---|--------------|
| NotStored | 0 | 데이터를 저장하지 않음 |
| Stored | 1 | 데이터를 저장 함 |
| StoredSerialNo | 2 | 사용하지 않음 |

| | | | | | | | |
|---|--|---|------|----|------------|----|--------------|
|  | | AT288N API Reference Guide for Android Developers | | | | | |
| All That Identification | | Android Developer Guide | | | 회사 | | ATID Co.,Ltd |
| 문서이름 | | 작성자 | SW 팀 | 날자 | 2023-06-12 | 버전 | v0.4 |

2.4. Constant

AT288N Device 타입을 정의 한다.

| 플래그 | 값 | 설명 |
|---------------------------|----|---|
| Model Type | | |
| MT_NONE | 0 | 모델 타입을 알 수 없음 |
| MT_AT288 | 1 | Standard AT288N |
| MT_AT288Japan | 2 | Japanese AT288N |
| Connection State | | |
| STATE_NONE | 0 | 아무런 이벤트가 발생하지 않음 |
| STATE_LISTEN | 1 | 외부로부터의 연결 대기 중 |
| STATE_CONNECTING | 2 | 연결 진행 중 |
| STATE_CONNECTED | 3 | 연결 완료 |
| Event Type | | |
| ET_NONE | 0 | 아무런 이벤트가 발생하지 않음 |
| ET_TIMEOUT | 3 | 명령에 대한 시간 초과 |
| ET_INVENTORY | 5 | Inventory 작업 중 |
| ET_READMEMORY | 6 | Read Memory 작업 중 |
| ET_WRITEMEMORY | 7 | Write Memory 작업 중 |
| ET_LOCK | 8 | Lock 작업 중 |
| ET_KILL | 9 | Kill 작업 중 |
| ET_GETPROPERTIES | 10 | 속성이 반환됨 |
| ET_COMMAND | 11 | 명령을 처리 중 |
| Tag Type | | |
| ISO18000_6B | 0 | ISO 18000-6B Tag |
| ISO18000_6C_GEN1 | 1 | ISO 18000-6C GEN1 Tag |
| ISO18000_6C_GEN2 | 2 | ISO 18000-6C GEN2 Tag |
| Continue Mode Type | | |
| CT_MULTIPLE | 0 | 다중 Tag 읽기 방식 |
| CT_SINGLE | 1 | 단일 Tag 읽기 방식 |
| CT_FILTERING | 2 | 다중 Tag 읽기 방식이지만, 같은 Tag를 두 번 이상 읽지 않는다. |
| Memory Bank Type | | |
| MT_RESERVED | 0 | Reserved memory |
| MT_EPC | 1 | EPC memory |

| MT_TID | 2 | TID memory | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------------------------|-----|---|-------|--|-------------|----|----|--|---|---|-------|---|---|--------|---|---|-----|---|---|-------|---|---|--------|---|---|--------|---|---|----------|---|---|------|--|---|----------|--|---|-------------|--|----|-------|--|----|-----------|--|----|-------------|--|----|--------|--|----|-----------|--|----|------------|--|----|-------------|
| MT_USER | 3 | User memory | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Inventory Format Type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PC_EPC | 0 | PC + EPC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SERIAL_PC_EPC | 1 | Serial Number + PC + EPC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ONLY_EPC | 2 | EPC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SERIAL_EPC | 3 | Serial Number + EPC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Property Codes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PROPERTY_BUZZER | 'b' | Buzzer 모드 value: 0(Disable), 1(Enable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PROPERTY_CONTINUE_MODE | 'c' | Inventory 기능 수행 시 Tag를 연속으로 읽을 것인지 한번만 읽을 것인지 결정 value: 0(Disable), 1(Enable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PROPERTY_ANTENNA_STATUS | 'e' | 사용하지 않음 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PROPERTY_GLOBAL_BAND | 'f' | 국가별 UHF 주파수 대역 <table border="1"> <thead> <tr> <th colspan="2">value</th><th>Description</th></tr> <tr> <th>MA</th><th>MI</th><th></th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>Korea</td></tr> <tr><td>2</td><td>1</td><td>Europe</td></tr> <tr><td>3</td><td>2</td><td>USA</td></tr> <tr><td>4</td><td>3</td><td>China</td></tr> <tr><td>5</td><td>4</td><td>Taiwan</td></tr> <tr><td>6</td><td>5</td><td>Brazil</td></tr> <tr><td>7</td><td>6</td><td>Malaysia</td></tr> <tr><td>8</td><td>7</td><td>Asia</td></tr> <tr><td></td><td>8</td><td>Japan 1W</td></tr> <tr><td></td><td>9</td><td>Japan 250mW</td></tr> <tr><td></td><td>10</td><td>India</td></tr> <tr><td></td><td>11</td><td>Indonesia</td></tr> <tr><td></td><td>12</td><td>Japan 125mW</td></tr> <tr><td></td><td>13</td><td>Israel</td></tr> <tr><td></td><td>14</td><td>Australia</td></tr> <tr><td></td><td>15</td><td>Newzealand</td></tr> <tr><td></td><td>16</td><td>Philippines</td></tr> </tbody> </table> | value | | Description | MA | MI | | 0 | 0 | Korea | 2 | 1 | Europe | 3 | 2 | USA | 4 | 3 | China | 5 | 4 | Taiwan | 6 | 5 | Brazil | 7 | 6 | Malaysia | 8 | 7 | Asia | | 8 | Japan 1W | | 9 | Japan 250mW | | 10 | India | | 11 | Indonesia | | 12 | Japan 125mW | | 13 | Israel | | 14 | Australia | | 15 | Newzealand | | 16 | Philippines |
| value | | Description | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MA | MI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | Korea | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 1 | Europe | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | 2 | USA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | 3 | China | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | 4 | Taiwan | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | 5 | Brazil | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | 6 | Malaysia | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | 7 | Asia | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 8 | Japan 1W | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 9 | Japan 250mW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 10 | India | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 11 | Indonesia | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 12 | Japan 125mW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 13 | Israel | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 14 | Australia | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 15 | Newzealand | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 16 | Philippines | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | | |
|----------------------------------|-----|---|----|-------------|
| | | | 17 | Singapore |
| | | | 18 | Thailand |
| | | | 19 | Uruguay |
| | | | 20 | Vietnam |
| | | | 21 | SouthAfrica |
| PROPERTY_PACKET_OPTION | 'i' | 사용하지 않음 | | |
| PROPERTY_ANTENNA_SWITCHING_COUNT | 'k' | 사용하지 않음 | | |
| PROPERTY_ANTENNA_SWITCHING_TIME | 'j' | Inventory 기능 수행 시 Antenna가 활성화되는 시간(단위 : ms) | | |
| PROPERTY_POWER_IDLE_TIME | 'o' | Inventory 기능 수행 시 Antenna의 유희시간(단위 : ms) | | |
| PROPERTY_POWER_GAIN | 'p' | Antenna 송신 파워 레벨. Value : 0 ~ 19 | | |
| PROPERTY_START_Q_VALUE | 'q' | Q 알고리즘의 시작 값 Value: 0 ~ 15 | | |
| PROPERTY_VERSION | 'v' | RFID Module F/W 버전 | | |
| PROPERTY_ACCESS_PASSWORD | 'w' | Access에 필요한 비밀번호 | | |
| PROPERTY_READER_MODE | 'x' | 사용하지 않음 | | |
| PROPERTY_AUTO_COMMAND | 'y' | 사용하지 않음 | | |
| PROPERTY_SELECTION_BANK | 'g' | Masking Selection 기능에서 Mask을 수행할 Tag의 메모리 Bank | | |
| PROPERTY_SELECTION_OFFSET | ',' | Masking Selection에서 Mask값을 비교하기 시작할 주소 (단위 : bit) | | |
| PROPERTY_SELECTION_ACTION | '8' | Masking Selection 기능에서 Mask값에 대한 동작을 정의 | | |
| PROPERTY_SELECTION_SESSION | 's' | Masking Selection에서 Mask값을 비교할 Inventory Session Value: 0(S0), 1(S1), 2(S2), 3(S3) | | |
| PROPERTY_MIN_Q_VALUE | 'l' | Q 알고리즘의 최소값 Value: 0 ~ 15 | | |
| PROPERTY_MAX_Q_VALUE | 'j' | Q 알고리즘의 최대값 Value: 0 ~ 15 | | |
| PROPERTY_LBT_CHANNEL | 'B' | LBT Channel 설정 값 | | |
| PROPERTY_PORT_ACTIVE | 'e' | 사용하지 않음 | | |
| PROPERTY_AUTO_POWEROFF | 'p' | AT288N이 off 될때 까지의 시간 | | |

| | | |
|--------------------------------|-----|---|
| | | (단위 : minute) |
| Extended Property Codes | | |
| PROPERTY_EX_POWER_STATE | '1' | AT288N의 전원 On/Off 상태 |
| PROPERTY_EX_POWER_GAIN | '2' | Antenna 송신 파워 레벨 Value : 0 ~ 19 |
| PROPERTY_EX_TAG_TYPE | '3' | AT288N이 인식하는 Tag Type |
| PROPERTY_EX_COMMUNICATION_TYPE | '4' | AT288N과 Host와의 통신 방법 |
| PROPERTY_EX_CONTINUE_MODE | '5' | Inventory Type |
| PROPERTY_EX_STORED_MODE | '6' | Tag 데이터 저장 모드 |
| PROPERTY_EX_USE_SERIAL_NO | '8' | Inventory 데이터에 Serial number 포함 여부 |
| PROPERTY_EX_BATTERY_STATE | '9' | AT288N의 배터리 상태 Value: 0(high), 1(low) |
| PROPERTY_EX_GET_STORED_DATA | 'a' | AT288N에 저장된 Tag 리스트를 Host로 전송 완료. |
| PROPERTY_EX_DELETE_STORED_DATA | 'b' | AT288N에 저장된 Tag 리스트를 삭제 완료. |
| PROPERTY_EX_NATIONAL_CODE | 'n' | 사용하지 않음 |
| PROPERTY_EX_SERIAL_NO | 's' | AT288N의 Serial Number |
| PROPERTY_EX_VERSION | 'v' | AT288N의 mainboard F/W 버전 |
| PROPERTY_EX_INVENTORY_FORMAT | '8' | Inventory 데이터의 형태 |
| Action Codes | | |
| ACTION_INVENTORY_6B_MULTIPLE | 'b' | 6B Type의 Tag에 대하여 Multiple 방식으로 Inventory를 시작함 |
| ACTION_INVENTORY_6B_SINGLE | 'a' | 6B Type의 Tag에 대하여 Single 방식으로 Inventory를 시작함 |
| ACTION_INVENTORY_6C_MULTIPLE | 'f' | 6C Type의 Tag에 대하여 Multiple 방식으로 Inventory를 시작함 |
| ACTION_INVENTORY_6C_SINGLE | 'e' | 6C Type의 Tag에 대하여 Single 방식으로 Inventory를 시작함 |
| ACTION_INVENTORY_6C_SELECTION | 'd' | 6C Type의 Tag에 대하여 Tag Selection 방식으로 Inventory를 시작함 |
| ACTION_INVENTORY_6C_VLC | 'j' | |
| ACTION_READ_MEMORY | 'r' | Tag Memory에서 데이터를 읽기 |

| | | |
|---------------------|-----|-------------------------|
| | | 시작함 |
| ACTION_WRITE_MEMORY | 'w' | Tag Memory에 데이터를 쓰기 시작함 |
| ACTION_LOCK | 'l' | Tag를 잠금 |
| ACTION_KILL | 'k' | Tag를 폐기 |
| ACTION_STOP | '3' | 동작이 중지됨 |
| | | |