API description document of ATP series optical fiber spectrometer Linux dynamic library

ABSTRACT

This document mainly describes the API interface description of the ATP series spectrometer under the Linux platform

OPTOSKY

Document History

Edition	Date	Author	Description
V1.0	2020-11-17	OPTOSKY	Initial Version

1. DESCRIPTION

libusb(https://libusb.info/)The API interface described in this document is developed based on libusb of Linux platform. Users can copy libusb library provided in this path to system library file path (/ usr / local / LIB), or install libusb by themselves （ https://libusb.info/).

Document description:

libOptoskySupport.so: Spectrometer operation supports dynamic library.

OptoskyDemo.c: API usage sample code.

OptoskySupport.h: API interface header file.

Makefile: The sample code compiles the makefile file.

2. Overview of API interface

API	EXPLAIN
optosky_open_spectrometer	Open the spectrometer
optosky_close_spectrometer	Close the spectrometer
optosky_get_library_version	Get dynamic library version number
optosky_get_vendor	Obtain spectrometer manufacturer information
optosky_get_PN	Get PN number
optosky_get_SN	Get SN number
optosky_get_integral_time_length	Get integral time length
optosky_get_integral_time_unit	Get integral time unit
optosky_get_wavelength_of_the_spec	Get wavelength range of spectrometer
optosky_get_integral_time	Get the current integral time
optosky_set_integral_time	Set integral time
optosky_set_average	Set the average number of scans
optosky_acquisition_dark_sync	Scan to obtain dark current (synchronous mode)
optosky_acquisition_spectrum_sync	Scanning acquisition spectrum (synchronous mode)
optosky_acquisition_dark_async	Scan dark current (asynchronous mode)
optosky_acquisition_spectrum_async	Scanning spectrum (asynchronous mode)
optosky_get_spectrum_data_async	Acquisition of spectrum data collected in asynchronous

API	EXPLAIN
	mode
optosky_set_external_GPIO_value	Set external GPIO port status
optosky_external_trigger_enable	Enable external trigger scanning function
optosky_external_trigger_disable	Turn off external trigger scanning

3. API interface description

(1) optosky_open_spectrometer

- Function prototype `INT_8S optosky_open_spectrometer(void);`
- Feature NO;
- Return value 0: Indicates that the device was opened successfully!
  -1: Indicates that the device has been turned on!

(2) optosky_close_spectrometer

- Function prototype `INT_8S optosky_close_spectrometer(void);`
- Feature NO;
- Return value 0: Indicates that the device was shut down successfully!
  -1： Indicates that the device is not turned on!

(3) optosky_get_library_version

- Function prototype `INT_8U *optosky_get_library_version(void);`
- Feature NO;
- Return value Dynamic library version number;

(4) optosky_get_vendor

- Function prototype `INT_8S optosky_get_vendor(INT_8S* vendor, INT_8U vendor_size);`
- Feature `vendor: Manufacturer information array;`
  `vendor_size：` Array length;
- Return value Indicates the length of manufacturer information string;
  -10: Indicates that the device is not turned on!

(5) optosky_get_PN

- Function prototype `INT_8S optosky_get_PN(INT_8S* pn, INT_8U pn_size);`
- Feature pn: PN number array;
  pn_size: Array length;
- Return value Represents the length of PN string;
  -10: Indicates that the device is not turned on!

(6) optosky_get_SN

- Function prototype `INT_8S optosky_get_SN(INT_8S* sn, INT_8U sn_size);`
- Feature sn: PN number array;
  sn_size: Array length;
- Return value Represents the length of SN string;
  -10: Indicates that the device is not turned on!

(7) optosky_get_integral_time_length

- Function prototype `__Attr_Integral_Length optosky_get_integral_time_length(void);`
- Feature NO;
- Return value IntegralTime_Size_16: Indicates that the integral time length is 2 bytes;
  IntegralTime_Size_32: Indicates that the integral time length is 4 bytes;

(8) optosky_get_integral_time_unit

- Function prototype `__Attr_Integral_Unit optosky_get_integral_time_unit(void);`
- Feature NO 无
- Return value IntegralTime_Unit_ms: Represents the integration time, in ms;
  IntegralTime_Unit_us: Represents the integration time unit is us;

(9) optosky_get_wavelength_of_the_spec

- Function prototype INT_16S optosky_get_wavelength_of_the_spec(FLOAT *wavelength ,INT_16U size);
- Feature wavelength: Wavelength range array;
  Size: The wavelength range length is obtained;
- Return value Represents the wavelength range length;
  -10: Indicates that the device is not turned on!

(10) optosky_get_integral_time

- Function prototype INT_8S optosky_get_integral_time(INT_32U *scanTime);
- Feature scanTime: Return integral time;
- Return value 0: Indicates success;
  -10: Indicates that the device is not turned on!

(11) optosky_set_integral_time

- Function prototype INT_8S optosky_set_integral_time(INT_32U scanTime);

- Feature scanTime: Integral time to be set;
- Return value 0: Indicates that the setting is successful;
  -10: Indicates that the device is not turned on!

(12) optosky_set_average

- Function prototype INT_8S optosky_set_average(INT_16U average);
- Feature average: The average acquisition times to be set;
- Return value 0: Indicates that the setting is successful;
  -10: Indicates that the device is not turned on!

(13) optosky_acquisition_dark_sync

- Function prototype INT_16S optosky_acquisition_dark_sync(INT_32U integrationTime, INT_16U *spectrum);
- Feature integrationTime: IIntegral time of spectral scanning;
  Spectrum: Spectral data returned;
- Return value Represents the length of the collected spectral data;
  -10: Indicates that the device is not turned on!

(14) optosky_acquisition_spectrum_sync

- Function prototype INT_16S optosky_acquisition_spectrum_sync(INT_32U integrationTime, INT_16U *spectrum);
- Feature integrationTime: Integral time of spectral scanning;
  Spectrum: Spectral data returned;
- Return value Represents the length of the collected spectral data;
  -10: Indicates that the device is not turned on!

(15) optosky_acquisition_dark_async

- Function prototype INT_8S optosky_acquisition_dark_async(INT_32U integrationTime);
- Feature integrationTime: Integral time of spectral scanning;
- Return value 0: Indicates that the set acquisition is successful;
  -10: Indicates that the device is not turned on!

(16) optosky_acquisition_spectrum_async

- Function prototype INT_8S optosky_acquisition_spectrum_async(INT_32U integrationTime);
- Feature integrationTime: Integral time of spectral scanning;
- Return value 0: Indicates that the set acquisition is successful;
  -10: Indicates that the device is not turned on!

(17) optosky_get_spectrum_data_async

- Function prototype INT_16S optosky_get_spectrum_data_async(INT_16U *spectrum);
- Feature spectrum: Spectral data returned;

- Return value 0: Indicates that the set collection is successful;
  -10: Indicates that the device is not turned on!

(18) optosky_set_external_GPIO_value

- Function prototype INT_8S optosky_set_external_GPIO_value (EXT_GPIO_PIN pin, EXT_GPIO_VALUE value);
- Feature pin: Control the number of GPIO ports (0 ~ 11);
  Value: Control the level state of GPIO port (0 / 1);
- Return value 0: Indicates that the setting is successful;
  -10: Indicates that the device is not turned on!

(19) optosky_external_trigger_enable

- Function prototype INT_8S optosky_external_trigger_enable(INT_16U integrationTime, void(*external_scan_callback)(INT_16U count, INT_16U *spectrum));
- Feature integrationTime: Spectral data returned;
  external_scan_callback: Register callback function (called when spectral data is received)
- Return value 0: Indicates that the setting is successful;
  -1: Means registered;
  -10: Indicates that the device is not turned on!

(20) optosky_external_trigger_disable

- Function prototype INT_8S optosky_external_trigger_disable(void);
- Feature spectrum: Spectral data returned;
- Return value 0: Indicates that the setting is successful;
  -1: Indicates that the setting failed;
  -10: Indicates that the device is not turned on!

4. Operation examples

1、Obtain wavelength range of spectrometer

```
FLOAT wavelength[5000] = {0};  /* The returned spectral range is saved as an array */
INT_16S ret = optosky_get_wavelength_of_the_spec(wavelength, 2048);
if(ret > 0) {
    INT_16U index = 0;
    for(; index<ret; index++) {
        printf("[%d]\t%f\n", index, wavelength[index]);
    }
}else {
    printf("Get the wavelength range of the spectrometer failed!\n");
}
```

2、Obtaining spectral data

```
INT_32U integrationTime = 10;     /* Integration time */
INT_16U spectrum[8000] = {0};     /* Saved array of scanned spectral data */

INT_16S ret = optosky_acquisition_spectrum_sync(integrationTime, spectrum);
if(ret > 0) {
    if(ret == 1) {
        printf("The spectrometer is busy now!\n");
        return;
    }
    INT_16U index = 0;
    for(; index<ret; index++) {
        printf("[%d]\t%d\n", index, spectrum[index]);
    }
    printf("spectrum length = %d\n", ret);
}else {
    printf("Get spectrum error[%d]!\n", ret);
}
```