Play MIDI

Sunmidiar2 Quick Start:

SunMidiar2, a powerful tool, authored by Sunplus Tech Co., Ltd mainly for making and editing Tone Colors and playing them on an ICE, offers a more effective way than ever to manage MIDI files and Instrument Library, to make and edit tone colors, and to play MIDI and tone colors on PC/ICE. With the user-friendly interface and flexible editing modes, the SunMidiar2 is capable of helping you to work easily and efficiently.

Step1: Installing and Running the SunMidiar2

The current version of the tool is capable of running under Windows98® and Windows2000®. The propositional minimum system requirements are:

- CPU clock: 133MHz
- Capacity of memory: 64MB
- Free hard disk space: 20MB

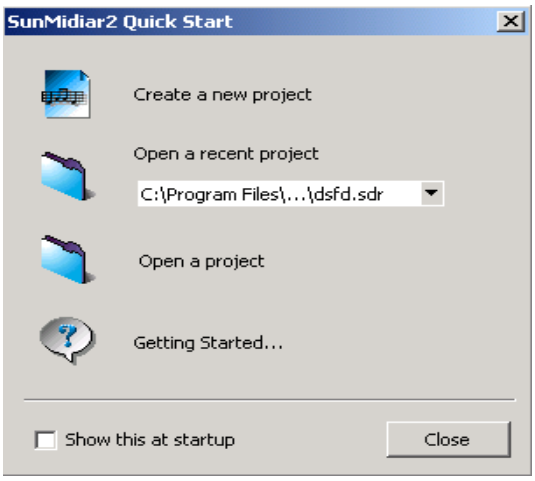
Take the following steps, you can install the SunMidiar2 on your computer:

1. Extract the zip file and execute the installation file (*setup. exe*).
2. Follow the on-screen prompts and the SunMidiar2 will be installed on your computer.

During installation, you will be asked to select one installation type: **Typical**, **Compact**, or **Custom**. SUNPLUS highly recommends you to install the tool with the "Typical" which applies for most users. Selecting "Compact" results in the minimal configuration installed on your computer. The "Custom" option is usually for advanced users.

To run the SunMidiar2, click [Start] →[Program Files] →[Sunplus] →[SunMidiar2] → [SunMidiar2] from the Windows task bar.

Especially, when SunMidiar2 is started up for the first time, the following **SunMidiar2 Quick Start** dialog box will be opened over the main user interface of SunMidiar2:



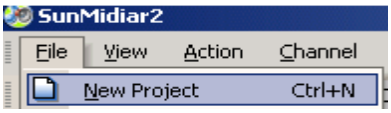
In the box, each icon is a button; you can directly create/open a project or get the Help by clicking it. For example, if you click the button before:

- Create a new project: A New Project dialog box will be popped up for you to create a new project.
- Open a recent project: Up to 8 recently-opened projects are listed here. When the tool is opened at first time, this item is disabled.
- *Open a project:* An Open dialog box will be opened for you to open a project.
- *Getting Started:* The Help document of this tool will appear.

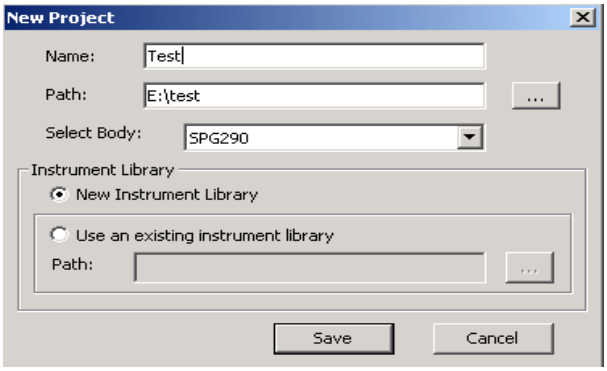
If you want to do nothing just click the Close button to close the box. Also, you can make this dialog box appeared/hidden when this tool is started up by checking/unchecking (/) **Show this at startup** and clicking the Close button. On the other hand, you can call it by clicking [File]→ [Quick Start] when it is hidden.

Step 2:Create a project

1. Create a new project using the New Project command or toolbar button.



2. Input some parameters in the New Project dialog box.



•Type "Test" as the project name in the **Name** text box

•Specify *' E:\test "* in the **Path** box

•Select **SPG290** as the body in the **Select Body** pull-down menu

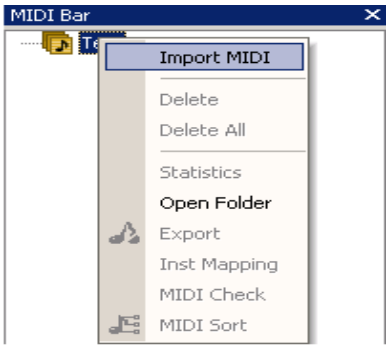
•Choose **New Instrument Library** and click the Save button.

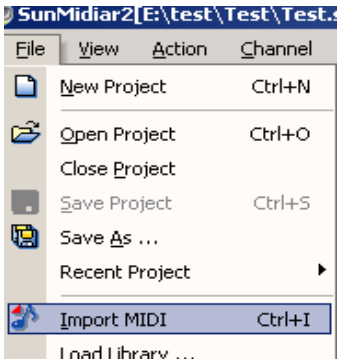
When you are finished, the **Test** project is created. Meanwhile, an empty folder " **Test**" is created in the path you specified and the path is shown on the title bar.

Step 3:Import a MIDI file into the project

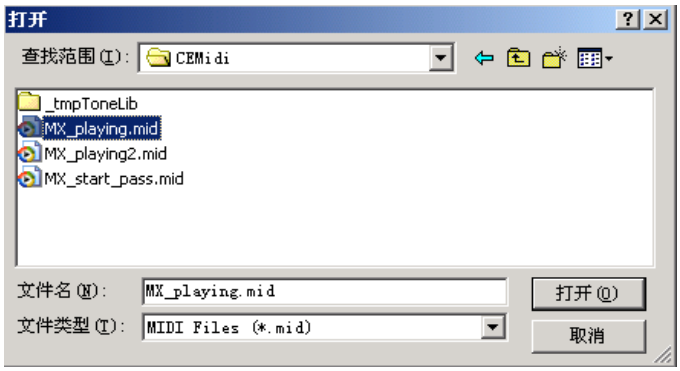
1. Start from clicking toolbar button.

2. Begin from clicking the Import MIDI menu command on the File menu or right-click menu of the MIDI tree, or pressing shortcut keys.

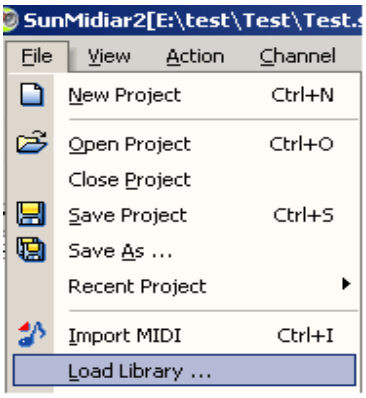


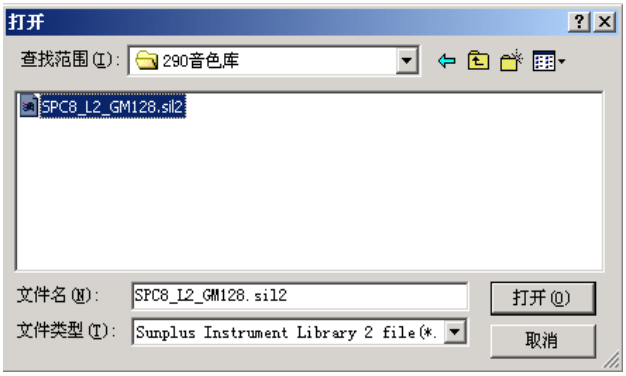


3. Select .mid file(s) in the Open dialog box
4. Click **Open** on the dialog box.

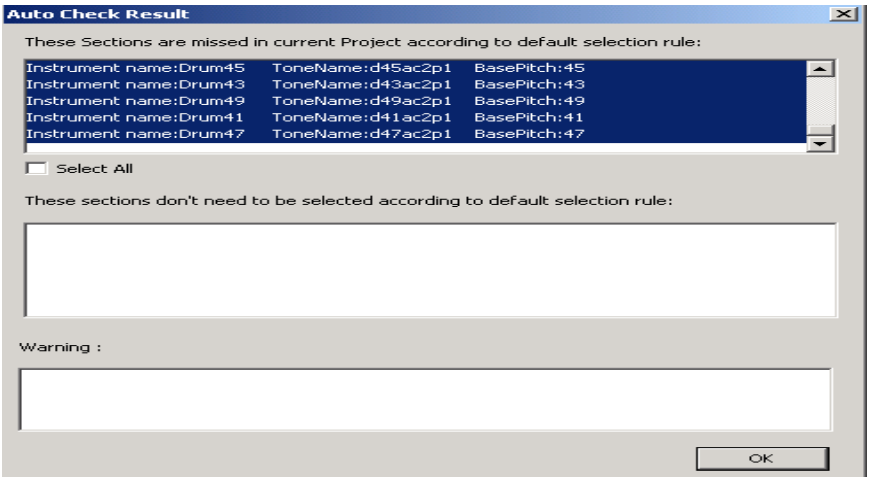


Step 4: Load Library and select .sil2 file in the Open dialog box and then click Open on the dialog box.



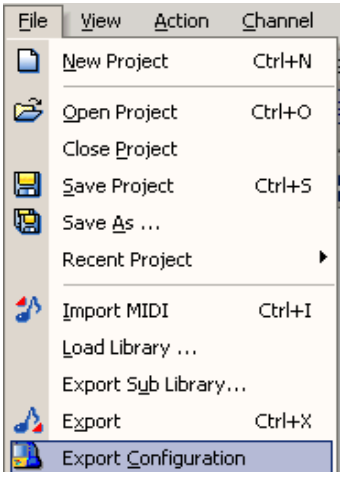


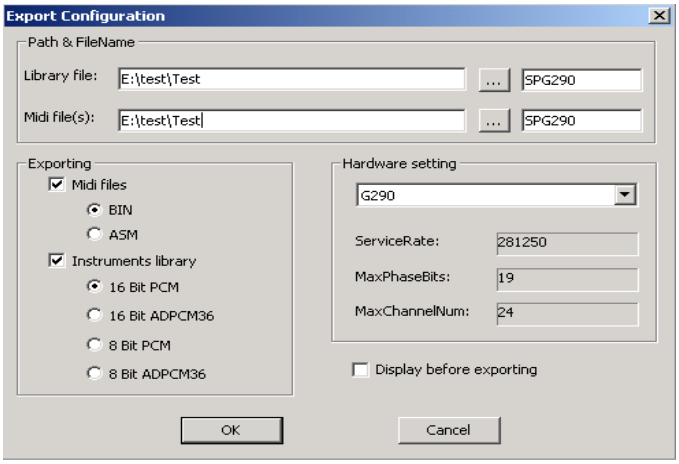
Notice: On the Auto Check Result dialog box, click OK button.



Step 5: Export the MIDI File

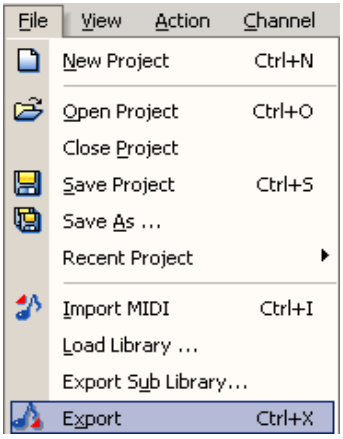
1. Set export configuration on the Export Configuration and click OK button.





2.Press corresponding toolbar buttons to export MIDI file

3. Press corresponding menu commands to export MIDI file



Step 6: Save the Project

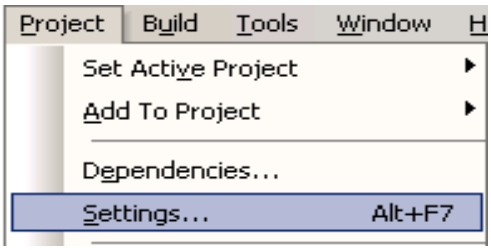
1.Click the  (Save Project) toolbar button.

2.Click [File]→[Save Project].

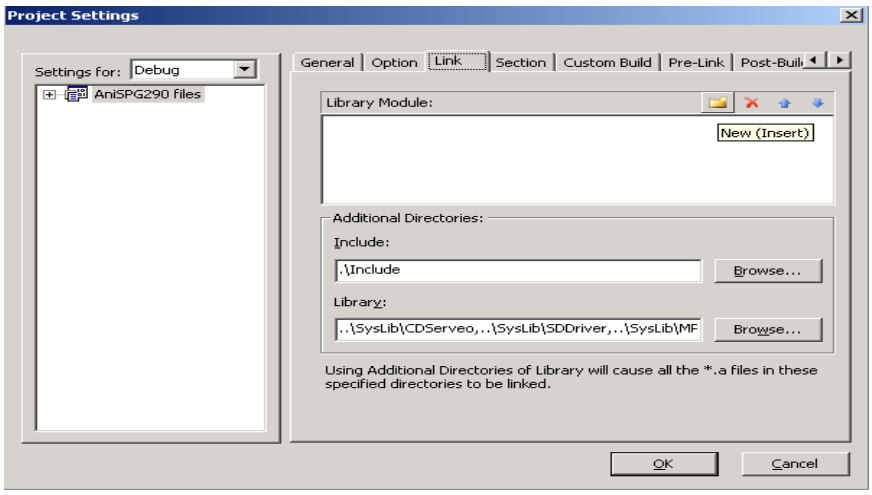
How to use those files in the program?

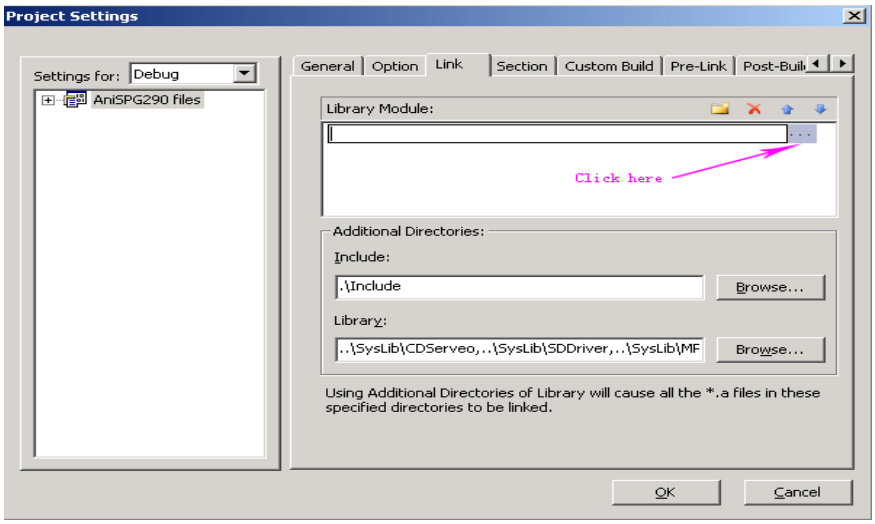
After exporting MIDI file with Sunmeiar2, we can use those midi files in our program. Please refer to the following steps.

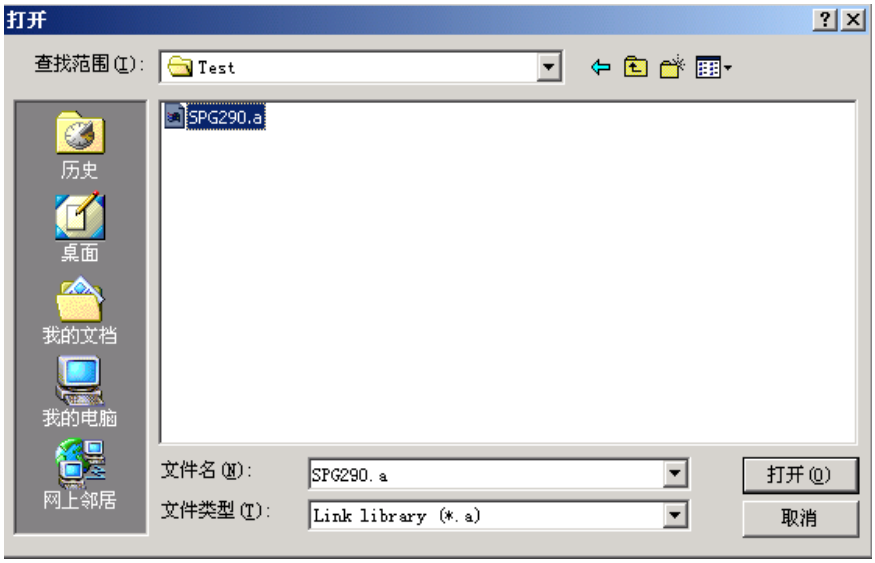
Step1: Click the Settings… menu command on the Project menu;



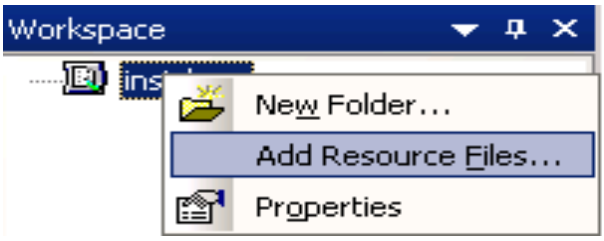
Step2:On the Project Settings dialog box, click Link button and then click New button to insert .a file which exported by Sunmdediar2 and SPUDrv.a in the Library Module text box (See the following graphics).







Step3: Double click .rc file in folder Resource files, and right click "instdrum" to add resource files (*.bin) that exported by Sunmediar2.



Step4: Rebuild all the files and all the resource files will be called in Resource.h .

```
//Resource.h
// SUNPLUS  S+core IDE generated include file.
#ifndef ANISPG290_RESOURCE_H__26500_6334__INCLUDED_
#define ANISPG290_RESOURCE_H__26500_6334__INCLUDED_

extern                                  unsigned
 binary E  SunPlus Studio SPG290 Sample Code SPU Music MIDI MX playing2 bin sta
#define                           RES MX PLAYING2 BIN
char*)&_binary_E__SunPlus Studio SPG290 Sample Code SPU Music MIDI MX playing2 
extern                                  unsigned
 binary E  SunPlus Studio SPG290 Sample Code SPU Music MIDI MX playing2 bin end
#define                         RES MX PLAYING2 BIN END
char*)&_binary_E__SunPlus_Studio_SPG290_Sample Code SPU Music MIDI MX playing2 
... ...


#endif //ANISPG290_RESOURCE_H__26500_6334__INCLUDED
```

Step5: Create Midi table array named 'L_MidiFiles'. Reference sample code.

```
//MIDITable.c
U8* L_MidiFiles[]=
{
    RES_MX_PLAYING2_BIN,
    RES_MX_PLAYING_BIN,
    RES_MX_START_PASS_BIN
};
```

Step6: Initialize SPU, Play Midi. Reference sample code.

```
//MIDITable.c
```

```
... ...
void UserPlayMidi(void)
{
    //Initialize SPU
    InitSPU();
    //set midi channel mask
    SetMidiChannelMask(0x00FFFFFF);
    //set midi volume
    SetMidiVolume(127);
    //stop midi
    StopMidi();
    //play midi
    PlayMidi(0, SS_PLAYMIDI_INFINITY);
}
```

Step7: Call SPU IRQ function 'void SPU_IRQ_Service(void)' in IRQ62. Reference sample code.

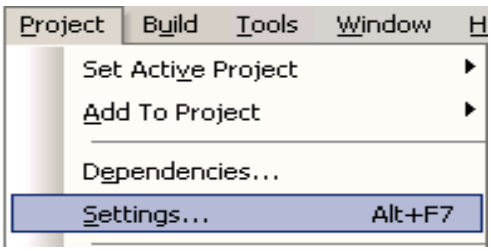
```
//Sys_IRQ.c
... ...
extern void SPU_IRQ_Service(void);
void IRQ62(void)     //SPU Beat Count ISR
{
    SPU_IRQ_Service();
}
```

Play MP3

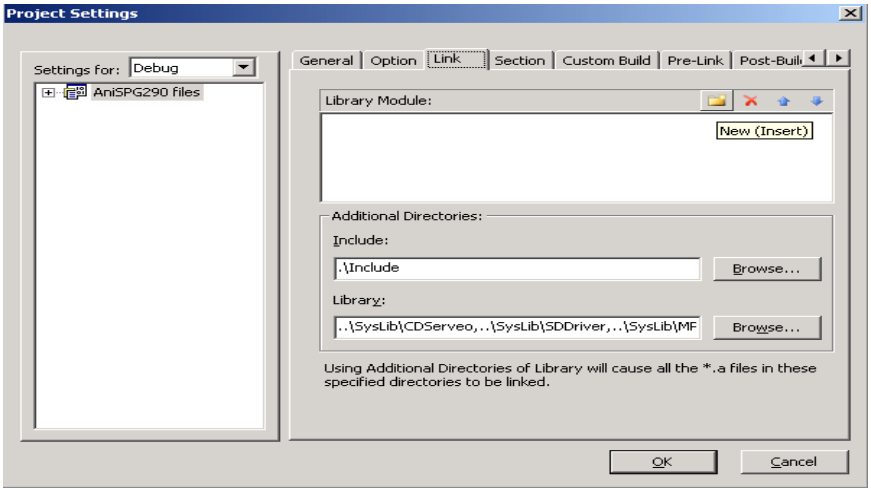
How to play mp3

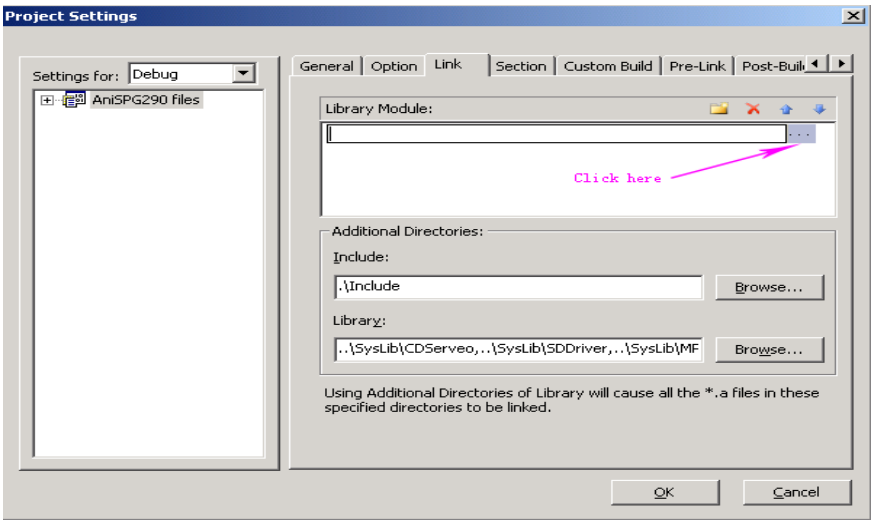
After preparing MP3 file, we can use these files in our program. Please refer to the following steps.

Step1: Click the Settings… menu command on the Project menu;

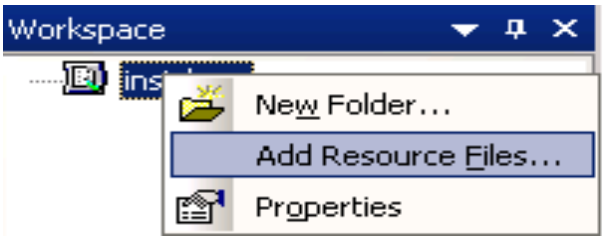


Step2:On the Project Settings dialog box, click Link button and then click New button to insert MP3Drv.a and MP3Core3.a which local in '..\SysLib\MP3' in the Library Module text box (See the following graphics).





Step3: Double click .rc file in folder Resource files, and right click "instdrum" to add resource files (*.mp3).



Step4: Rebuild all the files and all the resource files will be called in Resource.h .

```
//Resource.h
// SUNPLUS  S+core IDE generated include file.
#ifndef ANISPG290_RESOURCE_H__26500_6334__INCLUDED_
#define ANISPG290_RESOURCE_H__26500_6334__INCLUDED_

... ...
extern                                  unsigned
```

```
 binary E  SunPlus Studio SPG290 Sample Code SPU Music Mp3 BlueSprite mp3 start
#define                          RES BLUESPRITE MP3                          (u
char*)&_binary_E__SunPlus_Studio_SPG290_Sample Code SPU Music Mp3 BlueSprite mp
extern                                  unsigned
_binary_E__SunPlus_Studio_SPG290_Sample_Code_SPU_Music_Mp3_BlueSprite_mp3_end;
#define                        RES BLUESPRITE MP3 END                        (u
char*)&_binary_E__SunPlus_Studio_SPG290_Sample_Code_SPU_Music_Mp3_BlueSprite mp

#endif //ANISPG290_RESOURCE_H__26500_6334__INCLUDED
```

Step5: Create MP3 table array named 'L_MP3Files'. Reference sample code.

```
//MP3Table.c
U8* L_MP3Files[]=
{
    RES_BLUESPRITE_MP3
};
```

Step6: Play MP3. Reference sample code.

```
//MP3Table.c
... ...
U32 Get_MP3_Start_Address(U32 MP3_Index)
{
    U32 *Address;

    Address = (U32*)&L_MP3Files;
    Address = (U32*)(*(U32*)((U32)Address+MP3_Index));
    return((U32)Address);
}

void UserPlayMp3(void)
{
    U32 Address;

    //Init_MP3();
    Address = Get_MP3_Start_Address(0);
    Play_MP3(Address);
    Repeat_ON_MP3();     //MP3 play repeat enable
    MP3_Service_Loop();
    MP3_Service_Loop();
    MP3_Service_Loop();
```

```
    MP3_Service_Loop();
    MP3_Service_Loop();
    MP3_Service_Loop();
    MP3_Service_Loop();
    MP3_Service_Loop();
}
```

Step7: Call some functions about play mp3 in IRQ63. Reference sample code.

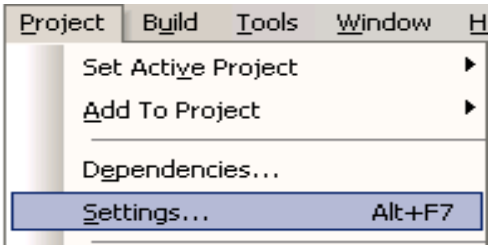
```
//Sys_IRQ.c
... ...
extern short  TempPCM[];
extern int Need_PCM_Flag;
extern void FillSoftFIFO(unsigned short *TempPCM);
extern void MP3_Service_Loop(void);
void IRQ63(void)
{
    FillSoftFIFO(TempPCM);
    Need_PCM_Flag = 1;                //  Set  flag  to  start  MP3
decoder
    *P_SPU_SoftIRQEN = 0xC000 | 0x0004 | 0x0003;   //  for  stereo,
8KBytes
    MP3_Service_Loop();
    MP3_Service_Loop();
    MP3_Service_Loop();
}
```

Play PCM / ADPCM

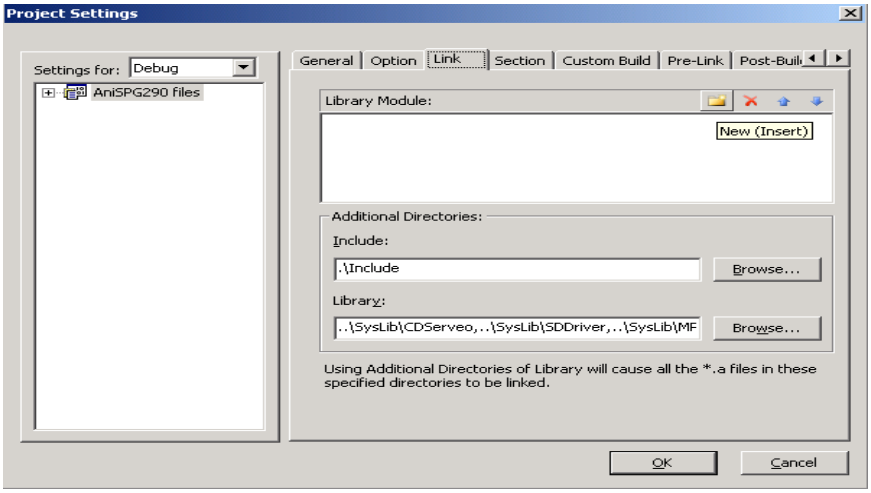
How to play PCM / ADPCM

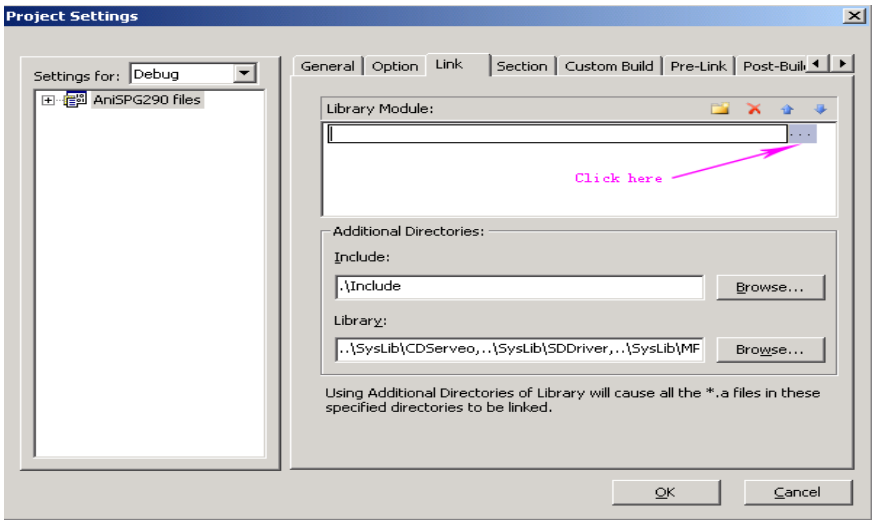
After preparing PCM or ADPCM file (*.drm), we can use these files in our program. Please refer to the following steps.

Step1: Click the Settings… menu command on the Project menu;

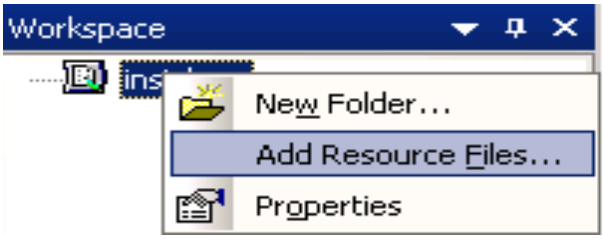


Step2:On the Project Settings dialog box, click Link button and then click New button to insert SPUDrv.a which local in '..\SysLib\SPU' in the Library Module text box (See the following graphics).





Step3: Double click .rc file in folder Resource files, and right click "instdrum" to add resource files (*.drm).



Step4: Rebuild all the files and all the resource files will be called in Resource.h .

```
//Resource.h
// SUNPLUS  S+core IDE generated include file.
#ifndef ANISPG290_RESOURCE_H__26500_6334__INCLUDED_
```

```
#define ANISPG290_RESOURCE_H__26500_6334__INCLUDED_

... ...
extern                                    unsigned
 binary E  SunPlus Studio SPG290 Sample Code SPU Music PCM ADPCM 16Bit Drm drm s
#define                              RES 16BIT DRM DRM
char*)& binary E  SunPlus Studio SPG290 Sample Code SPU Music PCM ADPCM 16Bit Dr
extern                                    unsigned
 binary E  SunPlus Studio SPG290 Sample Code SPU Music PCM ADPCM 16Bit Drm drm e
#define                             RES 16BIT DRM DRM END
char*)& binary E  SunPlus Studio SPG290 Sample Code SPU Music PCM ADPCM 16Bit Dr

extern                                    unsigned
_binary_E__SunPlus Studio SPG290 Sample Code SPU Music PCM ADPCM Adpcm drm start
#define                                RES ADPCM DRM
char*)& binary E  SunPlus Studio SPG290 Sample Code SPU Music PCM ADPCM Adpcm dr
extern                                    unsigned
_binary_E__SunPlus_Studio_SPG290_Sample_Code_SPU_Music_PCM_ADPCM_Adpcm_drm_end;
#define                              RES ADPCM DRM END
char*)& binary E  SunPlus Studio SPG290 Sample Code SPU Music PCM ADPCM Adpcm dr
#endif //ANISPG290_RESOURCE_H__26500_6334__INCLUDED
```

Step5: Create PCM or ADPCM table array named '`L_PCMFiles`' or '`L_ADPCMFiles`'. Reference sample code.

```
//PCMTable.c
U8* L_PCMFiles[]=
{
    RES_16BIT_DRM_DRM
};

U8* L_ADPCMFiles[]=
{
    RES_ADPCM_DRM
};
```

Step6: Play PCM or ADPCM. Reference sample code.

```
//PCMTable.c
... ...
void UserPlayPCM(void)
{
    InitSPU();//Initialize SPU
```

```
    SetMidiChannelMask(0x00FFFF00);//set bit0 ~ bit7 play PCM or
ADPCM

    TM_PlayPCM(L_PCMFiles[0], 63, 127);//play pcm, volume is 127
}

void UserPlayADPCM(void)
{
    InitSPU();
    SetMidiChannelMask(0x00FFFF00);

    TM_PlayPCM_FixCH(0, L_ADPCMFiles[0], 63, 127);//play ADPCM at CH0
}

void StopAllCH(void)
{
    U8 i;

    for (i = 0; i < 24; i ++)
    {
        TM_StopPCM(i);//Stop all Channels
    }
}
```

Step7: Call some functions about play PCM or ADPCM in IRQ62. Reference sample code.

```
//Sys_IRQ.c
... ...
extern void SPU_IRQ_Service(void);
void IRQ62(void)     //SPU Beat Count ISR
{
    SPU_IRQ_Service();
}
```