目录

1. HRG-TOF 相机编程指南概述

HRG TOF 相机 SDK 主要用于获取 TOF 相机图像数据，且可以获取及设置 TOF 相机相关参数，适用于 TOF 相机 HRG 系列产品。该编程手册汇总了 SDK 二次开发中的可能遇到的常见问题，方便用户在使用 HRG-TOF SDK 时，遇到相关或者类似问题时，能够快速找到相应的解决方法，提高开发效率。

2. 常见问题介绍

2.1. 无法连接相机?

首先，检查相机是否正常工作，正常工作时相机光源会闪起来，且有轻微的声音；其次，相机的网络端口默认 IP 地址是 192.168.0.6，USB 端口也虚拟成网口默认 IP 为 192.168.1.6，确保使用 SDK 的主机可以 ping 通相机。

2.2. 成功连接相机，无法获取图像数据?

检查使用 SDK 的主机防火墙是否关闭或者允许 SDK 应用程序通过防火墙。

2.3. 成相机数据噪声比较重，如何设置参数？

SDK 中提供幅度值滤波，通过设置幅度值阈值，可以将低于幅度值阈值的对应像素的深度值置为 0，直接滤除返回的弱光像素点。

2.4. 相机深度数据是什么坐标系下的？

获取的深度图是球坐标系下的值，如用相机照射一堵平的白墙，形成的深度图是一个曲面，转换到点云时是一个平面。

2.5. 使用测试程序无法可视化显示深度图、幅度图、点云？

在 Linux 平台下，提供的 demo 中 main.cpp 有控制是否使用 opencv 显示、pcl 显示的宏，通过宏可以控制是否使用幅度图映射到灰度图显示、深度图映射到彩色图显示、点云三维显示等功能。如果使用该功能，需要安装相对应的库，并同时在 CMakeLists 配置相关环境；

在 windows 下，提供的 demo 中 main.cpp 类似 Linux 平台，并同时需要配置 VS 的相关库环境。

2.6. 使用 USB 通信时，相机供电顺序是什么？

使用 USB 通信时，先使用电源给相机供电，然后再使用 USB 端口通信，否则会引起数据异常。

3. 函数调用实例

具体请参考提供的 tofsdk_test 工程中的 main.cpp。

注意：SDK 获取相机有 2 中方式，一种是通过回调函数获取幅度图和**球坐标系下**的深度图，另外一种是从数据队列中获取幅度图和**球坐标系下**的深度图。如 Linux 测试环境下提供的 demo 中 callback_show_test 测试工程和 capture_show_test 测试工程。

4. 函数说明

4.1. 连接相机 Hrg_LogConfig

函数：__API int Hrg_LogConfig(Hrg_Log_Level log_level, const char* path = "./log/")

参数：[in] log_level 设置日志等级

[in] path 设置日志保存路径

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于设置日志参数，一般用于定位相机信息，日志等级可参考枚举类型 Hrg_Log_Level。

4.2. 连接相机 Hrg_OpenDevice

函数：__API int Hrg_OpenDevice(const Hrg_Dev_Info *dev, Hrg_Dev_Handle *handle)

参数：[in] dev 设置连接方式的参数信息

[out] handle 返回该连接方式的句柄

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于和相机之间建立连接，目前 HRG 系列 TOF 相机支持网口和网口传输数据。

4.3. 关闭相机连接 Hrg_CloseDevice

函数：__API int Hrg_CloseDevice(const Hrg_Dev_Handle *handle)

参数：[in] handle 该连接方式的句柄

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于关闭连接，需与 Hrg_OpenDevice 配套使用。

4.4. 开启数据流 Hrg_StartStream

函数：_API int Hrg_StartStream(const Hrg_Dev_Handle *handle)

参数：[in] handle 该连接方式的句柄

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于打开数据流。

4.5. 设置帧模式 Hrg_StopStream

函数：_API int Hrg_StopStream(const Hrg_Dev_Handle *handle)

参数：[in] handle 该连接方式的句柄

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于关闭数据流。

4.6. 获取一帧数据 Hrg_GetFrame

函数：_API int Hrg_GetFrame(const Hrg_Dev_Handle *handle, Hrg_Frame *frame)

参数：[in] handle 该连接方式的句柄

[out] frame 获取的一帧数据

返回值：0 表示成功，-1 表示失败。

说明：

该接口主要用于获取一帧数据，帧详细信息参见结构体 Hrg_Frame。

4.7. 释放一帧数据 Hrg_FreeFrame

函数：__API int Hrg_FreeFrame(const Hrg_Dev_Handle *handle, Hrg_Frame *frame)

参数：[in] handle 该连接方式的句柄

[in] frame 获取的数据帧

返回值：0 表示成功，-1 表示失败。

说明：

该接口主要用于释放一帧数据，需要与 GetFrame 接口配对使用

4.8. 获取深度和幅度数据 Hrg_GetDepthI16andAmplitudeData

函数：__API int Hrg_GetDepthI16andAmplitudeData(const Hrg_Dev_Handle *handle,

const Hrg_Frame* frame,

int16_t* depth,

int16_t* amplitude);

参数：[in] handle 该连接方式的句柄

[int] frame 获取的一帧数据

[out] depth 提取 int16_t 类型球坐标系下的深度数据，单位为 mm

[out] amplitude 提取 int16_t 类型的幅度数据

返回值：0 表示成功，-1 表示失败。

说明：

该接口主要用于从获取一帧数据解析出深度数据和幅度数据，注意该接口需要 Hrg_Frame_Mode 为 Mode_RawPhases 时使用，同时 depth 和 amplitude 指针需要用户预分配内存。

4.9. 获取深度和幅度数据 Hrg_GetDepthF32andAmplitudeData

函数：__API int Hrg_GetDepthF32andAmplitudeData(const Hrg_Dev_Handle *handle,

const Hrg_Frame* frame,

float* depth,

int16_t* amplitude)

参数：[in] handle 该连接方式的句柄

[int] frame 获取的一帧数据

[out] depth 提取 float 类型球坐标系下的深度数据，单位为 m

[out] amplitude 提取 int16_t 类型的幅度数据

返回值：0 表示成功，-1 表示失败。

说明：

该接口主要用于从获取一帧数据解析出深度数据和幅度数据，注意该接口需要 Hrg_Frame_Mode 为 Mode_RawPhases 时使用，同时 depth 和 amplitude 指针需要用户预分配内存。

4.10. 获取点云数据 Hrg_GetXYZDataF32

函数：__API int Hrg_GetXYZDataF32(const Hrg_Dev_Handle *handle,

const int16_t* depth,

float* pcl,

int32_t pointCount);

参数：[in] handle 该连接方式的句柄

[in] depth 深度数据，数据类型为 int16_t，单位为 mm

[out] pcl 点云数据，包含（x,y,z）3 个元素，（0,0,0）为无效点，单位为 m

[in] pointCount 点云点的数量

返回值：0 表示成功，-1 表示失败。

说明：

该接口主要用于将 int16_t 类型的深度数据转换为点云数据。

4.11. 获取点云数据 Hrg_GetXYZDataF32_f

函数：__API int Hrg_GetXYZDataF32_f(const Hrg_Dev_Handle *handle,

const float* depth,

float* pcl,

int32_t pointCount)

参数：[in] handle 该连接方式的句柄

[in] depth 深度数据，数据类型为 float，单位为 m

[out] pcl 点云数据，包含（x,y,z）3 个元素，（0,0,0）为无效点，单位为 m

[in] pointCount 点云点的数量

返回值：0 表示成功，-1 表示失败。

说明：

该接口主要用于用于将 float 类型的深度数据转换为点云数据。

4.12. 获取点云数据 Hrg_GetXYZDataI16

函数：__API int Hrg_GetXYZDataI16(const Hrg_Dev_Handle *handle,

const int16_t* depth,

int16_t* pcl,

int32_t pointCount)

参数：[in] handle 该连接方式的句柄

[in] depth 深度数据，数据类型为 int16_t，单位为 mm

[out] pcl 点云数据，包含（x,y,z）3 个元素，（0,0,0）为无效点，单位为 mm

[in] pointCount 点云点的数量

返回值：0 表示成功，-1 表示失败。

说明：

该接口主要用于用于将 int16_t 类型的深度数据转换为 int16_t 类型点云数据。

4.13. 深度数据映射为 RGB 图 Hrg_DepthI16ToRGB

函数：__API int Hrg_DepthI16ToRGB(const Hrg_Dev_Handle *handle,

uint8_t *dst,

int dst_len,

const int16_t *src,

int src_len,

int16_t range_min_m,

int16_t range_max_m)

参数：[in] handle 该连接方式的句柄

[out] dst 映射后的 RGB 图像数据

[in] dst_len 映射后的 RGB 图像数据所占字节数

[out] src 深度数据，数据类型为 int16_t，单位为 mm

[in] src_len 深度数据像素点数量

[in] range_min_m 映射最小距离

[in] range_max_m 映射最大距离

返回值：0 表示成功，-1 表示失败。

说明：

该接口主要用于用于将 int16_t 类型的深度数据映射为 RGB 图像。

4.14. 深度数据映射为 RGB 图 Hrg_DepthF32ToRGB

函数：__API int Hrg_DepthF32ToRGB(const Hrg_Dev_Handle *handle,

uint8_t *dst,

int dst_len,

const float *src,

int src_len,

float range_min_m,

float range_max_m)

参数：[in] handle 该连接方式的句柄

[out] dst 映射后的 RGB 图像数据

[in] dst_len 映射后的 RGB 图像数据所占字节数

[in] src 深度图数据，数据类型为 float，单位为 mm

[in] src_len 深度图数据像素点数量

[in] range_min_m 映射最小距离

[in] range_max_m 映射最大距离

返回值：0 表示成功，-1 表示失败。

说明：

该接口主要用于用于将 float 类型的深度数据映射为 RGB 图像。

4.15. 幅度数据映射为灰度图 Hrg_AmplitudeToIR

函数：__API int Hrg_AmplitudeToIR(const Hrg_Dev_Handle *handle,

uint8_t *dst,

int dst_len,

const int16_t *src,

int src_len,

int balance)

参数：[in] handle 该连接方式的句柄

[out] dst 映射后的灰度图像数据

[in] dst_len 映射后的灰度图像数据所占字节数

[in] src 幅度图数据

[in] src_len 幅度图数据像素点数量

[in] balance 映射最大幅度值

返回值：0 表示成功，-1 表示失败。

说明：

该接口主要用于用于将幅度数据映射为灰度图像。

4.16. 获取原始四相位数据 Hrg_GetRawPhasesData

函数：__API int Hrg_GetRawPhasesData(const Hrg_Dev_Handle *handle,

const Hrg_Frame *frame,

int16_t **phase1,

int16_t **phase2,

int16_t **phase3,

int16_t **phase4)

参数：[in] handle 该连接方式的句柄

[in] frame 获取的一帧数据

[out] phase1 从一帧数据中提取的相位为 0° 原始数据

[out] phase2 从一帧数据中提取的相位为 90° 原始数据

[out] phase3 从一帧数据中提取的相位为 180° 原始数据

[out] phase4 从一帧数据中提取的相位为 270° 原始数据

返回值：0 表示成功，-1 表示失败。

说明：

该接口主要用于用于获取原始四个原始相位数据，注意该接口需要 Hrg_Frame_Mode 为 Mode_RawPhases 时使用。

4.17. 获取 WDR 原始相位数据 Hrg_GetRawPhasesDataWDRRate

函数：__API int Hrg_GetRawPhasesDataWDR(const Hrg_Dev_Handle *handle,

const Hrg_Frame *frame,

int16_t **phase1_1,

int16_t **phase2_1,

int16_t **phase3_1,

int16_t **phase4_1,

int16_t **phase1_2,

int16_t **phase2_2,

int16_t **phase3_2,

int16_t **phase4_2)

参数：[in] handle 该连接方式的句柄

[in] frame 获取的一帧数据

[out] phase1_1 从一帧数据中提取的短曝光相位为 0° 原始数据

[out] phase2_1 从一帧数据中提取的短曝光相位为 90° 原始数据

[out] phase3_1 从一帧数据中提取的短曝光相位为 180° 原始数据

[out] phase4_1 从一帧数据中提取的短曝光相位为 270° 原始数据

[out] phase1_2 从一帧数据中提取的长曝光相位为 0° 原始数据

[out] phase2_2 从一帧数据中提取的长曝光相位为 90° 原始数据

[out] phase3_2 从一帧数据中提取的长曝光相位为 180° 原始数据

[out] phase4_2 从一帧数据中提取的长曝光相位为 270° 原始数据

返回值：0 表示成功，-1 表示失败。

说明：

该接口主要用于用于获取 WDR 连续长短曝光的原始四个原始相位数据，注意该接口需要 Hrg_Frame_Mode 为 Mode_RawPhases 时使用。

4.18. 获取深度图数据 Hrg_GetDepthData

函数：__API int Hrg_GetDepthData(const Hrg_Dev_Handle *handle,

const Hrg_Frame* frame,

int16_t** depth);

参数：[in] handle 该连接方式的句柄

[in] frame 获取的一帧数据

[out] depth 获取球坐标系下的深度图数据，单位为 mm

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于获取深度图数据，注意需要 Hrg_Frame_Mode 为 Mode_DistAmp 时使用。

4.19. 获取幅度图数据 Hrg_GeAmplitudeData

函数：__API int Hrg_GeAmplitudeData(const Hrg_Dev_Handle *handle,

const Hrg_Frame* frame,

int16_t**amplitude)

参数：[in] handle 该连接方式的句柄

[in] frame 获取的一帧数据

[out] amplitude 获取的幅度图数据

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于获取幅度图数据，注意需要 Hrg_Frame_Mode 为 Mode_DistAmp 时使用。

4.20. 获取支持的距离模式 Hrg_GetRangeModeList

函数：__API int Hrg_GetRangeModeList(const Hrg_Dev_Handle *handle, Hrg_Range_Mode_List* rangeModeList)

参数：[in] handle 该连接方式的句柄

[out] rangeModeList 获取支持的距离模式列表

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于获取支持的距离模式。

4.21. 设置帧模式 Hrg_SetFrameMode

函数：__API int Hrg_SetFrameMode(const Hrg_Dev_Handle *handle, const Hrg_Frame_Mode frameMode)

参数：[in] handle 该连接方式的句柄

[in] frameMode 设置的帧模式

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于设置帧模式，在 Mode_DistAmp 模式下，深度图和幅度图计算由相机完成，在 Mode_RawPhases 模式下，深度图和幅度图计算由客户端 SDK 完成。

4.22. 获取帧模式 Hrg_GetFrameMode

函数：__API int Hrg_GetFrameMode(const Hrg_Dev_Handle *handle, Hrg_Frame_Mode *frameMode)

参数：[in] handle 该连接方式的句柄

[out] frameMode 获取的帧模式

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于获取帧模式，在 Mode_DistAmp 模式下，深度图和幅度图计算由相机完成，在 Mode_RawPhases 模式下，深度图和幅度图计算由客户端 SDK 完成。

4.23. 设置帧模式 Hrg_SetRangeMode

函数：__API int Hrg_SetRangeMode(const Hrg_Dev_Handle *handle, const Hrg_Range_Mode rangeMode)

参数：[in] handle 该连接方式的句柄

[in] rangeMode 设置的距离模式

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于设置距离模式。在不同波长光源下不同距离模式测量范围不同，具体如下：

940nm:

Mode_Range_None: 无（原始数据模式，未做标定，用户可自由设置参数）

Mode_Range_S: 约 25cm～120cm

Mode_Range_M: 约 35cm～180cm

Mode_Range_L: 约 50cm～220cm

Mode_Range_XL: 约 60cm～300cm

Mode_Range_Custom: 可根据客户定制参数，默认不支持

Mode_Range_WDR: 约 25cm～300cm，宽动态距离模式，实际帧率会降为原来的一一半

850nm:

Mode_Range_None: 无（原始数据模式，未做标定，用户可自由设置参数）

Mode_Range_S：约 25cm～130cm

Mode_Range_M：约 40cm～200cm

Mode_Range_L：约 50cm～250cm

Mode_Range_XL：约 70cm～350cm

Mode_Range_Custom：可根据客户定制参数，默认不支持

Mode_Range_WDR：约 25cm～350cm，宽动态距离模式，实际帧率会降为原来的一一半

4.24. 获取帧模式 Hrg_GetRangeMode

函数：__API int Hrg_GetRangeMode(const Hrg_Dev_Handle *handle, Hrg_Range_Mode* rangeMode)

参数：[in] handle 该连接方式的句柄

[out] rangeMode 设置的距离模式

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于获取距离模式。

4.25. 设置积分时间 Hrg_SetIntegrationTime

函数：__API int Hrg_SetIntegrationTime(const Hrg_Dev_Handle *handle, const uint32_t integrationTime)

参数：[in] handle 该连接方式的句柄

[in] integrationTime 设置的积分时间

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于设置积分时间， 只有在距离模式为 Mode_Range_None 时可以成功设置。

4.26. 获取积分时间 Hrg_GetIntegrationTime

函数：__API int Hrg_GetIntegrationTime(const Hrg_Dev_Handle *handle, uint32_t* integrationTime)

参数：[in] handle 该连接方式的句柄

[out] integrationTime 积分时间

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于获取设置积分时间。

4.27. 设置增益 Hrg_SetGain

函数：__API int Hrg_SetGain(const Hrg_Dev_Handle *handle, const uint32_t gain)

参数：[in] handle 该连接方式的句柄

[in] gain 增益

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于设置增益，只有在距离模式为 Mode_Range_None 时可以成功设置。。

4.28. 获取增益 Hrg_GetGain

函数：__API int Hrg_GetGain(const Hrg_Dev_Handle *handle, uint32_t* gain)

参数：[in] handle 该连接方式的句柄

[out] gain 增益

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于获取设置的增益。

4.29. 设置帧率 Hrg_SetFrameRate

函数：__API int Hrg_SetFrameRate(const Hrg_Dev_Handle *handle, const uint32_t frameRate)

参数：[in] handle 该连接方式的句柄

[in] frameRate 帧率

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于设置帧率，只有在距离模式为 Mode_Range_None 时可以成功设置。。

4.30. 获取增益 Hrg_GetFrameRate

函数：__API int Hrg_GetFrameRate(const Hrg_Dev_Handle *handle, uint32_t* frameRate)

参数：[in] handle 该连接方式的句柄

[out] frameRate 帧率

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于获取设置的帧率。

4.31. 设置帧率 Hrg_SetModulationFrequency

函数：__API int Hrg_SetModulationFrequency(const Hrg_Dev_Handle *handle,

const uint32_t modulationFrequency)

参数：[in] handle 该连接方式的句柄

[in] modulationFrequency 调制频率

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于设置调制频率，只有在距离模式为 Mode_Range_None 时可以成功设置。。

4.32. 获取增益 Hrg_GetModulationFrequency

函数：__API int Hrg_GetModulationFrequency(const Hrg_Dev_Handle *handle, uint32_t* modulationFrequency)

参数：[in] handle 该连接方式的句柄

[out] modulationFrequency 调制频率

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于获取设置的调制频率。

4.33. 设置幅度值阈值 Hrg_SetAmplitudeThreshold

函数：__API int Hrg_SetAmplitudeThreshold(const Hrg_Dev_Handle *handle, const uint32_t amplitudeThreshold)

参数：[in] handle 该连接方式的句柄

[in] amplitudeThreshold 幅度值阈值

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于设置幅度值阈值，低于幅度值阈值像素点对应的深度值置为 0。

4.34. 获取幅度值阈值 Hrg_GetAmplitudeThreshold

函数：__API int Hrg_GetAmplitudeThreshold(const Hrg_Dev_Handle *handle, uint32_t* amplitudeThreshold)

参数：[in] handle 该连接方式的句柄

[out] amplitudeThreshold 幅度值阈值

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于获取设置的幅度值阈值。

4.35. 设置距离偏差值 Hrg_SetDistanceOffset

函数：__API int Hrg_Se 幅度值阈 tDistanceOffset(const Hrg_Dev_Handle *handle, const int32_t distanceOffset)

参数：[in] handle 该连接方式的句柄

[in] distanceOffset 距离偏差值，单位 mm

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于设置距离偏差值，该距离偏差值可以将深度图值进行整体偏。distanceOffset 为正值时，整体深度数据减小 distanceOffset mm，distanceOffset 为负值时，整体幅度数据增加 distanceOffset mm。

4.36. 获取距离偏差值 Hrg_GetDistanceOffset

函数：__API int Hrg_GetDistanceOffset(const Hrg_Dev_Handle *handle, int32_t* distanceOffset)

参数：[in] handle 该连接方式的句柄

[out] distanceOffset 幅度值阈值

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于获取设置的距离偏差值。

4.37. 设置距离范围 Hrg_SetDepthRange

函数：__API int Hrg_SetDepthRange(const Hrg_Dev_Handle *handle,

const uint32_t minDepthRange,

const uint32_t maxDepthRange)

参数：[in] handle 该连接方式的句柄

[in] minDepthRange 最小距离值，单位 mm

[in] maxDepthRange 最大距离值，单位 mm

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于设置的距离范围值。

4.38. 获取距离范围 Hrg_GetDepthRange

函数：__API int Hrg_GetDepthRange(const Hrg_Dev_Handle *handle,

uint32_t* minDepthRange,

uint32_t* maxDepthRange)

参数：[in] handle 该连接方式的句柄

[out] minDepthRange 最小距离值，单位 mm

[out] maxDepthRange 最大距离值，单位 mm

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于获取设置的距离范围值。

4.39. 设置多参数 Hrg_SetAllParameters

函数：__API int Hrg_SetAllParameters(const Hrg_Dev_Handle *handle, const Hrg_Parameters* para)

参数：[in] handle 该连接方式的句柄

[in] para 设置的多参数

返回值：0 表示成功，-1 表示失败。

说明：

该接口用于设置的多参数，具体参数见 Hrg_Parameters 结构体描述。

4.40. 获取多参数 Hrg_GetAllParameters

函数：__API int Hrg_GetAllParameters(const Hrg_Dev_Handle *handle, Hrg_Parameters* para)

参数：[in] handle 该连接方式的句柄

[out] para 获取设置的多参数

回值：0 表示成功，-1 表示失败。

说明：

该接口用于获取设置的多参数，具体参数见 Hrg_Parameters 结构体描述。

4.41. 从相机下载文件 Hrg_DownloadFile

函数：__API int Hrg_DownloadFile(const Hrg_Dev_Handle *handle, const char* filename)

参数：[in] handle 该连接方式的句柄

[in] filename 下载的文件名

回值：0 表示成功，-1 表示失败。

说明：

该接口用于下载相机中的相关配置文件。

4.42. 上传文件带相机 Hrg_UploadFile

函数：__API int Hrg_UploadFile(const Hrg_Dev_Handle *handle, const char *filename)

参数：[in] handle 该连接方式的句柄

[in] filename 上传的文件名

回值：0 表示成功，-1 表示失败。

说明：

该接口用于上传相关配置文件到相机。

4.43. 读相机寄存器值 Hrg_ReadRegister

函数：__API int Hrg_ReadRegister(const Hrg_Dev_Handle *handle,

const uint16_t *address,

uint16_t *data,

const uint32_t registerCount)

参数：[in] handle 该连接方式的句柄

[in] address 寄存器地址列表

[out] data 获取的对应寄存器值列表

[in] registerCount 寄存器数量

回值：0 表示成功，-1 表示失败。

说明：

该接口用于读相机多个寄存器的值。

4.44. 写相机寄存器值 Hrg_WriteRegister

函数：__API int Hrg_WriteRegister(const Hrg_Dev_Handle *handle,

const uint16_t *address,

const uint16_t *data,

const uint32_t registerCount)

参数：[in] handle 该连接方式的句柄

[in] address 寄存器地址列表

[in] data 获取的对应寄存器值列表

[in] registerCount 寄存器数量

回值：0 表示成功，-1 表示失败。

说明：

该接口用于写相机多个寄存器的值。

4.45. 获取相机内部参数值 Hrg_GetCameraExtrinsicParameters

函数：__API int Hrg_GetCameraExtrinsicParameters(const Hrg_Dev_Handle *handle,

Hrg_Internal_Parameters *extrinsicPara)

参数：[in] handle 该连接方式的句柄

[in] extrinsicPara 相机参数

回值：0 表示成功，-1 表示失败。

说明：

该接口用于获取相机内部参数值，具体见结构体 Hrg_Internal_Parameters。

4.46. 设置图像翻转镜像 Hrg_SetFlipMirror

函数：__API int Hrg_SetFlipMirror(const Hrg_Dev_Handle *handle, const Hrg_Flip_Mirror flipMirror)

参数：[in] handle 该连接方式的句柄

[in] flipMirror 翻转镜像

回值：0 表示成功，-1 表示失败。

说明：

该接口用于设置图像翻转镜像等功能，具体见枚举类型 Hrg_Flip_Mirror。