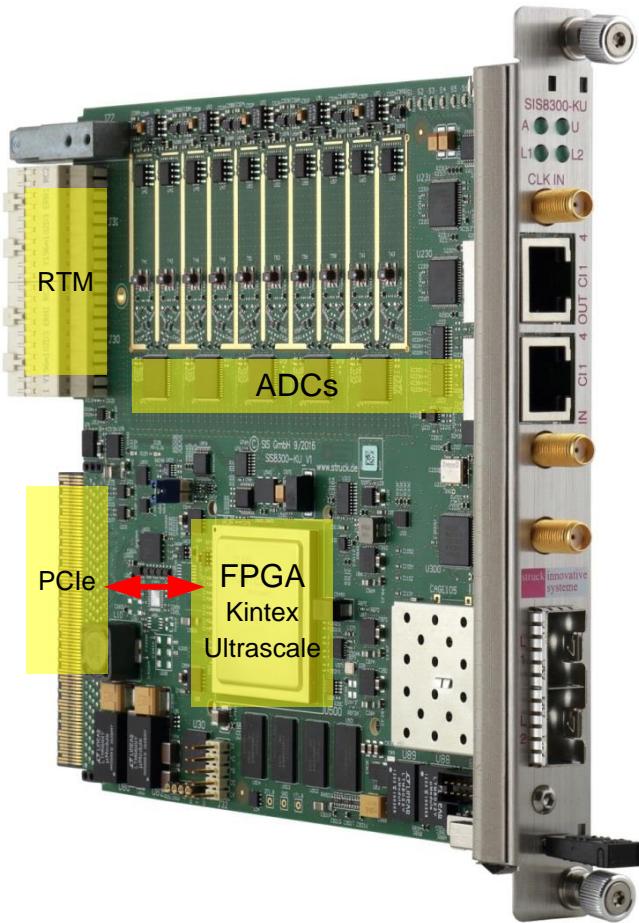




How to talk to the hardware from the OS illustrated with Struck Digitizer AMC/RTM Combination



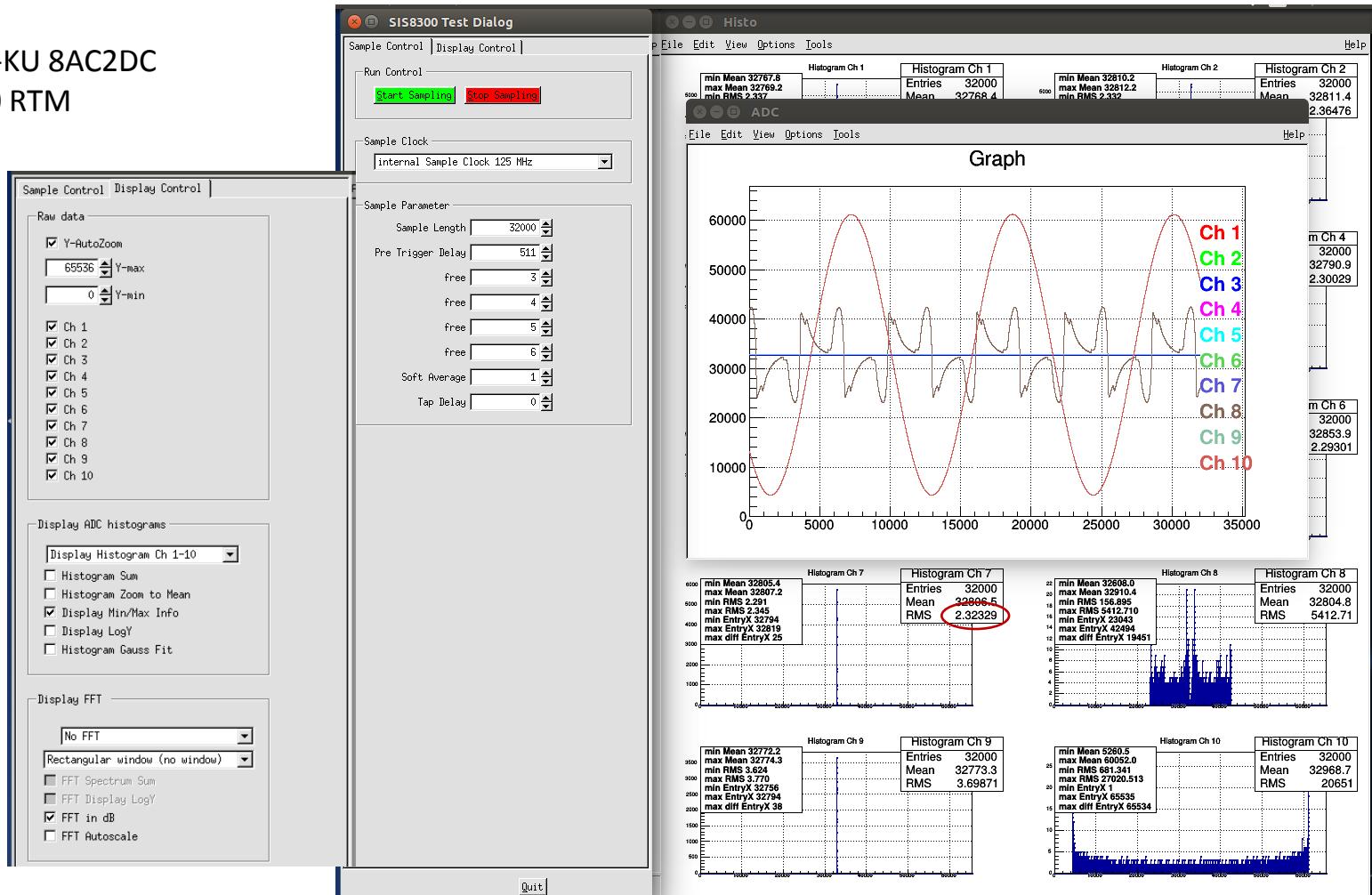
MTCA.4 Crate with a CPU, NAT-MCH and AMCs (SIS8300KU)



Our goal is to build an application to get and show ADC data

SIS8300-x ROOT GUI

Use SIS8300-KU 8AC2DC
with SIS8900 RTM



RMS for open AC/DC channel

PCI Express is used as Communication/Transfer bus

Applications:

register_read

..

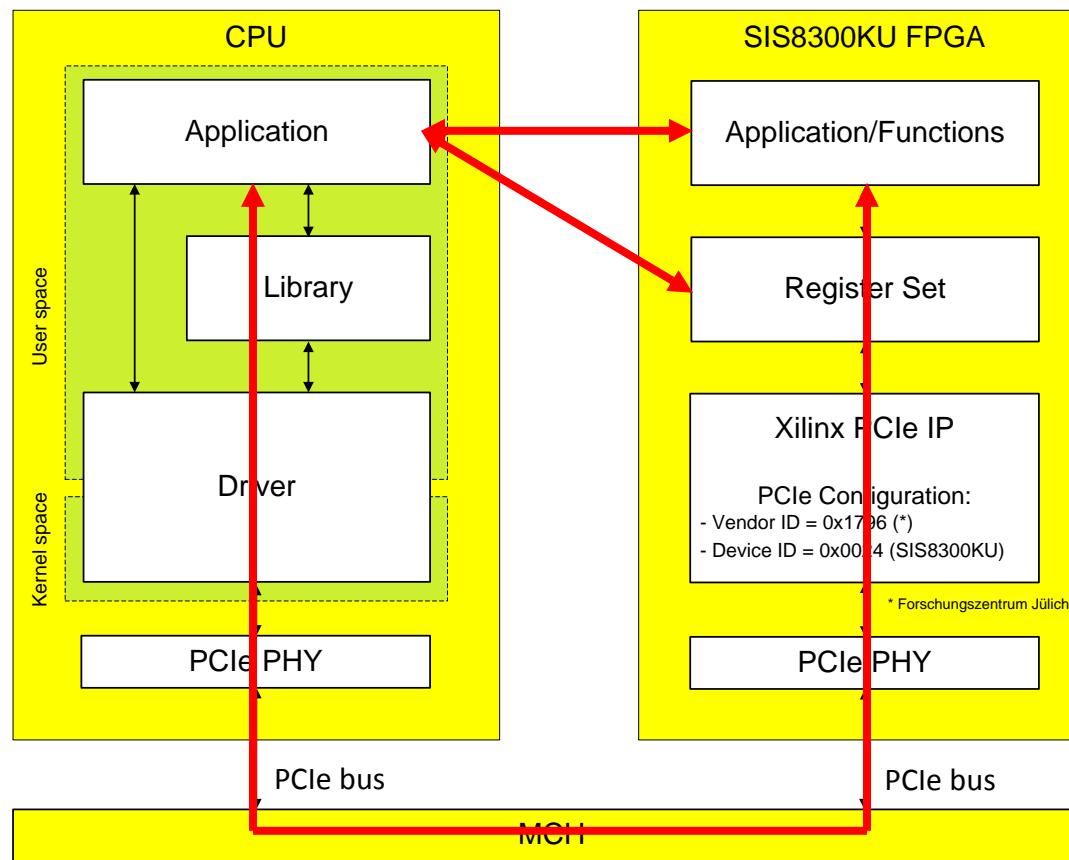
..

SIS8300-x ROOT GUI

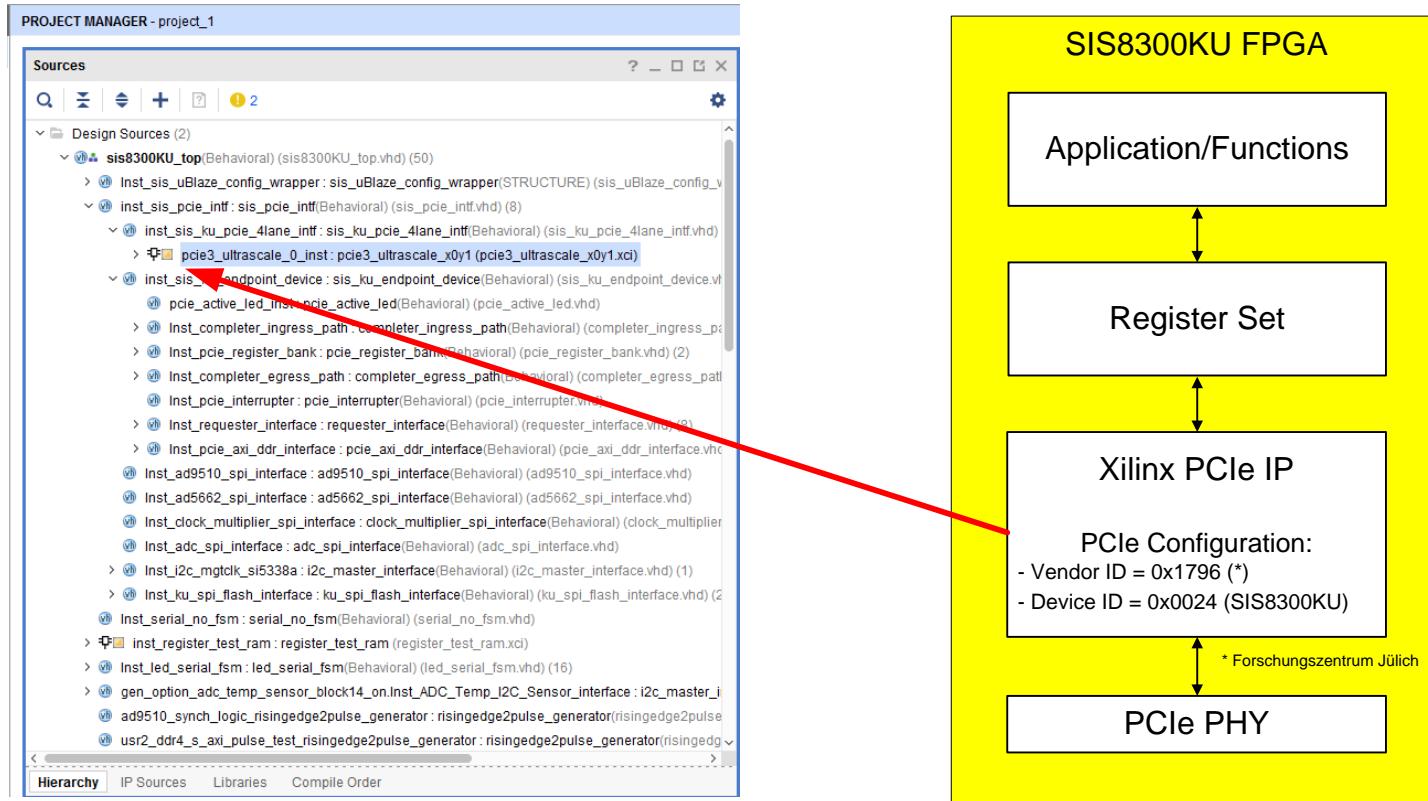
EPICS,

„Software side“

„Hardware side“



Struck SIS8300KU AXI-Based Xilinx Vivado FPGA Framework



SIS8300KU FPGA PCI Express Configuration

Re-customize IP

UltraScale FPGA Gen3 Integrated Block for PCI Express (4.4)

Documentation IP Location Switch to Defaults

Show disabled ports

Component Name: pcie3_ultrascale_x0y1

PF0 IDs PF0 BAR Legacy/MSI Cap MSI-X

PF0 - ID Initial Values

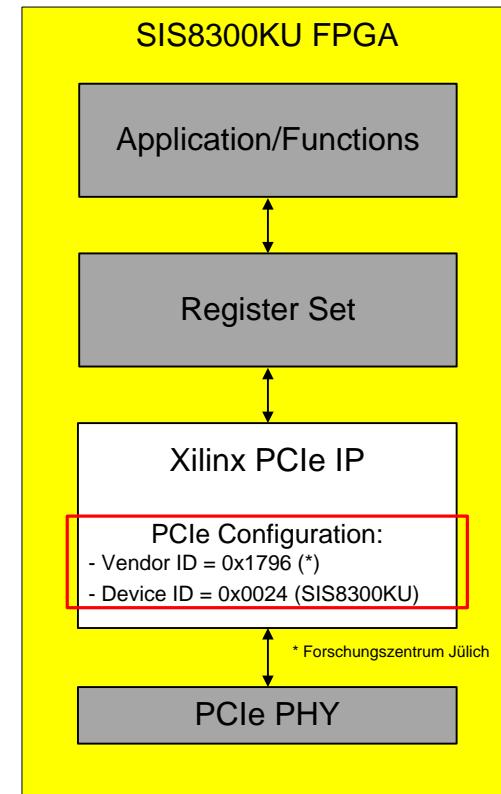
Vendor ID	1796	Range: 0000..FFFF
Device ID	24	Range: 0000..FFFF
Revision ID	00	Range: 00..FF
Subsystem Vendor ID	1796	Range: 0000..FFFF
Subsystem ID	24	Range: 0000..FFFF

Class Code

PF0 Use Class Code Lookup Assistant

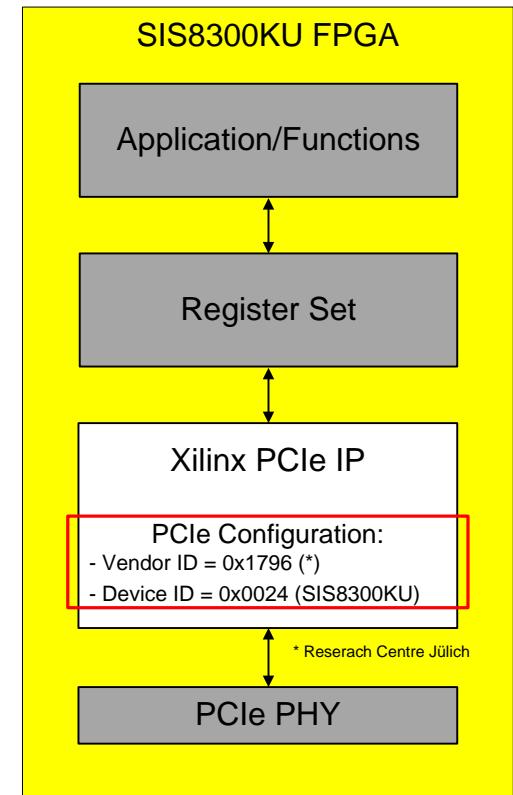
Base Class Menu	Simple communication controllers
Base Class Value	07
Sub Class Interface Menu	Generic XT compatible serial controller
Sub Class Value	00
Interface Value	00
Class Code	070000

m_axis_cq +
m_axis_rc +
pcie_7x_mngt +
pcie3_cfg_fc +
pcie3_cfg_msg_tx +
pcie3_cfg_mngt +
pcie3_per_func_status +
pcie3_cfg_status +
pcie3_transmit_fc +
pcie3_us_int_shared_logic +
user_clk -
user_reset -
user_link_up -
phy_rdy_out -



Check with „lspci“ if the OS has detected the SIS Devices

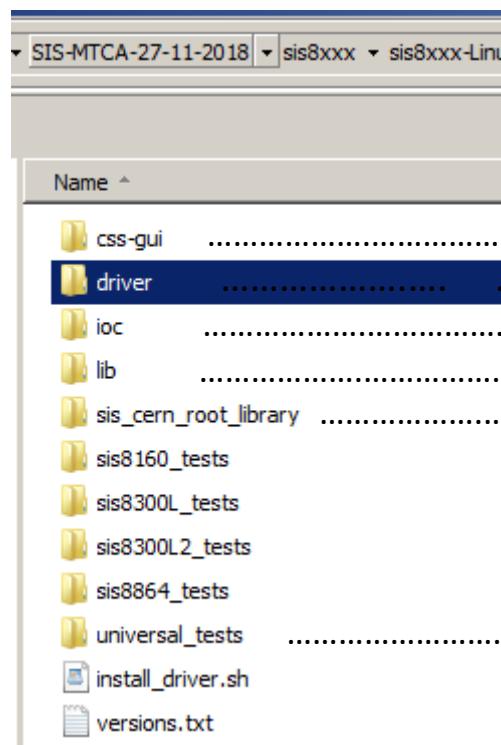
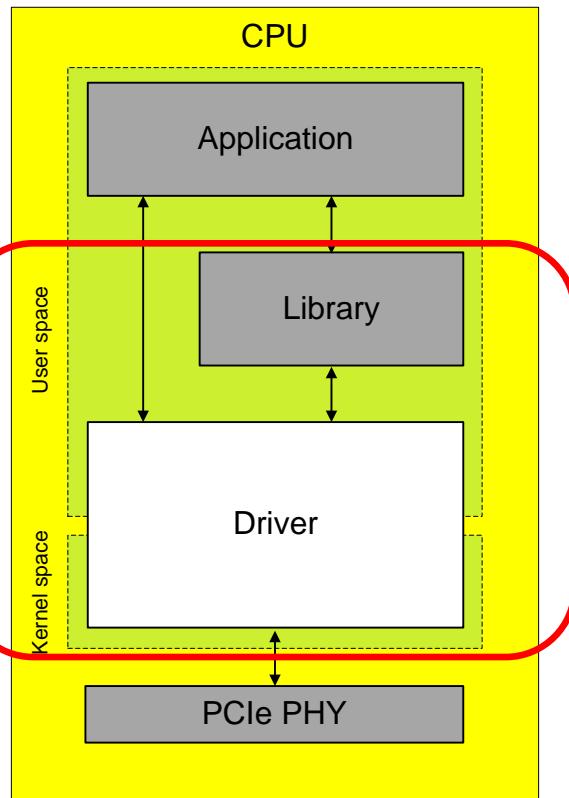
```
Activities Terminal ▾
File Edit View Search Terminal Help
th@th-CELSIUS-M740: ~/SIS/testroom/driver
th@th-CELSIUS-M740:~$ 
th@th-CELSIUS-M740:~$ 
th@th-CELSIUS-M740:~$ 
th@th-CELSIUS-M740:~$ 
th@th-CELSIUS-M740:~$ lspci | grep Jue
09:00.0 Unassigned class [ff00]: Research Centre Juelich Device 0019
0e:00.0 Serial controller: Research Centre Juelich Device 0024
th@th-CELSIUS-M740:~$
```



You will find the SIS8xxx driver on the DVD

„Software side“

API: Application Interface



- adc_test_root
- flash
- memory_test
- reg_led_test
- register
- rtm_i2c_test
- sis_root_gui_V2.3

Why we need a device driver like the SIS830x driver

A device driver enables the Access from a Userspace Application to a Hardware Device (Kernel space)

The driver provides:

Address-Mapped I/O (ioctl)

- the SIS8300KU „PCI bar 0 address space“ is mapped into the „processor address space“
- enables processor reads/writes from/to the SIS8300KU Register Set with the system call „ioctl“

Stream-Oriented I/O (DMA)

- are handled/triggered by writing to the „DMA“ and „IRQ“ register set
- enables data transfer from/to the User-Memory (PC) to/from the AMC (SIS8300KU) Memory

Interrupts

- are handled by writing to „IRQ“ register set

Install the SIS830x driver

```
File Edit View Search Terminal Help
th@th-CELSIUS-M740:~$ cd SIS/testroom/driver/
th@th-CELSIUS-M740:~/SIS/testroom/driver$ ls
dkms.conf      sis8300.h      sis8300_irq.h      sis8300_read.c    sis8300_write.h      sis8xxx_reg.h
Makefile        sis8300_ioctl.c  sis8300_list.c   sis8300_read.h    sis8325_reg.h      udev-rules
sis8300.c       sis8300_ioctl.h  sis8300_list.h  sis8300_reg.h    sis8800_ioctl_calls.h
sis8300_defs.h  sis8300_irq.c  sis8300_o.ur-safe sis8300_write.c  sis8800_reg.h
th@th-CELSIUS-M740:~/SIS/testroom/driver$ make
make -C /lib/modules/4.15.0-39-generic/build SUBDIRS=/home/th/SIS/testroom/driver modules
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-39-generic'
  CC [M] /home/th/SIS/testroom/driver/sis8300_ioctl.o
  CC [M] /home/th/SIS/testroom/driver/sis8300.o
  CC [M] /home/th/SIS/testroom/driver/sis8300_read.o
  CC [M] /home/th/SIS/testroom/driver/sis8300_irq.o
  CC [M] /home/th/SIS/testroom/driver/sis8300_list.o
  CC [M] /home/th/SIS/testroom/driver/sis8300_write.o
  LD [M] /home/th/SIS/testroom/driver/sis8300drv.o
Building modules, stage 2.
MODPOST 1 modules
  CC      /home/th/SIS/testroom/driver/sis8300drv.mod.o
  LD [M] /home/th/SIS/testroom/driver/sis8300drv.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-39-generic'
th@th-CELSIUS-M740:~/SIS/testroom/driver$ sudo make install
[sudo] password for th:
/sbin/rmmod sis8300drv
make -C /lib/modules/4.15.0-39-generic/build SUBDIRS=/home/th/SIS/testroom/driver modules_install
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-39-generic'
  INSTALL /home/th/SIS/testroom/driver/sis8300drv.ko
At main.c:160:
- SSL error:02001002:system library:fopen:No such file or directory: ../crypto/bio/bss_file.c:74
- SSL error:2006080:BIO routines:BIO_new_file:no such file: ../crypto/bio/bss_file.c:81
sign-file: certs/signing_key.pem: No such file or directory
  DEPMOD 4.15.0-39-generic
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-39-generic'
/sbin/depmod -a
/sbin/modprobe sis8300drv
cp -f udev-rules/92-struck.rules /etc/udev/rules.d/92-struck.rules
th@th-CELSIUS-M740:~/SIS/testroom/driver$ ls
dkms.conf      sis8300_defs.h  sis8300.h      sis8300_irq.h      sis8300_o.ur-safe sis8300_write.c    sis8800_ioctl_c
Makefile        sis8300drv.ko   sis8300_ioctl.c  sis8300_irq.o     sis8300_read.c    sis8300_write.h    sis8xxx_reg.h
modules.order   sis8300drv.mod.o sis8300_list.c  sis8300_list.h   sis8300_read.h    sis8300_write.o    udev-rules
Module.symvers  sis8300drv.mod.o sis8300_ioctl.o sis8300_list.h   sis8300_read.h    sis8300_write.o
sis8300.c       sis8300drv.o   sis8300_irq.c  sis8300_list.o  sis8300_read.o    sis8325_reg.h
th@th-CELSIUS-M740:~/SIS/testroom/driver$
```

1.

Copy the directory „sis8xxx-Linux“ from the DVD to a local directory on the PC

2.

Go to the directory „driver“

3.

Execute „make“

4.

Execute „sudo make install“ (as root)

5.

The kernel object „sis8300drv.ko“ is generated

You can check with „dmesg“ if the driver is loaded

```
th@th-CELSIUS-M740: ~/SIS/testroom/driver$  
th@th-CELSIUS-M740: ~/SIS/testroom/driver$  
th@th-CELSIUS-M740: ~/SIS/testroom/driver$ dmesg | grep sis8300  
[ 513.414399] sis8300: driver unloaded  
[ 513.913174] sis8300: Driver Version: v1.13 (c)SIS GmbH 2018-11-28  
[ 513.913177] sis8300: registering for devices:  
[ 513.913179] sis8300: Vendor/Device: 1796/0018  
[ 513.913181] sis8300: Vendor/Device: 1796/0019  
[ 513.913183] sis8300: Vendor/Device: 1796/0022  
[ 513.913184] sis8300: Vendor/Device: 1796/0023  
[ 513.913186] sis8300: Vendor/Device: 1796/0024  
[ 513.913187] sis8300: Vendor/Device: 1796/0028  
[ 513.913189] sis8300: Vendor/Device: 1796/0029  
[ 513.913283] sis8300-s3: device index 0  
[ 513.913414] sis8300-s3: PCIe link status: 4 lanes at 2.5Gb/s  
[ 513.913417] sis8300-s3: vendor id: 1796/device id: 0019  
[ 513.913440] sis8300-s3: bar0 mapped to:00000000ceec0a8c size:16384  
[ 513.913569] sis8300-s3: device number: major:0 minor:235  
[ 513.913585] sis8300-s3: dma sw buffer:00000000058330443 len:131072  
[ 513.913587] sis8300-s3: dma hw buffer:0000000031020000 len:131072  
[ 513.913602] sis8300-s3: dma_2 sw buffer:0000000013252112 len:131072  
[ 513.913603] sis8300-s3: dma_2 hw buffer:0000000031040000 len:131072  
[ 513.913604] sis8300-s3: irq line: 27  
[ 513.913623] sis8300-s3: module firmware/revision: 83022012  
[ 513.913624] sis8300-s3: alternative device handle: /dev/sis8300-0  
[ 513.913708] sis8300-s1: device index 1  
[ 513.913785] sis8300-s1: PCIe link status: 4 lanes at 5.0Gb/s  
[ 513.913787] sis8300-s1: vendor id: 1796/device id: 0024  
[ 513.913801] sis8300-s1: bar0 mapped to:00000000084c23d90 size:16384  
[ 513.913897] sis8300-s1: device number: major:0 minor:511  
[ 513.913912] sis8300-s1: dma sw buffer:0000000000b52a514 len:131072  
[ 513.913914] sis8300-s1: dma hw buffer:0000000031060000 len:131072  
[ 513.913928] sis8300-s1: dma_2 sw buffer:000000007bcd1985 len:131072  
[ 513.913930] sis8300-s1: dma_2 hw buffer:0000000031080000 len:131072  
[ 513.913931] sis8300-s1: module firmware/revision: 83031911  
[ 513.913936] sis8300-s1: module firmware/revision: 83031911  
[ 513.913937] sis8300-s1: alternative device handle: /dev/sis8300-1  
th@th-CELSIUS-M740: ~/SIS/testroom/driver$
```

```
th@th-CELSIUS-M740: ~/SIS/testroom/driver$  
th@th-CELSIUS-M740: ~/SIS/testroom/driver$  
th@th-CELSIUS-M740: ~/SIS/testroom/driver$  
th@th-CELSIUS-M740: ~/SIS/testroom/driver$ ls /dev/vsis*  
/dev/sis8300-0 /dev/sis8300-1 /dev/sis8300-s1 /dev/sis8300-s3  
th@th-CELSIUS-M740: ~/SIS/testroom/driver$
```

the driver generates two Device Handles/Names
for each hardware device:

1. an enumerated Name: sis8300-1
2. a PCIe Slot dependent Name: sis8300-s1

A Desy SIS8300 driver is available, also (<https://github.com/MicroTCA>)

The screenshot shows the GitHub organization page for **MicroTCA**. At the top, there is a navigation bar with tabs for **Repositories 6**, **People 0**, and **Projects 0**. A prominent banner at the top of the page says "Grow your team on GitHub" with a "Sign up" button. Below the banner, there are search fields for "Find a repository...", "Type: All", and "Language: All".

The organization page lists four repositories:

- upciedev**: universal PCIe driver, provides the base PCIe functionality to be used by the top level drivers. Language: C, 1 file, GPL-2.0, Updated on 22 Feb.
- pciedev**: Simple PCIe driver builded on top of upciedev. Language: C++, 1 file, GPL-2.0, Updated on 22 Feb.
- x1timer**: Device driver for DESY FLASH and XFEL x1timer, x2timer, and NAT NAMC-psTimer AMC Fast Timing System with ps resolution. Language: C, 1 file, GPL-2.0, Updated on 22 Feb.
- sis8300**: device driver for STRUCK SIS8300 board based on upciedev. Language: C, 1 file, GPL-2.0, Updated on 11 Dec 2017.

On the right side of the page, there are two boxes: "Top languages" (C, C++) and "People" (0). The "People" box contains the message: "This organization has no public members. You must be a member to see who's a part of this organization."

Register Read Example with the system call „ioctl“

Application: \universal_tests\register\regread_iocall.c



```
th@th-CELSIUS-M740:~/SIS/testroom/universal_tests/register$ ./regread_iocall /dev/sis8300-s1 0x1
reg = 0x00000081
th@th-CELSIUS-M740:~/SIS/testroom/universal_tests/register$ ./regread_iocall /dev/sis8300-s1 0x0
reg = 0x83031911
```

An Extract of Registers of the SIS830x FPGA Firmware Implementation (Register Set)

Register Address Map for Identification ...

Offset	Access	Function
0x00	R	Module Identifier/Firmware Version register
0x01	R	Serial number register
0x02	R/W	reserved
0x03	R/W	reserved
0x04	R/W	User Control/Status register (JK)
0x05	R	Firmware Options register
0x06	R/W	ADC Temperature Sensor interface register
0x07	R	PCIe Status register

Register Address Map for the Application like „Sample Block Length“

0x120	R/W	ADC ch1 Memory Sample Start Block Address / Actual Block Address register
0x121	R/W	ADC ch2 Memory Sample Start Block Address / Actual Block Address register
..
0x129	R/W	ADC ch10 Memory Sample Start Block Address / Actual Block Address register
0x12A	R/W	ADC chx Sample Block Length register
0x12B	R/W	ADC chx Ringbuffer Delay register (0 to 2046)

An Extract of Registers of the SIS830x FPGA Firmware Implementation (Register Set)

Register Address Map for Stream-Oriented I/O (DMA)

0x200	R/W	DMA_READ_DST_ADR_LO32
0x201	R/W	DMA_READ_DST_ADR_HI32
0x202	R/W	DMA_READ_SRC_ADR_LO32
0x203	R/W	DMA_READ_LEN
0x204	R/W	DMA_READ_CTRL
0x205	R/W	DMA Readout Sample byte swap control
0x210	R/W	DMA_WRITE_SRC_ADR_LO32
0x211	R/W	DMA_WRITE_SRC_ADR_HI32
0x212	R/W	DMA_WRITE_DST_ADR_LO32
0x213	R/W	DMA_WRITE_LEN
0x214	R/W	DMA_WRITE_CTRL

Register Address Map for Interrupts

0x220	R/W	IRQ Enable
0x221	R	IRQ Status
0x222	W	IRQ Clear
0x223	KA	IRQ Refresh

Relating to the SHAPI – Standard Hardware API



Standard Hardware API Design Guide

Guideline for designing Hardware Access APIs for MTCA.4 Systems

MTCA_DG.1 R1.0

January 9, 2017



Open Modular
Computing Specifications



SHAPI Register Address Map

Table 3-1: Standard Device Identification and Control Registers

31	16	15	0		
				0x00	ro
				0x04	ro
				0x08	ro
				0x0C	ro
				0x10	ro
				0x14	ro
				0x18	ro
				0x1C	ro
				0x20	ro
				0x24	ro
				0x28	ro
				0x2C	rw
				0x30	rw
				0x34	ro
				0x38	ro
				0x3C	rw

The SIS8300KU IOCTL Register Access implementation looks like the SHAPI IOCTL Register Access implementation

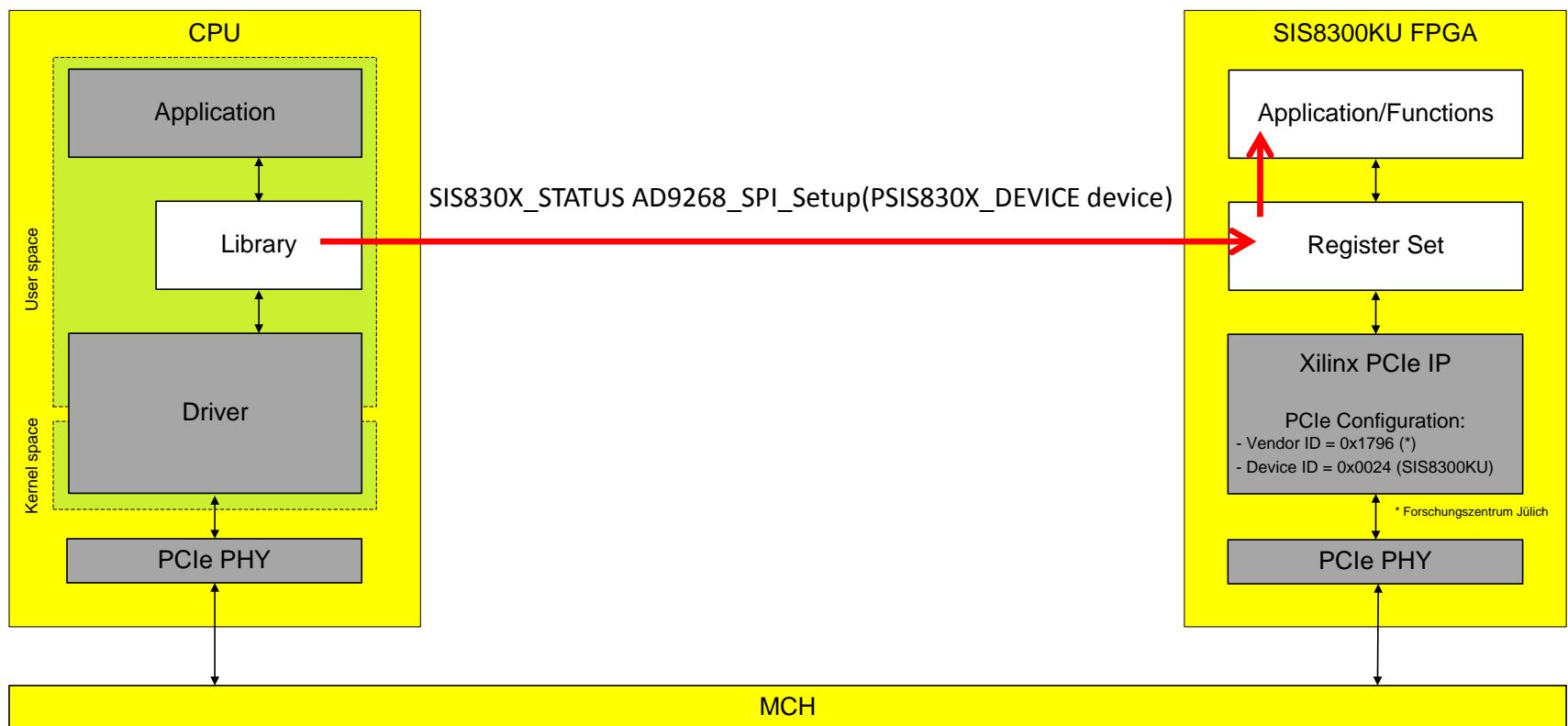
The SIS8300KU DMA/IRQ implementation is similar to the SHAPI DMA/IRQ implementation

The SIS8300KU Register Address Map is NOT identical to the SHAPI Register Address Map

SIS830x library <--> SIS830x FPGA Firmware

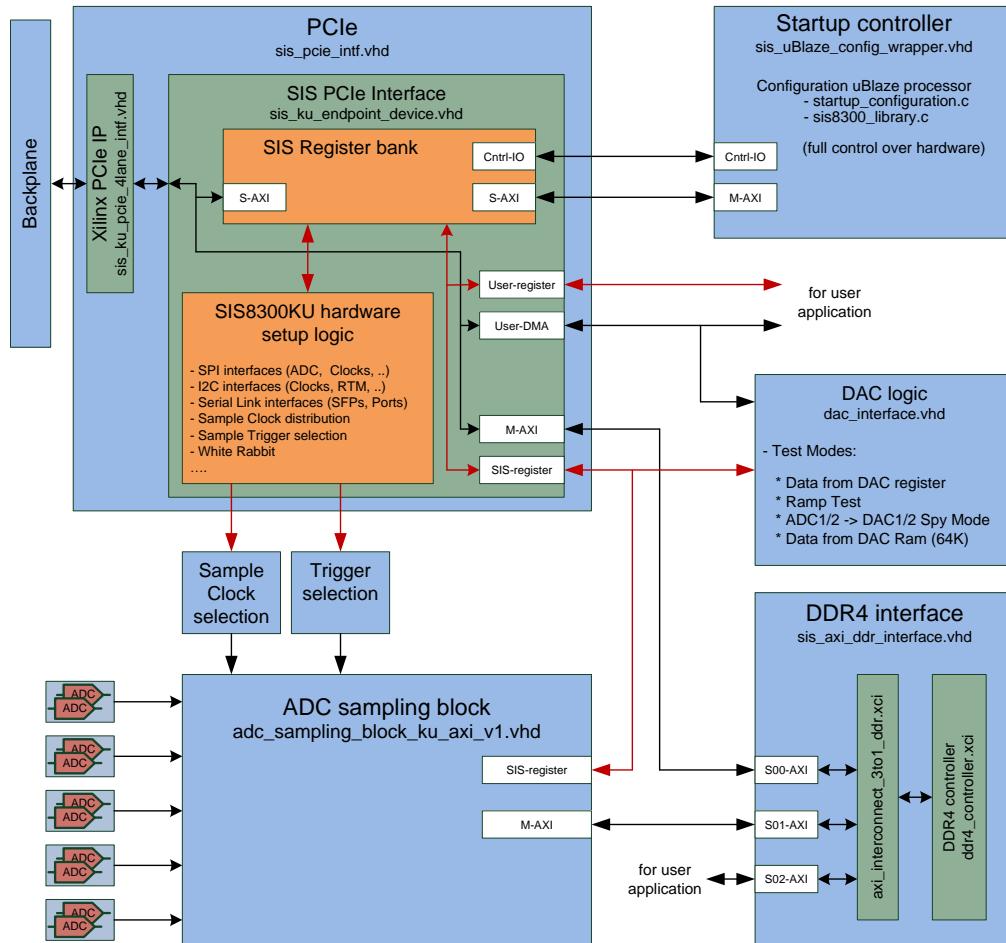
The Library contains functions/calls which correspond to the SIS8300KU FPGA Firmware Functions

a Library call contains multiple cycles to execute a “function” on the SIS8300KU
for example: setup of the ADC chips via SPI



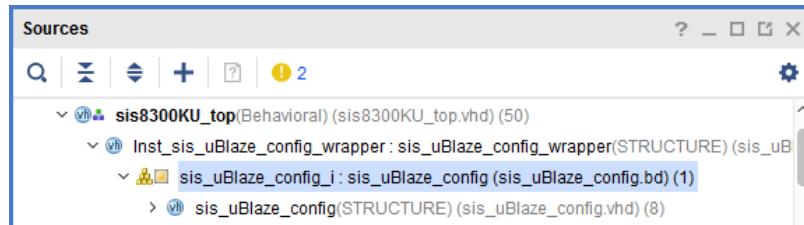
Block diagram: SIS8300KU AXI4-Based Xilinx Vivado FPGA Framework

Struck SIS8300-KU AXI-Based Xilinx FPGA Framework



Startup Controller
(initially proposed by ESS)
Allows the complete Configuration
of a SIS8300-KU Digitizer
and an attached RTM
without MTCA CPU Interaction

Vivado FPGA Framework: Startup Controller



Xilinx Software Development Kit: SDK

The screenshot shows the Xilinx SDK IDE. On the left, the file structure is displayed:

- Debug
- Release
- src
 - platform_config.h
 - platform.c
 - platform.h
 - rtm_i2c_library.c
 - rtm_i2c_library.h
 - sis_gpio_routines.c
 - sis_gpio_routines.h
 - sis8300_library.c
 - sis8300_library.h
 - sis8300_reg.h
 - startup_configuration.c
 - lscript.ld
- start_up_bsp

The code editor on the right contains C code for ADC setup:

```
/*****************************************************************************************************
 * ADC setup
 *****/
*(pcie_register_axi_ptr + SIS8300_USER_CONTROL_STATUS_REG) = 0x00001; // set user Led U
initialize_process_state = INITIALIZE_ADC_SETUP_STATE ; //
sis_write_control_reg(0x80000000 + (initialize_process_state & 0xff)); // enable pcie register access
status = 0 ;
for(i=0; i<5; i++){
    status = status | initialize_adc(pcie_register_axi_ptr, i);
}
if(status != 0) {
    initialize_process_state = INITIALIZE_ADC_SETUP_STATE + INITIALIZE_ERROR_STATE ; //
    sis_write_control_reg((initialize_process_state & 0xff)); // disable pcie register access
    #ifdef DEBUG_PRINTS
    print("error during initialize_adc\n\r");
    #endif
    do {
        loop_counter++;
    } while(1) ;
}
*(pcie_register_axi_ptr + SIS8300_USER_CONTROL_STATUS_REG) = 0x10000; // clr user Led U
```

Build the SIS830x library (libSIS830x.so)

```
th@th-CELSIUS-M740:~/SIS/testroom/lib$ ls libSIS830x/
ad9268.c      ads42lb69.c      libSIS830x.layout    si5326.c          sis830x.h      spiFlash.h
ad9268.h      ads42lb69.h      main.c            si5326.h          sis830xReg.h
ad9510.c      internal.c      Makefile          si5338aSynth.c   sis830xStat.h
ad9510.h      internal.h      obj              si5338aSynthDefault.h sis830xType.h
adc_ad9268.c   libSIS830x.cbp  rtmI2C.c        si5338aSynth.h   sis830xVer.h
adc_ad9268.h   libSIS830x.depend  rtmI2C.h       sis8300kuReg.h  spiFlash.c
th@th-CELSIUS-M740:~/SIS/testroom/lib$ make
for dir in libSIS830x/ libSIS8300/ libSIS8160/ libSIS8864/; do make -C $dir; done
make[1]: Entering directory '/home/th/SIS/testroom/lib/libSIS830x'
test -d obj/Release || mkdir -p obj/Release
gcc -Wall -fPIC -O2 -fPIC -I../../driver -c si5326.c -o obj/Release/si5326.o
gcc -Wall -fPIC -O2 -fPIC -I../../driver -c spiFlash.c -o obj/Release/spiFlash.o
```

1.
Go to the directory „lib“

2.
Execute „make“

3.
The library „libSIS830x.so“
is generated

```
th@th-CELSIUS-M740:~/SIS/testroom/lib$ ls libSIS830x/
ad9268.c      ads42lb69.c      libSIS830x.layout    rtM2C.h          sis8300kuReg.h  spiFlash.c
ad9268.h      ads42lb69.h      libSIS830x.so      si5326.c          sis830x.h      spiFlash.h
ad9510.c      internal.c      main.c            si5326.h          sis830xReg.h
ad9510.h      internal.h      Makefile          si5338aSynth.c   sis830xStat.h
adc_ad9268.c   libSIS830x.cbp  obj              si5338aSynthDefault.h sis830xType.h
adc_ad9268.h   libSIS830x.depend  rtmI2C.c       si5338aSynth.h   sis830xVer.h
th@th-CELSIUS-M740:~/SIS/testroom/lib$
```

An Extract of calls of the SIS830x Library (.../lib/ libSIS830x/sis830x.h)

```
// device handling
SIS830X_STATUS sis830x_GetNumberOfDevices(int *num);
SIS830X_STATUS sis830x_OpenDeviceOnIdx(int idx, PSIS830X_DEVICE device);

SIS830X_STATUS sis8325_GetNumberOfDevices(int *num);
SIS830X_STATUS sis8325_OpenDeviceOnIdx(int idx, PSIS830X_DEVICE device);

SIS830X_STATUS sis830x_OpenDeviceOnPath(char *path, PSIS830X_DEVICE device);
SIS830X_STATUS sis830x_CloseDevice(PSIS830X_DEVICE device);

// general purpose register i/o
SIS830X_STATUS sis830x_ReadRegister(PSIS830X_DEVICE device, uint32_t addr, uint32_t *data);
SIS830X_STATUS sis830x_WriteRegister(PSIS830X_DEVICE device, uint32_t addr, uint32_t data);

// memory i/o
SIS830X_STATUS sis830x_ReadMemory(PSIS830X_DEVICE device, uint32_t addr, uint32_t *data, uint32_t len);
SIS830X_STATUS sis830x_WriteMemory(PSIS830X_DEVICE device, uint32_t addr, uint32_t *data, uint32_t len);

// ad9510 divider setup
SIS830X_STATUS sis830x_AD9510_SPI_Setup(PSIS830X_DEVICE device, unsigned int* ch_divider_configuration_array, unsigned int ad9510_synch_cmd);

// adc ad9268 setup (SIS8300/SIS8300L/SIS8300L2/SIS8300KU)
SIS830X_STATUS sis830x_ADC_AD9268_SPI_Setup(PSIS830X_DEVICE device);
SIS830X_STATUS sis830x_ADC_AD9268_SPI_Read(PSIS830X_DEVICE device, unsigned int adc_device_no, unsigned int spi_addr, unsigned int* read_data);
```

Register Read Example with a SIS83xx Library call „sis830x_ReadRegister(...)"

Application: \universal_tests\register\regread.c

```
#include "../../lib/libSIS830x/sis830x.h"

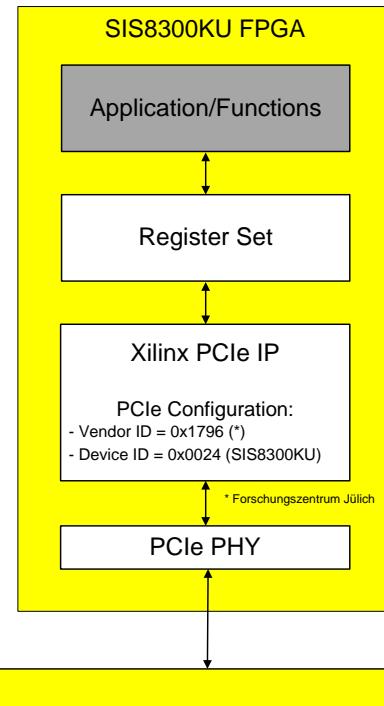
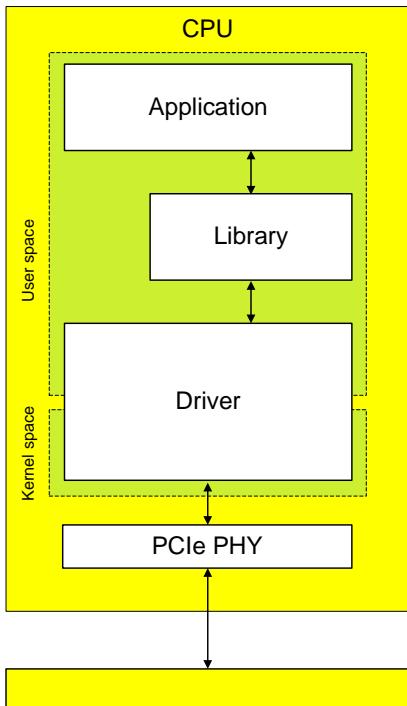
int main(int argc, char **argv){
    unsigned int offset, data;
    SIS830X_DEVICE fp;
    PSIS830X_DEVICE device;
    SIS830X_STATUS ret;
    fp.open = false;
    device = &fp;
    if(argc != 3){
        printf("usage: %s [device] [offset]\n", argv[0]);
        return -1;
    }

    ret = sis830x_OpenDeviceOnPath(argv[1], device);
    if(ret != Stat830xSuccess){
        printf("can't open device\n");
        return -1;
    }

    offset = (unsigned int)strtoul(argv[2], NULL, 16);
    ret = sis830x_ReadRegister(device, offset, &data);
    if(ret != Stat830xSuccess){
        printf("can't read register\n");
        return -1;
    }

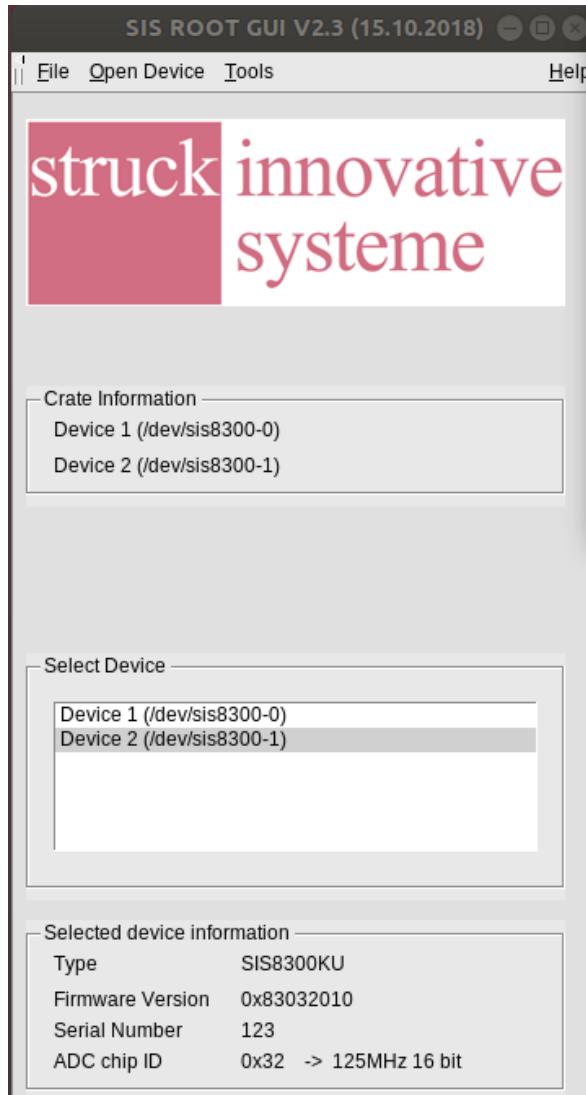
    printf("reg = 0x%08X\n", data);
    sis830x_CloseDevice(device);

    return 0;
}
```



```
th@th-CELSIUS-M740:~/SIS/testroom/universal_tests/register$ ./regread
reg = 0x83031911
th@th-CELSIUS-M740:~/SIS/testroom/universal_tests/register$ ./regread
reg = 0x00000081
/dev/sis8300-s1 0x0
/dev/sis8300-s1 0x1
```

SIS830x Root GUI Application

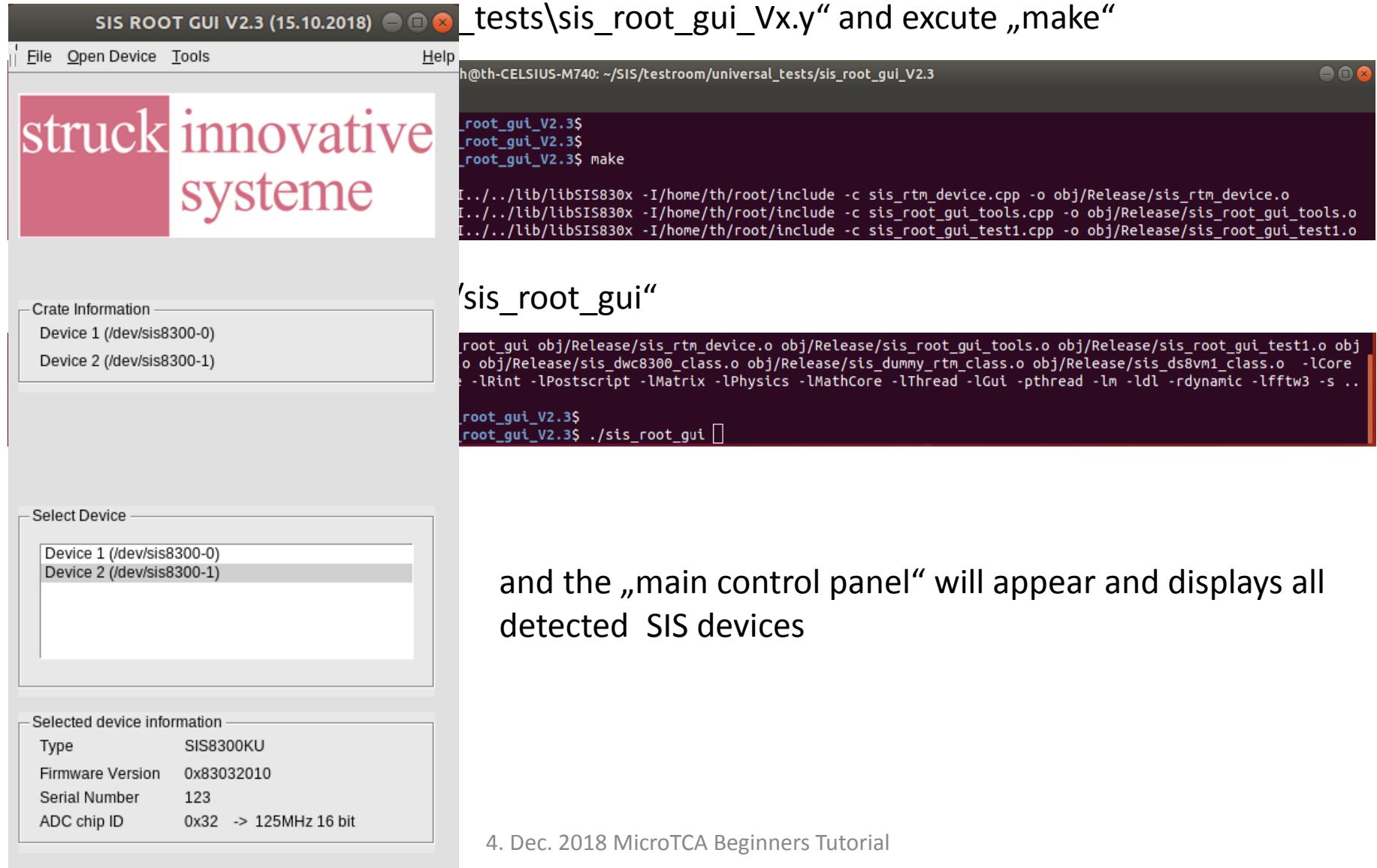


The „SIS830x Root GUI“ is a graphical application that provides access to several SIS ADC cards with different RTMs

The „SIS830x Root GUI“ application requires an installation of the CERN ROOT framework
→ <https://root.cern.ch/downloading-root>

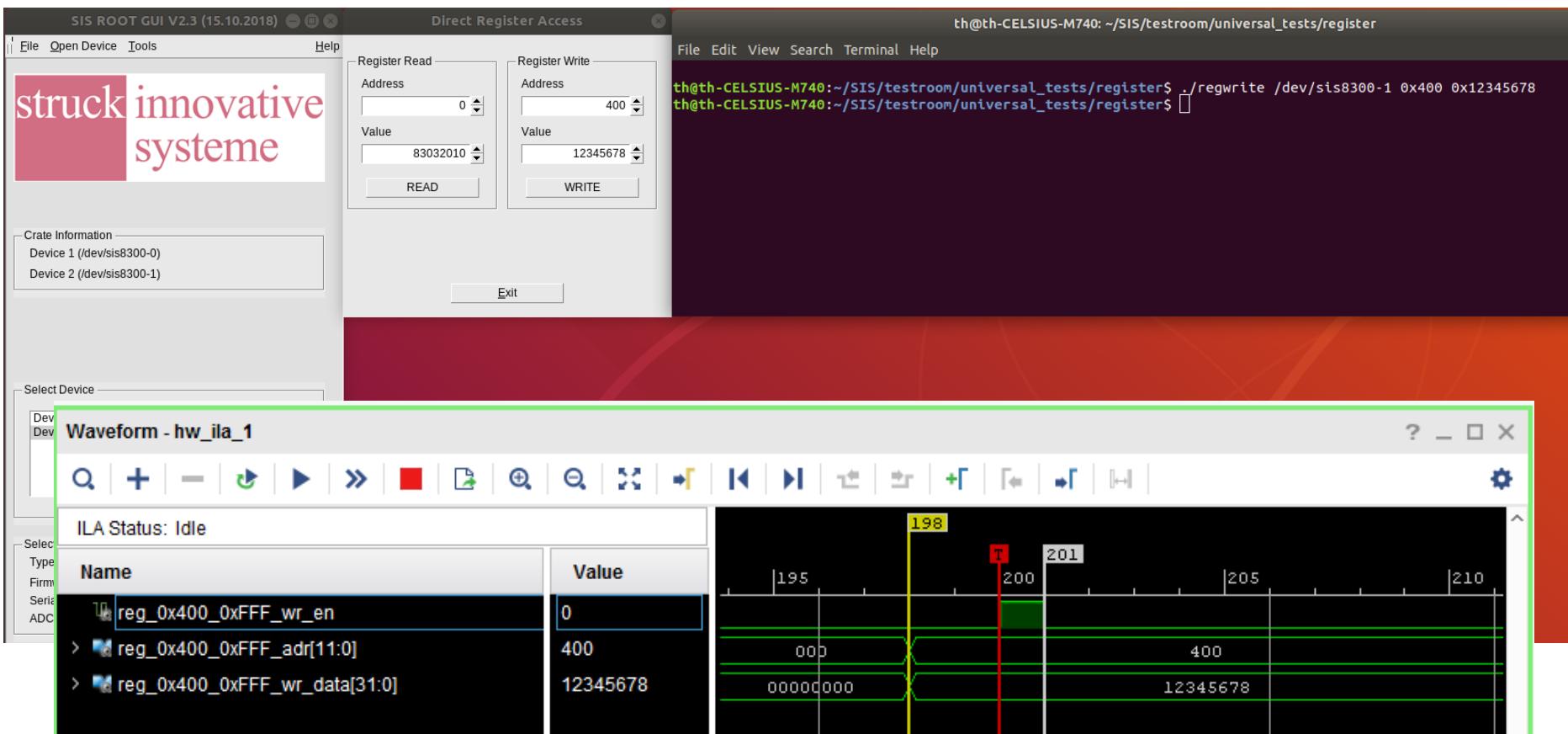
Device	Supported
SIS8300	yes
SIS8300 v2	yes
SIS8300L	yes
SIS8300L v2	yes
SIS8300L2	yes
SIS8300KU	yes
SIS8300KU-250MHz	yes
SIS8325	yes
SIS8900	yes
DWC8300	yes
DWC8VM1	yes
DS8VM1	yes

Build the SIS830x Root GUI Application



4. Dec. 2018 MicroTCA Beginners Tutorial

SIS830x Root GUI / Tools / Direct Register Access



Use Vivado chipscope to observe what happens inside the FPGA with a „Register Write“

SIS830x Root GUI / Tools / Direct Register Access

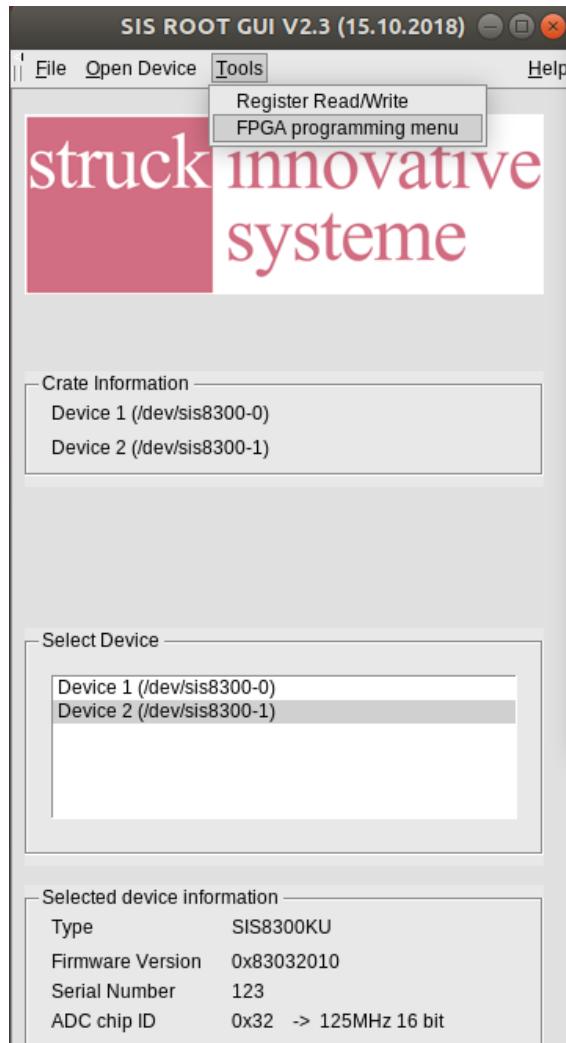
The screenshot displays three windows related to SIS830x testing:

- SIS ROOT GUI V2.3 (15.10.2018)**: A main application window with a logo for "struck innovative systeme". It shows "Device Information" for Device 1 (/dev/sis8300-0) and Device 2 (/dev/sis8300-1). A "Select Device" dropdown is also present.
- Direct Register Access**: A modal dialog with two sections: "Register Read" and "Register Write". Both sections have "Address" fields set to 400 and "Value" fields set to 12345678. The "Register Read" section has a "READ" button, and the "Register Write" section has a "WRITE" button. An "Exit" button is at the bottom.
- Terminal**: A terminal window showing command-line interactions:

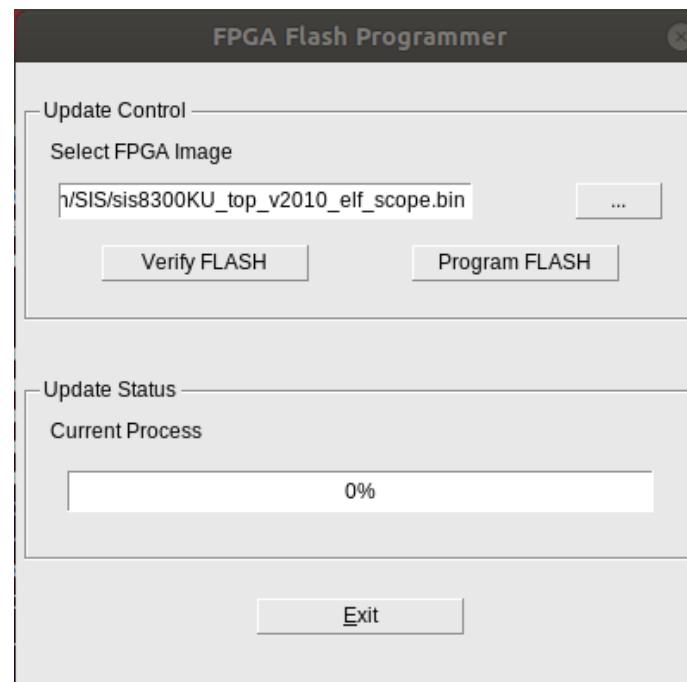
```
th@th-CELSIUS-M740:~/SIS/testroom/universal_tests/register$ ./regwrite /dev/sis8300-1 0x400 0x12345678
th@th-CELSIUS-M740:~/SIS/testroom/universal_tests/register$ ./regread /dev/sis8300-1 0x400
reg = 0x12345678
th@th-CELSIUS-M740:~/SIS/testroom/universal_tests/register$
```
- Waveform - hw_ilia_1**: A Vivado chipscope waveform viewer titled "Waveform - hw_ilia_1". It shows waveforms for several signals over time. The left pane lists signal names and their current values: reg_0x400_0xFFFF_adr[11:0] is 400, reg_0x400_0xFFFF_rd_data[31:0] is 12345678, reg_0x400_0xFFFF_rd_en is 0, inst_sis_PCIE_intf/inst_s...path/egress_fifo_wren is 0, and inst_sis_PCIE_intf/inst_s...th/egress_fifo_data[36:0] is 0012345678. The right pane shows the corresponding waveform traces with timestamp markers like 195, 200, 201, 204, 205, and 210.

Use Vivado chipscope to observe what happens inside the FPGA with a „Register Read“

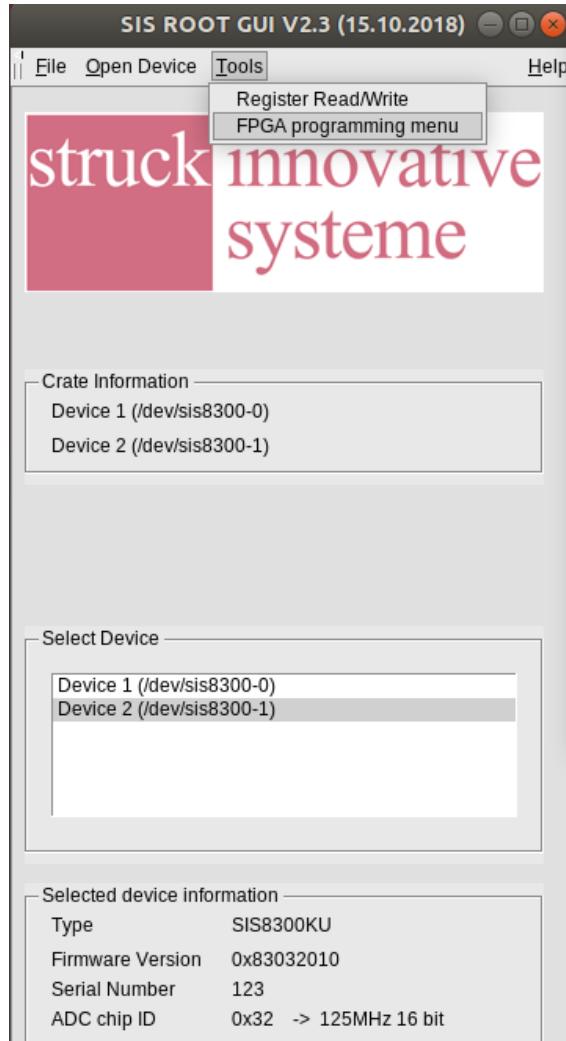
SIS830x Root GUI / Tools / FPGA programming menu



Select „Tools/FPGA programming menu“ and the „FPGA programming menu“ canvas will appear“

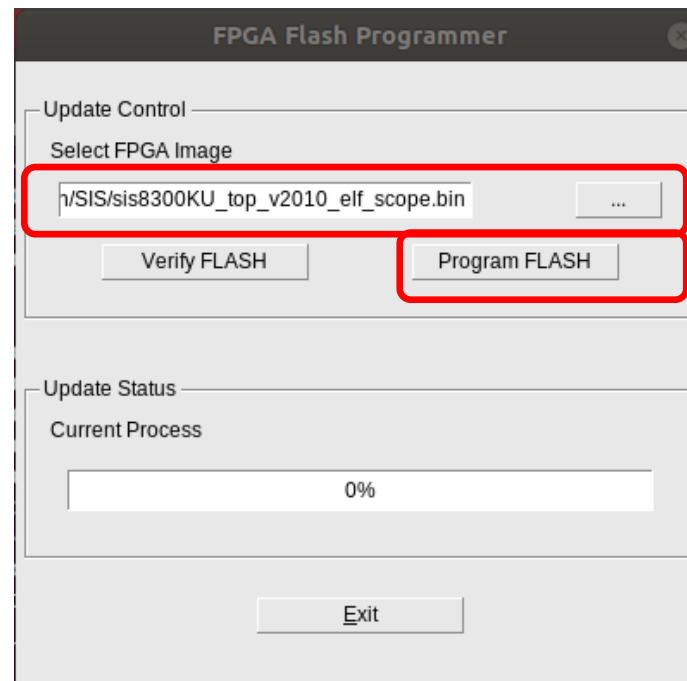


SIS830x Root GUI / Tools / FPGA programming menu

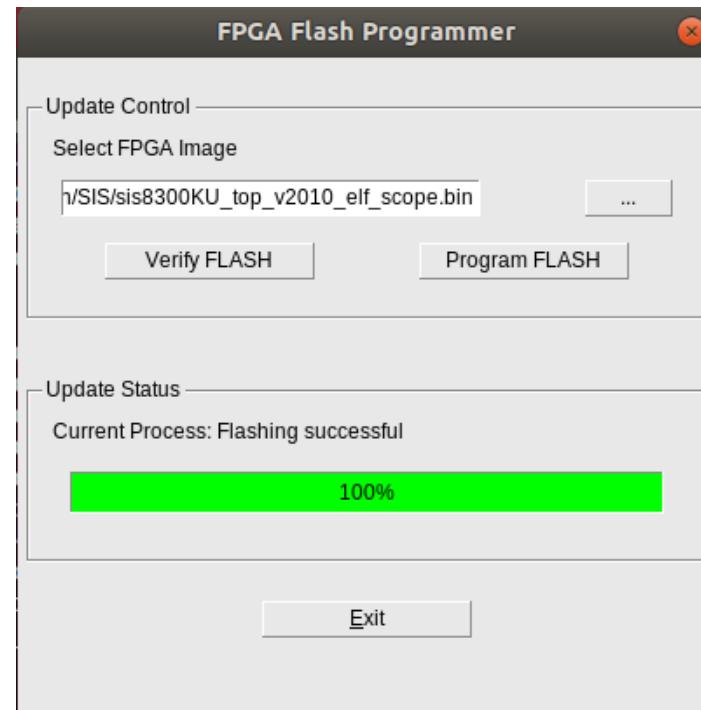
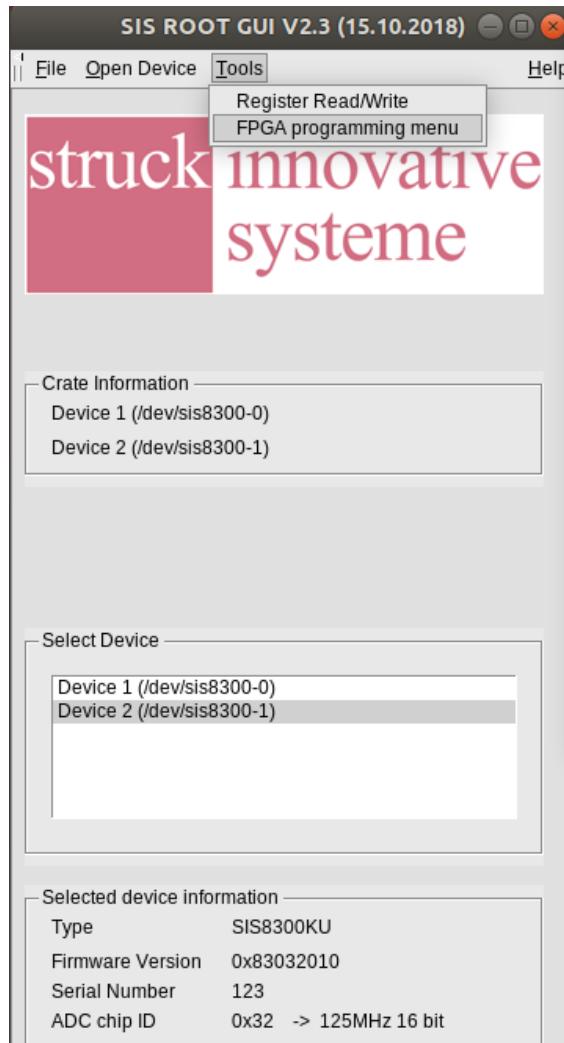


Select the „...“ button to get browser window to search for and select a FPGA image file (*.bin)

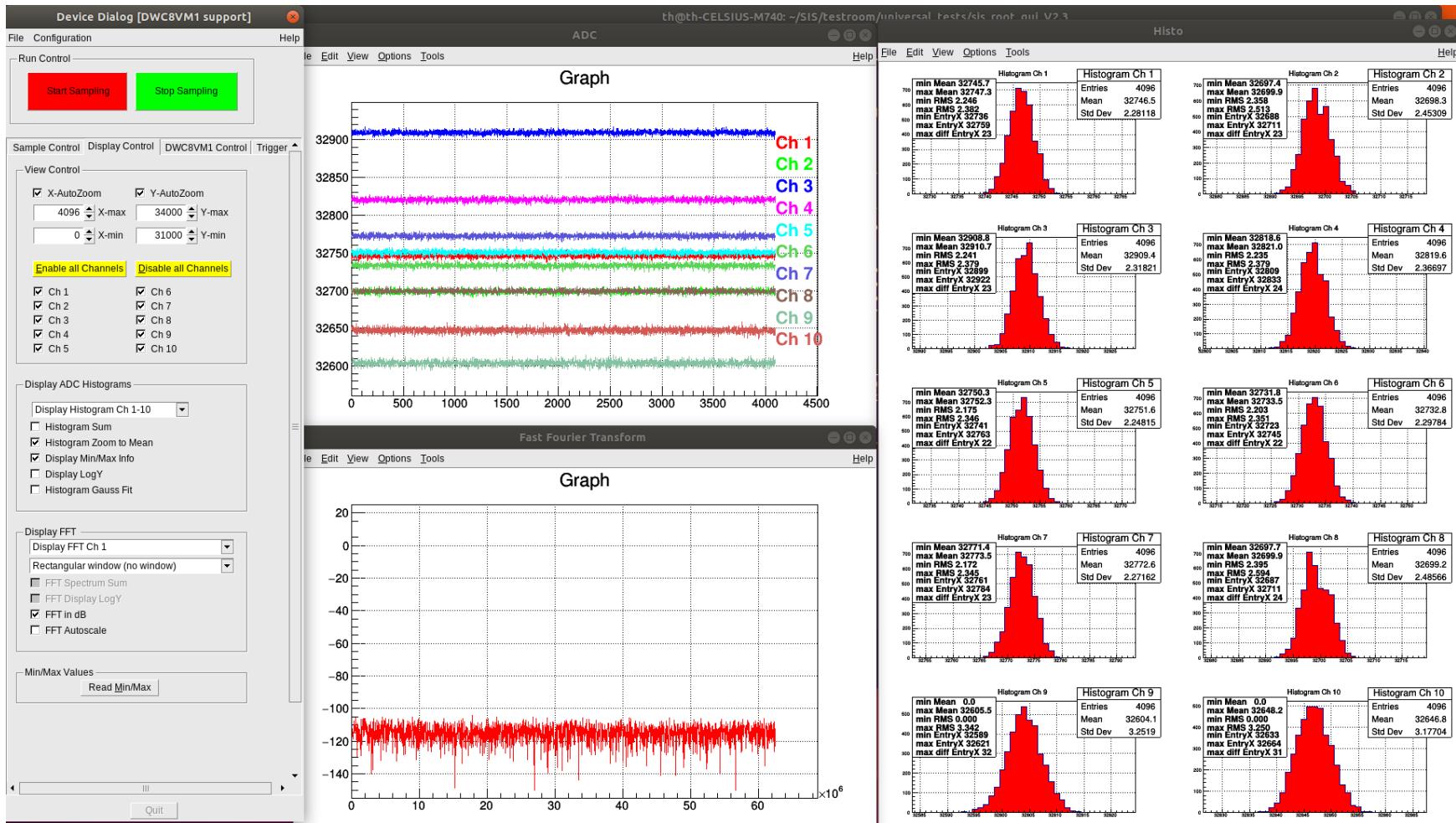
Start „Program FLASH“



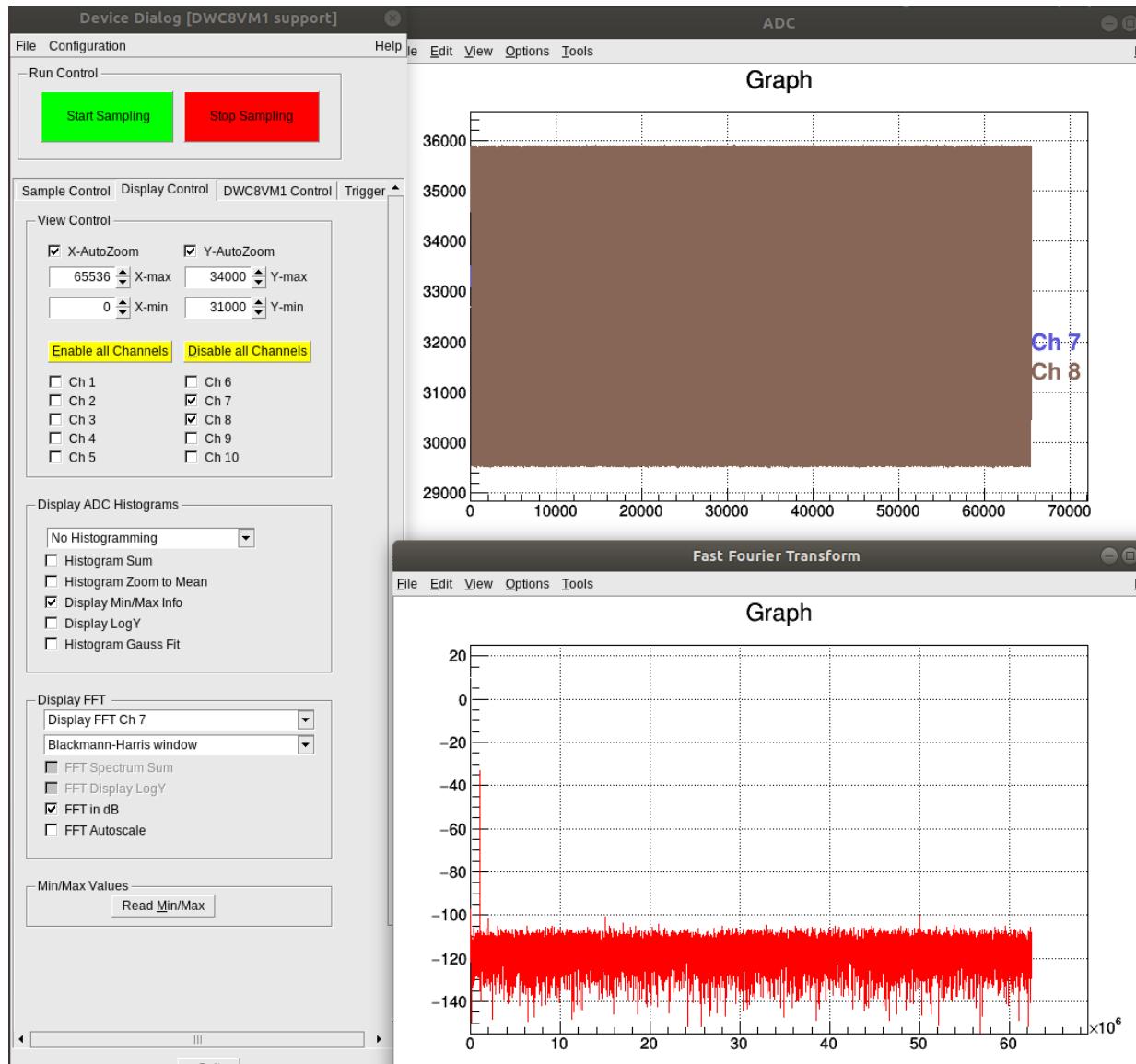
SIS830x Root GUI / Tools / FPGA programming menu



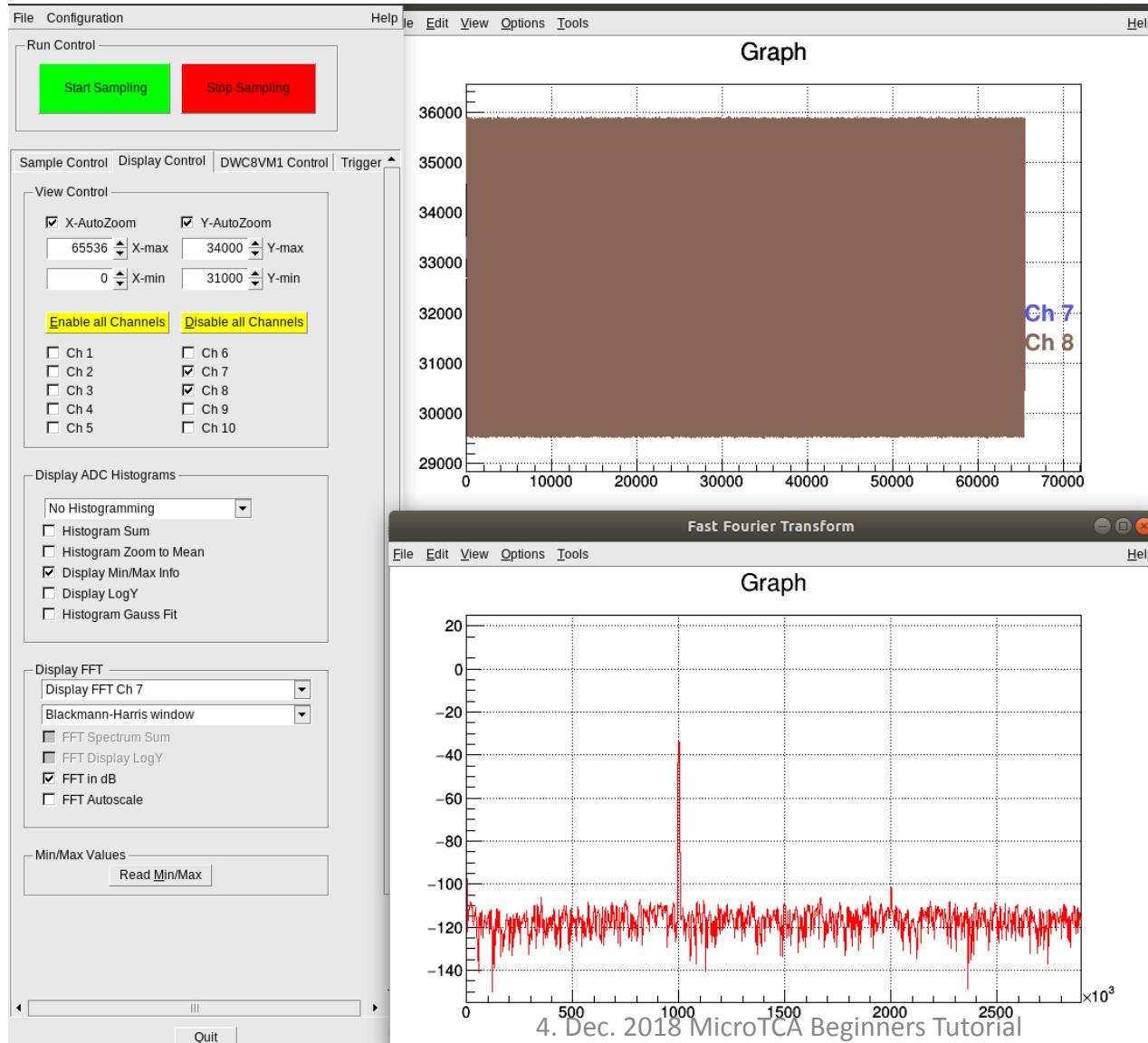
SIS830x Root GUI / Tools / FFT



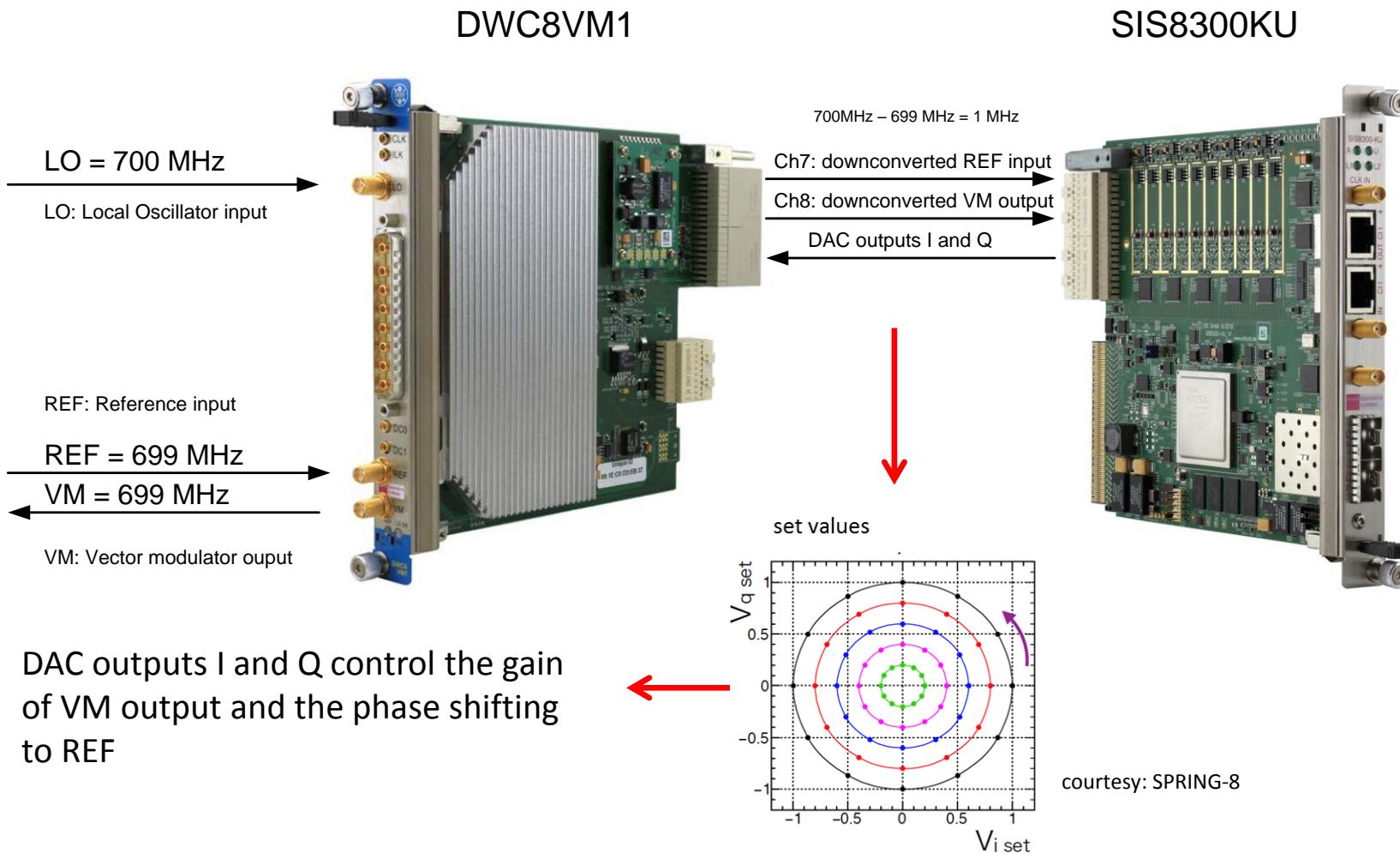
SIS830x Root GUI / Tools / FFT



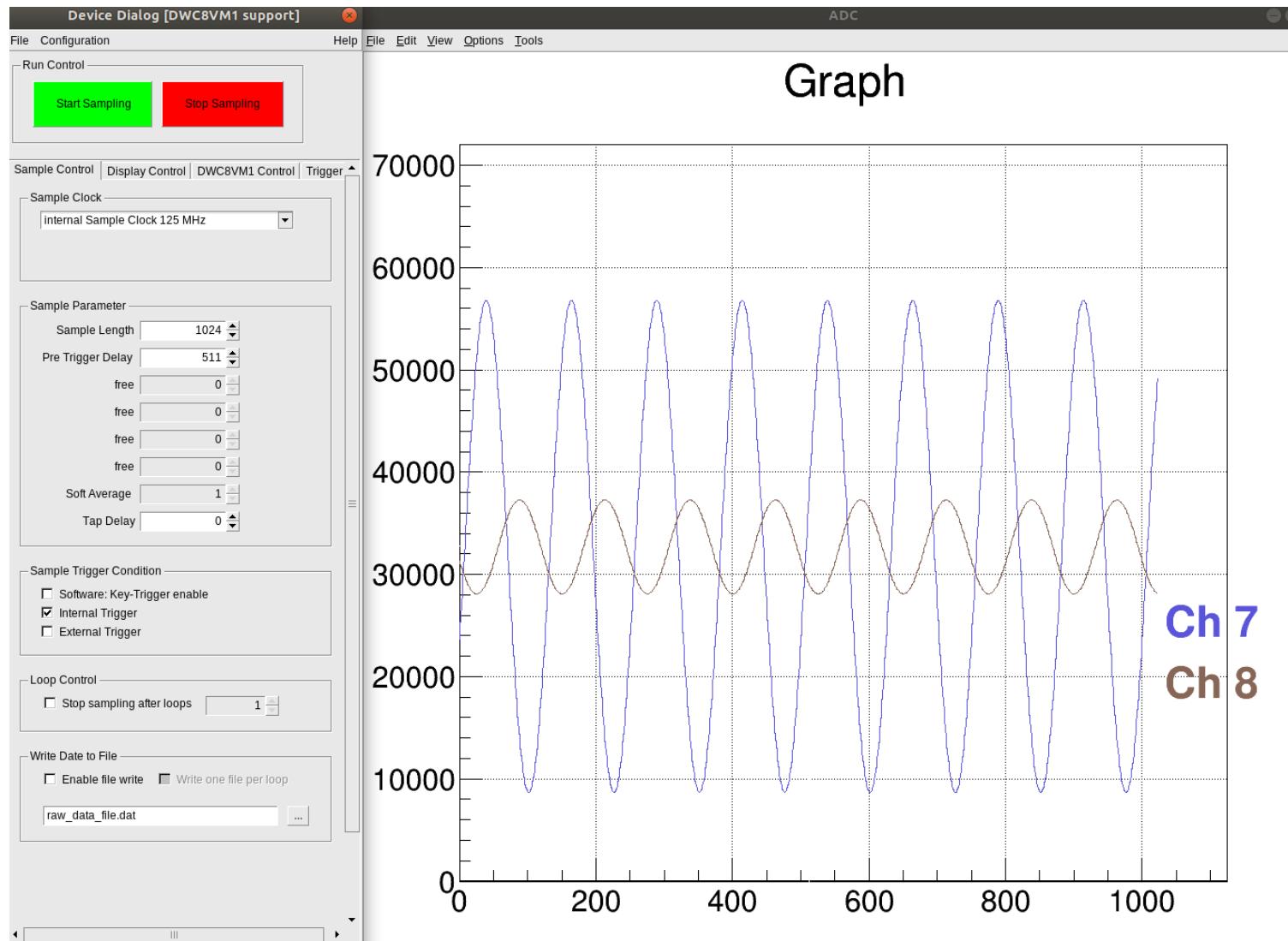
SIS830x Root GUI / Tools / FFT



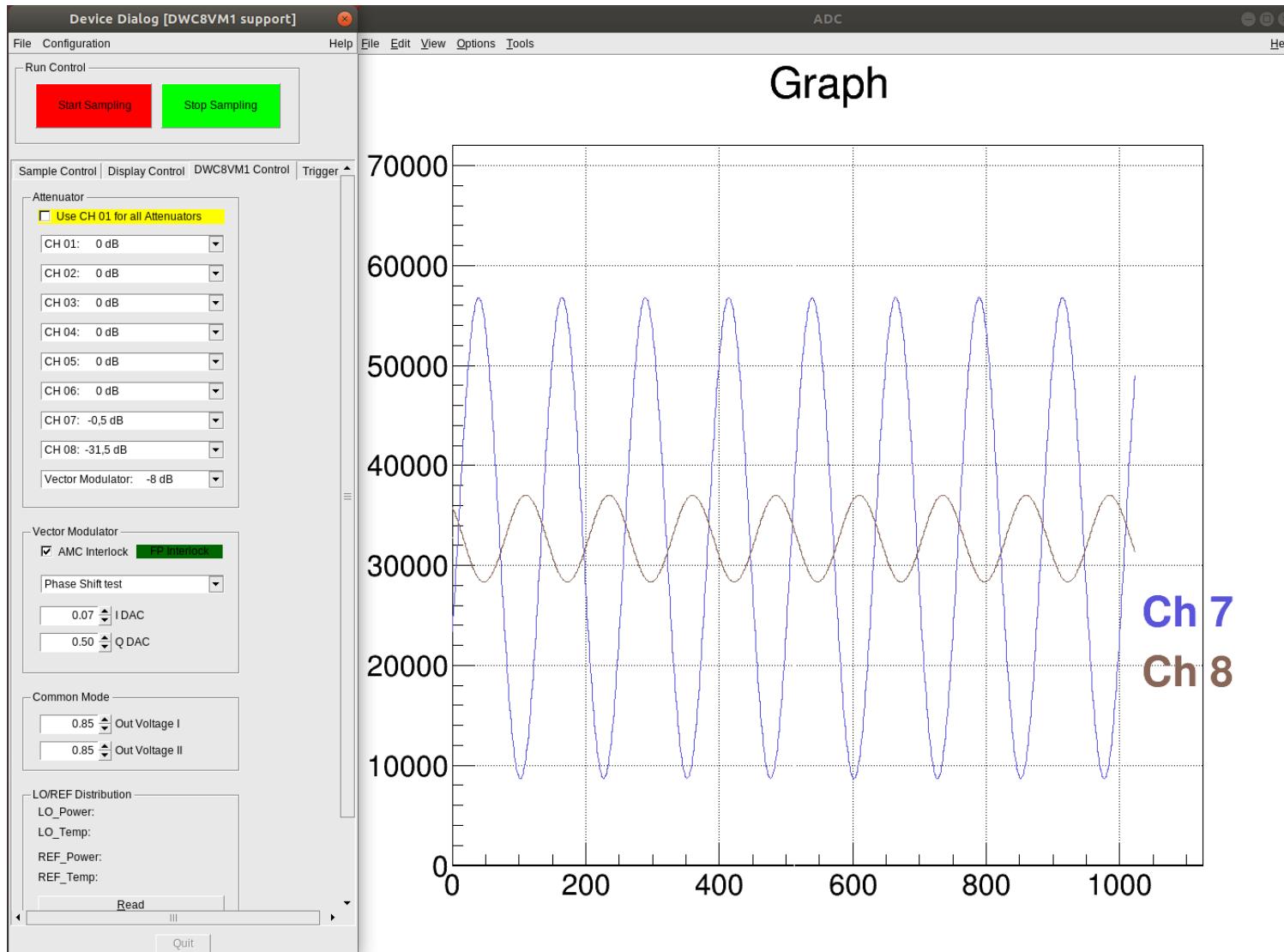
SIS830x Root GUI / Tools / Test with RTM DWC8VM1



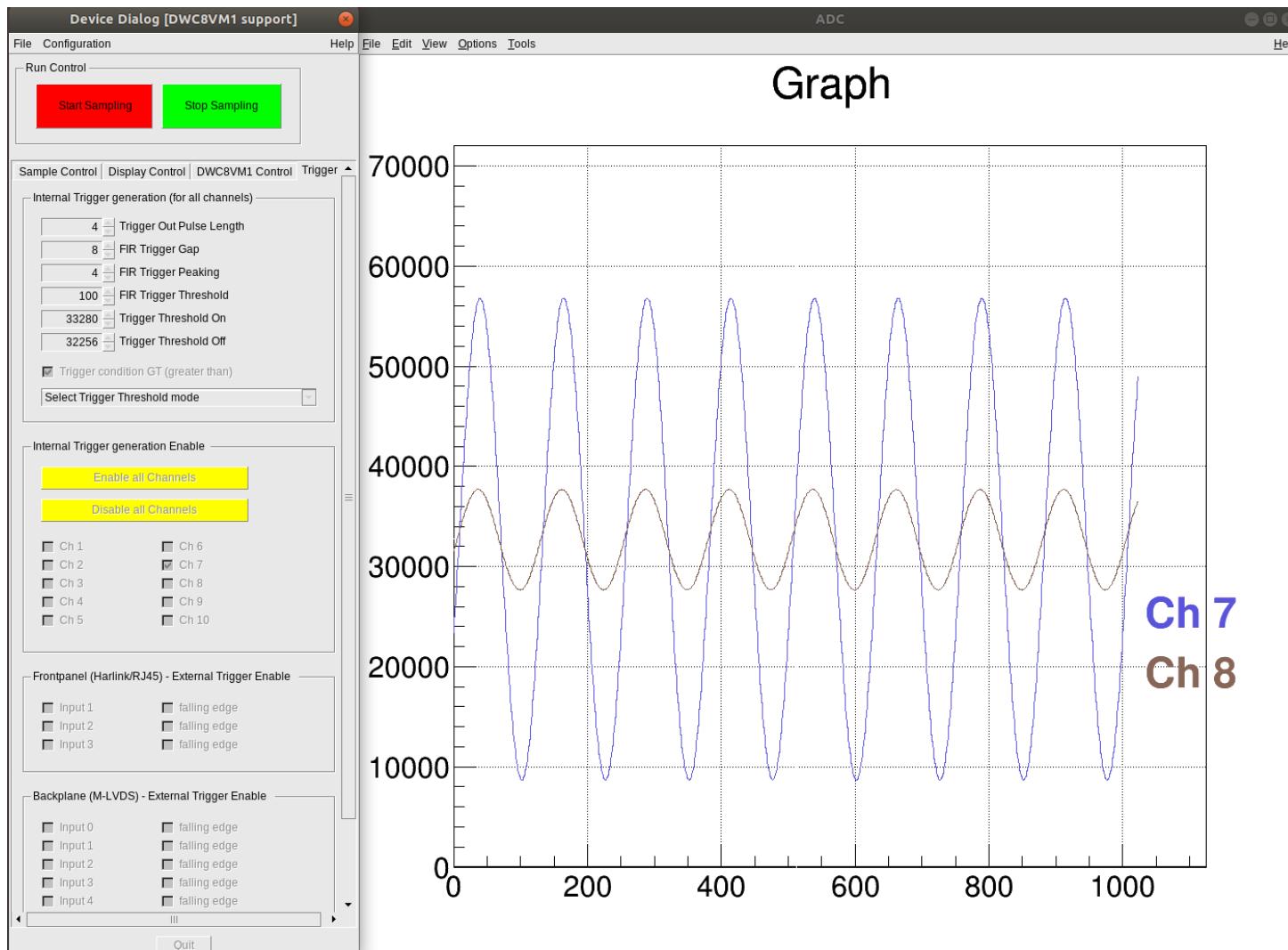
SIS830x Root GUI / Tools / Test with RTM DWC8VM1



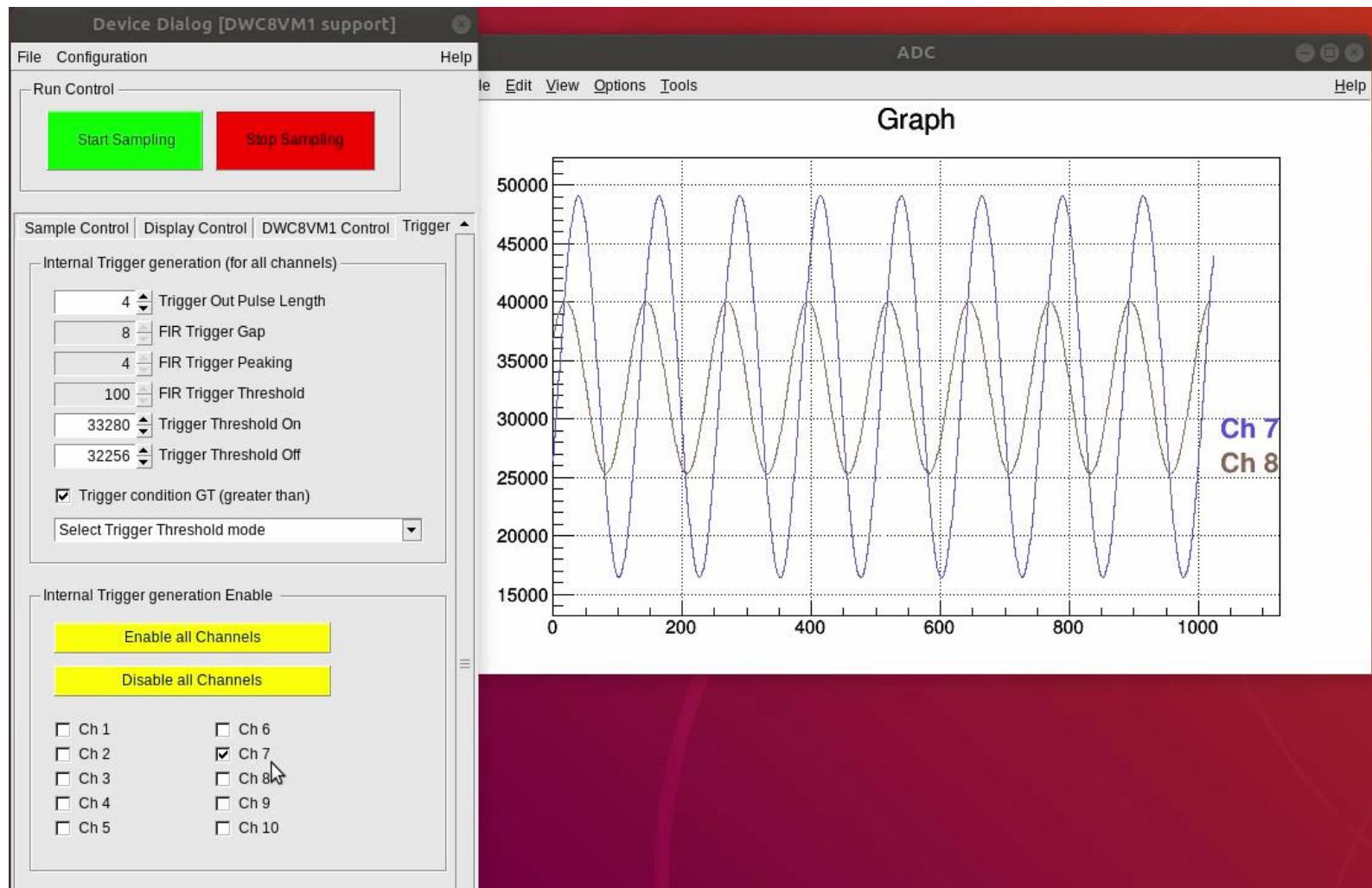
SIS830x Root GUI / Tools / Test with RTM DWC8VM1



SIS830x Root GUI / Tools / Test with RTM DWC8VM1



SIS830x Root GUI / Tools / Test with RTM DWC8VM1



Thank you!

7th MicroTCA Workshop
for Industry & Research

5 – 6 Dec 2018

CFEL, DESY, Hamburg

