



COMAIR5 Library API Specification

V1.0.2 – Nov 03, 2014

Important Notice

GENERALPLUS TECHNOLOGY INC. reserves the right to change this documentation without prior notice. Information provided by GENERALPLUS TECHNOLOGY INC. is believed to be accurate and reliable. However, GENERALPLUS TECHNOLOGY INC. makes no warranty for any errors which may appear in this document. Contact GENERALPLUS TECHNOLOGY INC. to obtain the latest version of device specifications before placing your order. No responsibility is assumed by GENERALPLUS TECHNOLOGY INC. for any infringement of patent or other rights of third parties which may result from its use. In addition, GENERALPLUS products are not authorized for use as critical components in life support devices/ systems or aviation devices/systems, where a malfunction or failure of the product may reasonably be expected to result in significant injury to the user, without the express written approval of Generalplus.

Table of Content

1	Introduction	6
2	Library Module and Header Files.....	7
3	Notice for Developing app.....	9
3.1	ipod touch project	9
3.2	Developing app with game engine.....	9
3.3	CPU loading issue on previous device.....	10
3.4	Unity3D compatibility	11
4	Constrains and Limitations	12
5	API Calling Sequence.....	13
5.1	Decode Calling Sequence	13
6	API Specification	14
6.1	Data Structure	14
6.1.1	S_ComAir5CommnadList	14
6.2	Exported APIs.....	14
6.2.1	InitComAir5	14
6.2.2	UnitComAir5	15
6.2.3	StartComAir5Decode	15
6.2.4	StopComAir5Decode.....	15
6.2.5	SetComAir5Property	16
6.2.6	GetComAir5Property.....	17
6.2.7	PFN_UserCallBack	17
6.2.8	PFN_UserRawDataCallBack.....	18
6.2.9	PlayComAir5Cmd	18
6.2.10	PlayComAir5CmdList	19
6.2.11	IsComAir5CmdPlaying	19
6.3	COMAIR5 Property Type.....	21
6.3.1	eComAir5Property_RegCode.....	21
6.3.2	eComAir5Property_CentralFreq.....	21
6.3.3	eComAir5Property_iDfValue.....	21
6.3.4	eComAir5Property_CmdCallBack	21
6.3.5	eComAir5Property_RawDataCallBack	21
6.3.6	eComAir5Property_Threshold.....	21

6.4	Error Code	22
7	Reference.....	23

Revision History

Revision	Date	Revised By	Remark
1.0.2	11/06/2014	Shawn Liu	Support threshold. Unity3D compatibility.
1.0.1	12/17/2013	Shawn Liu	Rename ComAir to COMAIR
1.0.0	08/02/2013	Shawn Liu	First release.

1 Introduction

The COMAIR5 library is to make it easily to interact with the devices which embed Generalplus COMAIR5 chips on iOS platform.

This library provides following functions:

- (1) Initialize audio record unit and COMAIR5 decode function on iOS platform.
- (2) Record and Decode COMAIR5 sound by build-in microphone on iOS platform.
- (3) Play COMAIR5 command.

This API specification describes all the APIs provided by the COMAIR5 library.

2 Library Module and Header Files

This COMAIR5 library is in **static library on iOS** platform. This library is written by **C++**, before linking our library you must add the linker flag **-lstdc++.6** (Figure 2-1) or specify **C++/objective-C++** compiler (Figure 2-2) to compile target project.

Static Library Module Name: **libComAir5Library_V1.0.1.a**

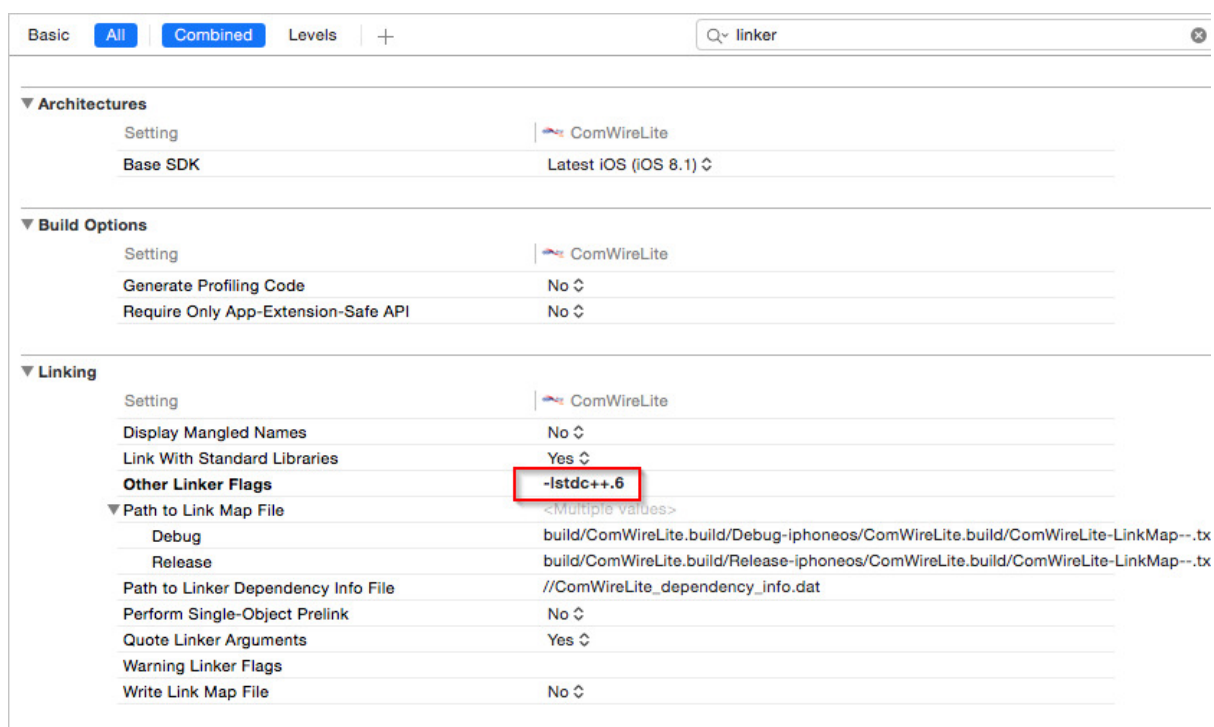


Figure 2-1 Linker Flags

Basic
All
Combined
Levels
+

▼ Apple LLVM 6.0 - Custom Compiler Flags

Setting
ComWireLite

Other C Flags

Other C++ Flags

Other Warning Flags

▼ Apple LLVM 6.0 - Language

Setting
ComWireLite

'char' Type Is Unsigned
No

Allow 'asm', 'inline', 'typeof'
Yes

C Language Dialect
GNU99 [-std=gnu99]

CodeWarrior/MS-Style Inline Assembly
Yes

► Compile Sources As
Objective-C++

Enable Linking With Shared Libraries
Yes

Enable Trigraphs
No

Generate Floating Point Library Calls
No

Increase Sharing of Precompiled Headers
No

Precompile Prefix Header
No

Prefix Header

Recognize Built-in Functions
Yes

Recognize Pascal Strings
Yes

Short Enumeration Constants
No

Use Standard System Header Directory Searching
Yes

Figure 2-2 Specify C++ Compile

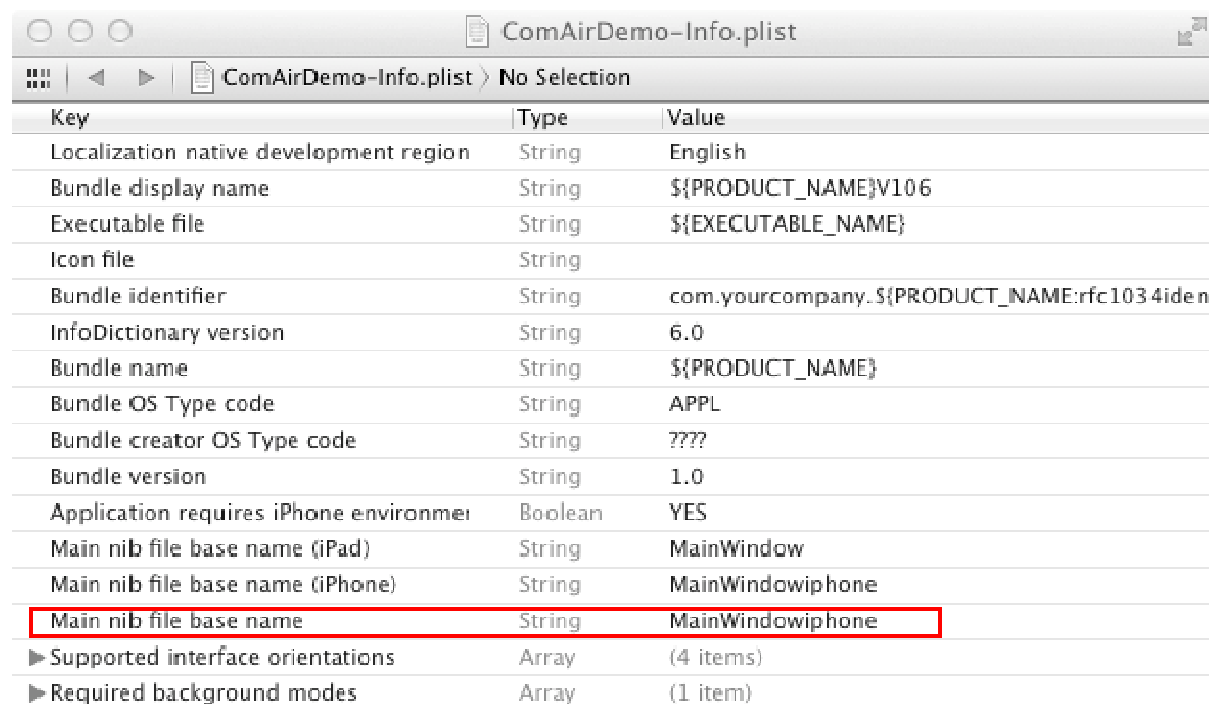
Header Files:

1. **ComAir5API.h**: Static library class definition which is used to manipulate COMAIR5 library.

3 Notice for Developing app

3.1 ipod touch project

By default it will always loads the ipad main nib, this will causes the screen goes dark and no response when run app on ipod touch. Please assign the attribute **Main nib file base name** to the iphone main nib in your .plist file.



Key	Type	Value
Localization native development region	String	English
Bundle display name	String	\${PRODUCT_NAME}V106
Executable file	String	\${EXECUTABLE_NAME}
Icon file	String	
Bundle identifier	String	com.yourcompany.\${PRODUCT_NAME:rfc1034iden
InfoDictionary version	String	6.0
Bundle name	String	\${PRODUCT_NAME}
Bundle iOS Type code	String	APPL
Bundle creator OS Type code	String	????
Bundle version	String	1.0
Application requires iPhone environment	Boolean	YES
Main nib file base name (iPad)	String	MainWindow
Main nib file base name (iPhone)	String	MainWindowiphone
Main nib file base name	String	MainWindowiphone
Supported interface orientations	Array	(4 items)
Required background modes	Array	(1 item)

Figure 3-1 .plist file for ipod touch

3.2 Developing app with game engine

Some game engine will configure the audio session settings during their own initial process; these settings may cause the COMAIR5 decoding process out of functional. If using these game engine, please make sure the audio session settings are set to appropriate value that are described below:

Settings	Appropriate Value
Audio Session Categories	AVAudioSessionCategoryPlayAndRecord
Audio Session Property: kAudioSessionProperty_PreferredHardwareSampleRate	Twice larger than COMAIR5 Central frequency

How to configure the Audio session please refers to the web site

[Audio Session Programming Guide.](http://developer.apple.com/library/ios/#documentation/Audio/Conceptual/AudioSessionProgrammingGuide/Introduction/Introduction.html)

<http://developer.apple.com/library/ios/#documentation/Audio/Conceptual/AudioSessionProgrammingGuide/Introduction/Introduction.html>

3.3 CPU loading issue on previous device

Some previous device like iPhone 3GS is easy to have insufficient CPU loading when running complicated Apps. This will cause the non-continuity when getting data from its microphone. User can increase buffer size to solve this problem.

```
int i32Ret = InitComAir5 ();
//Set to 0.04 sec. COMAIR5 library default set to 0.02 sec
Float32 preferredBufferSize = .04;
AudioSessionSetProperty(kAudioSessionProperty_PreferredHardwareIOBufferDuration, sizeof(preferredBufferSize), &preferredBufferSize);
```

3.4 Unity3D compatibility

During Unity3D initial process, the audio session will set to playback only and hardware sample rate down to 24Khz. ComAir/ComWire only works with the audio session which sets to PlaybackAndRecord and 48Khz hardware sample rate. Please modify audio session setting after initial ComAir/ComWire function:

```
// set audio session after ComAir/ComWire initial function
[[AVAudioSession sharedInstance] setCategory:AVAudioSessionCategoryPlayAndRecord
error:nil];
```

```
//iOS6
Float64 sampleRate = 48000.00;
AudioSessionSetProperty(kAudioSessionProperty_PreferredHardwareSampleRate,
                        sizeof(sampleRate),
                        &sampleRate);
```

```
//iOS7
[[AVAudioSession sharedInstance]setPreferredSampleRate:48000.00 error:nil];
```

```
[[AVAudioSession sharedInstance] setActive:YES error:nil];
```

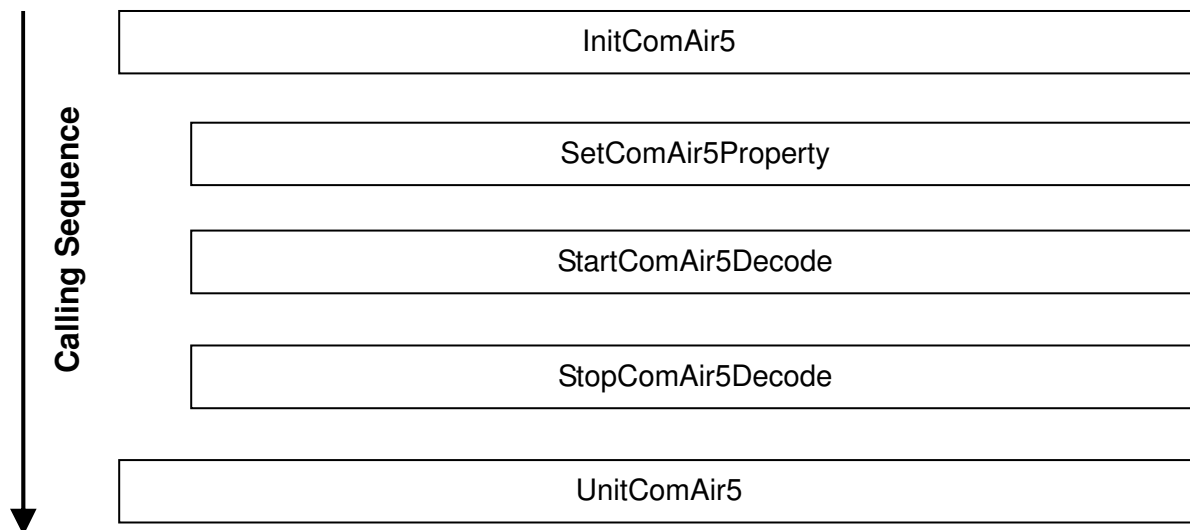
4 Constrains and Limitations

Constrains:

1. This library can only be used to interact with the devices which embed Generalplus COMAIR5 chips.
2. This library supports only iOS platforms.

5 API Calling Sequence

5.1 Decode Calling Sequence



1. The API surrounded with dotted line can be optionally called.
2. The indented APIs should be called after its upper protrusive APIs were called.

6 API Specification

6.1 Data Structure

6.1.1 S_ComAir5CommnadList

Prototype:

```
typedef struct tagComAir5CommandList
{
    int          i32command;
    Float32      f32Delay;
}S_ComAir5CommnadList
```

Description:

A data structure contains COMAIR5 command.

Parameters:

i64command:

The COMAIR5 command.

f32Delay:

The mute intervals (seconds) after playing COMAIR5 sound.

6.2 Exported APIs

6.2.1 InitComAir5

Prototype:

```
int          InitComAir5 (
    );
```

Description:

Initialize COMAIR5 Audio unit for decoding.

Return Value:

Return 0 (*COMAIR5_NOERR*) if this function succeeded. Otherwise, other value returned. Please refer to Section 6.4 for error code definition.

Parameters:

Remark:

This API should be called before any other ones.

6.2.2 UnitComAir5

Prototype:

```
int          UnitComAir5 (  
                );
```

Description:

Uninitialize COMAIR5 Audio unit.

Return Value:

Return 0 (*COMAIR5_NOERR*) if this function succeeded. Otherwise, other value returned. Please refer to Section 6.4 for error code definition.

Parameters:

Remark:

This API should be called before exit program.

6.2.3 StartComAir5Decode

Prototype:

```
int          StartComAir5Decode (  
                );
```

Description:

Start to decode COMAIR5 sound.

Return Value:

Return 0 (*COMAIR5_NOERR*) if this function succeeded. Otherwise, other value returned. Please refer to Section 6.4 for error code definition.

Parameters:

Remark:

Before calling this API, it must set the property *eComAir5Property_CmdCallBack* first.

6.2.4 StopComAir5Decode

Prototype:

```
int          StopComAir5Decode (  
                );
```

Description:

Stop decoding COMAIR5 sound.

Return Value:

Return 0 (*COMAIR5_NOERR*) if this function succeeded. Otherwise, other value returned. Please refer to Section 6.4 for error code definition.

Parameters:

Remark:

6.2.5 SetComAir5Property

Prototype:

```
int          SetComAir5Property (  
             eComAir5PropertyTarget eTarget,  
             eComAir5Property      eProperty,  
             void                   *vpValue,  
             Int                    i32ValueSize  
             );
```

Description:

Set COMAIR5 property. This function can set encode and decode COMAIR5 property separately.

Return Value:

Return 0 (*COMAIR5_NOERR*) if this function succeeded. Otherwise, other value returned. Please refer to Section 6.4 for error code definition.

Parameters:

[In] eTarget:

Specify target to set COMAIR5 property.

eComAir5PropertyTarget_Encode is for Encode;

eComAir5PropertyTarget_Decompile is for Decode. Please refer to Section 6.3 for more information.

[In] eProperty:

The COMAIR5 property type. Please refer to section 6.3 for property type.

[In] vpValue:

The property value.

[In] i32ValueSize:

The property size.

Remark:

6.2.6 GetComAir5Property

Prototype:

```
int          GetComAir5Property (  
    eComAir5PropertyTarget  eTarget,  
    eComAir5Property        eProperty,  
    void                    *vpValue  
    Int                     i32ValueSize  
);
```

Description:

Get COMAIR5 property. This function can get encode and decode COMAIR5 property separately.

Return Value:

Return 0 (*COMAIR5_NOERR*) if this function succeeded. Otherwise, other value returned. Please refer to Section 6.4 for error code definition.

Parameters:

[In] eTarget:

Specify target to get COMAIR5 property.

eComAir5PropertyTarget_Encode is for Encode;

eComAir5PropertyTarget_Decompile is for Decode. Please refer to Section 6.3 for more information.

[In] eProperty:

The COMAIR5 property type. Please refer to section 6.3 for property type.

[Out] vpValue:

The property value.

[In] i32ValueSize:

The property size.

Remark:

6.2.7 PFN_UserCallBack

Prototype:

```
int          PFN_UserCallBack (  
    int      i32Commnad  
);
```

Description:

This function will be called when catch the COMAIR5 command.

Return Value:

Return 0 (*COMAIR5_NOERR*) if this function succeeded. Otherwise, other value returned. Please refer to Section 6.4 for error code definition.

Parameters:

[Out] i32Commnad:

COMAIR5 command value..

Remark:

6.2.8 PFN_UserRawDataCallBack

Prototype:

```
int          PFN_UserRawDataCallBack (  
            unsigned char*    pbyRawData,  
            int                i32Size  
            );
```

Description:

This function will be called when decoding sound.

Return Value:

Return 0 (*COMAIR5_NOERR*) if this function succeeded. Otherwise, other value returned. Please refer to Section 6.4 for error code definition.

Parameters:

[Out] pbyRawData:

The PCM data.

[Out] i32Size:

The size of PCM data.

Remark:

6.2.9 PlayComAir5Cmd

Prototype:

```
int          PlayComAir5Cmd (  
            Int        i32command,  
            Float32     SoundVolume  
            );
```

Description:

Play COMAIR5 sound.

Return Value:

Return 0 (*COMAIR5_NOERR*) if thi s function succeeded. Otherwise, other value returned. Please refer to Section 6.4 for error code definition.

Parameters:

[in] i32command:

The COMAIR5 command.

[in] SoundVolume:

The level of sound volume. Only accepts the range from 0 to 1.0.

Remark:

6.2.10 PlayComAir5CmdList

Prototype:

```
int          PlayComAir5CmdList (  
    int          i32Cnt,  
    S_ComAir5CommnadList *pCommandList,  
    Float32      SoundVolume  
);
```

Description:

Play COMAIR5 command list by specified the delay between sounds.

Return Value:

Return 0 (*COMAIR5_NOERR*) if this function succeeded. Otherwise, other value returned. Please refer to Section 6.4 for error code definition.

Parameters:

[in] i32Cnt:

The command list count.

[in] pCommandList:

The pointer to S_ComAir5CommnadList.

[in] SoundVolume:

The level of sound volume. Only accepts the range from 0 to 1.0.

Remark:

6.2.11 IsComAir5CmdPlaying

Prototype:

```
BOOL          IsComAir5CmdPlaying (  
);
```

Description:

Return the COMAIR5 sound playing.

Return Value:

Return true if COMAIR5 sound is playing. Otherwise return false.

Parameters:

Remark

6.3 COMAIR5 Property Type

Property Type	Target	Action	Function	Value Type
<i>eComAir5Property_RegCode</i>	E/D	S	Register code.	String
<i>eComAir5Property_CentralFreq</i>	E/D	S/G	Central frequency.	Int
<i>eComAir5Property_iDfValue</i>	E/D	S/G	iDf value.	Int
<i>eComAir5Property_CmdCallBack</i>	D	S	Decode command call back function.	Pointer
<i>eComAir5Property_RawDataCallBack</i>	D	S	Decode raw data call back function.	Pointer
<i>eComAir5Property_Threshold</i>	D	S/G	Decode threshold.	int

D: Decode, E: Encode, S: Set, G: Get

6.3.1 eComAir5Property_RegCode

Set register code for Encoding/Decoding COMAIR5 command.

6.3.2 eComAir5Property_CentralFreq

Set/Get COMAIR5 decode/encode central frequency. The ipad/iphone frequency response from 20Hz to 20,000Hz .

6.3.3 eComAir5Property_iDfValue

Set/Get COMAIR5 frequency shift value.

6.3.4 eComAir5Property_CmdCallBack

Set the user call back function. User call back will be called when catch the command during decoding COMAIR5 sound.

6.3.5 eComAir5Property_RawDataCallBack

Set the user call back function for saving raw PCM data.

6.3.6 eComAir5Property_Threshold

Set the decode threshold. Default is 12.

6.4 Error Code

Symbol Name	Value	Description
<i>COMAIR5_NOERR</i>	0x0000	No error. API works normally.
<i>COMAIR5_AUDIONOTINIT</i>	0x0001	ComAir5 not initialize.
<i>COMAIR5_AUIDOUINTFAILED</i>	0x0002	Get audio unit failed.
<i>COMAIR5_ENABLEIORECFAILED</i>	0x0003	Enable Record failed.
<i>COMAIR5_PROPERTYNOTFOUND</i>	0x0004	Property not found.
<i>COMAIR5_PROPERTYOPERATIONFAILED</i>	0x0005	Get/Set property failed..
<i>COMAIR5_PROPERTYSIZEMISSMATCH</i>	0x0006	Property size is not correct.
<i>COMAIR5_PLAYCOMAIR5SOUNDFAILED</i>	0x0007	Play COMAIR5 sound failed.

7 Reference

The Objective-C Programming Language

<http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html>

iOS Developer library

<http://developer.apple.com/library/ios/navigation/>

iOS Application Programming Guide

<http://developer.apple.com/library/ios/#documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/Introduction/Introduction.html>

Multimedia Programming Guide

<http://developer.apple.com/library/ios/#documentation/AudioVideo/Conceptual/MultimediaPG/Introduction/Introduction.html>

Audio Session Programming Guide

<http://developer.apple.com/library/ios/#documentation/Audio/Conceptual/AudioSessionProgrammingGuide/Introduction/Introduction.html>