



ATID Co.,Ltd

# AT288N SDK Reference Guide for iPhone Developers

AT288N SDK Reference Guide

SW 팀

2023-06-12

## 개정 이력

버전	개정일자	개정사유 <sup>1</sup>	개정내역 <sup>2</sup>	작성자
v0.1	2017. 8. 31.	초안		류은주
v0.2	2019. 1. 22.	추가	typeBC/ scanTime 추가	류은주
v0.3	2023-06-12	수정	개발 툴 버전 수정	SW 팀

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

<sup>2</sup> 개정내역 : 개정이 발생하는 페이지 번호와 변경 내용을 기술

## Content

1. Intro.....	7
2. EAREader Class.....	8
2.1. Initialize Methods.....	8
2.1.1. initWithDevice.....	8
2.1.2. Disconnect.....	9
2.2. Reader Properties.....	10
2.2.1. peripheral.....	10
2.2.2. getAction.....	10
2.2.3. setDelegate .....	10
2.3. Action Methods.....	11
2.3.1. Inventory .....	11
2.3.2. readMemory.....	11
2.3.3. writeMemory .....	12
2.3.4. lock .....	12
2.3.5. stop.....	13
2.3.6. getSaveData.....	13
2.3.7. deleteSaveData.....	13
2.4. Device Properties.....	14
2.4.1. firmwareVersion.....	14
2.4.2. buzzer.....	14
2.4.3. continuousMode .....	14
2.4.4. powerGain .....	15
2.4.5. PowerOff .....	15

2.4.6.	IdleTime .....	15
2.4.7.	autoOffTime.....	16
2.4.8.	accessPassword.....	16
2.4.9.	inventorySession.....	16
2.4.10.	globalBand .....	17
2.4.11.	autosaveMode.....	18
2.4.12.	inventoryFormat.....	18
2.4.13.	lowBatteryState .....	18
2.4.14.	selectAction.....	19
2.4.15.	SelectBitPtr .....	19
2.4.16.	SelectBank .....	19
2.4.17.	singleMulti.....	20
2.4.18.	typeBC .....	20
2.4.19.	scanTime.....	20
3.	EADevice BluetoothLe Class.....	21
3.1.	Initialize Methods.....	21
3.1.1.	initWithPeripheral .....	21
3.2.	Properties.....	22
3.2.1.	delegate.....	22
3.2.2.	name .....	22
3.2.3.	address.....	22
3.2.4.	peripheral.....	23
3.3.	Methods.....	24
3.3.1.	disconnect .....	24

3.3.2.	writeData.....	24
4.	LockParam.....	25
4.1.	Properties.....	25
4.1.1.	killPassword .....	25
4.1.2.	accessPassword.....	25
4.1.3.	epc .....	25
4.1.4.	tid.....	26
4.1.5.	user.....	26
5.	EAResultType.....	27
5.1.	Methods.....	27
5.1.1.	msg.....	27
6.	EAResultData.....	28
6.1.	Properties.....	28
6.1.1.	result .....	28
6.1.2.	mData.....	28
7.	Deletagate Interfaces .....	29
7.1.	EAReaderDelegate .....	29
7.1.1.	readerInitialized.....	29
7.1.2.	updateDeviceState.....	29
7.1.3.	readTagResult .....	30
7.1.4.	changedActionState.....	30
7.1.5.	tagAccessResult .....	31
7.1.6.	opmodesetting.....	31
7.1.7.	powergainchange.....	32

7.1.8.	memoryaccessresult.....	32
8.	Enumerators .....	33
8.1.	BankType.....	33
8.2.	BuzzerState.....	33
8.3.	SessionType.....	33
8.4.	CommandType .....	34

## 1. Intro

본 문서는 AT288 SDK Library 를 사용하여 응용 프로그램을 개발하고자 하는 iOS 개발자들을 위하여 SDK Library 를 사용할 수 있는 SDK Library 사용 방법을 기술 하는데 그 목적이 있다.

본 문서에서 사용되는 개발 도구는 **Xcode 14.2** 이 사용되었고 개발 대상 플랫폼은 iOS 10.3 이상을 지원한다.

## 2. EReader Class

EReader Class 는 iPhone 에서 AT288N 장비와 Programmable Interface 를 제공한다.  
AT288N 장비와 iPhone 에 대한 접속 방법은 iOS SDK 로 제공되는 CoreBluetooth framework 를 사용한다.

### 2.1. Initialize Methods

일반적인 CocoaTouch Class 들 처럼 EReader 클래스도 초기화를 위한 함수를 제공한다.  
전통적으로 init 로 시작되는 초기화 함수로서 다음 initWithDevice 함수를 초기화로 제공한다.

#### 2.1.1. initWithDevice

iPhone 에서 AT288N BLE 를 사용하여 통신하는 EReader 개체를 생성하기 위한 메서드이다.

##### Syntax

```
- (id)initWithDevice:(EADevice *)device delegate:(id<EADelegate>)delegate;
```

##### Parameters

**device** : 이미 접속이 완료된 CoreBluetooth framework 의 CPeripheral Object 를 포함하고 있는 접속 정보를 관리하는 EADevice BluetoothLe Class 의 Instance 를 지정한다.

**delegate** : EReader 의 상태 변경 등의 이벤트를 통한 AT288N 장비로부터 정보 수신을 목적으로 하는 delegate protocol interface 를 지정한다.

##### Remarks

반드시 접속이 완료된 CPeripheral 개체가 포함되어 있는 EADevice BluetoothLe Class 의 Instance 를 지정하여야 한다. BLE 의 접속은 CoreBluetooth Framework 를 사용하고, 접속 이 후, AT288N 장비의 초기화 부터는 본 SDK 를 사용하여 통신한다.



## 2.1.2. Disconnect

iPhone 과 접속되어 있는 AT288N 장비와의 연결을 종료한다.

### Syntax

```
- (void)disconnect;
```

### Remarks

disconnect Method 는 내부적으로 CBPeripheral 의 disconnect 를 호출한다. 이 Method 가 호출되고 나면 EADevice BluetoothLe Instance 로 포함되어 있는 CBPeripheral Instance 를 사용할 수 없도록 무효화 된다.

## 2.2. Reader Properties

### 2.2.1. peripheral

AT288N 장비에 연결하기 위해 사용된 CBPeripheral Class 의 Instance 를 반환한다.

#### Syntax

```
- (CBPeripheral *)peripheral;
```

#### Remarks

AT288N 장비와 연결 상태를 확인하기 위한 Interface 는 CoreBluetooth framework 를 참조하도록 한다.

### 2.2.2. getAction

EAREader Instance 의 현재 동작 상태를 반환한다.

#### Syntax

```
- (CommandType)getAction;
```

#### Remarks

getAction Method 는 EAREader 가 AT288N 장비에 마지막 전송한 Action Command 를 반환하므로써 현재 EAREader 가 AT288N 장비에 어떤 작업을 명령하였는지, 현재 어떤 작업을 수행 중인지를 알 수 있도록 한다.

### 2.2.3. setDelegate

EAREader 의 Instance 로부터 이벤트를 수신할 수신 대상을 변경한다.

#### Syntax

```
- (void)setDelegate:(id<EAREaderDelegate>)delegate;
```

#### Parameters

**delegate** : 이벤트를 수신할 수 있는 EAREaderDelegate Protocol 을 구현한 Class 의 Instance 를 지정한다.

#### Remarks

EAREader Instance 로부터 이벤트를 수신하는 대상 개체를 변경하고자 할 경우, setDelegate Method 를 사용할 수 있다.

## 2.3. Action Methods

### 2.3.1. Inventory

AT288N 장비의 상태에 따라 알맞은 Inventory 기능을 자동으로 수행한다.

#### Syntax

```
- (ResultType)inventory;
```

#### Return

명령의 수행 결과를 ResultType 열거형으로 반환한다.

#### Remarks

inventory 메서드를 정상적으로 동작하면 NoError 를 반환한다. Inventory 메서드가 동작 중에는 다른 메서드를 사용할 수 없으므로 stop 메서드를 사용하여 동작을 중지 시키고 다른 메서드를 호출하여야 한다.

inventory 명령으로 AT288N 장비가 Tag 을 읽으면 EAREaderDelegate Protocol 의 readTagResult Method 가 호출된다.

### 2.3.2. readMemory

readMemory 는 Tag 의 Memory 를 직접 Read 하기 위한 명령어이다.

#### Syntax

```
- (ResultType)readMemory:(BankType)bank offset:(int)offset length:(int)length  
mask:(NSString *)mask;
```

#### Parameters

**bank** : 읽을 데이터의 Memory Bank 를 설정한다.

**offset** : 읽을 데이터의 시작 주소를 설정한다. (WORD 단위)

**length** : 읽을 데이터 길이를 설정한다.(WORD 단위)

**mask**: mask 설정할 string 을 입력한다.

#### Return

명령의 수행 결과를 ResultType 열거형으로 반환한다.

#### Remarks

readMemory 메서드를 정상적으로 동작하면 NoError 를 반환한다. readMemory 메서드가 동작 중에는 다른 메서드를 사용할 수 없으므로 stop 메서드를 사용하여 동작을 중지 시키고 다른 메서드를 호출하여야 한다.

## 2.3.3. writeMemory

writeMemory 는 Tag 의 Memory 를 직접 Write 하기 위한 명령어이다.

### Syntax

```
- (ResultType)writeMemory:(BankType)bank offset:(int)offset value:(NSString *)value
mask:(NSString *)mask;
```

### Parameters

**bank** : 데이터 쓰기를 수행할 태그의 Memory Bank 를 설정한다.

**offset** : 데이터가 쓰여지기 시작할 시작 주소를 설정한다.(WORD 단위)

**data** : 쓰여질 데이터를 Hex 값으로 입력한다.

**mask**: mask 설정할 string 을 입력한다.

### Return

명령의 수행 결과를 ResultType 열거형으로 반환한다.

### Remarks

writeMemory 메서드를 정상적으로 동작하면 NoError 를 반환한다. writeMemory 메서드가 동작 중에는 다른 메서드를 사용할 수 없으므로 stop 메서드를 사용하여 동작을 중지 시키고 다른 메서드를 호출하여야 한다.

## 2.3.4. lock

lock 메서드는 Tag Memory 를 접근할 수 없도록 Lock/unlock 때 사용하는 메소드이다.

### Syntax

```
- (ResultType)lock:(LockParam *)param tagmask:(NSString *)tagmask;
```

### Parameters

**param**: Memory Bank 의 잠금 옵션 **tagmask**: mask 설정할 string 을 입력한다.

### Return

명령의 수행 결과를 ResultType 열거형으로 반환한다.

### Remarks

Lock 의 수행 결과는 memoryaccessresult 이벤트로 반환되며, Lock 메서드가 성공하면 1 의 값으로 반환 되며, 실패하면 1 이 아닌 값으로 반환된다.

이벤트 핸들러가 호출되면 Lock 동작은 자동으로 중지되며, Lock 을 수행 중에 취소하고 싶다면 stop 메서드를 호출하면 된다.

## 2.3.5. stop

stop 메서드는 모든 Action 계열의 동작을 중지 시킨다.

### Syntax

```
- (ResultType)stop;
```

### Return

명령의 수행 결과를 ResultType 열거형으로 반환한다.

### Remarks

AT288N 장비가 동작 중인 모든 작업을 취소하고 상태를 중지 상태로 변경한다.

## 2.3.6. getSaveData

getSaveData 메서드는 AT288N 장비에 저장되어 있는 Tag readTagResult 이벤트로 반환한다.

### Syntax

```
- (void)getSaveData;
```

### Remarks

AT288N 장비로부터 readTagResult 이벤트가 발생하며, 메모리에 저장된 모든 태그가 반환되면 changeActionState 로 Stop 이 반환된다.

## 2.3.7. deleteSaveData

deleteSaveData 메서드는 AT288N 장비의 내부 메모리에 저장되어 있는 모든 Data List 를 삭제한다.

### Syntax

```
- (BOOL)deleteSaveData;
```

### Return

삭제가 완료되면 Reply 패킷을 응답한다.

### Remarks

AT288N 장비 내부 메모리에 저장된 모든 태그가 반환되면 changeActionState 로 Stop 이 반환된다.

## 2.4. Device Properties

### 2.4.1. firmwareVersion

getFirmwareVersion 메서드는 AT288N 장비의 Firmware 버전을 문자열로 반환한다.

#### Syntax

```
- (NSString *)firmwareVersion;
```

#### Remarks

AT288N 장비로부터 Firmware 버전이 정상적으로 반환되면 버전 문자열이 반환된다.

### 2.4.2. buzzer

buzzer 속성은 AT288N 장비의 Buzzer 상태를 설정하거나 반환한다.

#### Syntax

```
@property (nonatomic, assign) BuzzerState buzzer;
```

#### Type

AT288N 장비의 Buzzer 을 상태를 설정하는 BuzzerState 열거형

#### Remarks

AT288N 장비의 Buzzer 상태를 소리 없음, 소리 있음으로 설정할 수 있다.

### 2.4.3. continuousMode

continuousMode 속성은 AT288N 장비가 Inventory 기능을 수행할 때, Tag 를 연속적으로 읽을 것인지 한번만 읽을 것인지 여부를 반환하거나 설정한다.

#### Syntax

```
@property (nonatomic, assign) BOOL continuousMode;
```

#### Remarks

Inventory 동작 시에 Tag 를 연속 적으로 읽을 것인지 아닌지를 설정하거나 반환한다. 이 값이 YES 이면 Continuous Mode 가 설정된 상태이고, NO 이면 설정되지 않은 상태이다.

## 2.4.4. powerGain

powerGain 메서드는 AT288N 장비의 Power Gain 을 반환하거나 설정한다.

### Syntax

```
@property (nonatomic, assign) int powerGain;
```

### Remarks

powerGain 속성은 정수형으로 값은 파워출력값의 \*10 을 한 값이 설정되거나 반환된다.  
예를 들어 안테나 파워가 30dBm 이라면 300 이라는 값을 설정하거나 반환된다.

## 2.4.5. PowerOff

powerOff 메서드는 AT288N 장비의 전원 상태를 반환하거나 설정한다.

### Syntax

```
@property (nonatomic, assign) BOOL powerOff;
```

### Remarks

AT288 MI Device 가 Host 와 연결중인 경우, AT288 R2000 Device 의 파워를 끄면 0 값이 반환된다. 파워가 켜져 있는 경우는 1 이 반환된다. 설정은 0 만 가능하다.

## 2.4.6. IdleTime

idleTime 속성은 AT288N 장비가 Inventory 수행할 때, 유휴 시간을 반환하거나 설정한다.

### Syntax

```
@property (nonatomic, assign) int idleTime;
```

### Remarks

여러 대의 장비가 운영 되는 환경에서 Reader 간의 간섭을 최소화하고 오랜 시간 동안 Full Time 동작으로 인한 Reader 의 과열을 방지를 위해 사용한다. Reader 는 RF Channel Hopping 시 Power Idle Time(ms) 동안 RF Power Off 되었다가 Inventory Time(ms)동안 On 된다.

## 2.4.7. autoOffTime

autoOffTime 속성은 AT288N 장비가 아무런 동작을 하지 않을 경우, 자동으로 Off되는 시간을 설정하거나 반환한다.

### Syntax

```
@property (nonatomic, assign) int autoOffTime;
```

### Remarks

AT288N 장비의 Battery 사용 시간 절약을 위하여 일정 시간 사용하지 않으면 자동으로 Reader 가 꺼지도록 설정하는 기능이다. 설정 단위는 min 이다. Auto Off Time 기능을 사용하지 않으려면 이 값을 0 으로 설정한다.

## 2.4.8. accessPassword

AT288N 장비가 Lock 이 걸린 Tag 에 Access Command 를 수행할 때, 필요한 Access Password 를 설정하거나 반환한다.

### Syntax

```
@property (nonatomic, strong) NSString *accessPassword;
```

### Remarks

Access Password가 0으로 설정되는 Read/Write/Lock Command 실행 시 Access Password를 사용하지 않으며, 8Byte 단위의 Password가 설정되어 있으면 Access Password를 사용하게 된다.

## 2.4.9. inventorySession

AT288N 장비가 Inventory 를 수행할 때, 접근할 Tag 의 Session 을 설정하거나 반환한다.

### Syntax

```
@property (nonatomic, assign) SessionType inventorySession;
```

### Remarks

AT288N 장비가 Inventory 를 수행하면서, Inventory 동작과 관련된 Tag 의 세션의 상태 정보를 가지고 Inventory 작업을 수행한다. inventorySession 속성은 AT288N 장비가 Inventory 를 수행할 때, Tag 에 접근하는 Session 을 지정한다.



## 2.4.10. globalBand

globalBand 메서드는 AT288N 장비에 설정된 UHF 주파수 국가를 반환한다.

### Syntax

```
@property (nonatomic, assign) int globalBand;
```

### Remarks

AT288N 장비로부터 국가별 UHF 주파수가 정상적으로 반환될 경우, 정수로 반환 받은 정수형의 데이터는 다음 표와 같다.

플래그	값	설명
KOREA	0	한국 UHF 주파수
EURO	1	유럽 UHF 주파수
USA	2	미국 UHF 주파수
CHINA	3	중국 UHF 주파수
TAIWAN	4	타이완 UHF 주파수
BRAZIL	5	브라질 UHF 주파수
MALAYSIA	6	말레이시아 UHF 주파수
AISA	7	아시아 UHF 주파수
JAPAN 1W	8	일본 UHF 주파수
JAPAN 250mW	9	일본 UHF 주파수
INDIA	10	인도 UHF 주파수
INDONESIA	11	인도네시아 UHF 주파수
JAPAN 125mW	12	일본 UHF 주파수

## 2.4.11. autosaveMode

AT288N 장비가 Inventory 를 수행하면서, 내장 메모리에 detect 된 data 를 저장할 것인지 아닌지를 설정하거나 반환한다.

### Syntax

```
@property (nonatomic, assign) BOOL saveMode;
```

### Remarks

saveMode 가 YES 로 설정이 되면 AT288N 장비가 Inventory 한 Tag 정보를 내부 메모리에 저장한다.

## 2.4.12. inventoryFormat

inventoryFormat 메서드는 Inventory 된 data 의 반환 형식을 설정하거나 설정된 값을 반환한다.

### Syntax

```
@property (nonatomic, assign) int inventoryFormat;
```

### Remarks

반환값에 따른 설정 형식은 아래와 같이 4 개의 형식을 갖는다.

0 : PC+EPC

1 : Serial No. + PC+EPC

2 : EPC

3 : Serial No. + EPC

## 2.4.13. lowBatteryState

AT288N 장비의 배터리 상태가 low 인지 아닌지를 반환한다. 반환값은 0 또는 1 이고 0 이면 Hight Battey 상태이다.

### Syntax

```
- (BOOL)lowBatteryState;
```

## 2.4.14. selectAction

Mask Selection 기능을 위해 AT288N Device 에 설정되어 있는 Mask Action 값을 반환 하거나 설정 한다.

### Syntax

```
@property (nonatomic, assign) BOOL selectAction;
```

### Remarks

Masking Selection 옵션에서 Mask 의 값이 주어진 Tag 와의 일치할 때, Selection 기능을 동작 시킬지, 불일치 할 때 Selection 기능을 동작 시킬지 결정한다.

플래그	값	설명
<b>Matching</b>	0	Mask 가 주어진 Tag 와 일치 하는 경우 동작한다.
<b>NonMatching</b>	1	Mask 가 주어진 Tag 와 일치 하지 않는 경우 동작한다.

## 2.4.15. SelectBitPtr

Mask Selection 기능을 위해 AT288N Device 에 설정되어 있는 Mask 값의 비교가 시작될 시작 주소를 반환하거나 설정한다

### Syntax

```
@property (nonatomic, assign) int selectBitPtr;
```

### Remarks

시작 주소는 bit 단위로 지정 된다.

## 2.4.16. SelectBank

AT288N Device 에 설정되어 있는 Mask Selection 이 수행될 Tag 의 Memory Bank 를 반환 하거나 설정 한다.

### Syntax

```
@property (nonatomic, assign) int selectionBank;
```

### Remarks

Bank 열거형은 8.1 을 참조한다.

### 2.4.17. singleMulti

AT288N Device 에 설정되어 있는 Inventory 방식(single/multi/filter)을 반환 하거나 설정 한다.

#### Syntax

```
@property (nonatomic, assign) int singleMulti;
```

#### Remarks

반환된 값이 0 이면 Multiple 방식, 1 이면 Single 방식, 2 이면 Filter 방식이다.

AT288N Device 의 Inventory Type 버튼을 눌러 Inventory 방식 변경이 가능하다.

### 2.4.18. typeBC

AT288N Device 에 설정되어 있는 Inventory 할 target tag 의 type(6C/6B)을 반환 하거나 설정 한다.

#### Syntax

```
@property (nonatomic, assign) BOOL typeBC;
```

#### Remarks

반환된 값이 0 이면 6C 방식, 1 이면 6B 방식이다.

### 2.4.19. scanTime

연결된 AT288N Device 의 memory access 동작 관련 동작 최대 유지 시간을 반환 하거나 설정 한다.

#### Syntax

```
@property (nonatomic, assign) int scanTime;
```

#### Remarks

초기값은 5 초로 설정 되어 있다(5 초 이상으로만 설정 가능).

## 3. EADevice BluetoothLe Class

EADevice BluetoothLe Class 는 AT288N 장비와 BLE 연결을 한 후, CBPeripheral Class 의 Instance 를 Wrapping 하여 EAREader Class 에 전달하기 위한 역할을 수행한다.

### 3.1. Initialize Methods

#### 3.1.1. initWithPeripheral

##### Syntax

```
- (id)initWithPeripheral:(CBPeripheral *)peripheral delegate:(id<EADeviceInitializeDelegate>)callback;
```

##### Parameters

**peripheral** : 이미 접속이 완료된 CoreBluetooth framework 의 CPeripheral Object 를 포함하고 있는 접속 정보를 관리하는 EADevice BluetoothLe Class 의 Instance 를 지정한다.

**delegate** : EAREader 의 상태 변경 등의 이벤트를 통한 AT288N 장비로부터 정보를 수신을 목적으로 하는 delegate protocol interface 를 지정한다.

##### Remarks

반드시 접속이 완료된 CPeripheral 개체가 포함되어 있는 EADevice BluetoothLe Class 의 Instance 를 지정하여야 한다. BLE 의 접속은 CoreBluetooth Framework 를 사용하고, 접속 이 후, AT288N 장비의 초기화 부터는 본 SDK 를 사용하여 통신한다.

## 3.2. Properties

### 3.2.1. delegate

BLE 통신으로 데이터를 수신한 경우, 수신 데이터를 처리할 이벤트 대리자를 설정하거나 반환한다.

#### Syntax

```
@property (weak, nonatomic) id<EADevice ReadDataDelegate> delegate;
```

#### Remarks

데이터 수신 루틴을 구현하고 있는 EADevice ReadDataDelegate 인터페이스를 구현한 개체를 설정하여 데이터 수신 처리를 한다.

### 3.2.2. name

호스트 장비와 연결된 AT288N 장비의 BLE Device 이름을 반환한다.

#### Syntax

```
- (NSString *)name;
```

#### Remarks

호스트 장비와 연결된 AT288N 장비의 BLE Device 이름을 반환한다.

### 3.2.3. address

호스트 장비와 연결된 AT288N 장비의 BLE Device Address 를 반환한다.

#### Syntax

```
- (NSString *)address;
```

#### Remarks

호스트 장비와 연결된 AT288N 장비의 BLE Device 의 임시 MAC 주소를 반환한다.

## 3.2.4. peripheral

AT288N 장비와 연결하기 위해 사용된 CBPeripheral Class 의 Instance 를 반환한다.

### Syntax

```
- (CBPeripheral *)peripheral;
```

### Remarks

AT288N 장비와 연결된 BLE 연결의 상세 정보를 제공하기 위해 Cocoa Touch 의 CBPeripheral 개체를 참조할 수 있다.

## 3.3. Methods

### 3.3.1. disconnect

연결된 AT288N 장비의 BLE 모듈 연결을 종료한다.

#### Syntax

```
- (void)disconnect;
```

#### Remarks

AT288N 장비와 연결을 종료하고 CBPeripheral 개체를 무효화 한다.

### 3.3.2. writeData

AT288N 장비에 BLE 를 통하여 데이터를 전송한다.

#### Syntax

```
- (void)writeData:(NSData *)data;
```

#### Parameter

**data** : 전송하고자 하는 데이터를 가지고 있는 NSData 개체

#### Remarks

writeData 메서드를 통하여 AT288N 장비에 BLE 를 사용하여 데이터를 전송하고, 수신된 데이터를 EADevice ReadDataDelegate 인터페이스를 통하여 수신한다.



## 4. LockParam

### 4.1. Properties

#### 4.1.1. killPassword

Kill Password 영역을 제어할 것인지를 설정하거나 반환한다.

##### Syntax

```
@property (nonatomic) BOOL killPassword;
```

##### Remarks

lock, unlock, permalock 메서드를 통하여 작업하는 대상 영역이 Kill Password 영역을 대상으로 작업할 것인지를 지정한다.

#### 4.1.2. accessPassword

Access Password 영역을 제어할 것인지를 설정하거나 반환한다.

##### Syntax

```
@property (nonatomic) BOOL accessPassword;
```

##### Remarks

lock, unlock, permalock 메서드를 통하여 작업하는 대상 영역이 Access Password 영역을 대상으로 작업할 것인지를 지정한다.

#### 4.1.3. epc

EPC 메모리를 제어할 것인지 설정하거나 반환한다.

##### Syntax

```
@property (nonatomic) BOOL epc;
```

##### Remarks

lock, unlock, permalock 메서드를 통하여 작업하는 대상 영역이 EPC 메모리 영역을 대상으로 작업할 것인지를 지정한다.

## 4.1.4. tid

TID 메모리를 제어할 것인지 설정하거나 반환한다.

### Syntax

```
@property (nonatomic) BOOL tid;
```

### Remarks

lock, unlock, permalock 메서드를 통하여 작업하는 대상 영역이 TID 메모리 영역을 대상으로 작업할 것인지를 지정한다.

## 4.1.5. user

User 메모리를 제어할 것인지 설정하거나 반환한다.

### Syntax

```
@property (nonatomic) BOOL user;
```

### Remarks

lock, unlock, permalock 메서드를 통하여 작업하는 대상 영역이 User 메모리 영역을 대상으로 작업할 것인지를 지정한다.

## 5. EAResultType

ResultType 열거형에 대한 메시지 문자열 변환을 위한 Utility 클래스이다.

### 5.1. Methods

#### 5.1.1. msg

##### Syntax

```
+(NSString *)msg:(ResultType)code;
```

##### Parameters

**code** : 메시지로 변환하고자 하는 ResultType 열거형

##### Remarks

ResultType 에 대한 문자열을 반환한다.

## 6. EAResultData

메서드가 ResultType 과 데이터를 반환할 경우 사용되는 ResultType 반환 클래스이다.

### 6.1. Properties

#### 6.1.1. result

##### Syntax

```
ResultType mResult;
```

##### Remarks

반환된 ResultType 를 나타낸다.

#### 6.1.2. mData

##### Syntax

```
NSData * mData;
```

##### Remarks

반환된 데이터를 나타내는 NSData 개체.

## 7. Deletagate Interfaces

### 7.1. EReaderDelegate

#### 7.1.1. readerInitialized

EReader 개체가 초기화가 완료될 통지 하기 위한 인터페이스 메서드이다.

##### Syntax

```
- (void)readerInitialized:(EReader *)reader;
```

##### Parameters

**reader** : 초기화가 완료된 변경된 EReader 개체.

##### Remarks

EReader 개체가 AT288N 장비와 연결되어 초기화가 완료되면 호출된다.

#### 7.1.2. updateDeviceState

동작형 메서드인 loadStoredTag 나 saveStoredTag, removeAllStoredTags 등 메서드의 비동기 작업 종료를 알리기 위한 인터페이스 메서드이다.

##### Syntax

```
- (void)updateDeviceState:(ResultType)error;
```

##### Parameters

**error** : AT288N 장비의 상태가 업데이트된 결과를 나타내는 ResultType 열거형.

##### Remarks

Stored Tag 메모리에 접근하는 작업은 긴 작업 시간을 필요로 하므로 작업 결과를 비동기적으로 알리기 위해 Stored Tag 메모리에 접근하는 작업이 종료되면 호출된다.

## 7.1.3. readTagResult

AT288N 장비가 Inventory 작업 중에 읽은 Tag 의 EPC 데이터를 반환하기 위한 인터페이스 메서드 이다.

### Syntax

```
- (void)readTagResult: (NSString *)tag rssi:(float)rssi phase:(float)phase;
```

### Parameters

**tag** : AT288N 장비가 읽은 Tag 의 PC+EPC 데이터를 Hex 형태의 문자열.

**rssi** : AT288N 장비가 해당 Tag 를 읽을 때 rssi

**phase** : AT288N 장비가 Tag 를 읽을 때 phase

### Remarks

Inventory 메소드를 호출하고 AT288N 장비가 Tag 를 읽으면 이벤트가 호출된다.

## 7.1.4. changedActionState

AT288N 장비에 동작 상태가 변경되면 이를 통지하기 위한 인터페이스 메서드이다.

### Syntax

```
- (void)changedActionState:(CommandType)action;
```

### Parameters

**action** : 현재 AT288N 장비의 동작 상태를 나타내는 CommandType 열거형.

### Remarks

inventory, readMemory, writeMemory, lock, unlock, permaLock, stop 등의 메서드를 호출하면 AT288N 장비의 동작 상태가 변경된다. 이를 사용자에게 통지하기 위하여 이벤트가 호출된다.

## 7.1.5. tagAccessResult

AT288N 장비가 Tag 를 Access 한 결과를 통지하기 위한 인터페이스 메서드이다.

### Syntax

```
- (void)tagAccessResult:(ResultType)error actionState:(CommandType)action epc:(NSString *)epc data:(NSString *)data;
```

### Parameters

**error** : 동작이 완료된 Access 관련 메서드의 결과를 나타내는 ResultType 열거형.

**action** : AT288N 장비의 어떤 동작의 결과인지를 나타내는 CommandType 열거형.

**epc** : AT288N 장비가 접근한 Tag 를 나타내는 Tag EPC 값을 나타내는 Hex 형태의 문자열.

**data** : readMemory 메서드와 같이 Tag 에서 데이터 읽기 위한 Tag Access 인 경우, Tag 에서 읽은 데이터를 반환하기 위한 Hex 형태의 문자열

### Remarks

tagAccessResult 이벤트는 inventory 메서드를 제외한 readMemory, writeMemory, lock, unlock, permaLock I 등의 Tag 접근 메서드들에 대한 Tag 접근 결과로서 호출된다.

readMemory 메서드의 결과로 Tag 에서 읽은 데이터를 포함하기도 한다.

## 7.1.6. opmodesetting

AT288N Device 의 inventory 방식(single/multi/filter) 변경 되었을 때 설정 값을 반환하기 위한 인터페이스 메서드 이다.

### Syntax

```
- (void)opmodesetting:(int)mode;
```

### Parameters

**mode** : AT288N 장비에 설정된 Inventory 방식

### Remarks

AT288N 장비의 Inventory 방식이 바뀌면 이벤트가 호출된다.

## 7.1.7. powergainchange

AT288N Device 의 powergain 이 변경 되었을 때 설정 값을 반환하기 위한 인터페이스 메서드 이다.

### Syntax

```
- (void)powergainchange:(int)gain;
```

### Parameters

**gain** : AT288N 장비에 설정된 powergain 값

### Remarks

AT288N 장비의 설정된 powergain 값이 변경 되면 이벤트가 호출된다.

## 7.1.8. memoryaccessresult

AT288N Device 에서 Tag memory 에 접근하는 작업 후 결과를 반환하기 위한 인터페이스 메서드 이다.

### Syntax

```
- (void)memoryaccessresult:(int)result;
```

### Parameters

**result** : Tag memory 에 접근(lock/read/write)한 결과

### Remarks

Tag 의 memory 에 접근하여 작업한 결과가 AT288N 으로 부터 전달되면 이벤트가 호출된다.



## 8. Enumerators

### 8.1. BankType

EAREader 에서 접근할 태그의 메모리 बैं크를 나타낸다.

Flag	Value	Description
<b>Bank_Reserved</b>	0	Reserved 메모리 Bank 를 가리킨다.
<b>Bank_EPC</b>	1	EPC 메모리 Bank 를 가리킨다.
<b>Bank_TID</b>	2	TID 메모리 Bank 를 가리킨다.
<b>Bank_User</b>	3	User 메모리 Bank 를 가리킨다.

### 8.2. BuzzerState

EAREader 장비의 부저를 제어할 때, 부저의 상태를 정의한다.

Flag	Value	Description
<b>Buzzer_Off</b>	0	장비의 부저 소리를 끈다.
<b>Buzzer_ON</b>	1	장비의 부저 소리를 켜다.

### 8.3. SessionType

EAREader 가 Inventory 명령을 수행할 때, 대상이 되는 태그의 세션을 나타낸다.

Flag	Value	Description
<b>Session_S0</b>	0	inventoried S0
<b>Session_S1</b>	1	inventoried S1
<b>Session_S2</b>	2	inventoried S2
<b>Session_S3</b>	3	inventoried S3

## 8.4. CommandType

EAREader 의 비동기 명령이 무엇이 완료되었는지 나타낸다.

Flag	Value	Description
<b>CommandInventory</b>	0x66	인벤토리 작업 중
<b>CommandReadMemory</b>	0x69	Read Memory 작업 중
<b>CommandWriteMemory</b>	0x77	Write Memory 작업 중
<b>CommandKill</b>	0x6B	Kill Tag 작업 중
<b>CommandLock</b>	0x6C	Lock 작업 중
<b>CommandStop</b>	0x33	동작이 중지됨