API Functions

ClearDatabase

Clear the DAB programs stored in the module's database.

```
BOOL ClearDatabase(void);
```

Parameters

none

Return Values

Return `true` if successful, `false` if failed.

Remark

This function is unnecessary under normal operation. `DABAutoSearch()` will execute this function internally.

CloseRadioPort

Close the COM port of the radio.

```
BOOL CloseRadioPort(void);
```

Parameters

none

Return Values

Return `true` if successful, `false` if failed.

DABAutoSearch

Auto search DAB channels. Current stored DAB channels will be cleared.

```
BOOL DABAutoSearch(
    unsigned char startindex,
    unsigned char endindex
);
```

Parameters

startindex

[in] Starting index to search from. See remark below.

endindex

[in] Ending index to search to. See remark below.

Return Values

Return `true` if successful, `false` if failed.

Remark

If module is operating outside of China Band, calling `DABAutoSearch(0,40)` will speed up the searching process.

DAB Frequency Index

BAND 3

Frequency	Alias	Index
174.928MHz	5A	0
176.64MHz	5B	1
178.352MHz	5C	2
180.064MHz	5D	3
181.936MHz	6A	4
183.648MHz	6B	5
185.36MHz	6C	6
187.072MHz	6D	7
188.928MHz	7A	8
190.64MHz	7B	9
192.352MHz	7C	10
194.064MHz	7D	11
195.936MHz	8A	12
197.648MHz	8B	13
199.36MHz	8C	14
201.072MHz	8D	15
202.928MHz	9A	16
204.64MHz	9B	17
206.352MHz	9C	18
208.064MHz	9D	19

BAND 3

Frequency	Alias	Index
209.936MHz	10A	20
210.096MHz	10N	21
211.648MHz	10B	22
213.36MHz	10C	23
215.072MHz	10D	24
216.928MHz	11A	25
217.088MHz	11N	26
218.64MHz	11B	27
220.352MHz	11C	28
222.064MHz	11D	29
223.936MHz	12A	30
224.096MHz	12N	31
225.648MHz	12B	32
227.36MHz	12C	33
229.072MHz	12D	34
230.784MHz	13A	35
232.496MHz	13B	36
234.208MHz	13C	37
235.776MHz	13D	38
237.488MHz	13E	39
239.2MHz	13F	40

CHINA BAND

Frequency	Alias	Index

168.160 MHz	CN 6A	41
169.872 MHz	CN 6B	42
171.584 MHz	CN 6C	43
173.296 MHz	CN 6D	44
175.008 MHz	CN 6N	45
176.720 MHz	CN 7A	46
178.432 MHz	CN 7B	47
180.144 MHz	CN 7C	48
181.856 MHz	CN 7D	49
184.160 MHz	CN 8A	50
185.872 MHz	CN 8B	51
187.584 MHz	CN 8C	52
189.296 MHz	CN 8D	53
191.008 MHz	CN 8N	54

CHINA BAND

Frequency	Alias	Index
192.720 MHz	CN 9A	55
194.432 MHz	CN 9B	56
196.144 MHz	CN 9C	57
197.856 MHz	CN 9D	58
200.160 MHz	CN 10A	59
201.872 MHz	CN 10B	60
203.584 MHz	CN 10C	61
205.296 MHz	CN 10D	62
207.008 MHz	CN 10N	63
208.720 MHz	CN 11A	64
210.432	CN	65
		66
		67
		68
		69
		70
		71

MHz	11B
212.144	CN
MHz	11C
213.856	CN
MHz	11D
216.432	CN
MHz	12A
218.144	CN
MHz	12B
219.856	CN
MHz	12C
221.568	CN
MHz	12D

GetDataRate

Get the current DAB data rate.

```
int GetDataRate(void);
```

Parameters

none

Return Values

Return the current DAB data rate in kbps.

GetEnsembleName

Get the ensemble name of the current program.

```
BOOL GetEnsembleName(
    long dabIndex,
    char namemode,
    wchar_t * programName
);
```

Parameters

dabIndex

> [in] The DAB program index of 0 to `GetTotalProgram()-1` to get the ensemble from.

namemode

> [in] Setting this parameter to 0 for abbreviated name or 1 for long name. Only valid when mode is DAB

programName

> [out] pointer to the buffer of the returned text. This pointer will need to have at least 150 wchar_t characters allocated. In Windows wchar_t size is 2 bytes and in Linux wchar_t is 4 bytes.

Return Values

Return `true` if successful, `false` if failed.

Remark

Please read remark for function `GetProgramText()`.

GetFrequency

Get the currenct DAB frequency index in while DAB is auto searching.

```
char GetFrequency(void);
```

Parameters

none

Return Values

Return the DAB frequency index of the current auto search.

GetPlayIndex

Get the index of current playing DAB stream or the current playing frequency.

```
long GetPlayIndex(void);
```

Parameters

none

Return Values

If the radio is in DAB mode, return the current playing DAB index within 0 to `GetTotalProgram()-1`. If the radio is in FM mode, the current playing FM frequency in kHz is returned, eg 94500 is 94.5Mhz.

Remark

`GetPlayMode()` is required to be called before or after this function to determine the radio mode in order to differentiate the returned value.

GetPlayMode

Determine if the current mode is DAB or FM.

```
char GetPlayMode(void);
```

Parameters

none

Return Values

Return 0 when current mode is DAB or 1 when current mode is FM. Any other value is invalid and -1 when function failed.

GetPlayStatus

Determine if the current radio status is playing, searching, tuning, stop sorting or reconfiguring.

```
char GetPlayStatus(void);
```

Parameters

none

Return Values

Return the following values

```
0 = Playing
1 = Searching
2 = Tuning
3 = Stop
4 = Sorting
5 = Reconfiguring
```

Other value invalid and -1 if function failed.

Remark

Take note that when `OpenRadioPort()` is setup with a `true` value passed in to use hardware mute, this function need to be repeatedly called in a loop as the API will only release the hardware mute when the play status is 0 - playing. If the play status is other than 0

- playing, the volume will be muted and even when `SetVolume()` is called to the max volume, the sound will be very soft.

GetPreset

Get the preset DAB index or preset FM frequency. The module is able to store 10 DAB and 10 FM preset.

```
long GetPreset(
    char mode,
    char presetindex
);
```

Parameters

mode

[in] 0 to get DAB preset or 1 to get FM preset

presetindex

[in] Preset location from 0 to 9

Return Values

If mode is DAB (mode 0), this value contains the DAB program index. If mode is FM (mode 1), this value contains the FM frequency in kHz , eg 94500 for 94.5Mhz.

GetProgramName

Get the name of the current program.

```
BOOL GetProgramName(
    char mode,
    long dabIndex,
    char namemode,
    wchar_t * programName
);
```

Parameters

mode

> [in] 0 if mode is DAB or 1 if mode is FM.

dabIndex

> [in] Index of the DAB channel from 0 to `GetTotalProgram()-1` if mode is DAB or this parameter is ignored when mode is FM.

namemode

> [in] Setting this parameter to 0 for abbreviated name or 1 for long name. Only valid when mode is DAB.

programName

> [out] pointer to the buffer of the returned text. This pointer will need to have at least 150 wchar_t characters allocated. In Windows wchar_t size is 2 bytes and in Linux wchar_t is 4 bytes.

Return Values

Return `true` if successful, `false` if failed.

Remark

Please read remark for function `GetProgramText()`.

GetProgramText

Get the RDS text of the current stream.

```
char GetProgramText(
    wchar_t * programText
);
```

Parameters

programText

[out] pointer to the buffer of the returned text. This pointer will need to have at least 150 wchar_t characters allocated. In Windows wchar_t size is 2 bytes and in Linux wchar_t is 4 bytes.

Return Values

Return 0 if successful with text, -1 if failed. If the same text has been retrieved previously, 1 will be returned.

Remark

Linux programmer porting this function need to be aware of the different sizes of `wchar_t` when porting this function, proper routines are required to convert the text into a printable string. For example, capital "Z"

Windows `wchar_t` is `0x5A, 0x00`
Linux `wchar_t` is `0x5A, 0x00, 0x00, 0x00`

Printing strings of `wchar_t` in Linux requires `wprintf(L"%ls", buffer);`

GetProgramType

Get the current playing program type to be used to identify the genre.

```
char GetProgramType(
    char mode,
    long dabIndex
);
```

Parameters

mode

[in] 0 if current mode is DAB or 1 if current mode is FM.

dabindex

[in] the index of the DAB channel required to get the program type. Function will ignore this value in if `mode` is 1 (FM).

Return Values

Return the following values:

```
0 = <Prg Type N/A>
1 = News
2 = Current Affairs
3 = Information
4 = Sport
5 = Education
6 = Drama
7 = Arts
8 = Science
9 = Talk
10 = Pop Music
11 = Rock Music
12 = Easy Listening
13 = Light Classical
14 = Classical Music
15 = Other Music
16 = Weather
17 = Finance
18 = Children's
19 = Factual
20 = Religion
21 = Phone In
22 = Travel
23 = Leisure
24 = Jazz and Blues
25 = Country Music
26 = National Music
27 = Oldies Music
28 = Folk Music
29 = Documentary
30 = <Undefined>
31 = <Undefined>
```

GetSignalStrength

Get the signal strengh of the current playing stream.

```
char GetSignalStrength(
    int *biterror
);
```

Parameters

biterror

> [out] 0 if FM mode and bit error rate if DAB mode. ** ignore this out value until further notice.

Return Values

Signal strength in 0 to 100 percent.

GetStereo

Get the stereo reception status of the current playing stream.

```
char GetStereo(void);
```

Parameters

none

Return Values

Return the following values or -1 if function failed.

```
0: Stereo
1: Joint stereo
2: Dual channel
3: Single channel (mono)
```

GetStereoMode

Get the current stereo mode in the radio configuration.

```
char GetStereoMode(void);
```

Parameters

none

Return Values

Return 0 if current mode is forced mono, 1 if auto strereo or -1 if function failed.

GetTotalProgram

Get the total number of DAB programs stored in the module.

```
long GetTotalProgram(void);
```

Parameters

none

Return Values

Total DAB programs stored in the module.

GetVolume

Get the current volume.

```
char GetVolume(void);
```

Parameters

none

Return Values

Current volume in 0 to 16 or -1 if failed.

HardResetRadio

Hard reset the radio module by pulling the RESET pin LOW.

```
 BOOL HardResetRadio(void);
```

Parameters

none

Return Values

Return `true` if successful, `false` if failed.

IsSysReady

Check if the module is ready to receive command.

```
BOOL IsSysReady(void);
```

Parameters

none

Return Values

Return `true` if successful, `false` if failed.

NextStream

Forward to the next available stream in the current mode. When radio is in DAB mode, the dabindex will be incremented and then played. When the radio is in FM mode, search by increasing the FM frequency until a channel is found.

```
BOOL NextStream(void);
```

Parameters

none

Return Values

Return `true` if successful, `false` if failed.

OpenRadioPort

Open the COM port of the radio and set mute behavior.

```
BOOL OpenRadioPort(
    LPCSTR port,
    BOOL usehardmute
);
```

Parameters

port

```
    [in] COM port of the radio. Example "\\.\COM1", refer to
    http://support.microsoft.com/kb/115831 for details.
```

usehardmute

```
[in] true to enable or false disable hard mute.  Hard mute will turn on the
MOSFET on the board to shunt transitional noise like psss or pop sound.
```

Return Values

Return `true` if successful, `false` if failed.

PlayStream

Play radio stream in FM or DAB.

```
BOOL PlayStream(
    char mode,
    unsigned long channel
);
```

Parameters

mode

> [in] The mode of the radio, 0 is DAB and 1 is FM.

channel

> [in] When `mode` is DAB (mode 0), this value is the index of the DAB channels from 0 to `GetTotalProgram()-1`. When `mode` is FM (mode 1), this value is the FM frequency in kHz, eg 105000 is 105Mhz, eg 94500 is 94.5Mhz.

Return Values

Return `true` if successful, `false` if failed.

PrevStream

Backward to the previous available stream in the current mode. When radio is in DAB mode, the dabindex will be decremented and then played. When the radio is in FM mode, search by decresing the FM frequency until a channel is found.

```
BOOL PrevStream(void);
```

Parameters

none

Return Values

Return `true` if successful, `false` if failed.

SetPreset

Store program into preset location.

```
BOOL SetPreset(
    char mode,
    char presetindex,
    unsigned long channel
);
```

Parameters

mode

> [in] 0 to store DAB program, 1 to store FM program.

presetindex

> [in] Preset location to be stored from 0 to 9.

channel

> [in] If mode is DAB (mode 0), this parameter is the DAB program index. If mode is FM (mode 1), this parameter is the FM frequency to be stored in kHz, eg 94500 for 94.5Mhz.

Return Values

Return `true` if successful, `false` if failed.

SetStereoMode

Set radio to forced mono or auto detect stereo mode.

```
BOOL SetStereoMode(
    char mode
);
```

Parameters

mode

> [in] If mode is 0, the radio will be forced into mono mode. If mode is 1, the radio will auto detect stereo mode, switching to mono when reception is poor.

Return Values

Return `true` if successful, `false` if failed.

SetVolume

Set the volume of the radio.

```
BOOL SetVolume(
    char volume
);
```

Parameters

volume

[in] A char value of the volume from 0 to 16

Return Values

Return `true` if successful, `false` if failed.

StopStream

Stop currently played FM or DAB stream.

```
BOOL StopStream(void);
```

Parameters

none

Return Values

Return `true` if successful, `false` if failed.

VolumeMinus

Minus one volume step from the current volume.

```
char VolumeMinus(void);
```

Parameters

none

Return Values

Volume value of 0 to 15 after executing successfully or -1 if failed.

VolumeMute

Mute the volume.

```
void VolumeMute(void);
```

Parameters

none

Return Values

none

VolumePlus

Add one volume step to the current volume.

```
char VolumePlus(void);
```

Parameters

none

Return Values

Volume value of 0 to 16 after executing successfully or -1 if failed.