

Software Development Kit for Sensor.dll

Version 2.3



**PROFESSIONAL MAKER of
SBC/CompactPCI Products**

IBASE Technology Inc.

Email: sales@ibase.com.tw

<http://www.ibase.com.tw>

Tel: 886-2-26557588 Fax: 886-2-26557158

**NOTICE: THIS SOFTWARE IS THE PROPERTY OF IBASE TECHNOLOGY INC.
IT IS CONSIDERED A TRADE SECRET AND IS NOT TO BE DIVULGED
OR USED BY PARTIES WHO HAVE NOT RECEIVED WRITTEN
AUTHORIZATION.**

**ALL INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER
EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
PURPOSE.**

1. Sensor chip support list

VIA VT82C686A, VIA VT82C686B, Winbond 83781D, Winbond 83782D, Winbond 83783S,
Winbond 83627HF, Winbond 83627THF , Winbond 83697HF and ITE8712.

2. Driver Installation

2.1 For Windows 98 and Windows Me

(1) Copy "IBEM.VXD" file to <%WINDIR%>\SYSTEM\

2.2 For Windows NT 4.0, Windows 2000 and Windows XP

(1) Copy "IBEM.SYS" file to <%WINDIR%>\SYSTEM32\DRIVERS\

(2) Execute the RegEdit. The following parameters must be written to your registry.

HKLM,"System\CurrentControlSet\Services\IBEM ","ErrorControl",%REG_DWORD%,0x00000001

HKLM,"System\CurrentControlSet\Services\IBEM ","Type",%REG_DWORD%,0x00000001

HKLM,"System\CurrentControlSet\Services\IBEM ","Start",%REG_DWORD%,0x00000000

HKLM,"System\CurrentControlSet\Services\IBEM "," DisplayName ",%REG_SZ%," IBEM"

Or import "IBEM_NT.REG" from RegEdit

3. Export Function on Sensor.dll and notices

```
extern "C" __declspec(dllexport) __stdcall int DLLInitial(void);
```

```
extern "C" __declspec(dllexport) __stdcall int DLLRelease(void);
```

```
extern "C" __declspec(dllexport) __stdcall int Vcore(void);
```

```
extern "C" __declspec(dllexport) __stdcall int Vio(void);
```

```
extern "C" __declspec(dllexport) __stdcall int P5V(void);
```

```
extern "C"__declspec(dllexport) __stdcall int P12V(void);
extern "C"__declspec(dllexport) __stdcall int N12V(void);
extern "C"__declspec(dllexport) __stdcall int N5V(void);
extern "C"__declspec(dllexport) __stdcall int Fan1(void);
extern "C"__declspec(dllexport) __stdcall int Fan2(void);
extern "C"__declspec(dllexport) __stdcall int SysTemp(void);
extern "C"__declspec(dllexport) __stdcall int CpuTemp(void);
```

Some of the sensor chips may not support all of those functions. The unsupported functions may return undefined value. For example, VT82C686A / VT82C686B cannot detect –12V and –5V, the routine “N12V” and “N5V” do not function for this sensor chip.

This version of sensor.dll allow multi-instance at the same time, but the programmer should avoid this.

```
extern "C"__declspec(dllexport) __stdcall int DLLInitial(void);
```

Input : none

Output : return the status of driver, sensor and error code.

Bit0..Bit7 : sensor chip. For detail information, please refer to the “SENSOR.H” file and demo program.

Bit8..Bit15 : error code. For detail information, please refer to the “SENSOR.H” file and demo program.

This function should be invoked before you try to access the sensor chip, or the sensor function return value could be undefined.

```
extern "C"__declspec(dllexport) __stdcall int DLLRelease(void);
```

Input : none

Output : always return 1.

This function should be invoked before the program closed and it will release the device drive and memory for sensor.dll. If the program closed without call this routine may cause memory leak or the operating system un-stabile.

```
extern "C"__declspec(dllexport) __stdcall int Vcore(void);
```

Input : none

Output : Vcore (mV)

```
extern "C"__declspec(dllexport) __stdcall int Vio(void);
```

Input : none

Output : Vio (mV)

```
extern "C"__declspec(dllexport) __stdcall int P5V(void);
```

Input : none
Output : +5V (mV)

```
extern "C" __declspec(dllexport) __stdcall int P12V(void);
```

Input : none
Output : +12V (mV)

```
extern "C" __declspec(dllexport) __stdcall int N12V(void);
```

Input : none
Output : -12V (mV)

```
extern "C" __declspec(dllexport) __stdcall int N5V(void);
```

Input : none
Output : -5V (mV)

```
extern "C" __declspec(dllexport) __stdcall int Fan1(void);
```

Input : none
Output : Fan1 RPM (rpm)

```
extern "C" __declspec(dllexport) __stdcall int Fan2(void);
```

Input : none
Output : Fan2 RPM (rpm)

```
extern "C" __declspec(dllexport) __stdcall int SysTemp(void);
```

Input : none
Output : System temperature (°C)

```
extern "C" __declspec(dllexport) __stdcall int CpuTemp(void);
```

Input : none
Output : CPU temperature (°C)