DAB dongle demo(Win console interface)

1. DAB_OpenDevice

This function open the DAB device for console.

```
int DAB_OpenDevice
(
    void* parameter
);
```

Parameters

void* parameter：It always be NULL.

Return Value

BYTE: Return the open state.

Requirements

Function	Required library
DAB_OpenDevice	DabWin32.dll

2. DAB_Scan

This function scan the Ensemble and Service information according to frequency.

```
int DAB_Scan
(
    LWORD EnsFreq
);
```

Parameters

LWORD EnsFreq: the frequency. **EnsFreq = frequency*1000**

Return Value

int: Return Service Num or scan failed state.

Requirements

Function	Required library
DAB_scan	DabWin32.dll

3. DAB_GetInfo

This function get the service information according the index.

```
SERVINFORM DAB_GetInfo
(
    BYTE list
);
```

Parameters

BYTE list: the service index

Return Value

```
struct _SERVINFORM_
{
    BYTE serviceLable[32];
    BYTE EnsembleName[32];
    BYTE SubChannleId;
    WORD ServiceType;
    WORD BitRateKhz;
    BYTE   SrvCharSet;
    BYTE   TotalAudioNum;
    BYTE   TotalVideoNum;
    BYTE   TotalDataNum;
}SERVINFORM;
```

Return the service information include the Ensemble name, service label etc.

Requirements

Function	Required library
DAB_GetInfo	DabWin32.dll

4. DAB_SelService

This function selects the service specified by bySeviceIndex

```
int DAB_SelService
(
    BYTE bySeviceIndex
);
```

Parameters

BYTE bySeviceIndex: the index of service which is selected.

Return Value

int: Return the play state.

Requirements

Function	Required library
DAB_SelService	DabWin32.dll

5. DAB_SetChnlNum

This function sets the number of service channel.

```
int DAB_SetChnlNum
(
    BYTE byChnlNum;
);
```

Parameters

BYTE byChnlNum: the index of service which is selected.

Return Value

int: Return the set channel number.

Requirements

Function	Required library
DAB_SetChnlNum	DabWin32.dll

6. DAB_GetVersion

This function gets the version of software.

```
int DAB_GetVersion
(
    void
);
```

Parameters

void

Return Value

int: Return the version number.

Requirements

Function	Required library
DAB_GetVersion	DabWin32.dll

7. DAB_CloseDevice

This function closes the DAB device.

```
int DAB_CloseDevice
(
    void
);
```

Parameters

void

Return Value

int: Return the close state.

Requirements

Function	Required library
DAB_CloseDevice	DabWin32.dll

8. DAB_ReadMSC

This function reads the MSC data from SM342 to mplayer.

```
int DAB_ReadMSC
(
    char* buffer,
    int max_length
);
```

Parameters

Char *buffer: return the MSC data;

int max_length: the length of data buffer which is needed by mplayer.

Return Value

int: return the actual length of data buffer.

Requirements

Function	Required library
DAB_ReadMSC	DabWin32.dll

DAB dongle demo(mplayer vstream interface)

1. vstream_open

This function opens the DAB device for Mplayer, calling the function of DABvstream_open().

```
int vstream_open
(
    void* parameter
);
```

Parameters

void* parameter：It always be NULL.

Return Value

BYTE: Return the open state.

Requirements

Function	Required library
vstream_open	VSTREAM.dll

2. vstream_read

This function read the data from SM342 to mplayer. It calls the function of DABvstream_read().

```
int vstream_read
(
```

```
        char* buffer,
        int max_length
    );
```

Parameters

Char *buffer: return the data;

int max_length: the length of data buffer which is needed by mplayer.

Return Value

int: return the actual length of data buffer.

Requirements

Function	Required library
vstream_read	VSTREAM.dll

3. vstream_close

This function close the DAB function. It is called in the Mplsyer.

```
    int DABvstream_close
    (
        void
    );
```

Parameters

void

Return Value

int: Return the close state.

Requirements

Function	Required library
vstream_close	VSTREAM.dll

Sample code

```
if ( TRUE == DAB_OpenDevice (NULL) )
{
	Num = DAB_Scan();
}
else
{
	return;
}

if (Num > 0)
{
	//get the service information
	for (i = 0; i < Num; i ++)
	{
		InformStruct[i] = DAB_GetInfo(i );
	}
	//display the service label to navigation
	Display();
}
else
{
	return;
}

index = 2;
DAB_SelService( index );

delay(5);
……
index = 3;
DAB_SelService( index );
……
Sleep();
DAB_CloseDevice();
```