




Mr.LOOP SDK Release  
1.2

Supervised by  
Mr. LOOP Inc.

April 19, 2017

# Contents

<b>1</b>	<b>Main Page</b>	<b>1</b>
<b>2</b>	<b>Mr.LOOP SDK User Guide</b>	<b>2</b>
<b>3</b>	<b>Device Flowcahrt</b>	<b>4</b>
<b>4</b>	<b>SDK Simple Control Process</b>	<b>5</b>
<b>5</b>	<b>Windows Driver Install Tutorial</b>	<b>6</b>
<b>6</b>	<b>Disable USB suspend for high performance</b>	<b>11</b>
<b>7</b>	<b>Linux Driver Install Tutorial</b>	<b>15</b>
<b>8</b>	<b>macOS Driver Install Tutorial</b>	<b>16</b>
<b>9</b>	<b>Android SDK Install Tutorial</b>	<b>18</b>
<b>10</b>	<b>Mr.LOOP SDK SimpleTransfer Guide</b>	<b>24</b>
<b>11</b>	<b>File Index</b>	<b>25</b>
11.1	File List . . . . .	25
<b>12</b>	<b>File Documentation</b>	<b>26</b>
12.1	mrloopbf_release.h File Reference . . . . .	26
12.1.1	Macro Definition Documentation . . . . .	26
12.1.1.1	MRLOOPBF_SHARED_EXPORT . . . . .	26
12.1.2	Function Documentation . . . . .	26
12.1.2.1	ML_Close() . . . . .	26
12.1.2.2	ML_GetDevGen() . . . . .	27
12.1.2.3	ML_HiddenDebugMsg() . . . . .	27
12.1.2.4	ML_Init() . . . . .	27

12.1.2.5	ML_Receiver()	27
12.1.2.6	ML_SetMode()	28
12.1.2.7	ML_SetSpeed()	28
12.1.2.8	ML_Transfer()	28
12.2	Android API.java File Reference	29
12.2.1	Function Documentation	29
12.2.1.1	CloseDevice()	29
12.2.1.2	getDevices()	29
12.2.1.3	ML_GetDescriptors()	30
12.2.1.4	ML_Receive()	30
12.2.1.5	ML_SetMode()	30
12.2.1.6	ML_SetSpeed()	31
12.2.1.7	ML_Transfer()	31
<b>Index</b>		<b>32</b>

# Chapter 1

## Main Page

Mr.LOOP WiGig Software Development Kits

This library is provided to utilize our devices with limited warranty.

Permission is granted to our clients to use this library for any purpose, including commercial applications freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. This notice may not be removed or altered from any source distribution.

Copyright (C) 2015-2017 Mr.LOOP Cor. Ltd [software@mrloop.com.tw](mailto:software@mrloop.com.tw)



## Chapter 2

# Mr.LOOP SDK User Guide

This guide demonstrates how to utilize our SDK.

Please read Chapter 3 to understand workflow of our hardware. Chapter 4 show its control process of our APIs.

I/O rate depends on USB host-type and CPU deeply. Our devices doesn't work on any virtual machine.

Copyright (C) 2015-2017 Mr.LOOP Inc. All Rights Reserved.



## Technical Specs

### Windows test-case:

- CPU: Intel i5-4200U 2.3 Ghz above
- Chipset: Later than Intel® HM86
- RAM: least 8G
- HD: SSD
- Windows Platform Essential: .Net Framework 4.5

### Linux test-case:

- CPU: Intel i7-4790 2.30 GHz
- RAM: 8 Gb
- HD: SSD
- Kernal: 3.16
- OS: Ubuntu 14.04 LTS

### macOS test-case:

- CPU: Intel Core i7 2.30 GHz
- RAM: 8 Gb
- HD: SSD
- OS: macOS 10.11.6



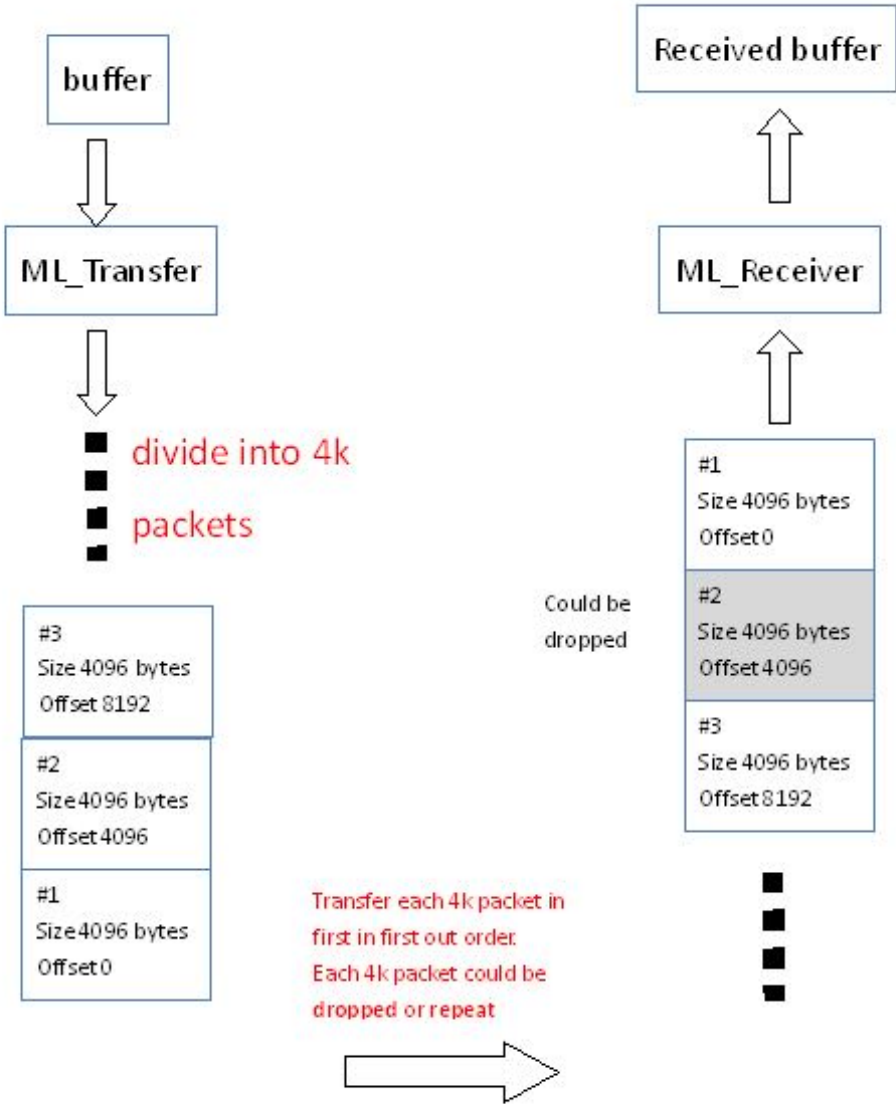
### Android test-case:

- OS: Android 4.2 above
- Privileges: USB

Chapter 3

Device Flowcahrt

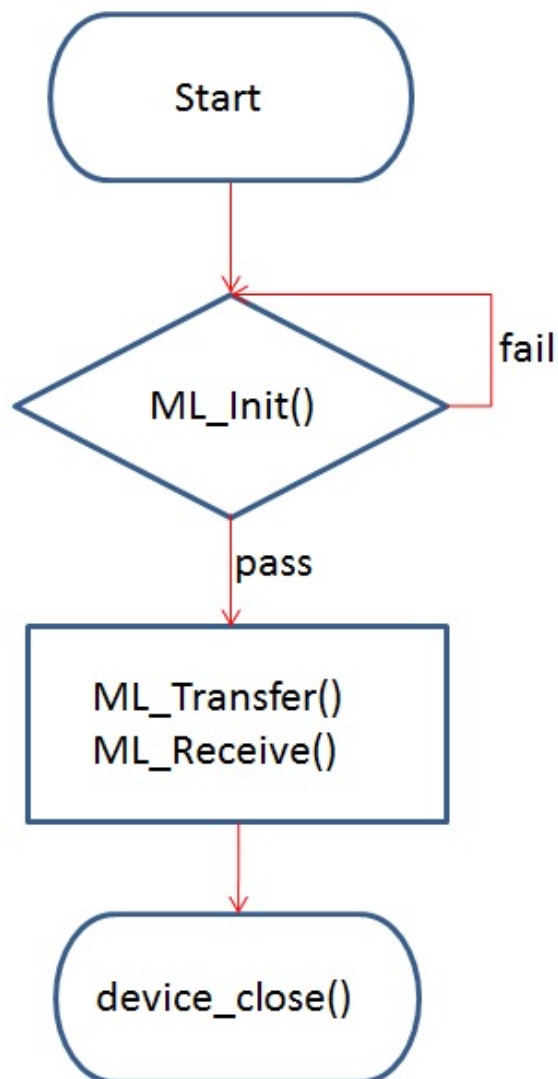
This flowchart describes our device how to work



## Chapter 4

# SDK Simple Control Process

This flowchart describes our SDK how to work





## Chapter 5

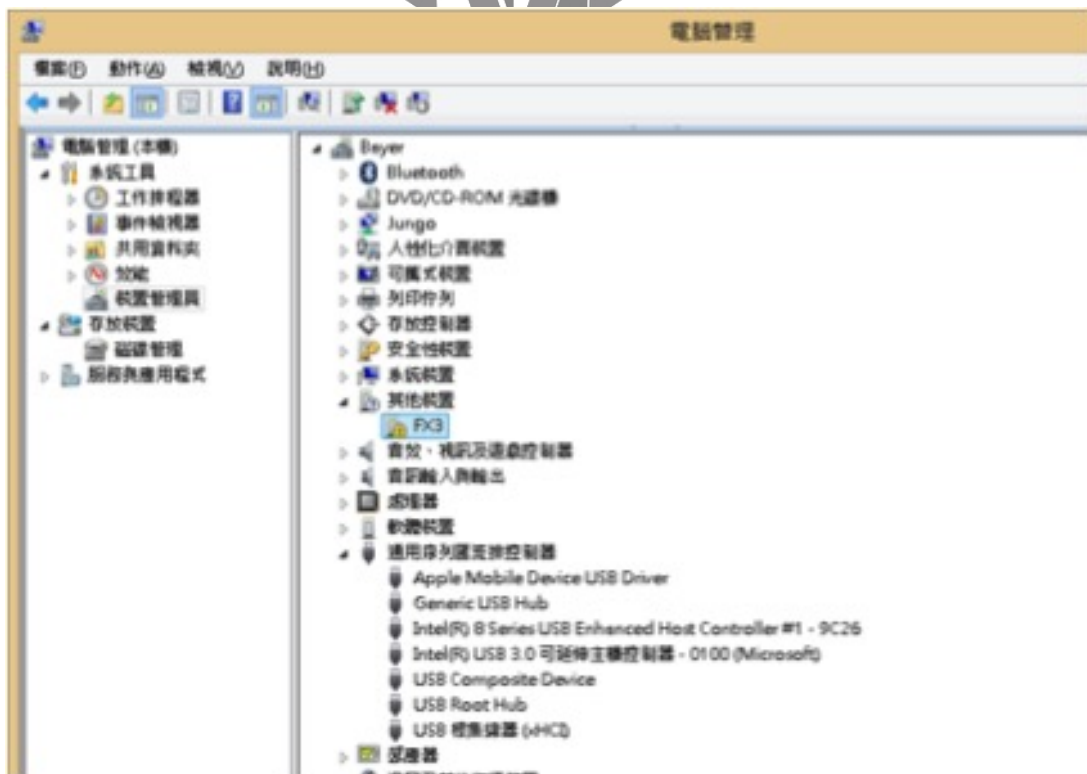
# Windows Driver Install Tutorial

### Using Driver.exe Install

1. Unzip WindowsDrivers.zip
2. run "InstallINF\_x64.exe" for 64bits OS and "InstallINF\_x86.exe" for 32bits OS.
3. If fail, following below guide manually first.

### Manual Install

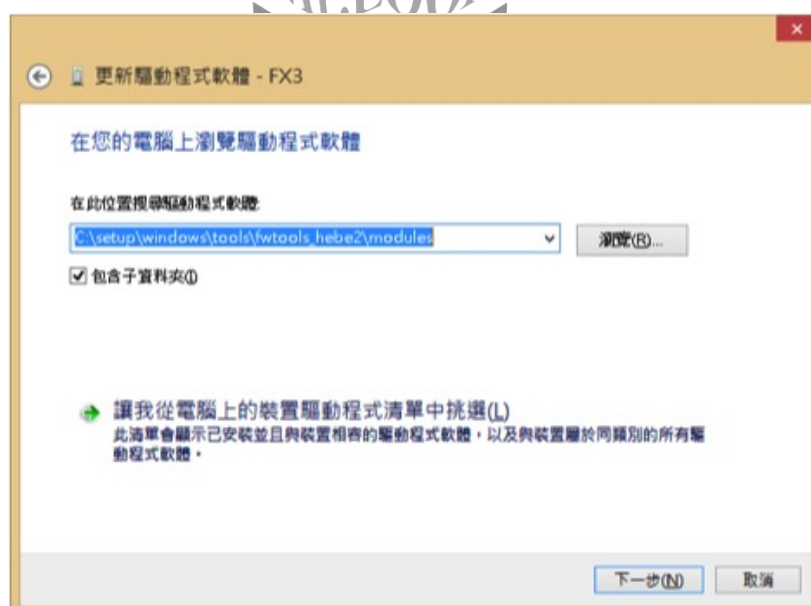
1. Run Setup to install SW  
Run Device Manager. Select “FX3” and mouse right click. Select “Update Driver Software”.



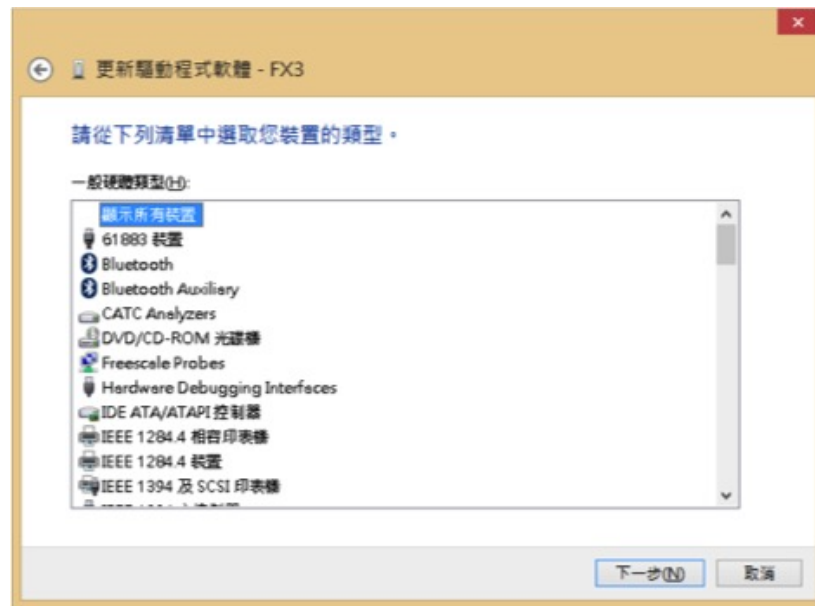
2. Select “Browse my computer for driver software”



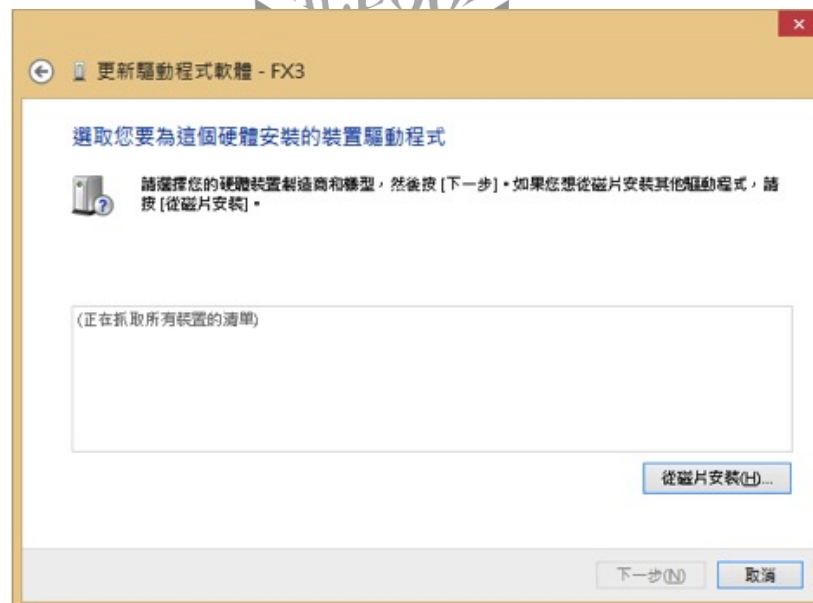
3. Select “Let me pick from a list of device drivers on my computer”



## 4. Select "Next"



## 5. Select "Have disk"



## 6. Select the path of "cyusb3.inf"

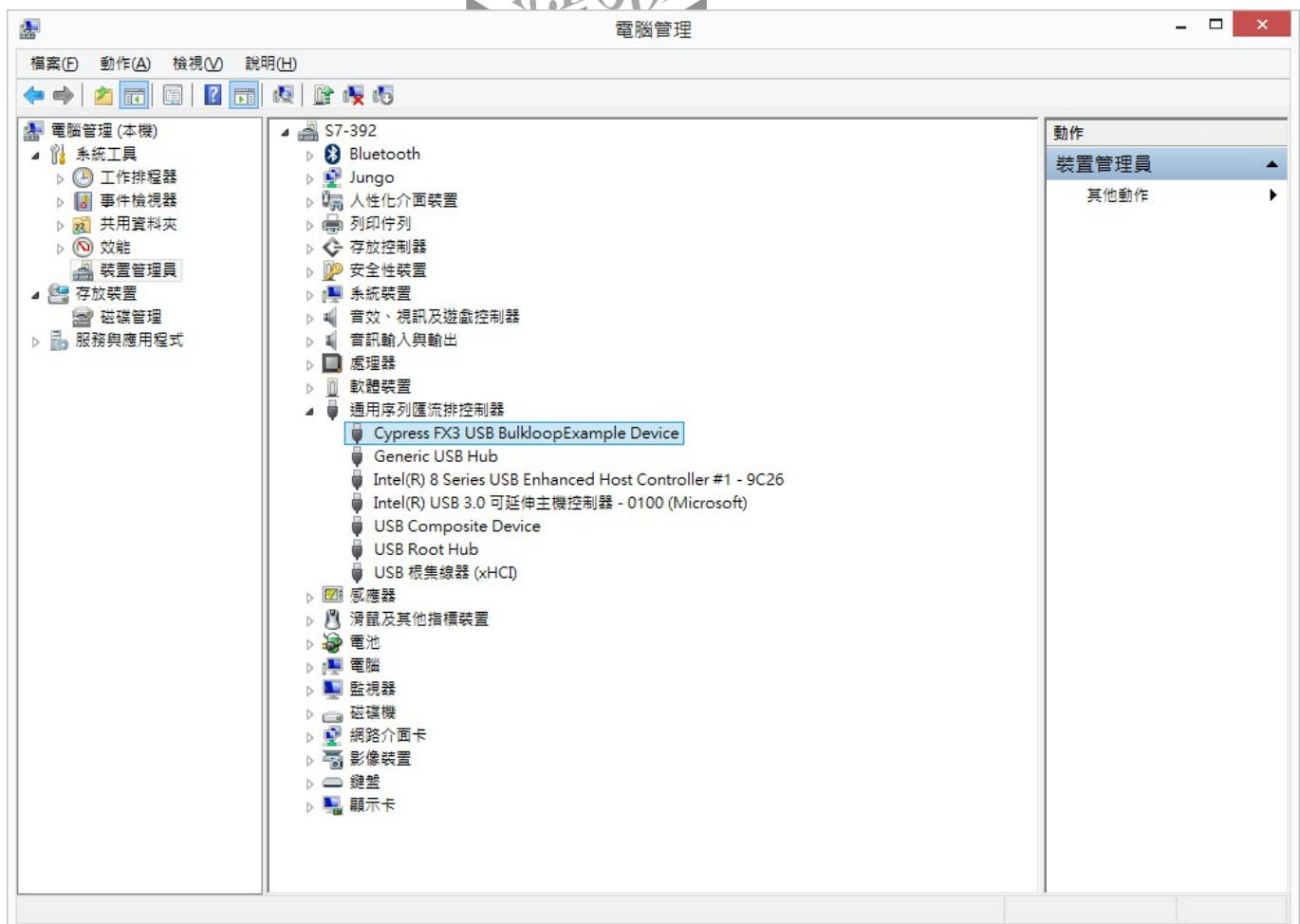
Default path is : "C:\Program Files (x86)\MrLoop\Dongle\driver\\*windows\_↵  
version\\*\*x64\_or\_x86"

"\*" : Depends on windows version, select win7, win8, win8.1. For win10, choose win8.1

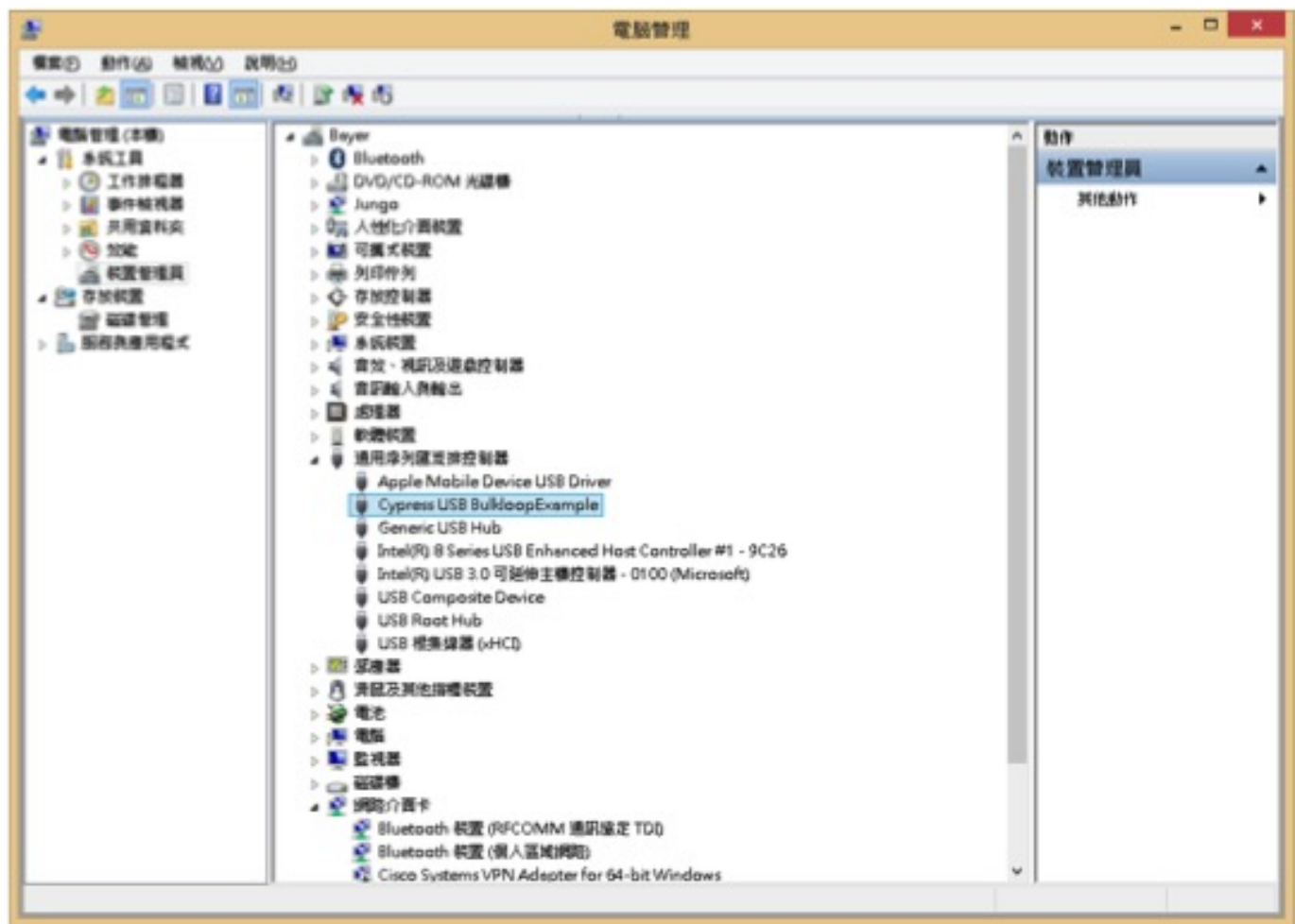
"\*\*": Depends on 32bit OS or 64bit OS



## 7. Select "Cypress USB BulkloopExample" and "Next"



8. When install successfully, you can find “CypressUSB BulkloopExample” on device manager.



## Chapter 6

# Disable USB suspend for high performance

### 1. Click power management



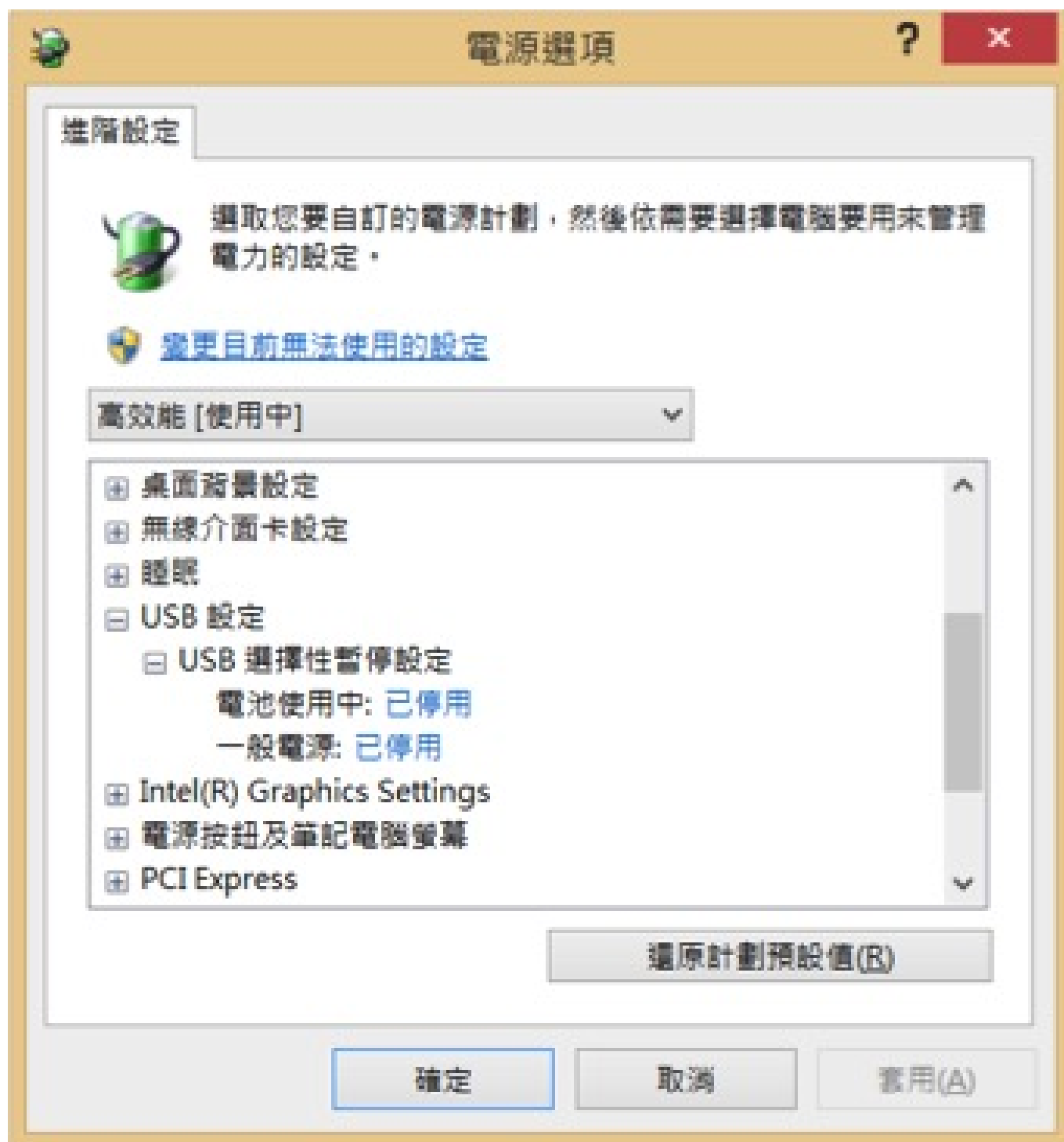
2. Click “High performance”
3. Click “change setting”



4. Click change power setting

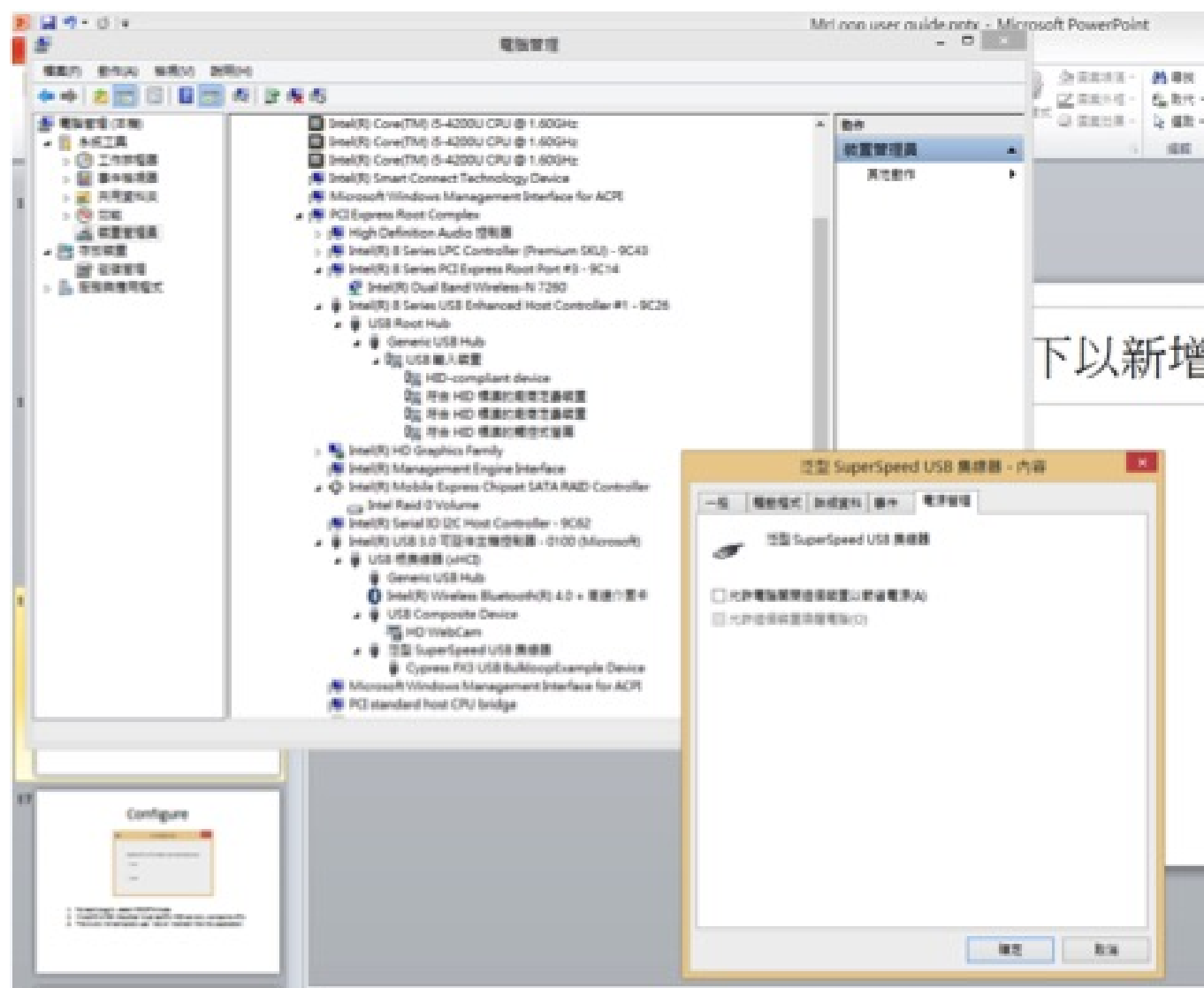


## 5. Disable USB selective suspend





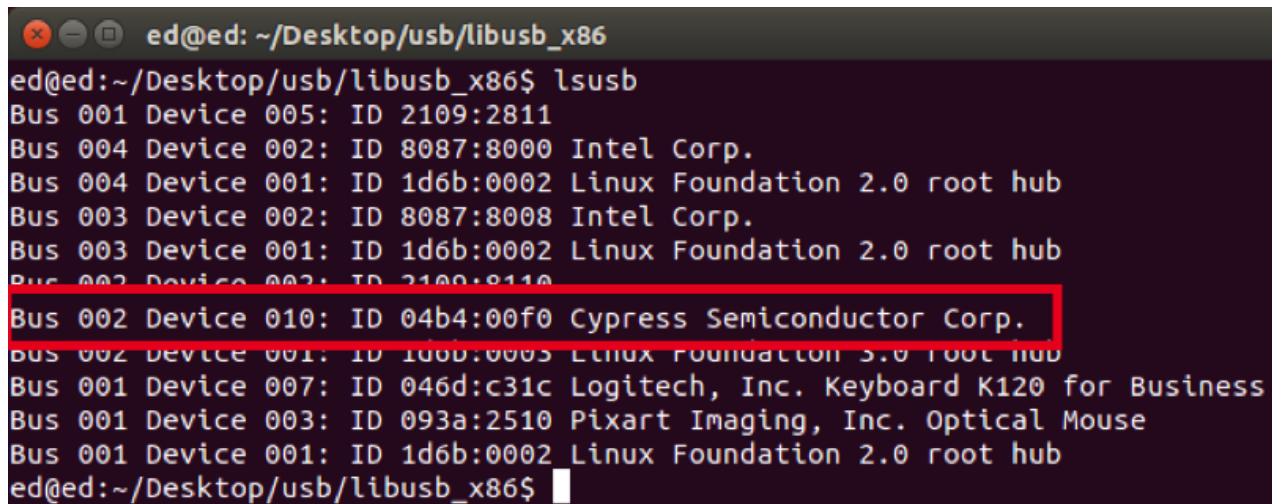
## 6.Disable USB hub power management



## Chapter 7

# Linux Driver Install Tutorial

1. Open terminal and sh ./install.sh
2. Support CPU architecture: x86\_64
3. Plug-in WiGig dongle
4. open terminal enter "lsusb" and find "Cypress Semiconductor Corp."

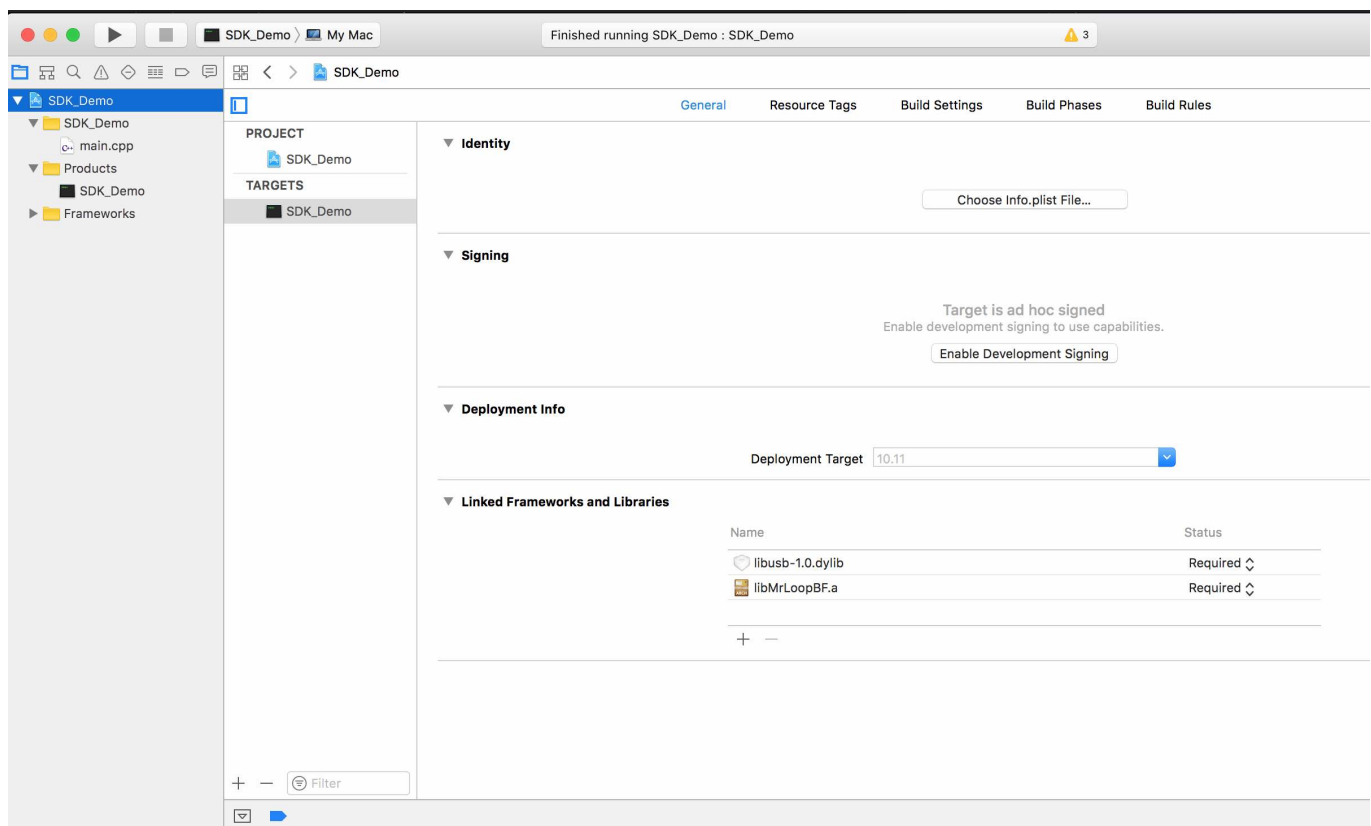


```
ed@ed: ~/Desktop/usb/libusb_x86
ed@ed:~/Desktop/usb/libusb_x86$ lsusb
Bus 001 Device 005: ID 2109:2811
Bus 004 Device 002: ID 8087:8000 Intel Corp.
Bus 004 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 003 Device 002: ID 8087:8008 Intel Corp.
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 002: ID 2109:2811
Bus 002 Device 010: ID 04b4:00f0 Cypress Semiconductor Corp.
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 007: ID 046d:c31c Logitech, Inc. Keyboard K120 for Business
Bus 001 Device 003: ID 093a:2510 Pixart Imaging, Inc. Optical Mouse
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
ed@ed:~/Desktop/usb/libusb_x86$
```

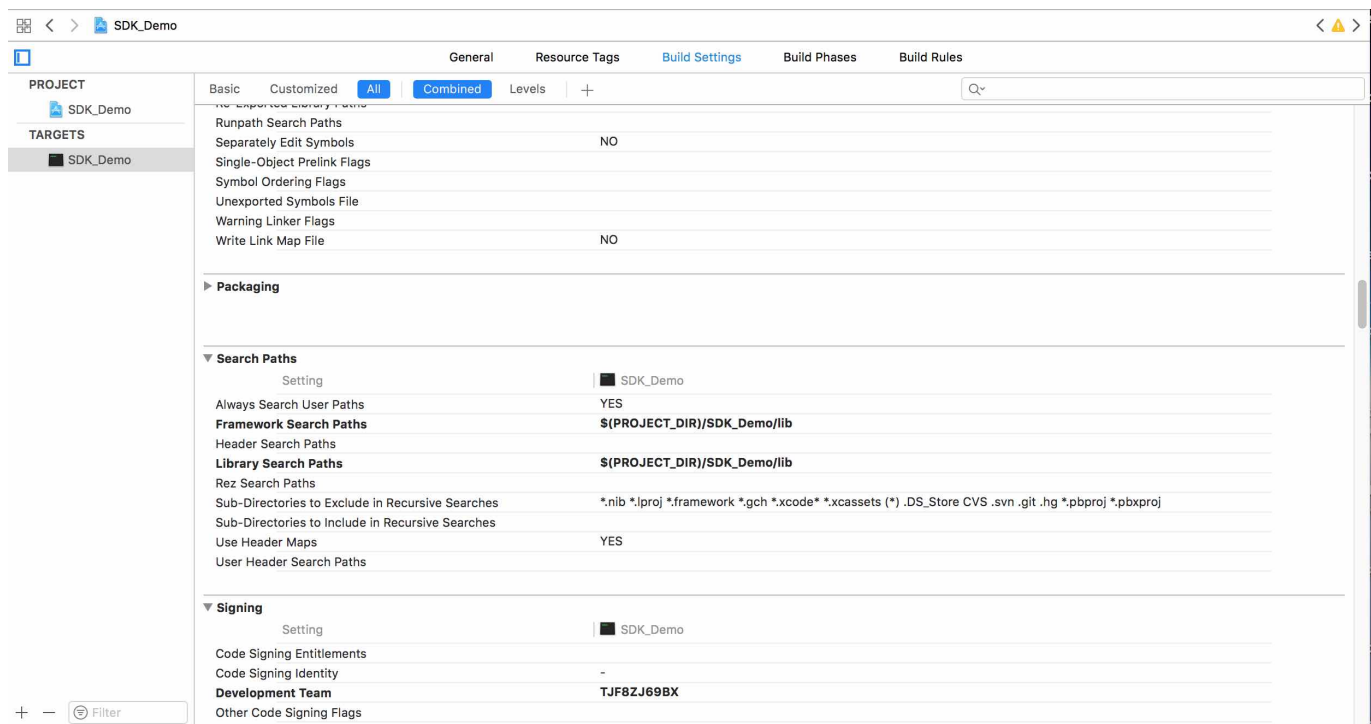
## Chapter 8

# macOS Driver Install Tutorial

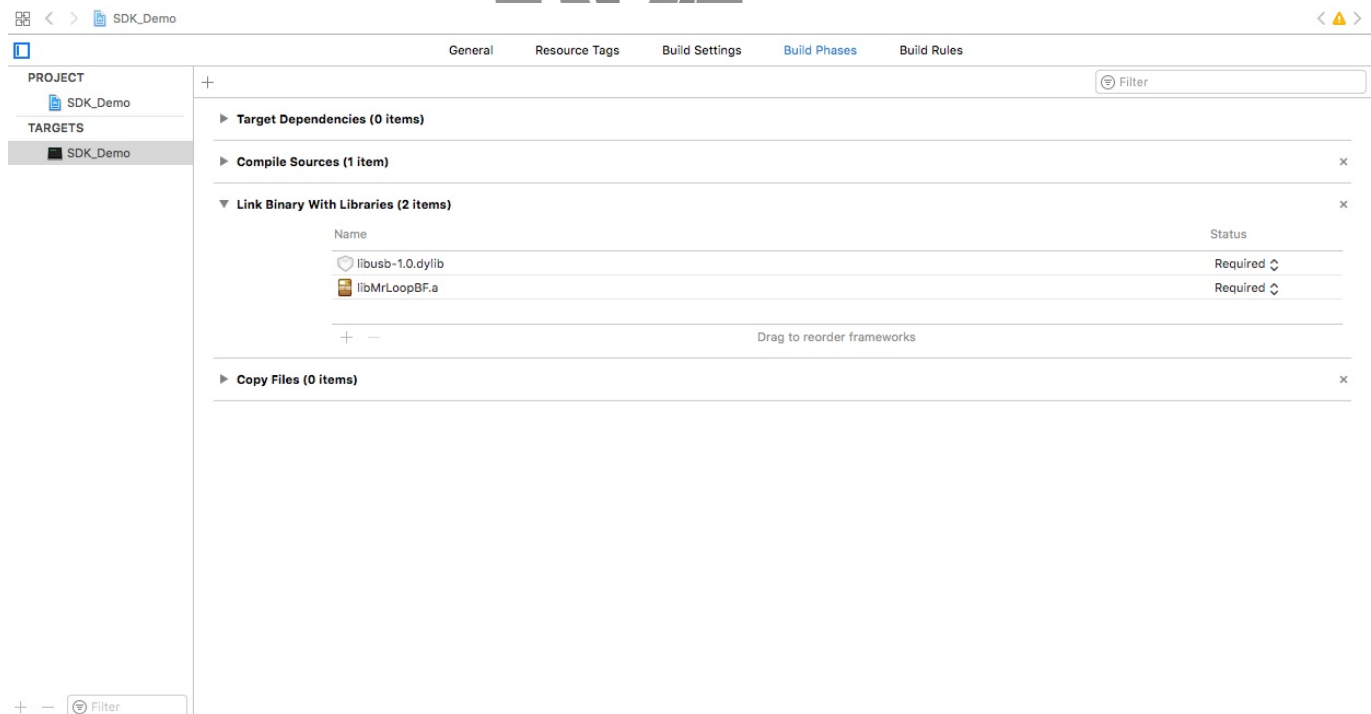
1. Unzip macOS\_x86\_64.zip
2. Install latest libusb by Homebrew or copy our attached file to DYLD\_LIBRARY\_PATH (Default Path: /usr/local/lib)
3. In Xcode > Targets > General , in "Linked Frameworks and Libraries" add libusb-1.0.dylib and libMrLoopBF.a



4. In Xcode > Targets > Build Settings , add relative path in both of "Frameworks Search Paths" and "Library Search Paths".



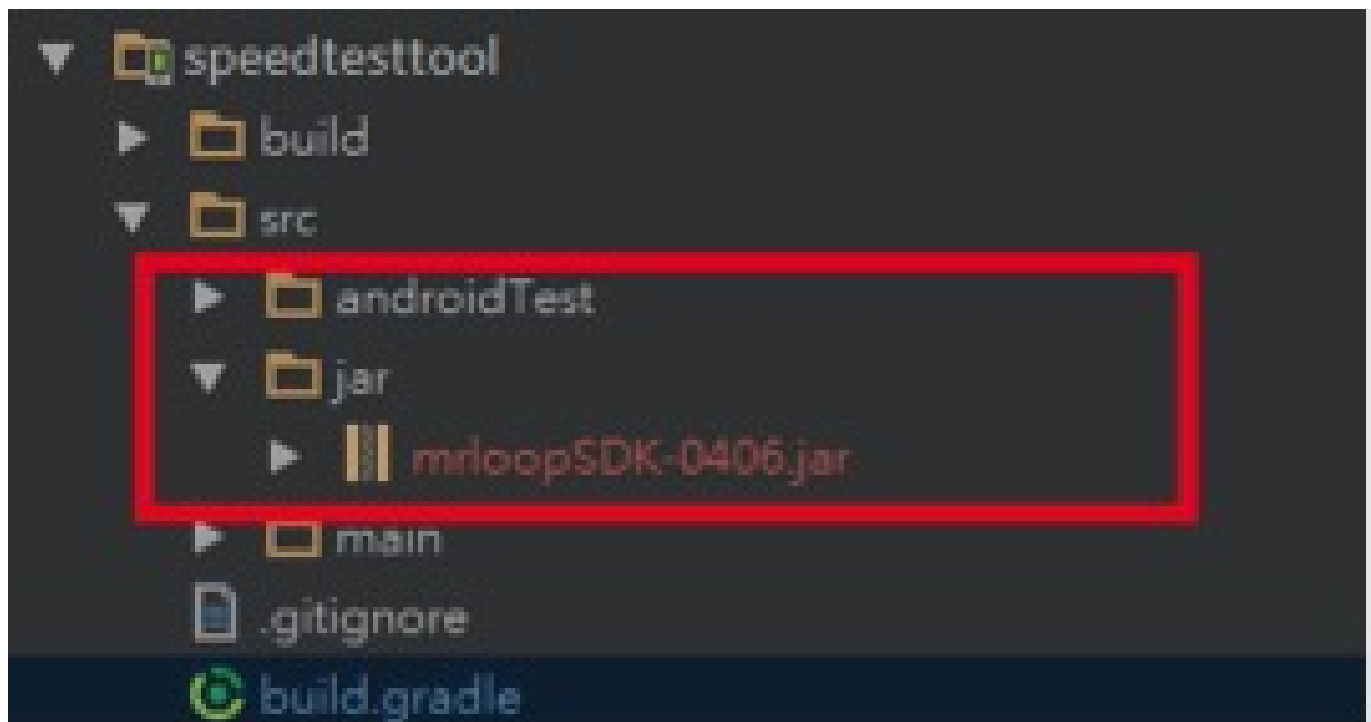
5. Finally, check Xcode > Targets > Build Phases, and make sure both of them had added in "Link Binary With Libraries"



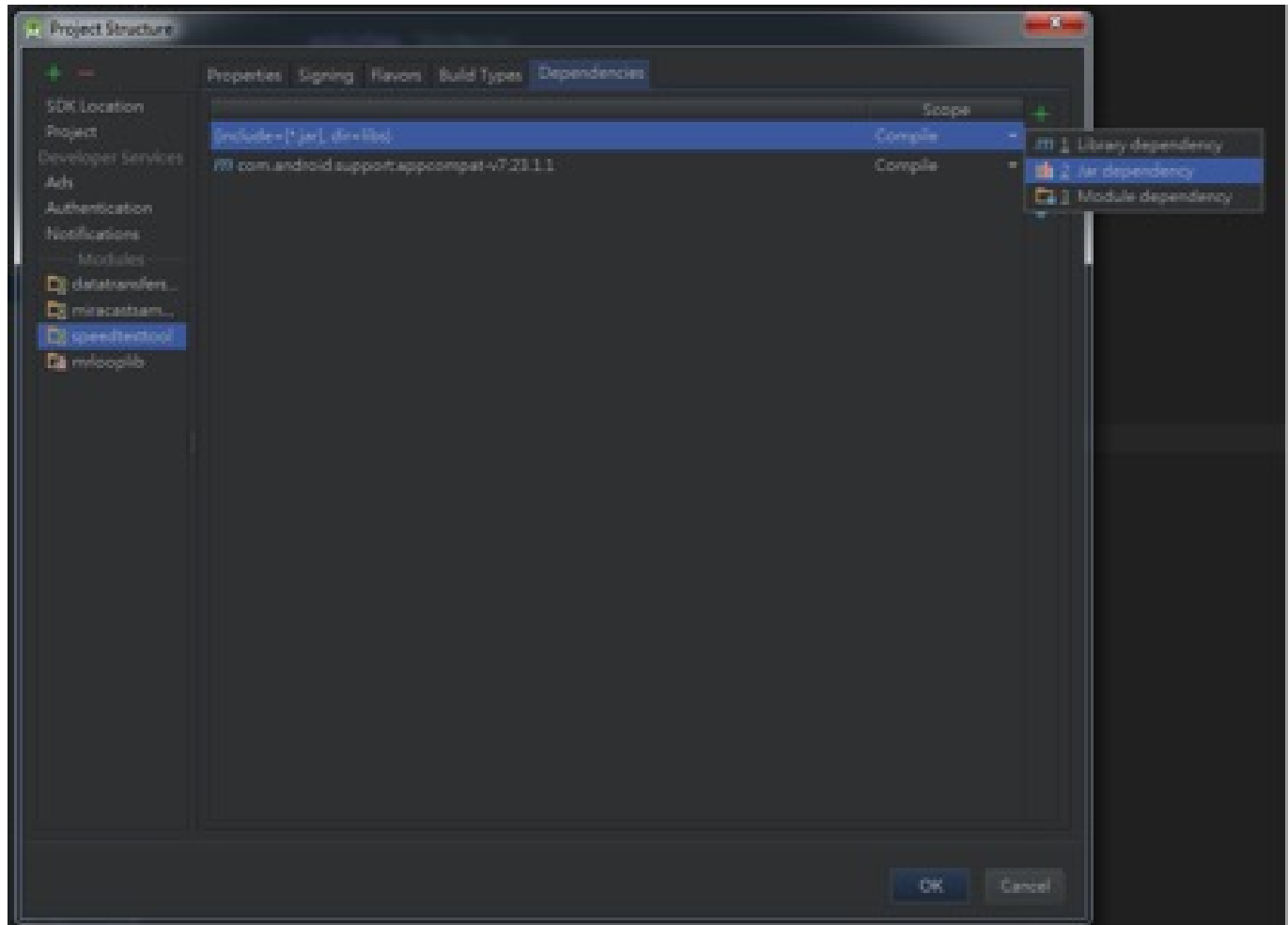
## Chapter 9

# Android SDK Install Tutorial

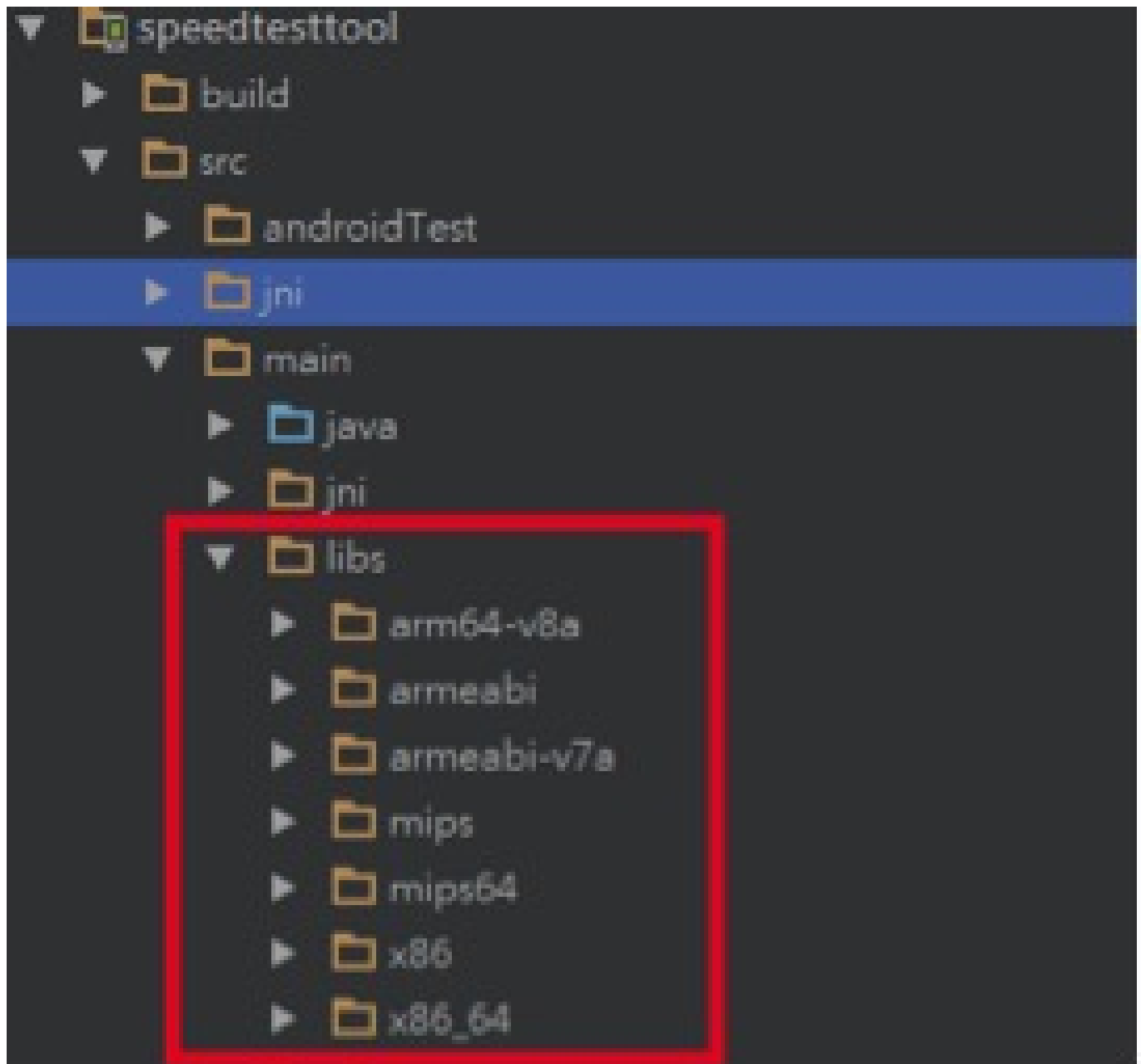
1. Copy jar folder to your Android module.



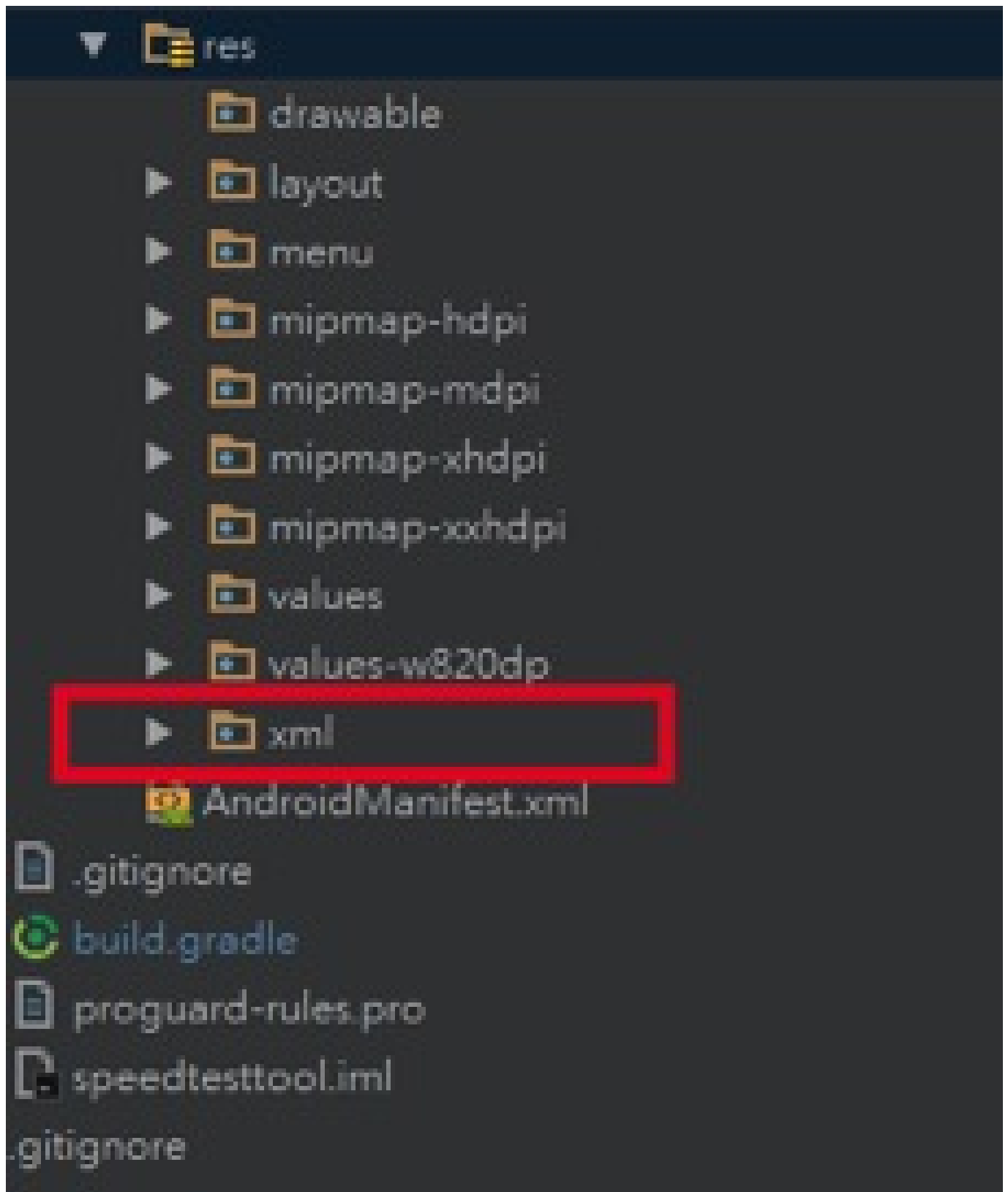
## 2. File -> Project Structre -> Dependencies Add jar dependency



### 3. Copy libs folder to Android modules



#### 4. Copy xml folder to res folder





## 5. Modify src\main\AndroidManifest.xml, and add intent-filter and meta-data

- <intent-filter>
- <action android:name="android.hardware.usb.action.USB\_DEVICE\_ATTACHED" />
- </intent-filter>
- <meta-data android:name="android.hardware.usb.action.USB\_DEVICE\_ATTACHED" android:resource="@xml/device\_filter" />

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.mrloop.speedtesttool" >

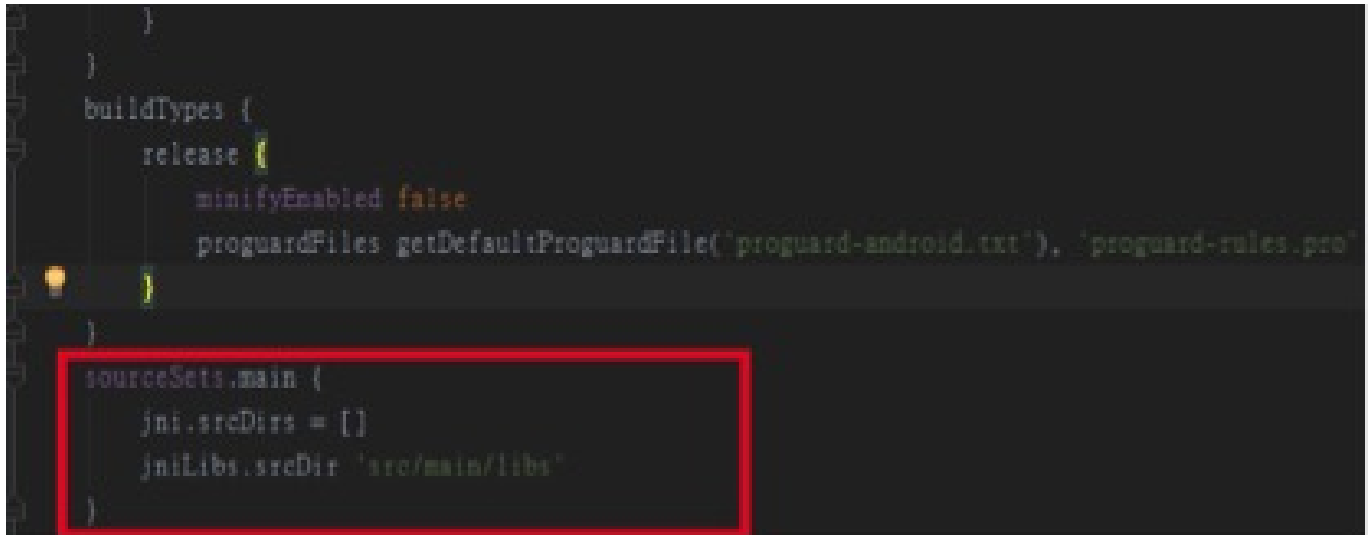
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="SpeedTestTool"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="SpeedTestTool"
            android:screenOrientation="landscape"
            android:configChanges="orientation|keyboardHidden"
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
            <intent-filter>
                <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
            </intent-filter>
            <meta-data
                android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
                android:resource="@xml/device_filter" />
        </activity>
    </application>

</manifest>
```

1. Modify module of build.gradle then add sourceSet in android.

- sourceSets.main {
- jni.srcDirs = []
- jniLibs.srcDir 'src/main/libs'}



```
    }  
    }  
    buildTypes {  
        release {  
            minifyEnabled false  
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'  
        }  
    }  
    sourceSets.main {  
        jni.srcDirs = []  
        jniLibs.srcDir 'src/main/libs'  
    }
```



## Chapter 10

# Mr.LOOP SDK SimpleTransfer Guide

- This tutorial describes how to drive our device with our SDK.
- 
- There are three states with hardware-driven:
  - 1. Enable devices.
  - 2. Devices work.
  - 3. Disable devices.
- 
- 
- We would implement each state in main function of SampleTransfer.cpp sequentially:
- 
- #Enable devices.
  - 1. Call `ML_HiddenDebugMsg()` to check if our library included and then disable error message output.
  - 2. Call `ML_Init()` to enable and open our device.
  - 3. Call `ML_SetSpeed(int)` to set up RF Speed threshold. Default value is 2.
  - 4. Call `ML_SetMode(int)` to recognize this peer is 1 (tx) or 2 (rx).
- 
- #Devices work.
  - 5. Developers always need to allocate enough buffers to be copied whatever the peer is Tx or Rx.
  - 6. CheckPktTx/CheckPktRx used to examine RF signals if packets received exactly.
- 
- #Disable devices
  - 7. Call `ML_Close()` to close devices before applications quit.

# Chapter 11

## File Index

### 11.1 File List

Here is a list of all files with brief descriptions:

<a href="#">mrloopbf_release.h</a>	26
<a href="#">Android API.java</a>	29



## Chapter 12

# File Documentation

### 12.1 mrloopbf\_release.h File Reference

#### Macros

- `#define MRLOOPBF_SHARED_EXPORT`

#### Functions

- `MRLOOPBF_SHARED_EXPORT int ML_Init ()`
- `MRLOOPBF_SHARED_EXPORT void ML_Close ()`
- `MRLOOPBF_SHARED_EXPORT void ML_HiddenDebugMsg ()`
- `MRLOOPBF_SHARED_EXPORT int ML_Transfer (uint8_t *In_byte_ptr, int In_length, ← ptr)`
- `MRLOOPBF_SHARED_EXPORT int ML_Receiver (uint8_t *In_byte_ptr, int *In_length)`
- `MRLOOPBF_SHARED_EXPORT bool ML_SetSpeed (uint8_t speed)`
- `MRLOOPBF_SHARED_EXPORT bool ML_SetMode (uint8_t mode)`
- `MRLOOPBF_SHARED_EXPORT uint8_t ML_GetDevGen (void)`

#### 12.1.1 Macro Definition Documentation

##### 12.1.1.1 MRLOOPBF\_SHARED\_EXPORT

```
#define MRLOOPBF_SHARED_EXPORT
```

#### 12.1.2 Function Documentation

##### 12.1.2.1 ML\_Close()

```
MRLOOPBF_SHARED_EXPORT void ML_Close ( )
```

Close the device and leave SDK .

#### Returns

no return.

### 12.1.2.2 ML\_GetDevGen()

```
MRLOOPBFSHARED_EXPORT uint8_t ML_GetDevGen (
    void )
```

Get Current USB Device Generation.

Returns

Integer value when succeeded, 0 if it fails. 1 is USB 1.1 2 is USB 2.0 3 is USB 2.1 4 is USB 3.0 5 is USB 3.1

### 12.1.2.3 ML\_HiddenDebugMsg()

```
MRLOOPBFSHARED_EXPORT void ML_HiddenDebugMsg ( )
```

Hide the debug output message.

Returns

no return.

### 12.1.2.4 ML\_Init()

```
MRLOOPBFSHARED_EXPORT int ML_Init ( )
```

Initial the device and SDK first.

Returns

If return 0, is initialization finish. The other return number is device driver, no device is attached or connect fail.

### 12.1.2.5 ML\_Receiver()

```
MRLOOPBFSHARED_EXPORT int ML_Receiver (
    uint8_t * In_byte_ptr,
    int * In_length )
```

To listen to RF. Timeout value is 1 second.

Parameters

<i>In_byte_ptr</i>	A buffer to receive data. The buffer size must be multiple of 4096bytes since the unit in RF transaction is 4096bytes. The buffer will be divided into 4k packets in transaction. Each packet may be dropped or repeat in RF transaction.
<i>In_length_ptr</i>	Input the size of the buffer and output the size of the receive packet. When it returns fail, *In_length_ptr will be 0.

## Returns

When it returns fail, `*In_length_ptr` will be 0.

ML\_Transfer does NOT guarantee that the packet is delivered without error. Even ML\_Transfer returns true, the packet could be dropped or repeated. The safe way is to put an index in the packet. Tx sends out the packet. Rx receives the packet and checks the index. Then Rx sends out one packet to note ack. If Tx does not receive the ack packet, then sends out the packet again or return error.

### 12.1.2.6 ML\_SetMode()

```
MRLOOPBFSHARED_EXPORT bool ML_SetMode (
    uint8_t mode )
```

is Set Mrloop WiGig Dongle RF rule

## Parameters

<i>mode</i>	Set mode value "1" is Master, mode value "2" is Slave.
-------------	--

## Returns

False is fail. Slave can Only connect to Master. Master can Only connect to Slave.

### 12.1.2.7 ML\_SetSpeed()

```
MRLOOPBFSHARED_EXPORT bool ML_SetSpeed (
    uint8_t speed )
```

is set Mrloop WiGig Dongle speed.

## Parameters

<i>speed</i>	It ranges between 1~7.
--------------	------------------------

## Returns

False is fail.

### 12.1.2.8 ML\_Transfer()

```
MRLOOPBFSHARED_EXPORT int ML_Transfer (
    uint8_t * In_byte_ptr,
    int In_length_ptr )
```

To send out packet. Timeout value is 1 second.

## Parameters

<i>In_byte_ptr</i>	The buffer to be sent out. The buffer size must be multiple of 4096bytes since the unit in RF transaction is 4096bytes. The buffer will be divided into 4k packets in transaction. Each packet may be dropped or repeat in RF transaction.
<i>In_length_ptr</i>	The size of the buffer.

## Returns

If return fail, there may be error on device driver or no device is attached.

ML\_Transfer does NOT guarantee that the packet is delivered without error. Even ML\_Transfer returns true, the packet could be dropped or repeated. The safe way is to put an index in the packet. Tx sends out the packet. Rx receives the packet and checks the index. Then Rx sends out one packet to note ack. If Tx does not receive the ack packet, then sends out the packet again or return error.

## 12.2 Android API.java File Reference

## Functions

- int [getDevices](#) (UsbManager mManager)
- String [ML\\_GetDescriptors](#) ()
- void [CloseDevice](#) ()
- int [ML\\_Transfer](#) (byte[] Txbuffer)
- int [ML\\_Receive](#) (byte[] Rxbuffer, int timeout)
- int [ML\\_SetMode](#) (byte mode)
- int [ML\\_SetSpeed](#) (byte Speed)

## 12.2.1 Function Documentation

## 12.2.1.1 CloseDevice()

```
void CloseDevice ( )
```

Close the device and leave SDK .

## Returns

no return.

## 12.2.1.2 getDevices()

```
int getDevices (
    UsbManager mManager )
```

Initial the device and SDK first.



## Parameters

<i>UsbManager</i>	It is android USB Manager.
-------------------	----------------------------

## Returns

If return 1, is initialization finish. The return number 0 is no device is attached or connect fail.

## 12.2.1.3 ML\_GetDescriptors()

```
String ML_GetDescriptors ( )
```

Get Current USB Device Generation.

## Returns

return string to know devices is USB 2.0 or USB 3.0 .

## 12.2.1.4 ML\_Receive()

```
int ML_Receive (
    byte [] Rxbuffer,
    int timeout )
```



To listen to RF. Timeout value is 1 second.

## Parameters

<i>In_byte_ptr</i>	A buffer to receive data. The buffer size must be multiple of 4096bytes since the unit in RF transaction is 4096bytes. The buffer will be divided into 4k packets in transaction. Each packet may be dropped or repeat in RF transaction.
<i>Timeout</i>	Set USB Timeout the unit is millisecond, please do not less then 500 ms.

## Returns

When it returns 0 is timeout or fail.

ML\_Transfer does NOT guarantee that the packet is delivered without error. Even ML\_Transfer returns true, the packet could be dropped or repeated. The safe way is to put an index in the packet. Tx sends out the packet. Rx receives the packet and checks the index. Then Rx sends out one packet to note ack. If Tx does not receive the ack packet, then sends out the packet again or return error.

## 12.2.1.5 ML\_SetMode()

```
int ML_SetMode (
    byte mode )
```

is set Mrloop WiGig Dongle speed.

## Parameters

<i>speed</i>	It ranges between 1~7.
--------------	------------------------

## Returns

0 is fail.

## 12.2.1.6 ML\_SetSpeed()

```
int ML_SetSpeed (
    byte Speed )
```

is Set Mrloop WiGig Dongle RF rule

## Parameters

<i>mode</i>	Set mode value "1" is Master, mode value "2" is Slave.
-------------	--

## Returns

0 is fail. Slave can Only connect to Master. Master can Only connect to Slave.

## 12.2.1.7 ML\_Transfer()

```
int ML_Transfer (
    byte [ ] Txbuffer )
```

To send out packet. Timeout value is 1 second.

## Parameters

<i>Txbuffer</i>	The buffer to be sent out. The buffer size must be multiple of 4096bytes since the unit in RF transaction is 4096bytes. The buffer will be divided into 4k packets in transaction. Each packet may be dropped or repeat in RF transaction.
-----------------	--

## Returns

If return 0 is transfer fail.

ML\_Transfer does NOT guarantee that the packet is delivered without error. Even ML\_Transfer returns true, the packet could be dropped or repeated. The safe way is to put an index in the packet. Tx sends out the packet. Rx receives the packet and checks the index. Then Rx sends out one packet to note ack. If Tx does not receive the ack packet, then sends out the packet again or return error.

# Index

- Android API.java, [29](#)
  - CloseDevice, [29](#)
  - getDevices, [29](#)
  - ML\_GetDescriptors, [30](#)
  - ML\_Receive, [30](#)
  - ML\_SetMode, [30](#)
  - ML\_SetSpeed, [31](#)
  - ML\_Transfer, [31](#)
- CloseDevice
  - Android API.java, [29](#)
- getDevices
  - Android API.java, [29](#)
- ML\_Close
  - mrloopbf\_release.h, [26](#)
- ML\_GetDescriptors
  - Android API.java, [30](#)
- ML\_GetDevGen
  - mrloopbf\_release.h, [26](#)
- ML\_HiddenDebugMsg
  - mrloopbf\_release.h, [27](#)
- ML\_Init
  - mrloopbf\_release.h, [27](#)
- ML\_Receive
  - Android API.java, [30](#)
- ML\_Receiver
  - mrloopbf\_release.h, [27](#)
- ML\_SetMode
  - Android API.java, [30](#)
  - mrloopbf\_release.h, [28](#)
- ML\_SetSpeed
  - Android API.java, [31](#)
  - mrloopbf\_release.h, [28](#)
- ML\_Transfer
  - Android API.java, [31](#)
  - mrloopbf\_release.h, [28](#)
- MRLOOPBF\_SHARED\_EXPORT
  - mrloopbf\_release.h, [26](#)
- mrloopbf\_release.h, [26](#)
  - ML\_Close, [26](#)
  - ML\_GetDevGen, [26](#)
  - ML\_HiddenDebugMsg, [27](#)
  - ML\_Init, [27](#)
  - ML\_Receiver, [27](#)
  - ML\_SetMode, [28](#)
  - ML\_SetSpeed, [28](#)
  - ML\_Transfer, [28](#)
  - MRLOOPBF\_SHARED\_EXPORT, [26](#)

