Shanghai Fudan Microelectronics Group Company Limited

复旦微电子

FM13DT160 App

SDK Interface Document Description

2020.09

SHANGHAI FUDAN MICROELECTRONICS GROUP CO., LTD PRODUCT BEST SUITED TO THE CUSTOMER'S APPLICATION; THEY DO NOT CONVEY ANY LICENSE UNDER ANY INTELLECTUAL PROPERTY RIGHTS, OR ANY OTHER RIGHTS, BELONGING TO SHANGHAI FUDAN MICROELECTRONICS GROUP CO., LTD OR A THIRD PARTY.

WHEN USING THE INFORMATION CONTAINED IN THIS DOCUMENTS, PLEASE BE SURE TO EVALUATE ALL INFORMATION AS A TOTAL SYSTEM BEFORE MAKING A FINAL DECISION ON THE APPLICABILITY OF THE INFORMATION AND PRODUCTS.

PURCHASERS ARE SOLELY RESPONSIBLE FOR THE CHOICE, SELECTION AND USE OF THE SHANGHAI FUDAN MICROELECTRONICS GROUP CO., LTD PRODUCTS AND SERVICES DESCRIBED HEREIN, AND SHANGHAI FUDAN MICROELECTRONICS GROUP CO., LTD ASSUMES NO LIABILITY WHATSOEVER RELATING TO THE CHOICE, SELECTION OR USE OF THE SHANGHAI FUDAN MICROELECTRONICS GROUP CO., LTD PRODUCTS AND SERVICES DESCRIBED HEREIN. UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED SHANGHAI FUDAN MICROELECTRONICS GROUP CO., LTD REPRESENTATIVE, SHANGHAI FUDAN MICROELECTRONICS GROUP CO., LTD PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE.

FUTURE ROUTINE REVISIONS WILL OCCUR WHEN APPROPRIATE, WITHOUT NOTICE. CONTACT SHANGHAI FUDAN MICROELECTRONICS GROUP CO., LTD SALES OFFICE TO OBTAIN THE LATEST SPECIFICATIONS AND BEFORE PLACING YOUR PRODUCT ORDER. PLEASE ALSO PAY ATTENTION TO INFORMATION PUBLISHED BY SHANGHAI FUDAN MICROELECTRONICS GROUP CO., LTD BY VARIOUS MEANS, INCLUDING SHANGHAI FUDAN MICROELECTRONICS GROUP CO., LTD HOME BLOCK (HTTP://WWW.FMSH.COM/).

PLEASE CONTACT SHANGHAI FUDAN MICROELECTRONICS GROUP CO., LTD LOCAL SALES OFFICE FOR THE SPECIFICATION REGARDING THE INFORMATION IN THIS DOCUMENT OR SHANGHAI FUDAN MICROELECTRONICS GROUP CO., LTD PRODUCTS.

Trademarks

Shanghai Fudan Microelectronics Group Co., Ltd name and logo, the “复旦” logo are trademarks or registered trademarks of Shanghai Fudan Microelectronics Group Co., Ltd or its subsidiaries in China.

CONTENTS

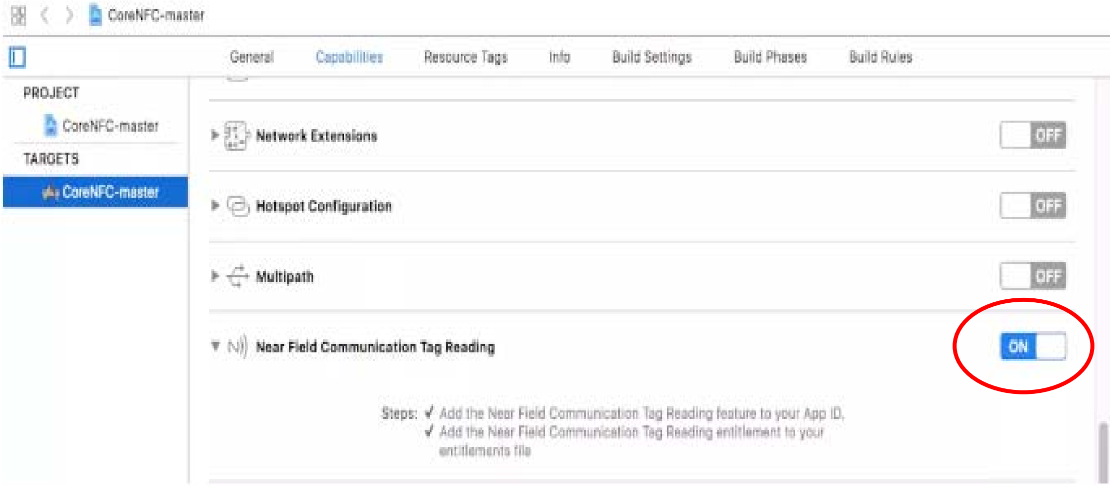
1 IOS

1.1 Static Library Use of process instructions

（1）Creat new OC project

（2）Add lib DT160SDK.a(static library) and NFCTagHelper.h(Interface header file) to the project

（3）Open the Near Field Communication Tag Reading option in your Project TARGETS->Capabilities



Network Extensions OFF

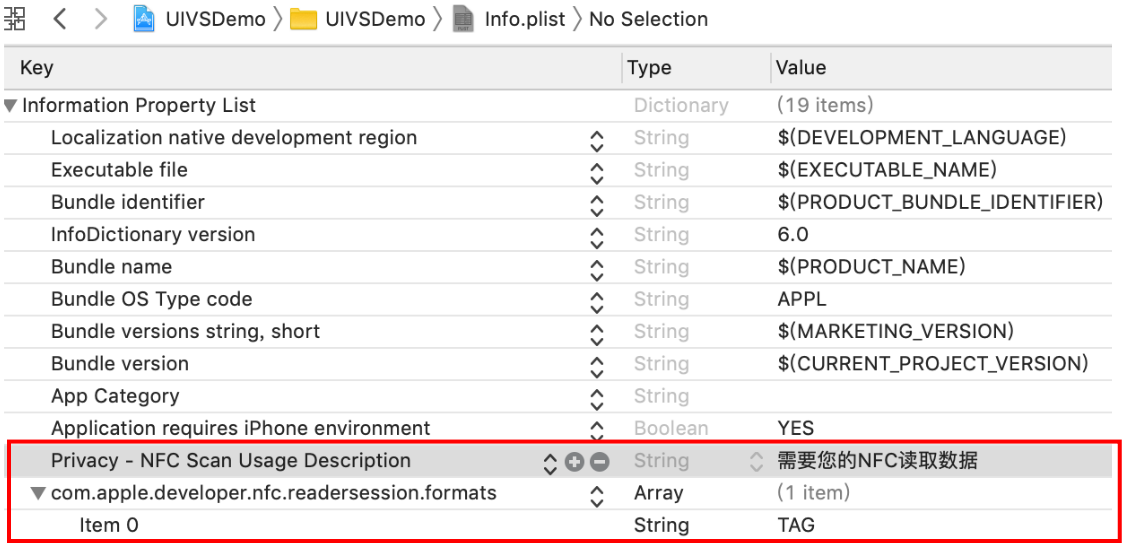
Hotspot Configuration OFF

Multipath OFF

Near Field Communication Tag Reading ON

Steps: ✓ Add the Near Field Communication Tag Reading feature to your App ID.
✓ Add the Near Field Communication Tag Reading entitlement to your entitlements file

（4）Add to your Project info.plist: Privacy -NFC Scan Usage Description NFC usage description and com.apple.developer.nfc.readersession.formats



Key	Type	Value
▼ Information Property List	Dictionary	(19 items)
Localization native development region	String	$(DEVELOPMENT_LANGUAGE)
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	$(MARKETING_VERSION)
Bundle version	String	$(CURRENT_PROJECT_VERSION)
App Category	String	
Application requires iPhone environment	Boolean	YES
Privacy - NFC Scan Usage Description	String	需要您的NFC读取数据
▼ com.apple.developer.nfc.readersession.formats	Array	(1 item)
Item 0	String	TAG

1.2 Interface description

1.2.1 Get SDK version number

```
-(NSString *)getLibVersion
->Returns SDK version number
```

1.2.2 Single measurement (field strength,voltage,temperature)

```
-(void)getBasicData：(void (^)(MeasureMsg *resultData))onComplete
```

Parameter Description：onComplete Callback method after scanning

Callback data description：

```
resultData.isSuccess        //Operational results，YES or NO
resultData.uid              //tag uID
resultData.fieldValue       //field strength
resultData.tempValue        //temperature
resultData.voltageValue     //voltage
resultData.message          //Error message，When isSuccess is NO, use
```

1.2.3 Check WakeUp State

```
-(void)checkWakeUp：(void (^)(MeasureMsg *resultData))onComplete
```

Parameter Description：onComplete Callback method after scanning

Callback data description：

```
resultData.isSuccess     //Operational results，YES or NO
resultData.uid           //tag uID
resultData.isWakeup      //WakeupState,YES:alreadyWakeup,NO:noWakeup
resultData.message       //Error message,When isSuccess is NO,use
```

1.2.4 Sleep

```
-(void)doSleep：(void (^)(MeasureMsg *resultData))onComplete
```

Parameter Description：onComplete Callback method after scanning

Callback data description：

```
resultData.isSuccess     //Operational results，YES or NO
resultData.uid           //tag uID
resultData.message       //Error message,When isSuccess is NO,use
```

1.2.5 Initializ UHF(Initialize Reg)

```
-(void)initUHF：(void (^)(MeasureMsg *resultData))onComplete
```

Parameter Description：onComplete Callback method after scanning

Callback data description：

```
resultData.isSuccess     //Operational results，YES or NO
resultData.uid           //tag uID
resultData.message       //Error message,When isSuccess is NO,use
```

1.2.6 LED ON

```
-(void)turnOnLED： (void (ˆ)(MeasureMsg *resultData))onComplete
Parameter Description：onComplete Callback method after scanning
Callback data description：
resultData.isSuccess     //Operational results，YES or NO
resultData.uid           //tag uID
resultData.message       //Error message,When isSuccess is NO,use
```

1.2.7 LED OFF

```
-(void)turnOffLED： (void (ˆ)(MeasureMsg *resultData))onComplete
Parameter Description：onComplete Callback method after scanning
Callback data description：
resultData.isSuccess     //Operational results，YES or NO
resultData.uid           //tag uID
resultData.message       //Error message,When isSuccess is NO,use
```

1.2.8 Start Logging

```
-(void)startLogging：
(NSInteger)delayMinutes intervalSeconds:(NSInteger)intervalSeconds
loggingCount:(NSInteger) loggingCount  minTemperature:(NSInteger)
minTemperature  maxTemperature:(NSInteger) maxTemperature
(void (ˆ)(MeasureMsg *resultData))onComplete
Parameter Description：
delayMinutes:    //(Unit:Minutes) Delay Time to start
intervalSeconds: //(Unit:seconds)
                 //The interval of temperature measurement in logging process
loggingCount:    //(Unit:times) temperature measurement Points
minTemperature:  //(Unit:℃)
//The temperature limit threshold for logging temperature measurement, the results
//below this temperature will be included in the Summary results.
maxTemperature:   //(Unit:℃)
//The temperature limit threshold for logging temperature measurement, the results
//above this temperature will be included in the Summary results.
onComplete Callback method after scanning
Callback data description：
resultData.isSuccess     //Operational results，YES or NO
resultData.uid           //tag uID
resultData.message       //Error message,When isSuccess is NO,use
```

1.2.9 Stop logging

```
-(void)stopLogging： (void (ˆ)(MeasureMsg *resultData))onComplete
Parameter Description：onComplete Callback method after scanning
Callback data description：
```

```
resultData.isSuccess    //Operational results，YES or NO
resultData.uid          //tag uID
resultData.message      //Error message,When isSuccess is NO,use
```

1.2.10 Read Logging Temperature Data

```
-(void)getLoggingResult: (void (^)(LoggingMsg *resultData))onComplete
Parameter Description：onComplete Callback method after scanning
Callback data description：
resultData.isSuccess    //Operational results，YES or NO
resultData.uid          //tag uID
resultData.message      //Error message,When isSuccess is NO,use
LoggingMsg the data structure see the 1.3 callback structure description
```

1.2.11 Custom Send Instruction

```
-(void)sendInstruct:(NSString*)instruction(void(^)(MeasureMsg
*resultData))onComplete
Parameter Description：Instruction:Custom Instruction
onComplete Callback method after scanning
Callback data description：
resultData.isSuccess  //Operational results，YES or NO
resultData.uid       //tag uID
resultData.message   // When the isSuccess is NO, it represents the error message
                     //and when the isSuccess is YES, it represents the returned data
```

1.3 Description of callback structure

```
@interface MeasureMsg: NSObject
// Operation result, YES success, NO failure
@property(nonatomic, assign) BOOL isSuccess;
//Tag UID
@property(nonatomic, copy) NSString *uid;
// Field strength
@property(nonatomic, copy) NSString *fieldValue;
// Temperature
@property(nonatomic, copy) NSString *tempValue;
// Voltage
@property(nonatomic, copy) NSString *voltageValue;
// Wake state, YES wake NO sleep
@property(nonatomic, assign) BOOL isWakeup;
// Abnormal error information used when isSuccess is NO
@property(nonatomic, copy) NSString *message;
@end
@interface LoggingMsg: NSObject
```

```
// Operation result, YES success, NO failure
@property(nonatomic, assign) BOOL isSuccess;
//Tag UID
@property(nonatomic, copy) NSString *uid;
//Measurement State:
//0 delayed start-up , 1 in logging, 2 logging anomaly ,3 logging completed
@property(nonatomic, assign) LoggingStatus opStatus;
//Timing temperature measurement start time, timestamp
@property(nonatomic, assign) NSInteger startTime;
// Total Temperature Measurement Point
@property(nonatomic, assign) NSInteger totalCount;
// Current Temperature Measurement Point
@property(nonatomic, assign) NSInteger recordedCount;
// Time delay in starting logging temperature measurement( unit:minutes)
@property(nonatomic, assign) NSInteger delayMinutes;
// Time interval, unit: seconds
@property(nonatomic, assign) NSInteger intervalSeconds;
// Current minimum temperature data
@property(nonatomic, assign) CGFloat recordedMinimum

// Current Maximum temperature data
@property(nonatomic, assign) CGFloat recordedMaximum
// The temperature Min limit threshold of logging temperature measurement, //the
results below this temperature will be included in the results of //exceeding the
limit
@property(nonatomic, assign) CGFloat validMinimum
// The temperature Max limit threshold of logging temperature measurement, //the
results above this temperature will be included in the results of //exceeding the
limit
@property(nonatomic, assign) CGFloat validMaximum
// Number of temperature measurements below the min limit threshold
@property(nonatomic, assign) NSInteger overLowCount
// Number of temperature measurements above the max limit threshold
@property(nonatomic, assign) NSInteger overHighCount
// Data detail for logging temperature measurement, array
@property(nonatomic, strong) NSMutableArray *temperaturesArray;
//Abnormal error information used when isSuccess is NO
@property(nonatomic, copy) NSString *message;

@end
```

2 Android

2.1 Flow Description for Static Library

Configuration NFC permissions:

```
<uses-permission android:name=″android.permission.NFC″></uses-permission>
```

Single case； GeneralNFC.getInstance()；

2.2 Interface description

2.2.1 Get SDK version number

```
public String getLibVersion()
```

->Returns SDK version number

2.2.2 Start NFC Reader Mode

```
public void startNFCReaderMode(Activity activity)
```

Parameter Description: activity currently displayed

You need to call this method before using the following method. To use it first， do not call it again when using the following method， it will fail.

2.2.3 Close NFC Reader Mode

```
public void closeNFCReaderMode()
```

No parameters

Call when the page is closed or NFC functionality is not required

2.2.4 External Call Tag Methods

```
public void setTag(Tag tag)
```

tag， external call method fetch TAG,2.2.2 method does not need to be used

2.2.5 Single measurement (field strength,voltage,temperature)

```
public void getBasicData(final OnResultCallback callback)
```

Parameter Description：OnResultCallback method after scanning

Callback data description：

```
void onResult (boolean status, String[]response)；
status               //Operational results，true Success，false Fail
response:The data are ordered as follows
fieldValue           // field strength
tempValue            // temperature
voltageValue         // voltage
```

2.2.6 Check WakeUp State

```
public void checkWakeUp(final OnResultCallback callback)
Parameter Description：OnResultCallback method after scanning
Callback data description：
void onResult (boolean status, String[] response);
status              // Operational results，true Success，false Fail
response            //null
```

2.2.7 Sleep

```
public void doSleep(final OnResultCallback callback)
Parameter Description：OnResultCallback method after scanning
Callback data description：
void onResult (boolean status, String[] response);
status              //Operational results，true Success，false Fail
response            //null
```

2.2.8 Initializ UHF (Initialize Reg)

```
public void initUHF(final OnResultCallback callback)
Parameter Description：OnResultCallback method after scanning
Callback data description：
void onResult (boolean status, String[] response);
status              //Operational results，true Success，false Fail
response            //null
```

2.2.9 LED ON

```
public void turnOnLED(final OnResultCallback callback)
Parameter Description：OnResultCallback method after scanning
Callback data description：
void onResult (boolean status, String[] response);
status              // Operational results，true Success，false Fail
response            // null
```

2.2.10 LED OFF

```
public void turnOffLED(final OnResultCallback callback)
Parameter Description：OnResultCallback method after scanning
Callback data description：
void onResult (boolean status, String[] response);
status             // Operational results，true Success，false Fail
response           //null
```

2.2.11 Check Status

```
public void checkStatus(final OnResultCallback callback)
Parameter Description: OnResultCallback method after scanning
Callback data description:
void onResult (boolean status, String[] response);
status        // Operating results: true: in the Logging process;
              // false :in the non-Logging process
response     // null
```

2.2.12 Start Logging

```
public void startLogging(int delayMinutes, int intervalSeconds, int loggingCount, int
minTemperature, int maxTemperature, final OnResultCallback callback)
Parameter Description:
delayMinutes      //(Unit:Minutes) Delay Time to start
intervalSeconds   //(Unit:seconds)The interval of temperature
                   // measurement in logging process
loggingCount      //(Unit: times) temperature measurement Points
minTemperature    //(Unit:℃)The temperature limit threshold for logging
                  //temperature measurement, the results below this
                  //temperature will be included in the Summary results
maxTemperature    //(Unit:℃)The temperature limit threshold for logging
                  //temperature measurement, the results above this
                  // temperature will be included in the Summary results.
OnResultCallback  // OnResultCallback method after scanning
Callback data description:
void onResult (boolean status, String[] response);
status            // Operational results, true Success, false Fail
response          //null
```

2.2.13 Stop Logging

```
public void stopLogging(final OnResultCallback callback)
Parameter Description: OnResultCallback method after scanning
Callback data description:
void onResult (boolean status, String[] response);
status              // Operational results, true Success, false Fail
response            //null
```

2.2.14 Read Logging Temperature Data

```
public void getLoggingResult(final OnResultCallback callback)
Parameter Description：OnResultCallback method after scanning

Callback data description：
void onResult (boolean status, String[] response);
status           //Operational results，true Success，false Fail
The response data are in the following order:
opStatus         //Measurement State, 0:delayed start-up , 1:in logging,
                 //2:logging anomaly ,3: logging completed
startTime        //Timing temperature measurement start time, timestamp
totalCount           //Total Temperature Measurement Point
recordedCount        //Current Temperature Measurement Point
delayMinutes         //Time delay in starting logging (unit:minutes)
intervalSeconds      // Time interval(unit: seconds)
recordedMinimum      // Current minimum temperature data
recordedMaximum      // Current maximum temperature data
validMinimum         // Low temperature  Limit
validMaximum         // High temperature  Limit
overLowCount         // Number of temperature below the min limit
overHighCount        // Number of temperature above the max limit
The temperature data is stored in the remaining array
```

2.2.15 Custom Send Instruction

```
public void sendInstruct(final OnResultCallback callback)
Parameter Description：OnResultCallback method after scanning
Callback data description：
void onResult (boolean status, String[] response);
status            //Operational results，true Success，false Fail
response          // The data sent back is the first in the array
```

2.2.16 Configure Original data mode

```
public void configPrimitiveMode (final OnResultCallback callback)
Parameter Description：OnResultCallback method after scanning
Callback data description：
void onResult (boolean status, String[] response);
status             //Operational results，true Success，false Fail
response           // null
```

2.2.17 Configure Normal data mode

```
public void configStandardMode (int mode,final OnResultCallback callback)
Parameter Description：
mode : Configuration mode
0:2 decimal places
1:3 decimal places
OnResultCallback method after scanning
Callback data description：
void onResult (boolean status， String[] response);
status              //Operational results，true Success，false Fail
response           // null
```

2.2.18 Setting Password

```
public void settingPassword (String pwd,byte[]address,
final OnResultCallback callback)
Parameter Description：
pwd:4 byte
address: 2 byte Address value to be encrypted
OnResultCallback method after scanning
Callback data description：
void onResult (boolean status， String[] response);
status              //Operational results，true Success，false Fail
response           // null
```

2.2.19 Update password

```
public void updatePassword (String oldPwd,String newPwd,byte[] address,
final OnResultCallback callback)
Parameter Description：
oldPwd:old password,4 byte
newPwd:new password,4byte
address: 2 byte Address value to be encrypted
OnResultCallback method after scanning
Callback data description：
void onResult (boolean status， String[] response);
status              //Operational results，true Success，false Fail
response           // null
```

2.3 Description of callback structure

```
public interface OnResultCallback {
    /**
     * Result callback
     * @param status
     * @param response Data information, array of var length
     */
    void onResult(boolean status,String... response);
    /**
     * Failed result callback
     * @param errorMsg Error Information
     */
    void onFailed(String errorMsg);
}
```

3 Revision history

Rev	Release date	Pages	Modifications
1.0	August 2020	14	Initial release
1.1	Sep 2020	16	Add some features to Android phone

Sales and Service

Shanghai Fudan Microelectronics Group Co., Ltd.
Address: Bldg No. 4, 127 Guotai Rd,
Shanghai City China.
Postcode: 200433
Tel: (86-021) 6565 5050
Fax: (86-021) 6565 9115

Shanghai Fudan Microelectronics (HK) Co., Ltd.
Address: Unit 506, 5/F., East Ocean Centre, 98 Granville Road, Tsimshatsui East, Kowloon, Hong Kong
Tel: (852) 2116 3288   2116 3338
Fax: (852) 2116 0882

Beijing Office
Address: Room 423, Bldg B, Gehua Building,
1 QingLong Hutong, Dongzhimen Alley north Street,
Dongcheng District, Beijing City, China.
Postcode: 100007
Tel: (86-010) 8418 6608
Fax: (86-010) 8418 6211

Shenzhen Office
Address: Room.1301, Century Bldg, No. 4002, Shengtingyuan Hotel, Huaqiang Rd (North),
Shenzhen City, China.
Postcode: 518028
Tel: (86-0755) 8335 0911   8335 1011   8335 2011 8335 0611
Fax: (86-0755) 8335 9011

Shanghai Fudan Microelectronics (HK) Ltd Taiwan Representative Office
Address: Unit 1225, 12F., No 252, Sec.1 Neihu Rd., Neihu Dist., Taipei City 114, Taiwan
Tel : (886-2) 7721 1889
Fax: (886-2) 7722 3888

Shanghai Fudan Microelectronics (HK) Ltd Singapore Representative Office
Address : 237, Alexandra Road, #07-01 The Alexcier, Singapore 159929
Tel : (65) 6472 3688
Fax: (65) 6472 3669

Shanghai Fudan Microelectronics Group Co., Ltd NA Office
Address :2490 W. Ray Road Suite#2
Chandler, AZ 85224 USA
Tel : (480) 857-6500 ext 18

Web Site: http://www.fmsh.com/