



USER MANUAL
BT Mag IOS SDK

Bluetooth Magnetic Stripe Reader

80125503-001-B

06/18/2015

Revision History

| Revision | Date | Description of Changes | By |
|----------|------------|---|----------|
| A | 04/02/2013 | Initial Release | C.H. |
| B | 06/18/2015 | Added 64bit architecture support. Support iOS v6.x, v7.x and v8.x. Added API: enableLogging. Changed the class name "EventThreadStarterBTMag" in SDK. | Xinhu Li |

Platform

IOS 6.0 and above

Library

Add the SDK static library to your xCode application project

In xCode software, open your project. Show the project navigator, then select project name, Right click ->Add Files to "...". Add the following 3 files.

BTmsr.h

BTmsrDelegate.h

libBTMagSDK.a

Include the following file in the project

```
# import "BTmsr.h"
```

```
# import "BTmsrDelegate.h"
```

API Summary

BTmsr Class

[BTmsrDidDisconnect;](#)

[cardDataReceived:\(NSData *\)CardData: \(Byte\)TracksStatus;](#)

[Open;](#)

[Close;](#)

[GetSecurityLevel;](#)

[GetEncryptionType;](#)

[GetFirmwareVersion;](#)

[GetSerialNumber;](#)

[GetEnzyModuleFWVersion;](#)

[directIO:\(NSData*\)rawBytes;](#)

[GetAllSettings;](#)

[GetEnzyBDAddress;](#)

[SetAllSettingDefault;](#)

[SetMSREnable;](#)

[SetDecodingMethod:](#)
[SetErrorIndication:](#)
[SetEnhance:](#)
[GetDecodingMethod:](#)
[SetEncryptionType:](#)
[SetKeyManagementType:](#)
[GetKeyManagementType:](#)
[SetTracksSelection:](#)
[GetTracksSelection:](#)
[SetTrackSeparator:](#)
[GetTrackSeparator:](#)
[SetTerminator:](#)
[GetTerminator:](#)
[SetTrackPrefix:](#)
[GetTrackPrefix:](#)
[SetTrackSuffix:](#)
[GetTrackSuffix:](#)
[GetCardType:](#)
[GetTrack1:](#)
[GetTrack2:](#)
[GetTrack3:](#)
[GetTrack1_EncryptedData:](#)
[GetTrack2_EncryptedData:](#)
[GetTrack3_EncryptedData:](#)
[GetKSN:](#)
[GetSessionID:](#)
[enableLogging:](#)

API Description

BTmsr Class

void BTmsrDidDisconnect:

Description: Disconnect to the Bluetooth device. This is a call back function, which needs to be implemented in application. When the BTMag device is powered off from connection status, this function will be called.

Parameter: Nil

Return: Nil

```
Example: -(void)BTmsrDidConnect
        {
            NSLog(@"disconnected.");
        }
```

void cardDataReceived:(NSData *) CardData: (Byte) TracksStatus

BT Mag SDK User Manual

Description: Receive swipe Card Data. This is a call back function which is needed to be implemented in application. When swipe a card in the BTMag device, this function will be called.

CardData:

The card data is filling the ID Tech format. For detailed information, please refer to 80125501-001 BTMag User Manual.

TracksStatus :

Parameter: The flag byte of all 3 tracks decode status. Detail:
Bit 0: 1-track1 present; 0- No track1 present.
Bit 1: 1-track2 present; 0- No track1 present.
Bit 2: 1-track3 present; 0- No track1 present.
Bit 3: 1-track1 decode nicely. 0- track1 decode error.
Bit 4: 1-track2 decode nicely. 0- track2 decode error.
Bit 5: 1-track3 decode nicely. 0- track3 decode error.

Return: Nil

Example:

```
void cardDataReceived:(NSData *) CardData: (Byte) TracksStatus
{
    NSLog(@"whole card data: %@", CardData.description);
    Case 1: Only one byte data, 0x48 indicates one swiping is coming ;
    Case 2: Only one byte data, 0x49 indicates bad swiping and no card data will
    be sent.
    Case 3: after good swiping, card data.
}
```

BOOL open:

Description: Open all connection and start receiving data

Parameter: Nil

Return: If successful, then return True, else False.

Example: BOOL result=[BTmsrDevice open];

void close:

Description: Close all connection and stop receiving data

Parameter: Nil

Return: Nil

Example: [BTmsrDevice close];

Byte GetSecurityLevel:

Description: Get the security level of the device

BT Mag SDK User Manual

Parameter: Nil

Return:

- 0x31 Security Level 1

By default, readers from the factory are configured to have this security level. There is no encryption process, no key serial number transmitted with decoded data. The reader functions as a non-encrypting reader and the decoded track data is sent out in default mode.

- 0x32 Security Level 2

Key Serial Number and Base Derivation Key have been injected but the encryption process is not yet activated. The reader will send out decoded track data in default format. Set the encryption type to TDES and AES will change the reader to security level 3.

- 0x33 Security Level 3

Both Key Serial Number and Base Derivation Keys are injected and encryption mode is turned on. For payment cards, both encrypted data and masked clear text data are sent out. When the encryption is turned on, level 3 is the default security level.

Example: Byte result=[BTmsrDevice GetSecurityLevel];

Byte GetEncryptionType:

Description: Get the encryption type of the device

Parameter: Nil

Return: 0x30: Encryption Disabled; 0x31: TDES; 0x32: .AES

Example: Byte result=[BTmsrDevice GetEncryptionType];

NSString* GetFirmwareVersion:

Description: Get the firmware Version of the device

Parameter: Nil

Return: firmware Version description

Example: NSString* result=[BTmsrDevice GetFirmwareVersion];

NSString* GetSerialNumber:

Description: Get the Serial Number of the device

Parameter: Nil

Return: Serial Number

Example: NSString* result=[BTmsrDevice GetSerialNumber];

NSString* GetEnzyModuleFWVersion:

Description: Get the enzy module FW version of the device

Parameter: Nil

Return: enzy module FW version of the device

BT Mag SDK User Manual

Example: NSString* result=[BTmsrDevice GetEnzyModuleFWVersion];

NSData* directIO: (NSData*)rawBytes

Description: Send command directly
Parameter: NSData* rawByte={0x52,0x22};
Return: Response of the command
Example: NSData* result=[BTmsrDevice direction:rawByte];

NSData* GetAllSettings:

Description: Get all the settings of the device
Parameter: Nil
Return: All the setting
Example: NSData* result=[BTmsrDevice GetAllSettings];

NSString* GetEnzyBDAddress:

Description: Get the enzy BD Address of the device
Parameter: Nil
Return: enzy BD Address
Example: NSString* result=[BTmsrDevice GetEnzyBDAddress];

BOOL SetAllSettingDefault:

Description: Set all settings default to the device
Parameter: Nil
Return: True: Set success, False: Failure
Example: BOOL result=[BTmsrDevice SetAllSettingDefault];

BOOL SetMSREnable: (BOOL)Value

Description: Set MSR Enable or Disable
Parameter: BOOL True or False
Return: True: Set success, False: Failure
Example: BOOL result=[BTmsrDevice SetMSREnable:b];

BOOL SetDecodingMethod: (unsigned char)DecodingMethod

Description: Set Decoding Method
Parameter: unsigned char c=
0x31: Decoding in Both Directions. If the encryption feature is enabled, the key management method used is DUKPT,
0x33: Moving stripe along head against direction of encoding. If the encryption feature is enabled, the key management method used is DUKPT.
0x32: Moving stripe along head in direction of encoding. If the encryption

BT Mag SDK User Manual

feature is enabled, the key management method used is DUKPT.

0x30: Raw Data Decoding in Both Directions, send out in ID TECH mode.

0x34: Raw Data Decoding in Both Directions, send out in other mode. If the encryption feature is enabled, the key management method used is fixed key.

Return: True: Set success, False: Failure

Example: `BOOL result=[BTmsrDevice SetDecodingMethod:c];`

BOOL SetErrorIndication:

Description: Set error indication

Parameter: Nil

Return: True: Set success, False: Failure

Example: `BOOL result=[BTmsrDevice SetErrorIndication];`

BOOL SetEnhance:

Description: Set reader to Enhanced Encryption Format

Parameter: Nil

Return: True: Set success, False: Failure

Example: `BOOL result=[BTmsrDevice SetEnhance];`

Byte GetDecodingMethod:

Description: Get Decoding Method of the device

Parameter: Nil

Return: 0x31: Decoding in Both Directions. If the encryption feature is enabled, the key management method used is DUKPT,

0x33: Moving stripe along head against direction of encoding. If the encryption feature is enabled, the key management method used is DUKPT.

0x32: Moving stripe along head in direction of encoding. If the encryption feature is enabled, the key management method used is DUKPT.

0x30: Raw Data Decoding in Both Directions, send out in ID TECH mode.

0x34: Raw Data Decoding in Both Directions, send out in other mode. If the encryption feature is enabled, the key management method used is fixed key.

Example: `Byte result=[BTmsrDevice GetDecodingMethod];`

BOOL SetEncryptionType: (unsigned char)Encryption_Type

Description: Set encryption type

Parameter: unsigned char c=

0x30: Encryption Disabled, 0x31: TDES, 0x32: DES. Once the reader is in Security level 3, the encryption method cannot be disabled.

Return: True: Set success, False: Failure

Example: `BOOL result=[BTmsrDevice SetEncryptionType:c];`

BT Mag SDK User Manual

BOOL SetKeyManagementType: (unsigned char)KeyManagementType

Description: Set Key Management type
Parameter: unsigned char c=
0x30: FIXED Key, 0x31: DUKPT Key.
Return: True: Set success, False: Failure
Example: `BOOL result=[BTmsrDevice SetKeyManagementType:c];`

Byte GetKeyManagementType:

Description: Get the key management type of the device
Parameter: Nil
Return: 0x30: FIXED Key, 0x31: DUKPT Key.
Example: `Byte* result=[BTmsrDevice GetKeyManagementType];`

BOOL SetTracksSelection: (unsigned char)TracksFlag

Description: There are up to three tracks of encoded data on a magnetic stripe. This option selects the tracks that will be read and decoded.
Parameter: unsigned char c=
0x30: Any Track,
0x31:Track 1 only,
0x32:Track 2 only,
0x33:Track 1&Track 2,
0x34:Track 3 only
0x35:Track 1&Track 3,
0x36:Track 2&Track 3,
0x37:ALL Three Tracks,
0x38:Any Track 1&2
0x39:Any Track 2&3
Return: True: Set success, False: Failure
Example: `BOOL result=[BTmsrDevice SetTracksSelection:c];`

Byte GetTracksSelection

Description: Get Tracks Selection of the device
Parameter: Nil
Return: 0x30: Any Track,
0x31:Track 1 only,
0x32:Track 2 only,
0x33:Track 1&Track 2,
0x34:Track 3 only
0x35:Track 1&Track 3,
0x36:Track 2&Track 3,
0x37:ALL Three Tracks,
0x38:Any Track 1&2
0x39:Any Track 2&3

BT Mag SDK User Manual

Example: Byte result=[BTmsrDevice GetTracksSelection];

BOOL SetTrackSeparator: (unsigned char)TrackSeparator

Description: This option allows the user to select the character to be used to separate data decoded by a multiple-track reader. Only one ASCII Character. The default value is CR (0Dh), 0h means no track separator.

Parameter: unsigned char c=
0x0d: TRACK SEPARATOR CR,
0x00:TRACK SEPARATOR NONE,
any ASCII character

Return: True: Set success, False: Failure

Example: BOOL result=[BTmsrDevice SetTrackSeparator:c];

Byte GetTrackSeparator:

Description: Get Track Separator of the device

Parameter: Nil

Return: 0x0d: TRACK SEPARATOR CR,
0x00:TRACK SEPARATOR NONE,
any ASCII character

Example: Byte result=[BTmsrDevice GetTrackSeparator];

BOOL SetTerminator: (unsigned char)Terminator

Description: Terminator characters are used to end a string of data in some application.

Parameter: unsigned char c=
0x0d: TRACK SEPARATOR CR,
0x00:TRACK SEPARATOR NONE,
any ASCII character

Return: True: Set success, False: Failure

Example: BOOL result=[BTmsrDevice SetTerminator:c];

Byte GetTerminator:

Description: Get Terminator of the device

Parameter: Nil

Return: 0x0d: TRACK SEPARATOR CR,
0x00:TRACK SEPARATOR NONE,
any ASCII character

Example: Byte result=[BTmsrDevice GetTerminator];

BOOL SetTrackPrefix: (short)TrackID: (unsigned char) Prefix

Description: Characters can be added to the beginning of a track data. These can be special characters to identify the specific track to the receiving host, or any other character string.

BT Mag SDK User Manual

Up to six ASCII characters can be defined.

Parameter: short TrackID=
0x01: TRACK1,
0x02: TRACK2,
0x03: TRACK3,

unsigned char c=maximum 6 characters

Return: True: Set success, False: Failure

Example: BOOL result=[BTmsrDevice SetTrackPrefix:TrackID:c];

Byte GetTrackPrefix: (short) TrackID

Description: Get the Prefix of the TrackID Track

Parameter: short TrackID=
0x01: TRACK1,
0x02: TRACK2,
0x03: TRACK3,

Return: Prefix of the TrackID track

Example: Byte result=[BTmsrDevice GetTrackPrefix:TrackID];

BOOL SetTrackSuffix: (short)TrackID: (unsigned char) Prefix

Description: Characters can be added to the end of track data. These can be special characters to identify the specific track to the receiving host, or any other character string. Up to six ASCII characters can be defined.

Parameter: short TrackID=
0x01: TRACK1,
0x02: TRACK2,
0x03: TRACK3,

unsigned char c=maximum 6 characters

Return: True: Set success, False: Failure

Example: BOOL result=[BTmsrDevice SetTrackPSuffix:TrackID:c];

Byte GetTrackSuffix: (short) TrackID

Description: Get the Suffix of the TrackID Track

Parameter: short TrackID=
0x01: TRACK1,
0x02: TRACK2,
0x03: TRACK3,

Return: Suffix of the TrackID track

Example: Byte result=[BTmsrDevice GetTrackSuffix:TrackID];

NSString* GetCardType:

Description: Get the card type
Parameter: Nil
Return: “ISO”(0x80),
“AAMVA”(0x81),
“raw”(0x84)
Example: NSString* result=[BTmsrDevice GetCardType];

NSString* GetTrack1:

Description: Get data of Track 1
Parameter: Nil
Return: Data of Track 1
Example: NSString* result=[BTmsrDevice GetTrack1];

NSString* GetTrack2:

Description: Get data of Track 2
Parameter: Nil
Return: Data of Track 2
Example: NSString* result=[BTmsrDevice GetTrack2];

NSString* GetTrack3:

Description: Get data of Track 3
Parameter: Nil
Return: Data of Track 3
Example: NSString* result=[BTmsrDevice GetTrack3];

NSData* GetTrack1_EncryptedData:

Description: Get Encrypted data of Track 1
Parameter: Nil
Return: Encrypted Data of Track 1
Example: NSData* result=[BTmsrDevice GetTrack1_EncryptedData];

NSData* GetTrack2_EncryptedData:

Description: Get Encrypted data of Track 2
Parameter:
Return: Encrypted Data of Track 2
Example: NSData* result=[BTmsrDevice GetTrack2_EncryptedData];

NSData* GetTrack3_EncryptedData:

Description: Get Encrypted data of Track 3
Parameter: Nil
Return: Encrypted Data of Track 3

BT Mag SDK User Manual

Example: NSData* result=[BTmsrDevice GetTrack3_EncryptedData];

NSData* GetKSN:

Description: Get KSN of the device

Parameter: Nil

Return: KSN

Example: NSData* result=[BTmsrDevice GetKSN];

NSData* GetSessionID:

Description: Get Session ID of the device

Parameter: Nil

Return: Session ID

Example: NSData* result=[BTmsrDevice GetSessionID];

+(void) enableLogging:

Description: Set SDK console log enable or disable.

Parameter: BOOL enable

Return: Nil

Example: [BTmsr enableLogging : YES];