



ComAir5 RX Library User's Manual V1.0

08/21/2013

0 Table of Content

0	TABLE OF CONTENT	2
1	REVISION HISTORY.....	3
1.1	Document History	3
2	COMAIR5 RX SUMMARY.....	4
2.1	Transmission Distance.....	4
2.2	Speaker and Microphone Selection	4
2.3	Environment Consideration.....	4
2.4	Filter in the Electrical Circuit	4
2.5	The Data Format.....	4
3	LIBRARY SERVICE LOOPS	5
4	API OF COMAIR5_RX	7
4.1	Hardware Dependent Functions Function: Initializes ComAir5_RX.....	7
4.1.1	Function: Initialize ComAir5 Rx module, sets Interrupt source, Timer and ADC before running 7	
4.2	Service Loop Functions: Service loop for ComAir5 RX.....	7
4.2.1	Function: Rx foreground service loop.....	7
4.3	ComAir5 Control.....	7
4.3.1	Function: Start Receiving	7
4.3.2	Function: Stop Receiving	8
4.4	ISR Functions: Interrupt service routine for ComAir5 RX	8
4.5	User Functions: for ComAir5_RX	8
4.5.1	Function: Hardware Initial for Rx of ComAir5	9
4.5.2	Function: Hardware Get ADC data for ComAir5_RX	9
4.5.3	Function: Hardware Stop for ComAir5_RX.....	9
5	RESOURCES LIST OF COMAIR5_RX.....	10
5.1	TABLE 1: RAM Size (Unit: Decimal Word).....	10
5.2	TABLE 2: ROM Size (Unit: Decimal Word).....	10
5.3	TABLE 3: Hardware Resources VS Library	10
5.4	TABLE 5: CPU Usage Rate (approximate).....	10
5.5	TABLE 6: Name of Overlap RAM in the library	10

1 Revision History

1.1 Document History

Revision	Date	By	Remark
V1.0	08/21/2013	Porter Yang	Original Version

2 ComAir5 RX Summary

The ComAir5_RX algorithm provides the ability to transmit-receive commands through the air. This algorithm can be used in communication areas. Below are some recommended considerations developers should consider.

2.1 Transmission Distance

The maximum transmission distance between transmitter and receiver depends on the room acoustics, echoes, and environment noise. Also the transmission power or sound amplitude plays an important role. Generally speaking, if the separation distance is up to 5meters, the transmission power should be larger than 50dB.

2.2 Speaker and Microphone Selection

The separation distances talked above assume selection of proper high-quality microphones and speakers. Many brands of electret microphones have flat frequency responses to 20 kHz, which is more than adequate. However, not all microphones meet this specification. Avoid microphones that do not have flat frequency responses to 20 kHz.

Selection of speakers is more difficult because many, but not all, inexpensive speakers are satisfactory and meet specifications. Thus, the best approach is to try several speakers and select one that produces audio signals that are sufficiently loud and that works as a sonic tone component to the distances required by the application.

2.3 Environment Consideration

If there is any sound absorbing material in front of the microphone or speaker, the performance can degrade a lot. The material covering microphones and speakers must be the minimum possible.

2.4 Filter in the Electrical Circuit

The electrical filters connected to microphone or speaker should have upper cut off frequency not less than working frequency. The proper frequency response can be achieved by proper selection of resistors and capacitors.

2.5 The Data Format

1 tone	3 tone	1 tone
SYNC	DATA	CHECKSUM

SYNC: Synchronization signal.

DATA: Total of 80 command

CHECKSUM: check the accuracy of data transmitted.

To send a command takes 1.1 second.

3 Library Service loops

The ComAir5 library can support the foreground service loops. In foreground service loop, users have to put the service loop in the main routine to keep entering the service regularly. Inside the ComAir5 service loop, there will be a mechanism to decide any task should be carried on. Some overhead will produced inevitably. The amount of overhead varies and depends on the payload of CPU.

Example:

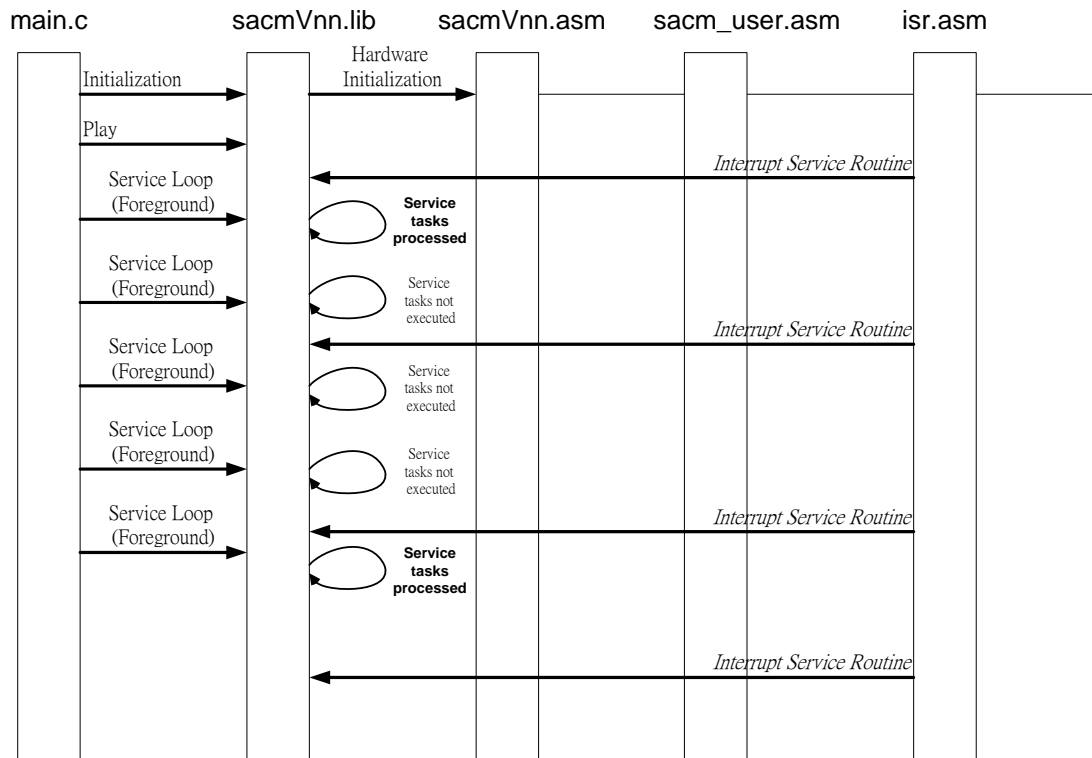
Foreground service loop:

In main.c

```
int main()
{
    System_Initial();                // System initial
    ComAir5_RX_Init(17000, 200);     // ComAir5_RX initial
    ComAir5_RX_Start(12);            // Start Receiving command
    while(1)
    {
        ComAir5_RX_ServiceLoop();    // Foreground Service loop
        System_ServiceLoop();        // Service loop for watchdog clear
    } // end of while
    return 0;
} // end of main
```

In isr.asm:

```
_FIQ:
    push R1, R5 to [SP];            // save registers
    call F_ISR_Service_ComAir5_RX   // Interrupt service routine
    R1 = C_IRQ0_TMA;
    [P_INT_Clear] = R1;
    pop R1, R5 from [SP];           // restore registers
    reti;
```



Timing diagram: Auto mode, Foreground service loop

4 API of ComAir5_RX

4.1 Hardware Dependent Functions Function: Initializes ComAir5_RX

4.1.1 Function: Initialize ComAir5 Rx module, sets Interrupt source, Timer and ADC before running

Syntax:

C: int ComAir5_RX_Init(int CentFreq, int DeltaFreq)

ASM: R1: CentFreq
R2: DeltaFreq
Call F_ComAir5_RX_Init

Parameters: 1: CentFreq: User directly sets central frequency
2: DeltaFreq User directly sets delta frequency

Parameters: CentFreq, DeltaFreq

Return Value: None

Library: ComAir5_RX_vXXX

Remark:

1. This function initializes the Kernel of ComAir5_RX. It also initializes the Timer A, ADC and enables the Timer A FIQ at the sample rate on 48KHz.
2. The hardware setting is opened for user's reference (see F_CA_GPD8_HW_Init in ComAir5_RX_User.asm).
3. Example: User directly sets 17000Hz as cetral frequency and 200Hz as delta frequency.
ComAir5_RX_Init(17000, 200);

4.2 Service Loop Functions: Service loop for ComAir5 RX

4.2.1 Function: Rx foreground service loop

Syntax:

C: void ComAir5_RX_ServiceLoop(void)

ASM: Call F_ComAir5_RX_ServiceLoop

Parameters: None

Return Value: None

Library: ComAir5_RX_vXXX

Remark: Foreground service loop has to be placed in main loop.

4.3 ComAir5 Control

4.3.1 Function: Start Receiving

Syntax:

C: void ComAir5_RX_Start(int threshold)

ASM: R1 = threshold
Call F_ComAir5_RX_Start

Parameters: threshold: 0~20
Return Value: N/A
Library: ComAir5_RX_vXXX
Remark: Threshold is from 0 to 20. Larger will stricter.

4.3.2 Function: Stop Receiving

Syntax:

C: void ComAir5_RX_Stop(void);
ASM: Call F_ComAir5_RX_Stop
Parameters: N/A
Return Value: N/A
Library: ComAir5_RX_vXXX
Remark: N/A

4.4 ISR Functions: Interrupt service routine for ComAir5 RX

Syntax:

C: void ISR_Service_ComAir5_RX(void)
ASM: Call F_ISR_Service_ComAir5_RX
Parameters: None
Return Value: None
Library: ComAir5_RX_vXXX
Remark: 1. This function is used in assembly only and it can be hooked on the _FIQ, _IRQ1 or _IRQ2: label.
(See isr.asm for details)
2. The F_ISR_Service_ComAir5_RX will not take up any time to process the Interrupt routine except minor overheads if the program is not in beat tracking mode. It is possible for users to place user-define function in the same FIQ or IRQ.

EX:

```
_FIQ:
    push r1,r5 to [sp];

    call F_ISR_Service_ComAir5_RX
    call F_User_ISR

    R1 = C_IRQ0_TMA;
    [P_INT_Clear] = R1;
    pop r1,r5 from [sp]
    reti
```

3. This function will not destroy the content in R1-R5 register. Programmers have not to protect the registers externally.
4. The Timer A FIQ is working on 48KHz.

4.5 User Functions: for ComAir5_RX

4.5.1 Function: Hardware Initial for Rx of ComAir5

Syntax:

C: None

ASM: Call F_ComAir5_RX_HW_Init

Parameters: None

Return Value: None

Library: ComAir5_RX_User.asm

Remark:

1. Set timer as 48KHz
2. Setup ADC

4.5.2 Function: Hardware Get ADC data for ComAir5_RX

Syntax:

C: None

ASM: Call F_ComAir5_RX_HW_GetADC

Parameters: None

Return Value: R3: signed ADC data

Library: ComAir5_RX_User.asm

Remark: Get signed ADC data

4.5.3 Function: Hardware Stop for ComAir5_RX

Syntax:

C: None

ASM: Call F_ComAir5_RX_HW_Stop

Parameters: None

Return Value: None

Library: ComAir5_RX_User.asm

Remark: None

5 Resources List of ComAir5_RX

5.1 TABLE 1: RAM Size (Unit: Decimal Word)

	IRAM	ISRAM	RAM	SRAM	ORAM	OSRAM
ComAir5_RX					987	

5.2 TABLE 2: ROM Size (Unit: Decimal Word)

	TEXT	CODE	DATA	USER DEFINE
ComAir5_RX	4583	906		

5.3 TABLE 3: Hardware Resources VS Library

	Interrupt	Timer Setting	Audio
ComAir5_RX	TMA FIQ	48 KHz	ADC

5.4 TABLE 5: CPU Usage Rate (approximate)

	GPCE063A CPU Usage Rate (49 MHz) while ComAir5_RX
ComAir5_Rx	49% (2.6ms/5.3ms)

5.5 TABLE 6: Name of Overlap RAM in the library

Table: Name of Overlap RAM in the library

	Overlap RAM definition	
Algorithm	Overlap RAM Label	Size(word)
ComAir5_RX	OVERLAP_ComAir5_RX_API_BLOCK	0xF3
ComAir5_RX	GPTD8_ORAM_BLOCK	0x2E8