HAMAMATSU
PHOTON IS OUR BUSINESS

Application Note for S11684 and S11685

Document number : K44-B60015

Dec. 25, 2014

HAMAMATSU PHOTONICS K.K.

Solid State Division The 6th Mfg. The 44 Dept.

Contents

HAMAMATSU
PHOTON IS OUR BUSINESS

How to use S11684/S11685 (1)

The following show how to use S11684 and S11685.

1. Install driver software
When you use the sensor for the first time, install driver software in accordance with the manual stored in the attached CD.

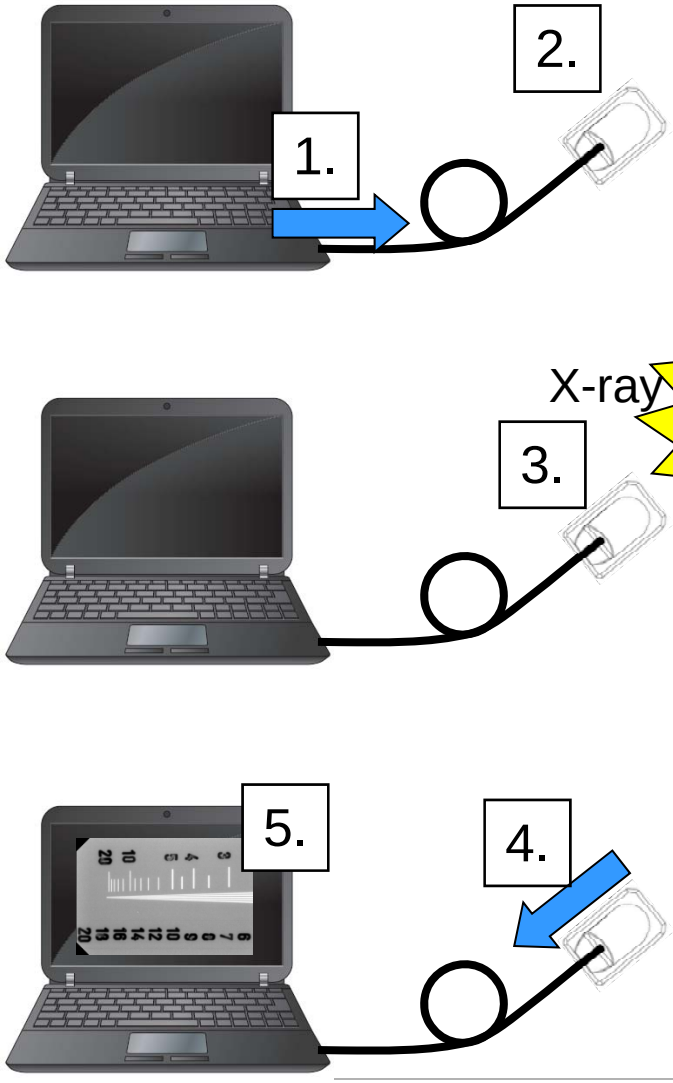
2. Install evaluation software
When you use the sensor for the first time, install evaluation software in accordance with the manual stored in the attached CD.

3. Connect the sensor to a PC
Connect the USB connector to a USB port.

4. Take X-rays
Emit X-rays to the sensor after the evaluation software is ready to acquire an X-ray image.

HAMAMATSU
PHOTON IS OUR BUSINESS

How to use S11684/S11685 (2)



1: A PC gives orders to acquire an image.

2: The sensor is waiting for X-ray irradiation.

3: After detecting X-ray irradiation with the monitoring photodiode of the sensor, the sensor starts image integration.

4: The sensor sends image data.

5: A PC displays an image on an evaluation software.

How to extend the USB cable

It is recommended to use a USB 2.0 repeater cable or a self-powered hub
if it is required to extend the USB cable.

Do not use a bus-powered hub or an extender cable.
If V_{BUS} is lower than 4.75 V, the sensor may not work properly.

For example, SANWA KB-USB-R205 is one of the candidates for a repeater cable.
There is no problem with the performance of an image sensor with the repeater cable.

However, if the USB cable is extended with a repeater or a hub, the amount of electromagnetic radiation might not be compliant with some EMC (electro magnetic compatibility) standards.

Repeater cable
(SANWA KB-USB-R205)

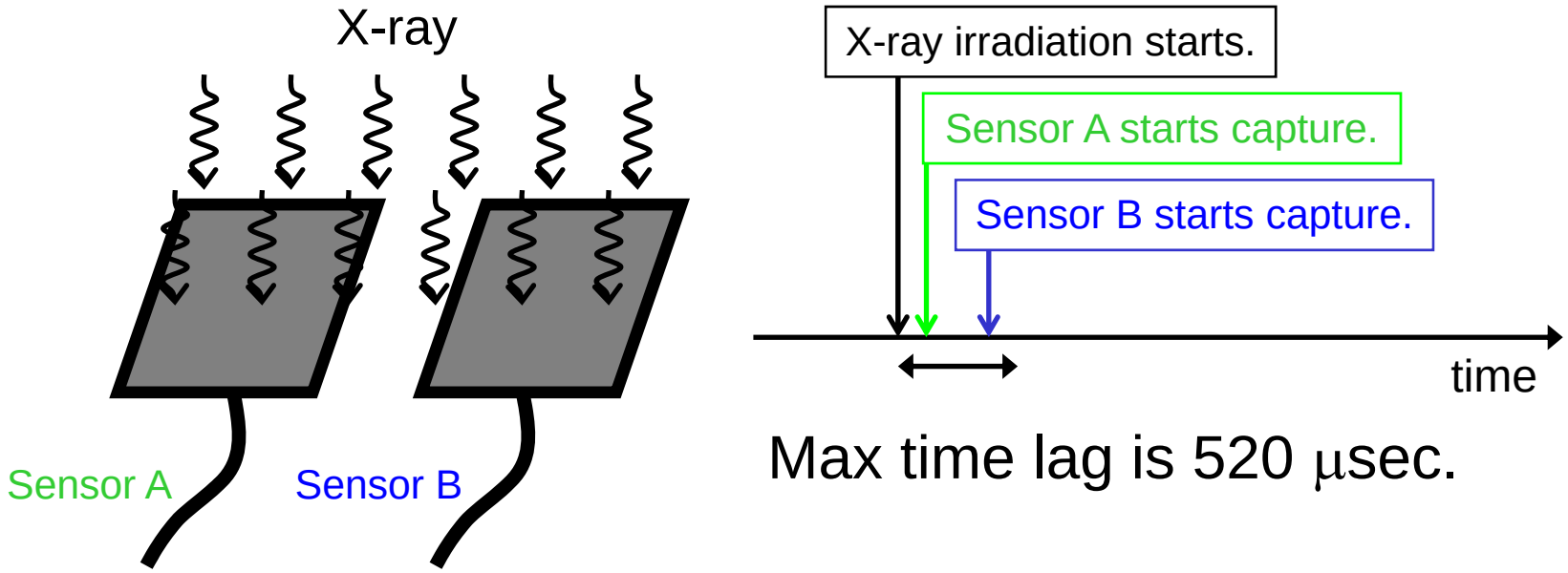
Self-powered hub

Bus-powered hub

Extender cable

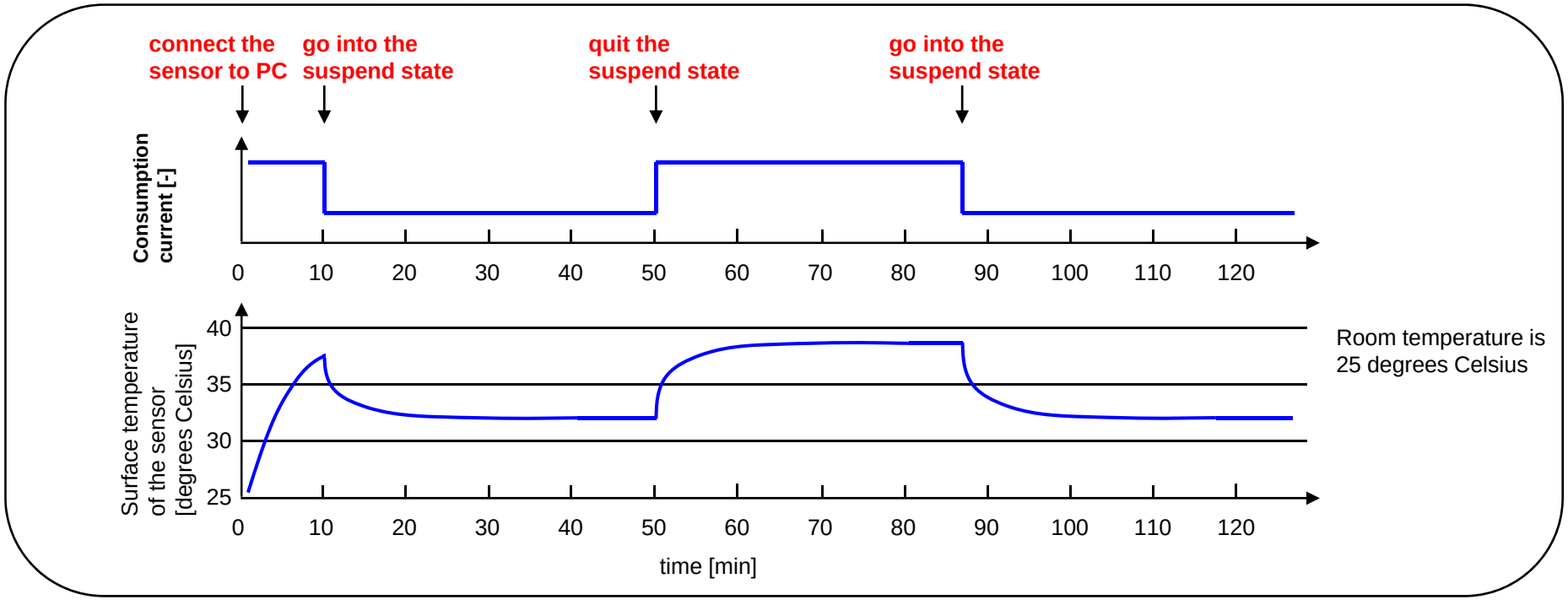
Time lag of acquiring an image

The sensor has a time lag between the start of acquiring an image and the start of irradiation.
The time lag after X-ray irradiation is 520 μsec maximum. Therefore, there is a possibility that **the sensor would not detect the irradiation shorter than 520 μsec**.
When more than one sensor is connected to one PC, each sensor has a time lag.
The sensors will start capture within 520 μsec, but **they do not start capture at the same time**.



How to lower the consumption current of the sensor

You are able to lower the consumption current of the sensor with the **suspend state**.
As a result, the temperature of the sensor becomes lower.
However, **Windows XP does not support the suspend state**.
The consumption current will be much smaller with the suspend state, but it is larger than the current regulated by the USB2.0 specification.



For more information, please refer to the function manual.

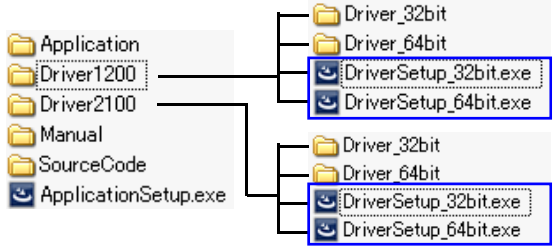
How to enable the suspend state on Windows

You are able to get the sensor in the suspend state with the following two ways.
(A) Calling CMOS_USB.dll's function which get the sensor in the suspend state
(For more information on the functions, please refer to the function manual stored on the attached CD.)
(B) Connecting the sensor and calling no functions for a certain time
However, it depends on INF file whether or not the sensor goes into the suspend state.
The INF file, "IntraOralXrayImageSensor.inf", has information on driver settings including the suspend settings.
There are two versions of the file, 1.2.0.0 and 2.1.0.0 on the attached CD.

Settings	Version 2.1.0.0	Version 1.2.0.0
Is the way (A) enabled?	Yes	No
Is the way (B) enabled?	Yes 10 min after connection	No

Version 1.2.0.0 is provided for Windows XP users because the suspend state does not work properly on Windows XP.

The settings will be applied to your PC by installing driver software with "DriverSetup_32bit.exe" or "DriverSetup_64bit.exe". Installers are stored on the attached CD.



These install driver software in accordance with ver.1.2.0.0 of INF file

These install driver software in accordance with ver.2.1.0.0 of INF file

If our settings of the INF files do not meet your demand, please make your own ".inf" file in accordance with appendix.

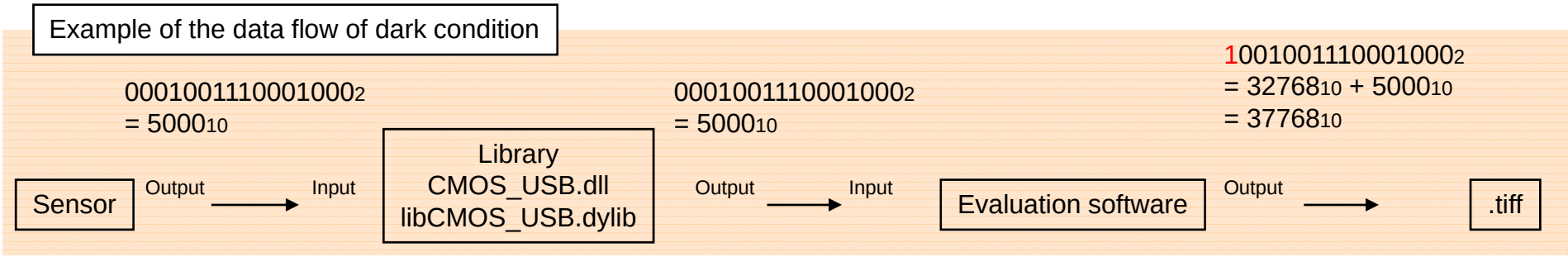
Re: AD output from the sensors

The image data flow is the following. Our evaluation software, S11684S11685.exe, is used to acquire a tiff image.

(1) The output of each pixel is 16 bits data. It is from about 5000 LSB (dark condition) to about 10000 LSB (saturation condition). The sensor outputs the data to CMOS_USB.dll or libCMOS_USB.dylib.

(2) CMOS_USB.dll and libCMOS_USB.dylib output 16 bits image data same as input.

(3) S11684S11685.exe changes the MSB of the data into 1 when creating a tiff image.



HAMAMATSU
PHOTON IS OUR BUSINESS

How to create a digital signature

Digital signatures allow the administrator or end-user who is installing Windows-based software to know whether a legitimate publisher has provided the software package.

For 64-bit versions of Windows Vista and later versions of Windows, driver code signing policy requires that all driver code have a digital signature.

Creating a digital signature requires a code-signing certificate, also referred to as a Software Publisher Certificate (SPC) from a commercial certification authority (CA).

For more details on digital signature, refer to the following website.

http://msdn.microsoft.com/en-us/windows/hardware/gg487328.aspx

HAMAMATSU
PHOTON IS OUR BUSINESS

How to correct images

The procedure for correcting images is shown in our handbook.
The handbook can be downloaded from the following our website.

http://www.hamamatsu.com/resources/pdf/ssd/e05_handbook_image_sensors.pdf
See page 32, 33, “Correction”.

http://www.hamamatsu.com/resources/pdf/ssd/e09_handbook_xray_detectors.pdf
See page 9, “Image correction”.

HAMAMATSU
PHOTON IS OUR BUSINESS

An example of how to make flat field correction

In no event shall we be liable for any trouble and damage arising from the use of flat field correction shown in this page. If you use it, it is necessary to verify your application software.

Flat field correction compensates for slight variations in sensitivity over the effective pixels.
For example, an image corrected by flat field is obtained as the following.

$$I_C(x,y) = I_O(x,y) \times I_{CORRECTION}(x,y)$$

Where,
$I_C(x,y)$: an image corrected by flat field
$I_O(x,y)$: an image being corrected
$I_{CORRECTION}(x,y)$: a correction coefficient of flat field correction

- $I_O(x,y)$ is obtained as the following.
  $I_O(x,y) = I_X(x,y) - I_D(x,y)$
  Where,
  $I_X(x,y)$: an X-ray image with an object
  $I_D(x,y)$: a dark image

- $I_{CORRECTION}(x,y)$ is obtained as the following.
  $I_{CORRECTION}(x,y) = const / (I_F(x,y) - I_D(x,y))$
  Where,
  $I_F(x,y)$: a flat X-ray image
  const : a mean value of pixels of $(I_F(x,y) - I_D(x,y))$

When making the correction, it is useful to note the following points.
- Acquire $(I_F(x,y)-I_D(x,y))$ whose "const" is around 3500 LSB.

- Obtain the averaged image by processing at least 3 images of $(I_F(x,y) - I_D(x,y))$.

- Select pixels from effective pixels except defect lines when calculating "const".

How to make flat field correction with the evaluation software, S11684S11685.exe

In no event shall we be liable for any trouble and damage arising from the use of flat field correction shown in this page. If you use it, it is necessary to verify your application software.

The evaluation software, S11684S11685.exe, attached to CD could be helpful to make flat field correction.
S11684S11685.exe can obtain $I_{CORRECTION\_HPK}(x,y)$ as a TIFF image.
$I_{CORRECTION\_HPK}(x,y)$ is expressed by the following.

$$I_{CORRECTION\_HPK}(x,y) = I_{CORRECTION}(x,y) \times 1000$$

The instruction manual of the software shows how to obtain $I_{CORRECTION\_HPK}(x,y)$.
With $I_{CORRECTION\_HPK}(x,y)$, your application software can make flat field correction by the following.

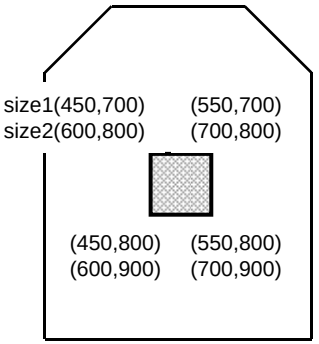
$$I_C(x,y) = I_O(x,y) \times I_{CORRECTION\_HPK}(x,y) / 1000$$

S11684S11685.exe obtains $I_{CORRECTION\_HPK}(x,y)$ with the following procedure.

1.Acquire images of $(I_F(x,y)-I_D(x,y))$
2.Obtain an average image of the images of $(I_F(x,y)-I_D(x,y))$
(defined as $I_{AVERAGE}(x,y)$)
3.Obtain "const" from an average value of 100*100 pixels of $I_{AVERAGE}(x,y)$.
(100*100 area is shown in Fig.1.)
4.Obtain $I_{CORRECTION\_HPK}(x,y)$ from the following

$$I_{CORRECTION\_HPK}(x,y) = const / I_{AVERAGE}(x,y) \times 1000$$

Fig.1;Area of const



HAMAMATSU
PHOTON IS OUR BUSINESS

How to get serial No. programmatically.

We can see a serial No. on the USB A connecter of the sensor.
However, application software does not know the serial No. of the sensor.
A function, HPK_GetSensorInformation, returns the structure containing the serial No.
The structure contains ushort XXXX and uchar YY.
The 6 number, the serial No. of S11684, is YYXXXX.

Before calling “HPK_GetSensorInformation”, application software has to open
a device handle for the connected sensor.
By calling HPK_GetSensorInformation with the device handle,
application software will be able to link a device handle with the Serial No.

Note : Once the sensor is disconnected, the device handle will be invalid.
See “Precautions for a device handle when connecting and disconnecting the sensors”
in order to deal with a device handle properly.

For example.
When 2 sensors are connected, application software need to call the following.

```
USB_OpenDevice(0x4400);  // handle1(e.g.) will be returned
USB_OpenDevice(0x4400);  // handle2(e.g.) will be returned
USB_OpenPipe(handle1);     // pipe handle1(e.g.) will be returned
USB_OpenPipe(handle2);     // pipe handle2(e.g.) will be returned
HPK_GetSensorInformation(handle1, &pIntegParam, &pSensorInfo); // SN will be returned
HPK_GetSensorInformation(handle2, &pIntegParam, &pSensorInfo); // SN will be returned
```

HAMAMATSU
PHOTON IS OUR BUSINESS

How to detect the connection and disconnection of the sensor programmatically

If you use the function and the method shown in the following, it is necessary to verify your application software.

The following function and method could be useful to know whether the sensor is connected or disconnected.
“RegisterDeviceNotification” function (*http://msdn.microsoft.com/en-us/library/aa363431(v=vs.85).aspx*)
“WndProc” method (*http://msdn.microsoft.com/en-us/library/system.windows.forms.form.wndproc.aspx*)

With the events generated by OS, application software can receive information including arrival and removal of USB devices and the device interface GUID. Application software can identify the events of S11684 and S11685 by comparing the GUID with that of the other USB devices. The GUID of S11684 and S11685 is described in IntraOralXrayImageSensor.inf stored in the attached CD .

For Mac OS X, application software can receive the event too. You are able to implement that in accordance with the following sample code.
https://developer.apple.com/library/mac/samplecode/USBPrivateDataSample/Introduction/Intro.html

If you use the functions mentioned above, it is necessary to verify that your application software works properly.

In order to operate the sensors properly after connecting and disconnecting, it is recommended that application software call functions in accordance with “Precautions for a device handle when connecting and disconnecting the sensors.”

HAMAMATSU
PHOTON IS OUR BUSINESS

Precautions for calling functions of CMOS_USB.dll and libCMOS_USB.dylib

1) Application software has to call the function of CMOS_USB.dll one by one for one sensor and must not call the functions at the same time. For example, application software cannot get serial number while the sensor is acquiring an image. However, if more than one sensor is connected, application software can get serial number of a sensor while another sensor is acquiring an image.

2) If CMOS_USB.dll returns an error code, it will be necessary to disconnect and reconnect the sensor in order to operate the sensor again.

3) The sensor must not go into the suspend state while a function is being processed.

The above precautions are important. Therefore, please also refer to the following documents.
S11684S11685_FunctionManual_E_for_windows.pdf
S11684S11685_FunctionManual_E_for_Mac.pdf

Precautions for a device handle when connecting and disconnecting the sensors

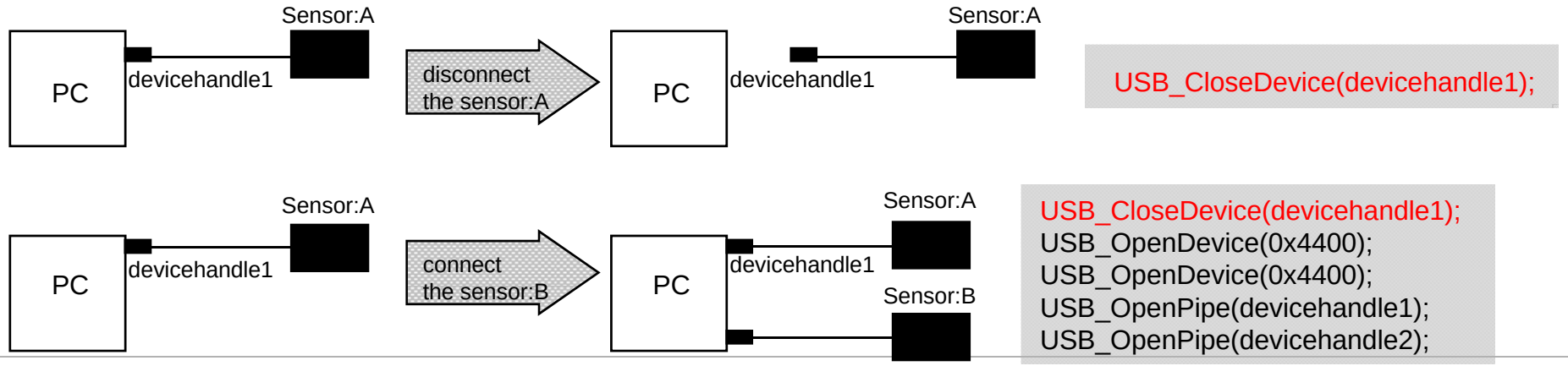
In order to start communication with the sensor, application software has to open the device handle. The device handle is necessary for application software in order to specify the sensor which receives functions.
Application software has to control the device handles in the following situations.

- When you disconnect a sensor

After you disconnect the sensor, your application software has to call “USB_CloseDevice” with the device handle of the disconnected sensor.

- When you connect an additional sensor

If there have been the connected sensors and the device handles have been already opened, it is necessary that application software call “USB_CloseDevice” with all of the opened device handles before you open the device handle of an additional sensor. After calling “USB_CloseDevice”, application software can open all device handles properly. Without “USB_CloseDevice”, there is possibility that application software fails in “USB_OpenDevice.”



HAMAMATSU
PHOTON IS OUR BUSINESS

Precautions for the initialization of the sensor just after being connected

The sensors initialize themselves just after being connected. This initialization takes several seconds (around 1.8 sec for S11684, 2.5 sec for S11685).
However, the sensor does not inform a PC of the end of the initialization. Therefore, we cannot detect the end programmatically.

During the initialization, application software can call the functions of CMOS_USB.dll or libCMOS_USB.dylib.
However, even if application software acquire an image data during the initialization, the data is improper.
Monitor photodiode data acquired during the initialization is also improper.
Therefore, you must not use data acquired during the initialization for diagnosis and so forth.
The following functions work properly during the initialization and returned values are proper.
- “USB_OpenDevice”
- “USB_OpenTargetDevice”
- “USB_OpenPipe”
- “USB_CloseDevice”
- “HPK_GetSensorInformation”

It is recommended that application software have some measures in order not to use the data acquired during the initialization.
For example, **application software should not call any functions for at least 3 seconds after connecting the sensor.**

Appendix

The setting method of suspend state

How to use suspend state

Please refer to the following procedures in order to the use suspend state.

* We explain the procedures with Windows 7 SP1 64bit PC.

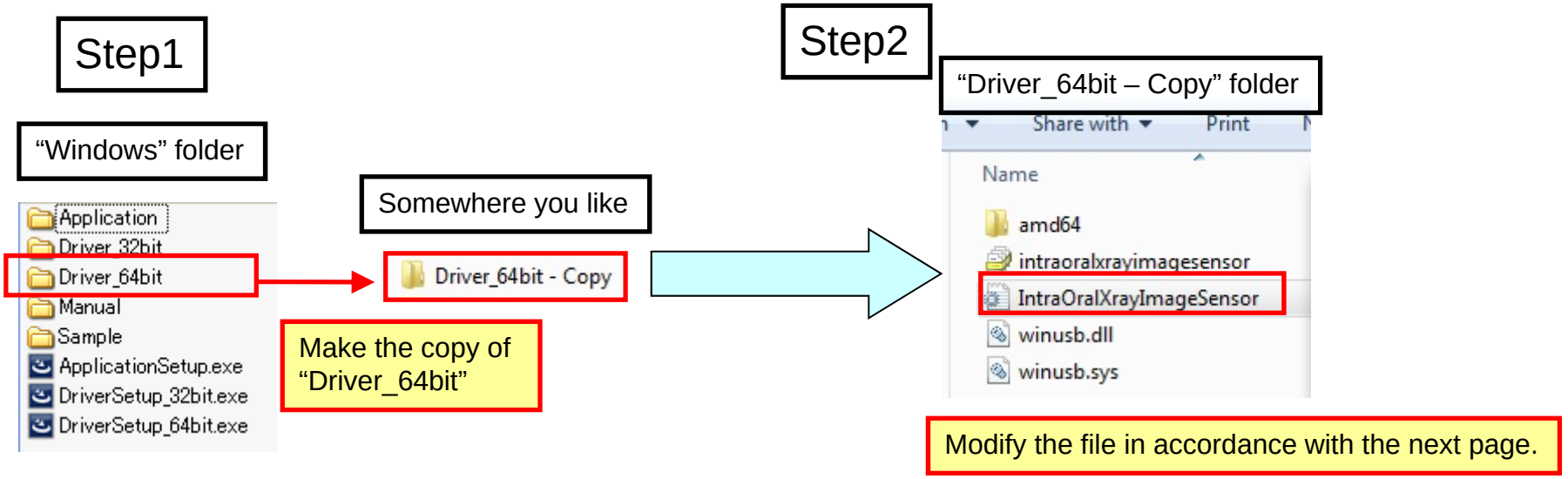
You have to modify the INF file attached to the CD-ROM in order to use the suspend state.
If you follow the procedures, the device becomes the suspend state on a certain time after you do not send any commands.

●Table of contents

Section 1: How to make the INF file that is available for the suspend state

Step1: The following figure shows the contents of the folder named “Windows” attached to CD-ROM.
Copy the “Driver_64bit” folder into somewhere you like if the CPU of your PC is 64-bit.
If the CPU of your PC is 32-bit, copy the “Driver_32bit” folder like the 64-bit CPU.

Step2: Modify IntraOralXrayImageSensor.inf in accordance with the next page.
- If you disable the suspend state, please use 1.2.0.0 of INF.
- If you enable the suspend state, please use 2.1.0.0 of INF.
- If you enable the suspend state with different conditions from 2.1.0.0 of INF, please modify IntraOralXrayImageSensor.inf.



Section 1: How to make the INF file that is available for the suspend state

Before changing the contents
(This is ver.2.1.0.0 of INF)

```
;[5]
[USB_Install.HW]
AddReg = Dev_AddReg

[Dev_AddReg]
HKR,, DeviceInterfaceGUIDs, 0x10000, "{6C477250-0AED-44a8-9
HKR,,"DeviceIdleEnabled",0x00010001,1
HKR,,"DefaultIdleState",0x00010001,1
HKR,,"DefaultIdleTimeout",0x00010001,600000
;HKR,,"UserSetDeviceIdleEnabled",0x00010001,1
;HKR,,"SystemWakeEnabled",0x00010001,1

;[6]
[USB_Install.CoInstallers]
AddReg = CoInstallers_AddReg
```

After changing the contents

Ex.1

```
[Dev_AddReg]
HKR,, DeviceInterfaceGUIDs, 0x10000, "{6C477250-0AED
HKR,,"DeviceIdleEnabled",0x00010001,1
;HKR,,"DefaultIdleState",0x00010001,1
;HKR,,"DefaultIdleTimeout",0x00010001,600000
;HKR,,"UserSetDeviceIdleEnabled",0x00010001,1
;HKR,,"SystemWakeEnabled",0x00010001,1
```

Ex.2

```
[Dev_AddReg]
HKR,, DeviceInterfaceGUIDs, 0x10000, "{6C477250-0AED
HKR,,"DeviceIdleEnabled",0x00010001,1
HKR,,"DefaultIdleState",0x00010001,1
HKR,,"DefaultIdleTimeout",0x00010001,30000
;HKR,,"UserSetDeviceIdleEnabled",0x00010001,1
;HKR,,"SystemWakeEnabled",0x00010001,1
```

The following table is same as one of page 8. Above 2 examples are added to the table.

Settings	Version 1.2.0.0	Version 2.1.0.0	Ex.1	Ex.2
Is the way (A) enabled?	No	Yes	Yes	Yes
Is the way (B) enabled?	No	Yes 10 min after connection	No	Yes 30 sec after connection

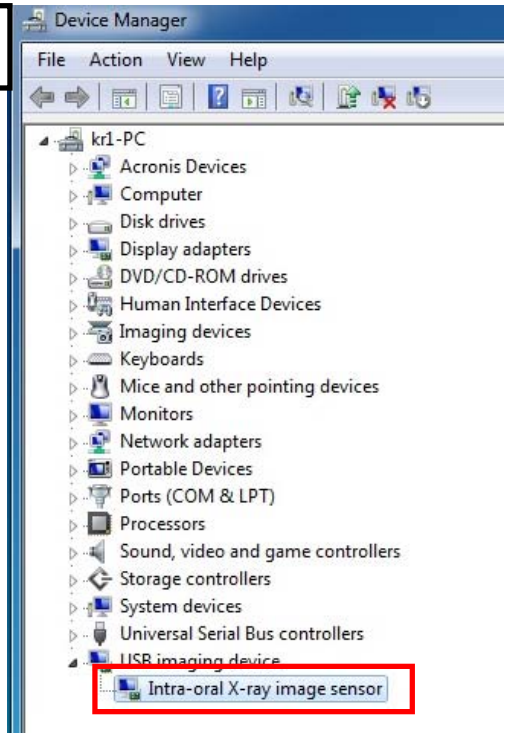
***If you want to know the meanings of the registry keys, please refer to the following URL:**
http://msdn.microsoft.com/en-us/library/windows/hardware/ff728834(v=vs.85).aspx

Section 1: How to make the INF file that is available for the suspend state

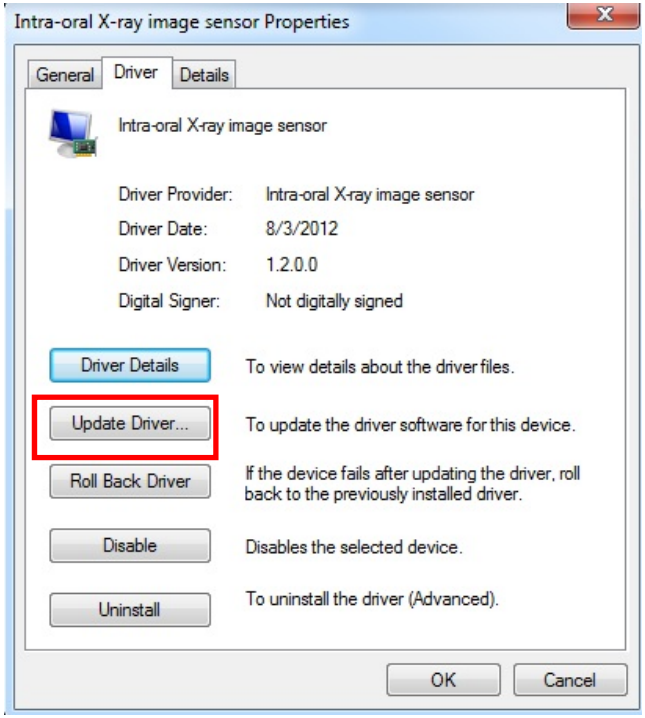
Step3: Connect the sensor to a PC. You can see “Intra-oral X-ray image sensor” in Device Manager as shown below.

Step4: Double-click “Intra-oral X-ray image sensor”, and you can see the following wizard screen. Click the “Update Driver” button.

Step3

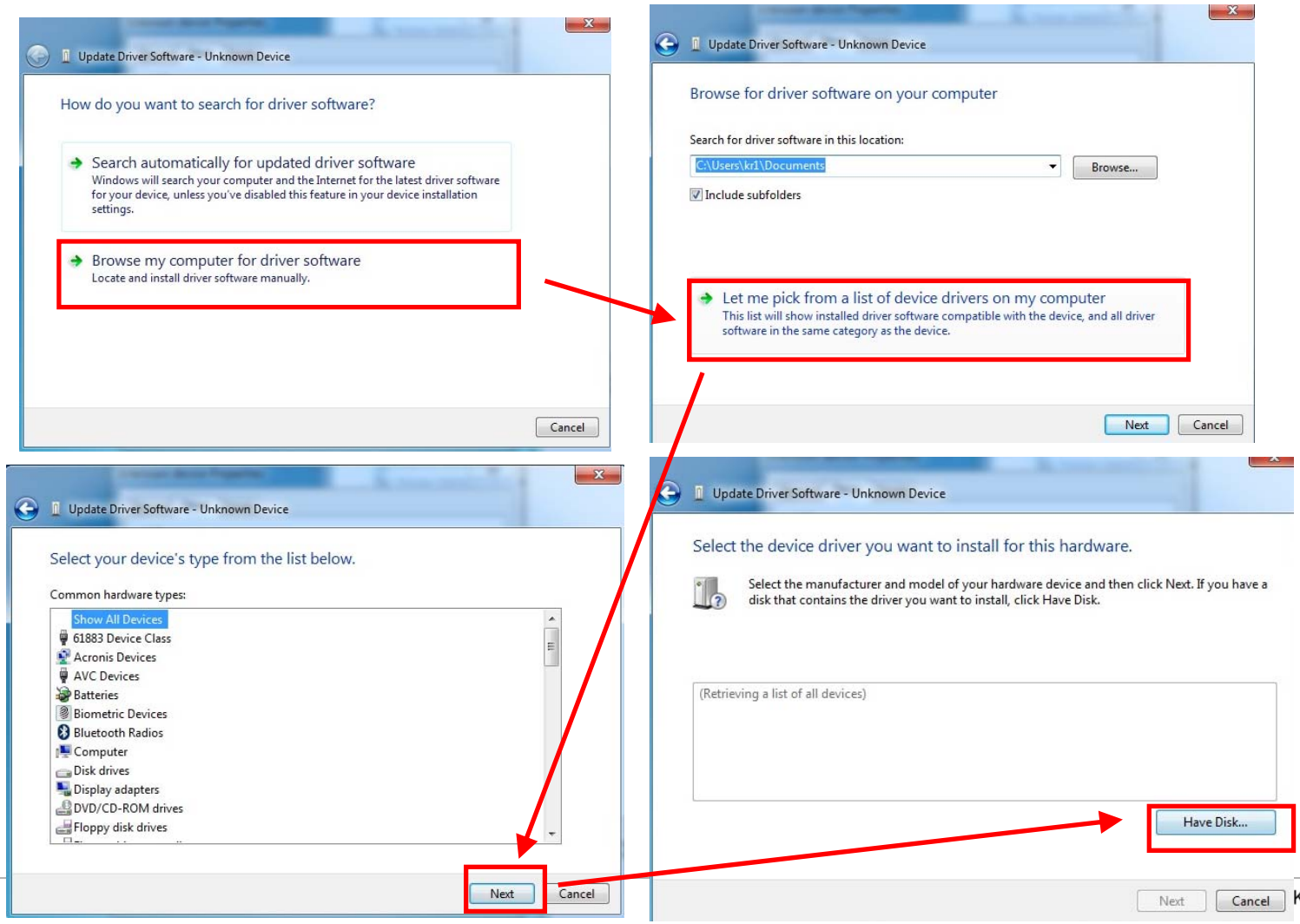


Step4

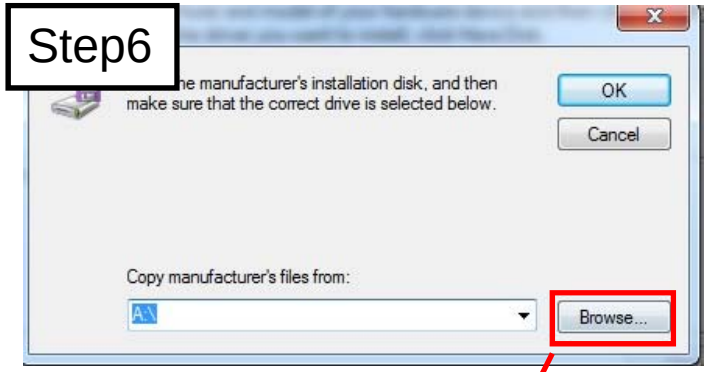


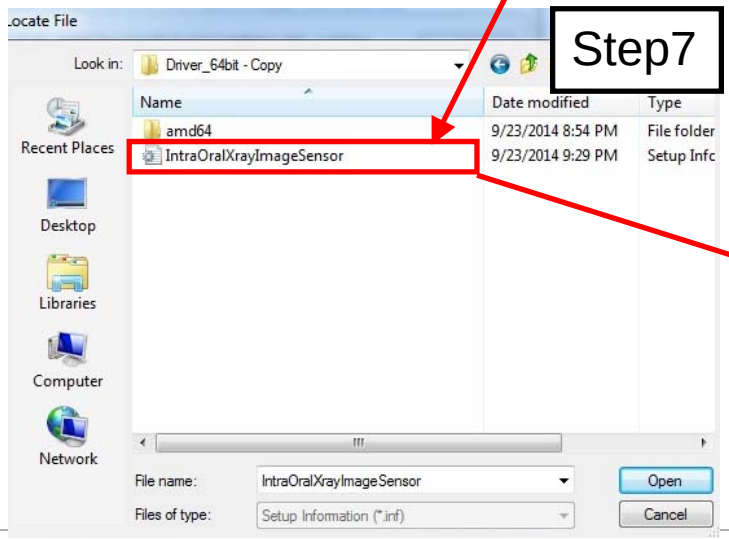
Section 1: How to make the INF file that is available for the suspend state

Step5: Follow the procedures mentioned below.



Section 1: How to make the INF file that is available for the suspend state

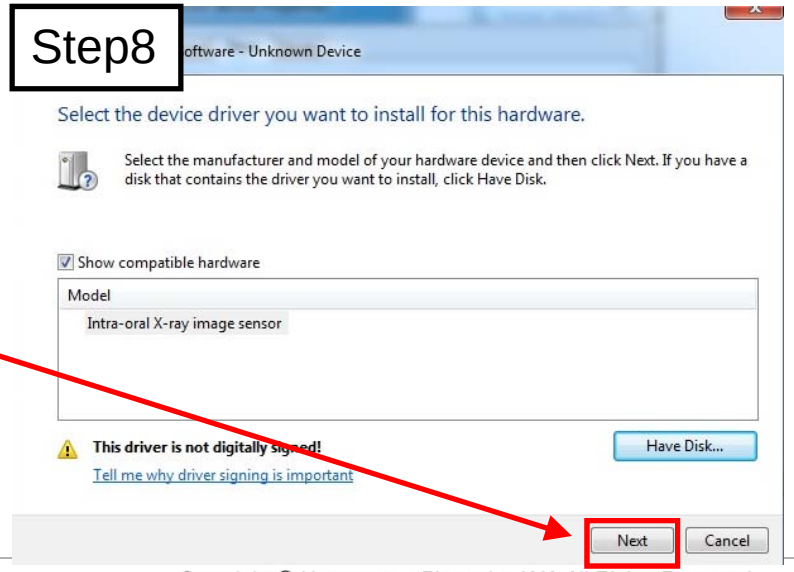




Step6: Click the “Browse” button.

Step7: Select the modified “IntraOralXrayImageSensor.inf”.

Step8: Click the “Next” button.

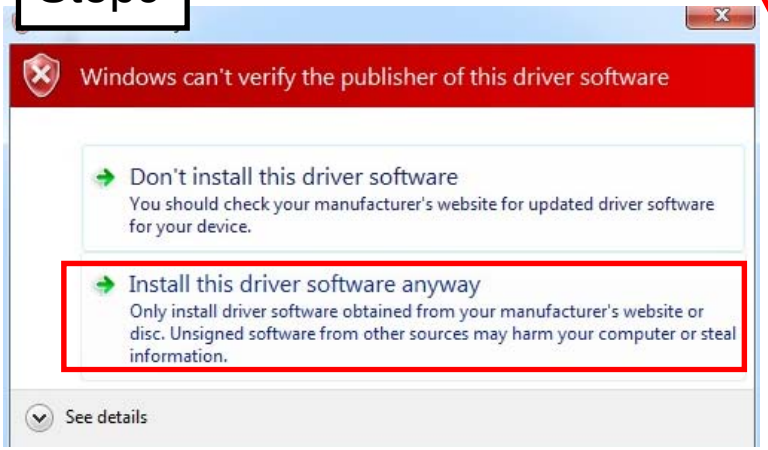


Section 1: How to make the INF file that is available for the suspend state

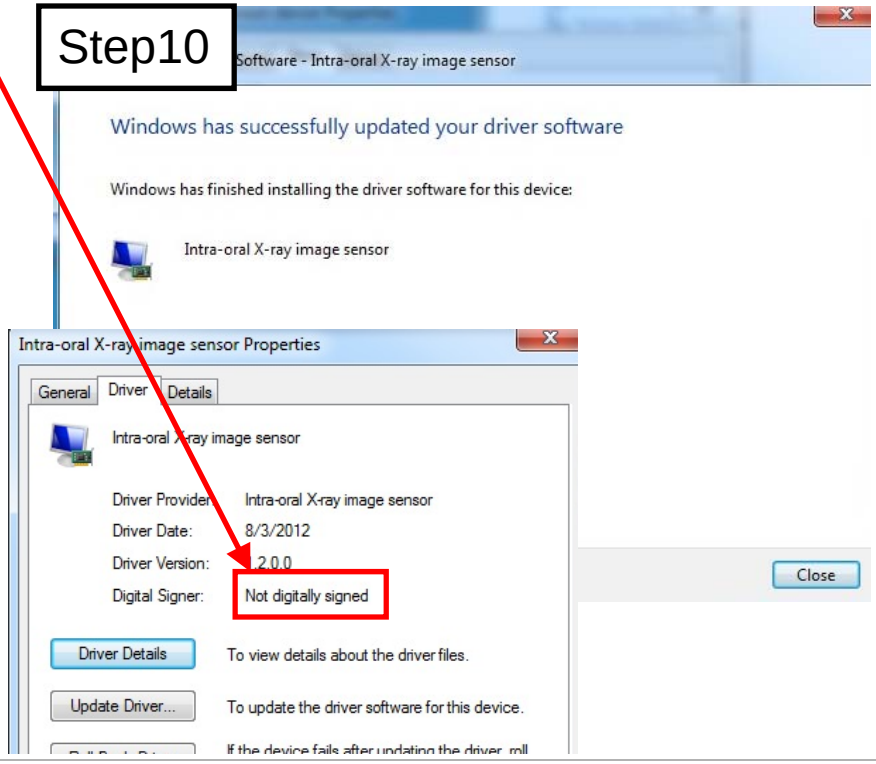
Step9: Click “Install this driver software anyway” button. Do not care this message.

Step10: The change of INF file is finished. You can confirm whether the change is finished properly or not by confirming Properties of “Intra-oral X-ray image sensor” in Device Manager as shown below. If “Digital Signer” item is “Not digitally signed”, the change is finished properly.

Step9



Step10



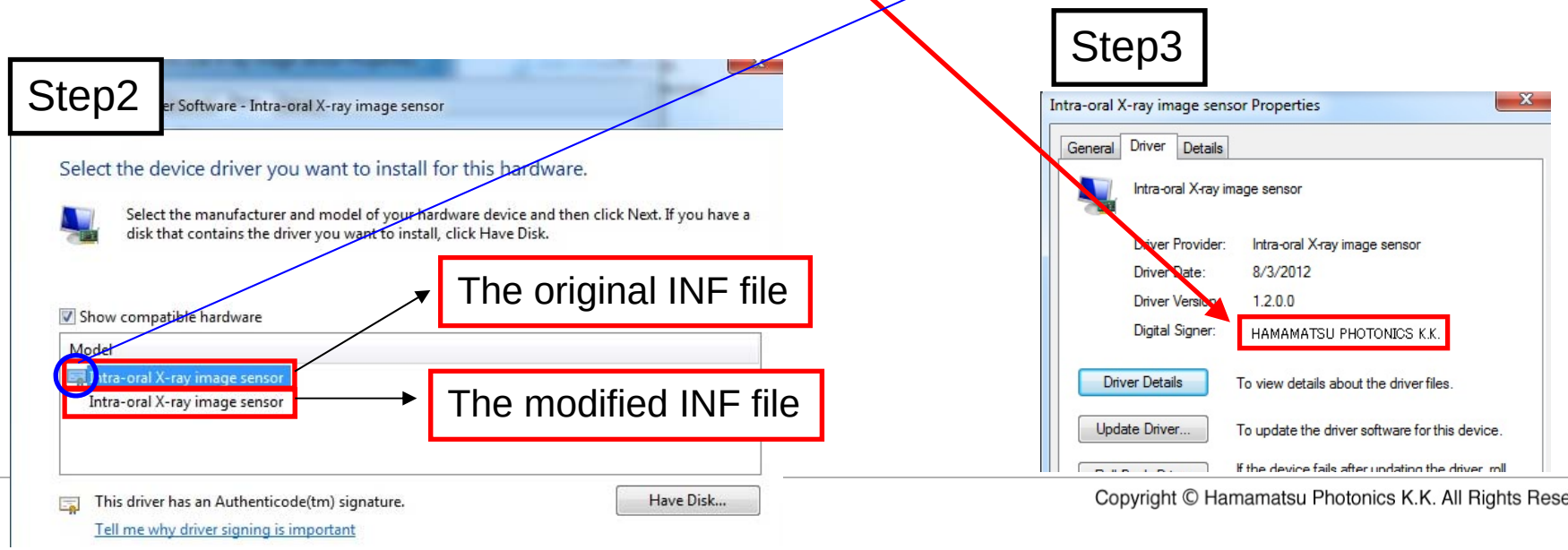
Section 2: How to restore the INF file

If you want to use the INF file that is not available for the suspend state again, please refer to the following procedures.

Step1: Follow Step3 to Step5 of section 1 as mentioned before, and you can see the following wizard.

Step2: Select the “Intra-oral X-ray image sensor” with the Authenticode signature mark .
The INF file is the file that is not available for suspend state and that is in the CD-ROM originally.

Step3: If the “Digital Signer” item is “HAMAMATSU PHOTONICS K.K.”, the restoration is finished properly.

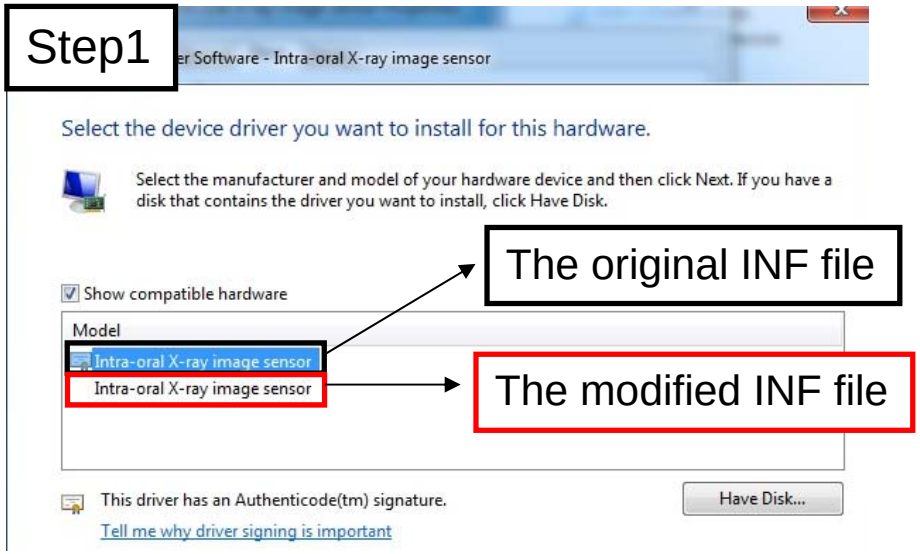


Section 3: How to delete the INF file that is available for the suspend state.

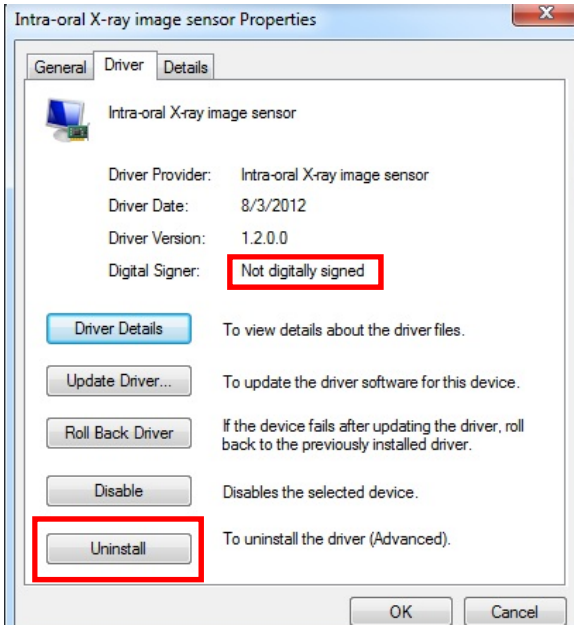
Step1: Set the INF file that is available for suspend state by choosing the modified INF file as shown below.

Step2: Open Properties of “Intra-oral X-ray image sensor” and confirm that “Digital Signer” is “Not digitally signed”. Click “Uninstall” button.

Step1



Step2



HAMAMATSU
PHOTON IS OUR BUSINESS

www.hamamatsu.com