



GPL162XXA SACM Library Quick Start Manual

V0.1 - Mar. 13, 2008

Preliminary

Table of Content

	<u>PAGE</u>
GPL162XXA SACM Library	1
Quick User Manual.....	1
1 Function List.....	4
2 Application Interface	5
2.1 SACM_Initial.....	5
2.2 SACM_Codec	5
2.3 SACM_Volume	5
2.4 SACM_Speed.....	6
2.5 User_SetStartAddrManual.....	6
2.6 USER_SetRECStartAddr	7
2.7 SACM_Play	7
2.8 SACM_ServiceLoop	8
2.9 SACM_Pause	8
2.10 SACM_Resume.....	9
2.11 SACM_Stop	9
2.12 SACM_Status	9
2.13 SACM_DVR1800_BITRATE	10
2.14 SACM_Rec	10
2.15 SACM_MP3_SetFS	10
2.16 SACM_MP3_Get_Time	11
3 Specification and System Requirement of SACM.....	12
4 CPU Performance of SACM.....	13
5 Service Loops for SACM.....	14



Revision History

Revision	Date	By	Remark
0.1	2008/3/13	Eddie Wang	First edition

Generalplus Confidential
For Public International Ltd. Use
Only

1 Function List

Index	Syntax
1	void SACM_Initial();
2	void SACM_Codec(int codec);
3	void SACM_Volume(int volume);
4	void SACM_Speed(int speed);
5	void USER_SetStartAddr(int index);
6	void USER_SetRECStartAddr(long address);
7	void SACM_Play(int Speech index, int Channel ,int Ramp);
8	void SACM_ServiceLoop(void);
9	void SACM_Pause(void);
10	void SACM_Resume(void);
11	void SACM_Stop(void);
12	unsigned int SACM_Status(void);
13	void SACM_DVR1800_BITRATE(int BitRate);
14	void SACM_Rec(int RceMonitor, int bit_rate);
15	void SACM_MP3_SetFS(void);
16	int SACM_MP3_Get_Time(void);

2 Application Interface

2.1 SACM_Initial

API Name		SACM_Initial(void)
Function		Kernel variables initialization and calls HW initial
Description		Initial SACM Speech.
Header File	C	SACM_API.h
	ASM	
Syntax	C	void SACM_Initial(void);
	ASM	
Parameters		void
Return Values		None
Remarks		This Function must be called when SACM speech is played or record function is applied.

2.2 SACM_Codec

API Name		SACM_Codec(int codec)
Function		decoded speech type selection
Description		decoded speech type selection
Header File	C	SACM_API.h
	ASM	
Syntax	C	void SACM_Codec(int codec);
	ASM	
Parameters		codec: depends on how many speech is used. Generalplus suggests programmers using constant value that defined in algorithm.h
Return Values		
Remarks		Set the speech type

2.3 SACM_Volume

API Name		SACM_Volume(int volume)
Function		speech volume setup
Description		Sets up speech volume
Header File	C	SACM_API.h
	ASM	
Syntax	C	void SACM_Volume(int volume);
	ASM	

API Name	SACM_Volume(int volume)
Parameters	volume: 0(mute) ~ 15(max)
Return Values	void
Remarks	

2.4 SACM_Speed

API Name		SACM_Speed(int speed)
Function		speech play speed setup
Description		speech play speed setup
Header File	C	SACM_API.h
	ASM	
Syntax	C	void SACM_Speed(int speed);
	ASM	
Parameters		speed: 0(slowest) ~ 24(fastest)
Return Values		None

2.5 User_SetStartAddrManual

API Name		User_SetStartAddrManual
Function		Speech source data address defined by users
Description		Speech source data address defined by users
Header File	C	SACM_API.h
	ASM	
Syntax	C	void User_SetStartAddrManual (long address);
	ASM	
Parameters		index: speech index.
Return Values		None
Remarks		This Function is used when play the resource in Memory, For example: { ... User_SetStartAddrManual(LB2LP(_RES_VERD24M_A18_sa)); SACM_Play(SpeechIndex, DAC1+DAC2, 0); ... }

2.6 USER_SetRECStartAddr

API Name		USER_SetRECStartAddr
Function		DVR encoded data store address defined by users.
Description		DVR encoded data store address defined by users.
Header File	C	SACM_API.h
	ASM	
Syntax	C	void USER_SetRECStartAddr(long address);
	ASM	
Parameters		address: encoded data store address.
Return Values		None
Remarks		<p>This Function is used for recording into Memory, For example:</p> <pre> { ... USER_SetRECStartAddr(RECStartAddr + 2); SACM_Rec(RecMonitorOff, BIT_RATE); ... }</pre>

2.7 SACM_Play

API Name		SACM_Play(int Speech index, int Channel ,int Ramp)
Function		
Description		Releases Sound resource and ISR
Header File	C	SACM_API.h
	ASM	
Syntax	C	void SACM_Play(int Speech index, int Channel ,int Ramp);
	ASM	
Parameters		<p>Parameter:</p> <p>Speech index: -1 : Manual mode 0 - max index : Auto mode</p> <p>Channel: 1: DAC1 on 2: DAC2 on 3: DAC1,2 on</p> <p>Ramp: 0: ramp up/down off 1: ramp up on 2: ramp down on 3: ramp up/down on</p>
Return Values		None
Remarks		About Speech index:

API Name		SACM_Play(int Speech index, int Channel ,int Ramp)
		Please choose Manual mode if speech played from FAT; Please choose Auto mode if speech played from memory.
Examples		<pre> 1. Play from Memory: { ... CodecType = C_CODEC_A1800; SACM_Codec(CodecType); SACM_Play(SpeechIndex, DAC1+DAC2 ,0); ... }</pre>

2.8 SACM_ServiceLoop

API Name		SACM_ServiceLoop
Function		Decoding Resource data
Description		Decodes bit stream(or encoded data)
Header File	C	SACM_API.h
	ASM	
Syntax	C	void SACM_ServiceLoop(void);
	ASM	
Parameters		None
Return Values		None
Remarks		

2.9 SACM_Pause

API Name		SACM_Pause(void)
Function		speech pause
Description		Pauses the current speech
Header File	C	SACM_API.h
	ASM	
Syntax	C	void SACM_Pause(void);
	ASM	
Parameters		None
Return Values		None
Remarks		None

2.10 SACM_Resume

API Name		SACM_Resume(void)
Function		speech resume
Description		Resumes the paused speech.
Header File	C	SACM_API.h
	ASM	
Syntax	C	void SACM_Resume(void);
	ASM	
Parameters		
Return Values		None
Remarks		

2.11 SACM_Stop

API Name		SACM_Stop(void)
Function		speech stop
Description		Stop playing current speech
Header File	C	SACM_API.h
	ASM	
Syntax	C	void SACM_Stop(void);
	ASM	
Parameters		None
Return Values		None

2.12 SACM_Status

API Name		SACM_Status(void)
Function		Current status
Description		Gets current status: stop, play, or pause.
Header File	C	SACM_API.h
	ASM	
Syntax	C	unsigned int SACM_Status(void);
	ASM	
Parameters		None
Return Values		current status
Remarks		Return value: C_Snd_Stop(0x0000): sound play or record stop; C_Snd_Playing(0x0001): sound playing; C_Snd_DVRing(0x0002): sound recording; C_Snd_Pause(0x0004): sound play or record pause.

2.13 SACM_DVR1800_BITRATE

API Name		SACM_DVR1800_BITRATE
Function		DVR1800 bit rate setup
Description		Sets up DVR1800 bit rate
Header File	C	SACM_API.h
	ASM	
Syntax	C	void SACM_DVR1800_BITRATE(int BitRate);
	ASM	
Parameters		BitRate: The DVR1800 supported bit rate is 72000bps, 96000bps, 12kbps, 14.4kbps, 16kbps, 20kbps, 24kbps, and 32kbps
Return Values		

2.14 SACM_Rec

API Name		SACM_Rec(int RceMonitor, int bit_rate)
Function		speech DVR record start
Description		Starts recording to memory
Header File	C	SACM_API.h
	ASM	
Syntax	C	void SACM_Rec(int RceMonitor, int bit_rate);
	ASM	
Parameters		RceMonitor: 0 off 1 on bit_rate:
Return Values		None
Remarks		
Examples		<pre> 1. Record to Memory: { ... CodecType = C_CODEC_DVR1800; SACM_Codec(CodecType); USER_SetRECStartAddr(RECStartAddr + 2); SACM_Rec(RecMonitorOff,BIT_RATE); } </pre>

2.15 SACM_MP3_SetFS

API Name		SACM_MP3_SetFS
Function		Get MP3 bit rate; sample rate info of MP3 source data before playing MP3.
Description		Gets MP3 bit rate; sample rate info of MP3 source data before playing MP3.

API Name			SACM_MP3_SetFS
Header File	C		SACM_API.h
	ASM		
Syntax	C		void SACM_MP3_SetFS(void);
	ASM		
Parameters			None
Return Values			None
Remarks			programmers should call this function first before playing MP3.

2.16 SACM_MP3_Get_Time

API Name			SACM_MP3_Get_Time
Function			Get the MP3 current play time.
Description			Get the MP3 current play time.
Header File	C		SACM_API.h
	ASM		
Syntax	C		int SACM_MP3_Get_Time(void);
	ASM		
Parameters			No input parameter.
Return Values			The current MP3 playtime, the unit is second

3 Specification and System Requirement of SACM

Generalplus supports many speech algorithms that are able to play back high quality speech with efficient memory space reservation. There are three algorithms supported in GPL162XXA Mask ROM - S880, DVR1800, and MP3. The specifications of three algorithms are shown in Table 1.

Name	Sample rate (Hz)	Bit rate (k bits/s)
S880	16k	7.2, 9.6, 12.8, 14.4, 16, 18.4, 20, 23.2 and 24
DVR1800	16K	7.2, 9.6, 12k, 14.4k, 16k, 20k, 24k, 32k
MP3	8k-48k	8-320

Table 1 specifications of S880, DVR1800, and MP3

In the CODE size of these three algorithms, S880 and MP3 are totally built in the Mask ROM and DVR1800 is partially built in it. In the RAM size shown in Table 2, it consists of working RAM, decode-in RAM, and DAC-out RAM in which the system requirements are depicted in Table 2.

Name	RAM size (word)	Code size (word)
S880	1297	None
DVR1800	1715	11194
MP3	7889	None

Table 2 system requirement of S880, DVR1800, and MP3

4 CPU Performance of SACM

The CPU performance of S880, DVR1800, and MP3 are shown in Table 3. The test program runs at 96MHz CPU clock with 4KB I-cache is turned ON. Please note that the test program runs without any other application which means the hit rate of I-cache may vary depending on user's application case by case.

<i>Name</i>	<i>CPU performance</i>	<i>Test pattern</i>
S880	16%	7.2kbps
DVR1800	8%	32kbps
MP3	28.6%	44.1k sample, 320kbps

Table 3 CPU performance of S880, DVR1800, and MP3

5 Service Loops for SACM

In background service loop, the program will check whether or not the service-loop tasks should take place right after the exiting interrupt, and therefore, the tasks of demanding service loop in background is able to lead more efficiency and flexibility for user-defined tasks in main program. Since the background service loop eliminates some unnecessary overhead, the background service loop programming can conserve more CPU resources and increase efficiency as well.

Example:

Background service loop:

In main.c:

```
int main()
```

```
{  
    System_Initial();  
    CodecType = C_CODEEC_DVR1800;  
    SACM_Initial();  
    SACM_Codec(CodecType);  
    SACM_Play(0,DAC1+DAC2,0);  
    while (1)  
    {  
        System_ServiceLoop();  
    }  
    return 0;  
}
```

In isr.asm

```
_IRQ0:
```

```
    //add your code here
```

```
    call _F_SACM_IRQ0_Service
```

```
_IRQ1:
```

```
    call _F_SACM_IRQ1_Service
```

```
_IRQ3:
```

```
    call _F_SACM_IRQ3_Service
```