AsReader GUN iOS Demo

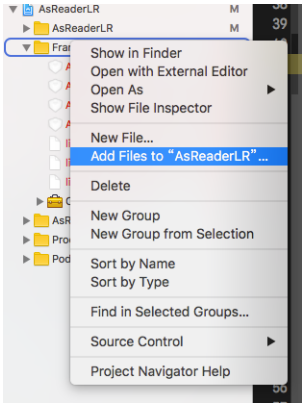
AsReader GUN iOS Demo App Coding Guide

AsReader, Inc.
2017-02-22

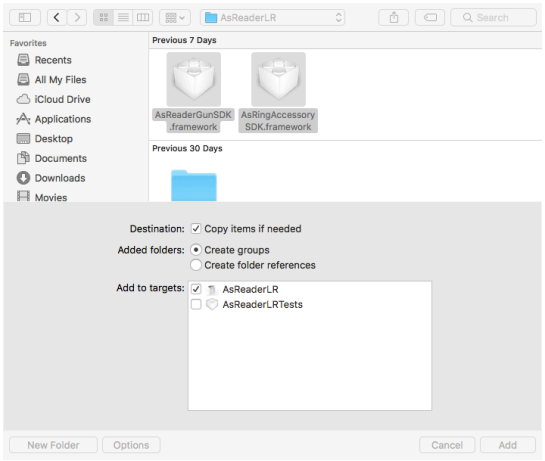
1.Create Project

1.1 Import SDKs

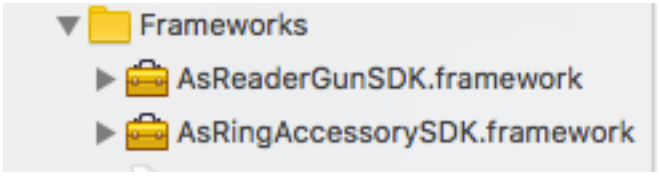
1.1.1. Choose “Add File to..” to add SDK files to Project.



1.1.2. Select both AsRing and AsReader SDK，make sure “Copy items if needed” selected, click “Add” to add SDKs.



1.1.3 The result of above should be look like as follows:



1.2 Config the plist file

Add this item in Info.plist : Supported external accessory protocols
Set the value to : jp.co.asx.asring.plus

2. Start Coding

2.1 import the header in Appdelegate or else where you use AsReader.

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

2.2 Define properties if needed to hold the instance of AsReader instances on both communication lay and application layer.

```
@property (strong, nonatomic) AsReaderGUN *mAsReaderGUN;
```

```
@property (strong, nonatomic) AsReader *mReader;
```

2.3 Initialize the AsReaderGUN instance.

```
appDelegate=(AppDelegate*)[[UIApplication    sharedApplication]
delegate]
```

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

```
appDelegate.mAsReaderGUN.deviceModel= @"com.asreader.gun";
```

```
-(void)viewWillAppear:(BOOL)animated {
    [super viewWillAppear:animated];
    NSNotificationCenter *center = [NSNotificationCenter
defaultCenter];
    [center addObserver:self
selector:@selector(AsReaderGUNConnected:)
name:@"AsReaderGUNConnected" object:nil];
    [center addObserver:self
selector:@selector(AsReaderGUNDisconnected:)
name:@"AsReaderGUNDisconnected" object:nil];
}
```

2.4 Enable receiving notification when AsReader connected to iOS.

2.5 remove notification listener in “viewDidDisappear”

```
- (void)viewDidDisappear:(BOOL)animated{
    [super viewDidDisappear:animated];
    [[NSNotificationCenter defaultCenter] removeObserver:self
name:@"AsReaderGUNConnected" object:nil];
    [[NSNotificationCenter defaultCenter] removeObserver:self
name:@"AsReaderGUNDisconnected" object:nil];
}
```

2.6 Accepting notification when connected

Create AsReader instance(application layer instance) when AsReader connected to iOS device.

```
- (void)AsReaderGUNConnected:(NSNotification *)notification {
    dispatch_async(dispatch_get_main_queue(), ^{
        appDelegate.mReader = [[AsReader alloc]
initWithAsReaderGUN:appDelegate.mAsReaderGUN delegate:self];
    });
}
```

2.7 Accepting notification when disconnected

Destroy the AsReader instance when disconnected.

```
- (void)AsReaderGUNDisconnected:(NSNotification *)notification
{
    dispatch_async(dispatch_get_main_queue(), ^{
        appDelegate.mReader = nil;
    });
}
```

3.Scan Barcode

3.1 Config

```
[appDelegate.mReader setTagDataType:TAG_DATA_TYPE_HEX];
```

```
appDelegate.mReader.isUseKeyAction = [UserData getIsUseKeyAction];
```

```
[appDelegate.mReader setBarcodeMode:YES isKeyAction:YES];
```

3.3 Start to scan

```
[appDelegate.mReader startDecode];
```

3.4 Receive the scanned barcode data

Receive data as NSString

```
- (void)detectBarcode:(BarcodeType)barcodeType
  codeId:(NSString *)codeId barcode:(NSString *)barcode;
```

Receive data as NSData

```
- (void)detectBarcode:(BarcodeType)barcodeType
codeId:(NSString *)codeId barcodeData:(NSData *)barcodeData;
```

4.RFID Inventory

4.1 Start

```
[appDelegate.mReader inventory];
```

4.2 Stop

```
[appDelegate.mReader stop];
```

4.3 Receiving RFID data

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