



**KOREA
SPECTRAL
PRODUCTS**

서울시 구로구 디지털로 273.
에이스트원타워2차 402,407,408호
Tel) 02-2109-8871
www.ksp.co.kr

SDK for SM301-EX Manual

ver. 2.12.xx

CONTENTS

1. File Organization

1) Library	1
2) Driver	2
3) SDKs	2

2. Overview

1) Utilizing the SDK Functions	3
2) Using Curve Fitting to Calibrate	4

3. SDK Functions

1) spTestBoard	5
2) spSetIniVarEx	5
3) spSetIntTimeEx	5
4) spBalanceEx	6
5) spStoreAllEx	6
6) spSetTeCoolerEx	7
7) spReadStatusTE	7
8) spReadDataEx	8
9) spPolyCalc	8
10) spPolyFit	9
11) spGetWLTable	9
12) spDevInfo	10
13) spTECoolerPWR	10
14) spTECoolerReadTECPT	11
15) spSetWellDepth	12
16) spSetShutterPos	12
17) spConfigTriggerDelay	13
18) spConfigTrigger	13
19) spSetExtTrgMode	14
20) spReadDataEx_ExtTrg	14
21) spPerPixelUnifmCorrection	15

1. File Organization

1) Library

SM301-EX 구동 Functions을 사용하기 위한 Library file

ArrayBoardR6_Lib.dll

ArrayBoardR6_Lib.tlb

SMdbUSBABR6.dll

SMdbUSBABR6.lib

SDK Function을 호출하기 위해서는 개발 PC에 ArrayBoardR6_Lib.tlb 파일을 사용하여 COM 등록을 하여야 한다. 등록 방법은 두 가지가 있다.

-
- A. SM32ProForSM301vR6 설치 시 자동으로 등록
 - B. ArrayBoardR6_Lib.tlb 파일을 사용하여 수동 등록
 - 제공된 CD의 **"/SDKs/Com"** 폴더를 PC로 복사한다.
 - 명령 프롬프트를 관리자 권한으로 실행하여 복사한 **"/SDKs/Com"** 폴더로 경로 이동
 - command 입력

regasm.exe ArrayBoardR6_Lib.dll /tlb:ArrayBoardR6_lib.tlb

- C. 제공된 CD의 **"/SDKs/Com"** 폴더에 ArrayBoardR6_Lib.reg 파일을 관리자 권한으로 실행
-

A 방법으로 등록 시 정상 등록되지 않는다면 **B 방법** 혹은 **C 방법**으로 수동 등록

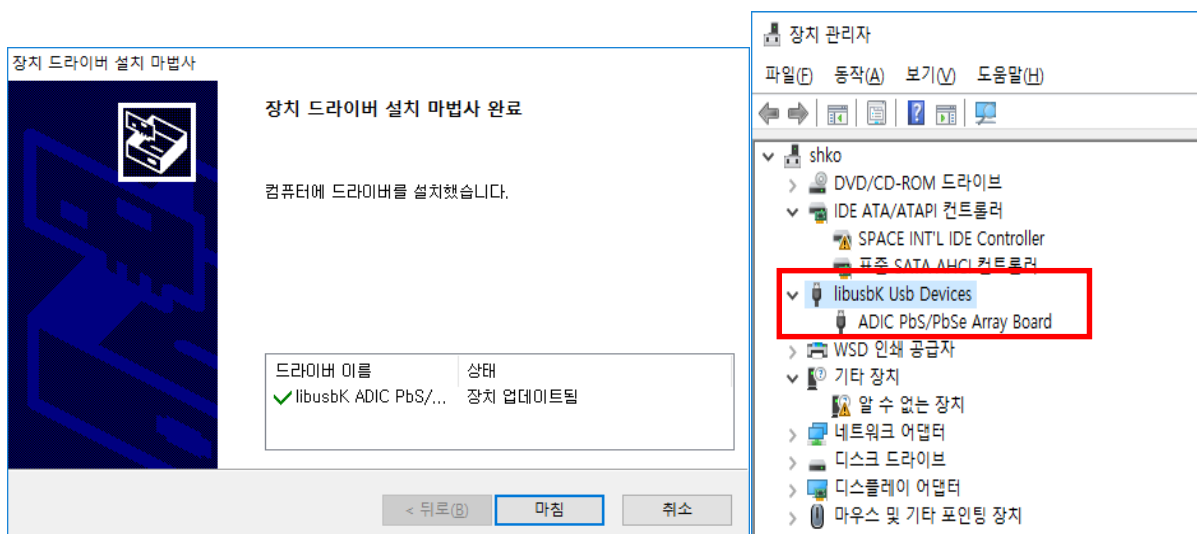
SMdbUSBABR6.dll과 ArrayBoardR6_Lib.dll은 같은 폴더에 있어야하며 개발 환경에 맞게 SMdbUSBABR6.dll과 SMdbUSBABR6.lib 파일을 include하여 사용하면 된다.

2) Driver

드라이버 설치 방법은 2가지가 있고, 설치 시 PC에 SM301-EX를 연결 후 진행해야 한다.

A. SM32ProForSM301vR6 설치 시 자동으로 설치

B. 제공된 CD의 "/Driver/" 폴더에 InstallDriver.exe 파일을 실행하여 설치 진행



드라이버 설치가 정상적으로 완료되면 위와 같이 녹색 체크와 함께 완료 메시지가 뜨고 장치관리자에 "libusbK Usb Devices" 아래 "ADIC PbS/PbSe Array Board"가 나타난다.

3) SDKs

위의 1) library com등록과 driver 설치가 정상적으로 완료되면 예제 코드 실행이 가능하다.

개발 환경 : Visual Studio 6.0 C++

SM301-EX Function들을 구동 예제 code로 DLL Include 방법 및 간단한 구동 방법을 확인할 수 있다.

2. Overview

1) Utilizing the SDK Functions

모든 SDK Functions은 board 제조사에서 제공한 ArrayBoardR6_Lib.dll을 기반으로 한다.
공급사에서 제공한 함수들을 사용하기 편하게 기능별로 묶어 SPdbUSBABR6.dll로 제공한다.

먼저 SM301-EX가 연결된 후 제일 먼저 **"spTestBoard"** 함수를 호출하여 PC에 연결된 모든 SM301-EX의 연결을 확인한다. 다음으로 **"spSetIniVarEX"** 함수를 호출하면 내부에서 구동 관련 설정 및 마지막으로 설정된 board의 설정값을 적용한다. 마지막으로 설정된 값은 board에 **"spStoreAllEx"** 함수가 호출 되었을 때의 설정 값들이다.

데이터를 수집하기 몇가지 준비 사항이 있다.

A. Integration Time 설정

- Integraion Time은 detector의 pixel에 빛에 노출시키는 시간이다. 설정 값이 높을 수록 약한 광의 신호를 감지할 수 있지만, 노이즈도 같이 증가한다.
- **"spSetIntTimeEx"** 함수로 설정 가능하며, 10us ~ 210ms 까지 설정이 가능하다.

B. TE-Cooler 구동

- TE-Cooler를 구동시켜 안정화를 시켜야 한다.
- **"spSetTeCoolerEx"**함수로 온도를 설정하고 **"spTECoolerPWR"**함수로 on/off한다.
- TE-Cooler의 전원을 on 시킨 후에는 **"spReadStatusTE"** 함수로 안정화 상태를 확인하여 안정화 된 이후에는 Balance 작업을 진행 할 수 있다.

C. Balance 작업

- PbS/PbSe detector는 데이터 수집 이전에 내부에서 board level의 신호 보정 작업을 해야 한다.
- TE-Cooler on 하여 안정화가 된 이후에 **"spBalanceEx"** 함수를 호출하면 내부에서 보정 작업을 진행한다.

위의 준비과정이 끝나면 **"spReadDataEx"** 함수로 detector signal을 확인 할 수 있다.
시간이 지나면 base line의 변형이 발생하기 때문에 Balance작업은 주기적으로 해주어야 한다.

2) Using Curve Fitting to Calibrate

Curve Fitting은 각 픽셀 별 Wavelength의 매칭을 위한 작업이다. SM301-EX와 같이 제공된 Data Sheet의 Calibration Data를 사용하여 계산한다. Calibration Data는 SM301-EX제작 시 알려진 파장의 광원을 측정 하여 광원이 관측되는 pixel과 파장을 매칭시킨 몇개의 Point를 정리한 data 이다.

이렇게 찾아낸 Point들을 "spPolyFit" 함수와 "spPolyCalc" 함수를 사용하여 다항식 계산을 하면 모든 Pixel의 매칭되는 파장을 알 수 있다. 일반적으로 3차다항식 함수를 사용하는 것이 정확한 파장 정보를 계산 할 수 있지만 교정 램프의 특성에 따라 매칭되는 Point의 개수가 적을 경우 더 낮은 차수의 다항식계산을 사용할 수도 있다.

3. SDK Functions

1) spTestBoard

long spTestBoard

(
)

연결된 Array Board를 찾는다. 최대 8대의 보드를 연결 할 수 있다.

RETURN

연결된 보드의 수, 0보다 작을 시 Error

2) spSetIniVarEx

long spSetIniVarEx

(
)

long IChannel

detector bias 설정, bad pixel 정보 및 board에 마지막으로 spStoreAllEx 함수로 저장된 설정 값을 불러와 적용한다. spTestBoard 함수 다음에 사용되어야 하며, 이 함수를 사용하지 않고 SM301-Ex를 구동하면 정상동작하지 않는다.

IChannel : 동작할 Spectrometer의 channel number

RETURN

정상 동작 할 경우 SP_NO_ERROR(1), 그렇지 않을 경우 음수를 반환

3) spSetIntTimeEx

long spSetIntTimeEx

(
)

long IChannel

long IIntTime

)

광 노출 시간을 설정 한다.

IChannel : 동작할 Spectrometer의 channel number

IIntTime : 1 ~ 65535의 digital 값이 입력되며, 실제 노출 시간은 다음과 같이 계산된다.

$$\text{Integration Time} = 3.2\mu\text{s} * (\text{IIntTime} - 1) + 4.025\mu\text{s}$$

설정 범위는 약 10 μs ~ 210ms 이다.

RETURN

정상 동작 할 경우 SP_NO_ERROR(1), 그렇지 않을 경우 음수를 반환

4) spBalanceEx

```
long spBalanceEx  
(  
    long IChannel  
    long IGlobalSkimOption  
)
```

각 픽셀의 Array를 교정하는 작업을 한다. Dac Coefficients를 결정하여 board level의 교정을 하고 S/W level의 교정으로 base을 7000, gain을 1.5로 설정 하여 광의 신호를 정확하게 확인 할 수 있도록 한다.

IChannel : 동작할 Spectrometer의 channel number

IGlobalSkimOption : 0 -> gskim value 사용하지 않고 보정

1 -> gskim value 자동설정 후 보정

2 -> board의 설정된 gskim을 사용하여 보정

RETURN

정상 동작 할 경우 SP_NO_ERROR(1), 그렇지 않을 경우 음수를 반환

5) spStoreAllEx

```
long spStoreAllEx  
(  
    long IChannel  
)
```

장비의 설정 상태 및 parameter를 내부 EEPROM에 저장한다.

IChannel : 동작할 Spectrometer의 channel number

RETURN

정상 동작 할 경우 SP_NO_ERROR(1), 그렇지 않을 경우 음수를 반환

6) spSetTeCoolerEx

```
long spSetTeCoolerEx
(
    long IChannel
    double dTECtemp
)
```

TE-Cooler의 동작 온도를 설정 한다.

IChannel : 동작할 Spectrometer의 channel number

dTECtemp : 설정할 온도를 입력 (설정 범위는 -21.1 ~ 28.5)

RETURN

정상 동작 할 경우 SP_NO_ERROR(1), 그렇지 않을 경우 음수를 반환

7) spReadStatusTE

```
long spReadStatusTE
(
    long IChannel
    long INumAvg
    long *IStatusTE
    double *dTeVal
)
```

TE-Cooler의 상태정보를 받아온다. IStatusTE[3]와 dTeVal[4]변수에 배열로 정보를 출력

IStatusTE는 TE-Cooler의 상태, dTeVal은 TE element의 volate 및 current 정보

IChannel : 동작할 Spectrometer의 channel number

INumAvg : 입력 숫자만큼 측정 후 평균하여 출력

IStatusTE[0] : TEC On/Off 상태 (1 = On, 0 = Off)

IStatusTE[1] : TEC 동작 상태 (1 = Cooling, 0 = Heating)

IStatusTE[2] : 안정화 여부 (1 = Stable, 0 = Not Stable)

dTeVal[0] : ITEC - a voltage related to the current in the TE element

dTeVal[1] : TMON - a voltage that provides temperature stability information

dTeVal[2] : VTEC - a voltage related to the voltage across the TE element

dTeVal[3] : VREF - the internal reference voltage for the TE controller

RETURN

정상 동작 할 경우 SP_NO_ERROR(1), 그렇지 않을 경우 음수를 반환

8) spReadDataEx

```
long spReadDataEx  
(  
    long IChannel  
    long *pIPixelArray  
)
```

detector pixel array data를 가져온다.

IChannel : 동작할 Spectrometer의 channel number

pIPixelArray : array data를 출력 (총 pixel 개수 = 256)

RETURN

정상 동작 할 경우 SP_NO_ERROR(1), 그렇지 않을 경우 음수를 반환

9) spPolyCalc

```
short spPolyCalc  
(  
    double *coefs  
    short order  
    double x  
    double *y  
)
```

이 함수는 다음 수식을 계산한다.

$$y = a_0 + a_1 \cdot x_1 + a_2 \cdot x_2 + \dots + a_N \cdot x_N$$

coefs : 다항식 계수 배열에 대한 포인터
spPolyFit을 사용하여 계산

x : 계산할 Pixel number

y : 계산된 값

RETURN

정상 동작 할 경우 SP_NO_ERROR(1), 그렇지 않을 경우 음수를 반환

10) spPolyFit

short spPolyFit

```
(
    double *x      //Array of independent variables
    double *y      //Array of dependent variables
    short numPts   //Number of points in independent and dependent arrays
    double *coefs  //Pointer of array to hold calculated coefficients[index : 0 - order]
    short order    //Order of polynomial
)
```

이 함수는 각 pixel 당 wavelength를 계산하는 다항식 함수를 찾는데 사용.

이 기능은 교정을 위한 기능이며, calibration data를 사용하여 계산한다.

calibration data는 교정 광원 또는 어떠한 narrow band filters를 투과하는 광원을 사용하여 wavelength를 알고있는 빛을 spectrometer로 측정하여 반응하는 pixel을 매칭한 data이다.

calibration data는 x와 y에 입력하며 배열 인덱스의 범위는 0 ~ numPts-1 이다.

계산 결과 나온 다항식 계수는 coefs배열에 출력되며 이후 spPolyCalc와 함께 사용하여 픽셀의 파장을 계산한다.

numPts : calibration data의 포인트 수

coefs : 다항식 계수 배열. 0 ~ order-1 의 범위

order : 다항식의 차원 수. 대부분의 경우 파장 보정에 최적인 3차 다항식 계산을 사용

RETURN

정상 동작 할 경우 SP_NO_ERROR(1), 그렇지 않을 경우 음수를 반환

11) spGetWLTable

long spGetWLTable

```
(
    long IChannel
    double *dWLTable
)
```

이 함수는 연결된 장치의 wavelength table을 가져온다. 간단히 wavelength table을 가져올때 사용. EEPROM에 저장되어있는 calibration data를 불러와 내부에서 다항식 계산을 하여 wavelength calibration을 한다.

IChannel : 동작할 Spectrometer의 channel number

dWLTable : 계산된 각 pixel별 wavelength 값 출력 배열

RETURN

정상 동작 할 경우 SP_NO_ERROR(1), 그렇지 않을 경우 음수를 반환

12) spDevInfo

long spDevInfo

```
(  
    long IChannel  
    char *strModel  
    char *strSerial  
)
```

이 함수는 연결된 장치의 model명과 serial number를 가져온다.

IChannel : 동작할 Spectrometer의 channel number

strModel : Model명 출력 배열

strSerial : serial number 출력 배열

RETURN

정상 동작 할 경우 SP_NO_ERROR(1), 그렇지 않을 경우 음수를 반환

13) spTECoolerPWR

long spSetTec

```
(  
    long IChannel  
    long long ITPowerState  
)
```

TECooler Power on/off 설정을 한다.

IChannel : 동작할 Spectrometer의 channel number

ITPowerState : 0 -> off, 1 -> on

RETURN

정상 동작 할 경우 SP_NO_ERROR(1), 그렇지 않을 경우 음수를 반환

14) spTECoolerReadTECPT

```
long spTECoolerReadTECPT
(
    long IChannel
    long *ITECSetPoint
)
```

spSetTeCoolerEx 에서 설정된 TEC Set Point의 값을 불러온다.

IChannel : 동작할 Spectrometer의 channel number

ITECSetPoint : 0 ~ 255의 digital 값으로 받아온다.

이 값은 다음과 같은 식으로 계산하여 온도로 변환해야 한다.

Variables / Coefficients:

- 1) TECSetpoint = 8 bit digital word loaded to digital pot, values range from 0 to 255 (inclusive)
- 2) TEPotRatio = TECSetpoint / 255, this is the ratio of the pot. This has values of 0 to 1 as is a Double data type.
- 3) Thermistor Coefficient A = 0.0033538646, Data type is Double
- 4) Thermistor Coefficient B = 0.0002565409, Data type is Double
- 5) Thermistor Coefficient C = 0.0000019243889, Data type is Double
- 6) Thermistor Coefficient D = 0.00000010969244, Data type is Double
- 7) dtmp = temporary variable, data type is Double

$$dtmp = \text{LN} \{ 3 * [(5 * \text{TEPotRatio} + 2) / (7 - 5 * \text{TEPotRatio})] \}$$

Note: LN is natural logarithm

$$\text{TempK} = 1 / (A + B * dtmp + C * dtmp^2 + D * dtmp^3); \text{ in Kelvin}$$

$$\text{TempC} = \text{TempK} - 273.15; \text{ in Celsius}$$

RETURN

정상 동작 할 경우 SP_NO_ERROR(1), 그렇지 않을 경우 음수를 반환

15) spSetWellDepth

```
long spSetWellDepth  
(  
    long IChannel  
    long IWellDepth  
)
```

Charge well size를 설정 한다.

IChannel : 동작할 Spectrometer의 channel number

IWellDepth : 0 ~ 7 값 입력

- 0 = Set 1pF charge well
- 1 = Set 4pF charge well
- 2 = Set 7pF charge well
- 3 = Set 10pF charge well
- 4 = Set 11pF charge well
- 5 = Set 14pF charge well
- 6 = Set 17pF charge well
- 7 = Set 20pF charge well

RETURN

정상 동작 할 경우 SP_NO_ERROR(1), 그렇지 않을 경우 음수를 반환

16) spSetShutterPos

```
long spSetShutterPos  
(  
    long IChannel  
    long IShutterPos  
)
```

Shutter Open/Close 설정을 한다.

IChannel : 동작할 Spectrometer의 channel number

IShutterPos : 0 -> Close, 1 -> Open

RETURN

정상 동작 할 경우 SP_NO_ERROR(1), 그렇지 않을 경우 음수를 반환

17) spConfigTriggerDelay

```
long spConfigTriggerDelay
(
    long IChannel
    long ITrigDelayValue
    long ITrigDelayMode
)
```

external trigger mode에서 입력 signal의 time out 사용 여부와 값을 설정 한다.

IChannel : 동작할 Spectrometer의 channel number

ITrigDelayValue : time out 값. 0 ~ 65535의 digital 값 입력

$2.26\mu s + (ITrigDelayValue - 1) * 0.2\mu s$ 으로 Maximum = 13.11ms

ITrigDelayMode : time out 사용 여부

0 : Disabled, 1 : Enabled

RETURN

정상 동작 할 경우 SP_NO_ERROR(1), 그렇지 않을 경우 음수를 반환

18) spConfigTrigger

```
long spConfigTriggerDelay
(
    long IChannel
    long ITriggerPolarity
)
```

external trigger mode사용 시 동작 모드 설정

IChannel : 동작할 Spectrometer의 channel number

ITriggerPolarity : falling edge와 rising edge 모드를 설정

0 : falling edge, 1 : rising edge

RETURN

정상 동작 할 경우 SP_NO_ERROR(1), 그렇지 않을 경우 음수를 반환

19) spSetExtTrgMode

```
long spSetExtTrgMode
(
    long lChannel
    long lExtTrgState
)
```

external trigger mode 사용 여부 설정. external trigger mode가 활성화 중에는 spReadDataEx_ExtTrg 함수 외의 다른 함수는 사용할 수 없다. 만약 다른 함수를 사용해야 한다면 external trigger mode를 비활성화 후 사용해야 한다.

lChannel : 동작할 Spectrometer의 channel number

lExtTrgState : 0 -> Disabled, 1 -> Enabled

RETURN

정상 동작 할 경우 SP_NO_ERROR(1), 그렇지 않을 경우 음수를 반환

20) spReadDataEx_ExtTrg

```
long spReadDataEx_ExtTrg
(
    long lChannel
    long *plPixelArray
)
```

external trigger mode가 활성화일 때만 사용가능하며, array data를 가져올 수 있다.

spConfigTriggerDelay함수에서 설정한 time out 값 동안 external trigger signal의 입력을 기다리며 time out이 지나면 SP_EXT_TRG_WAITING(-99)를 반환한다.

lChannel : 동작할 Spectrometer의 channel number

plPixelArray : array data를 출력 (총 pixel 개수 = 256)

RETURN

정상 동작 할 경우 SP_NO_ERROR(1), 그렇지 않을 경우 음수를 반환

external trigger signal 입력 time out 시 SP_EXT_TRG_WAITING(-99)를 반환

21) spPerPixelUnifmCorrection

```
long spPerPixelUnifmCorrection  
(  
    long IChannel  
)
```

S/W level의 base line 보정을 할 때 사용한다. 현재 측정 환경에서 base line을 7000으로 맞추고 각 pixel의 gain을 1.5로 설정한다.

IChannel : 동작할 Spectrometer의 channel number

RETURN

정상 동작 할 경우 SP_NO_ERROR(1), 그렇지 않을 경우 음수를 반환