adcDLL1.dll description

1. void DevDetect(unsigned char *DvDetect);

Check device.

If *DvDetect = 0 – device is not initialized

If *DvDetect = 1 – device is not initialized

2. unsigned char* ReadAndWriteToDevice(unsigned char *InputReport1, unsigned char*OutputReport1, int DevDet);

General purpose command. Is used to set acquisition parameters, start exposure, write and read from Flash memory.

usigned char *InputReport1 – 64 bytes array;

unsigned char* OutputReport1 – 64 bytes array with command to be written to device;

int DevDet – if 1 – make device initialization; if 0 – do not make device initialization (should be 0 if device was initialized already).

3. unsigned char* ReadAndWriteToDevice_new(unsigned char *InputReport1, unsigned char *OutputReport1, int DevDet)

General purpose command. Is used to set acquisition parameters, start exposure, write and read from Flash memory.

unsigned char *InputReport1 – 64 bytes array;

unsigned char* OutputReport1 – 64 bytes array with command to be written to device;

int DevDet – if 1 – make device initialization; if 0 – do not make device initialization (should be 0 if device was initialized already).

4. unsigned char Device_List_and_Init(unsigned char DevInitNmb)

DevInitNmb – If set to 0, Device_List_and_Init will return number of spectrometers connected to the computer. Otherwise should be set equal to the spectrometer number you want to initialize

5. int GetSpectra(signed __int16 *InputSpec1, unsigned char SpecNmb, unsigned __int16 startPix, unsigned __int16 endPix, unsigned char Fast, unsigned char test1, unsigned __int16 tot_startPix, unsigned __int16 tot_endPix)

Command to read acquired data.

“*InputSpec1” = 3653 elements (signed __int16) array,

“SpecNmb” = 1,

“endPix” = 3652,

“StartPix” = 0,

“Fast” = 0 or 1,

“test1” = 0.

“tot_startPix “ = 33

“tot_endPix” = 3685

Set acquisition parameters, start exposure and read spectra from device

Following steps should be made:

1. Set parameters and start acquisition. Send command *ReadAndWriteToDevice* with following parameters:

OutputReport1:

byte 1 = 1

byte 2 = low 8 bits of “Exposure time” (unsigned __int16) (exposure time (ms) = 2.375 (ms)* “Exposure time” (unsigned __int16))

byte 3 = “scans number” (char)

byte 4 = “blank scans number” (char)

byte 5 = 1

byte 6 = “use trigger” (bool) + 2*“keep trigger” (bool)

byte 7 = high 8 bits of “Exposure time” (unsigned __int16)

2. wait 2.375 (ms)* “Exposure time” (unsigned __int16)

3. Check acquisition status. Send command *ReadAndWriteToDevice* with following parameters:

OutputReport1:

byte 1 = 2

Return data (unsigned char *InputReport1):

Byte 3: if “1” – acquisition in progress, if “0” - acquisition is complete and data is ready to be transferred to computer.

4. Reset address. Send command *ReadAndWriteToDevice* with following parameters:

OutputReport1:

byte 1 = 3

5. Read one acquired spectra from device. Realize *GetSpectra* with following parameters:

“*InputSpec1” = 3653 elements (signed __int16) array (spectral data are written to this array),

“SpecNmb” = 1,

“endPix” = 3652,

“StartPix” = 0,

“Fast” = 0 or 1,

“test1” = 0.

“tot_startPix “ = 33

“tot_endPix” = 3685

6. Realize this point if “scans number” > 1. Send command *ReadAndWriteToDevice* with following parameters:

OutputReport1:

byte 1 = 9

byte 2 = 0x01

byte 3 = 0x80

7. Repeat 5 and 6 “scans number” times

8. Repeat 4

Read flash memory

“read_bytes” – bytes to read at “address” (3 bytes),

“Reading cycles” = (int)(“read_bytes” / 64)+1, j – current cycle number (j=0.. “Reading cycles”-1)

“current address” = “address” + j*64;

Make following steps “Reading cycles” times (where j – current cycle number):

1. Send command *ReadAndWriteToDevice_new* with following parameters:

OutputReport1:

byte 1 = 161,

byte 2 = 3rd (high) byte of the “current address”,

byte 3 = 2nd byte of the “current address”,

byte 4 = 1st (low) byte of the “current address”,

Return data (unsigned char *InputReport1):

64 bytes from flash memory starting from address “current address”.

Read calibration coefficients

Wavelength vs. CCD pixel calibration is perform via 2nd order polynomial approximation (Ax^2+Ax+C, where x – CCD pixel).

Baseline subtraction coefficients : a and b;

Calibration coefficients are written in Flash memory in ASCII format.

Read 80 bytes starting from address = 0;

0 – 15 bytes represent coefficient A

16 – 31 bytes represent coefficient B

32 – 47 bytes represent coefficient C

48 – 63 bytes represent coefficient a

64 – 79 bytes represent coefficient b

Read correction spectra

Read 7306 bytes starting from address = 4096.

i=1..7306 bytes contain 3653 two-bytes data points. i=2*n-1 low byte, i=2*i – high byte, where n=1..3653.

“correction spectra” = “3653 two-bytes data points” / 32768;

Spectra obtained from spectrometer should be calculated according to following instruction:

“*InputSpec1” / “correction spectra”;