AsReader®

AsReader Wired BOX-Type

Objective-C SDK Reference Guide

Asterisk Inc.

Revision History

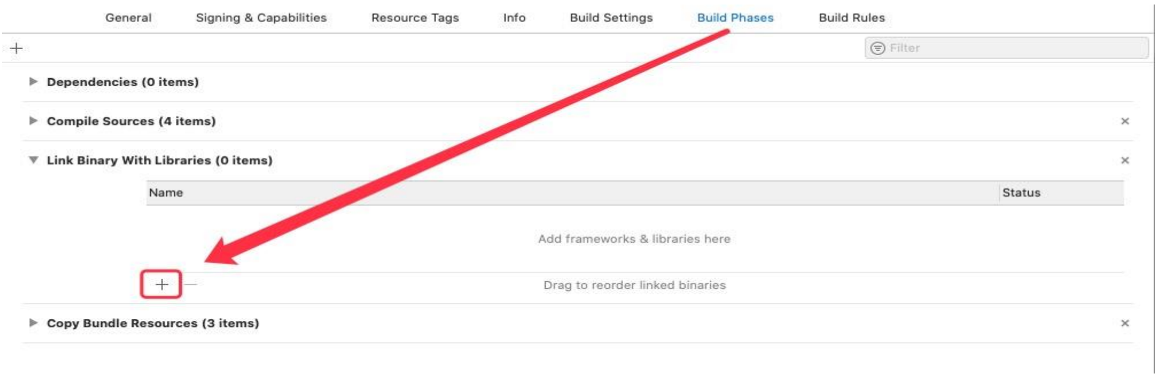
Ver.	Description	Revised by	Date
1.0	Initial version	Trinity Liu	2021/04/29
1.1	1. Adds property isShowPrintNSLog 2. Modifies function setRecharging 3. Adds function getRechargedStatus 4. Adds function tX 5. Adds function getRFIDModuleVersion 6. Modifies delegate name readTimeForMultiAntennaReceivedAnt1 7. Modifies delegate name didSetReadTimeForMultiAntenna 8. Adds delegate setRecharging 9. Adds delegate rfidModuleVersionReceived 10. Adds delegate receivedRechargedStatus 11. Modifies Enum AsReaderWiredBOXTypeSDKBOXNetworkState Adds Enum AsReaderWiredBOXTypeSDKResultSetRecharging	Trinity Liu	2022/4/6
1.2	1.Adds property connectMethod 2.Adds enum AsReaderWiredBOXTypeConnectMethod	Trinity Liu	2022/5/16
1.3	Modifies the parameter type of the function setFHLBTParamReadTime: the rfLevel parameter type is changed from uint16_t to int	Trinity Liu	2022/5/23

Contents

1. SDK Usage

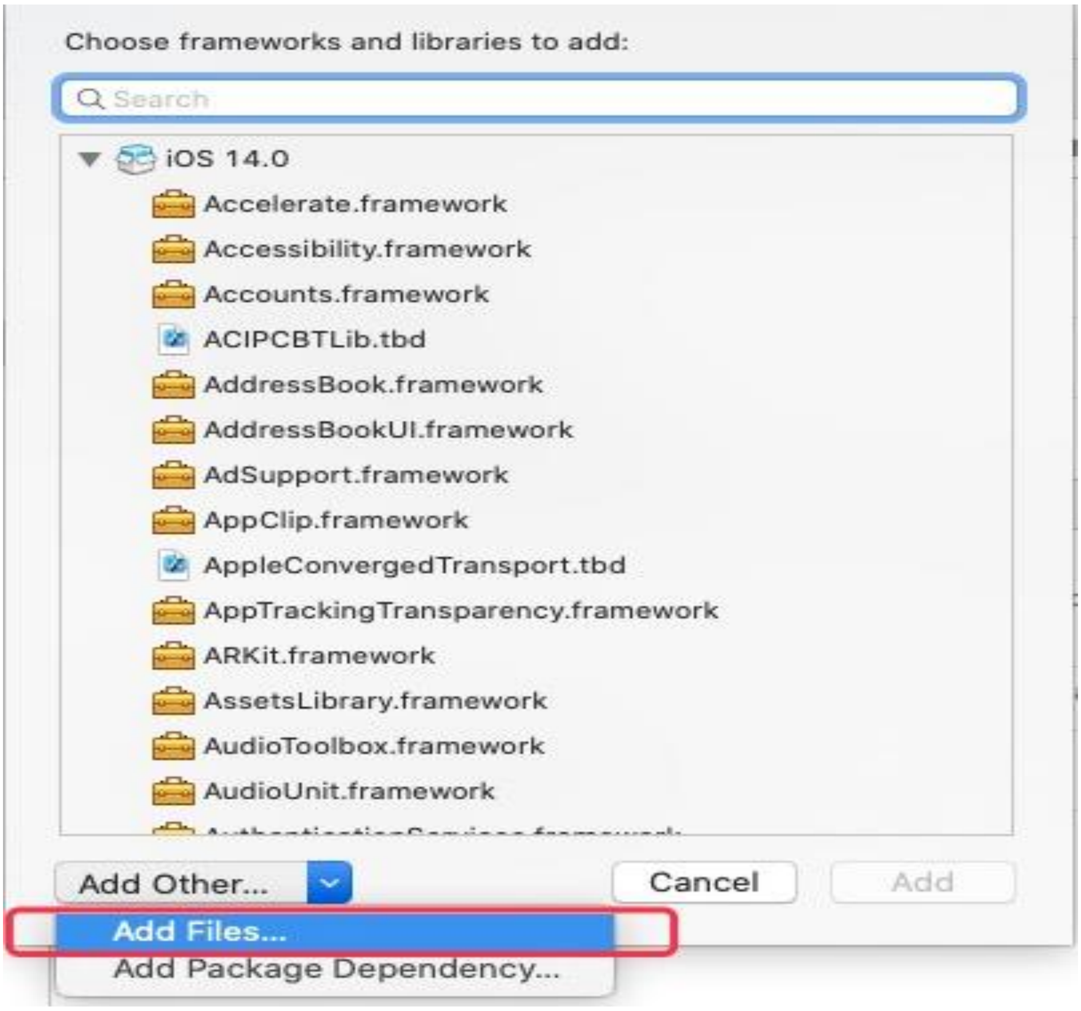
1.1. Add SDK

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

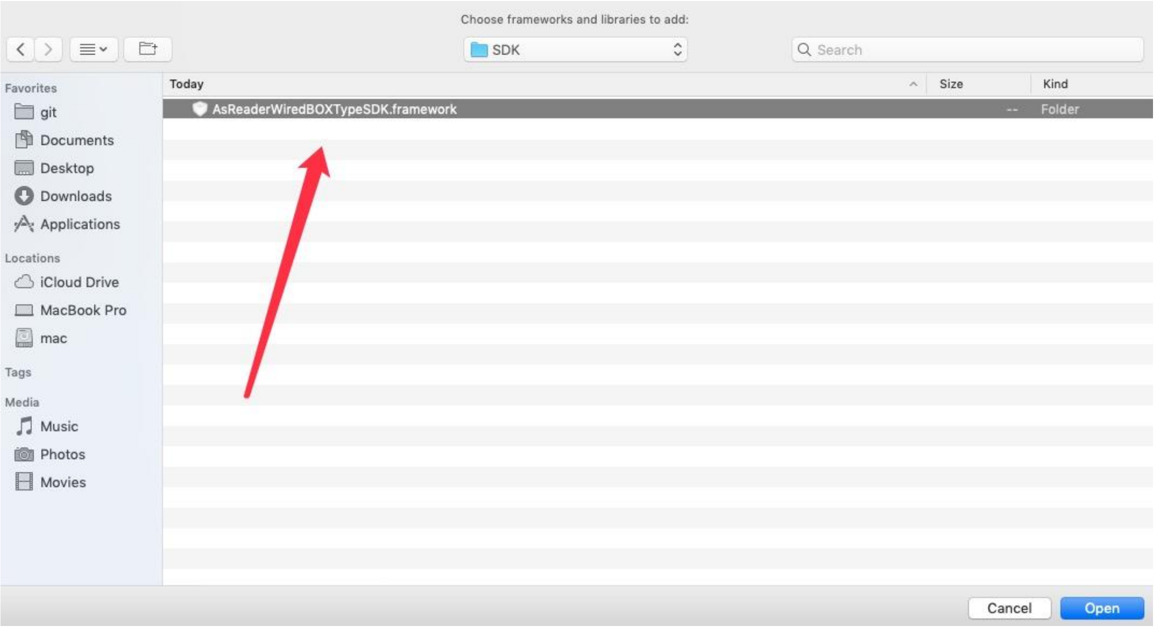


General
Signing & Capabilities
Resource Tags
Info
Build Settings
Build Phases
Build Rules
+
Filter
Dependencies (0 items)
Compile Sources (4 items)
Link Binary With Libraries (0 items)
Name
Status
Add frameworks & libraries here
+ —
Drag to reorder linked binaries
Copy Bundle Resources (3 items)

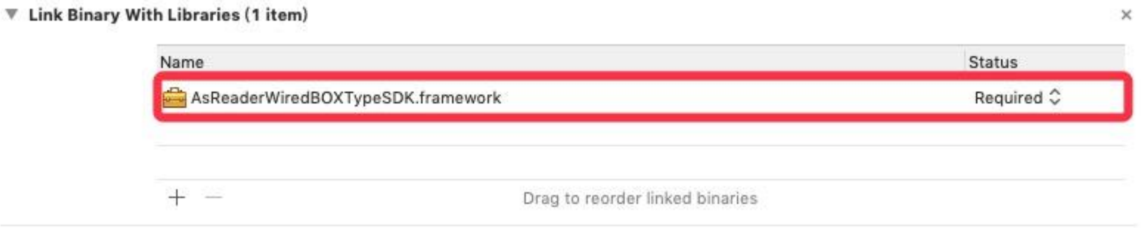
2) Select “Add Other…”, “Add Files…”



3) Add AsReaderWiredBOXTypeSDK.framework



4) SDK is added.


Link Binary With Libraries (1 item)
Name
Status
AsReaderWiredBOXTypeSDK.framework
Required
+ —
Drag to reorder linked binaries

1.2. Import Header Files

Objective-C projects need to import header files in classes that use the SDK. The following is a reference.

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

1.3. SDK Usage

1.3.1. create and initialize an AsReaderWiredBOXType object (singleton pattern).

```
AsReaderWiredBOXType*asReaderWiredBOXType = [AsReaderWiredBOXType
sharedInstance];
```

1.3.2. Set delegates

```
asReaderWiredBOXType.delegate = self;
```

1.3.3. Start to search

```
[asReaderWiredBOXType startSearchDevice];
```

Receive the delegate “searchingIP” that retrieved the IP (2.3.1)

1.3.4. Connect to the AsReaderWiredBOXType object.

Connect through the IP address obtained by this delegate "searchingIP” (see 2.3.1)

```
[asReaderWiredBOXType connectServer:@”192.168.1.10”port:9600];
```

The result of a change in device connection status is notified by delegate “networkStateChanged" (see 2.3.2).

1.3.5. Start to receive RFID data

```
[asReaderWiredBOXType startReceiveRFIDData];
```

1.3.6. Start to inventory RFID tags

```
[asReaderWiredBOXType startInventoryWithTagsMaxTagCount:0
                                                 readTime:0
                                              repeatCycle:0
                                            enableAntenna1:YES
                                            enableAntenna2:YES
                                            enableAntenna3:YES
                                            enableAntenna4:YES
                                            enableAntenna5:YES
                                            enableAntenna6:YES
                                            enableAntenna7:YES
                                            enableAntenna8:YES];
```

Receive the result returned from the delegate “inventoryStartedWithStatus” (see 2.3.4) that returns the result of the execution of the event "Start Inventory".
Receive the results which returned from the delegate “receivedPCEPCData” (see 2.3.6) that can return the RFID data read by the device.

1.3.7. Stop taking inventory of RFID tags

```
[asReaderWiredBOXType stopInventory];
```

Receive the result returned from the delegate “inventoryStoppedWithStatus” (see 2.3.5) that returns the result of the execution of the event "Stop Inventory".

1.3.8. Disconnect from the AsReaderWiredBOXType object.

```
[asReaderWiredBOXType disconnectServer];
```

The result of a change in device connection status is notified by delegate “networkStateChanged" (see 2.3.2).

2. AsReaderWiredBOXType Class

ASReaderWiredBOXType Object has the functions of searching/connecting/disconnecting Wired BOX-Type devices, inventorying RFID tags, stopping inventory, setting inventory related parameters, receiving HID data from USB connected devices, etc.

2.1. Properties

Property Name	Property	Type	Descriptions
host	nonatomic, strong, readonly	NSString	IP address.
delegate	nonatomic, assign	AsReaderWiredBOXTypeDelegate	Delegate (see 2.3).
isConnected	nonatomic, assign, readonly	BOOL	Connection states.
isShowPrintNSLog	nonatomic, assign	BOOL	Whether to print Log
connectMethod	nonatomic, assign	AsReaderWiredBOXTypeConnectMethod	AsReaderWiredBOXTypeConnectMethod Enum (See 4.1.12)

2.2. Functions

2.2.1. sharedInstance

+ (AsReaderWiredBOXType *) sharedInstance;				
	Parameter Name	**In/Out**	**Type**	**Descriptions**
Return Values	-	Out	AsReaderWiredBOXType	AsReaderWiredBOXType object.

■ **Function Descriptions:**
Create and initialize an AsReaderWiredBOXType object (singleton pattern).
■ **Sample Code**:

```
AsReaderWiredBOXType *asReaderWiredBOXType = [AsReaderWiredBOXType
sharedInstance];
```

2.2.2. getAsWiredBOXPrinter

- (AsWiredBOXPrinter *) getAsWiredBOXPrinter;				
	Parameter Name	**In/Out**	**Type**	**Descriptions**
Return Values	-	Out	AsWiredBOXPrinter	AsWiredBOXPrinter object.

■ **Function Descriptions:**

Creates AsWiredBOXPrinter object.

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
AsWiredBOXPrinter *asPrinter = [asReaderWiredBOXType getAsWiredBOXPrinter];
```

2.2.3. startSearchDevice

- (BOOL) startSearchDevice;				
	Parameter Name	**In/Out**	**Type**	**Descriptions**
Return Values	-	Out	BOOL	YES: The command was sent successfully. NO: The command failed to be sent.

■ **Function Descriptions:**

Searches for Wired BOX-Type devices in the same network segment.

After this function is executed, call back the “searchingIP” (see 2.3.1), “searchedUPDSearchingDeviceInfo” (see 2.3.3) delegate

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
BOOL status = [asReaderWiredBOXType startSearchDevice];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.4. connectServer

- (BOOL) connectServer:(NSString*) ip port:(int) port;				
	Parameter Name	**In/Out**	**Type**	**Descriptions**
Parameter	ip	In	NSString	IP address.
Parameter	port	In	int	Port number. (Default port: 9600)

Return Values	-	Out	BOOL	YES: The command was sent successfully. NO: The command failed to be sent.

■ **Function Descriptions:**

Specify the TCP port and IP and connect to the device.

After this function is executed, the connection state of the device is invoked by the delegate function "networkStateChanged" (see 2.3.2).

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
BOOL status = [asReaderWiredBOXType connectServer:@”192.168.1.10” port:9600];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.5. disconnectServer

- (BOOL) disconnectServer;

	Parameter Name	In/Out	Type	Descriptions
Return Values	-	Out	BOOL	YES: The command was sent successfully. NO: The command failed to be sent.

■ **Function Descriptions:**

Disconnect from the Wired BOX-Type device.

After this function is executed, the disconnected state is invoked by the delegate function "networkStateChanged" “networkStateChanged” (see 2.3.2).

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
BOOL status = [asReaderWiredBOXType disconnectServer];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.6. startReceiveHIDData

- (BOOL) startReceiveHIDData;

	Parameter Name	In/Out	Type	Descriptions
Return	-	Out	BOOL	YES: The command was sent successfully.

Values				NO: The command failed to be sent.

■ **Function Descriptions:**

Starts to receive data from the HID device connected by USB and Wired BOX-Type device, so that the App can communicate with the HID module. If a Wired BOX-Type device is connected to a HID device, the App can receive data from the HID device.

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
BOOL status = [asReaderWiredBOXType startReceiveHIDData];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.7. stopReceiveHIDData

- (BOOL) stopReceiveHIDData;

	Parameter Name	In/Out	Type	Descriptions
Return Values	-	Out	BOOL	YES: The command was sent successfully. NO: The command failed to be sent.

■ **Function Descriptions:**

Stops receiving HID data, so that the application can no longer communicate with the HID module, and the application can no longer receive the data from the HID input device.

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
BOOL status = [asReaderWiredBOXType stopReceiveHIDData];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.8. startReceiveRFIDData

- (BOOL) startReceiveRFIDData;

	Parameter Name	In/Out	Type	Descriptions
Return Values	-	Out	BOOL	YES: The command was sent successfully. NO: The command failed to be sent.

■ **Function Descriptions:**

Starts receiving or sending RFID data. The App can receive data from the RFID module inside the Wired BOX-Type device.

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
BOOL status = [asReaderWiredBOXType startReceiveRFIDData];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.9. stopReceiveRFIDData

- (BOOL) stopReceiveRFIDData;

	Parameter Name	In/Out	Type	Descriptions
Return Values	-	Out	BOOL	YES: The command was sent successfully. NO: The command failed to be sent.

■ **Function Descriptions:**

Stops receiving/sending RFID data. The App can not receive data from the RFID module inside the Wired BOX-Type device.

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
BOOL status [asReaderWiredBOXType stopReceiveRFIDData];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.10. startInventoryWithTagsMaxTagCount

```
- (BOOL) startInventoryWithTagsMaxTagCount:(uint8_t) maxTagCount
                                  readTime:(uint8_t) readTime
                               repeatCycle:(uint16_t) repeatCycle
                            enableAntenna1:(BOOL) antenna1
                            enableAntenna2:(BOOL) antenna2
                            enableAntenna3:(BOOL) antenna3
                            enableAntenna4:(BOOL) antenna4
                            enableAntenna5:(BOOL) antenna5
                            enableAntenna6:(BOOL) antenna6
                            enableAntenna7:(BOOL) antenna7
                            enableAntenna8:(BOOL) antenna8;
```

	Parameter Name	In/Out	Type	Descriptions

Parameter	maxTagCount	In	uint8_t	The maximum number of RFID tags that can be inventory in one time inventory.
Parameter	readTime	In	uint8_t	The maximum amount of time one time inventory can last.
Parameter	repeatCycle	In	uint16_t	The maximum number of query cycles for one time inventory.
Parameter	antenna1	In	BOOL	Sets whether to use antenna 1. YES: use. NO: not use.
Parameter	antenna2	In	BOOL	Sets whether to use antenna 2. YES: use. NO: not use.
Parameter	antenna3	In	BOOL	Sets whether to use antenna 3. YES: use. NO: not use.
Parameter	antenna4	In	BOOL	Sets whether to use antenna 4. YES: use. NO: not use.
Parameter	antenna5	In	BOOL	Sets whether to use antenna 5. YES: use. NO: not use.
Parameter	antenna6	In	BOOL	Sets whether to use antenna 6. YES: use. NO: not use.
Parameter	antenna7	In	BOOL	Sets whether to use antenna 7. YES: use. NO: not use.
Parameter	antenna8	In	BOOL	Sets whether to use antenna 8. YES: use. NO: not use.
Return Values	-	Out	BOOL	YES: The command was sent successfully. NO: The command failed to be sent.

■ **Function Descriptions:**

Takes inventory of RFID tags.

After this function is executed, call back the “inventoryStartedWithStatus” (see 2.3.4), “receivedPCEPCData” (see 2.3.6) delegate.

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
BOOL status = [asReaderWiredBOXType startInventoryWithTagsMaxTagCount:0
                                                          readTime:0
                                                       repeatCycle:0
                                                   enableAntenna1:YES
                                                   enableAntenna2:YES
                                                   enableAntenna3:YES
                                                   enableAntenna4:YES
                                                   enableAntenna5:YES
                                                   enableAntenna6:YES
                                                   enableAntenna7:YES
                                                   enableAntenna8:YES];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.11. stopInventory

- (BOOL) stopInventory;				
	Parameter Name	**In/Out**	**Type**	**Descriptions**
Return Values	-	Out	BOOL	YES: The command was sent successfully. NO: The command failed to be sent.

■ **Function Descriptions:**

Stops inventory.

After this function is executed, call back the “inventoryStoppedWithStatus” (see 2.3.5) delegate.

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
BOOL status = [asReaderWiredBOXType stopInventory];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.12. updateRegistry

- (BOOL) updateRegistry;				
	Parameter Name	**In/Out**	**Type**	**Descriptions**

Return Values	-	Out	BOOL	YES: The command was sent successfully. NO: The command failed to be sent.

■ **Function Descriptions:**

Use it when you want to permanently save your RFID settings after changing them.

After this function is executed, call back the “updatedRegistryStatus” (see 2.3.7) delegate.

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
BOOL status = [asReaderWiredBOXType updateRegistry];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.13. setRecharging

- (ASREADERWIREDBOXTYPESDKRESULTSETRECHARGING)SETRECHARGING:(BOOL)STATUS;

	Parameter Name	In/Out	Type	Descriptions
Parameter	status	In	BOOL	YES: Charging the phone. NO: not charging the phone.
Return Values	-	Out	AsReaderWiredBOXTypeSDKResultSetRecharging	AsReaderWiredBOXTypeSDKResultSetRecharging, Enum (See 4.1.11)

■ **Function Descriptions**:

Sets whether to charge the connected phone while charging Wired BOX-Type.

Once this function is executed, the delegate "setRecharging" (see 2.3.19) is called back.

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
AsReaderWiredBOXTypeSDKResultSetRecharging status = [asReaderWiredBOXType
setRecharging:YES];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.14. getRechargedStatus

- (BOOL)getRechargedStatus;

	Para meter Name	**In/Out**	**Type**	**Descriptions**
Return Values	-	Out	BOOL	YES: Get charging status successfully NO: Get charging status failed

Function Descriptions:
Gets the charging status of the phone which connected with the Wired BOX-Type device.
Once this function is executed, the delegate "receivedRechargedStatus" (see 2.3.21) is called back.
Sample Code: (Note: asReaderWiredBOXType is an instance of AsReaderWiredBOXType class.)

```
BOOL status = [asReaderWiredBOXType getRechargedStatus];
if (status) {
// Gets charging status successfully.
} else {
// Gets charging status failed.
}
```

2.2.15. getSession

- (BOOL) getSession;				
	Parameter Name	**In/Out**	**Type**	**Descriptions**
Return Values	-	Out	BOOL	YES: The command was sent successfully. NO: The command failed to be sent.

■ **Function Descriptions**:
To get the session value in the RFID settings.
After this function is executed, call back the “receivedSession” (see 2.3.9) delegate.
■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
BOOL status = [asReaderWiredBOXType getSession];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.16. setSession

- (BOOL) setSession:(uint8_t) session;				
	Parameter Name	**In/Out**	**Type**	**Descriptions**

Parameter	session	In	uint8_t	S0 (0x00) 、S1 (0x01) 、S2 (0x02) 、S3 (0x03)
Return Values	-	Out	BOOL	YES: The command was sent successfully. NO: The command failed to be sent.

■ **Function Descriptions:**

Sets the session value in the RFID settings.

After this function is executed, call back the “didSetSessionStatus” (see 2.3.10) delegate.

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
BOOL status = [asReaderWiredBOXType setSession:0];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.17. getOutputPowerLevelForMultiAntenna

- (BOOL) getOutputPowerLevelForMultiAntenna;

	Parameter Name	In/Out	Type	Descriptions
Return Values	-	Out	BOOL	YES: The command was sent successfully. NO: The command failed to be sent.

■ **Function Descriptions:**

Gets the output power of each antenna in the RFID settings.

After this function is executed, call back the “receivedOuputPowerAntenna1” (see 2.3.11) delegate.

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
BOOL status = [asReaderWiredBOXType getOutputPowerLevelForMultiAntenna];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.18. setOutputPowerLevelAntenna1

```
-(BOOL) setOutputPowerLevelAntenna1:(uint16_t) powerLevel1
                          Antenna2:(uint16_t) powerLevel2
                          Antenna3:(uint16_t) powerLevel3
                          Antenna4:(uint16_t) powerLevel4
```

Antenna5:(uint16_t) powerLevel5
Antenna6:(uint16_t) powerLevel6
Antenna7:(uint16_t) powerLevel7
Antenna8:(uint16_t) powerLevel8;

	Parameter Name	In/Out	Type	Descriptions
Parameter	powerLevel1	In	uint16_t	The output power of antenna 1.
Parameter	powerLevel2	In	uint16_t	The output power of antenna 2.
Parameter	powerLevel3	In	uint16_t	The output power of antenna 3.
Parameter	powerLevel4	In	uint16_t	The output power of antenna 4.
Parameter	powerLevel5	In	uint16_t	The output power of antenna 5.
Parameter	powerLevel6	In	uint16_t	The output power of antenna 6.
Parameter	powerLevel7	In	uint16_t	The output power of antenna 7.
Parameter	powerLevel8	In	uint16_t	The output power of antenna 8.
Return Values	-	Out	BOOL	YES: The command was sent successfully. NO: The command failed to be sent.

■ **Function Descriptions:**

Sets the output power of each antenna in the RFID settings.

After this function is executed, call back the “didSetOutputPowerForMultiAntenna” (see 2.3.12) delegate

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
BOOL status = [asReaderWiredBOXType setOutputPowerLevelAntenna1:10
                                                       Antenna2:10
                                                       Antenna3:10
                                                       Antenna4:10
                                                       Antenna5:10
                                                       Antenna6:10
                                                       Antenna7:10
                                                       Antenna8:10];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.19. getOutputPowerLevel

- (BOOL) getOutputPowerLevel;

	Parameter Name	In/Out	Type	Descriptions
Return Values	-	Out	BOOL	YES: The command was sent successfully. NO: The command failed to be sent.

■ **Function Descriptions**:
Gets the common output value of each antenna in the current RFID settings, and the maximum and minimum output power that can be set.
After this function is executed, call back the “receivedOutputPowerLevel” (see 2.3.13) delegate
■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)
BOOL status = [asReaderWiredBOXType getOutputPowerLevel];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}				

2.2.20. setOutputPowerLevel

- (BOOL) setOutputPowerLevel:(uint16_t) power;				
	Parameter Name	**In/Out**	**Type**	**Descriptions**
Parameter	power	In	uint16_t	Output power (output power range of the Japan version: 18 ~ 23dbm, output power range of the non-Japan version: 18 ~ 27dbm).
Return Values	-	Out	BOOL	YES: The command was sent successfully. NO: The command failed to be sent.

■ **Function Descriptions**:
Sets the common output value of each antenna in the RFID settings
After this function is executed, call back the “didSetOutputPowerLevelStatus” (see 2.3.14) delegate
■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)
BOOL status = [asReaderWiredBOXType setOutputPowerLevel:20];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}				

2.2.21. getFHLBTParam

- (BOOL) getFHLBTParam;				
	Parameter Name	**In/Out**	**Type**	**Descriptions**
Return Values	-	Out	BOOL	YES: The command was sent successfully. NO: The command failed to be sent.

■ **Function Descriptions**:

Gets the values of FH and LBT in the RFID settings.

After this function is executed, call back the “receivedFHLBTData” (see 2.3.15) delegate

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
BOOL status = [asReaderWiredBOXType getFHLBTParam];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.22. setFHLBTParamReadTime

- (BOOL) setFHLBTParamReadTime:(uint16_t) readTime
idleTime:(uint16_t) idleTime
carrierSenseTime:(uint16_t) carrierSenseTime
rfLevel:(uint16_t) rfLevel
frequencyHopping:(uint8_t) frequencyHopping
listenBeforeTalk:(uint8_t) listenBeforeTalk
continuousWave:(uint8_t) continuousWave;

	Parameter Name	**In/Out**	**Type**	**Descriptions**
Parameter	readTime	In	uint16_t	the reading time (Unit: ms)
Parameter	idleTime	In	uint16_t	the idle time (Unit: ms).
Parameter	carrierSenseTime	In	uint16_t	Carrier listening time, fixed value: 50.
Parameter	rfLevel	In	int	Carrier listening RF level, fixed value: -740.
Parameter	frequencyHopping	In	uint8_t	Enable FH (0x01 or above); disabled FH (0x00).
Parameter	listenBeforeTalk	In	uint8_t	Enable LBT (0x01 or above); disabled LBT (0x00).
Parameter	continuousWave	In	uint8_t	CW, fixed value:0x00
Return Values	-	Out	BOOL	YES: The command was sent successfully.

				NO: The command failed to be sent.

■ **Function Descriptions**:

Sets the parameters of FH and LBT in the RFID settings.

After this function is executed, call back the "didSetFHLBTStatus" (see 2.3.16) delegate

Note: To set up the Hopping is enabled, you need to set up ParameterfrequencyHopping as 0x02, Parameter listenBeforeTalk as 0x01;

To set up the Hopping disabled, you need to set up ParameterfrequencyHopping as 0x01, ParameterlistenBeforeTalk as 0x02.

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
BOOL status = [asReaderWiredBOXType setFHLBTParamReadTime:10
                                                  idleTime:10
                                          carrierSenseTime:50
                                                   rfLevel:-749
                                          frequencyHopping:1
                                          listenBeforeTalk:2
                                            continuousWave:0];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.23. getReadTimeForMultiAntenna

- (BOOL) getReadTimeForMultiAntenna;				
	Parameter Name	**In/Out**	**Type**	**Descriptions**
Return Values	-	Out	BOOL	YES: The command was sent successfully. NO: The command failed to be sent.

■ **Function Descriptions**:

Gets the read time of each antenna in the RFID settings.

After this function is executed, call back the "readTimeForMultiAntennaReceivedAnt1" (see 2.3.17) delegate

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
BOOL status = [asReaderWiredBOXType getReadTimeForMultiAntenna];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.24. setReadTimeForMultiAntennaReceivedAnt1

-(BOOL) setReadTimeForMultiAntennaReceivedAnt1:(uint16_t) ant1
Ant2:(uint16_t) ant2
Ant3:(uint16_t) ant3
Ant4:(uint16_t) ant4
Ant5:(uint16_t) ant5
Ant6:(uint16_t) ant6
Ant7:(uint16_t) ant7
Ant8:(uint16_t) ant8;

	Parameter Name	**In/Out**	**Type**	**Descriptions**
Parameter	ant1	In	uint16_t	The read time of antenna 1.
Parameter	ant2	In	uint16_t	The read time of antenna 2.
Parameter	ant3	In	uint16_t	The read time of antenna 3.
Parameter	ant4	In	uint16_t	The read time of antenna 4.
Parameter	ant5	In	uint16_t	The read time of antenna 5.
Parameter	ant6	In	uint16_t	The read time of antenna 6.
Parameter	ant7	In	uint16_t	The read time of antenna 7.
Parameter	ant8	In	uint16_t	The read time of antenna 8.
Return Values	-	Out	BOOL	YES: The command was sent successfully. NO: The command failed to be sent.

■ **Function Descriptions**:

Sets the read time for each antenna in the RFID settings.

After this function is executed, call back the “didSetReadTimeForMultiAntenna” (see 2.3.18) delegate

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
BOOL status = [asReaderWiredBOXType setReadTimeForMultiAntennaReceivedAnt1:10
Ant2:10 Ant3:10 Ant4:10 Ant5:10 Ant6:10 Ant7:10 Ant8:10];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.25. setDHCPOnMac

- (BOOL) setDHCPOnMac:(NSString*) macAddr;

	Parameter Name	In/Out	Type	Descriptions
Parameter	macAddr	In	NSString	MAC address.
Return Values	-	Out	BOOL	YES: The command was sent successfully. NO: The command failed to be sent.

■ **Function Descriptions**:

Sets a dynamic IP address.

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
BOOL status = [asReaderWiredBOXType setDHCPOnMac:@"AC:BC:32:02:8D:32"];
if (status) {
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.26. setDHCPOffMac

-(BOOL) setDHCPOffMac:(NSString*) macAddr
IP:(NSString*) ip
Subnet:(NSString*) subnet
Gateway:(NSString*) gateway
DNSServer:(NSString*) dnsServer;

	Parameter Name	In/Out	Type	Descriptions
Parameter	macAddr	In	NSString	MAC address.
Parameter	ip	In	NSString	IP address.
Parameter	subnet	In	NSString	Subnet mask.
Parameter	gateway	In	NSString	Gateway.
Parameter	dnsServer	In	NSString	Domain name system server.
Return Values	-	Out	BOOL	YES: The command was sent successfully. NO: The command failed to be sent.

■ **Function Descriptions**:

Sets a static IP address and the network information.

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
BOOL status = [asReaderWiredBOXType setDHCPOffMac:@"AC:BC:32:02:8D:32"
IP:@"192.168.1.10" Subnet:@"255.255.255.255" Gateway:@"255.255.255.255"
DNSServer:@"255.255.255.255"];
if (status) {
```

```
//The command was sent successfully.
} else {
//The command failed to be sent.
}
```

2.2.27. tX

- (BOOL)tX:(NSData *)data;

	Parameter Name	In/Out	Type	Descriptions
Parameter	data	In	NSData	Data command
Return Values	-	Out	BOOL	YES: Send command successfully NO: Send command failed

Function Descriptions:

Send data commands to the Wired BOX-Type device.

Sample Code: (Note: asReaderWiredBOXType is an instance of AsReaderWiredBOXType class.)

```
Byte buf[6]    = {0x00,0xB1,0x00,0x00,0x7E,0x30};
NSData *data = [NSData dataWithBytes:buf length:8];
BOOL status = [asReaderWiredBOXType tx:data];
if (status) {
// Sends command successfully.
} else {
// Sends command faild.
}
```

2.2.28. getRFIDModuleVersion

- (BOOL) getRFIDModuleVersion;

	Parameter Name	In/Out	Type	Descriptions
Return Values	-	Out	BOOL	YES: Get firmware version of the RFID module successfully NO: Get firmware version of the RFID module failed

Function Descriptions:

Gets the RFID module firmware version of the Wired BOX-Type.

Once this function is performed, the entrusted "rfidModuleVersionReceived" (see 2.3.20) is the callback.

Sample Code: (Note: asReaderWiredBOXType is an instance of AsReaderWiredBOXType class.)

```
BOOL status = [asReaderWiredBOXType getRFIDModuleVersion];
if (status) {
// Gets firmware version successfully
} else {
// Gets firmware version failed
}
```

2.3. Delegates

2.3.1. searchingIP

-(void) AsReaderWiredBOXType:(AsReaderWiredBOXType*) AsReaderWiredBOXType searchingIP:(NSString*) ip isFinish:(BOOL) isFinish;

	Parameter Name	**In/Out**	**Type**	**Descriptions**
Parameter	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType object.
Parameter	ip	Out	NSString	The IP address of the device.
Parameter	isFinish	Out	BOOL	The searching status. YES: Searching finished. NO: Searching continuously.

■ **Function Descriptions**:

Receives the IP address of the Wired BOX-Type device.

Once calling back the function startSearchDevice (see 2.2.3), this delegate will be called back.

■ **Sample Code:**

```
NSMutableArray *arrayIPs = [[NSMutableArray alloc] init];
- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType
searchingIP:(NSString*) ip isFinish:(BOOL) isFinish{
    if(isFinish == YES) {
        // Searching finished.
    } else {
        // Searching continuously.
      if (![arrayIPs containsObject:ip]) {
            [arrayIPs addObject:ip];
        }
    }
}
```

2.3.2. networkStateChanged

<table>
<tr><th colspan="5">-(void) AsReaderWiredBOXType:(AsReaderWiredBOXType*) AsReaderWiredBOXType networkStateChanged:(AsReaderWiredBOXTypeSDKBOXNetworkState) state error:(NSError *) error;</th></tr>
<tr><th></th><th>Parameter Name</th><th>In/Out</th><th>Type</th><th>Descriptions</th></tr>
<tr><td><b>Parameter</b></td><td>AsReaderWiredBOXType</td><td>Out</td><td>AsReaderWiredBOXType</td><td>AsReaderWiredBOXType object.</td></tr>
<tr><td><b>Parameter</b></td><td>state</td><td>Out</td><td>AsReaderWiredBOXTypeSDKBOXNetworkState</td><td>AsReaderWiredBOXTypeSDKBOXNetworkState,ENUM(see 4.1.2)</td></tr>
<tr><td><b>Parameter</b></td><td>error</td><td>Out</td><td>NSError</td><td>Error message generated when network state changes.</td></tr>
</table>

■ **Function Descriptions**:

This delegate will be called back once the Wired BOX-Type device connection status changes. Once calling back the function connectServer (see 2.2.4), disconnectServer (see 2.2.5), this delegate will be called back.

■ **Sample Code**:

```
- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType
networkStateChanged:(AsReaderWiredBOXTypeSDKBOXNetworkState) state error:(NSError
*) error{
        if (state == AsReaderWiredBOXTypeSDKBOXNetworkStateConnected) {
            //Connected.
} else if (state == AsReaderWiredBOXTypeSDKBOXNetworkStateDisconnected) {
            //Disconnected.
} else if (state == AsReaderWiredBOXTypeSDKBOXNetworkStateRecievedData) {
            //Received some data.
} else {
            //error: the corresponding content returned when a network error occurred
}
}
```

2.3.3. searchedUPDSearchingDeviceInfo

<table>
<tr><th colspan="5">-(void) AsReaderWiredBOXType:(AsReaderWiredBOXType*) AsReaderWiredBOXType searchedUPDSearchingDeviceInfo:(NSArray *) arrInfos;</th></tr>
<tr><th></th><th>Parameter Name</th><th>In/Out</th><th>Type</th><th>Descriptions</th></tr>
</table>

Parameter	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType object.
Parameter	arrInfos	Out	NSArray	An array of device IP address.

■ **Function Descriptions**:

Receives the got IP address of the Wired BOX-Type device.

Once calling back the function startSearchDevice (see 2.2.3), this delegate will be called back.

■ **Sample Code**:

```
- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType
searchedUPDSearchingDeviceInfo:(NSArray *) arrInfos {
// arrInfos An array of device IP address.
}
```

2.3.4. inventoryStartedWithStatus

-(void) AsReaderWiredBOXType:(AsReaderWiredBOXType*) AsReaderWiredBOXType inventoryStartedWithStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode) statusCode;

	Parameter Name	**In/Out**	**Type**	**Descriptions**
Parameter	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType object.
Parameter	statusCode	Out	AsReaderWiredBOXTypeSDKBOXStatusCode	AsReaderWiredBOXTypeSDKBOXStatusCode,ENUM(see 4.1.9).

■ **Function Descriptions**:

Receives the state of starting inventory.

Once calling back the function startInventoryWithTagsMaxTagCount (see 2.2.10), this delegate will be called back.

■ **Sample Code**:

```
- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType
inventoryStartedWithStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode) statusCode {
// statusCode AsReaderWiredBOXTypeSDKBOXStatusCode,ENUM(see 4.1.9)
}
```

2.3.5. inventoryStoppedWithStatus

-(void) AsReaderWiredBOXType:(AsReaderWiredBOXType*) AsReaderWiredBOXType inventoryStoppedWithStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode) statusCode;

	Parameter Name	In/Out	Type	Descriptions
Parameter	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType object.
Parameter	statusCode	Out	AsReaderWiredBOXTypeSDKBOXStatusCode	AsReaderWiredBOXTypeSDKBOXStatusCode,ENUM(see 4.1.9).

■ **Function Descriptions**:

Receives the state of stopping inventory.

Once calling back the function stopInventory (see 2.2.11), this delegate will be called back.

■ **Sample Code**:

```
- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType
inventoryStoppedWithStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode) statusCode {
// statusCode AsReaderWiredBOXTypeSDKBOXStatusCode,ENUM(see 4.1.9).
}
```

2.3.6. receivedPCEPCData

- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType receivedPCEPCData:(NSData *) pcEpc selectAntenna:(AsReaderWiredBOXTypeSDKBOXSelectAntenna) antenna;

	Parameter Name	In/Out	Type	Descriptions
Parameter	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType object.
Parameter	pcEpc	Out	NSData	The read RFID data.
Parameter	antenna	Out	AsReaderWiredBOXTypeSDKBOXSelectAntenna	AsReaderWiredBOXTypeSDKBOXSelectAntenna,ENUM(see 4.1.3).

■ **Function Descriptions**:

Receives the read RFID data.

Once calling back the function startInventoryWithTagsMaxTagCount (see 2.2.10), this delegate will be called back.

■ **Sample Code**:

```
- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType
receivedPCEPCData:(NSData *) pcEpc
selectAntenna:(AsReaderWiredBOXTypeSDKBOXSelectAntenna) antenna {
// pcEpc The read RFID data.
// antenna AsReaderWiredBOXTypeSDKBOXSelectAntenna,ENUM(see 4.1.3).
}
```

2.3.7. updatedRegistryStatus

-(void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType updatedRegistryStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode) statusCode;

	Parameter Name	In/Out	Type	Descriptions
Parameter	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType object.
Parameter	statusCode	Out	AsReaderWiredBOXTypeSDKBOXStatusCode	AsReaderWiredBOXTypeSDKBOXStatusCode,ENUM(see 4.1.9).

■ **Function Descriptions:**

Calls back this function after the RFID settings are permanently saved.

Once calling back the function updateRegistry (see 2.2.12), this delegate will be called back.

■ **Sample Code:**

```
- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType
updatedRegistryStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode) statusCode {
//statusCode AsReaderWiredBOXTypeSDKBOXStatusCode,ENUM(see 4.1.9).
}
```

2.3.8. HIDDataRecived

-(void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType HIDDataRecived:(NSData *) HIDData;

	Parameter Name	In/Out	Type	Descriptions
Parameter	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType object.
Parameter	HIDData	Out	NSData	HID interface data.

■ **Function Descriptions:**

Receives HID data read by the HID input device.

■ **Sample Code:**

```
- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType
HIDDataRecived:(NSData *) HIDData {
// HIDData HID interface data.
}
```

2.3.9. receivedSession

<table>
<tr><th colspan="5">- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType receivedSession:(uint8_t) session;</th></tr>
<tr><th></th><th>Parameter Name</th><th>In/Out</th><th>Type</th><th>Descriptions</th></tr>
<tr><td><b>Parameter</b></td><td>AsReaderWiredBOXType</td><td>Out</td><td>AsReaderWiredBOXType</td><td>AsReaderWiredBOXType object.</td></tr>
<tr><td><b>Parameter</b></td><td>session</td><td>Out</td><td>uint8_t</td><td>Set the value of RFID session.</td></tr>
<tr><td colspan="5"><b>■ Function Descriptions:</b>
Receive the value of Session.
Once calling back the function getSession (see 2.2.14), this delegate will be called back.
<b>■ Sample Code:</b>
- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType receivedSession:(uint8_t) session {
// session Set the value of RFID session.
}</td></tr>
</table>

2.3.10. didSetSessionStatus

<table>
<tr><th colspan="5">- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType didSetSessionStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode) state;</th></tr>
<tr><th></th><th>Parameter Name</th><th>In/Out</th><th>Type</th><th>Descriptions</th></tr>
<tr><td><b>Parameter</b></td><td>AsReaderWiredBOXType</td><td>Out</td><td>AsReaderWiredBOXType</td><td>AsReaderWiredBOXType object.</td></tr>
<tr><td><b>Parameter</b></td><td>state</td><td>Out</td><td>AsReaderWiredBOXTypeSDKBOXStatusCode</td><td>AsReaderWiredBOXTypeSDKBOXStatusCode,ENUM (see 4.1.9).</td></tr>
<tr><td colspan="5"><b>■ Function Descriptions:</b>
Receives the result of setting the session.
Once calling back the function setSession (see 2.2.16), this delegate will be called back.
<b>■ Sample Code:</b>
- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType didSetSessionStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode) state {
// state AsReaderWiredBOXTypeSDKBOXStatusCode,ENUM(see 4.1.9)
}</td></tr>
</table>

2.3.11. receivedOuputPowerAntenna1

- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType
receivedOuputPowerAntenna1:(uint16_t) antenna1
Antenna2:(uint16_t) antenna2
Antenna3:(uint16_t) antenna3
Antenna4:(uint16_t) antenna4
Antenna5:(uint16_t) antenna5
Antenna6:(uint16_t) antenna6
Antenna7:(uint16_t) antenna7
Antenna8:(uint16_t) antenna8
Min:(uint16_t) min
Max:(uint16_t) max;

	Parameter Name	**In/Out**	**Type**	**Descriptions**
Parameter	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType object.
Parameter	antenna1	Out	uint16_t	The output power of antenna 1.
Parameter	antenna2	Out	uint16_t	The output power of antenna 2.
Parameter	antenna3	Out	uint16_t	The output power of antenna 3.
Parameter	antenna4	Out	uint16_t	The output power of antenna 4.
Parameter	antenna5	Out	uint16_t	The output power of antenna 5.
Parameter	antenna6	Out	uint16_t	The output power of antenna 6.
Parameter	antenna7	Out	uint16_t	The output power of antenna 7.
Parameter	antenna8	Out	uint16_t	The output power of antenna 8.
Parameter	min	Out	uint16_t	the minimum power output of the antenna
Parameter	max	Out	uint16_t	the maximum power output of the antenna

■ **Function Descriptions:**

Receives the power output of each antenna.

Once calling back the function getOutputPowerLevelForMultiAntenna (see 2.2.17), this delegate will be called back.

■ **Sample Code:**

```
- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType receivedOuputPowerAntenna1:(uint16_t) antenna1 Antenna2:(uint16_t) antenna2 Antenna3:(uint16_t) antenna3 Antenna4:(uint16_t) antenna4 Antenna5:(uint16_t) antenna5 Antenna6:(uint16_t) antenna6 Antenna7:(uint16_t) antenna7 Antenna8:(uint16_t) antenna8 Min:(uint16_t) min Max:(uint16_t) max {
// antenna1: the output power of antenna 1.
// antenna2: the output power of antenna 2.
// antenna3: the output power of antenna 3.
// antenna4: the output power of antenna 4.
// antenna5: the output power of antenna 5.
// antenna6: the output power of antenna 6.
// antenna7: the output power of antenna 7.
// antenna8: the output power of antenna 8.
// min: the minimum power output of the antenna
// max: the maximum power output of the antenna
}
```

2.3.12. didSetOutputPowerForMultiAntenna

- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType didSetOutputPowerForMultiAntenna:(AsReaderWiredBOXTypeSDKCommonStatus) statusCode;

	Parameter Name	In/Out	Type	Descriptions
Parameter	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType object
Parameter	state	Out	AsReaderWiredBOXTypeSDKCommonStatus	AsReaderWiredBOXTypeSDKCommonStatus,ENUM (see 4.1.10)

■ **Function Descriptions:**

Receives the execution result of setting the output power value for each antenna.

Once calling back the function setOutputPowerLevelAntenna1 (see 2.2.17), this delegate will be called back.

■ **Sample Code:**

```
- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType didSetOutputPowerForMultiAntenna:(AsReaderWiredBOXTypeSDKCommonStatus) statusCode {
```

```
// statusCode AsReaderWiredBOXTypeSDKCommonStatus,ENUM(see 4.1.10)
}
```

2.3.13. receivedOutputPowerLevel

- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType receivedOutputPowerLevel:(uint16_t) powerLevel maxPower:(uint16_t) maxPower minPower:(uint16_t) minPower;

	Parameter Name	**In/Out**	**Type**	**Descriptions**
Parameter	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType object.
Parameter	powerLevel	Out	uint16_t	The current output power.
Parameter	maxPower	Out	uint16_t	The maximum output power that can be set.
Parameter	minPower	Out	uint16_t	The minimum output power that can be set.

■ **Function Descriptions:**

Receives the common output power of the antennas and the range of output power that can be set.

Once calling back the function getOutputPowerLevel (see 2.2.19), this delegate will be called back.

■ **Sample Code:**

```
- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType
receivedOutputPowerLevel:(uint16_t) powerLevel maxPower:(uint16_t) maxPower
minPower:(uint16_t) minPower {
// powerLevel: the current output power.
// maxPower: the maximum output power that can be set.
// minPower: the minimum output power that can be set.
}
```

2.3.14. didSetOutputPowerLevelStatus

- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType didSetOutputPowerLevelStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode) statusCode;

	Parameter Name	**In/Out**	**Type**	**Descriptions**
Parameter	AsReaderWired	Out	AsReaderWiredBOXT	AsReaderWiredBOXType

	BOXType		ype	object.
Parameter	statusCode	Out	AsReaderWiredBOXTypeSDKBOXStatusCode	AsReaderWiredBOXTypeSDKBOXStatusCode,ENUM(see 4.1.9).

■ **Function Descriptions:**

Receives the execution result of the common output power of the antennas.

Once calling back the function setOutputPowerLevel (see 2.2.20), this delegate will be called back.

■ **Sample Code:**

```
- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType didSetOutputPowerLevelStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode) statusCode {
// statusCode AsReaderWiredBOXTypeSDKBOXStatusCode,ENUM(see 4.1.9).
}
```

2.3.15. receivedFHLBTData

- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType receivedFHLBTData:(NSData *) data readTime:(int) readTime idleTime:(int) idleTime;

	Parameter Name	In/Out	Type	Descriptions
Parameter	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType object
Parameter	data	Out	NSData	Read time (16bit). Idle time (16bit). Carrier listening time (16bit). Carrier listening RF level (16bit). FH (8bit). LBT (8bit). CW (8bit).
Parameter	readTime	Out	int	Read time.
Parameter	idleTime	Out	int	Idle time.

■ **Function Descriptions:**

Receives the values of FH and LBT.

Once calling back the function getFHLBTParam (see 2.2.21), this delegate will be called back.

■ **Sample Code:**

```
- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType receivedFHLBTData:(NSData *) data readTime:(int) readTime idleTime:(int) idleTime {
```

```
// data
Read time (16bit).
Idle time (16bit).
Carrier listening time (16bit).
Carrier listening RF level (16bit).
FH (8bit)
LBT (8bit)
CW (8bit)
// readTime
Read time
// idleTime
Idle time.
}
```

2.3.16. didSetFHLBTStatus

- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType didSetFHLBTStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode) statusCode;

	Parameter Name	In/Out	Type	Descriptions
Parameter	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType object.
Parameter	statusCode	Out	AsReaderWiredBOXTypeSDKBOXStatusCode	AsReaderWiredBOXTypeSDKBOXStatusCode,ENUM(see 4.1.9).

■ **Function Descriptions:**

Receives the result of setting the FH and LBT.

Once calling back the function setFHLBTParamReadTime (see 2.2.21), this delegate will be called back.

■ **Sample Code:**

```
- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType
didSetFHLBTStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode) statusCode {
// statusCode AsReaderWiredBOXTypeSDKBOXStatusCode,ENUM(see 4.1.9)
}
```

2.3.17. readTimeForMultiAntennaReceivedAnt1

- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType readTimeForMultiAntennaReceivedAnt1:(uint16_t) ant1 Ant2:(uint16_t) ant2 Ant3:(uint16_t) ant3 Ant4:(uint16_t) ant4 Ant5:(uint16_t) ant5 Ant6:(uint16_t) ant6 Ant7:(uint16_t) ant7 Ant8:(uint16_t) ant8;

	Parameter Name	In/Out	Type	Descriptions
Parameter	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType object.
Parameter	ant1	Out	uint16_t	The read time of antenna 1.
Parameter	ant2	Out	uint16_t	The read time of antenna 2.
Parameter	ant3	Out	uint16_t	The read time of antenna 3.
Parameter	ant4	Out	uint16_t	The read time of antenna 4.
Parameter	ant5	Out	uint16_t	The read time of antenna 5.
Parameter	ant6	Out	uint16_t	The read time of antenna 6.
Parameter	ant7	Out	uint16_t	The read time of antenna 7.
Parameter	ant8	Out	uint16_t	The read time of antenna 8.

■ **Function Descriptions:**

Receives the read time of the antennas.

Once calling back the function getReadTimeForMultiAntenna (see 2.2.22), this delegate will be called back.

■ **Sample Code:**

```
- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType
readTimeForMultiAntennaReceivedAnt1:(uint16_t) ant1 Ant2:(uint16_t) ant2 Ant3:(uint16_t)
ant3 Ant4:(uint16_t) ant4 Ant5:(uint16_t) ant5 Ant6:(uint16_t) ant6 Ant7:(uint16_t) ant7
Ant8:(uint16_t) ant8 {
// ant1         The read time of antenna 1.
// ant2         The read time of antenna 2.
// ant3         The read time of antenna 3.
// ant4         The read time of antenna 4.
// ant5         The read time of antenna 5.
// ant6         The read time of antenna 6.
// ant7         The read time of antenna 7.
// ant8         The read time of antenna 8.
}
```

2.3.18. didSetReadTimeForMultiAntenna

- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType didSetReadTimeForMultiAntenna:(AsReaderWiredBOXTypeSDKCommonStatus) status;

	Parameter Name	In/Out	Type	Descriptions

Parameter	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType object.
Parameter	status	Out	AsReaderWiredBOXTypeSDKCommonStatus	AsReaderWiredBOXTypeSDKCommonStatus,ENUM(see 4.1.10).

■ **Function Descriptions:**

Receives the result of setting the read time of each antenna.

Once calling back the function setReadTimeForMultiAntennaReceivedAnt1 (see 2.2.23), this delegate will be called back.

■ **Sample Code:**

```
- (void) AsReaderWiredBOXType:(AsReaderWiredBOXType *) AsReaderWiredBOXType
didSetReadTimeForMultiAntenna:(AsReaderWiredBOXTypeSDKCommonStatus) status {
// status AsReaderWiredBOXTypeSDKCommonStatus,ENUM(see 4.1.10)
}
```

2.3.19. setRecharging

- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType setRecharging:(AsReaderWiredBOXTypeSDKBOXStatusCode)statusCode;

	Parameter Name	In/Out	Type	Descriptions
Parameter	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType object
Parameter	statusCode	Out	AsReaderWiredBOXTypeSDKBOXStatusCode	AsReaderWiredBOXTypeSDKBOXStatusCode, Enum (see 4.1.9).

Function Descriptions:

Receives the execution results of the charging setting for the iOS device connected to the Wired BOX-Type device.

Once the function setRecharging (see 2.2.13) is called, this delegate is called back.

Sample Code:

```
- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType
setRecharging:(AsReaderWiredBOXTypeSDKBOXStatusCode)statusCode {
// statusCode AsReaderWiredBOXTypeSDKBOXStatusCode, Enum (see 4.1.9).
}
```

2.3.20. rfidModuleVersionReceived

- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType rfidModuleVersionReceived:(NSString *)rfidModuleVersion;

	Parameter Name	In/Out	Type	Descriptions
Parameter	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType object
Parameter	rfidModuleVersion	Out	NSString	Firmware version of RFID module

Function Descriptions:
Receives the RFID module firmware version of the Wired BOX-Type device.
Once the function getRFIDModuleVersion (see 2.2.28) is called, this delegate is called back.
Sample Code:
- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType rfidModuleVersionReceived:(NSString *)rfidModuleVersion {
// rfidModuleVersion: The RFID module firmware version of the Wired BOX.
}				

2.3.21. receivedRechargedStatus

- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType receivedRechargedStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode)statusCode;				
	Parameter Name	**In/Out**	**Type**	**Descriptions**
Parameter	AsReaderWiredBOXType	Out	AsReaderWiredBOXType	AsReaderWiredBOXType object
Parameter	statusCode	Out	AsReaderWiredBOXTypeSDKBOXStatusCode	AsReaderWiredBOXTypeSDKBOXStatusCode, Enum (see 4.1.9).

Function Descriptions:
Receives the charging status of the phone which is connected to the Wired BOX-Type device.
Once the function getRechargedStatus (see 2.2.14) is called, this delegate is called back.
 	
Sample Code:
- (void)AsReaderWiredBOXType:(AsReaderWiredBOXType *)AsReaderWiredBOXType receivedRechargedStatus:(AsReaderWiredBOXTypeSDKBOXStatusCode)statusCode {
// statusCode AsReaderWiredBOXTypeSDKBOXStatusCode, Enum (see 4.1.9).
}				

3. AsWiredBOXPrinter Class

AsWiredBOXPrinter object can be used to get printer status, to print text, images, and QR codes, and to cut paper, open the money box, and perform other user-defined functions.

3.1. Properties

Property Name	Property	Type	Descriptions
printDelegate	nonatomic, weak	AsReaderWiredBOXTypePrinterDelegate	Delegate (see 3.3).
isConnected	nonatomic, assign, readonly	BOOL	Connection status.

3.2. Functions

3.2.1. initWithAsReaderWiredBOXType

-(id) initWithAsReaderWiredBOXType:(AsReaderWiredBOXType *) asReaderWiredBOXType;				
	Parameter Name	**In/Out**	**Type**	**Descriptions**
Parameter	asReaderWiredBOXType	In	AsReaderWiredBOXType	AsReaderWiredBOXType object.
Return Values	-	Out	AsWiredBOXPrinter	AsWiredBOXPrinter object.

■ **Function Descriptions:**

Creates AsWiredBOXPrinter object.

■ **Sample Code:** (Note: asReaderWiredBoxType is an instance of AsReaderWiredBoxType.)

```
AsWiredBOXPrinter *asWiredBOXPrinter = [[AsWiredBOXPrinter alloc]
initWithAsReaderWiredBOXType:asReaderWiredBOXType];
```

3.2.2. getPrinterStatus

-(void) getPrinterStatus;
■ **Function Descriptions:**

Gets the current status of the printer.
After this function is executed, call back the “AsReaderWiredBOXTypePrinterStatus” (see 3.3.1) delegate.

■ **Sample Code:** (Note: asPrinter is an instance of AsWiredBOXPrinter class.)

```
[asPrinter getPrinterStatus];
```

3.2.3. addPrintText

-(void) addPrintText:(NSString*) text
font:(AsReaderWiredBOXTypeSDKMNPrintFont) font
printerAlignment:(AsReaderWiredBOXTypeSDKPrinterAlignment) printerAlignment;

	Parameter Name	**In/Out**	**Type**	**Descriptions**
Parameter	text	In	NSString	Data to be printed.
Parameter	font	In	AsReaderWiredBOXTypeSDKMNPrintFont	AsReaderWiredBOXTypeSDKMNPrintFont,ENUM (see 4.1.6).
Parameter	printerAlignment	In	AsReaderWiredBOXTypeSDKPrinterAlignment	AsReaderWiredBOXTypeSDKPrinterAlignment,ENUM (see 4.1.5).

■ **Function Descriptions:**

Add the print data for each line.

■ **Sample Code:** (Note: asPrinter is an instance of AsWiredBOXPrinter class.)

```
[asPrinter addPrintText:@"123456" font:AsReaderWiredBOXTypeSDKMNPrintFont_1
printerAlignment:AsReaderWiredBOXTypeSDKPrinterAlignmentRight];
```

3.2.4. addPrintQRCode

-(void) addPrintQRCode:(NSString*) qrCode
size:(AsReaderWiredBOXTypeSDKQrDotSize) size
printerAlignment:(AsReaderWiredBOXTypeSDKPrinterAlignment) printerAlignment;

	Parameter Name	**In/Out**	**Type**	**Descriptions**
Parameter	qrCode	In	NSString	The QRCode to be printed.
Parameter	size	In	AsReaderWiredBOXTypeSDKQrDotSize	AsReaderWiredBOXTypeSDKQrDotSize,ENUM (see 4.1.8).
Parameter	printerAlignment	In	AsReaderWiredBOXTypeSDKPrinterAlignment	AsReaderWiredBOXTypeSDKPrinterAlignment,ENUM (see 4.1.5).

■ **Function Descriptions:**

Add the QRCode to print.

■ **Sample Code:** (Note: asPrinter is an instance of AsWiredBOXPrinter class.)

[asPrinter addPrintQRCode:@"123456" size: AsReaderWiredBOXTypeSDKQrDotSize_1 printerAlignment: AsReaderWiredBOXTypeSDKPrinterAlignmentLeft];

3.2.5. addPrintImage

-(void) addPrintImage:(UIImage *) image
width:(float) width
height:(float) height
printerAlignment:(AsReaderWiredBOXTypeSDKPrinterAlignment) printerAlignment;

	Parameter Name	In/Out	Type	Descriptions
Parameter	image	In	UIImage	The images to be printed.
Parameter	width	In	float	The width of image to be printed.
Parameter	height	In	float	The height of image to be printed.
Parameter	printerAlignment	In	AsReaderWiredBOXTypeSDKPrinterAlignment	AsReaderWiredBOXTypeSDKPrinterAlignment,ENUM (see 4.1.5).

■ **Function Descriptions:**

Add the image to print.

■ **Sample Code:** (Note: asPrinter is an instance of the AsWiredBOXPrinter class, pic is the name of the local picture)

UIImage *image = [UIImage imageNamed:@”pic”];

[asPrinter addPrintImage:image width:200 height:200 printerAlignment: AsReaderWiredBOXTypeSDKPrinterAlignmentCenter];

3.2.6. addCutCommand

-(void) addCutCommand:(AsReaderWiredBOXTypeSDKCutType) cutType;

	Parameter Name	In/Out	Type	Descriptions
Parameter	cutType	In	AsReaderWiredBOXTypeSDKCutType	AsReaderWiredBOXTypeSDKCutType,ENUM (see 4.1.4).

■ **Function Descriptions:**

Add a paper cutting instructions.

■ **Sample Code:** (Note: asPrinter is an instance of AsWiredBOXPrinter class.)

[asPrinter addCutCommand:AsReaderWiredBOXTypeSDKCutTypeHalf];

3.2.7. openDrawer

-(void) openDrawer;
■ **Function Descriptions:** Open the drawer. ■ **Sample Code:** (Note: asPrinter is an instance of AsWiredBOXPrinter class.) [asPrinter openDrawer];

3.2.8. cleanPrintData

-(void) cleanPrintData;
■ **Function Descriptions:** Clear the added print data. ■ **Sample Code:** (Note: asPrinter is an instance of AsWiredBOXPrinter class.) [asPrinter cleanPrintData];

3.2.9. doPrint

-(void) doPrint;
■ **Function Descriptions:** Prints the previously added print data, and the added print data will be emptied after successful printing. After this function is executed, call back the “AsReaderWiredBOXTypePrinterConnected” (see 3.3.2), “retryPrintOnError” (see 3.3.3), “whenPrintSuccessed” (see 3.3.4) delegate ■ **Sample Code:** (Note: asPrinter is an instance of AsWiredBOXPrinter class.) [asPrinter doPrint];

3.2.10. txRawData

-(void) txRawData:(NSData *) txRawData;				
	Parameter Name	**In/Out**	**Type**	**Descriptions**
Parameter	txRawData	In	NSData	Functions included with the printer, but not supported by the SDK.
■ **Function Descriptions:** Users can communicate directly with the printer through this function to realize functions not supported by the SDK. After this function is executed, if the printer returns the result of execution, the delegate "				

rxRawData " (see 3.3.5) will be called back.

■ **Sample Code:** (Note: asPrinter is an instance of the AsWiredBOXPrinter class, data is NSData type.)

[asPrinter txRawData:data];

3.3. Delegates

3.3.1. AsReaderWiredBOXTypePrinterStatus

-(void) AsReaderWiredBOXTypePrinterStatus:(AsReaderWiredBOXTypeSDKPrinterStatus) printerStatus;

	Parameter Name	**In/Out**	**Type**	**Descriptions**
Parameter	printerStatus	Out	AsReaderWiredBOXTypeSDKPrinterStatus	AsReaderWiredBOXTypeSDKPrinterStatus,ENUM (see 4.1.1).

■ **Function Descriptions:**

Receives the status of the printer.

Once calling back the function getPrinterStatus (see 3.2.2), this delegate will be called back.

■ **Sample Code:**

```
-(void) AsReaderWiredBOXTypePrinterStatus:(AsReaderWiredBOXTypeSDKPrinterStatus) printerStatus {
// printerStatus AsReaderWiredBOXTypeSDKPrinterStatus,ENUM (see 4.1.1).
}
```

3.3.2. AsReaderWiredBOXTypePrinterConnected

- (void) AsReaderWiredBOXTypePrinterConnected:(BOOL) isConnected;

	Parameter Name	**In/Out**	**Type**	**Descriptions**
Parameter	isConnected	Out	BOOL	YES: Connected. NO: Disconnected.

■ **Function Descriptions:**

Returns the connection status of the printer.

Once calling back the function doPrint (see 3.2.9), this delegate will be called back.

■ **Sample Code:**

```
- (void) AsReaderWiredBOXTypePrinterConnected:(BOOL) isConnected {
```

```
// isConnected YES: Connected.
NO: Disconnected.
}
```

3.3.3. retryPrintOnError

- (BOOL) retryPrintOnError;

	Parameter Name	In/Out	Type	Descriptions
Return Values	-	In	BOOL	YES: Reprint. NO: Not reprint.

■ **Function Descriptions:**

Try to reprint without clearing the data.

Once calling back the function doPrint (see 3.2.9), this delegate will be called back.

■ **Sample Code:**

```
- (BOOL) retryPrintOnError{
      return YES;
}
```

3.3.4. whenPrintSuccessed

- (void) whenPrintSuccessed;

■ **Function Descriptions:**

This function is called after printing data.

Once calling back the function doPrint (see 3.2.9), this delegate will be called back.

■ **Sample Code:**

```
- (void) whenPrintSuccessed {
     // Prints data successfully.
}
```

3.3.5. rxRawData

- (void) rxRawData:(NSData *) rxRawData;

	Parameter Name	In/Out	Type	Descriptions
Parameter	rxRawData	Out	NSData	Receives the execution results returned by the printer.

■ **Function Descriptions:**

Receives the execution results of functions that come with the printer but are not supported by the SDK.

Once calling back the function txRawData (see 3.2.10), this delegate will be called back.

■ **Sample Code:**

```
- (void) rxRawData:(NSData *) rxRawData {
     // rxRawData Receives the execution results returned by the printer.
}
```

4. AsReaderWiredBOXTypeSDKDefine Class

AsReaderWiredBOXTypeSDKDefine object is an enumerated type of each parameter.

4.1. Enum

4.1.1. AsReaderWiredBOXTypeSDKPrinterStatus

The printer status:

Definition	Value	Descriptions
AsReaderWiredBOXTypeSDKPrinterStatus_NOERROR	0	No error.
AsReaderWiredBOXTypeSDKPrinterStatus_NOPAPER	1	No paper.
AsReaderWiredBOXTypeSDKPrinterStatus_InstitutionalFailure	2	The machine is out of order.
AsReaderWiredBOXTypeSDKPrinterStatus_CoverOpen	3	The cover of the machine is open.

4.1.2. AsReaderWiredBOXTypeSDKBOXNetworkState

The network status:

Definition	Value	Descriptions
AsReaderWiredBOXTypeSDKBOXNetworkStateConnected_	0	Connected
AsReaderWiredBOXTypeSDKBOXNetworkStateDisconnected	1	Disconnected
AsReaderWiredBOXTypeSDKBOXNetworkStateConnecting	2	Connecting
AsReaderWiredBOXTypeSDKBOXNetworkStateRecievedData	3	Received data
AsReaderWiredBOXTypeSDKBOXNetworkStateError	4	Error occurred

4.1.3. AsReaderWiredBOXTypeSDKBOXSelectAntenna

The selected antenna:

Definition	Value	Descriptions
AsReaderWiredBOXTypeSDKBOXSelectAntennaNone_	0	No antenna
AsReaderWiredBOXTypeSDKBOXSelectAntenna1	1	Antenna 1
AsReaderWiredBOXTypeSDKBOXSelectAntenna2	2	Antenna 2
AsReaderWiredBOXTypeSDKBOXSelectAntenna3	3	Antenna 3
AsReaderWiredBOXTypeSDKBOXSelectAntenna4	4	Antenna 4
AsReaderWiredBOXTypeSDKBOXSelectAntenna5	5	Antenna 5
AsReaderWiredBOXTypeSDKBOXSelectAntenna6	6	Antenna 6
AsReaderWiredBOXTypeSDKBOXSelectAntenna7	7	Antenna 7
AsReaderWiredBOXTypeSDKBOXSelectAntenna8	8	Antenna 8

4.1.4. AsReaderWiredBOXTypeSDKCutType

Paper cutting status:

Definition	Value	Descriptions
AsReaderWiredBOXTypeSDKCutTypeFull_	0x69	All cut.
AsReaderWiredBOXTypeSDKCutTypeHalf	0x6D	Half cut

4.1.5. AsReaderWiredBOXTypeSDKPrinterAlignment

Location of the print content:

Definition	Value	Descriptions
AsReaderWiredBOXTypeSDKPrinterAlignmentLeft	0	In the left of the page.
AsReaderWiredBOXTypeSDKPrinterAlignmentCenter	1	In the middle of the page.
AsReaderWiredBOXTypeSDKPrinterAlignmentRight	2	In the right of the page.

4.1.6. AsReaderWiredBOXTypeSDKMNPrintFont

The font-size of the text to be printed:

Definition	Value	Descriptions
AsReaderWiredBOXTypeSDKMNPrintFont_1	0x00	Font-size 1
AsReaderWiredBOXTypeSDKMNPrintFont_2	0x11	Font-size 2
AsReaderWiredBOXTypeSDKMNPrintFont_3	0x22	Font-size 3
AsReaderWiredBOXTypeSDKMNPrintFont_4	0x33	Font-size 4
AsReaderWiredBOXTypeSDKMNPrintFont_5	0x44	Font-size 5
AsReaderWiredBOXTypeSDKMNPrintFont_6	0x55	Font-size 6
AsReaderWiredBOXTypeSDKMNPrintFont_7	0x66	Font-size 7
AsReaderWiredBOXTypeSDKMNPrintFont_8	0x77	Font-size 8

4.1.7. AsReaderWiredBOXTypeSDKQrModel

QR Model:

Definition	Value	Descriptions
AsReaderWiredBOXTypeSDKQrModel_1	49	QR Model1
AsReaderWiredBOXTypeSDKQrModel_2	50	QR Model2

4.1.8. AsReaderWiredBOXTypeSDKQrDotSize

QR dot size:

Definition	Value	Descriptions
AsReaderWiredBOXTypeSDKQrDotSize_1	1	QR dot size 1x1
AsReaderWiredBOXTypeSDKQrDotSize_2	2	QR dot size 2x2
AsReaderWiredBOXTypeSDKQrDotSize_3	3	QR dot size 3x3
AsReaderWiredBOXTypeSDKQrDotSize_4	4	QR dot size 4x4
AsReaderWiredBOXTypeSDKQrDotSize_5	5	QR dot size 5x5
AsReaderWiredBOXTypeSDKQrDotSize_6	6	QR dot size 6x6

AsReaderWiredBOXTypeSDKQrDotSize_7	7	QR dot size 7x7

4.1.9. AsReaderWiredBOXTypeSDKBOXStatusCode

The status of setting:

Definition	Value	Descriptions
AsReaderWiredBOXTypeSDKBOXStatusCodeUnknown	-1	Unknown
AsReaderWiredBOXTypeSDKBOXStatusCodeSuccess	0	Success
AsReaderWiredBOXTypeSDKBOXStatusCodeFailed	1	Failed

4.1.10. AsReaderWiredBOXTypeSDKCommonStatus

The status of setting:

Definition	Value	Descriptions
AsReaderWiredBOXTypeSDKCommonStatusNone	0	Unknown
AsReaderWiredBOXTypeSDKCommonStatusSuccess	1	Success
AsReaderWiredBOXTypeSDKCommonStatusFailure	2	Failed

4.1.11. AsReaderWiredBOXTypeSDKResultSetRecharging

Charging status

Definition	Value	Descriptions
AsReaderWiredBOXTypeSDKResultSetRechargingSucceed	0	Charge successfully
AsReaderWiredBOXTypeSDKResultSetRechargingFailed_Type_C_Busy	1	Failed to charge due to inventory taking
AsReaderWiredBOXTypeSDKResultSetRechargingFailed	2	Chage failed

4.1.12. AsReaderWiredBOXTypeConnectMethod

Connection type

Definition	**Value**	**Descriptions**
AsReaderWiredBOXTypeConnectMethod_TCP	0	TCP connection
AsReaderWiredBOXTypeConnectMethod_MQTT	1	MQTT connection