

XEMA 系列相机接口文档

目录

XEMA 系列相机接口文档.....	1
版本记录.....	4
接口描述.....	5
1. Connect.....	5
2. Disconnect.....	5
3. Get camera resolution.....	5
4. Capture camera data.....	5
5. Get Depth Data.....	6
6. Get HeightMap Data.....	6
7. Get Brightness Data.....	6
8. Get Standard Plane Param.....	6
9. Get Height Map Data Base Param.....	7
10. Get Pointcloud Data.....	7
11. Get calibration parameters.....	7
12. Set Led Current parameters.....	7
13. Get Led Current parameters.....	8
14. Set Hdr parameters.....	8
15. Get Hdr parameters.....	8
16. Set Standard Plane parameters.....	8
17. Get Standard Plane parameters.....	9
18. Set Camera Exposure Parameters.....	9
19. Get Camera Exposure Parameters.....	9
20. Get Mixed Hdr Parameters.....	9
21. Set Camera Confidence Parameters.....	10
22. Set Mixed Hdr Parameters.....	9
23. Get Camera Confidence Parameters.....	10
24. Set Camera Gain Parameters.....	10
25. Get Camera Gain Parameters.....	10
26. Set Pointcloud Smoothing Parameters.....	10
27. Get Pointcloud Smoothing Parameters.....	11
28. Set Radius Filter Parameters.....	11
29. Get Radius Filter Parameters.....	11
30. Set Outlier Filter Parameters.....	11
31. Get Outlier Filter Parameters.....	12
32. Set Multiple Exposure Model Parameters.....	12
33. Set Repetition Exposure Num Parameters.....	12
34. Update Device List.....	12
35. Get All Device Base Info.....	12
36. Set Depth Filter Parameters.....	13

37. Get Depth Filter Parameters	13
38. Set Capture Engine	13
39. Get Capture Engine	13
40. Set Param Gray Rectify	13
41. Get Param Gray Rectify	14
42. Set Param Brightness Hdr Exposure	14
43. Get Param Brightness Hdr Exposure	14
44. Set Param Brightness Exposure Model	14
45. Get Param Brightness Exposure Model	14
46. Set Param Brightness Gain	15
47. Get Param Brightness Gain	15
48. Get Camera Channels	15
49. Get Color Brightness Data	15
50. Get Undistort Color Brightness Data	15
51. Get Sdk Version	16
52. Get Sdk Version	16
53. Capture Brightness Data	16
54. Set Param Reflect Filter	16
55. Get Param Reflect Filter	16
例程	17
错误码	18

版本记录

时间	版本	作者	描述
2022.02.21	V1.0.1	张观锦	完成 9 个基本函数
2022.03.18	V1.0.2	张观锦	完成 17 个基本函数
2022.05.13	V1.0.3	张观锦	完成 19 个基本函数
2022.06.16	V1.0.4	张观锦	完成 27 个基本函数
2022.08.05	V1.0.5	张观锦	完成 29 个基本函数
2022.09.09	V1.0.6	张观锦	完成 33 个基本函数
2022.11.21	V1.0.7	张观锦	完成 35 个基本函数 (修改两个函数)
2023.01.29	V1.1.0	张观锦	完成 37 个基本函数
2023.04.23	V1.2.0	张观锦	完成 39 个基本函数
2023.06.15	V1.3.0	张观锦	完成 47 个基本函数
2023.07.26	V1.4.0	张观锦	完成 50 个基本函数
2023.08.16	V1.4.1	张观锦	完成 51 个基本函数
2023.08.25	V1.5.0	张观锦	完成 52 个基本函数
2023.09.20	V1.5.1	张观锦	完成 53 个基本函数
2023.12.21	V1.5.2	张观锦	完成 55 个基本函数

接口描述

1-33 接口在头文件 open_cam3d_sdk.h 中，34-35 接口在头文件 enumerate.h 中。错误码定义在 camera_status.h 中。

1. Connect

```
int DfConnect(const char* camera_id);
//函数名: DfConnect
//功能: 连接相机
//输入参数: camera_id (相机 ip 地址)
//输出参数: 无
//返回值: 类型 (int) :返回 0 表示连接成功;返回-1 表示连接失败.
```

2. Disconnect

```
int DfDisconnect(const char* camera_id);
//函数名: DfDisconnect
//功能: 断开相机连接
//输入参数: camera_id (相机 ip 地址)
//输出参数: 无
//返回值: 类型 (int) :返回 0 表示断开成功;返回-1 表示断开失败.
```

3. Get camera resolution

```
int DfGetCameraResolution(int* width, int* height);
//函数名: DfGetCameraResolution
//功能: 获取相机分辨率
//输入参数: 无
//输出参数: width(图像宽)、height(图像高)
//返回值: 类型 (int) :返回 0 表示获取参数成功;返回-1 表示获取参数失败。
```

4. Capture camera data

```
int DfCaptureData(int exposure_num, char* &timestamp);
//函数名: DfCaptureData
//功能: 采集点云数据并阻塞至返回结果
//输入参数: exposure_num (曝光次数) : 1 (为单曝光) 、大于 1 为多曝光 (hdr、
重复曝光) 。
//输出参数: timestamp(时间戳)
//返回值: 类型 (int) :返回 0 表示获取采集点云成功;返回-1 表示采集点云失败。
```

5. Get Depth Data

```
int DfGetDepthData(unsigned short* depth);
//函数名: DfGetDepthData
//功能: 获取深度图数据
//输入参数: 无
//输出参数: depth(深度图)
//返回值: 类型 (int) :返回 0 表示获取数据成功;返回-1 表示采集数据失败
    参数是相应图像的数据指针, 调用方需要根据 DfGetCameraResolution 返回的结果 提
    前提前申请好内存空间。数据在内存中将以 H (行) /W (列) /C (通道) 的顺序排列。
```

6. Get HeightMap Data

```
//函数名: DfGetHeightMapData
//功能: 获取高度映射图数据
//输入参数: 无
//输出参数: height_map(高度映射图)
//返回值: 类型 (int) :返回 0 表示获取数据成功;返回-1 表示采集数据失败.参数是相
    应图像的数据指针, 调用方需要根据 DfGetCameraResolution 返回的结果 提前申请好
    内存空间。数据在内存中将以 H (行) /W (列) /C (通道) 的顺序排列。
```

7. Get Brightness Data

```
int DfGetBrightnessData(unsigned short* depth);
//函数名: DfGetBrightnessData
//功能: 获取亮度图数据
//输入参数: 无
//输出参数: brightness(亮度图)
//返回值: 类型 (int) :返回 0 表示获取数据成功;返回-1 表示采集数据失败。
    参数是相应图像的数据指针, 调用方需要根据 DfGetCameraResolution 返回的结果 提
    前申请好内存空间。数据在内存中将以 H (行) /W (列) /C (通道) 的顺序排列。
```

8. Get Standard Plane Param

```
int DfGetStandardPlaneParam(float* R,float* T);
//函数名: DfGetStandardPlaneParam
//功能: 获取基准平面参数
//输入参数: 无
//输出参数: R(旋转矩阵: 3*3)、T(平移矩阵: 3*1)
//返回值: 类型 (int) :返回 0 表示获取数据成功;返回-1 表示采集数据失败.
```

9. Get Height Map Data Base Param

```
int DfGetHeightMapDataBaseParam(float* R, float* T, float* height_map);  
//函数名: DfGetHeightMapDataBaseParam  
//功能: 获取校正到基准平面的高度映射图  
//输入参数: R(旋转矩阵)、T(平移矩阵)  
//输出参数: height_map(高度映射图)  
//返回值: 类型 (int) :返回 0 表示获取数据成功;返回-1 表示采集数据失败。
```

10. Get Pointcloud Data

```
int DfGetPointcloudData(float* point_cloud);  
//函数名: DfGetPointcloudData  
//功能: 获取点云数据 (深度图转点云)  
//输入参数: 无  
//输出参数: point_cloud(点云)  
//返回值: 类型 (int) :返回 0 表示获取数据成功;返回-1 表示采集数据失败  
参数是相应图像的数据指针, 调用方需要根据 DfGetCameraResolution 返回的结果 提前申请好内存空间。数据在内存中将以 H (行) /W (列) /C (通道) 的顺序排列。
```

11. Get calibration parameters

```
int DfGetCalibrationParam(struct CalibrationParam* calibration_param);  
//函数名: DfGetCalibrationParam  
//功能: 获取相机标定参数  
//输入参数: 无  
//输出参数: calibration_param (相机标定参数结构体)  
//返回值: 类型 (int) :返回 0 表示获取标定参数成功;返回-1 表示获取标定参数失败.
```

12. Set Led Current parameters

```
int DfSetParamLedCurrent(int led);  
//函数名: DfSetParamLedCurrent  
//功能: 设置 LED 电流  
//输入参数: led (电流值)  
//输出参数: 无  
//返回值: 类型 (int) :返回 0 表示获取标定参数成功;返回-1 表示获取标定参数失败.
```

13. Get Led Current parameters

```
int DfGetParamLedCurrent(int led);
//函数名: DfGetParamLedCurrent
//功能: 获取 LED 电流
//输入参数: 无
//输出参数: led (电流值)
//返回值: 类型 (int) :返回 0 表示获取标定参数成功;返回-1 表示获取标定参数失败.
```

14. Set Hdr parameters

```
int DfSetParamMixedHdr(int num, int exposure_param[6], int led_param[6]);
//函数名: DfSetParamMixedHdr
//功能: 设置混合多曝光参数 (最大曝光次数为 6 次)
//输入参数: num (曝光次数)、exposure_param[6] (6 个曝光参数、前 num 个有效)、
led_param[6] (6 个 led 亮度参数、前 num 个有效)
//输出参数: 无
//返回值: 类型 (int) :返回 0 表示获取标定参数成功;返回-1 表示获取标定参数失败.
```

15. Get Hdr parameters

```
int DfGetParamMixedHdr(int& num, int exposure_param[6], int led_param[6]);
//函数名: DfGetParamMixedHdr
//功能: 获取混合多曝光参数 (最大曝光次数为 6 次)
//输入参数: 无
//输出参数: num (曝光次数)、exposure_param[6] (6 个曝光参数、前 num 个有效)、
led_param[6] (6 个 led 亮度参数、前 num 个有效)
//返回值: 类型 (int) :返回 0 表示获取标定参数成功;返回-1 表示获取标定参数失败.
```

16. Set Standard Plane parameters

```
int DfSetParamStandardPlaneExternal(float* R, float* T);
//函数名: DfSetParamStandardPlaneExternal
//功能: 设置基准平面的外参
//输入参数: R(旋转矩阵: 3*3)、T(平移矩阵: 3*1)
//输出参数: 无
//返回值: 类型 (int) :返回 0 表示获取数据成功;返回-1 表示采集数据失败.
```

17. Get Standard Plane parameters

```
int DfGetParamStandardPlaneExternal(float* R, float* T);
//函数名: DfGetParamStandardPlaneExternal
//功能: 获取基准平面的外参
//输入参数: 无
//输出参数: R(旋转矩阵: 3*3)、T(平移矩阵: 3*1)
//返回值: 类型 (int) :返回 0 表示获取数据成功;返回-1 表示采集数据失败.
```

18. Set Camera Exposure Parameters

```
int DfSetParamCameraExposure(float exposure);
//函数名: DfSetParamCameraExposure
//功能: 设置相机曝光时间
//输入参数: exposure(相机曝光时间)
//输出参数: 无
//返回值: 类型 (int) :返回 0 表示获取数据成功;返回-1 表示采集数据失败.
```

19. Get Camera Exposure Parameters

```
int DfGetParamCameraExposure(float& exposure);
//函数名: DfGetParamCameraExposure
//功能: 获取相机曝光时间
//输入参数: 无
//输出参数: exposure(相机曝光时间)
//返回值: 类型 (int) :返回 0 表示获取数据成功;返回-1 表示采集数据失败.
```

20. Get Mixed Hdr Parameters

```
int DfGetParamMixedHdr(int& num, int exposure_param[6], int led_param[6]);
//函数名: DfGetParamMixedHdr
//功能: 获取混合多曝光参数 (最大曝光次数为 6 次)
//输入参数: 无
//输出参数: num (曝光次数)、exposure_param[6] (6 个曝光参数、前 num 个有效)、
led_param[6] (6 个 led 亮度参数、前 num 个有效)
//返回值: 类型 (int) :返回 0 表示获取标定参数成功;返回-1 表示获取标定参数失败.
```

21. Set Mixed Hdr Parameters

```
int DfSetParamMixedHdr(int num, int exposure_param[6], int led_param[6]);
//函数名: DfSetParamMixedHdr
//功能: 设置混合多曝光参数 (最大曝光次数为 6 次)
//输入参数: num (曝光次数)、exposure_param[6] (6 个曝光参数、前 num 个有效)、
```

```
led_param[6] (6个led亮度参数、前num个有效)
//输出参数: 无
//返回值: 类型 (int) :返回0表示获取标定参数成功;返回-1表示获取标定参数失败.
```

22. Set Camera Confidence Parameters

```
int DfSetParamCameraConfidence(float confidence);
//函数名: DfSetParamCameraConfidence
//功能: 设置相机置信度
//输入参数: confidence(相机置信度)
//输出参数: 无
//返回值: 类型 (int) :返回0表示获取数据成功;返回-1表示采集数据失败.
```

23. Get Camera Confidence Parameters

```
int DfGetParamCameraConfidence(float& confidence);
//函数名: DfGetParamCameraConfidence
//功能: 获取相机置信度
//输入参数: 无
//输出参数: confidence(相机置信度)
//返回值: 类型 (int) :返回0表示获取数据成功;返回-1表示采集数据失败.
```

24. Set Camera Gain Parameters

```
int DfSetParamCameraGain(float gain);
//函数名: DfSetParamCameraGain
//功能: 设置相机增益
//输入参数: gain(相机增益)
//输出参数: 无
//返回值: 类型 (int) :返回0表示获取数据成功;返回-1表示采集数据失败.
```

25. Get Camera Gain Parameters

```
int DfGetParamCameraGain(float& gain);
//函数名: DfGetParamCameraGain
//功能: 获取相机增益
//输入参数: 无
//输出参数: gain(相机增益)
//返回值: 类型 (int) :返回0表示获取数据成功;返回-1表示采集数据失败.
```

26. Set Pointcloud Smoothing Parameters

```
int DfSetParamSmoothing(int smoothing);
//函数名: DfSetParamSmoothing
```

```
//功能： 设置点云平滑参数  
//输入参数： smoothing(0:关、1-5:平滑程度由低到高)  
//输出参数： 无  
//返回值： 类型 (int) :返回 0 表示获取数据成功;返回-1 表示采集数据失败.
```

27. Get Pointcloud Smoothing Parameters

```
int DfGetParamSmoothing(int& smoothing);  
//函数名： DfGetParamSmoothing  
//功能： 获取点云平滑参数  
//输入参数： 无  
//输出参数： smoothing(0:关、1-5:平滑程度由低到高)  
//返回值： 类型 (int) :返回 0 表示获取数据成功;返回-1 表示采集数据失败.
```

28. Set Radius Filter Parameters

```
int DfSetParamRadiusFilter(int use,float radius,int num);  
//函数名： DfSetParamRadiusFilter  
//功能： 设置点云半径滤波参数  
//输入参数： use(开关：1 开、0 关)、radius(半径) 、 num (有效点)  
//输出参数： 无  
//返回值： 类型 (int) :返回 0 表示获取数据成功;返回-1 表示采集数据失败.
```

29. Get Radius Filter Parameters

```
int DfGetParamRadiusFilter(int& use, float& radius, int& num);  
//函数名： DfGetParamRadiusFilter  
//功能： 获取点云半径滤波参数  
//输入参数： 无  
//输出参数： use(开关：1 开、0 关)、radius(半径) 、 num (有效点)  
//返回值： 类型 (int) :返回 0 表示获取数据成功;返回-1 表示采集数据失败.
```

30. Set Outlier Filter Parameters

```
int DfSetParamOutlierFilter(float threshold);  
//函数名： DfSetParamOutlierFilter  
//功能： 设置外点过滤阈值  
//输入参数： threshold(阈值 0-100)  
//输出参数： 无  
//返回值： 类型 (int) :返回 0 表示获取数据成功;返回-1 表示采集数据失败.
```

31. Get Outlier Filter Parameters

```
int DfGetParamOutlierFilter(float& threshold);
//函数名: DfGetParamOutlierFilter
//功能: 获取外点过滤阈值
//输入参数: 无
//输出参数: threshold(阈值 0-100)
//返回值: 类型 (int) :返回 0 表示获取数据成功;返回-1 表示采集数据失败.
```

32. Set Multiple Exposure Model Parameters

```
int DfSetParamMultipleExposureModel(int model);
//函数名: DfSetParamMultipleExposureModel
//功能: 设置多曝光模式
//输入参数: model(1: HDR(默认值)、2: 重复曝光)
//输出参数: 无
//返回值: 类型 (int) :返回 0 表示设置参数成功;否则失败。
```

33. Set Repetition Exposure Num Parameters

```
int DfSetParamRepetitionExposureNum(int num);
//函数名: DfSetParamRepetitionExposureNum
//功能: 设置重复曝光数
//输入参数: num(2-10)
//输出参数: 无
//返回值: 类型 (int) :返回 0 表示设置参数成功;否则失败。
```

34. Update Device List

```
int DfUpdateDeviceList(int& device_num);
//函数名: DfUpdateDeviceList
//功能: 获取可连接设备数
//输入参数: device_num(设备数)
//输出参数: 无
//返回值: 类型 (int) :返回 0 表示连接成功;返回-1 表示连接失败.
```

35. Get All Device Base Info

```
int Df GetAllDeviceInfo(DeviceBaseInfo* pDeviceInfo, int* pBufferSize);
//函数名: Df GetAllDeviceInfo
//功能: 获取设备基本信息
//输入参数: pDeviceInfo(设备信息)、pBufferSize (设备结构体内存尺寸)
//输出参数: 无
//返回值: 类型 (int) :返回 0 表示连接成功;返回-1 表示连接失败.
```

36. Set Depth Filter Parameters

```
int DfSetParamDepthFilter(int use, float depth_filter_threshold);
//函数名: DfSetParamRadiusFilter
//功能: 设置深度图滤波参数
//输入参数: use(开关: 1 开、0 关)、depth_filterthreshold(过滤的噪声阈值)
//输出参数: 无
//返回值: 类型 (int) :返回 0 表示设置参数成功;否则失败。
```

37. Get Depth Filter Parameters

```
int DfGetParamDepthFilter(int& use, float& depth_filter_threshold);
//函数名: DfSetParamRadiusFilter
//功能: 获取深度图滤波参数
//输入参数: 无
//输出参数: use(开关: 1 开、0 关)、depth_filterthreshold(距离过滤的噪声阈值)
//返回值: 类型 (int) :返回 0 表示获取参数成功;否则失败。
```

38. Set Capture Engine

```
int DfSetCaptureEngine(XemaEngine engine);
//函数名: DfSetCaptureEngine
//功能: 设置采集引擎
//输入参数: engine
//输出参数:
//返回值: 类型 (int) :返回 0 表示设置参数成功;返回-1 表示设置参数失败。
```

39. Get Capture Engine

```
int DfGetCaptureEngine(XemaEngine engine);
//函数名: DfGetCaptureEngine
//功能: 获取采集引擎
//输入参数: engine
//输出参数:
//返回值: 类型 (int) :返回 0 表示设置参数成功;返回-1 表示设置参数失败。
```

40. Set Param Gray Rectify

```
int DfSetParamGrayRectify(int use, int radius, float sigma);
//函数名: DfSetParamGrayRectify
//功能: 设置点云灰度补偿参数
//输入参数: use(开关: 1 开、0 关)、radius(半径: 3、5、7、9) 、sigma (补偿强度,
范围 0-100)
//输出参数: 无
```

//返回值：类型（int）：返回 0 表示设置参数成功；否则失败。

41. Get Param Gray Rectify

```
int DfGetParamGrayRectify(int& use, int& radius, float& sigma);
//功能： 获取点云灰度补偿参数
//输入参数： 无
//输出参数： use(开关：1 开、0 关)、radius(半径：3、5、7、9) 、sigma (补偿强度，范围 0-100)
//返回值： 类型（int）：返回 0 表示获取参数成功；否则失败。
```

42. Set Param Brightness Hdr Exposure

```
int DfSetParamBrightnessHdrExposure(int num, int exposure_param[10]);
//函数名： DfSetParamBrightnessHdrExposure
//功能： 设置亮度图多曝光参数（最大曝光次数为 10 次）
//输入参数： num （曝光次数）、exposure_param[6]（6 个曝光参数、前 num 个有效）
//输出参数： 无
//返回值： 类型（int）：返回 0 表示设置参数成功；否则失败。
```

43. Get Param Brightness Hdr Exposure

```
int DfGetParamBrightnessHdrExposure(int& num, int exposure_param[10]);
//函数名： DfGetParamBrightnessHdrExposure
//功能： 获取亮度图多曝光参数（最大曝光次数为 10 次）
//输入参数： 无
//输出参数： num （曝光次数）、exposure_param[10]（10 个曝光参数、前 num 个有效）
//返回值： 类型（int）：返回 0 表示设置参数成功；否则失败。
```

44. Set Param Brightness Exposure Model

```
int DfSetParamBrightnessExposureModel(int model);
//函数名： DfSetParamBrightnessExposureModel
//功能： 设置亮度图曝光模式
//输入参数： model (1: 单曝光、2: 曝光融合)
//输出参数： 无
//返回值： 类型（int）：返回 0 表示设置参数成功；否则失败。
```

45. Get Param Brightness Exposure Model

```
int DfGetParamBrightnessExposureModel(int& model);
//函数名： DfGetParamBrightnessExposureModel
//功能： 获取亮度图曝光模式
//输入参数： 无
```

```
//输出参数: model (1: 单曝光、2: 曝光融合)  
//返回值: 类型 (int) :返回 0 表示设置参数成功;否则失败。
```

46. Set Param Brightness Gain

```
int DfSetParamBrightnessGain(float gain);  
//函数名: DfSetParamBrightnessGain  
//功能: 设置亮度图增益  
//输入参数: gain(亮度图增益)  
//输出参数: 无  
//返回值: 类型 (int) :返回 0 表示设置参数成功;否则失败。
```

47. Get Param Brightness Gain

```
int DfGetParamBrightnessGain(float& gain);  
//函数名: DfGetParamBrightnessGain  
//功能: 获取亮度图增益  
//输入参数: 无  
//输出参数: gain(亮度图增益)  
//返回值: 类型 (int) :返回 0 表示设置参数成功;否则失败。
```

48. Get Camera Channels

```
int DfGetCameraChannels(int* channels);  
//函数名: DfGetCameraChannels  
//功能: 获取相机图像通道数  
//输入参数: 无  
//输出参数: channels(通道数)  
//返回值: 类型 (int) :返回 0 表示获取参数成功;返回-1 表示获取参数失败.
```

49. Get Color Brightness Data

```
int DfGetColorBrightnessData(unsigned char* brightness, XemaColor color);  
//函数名: DfGetColorBrightnessData  
//功能: 获取亮度图  
//输入参数: 无  
//输出参数: brightness(亮度图),color(亮度图颜色类型)  
//返回值: 类型 (int) :返回 0 表示获取数据成功;返回-1 表示采集数据失败.
```

50. Get Undistort Color Brightness Data

```
int DfGetUndistortColorBrightnessData(unsigned char* brightness, XemaColor color);  
//函数名: DfGetUndistortColorBrightnessData  
//功能: 获取去畸变后的彩色亮度图
```

```
//输入参数: 无  
//输出参数: brightness(亮度图)  
//返回值: 类型 (int) :返回 0 表示获取数据成功;返回-1 表示采集数据失败.
```

51. Get Sdk Version

```
int DfGetSdkVersion(char version[64]);  
//函数名: DfGetSdkVersion  
//功能: 获取 sdk 版本  
//输入参数: 无  
//输出参数: version(版本)  
//返回值: 类型 (int) :返回 0 表示设置参数成功;否则失败。
```

52. Get Sdk Version

```
int DfCaptureBrightnessData(unsigned char* brightness, XemaColor color);  
//函数名: DfCaptureBrightnessData  
//功能: 获取亮度图  
//输入参数: color(图像颜色类型)  
//输出参数: brightness(亮度图)  
//返回值: 类型 (int) :返回 0 表示获取数据成功;返回-1 表示采集数据失败.
```

53. Capture Brightness Data

```
int DfCaptureBrightnessData(unsigned char* brightness, XemaColor color);  
//函数名: DfCaptureBrightnessData  
//功能: 获取亮度图  
//输入参数: color(图像颜色类型)  
//输出参数: brightness(亮度图)  
//返回值: 类型 (int) :返回 0 表示获取数据成功;返回-1 表示采集数据失败.
```

54. Set Param Reflect Filter

```
int DfSetParamReflectFilter(int use, float param_b);  
//函数名: DfSetParamReflectFilter  
//功能: 设置反射滤波参数  
//输入参数: use(开关: 1 开、0 关)、param_b (过滤系数: 范围 0-100)  
//输出参数: 无  
//返回值: 类型 (int) :返回 0 表示设置参数成功;否则失败。
```

55. Get Param Reflect Filter

```
int DfGetParamReflectFilter(int& use, float& param_b);
```

```
//函数名: DfGetParamReflectFilter  
//功能: 获取反射滤波参数  
//输入参数: 无  
//输出参数: use(开关: 1 开、0 关)、param_b (过滤系数: 范围 0-100)  
//返回值: 类型 (int) :返回 0 表示设置参数成功;否则失败。
```

例程

详细请看 example.cpp

属性说明

```
//相机标定参数结构体  
struct CalibrationParam  
{  
    //相机内参  
    double intrinsic[3*3];  
    //相机外参  
    double extrinsic[4*4];  
    //相机畸变  
    //<k1,k2,p1,p2,k3,k4,k5,k6,s1,s2,s3,s4>暂时只使用 5 个畸变参数  
    double distortion[1*12];  
};  
  
//设备基本信息结构体  
struct DeviceBaseInfo  
{  
    //相机内参  
    char mac[64];  
    //相机外参  
    char ip[64];  
};  
  
//采集引擎  
enum class XemaEngine  
{  
    Normal = 0,  
    Reflect = 1,  
};  
  
enum class XemaColor  
{  
    Rgb = 0,  
    Bgr = 1,
```

```
Bayer = 2,  
Gray= 3,  
};
```

错误码

错误码	码值	描述
DF_SUCCESS	0	成功
DF_FAILED	-1	失败
DF_UNKNOWN	-2	未知命令
DF_BUSY	-3	相机占用
DF_NOT_CONNECT	-4	相机未连接
DF_ERROR_NETWORK	-5	网络出错
DF_ERROR_2D_CAMERA	-6	2d 相机故障
DF_ERROR_INVALID_PARAM	-7	无效参数
DF_ERROR_LIGHTCRAFTER_SET_MODEL	-8	光机投影模式设置出错
DF_ERROR_LIGHTCRAFTER_SET_TRIGGEROUT	-9	光机触发设置出错
DF_ERROR_LIGHTCRAFTER_SET_CURRENT	-10	光机设置亮度出错
DF_ERROR_LIGHTCRAFTER_SET_PATTERN_ORDER	-11	光机条纹设置出错
DF_ERROR_CAMERA_STREAM	-12	相机操作流出错
DF_ERROR_CAMERA_GRAP	-13	相机采图出错
DF_FRAME_CAPTURING	-14	相机正在采集帧数据
DF_ERROR_LOST_TRIGGER	-15	没有触发信号
DF_ERROR_LOST_PATTERN_SETS	-16	光机缺少条纹图