AsReader

AsReader SDK Reference Guide for iOS Developers

AsReader SDK Reference Guide

AsReader, Inc.

2017-02-20

Record of Revision

Ver.	Date	Description of the Change	Note	Name

Content

1. Intro

The main purpose of this document is to build the developing environment where SDK Library can be used by developers who want to develop iOS applications using AsReader SDK Library and inform users about SDK Library instructions.

Development tool in this document is Xcode7.3.1 or above and Platform to be developed supports iOS 8.0 or higher.

2. AsReader Class

AsReader Class provides AsReader and Programmable Interface on iOS.

2.1. Initialize Methods

AsReader Class provides functions for initialization like general CocoaTouch Classes do. Typically, they start with 'init'. 'initWithAsReaderGUN' is the function.

2.1.1. initWithAsReaderGUN

This is a function for generating an AsReader instance for controlling AsReader Gun Type. AsReader instance reqirese a AsRingDevice instance to get connected with iOS device.

Syntax

```
- (id)initWithAsReaderGUN:(AsReaderGUN *)asReaderGUN
delegate:(id<AsReaderDelegate>)delegate;
```

Parameters

device : Assigns AsReaderGUN instance

delegate : Assigns a delegate which will receive information or notifications from AsReader device on events like changing status of AsReader.

2.1.2. Disconnect

Disconnecting AsReader from iOS device.

Syntax

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

Remarks

Disconnecting method makes AsReader disconnect internally. Once this method is called out, Asreader instance of will be dastroyed.

2.2. Reader Properties

2.2.1. mAsReaderGUN

This propert is used to set or get AsReaderGUN instance used in AsReader instance.

Syntax

```
@property (nonatomic, strong)  AsReaderGUN *mAsReaderGUN;
```

Remarks

AsReader instance requires one AsReaderGUN instance to work properly.

2.2.2. Algorithm

This propert is used set AsReaderGUN's Q type.

Syntax

```
@property (nonatomic, assign) AlgorithmType algorithm;
```

Remarks

The AlgorithmType value=0 is Fixed Q,value = 1 is Dynamic Q.

2.2.3. minQ

This propert is used set min Q value.

Syntax

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

Remarks

The minQ value between 0 – 15.

2.2.4. maxQ

This propert is used set max Q value.

Syntax

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

Remarks

The maxQ value between 0 – 15.

2.2.5. qValue

This propert is used set Fixed value.

Syntax

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

Remarks

The qValue between 0 – 15.

2.2.6. getAction

This is the method to return the status of current action of AsReader Instance.

Syntax

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

Remarks

GetAction method shows what operation is done to AsReader, or what kind of operation the AsReader is on currently by returning the last Action Command that has been transmitted to AsReader from AsReader.

2.2.7. setDelegate

It changes the target that receives the event from AsReader's Instance.

Syntax

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

Parameters

delegate : It assigns the instance of the class, which implements AsReaderDelegate Protocol that can receive the event.

Remarks

setDelegate method can be used for changing the target object receiving events from AsReader Instance.

2.2.8. setScanMode

Set AsReader to scan the barcode or RFID.

Syntax

```
(void)setScanMode:(ScanMode)scanMode;
```

Parameters

ScanMode: Scan type.

Remarks

This mode is not permanent, it will be reset to default after app reinitiate, RFID is the default mode, application should be responsible for initializing the mode on initializing process of every functional type.

2.3. Action Methods

2.3.1. Inventory

Automatically operates the proper function of Inventory.

Syntax

```
- (ResultCode)inventory;
```

Return

Returns the outcome by ResultCode enumeration type.

Remarks

If the inventory method operates normally, it returns NoError. When the inventory method is running, another method is not available. To call out another method, stop the current operation by stop method.

Once AsReader read a tag by inventory command, the readTag method from AsReaderDelegate Protocol will be called out.

2.3.2. readMemory

ReadMemory is a command to read directly memory of Tag.

Syntax

```
- (ResultCode)readMemory:(MemoryBank)bank offset:(int)offset
length:(int)length;
```

Parameters

bank : Sets memory bank of data.

offset : Sets the initial address of data. (unit: word)

length : Sets the length of data.(unit: word)

Return

Returns the outcome by ResultCode enumeration type.

Remarks

If the inventory method operates normally, it returns NoError. When the inventory method is running, another method is not available. To call out another method, stop the current operation by stop method.

2.3.3. writeMemory

WriteMemory is a command to write directly the memory of Tag.

Syntax

```
- (ResultCode)writeMemory:(MemoryBank)bank offset:(int)offset
value:(NSString *)value;
```

Parameters

bank : Sets Memory Bank of Tag

offset : Sets start address of data. (unit: word)

data : Inputs data as a value of Hex.

Return

Returns the outcome by ResultCode enumeration type.

Remarks

If the inventory method operates normally, it returns NoError. When the inventory method is running, another method is not available. To call out another method, stop the current operation by stop method.

2.3.4. lock

lock method is command to lock Tag memory.

Syntax

```
- (ResultCode)lock:(LockParam *)param;
```

Parameters

param : The option for locking Memory Bank.

Return

Returns the outcome by ResultCode enumeration type.

Remarks

The outcome of Lock returns as accessResult event. If lock method succeeds, it returns NoError, if not, it returns some value but NoError.
Once event handler is called out, lock process will be automatically terminated. Use stop method to cancel lock during lock process.

2.3.5. unlock

This method is to unlock Tag memory using lock method

Syntax

```
- (ResultCode)unlock:(LockParam *)param;
```

Parameters

param : The option for locking Memory Bank

Return

Returns the outcome by ResultCode enumeration type.

Remarks

The outcome of unlock returns as accessResult event. If unlock method succeeds, it returns NoError, if not, it returns some value but NoError.
Once event handler is called out, unlock process will be automatically terminated. Use stop method to cancel unlock during unlock process.

2.3.6. permaLock

Syntax

```
- (ResultCode)permaLock:(LockParam *)param;
```

Parameters

param : The option for locking Memory Bank

Return

Returns the outcome by ResultCode enumeration type.

Remarks

The outcome of permaLock returns as accessResult event. If permaLock method succeeds, it returns NoError, if not, it returns some value but NoError.
Once event handler is called out, permaLock process will be automatically terminated. Use stop method to cancel permaLock during permaLock process.

2.3.7. kill

Kill method is a command to kill Tag, which makes no response with AsReader device.

Syntax

```
- (ResultCode)kill:(NSString *)killPassword;
```

Parameters

killpassword : Kill password for tag to be applied

Return

Returns the outcome by ResultCode enumeration type.

Remarks

It is not restorable once kill method is applied.

2.3.8. stop

This method is to discontinue all of action operation.

Syntax

```
- (ResultCode)stop;
```

Return

Returns the outcome by ResultCode enumeration type.

Remarks

Makes AsReader idle, cancel all of the functions.

2.3.9. defaultParameter

Reset all parameters to default.

Syntax

```
- (ResultCode)defaultParameter;
```

Return

Returns the outcome by ResultCode enumeration type.

Remarks

Reset all Parameter and shows the result. NoError will be returned if all functions are working properly.

2.3.10. saveParameter

Save all parameters, to memory.

Syntax

```
- (ResultCode)saveParameter;
```

Return

Returns the outcome by ResultCode enumeration type.

Remarks

When it is in success, NoError will be returned

2.4. Device Properties

2.4.1. firmwareVersion

This is the method to return the firmware version as string.

Syntax

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

Remarks

If the firmware version can be returned normally, the version of string will be returned

2.4.2. regionName

This is the method to return the region name of RF module as string .

Syntax

```
- (NSString *)regionName;
```

Remarks

If the region name can be returned normally, the region name is string will be returned

2.4.3. serialNumber

This is the method to return the serial Number to string.

Syntax

```
- (NSString *)serialNumber;
```

Remarks

If the serial number can be returned normally, the serial number of string will be returned

2.4.4. rFModuleVersion

This is the method to return the RF module version to string.

Syntax

```
- (NSString *)rFModuleVersion;
```

Remarks

If the RF module version can be returned normally, the RF module version of string will be returned

2.4.5. powerGainRange

This is the method to return the antenna output level of AsReader

Syntax

```
- (CMinMaxValue)powerGainRange;
```

Remarks

It shows minimum and maximum value of Antenna level by Min, Max of CMinMax. The returning value will be 10 multiplied by the output level. For instances, the value of 300 and 110 mean 30dBm and 11dBm.

2.4.6. buzzer

buzzer is a method to set or show the buzzer state.

Syntax

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

Type

Sets Buzzer state by enumeration type.

Remarks

Mute, low volume or louder volume can be set.

2.4.7. continuousMode

It returns or sets tag to read the mode continuously or singularly.

Syntax

```
@property (nonatomic, assign) BOOLcontinuousMode;
```

Remarks

It returns or sets tag to read the mode continuously or not when inventory running. The output, YES means 'in continuous Mode', and NO means 'not in continuous Mode'.

2.4.8. limitTagCount

It returns or sets the maximum number of Tags that will detect on Inventory.

Syntax

```
@property (nonatomic, assign) intlimitTagCount;
```

Remarks

It will not run if Continuous Mode is One Tag Mode. It will run only with Multi Tag Mode. When the value is 0, Inventory could stop only with Stop Operation command.

2.4.9. powerGain

It returns or sets power Gain of AsReader.

Syntax

```
@property (nonatomic, assign) intpowerGain;
```

Remarks

It shows the outcome in the integer type value. The returning or set value will be 10 multiplied by the output level. For instances, the value of 300 means 30dBm.

2.4.10. operationTime

This method returns or sets the running time of AsReader.

Syntax

```
@property (nonatomic, assign) intoperationTime;
```

Remarks

If 0 is set for operationTime value, it will run continuously until the stop method stops it. The unit of the operationTime value is ms.

2.4.11. inventoryTime

This method returns or sets the time taken for the antenna operation during the inventory running

Syntax

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

Remarks

Reader performs Inventory during its Inventory Time, and stops performing Inventory during its Read Idle Time. This repeats alternately.

2.4.12. IdleTime

This method returns or sets the idle time when AsReader Device is running the inventory.

Syntax

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

Remarks

To prevent interferences in situations where many AsReader Devices are being used. Also used to prevent overheating when the device is used in Full Time. When RF Channel Hopping, Reader is set as RF Power Off during Power Idle Time (ms) and On during Inventory Time (ms).

2.4.13. autoOffTime

AutoOffTime method returns or sets the time taken to be off automatically when AsReader Device is not being used.

Syntax

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

Remarks

Used to automatically reduce the battery usage time when AsReader Device is not being used for the set time. Unit is in sec. If the user does not want to use Auto Off Time, set the value as 0.

2.4.14. accessPassword

This method returns or sets required Access Password when performing Access Command on the Tag that is in Lock.

Syntax

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

Remarks

When performing Read/Write/Lock Command where Access Password is set as 0, Access Password is not used. When the password is set in 8 Byte unit, Access Password is used.

2.4.15. inventorySession

This method returns or sets the Session of its accessing tag when performing Inventory.

Syntax

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

Remarks

When AsReader performs the inventory, it has the information of the state of Tag session. inventorySession method assigns which session will access the Tag when AsReader is performing the inventory.

2.4.16. sessionFlag

Assigns settings of session Flag to be inventory during inventory operation approached by Tag.

Syntax

```
@property (nonatomic, assign) SessionFlag sessionFlag;
```

Remarks

It assigns only the tag, which is in state of designated sessionFlag to, be inventory.

This is an addition to v1.3.x or higher

2.4.17. selectionMask

When AsReader is performing Inventory, this method returns Session that will carry on the Action of its Selection Mask. AsReader has 0~7 Selection Mask slots, a total 8 slots. This means that Tag can be get up to 8 Selection Mask

Syntax

```
@property (nonatomic, strong) AsSelectMaskParam *selectionMask;
```

Remarks

It returns or sets the array pointer of AsSelectionMaskParam Class' instance, which is wrapping the Selection Mask information corresponding to Gen2 regulation.

2.4.18. usedSelectionMask

Returns or sets whether the information of set Selection Mask is used or not in the inventory operation

Syntax

```
@property (nonatomic, assign) BOOLuseSelectionMask;
```

Remarks

If NO is set, the inventory method will perform the inventory on all of the general Tags.
If YES is set, it will perform only the tag filtered by Selection Mask.

2.4.19. removeSelectionMask

Delete Selection mask information in AsReader.

Syntax

```
- (void)removeSelectionMask:(int)index;
```

Parameters

index : Mask index parameter

Remarks

It deletes the Selection Mask of assigned index.

2.4.20. clearSelectionMask

Delete all of Selection mask information in AsReader.

Syntax

```
- (void)clearSelectionMask;
```

Remarks

Delete all of Selection mask information in AsReader.

2.4.21. isUseKeyAction

This method returns or set whether the AsReader device uses a hardware key or not.

Syntax

```
@property (nonatomic, assign) BOOLisUseKeyAction;
```

Remarks

If it is set to YES, AsReader device uses a hardware key.

If it is set to NO, no hardware key is used.

2.4.22. storedMode

This method returns or sets whether the Tag will be saved in the internal memory when performing the inventory.

Syntax

```
@property (nonatomic, assign) BOOLstoredMode;
```

Remarks

If storedMode is set to YES, the inventoried Tag information will be saved in the internal memory.

2.4.23. storedCount

It returns the number of Tag saved in the internal memory of AsReader.

Syntax

```
- (int)storedCount;
```

2.4.24. reportMode

This method returns or sets whether it occurs read Tag event when the stored memory reads Tag data while AsReader is performing the inventory.

Syntax

```
@property (nonatomic, assign) BOOLreportMode;
```

Remarks

If report mode is set to YES, read tag event does not occur when the stored memory reads the tag value.

If it is NO, read tag event occurs even if the stored memory reads the tag value.

2.4.25. batteryStatus

Returns the status of remaining battery of AsReader. The returned value is from 0 to 4. If 4 is shown, it is charging or fully charged.

Syntax

```
- (int)batteryStatus;
```

2.4.26. rssiMode

This method returns or sets whether AsReader reports RSSI value or not when it reads the tag value.

Syntax

```
@property (nonatomic, assign) BOOLrssiMode;
```

Remarks

If RSSI Mode is set to YES, readTag event returns RSSI value as well as Tag value.

If it is NO, only tag value will be returned. The default setting is NO.

2.4.27. clearEpcMask

Deletes all the EPC Mask Memory when AsReader is performing Inventory

Syntax

```
- (ResultCode)clearEpcMask;
```

2.4.28. saveEpcMask

Saves and applies the added EPC Mask Memory to AsReader

Syntax

```
- (ResultCode)saveEpcMask;
```

2.4.29. epcMaskCount

Returns the number of EPC Mask memory, which is saved in AsReader.

Syntax

```
- (int)epcMaskCount;
```

2.4.30. addEpcMask

Adds EPC Mask Memory that Masking will be applied to when AsReader is running Inventory.

Syntax

```
- (ResultCode)addEpcMask:(int) offset length:(int)length
mask:(NSString *)mask;
- (ResultCode)addEpcMask:(AsSelectMaskEPCParam *)masks;
```

Parameters

offset : Sets the initial address of Mask in bit unit. It must be 16bit or larger.

length : Sets the length of the Mask in bit unit.

mask : Sets the Mask value as Character string in HEX form.

masks : Array pointer of SelectMaskEpcParam class that contains the Mask information.

2.4.31. getEpcMask

Returns the EPC Mask information from the index given among its memory EPC Mask value in AsReader

Syntax

```
- (AsSelectMaskEPCParam *)getEpcMask:(int)index;
```

Parameters

index : Value where Memory EPC Mask is saved.

2.4.32. epcMaskMatchingMode

Returns Matching Mode status of epcMask in AsReader which filters whether EPC Mask matched Tag or unmatched one.

Syntax

```
@property (nonatomic, assign) BOOLepcMaskMatchMode;
```

Remarks

If epcMaskMatchingMode is set to YES, it will mask the matched value with epcMask.

If it is NO, it will mask unmatched value with epcMask.

2.4.33. getLbtMask

This method returns a mask value array of the frequency table information.

Syntax

```
- (NSArray *)getLBTMask;
```

2.4.34. getLbt

This method returns frequency table information set in the AsReader.

Syntax

```
- (NSArray *)getLBT;
```

2.4.35. setLbt

This method sets up the frequency table information

Syntax

```
- (void)setLBT:(NSArray *)table;
```

Parameters

Table : Parameter for frequency table settings.

2.4.36. getLbtFrequency

This method returns frequency information of AsReader.

Syntax

```
- (NSString *)getLBTFrequency:(int)slot;
```

2.5. Barcode Methods

2.5.1. setBarcodeMode

This method sets Barcode Scanner enable.

Syntax

```
- (ResultCode)setBarcodeMode:(BOOL)enabled;
```

Parameter

enabled : BOOL type to decide enable/disable Barcode Mode.

Remarks

If setBarcodeMode is set to YES, Barcode, an option device, cannot run commands related to UHF functions but other Barcode commands, which are followed by.

If setBarcodeMode is set to NO, it is able to command related to UFH.

2.5.2. startDecode

To start Barcode decoding.

Syntax

```
- (ResultCode)startDecode;
```

Remarks

If startDecode method works normally, Barcode, an option device, launches an aimer and scans a barcode. To stop the Barcode scan, use stopDecode method.
Once a barcode is scanned, scanned barcode information by AsReaderEventListener's onDetectBarcode will be returned. It will return No Read event by DetectBarcode method if the barcode scan is failed.

2.5.3. stopDecode

To Stop Barcode decoding.

Syntax

```
- (ResultCode)stopDecode;
```

Remarks

It stops the barcode scan while the barcode module, an option device of AsReader, is scanning.

2.5.4. setBarcodeParam

This method saves the settings in the barcode module.

Syntax

```
- (ResultCode)setBarcodeParam:(NSArray *)paramData;
```

Parameters

params : The pointer of NSArray object which is containing the barcode settings.

Remarks

setBarcodeParam method has the Barcode Module settings in the byte array type.

It wraps the byte array of Barcode Option settings and transmits it to NSArray object.

2.5.5. getBarcodeParam

This method returns the settings in barcode module

Syntax

```
- (NSArray *)getBarcodeParam:(NSArray *)paramData;
```

Parameters

names : NSArray object's pointer that contains the name of settings to be returned.

Remarks

getBarcodeParam method inquires the settings in the barcode module. It wraps the barcode parameter of code byte array into the NSArray object and transmit so that it could return the byte array in order of names and settings as the NSArray object.

2.5.6. isBarcodeModule

This method returns the barcode module available or unavailable

Syntax

```
- (BOOL) isBarcodeModule;
```

Parameters

names : return now AsReader barcode module status is available or unavailable

Remarks

This method returns barcode module status , when readerInitialized method return the barcode module status changed ;

3. AsReaderGUN Class

AsReaderGUN Class represents AsReaderGUN device hardware on AsRing Accessory Control Protocol(AACP) layer, provides software interfaces for controlling the communication between iOS and AsReader. .

3.1. Properties

3.1.1. delegate

A delegate to receive event notification or data from AsRing+.

Syntax

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

Remarks

Set or get an instance of delegate which implements AsReaderGUNReadDataDelegate .

3.1.2. deviceModel

Returns the name of model identifier of AsReader.

Syntax

```
- (NSString *)deviceModel;
```

Remarks

Returns the model identifier of AsReader.

3.1.3. address

Returns the address of AsReader assigned by AsRing+.

Syntax

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

Remarks

Returns the address of AsReader assigned by AsRing+

3.2. Methods

3.2.1. disconnect

Terminates the connection with AsReader.

Syntax

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

Remarks

It terminates the connection with AsReader and invalidates the object.

3.2.2. writeData

Transmits data to AsReader.

Syntax

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

Parameter

data : NSData object containing the data to be transmitted.

Remarks

It transmits data to AsReader by writeData method and receives data by AsRingDeviceReadDataDelegate interface.

4. CMinMaxValue

4.1. Properties

4.1.1. min

Syntax

```
int min;
```

Remarks

Storing minimum value.

4.1.2. max

Syntax

```
int max;
```

Remarks

Storing maximum value.

5. LockParam

5.1. Properties

5.1.1. killPassword

Returns or sets whether or not to control kill Password area.

Syntax

```
@property (nonatomic) BOOLkillPassword;
```

Remarks

It assigns whether a designated area will be the Kill Password area through lock, unlock or permalock method.

5.1.2. accessPassword

Returns or sets whether or not to control Access Password area.

Syntax

```
@property (nonatomic) BOOLaccessPassword;
```

Remarks

It assigns whether a designated area will be the Access Password area through lock, unlock or permalock method.

5.1.3. epc

Returns or sets whether or not to control EPC memory.

Syntax

```
@property (nonatomic) BOOLepc;
```

Remarks

It assigns whether a designated area will be the EPC memory area through lock, unlock or permalock method.

5.1.4. tid

Returns or sets whether or not to control TID memory.

Syntax

```
@property (nonatomic) BOOLtid;
```

Remarks

It assigns whether a designated area will be the TID memory area through lock, unlock or permalock method.

5.1.5. user

Returns or sets whether or not to control User memory.

Syntax

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

Remarks

It assigns whether a designated area will be the user memory area through lock, unlock or permalock method.

6. AsResultCode

This is a utility class for converting message string on ResultCode enumeration type.

6.1. Methods

6.1.1. msg

Syntax

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

Parameters

code : ResultCode enumeration type to be converted to a message.

Remarks

Returns string of ResultCode.

7. AsResultData

This is a ResultCode-returning class that is used when method returns ResultCode and data.

7.1. Properties

7.1.1. result

Syntax

```
ResultCodemResult;
```

Remarks

Shows returned ResultCode.

7.1.2. mData

Syntax

```
NSData * mData;
```

Remarks

NSDataobject that shows returned data.

8. AsSelectMaskParam

AsSelectMaskParam class is an interface class, which can be used to return or set selection Mask on Gen2 standard in AsReader. AsReader can set up to maximum of 8 Selection Masks.

8.1. Initialize Methods

8.1.1. initWithIndex

Create a standard AsSelectMaskParam object with radix value of Selection Mask.

Syntax

```
- (id)initWithIndex:(int)index;
```

Parameters

index : Integer type which shows the order of Selection Mask that has 0 - 7 values.

Remarks

initWithIndex method creates the instance of initialized AsSelectMaskParam Class with index value and basic Selection Mask value.

8.1.2. initWithParameterIndex

Creates AsSelectMaskParam object which makes up Selection mask information with given Parameter values.

Syntax

```
- (id)initWithParameterIndex:(int)index
target:(MaskTargetType)maskTarget
action:(MaskActionType)maskAction bank:(MemoryBank)maskBank
offset:(int)maskOffset mask:(NSString *)maskData
used:(BOOL)usedMask;
```

Parameters

dex : Integer type, which shows Selection Mask order that has a value of 0-7.

maskTarget : Enumeration type of MaskTargetType that shows Session of Tag when Selection Mask is applied.

maskAction : Enumeration type of MaskActionType that shows Session of Tag when Selection Mask is applied.

maskBank : Enumeration type of MemoryBank that shows memorybank of Tag which Mask data of Selection Mask will be compared.

maskOffset : Integer type (unit: bit) which shows the initial address where Mask data of Selection Mask will be compared.

maskData : NSString type of Hex form which shows Mask data which is set as comparison for Selection Mask

maskLength : Integer type (unit: bit) which shows the length that will be used to compare with Mask data which is set as comparison.

usedMask : BOOL type that shows whether to use created AsSelectMaskParam object as Selection Mask from AsReader.

Remarks

With given information of Parameter, it creates AsSelectMaskParam object. Give attention that the length of mask is assigned by the bit. To make PC portion as an initial address of Selection Mask, set 16 for the initial address since 1word is PC value at 1word spot in EPC memory bank. Selection-Masked data compares the all given Mask data by length.

8.1.3. initWithParameterLength

With given Parameter values, it creates AsSelectMaskParam objects that make up Selection mask information.

Syntax

```
- (id)initWithParameterLength:(int)index
target:(MaskTargetType)maskTarget
action:(MaskActionType)maskAction bank:(MemoryBank)maskBank
offset:(int)maskOffset mask:(NSString *)maskData
length:(int)maskLength used:(BOOL)usedMask;
```

Parameters

index : Integer type that shows Selection Mask order that has a value of 0-7.

maskTarget : Enumeration type of MaskTargetType that shows Session of Tag when Selection Mask is applied.

maskAction : Enumeration type of MaskActionType that shows Session of Tag when Selection Mask is applied.

maskBank : Enumeration type of MemoryBank that shows memorybank of Tag which Mask data of Selection Mask will be compared.

maskOffset : Integer type (unit: bit) which shows the initial address where Mask data of Selection Mask will be compared.

maskData : NSString type of Hex form which shows Mask data which is set as comparison for Selection Mask

maskLength : Integer type (unit: bit) which shows the length that will be used to compare with Mask data which is set as comparison.

usedMask : BOOL type that shows whether to use created AsSelectMaskParam object as Selection Mask from AsReader.

Remarks

With given information of Parameter, it creates AsSelectMaskParam object. Give attention that the length of mask is assigned by the bit. To make PC portion as an initial address of Selection Mask, set 16 for the initial address and the length since 1word is PC value at 1word spot in EPC memory bank. This method can be used to assign a part of the give mask data.

8.2. Properties

8.2.1. index

Returns notation that shows the sequence of Selection Mask.

Syntax

```
- (int)index;
```

8.2.2. target

Returns or sets the session of Tag where Selection Mask will be applied.

Syntax

```
@property (nonatomic) MaskTargetType target;
```

8.2.3. action

Returns or sets the Session status of Tag when Selection Mask is applied

Syntax

```
@property (nonatomic) MaskActionType action;
```

8.2.4. bank

Returns or sets memory bank of Tag that will be compared with Mask data of Seletion Mask.

Syntax

```
@property (nonatomic) MemoryBank bank;
```

8.2.5. offset

Returns or sets the initial address by the bit where Mask data of Selection Mask will start to be compared with.

Syntax

```
@property (nonatomic) int offset;
```

8.2.6. mask

Returns or sets Mask data as Hex type string which is compared with Selection Mask.

Syntax

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

8.2.7. length

Returns or sets the length that will be used in comparing Mask data of comparison target.

Syntax

```
@property (nonatomic) int length;
```

8.2.8. used

Returns or sets whether or not to use Selection Mask at AsReader

Syntax

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

9. AsSelectMaskEPCParam

With Selection Mask of Gen2 standard, AsReader can only do up to maximum of 8 Seletion Mask. However, with a way of internal implementation in comparing just EPC value, it provides a function that can mask up to maximum of 100 EPC value. This function enables AsSelectMaskEPCParam class, which is an interface class to return or set EPC Mask in AsReader.

9.1. Properties

9.1.1. offset

Returns or sets the initial address by the bit where EPC data of EPC Mask will start to be compared with.

Syntax

```
@property (nonatomic) int offset;
```

9.1.2. length

Returns or sets the length that will be used in comparing EPC data of comparison target.

Syntax

```
@property (nonatomic) int length;
```

9.1.3. mask

Returns or sets EPC data as Hex type string which is compared with EPC Mask.

Syntax

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

10. LbtItem

10.1. Initialize Methods

10.1.1. init

CreatesLbtItem object.

Syntax

```
-(id)init;
```

10.1.2. initWithSlot

With a given Parameter, it creates initialized LbtItem object.

Syntax

```
-(id)initWithSlot:(int)slot isUsed:(BOOL)isUsed;
```

Parameters

slot : It is Integer type which shows frequency slot that sets frequency for LBT frequency table.

isUsed : It is BOOL type that asks whether to use frequency of designated slot of LBT frequency table.

Remarks

It creates initialized LbtItem object by specifying whether to use slot numbers.

10.2. Properties

10.2.1. mSlot

Returns or sets notation that shows the location of the frequency of LBT frequency table.

Syntax

```
@property (nonatomic) int mSlot;
```

10.2.2. mIsUsed

Returns or sets whether to use a frequency designated on the LBT frequency table.

Syntax

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

11. Deletegate Interfaces

11.1. AsReaderDelegate

11.1.1. readerInitialized

This is an interface method to notify the completion of barcode module of initialization AsReader object.

Syntax

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

Parameters

reader : Updated AsReader barcode module object with the initialization.

Remarks

When AsReader object is connected with AsReader and its initialization is finished and barcode module is initialization finished, it will call

11.1.2. updateDeviceState

This is an interface method to notify the end of the operation of asynchronous working for loadStoredTag, which is Acting Method, or saveStoredTag method or removeAllStoredTags method.

Syntax

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

Parameters

ResultCode enumeration type that shows an updated result of AsReader

Remarks

Since it takes long working time to access to stored tag memory, it will call when the accessing is finished for notifying the result of the operation asynchronously.

11.1.3. readTag

It is an interface method to return EPC data of Tag that reads by AsReader during the inventory working.

Syntax

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

Parameters

tag : Character string of Hex type of PC+EPC data of Tag which AsReader read.

rssi : RSSI Value of Tag that AsReader read

phase : Phase value of Tag which AsReader read

Remarks

Call inventory method and if AsReader reads Tag then the event is called.

rssi and phase is added in v1.3.X or higher.

11.1.4. changedActionState

This is an interface method to remind change of operational status in AsReader.

Syntax

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

Parameters

action : CommandType enumeration type shows operational status in AsReader , resultCode enumeration type shows command status.

Remarks

Operational status of AsReader changes when inventory, readMemory, writeMemory, lock, unlock, permaLock, etc. methods are called. The event will be called to notify the user of it.

11.1.5. detectBarcode

This is an interface method to notify when AsReader scans a barcode in barcode mode

Syntax

```
- (void)detectBarcode:(BarcodeType)barcodeTypecodeId:(NSString
*)codeId barcode:(NSString *)barcode;
```

Parameters

barcodeType : BarcodeType enumeration type that shows types of scanned Barcode.

codeId : Character string that shows Code ID of scanned Barcode.

barcode : ASCII character string that shows data of scanned Barcode.

Remarks

This event called when AsReader is in Barcode Mode and calls startDecode method and Barcode Scanner, which is an Option Device, reads Barcode. Barcode Data is automatically decoded inside of AsReader and returned as ASCII character string type.

11.1.6. detectBarcode

This is an interface method to notify when AsReader scans a barcode NSdate in barcode mode

Syntax

```
- (void)detectBarcode:(BarcodeType)barcodeTypecodeId:(NSString
*)codeId barcodeData:(NSdata *)barcodeData;
```

Parameters

barcodeType : BarcodeType enumeration type that shows types of scanned Barcode.

codeId : Character string that shows Code ID of scanned Barcode.

`barcodeData`**:** ASCII character nsdata that shows data of scanned Barcodedata.

Remarks

This event called when AsReader is in Barcode Mode and calls startDecode method and Barcode Scanner, which is an Option Device, reads Barcode. Barcode Data is automatically decoded inside of AsReader and returned as ASCII character string type.

11.1.7. accessResult

This is an interface method to notify the result whether AsReader accesses Tag.

Syntax

```
- (void)accessResult:(ResultCode)error
actionState:(CommandType)action epc:(NSString *)epc data:(NSString
*)data rssi:(float)rssi phase:(float)phase;
```

Parameters

error : ResultCode enumeration type which shows the result of stopped operating Access

action : Enumeration type of commandType which shows what AsReader operation result it is.

epc : Character string of Hex type which shows Tag EPC Value that shows Tag is accessed by AsReader.

data : If it is Tag Access to read data in Tag like readMemory method, it is string of Hex type for returning data which was read in Tag

rssi : RSSI value of which AsReader accessed Tag.

phase : Phase value of which AsReader accessed Tag.

Remarks

accessResult event will be called as a result of accessing to Tag as to methods for accessing to Tag like readMemory, writeMemory, lock, unlock, permaLock, kill methods. In this event, inventory method is not included. As a result of readMemory method, Data, which reads in Tag, is also included. Rssi and phase are added in v1.3.X or higher

11.1.8. onAsReaderLeftModeKeyEvent

This method is *to notify the asreader click the left button method*

Syntax

```
- (BOOL)onAsReaderLeftModeKeyEvent:(BOOL)status;
```

Parameters

`status: YES press,NO release`

Remarks

onAsReaderLeftModeKeyEvent event will be called the left button press or lift. This deleg ate can be override on application level, the default implementation in SDK is the switch the scan mode between RFID and Barcode.

11.1.9. onAsReaderRightModeKeyEvent

This method is *to notify the asreader click the right button method*

Syntax

```
- (BOOL)onAsReaderRightModeKeyEvent:(BOOL)status;
```

Parameters

`status: YES press,NO release`

Remarks

onAsReaderRightModeKeyEvent event will be called the Right button press or lift. This del egate can be override on application level, the default implementation in SDK is the switc h the scan mode between RFID and Barcode.

11.1.10. onAsReaderTriggerKeyEvent

This method is *to notify the asreader click the trigger button method*

Syntax

```
- (BOOL)onAsReaderTriggerKeyEvent:(BOOL)status;
```

Parameters

status: YES press,NO release

Remarks

onAsReaderTriggerKeyEvent event will be called the trigger button press or lift. This delegate can be override on application level, the default implementation in SDK is start scanning RFID or Barcode according to the scan mode setting

12. Enumerators

12.1. ResultCode

It shows the operation result of order or property of AsReader.

Flag	Value	Description
ResultNoError	0x0000	Succeed in result.
ResultOtherError	0x0001	An error has occurred due to unknown reason.
ResultUndefined	0x0002	An Undefined Error
ResultMemoryOverrun	0x0003	Accessing to memory out of range.
ResultMemoryLocked	0x0004	Memory is locked.
ResultInsufficientPower	0x000B	Battery power is low.
ResultNonSpecificError	0x000F	Not a specific error.
ResultInOperation	0xE000	In operation.
ResultOutOfRange	0xE001	Out of range.
ResultNotConnected	0xE100	Not connected to Device.
ResultInvalidParameter	0xE200	Invalid parameter transmitted.
ResultInvalidResponse	0xE300	Returned invalid parameter.
ResultNotSupportFirmware	0xEE00	Unsupported firmware.
ResultTimeout	0xEFFF	Exceeded allowed accessing time.
ResultHandleMismatch	0xF001	Handle mismatch.
ResultCRCError	0xF002	CRC error on tag response.
ResultNoTagReply	0xF003	No Tag Reply.
ResultInvalidPassword	0xF004	Invalid password.
ResultZeroKillPassword	0xF005	Zero kill password.
ResultTagLost	0xF006	Tag lost.
ResultCommandFormatError	0xF007	Command format error.
ResultReadCountInvalid	0xF008	Read count invalid.
ResultOutOfRetries	0xF009	Out of retries.
ResultParamError	0xFFFB	Parameter error.
ResultBusy	0xFFFC	Busy.
ResultInvalidCommand	0xFFFD	Invalid Command.
ResultLowBattery	0xFFFE	Low Battery
ResultOperationFailed	0xFFFF	Operation failed

12.2. MemoryBank

It shows the memory bank of Tag which will access to from AsReader.

Flag	Value	Description
Bank_Reserved	0	Refers to Reseved memory Bank.
Bank_EPC	1	Refers to EPC memory Bank.
Bank_TID	2	Refers to TID memory Bank.

Bank_User	3	Refers to User memory Bank.

12.3. BuzzerState

It defines the status of Buzzer when controlling AsReader's Buzzer.

Flag	Value	Description
Buzzer_Off	0	Turn off Buzzer.
Buzzer_Low	1	Turn Buzzer down.
Buzzer_High	2	Turn Buzzer up.

12.4. VibratorState

It defines the status of Vibrator when controlling AsReader's Vibrator.

Flag	Value	Description
Vibrator_Off	0	Turn off Vibrator.
Vibrator_On	1	Turn Vibrator down.

12.5. SessionType

It shows the session of target Tag when AsReader performs Inventory order.

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

12.6. SessionFlag

It shows the status of session flag which is targeted by Selection Mask of AsReader.

Flag	Value	Description
SessionFlag_AB	0	A or B
SessionFlag_A	1	A only
SessionFlag_B	2	B only

12.7. MaskTargetType

IT shows the Selection Mask target session of AsReader.

Flag	Value	Description
MaskTarget_S0	0	inventoried S0
MaskTarget_S1	1	inventoried S1
MaskTarget_S2	2	inventoried S2
MaskTarget_S3	3	inventoried S3
MaskTarget_SL	4	Selection Flags

12.8. MaskActionType

It shows the action status of Selection Mask of AsReader.

Flag	Value	Description
MaskAction_AB	0	Tag Matching : assert SL or inventoried → A Tag Not-Matching : retract SL or inventoried → B
MaskAction_AN	1	Tag Matching : assert SL or inventoried → A Tag Not-Matching : do nothing
MaskAction_NB	2	Tag Matching : do nothing Tag Not-Matching : retract SL or inventoried → B
MaskAction_MN	3	Tag Matching : negate SL or (A → B, B → A) Tag Not-Matching : do nothing
MaskAction_BA	4	Tag Matching : retract SL or inventoried → B Tag Not-Matching : assert SL or inventoried → A
MaskAction_BN	5	Tag Matching : retract SL or inventoried → B Tag Not-Matching : do nothing
MaskAction_NA	6	Tag Matching : do nothing Tag Not-Matching : assert SL or inventoried → A
MaskAction_NM	7	Tag Matching : do nothing Tag Not-Matching : negate SL or (A → B, B → A)

12.9. MaskType

It shows the function of Selection Mask of AsReader.

Flag	Value	Description
MaskType_Selection	0	Gen2 standard Mask Selection.
MaskType_EPC	1	EPC Mask Selection.

12.10. AlgorithmType

IT shows Q value config.

Flag	Value	Description
FixedQ	0	Fixed Q
DynamicQ	1	Dynamic Q

12.11. CommandType

It shows which asynchronous order of AsReader has been completed.

Flag	Value	Description
CommandInventory	0x66	Inventory in progress.
CommandReadMemory	0x72	Read Memory in progress.
CommandWriteMemory	0x77	Write Memory in progress.
CommandKill	0x6B	Kill Tag in progress.
CommandLock	0x6C	Lock in progress.
CommandUnlock	0x6D	Unlock in progress.
CommandPermaLock	0x70	PermaLock in progress.
CommandStop	0x73	Operation Stopped.