

力反馈设备 HFD SDK 调用说明

所需文件：

头文件： HFD_OPEN.h, hfdVector.h, hfdDefines.h

库文件： HFD_API.lib

DLL 文件： HFD_API.dll

如果采用**静态调用**动态链接库的方式，则需要先在项目中把.lib 文件所在的路径以及.h 文件所在的路径添加到项目的“附加库目录”以及“附加包含目录”，并在“附加依赖项”中添加相应的 lib 文件；

第一步：调用 `int hfdOpen(void)`函数

函数描述：打开一个力反馈设备的连接

函数返回：1-调用成功，0-调用失败

第二步：调用 `int hfdInit(char ID /*= -1*/)`函数

函数描述：初始化设备

输入参数：默认为-1，可不输入参数

函数返回：1-调用成功，0-调用失败

第三步：调用 `int hfdCalibrateDevice(char ID/*= -1*/)`函数

函数描述：从芯片读取设备校准数据，完成设备校准

输入参数：默认为-1，可不输入参数

函数返回：1-调用成功，0-调用失败

前提：设备在上电后，已经通过 HFDDLL_Test 软件完成了校准操作，并点击了 Save Calibration 按钮，将校准数据保存于芯片中

执行完第三步后，就可以通过调用函数 `hfdGetPosition()`正确获取设备**末端点的位置**以及通过调用函数 `hfdGetOrientationFrame()`正确获取**手柄的姿态**了。但是为了能够持续获取设备末端点的位置以及手柄的姿态，需要**创建一个线程**，在线程中持续调用函数 `hfdGetPosition()`与函数 `hfdGetOrientationFrame()`；

第四步：调用 `hfdGetPosition(double *px, double *py, double *pz, char ID)`函数

函数描述：获取设备末端点的位置

函数参数：

px：X 轴位置分量，左右方向为 X 轴，正方向为水平向右

py：Y 轴位置分量，上下方向为 Y 轴，正方向为竖直向上

pz：Z 轴位置分量，前后方向为 Z 轴，正方向为垂直向外

ID：设备 ID，默认为-1，可不输入参数

函数返回：1-调用成功，0-调用失败

第五步：调用 `int hfdGetOrientationFrame(double matrix[3][3], char ID)`函数

函数描述：获取设备手柄的姿态

函数参数：

matrix[3][3]，3*3 的矩阵，用来接收末端姿态

ID：设备 ID，默认为-1，可不输入参数

函数返回：1-调用成功，0-调用失败

此时，由于电机阻尼的存在，用户在移动设备手柄时会感觉有粘滞阻力，为了能有重力补偿以及消除粘滞阻力，需要进行如下设置。

第六步：调用 `int hfdEnableForce(unsigned char val, char ID)`函数使能设备力矩模式

函数描述：使能设备的力矩模式

函数参数：

val：取 HFD_ON，打开（HFD_ON 在 hfdDefines.h 文件中定义）

ID：设备 ID，默认为-1，可不输入参数

函数返回：1-调用成功，0-调用失败

第七步：调用 `int hfdEnableExpertMode(void)` 函数启用专家模式，以使后续调用 `hfdEnableDevice()`函数能够成功

函数描述：使能专家模式

函数返回：1-调用成功，0-调用失败

第八步：调用 `int hfdEnableDevice(bool val, char ID)` 函数**使能设备**，`val` 参数设置为 `true`

函数描述：使能或失能力反馈设备

函数参数：

`val`: `true`-使能力反馈设备, `false`: 失能力反馈设备

`ID`: 设备 ID, 默认为-1, 可不输入参数

函数返回：1-调用成功, 0-调用失败

Note：在调用 `hfdEnableDevice()` 函数前需要调用 `hfdEnableExpertMode()` 开启专家模式, 在调用完 `hfdEnableDevice()` 函数后需要调用 `hfdDisableExpertMode()` 关闭专家模式;

第九步：调用 `int hfdSetGravityCompensation(int val, char ID)` 函数开启重力补偿功能

函数描述：开启或者关闭重力补偿功能

函数参数：

`val`: `HFD_ON`-开启重力补偿, `HFD_OFF`-关闭重力补偿

`ID`: 设备 ID, 默认为-1, 可不输入参数

函数返回：1-调用成功, 0-调用失败

执行完第九步后, 设备将具有重力补偿功能, 能够在运动空间内任意一点悬停, 用户可自由拖动设备末端点运动。

第十步：调用 `int hfdSetForce(double fx, double fy, double fz, char ID)` 函数使设备输出反馈力

函数描述：用来在 X,Y,Z 三个方向输出反馈力

函数参数：

`fx`: X 方向输出的力, 左右方向为 X 轴, 正方向为水平向右

`fy`: Y 方向输出的力, 上下方向为 Y 轴, 正方向为竖直向上

`fz`: Z 方向输出的力, 前后方向为 Z 轴, 正方向为垂直向外

`ID`: 设备 ID, 默认为-1, 可不输入参数

函数返回：1-调用成功, 0-调用失败

第十一步：调用 `int hfdEnableDevice(bool val, char ID)` 函数**失能设备**，`val` 参数设置为 `false`

函数描述：使能或失能力反馈设备

函数参数：

`val`: `true`-使能力反馈设备，`false`: 失能力反馈设备

`ID`: 设备 ID，默认为-1，可不输入参数

函数返回：1-调用成功，0-调用失败

Note：在调用 `hfdEnableDevice()` 函数前需要调用 `hfdEnableExpertMode()` 开启专家模式，在调用完 `hfdEnableDevice()` 函数后需要调用 `hfdDisableExpertMode()` 关闭专家模式；

第十二步：调用 `int hfdClose(char ID)` 函数关闭设备

函数描述：关闭力反馈设备

函数参数：

`ID`: 设备 ID，默认为-1，可不输入参数

函数返回：1-调用成功，0-调用失败

其他可以加入之前创建的线程中的函数，分别可以用于获取不同的数据：

`int hfdGetButton(int index, char ID)`

函数描述：用于获取末端手柄上按钮的状态

函数参数：

`index`: 按钮索引，默认取 0

`ID`: 设备 ID，默认为-1，可不输入参数

函数返回：1-调用成功，0-调用失败

`int hfdGetLinearVelocity(double *vx, double *vy, double *vz, char ID)`

函数描述：获取末端点在笛卡尔坐标系中的线性速度

函数参数：

`vx`: X 方向速度，左右方向为 X 轴，正方向为水平向右

`vy`: Y 方向速度，上下方向为 Y 轴，正方向为竖直向上

`vz`: Z 方向速度，前后方向为 Z 轴，正方向为水平向外

`ID`: 设备 ID，默认为-1，可不输入参数

函数返回：1-调用成功，0-调用失败

int hfdGetOrientationRad(double *oa, double *ob, double *og, char ID)

函数描述：获取末端旋转三自由度各关节的角度[rad]

函数参数：

oa：绕 Yaw 轴的旋转角度

ob：绕 Pitch 轴的旋转角度

og：绕手柄自身轴(Roll 轴)的旋转角度

ID：设备 ID，默认为-1，可不输入参数

函数返回：1-调用成功，0-调用失败

int hfdGetJointAngles(double j[HFD_MAX_DOF],char ID)

函数描述：获取设备各关节的角度[rad]

函数参数：

J[HFD_MAX_DOF]:用于存放关节角度的数组, HFD_MAX_DOF 在 hfdDefines.h

文件中定义

ID：设备 ID，默认为-1，可不输入参数

函数返回：1-调用成功，0-调用失败

其他 API 函数在 HFD_OPEN.h 文件中，现不作一一说明