

SIS3153

USB Windows Programmers Guide

SIS GmbH
Harksheider Str. 102A
22399 Hamburg
Germany

Phone: ++49 (0) 40 60 87 305 0
Fax: ++49 (0) 40 60 87 305 20

email: info@struck.de
<http://www.struck.de>

Version: sis3153-M-usb-2-v100-windows_programmers_guide.doc as
of 28.10.2013

Revision Table:

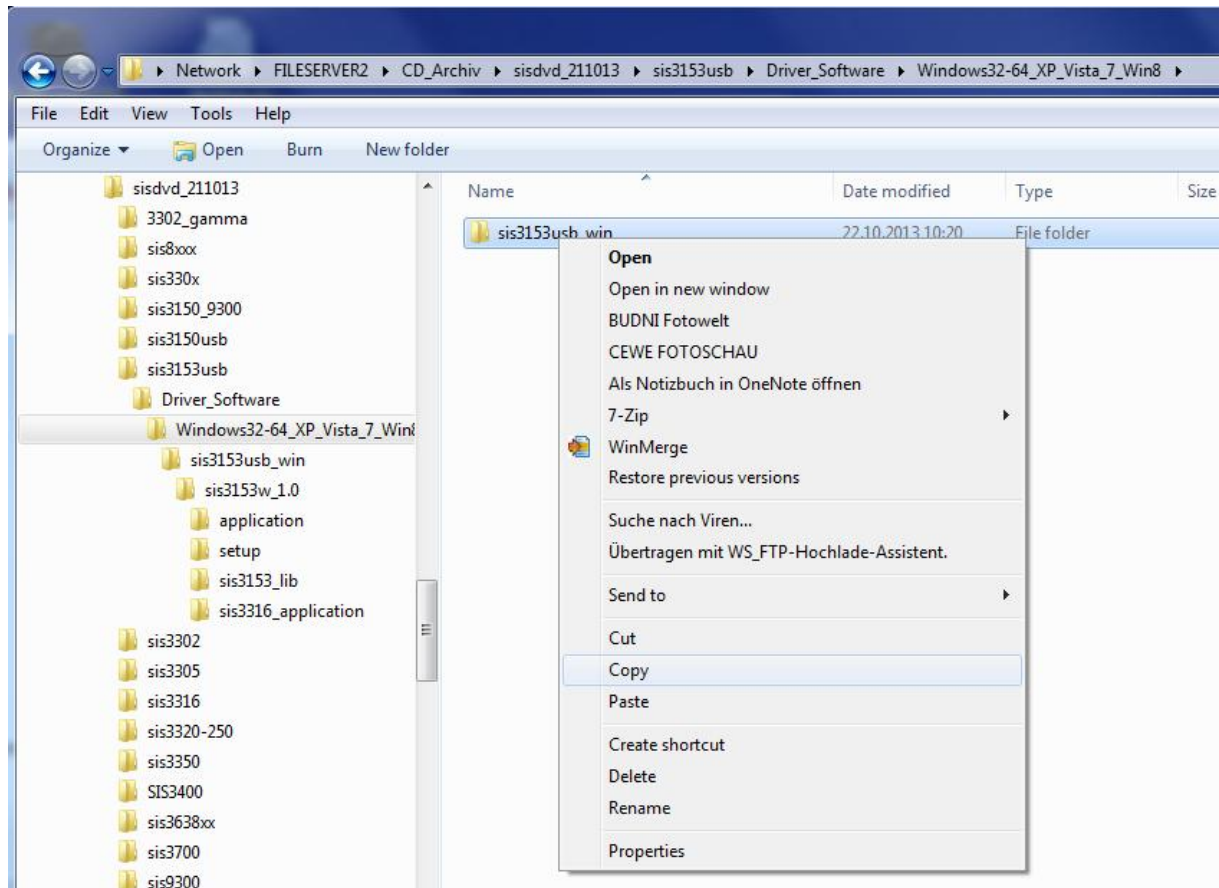
| Revision | Date | Modification |
|----------|------------|------------------------|
| 1.00 | 28.10.2013 | First official release |
| | | |

1 Table of contents

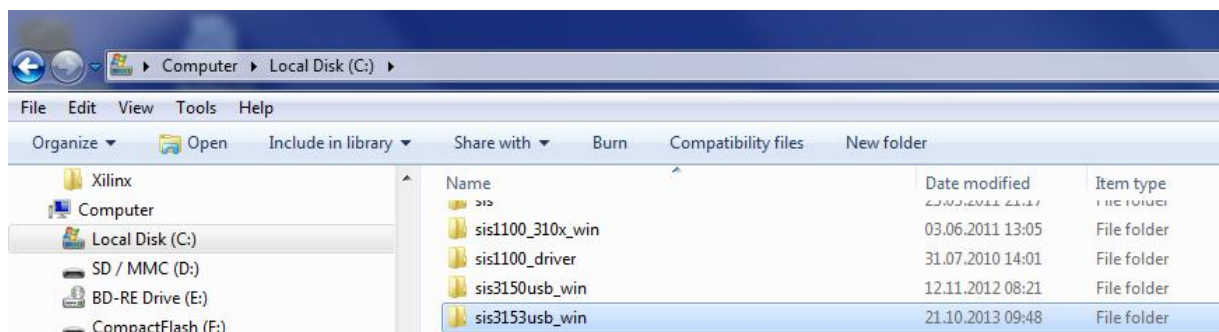
| | | |
|-------|---|----|
| 1 | Table of contents..... | 3 |
| 2 | Windows Driver Installation..... | 4 |
| 2.1 | Initial driver installation (Win7)..... | 5 |
| 2.2 | Update driver (Win7)..... | 10 |
| 3 | CVI SIS3153 USB base program..... | 13 |
| 4 | Labview (Windows) Interface..... | 14 |
| 4.1 | close_device..... | 15 |
| 4.2 | find_all_devices..... | 15 |
| 4.3 | open_device | 15 |
| 4.4 | gHandle.vi..... | 15 |
| 4.5 | sis3153_usb_register_single_read..... | 16 |
| 4.6 | sis3153_usb_register_single_write..... | 16 |
| 4.7 | sis3153_usb_register_dma_read..... | 16 |
| 4.8 | sis3153_usb_register_dma_write | 16 |
| 4.9 | sis3153_usb_vme_single_read..... | 17 |
| 4.10 | sis3153_usb_vme_single_write..... | 17 |
| 4.11 | sis3153_usb_vme_A32D32_read..... | 17 |
| 4.12 | sis3153_usb_vme_A32D32_write..... | 17 |
| 4.13 | sis3153_usb_vme_dma_read..... | 18 |
| 4.14 | sis3153_usb_vme_dma_write | 18 |
| 4.15 | Example 1: Connection Type..... | 19 |
| 4.16 | Example 2: Toggle USER LED..... | 20 |
| 4.17 | Example 3: The sampling test | 21 |
| 5 | VC Sourcecode examples..... | 24 |
| 5.1 | sis3153usb_vme_speed_test..... | 24 |
| 5.1.1 | VC project settings..... | 25 |
| 5.1.2 | Open driver..... | 26 |
| 5.1.3 | Read/write Cycles | 27 |
| 5.2 | sis3153usb_read_sis3305_memory_speed_test..... | 28 |
| 5.3 | sis3316_main_test (with sis3153usb class library)..... | 29 |
| 5.3.1 | VC project settings (with sis3153usb class library) | 30 |
| 5.3.2 | Open driver (with sis3153usb class library) | 31 |
| 5.3.3 | Read/write Cycles (with sis3153usb class library)..... | 32 |
| 6 | Index..... | 33 |

2 Windows Driver Installation

For a quick start and to simplify the use of the example “applications” we recommend to copy the directory “Z:\sisdvd_ddmmyy\sis3153usb\Driver_Software\Windows32-64_XP_Vista_7_Win8\sis3153usb_win “ from the DVD to the drive C: on your computer. All paths of the CVI and VC projects will be correct.



DVD directory



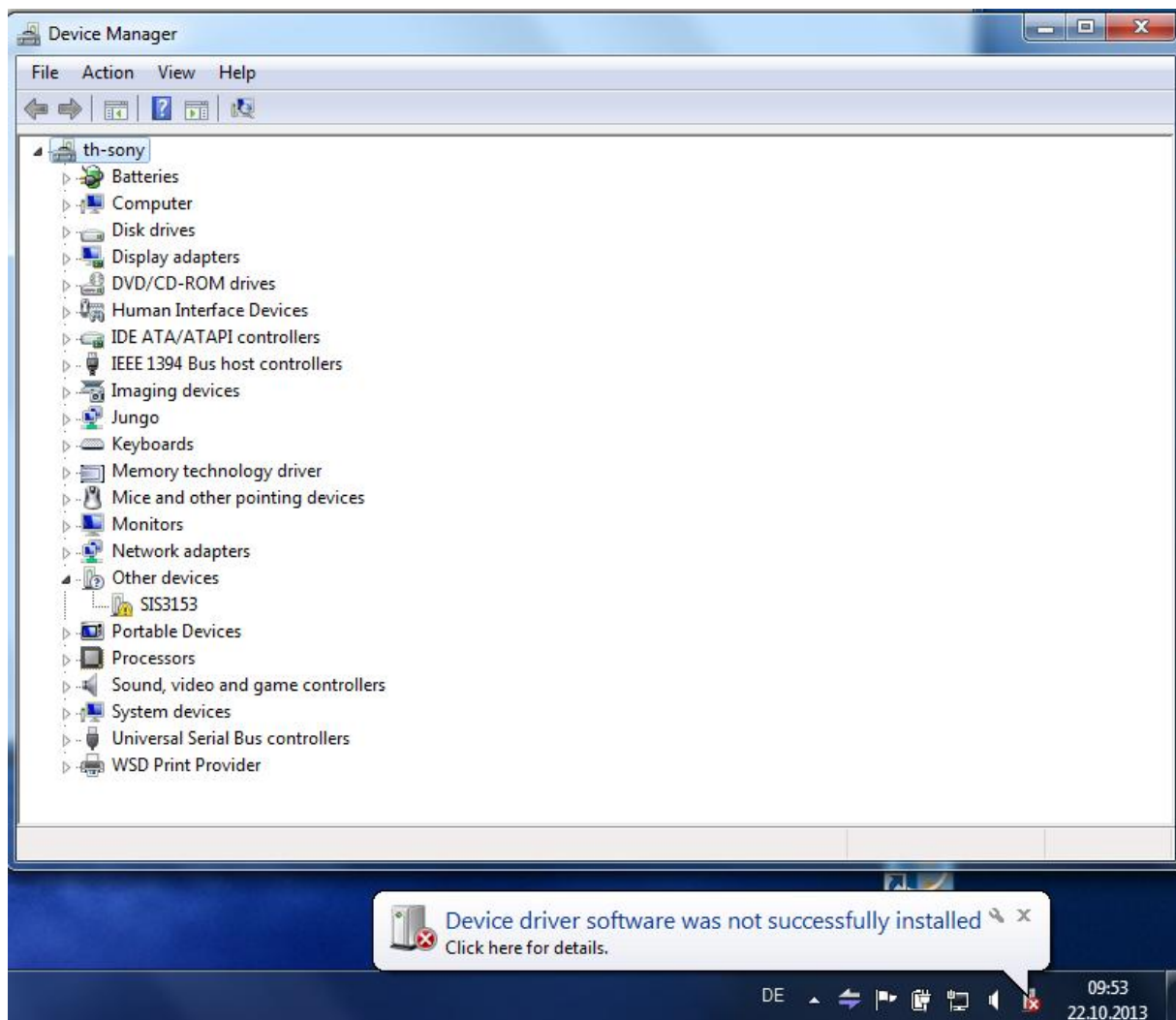
PC C: directory

2.1 Initial driver installation (Win7)

Follow these guidelines when the SIS3153 USB device has never been installed on the current computer.

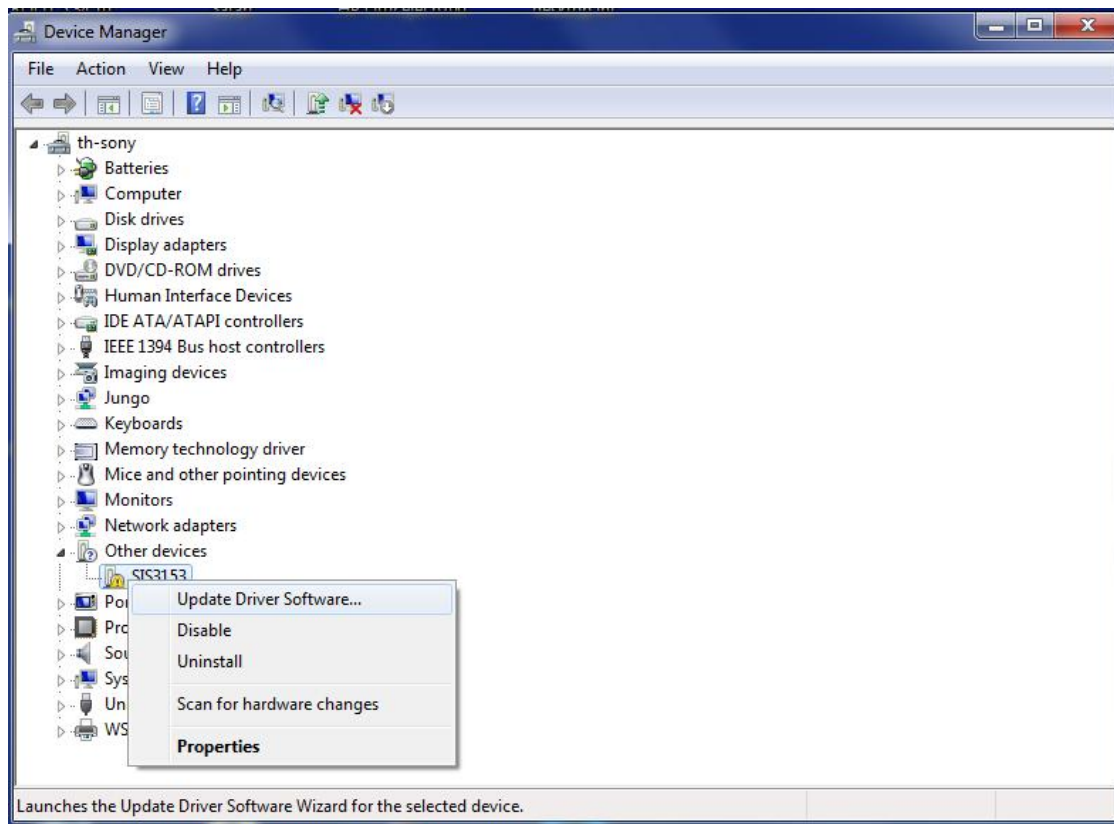
You will get the windows message “Device driver software was not successfully installed”.

Open the Device manager:

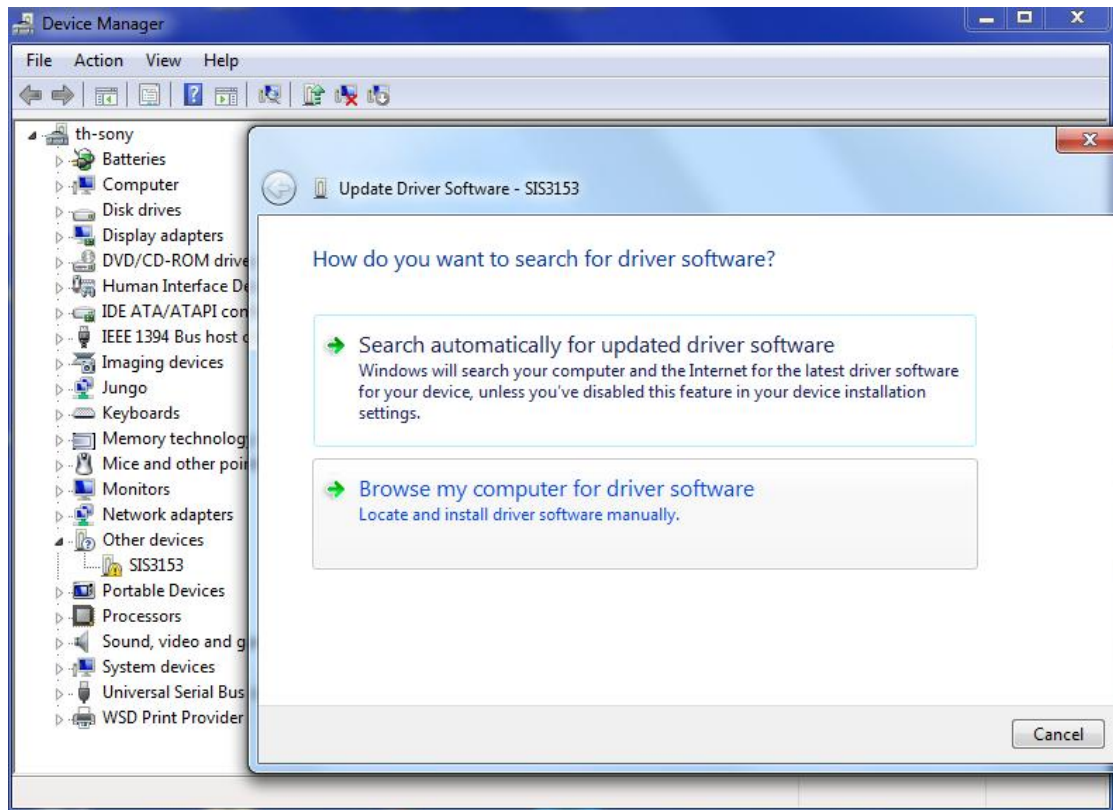


Upon insertion of the USB cable a windows messages will appear.

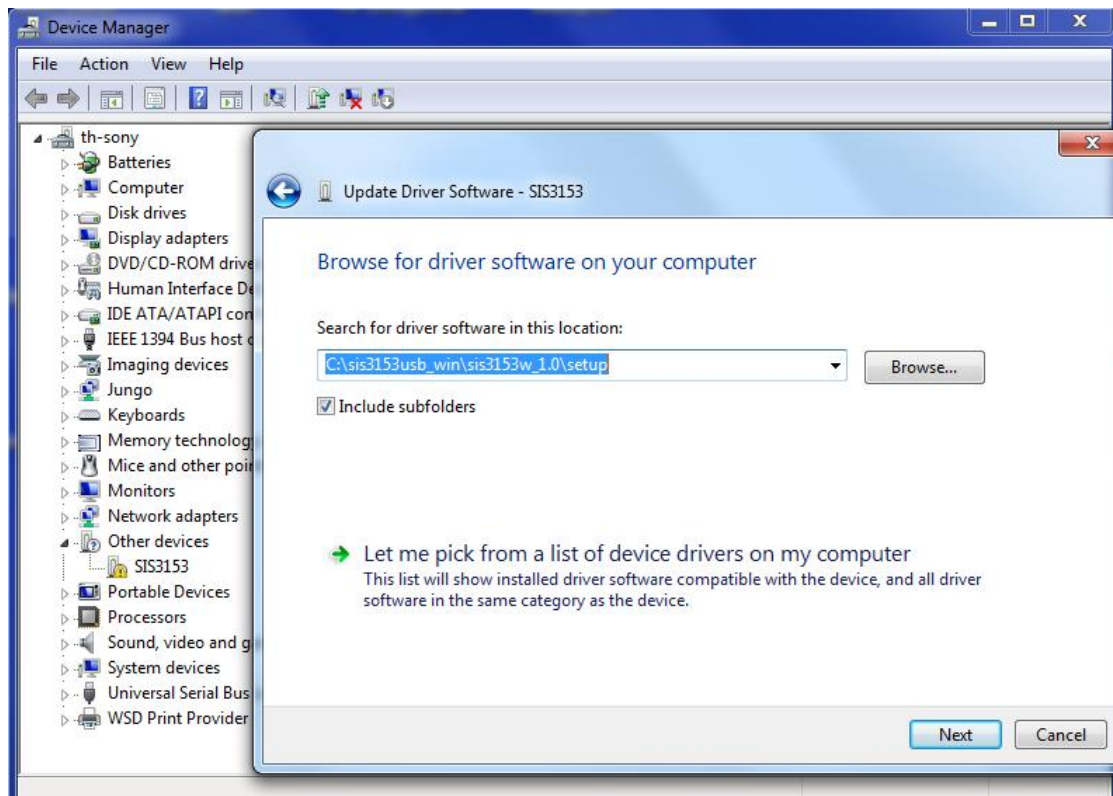
Make a “right click” on the device SIS3153 and click “Update Driver Software” as shown below.



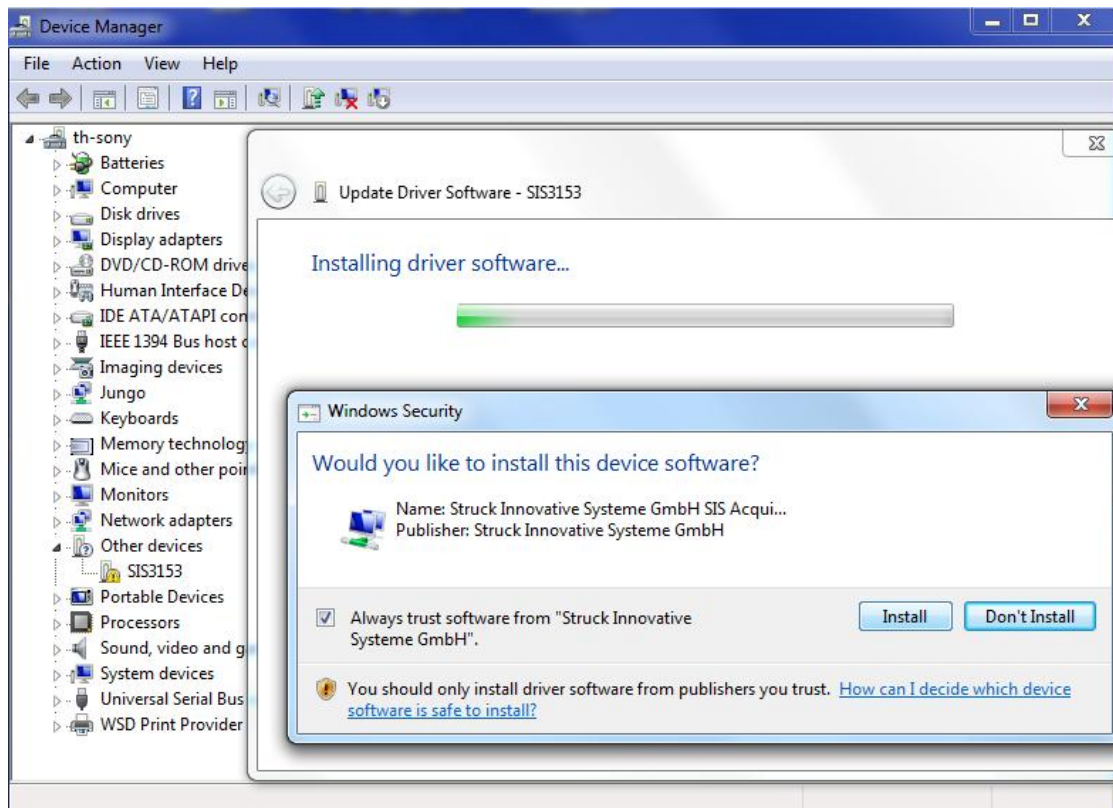
Select “Browse my computer for driver software”.



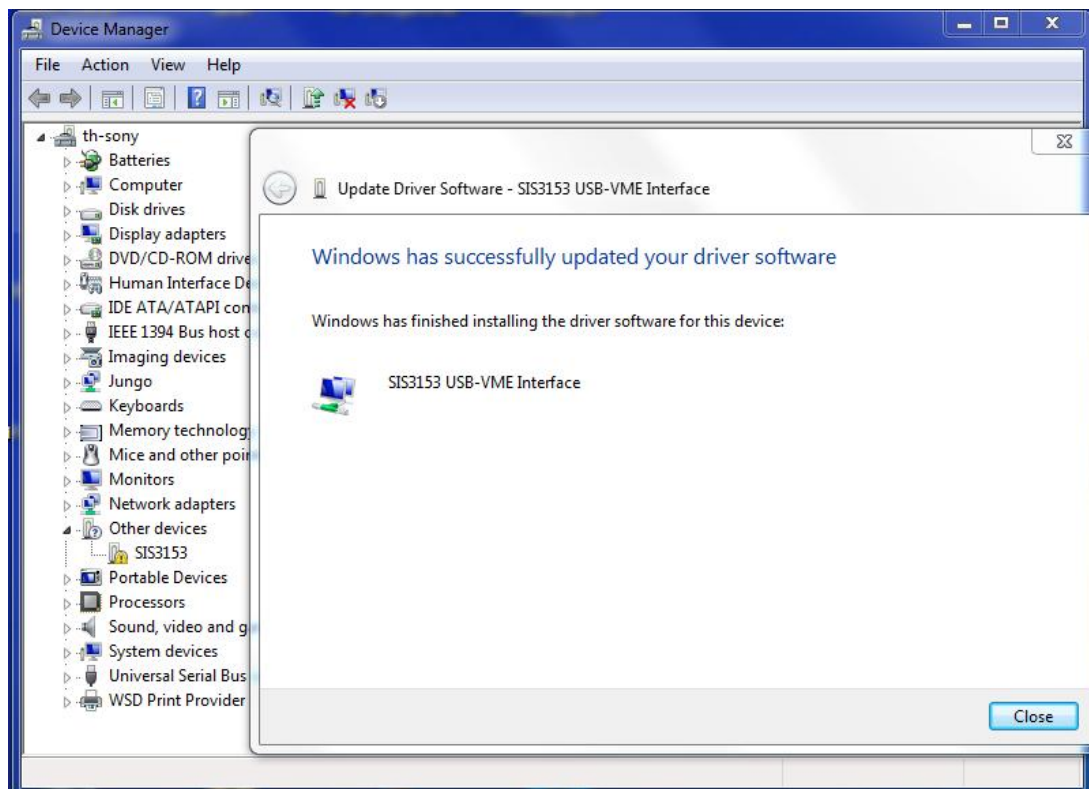
Select the “setup” directory (local disk C: or DVD) and click “Next”.



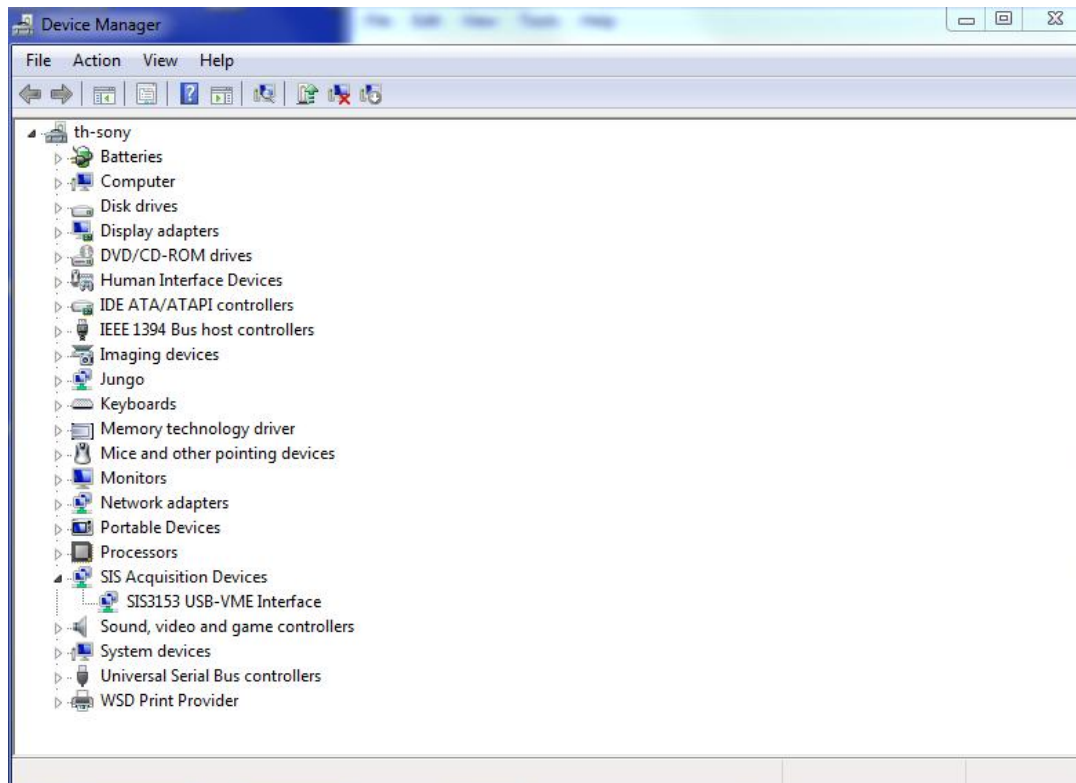
Select “Install” and the driver will be installed:



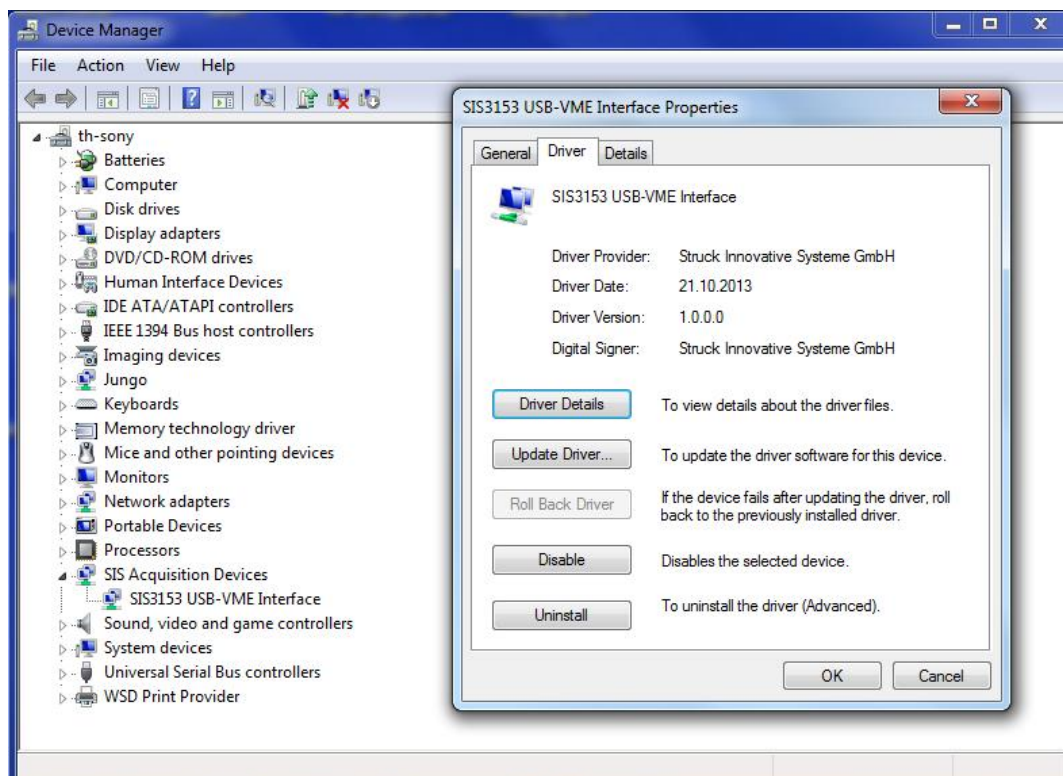
After installing you will get this windows message:



The driver “SIS3153 USB-VME interface” is installed.

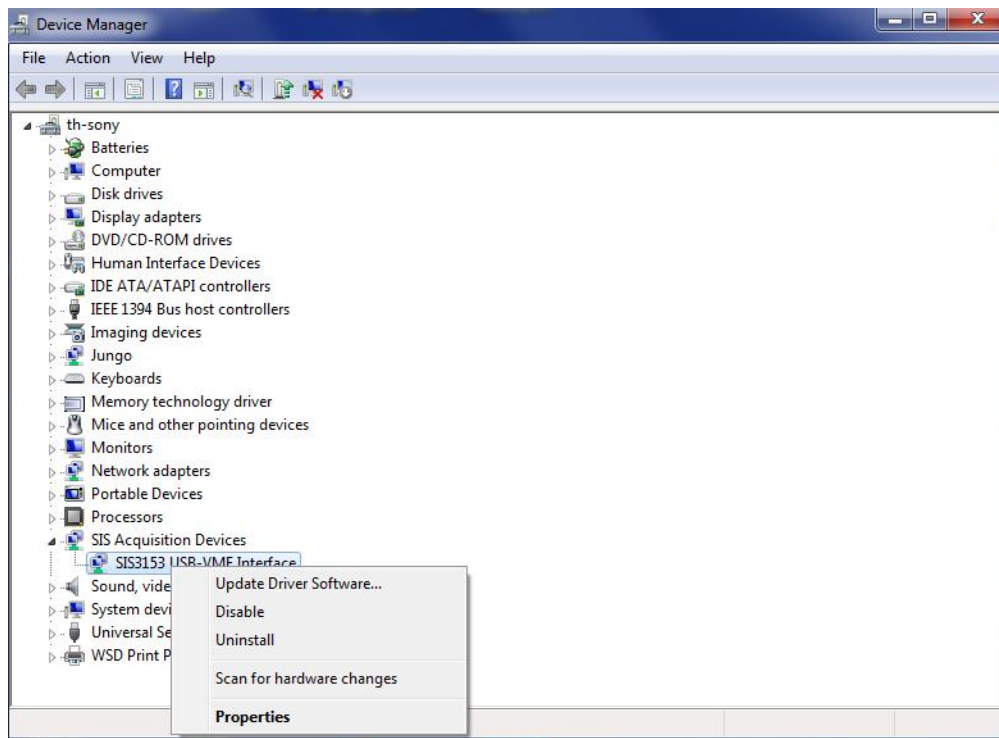


Properties of the device “SIS3153 USB-VME Interface”:

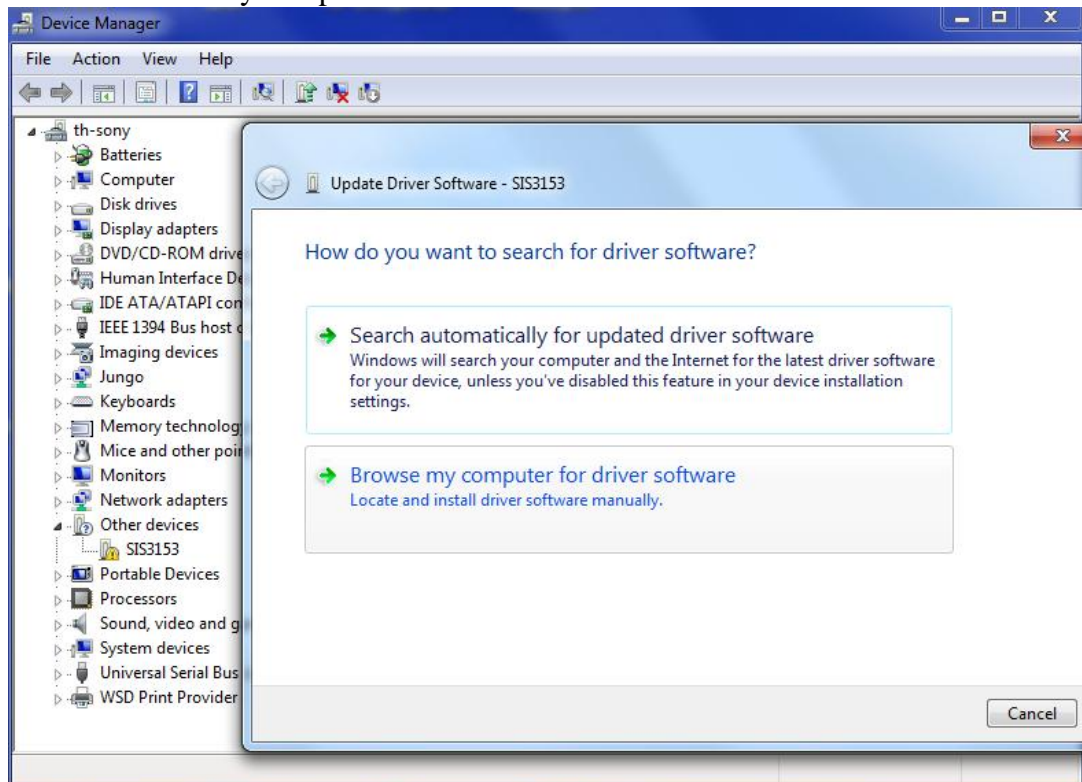


2.2 Update driver (Win7)

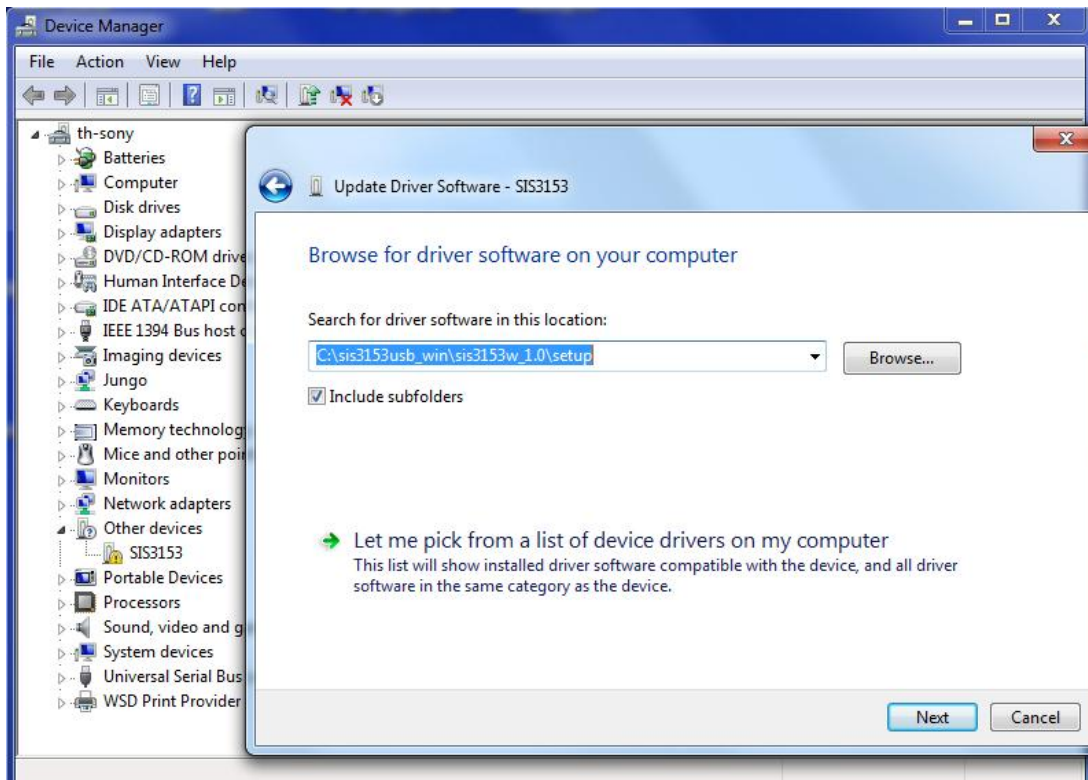
Open the Device Manager and make a “right click” on the device “SIS3153 USB-VME Interface” and click “Update Driver Software ..”.



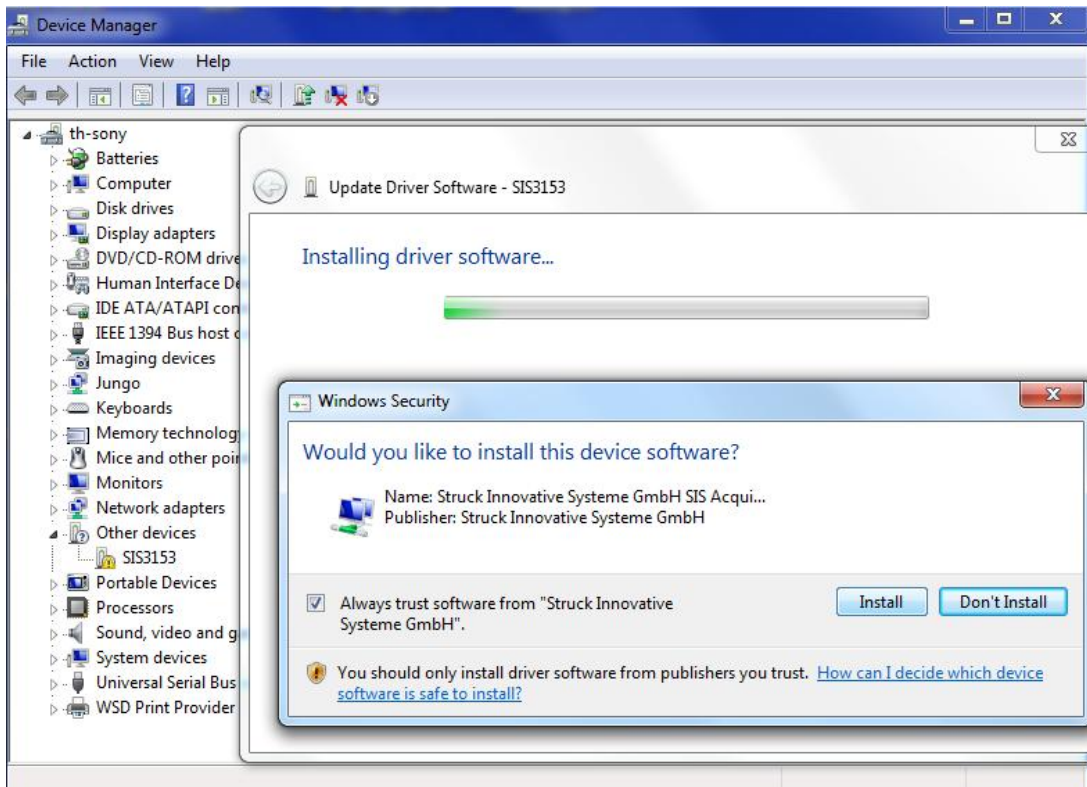
Select “Browse my computer for driver software”.



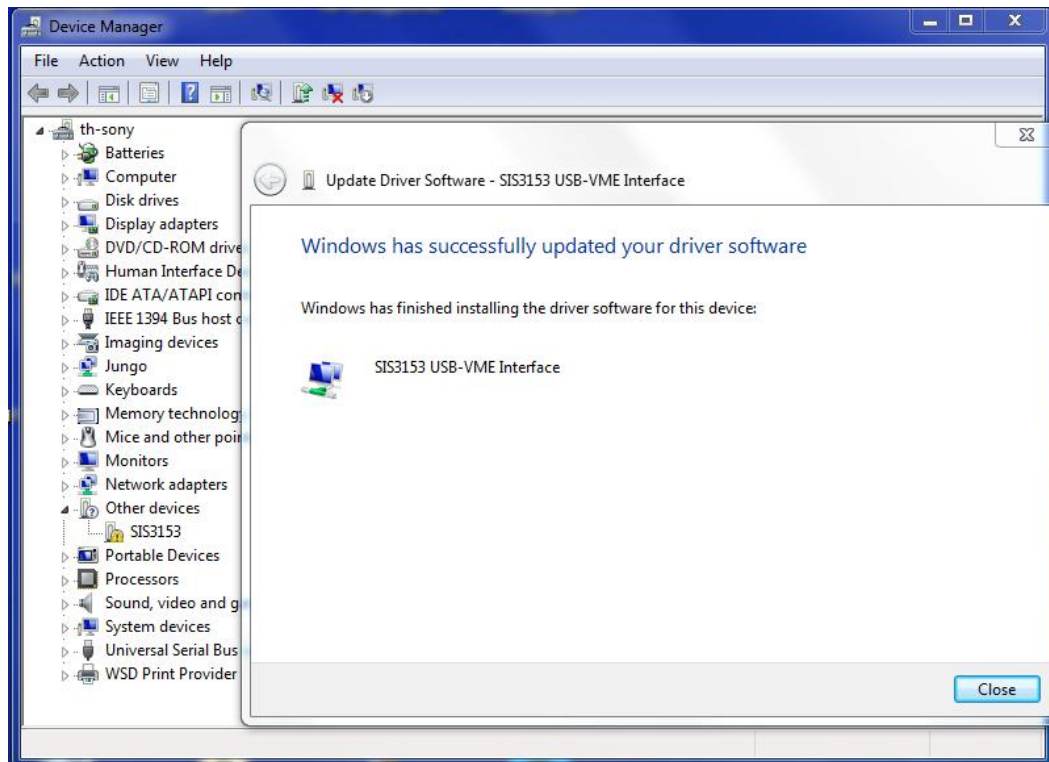
Select the “setup” directory (local disk C: or DVD) and click “Next”.



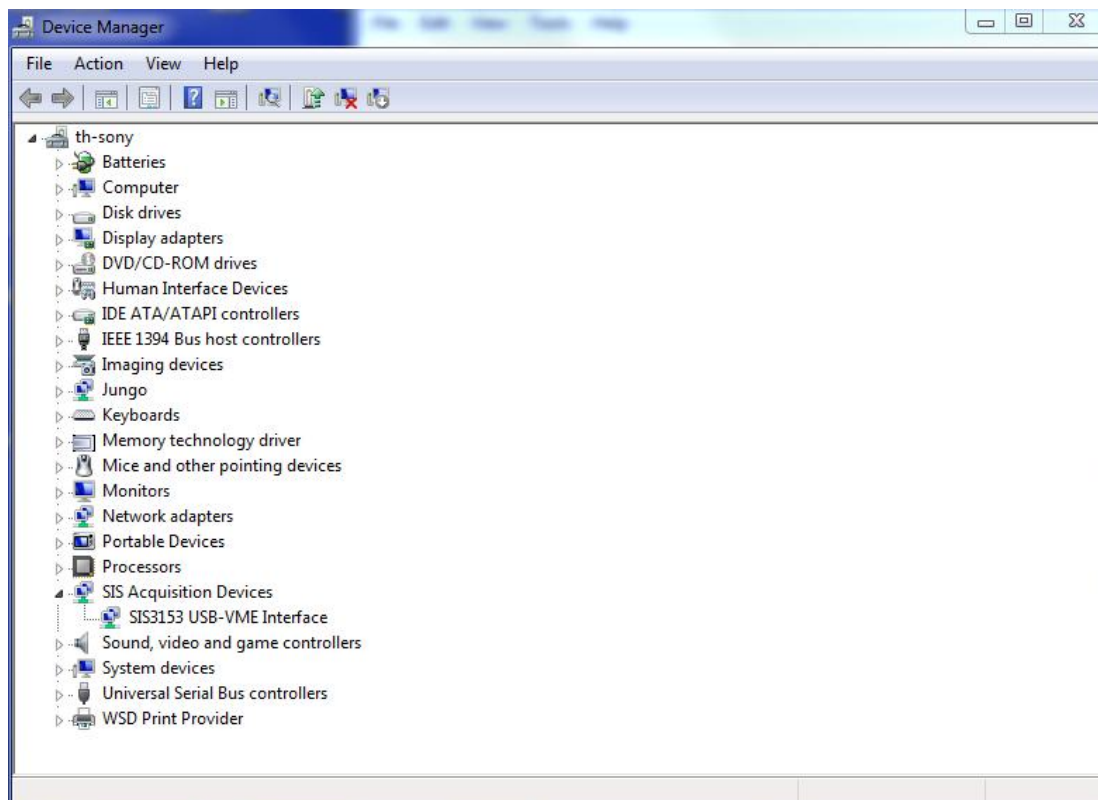
Select “Install” and the driver will be installed:



After installing you will get this windows message:

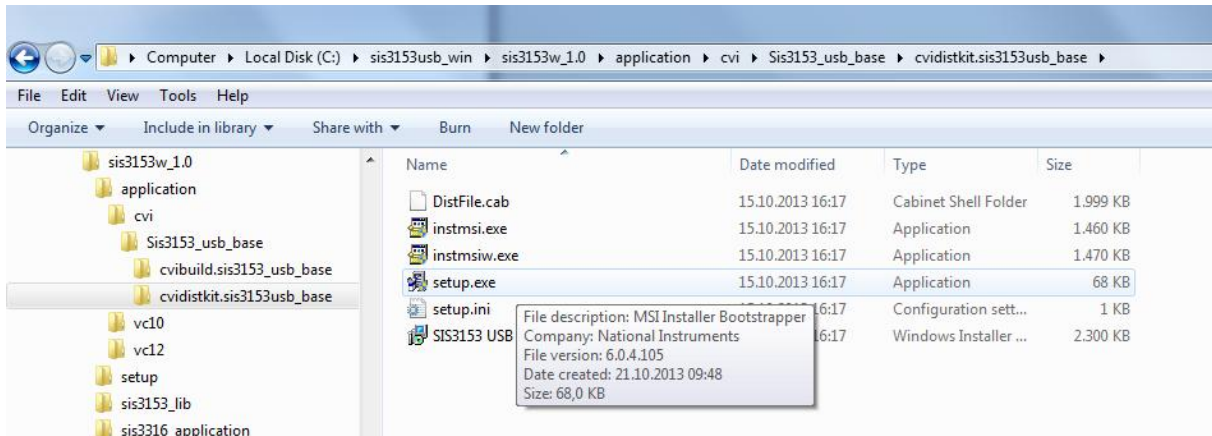


The driver “SIS3153 USB-VME interface” is installed.

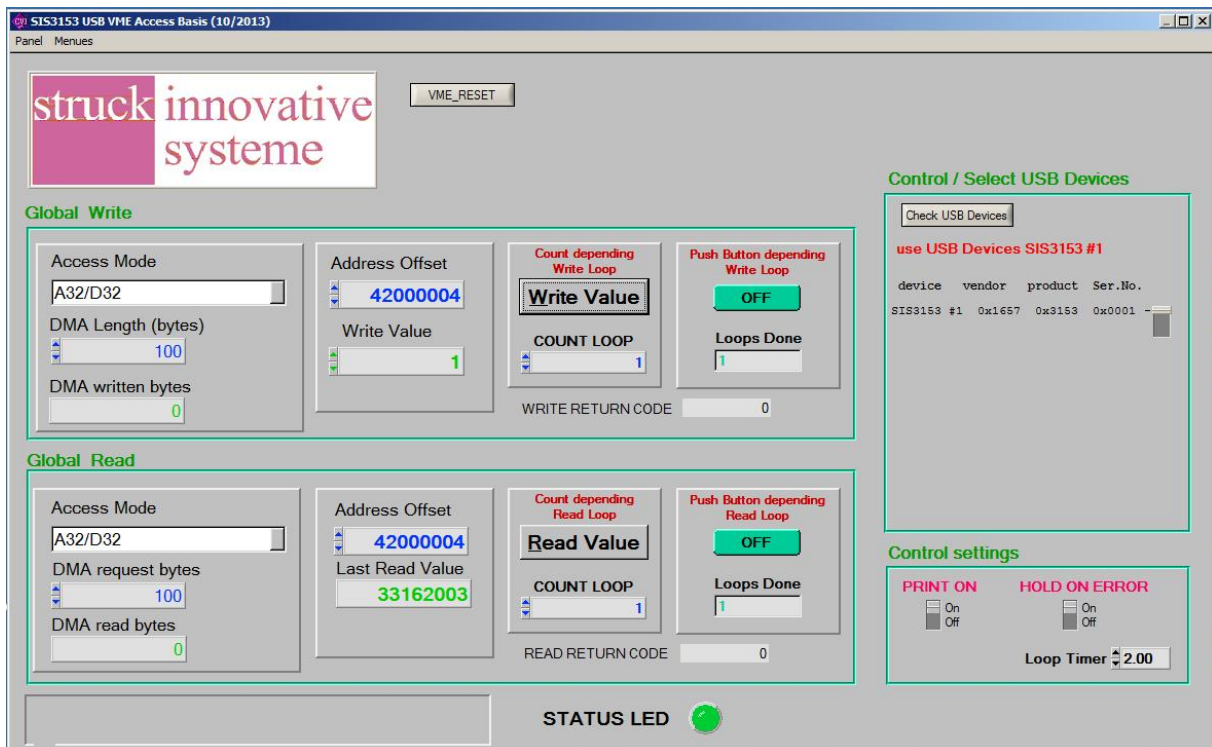


3 CVI SIS3153 USB base program

The SIS3153 USB base program is a convenient tool to execute VME cycles with the USB to VME interface. It has a graphical user interface and is based on National Instruments Labwindows CVI development environment. The program with the CVI runtime engine (RTE) can be installed by running the setup executable in the cvidistkit directory.



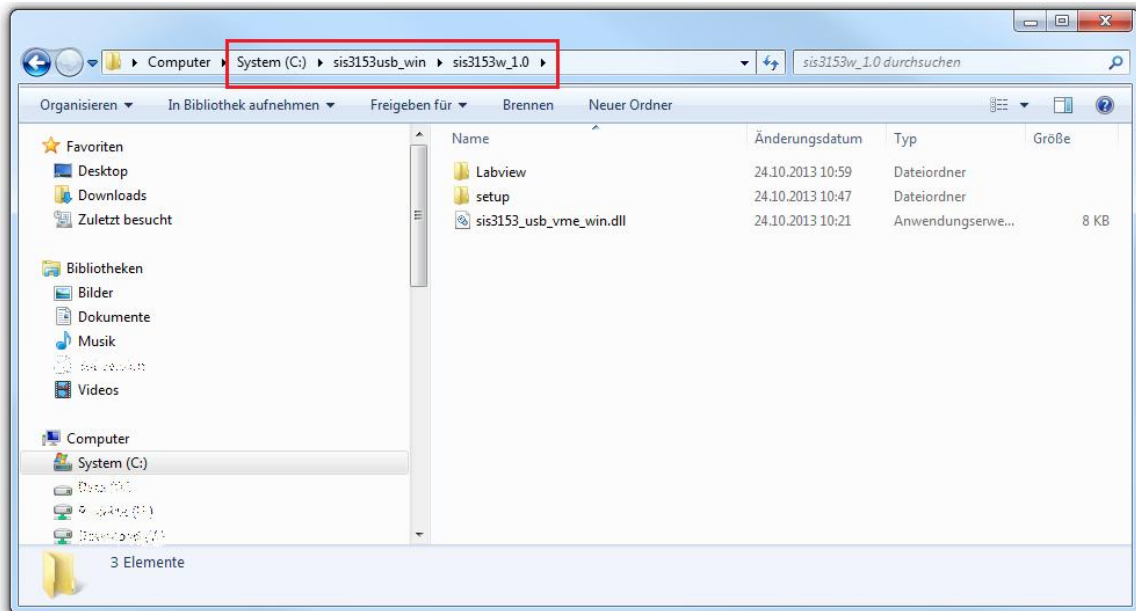
The user interface of the program is shown below.



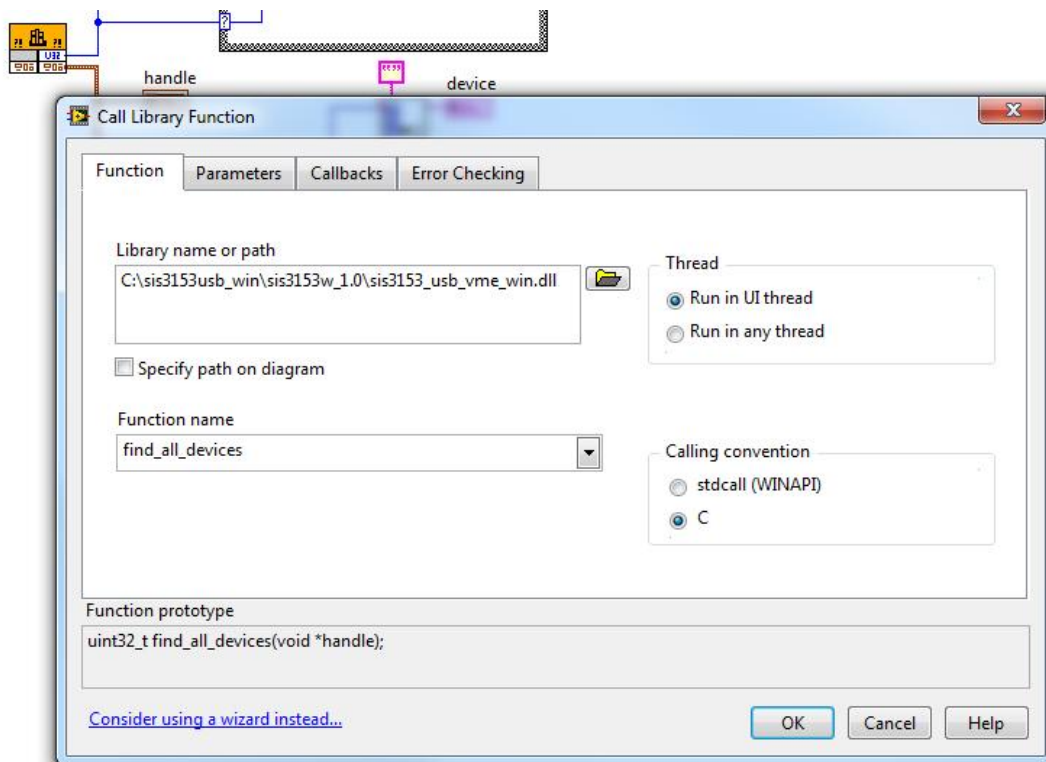
4 Labview (Windows) Interface

The SIS3153 USB Labview interface uses the same DLL as the standard Windows driver. At this point in time the VIs are limited to one SIS3153 USB interface.

Copy the sis3153usb_win directory to your c: drive.

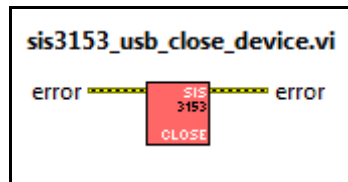


The DLL is assumed to be in `c:\sis3153usb_win\sis3153w_1.0\sis3153_usb_vme_win.dll`



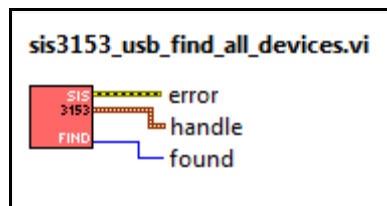
4.1 close_device

The close_device VI is used to close the instance of the driver (after powering off the VME crate eg.). Without closing the driver you will no new handle with a next open_device. Please note, that the close_device VI needs the last valid handle as input parameter (i.e. you can not use the global handle from a failed call to the find_all_devices VI to close the driver).



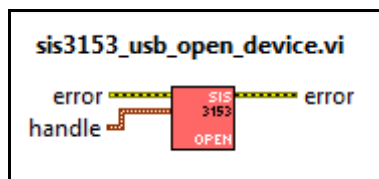
4.2 find_all_devices

This VI scans for SIS3153 USB devices, reports the number of found devices, lists the first device, opens a handle to the first device. Serial number and firmware revision of the first device are reported, also. To extract the information of the handle, please use the "unbundle VI" from the Labview standard library. The error LED stays off upon successful completion.



4.3 open_device

This VI open the first device and generates a simplified handle, which is used by the other VIs.

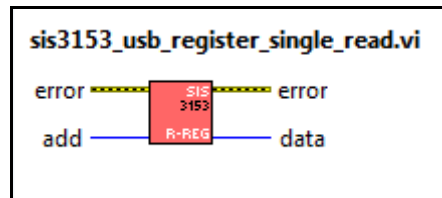


4.4 gHandle.vi

This VI holds the global variables *Handle*, *shortHandle* and *VME base address*, that are used for VME access through the first found interface in the VME calls. *shortHandle* is a simplified version of *Handle* and is used by the following VIs.

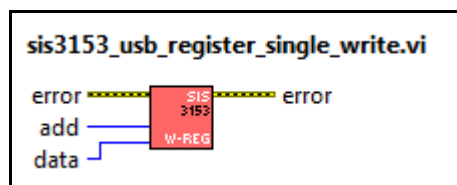
4.5 sis3153_usb_register_single_read

Single word read from register space. A example how to use this VI are attached in chapter 4.15.



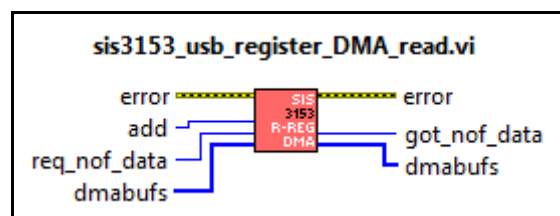
4.6 sis3153_usb_register_single_write

Single word write to register space. Used to control SIS3153 properties. A example how to use this VI are attached in chapter 4.16.



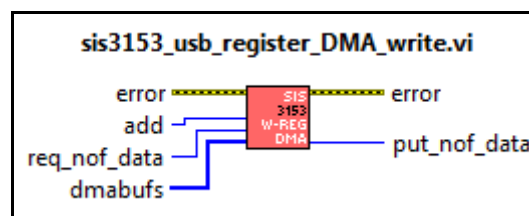
4.7 sis3153_usb_register_dma_read

Block transfer read from register address space.



4.8 sis3153_usb_register_dma_write

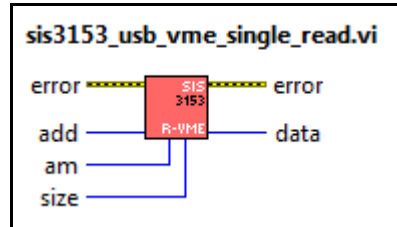
Block transfer write to register address space.



4.9 sis3153_usb_vme_single_read

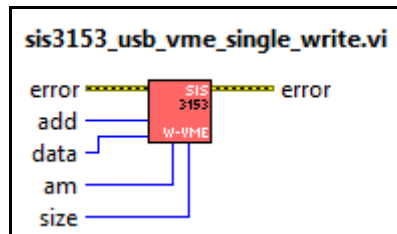
This VI allows the execution of VME single read cycles. Input parameters are *address*, *am* (address modifier) and *size* (4 bytes, 2 bytes or 1 byte).

Successful execution is flagged by a “zero” error code.



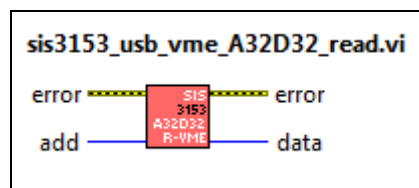
4.10 sis3153_usb_vme_single_write

This VI is the single write counterpart to vme_single_read with the same parameter set.



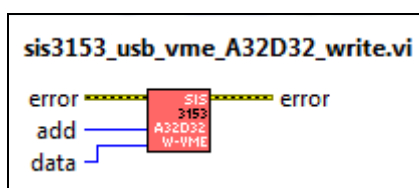
4.11 sis3153_usb_vme_A32D32_read

This is a special version of the sis3153_usb_vme_single_read.vi. Internally *am* are fixed to 0x9 and *size* to 0x4. Additionally the value *add* are merged with the *VME base address*.



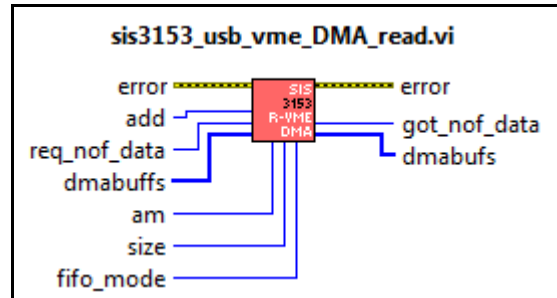
4.12 sis3153_usb_vme_A32D32_write

This is a special version of the sis3153_usb_vme_single_write.vi. Internally *am* are fixed to 0x9 and *size* to 0x4. Additionally the value *add* are merged with the *VME base address*.



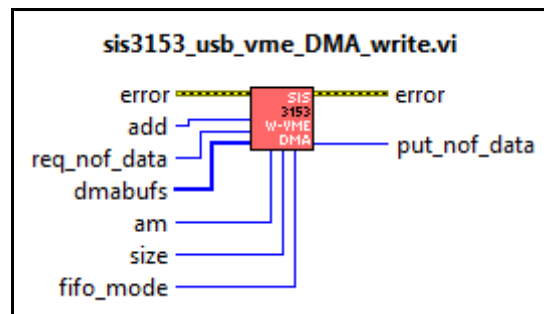
4.13 sis3153_usb_vme_dma_read

This VI is the block transfer VME read cycle VI. An array with *size* elements and *req_nof_data* words is read from the specified address. Address increment is switched off with *fifo_mode*=1. *got_nof_data* displays the number of retrieved words and 4 array elements are displayed in *dmabufs*. You can request a large number of words and check *got_nof_words* for the actual word count after a bus error (after reading data from a SIS3820 multiscaler in FIFO mode e.g.).



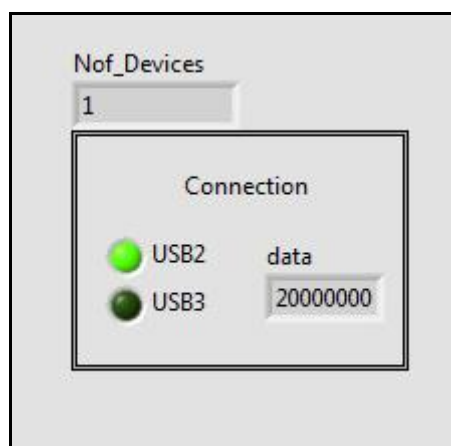
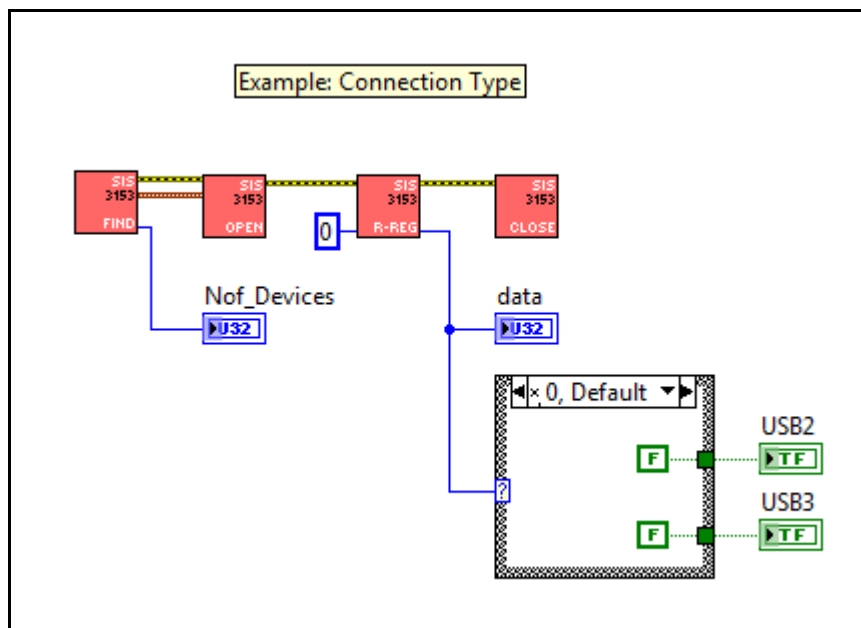
4.14 sis3153_usb_vme_dma_write

This is the block transfer VME write cycle VI. An array with *size* elements and *req_nof_data* words is written to the specified address. Address increment is switched off with *fifo_mode*=1. *put_nof_words* displays the number of written words (which matches the requested value unless an error has occurred).



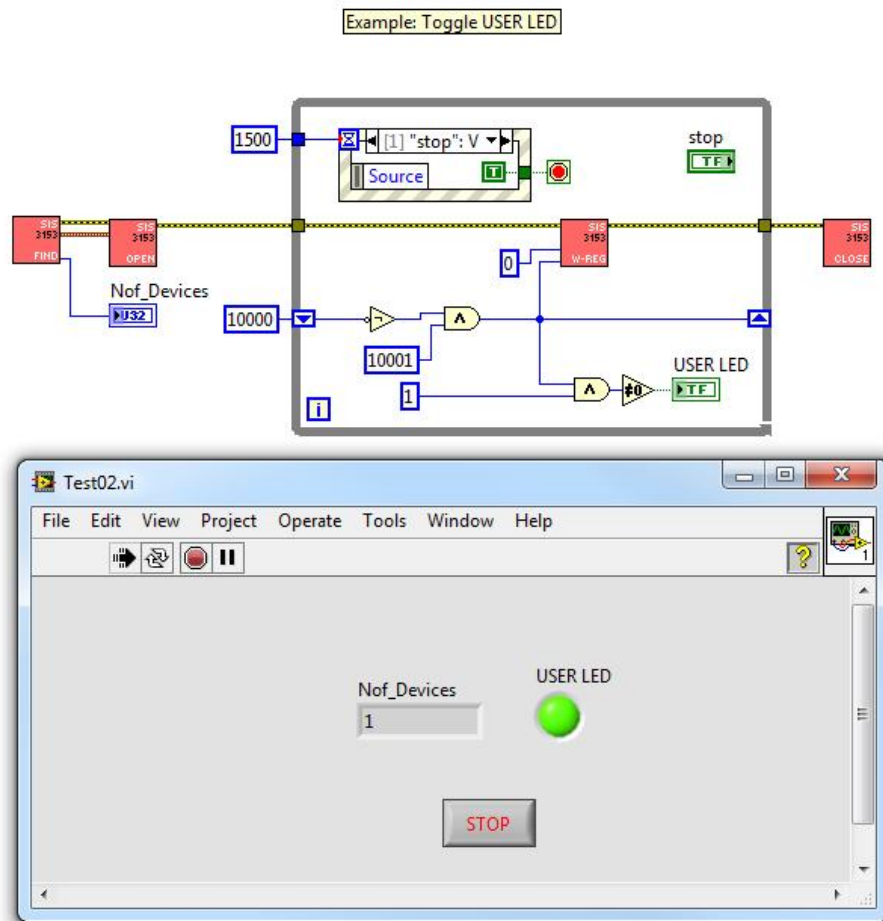
4.15 Example 1: Connection Type

The example below (Test01.vi) demonstrates how to use the VIs. To recognize the type of the connection, between the SIS3150 and a PC, the value of the register at the address 0x0 must be read. A value of 0x20000000 indicates a USB2.0 connection, a value of 0x30000000 indicates a USB3.0 connection.



4.16 Example 2: Toggle USER LED

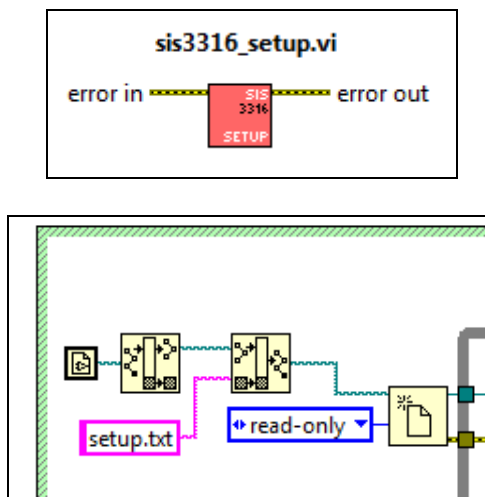
The example below (Test02.vi) illustrates toggling the User LED of the SIS3153 module.



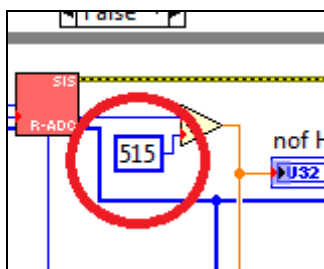
4.17 Example 3: The sampling test

For this test an additional SIS3316 was used. The test (`sample_test.vi`) checks the interaction of the different components of the SIS3316 device that are involved in the data collection (e.g. triggering). In addition to the standard VI two additional VIs are used in this test. The `sis3316_setup.vi` initializes the target device with default values (e.g. the sampling frequency is set to 125MHz). Several registers must be initialized to set up the SIS3316. The initial values are stored in a table (tabulator separated) in the text file `setup.txt`. The structure of the table is shown below:

| | | | |
|-------------------|---------|-----------------|---------|
| 1 = init. command | address | command | comment |
| 2 = sleep comm. | 0 | sleep time [ms] | |

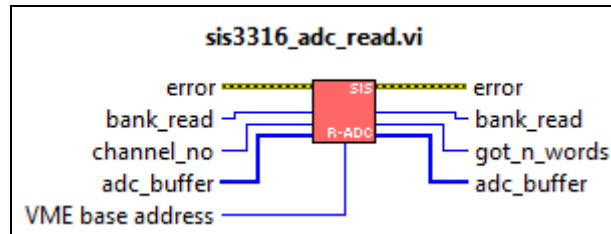


Feel free to experiment with the default values. If the data format and/or the sample lengths have changed, the constant in `sample_test.vi` has to be adapted.

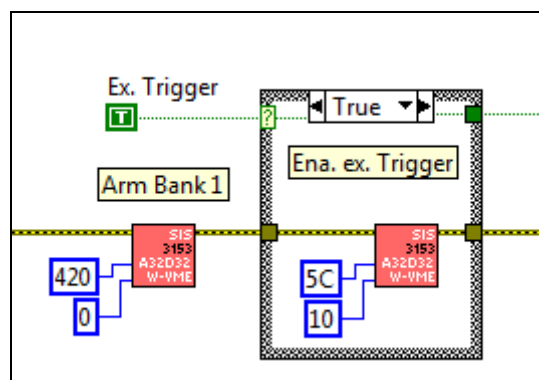


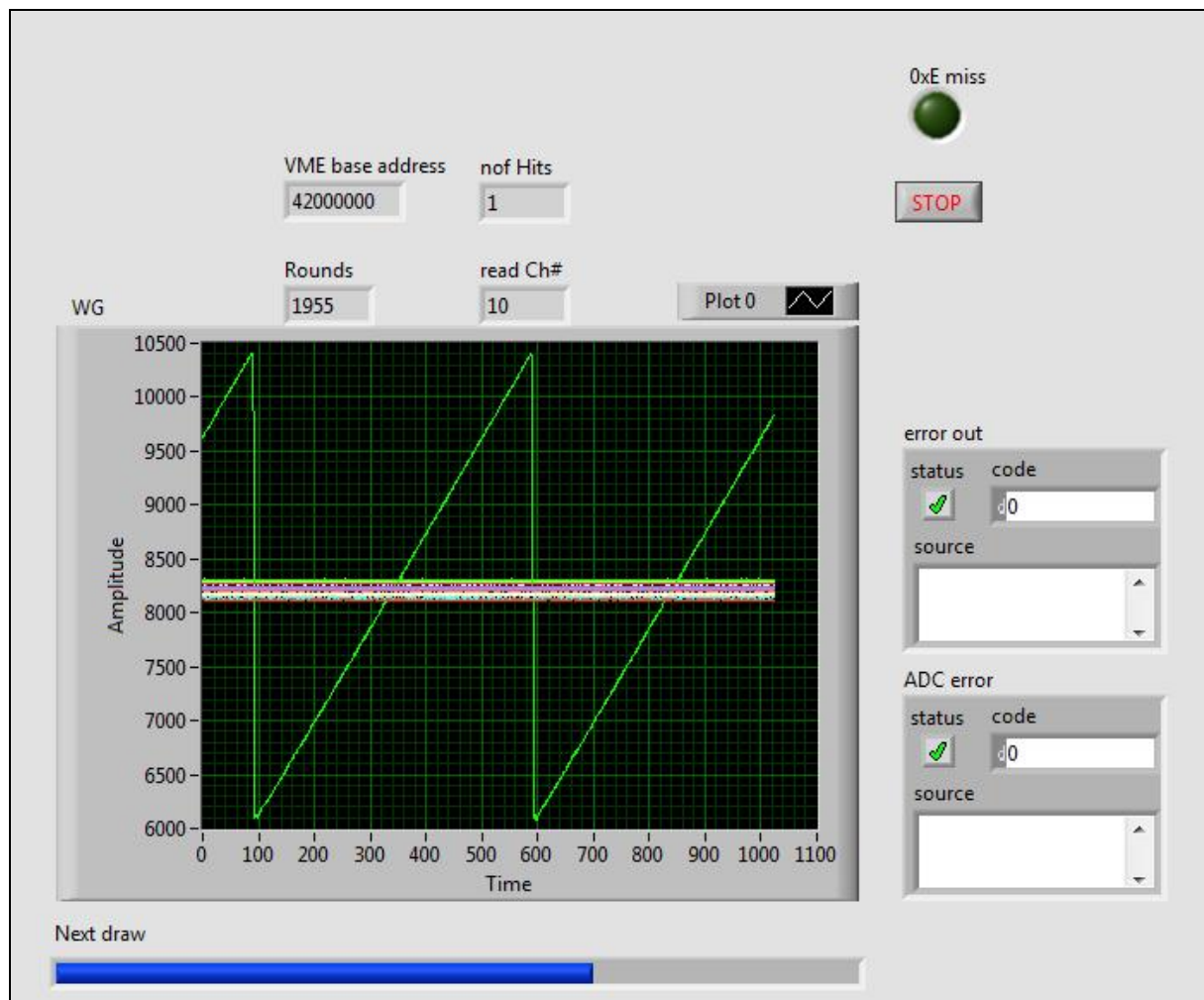
This constant corresponds to the number of long words which represent an event. Refer to the main SIS3316 manual for more information about the Hit/Event Data Format.

The `adc_read.vi` equates to the function `read_MBLT64_Channel1_PreviousBankDataBuffer()`. It allows to request large data packets from the device very comfortably. Refer to the main manual for more information.



Note: It is possible to switch between external and software trigger. Set the “Ex. Trigger” flag (in `sample_test.vi`) to enable the external trigger. Clear the flag to enable the software trigger.





Note: For an optimal performance, only the first hit of each channel will be shown

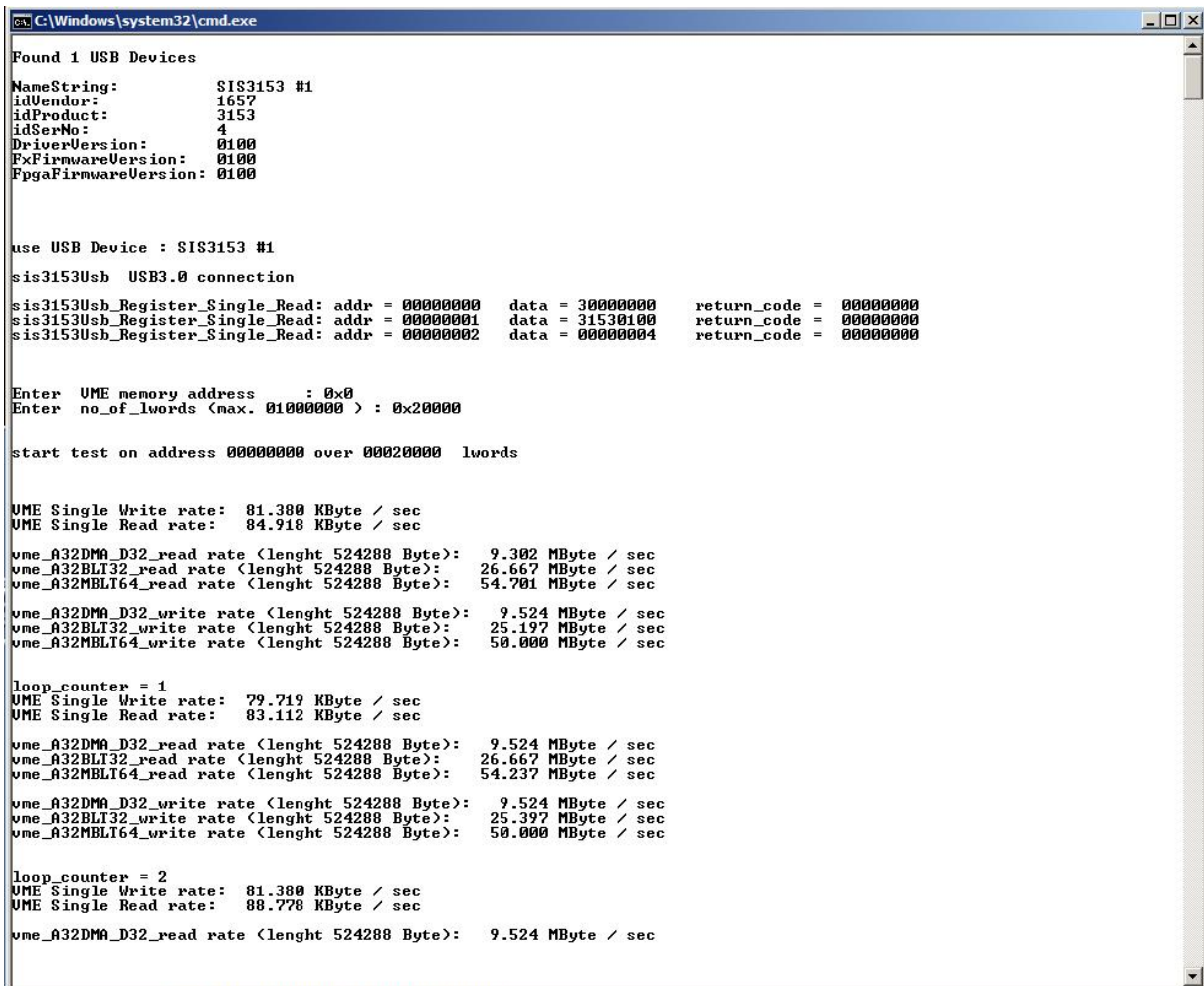
5 VC Sourcecode examples

Some VC10 and VC12 example projects demonstrate the use of the SIS3153 USB-VME interface module.

5.1 sis3153usb_vme_speed_test

This program executes read/write VME cycles (A32_D32, A32DMA_D32, A32_BLT32, A32_MBLT64) from/to a general VME Memory and calculates the data rate.

An old and slow VME Memory was used:



```
C:\Windows\system32\cmd.exe
Found 1 USB Devices
NameString:          SIS3153 #1
idVendor:            1657
idProduct:           3153
idSerNo:             4
DriverVersion:       0100
FirmwareVersion:     0100
FpgaFirmwareVersion: 0100

use USB Device : SIS3153 #1
sis3153usb USB3.0 connection
sis3153usb_Register_Single_Read: addr = 00000000 data = 30000000 return_code = 00000000
sis3153usb_Register_Single_Read: addr = 00000001 data = 31530100 return_code = 00000000
sis3153usb_Register_Single_Read: addr = 00000002 data = 00000004 return_code = 00000000

Enter VME memory address : 0x0
Enter no_of_lwords (max. 01000000) : 0x20000

start test on address 00000000 over 00020000 lwords

VME Single Write rate: 81.380 KByte / sec
VME Single Read rate: 84.918 KByte / sec

vme_A32DMA_D32_read rate (length 524288 Byte): 9.302 MByte / sec
vme_A32BLT32_read rate (length 524288 Byte): 26.667 MByte / sec
vme_A32MBLT64_read rate (length 524288 Byte): 54.701 MByte / sec

vme_A32DMA_D32_write rate (length 524288 Byte): 9.524 MByte / sec
vme_A32BLT32_write rate (length 524288 Byte): 25.197 MByte / sec
vme_A32MBLT64_write rate (length 524288 Byte): 50.000 MByte / sec

loop_counter = 1
VME Single Write rate: 79.719 KByte / sec
VME Single Read rate: 83.112 KByte / sec

vme_A32DMA_D32_read rate (length 524288 Byte): 9.524 MByte / sec
vme_A32BLT32_read rate (length 524288 Byte): 26.667 MByte / sec
vme_A32MBLT64_read rate (length 524288 Byte): 54.237 MByte / sec

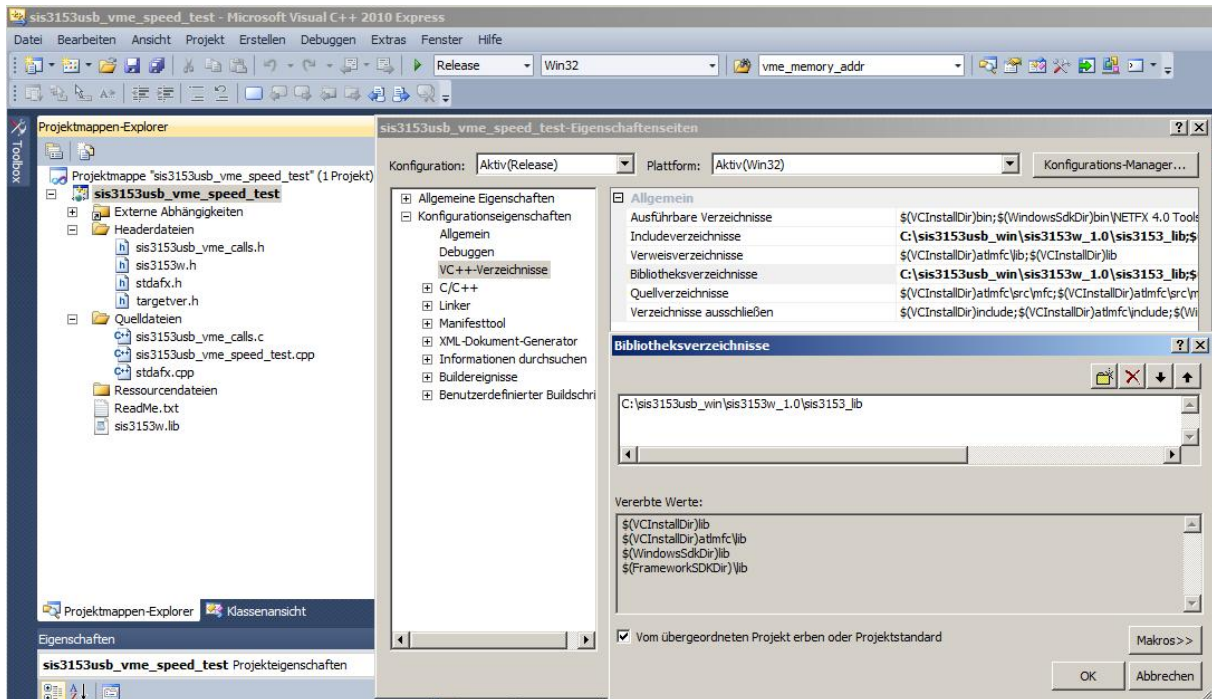
vme_A32DMA_D32_write rate (length 524288 Byte): 9.524 MByte / sec
vme_A32BLT32_write rate (length 524288 Byte): 25.397 MByte / sec
vme_A32MBLT64_write rate (length 524288 Byte): 50.000 MByte / sec

loop_counter = 2
VME Single Write rate: 81.380 KByte / sec
VME Single Read rate: 88.778 KByte / sec

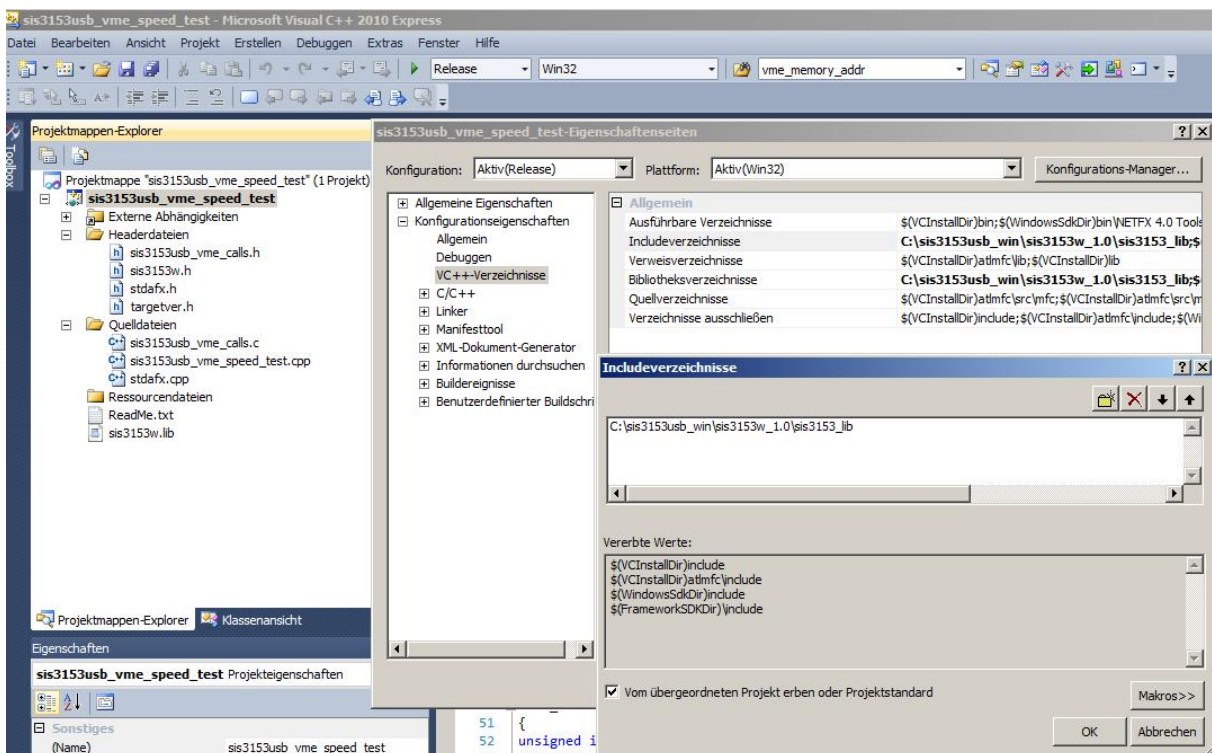
vme_A32DMA_D32_read rate (length 524288 Byte): 9.524 MByte / sec
```

5.1.1 VC project settings

This screenshot shows the setting of the library path:



This screenshot shows the setting of the include path:



5.1.2 Open driver

This screenshot shows how to open a SIS3153USB device:

```
75  /*****
76  *      init sis3153USB
77  *****/
78
79  do {
80      FindAll_SIS3153USB_Devices(gl_sis3153_device_information_struct, &gl_nof_usbdevices, MAX_USB_DEV_NUMBER) ;
81      if (gl_nof_usbdevices == 0) {
82          printf("Found no USB Devices !!!!!\n") ;
83          Sleep(500);
84      }
85  } while (gl_nof_usbdevices == 0) ;
86  if (gl_nof_usbdevices == 0) {
87      gl_USBDevice = NULL ;
88      printf("Found no USB Devices !!!!!\n") ;
89      Sleep(1000);
90      program_stop_and_wait();
91      return -1 ;
92  }else{
93      printf("\nFound %d USB Devices\n\n", gl_nof_usbdevices);
94
95      for(i = 0;i < gl_nof_usbdevices;i++){
96          printf("NameString:      %s\n", gl_sis3153_device_information_struct[i].readableName);
97          printf("idVendor:          %04X\n", gl_sis3153_device_information_struct[i].idVendor);
98          printf("idProduct:         %04X\n", gl_sis3153_device_information_struct[i].idProduct);
99          printf("idSerNo:           %d\n", gl_sis3153_device_information_struct[i].idSerNo);
100         printf("DriverVersion:      %04X\n", gl_sis3153_device_information_struct[i].idDriverVersion);
101         printf("FirmwareVersion:    %04X\n", gl_sis3153_device_information_struct[i].idFirmwareVersion);
102         printf("FpgaFirmwareVersion: %04X\n", gl_sis3153_device_information_struct[i].idFpgaFirmwareVersion);
103     }
104     printf("\n\n");
105
106     // use first device
107     gl_usb_index = 1 ;
108     printf("use USB Device : %s\n\n", gl_sis3153_device_information_struct[gl_usb_index-1].readableName);
109
110     return_code = Sis3153usb_OpenDriver (&gl_sis3153_device_information_struct[gl_usb_index-1] );
111     gl_USBDevice = gl_sis3153_device_information_struct[gl_usb_index-1].hUsb;
112     if(return_code != 0) {
113         gl_USBDevice = NULL ;
114         printf("ERROR Sis3153usb_OpenDriver \n\n") ;
115     }
```

1. FindAll_SIS3153USB_Devices(...)
2. Sis3153usb_OpenDriver

5.1.3 Read/write Cycles

This screenshot shows a VME read cycle (A32_D32, AM=9):

```
return_code = sis3153Usb_Vme_Single_Read(gl_USBDevice, addr, 0x9, 4, &data) ;    //  
if (return_code != 0) {  
    printf("sis3153Usb_Vme_Single_Read: addr = %08X    return_code = %08X \n", addr, return_code);  
    program_stop_and_wait();  
}
```

This screenshot shows a VME write cycle (A32_D32, AM=9):

```
return_code = sis3153Usb_Vme_Single_Write(gl_USBDevice, addr, 0x9, 4, 0x0) ;    //  
if (return_code != 0) {  
    printf("sis3153Usb_Vme_Single_Write: addr = %08X    data = %08X    return_code = %08X \n", addr, data, return_code);  
    program_stop_and_wait();  
}
```

5.2 sis3153usb_read_sis3305_memory_speed_test

This program executes read VME cycles (A32DMA_D32, A32_BLT32, A32_MBLT64, A32_2eVME, A32_2eSST160, A32_2eSST267, A32_2eSST320,) from the SIS3305 ADC module and calculates the data rate.

```

C:\Windows\system32\cmd.exe
Found 1 USB Devices
NameString:          SIS3153 #1
idVendor:            1657
idProduct:           3153
idSerNo:             4
DriverVersion:       0100
FirmwareVersion:     0100
FpgaFirmwareVersion: 0100

Use USB Device : SIS3153 #1
sis3153usb USB3.0 connection

sis3153usb_Register_Single_Read: addr = 00000000  data = 30000000  return_code = 00000000
sis3153usb_Register_Single_Read: addr = 00000001  data = 31530100  return_code = 00000000
sis3153usb_Register_Single_Read: addr = 00000002  data = 00000004  return_code = 00000000

Enter SIS3305 VME base address      : 0x42000000
sis3305 Ident/Version register = 3305200C

start test on address 0x42000000 over 0x00100000 1words

vme_A32DMA_D32FIFO_read rate:      15.267 MByte / sec
vme_A32BLT32FIFO_read rate:       25.478 MByte / sec
vme_A32MBLT64FIFO_read rate:      47.904 MByte / sec
vme_A32_2EUMEFIFO_read rate:      86.022 MByte / sec
vme_A32_2ESST160FIFO_read rate:   127.389 MByte / sec
vme_A32_2ESST267FIFO_read rate:   147.059 MByte / sec
vme_A32_2ESST320FIFO_read rate:   175.439 MByte / sec

loop_counter = 1
vme_A32DMA_D32FIFO_read rate:      15.267 MByte / sec
vme_A32BLT32FIFO_read rate:       25.478 MByte / sec
vme_A32MBLT64FIFO_read rate:      47.904 MByte / sec
vme_A32_2EUMEFIFO_read rate:      86.022 MByte / sec
vme_A32_2ESST160FIFO_read rate:   127.389 MByte / sec
vme_A32_2ESST267FIFO_read rate:   147.059 MByte / sec
vme_A32_2ESST320FIFO_read rate:   173.913 MByte / sec

loop_counter = 2
vme_A32DMA_D32FIFO_read rate:      15.267 MByte / sec
vme_A32BLT32FIFO_read rate:       25.478 MByte / sec
vme_A32MBLT64FIFO_read rate:      47.904 MByte / sec
vme_A32_2EUMEFIFO_read rate:      86.022 MByte / sec
vme_A32_2ESST160FIFO_read rate:   127.389 MByte / sec
vme_A32_2ESST267FIFO_read rate:   147.059 MByte / sec

```

5.3 sis3316_main_test (with sis3153usb class library)

You will find this project in the directory:

sis3153usb_win\sis3153w_1.0\sis3316_application\vc10\test

This program tests the SIS3316 ADC module:

```

C:\Windows\system32\cmd.exe

sis3153usb device open OK    <found 1 vme interface device[s1]>

idVendor:      0x1657
idProduct:     0x3153
idSerNo:       4
idDriverVersion: 0x0100
idFirmwareVersion: 0x0100
idFpgaFirmwareVersion: 0x0100

Enter  UME start address of first module      : 0x42000000

Enter  number of modules <max=16>             : 1

UME-Control FPGA:      Version and Ident register: 0x33162003 at addr = 0x42000004 return_code = 0x00000000
ADC FPGA 1 - Ch 1 to 4: Version register:      0x02509914 at addr = 0x42001100 return_code = 0x00000000
ADC FPGA 1 - Ch 5 to 8: Version register:      0x02509914 at addr = 0x42002100 return_code = 0x00000000
ADC FPGA 1 - Ch 9 to 12: Version register:      0x02509914 at addr = 0x42003100 return_code = 0x00000000
ADC FPGA 1 - Ch 13 to 16: Version register:      0x02509914 at addr = 0x42004100 return_code = 0x00000000

*****
*      Status of all modules are OK      *
*****

Start sis3316 Register Test with module 0x42000000 .....
Start sis3316 Memory Test with module 0x42000000 .....
Increment:
  group 1  W-blt32 R-blt32-C R-mblt64-C R-2eUME-C R-2eSST160-C R-2eSST267-C R-2eSST320-C
  group 2  W-blt32 R-blt32-C R-mblt64-C R-2eUME-C R-2eSST160-C R-2eSST267-C R-2eSST320-C
  group 3  W-blt32 R-blt32-C R-mblt64-C R-2eUME-C R-2eSST160-C R-2eSST267-C R-2eSST320-C
  group 4  W-blt32 R-blt32-C R-mblt64-C R-2eUME-C R-2eSST160-C R-2eSST267-C R-2eSST320-C
Test Pattern:
  group 1  W-blt32 R-blt32-C R-mblt64-C R-2eUME-C R-2eSST160-C R-2eSST267-C R-2eSST320-C
  group 2  W-blt32 R-blt32-C R-mblt64-C R-2eUME-C R-2eSST160-C R-2eSST267-C R-2eSST320-C
  group 3  W-blt32 R-blt32-C R-mblt64-C R-2eUME-C R-2eSST160-C R-2eSST267-C R-2eSST320-C
  group 4  W-blt32 R-blt32-C R-mblt64-C R-2eUME-C R-2eSST160-C R-2eSST267-C R-2eSST320-C
Random Pattern:
  group 1  W-blt32 R-blt32-C R-mblt64-C R-2eUME-C R-2eSST160-C R-2eSST267-C R-2eSST320-C
  group 2  W-blt32 R-blt32-C R-mblt64-C R-2eUME-C R-2eSST160-C R-2eSST267-C R-2eSST320-C
  group 3  W-blt32 R-blt32-C R-mblt64-C R-2eUME-C R-2eSST160-C R-2eSST267-C R-2eSST320-C
  group 4  W-blt32 R-blt32-C R-mblt64-C R-2eUME-C R-2eSST160-C R-2eSST267-C R-2eSST320-C
Increment/Test Pattern:
  group 1  W-blt32 R-blt32-C R-mblt64-C R-2eUME-C R-2eSST160-C R-2eSST267-C R-2eSST320-C
  group 2  W-blt32 R-blt32-C R-mblt64-C R-2eUME-C R-2eSST160-C R-2eSST267-C R-2eSST320-C
  group 3  W-blt32 R-blt32-C R-mblt64-C R-2eUME-C R-2eSST160-C R-2eSST267-C R-2eSST320-C
  group 4  W-blt32 R-blt32-C R-mblt64-C R-2eUME-C R-2eSST160-C R-2eSST267-C R-2eSST320-C

loop_counter = 1    error_loop_counter = 0    register_error = 0    memory_error = 0    link_error = 0

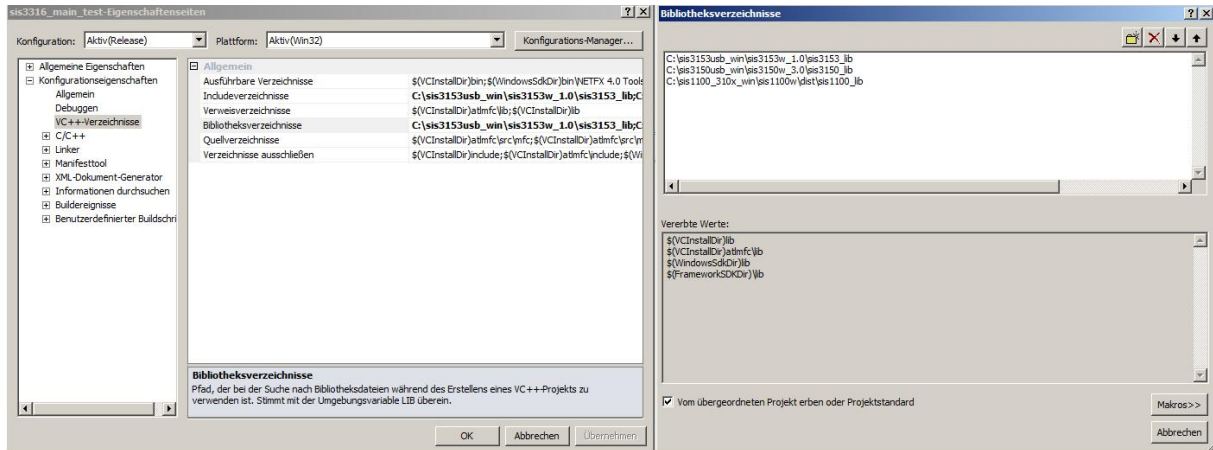
vme_fpga_link_error_counter_array[3] downto 0] = 0    0    0    0
adc_fpga_link_error_counter_array[3] downto 0] = 0    0    0    0

Start sis3316 Register Test with module 0x42000000 .....
Start sis3316 Memory Test with module 0x42000000 .....
Increment:
  group 1  W-blt32 R-blt32-C R-mblt64-C R-2eUME-C R-2eSST160-C R-2eSST267-C R-2eSST320-C
  group 2  W-blt32 R-blt32-C R-mblt64-C R-2eUME-

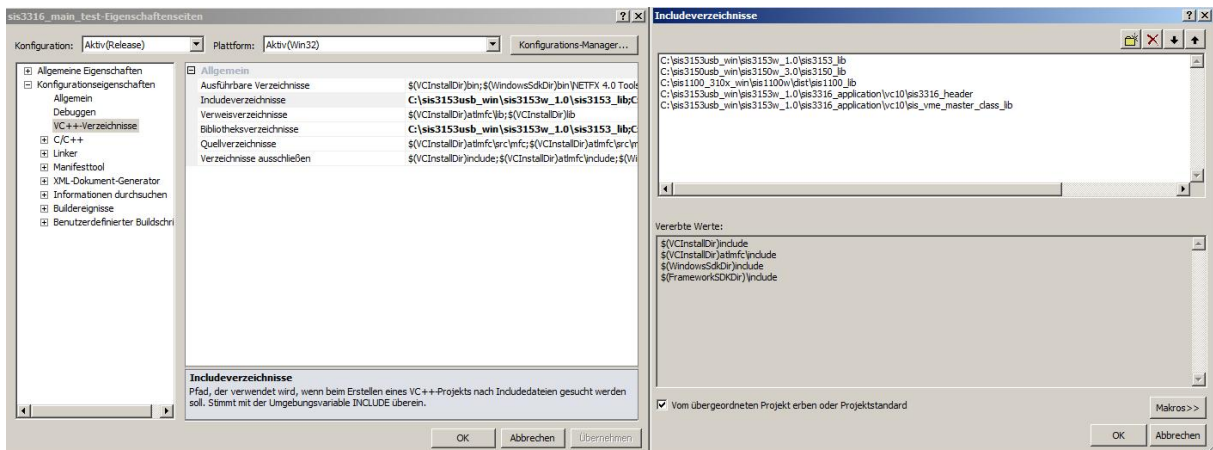
```

5.3.1 VC project settings (with sis3153usb class library)

This screenshot shows the setting of the library path:



This screenshot shows the setting of the include path:



5.3.2 Open driver (with sis3153usb class library)

This screenshot shows how to open one of the SIS VME interfaces with the call “vme_crate->vmeopen” from the user code:

```
// open Vme Interface device
return_code = vme_crate->vmeopen (); // open Vme interface
vme_crate->get_vmeopen_messages (char_messages, &nof_found_devices); // open Vme interface
printf("\n%s (found %d vme interface device[s])\n\n", char_messages, nof_found_devices);
```

A #define in the user code (sis3316_main_test.cpp) defines which SIS VME interface driver will be linked to the

“vme_crate->vmeopen” (see “vme_interface_class.h” in directory:\sis_vme_master_class_lib).

If the #define is set to “#define USB3_VME_INTERFACE” the following “vme_crate->vmeopen” will be executed:

```
int sis3153::vmeopen ( void )
{
    int status;
    // struct SIS3153USB_Device_Struct info[MAX_SIS3153USB_DEVICES];
    UINT data;
    UINT found=0;

    status = FindAll_SIS3153USB_Devices(device_info, &found, MAX_SIS3153USB_DEVICES);
    this->nof_devices_found = found ;
    if((found==0) || (found < (this->used_device_no + 1))){
        this->sis3153_device = NULL ;
        if(found==0) {
            strcpy (this->char_messages , "sis3153usb device open failed: no sis3153usb device found");
        }
        else {
            strcpy (this->char_messages , "sis3153usb device open failed: selected sis3153usb device not found");
        }
        //cout << "FindAll_SIS3153USB_Devices no" << found << endl;
        return -1;
    }
    status = Sis3153usb_OpenDriver( &device_info[this->used_device_no]) ;
    if (status != 0) {
        strcpy (this->char_messages , "sis3153usb device open failed");
        return -1;
    }
    this->sis3153_device = device_info[this->used_device_no].hUsb ;
    //sis3153usb_Register_Single_Write ( this->sis3153_device, 0x0, /* USB Control/Status Register */ 0x1); // set Led A
    sis3153usb_Register_Single_Write ( this->sis3153_device, 0x10, /* USB VME Master Control/Status Register */ 0xC000); // set VME System Controller Berr to 100us

    this->idVendor = this->device_info[this->used_device_no].idVendor ;
    this->idProduct = this->device_info[this->used_device_no].idProduct ;
    this->idSerNo = this->device_info[this->used_device_no].idSerNo ;
    this->idDriverVersion = this->device_info[this->used_device_no].idDriverVersion;
    this->idFirmwareVersion = this->device_info[this->used_device_no].idFirmwareVersion;
    this->idPpgaFirmwareVersion = this->device_info[this->used_device_no].idPpgaFirmwareVersion;

    strcpy (this->char_messages , "sis3153usb device open OK");
    return 0;
}
```

5.3.3 Read/write Cycles (with sis3153usb class library)

VME write cycle (A32_D32, AM=9):

```
addr = module_base_addr + SIS3316_MODID ;  
return_code = vme_crate->vme_A32D32_read ( addr, &data);
```

VME read cycle (A32_D32, AM=9):

```
addr = module_base_addr + SIS3316_MODID ;  
return_code = vme_crate->vme_A32D32_read ( addr, &data);
```

VME read cycle (A32_2eSST320):

```
return_code = vme_crate->vme_A32_2ESST320FIFO_read (module_addr + SIS3316_FPGA_ADC1_MEM_BASE+(i_adc*SIS3316_FPGA_ADC_MEM_OFFSET), gl_rblt_data, nof_lwords, &read_nof_words );  
if((return_code != 0) || (nof_lwords != read_nof_words)){  
    printf("\nError vme_A32_2ESST320FIFO_read : return_code = 0x%08x req_nof_lwords = 0x%08 read_nof_words = 0x%08x\n",return_code, nof_lwords, read_nof_words);  
    program_stop_and_wait();  
}
```


6 Index

| | | | |
|---------------------------------|----|----------------------------------|----|
| AM..... | 17 | RTE..... | 13 |
| closedriver..... | 15 | sample_test.vi..... | 21 |
| Connection Type..... | 19 | sampling test..... | 21 |
| CVI..... | 13 | SIS3153 USB base..... | 13 |
| default values..... | 21 | Toggle USER LED..... | 20 |
| DLL..... | 14 | trigger..... | 22 |
| FindAll_SIS3150USB_Devices..... | 15 | external..... | 22 |
| got_num_words..... | 18 | software..... | 22 |
| handle..... | 15 | udp_adc_read.vi..... | 22 |
| handle.gbl..... | 15 | Update driver (Win7)..... | 10 |
| Labview..... | 14 | vme_A32D32_read..... | 17 |
| mode..... | | vme_A32D32_write..... | 17 |
| FIFO..... | 18 | vme_dma_read..... | 18 |
| open_device..... | 15 | vme_dma_write..... | 18 |
| register_dma_read..... | 16 | vme_read..... | 17 |
| register_dma_write..... | 16 | vme_write..... | 17 |
| register_sgl_read..... | 16 | Windows Driver Installation..... | 4 |
| register_sgl_write..... | 16 | | |
| req_num_data..... | 18 | | |