AsReader®

AsReaderP252B SDK

Objective-C SDK Reference Guide

Asterisk Inc.

Revision History

Version	Description	Date
V1.0	Initial version	2024/3/15

Contents

Introduction

This manual provides the following information to developers developing Objective-C applications using the SDK.

- How the development environment is built
- Description of various SDK library functions

Development tools:

- Xcode 13.1

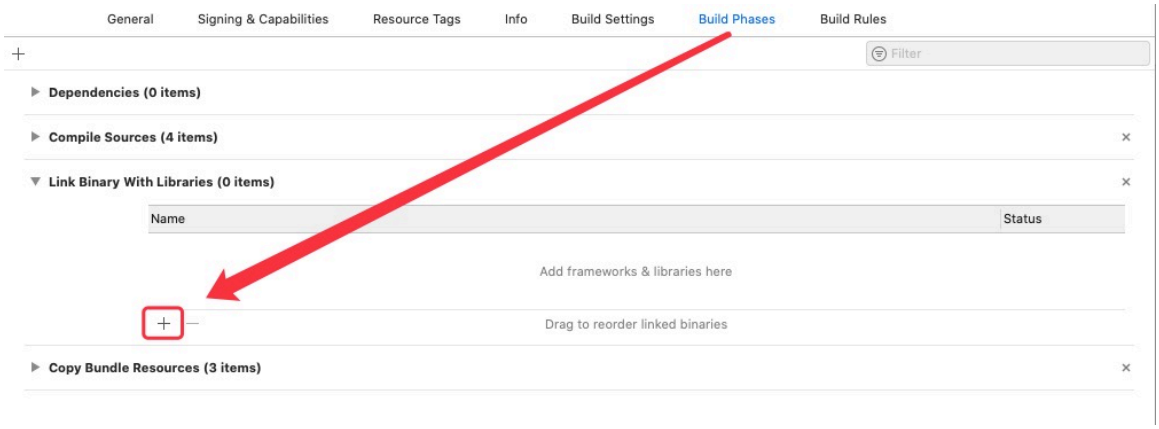
System requirements:

- iOS 12.0+

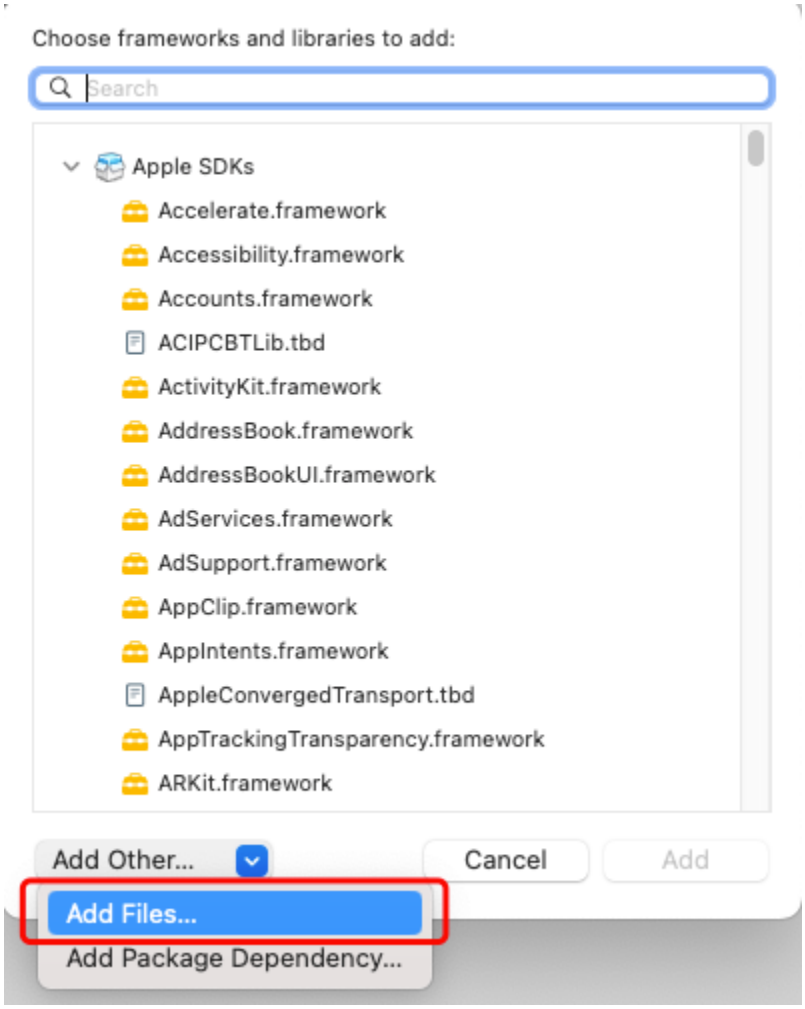
1 SDK Usage

1.1. Add SDK

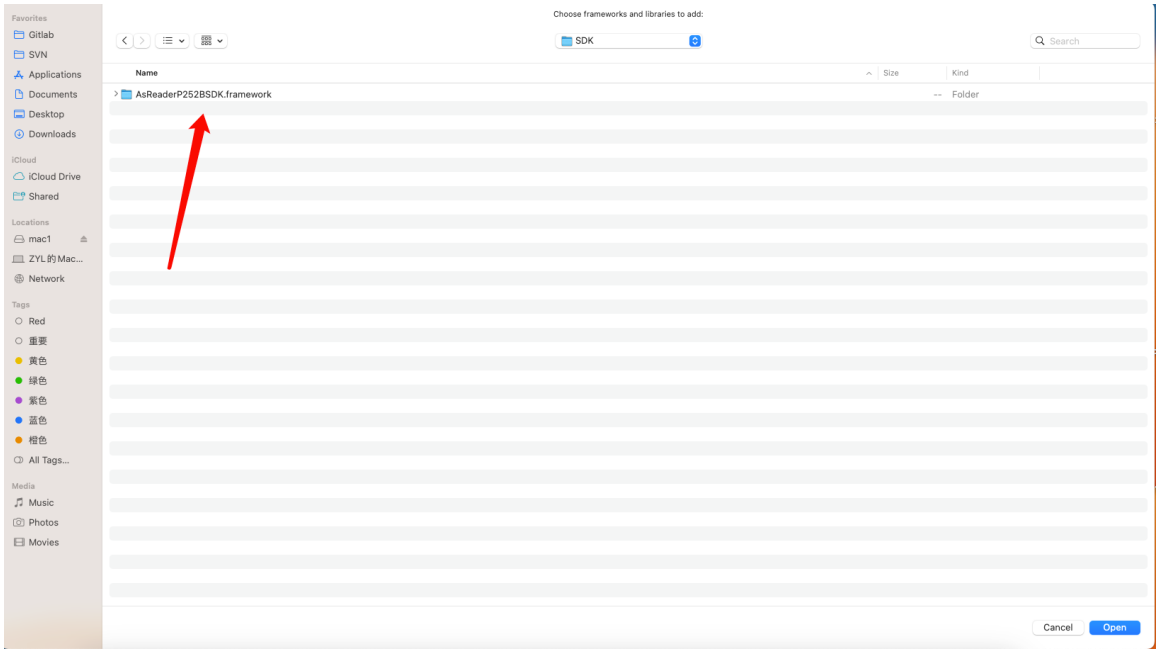
1. TARGET -> Build phases -> Link Binary with Libraries



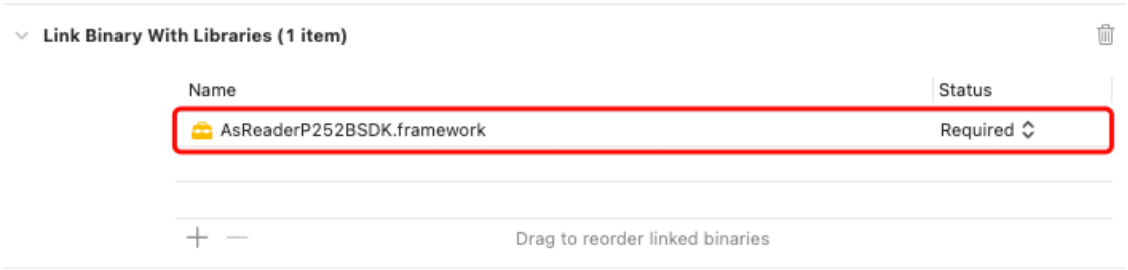
2. Select “Add Other…”→ “Add Files…”



3. Add AsReaderP252BSDK.framework



4. The SDK is shown as follows after being added:



1.2. Import header files

For Objective-C project, header files need to be imported in classes where you want to use the SDK. Reference is as follows:

#import <AsReaderP252BSDK/AsReaderP252BSDK.h>

1.3. Use of SDK

1.3.1. Create and initialize the object of AsReaderP252BDeviceManager (singleton mode)

```
AsReaderP252BDeviceManager *bluetoothManager = [AsReaderP252BDeviceManager shared];
```

1.3.2. Assign delegates

```
bluetoothManager.delegate = self;
```

1.3.3. Start search

```
[bluetoothManager startDiscovered];
```

Receives the array of the AsReader device searched via Bluetooth.

```
- (void)didDiscoverAsReaderP252BDeviceList:(NSArray<AsReaderP252B *>
*)asReaderP252BList{
   //asReaderP252BList: The array elements are objects of AsReaderP252B.
}
```

1.3.4. Connect to the AsReader P252B device

Obtain the AsReaderP252B in asReaderP252BList for connection.

Function of connecting to AsReaderP252B: (asReaderP252B is the object of AsReaderP252B)

```
[asReaderP252B connect];
```

The delegate function for receiving connection results.

```
-(void)whenAsReaderP252BIsConnected:(BOOL)isConnected{
      if (isConnected){
          //Connected
      } else {
          //Disconnected
      }
}
```

1.3.5. Start inventory of RFID tags

1.3.5.1. inventory

```
AsReaderP252BResultCode resultCode = [asReaderP252B inventory];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

1.3.5.1.1. Delegate for receiving the execution result for starting inventory

```
-(void)changedAsReaderP252BActionState:(AsReaderP252BCommandType)action
resultCode:(NSInteger)resultCode{
   //action: Enum AsReaderP252BCommandType (see 7.1.1)
   //resultCode: Enum AsReaderP252BResultCode (see 8.4.1)
}
```

1.3.5.1.2. Delegate for receiving data of inventory

```
- (void)readAsReaderP252BTag:(NSString *)tag
                        rssi:(float)rssi
                       phase:(float)phase
                   frequency:(float)frequency{
   //tag: The PCEPC data of RFID tag. (Hex)
   //rssi: The RSSI data of RFID tag.
   //phase: The phase value of RFID tag.
   //frequency: The frequency of RFID tag when it is inventoried.
}
```

1.3.6. Stop inventory of RFID tags

```
AsReaderP252BResultCode resultCode = [asReaderP252B stop];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

The delegate function for receiving the execution results of stopping inventory:

```
-(void)changedAsReaderP252BActionState:(AsReaderP252BCommandType)action
resultCode:(NSInteger)resultCode{
   //action: Enum AsReaderP252BCommandType (see 7.1.1)
   //resultCode: Enum AsReaderP252BResultCode (see 8.4.1)
}
```

1.3.7. Start barcode scanning

1.3.7.1. startDecode

```
AsReaderP252BResultCode resultCode = [asReaderP252B startDecode];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

1.3.7.1.1. Delegate for receiving the execution result for starting scan

```
-(void)accessAsReaderP252BResult:(AsReaderP252BResultCode)error
                     actionState:(AsReaderP252BCommandType)action
                             epc:(NSString *)epc
                            data:(NSString *)data
                            rssi:(float)rssi
                           phase:(float)phase
                       frequency:(float)frequency{
  //error: Enum AsReaderP252BResultCode (see 8.4.1)
  //action: Enum AsReaderP252BCommandType (see 7.1.1)
}
```

1.3.7.1.2. Delegate for receiving data of scanning

```
-(void)detectBarcode:(AsReaderP252BBarcodeType)barcodeType
         barcodeData:(NSData *)barcodeData{
  //barcodeType: Enum AsReaderP252BBarcodeType (see 4.2.1).
  //barcodeData: The scanned barcode data.
}
```

1.3.8. Stop barcode scanning

```
AsReaderP252BResultCode resultCode = [asReaderP252B stop];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

The delegate function for receiving the execution results of stopping the scan.

```
-(void)changedAsReaderP252BActionState:(AsReaderP252BCommandType)action
resultCode:(NSInteger)resultCode{
  //action: Enum AsReaderP252BCommandType (see 7.1.1)
  //resultCode: Enum AsReaderP252BResultCode (see 8.4.1)
}
```

1.3.9. Disconnect from the AsReaderP252B device

```
[asReaderP252B disconnect];
```

The delegate function for receiving disconnection results.

```
-(void)whenAsReaderP252BIsConnected:(BOOL)isConnected{
    if (isConnected){
        //Connected.
    } else {
        //Disconnected.
    }
}
```

2 AsReaderP252B

The objects of AsReaderP252B are used for connecting and disconnecting the ASR-P252B device, inventory of RFID data, stopping inventory, doing some related settings of inventory or scanning, reading tags, writing tags, locking tags and killing tags, and scanning barcodes, etc.

2.1. Properties

Property names	Properties	Types	Descriptions
minQ	nonatomic, assign	int	The minimum Q value of the inventory.
maxQ	nonatomic, assign	int	The maximum Q value of the inventory.
qValue	nonatomic, assign	int	The Q value of the inventory. Range: 0~15.
idleTime	nonatomic, assign	int	The duration of the radio waves non-emitted when inventorying. (Unit: 10ms) Range: 0~65535.
powerGain	nonatomic, assign	int	The power of the inventory. Range: 2~30.
sleepTime	nonatomic, assign	int	The auto sleep time when the AsReader device is not connected. (Unit: s) Range: 0~1800.
autoOffTime	nonatomic, assign, readonly	int	The auto off time when the AsReader device is not connected. (Unit: s) Range: 0~1800.
operationTime	nonatomic, assign	int	The duration of an inventory. (Unit: s) Range: 0~1800.
linkProfileValue	nonatomic, assign	int	The link profile value of the inventory. Range: 0~3.
rssiMode	nonatomic, assign, readonly	BOOL	Whether to get RSSI data when inventorying RFID tags. YES: Gets RSSI data. NO: Do not get RSSI data.
continuousMode	nonatomic, assign	BOOL	Whether to inventory continuously. YES: Continuous mode valid. NO: Continuous mode invalid.
serialNumber	nonatomic, strong	NSString	The serial number of the AsReader device.
accessPassword	nonatomic, strong	NSString	The access password of the RFID tag.

Buzzer	nonatomic, assign	AsReaderP252BbuzzerState	The buzzer status of the AsReader device. Enum AsReaderP252BbuzzerState (see 8.4.4)
sessionFlag	nonatomic, assign	AsReaderP252BsessionFlag	The session flag of the inventory. Enum AsReaderP252BsessionFlag (see 8.4.6)
inventorySession	nonatomic, assign	AsReaderP252BsessionType	The session value of the inventory. Enum AsReaderP252BsessionType (see 8.4.5)

2.2. Functions

2.2.1. setDelegate

- (void)setDelegate:(id<AsReaderP252BDelegate>)delegate;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	delegate	In	AsReaderP252BDelegate	AsReaderP252BDelegate (see 3)
■Function description: Sets AsReaderP252BDelegate. **■Sample code:** (Note: asReaderP252B is the object of AsReaderP252B.) [asReaderP252B setDelegate:self];				

2.2.2. batteryStatus

- (int)batteryStatus;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	int	The battery power level of the AsReader device. Range: 0, 1, 2, 3, 4 0: 0 1: 25% 2: 50% 3: 75% 4: 100%
■Function description: Gets the battery power level of the AsReader device. **■Sample code:** (Note: asReaderP252B is the object of AsReaderP252B.) int battery = [asReaderP252B batteryStatus];				

2.2.3. regionName

- (NSString *)regionName;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	NSString	The region or country where the AsReader device is set to meet the regulations.

■Function description:
Gets the region or country where the AsReader device is set to meet the regulations.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)

```
NSSting *regionName = [asReaderP252B regionName];
```

2.2.4. serialNumber

- (NSString *)serialNumber;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	NSString	The serial number of the AsReader device.

■Function description:
Gets the serial number of the AsReader device.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)

```
NSSting *serialNumber = [asReaderP252B serialNumber];
```

2.2.5. rFModuleVersion

- (NSString *)rFModuleVersion;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	NSString	The firmware version of the RFID module.

■Function description:
Gets the firmware version of the RFID module.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)

```
NSSting *rFModuleVersion = [asReaderP252B rFModuleVersion];
```

2.2.6. hardwareVersion

- (NSString *)hardwareVersion;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	NSString	The hardware version of the AsReader device.

■**Function description:**
Gets the hardware version of the AsReader device.

■**Sample code:** (Note: asReaderP252B is the object of AsReaderP252B.)
NSSting *hardwareVersion = [asReaderP252B hardwareVersion];

2.2.7. firmwareVersion

- (NSString *)firmwareVersion;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	NSString	The firmware version of the AsReader device.

■**Function description:**
Gets the firmware version of the AsReader device.

■**Sample code:** (Note: asReaderP252B is the object of AsReaderP252B.)
NSSting *firmwareVersion = [asReaderP252B firmwareVersion];

2.2.8. getMessageWithAsReaderP252BResultCode

-(NSString*)getMessageWithAsReaderP252BResultCode:(AsReaderP252BResultCode)code;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	code	In	AsReaderP252BResultCode	Enum AsReaderP252BResultCode (see 8.4.1)
Return value	-	Out	NSString	Error message.

■**Function description:**
Converts result code to error message.

■**Sample code:** (Note: asReaderP252B is the object of AsReaderP252B.)
AsReaderP252BResultCode error; // Error codes returned by the SDK.
NSSting *errorString = [asReaderP252B getMesageWithAsReaderP252BResultCode:error];

2.2.9. getAction

- (AsReaderP252BCommandType)getAction;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	AsReaderP252BCommandType	Action command. Enum AsReaderP252BCommandType (see 7.1.1)

■**Function description:**
Gets the action status of the AsReader device.

■**Sample code:** (Note: asReaderP252B is the object of AsReaderP252B.)
```
AsReaderP252BCommandType type = [asReaderP252B getAction];
```

2.2.10. powerGainRange

- (AsReaderP252BMinMaxValue)powerGainRange;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	AsReaderP252BMinMaxValue	The power range can be set. AsReaderP252BMinMaxValue (see 8.5.1)

■**Function description:**
Gets the power range of the AsReader device.

■**Sample code:** (Note: asReaderP252B is the object of AsReaderP252B.)
```
int min = asReaderP252B.powerGainRange.min; // The minimum Q value of the inventory.
int max = asReaderP252B.powerGainRange.max; // The maximum Q value of the inventory.
```

2.2.11. inventory

- (AsReaderP252BResultCode)inventory;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	AsReaderP252BResultCode	Function execution results. Enum AsReaderP252BResultCode (see 8.4.1)

■Function description:
The AsReader device starts to inventory RFID tags.
Once the function is executed, the delegate functions “readAsReaderP252BTag” (see 3.3), “changedAsReaderP252BActionState” (see 3.4) will be called back.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)

```
AsReaderP252BResultCode resultCode = [asReaderP252B inventory];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

2.2.12. inventoryHumidityTag

- (AsReaderP252BResultCode)inventoryHumidityTag;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	AsReaderP252BResultCode	Function execution results. Enum AsReaderP252BResultCode (see 8.4.1)

■Function description:
The AsReader device starts to inventory RFID tags with humidity chip.
Once the function is executed, the delegate functions “changedAsReaderP252BActionState” (see 3.4), “readAsReaderP252BHumidityTag” (see 3.7) will be called back.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)

```
AsReaderP252BResultCode resultCode = [asReaderP252B inventoryHumidityTag];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

2.2.13. inventoryTemperatureTagWithType

-(AsReaderP252BResultCode)inventoryTemperatureTagWithType:(AsReaderP252BTemperatureTagType)type;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	type	In	AsReaderP252BTemperatureTagType	The type of temperature tag. Enum AsReaderP252BTemperatureTagType (see 8.4.3)
Return value	-	Out	AsReaderP252BResultCode	Function execution results. Enum AsReaderP252BResultCode (see 8.4.1)

■Function description:
The AsReader device starts to inventory RFID tags with temperature chip.
Once the function is executed, the delegate functions “changedAsReaderP252BActionState” (see 3.4) and “readAsReaderP252BTemperatureTag” (see 3.6) will be called back.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)

```
AsReaderP252BResultCode resultCode = [asReaderP252B
inventoryTemperatureTagWithType:AsReaderP252BTemperatureTagType_0];
    // Inventory RFID tags with Magnus-S3 temperature chip
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

2.2.14. stop

- (AsReaderP252BResultCode)stop;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	AsReaderP252B ResultCode	Function execution results. Enum AsReaderP252BResultCode (see 8.4.1)

■Function description:
Stops inventorying tags/scanning barcodes.
This function returns the execution result without waiting for the execution result of the AsReader device.
Once the function is executed, the delegate function “changedAsReaderP252BActionState” (see 3.4) will be called back.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)

```
AsReaderP252BResultCode resultCode = [asReaderP252B stop];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

2.2.15. stopSync

- (AsReaderP252BResultCode)stopSync;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	AsReaderP252BResultCode	Function execution results. Enum AsReaderP252BResultCode (see 8.4.1)

■Function description:
Stops inventorying tags/scanning barcodes.
This function needs to wait for the execution result of the AsReader device before returning the execution result.
Once the function is executed, the delegate function “changedAsReaderP252BActionState” (see 3.4) will be called back.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)

```
AsReaderP252BResultCode resultCode = [asReaderP252B stopSync];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

2.2.16. readMemory

-(AsReaderP252BResultCode)readMemory:(AsReaderP252BMemoryBank)bank offset:(int)offset length:(int)length;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	bank	In	AsReaderP252BMemoryBank	The memory bank of the RFID tag. Enum AsReaderP252BMemoryBank (see 8.4.2)
Parameter	offset	In	int	The start address of the memory bank. (Unit: Word)
Parameter	length	In	int	The length of the memory bank to be read. (Unit: Word)
Return value	-	Out	AsReaderP252BResultCode	Function execution results. Enum AsReaderP252BResultCode (see 8.4.1)

■**Function description:**
To read memory bank of the RFID tag.
Once the function is executed, the delegate function “accessAsReaderP252BResult” (see 3.5) will be called back.

■**Sample code:** (Note: asReaderP252B is the object of AsReaderP252B.)

```
AsReaderP252BResultCode resultCode = [asReaderP252B
readMemory:AsReaderP252BBank_EPC offset:2 length:4];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

2.2.17. writeMemory

-(AsReaderP252BResultCode)writeMemory:(AsReaderP252BMemoryBank)bank offset:(int)offset value:(NSString *)value;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	bank	In	AsReaderP252BMemoryBank	The memory bank of the RFID tag. Enum AsReaderP252BMemoryBank (see 8.4.2)
Parameter	offset	In	int	The start address of the memory bank. (Unit: Word)
Parameter	value	In	NSString	The data to be written to the RFID tag. (Hex)
Return value	-	Out	AsReaderP252BResultCode	Function execution results. Enum AsReaderP252BResultCode (see 8.4.1)

■**Function description:**
Writes data to the target memory bank of the RFID tag.
Once the function is executed, the delegate function “accessAsReaderP252BResult” (see 3.5) will be called back.

■**Sample code:** (Note: asReaderP252B is the object of AsReaderP252B.)

```
AsReaderP252BResultCode resultCode = [asReaderP252B
writeMemory:AsReaderP252BBank_EPC offset:2 value:@"1234"];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

2.2.18. lock

- (AsReaderP252BResultCode)lock:(AsReaderP252BLockParam *)param;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	param	In	AsReaderP252BLockParam	The object of AsReaderP252BLockParam. (see 8.1)
Return value	-	Out	AsReaderP252BResultCode	Function execution results. Enum AsReaderP252BResultCode (see 8.4.1)

■Function description:
Locks a target memory bank of the RFID tag.
Once the function is executed, the delegate function “accessAsReaderP252BResult” (see 3.5) will be called back.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B, param is the object of AsReaderP252BLockParam.)

```
AsReaderP252BResultCode resultCode = [asReaderP252B lock:param];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

2.2.19. unlock

- (AsReaderP252BResultCode)unlock:(AsReaderP252BLockParam *)param;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	param	In	AsReaderP252BLockParam	The object of AsReaderP252BLockParam. (see 8.1)
Return value	-	Out	AsReaderP252BResultCode	Function execution results. Enum AsReaderP252BResultCode (see 8.4.1)

■Function description:
Unlocks the locked memory bank of the RFID tag. After unlocking, you can use the default password to overwrite the tag data.
Once the function is executed, the delegate function “accessAsReaderP252BResult” (see 3.5) will be called back.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B, param is the object of AsReaderP252BLockParam.)

```
AsReaderP252BResultCode resultCode = [asReaderP252B unlock:param];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

2.2.20. permaLock

- (AsReaderP252BResultCode)permaLock:(AsReaderP252BLockParam *)param;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	param	In	AsReaderP252BLockParam	The object of AsReaderP252BLockParam (see 8.1)
Return value	-	Out	AsReaderP252BResultCode	Function execution results. Enum AsReaderP252BResultCode (see 8.4.1)

■Function description:
Permanently locks a target memory bank of the RFID tag.
Permanently locked tag data cannot be changed or unlocked.
Once the function is executed, the delegate function “accessAsReaderP252BResult” (see 3.5) will be called back.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B, param is the object of AsReaderP252BLockParam.)

```
AsReaderP252BResultCode resultCode = [asReaderP252B permaLock:param];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

2.2.21. kill

- (AsReaderP252BResultCode)kill:(NSString *)killPassword				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	killPassword	In	NSString	Kill password.
Return value	-	Out	AsReaderP252BResultCode	Function execution results. Enum AsReaderP252BResultCode (see 8.4.1)

■Function description:
Kills a tag. The killed tag cannot be used.
Once the function is executed, the delegate function “accessAsReaderP252BResult” (see 3.5) will be called back.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)

```
AsReaderP252BResultCode resultCode = [asReaderP252B kill: @"12345678"];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

2.2.22. defaultParameter

- (AsReaderP252BResultCode)defaultParameter;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	AsReaderP252BResultCode	Function execution results. Enum AsReaderP252BResultCode (see 8.4.1)

■Function description:
Restores default settings.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)

```
AsReaderP252BResultCode resultCode = [asReaderP252B defaultParameter];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

2.2.23. saveParameter

- (AsReaderP252BResultCode)saveParameter;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	AsReaderP252BResultCode	Function execution results. Enum AsReaderP252BResultCode (see 8.4.1)

■Function description:
Saves the parameters to the AsReader device.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)

```
AsReaderP252BResultCode resultCode = [asReaderP252B saveParameter];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

2.2.24. startDecode

- (AsReaderP252BResultCode)startDecode;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	AsReaderP252BResultCode	Function execution results. Enum AsReaderP252BResultCode (see 8.4.1)

■Function description:
The AsReader device starts to scan barcodes.
Once the function is executed, the delegate function “detectBarcode” (see 3.12) will be called back.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)

```
AsReaderP252BResultCode resultCode = [asReaderP252B startDecode];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

2.2.25. setBarcodeTimeOut

- (AsReaderP252BResultCode)setBarcodeTimeOut:(int)barcodeTimeOut;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	barcodeTimeOut	In	int	The timeout when the AsReader device scans a barcode. (Unit: s) Range: 4~300.
Return value	-	Out	AsReaderP252BResultCode	Function execution results. Enum AsReaderP252BResultCode (see 8.4.1)

■Function description:
Sets the timeout when the AsReader device scans a barcode.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)

```
AsReaderP252BResultCode resultCode = [asReaderP252B setBarcodeTimeOut:4];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

2.2.26. getBarcodeTimeOut

- (int)getBarcodeTimeOut;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	int	The timeout when the AsReader device scans a barcode. (Unit: s)

■Function description:
Gets the timeout when the AsReader device scans a barcode.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)
```
int barcodeTimeout = [asReaderP252B getBarcodeTimeOut];
```

2.2.27. epcMaskCount

- (int)epcMaskCount;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	int	Number of EPC masks.

■Function description:
Gets the number of EPC masks.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)
```
int count= [asReaderP252B epcMaskCount];
```

2.2.28. addEpcMask

- (AsReaderP252BResultCode)addEpcMask:(int)offset length:(int)length mask:(NSString *)mask;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	offset	In	int	The start address of the mask. (Unit: bit)
Parameter	length	In	int	The length of the mask. (Unit: bit)
Parameter	mask	In	NSString	The mask data. (Hex)
Return value	-	Out	AsReaderP252BResultCode	Function execution results. Enum AsReaderP252BResultCode (see 8.4.1)

■Function description:
Adds EPC mask.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)

```
AsReaderP252BResultCode resultCode = [asReaderP252B addEpcMask:32 length:16
mask:@"1234"];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

2.2.29. addEpcMask

-(AsReaderP252BResultCode)addEpcMask:(AsReaderP252BSelectMaskEPCParam *)mask;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	mask	In	AsReaderP252BSelectMaskEPCParam	The object of AsReaderP252BSelectMaskEPCParam. (see 8.2)
Return value	-	Out	AsReaderP252BResultCode	Function execution results. Enum AsReaderP252BResultCode (see 8.4.1)

■Function description:
Adds EPC mask as the object of AsReaderP252BSelectMaskEPCParam.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B, mask is the object of AsSelectMaskEPCParam.)

```
AsReaderP252BResultCode resultCode = [asReaderP252B addEpcMask:mask];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

2.2.30. getEpcMask

- (AsReaderP252BSelectMaskEPCParam *)getEpcMask:(int)index;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	index	In	int	Index of the EPC mask.
Return value	-	Out	AsReaderP252BSelectMaskEPCParam	The object of AsReaderP252BSelectMaskEPCParam. (see 8.2)

■Function description:
Gets the EPC mask for the specified index.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)
```
AsReaderP252BSelectMaskEPCParam *mask= [asReaderP252B getEpcMask:0];
```

2.2.31. clearEpcMask

- (AsReaderP252BResultCode)clearEpcMask;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	AsReaderP252BResultCode	Function execution results. Enum AsReaderP252BResultCode (see 8.4.1)

■Function description:
Removes EPC mask data.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)
```
AsReaderP252BResultCode resultCode = [asReaderP252B clearEpcMask];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

2.2.32. setFrequencyAutomatic

- (void)setFrequencyAutomatic:(BOOL)isAutomatic;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	isAutomatic	In	BOOL	Whether to use frequency hopping. YES: Use. NO: Do not use.

■Function description:
Sets whether to use frequency hopping when inventorying.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)
```
[asReaderP252B setFrequencyAutomatic:YES];
```

2.2.33. getFrequencyAutomatic

- (BOOL)getFrequencyAutomatic;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	BOOL	Whether to use frequency hopping. YES: Use. NO: Do not use.

■Function description:
Gets whether to use frequency hopping when inventorying.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)
```
BOOL status = [asReaderP252B getFrequencyAutomatic];
if (status) {
   //The frequency hopping is valid.
} else {
   //The frequency hopping is invalid.
}
```

2.2.34. setFrequency

- (void)setFrequency:(NSArray *)table;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	table	In	NSArray	The frequency list. The array elements are objects of AsReaderP252BLbtItem. (see 8.3)

■Function description:
Sets the frequency list of the AsReader device.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)
```
[asReaderP252B setFrequency:array];
```

2.2.35. getFrequency

- (NSArray *)getFrequency;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	NSArray	The frequency list.

■Function description:
Gets the frequency list of the AsReader device.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)
```
NSArray * lbtArray = [asReaderP252B getFrequency];
```

2.2.36. setScanMode

- (void)setScanMode:(AsReaderP252BScanMode)scanMode;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	scanMode	In	AsReaderP252BScanMode	Barcode mode or RFID mode. Enum AsReaderP252BScanMode (see 7.1.2)

■Function description:
Sets whether the AsReader device is in barcode mode or RFID mode.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)
```
[asReader setScanMode: AsReaderP252BRFIDScanMode];
```

2.2.37. getBaudRateList

- (NSMutableArray *)getBaudRateList;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	NSMutableArray	The baud rate list.

■Function description:
Gets the baud rate list of the AsReader device.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)
```
NSMutableArray *array = [asReaderP252B getBaudRateList];
```

2.2.38. setBaudRate

- (AsReaderP252BResultCode)setBaudRate:(int)baudRate;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	baudRate	In	int	The index of baud rate list.
Return value	-	Out	AsReaderP252BResultCode	Function execution results. Enum AsReaderP252BResultCode (see 8.4.1)

■Function description:
Sets the index of baud rate list.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)

```
AsReaderP252BResultCode resultCode = [asReaderP252B setBaudRate:1];
if (resultCode == AsReaderP252BResultNoError) {
    //Function execution succeeded.
}else{
    //Function execution failed.
}
```

2.2.39. getBaudRate

- (int)getBaudRate;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	int	The index of baud rate list.

■Function description:
Gets the index of baud rate list.

■Sample code: (Note: asReaderP252B is the object of AsReaderP252B.)

```
int baudRate= [asReaderP252B getBaudRate];
```

3 AsReaderP252BDelegate

3.1. whenAsReaderP252BIsConnected

- (void)whenAsReaderP252BIsConnected:(BOOL)isConnected;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	isConnected	Out	BOOL	Connection status. YES: Connected. NO: Disconnected.

■Function description:
Receives the connection status of the AsReader device.
The delegate function will be called back after calling the function “connect” (see 6.2.1) and “disconnect” (see 6.2.2).

■Sample code:
```
-(void)whenAsReaderP252BIsConnected:(BOOL)isConnected{
      if (isConnected){
          //Connected.
      } else {
          //Disconnected.
      }
}
```

3.2. updateAsReaderP252BDeviceState

- (void)updateAsReaderP252BDeviceState:(AsReaderP252BResultCode)error;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	error	Out	AsReaderP252BResultCode	Enum AsReaderP252BResultCode (see 8.4.1)

■Function description:
The delegate function will be called back if an error occurs after calling some functions/properties.
*The functions and properties are as follows:
2.1. Properties, 2.2.2. batteryStatus, 2.2.3. regionName, 2.2.4. serialNumber, 2.2.5. rFModuleVersion, 2.2.7. firmwareVersion, 2.2.22. defaultParameter, 2.2.27. epcMaskCount, 2.2.28. addEpcMask, 2.2.29. addEpcMask, 2.2.30. getEpcMask, 2.2.31. clearEpcMask, 2.2.34. setFrequency, 2.2.35. getFrequency, 2.2.36. setScanMode

■Sample code:
```
- (void)updateAsReaderP252BDeviceState:(AsReaderP252BResultCode)error {
   //error: Enum AsReaderP252BResultCode (see 8.4.1)
}
```

3.3. readAsReaderP252BTag

- (void)readAsReaderP252BTag:(NSString *)tag
rssi:(float)rssi
phase:(float)phase
frequency:(float)frequency;

	Names	In/Out	Types	Descriptions
Parameter	tag	Out	NSString	The PCEPC data of the RFID tag. (Hex)
Parameter	rssi	Out	float	The RSSI data of the RFID tag.
Parameter	phase	Out	float	The phase value of the RFID tag.
Parameter	frequency	Out	float	The frequency of the RFID tag when it is inventoried.

■Function description:
Receives the data of the inventoried tag.
The delegate function will be called back after calling the function “inventory” (see 2.2.11).

■Sample code:

```
- (void)readAsReaderP252BTag:(NSString *)tag
                        rssi:(float)rssi
                       phase:(float)phase
                   frequency:(float)frequency{
   //tag: The PCEPC data of RFID tag. (Hex)
   //rssi: The RSSI data of RFID tag.
   //phase: The phase value of RFID tag.
   //frequency: The frequency of RFID tag when it is inventoried.
}
```

3.4. changedAsReaderP252BActionState

-(void)changedAsReaderP252BActionState:(AsReaderP252BCommandType)action resultCode:(NSInteger)resultCode;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	action	Out	AsReaderP252BCommandType	Enum AsReaderP252BCommandType (see 7.1.1)
Parameter	resultCode	Out	AsReaderP252BResultCode	Enum AsReaderP252BResultCode (see 8.4.1)

■Function description:
Receives the execution results for starting inventory/stopping inventory.
The delegate function will be called back after calling the function “inventory” (see 2.2.11), “inventoryHumidityTag” (see 2.2.12), “inventoryTemperatureTagWithType” (see 2.2.13), “stop” (see 2.2.14), “stopSync” (see 2.2.15).

■Sample code:

```
-(void)changedAsReaderP252BActionState:(AsReaderP252BCommandType)action
resultCode:(NSInteger)resultCode{
   //action: Enum AsReaderP252BCommandType (see 7.1.1)
   //resultCode: Enum AsReaderP252BResultCode (see 8.4.1)
}
```

3.5. accessAsReaderP252BResult

-(void)accessAsReaderP252BResult:(AsReaderP252BResultCode)error
actionState:(AsReaderP252BCommandType)action
epc:(NSString *)epc
data:(NSString *)data
rssi:(float)rssi
phase:(float)phase
frequency:(float)frequency;

	Names	In/Out	Types	Descriptions
Parameter	error	Out	AsReaderP252BResultCode	Enum AsReaderP252BResultCode (see 8.4.1)
Parameter	action	Out	AsReaderP252BCommandType	Enum AsReaderP252BCommandType (see 7.1.1)
Parameter	epc	Out	NSString	The EPC data of RFID tag. (Hex)
Parameter	data	Out	NSString	The read RFID tag data. (Hex)
Parameter	rssi	Out	float	The RSSI data of RFID tag.
Parameter	phase	Out	float	The phase value of RFID tag.
Parameter	frequency	Out	float	The frequency of RFID tag when it is inventoried.

■Function description:
Receives the execution results for reading tag/writing tag/locking tag/killing tag.
The delegate function will be called back after calling the function “readMemory” (see 2.2.16), “writeMemory” (see 2.2.17), “lock” (see 2.2.18), “unlock” (see 2.2.19), “permaLock” (see 2.2.20), “kill” (see 2.2.21), “startDecode” (see 2.2.24).

■Sample code:

```
-(void)accessAsReaderP252BResult:(AsReaderP252BResultCode)error
                  actionState:(AsReaderP252BCommandType)action
                          epc:(NSString *)epc
                          data:(NSString *)data
                            rssi:(float)rssi
                          phase:(float)phase
                      frequency:(float)frequency{
   //error: Enum AsReaderP252BResultCode (see 8.4.1)
   //action: Enum AsReaderP252BCommandType (see 7.1.1)
   //epc: The EPC data of RFID tag. (Hex)
   //data: The read RFID tag data. (Hex)
   //rssi: The RSSI data of RFID tag.
   //phase: The phase value of RFID tag.
   //frequency: The frequency of RFID tag when it is inventoried.
}
```

3.6. readAsReaderP252BTemperatureTag

-(void)readAsReaderP252BTemperatureTag:(NSString*)tag
rssi:(float)rssi
phase:(float)phase
frequency:(float)frequency
temperature:(float)temperature;

	Names	In/Out	Types	Descriptions
Parameter	tag	Out	NSString	The PCEPC data of RFID tag. (Hex)
Parameter	rssi	Out	float	The RSSI data of RFID tag.
Parameter	phase	Out	float	The phase value of RFID tag.
Parameter	frequency	Out	float	The frequency of RFID tag when it is inventoried.
Parameter	temperature	Out	float	The temperature.

■Function description:
Receives the data of the RFID tag with a temperature chip.
The delegate function will be called back after calling the function "inventoryTemperatureTagWithType" (see 2.2.13).

■Sample code:

```
-(void)readAsReaderP252BTemperatureTag:(NSString*)tag
                                  rssi:(float)rssi
                                 phase:(float)phase
                             frequency:(float)frequency
                           temperature:(float)temperature{
   //tag: The PCEPC data of RFID tag. (Hex)
   //rssi: The RSSI data of RFID tag.
   //phase: The phase value of RFID tag.
   //frequency: The frequency of RFID tag when it is inventoried.
   //temperature: The temperature.
}
```

3.7. readAsReaderP252BHumidityTag

-(void)readAsReaderP252BHumidityTag:(NSString*)tag
rssi:(float)rssi
phase:(float)phase
frequency:(float)frequency
humidity:(float)humidity;

	Names	In/Out	Types	Descriptions
Parameter	tag	Out	NSString	The PCEPC data of RFID tag. (Hex)
Parameter	rssi	Out	float	The RSSI data of RFID tag.
Parameter	phase	Out	float	The phase value of RFID tag.
Parameter	frequency	Out	float	The frequency of RFID tag when it is inventoried.
Parameter	humidity	Out	float	The humidity.

■Function description:
Receives the data of the RFID tag with a humidity chip.
The delegate function will be called back after calling the function “inventoryHumidityTag” (see 2.2.12).

■Sample code:

```
-(void)readAsReaderP252BHumidityTag:(NSString*)tag
                                rssi:(float)rssi
                               phase:(float)phase
                           frequency:(float)frequency
                            humidity:(float)humidity{
   //tag: The PCEPC data of RFID tag. (Hex)
   //rssi: The RSSI data of RFID tag.
   //phase: The phase value of RFID tag.
   //frequency: The frequency of RFID tag when it is inventoried.
   //humidity: The humidity.
}
```

3.8. onAsReaderP252BTriggerKeyEvent

-(BOOL)onAsReaderP252BTriggerKeyEvent:(BOOL)status;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	status	Out	BOOL	The status of the Trigger button. YES: Press the Trigger button. NO: Release the Trigger button.
Return value	-	In	BOOL	YES: Execute the default actions of SDK. Press the Trigger button: Start inventory/start scanning. Release the Trigger button: Stop inventory/stop scanning. NO: Default actions of SDK are not executed.

■Function description:
Once the Trigger button of the AsReader device is pressed or released, this function will be called back.

■Sample code:
```
-(BOOL)onAsReaderP252BTriggerKeyEvent:(BOOL)status{
   //status: The status of the Trigger button.
             YES: Press the Trigger button.
             NO: Release the Trigger button.
}
```

3.9. onAsReaderP252BModeKeyEvent

- (BOOL)onAsReaderP252BModeKeyEvent:(BOOL)status;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	status	Out	BOOL	The status of the Mode button. YES: Press the Mode button. NO: Release the Mode button.
Return value	-	In	BOOL	YES: Switch the Barcode mode/RFID mode. NO: Keep the current mode.

■Function description:
Once the Mode button of the AsReader device is pressed or released, this function will be called back.

■Sample code:
```
- (BOOL)onAsReaderP252BModeKeyEvent:(BOOL)status{
   //status: The status of the Mode button.
             YES: Press the Mode button.
             NO: Release the Mode button.
}
```

3.10. whenAsReaderP252BReceivedData

- (void)whenAsReaderP252BReceivedData:(NSData *)receivedData;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	receivedData	Out	NSData	Command returned by the AsReader device.

■Function description:
Receives the command returned by the AsReader device.

■Sample code:
```
- (void)whenAsReaderP252BReceivedData:(NSData *)receivedData{
   //receivedData: Command returned by the AsReader device.
}
```

3.11. whenAsReaderP252BWriteData

- (void)whenAsReaderP252BWriteData:(NSData *)writeData;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	writeData	Out	NSData	The command sent to the AsReader device.

■Function description:
Receives the command sent to the AsReader device.

■Sample code:
```
- (void)whenAsReaderP252BWriteData:(NSData *)writeData{
   //writeData: The command sent to the AsReader device.
}
```

3.12. detectBarcode

-(void)detectBarcode:(AsReaderP252BBarcodeType)barcodeType barcodeData:(NSData *)barcodeData;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	barcodeType	Out	AsReaderP252BBarcodeType	The barcode type. Enum AsReaderP252BBarcodeType (see 4.2.1)
Parameter	barcodeData	Out	NSData	The scanned barcode data.

■Function description:
Receives scanned barcode data.
The delegate function will be called back after calling the function “startDecode” (see 2.2.24).

■Sample code:
```
-(void)detectBarcode:(AsReaderP252BBarcodeType)barcodeType
        barcodeData:(NSData *)barcodeData{
   //barcodeType: Enum AsReaderP252BBarcodeType (see 4.2.1)
   //barcodeData: The scanned barcode data.
}
```

4 AsReaderP252BBarcode

4.1. Functions

4.1.1. getBarcodeString

+ (NSString *)getBarcodeString:(AsReaderP252BBarcodeType)barcodeType;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	barcodeType	In	AsReaderP252BBarcodeType	Enum AsReaderP252BBarcodeType (see 4.2.1)
Return value	-	Out	NSString	The barcode types in AsReaderP252BBarcodeType.
■Function description: Gets the barcode type. **■Sample code:** NSString *barcodeType = [AsReaderP252BBarcode getBarcodeString:AsReaderP252BBarcodeTypeCode39];				

4.2. Enum

4.2.1. AsReaderP252BBarcodeType

Definitions	Descriptions
AsReaderP252BBarcodeTypeNoRead	UNKNOWNTYPE.
AsReaderP252BBarcodeTypeCode39	Code39.
AsReaderP252BBarcodeTypeCode11	Code11.
AsReaderP252BBarcodeTypeCodabar	Codabar.
AsReaderP252BBarcodeTypeEAN13	EAN-13.
AsReaderP252BBarcodeTypeCode128	Code128.
AsReaderP252BBarcodeTypeEAN13With2Supps	EAN-13 with 2 Supps.
AsReaderP252BBarcodeTypeIndustrial2Of5	Industrial 2 of 5.
AsReaderP252BBarcodeTypeEAN13With5Supps	EAN-13 with 5 Supps.
AsReaderP252BBarcodeTypeIATA2Of5	IATA 2 of 5.
AsReaderP252BBarcodeTypeMSI	MSI.
AsReaderP252BBarcodeTypeInterleaved2Of5	Interleaved 2 of 5.
AsReaderP252BBarcodeTypeEAN128	EAN-128.
AsReaderP252BBarcodeTypeCode93	Code93.
AsReaderP252BBarcodeTypeUPCE1	UPC-E1.
AsReaderP252BBarcodeTypeUPCA	UPC-A.
AsReaderP252BBarcodeTypeUPCE1With2Supps	UPC-E1 with 2 Supps.
AsReaderP252BBarcodeTypeUPCAWith2Supps	UPC-A with 2 Supps.
AsReaderP252BBarcodeTypeUPCAWith5Supps	UPC-A with 5 Supps.
AsReaderP252BBarcodeTypeUPCE1With5Supps	UPC-E1 with 5 Supps.
AsReaderP252BBarcodeTypeTriopticCode39	Trioptic Code39.
AsReaderP252BBarcodeTypeUPCE0	UPC-E.
AsReaderP252BBarcodeTypeBooklandEAN	Bookland EAN.
AsReaderP252BBarcodeTypeUPCE0With2Supps	UPC-E with 2 Supps.
AsReaderP252BBarcodeTypeCouponCode	Coupon Code.

AsReaderP252BBarcodeTypeUPCE0With5Supps	UPC-E with 5 Supps.
AsReaderP252BBarcodeTypeGS1DataBarLimitedRSSLimited	GS1 DataBar Limited (RSS-Limited).
AsReaderP252BBarcodeTypeEAN8	EAN-8.
AsReaderP252BBarcodeTypeGS1DataBarRSS14	GS1 DataBar (RSS-14).
AsReaderP252BBarcodeTypeEAN8With2Supps	EAN-8 with 2 Supps.
AsReaderP252BBarcodeTypeGS1DataBarExpandedRSSExpanded	GS1 DataBar Expanded (RSS-Expanded).
AsReaderP252BBarcodeTypeEAN8With5Supps	EAN-8 with 5 Supps.
AsReaderP252BBarcodeTypeMatrix2Of5	Matrix 2 of 5.
AsReaderP252BBarcodeTypeChinaPostChinese2Of5	China Post (Chinese 2 of 5).
AsReaderP252BBarcodeTypeCode32	Code32.
AsReaderP252BBarcodeTypeUKPlessey	UK Plessey.
AsReaderP252BBarcodeTypeISBT128	ISBT128.
AsReaderP252BBarcodeTypePDF417	PDF417.
AsReaderP252BBarcodeTypeAztec	Aztec.
AsReaderP252BBarcodeTypeMicroPDF417	MicroPDF417.
AsReaderP252BBarcodeTypeQR	QR.
AsReaderP252BBarcodeTypeDataMatrix	DataMatrix.
AsReaderP252BBarcodeTypeMicroQR	Micro QR.
AsReaderP252BBarcodeTypeHanXinCode	HanXin Code.
AsReaderP252BBarcodeTypeMaxicode	Maxicode.
AsReaderP252BBarcodeTypeITF14	ITF-14.
AsReaderP252BBarcodeTypeITF6	ITF-6.
AsReaderP252BBarcodeTypeAIM128	AIM 128.
AsReaderP252BBarcodeTypeISSN	ISSN.
AsReaderP252BBarcodeTypeISBN	ISBN.
AsReaderP252BBarcodeTypeGS1Databar	GS1-Databar.

5 AsReaderP252BDeviceManager

5.1. Properties

Property names	Properties	Types	Descriptions
delegate	nonatomic, weak	AsReaderP252BDeviceManagerDelegate	AsReaderP252BDeviceManagerDelegate (see 5.3)

5.2. Functions

5.2.1. shared

+ (AsReaderP252BDeviceManager *)shared;				
	Names	**In/Out**	**Types**	**Descriptions**
Return value	-	Out	AsReaderP252BDeviceManager	The object of AsReaderP252BDeviceManager. (see 5)
■Function description: Create and initialize the object of AsReaderP252BDeviceManager (singleton mode). **■Sample code:** AsReaderP252BDeviceManager *asReaderP252BDeviceManager= [AsReaderP252BDeviceManager shared];				

5.2.2. startDiscovered

- (void)startDiscovered;
■Function description: Starts to search for AsReader devices via Bluetooth. Once this function is executed, the delegate function “didDiscoverAsReaderP252BDeviceList” (see 5.3.1) will be called back. **■Sample code:** (Note: asReaderP252BDeviceManager is the object of AsReaderP252BDeviceManager.) [asReaderP252BDeviceManager startDiscovered];

5.2.3. stopDiscovered

- (void)stopDiscovered;
■Function description: Stops to search for AsReader devices via Bluetooth. **■Sample code:** (Note: asReaderP252BDeviceManager is the object of AsReaderP252BDeviceManager.) [asReaderP252BDeviceManager stopDiscovered];

5.3. Delegate

5.3.1. didDiscoverAsReaderP252BDeviceList

-(void)didDiscoverAsReaderP252BDeviceList:(NSArray<AsReaderP252B*> *)asReaderP252BList;				
	Names	**In/Out**	**Types**	**Descriptions**
Parameter	asReaderP252B List	Out	NSArray	The list of the AsReader devices found via Bluetooth. (The array elements are objects of AsReaderP252B.)

■Function description:
Receives the array of the AsReader devices found via Bluetooth. (The function will be called back multiple times.)
The delegate function will be called back after calling the function “startDiscovered” (see 5.2.2).
 	
■Sample code:
- (void)didDiscoverAsReaderP252BDeviceList:(NSArray<AsReaderP252B *> *)asReaderP252BList{
  // asReaderP252BList: The array elements are objects of AsReaderP252B.
}				

6 AsReaderP252BDevice

6.1. Properties

Property names	Properties	Types	Descriptions
currentAccessory	nonatomic, strong, readonly	EAAccessory	Gets the current object of EAAccessory.
deviceName	nonatomic, strong, readonly	NSString	Gets the AsReader device name.
isConnected	nonatomic, assign, readonly	BOOL	Gets the connection status of the AsReader device. YES: Connected. NO: Disconnected.

6.2. Functions

6.2.1. connect

- (void)connect;
■**Function description:** Connects to the AsReader device. Once this function is executed, the delegate function “whenAsReaderP252BIsConnected” (see 3.1) will be called back. ■**Sample code:** (Note: asReaderP252B is the object of AsReaderP252B.) [asReaderP252B connect];

6.2.2. disconnect

- (void)disconnect;
■**Function description:** Disconnects from the AsReader device. Once this function is executed, the delegate function “whenAsReaderP252BIsConnected” (see 3.1) will be called back. ■**Sample code:** (Note: asReaderP252B is the object of AsReaderP252B.) [asReaderP252B disconnect];

7 AsReaderP252BPacket

7.1. Enum

7.1.1. AsReaderP252BCommandType

Definitions	Descriptions
AsReaderP252BCommandInventory	Inventory in progress.
AsReaderP252BCommandReadMemory	Read Memory in progress.
AsReaderP252BCommandWriteMemory	Write Memory in progress.
AsReaderP252BCommandKill	Kill Tag in progress.
AsReaderP252BCommandLock	Lock in progress.
AsReaderP252BCommandUnlock	Unlock in progress.
AsReaderP252BCommandPermaLock	Perma Lock in progress.
AsReaderP252BCommandStop	Operation Stopped.
AsReaderP252BCommandDefaultParam	Default Param in progress.
AsReaderP252BCommandSaveParam	Save Param in progress.
AsReaderP252BCommandBuzzerStart	Start Buzzer in progress.
AsReaderP252BCommandDecodeStart	Scan Barcode in progress.

7.1.2. AsReaderP252BScanMode

Definitions	Descriptions
AsReaderP252BRFIDScanMode	RFID mode.
AsReaderP252BBarcodeScanMode	Barcode mode.

8 AsReaderP252BRfidValues

8.1. AsReaderP252BLockParam

8.1.1. Properties

Property names	Properties	Types	Descriptions
killPassword	nonatomic	BOOL	YES: Lock the kill password. NO: Do not lock the kill password.
accessPassword	nonatomic	BOOL	YES: Lock the access password. NO: Do not lock the access password.
epc	nonatomic	BOOL	YES: Lock the EPC bank. NO: Do not lock the EPC bank.
tid	nonatomic	BOOL	YES: Lock the TID bank. NO: Do not lock the TID bank.
user	nonatomic	BOOL	YES: Lock the User bank. NO: Do not lock the User bank.

8.2. AsReaderP252BSelectMaskEPCParam

8.2.1. Properties

Property names	Properties	Types	Descriptions
offset	nonatomic	int	The start address of the mask.
length	nonatomic	int	The length of the mask.
mask	nonatomic, strong	NSString	The mask data. (Hex)

8.3. AsReaderP252BLbtItem

8.3.1. Properties

Property names	Properties	Types	Descriptions
mSlot	nonatomic	int	The frequency position of the LBT frequency list.
mIsUsed	mIsUsed	BOOL	Whether to use the frequency specified in the LBT frequency list. YES: Use. NO: Do not use.
frequency	nonatomic, strong	NSString	The frequency of the LBT frequency list.

8.4. Enum

8.4.1. AsReaderP252BResultCode

Definitions	Descriptions
AsReaderP252BResultNoError	Succeed in result.
AsReaderP252BResultOtherError	An error has occurred due to unknown reason.
AsReaderP252BResultNotConnected	Not connected to Device.
AsReaderP252BResultInvalidParameter	Invalid Parameter transmitted.
AsReaderP252BResultUndefined	An undefined error.
AsReaderP252BResultMemoryOverrun	Accessing to memory out of range.
AsReaderP252BResultMemoryLocked	Memory is locked.
AsReaderP252BResultInsufficientPower	Battery power is low.
AsReaderP252BResultHandleMismatch	Handle mismatch.
AsReaderP252BResultCRCError	CRC error on tag response.
AsReaderP252BResultCommandFormatError	Command format error.
AsReaderP252BResultOutOfRetries	Access password error, kill password error, and other errors.
AsReaderP252BResultTimeout	Exceeded allowed accessing time.

8.4.2. AsReaderP252BMemoryBank

Definitions	Descriptions
AsReaderP252BBank_Reserved	Refers to Reserved memory Bank.
AsReaderP252BBank_EPC	Refers to EPC memory Bank.
AsReaderP252BBank_TID	Refers to TID memory Bank.
AsReaderP252BBank_User	Refers to User memory Bank.

8.4.3. AsReaderP252BTemperatureTagType

Definitions	Descriptions
AsReaderP252BTemperatureTagType_0	The type of temperature tag is Magnus-S3.

8.4.4. AsReaderP252BBuzzerState

Definitions	Descriptions
AsReaderP252BBuzzer_Off	Turn off the buzzer.
AsReaderP252BBuzzer_Low	Buzzer low.
AsReaderP252B3Buzzer_High	Buzzer high.

8.4.5. AsReaderP252BSessionType

Definitions	Descriptions
AsReaderP252BSession_S0	inventoried S0
AsReaderP252BSession_S1	inventoried S1
AsReaderP252BSession_S2	inventoried S2
AsReaderP252BSession_S3	inventoried S3

8.4.6. AsReaderP252BSessionFlag

Definitions	Descriptions
AsReaderP252BSessionFlag_A	A only
AsReaderP252BSessionFlag_B	B only
AsReaderP252BSessionFlag_AB	A or B

8.5. Structure

8.5.1. AsReaderP252BMinMaxValue

Definitions	Types	Descriptions
min	int	The minimum value of the power.
max	int	The maximum value of the power.

AsReaderP252B SDK

Objective-C SDK Reference Guide

Mar. 2024 1st Edition

AsReader Inc.

111 SW 5th Ave., Ste 3150

Portland, OR 97204-3656 U.S.A.

Tel.: (503) 770-2777 x102

Asterisk Inc.

AsTech Osaka Building 5F,

2-2-1, Kikawa-nishi,

Yodogawa-ku, Osaka, 532-0013, JAPAN