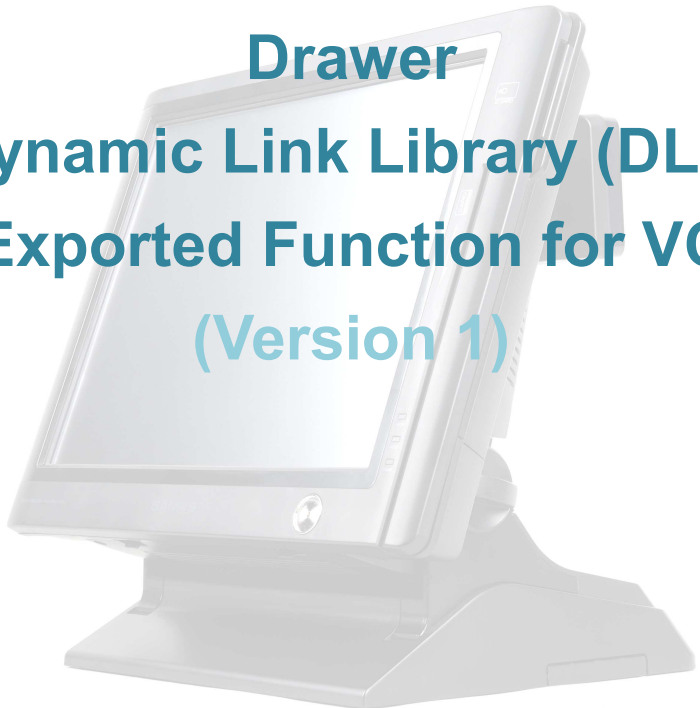


SPT-3000 Series



Drawer Dynamic Link Library (DLL) Exported Function for VC (Version 1)

www.sam4s.com

Revision A (July 02, 2009)

Contents

DRAWER CONTROL DLL (VC_DRAWERDLL.DLL)	3
EXPORTED FUNCTIONS OF DLL	4
DRAWERINIT	4
SETVOLTAGE	5
SETDRAWER	6
DRAWEROPEN	7
READSTART	8



Drawer control DLL (vc_DrawerDll.dll)

You can control SPT-3000's Drawer module from this file. You can load or de-load dll library dynamically, and refer the way how to as below.

//how to load dll

```
hDll = LoadLibrary("vc_DrawerDll.dll");
```

//how to release dll

```
FreeLibrary(hDll);
```



Exported functions of DLL

vc_DrawerDll.dll has several interfaces. ALL OF THEM CAN BE CALLED IN THE CONDITION OF DLL LOADED.

DrawerInit

bool DrawerInit();

Return Value

TRUE

When a drawer device is founded in system

FALSE

When a drawer device is not founded in system

Parameters

none

Remarks

Notice that a drawer device is founded or not in system.

Example (Pseudo codes)

```
typedef int (*DrawerInitFunc());
```

```
DrawerInitFunc lpDrawerInitFunc;
```

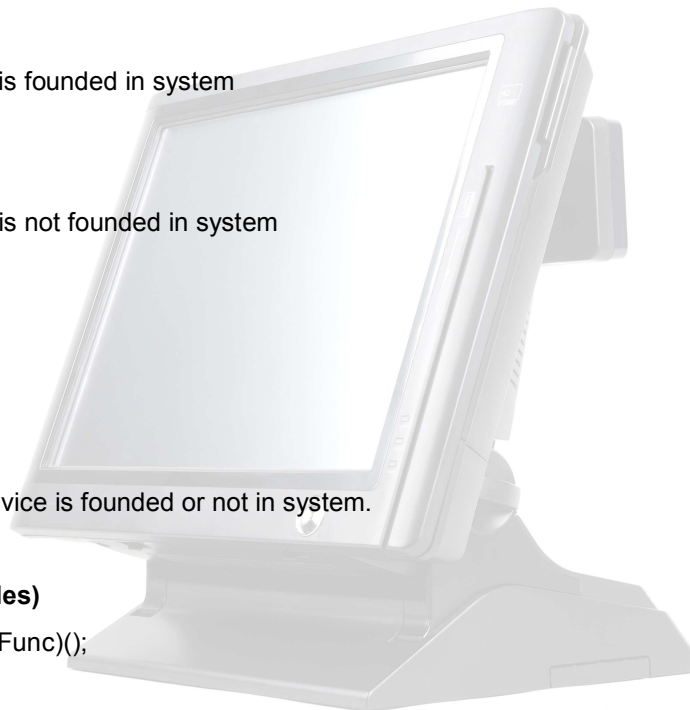
```
lpDrawerInitFunc = (DrawerInitFunc)GetProcAddress(hDll,"DrawerInit");
```

```
if (lpDrawerInitFunc() == FALSE)
```

```
{
```

```
    AfxMessageBox("Drawer device is not founded.");
```

```
}
```



SetVoltage

```
void SetVoltage(int m_Voltage);
```

Return Value

none

Parameters

m_Voltage

This means a drawer's voltage setting value. A number '12' means '12 Voltage', and a number '24' means '24 Voltage'.

Remarks

Set a drawer's voltage value.

Example (Pseudo codes)

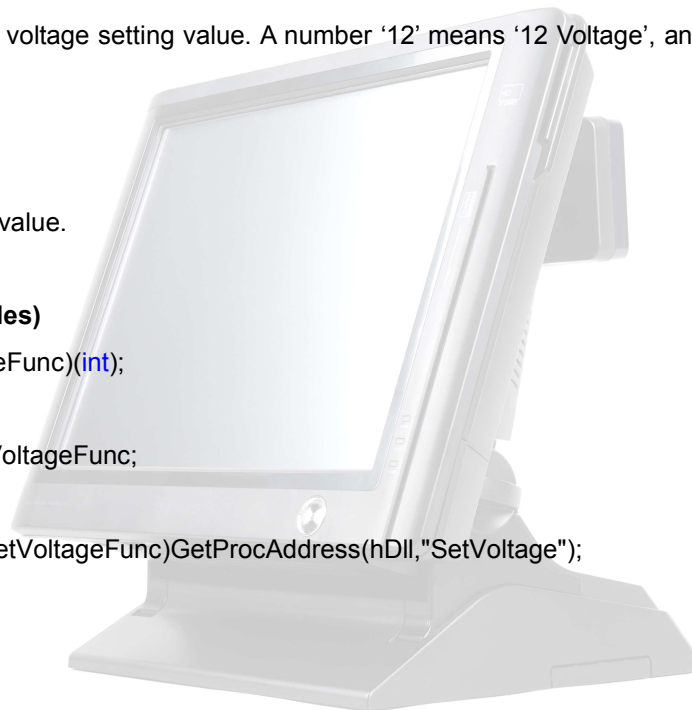
```
typedef int (*SetVoltageFunc)(int);
```

```
SetVoltageFunc lpSetVoltageFunc;
```

```
lpSetVoltageFunc = (SetVoltageFunc)GetProcAddress(hDll,"SetVoltage");
```

```
lpSetVoltageFunc(12);
```

```
GetDlgItem(IDC_STAT_SVOL)->SetWindowText("12 V");
```



SetDrawer

void SetDrawer(**int** m_DrawerNo)

Return Value

none

Parameters

m_DrawerNo

This means a drawer's number. A number '1' means 'Drawer 1', and a number '2' means 'Drawer 2'.

Remarks

Set a drawer's number.

Example (Pseudo codes)

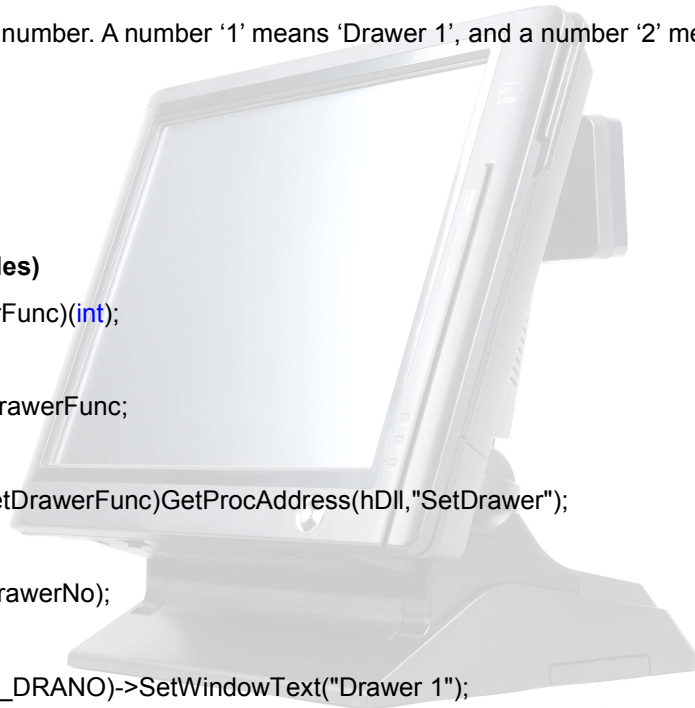
```
typedef int (*SetDrawerFunc)(int);
```

```
SetDrawerFunc lpSetDrawerFunc;
```

```
lpSetDrawerFunc = (SetDrawerFunc)GetProcAddress(hDll,"SetDrawer");
```

```
lpSetDrawerFunc(m_DrawerNo);
```

```
GetDlgItem(IDC_STAT_DRANO)->SetWindowText("Drawer 1");
```



DrawerOpen

void DrawerOpen(**int** m_DrawerNo)

Return Value

none

Parameters

m_DrawerNo

This means a drawer's number. A number '1' means 'Drawer 1', and a number '2' means 'Drawer 2'.

Remarks

Open a drawer determined by 'm_DrawerNo' value.

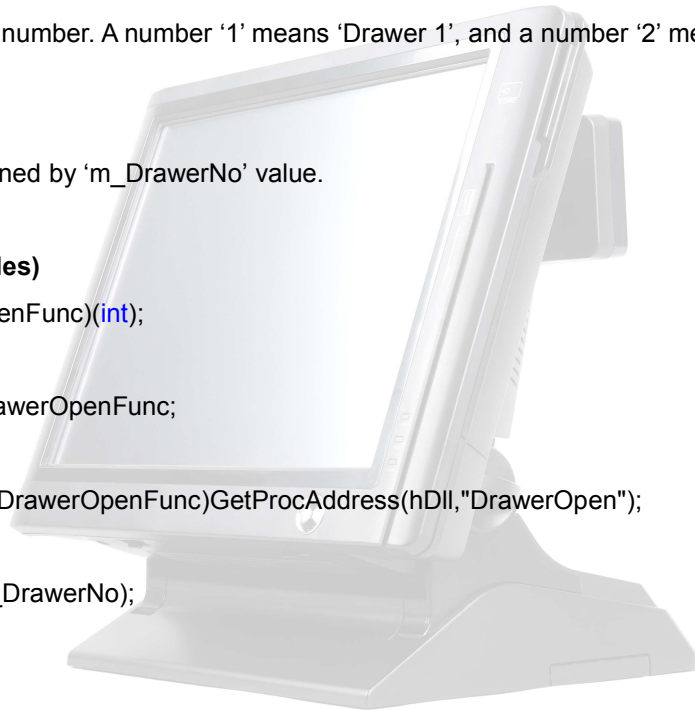
Example (Pseudo codes)

```
typedef int (*DrawerOpenFunc)(int);
```

```
DrawerOpenFunc lpDrawerOpenFunc;
```

```
lpDrawerOpenFunc = (DrawerOpenFunc)GetProcAddress(hDll,"DrawerOpen");
```

```
lpDrawerOpenFunc(m_DrawerNo);
```



ReadStart

Int ReadStart()

Return Value

1

OPENED-Ready : Drawer is opened.

2

CLOSED-Charging: Charging during a drawer is closed. Charging time usually takes about 4.5 sec.

3

OPENED- Charging: Charging during a drawer is opened. Charging time usually takes about 4.5 sec.

4

CLOSED- Ready: Drawer is closed.

Parameters

none

Remarks

Read a current drawer status.

Example (Pseudo codes)

```
typedef int (*ReadStartFunc());
```

```
ReadStartFunc lpReadStartFunc;
```

```
lpReadStartFunc = (ReadStartFunc)GetProcAddress(hDll,"ReadStart");
```

```
int result = lpReadStartFunc();
```

```
if (result == 1)
```

```
{
```

```
    GetDlgItem(IDC_STAT_DRASTAT)->SetWindowText("OPENED-Ready");
```

```
}
```

```
else if (result == 2)
```



```
{  
    GetDlgItem(IDC_STAT_DRASTAT)->SetWindowText("CLOSED-Charging");  
}  
else if (result == 3)  
{  
    GetDlgItem(IDC_STAT_DRASTAT)->SetWindowText("OPENED-Charging");  
}  
else if (result == 4)  
{  
    GetDlgItem(IDC_STAT_DRASTAT)->SetWindowText("CLOSED-Ready");  
}
```

