

**AsReader**®

# AsReaderGUN

## SDK Reference Guide

Applies to ASR-R250G and ASR-L251G

## Revision History

| Version | Modified Contents   | Date       |
|---------|---|------------|
| 1.0     | Initial version   | 2020/9/18  |
| 1.1     | Add an appendix   | 2020/10/20 |
| 1.2     | Delete "stopDecode"<br>Add "setBarcodeMode (With Custom prefix suffix )"  | 2021/4/20  |
| 1.3     | Add description of "continuousMode"   | 2022/4/7   |
| 1.4     | Several considerations have been added to the sections "3.2.4 setDelegate" and "4.1 readerInitialized".<br>Add the function getTagState, TagState | 2024/9/5   |

# Contents

|   |           |
|---|-----------|
| <b>1. Preparation for SDK Usage</b>                                 | <b>8</b>  |
| 1.1. Add SDK  | 8         |
| 1.2. Import the Header File   | 10        |
| <b>2. AsReaderGUN Class</b>   | <b>11</b> |
| 2.1. Properties   | 11        |
| 2.1.1. @property(strong, nonatomic) NSString *deviceModel;          | 11        |
| 2.1.2. @property (assign, nonatomic, readonly) BOOL isConnect;      | 11        |
| 2.2. Function   | 11        |
| 2.2.1. initWithDeviceModel  | 11        |
| 2.2.2. deviceModel  | 12        |
| 2.2.3. address  | 12        |
| 2.2.4. getAsReaderGUNVersion  | 12        |
| <b>3. AsReader Class</b>  | <b>14</b> |
| 3.1. Properties   | 14        |
| 3.1.1. @property (nonatomic, strong) AsReaderGUN *mAsReaderGUN;     | 14        |
| 3.1.2. @property (nonatomic, assign) BuzzerState buzzer;            | 14        |
| 3.1.3. @property (nonatomic, assign) VibratorState vibrator;        | 14        |
| 3.1.4. @property (nonatomic, assign) int operationTime;             | 14        |
| 3.1.5. @property (nonatomic, assign) int inventoryTime;             | 14        |
| 3.1.6. @property (nonatomic, assign) int idleTime;                  | 15        |
| 3.1.7. @property (nonatomic, assign) int sleepTime;                 | 15        |
| 3.1.8. @property (nonatomic, assign) int autoOffTime;               | 15        |
| 3.1.9. @property (nonatomic, strong) NSString *accessPassword;      | 15        |
| 3.1.10. @property (nonatomic, assign) SessionType inventorySession; | 15        |

|  |    |
|--|----|
| 3.1.11. @property (nonatomic, assign) SessionFlag sessionFlag;     | 15 |
| 3.1.12. @property (nonatomic, strong) NSString *serialNumber;      | 16 |
| 3.1.13. @property (nonatomic, assign) BOOL continuousMode;         | 16 |
| 3.1.14. @property (nonatomic, assign) int powerGain;               | 16 |
| 3.1.15. @property (nonatomic, assign) BOOL isUseKeyAction;         | 16 |
| 3.1.16. @property (nonatomic, assign) SelectFlag useSelectionMask; | 16 |
| 3.1.17. @property (nonatomic, assign) BOOL rssiMode;               | 17 |
| 3.1.18. @property (nonatomic, assign) BOOL epcMaskMatchMode;       | 17 |
| 3.1.19. @property (nonatomic, assign) AlgorithmType algorithm;     | 17 |
| 3.1.20. @property (nonatomic, assign) int minQ;                    | 17 |
| 3.1.21. @property (nonatomic, assign) int maxQ;                    | 17 |
| 3.1.22. @property (nonatomic, assign) int qValue;                  | 17 |
| 3.1.23. @property (nonatomic, assign) int linkProfileValue;        | 18 |
| 3.1.24. @property (nonatomic, assign) int defaultLinkProfileValue; | 18 |
| 3.1.25. @property (nonatomic, assign) int maskTypeValue;           | 18 |
| 3.2. Function  | 18 |
| 3.2.1. initWithAsReaderGUN   | 18 |
| 3.2.2. disconnect  | 19 |
| 3.2.3. getAction   | 19 |
| 3.2.4. setDelegate   | 20 |
| 3.2.5. setScanMode   | 20 |
| 3.2.6. getScanMode   | 20 |
| 3.2.7. inventory   | 21 |
| 3.2.8. readMemory  | 21 |
| 3.2.9. writeMemory   | 22 |
| 3.2.10. lock   | 23 |
| 3.2.11. unlock   | 23 |

|                                       |    |
|---------------------------------------|----|
| 3.2.12. permaLock                     | 24 |
| 3.2.13. kill                          | 24 |
| 3.2.14. stop                          | 25 |
| 3.2.15. stopSync                      | 25 |
| 3.2.16. defaultParameter              | 26 |
| 3.2.17. saveParameter                 | 26 |
| 3.2.18. regionName                    | 27 |
| 3.2.19. serialNumber                  | 27 |
| 3.2.20. rFModuleVersion               | 28 |
| 3.2.21. firmwareVersion               | 28 |
| 3.2.22. powerGainRange                | 28 |
| 3.2.23. batteryStatus                 | 29 |
| 3.2.24. clearEpcMask                  | 29 |
| 3.2.25. saveEpcMask                   | 29 |
| 3.2.26. epcMaskCount                  | 30 |
| 3.2.27. addEpcMask                    | 30 |
| 3.2.28. addEpcMask                    | 31 |
| 3.2.29. getEpcMask                    | 31 |
| 3.2.30. getLBTMask                    | 32 |
| 3.2.31. getLBT                        | 32 |
| 3.2.32. setLBT                        | 32 |
| 3.2.33. getLBTFrequency               | 33 |
| 3.2.34. startBuzzerWithBuzzerTime     | 33 |
| 3.2.35. startVibratorWithVibratorTime | 34 |
| 3.2.36. startDecode                   | 34 |
| 3.2.37. setBarcodeParam               | 35 |
| 3.2.38. getBarcodeParam               | 36 |

|   |           |
|---|-----------|
| 3.2.39. usedSelectionMask                           | 36        |
| 3.2.40. getSelectionMask                            | 37        |
| 3.2.41. setSelectionMask                            | 37        |
| 3.2.42. removeSelectionMask                         | 37        |
| 3.2.43. clearSelectionMask                          | 38        |
| 3.2.44. getAlgorithm                                | 38        |
| 3.2.45. setBarcodeMode                              | 38        |
| 3.2.46. setBarcodeMode (With Custom prefix suffix ) | 39        |
| 3.2.47. isBarcodeModule                             | 40        |
| 3.2.48. isRFIDModule                                | 40        |
| 3.2.49. getTagState                                 | 41        |
| 3.3. Enum   | 42        |
| 3.3.1. AlgorithmType                                | 42        |
| <b>4. AsReaderDelegate Class</b>                    | <b>43</b> |
| 4.1. readerInitialized                              | 43        |
| 4.2. updateDeviceState                              | 43        |
| 4.3. readTag  | 44        |
| 4.4. changedActionState                             | 44        |
| 4.5. detectBarcode                                  | 45        |
| 4.6. detectBarcode                                  | 46        |
| 4.7. accessResult                                   | 46        |
| 4.8. onAsReaderLeftModeKeyEvent                     | 47        |
| 4.9. onAsReaderRightModeKeyEvent                    | 47        |
| 4.10. onAsReaderTriggerKeyEvent                     | 48        |
| <b>5. AsRfidValues Class</b>                        | <b>49</b> |
| 5.1. Enum   | 49        |

|                           |           |
|---------------------------|-----------|
| 5.1.1. ResultCode         | 49        |
| 5.1.2. MemoryBank         | 50        |
| 5.1.3. BuzzerState        | 50        |
| 5.1.4. VibratorState      | 50        |
| 5.1.5. SessionType        | 50        |
| 5.1.6. SessionFlag        | 51        |
| 5.1.7. SelectFlag         | 51        |
| 5.1.8. MaskTargetType     | 51        |
| 5.1.9. MaskActionType     | 51        |
| 5.1.10. MaskType          | 52        |
| 5.1.11. TagState          | 52        |
| 5.2. Struct               | 53        |
| 5.2.1. CMinMaxValue       | 53        |
| 5.3. LockParam            | 54        |
| 5.3.1. Properties         | 54        |
| 5.4. AsResultCode         | 55        |
| 5.4.1. Function           | 55        |
| 5.5. AsSelectMaskParam    | 55        |
| 5.5.1. Properties         | 55        |
| 5.5.2. Function           | 57        |
| 5.6. AsSelectMaskEPCParam | 59        |
| 5.6.1. Properties         | 59        |
| 5.7. LbtItem              | 60        |
| 5.7.1. Properties         | 60        |
| 5.7.2. Function           | 60        |
| <b>6. AsPacket Class</b>  | <b>62</b> |

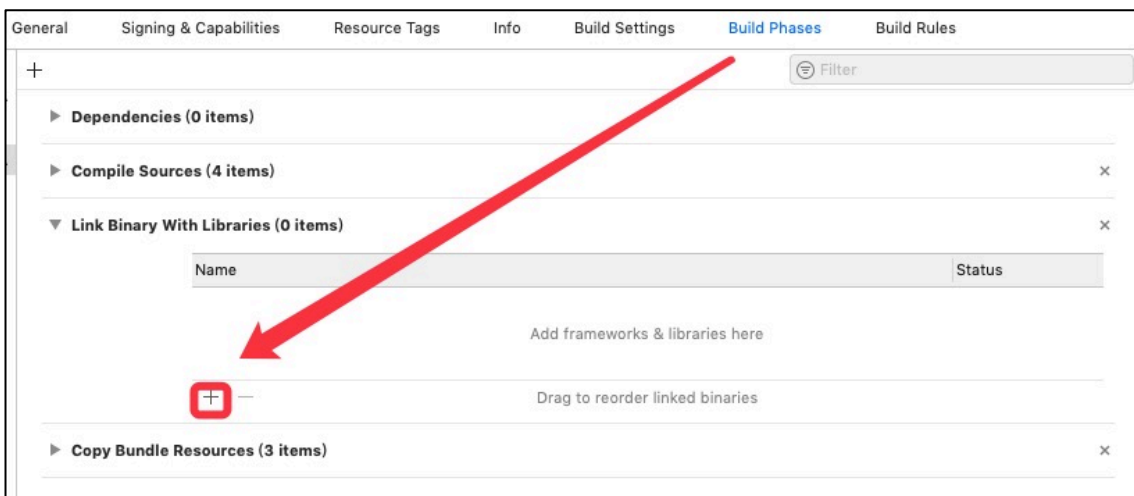
---

|   |           |
|---|-----------|
| 6.1. Enum   | 62        |
| 6.1.1. CommandType  | 62        |
| 6.1.2. ScanMode   | 62        |
| <b>7. AsBarcodeType Class</b>                             | <b>63</b> |
| 7.1. Function   | 63        |
| 7.1.1. getBarcodeString                                   | 63        |
| 7.2. Enum   | 63        |
| 7.2.1. BarcodeType  | 63        |
| <b>8. AsParamName Class</b>                               | <b>66</b> |
| 8.1. Function   | 66        |
| 8.1.1. getName  | 66        |
| 8.2. Enum   | 66        |
| 8.2.1. ParamName  | 66        |
| <b>9. AsParamValue Class</b>                              | <b>69</b> |
| 9.1. Properties   | 69        |
| 9.1.1. @property (assign, readwrite) ParamName paramName; | 69        |
| 9.1.2. @property (assign, readwrite) unsigned int value;  | 69        |
| 9.2. Function   | 70        |
| 9.2.1. setEnabled   | 70        |
| <b>10. AsMaskActionType Class</b>                         | <b>71</b> |
| 10.1. Function  | 71        |
| 10.1.1. toString  | 71        |
| <b>11. Appendix</b>                                       | <b>72</b> |
| 11.1. Parameter Informations                              | 72        |

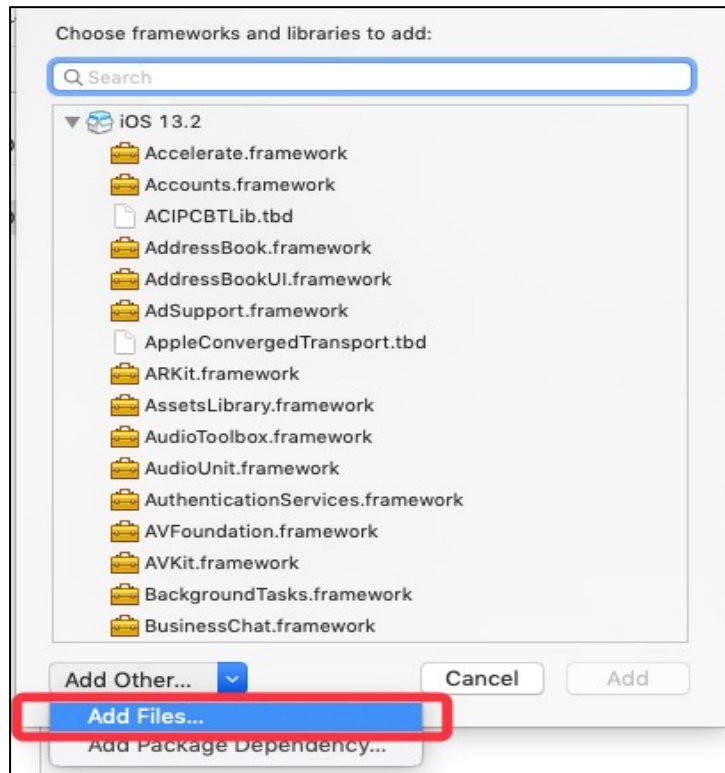
# 1. Preparation for SDK Usage

## 1.1. Add SDK

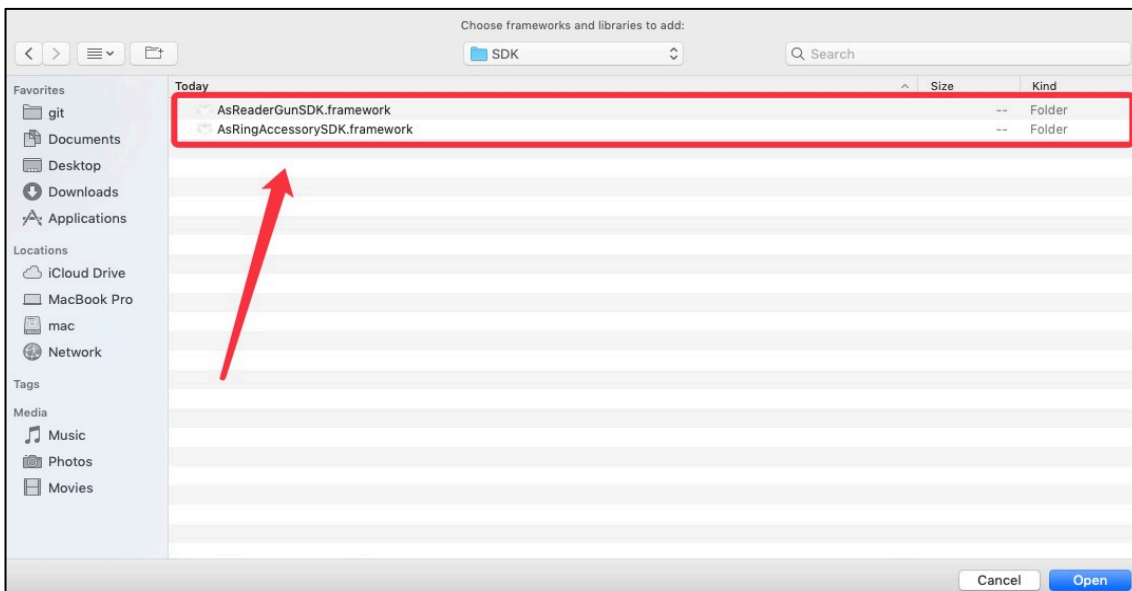
1. TARGET -> Build phases -> Link Binary with Libraries



2. Select “Add Other...”, “Add Files...”



3. Add AsReaderGUNSDK.framework, AsRingAccessorySDK.framework



**4. Complete as shown**



**5. Add Protocols**

In the info.plist file, add the protocols corresponding to the following devices to the "Supported external accessory protocols".

- jp.co.asx.asring.plus
- jp.co.asx.asreader.gun



**1.2. Import the Header File**

The ObjectiveC project needs to import the header file in the class where you want to use the SDK. Please refer to the following code:

```
#import <AsReaderGUNSDK/AsReaderGUNSDK.h>
```

## 2. AsReaderGUN Class

### 2.1. Properties

#### 2.1.1. @property(strong, nonatomic) NSString \*deviceModel;

**Description:**

Sets or gets the protocol for the connection of the AsReaderGUN accessories. (For example: com.asreader.gun)

#### 2.1.2. @property (assign, nonatomic, readonly) BOOL isConnected;

**Description:**

Gets the connection status of the AsReaderGUN.

YES: connected

NO: disconnected

### 2.2. Function

#### 2.2.1. initWithDeviceModel

| Function    | - (instancetype)initWithDeviceModel:(NSString *)deviceModel; |             |  |
|-------------|--|-------------|--|
| Parameters  | IN/OUT   | Types       | Descriptions   |
| deviceModel | IN   | NSString    | The connection protocol for the AsReaderGUN accessory. (For example: com.asreader.gun) |
| -           | OUT  | AsReaderGUN | AsReaderGUN object   |

**Function Description:**

This function is used to initialize the AsReaderGUN object.

**Sample code:**

```
AsReaderGUN * AsReaderGUN = [[AsReaderGUN alloc]
initWithDeviceModel:@"com.asreader.gun"];
```

### 2.2.2. deviceModel

|  |                            |              |  |
|--|----------------------------|--------------|--|
| <b>Function</b>  | - (NSString *)deviceModel; |              |  |
| <b>Parameters</b>  | <b>IN/OUT</b>              | <b>Types</b> | <b>Descriptions</b>                                    |
| -  | OUT                        | NSString     | The connection protocol for the AsReaderGUN accessory. |
| <p><b>Function Description:</b><br/>Gets the connection protocol for the AsReaderGUN accessory.</p> <p><b>Sample code:</b><br/>NSString * deviceModel = [AsReaderGUN deviceModel];</p> |                            |              |  |

### 2.2.3. address

|   |                        |              |  |
|---|------------------------|--------------|--|
| <b>Function</b>   | - (NSString *)address; |              |  |
| <b>Parameters</b>   | <b>IN/OUT</b>          | <b>Types</b> | <b>Descriptions</b>                          |
| -   | OUT                    | NSString     | The address AsRing+ assigned to AsReaderGUN. |
| <p><b>Function Description:</b><br/>Get the address AsRing+ assigned to AsReaderGUN.</p> <p><b>Sample code:</b><br/>NSString * address = [AsReaderGUN address];</p> |                        |              |  |

### 2.2.4. getAsReaderGUNVersion

|                   |                                      |              |                     |
|-------------------|--------------------------------------|--------------|---------------------|
| <b>Function</b>   | - (NSString *)getAsReaderGUNVersion; |              |                     |
| <b>Parameters</b> | <b>IN/OUT</b>                        | <b>Types</b> | <b>Descriptions</b> |
| -                 | OUT                                  | NSString     | SDK version         |

■ **Function Description:**

Get the version of the SDK

■ **Sample code:**

```
NSString * sdkVersion= [AsReaderGUN getAsReaderGUNVersion];
```

## 3. AsReader Class

### 3.1. Properties

For the storage destination and the default value of each item, please refer to Appendix [11.1](#).

#### 3.1.1. **@property (nonatomic, strong) AsReaderGUN \*mAsReaderGUN;**

**Description:**

Gets or sets the interface of AsReaderGUN.

#### 3.1.2. **@property (nonatomic, assign) BuzzerState buzzer;**

**Description:**

Gets or sets the buzzer state of AsReader.  
Enumerated type BuzzerState (see [5.1.3](#))

#### 3.1.3. **@property (nonatomic, assign) VibratorState vibrator;**

**Description:**

Gets or sets the vibrator state of AsReader.  
Enumerated type VibratorState (see [5.1.4](#))

#### 3.1.4. **@property (nonatomic, assign) int operationTime;**

**Description:**

Gets or sets the operation time of the RFID module of AsReaderGUN. (unit: ms)

#### 3.1.5. **@property (nonatomic, assign) int inventoryTime;**

**Description:**

Gets or sets the inventory time of the RFID module of AsReaderGUN. (unit: ms)

**3.1.6. @property (nonatomic, assign) int idleTime;**

**Description:**

Gets or sets the idle time of the RFID module of AsReaderGUN. (unit: ms)

**3.1.7. @property (nonatomic, assign) int sleepTime;**

**Description:**

Gets or sets the automatic sleep time of AsReaderGUN. (unit: s)

**Note:** ASR-R250G is timed when the connection is disconnected, and ASR-L251G is timed when the connection is successful and no operation has been performed.

**3.1.8. @property (nonatomic, assign) int autoOffTime;**

**Description:**

Gets or sets the automatic shut-off time after AsReaderGUN disconnects. (unit: s)

**3.1.9. @property (nonatomic, strong) NSString \*accessPassword;**

**Description:**

Gets or sets the Access password for the RFID tag.

**3.1.10. @property (nonatomic, assign) SessionType inventorySession;**

**Description:**

Gets or sets the Session value of the RFID module of AsReaderGUN.

Enumerated type SessionType (see [5.1.5](#))

**3.1.11. @property (nonatomic, assign) SessionFlag sessionFlag;**

**Description:**

Gets or sets the Session Flag of AsReaderGUN's RFID module.

Enumerated type SessionFlag (see [5.1.6](#))

**3.1.12. @property (nonatomic, strong) NSString \*serialNumber;****Description:**

Gets or sets the serial number of the AsReaderGUN device.

**3.1.13. @property (nonatomic, assign) BOOL continuousMode;****Description:**

Gets or sets whether the RFID module of AsReaderGUN reads continuously.

To use the Continuous mode, ContinuousMode=YES must be set in app.

For firmware reasons, the continuous scan mode is off by default each time the power is turned on or the AsReader and App are successfully connected.

YES: Continuous scan is valid

NO: Continuous scan is invalid

**3.1.14. @property (nonatomic, assign) int powerGain;****Description:**

Gets or sets the Power value for RFID tag inventory.

Struct CMinMaxValue (see [5.2.1](#))

**3.1.15. @property (nonatomic, assign) BOOL isUseKeyAction;****Description:**

Gets or sets whether the AsReaderGUN uses the hardware key.

YES: uses the hardware key.

NO: doesn't use the hardware key.

**3.1.16. @property (nonatomic, assign) SelectFlag useSelectionMask;****Description:**

Gets or sets the Select Flag of AsReaderGUN in the RFID module.

Enumerated type SelectFlag (see [5.1.7](#))

**3.1.17. @property (nonatomic, assign) BOOL rssiMode;****Description:**

Gets or sets whether to get RSSI data when inventorying RFID tags.

YES: get the RSSI data

NO: does not get RSSI data

**3.1.18. @property (nonatomic, assign) BOOL epcMaskMatchMode;****Description:**

Gets or sets whether to use the EPC Mask function

YES: use the EPC mask function

NO: does not use the EPC mask function

**3.1.19. @property (nonatomic, assign) AlgorithmType algorithm;****Description:**

Gets or sets the Q value to be fixed or dynamic when inventorying RFID tags.

Enumerated type AlgorithmType (see [3.3.1](#))

**3.1.20. @property (nonatomic, assign) int minQ;****Description:**

Gets or sets the min Q value when inventorying RFID tags.

Range: 0~15

**3.1.21. @property (nonatomic, assign) int maxQ;****Description:**

Gets or sets the max Q value when inventorying RFID tags.

Range: 0~15

**3.1.22. @property (nonatomic, assign) int qValue;****Description:**

Gets or sets the current Q value.

Range: 0~15

**3.1.23. @property (nonatomic, assign) int linkProfileValue;**

**Description:**  
 Gets or sets the current Link Profile value for the RFID module of AsReaderGUN.  
 Range: 0~3

**3.1.24. @property (nonatomic, assign) int defaultLinkProfileValue;**

**Description:**  
 Gets or sets the default Link Profile value for the RFID module of AsReaderGUN.  
 Range: 0~3  
**Note: Only support ASR-R250G.**

**3.1.25. @property (nonatomic, assign) int maskTypeValue;**

**Description:**  
 Gets or sets the Mask type for the RFID module of AsReaderGUN.  
 0: NoMask, 1: SelectionMask, 2: EPCMask

## 3.2. Function

### 3.2.1. initWithAsReaderGUN

|                   |  |                  |  |
|-------------------|--|------------------|--|
| <b>Function</b>   | -(id)initWithAsReaderGUN:(AsReaderGUN*)device delegate:(id<AsReaderDelegate>)delegate; |                  |  |
| <b>Parameters</b> | <b>IN/OUT</b>  | <b>Types</b>     | <b>Descriptions</b>  |
| device            | IN   | AsReaderGUN      | AsReaderGUN object.  |
| delegate          | IN   | AsReaderDelegate | Set a delegate to handle events such as AsReaderGUN's buttons being pressed or released. This delegate can receive information or notifications from the AsReaderGUN device. |

|  |     |          |                 |
|--|-----|----------|-----------------|
| -  | OUT | AsReader | AsReaderobject. |
| <p>■ <b>Function Description:</b><br/>The function used to initialize the AsReader object<br/><b>Note:</b> This function is called once receiving "AsReaderGUNConnected" notification.</p> <p>■ <b>Sample code:</b><br/>(Note: AsReaderGUN is the object of the AsReaderGUN class)</p> <pre>- (void)AsReaderGUNConnected:(NSNotification *)notification {     dispatch_async(dispatch_get_main_queue(), ^{         AsReader * asReader = [[AsReader alloc] initWithAsReaderGUN:AsReaderGUN delegate:self];     }); }</pre> |     |          |                 |

### 3.2.2. disconnect

|  |                     |              |                     |
|--|---------------------|--------------|---------------------|
| <b>Function</b>  | - (void)disconnect; |              |                     |
| <b>Parameters</b>  | <b>IN/OUT</b>       | <b>Types</b> | <b>Descriptions</b> |
| -  | -                   | -            | -                   |
| <p>■ <b>Function Description:</b><br/>Disconnect from the AsReaderGUN.</p> <p>■ <b>Sample code:</b> (Note: asReader is the object of the AsReader class.)<br/>[asReader disconnect];</p> |                     |              |                     |

### 3.2.3. getAction

|   |                           |              |   |
|---|---------------------------|--------------|---|
| <b>Function</b>   | - (CommandType)getAction; |              |   |
| <b>Parameters</b>   | <b>IN/OUT</b>             | <b>Types</b> | <b>Descriptions</b>   |
| -   | OUT                       | CommandType  | Command type<br>Enumerated type CommandType<br>(see <a href="#">6.1.1</a> ) |
| <p>■ <b>Function Description:</b><br/>Gets the status of the current action of the AsReader instance.</p> <p>■ <b>Sample code:</b> (Note: asReader is the object of AsReader class.)<br/>CommandType type = [asReader getAction];</p> |                           |              |   |

### 3.2.4. setDelegate

|  |   |                  |                     |
|--|---|------------------|---------------------|
| <b>Function</b>  | - (void)setDelegate:(id<AsReaderDelegate>)delegate; |                  |                     |
| <b>Parameters</b>  | <b>IN/OUT</b>                                       | <b>Types</b>     | <b>Descriptions</b> |
| delegate   | IN  | AsReaderDelegate | AsReaderDelegate    |
| <p>■ <b>Function Description:</b><br/>Set delegate for AsReader.</p> <p>■ <b>Sample code:</b> (Note: asReader is the object of AsReader class.)<br/>[asReader setDelegate:self];</p> <p><b>Note:</b> This delegate must be set, otherwise data cannot be returned.</p> |   |                  |                     |

### 3.2.5. setScanMode

|  |   |              |   |
|--|---|--------------|---|
| <b>Function</b>  | - (void)setScanMode:(ScanMode)scanMode; |              |   |
| <b>Parameters</b>  | <b>IN/OUT</b>                           | <b>Types</b> | <b>Descriptions</b>   |
| scanMode   | IN                                      | ScanMode     | Scan mode.<br>Enumerated type ScanMode (see <a href="#">6.1.2</a> ) |
| <p>■ <b>Function Description:</b><br/>Set the scan mode of AsReaderGUN.</p> <p>■ <b>Sample code:</b> (Note: asReader is the object of AsReader class.)<br/>[asReader setScanMode: RFIDScanMode];</p> |   |              |   |

### 3.2.6. getScanMode

|   |                          |              |   |
|---|--------------------------|--------------|---|
| <b>Function</b>   | - (ScanMode)getScanMode; |              |   |
| <b>Parameters</b>   | <b>IN/OUT</b>            | <b>Types</b> | <b>Descriptions</b>   |
| -   | OUT                      | ScanMode     | Scan mode.<br>Enumerated type ScanMode (see <a href="#">6.1.2</a> ) |
| <p>■ <b>Function Description:</b><br/>Gets the current scan mode of AsReaderGUN.</p> <p>■ <b>Sample code:</b> (Note: asReader is the object of AsReader class.)<br/>ScanMode mode = [asReader getScanMode];</p> |                          |              |   |

### 3.2.7. inventory

| Function   | -(ResultCode)inventory; |            |  |
|--|-------------------------|------------|--|
| Parameters   | IN/OUT                  | Types      | Descriptions   |
| -  | OUT                     | ResultCode | The result that will be returned when the function is called.<br>Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |
| <p><b>■ Function Description:</b><br/>Start to inventory tags.<br/>It is called back in the delegate functions changedActionState (see <a href="#">4.4</a>) and readTag (see <a href="#">4.3</a>).</p> <p><b>■ Sample code:</b> (Note: asReader is the object of AsReader class.)<br/> <pre>ResultCode resultCode = [asreader inventory]; if (resultCode == ResultNoError) {     //Called function successfully. }else{     //Called function failed. }</pre> </p> |                         |            |  |

### 3.2.8. readMemory

| Function   | -(ResultCode)readMemory:(MemoryBank)bank offset:(int)offset length:(int)length; |            |  |
|------------|---|------------|--|
| Parameters | IN/OUT  | Types      | Descriptions   |
| bank       | IN  | MemoryBank | The memory banks of a tag.<br>Enumerated type MemoryBank (see <a href="#">5.1.2</a> )                                    |
| offset     | IN  | int        | The offset of target banks. (Unit: word)   |
| length     | IN  | int        | Length of data to be read. (Unit: word)  |
| -          | OUT   | ResultCode | The result that will be returned when the function is called.<br>Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |

**Function Description:**

This function is used to inventory the data of the specific memory bank of the RFID tag.

It is called back in the delegate function `accessResult` (see [4.7](#)).

**Sample code:** (Note: `asReader` is the object of `AsReader` class.)

```

ResultCode resultCode = [asreader readMemory:Bank_EPC offset:16 length:4];
if (resultCode == ResultNoError) {
    //Called function successfully.
}else{
    //Called function failed.
}
    
```

### 3.2.9. writeMemory

| Function   | -(ResultCode)writeMemory:(MemoryBank)bank offset:(int)offset value:(NSString *)value; |            |  |
|------------|---|------------|--|
| Parameters | IN/OUT  | Types      | Descriptions   |
| bank       | IN  | MemoryBank | The memory banks of a tag. Enumerated type <code>MemoryBank</code> (see <a href="#">5.1.2</a> )                                    |
| offset     | IN  | int        | The offset of target banks. (Unit: word)   |
| value      | IN  | NSString   | Data to write. (the data is a hexadecimal value string)  |
| -          | OUT   | ResultCode | The result that will be returned when the function is called. Enumerated type <code>ResultCode</code> (see <a href="#">5.1.1</a> ) |

**Function Description:**

This function is used to write data to the specific memory bank of the RFID tag.

It is called back in the delegate function `accessResult` (see [4.7](#)).

**Sample code:** (Note: `asReader` is the object of `AsReader` class.)

```

ResultCode resultCode = [asreader writeMemory:Bank_EPC offset:16
value:@"1234"];
if (resultCode == ResultNoError) {
    //Called function successfully.
}else{
    //Called function failed.
}
    
```

### 3.2.10. lock

| Function  | - (ResultCode)lock:(LockParam *)param; |            |  |
|---|--|------------|--|
| Parameters  | IN/OUT                                 | Types      | Descriptions   |
| param   | IN                                     | LockParam  | LockParamobject.   |
| -   | OUT                                    | ResultCode | The result that will be returned when the function is called.<br>Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |
| <p><b>■Function Description:</b><br/>This function is used to lock a specific memory bank of RFID tags. It is called back in the delegate function accessResult (see <a href="#">4.7</a>).</p> <p><b>■Sample code:</b> (Note: asReader is the object of AsReader class; param is the object of LockParam class.)<br/> <pre>ResultCode resultCode = [asreader lock:param]; if (resultCode == ResultNoError) {     //Called function successfully. }else{     //Called function failed. }</pre></p> |  |            |  |

### 3.2.11. unlock

| Function   | - (ResultCode)unlock:(LockParam *)param; |            |  |
|--|--|------------|--|
| Parameters   | IN/OUT                                   | Types      | Descriptions   |
| param  | IN                                       | LockParam  | LockParamobject.   |
| -  | OUT                                      | ResultCode | The result that will be returned when the function is called.<br>Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |
| <p><b>■Function Description:</b><br/>Function is to unlock the tag memory bank which was locked. After unlocking, the default password can be used to rewrite the RFID tag data. It is called back in the delegate function accessResult (see <a href="#">4.7</a>).</p> <p><b>■Sample code:</b> (Note: asReader is the object of AsReader class; param is the object of LockParam class.)<br/> <pre>ResultCode resultCode = [asreader unlock:param];</pre></p> |  |            |  |

```

if (resultCode == ResultNoError) {
    //Called function successfully.
}else{
    //Called function failed.
}
    
```

### 3.2.12. permaLock

| Function   | - (ResultCode)permaLock:(LockParam *)param; |            |  |
|--|---|------------|--|
| Parameters   | IN/OUT                                      | Types      | Descriptions   |
| param  | IN  | LockParam  | LockParamobject.   |
| -  | OUT   | ResultCode | The result that will be returned when the function is called.<br>Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |
| <p><b>■ Function Description:</b><br/>This function is used to permanently lock the data of the specific memory bank of the tag. Permanently locked tag memory bank data can not be changed and unlocked. It is called back in the delegate function accessResult (see <a href="#">4.7</a>).</p> <p><b>■ Sample code:</b> (Note: asReader is the object of AsReader class; param is the object of LockParam class.)<br/> <pre> ResultCode resultCode = [asreader permaLock:param]; if (resultCode == ResultNoError) {     //Called function successfully. }else{     //Called function failed. }                     </pre> </p> |   |            |  |

### 3.2.13. kill

| Function     | - (ResultCode)kill:(NSString *)killPassword; |            |   |
|--------------|--|------------|---|
| Parameters   | IN/OUT                                       | Types      | Descriptions  |
| killPassword | IN   | NSString   | The password for killing the tags.                            |
| -            | OUT  | ResultCode | The result that will be returned when the function is called. |

|  |  |  |   |
|--|--|--|---|
|  |  |  | Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |
| <p><b>■Function Description:</b><br/>         This function is used to kill RFID tags. Killed tags can no longer be used. It is called back in the delegate function accessResult (see <a href="#">4.7</a>).</p> <p><b>■Sample code:</b> (Note: asReader is the object of AsReader class.)<br/> <pre>ResultCode resultCode = [asreader kill: @"00000000"]; if (resultCode == ResultNoError) {     //Called function successfully. }else{     //Called function failed. }</pre></p> |  |  |   |

### 3.2.14. stop

|  |                     |              |  |
|--|---------------------|--------------|--|
| <b>Function</b>  | - (ResultCode)stop; |              |  |
| <b>Parameters</b>  | <b>IN/OUT</b>       | <b>Types</b> | <b>Descriptions</b>  |
| -  | OUT                 | ResultCode   | The result that will be returned when the function is called.<br>Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |
| <p><b>■Function Description:</b><br/>         Stop taking inventory of tags. It is called back in the delegate function changedActionState (see <a href="#">4.4</a>).</p> <p><b>■Sample code:</b> (Note: asReader is the object of AsReader class.)<br/> <pre>ResultCode resultCode = [asreader stop]; if (resultCode == ResultNoError) {     //Called function successfully. }else{     //Called function failed. }</pre></p> |                     |              |  |

### 3.2.15. stopSync

|                   |                         |              |                     |
|-------------------|-------------------------|--------------|---------------------|
| <b>Function</b>   | - (ResultCode)stopSync; |              |                     |
| <b>Parameters</b> | <b>IN/OUT</b>           | <b>Types</b> | <b>Descriptions</b> |

|  |     |            |  |
|--|-----|------------|--|
| -  | OUT | ResultCode | The result that will be returned when the function is called.<br>Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |
| <p>■ <b>Function Description:</b><br/>Stops scanning in the synchronized thread.<br/>It is called back in the delegate function changedActionState (see <a href="#">4.4</a>).</p> <p>■ <b>Sample code:</b> (Note: asReader is the object of AsReader class.)<br/> <pre>ResultCode resultCode = [asreader stopSync]; if (resultCode == ResultNoError) {     //Called function successfully. }else{     //Called function failed. }</pre> </p> |     |            |  |

### 3.2.16. defaultParameter

|   |                                 |              |  |
|---|---------------------------------|--------------|--|
| <b>Function</b>   | - (ResultCode)defaultParameter; |              |  |
| <b>Parameters</b>   | <b>IN/OUT</b>                   | <b>Types</b> | <b>Descriptions</b>  |
| -   | OUT                             | ResultCode   | The result that will be returned when the function is called.<br>Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |
| <p>■ <b>Function Description:</b><br/>Restores all parameters to their default values.</p> <p>■ <b>Sample code:</b> (Note: asReader is the object of AsReader class.)<br/> <pre>ResultCode resultCode = [asreader defaultParameter]; if (resultCode == ResultNoError) {     //Called function successfully. }else{     //Called function failed. }</pre> </p> |                                 |              |  |

### 3.2.17. saveParameter

|                   |                              |              |                     |
|-------------------|------------------------------|--------------|---------------------|
| <b>Function</b>   | - (ResultCode)saveParameter; |              |                     |
| <b>Parameters</b> | <b>IN/OUT</b>                | <b>Types</b> | <b>Descriptions</b> |

|  |     |            |  |
|--|-----|------------|--|
| -  | OUT | ResultCode | The result that will be returned when the function is called.<br>Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |
| <p>■ <b>Function Description:</b><br/>Save all parameter values to the device memory.</p> <p>■ <b>Sample code:</b> (Note: asReader is the object of AsReader class.)<br/> <pre>ResultCode resultCode = [asreader saveParameter]; if (resultCode == ResultNoError) {     //Called function successfully. }else{     //Called function failed. }</pre></p> |     |            |  |

### 3.2.18. regionName

|  |                          |              |                           |
|--|--------------------------|--------------|---------------------------|
| <b>Function</b>  | -(NSString *)regionName; |              |                           |
| <b>Parameters</b>  | <b>IN/OUT</b>            | <b>Types</b> | <b>Descriptions</b>       |
| -  | OUT                      | NSString     | region name of RF module. |
| <p>■ <b>Function Description:</b><br/>Gets the current region information for AsReaderGUN.</p> <p>■ <b>Sample code:</b> (Note: asReader is the object of AsReader class.)<br/> <pre>NSString * regionName = [asreader regionName];</pre></p> |                          |              |                           |

### 3.2.19. serialNumber

|   |                            |              |                     |
|---|----------------------------|--------------|---------------------|
| <b>Function</b>   | -(NSString *)serialNumber; |              |                     |
| <b>Parameters</b>   | <b>IN/OUT</b>              | <b>Types</b> | <b>Descriptions</b> |
| -   | OUT                        | NSString     | Serial number       |
| <p>■ <b>Function Description:</b><br/>Gets the current serial number for AsReaderGUN.</p> <p>■ <b>Sample code:</b> (Note: asReader is the object of AsReader class.)<br/> <pre>NSString * serialNumber = [asreader serialNumber];</pre></p> |                            |              |                     |

### 3.2.20. rFModuleVersion

|  |                                |              |                                      |
|--|--------------------------------|--------------|--------------------------------------|
| <b>Function</b>  | - (NSString *)rFModuleVersion; |              |                                      |
| <b>Parameters</b>  | <b>IN/OUT</b>                  | <b>Types</b> | <b>Descriptions</b>                  |
| -  | OUT                            | NSString     | The version number of the RF module. |
| <p>■ <b>Function Description:</b><br/>Get the RF module version of AsReaderGUN.</p> <p>■ <b>Sample code:</b> (Note: asReader is the object of AsReader class.)<br/> <pre>NSString * rFModuleVersion = [asreader rFModuleVersion];</pre> </p> |                                |              |                                      |

### 3.2.21. firmwareVersion

|   |                                |              |                     |
|---|--------------------------------|--------------|---------------------|
| <b>Function</b>   | - (NSString *)firmwareVersion; |              |                     |
| <b>Parameters</b>   | <b>IN/OUT</b>                  | <b>Types</b> | <b>Descriptions</b> |
| -   | OUT                            | NSString     | FW version          |
| <p>■ <b>Function Description:</b><br/>Get the FW version of AsReaderGUN.</p> <p>■ <b>Sample code:</b> (Note: asReader is the object of AsReader class.)<br/> <pre>NSString * firmwareVersion = [asreader firmwareVersion];</pre> </p> |                                |              |                     |

### 3.2.22. powerGainRange

|  |                                 |              |  |
|--|---------------------------------|--------------|--|
| <b>Function</b>  | - (CMinMaxValue)powerGainRange; |              |  |
| <b>Parameters</b>  | <b>IN/OUT</b>                   | <b>Types</b> | <b>Descriptions</b>  |
| -  | OUT                             | CMinMaxValue | Range that Power can be set.<br>Struct CMinMaxValue (see <a href="#">5.2.1</a> ) |
| <p>■ <b>Function Description:</b><br/>Gets the values of the maximum and minimum power that the AsReaderGUN can be set.</p> <p>■ <b>Sample code:</b> (Note: asReader is the object of AsReader class.)<br/> <pre>int min = asReader.powerGainRange.min; // The minimum value of power can be set int max = asReader.powerGainRange.max; // The maximum power can be set</pre> </p> |                                 |              |  |

### 3.2.23. batteryStatus

|  |                       |              |   |
|--|-----------------------|--------------|---|
| <b>Function</b>  | - (int)batteryStatus; |              |   |
| <b>Parameters</b>  | <b>IN/OUT</b>         | <b>Types</b> | <b>Descriptions</b>                                 |
| -  | OUT                   | int          | the amount of battery left.<br>Range: 0, 1, 2, 3, 4 |
| <p>■ <b>Function Description:</b><br/>Get the remaining battery power of AsReaderGUN.</p> <p>■ <b>Sample code:</b><br/>int battery = [asReader batteryStatus];</p> |                       |              |   |

### 3.2.24. clearEpcMask

|   |                             |              |   |
|---|-----------------------------|--------------|---|
| <b>Function</b>   | - (ResultCode)clearEpcMask; |              |   |
| <b>Parameters</b>   | <b>IN/OUT</b>               | <b>Types</b> | <b>Descriptions</b>   |
| -   | OUT                         | ResultCode   | The result that will be returned when the function is called<br>Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |
| <p>■ <b>Function Description:</b><br/>Remove EPC mask data from AsReaderGUN memory.</p> <p>■ <b>Sample code:</b> (Note: asReader is the object of AsReader class.)<br/>ResultCode resultCode = [asreader clearEpcMask];<br/>if (resultCode == ResultNoError) {<br/>    //Called function successfully.<br/>}else{<br/>    //Called function failed.<br/>}</p> |                             |              |   |

### 3.2.25. saveEpcMask

|                   |                            |              |  |
|-------------------|----------------------------|--------------|--|
| <b>Function</b>   | - (ResultCode)saveEpcMask; |              |  |
| <b>Parameters</b> | <b>IN/OUT</b>              | <b>Types</b> | <b>Descriptions</b>  |
| -                 | OUT                        | ResultCode   | The result that will be returned when the function is called |

|  |  |  |   |
|--|--|--|---|
|  |  |  | Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |
| <p>■ <b>Function Description:</b><br/>Save the EPC mask data into AsReaderGUN memory.</p> <p>■ <b>Sample code:</b> (Note: asReader is the object of AsReader class.)<br/> <pre>ResultCode resultCode = [asreader saveEpcMask]; if (resultCode == ResultNoError) {     //Called function successfully. }else{     //Called function failed. }</pre></p> |  |  |   |

### 3.2.26. epcMaskCount

|  |                     |              |                      |
|--|---------------------|--------------|----------------------|
| <b>Function</b>  | -(int)epcMaskCount; |              |                      |
| <b>Parameters</b>  | <b>IN/OUT</b>       | <b>Types</b> | <b>Descriptions</b>  |
| -  | OUT                 | int          | Number of EPC masks. |
| <p>■ <b>Function Description:</b><br/>Gets the number of EPC masks stored in memory of AsReaderGUN.</p> <p>■ <b>Sample code:</b> (Note: asReader is the object of AsReader class.)<br/> <pre>int epcMaskCount = [asReader epcMaskCount];</pre></p> |                     |              |                      |

### 3.2.27. addEpcMask

|                   |   |              |   |
|-------------------|---|--------------|---|
| <b>Function</b>   | -(ResultCode)addEpcMask:(int)offset length:(int)length mask:(NSString *)mask; |              |   |
| <b>Parameters</b> | <b>IN/OUT</b>   | <b>Types</b> | <b>Descriptions</b>   |
| offset            | IN  | int          | The offset of the mask to be add. (Unit: bit)                 |
| length            | IN  | int          | The length of the mask to be add. (Unit: bit)                 |
| mask              | IN  | NSString     | The mask data. (a NSString in HEX form)                       |
| -                 | OUT   | ResultCode   | The result that will be returned when the function is called. |

|   |  |  |   |
|---|--|--|---|
|   |  |  | Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |
| <p><b>■ Function Description:</b><br/>Adds EPC masks.</p> <p><b>■ Sample code:</b> (Note: asReader is the object of AsReader class.)<br/>         ResultCode resultCode = [asreader addEpcMask:32 length:16 mask:@"1234"];<br/>         if (resultCode == ResultNoError) {<br/>             //Called function successfully.<br/>         }else{<br/>             //Called function failed.<br/>         }</p> |  |  |   |

### 3.2.28. addEpcMask

|  |  |                      |   |
|--|--|----------------------|---|
| <b>Function</b>  | - (ResultCode)addEpcMask:(AsSelectMaskEPCParam *)mask; |                      |   |
| <b>Parameters</b>  | <b>IN/OUT</b>  | <b>Types</b>         | <b>Descriptions</b>   |
| mask   | IN   | AsSelectMaskEPCParam | AsSelectMaskEPCParamobject.   |
| -  | OUT  | ResultCode           | The result that will be returned when the function is called<br>Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |
| <p><b>■ Function Description:</b><br/>Adds EPC masks in the form of AsSelectMaskEPCParamobject.</p> <p><b>■ Sample code:</b> (Note: asReader is the object of AsReader class. mask is the object of class AsSelectMaskEPCParam.)<br/>         ResultCode resultCode = [asreader addEpcMask:mask];<br/>         if (resultCode == ResultNoError) {<br/>             //Called function successfully.<br/>         }else{<br/>             //Called function failed.<br/>         }</p> |  |                      |   |

### 3.2.29. getEpcMask

|                 |  |
|-----------------|--|
| <b>Function</b> | - (AsSelectMaskEPCParam *)getEpcMask:(int)index; |
|-----------------|--|

| Parameters | IN/OUT | Types                | Descriptions               |
|------------|--------|----------------------|----------------------------|
| index      | IN     | int                  | Indexes of the EPC masks   |
| -          | OUT    | AsSelectMaskEPCParam | AsSelectMaskEPCParamobject |

**■ Function Description:**  
 Gets the EPC mask for the specified index.

**■ Sample code:** (Note: asReader is the object of AsReader class.)  
 AsSelectMaskEPCParam \* mask= [asreader getEpcMask:0];

### 3.2.30. getLBTMask

| Function   | -(NSArray *)getLBTMask; |         |              |
|------------|-------------------------|---------|--------------|
| Parameters | IN/OUT                  | Types   | Descriptions |
| -          | OUT                     | NSArray | Mask array   |

**■ Function Description:**  
 Gets a mask array of frequency table.

**■ Sample code:** (Note: asReader is the object of AsReader class.)  
 NSArray \* lbtArray = [asreader getLBTMask];

### 3.2.31. getLBT

| Function   | -(NSArray *)getLBT; |         |                 |
|------------|---------------------|---------|-----------------|
| Parameters | IN/OUT              | Types   | Descriptions    |
| -          | OUT                 | NSArray | LBT information |

**■ Function Description:**  
 Gets the LBT information.

**■ Sample code:** (Note: asReader is the object of AsReader class.)  
 NSArray \* lbtArray = [asreader getLBT];

### 3.2.32. setLBT

| Function | -(void)setLBT:(NSArray *)table; |  |  |
|----------|---------------------------------|--|--|
|----------|---------------------------------|--|--|

| Parameters  | IN/OUT | Types   | Descriptions  |
|---|--------|---------|---|
| table   | IN     | NSArray | LBT information<br>(Whosearray element is LbtItem object, see <a href="#">5.7</a> ) |
| <p>■ <b>Function Description:</b><br/>Sets LBT.</p> <p>■ <b>Sample code:</b> (Note: asReader is the object of AsReader class. Array is the object of the class NSArray (whose array element is LbtItem object).)<br/>[asreader setLBT:array];</p> |        |         |   |

### 3.2.33. getLBTFrequency

| Function  | - (NSString *)getLBTFrequency:(int)slot; |          |   |
|---|--|----------|---|
| Parameters  | IN/OUT                                   | Types    | Descriptions                                      |
| slot  | IN                                       | int      | The frequency position of the LBT frequency table |
| -   | OUT                                      | NSString | Band information                                  |
| <p>■ <b>Function Description:</b><br/>Gets the frequency band information.</p> <p>■ <b>Sample code:</b> (Note: asReader is the object of AsReader class.)<br/>NSString * frequency =[asreader getLBTFrequency:0];</p> |  |          |   |

### 3.2.34. startBuzzerWithBuzzerTime

| Function   | - (ResultCode)startBuzzerWithBuzzerTime:(int)buzzerTime; |            |   |
|--|--|------------|---|
| Parameters   | IN/OUT   | Types      | Descriptions  |
| buzzerTime   | IN   | int        | Buzzer time (unit: ms)  |
| -  | OUT  | ResultCode | The result that will be returned when the function is called<br>Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |
| <p>■ <b>Function Description:</b><br/>Makes the buzzer to beep for a certain time.</p> <p>■ <b>Sample code:</b>(Note: asReader is the object of the AsReader class.)<br/>ResultCode resultCode = [asreader startBuzzerWithBuzzerTime:100];</p> |  |            |   |

```

if (resultCode == ResultNoError) {
//Called function successfully.
}else{
    //Called function failed.
}
    
```

### 3.2.35. startVibratorWithVibratorTime

| Function  | - (ResultCode)startVibratorWithVibratorTime:(int)vibratorTime; |            |  |
|---|--|------------|--|
| Parameters  | IN/OUT   | Types      | Descriptions   |
| vibratorTime  | IN   | int        | Vibrating time (unit: ms)  |
| -   | OUT  | ResultCode | The result that will be returned when the function is called.<br>Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |
| <p>■ <b>Function Description:</b><br/>Makes the vibrator to vibrate for a certain time.</p> <p>■ <b>Sample code:</b>(Note: asReader is the object of the AsReader class.)<br/>                     ResultCode resultCode = [asreader startVibratorWithVibratorTime:100];<br/>                     if (resultCode == ResultNoError) {<br/>                     //Called function successfully.<br/>                     }else{<br/>                     //Called function failed.<br/>                     }<br/> </p> |  |            |  |

### 3.2.36. startDecode

| Function   | - (ResultCode)startDecode; |            |   |
|------------|----------------------------|------------|---|
| Parameters | IN/OUT                     | Types      | Descriptions  |
| -          | OUT                        | ResultCode | The result that will be returned when the function is called<br>Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |

**■ Function Description:**  
 Starts to scan barcodes.  
 It is called back in the delegate functions `changedActionState` (see [4.4](#)) and the `detectBarcode` (see [4.5](#), [4.6](#)).

**Note:**  
 To stop scanning barcode, please apply **3.2.15. stopSync**

**■ Sample code:**(Note: `asReader` is the object of the `AsReader` class.)  

```

  ResultCode resultCode = [asreader startDecode];
  if (resultCode == ResultNoError) {
    //Called function successfully.
  }else{
    //Called function failed.
  }

```

**3.2.37. setBarcodeParam**

|                   |  |              |   |
|-------------------|--|--------------|---|
| <b>Function</b>   | - (ResultCode)setBarcodeParam:(AsParamValue *)paramData; |              |   |
| <b>Parameters</b> | <b>IN/OUT</b>  | <b>Types</b> | <b>Descriptions</b>   |
| paramData         | IN   | AsParamValue | AsParamValueobject  |
| -                 | OUT  | ResultCode   | The result that will be returned when the function is called<br>Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |

**■ Function Description:**  
 Sets barcode configuration information.

**■ Sample code:** (Note: `asReader` is the object of class `AsReader`. `asParamValue` is the object of class `AsParamValue`. )  

```

  ResultCode resultCode = [asreader setBarcodeParam: asParamValue];
  if (resultCode == ResultNoError) {
    //Called function successfully.
  }else{
    //Called function failed.
  }

```

### 3.2.38. getBarcodeParam

| Function  | - (AsParamValue *)getBarcodeParam:(NSNumber *)paramData; |              |   |
|---|--|--------------|---|
| Parameters  | IN/OUT   | Types        | Descriptions  |
| paramData   | IN   | NSNumber     | Barcode parameters to be obtained<br>See <a href="#">8.2.1</a> ParamName. |
| -   | OUT  | AsParamValue | AsParamValueobject.   |
| <p>■ <b>Function Description:</b><br/>Gets barcode configuration information.</p> <p>■ <b>Sample code:</b>(Note: asReader is the object of the AsReader class.)<br/>AsParamValue * asParamValue= [asreader getBarcodeParam: [NSNumber numberWithInt:UPCA]];</p> |  |              |   |

### 3.2.39. usedSelectionMask

| Function  | - (BOOL)usedSelectionMask:(int)index; |       |   |
|---|---------------------------------------|-------|---|
| Parameters  | IN/OUT                                | Types | Descriptions  |
| index   | IN                                    | int   | The index of mask   |
| -   | OUT                                   | BOOL  | Indicates whether the mask corresponding to the current index is being used.<br>YES: Being used.<br>NO: Not being used. |
| <p>■ <b>Function Description:</b><br/>Returns whether the mask pointed to by a specified index in AsReaderGUN is being used.</p> <p>■ <b>Sample code:</b>(Note: asReader is the object of the AsReader class.)<br/>BOOL isUsed = [asreader usedSelectionMask:0];<br/>if (isUsed) {<br/>//The mask corresponding to the current index is being used.<br/>}<br/>else{<br/>// The mask corresponding to the current index is not being used.<br/>}</p> |                                       |       |   |

### 3.2.40. getSelectionMask

|   |   |                   |                          |
|---|---|-------------------|--------------------------|
| <b>Function</b>   | - (AsSelectMaskParam *)getSelectionMask:(int)index; |                   |                          |
| <b>Parameters</b>   | <b>IN/OUT</b>                                       | <b>Types</b>      | <b>Descriptions</b>      |
| index   | IN  | int               | The index of mask        |
| -   | OUT   | AsSelectMaskParam | AsSelectMaskParam object |
| <p>■ <b>Function Description:</b><br/>Gets the Selection mask for a specified index.</p> <p>■ <b>Sample code:</b>(Note: asReader is the object of the AsReader class.)<br/>AsSelectMaskParam * asSelectMaskParam = [asreader getSelectionMask:0];</p> |   |                   |                          |

### 3.2.41. setSelectionMask

|  |  |                   |                          |
|--|--|-------------------|--------------------------|
| <b>Function</b>  | - (void)setSelectionMask:(int)index<br>withParam:(AsSelectMaskParam *)param; |                   |                          |
| <b>Parameters</b>  | <b>IN/OUT</b>  | <b>Types</b>      | <b>Descriptions</b>      |
| index  | IN   | int               | The index of mask        |
| param  | IN   | AsSelectMaskParam | AsSelectMaskParam object |
| <p>■ <b>Function Description:</b><br/>Sets the Selection mask where the specified index points.</p> <p>■ <b>Sample code:</b> (Note: asReader is the object of the AsReader class.<br/>asSelectMaskParam is the object of the class AsSelectMaskParam.)<br/>[asreader setSelectionMask:0 withParam: asSelectMaskParam];</p> |  |                   |                          |

### 3.2.42. removeSelectionMask

|  |   |              |                     |
|--|---|--------------|---------------------|
| <b>Function</b>  | - (void)removeSelectionMask:(int)index; |              |                     |
| <b>Parameters</b>  | <b>IN/OUT</b>                           | <b>Types</b> | <b>Descriptions</b> |
| index  | IN                                      | int          | The index of mask   |
| <p>■ <b>Function Description:</b><br/>Removes the Selection mask at the location to which the specified index points to.</p> <p>■ <b>Sample code:</b>(Note: asReader is the object of the AsReader class.)<br/>[asreader removeSelectionMask:0];</p> |   |              |                     |

### 3.2.43. clearSelectionMask

|  |                             |              |                     |
|--|-----------------------------|--------------|---------------------|
| <b>Function</b>  | - (void)clearSelectionMask; |              |                     |
| <b>Parameters</b>  | <b>IN/OUT</b>               | <b>Types</b> | <b>Descriptions</b> |
| -  | -                           | -            | -                   |
| <p>■ <b>Function Description:</b><br/>Remove all Selection masks.</p> <p>■ <b>Sample code:</b>(Note: asReader is the object of the AsReader class.)<br/>[asreader clearSelectionMask];</p> |                             |              |                     |

### 3.2.44. getAlgorithm

|  |                                |               |  |
|--|--------------------------------|---------------|--|
| <b>Function</b>  | - (AlgorithmType)getAlgorithm; |               |  |
| <b>Parameters</b>  | <b>IN/OUT</b>                  | <b>Types</b>  | <b>Descriptions</b>  |
| -  | OUT                            | AlgorithmType | Type of Q<br><br>Enumerated type AlgorithmType<br>(see <a href="#">3.3.1</a> ) |
| <p>■ <b>Function Description:</b><br/>Gets the anti-collision mode.</p> <p>■ <b>Sample code:</b>(Note: asReader is the object of the AsReader class.)<br/>AlgorithmType type= [asreader getAlgorithm];<br/>if (type = FixedQ) {<br/>// Currently in Fixed Q mode<br/>}else if (type = DynamicQ) {<br/>// Currently in Dynamic Q mode<br/>}</p> |                                |               |  |

### 3.2.45. setBarcodeMode

|                   |   |              |                             |
|-------------------|---|--------------|-----------------------------|
| <b>Function</b>   | -(ResultCode)setBarcodeMode:(BOOL)enabled<br>isKeyAction:(BOOL)isKeyOn; |              |                             |
| <b>Parameters</b> | <b>IN/OUT</b>   | <b>Types</b> | <b>Descriptions</b>         |
| enabled           | IN  | BOOL         | Enable/disable barcode mode |
| isKeyOn           | IN  | BOOL         | Not use                     |

|  |     |            |                                      |
|--|-----|------------|--------------------------------------|
| -  | OUT | ResultCode | The execution result of the function |
| <p><b>■Function Description:</b><br/>Set the current device mode to Barcode mode.</p> <p><b>Note:</b><br/>In case you wish to customize prefixes/suffixes, please apply 3.2.46. setBarcodeMode (With Custom prefix suffix ) instead of this function.<br/>In the case that Barcode mode is set to be valid, any RFID-related commands cannot be executed.<br/>If Barcode mode is set to be invalid, the device can only execute RFID-related commands.</p> <p><b>■Sample code:</b> (Note: asraeder is an instance of AsReader class)<br/>ResultCode resultCode = [asraeder setBarcodeMode:YES isKeyAction:YES];<br/>if (resultCode == ResultNoError) {<br/>    //The function was called successfully.<br/>}else{<br/>    //The function is failed called.<br/>}</p> |     |            |                                      |

**3.2.46. setBarcodeMode (With Custom prefix suffix )**

|  |  |              |                                      |
|--|--|--------------|--------------------------------------|
| <b>Function</b>  | -(ResultCode)setBarcodeMode:(BOOL)enabled isKeyAction:(BOOL)isKeyOn isCustomPreSuffixOn:(BOOL)isCustomPreSuffixOn; |              |                                      |
| <b>Parameters</b>  | <b>IN/OUT</b>  | <b>Types</b> | <b>Descriptions</b>                  |
| enabled  | IN   | BOOL         | Enable/disable barcode mode          |
| isKeyOn  | IN   | BOOL         | Not use                              |
| isCustomPre SuffixOn   | in   | BOOL         | Enable/disable prefix/suffix         |
| -  | OUT  | ResultCode   | The execution result of the function |
| <p><b>■Function Description:</b><br/>Set the current device mode to Barcode mode and set prefix , suffix.</p> <p><b>Note:</b><br/>In case you wish to customize prefixes/suffixes, please apply this function instead of 3.2.45.<br/>In the case that Barcode mode is set to be valid, any RFID-related commands cannot be executed.</p> |  |              |                                      |

If Barcode mode is set to be invalid, the device can only execute RFID-related commands.

■ **Sample code:** (Note: asraeder is an instance of AsReader class)  
 ResultCode resultCode = [asraeder setBarcodeMode:YES isKeyAction:YES isCustomPreSuffixOn:YES];  
 if (resultCode == ResultNoError) {  
     //The function was called successfully.  
 }else{  
     //The function is failed called.  
 }

### 3.2.47. isBarcodeModule

| Function  | - (BOOL)isBarcodeModule; |       |   |
|---|--------------------------|-------|---|
| Parameters  | IN/OUT                   | Types | Descriptions  |
| -   | OUT                      | BOOL  | Gets whether Barcode module is supported or not.<br>YES: support<br>NO: not supported |
| <p>■ <b>Function Description:</b><br/>           Gets whether the current AsReaderGUN supports the Barcode module.</p> <p>■ <b>Sample code:</b>(Note: asReader is the object of the AsReader class.)<br/>           BOOL isSupportBarcodeModule = [asreader isBarcodeModule];<br/>           if (isSupportBarcodeModule) {<br/>               // Supports the Barcode module.<br/>           }else{<br/>               // Does not support the Barcode module.<br/>           }</p> |                          |       |   |

### 3.2.48. isRFIDModule

| Function   | - (BOOL)isRFIDModule; |       |              |
|------------|-----------------------|-------|--------------|
| Parameters | IN/OUT                | Types | Descriptions |

|   |     |      |   |
|---|-----|------|---|
| -   | OUT | BOOL | Whether RFID module is supported or not.<br>YES: support<br>NO: not supported |
| <p>■ <b>Function Description:</b><br/>Gets whether the current AsReaderGUN supports the RFID module.</p> <p>■ <b>Sample code:</b>(Note: asReader is the object of the AsReader class.)<br/> <pre> BOOL isSupportRFIDModule = [asreader isRFIDModule]; if (isSupportRFIDModule) { // Supports the RFID module. }else{ // Does not support the RFID module. } </pre> </p> |     |      |   |

### 3.2.49. getTagState

| Function   | - (TagState)getTagState:(NSString *)pcEPCData; |          |  |
|--|--|----------|--|
| Parameters   | IN/OUT   | Types    | Descriptions   |
| pcEPCData  | IN   | NSString | The PCEPC value  |
| -  | OUT  | TagState | It returns the result of the function being called.<br>Enumeration type TagState (see : <a href="#">5.1.11</a> ) |
| <p>■ <b>Function Description:</b><br/>Whether the EPC Bank of the RFID tag is locked.</p> <p>■ <b>Sample code:</b> (Note: asReader is the object of the AsReader class. pcEpc is the object of the String class.)<br/> <pre> TagState state = [mReader getTagState:pcEpc]; </pre> </p> |  |          |  |

## 3.3. Enum

### 3.3.1. AlgorithmType

| <b>Definitions</b> | <b>Descriptions</b> |
|--------------------|---------------------|
| FixedQ = 0         | Fixed Q             |
| DynamicQ = 1       | Dynamic Q           |

## 4. AsReaderDelegate Class

### 4.1. readerInitialized

|  |   |              |                     |
|--|---|--------------|---------------------|
| <b>Function</b>  | - (void)readerInitialized:(AsReader *)reader; |              |                     |
| <b>Parameters</b>  | <b>IN/OUT</b>                                 | <b>Types</b> | <b>Descriptions</b> |
| reader   | IN  | AsReader     | Object of AsReader  |
| <p>■ <b>Function Description:</b><br/> Once connected to the AsReaderGun successfully, this function is called to obtain the initialized AsReader object.</p> <p><b>Note:</b> The <code>setDelegate (3.2.4)</code> function must be called in this delegate callback, otherwise the data cannot be returned.</p> <p>■ <b>Sample code:</b></p> <pre>- (void)readerInitialized:(AsReader *)reader { } </pre> |   |              |                     |

### 4.2. updateDeviceState

|   |  |              |   |
|---|--|--------------|---|
| <b>Function</b>   | - (void)updateDeviceState:(ResultCode)error; |              |   |
| <b>Parameters</b>   | <b>IN/OUT</b>                                | <b>Types</b> | <b>Descriptions</b>   |
| error   | IN   | ResultCode   | The updated result of AsReader. Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |
| <p>■ <b>Function Description:</b><br/> This function is called back if an error occurs while executing some functions/properties (※). It returns the result of AsReader's execution of the command. (For example: Error occurs when obtaining whether EAN13 barcode type is supported.)</p> <p>(※): For the functions/properties, please refer to the following sections:</p> |  |              |   |

3.1.2 ~ 3.1.25, 3.2.1, 3.2.16, 3.2.18 ~ 3.2.21, 3.2.23 ~ 3.2.33, 3.2.37 ~ 3.2.46

■ **Sample code:**

```
- (void)updateDeviceState:(ResultCode)error {
    NSString * errorMessage = [AsResultCode msg:error];
}
```

### 4.3. readTag

|  |   |              |  |
|--|---|--------------|--|
| <b>Function</b>  | - (void)readTag:(NSString *)tag rssi:(float)rssi phase:(float)phase frequency:(float)frequency; |              |  |
| <b>Parameters</b>  | <b>IN/OUT</b>   | <b>Types</b> | <b>Descriptions</b>                                    |
| tag  | IN  | NSString     | PCEPC number of the tag be inventoried. (Hex NSString) |
| rssi   | IN  | float        | RSSI value of the tag                                  |
| phase  | IN  | float        | Phase value of the tag                                 |
| frequency  | IN  | float        | Frequency value of the tag                             |
| <p>■ <b>Function Description:</b><br/>                 Once inventory(see3.2.7) is called or the Trigger key is pressed, this delegate function will be called back.</p> |   |              |  |

### 4.4. changedActionState

|                   |  |              |  |
|-------------------|--|--------------|--|
| <b>Function</b>   | - (void)changedActionState:(CommandType)action resultCode:(NSInteger)resultCode; |              |  |
| <b>Parameters</b> | <b>IN/OUT</b>  | <b>Types</b> | <b>Descriptions</b>  |
| action            | IN   | CommandType  | Action type<br>Enumerated type CommandType<br>(see <a href="#">6.1.1</a> ) |
| resultCode        | IN   | NSInteger    | The result that will be returned when the function is called.              |

|  |  |  |   |
|--|--|--|---|
|  |  |  | Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |
| <p>■ <b>Function Description:</b><br/>Once a function like inventory (see <a href="#">3.2.7</a>) or stop (see <a href="#">3.2.14</a>) is called, this function is called back to receive the execution result.</p> <p>■ <b>Sample code:</b></p> <pre>- (void)changedActionState:(CommandType)action resultCode:(NSInteger)resultCode { // action: Type of command // resultCode: The result of command execution }</pre> |  |  |   |

## 4.5. detectBarcode

|  |  |              |   |
|--|--|--------------|---|
| <b>Function</b>  | - (void)detectBarcode:(BarcodeType)barcodeType<br>codelId:(NSString *)codelId barcode:(NSString *)barcode; |              |   |
| <b>Parameters</b>  | <b>IN/OUT</b>  | <b>Types</b> | <b>Descriptions</b>   |
| barcodeType  | IN   | BarcodeType  | Barcode type.<br>Enumerated type BarcodeType (see <a href="#">7.2.1</a> ) |
| codelId  | IN   | NSString     | The code ID of the barcode.   |
| barcode  | IN   | NSString     | barcode data  |
| <p>■ <b>Function Description:</b><br/>Once the function startDecode (see <a href="#">3.2.26</a>) is called, this function is called back.</p> <p>■ <b>Sample code:</b></p> <pre>- (void)detectBarcode:(BarcodeType)barcodeType codelId:(NSString *)codelId barcode:(NSString *)barcode{ // barcodeType: Barcode type // codelId: The code ID of the barcode // barcode: Barcode data }</pre> |  |              |   |

## 4.6. detectBarcode

|   |  |              |   |
|---|--|--------------|---|
| <b>Function</b>   | - (void)detectBarcode:(BarcodeType)barcodeType<br>codeId:(NSString *)codeId barcodeData:(NSData *)barcodeData; |              |   |
| <b>Parameters</b>   | <b>IN/OUT</b>  | <b>Types</b> | <b>Descriptions</b>   |
| barcodeType   | IN   | BarcodeType  | Barcode type.<br>Enumerated type BarcodeType (see <a href="#">7.2.1</a> ) |
| codeId  | IN   | NSString     | The code ID of the barcode.   |
| barcodeData   | IN   | NSData       | barcode data  |
| <p>■ <b>Function Description:</b><br/>Once the function startDecode (see <a href="#">3.2.26</a>) is called, this function is called back.</p> |  |              |   |

## 4.7. accessResult

|                   |   |              |   |
|-------------------|---|--------------|---|
| <b>Function</b>   | - (void)accessResult:(ResultCode)error<br>actionState:(CommandType)action epc:(NSString *)epc<br>data:(NSString *)data rssi:(float)rssi phase:(float)phase<br>frequency:(float)frequency; |              |   |
| <b>Parameters</b> | <b>IN/OUT</b>   | <b>Types</b> | <b>Descriptions</b>   |
| error             | IN  | ResultCode   | The executionresult of the function<br>Enumerated type ResultCode (see <a href="#">5.1.1</a> )  |
| action            | IN  | CommandType  | AsReaderGUN's operation command.<br>Enumerated type CommandType<br>(see <a href="#">6.1.1</a> ) |
| epc               | IN  | NSString     | Tag's EPC value (Hex)   |
| data              | IN  | NSString     | The inventoried tag data (Hex)  |
| rssi              | IN  | float        | RSSI value of the tag.  |
| phase             | IN  | float        | Phase value of the tag.   |
| frequency         | IN  | float        | Frequency value of the tag.   |

**Function Description:**

This function is called when writeMemory, readMemory, lock, unlock, permaLock, and kill are executed.

\* writeMemory, readMemory, lock, unlock, permaLock, and kill, please refer to sections [\(3.2.9\)](#) [\(3.2.8\)](#) [\(3.2.10\)](#) [\(3.2.11\)](#) [\(3.2.12\)](#) [\(3.2.13\)](#).

## 4.8. onAsReaderLeftModeKeyEvent

|   |  |              |  |
|---|--|--------------|--|
| <b>Function</b>   | - (BOOL)onAsReaderLeftModeKeyEvent:(BOOL)status; |              |  |
| <b>Parameters</b>   | <b>IN/OUT</b>                                    | <b>Types</b> | <b>Descriptions</b>  |
| status  | IN   | BOOL         | The state of the Mode key to the left of AsReaderGUN<br>YES: The trigger button is pressed<br>NO: The trigger button is released |
| -   | OUT  | BOOL         | YES: switch mode between Barcode and RFID.<br>NO: keep the current mode.   |
| <p><b>Function Description:</b><br/>Once the left Mode key of AsReaderGUN is pressed or released, the function is called back.</p> <p><b>Sample code:</b></p> <pre>- (BOOL)onAsReaderLeftModeKeyEvent:(BOOL)status{ return YES; }</pre> |  |              |  |

## 4.9. onAsReaderRightModeKeyEvent

|                   |   |              |   |
|-------------------|---|--------------|---|
| <b>Function</b>   | - (BOOL)onAsReaderRightModeKeyEvent:(BOOL)status; |              |   |
| <b>Parameters</b> | <b>IN/OUT</b>                                     | <b>Types</b> | <b>Descriptions</b>                                   |
| status            | IN  | BOOL         | The state of the Mode key to the right of AsReaderGUN |

|  |     |      |  |
|--|-----|------|--|
|  |     |      | YES: The trigger button is pressed<br>NO: The trigger button is released |
| -  | OUT | BOOL | YES: switch mode between Barcode and RFID.<br>NO: keep the current mode. |
| <p><b>■Function Description:</b><br/>Once the right Mode key of AsReaderGUN is pressed or released, the function is called back.</p> <p><b>■Sample code:</b></p> <pre>- (BOOL) onAsReaderRightModeKeyEvent:(BOOL)status{ return YES; }</pre> |     |      |  |

#### 4.10. onAsReaderTriggerKeyEvent

|  |   |              |  |
|--|---|--------------|--|
| <b>Function</b>  | - (BOOL)onAsReaderTriggerKeyEvent:(BOOL)status; |              |  |
| <b>Parameters</b>  | <b>IN/OUT</b>                                   | <b>Types</b> | <b>Descriptions</b>  |
| status   | IN  | BOOL         | The Trigger button state of AsReaderGUN.<br>YES: The trigger button is pressed<br>NO: The trigger button is released                                 |
| -  | OUT   | BOOL         | YES: perform the SDK default operation, press The trigger button to start the scan and release to stop the scan<br>NO: does not perform SDK defaults |
| <p><b>■Function Description:</b><br/>Once the Trigger button of AsReaderGUN is pressed or released, the function is called back.</p> <p><b>■Sample code:</b></p> <pre>- (BOOL) onAsReaderTriggerKeyEvent:(BOOL)status{ return YES; }</pre> |   |              |  |

## 5. AsRfidValues Class

### 5.1. Enum

#### 5.1.1. ResultCode

| Definitions                       | Descriptions                                 |
|-----------------------------------|--|
| 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.   |
| ResultOverHeated = 0xFFF6       | Over Heated.        |
| ResultModuleOverHeated = 0x0307 | Module Over Heated. |

### 5.1.2. MemoryBank

| Definitions   | Descriptions                   |
|---------------|--------------------------------|
| Bank_Reserved | Refers to Reseved memory Bank. |
| Bank_EPC      | Refers to EPC memory Bank.     |
| Bank_TID      | Refers to TID memory Bank.     |
| Bank_User     | Refers to User memory Bank.    |

### 5.1.3. BuzzerState

| Definitions | Descriptions        |
|-------------|---------------------|
| Buzzer_Off  | Turn off the buzzer |
| Buzzer_Low  | Buzzer low          |
| Buzzer_High | Buzzer high         |

### 5.1.4. VibratorState

| Definitions  | Descriptions          |
|--------------|-----------------------|
| Vibrator_Off | Turn off the vibrator |
| Vibrator_On  | Turn on vibrator      |

### 5.1.5. SessionType

| Definitions | Descriptions   |
|-------------|----------------|
| Session_S0  | inventoried S0 |
| Session_S1  | inventoried S1 |
| Session_S2  | inventoried S2 |

|            |                |
|------------|----------------|
| Session_S3 | inventoried S3 |
|------------|----------------|

### 5.1.6. SessionFlag

| Definitions    | Descriptions |
|----------------|--------------|
| SessionFlag_A  | A only       |
| SessionFlag_B  | B only       |
| SessionFlag_AB | A or B       |

### 5.1.7. SelectFlag

| Definitions        | Descriptions    |
|--------------------|-----------------|
| SelectFlag_NotUsed | Not use flag    |
| SelectFlag_SL      | SL is valid.    |
| SelectFlag_NOT_SL  | SL is invalid.  |
| SelectFlag_All     | Both are valid. |

### 5.1.8. MaskTargetType

| Definitions   | Descriptions    |
|---------------|-----------------|
| MaskTarget_S0 | inventoried S0  |
| MaskTarget_S1 | inventoried S1  |
| MaskTarget_S2 | inventoried S2  |
| MaskTarget_S3 | inventoried S3  |
| MaskTarget_SL | Selection Flags |

### 5.1.9. MaskActionType

| Definitions   | Descriptions  |
|---------------|---|
| MaskAction_AB | Tag Matching: assert SL or inventoried → A<br>Tag Not-Matching: retract SL or inventoried → B |
| MaskAction_AN | Tag Matching: assert SL or inventoried → A<br>Tag Not-Matching: do nothing                    |
| MaskAction_NB | Tag Matching: do nothing<br>Tag Not-Matching: retract SL or inventoried → B                   |
| MaskAction_MN | Tag Matching: negate SL or (A → B, B → A)   |

|               |   |
|---------------|---|
|               | Tag Not-Matching: do nothing  |
| MaskAction_BA | Tag Matching: retract SL or inventoried → B<br>Tag Not-Matching: assert SL or inventoried → A |
| MaskAction_BN | Tag Matching: retract SL or inventoried → B<br>Tag Not-Matching: do nothing                   |
| MaskAction_NA | Tag Matching: do nothing<br>Tag Not-Matching: assert SL or inventoried → A                    |
| MaskAction_NM | Tag Matching: do nothing<br>Tag Not-Matching: negate SL or (A → B, B → A)                     |

### 5.1.10. MaskType

| Definitions        | Descriptions    |
|--------------------|-----------------|
| MaskType_NO_MASK   | No MASK.        |
| MaskType_Selection | Selection MASK. |
| MaskType_EPC       | EPC MASK.       |

### 5.1.11. TagState

| Definitions     | Descriptions   |
|-----------------|--|
| State_Unknown   | Failed to get the EPC information of the target tag. |
| State_Error     | Other errors   |
| State_NoReply   | An error occurred while getting the Lock status.     |
| State_UnLock    | The EPC Bank of the RFID tag is unlocked.            |
| State_Lock      | The EPC Bank of the RFID tag is locked.              |
| State_PermaLock | The EPC Bank of the RFID tag is permanently locked.  |

## 5.2. Struct

### 5.2.1. CMinMaxValue

The settable range of the RFID output power value:

| <b>Definitions</b> | <b>Types</b> | <b>Descriptions</b>    |
|--------------------|--------------|------------------------|
| min                | int          | Minimum settable power |
| max                | int          | Maximum settable power |

## 5.3. LockParam

### 5.3.1. Properties

#### 5.3.1.1. @property (nonatomic) BOOL killPassword;

**Description:**

Gets or sets whether to control the Kill password area.

YES: controls the Kill password area

NO: do not control the Kill password area

#### 5.3.1.2. @property (nonatomic) BOOL accessPassword;

**Description:**

Gets or sets whether to control the Access password area.

YES: controls the Access password area

NO: do not control the Access password area

#### 5.3.1.3. @property (nonatomic) BOOL epc;

**Description:**

Gets or sets whether to control the EPC bank.

YES: controls the EPC bank

NO: do not control the EPC bank

#### 5.3.1.4. @property (nonatomic) BOOL tid;

**Description:**

Gets or sets whether to control the TID bank.

YES: controls the TID bank

NO: do not control the TID bank

#### 5.3.1.5. @property (nonatomic) BOOL user;

**Description:**

Gets or sets whether to control the User bank.  
 YES: controls the User bank  
 NO: do not control the User bank

## 5.4. AsResultCode

### 5.4.1. Function

#### 5.4.1.1. msg

|  |                                    |              |  |
|--|------------------------------------|--------------|--|
| <b>Function</b>  | +(NSString *)msg:(ResultCode)code; |              |  |
| <b>Parameters</b>  | <b>IN/OUT</b>                      | <b>Types</b> | <b>Descriptions</b>  |
| code   | IN                                 | ResultCode   | The result of the function being called<br>Enumerated type ResultCode (see <a href="#">5.1.1</a> ) |
| -  | OUT                                | NSString     | Descriptions of the ResultCode   |
| <p>■ <b>Function Description:</b><br/>                 Convert the ResultCode to its corresponding text description.</p> <p>■ <b>Sample code:</b><br/>                 NSString * errorMessage = [AsResultCode msg:ResultNoError];</p> |                                    |              |  |

## 5.5. AsSelectMaskParam

### 5.5.1. Properties

#### 5.5.1.1. @property (nonatomic) MaskTargetType target;

**Description:**  
 Gets or sets the MaskTargetType for the current Mask.  
 Enumerated type MaskTargetType (see [5.1.8](#))

**5.5.1.2. @property (nonatomic) MaskActionType action;****Description:**

Gets or sets the MaskActionType for the current Mask.  
Enumerated type MaskActionType (see [5.1.9](#))

**5.5.1.3. @property (nonatomic) MemoryBank bank;****Description:**

Gets or sets the MemoryBank for the current Mask.  
Enumerated type MemoryBank (see [5.1.2](#))

**5.5.1.4. @property (nonatomic) int offset;****Description:**

Gets or sets the offset for the current Mask.

**5.5.1.5. @property (strong, nonatomic) NSString \*mask;****Description:**

Gets or sets the current Mask. (Hex)

**5.5.1.6. @property (nonatomic) int length;****Description:**

Gets or sets the length of the current Mask.

**5.5.1.7. @property (nonatomic) BOOL used;****Description:**

Gets or sets whether the current Mask is used.  
YES: Use Mask  
NO: Not use Mask

## 5.5.2. Function

### 5.5.2.1. index

|  |               |              |                     |
|--|---------------|--------------|---------------------|
| <b>Function</b>  | -(int)index;  |              |                     |
| <b>Parameters</b>  | <b>IN/OUT</b> | <b>Types</b> | <b>Descriptions</b> |
| -  | OUT           | int          | Mask order          |
| <p>■ <b>Function Description:</b><br/>Gets the current Mask order of the AsSelectMaskParam object.</p> <p>■ <b>Sample code:</b><br/>int index = [param index];</p> |               |              |                     |

### 5.5.2.2. initWithIndex

|   |                                |                   |                          |
|---|--------------------------------|-------------------|--------------------------|
| <b>Function</b>   | -(id)initWithIndex:(int)index; |                   |                          |
| <b>Parameters</b>   | <b>IN/OUT</b>                  | <b>Types</b>      | <b>Descriptions</b>      |
| index   | IN                             | int               | Mask order               |
| -   | OUT                            | AsSelectMaskParam | AsSelectMaskParam object |
| <p>■ <b>Function Description:</b><br/>Creates an AsSelectMaskParam object due to the index value and Selection Mask value.</p> <p>■ <b>Sample code:</b><br/>AsSelectMaskParam *param = [[AsSelectMaskParam alloc] initWithIndex:0];</p> |                                |                   |                          |

### 5.5.2.3. initWithParameterIndex

|                   |  |              |                     |
|-------------------|--|--------------|---------------------|
| <b>Function</b>   | -(id)initWithParameterIndex:(int)index<br>target:(MaskTargetType)maskTarget<br>action:(MaskActionType)maskAction bank:(MemoryBank)maskBank<br>offset:(int)maskOffset mask:(NSString *)maskData<br>used:(BOOL)usedMask; |              |                     |
| <b>Parameters</b> | <b>IN/OUT</b>  | <b>Types</b> | <b>Descriptions</b> |
| index             | IN   | int          | Mask order (0~7)    |

|   |    |                |   |
|---|----|----------------|---|
| maskTarget  | IN | MaskTargetType | Target value for Selection Masks<br>Enumerated type<br>MaskTargetType (see <a href="#">5.1.8</a> )    |
| maskAction  | IN | MaskActionType | Action value for Selection Masks<br>Enumerated type MaskActionType<br>(see <a href="#">5.1.9</a> )    |
| maskBank  | IN | MemoryBank     | MemoryBank value for Selection<br>Masks<br>Enumerated type MemoryBank<br>(see <a href="#">5.1.2</a> ) |
| maskOffset  | IN | int            | Offset  |
| maskData  | IN | NSString       | Mask data (Hex)   |
| usedMask  | IN | BOOL           | Whether to use the current Mask   |
| <p>■ <b>Function Description:</b><br/>It creates AsSelectMaskParam object which makes up Selection Mask information with given parameter values.</p> <p>■ <b>Sample code:</b><br/>AsSelectMaskParam *param = [[AsSelectMaskParam alloc]<br/>initWithParameterIndex:0 target:MaskTarget_SL action:MaskAction_AB<br/>bank:Bank_EPC offset:32 mask:@"1234"used:YES];</p> |    |                |   |

#### 5.5.2.4. initWithParameterLength

|                   |  |                |  |
|-------------------|--|----------------|--|
| <b>Function</b>   | -(id)initWithParameterLength:(int)index<br>target:(MaskTargetType)maskTarget<br>action:(MaskActionType)maskAction bank:(MemoryBank)maskBank<br>offset:(int)maskOffset mask:(NSString *)maskData<br>length:(int)maskLength used:(BOOL)usedMask; |                |  |
| <b>Parameters</b> | <b>IN/OUT</b>  | <b>Types</b>   | <b>Descriptions</b>  |
| index             | IN   | int            | Mask order (0~7)   |
| maskTarget        | IN   | MaskTargetType | Target value for the Selection<br>Mask<br>Enumerated type<br>MaskTargetType (see <a href="#">5.1.8</a> ) |
| maskAction        | IN   | MaskActionType | Action value for Selection Masks   |

|            |    |            |   |
|------------|----|------------|---|
|            |    |            | Enumerated type MaskActionType (see <a href="#">5.1.9</a> )                                     |
| maskBank   | IN | MemoryBank | MemoryBank value for Selection Masks<br>Enumerated type MemoryBank (see <a href="#">5.1.2</a> ) |
| maskOffset | IN | int        | Offset  |
| maskData   | IN | NSString   | Mask data (Hex)   |
| maskLength | IN | int        | Mask length (Unit: bit)   |
| usedMask   | IN | BOOL       | Whether to use the current Mask   |

**■ Function Description:**

it creates AsSelectMaskParam objects that make up Selection Mask information with given parameter values.

**■ Sample code:**

```
AsSelectMaskParam *param = [[AsSelectMaskParam alloc]
initWithParameterIndex:0 target:MaskTarget_SL action:MaskAction_AB
bank:Bank_EPC offset:32 mask:@"1234" length:16 used:YES];
```

## 5.6. AsSelectMaskEPCParam

### 5.6.1. Properties

#### 5.6.1.1. @property (nonatomic) int offset;

**Description:**

Gets or sets the offset of the Selection mask data.

#### 5.6.1.2. @property (nonatomic) int length;

**Description:**

Gets or sets the length of the target tag data.

**5.6.1.3. @property (strong, nonatomic) NSString \*mask;**

**Description:**  
Gets or sets the Selection Mask data (Hex).

## 5.7. LblItem

### 5.7.1. Properties

**5.7.1.1. @property (nonatomic) int mSlot;**

**Description:**  
Gets or sets the frequency position of the LBT frequency table.

**5.7.1.2. @property (nonatomic) BOOL mIsUsed;**

**Description:**  
Gets or sets whether to use the specific frequency of the LBT frequency table.  
YES: used  
NO: not used

**5.7.1.3. @property (strong, nonatomic) NSString \*frequency;**

**Description:**  
Gets or sets the frequency value of the LBT frequency table.

### 5.7.2. Function

**5.7.2.1. init**

|                 |            |
|-----------------|------------|
| <b>Function</b> | -(id)init; |
|-----------------|------------|

| Parameters  | IN/OUT | Types   | Descriptions   |
|---|--------|---------|----------------|
| -   | OUT    | LbtItem | LbtItem object |
| <p>■ <b>Function Description:</b><br/>Initialize the LbtItem object.</p> <p>■ <b>Sample code:</b><br/>LbtItem *lbtI = [[LbtItem alloc] init];</p> |        |         |                |

### 5.7.2.2. initWithSlot

| Function  | -(id)initWithSlot:(int)slot isUsed:(BOOL)isUsed; |       |   |
|---|--|-------|---|
| Parameters  | IN/OUT   | Types | Descriptions  |
| slot  | IN   | int   | The frequency position of the LBT frequency table   |
| isUsed  | IN   | BOOL  | Whether to use the specified frequency of the LBT frequency table.<br>YES: used<br>NO: not used |
| -   | OUT  | Id    | LbtItem object  |
| <p>■ <b>Function Description:</b><br/>Creates and Initializes LbtItem.</p> <p>■ <b>Sample code:</b><br/>LbtItem *lbtI = [LbtItem alloc] initWithSlot:1 isUsed:YES];</p> |  |       |   |

## 6. AsPacket Class

### 6.1. Enum

#### 6.1.1. CommandType

| Definitions                 | Descriptions                |
|-----------------------------|-----------------------------|
| 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     | Perma Lock in progress.     |
| CommandBlockWrite = 0x57    | Block Write in progress.    |
| CommandBlockErase = 0x45    | Block Erase in progress.    |
| CommandStop = 0x73          | Operation Stopped.          |
| CommandDefaultParam = 0x61  | Default Param in progress.  |
| CommandSaveParam = 0x53     | Save Param in progress.     |
| CommandDecodeStart = 0x64   | Scan Barcode in progress.   |
| CommandBuzzerStart = 0x75   | Start Buzzer in progress.   |
| CommandVibratorStart = 0x76 | Start Vibrator in progress. |

#### 6.1.2. ScanMode

| Definitions     | Descriptions |
|-----------------|--------------|
| RFIDScanMode    | RFID mode    |
| BarcodeScanMode | Barcode mode |

## 7. AsBarcodeType Class

### 7.1. Function

#### 7.1.1. getBarcodeString

| Function  | +(NSString *)getBarcodeString:(BarcodeType)barcodeType; |             |  |
|---|---|-------------|--|
| Parameters  | IN/OUT  | Types       | Descriptions   |
| barcodeType   | IN  | BarcodeType | Enumerated type BarcodeType (see <a href="#">7.2.1</a> ) |
| -   | OUT   | NSString    | Text of the barcode type                                 |
| <p>■ <b>Function Description:</b><br/>The BarcodeType can be obtained as a string from the specified BarcodeType enumeration (see 7.2.1).</p> <p>■ <b>Sample code:</b><br/>NSString * barcodeName = [AsBarcodeType getBarcodeString:<br/>BarcodeTypeEAN13];</p> |   |             |  |

## 7.2. Enum

### 7.2.1. BarcodeType

The following are the enumeration definitions of the read barcode types.

| Enum Names                | Descriptions   |
|---------------------------|--|
| BarcodeTypeNoRead         | NR (It will be returned when no barcode is scanned within a certain period of time.) |
| BarcodeTypeAustralianPost | Australian Post  |
| BarcodeTypeAztecCode      | Aztec Code   |
| BarcodeTypeBooklandEAN    | Bookland EAN   |
| BarcodeTypeBritishPost    | British Post   |

|                            |                                  |
|----------------------------|----------------------------------|
| BarcodeTypeCanadianPost    | Canadian Post                    |
| BarcodeTypeChinaPost       | China Post                       |
| BarcodeTypeCodabar         | Codabar                          |
| BarcodeTypeCodablockF      | Codablock F                      |
| BarcodeTypeCode11          | Code 11                          |
| BarcodeTypeCode128         | Code 128                         |
| BarcodeTypeCode16K         | Code 16K                         |
| BarcodeTypeCode32          | Code 32                          |
| BarcodeTypeCode39          | Code 39                          |
| BarcodeTypeCode49          | Code 49                          |
| BarcodeTypeCode93          | Code 93                          |
| BarcodeTypeComposite       | EAN-UCC Composite Code           |
| BarcodeTypeD2of5           | Discreate 2 of 5                 |
| BarcodeTypeDataMatrix      | Data Matrix                      |
| BarcodeTypeEAN128          | UCC/EAN-128                      |
| BarcodeTypeEAN13           | EAN-13                           |
| BarcodeTypeEAN13CouponCode | EAN-13 with Extended Coupon Code |
| BarcodeTypeEAN8            | EAN-8                            |
| BarcodeTypeI2of5           | Interleaved 2 of 5               |
| BarcodeTypeIATA            | IATA 2 of 5                      |
| BarcodeTypeISBT128         | ISBT 128                         |
| BarcodeTypeISBT128Concat   | ISBT-128 Concat.                 |
| BarcodeTypeJapanesePost    | Japanese Post                    |
| BarcodeTypeKixPost         | Kix (Netherlands) Post           |
| BarcodeTypeKoreaPost       | Korea Post                       |
| BarcodeTypeMacroMicroPDF   | Macro Micro PDF                  |
| BarcodeTypeMaxiCode        | MaxiCode                         |
| BarcodeTypeMicroPDF        | Micro PDF 417                    |
| BarcodeTypeMSI             | MSI                              |
| BarcodeTypeParameterFNC3   | Parameter (FNC3)                 |
| BarcodeTypePDF417          | PDF-417                          |
| BarcodeTypePlanetCode      | Planet Code                      |
| BarcodeTypePlesseyCode     | Plessey Code                     |

|                            |                       |
|----------------------------|-----------------------|
| BarcodeTypePostnet         | Postnet               |
| BarcodeTypeQRCode          | QR Code               |
| BarcodeTypeR2of5           | Straight 2 of 5       |
| BarcodeTypeRSS             | RSS                   |
| BarcodeTypeScanletWebcode  | Scanlet Webcode       |
| BarcodeTypeTelepen         | Telepen               |
| BarcodeTypeTLC39           | TCIF Linked Code 39   |
| BarcodeTypeTriopticCode    | Trioptic Code 39      |
| BarcodeTypeUPCA            | UPC-A                 |
| BarcodeTypeUPCE            | UPC-E                 |
| BarcodeTypeVeriCode        | VeriCode              |
| BarcodeTypeX2of5           | Matrix 2 of 5         |
| BarcodeTypeRSSLimited      | DataBar               |
| BarcodeTypeChineseSensible | Chinese-Sensible Code |

## 8. AsParamName Class

### 8.1. Function

#### 8.1.1. getName

| Function   | +(NSString *)getName:(ParamName)paramName; |           |  |
|--|--|-----------|--|
| Parameters   | IN/OUT                                     | Types     | Descriptions   |
| paramName  | IN   | ParamName | Enumerated type ParamName (see <a href="#">8.2.1</a> ) |
| -  | OUT  | NSString  | Text of the barcode type.                              |
| <p>■ <b>Function Description:</b><br/>Gets the barcode type as a string from parameters set with independent variables.</p> <p>■ <b>Sample code:</b><br/>NSString * barcodeName = [AsParamName getName:EAN13];</p> |  |           |  |

### 8.2. Enum

#### 8.2.1. ParamName

Enumeration of barcode types can be set:

| Enum Names           | Descriptions               |
|----------------------|----------------------------|
| Codabar              | Codabar                    |
| Code39               | Code 39                    |
| Code32Pharmaceutical | Code 32 Pharmaceutical     |
| I2of5                | Interleaved 2 of 5         |
| NEC2of5              | NEC 2 of 5                 |
| Code93               | Code 93                    |
| R2of5                | Straight 2 of 5 Industrial |

|                        |  |
|------------------------|--|
| A2of5                  | Straight 2 of 5 IATA                   |
| X2of5                  | Matrix 2 of 5                          |
| Code11                 | Code 11                                |
| Code128                | Code 128                               |
| GS1128                 | GS1-128                                |
| Telepen                | Telepen                                |
| UPCA                   | UPC-A                                  |
| UPCACouponCode         | UPC-A/EAN-13 with Extended Coupon Code |
| CouponGS1DataBarOutput | Coupon GS1 DataBar Output              |
| UPCE0                  | UPC-E0                                 |
| UPCE1                  | UPC-E1                                 |
| EAN13                  | EAN/JAP-13                             |
| EAN8                   | EAN/JAP-8                              |
| MSI                    | MSI                                    |
| RSS14                  | RSS-14                                 |
| RSSLimit               | RSS Limited                            |
| RSSExp                 | RSS Expanded                           |
| TriopticCode           | Trioptic Code                          |
| CodablockA             | Codablock A                            |
| CodablockF             | Codablock F                            |
| PDF417                 | PDF 417                                |
| MacroPDF417            | MacroPDF417                            |
| MicroPDF               | MicroPDF 417                           |
| ComCode                | EAN/UCC Composite Code                 |
| GS1Emulation           | GS1 Emulation                          |
| TLC39                  | TCIF Linked Code 39                    |
| ChinaPost              | China Post                             |
| KoreaPost              | Korea Post                             |
| QRCode                 | QR Code                                |
| Matrix                 | Data Matrix                            |
| MaxiCode               | MaxiCode                               |
| AztecCode              | Aztec Code                             |
| HanXinCode             | Chinese Sensible (Han Xin) Code        |

|                      |                        |
|----------------------|------------------------|
| PostalCodes          | 2D Postal Codes        |
| Code39Pharmaceutical | Code 39 Pharmaceutical |

## 9. AsParamValue Class

### 9.1. Properties

#### 9.1.1. @property (assign, readwrite) ParamName paramName;

**Description:**

Gets or sets the type of barcode that AsReaderGUN can scan.  
Enumerated type ParamName (see [8.2.1](#))

#### 9.1.2. @property (assign, readwrite) unsigned int value;

**Description:**

Gets or sets whether the specified barcode type can be scanned by  
AsReaderGUN.  
1: Can be scanned  
0: Can not be scanned

## 9.2. Function

### 9.2.1. setEnabled

|  |                                 |              |  |
|--|---------------------------------|--------------|--|
| <b>Function</b>  | - (void)setEnabled:(BOOL)value; |              |  |
| <b>Parameters</b>  | <b>IN/OUT</b>                   | <b>Types</b> | <b>Descriptions</b>  |
| value  | IN                              | BOOL         | Sets whether to allow scanning of a specified barcode type.<br>YES: scanning is allowed<br>NO: scanning is not allowed |
| <p>■ <b>Function Description:</b><br/>Sets whether AsReaderGUN can scan the specified barcode type set by ParamName.</p> <p>■ <b>Sample code:</b><br/>AsParamValue * paramValue = [asReader getBarcodeParam: EAN13];<br/>[paramValue setEnabled:NO];</p> |                                 |              |  |

## 10. AsMaskActionType Class

### 10.1. Function

#### 10.1.1. toString

| Function  | +(NSString*)toString:(MaskActionType)actionType<br>targetType:(MaskTargetType)targetType; |                |  |
|---|---|----------------|--|
| Parameters  | IN/OUT  | Types          | Descriptions   |
| actionType  | IN  | MaskActionType | Mask action type<br>Enumerated type MaskActionType<br>(see <a href="#">5.1.9</a> ) |
| targetType  | IN  | MaskTargetType | Mask target type<br>Enumerated type<br>MaskTargetType (see <a href="#">5.1.8</a> ) |
| -   | OUT   | NSString       | Description of Mask action type<br>and tag type                                    |
| <p>■ <b>Function Description:</b><br/>According the MaskActionType and MaskTargetType, gets the corresponding explanation through the string.</p> <p>■ <b>Sample code:</b><br/>NSString * actionMessage = [AsMaskActionType toString:MaskAction_AB<br/>targetType:MaskTarget_S0];</p> |   |                |  |

# 11. Appendix

## 11.1. Parameter Informations

※Only for ASR-L251G.

| Properties              | Default Value       | Store in L251G F/W | Save in SDK instance | Restore to the Default Value When Power On |
|-------------------------|---------------------|--------------------|----------------------|--|
| buzzer                  | ON (HIGH)           | √                  |                      |  |
| vibrator                | ON                  | √                  |                      |  |
| operationTime           | 0 (ms):Not use time |                    | √                    |  |
| inventoryTime           | 400(ms)             |                    |                      | √  |
| idleTime                | 300(ms)             |                    |                      | √  |
| sleepTime               | 65535(s)            | √                  |                      |  |
| autoOffTime             | 65535(s)            | √                  |                      |  |
| accessPassword          | "0"                 |                    |                      | √  |
| inventorySession        | 0:S0                |                    |                      | √  |
| sessionFlag             | 2:A or B            |                    |                      | √  |
| serialNumber            |                     | √                  |                      |  |
| continuousMode          | YES                 |                    |                      | √  |
| powerGain               | 300:30.0 dBm        |                    |                      | √  |
| isUseKeyAction          |                     |                    |                      |  |
| useSelectionMask        | No                  |                    | √                    |  |
| rssiMode                | No                  |                    | √                    |  |
| epcMaskMatchMode        | No                  |                    | √                    |  |
| algorithm               | 1:Dynamic           |                    |                      | √  |
| minQ                    | 0                   |                    |                      | √  |
| maxQ                    | 15                  |                    |                      | √  |
| qValue                  | 4                   |                    |                      | √  |
| linkProfileValue        | 1                   |                    |                      | √  |
| defaultLinkProfileValue |                     |                    |                      |  |
| maskTypeValue           | 0:NoMASK            |                    | √                    |  |