EFW filter wheel Software Development Kit

Revision:2, 1
2017.2.14

Table of Contents

Change History

Change date	revision	comment
2017.2.14	2.1	Add API EFWCalibrate
2016.11.15	2.0	Add EFW_ERROR_CODE： EFW_ERROR_CLOSED Add EFWGetProductIDs EFWOpen: change argument to ID

1 Introduction

This SDK is used to operate EFW serial filter wheel, can be used by C, C++, C# and other develop tools, is suit for Windows, Linux, OSX operating system of x86 and x64.

Header file: EFW_filter.h

Under Windows the import library and dynamic library: EFW_filter.lib、EFW_filter.dll

Under Linux the dynamic library and static library: EFW_filter.so、EFW_filter.a

Under OSX the dynamic library and static library: EFW_filter.dylib、EFW_filter.a

Installation method:

Under Windows, extract the downloaded zip file to any directory, and add DLL's path to system environment variables, sometimes logout and re-login is required

2 Definition of enum-type and struct

```
2.1 typedef enum _EFW_ERROR_CODE
{
	EFW_SUCCESS = 0,
	EFW_ERROR_INVALID_INDEX,
	EFW_ERROR_INVALID_ID,
	EFW_ERROR_INVALID_VALUE,
	EFW_ERROR_CLOSED, //not opened
	EFW_ERROR_REMOVED, //failed to find the filter wheel, maybe the filter wheel has been
removed
	EFW_ERROR_MOVING,//filter wheel is moving
	EFW_ERROR_GENERAL_ERROR,//other error
	EFW_ERROR_CLOSED,
	EFW_ERROR_END = -1
}EFW_ERROR_CODE;
```

Returned error code

```
2.2 typedef struct _EFW_INFO
{
	int ID;
	char Name[64];
	int slotNum;
} EFW_INFO;
```

Filter wheel information

3 Function declaration

3.1 EFWGetNum
Syntax： int EFWGetNum()
Descriptions:
This should be the first API to be called, get number of connected EFW filter wheel, call this API to refresh device list if EFW is connected or disconnected

Return: number of connected EFW filter wheel. 1 means 1 filter wheel is connected.

3.2 EFWGetID
Syntax: EFW_ERROR_CODE EFWGetID(int index, int* ID)
Descriptions:
Get ID of filter wheel

Paras:
int index: the index of filter wheel, from 0 to N - 1, N is returned by EFWGetNum()

int* ID: pointer to ID. if the filter wheel is not opened, the ID is negative, otherwise the ID is a unique integer between 0 to EFW_ID_MAX - 1, after opened, all the operation is base on this ID, the ID will not change before the filter wheel is closed.

Return:
EFW_ERROR_INVALID_INDEX: index value is invalid
EFW_SUCCESS: operation succeeds

3.3 EFWGetProperty
Syntax: EFW_ERROR_CODE EFWGetProperty(int ID, EFW_INFO *pInfo)

Descriptions:
Get property of filter wheel.

Paras:
int ID: the ID of filter wheel

EFW_INFO *pInfo: pointer to structure containing the property of EFW

Return:
EFW_ERROR_INVALID_ID: invalid ID value
EFW_ERROR_CLOSED: the filter wheel is closed
EFW_ERROR_MOVING: slot number detection is in progress, generally this error will happen soon after filter wheel is connected.
EFW_SUCCESS: operation succeeds

3.4 EFWOpen
Syntax: EFW_ERROR_CODE EFWOpen(int ID)
Descriptions:
Open filter wheel

Paras:
int ID: the ID of filter whee

Return:
EFW_ERROR_INVALID_ID: invalid ID value
EFW_ERROR_GENERAL_ERROR: number of opened filter wheel reaches the maximum value.

EFW_ERROR_REMOVED: the filter wheel is removed.
EFW_SUCCESS: operation succeeds

3.5 EFWGetPosition
Syntax: EFW_ERROR_CODE EFWGetPosition(int ID, int *pPosition)

Descriptions:
Get position of slot

Paras:
int ID: the ID of filter wheel

int *pPosition: pointer to slot position, this value is between 0 to M - 1, M is slot number
this value is -1 if filter wheel is moving

Return:
EFW_ERROR_INVALID_ID: invalid ID value
EFW_ERROR_CLOSED: the filter wheel is closed
EFW_SUCCESS: operation succeeds
EFW_ERROR_ERROR_STATE: filter wheel is in error state

3.6 EFWSetPosition
Syntax: EFW_ERROR_CODE EFWSetPosition(int ID, int Position)

Descriptions:
Set position of slot

Paras:
int ID: the ID of filter wheel

int Position: slot position, this value is between 0 to M - 1, M is slot number

Return:
EFW_ERROR_INVALID_ID: invalid ID value
EFW_ERROR_CLOSED: the filter wheel is closed
EFW_SUCCESS: operation succeeds
EFW_ERROR_INVALID_VALUE: Position value is invalid
EFW_ERROR_MOVING: filter wheel is moving, should wait until idle
EFW_ERROR_ERROR_STATE: filter wheel is in error state

3.7 EFWSetDirection
Syntax: EFW_ERROR_CODE EFWSetDirection(int ID, bool bUnidirectional)
Descriptions:
Set unidirection of filter wheel

Paras:
int ID: the ID of filter wheel

bool bUnidirectional: if set as true, the filter wheel will rotate along one direction

Return:
EFW_ERROR_INVALID_ID: invalid ID value

EFW_ERROR_CLOSED: the filter wheel is closed
EFW_SUCCESS: operation succeeds

3.8 EFWGetDirection
Syntax: EFW_ERROR_CODE EFWGetDirection(int ID, bool *bUnidirectional)

Descriptions:
Get unidirection of filter wheel

Paras:
int ID: the ID of filter wheel

bool *bUnidirectional: pointer to unidirection value .

Return:
EFW_ERROR_INVALID_ID: invalid ID value
EFW_ERROR_CLOSED: the filter wheel is closed
EFW_SUCCESS: operation succeeds

3.9 EFWClose
Syntax: EFW_ERROR_CODE EFWClose(int ID)

Descriptions:
Close filter wheel

Paras:
int ID: the ID of filter wheel

Return:
EFW_ERROR_INVALID_ID: invalid ID value
EFW_SUCCESS: operation succeeds

3.10 EFWGetProductIDs
Syntax: int EFWGetProductIDs(int* pPIDs)

Descriptions:
get the product ID of each wheel, at first set pPIDs as 0 and get length and then malloc a buffer to load the PIDs

Paras:
int* pPIDs: pointer to array of PIDs

Return: length of the array.

3.11 EFWCalibrate
Syntax: EFW_ERROR_CODE EFWCalibrate(int ID)
Descriptions:
calibrate filter wheel

Paras:
int ID: the ID of filter wheel

Return:
EFW_ERROR_INVALID_ID: invalid ID value
EFW_ERROR_CLOSED: not opened
EFW_SUCCESS: operation succeeds
EFW_ERROR_MOVING: filter wheel is moving, should wait until idle
EFW_ERROR_ERROR_STATE: filter wheel is in error state
EFW_ERROR_REMOVED: filter wheel is removed

4 Suggested call sequence

Get count of connected filter wheels--> EFWGetNum
Get filter wheels' ID-> EFWGetID
Get filter wheels' name--> EFWGetProperty
Open filter wheel --> EFWOpen（Notes： this SDK can operate multiple filter wheels，distinguish by each filter wheel's ID）
Rotate--> EFWSetPosition
Close filter wheel-->EFWClose