TOSHIBA

TOSHIBA Windows POS Terminal

ST-A10/ST-A20/ST-B10/ST-B20/ST-C10/ ST-M30

Sensor Library External Interface Specification

First Edition : January 9, 2009
Second Edition : October 5, 2009
Third Edition : March 31, 2011
Fourth Edition : October 21, 2011
Fifth Edition : December 12, 2011

TOSHIBA TEC CORPORATION

Revision Record

No. EAA-02760

Sensor Library External Interface Specification

Rev. No.	Date	Pages	Description
---	Jan. 9, 2009	---	Original issue
1	Oct. 5,2009	1	Changes were made to support POSReady 2009
2	Mar. 31,2011	1	Changes were made to support Windows 7 Add the model: ST-A20, ST-B10, ST-C10
3	Oct 21, 2011	3,4,19,20,25	Add battery temperature for ST-C10 support
4	Dec 12, 2011	1-5, 21-28	Add support for ST-M30

Table of Contents

1. General 1
1.1 Introduction 1
1.2 Scope 1
1.3 Structure 2
1.4 Features 3
1.4.1 Types and Characteristics of Fan 3
1.4.2 Sensor Messages 4
2. Functions 5
2.1 Function List 5
2.2 Details of Functions 5
2.2.1 OpenSensor 6
2.2.2 CloseSensor 7
2.2.3 ReadLTemp 8
2.2.4 ReadRTemp 9
2.2.5 ReadFanRota 10
2.2.6 ReadSensorVol15 11
2.2.7 ReadSensorVol25 12
2.2.8 ReadSensorVol50 13
2.2.9 ReadSensorVol120 14
2.2.10 ReadSensorVolCPUCore 15
2.2.11 ReadSensorAll 16
2.2.12 SensorWatchStart 18
2.2.13 GetMachineType 20
2.2.14 ReadBatteryTemp 21
2.2.15 ReadPowerTemp 22
2.2.16 ReadSensorVol33 23
2.2.17 ReadSensorVolCmosBat 24
2.2.18 ReadSensorVolDIMM 25
2.2.19 ReadSensorAllEx 26
3. Remarks 28
3.1 Registry 28
3.2 Usage Example 28
3.3 Log 30
3.4 Method to Confirm Driver Installation 36

1. General

1.1 Introduction

This document describes the external interface specifications of the library which reads status (fan rotational speed, voltage, temperature) of the sensor for the POS terminal.

The following header files provide the definitions required to access the library.
SENSOR.H (Should be included.)
StrSensor.h (Included in SENSOR.H and should not be included by a user.)

Windows, WindowsXP, Windwos2000, WindowsVista, WEPOS, POSReady 2009, POSReady 7 and Windows 7 are registered trademarks of Microsoft Corporation.

Celeron and Pentium are registered trademarks of Intel Corporation.

1.2 Scope

The sensor library supports performance on the following hardware and operating systems.

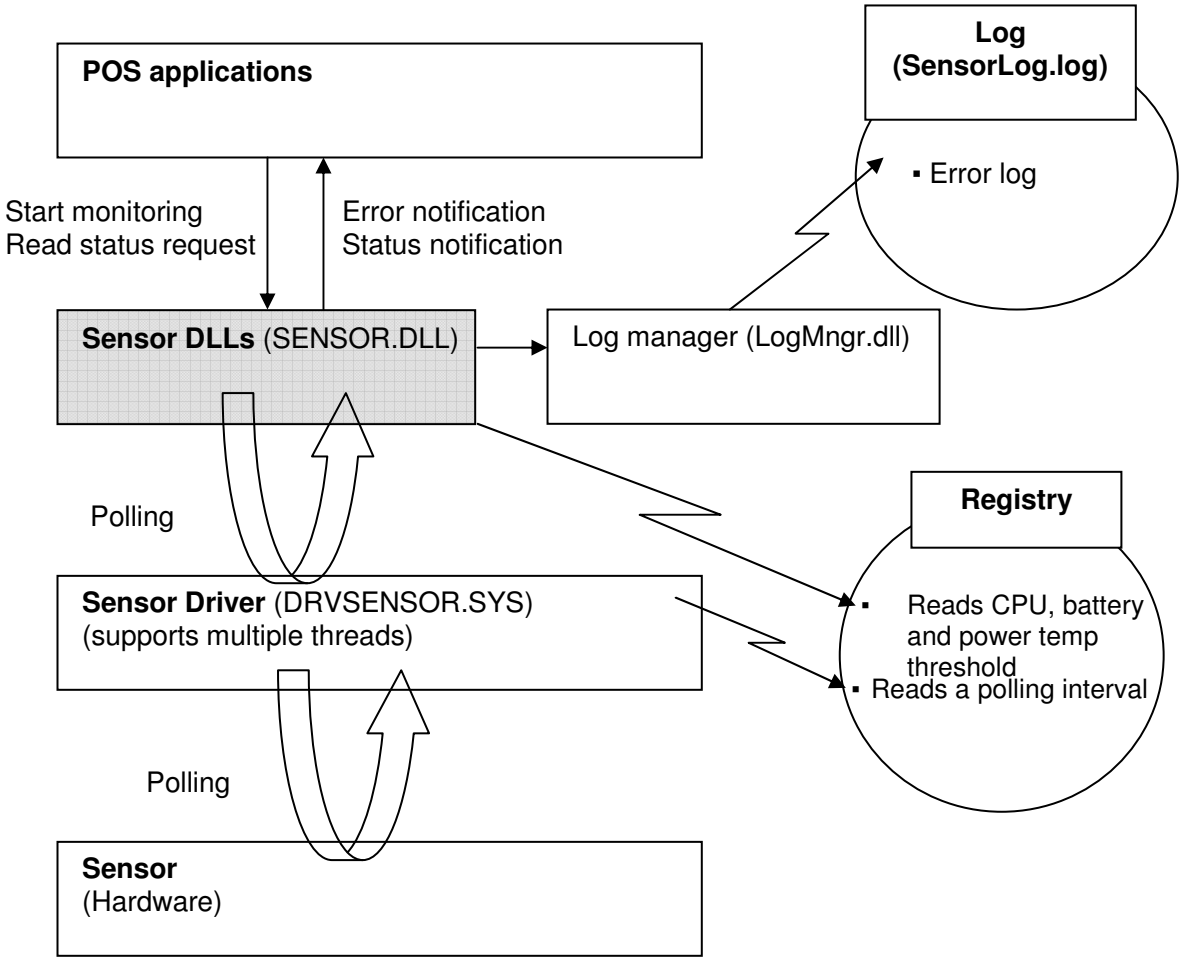
OS / Hardware	Windows 2000 Professional SP4	Windows XP Professional SP3	WEPOS1.1	Windows Embedded POSReady 2009	Windows Vista Business SP1	Windows 7 Professional SP1 (32bits)
ST-A10	✓	✓	✓	✓		
ST-A20	✓	✓	✓	✓	✓	✓
ST-B10		✓	✓	✓		
ST-B20	✓	✓	✓	✓	✓	✓
ST-C10		✓	✓	✓		✓
ST-M30		✓	✓	✓		✓

OS / Hardware	Windows 7 Professional SP1 (64bits)	POSReady 7 (32bits)	POSReady 7 (64bits)
ST-A10			
ST-A20		✓	
ST-B10			
ST-B20		✓	
ST-C10		✓	
ST-M30	✓	✓	✓

✓ - means the OS is supported on the POS models platform

Successful installation of the TEC sensor driver (DRVSENSOR.SYS) is a prerequisite.

1.3 Structure



In order for the POS applications to access the HW sensor, they can call the sensor driver via the sensor DLLs, and then the sensor driver accesses the hardware sensor to perform reading.

In addition, in order to notify if an error is occurring, the sensor DLL will poll at an interval which can be configured in the registry.

DLL will post windows message when sensor (hardware) abnormal status detect:

- ✓ CPU temperature increase above the threshold or decrease below the threshold
- ✓ Battery temperature increase above the threshold or decrease below the threshold
- ✓ Power temperature increase above the threshold or decrease below the threshold
- ✓ The fan start or stop rotating

Please note that DLL will not post any message if CPU temperature increases or decrease below the threshold.

1.4 Features

1.4.1 Types and Characteristics of Fan

Currently, fan rotation monitoring is only supports on ST-B20 and ST-M30.

The ST-B20 only supports the CPU fan reading only.
The ST-M30 only supports the CPU and case (system) fan reading only.

Position of Fan	Attachment		Read of Rotational Speed	Name Used by Sensor Driver
	ST-B20	ST-M30		
CPU	Yes	Yes	Yes	CPU_FAN_R
HDD	No	No	Yes	HDD_FAN_R
Case	No	Yes	Yes	CASE_FAN_R
Power supply	No	No	No	Not used

1.4.2 Sensor Messages

The sensor library monitors CPU temperature, battery temperature, power temperature, fan rotational speed and issues a user-defined Windows message if an error occurs.

If a user wishes to receive these sensor messages, the user must register SENSOR_MESSAGE to Windows using the RegisterWindowMessage function. Then user can receive a sensor message using a message identifier returned by this function. The SENSOR_MESSAGE string is defined by SENSOR.H.

If fan rotational speed, battery/power/CPU temperature reaches the threshold, a sensor message is issued and the detail of the sensor message is set to lParam. The wParam and lParam values and meaning are as given in the table below.

wParam	LParam Value	Meaning
CPU_TEMP_MESSAGE (0x0000)	TEMP_HIGH (0x0001)	The CPU temperature has exceeded the threshold (is above the threshold).
	TEMP_LOW (0x0000)	The CPU temperature has gone down below the threshold.
CPU_FAN_MESSAGE (0x0001)	FAN_STOP (0x0001)	The CPU fan stops rotating.
	FAN_START (0x0000)	The CPU fan starts rotating.
HDD_FAN_MESSAGE (0x0002)	FAN_STOP (0x0001)	The HDD fan stops rotating.
	FAN_START (0x0000)	The HDD fan starts rotating.
CASE_FAN_MESSAGE (0x0003)	FAN_STOP (0x0001)	The case fan stops rotating.
	FAN_START (0x0000)	The case fan starts rotating.
BATT_TEMP_MESSAGEE (0x0004)	TEMP_ABN (0x0001)	The battery temperature has exceeded or below the threshold.
	TEMP_NOR (0x0000)	The battery temperature has return to the normal threshold.
PWR_TEMP_MESSAGEE (0x0005)	TEMP_ABN (0x0001)	The power temperature has exceeded or below the threshold.
	TEMP_NOR (0x0000)	The power temperature has return to the normal threshold.

The Windows message and the RegisterWindowMessage functions are standard Windows functions.
For details, refer to the help file for the development environment and/or the MSDN Library.

2. Functions

2.1 Function List

Function Name	Description of Function
OpenSensor	Opens the sensor driver.
CloseSensor	Closes the sensor driver.
ReadLTemp	Reads a local temperature (also know as system temperature for example LPC47M287 chip temperature).
ReadRTemp	Reads a remote temperature (CPU temperature).
ReadFanRota	Reads fan rotation speed for different fan type.
ReadSensorVol15	Reads +1.5V voltage.
ReadSensorVol25	Reads +2.5V voltage.
ReadSensorVol50	Reads +5V voltage.
ReadSensorVol120	Reads +12V voltage.
ReadSensorVolCPUCore	Reads CPU core voltage.
ReadSensorAll	Reads all parameters (using old STR_SENSOR structure).
SensorWatchStart	Starts the SENSOR monitoring.
GetMachineType	Gets POS model information.
ReadBatteryTemp	Reads a battery temperature
ReadPowerTemp	Reads a power temperature
ReadSensorVol33	Reads a +3.3V voltage.
ReadSensorVolCmosBat	Reads on-board CMOS battery voltage.
ReadSensorVolDIMM	Reads DIMM voltage.
ReadSensorAllEx	Reads all parameter by passing in structure size(output buffer size).

2.2 Details of Functions

Refer to the subsequent pages.

2.2.1 OpenSensor

Function

Opens a specified sensor device.

Format

HANDLE hObject = OpenSensor (**LPCTSTR** lpName);

Parameters

lpName Specifies a pointer to the name “SENSOR” string of the sensor to be opened.

Return value

If the function succeeds, an object handle to the sensor object is returned.
If the function fails, INVALID_HANDLE_VALUE (-1) is returned.
In case of an error, refer to an extended error code. To get extended error information, the GetLastError function is used. For how to use the GetLastError function, refer to the help file for the development environment and/or the MSDN Library.

Remarks

To close the sensor object handle, the CloseSensor function is used.

Notes

Only the thread process which called the OpenSensor function can access each function.

Extended error codes and handling

SENSOR_ERROR_DUPLICATION_OPEN (0x00000001)
The specified sensor device is already open. The same process cannot open the device which is already open.

SENSOR _ERROR_OPEN_REG (0x00000002)
The name to open the sensor device, “SENSOR” is wrong. Check the name to open the device.

SENSOR_INSIDE_ERROR (0x00000004)
An internal error occurred.

SENSOR_PARAM_ERROR (0x00000005)
The lpName value is NULL.

SENSOR_CREATETHREAD_ERROR (0x00000007)
An initiation of the thread, which polls a status, failed.

SENSOR_ERROR_REG (0x00000009)
Reading of the registry failed. Using the Device Manager, make sure that the sensor driver is successfully installed. If this error occurs again even when the sensor driver is successfully installed, there is a possibility that something is wrong with the software.

SENSOR_NOT_READY (0x0000000a)
The sensor is not initialized. Try again after a short while.

2.2.2 CloseSensor

Function

Closes the OpenSensor object handle and notifies the driver that an access to the sensor is no longer necessary. When the SensorWatchStart function is used, status monitoring is stopped.

Format

DWORD = CloseSensor (**HANDLE** hObject);

Parameters

hObject Specifies the sensor object handle returned by OpenSensor function.

Return value

If the function succeeds, zero is returned.
If the function fails, an error code is returned.

Remarks

When the SensorWatchStart function is used, status monitoring is stopped. However, if other thread has started to monitor the SensorWatchStart function, the monitoring thread continues.

Error codes and handling

SENSOR_ERROR_NOT_OPEN (0x00000003)
The sensor driver is not open. It does not have to be closed.

SENSOR_INSIDE_ERROR (0x00000004)
An internal error occurred.

SENSOR_ERROR_INVALID_HANDLE (0x00000008)
The specified object handle is wrong.

2.2.3 ReadLTemp

Function

Reads a local temperature (also know as system temperature for example LPC47M287 chip temperature).

Format

DWORD = ReadLTemp (**HANDLE** hObject, **LPLONG** lpLTemp);

Parameters

hObject	Specifies a sensor object handle returned by the OpenSensor function.
LpLTemp	A pointer to a buffer which stores a local temperature (e.g. LPC47M287 chip temperature) The buffer size is 4 byte.

Return value

If the function succeeds, zero is returned.
If the function fails, an error code is returned.

Remarks

The lpLTemp max and min range is from –127 ℃ to 127 ℃.

Error codes and handling

SENSOR_ERROR_NOT_OPEN (0x00000003)
The sensor driver is not open. Only call this function after the OpenSensor function succeeds.

SENSOR_INSIDE_ERROR (0x00000004)
An internal error occurred.

SENSOR_PARAM_ERROR (0x00000005)
There is an error with the specified parameter. Check the parameter.

SENSOR_ERROR_INVALID_HANDLE (0x00000008)
The specified object handle is wrong.

SENSOR_POLLING_ERROR (0x0000000c)
A reading operation failed at the previous polling due to a hardware error.

2.2.4 ReadRTemp

Function

Reads a remote temperature (CPU temperature).

Format

DWORD = ReadRTemp (**HANDLE** hObject, **LPLONG** lpRTemp);

Parameters

hObject	Specifies a sensor object handle returned by the OpenSensor function.
LpRTemp	A pointer to a buffer which stores a remote temperature (CPU temperature) The buffer size is 4 byte.

Return value

If the function succeeds, zero is returned.
If the function fails, an error code is returned.

Remarks

The lpRTemp max and min range is from –127 ℃ to 127 ℃. The threshold value for lpRTemp is vary depend on the POS model.

Error codes and handling

SENSOR_ERROR_NOT_OPEN (0x00000003)
The sensor driver is not open. Call this function after the OpenSensor function succeeds.

SENSOR_INSIDE_ERROR (0x00000004)
An internal error occurred.

SENSOR_PARAM_ERROR (0x00000005)
There is an error with the specified parameter. Check the parameter.

SENSOR_ERROR_INVALID_HANDLE (0x00000008)
The specified object handle is wrong.

SENSOR_POLLING_ERROR (0x0000000c)
A reading operation failed at the previous polling due to a hardware error.

2.2.5 ReadFanRota

Function

Reads a rotational speed of a specified fan.

Format

DWORD = ReadFanRota (**HANDLE** hObject, **DWORD** dwFan, **LPDWORD** lpFRota);

Parameters

hObject	Specifies a sensor object handle returned by the OpenSensor function.
dwFan	A buffer which stores rotation speed of the specify fan type. Three types of fan is supported by specify the following parameter. For details of each parameter, refer to Section 1.4.1. CPU_FAN, HDD_FAN, CASE_FAN
lpFRota	A pointer to a buffer which stores a rotational speed of the specified fan The buffer size is 4 byte.

Return value

If the function succeeds, zero is returned.
If the function fails, an error code is returned.

Notes

1. The unit of rotational speeds is rpm. If a fan is not working or connected, zero is stored in lpFRota.
2. The rotational speed of the not attached fan or the fan's registry is configuring to 0, the value will always return 0xFFFFFFFFh.

Error codes and handling

SENSOR_ERROR_NOT_OPEN (0x00000003)
The sensor driver is not open. Call this function after the OpenSensor function succeeds.

SENSOR_INSIDE_ERROR (0x00000004)
An internal error occurred.

SENSOR_PARAM_ERROR (0x00000005)
There is an error with the specified parameter. Check the parameter.

SENSOR_ERROR_INVALID_HANDLE (0x00000008)
The specified object handle is wrong.

SENSOR_POLLING_ERROR (0x0000000c)
A reading operation failed at the previous polling due to a hardware error.

2.2.6 ReadSensorVol15

Function

Reads a +1.5V voltage of the voltage monitoring sensor.
If the sensor operates on the ST-A10, ST-A20, ST-B10 or ST-B20, the VCC voltage detects by the voltage monitoring sensor will be returned.
If the sensor operates on the ST-C10, a DCIN voltage is read by the voltage monitoring sensor.
This voltage information is not supported on ST-M30.

Format

DWORD = ReadSensorVol15 (**HANDLE** hObject, **LPDWORD** lpVolt)

Parameters

hObject	Specifies a sensor object handle returned by the OpenSensor function.
lpVolt	A pointer to a buffer which stores a voltage value The unit is mV and the buffer size is 4 byte.

Return value

If the function succeeds, zero is returned.
If the function fails, an error code is returned.

Error codes and handling

SENSOR_ERROR_NOT_OPEN (0x00000003)
The sensor driver is not open. Call this function after the OpenSensor function succeeds.

SENSOR_INSIDE_ERROR (0x00000004)
An internal error occurred.

SENSOR_PARAM_ERROR (0x00000005)
There is an error with the specified parameter. Check the parameter.

SENSOR_ERROR_INVALID_HANDLE (0x00000008)
The specified object handle is wrong.

SENSOR_POLLING_ERROR (0x0000000c)
A reading operation failed at the previous polling due to a hardware error.

2.2.7 ReadSensorVol25

Function

Reads a +2.5V voltage of the voltage monitoring sensor.
If the sensor operates on the ST-B20, the VCC voltage will be read by the voltage monitoring sensor.
This voltage information is not supported on ST-C10 and ST-M30.

Format

DWORD = ReadSensorVol25 (**HANDLE** hObject, **LPDWORD** lpVolt);

Parameters

hObject	Specifies a sensor object handle returned by the OpenSensor function.
lpVolt	A pointer to a buffer which stores a voltage value The unit is mV and the buffer size is 4 byte.

Return value

If the function succeeds, zero is returned.
If the function fails, an error code is returned.

Error codes and handling

SENSOR_ERROR_NOT_OPEN (0x00000003)
The sensor driver is not open. Call this function after the OpenSensor function succeeds.

SENSOR_INSIDE_ERROR (0x00000004)
An internal error occurred.

SENSOR_PARAM_ERROR (0x00000005)
There is an error with the specified parameter. Check the parameter.

SENSOR_ERROR_INVALID_HANDLE (0x00000008)
The specified object handle is wrong.

SENSOR_POLLING_ERROR (0x0000000c)
A reading operation failed at the previous polling due to a hardware error.

2.2.8 ReadSensorVol50

Function

Reads a +5.0V voltage of the voltage monitoring sensor.
If the sensor operates on the ST-B20, the VCC voltage will be read by the voltage monitoring sensor.
This voltage information is not supported on ST-M30.

Format

DWORD = ReadSensorVol50 (**HANDLE** hObject, **LPDWORD** lpVolt);

Parameters

hObject	Specifies a sensor object handle returned by the OpenSensor function.
lpVolt	A pointer to a buffer which stores a voltage value The unit is mV and the buffer size is 4 byte.

Return value

If the function succeeds, zero is returned.
If the function fails, an error code is returned.

Error codes and handling

SENSOR_ERROR_NOT_OPEN (0x00000003)
The sensor driver is not open. Call this function after the OpenSensor function succeeds.

SENSOR_INSIDE_ERROR (0x00000004)
An internal error occurred.

SENSOR_PARAM_ERROR (0x00000005)
There is an error with the specified parameter. Check the parameter.

SENSOR_ERROR_INVALID_HANDLE (0x00000008)
The specified object handle is wrong.

SENSOR_POLLING_ERROR (0x0000000c)
A reading operation failed at the previous polling due to a hardware error.

2.2.9 ReadSensorVol120

Function

Reads a +12.0V voltage of the voltage monitoring sensor.
If the sensor operates on the ST-B20, the VCC voltage will be read by the voltage monitoring sensor.

Format

DWORD = ReadSensorVol120 (**HANDLE** hObject, **LPDWORD** lpVolt);

Parameters

hObject	Specifies a sensor object handle returned by the OpenSensor function.
lpVolt	A pointer to a buffer which stores a voltage value The unit is mV and the buffer size is 4 byte.

Return value

If the function succeeds, zero is returned.
If the function fails, an error code is returned.

Error codes and handling

SENSOR_ERROR_NOT_OPEN (0x00000003)
The sensor driver is not open. Call this function after the OpenSensor function succeeds.

SENSOR_INSIDE_ERROR (0x00000004)
An internal error occurred.

SENSOR_PARAM_ERROR (0x00000005)
There is an error with the specified parameter. Check the parameter.

SENSOR_ERROR_INVALID_HANDLE (0x00000008)
The specified object handle is wrong.

SENSOR_POLLING_ERROR (0x0000000c)
A reading operation failed at the previous polling due to a hardware error.

2.2.10 ReadSensorVolCPUCore

Function

Reads a CPU core voltage of the voltage monitoring sensor.
This voltage information is not supported on ST-C10.

Format

DWORD = ReadSensorVolCPUCore (**HANDLE** hObject, **LPDWORD** lpVolt);

Parameters

hObject	Specifies a sensor object handle returned by the OpenSensor function.
lpVolt	A pointer to a buffer which stores a voltage value The unit is mV and the buffer size is 4 byte.

Return value

If the function succeeds, zero is returned.
If the function fails, an error code is returned.

Remarks

A CPU core voltage differs depending on a type of and load on the CPU.
Reference values: Celeron 2.0 GHz…1.525V - 1.370V, Pentium4 2.8 GHz…1.525V - 1.340V
Celeron 1.2 GHz…1.585V - 1.390V (Typ. 1.5V)

Error codes and handling

SENSOR_ERROR_NOT_OPEN (0x00000003)
The sensor driver is not open. Call this function after the OpenSensor function succeeds.

SENSOR_INSIDE_ERROR (0x00000004)
An internal error occurred.

SENSOR_PARAM_ERROR (0x00000005)
There is an error with the specified parameter. Check the parameter.

SENSOR_ERROR_INVALID_HANDLE (0x00000008)
The specified object handle is wrong.

SENSOR_POLLING_ERROR (0x0000000c)
A reading operation failed at the previous polling due to a hardware error.

2.2.11 ReadSensorAll

Function

Reads the entire parameters for example fan rotational speed, voltage and temperature information which are supported by the sensor library.
This function will still using the old strucutre for STR_SENSOR. Please refer for ReadSensorAllEx for more details.

Format

DWORD = ReadSensorAll (**HANDLE** hObject, **PSTR_SENSOR** lpStrsensor);

Parameters

hObject Specifies a sensor object handle returned by the OpenSensor function.
lpStrsensor A pointer to a buffer which stores all readable parameters and STR_SENSOR structure. The units used are ℃ for temperature, rpm for rotational speed, and mV for voltage. The STR_SENSOR structure is defined as follows. (Defined by Str_Sensor.h)

```
typedef struct STR_SENSOR {
    LONG        Local_Temp;          // Lcal temperature (on chip)
    LONG        Remote_Temp;         // Remote(CPU) temperature
    DWORD       CPU_Fan_Read_Rota;   // CPU fan rotational speed
    DWORD       HDD_Fan_Read_Rota;   // HDD fan rotational speed
    DWORD       Case_Fan_Read_Rota;  // Case fan rotational speed
    DWORD       Voltage_15;          // +1.5V voltage *[1]
    DWORD       Voltage_25;          // +2.5V voltage *
    DWORD       Voltage_50;          // +5V voltage
    DWORD       Voltage_120;         // +12V voltage
    DWORD       Voltage_Core;        // CPUcore voltage
} STR_SENSOR, *PSTR_SENSOR
```

The rotational speed of the fan, for which zero (not attached) is set in the registry, is always FFFFFFFFh.
If voltage or temperature value always return 0, this means that item might no supported on the POS model. Please refer to each specify reading function for more information.

Return value

If the function succeeds, zero is returned.
If the function fails, an error code is returned.

Error codes and handling

SENSOR_ERROR_NOT_OPEN (0x00000003)
The sensor driver is not open. Call this function after the OpenSensor function succeeds.

SENSOR_INSIDE_ERROR (0x00000004)
An internal error occurred.

SENSOR_PARAM_ERROR (0x00000005)
There is an error with the specified parameter. Check the parameter.

SENSOR_ERROR_INVALID_HANDLE (0x00000008)

[1] If the sensor operates on the ST-B20, a VCC voltage of the voltage monitoring sensor is stored in VOLTAGE_15. If the sensor operates on the ST-C10, a DCIN voltage of the voltage monitoring sensor is stored in VOLTAGE_15

The specified object handle is wrong.

SENSOR_POLLING_ERROR (0x0000000c)
A reading operation failed at the previous polling due to a hardware error.

2.2.12 SensorWatchStart

Function

Starts monitoring the sensor and fan rotational speed sensor.

Format

DWORD = SensorWatchStart (**HANDLE** hObject, **HANDLE** hWnd);

Parameters

hObject	Specifies a sensor object handle returned by the OpenSensor function.
hWnd	An application window handle which receives a sensor monitoring message

Return value

If the function succeeds, zero is returned.
If the function fails, an error code is returned.

Remarks

1. When this function is called during a normal operation, zero, which indicates a successful completion, is immediately returned. If an event causing a message to be issued occurs, the sensor library sends a sensor message to an application specified by hWnd.
2. The sensor monitors an event described in Section 1.4.2 and if such an event occurs, the sensor notifies an application specified by hWnd of the event.
3. Once the monitoring starts, a sensor message is issued whenever an event causing the message to be issued occurs.
4. The monitoring continues until the CloseSensor function is performed.
5. When several processes are calling the SensorWatchStart function and one of the processes closes by CloseSensor, the monitoring continues for other processes because the monitoring is performed for each process.
6. The CPU, battery and power temperature threshold to be notified to an application is stored in the registry described in Chapter 3. A monitoring polling interval from the sensor library to the sensor driver is set by the registry.
7. If the CPU temperature is already above the threshold, battery or power temperature is abnormal or if any of the fans is not working (that is, it is in an error state) when this function is called, a sensor message is immediately issued.

Notes

Event monitoring does not start until this function is called.
When this function is called from one thread using two or more kinds of application window handle as parameters, a sensor message is sent only to the window handle application which is used when this function is called last. In this case, the messages sent to the applications previously registered are not sent to the applications which are registered later.

Example:

CPU fan stops.
↓
SensorWatchStart(hObject, handle of A); → CPU fan stop message to A
↓
SensorWatchStart(hObject, handle of B); → No message notification
↓
CPU fan starts rotating. → CPU fan rotation start message to B (No message to A)
↓
CPU fan stops. → CPU fan stop message to B

Error codes and handling

SENSOR_ERROR_NOT_OPEN (0x00000003)
The sensor driver is not open. Call this function after the OpenSensor function succeeds.

SENSOR_INSIDE_ERROR (0x00000004)
An internal error occurred.

SENSOR_PARAM_ERROR (0x00000005)
There is an error with the specified parameter. Check the parameter.

SENSOR_ERROR_INVALID_HANDLE (0x00000008)
The specified object handle is wrong.

2.2.13 GetMachineType

Function

Reads a POS model.

Format

DWORD = GetMachineType (**HANDLE** hObject, **LPLONG** lpType);

Parameters

hObject	Specifies a sensor object handle returned by the OpenSensor function.
lpType	A pointer to a buffer which stores a model value The buffer size is 4 byte.

Return value

If the function succeeds, zero is returned.
If the function fails, an error code is returned.

Remarks

If this function succeeds, a value which indicates a model is stored in the buffer specified by lpType.

1: ST-A20/ST-B20
2: ST-A10/ST-B10
3: ST-C10
4: ST-M30
Others: Identification error (A model identification error is logged in the event log.)

Error codes and handling

SENSOR_ERROR_NOT_OPEN (0x00000003)
The sensor driver is not open. Call this function after the OpenSensor function succeeds.

SENSOR_INSIDE_ERROR (0x00000004)
An internal error occurred.

SENSOR_PARAM_ERROR (0x00000005)
There is an error with the specified parameter. Check the parameter.

SENSOR_ERROR_INVALID_HANDLE (0x00000008)
The specified object handle is wrong.

2.2.14 ReadBatteryTemp

Function

Reads a battery temperature.
Currently this information only supports on ST-C10.

Format

DWORD = ReadBatteryTemp (**HANDLE** hObject, **LPLONG** lpBattTemp);

Parameters

hObject	Specifies a sensor object handle returned by the OpenSensor function.
lpBattTemp	A pointer to a buffer which stores a battery temperature The buffer size is 4 byte.

Return value

If the function succeeds, zero is returned.
If the function fails, an error code is returned.

Remarks

The lpBattTemp max and min range is from -127℃ to 127℃.
The lpBattTemp threshold value on ST-C10 range is from -25 ℃ to 60 ℃.

Error codes and handling

SENSOR_ERROR_NOT_OPEN (0x00000003)
The sensor driver is not open. Call this function after the OpenSensor function succeeds.

SENSOR_INSIDE_ERROR (0x00000004)
An internal error occurred.

SENSOR_PARAM_ERROR (0x00000005)
There is an error with the specified parameter. Check the parameter.

SENSOR_ERROR_INVALID_HANDLE (0x00000008)
The specified object handle is wrong.

SENSOR_POLLING_ERROR (0x0000000c)
A reading operation failed at the previous polling due to a hardware error.

2.2.15 ReadPowerTemp

Function

Reads a power temperature.
Currently this information only supports on ST-M30.

Format

DWORD = ReadpowerTemp (**HANDLE** hObject, **LPLONG** lpPwrTemp);

Parameters

hObject	Specifies a sensor object handle returned by the OpenSensor function.
lpPwrTemp	A pointer to a buffer which stores a power temperature The buffer size is 4 byte.

Return value

If the function succeeds, zero is returned.
If the function fails, an error code is returned.

Remarks

The lpPwrTemp max and min range is -127 ℃ to 127 ℃.
The lpPwrTemp threshold value on ST-M30 range is from 0℃ to 65 ℃.

Error codes and handling

SENSOR_ERROR_NOT_OPEN (0x00000003)
The sensor driver is not open. Call this function after the OpenSensor function succeeds.

SENSOR_INSIDE_ERROR (0x00000004)
An internal error occurred.

SENSOR_PARAM_ERROR (0x00000005)
There is an error with the specified parameter. Check the parameter.

SENSOR_ERROR_INVALID_HANDLE (0x00000008)
The specified object handle is wrong.

SENSOR_POLLING_ERROR (0x0000000c)
A reading operation failed at the previous polling due to a hardware error.

2.2.16 ReadSensorVol33

Function

Reads a +3.3V voltage of the voltage monitoring sensor.
Currently this information only supports on ST-M30.

Format

DWORD = ReadSensorVol33 (**HANDLE** hObject, **LPDWORD** lpVolt);

Parameters

hObject	Specifies a sensor object handle returned by the OpenSensor function.
lpVolt	A pointer to a buffer which stores a voltage value The unit is mV and the buffer size is 4 byte.

Return value

If the function succeeds, zero is returned.
If the function fails, an error code is returned.

Error codes and handling

SENSOR_ERROR_NOT_OPEN (0x00000003)
The sensor driver is not open. Call this function after the OpenSensor function succeeds.

SENSOR_INSIDE_ERROR (0x00000004)
An internal error occurred.

SENSOR_PARAM_ERROR (0x00000005)
There is an error with the specified parameter. Check the parameter.

SENSOR_ERROR_INVALID_HANDLE (0x00000008)
The specified object handle is wrong.

SENSOR_POLLING_ERROR (0x0000000c)
A reading operation failed at the previous polling due to a hardware error.

2.2.17 ReadSensorVolCmosBat

Function

Reads on-board CMOS battery voltage of the voltage monitoring sensor.
Currently this information only supports on ST-M30.

Format

DWORD = ReadSensorVolCmosBat (**HANDLE** hObject, **LPDWORD** lpVolt);

Parameters

hObject	Specifies a sensor object handle returned by the OpenSensor function.
lpVolt	A pointer to a buffer which stores a voltage value The unit is mV and the buffer size is 4 byte.

Return value

If the function succeeds, zero is returned.
If the function fails, an error code is returned.

Error codes and handling

SENSOR_ERROR_NOT_OPEN (0x00000003)
The sensor driver is not open. Call this function after the OpenSensor function succeeds.

SENSOR_INSIDE_ERROR (0x00000004)
An internal error occurred.

SENSOR_PARAM_ERROR (0x00000005)
There is an error with the specified parameter. Check the parameter.

SENSOR_ERROR_INVALID_HANDLE (0x00000008)
The specified object handle is wrong.

SENSOR_POLLING_ERROR (0x0000000c)
A reading operation failed at the previous polling due to a hardware error.

2.2.18 ReadSensorVolDIMM

Function

Reads DIMM voltage of the voltage monitoring sensor.
Currently this information only supports on ST-M30.

Format

DWORD = ReadSensorVolDIMM (**HANDLE** hObject, **LPDWORD** lpVolt);

Parameters

hObject	Specifies a sensor object handle returned by the OpenSensor function.
lpVolt	A pointer to a buffer which stores a voltage value The unit is mV and the buffer size is 4 byte.

Return value

If the function succeeds, zero is returned.
If the function fails, an error code is returned.

Error codes and handling

SENSOR_ERROR_NOT_OPEN (0x00000003)
The sensor driver is not open. Call this function after the OpenSensor function succeeds.

SENSOR_INSIDE_ERROR (0x00000004)
An internal error occurred.

SENSOR_PARAM_ERROR (0x00000005)
There is an error with the specified parameter. Check the parameter.

SENSOR_ERROR_INVALID_HANDLE (0x00000008)
The specified object handle is wrong.

SENSOR_POLLING_ERROR (0x0000000c)
A reading operation failed at the previous polling due to a hardware error.

2.2.19 ReadSensorAllEx

Function

Reads all parameter base on the buffer size.

Format

DWORD = ReadSensorAllEx (
HANDLE hObject,
PSTR_SENSOR_EX lpStrsensor,
DWORD sizeofStrSensor);

Parameters

hObject	Specifies a sensor object handle returned by the OpenSensor function.
lpStrsensor	A pointer to a STR_SENSOR_EX structure buffer which stores all readable parameters. The units used are ℃ for temperature, rpm for rotational speed, and mV for voltage. The STR_SENSOR_EX structure is defined as follows. (Defined by Str_Sensor.h)
sizeofStrSensor	size of the STR_SENSOR_EX structure (lpStrsensor buffer size)

Starting forward v1.002, a new defines for the STR_SENSOR_EX will be used:

```
typedef struct STR_SENSOR_EX {
    LONG  Local_Temp;
    LONG  Remote_Temp;
    DWORD CPU_Fan_Read_Rota;
    DWORD HDD_Fan_Read_Rota;
    DWORD Case_Fan_Read_Rota;
    DWORD Voltage_15;
    DWORD Voltage_25;
    DWORD Voltage_50;
    DWORD Voltage_120;
    DWORD Voltage_Core;
    LONG  Batt_Temp;
    DWORD Voltage_Bat;
    DWORD Voltage_33;
    DWORD Voltage_DIMM;
    LONG  Power_Temp;

    DWORD Reserved16;       // Reserved item for future new hw reading
    DWORD Reserved17;
    DWORD Reserved18;
    DWORD Reserved19;
    DWORD Reserved20;
    DWORD Reserved21;
    DWORD Reserved22;
    DWORD Reserved23;
    DWORD Reserved24;
    DWORD Reserved25;
    DWORD Reserved26;
    DWORD Reserved27;
    DWORD Reserved28;
    DWORD Reserved29;
    DWORD Reserved30;
    DWORD Reserved31;
    DWORD Reserved32;
} STR_SENSOR_EX, *PSTR_SENSOR_EX;
```

In this case, if application calls this API by using old STR_SENSOR structure, as long as the application sends correct value of sizeofStrSensor (for old structure the size is 40bytes), the DLL will

still return all readable parameter for the old STR_SENSOR by only copy 40bytes data from current return data.

Return value

If the function succeeds, zero is returned.
If the function fails, an error code is returned.

Error codes and handling

SENSOR_ERROR_NOT_OPEN (0x00000003)
The sensor driver is not open. Call this function after the OpenSensor function succeeds.

SENSOR_INSIDE_ERROR (0x00000004)
An internal error occurred.

SENSOR_PARAM_ERROR (0x00000005)
There is an error with the specified parameter. Check the parameter.

SENSOR_ERROR_INVALID_HANDLE (0x00000008)
The specified object handle is wrong.

SENSOR_POLLING_ERROR (0x0000000c)
A reading operation failed at the previous polling due to a hardware error.

3. Remarks

3.1 Registry

A list of registries used by the sensor library and a default value will be used when the registry cannot be read for some reasons define as follows:

[HKEY_LOCAL_MACHINE\SYSTEM\TEC\POS DRIVERS\DRVSENSOR\SENSOR]

Key	Format	Data	Default (Same even in case of no registry)
CPUNoticeTemp	REG_DWORD	CPU temperature notification threshold for P4 model (unit: °C)	00000064h (100°C)
LibInterval	REG_DWORD	Library polling interval (unit: sec.)	00000001h (1 sec.)
Devicename	REG_SZ	Sensor device name	DRVSENSOR (Mandatory)
DebugLogLevel	REG_SZ	Log level	1
DebugLogFile	REG_SZ	Path to log file	C:\OPOS\TEC\LOG\SENSOR.LOG
DLLPath	REG_SZ	Path to DLL	C:\OPOS\TEC\DLL\SENSOR.DLL
CPUNoticeTemp3	REG_DWORD	CPU temperature notification threshold for P3 model (unit: °C)	00000064h (100°C)
BattNoticeHighTemp	REG_DWORD	Battery temperature high notification threshold (unit: °C)	0000003Ch (60°C)
BattNoticeLowTemp	REG_DWORD	Battery temperature low notification threshold (unit: °C)	FFFFFFE7h (-25°C)
PWRNoticeHighTemp	REG_DWORD	Power temperature high notification threshold (unit: °C)	00000041h (65°C)
PWRNoticeLowTemp	REG_DWORD	Power temperature low notification threshold (unit: °C)	00000000h (0°C)

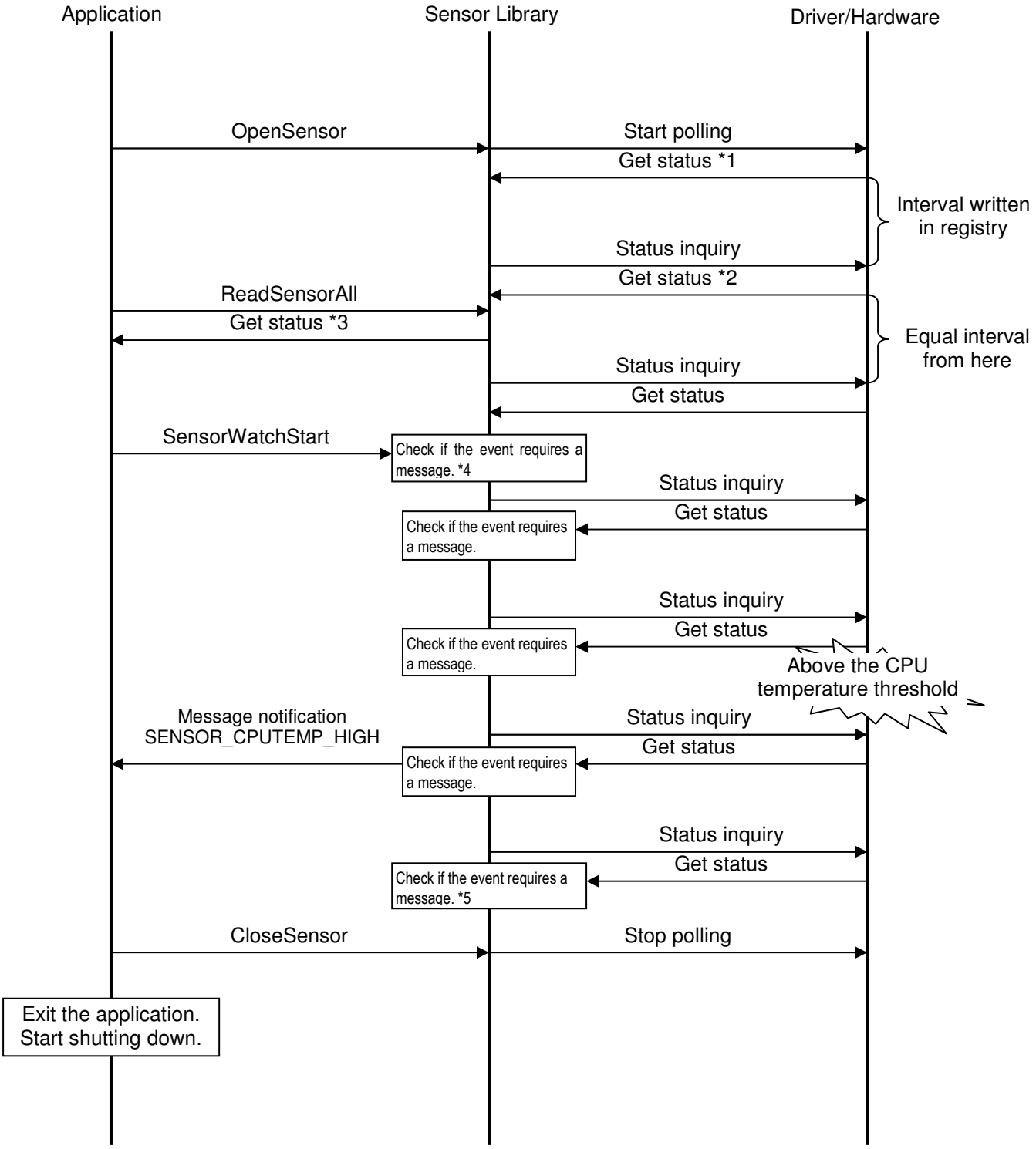
Note: Set CPUNoticeTemp and CPUNoticeTemp3 to 0 to 127℃. Other values are ignored. The CPU temperature threshold differs depending on the hardware models. For this reason, CPUNoticeTemp and CPUNoticeTemp3 must be set for each model. Set a value with reference to the Appendix to the Windows CPU Sensor Library Specification.

3.2 Usage Example

The next page shows a usage example of the sensor library using the sequence diagram

<Procedure>
1. Open the sensor. (OpenSensor)
2. Read all statuses. (ReadSensorAll)
3. Start monitoring (SensorWatchStart)
   [An event causing a message to be issued occurs (Here, the CPU temperature exceeded the threshold.)]
4. Issue the message. (SENSOR_CPUTEMP_HIGH)
5. Close the sensor (CloseSensor).
6. Exit the application and shut down.

<Usage Example Sequence Diagram>



Notes:

*1. The library holds the obtained status.
*2. The library updates a status which it holds.
*3. The library sends a message about the data it holds.
*4. Referring to the data, the library immediately sends a message if the CPU temperature has exceeded the threshold or the fan is not working.
*5. Even after sending a message, the library sends another message if an event causing the message to be issued occurs.

3.3 Log

When the sensor library in operation detects an error, it logs the error in a log file. Nothing is logged during normal operations.
In addition, if an error causing a message to be issued occurs, the error is logged in an event log.

File name

SENSOR.log
A path to a log file is written in the registry (Refer to Section 3.1.) The default is C:\OPOS\TEC\LOG.

Size

Max. 4 Mbyte. Data is overwritten from the oldest.

Format

<date of error occurrence>,<time of error occurrence>,[function name]<error code *1>,<details of error>
For an OpenSensor error code *1, a user can get a value using GetLastError.

Log level

Log level is Level 0 to Level 2 and written in the registry. (Refer to Section 3.1.) An operation at each log level is as follows:

0: Log is not written.
1: Error logs and message notifications are written in the log. (Default)
2: In addition to Level 1, a trace log of the functions is written.
3 or higher: Same operation as Level 2.

Details of errors

The subsequent pages show the error codes returned by the sensor library and details of the corresponding log.

(1) Log of unrecoverable errors

Error Code or GetLastError Return Value	Detailed Error Message	Descriptions and Error Handling
SENSOR_CREATETHREAD_ERROR (0x00000007)	Initiation of a thread, which performs polling, failed.	If any of these error occurs, there is a possibility that something is wrong with the software.
SENSOR_INSIDE_ERROR (0x00000004)	A TIMER_START error (polling start error) occurred in the driver.	
SENSOR_INSIDE_ERROR (0x00000004)	A RegisterWindowMessage function error occurred. Error Code is ****.	
SENSOR_INSIDE_ERROR (0x00000004)	An INVALID_PARAMETER error (parameter error) occurred in the driver.	
SENSOR_INSIDE_ERROR (0x00000004)	WaitForSingleObject failed.	
SENSOR_INSIDE_ERROR (0x00000004)	CloseHandle failed.	
SENSOR_INSIDE_ERROR (0x00000004)	An INSUFFICIENT_BUFFER error (too small parameter or NULL) occurred in the driver.	
SENSOR_INSIDE_ERROR (0x00000004)	A FILE_NOT_FOUND error (The driver is not open.) occurred in the driver.	
SENSOR_INSIDE_ERROR (0x00000004)	An ACCESS_DENIED error (already open) occurred in the driver.	
SENSOR_INSIDE_ERROR (0x00000004)	Other internal error occurred in the driver. Error Code = **	

(2) Log of recoverable errors by retry

Error Code or GetLastError Return Value	Detailed Error Message	Descriptions and Error Handling
SENSOR_NOT_READY (0x0000000a)	A NOT_READY error (The driver is not initialized.) occurred in the driver.	The sensor device is not initialized. Try again after a short while.

Error Code or GetLastError Return Value	Detailed Error Message	Descriptions and Error Handling
SENSOR_FILE_NOT_FOUND (0x0000000b)	A FILE_NOT_FOUND error (The driver is not found.) occurred in the driver.	Using the Device Manager, make sure that the sensor driver is successfully installed. If this error occurs again even when the sensor driver is successfully installed, there is a possibility that something is wrong with the software.
SENSOR_POLLING_ERROR (0x0000000c)	An error occurred at a previous polling.	Try again after a polling interval time expires. If the same error persists or frequently occurs, there is a possibility that something is wrong with the hardware.

(3) Parameter error log

Error Code or GetLastError Return Value	Detailed Error Message	Descriptions and Error Handling
7SEG_PARAM_ERROR (0x00000005)	The first parameter (handle) is not the same as that obtained by the OpenSensor function.	Check the function parameter setting and retry.
	The second parameter (pointer) is NULL.	
	The second parameter value (DWORD value) is invalid. (Value)	
	The third parameter (pointer) is NULL.	

(4) Registry-related error log

Error Code or GetLastError Return Value	Detailed Error Message	Descriptions and Error Handling
None (Continues with the default.)	A CPU temperature threshold value cannot be obtained from the registry.	Check the driver is successfully installed.
	A battery temperature threshold value cannot be obtained from the registry.	
	A power temperature threshold value cannot be obtained from the registry.	
	A polling interval value cannot be obtained from the registry.	
	DLL path cannot be obtained from the registry.	

SENSOR_ERROR_REG (0x00000009)	A sensor device name cannot be obtained from the registry.	

(5) Message notification log

Error Code or GetLastError Return Value	Remark	Detailed Error Message	Descriptions and Error Handling
None (Not an error)	--	The CPU temperature is ** ℃ and is above the threshold (** ℃). #3	These are logged until the last Close function is performed after the SensorOpen function is called, evern when the SensorWatchStart function is not called and no message notifications are sent. The threshold is as written in the registry. These logs are written in the application log of the event log. (Refer to the below.)
		The CPU temperature is ** ℃ and is below the threshold (** ℃).	
	Only for fan attach machine.	The CPU fan stopped (is not working).	
		The CPU fan started rotating.	
		The HDD fan stopped (is not working).	
		The HDDfan started rotating.	
		The case fan stopped (is not working).	
		The case fan started rotating.	
	Only on ST-C10	Return battery temp value = **。Exceed max value(**℃)。	
		Normal batt temp = **。Within max value(**℃)。	
		Return battery temp value = **。lower than min value(**℃)。	
		Normal batt temp = **。Within min value(**℃)。	
	Only on ST-M30	Return power temp value = **。Exceed max value(**℃)。	
		Normal power temp = **。Within max value(**℃)。	
		Return power temp value = **。lower than min value(**℃)。	
		Normal power temp = **。Within min value(**℃)。	

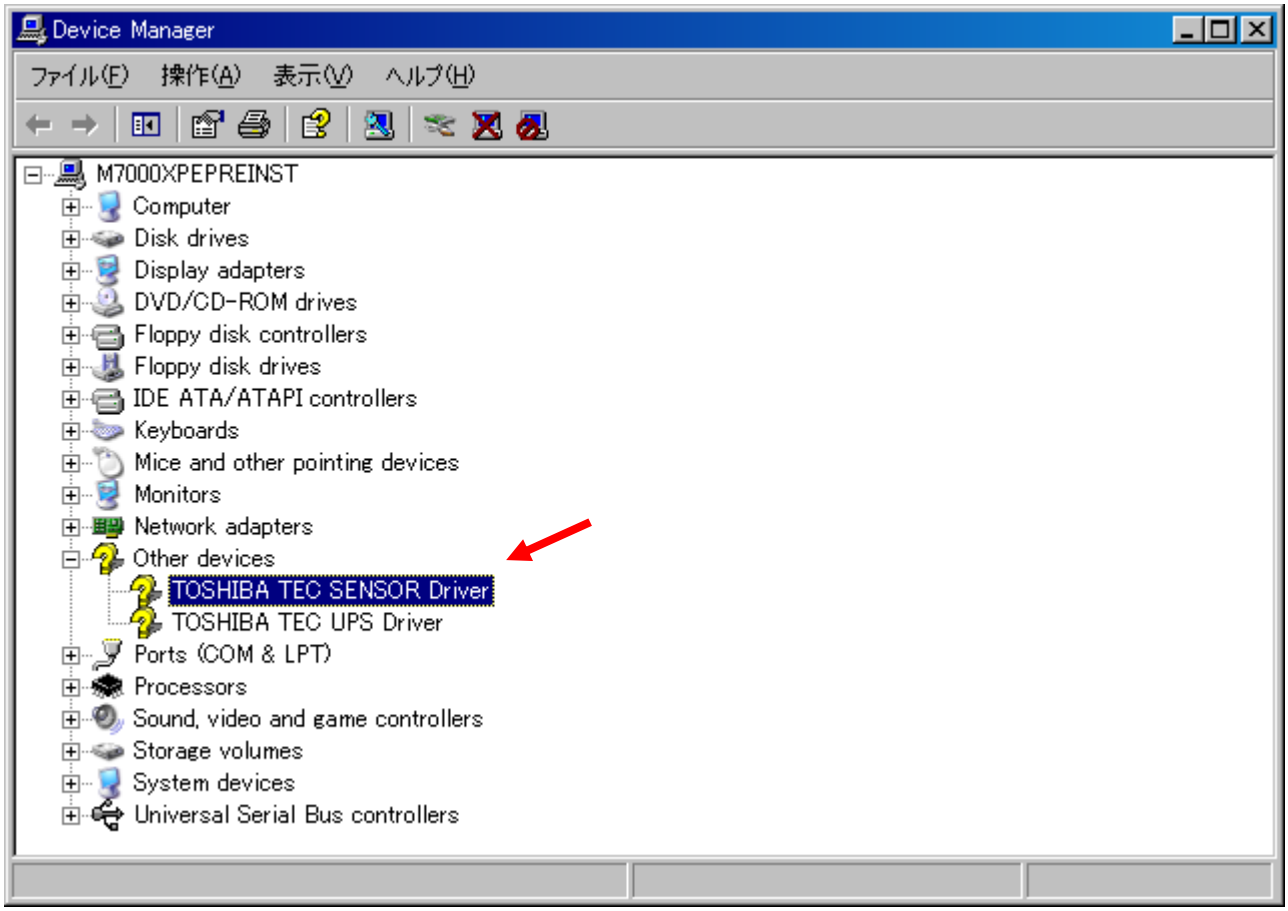
Event log which DLL writes

Event ID	Type	Description	Existence of dump data
256	Error	The CPU temperature is above the threshold.	No
257	Information	The CPU temperature is below the threshold.	No
258	Error	The CPU fan stopped (is not working).	No
259	Information	The CPU fan started rotating.	No
260	Error	The HDD fan stopped (is not working).	No
261	Information	The HDDfan started rotating.	No
262	Error	The case fan stopped (is not working).	No
263	Information	The case fan started rotating.	No
264	Error	The battery temperature is above the threshold	No

265	Error	The battery temperature is below the threshold	No
266	Information	The battery temperature is within the threshod	No
267	Error	The power temperature is above the threshold	No
268	Error	The power temperature is below the threshold	No
269	Information	The power temperature is within the threshod	No

3.4 Method to Confirm Driver Installation

Use the Device Manager to confirm the sensor driver has been successfully installed.
Open the Device Manager and make sure “TOSHIBA TEC SENSOR Driver” is displayed on the screen as shown below.



Note: If a yellow exclamation mark “!” or a red “x” is shown, or “TOSHIBA TEC SENSOR Driver” is not displayed under “Other devices”, the sensor driver has not been loaded successfully.