



ATID Co.,Ltd

AT288N SDK Reference Guide for iPhone Developers

AT288N SDK Reference Guide

SW Team

2023-06-12

Revised History

| Version | Date of revision | Reason for revision ¹ | Revised Content ² | Write |
|---------|------------------|----------------------------------|------------------------------|-----------|
| v0.1 | 2017. 8. 31. | draft | | Ryu Eunju |
| v0.2 | 2019. 01. 22 | add | "typeBC", "scanTime" add | Ryu Eunju |
| v0.3 | 2023-06-12 | Edit | Platform tool version | SW Team |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

¹Reason for revision : This section explains whether the new content is the addition, revision, or removal of the original document

²Revised content : Description on revised contents and the revised pages

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..... | 25 |
| 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. | EARedReaderDelegate | 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

The purpose of this document is to describe how to use SDK Library for developers wishing to develop iPhone application programs by using SDK Library.

Xcode 14.2 was used as a development tool. Development platform supports iOS 10.3 or higher.

2. EReader Class

EReader Class provides AT288N and Programmable Interface on iPhone. CoreBluetooth framework provided as iOS SDK is used to connect AT288N to an iPhone.

2.1. Initialize Methods

Like ordinary CocoaTouchClass, EReader class also provides a function for initialization. Initialization function beginning with init. provides initWithDevice function and initWithBTDevice function as initialization.

2.1.1. initWithDevice

Method to create EReader entity communicating by using AT288N BLE on iPhone.

Syntax

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

Parameters

device : designates the Instance of EADevice BluetoothLe Class controlling access information that includes CPeripheral Object of the CoreBluetooth framework that the access is already completed.

delegate : designates the delegate protocol interface to receive information from ATID device via events such as a change in state of EReader.

Remarks

Make sure to designate the Instance of EADevice BluetoothLe Class that the CPeripheral which the access is already completed is included. For access to BLE, use CoreBluetooth Framework. After access, SDK is used to communicate since initialization of AT288N.

2.1.2. Disconnect

Terminates connection to ATID device connected to iPhone

Syntax

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

Remarks

Disconnect Method calls disconnect of CBPeripheral. After this Method is called, CBPeripheral Instance included as EADevice BluetoothLe Instance is disabled.

2.2. Reader Properties

2.2.1. peripheral

Returns to Instance of CBPeripheral Class used to connect to AT288N.

Syntax

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

Remarks

Refer to CoreBluetooth framework for Interface to check state of connection to AT288N.

2.2.2. getAction

Returns current active state of EAREader Instance.

Syntax

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

Remarks

getAction Method returns Action Command which EAREader transmitted to AT288N last to let you know what task current EAREader commanded to ATID device and what task is being performed.

2.2.3. setDelegate

Changes the recipient that will receive events from Instance of EAREader.

Syntax

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

Parameters

delegate : designates the Instance of Class that implements the EAREaderDelegate Protocol which can receive an event.

Remarks

If you wish to change recipient that receives an event from EAREader Instance, you can use the setDelegate Method.

2.3. Action Methods

2.3.1. Inventory

Automatically Perform Inventory function depending on the state of AT288N.

Syntax

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

Return

Returns a result of execution of an instruction in a form of ResultType.

Remarks

Return NoError when operating inventory method normally. While Inventory method is working, you cannot use another method and thus you stop motion by using stop method and should call another method.

If AT288N reads tag by inventory command, readTagResultMethod of EAREaderDelegate Protocol will be called.

2.3.2. readMemory

readMemory is a command to read Tag memory directly.

Syntax

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

Parameters

bank : set Memory Bank of data to read.

offset : set initial address of data to read. (in WORD)

length : set the length of data to read.(in WORD)

mask: set initial mask of data to read.

Return

Return a result of execution of an instruction in a form of ResultType enumeration.

Remarks

Return NoError if operating readMemory method normally. While readMemory method is working, you cannot use another method and thus you stop motion by using stop method and should call another method..

2.3.3. writeMemory

writeMemory is a command to write memory of tag directly.

Syntax

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

Parameters

bank : sets Memory Bank of tag to perform data writing.

offset : sets initial address for data to begin writing.(in WORD)

data : enters data to write in Hex value.

mask: Sets mask to write string.

Return

This function returns a result of execution of an instruction in a form of ResultType enumeration.

Remarks

Returns NoError if operating writeMemory method normally. While writeMemory method is working, you cannot use another method and thus you stop motion by using stop method and should call another method

2.3.4. lock

lock method is used in locking to prevent access to Tag Memory.

Syntax

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

Parameters

param: lock option of Memory Bank

tagmask: Set mask of string

Return

This function returns a result of execution of an instruction in a form of ResultType enumeration.

Remarks

A result of execution of Lock is returned as readTagResult event. If Lock method succeeds, returns as NoError. If Lock method fails, returns as a value other than NoError

If event handler is called, Lock motion will stop automatically. If you wish to cancel while lock is being performed, you only have to call stop method.

2.3.5. stop

stop method stops motions of all Action series.

Syntax

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

Return

This function returns a result of execution of an instruction in a form of ResultType enumeration.

Remarks

Cancels all works which AT288N is operating and change state to stop state.

2.3.6. getSaveData

getSaveData method returns Tag readTagResult event in the stored data of AT288N

Syntax

```
- (void)getSaveData:
```

Remarks

A result of execution of getSaveData is returned as readTagResult event. If getSaveData method succeeds, returns as changeActionState.

2.3.7. deleteSaveData

deleteSaveData method deletes all data lists stored in the internal memory of the AT288N.

Syntax

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

Return

This function returns a result of execution of an instruction in a form of Reply packet.

Remarks

If all values stored in internal memory of AT288N are deleted, changeActionState will be returned.

2.4. Device Properties

2.4.1. firmwareVersion

getFirmwareVersion method returns Firmware version of AT288N as character string.

Syntax

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

Remarks

If Firmware version is returned normally from the ATID device, character string will be returned.

2.4.2. buzzer

buzzer 속성은 AT288N 장비의 Buzzer 상태를 설정하거나 반환한다.

Buzzer returns or sets whether to buzzer status of AT288N.

Syntax

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

Type

Buzzer method returns Buzzer status of AT288N as BuzzerState string.

Remarks

Set AT288N buzzer status where on or off

2.4.3. continuousMode

continuousMode property returns or sets whether to read tag once or consecutively when ATID device performs Inventory function.

Syntax

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

Remarks

This function returns or sets whether to read tag once or consecutively during Inventory operation. If this value is YES, it means Continuous Mode has been set. If this value is NO, it means Continuous Mode has not been set.

2.4.4. powerGain

powerGain method returns or sets Power Gain of AT288N.

Syntax

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

Remarks

powerGain property is integer type. Value *10 of power output value is set or returned. For example, if antenna power is 30dBm, value of 300 will be set or returned.

2.4.5. PowerOff

powerOff method sets or returns AT288N power status

Syntax

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

Remarks

While AT288N MI device connecting with Hoset device, AT288N R2000 device function allows reader to turn off ,if AT288N is not used for. If you turn on returns value to 1, you have only to set this value to 0.

2.4.6. IdleTime

idleTime property returns or sets idle time when AT288N performs Inventory.

Syntax

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

Remarks

This function is used to minimize interference among readers in an environment where several AT288N are operated and prevent overheating of reader resulting from long full time operation. In the event of RF Channel Hopping, during Power Idle Time(ms), reader powers off and during Inventory Time(ms), reader powers on.

2.4.7. autoOffTime

autoOffTime property sets or returns time that AT288N turns off automatically when it does not operate.

Syntax

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

Remarks

This function allows reader to turn off automatically if AT288N is not used for a certain time to save Battery to device. Unit of setting autoOffTime is sec. If you do not want to use Auto Off Time function, you have only to set this value to 0.

2.4.8. accessPassword

This function sets or returns Access Password needed when you perform Access Command for tag that lock is placed on AT288N.

Syntax

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

Remarks

When implementing Read/Write/Lock Command that Access Password is set to 0, Access Password is not used. Password of 8Byte unit is set. Access Password is used.

2.4.9. inventorySession

This function sets or returns session of tag to access when AT288N performs Inventory.

Syntax

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

Remarks

This function performs Inventory with information on tag session state related to inventory motion while AT288N performs Inventory. inventorySession property designates session that AT288N accesses tag when it performs Inventory.

2.4.10. globalBand

globalBand method returns country setting of UHF frequency in AT288N.

Syntax

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

Remarks

UHF Frequency Band supported by AT288N Devices for each country

| 플래그 | 값 | 설명 |
|-------------|----|-------------------------|
| KOREA | 0 | Korea UHF Frequency |
| EURO | 1 | EU UHF Frequency |
| USA | 2 | USA UHF Frequency |
| CHINA | 3 | China UHF Frequency |
| TAIWAN | 4 | Taiwan UHF Frequency |
| BRAZIL | 5 | Brazil UHF Frequency |
| MALAYSIA | 6 | Malaysia UHF Frequency |
| AISA | 7 | ASIA UHF Frequency |
| JAPAN 1W | 8 | Japan UHF Frequency |
| JAPAN 250mW | 9 | Japan UHF Frequency |
| INDIA | 10 | India UHF Frequency |
| INDONESIA | 11 | Indonesia UHF Frequency |
| JAPAN 125mW | 12 | Japan UHF Frequency |

2.4.11. autosaveMode

This function returns or set whether to store data detected in embedded memory while AT288N performs Inventory

Syntax

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

Remarks

If autosaveMode is set to YES, AT288N stores tag information inventoried or barcode information scanned in embedded memory.

2.4.12. inventoryFormat

inventoryFormat method set inventory data transmitted or return set up value.

Syntax

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

Remarks

Set return value 4types of them as below.

0 : PC+EPC

1 : Serial No. + PC+EPC

2 : EPC

3 : Serial No. + EPC

2.4.13. lowBatteryState

Return AT288N battery status whether battery status low or not, return value is 0 or 1 , if return value is 0 which is High Battery status.

Syntax

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

2.4.14. selectAction

Define action on mask value in Masking Selection function in AT288N

Syntax

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

Remarks

In Masking Selection option, if the tag data match with Mask value, whether selection feature run or not.

| Flag | Value | Remark |
|-------------|-------|---------------------------|
| Matching | 0 | Matching with Tag data |
| NonMatching | 1 | Un-matching with Tag data |

2.4.15. SelectBitPtr

For Mask selection feature, define return initial address or set up when AT288N starts compare to set up Mask value.

Syntax

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

Remarks

Set initial address in bit.

2.4.16. SelectBank

Designate Tag memory bank to perform Mask in masking selection function

Syntax

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

Remarks

Enumeration Bank refers to 8.1

2.4.17. singleMulti

Whether return or set up to perform in Inventory function in single, multi or filter in AT288N.

Syntax

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

Remarks

Multiple is returned 0, Single is returned 1 and Filter is returned 2.

Press Type button in AT288N Inventory function is enable to chance.

2.4.18. typeBC

Perform Inventory function depending on return type6C/6B) target tag or set for AT288N.

Syntax

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

Remarks

If the value 6C returns 0, 6B returns 1.

2.4.19. scanTime

It can return or set a memory access duration time(sec) for connected AT288N.

Syntax

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

Remarks

Initially set 5 sec(minimum value is 5 sec).

3. EADevice BluetoothLe Class

EADevice BluetoothLe Class plays a role to wrap and deliver CBPeripheral Class Instance to EAREader Class after connecting AT288N to BLE tool.

3.1. Initialize Methods

3.1.1. initWithPeripheral

Syntax

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

Parameters

peripheral : it designates EADevice BluetoothLe Class Instance which manages the linking information including CPeripheral Object of CoreBluetooth framework already completely linked.

delegate : it designates the delegate protocol interface aiming at receiving information from AT288N through events such as the status change of EAREader, and etc.

Remarks

Make sure to specify the EADevice BluetoothLe Class Instance including the already linked CPeripheral Object. As for BLE linkage, use CoreBluetooth Framework and communicate by using this SDK from the initialization of AT288N after being linked.

3.2. Properties

3.2.1. delegate

When receiving data through BLE communication, set an agent to deal with the received data or retrocede it.

Syntax

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

Remarks

Set the object actualized EADevice ReadDataDelegate interface, which is performing the receiving routine, to take care of the receipt of data.

3.2.2. name

Retrocede the name of BLE Device of AT288N connected to the Host tool.

Syntax

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

Remarks

Retrocede the name of BLE Device of AT288N connected to the Host tool.

3.2.3. address

Retrocede the BLE Device Address of AT288N connected to the Host tool.

Syntax

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

Remarks

Retrocede the temporary MAC Address of BLE Device of AT288N connected to the Host tool.

3.2.4. peripheral

Retrocede CBPeripheral Clas Instance used for connecting AT288N.

Syntax

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

Remarks

You may refer to the CBPeripheral Object of Cocoa Touch for providing the detail information of BLE linkage connected to AT288N.

3.3. Methods

3.3.1. disconnect

It terminates the BLE module linkage connected to AT288N.

Syntax

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

Remarks

Terminate the linkage with AT288N and nullify CBPeripheral Object.

3.3.2. writeData

It transmits data through BLE to AT288N.

Syntax


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

Parameter

data : NSData object having data intending to transmit

Remarks

Transmit data by using BLE of ATID tool through writeData method and receive the received data through EADevice ReadDataDelegate Interface.

| | | | | | | | |
|--|--|--|---------|------|------------|--------------|------|
|  | | AT288N SDK Reference Guide for iPhone Developers | | | | | |
| AT288N SDK Reference Guide | | | | | Company | ATID Co.,Ltd | |
| Doc. | | Drafter | SW Team | Date | 2023-06-12 | Ver. | v0.3 |

4. LockParam

4.1. Properties

4.1.1. killPassword

Set whether the scope of Kill Password is controlled or retrocede it.

Syntax

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

Remarks

Designate whether the target scope working through the methods of lock, unlock, and permalock will work on the scope of Kill Password as the subject.

4.1.2. accessPassword

Set whether Access Password scope is controlled or retrocede it.

Syntax

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

Remarks

Designate whether the target scope working through the methods of lock, unlock, and permalock will work on the scope of Access Password as the target.

4.1.3. epc

Set whether EPC memory is controlled or retrocede it.

Syntax

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

Remarks

Designate whether the target scope working through the methods of lock, unlock, and permalock will work on the scope of EDC memory as the target.

4.1.4. tid

Set whether TID memory is controlled or retrocede it.

Syntax

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

Remarks

Designate whether the target scope working through the methods of lock, unlock, and permalock will work on the scope of TID memory as the target.

4.1.5. user

Set whether User memory is controlled or retrocede it.

Syntax

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

Remarks

Designate whether the target scope working through the methods of lock, unlock, and permalock will work on the scope of User memory as the target

5. EAResultType

Utility Class is for converting message strings to the enumeration of ResultType.

5.1. Methods

5.1.1. msg

Syntax

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

Parameters

code : ResultType enumeration to be converted a message string

Remarks

Convert message strings of ResultType

6. EAResultData

ResultType Return Class is used if method is to retrocede ResultType and data.

6.1. Properties

6.1.1. result

Syntax

```
ResultType mResult;
```

Remarks

Show the retroceded ResultType.

6.1.2. mData

Syntax

```
NSData * mData;
```

Remarks

NSData object showing the retroceded data.

7. Deletagate Interfaces

7.1. EAREaderDelegate

7.1.1. readerInitialized

It is an interface method to notice the completion of initializing an EAREader object.

Syntax

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

Parameters

reader : a changed EAREader object completely initialized

Remarks

It is called when the initialization of EAREader object connected to AT288N is complete.

7.1.2. updateDeviceState

It is an interface method to notice the termination of asynchronous work of operational method such as loadStoredTag, saveStoredTag, removeAllStoredTags, or etc.

Syntax

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

Parameters

error : it is a ResultType indicating the consequence of the updated state of AT288N

Remarks

It is called when the work accessing to Stored Tag memory is ended to notice the consequence of the work asynchronously as it takes a long work time.

7.1.3. readTagResult

It is an interface method to retrocede the EPC data of the tag read while conducting inventory over AT288N.

Syntax

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

Parameters

Tag: string in Hex type of PC and EPC data of the tag read by AT288N.

rssi: rssi when AT288N reads the corresponding tag

phase: phase when AT288N reads tag

Remarks

Call inventory method and an event will be called when AT288N reads tag.

7.1.4. changedActionState

It is an interface method to notice when the operation of AT288NI is changed

Syntax

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

Parameters

action : it is a CommandType enumeration indicating the current operational state of AT288N

Remarks

The operational state of AT288N is changed when method such as inventory, readMemory, writeMemory, lock, unlock, permaLock, stop, and etc. An event is called to notify this to the user.

7.1.5. tagAccessResult

It is an interface method to notice the consequence of tag accessed by AT288N.

Syntax

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

Parameters

error: it is a ResultType indicating the result of the method relating to access operation is completed.

action: it is a CommandType enumeration indicating the consequence of a type of operation of AT288N.

epc: it is a HexType string indicating the EPC value showing the tag accessed by AT288N.

data: it is a HexType string to retrocede the data read in the Tag in case of Tag Access in which reads data in tag such as readMemory method.

Remarks

tagAccessResult event is called as the result of accessing to tag in regard to tag access method such as readMemory, writeMemory, lock, unlock, permaLock, and etc., except inventory method. It also includes the data read in tag as the result of readMemory method.

7.1.6. opmodesetting

It is an interface method to notice the return value of AT288N device such as single, multi or filter.

Syntax

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

Parameters

mode : AT288N Inventory set up method

Remarks

Inventory function passed to event

7.1.7. powergainchange

Retrocede interface method ,if Powergain value change in AT288N.

Syntax

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

Parameters

gain : Powergain value on AT288N.

Remarks

Retrocede event, if powergain value change in AT288N

7.1.8. memoryaccessresult

This function sets or returns Access device tag memory on AT288N.

Syntax


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

Parameters

result : access to lock, read or write tag memory result.

Remarks

This function access to Tag memory , AT288N is used if is to retrocede event and data.

| | | | | | | | |
|--|--|--|---------|------|------------|--------------|------|
|  | | AT288N SDK Reference Guide for iPhone Developers | | | | | |
| AT288N SDK Reference Guide | | | | | Company | ATID Co.,Ltd | |
| Doc. | | Drafter | SW Team | Date | 2023-06-12 | Ver. | v0.3 |

8. Enumerators

8.1. BankType

Designate Tag memory bank accessible in EARReader.

| Flag | Value | Description |
|----------------------|-------|----------------------|
| Bank_Reserved | 0 | Reserved memory bank |
| Bank_EPC | 1 | EPC memory bank |
| Bank_TID | 2 | TID memory bank |
| Bank_User | 3 | User memory bank |

8.2. BuzzerState

Define manage buzzer status for EARReader device.

| Flag | Value | Description |
|-------------------|-------|-----------------|
| Buzzer_Off | 0 | Turn off buzzer |
| Buzzer_ON | 1 | Turn on Buzzer |

8.3. SessionType

It shows tag session which is the subject when EARReader conducts inventory.

| Flag | Value | Description |
|-------------------|-------|----------------|
| Session_S0 | 0 | inventoried S0 |
| Session_S1 | 1 | inventoried S1 |
| Session_S2 | 2 | inventoried S2 |
| Session_S3 | 3 | inventoried S3 |

8.4. CommandType

It shows what is done in asynchronous command of EAREader.

| Flag | Value | Description |
|---------------------------|-------|-------------------------------|
| CommandInventory | 0x66 | Under inventory |
| CommandReadMemory | 0x69 | Under working on Read Memory |
| CommandWriteMemory | 0x77 | Under working on Write Memory |
| CommandKill | 0x6B | Under working on Kill Tag |
| CommandLock | 0x6C | Under working on Lock |
| CommandStop | 0x33 | Cease of operation |