



Null Touch CSDK Reference

V1.0



HuanTech.com

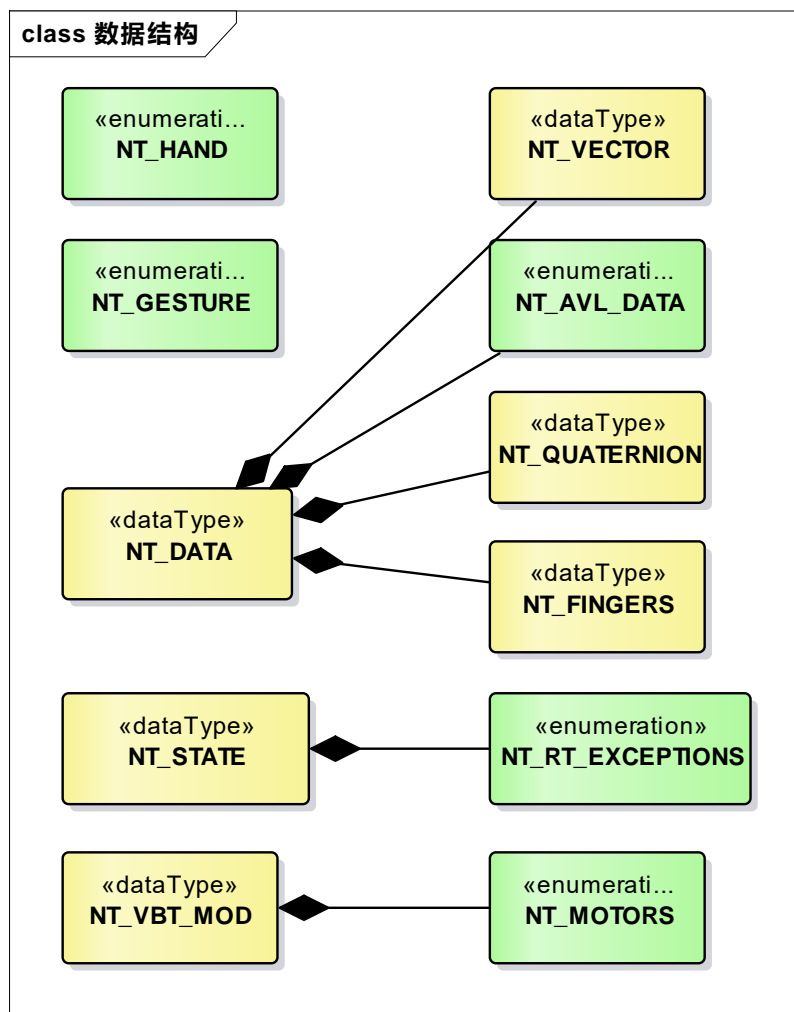
修订历史记录

日期	版本	说明
2018/2/5	1.0	文档格式版面修改

目录

Null Touch CSDK Reference	1
修订历史记录	2
目录	3
一、 数据结构	4
NT_RET_STATE	5
NT_FILE_OPERATE	5
NT_ERR_CNN	5
NT_QUATERNION	5
NT_VECTOR	6
NT_FINGERS	6
NT_RT_EXCEPTIONS	6
NT_STATE	7
NT_TYPE	7
NT_DATA	7
NT_MOTORS	8
NT_VBT_MOD	8
NT_HAND	8
NT_GESTURE	8
NT_BCONNECT_STATE	9
NT_CALIBRATE_STATE	9
二、 宏定义	9
三、 API	9
1. 概述	9
2. API 的详细描述	11
NT_ReadFlexConfig	11
NT_WriteConfig	11
NT_ConnectGlove	11
NT_GetState	12
NT_CalibrateFlex	12
NT_GetData	13
NT_SetVibration	13
NT_Exit	13
NT_GetCalibrateFlexState	13
NT_GetVersion	14

一、 数据结构



数据结构的组合关系(待更新)

NT_RET_STATE

描述：Null Touch SDK 成功、失败状态定义。

```
typedef enum {  
    NT_TRUE = 0, //成功  
    NT_FALSE = 1, //失败  
} NT_RET_STATE;
```

NT_FILE_OPERATE

描述：Null Touch 文件操作结果。

```
typedef enum {  
    NT_FILE_OK = 0,  
    NT_FILE_ERR = 1,  
    NT_NO_PERMISSION = 2,  
    NT_NO_CALIBREATION_VALUE = 3,  
    NT_NO_WRITE = 4,  
} NT_FILE_OPERATE;
```

NT_ERR_CNN

描述：Null Touch 连接错误码定义。

```
typedef enum {  
    NT_TWO_CNN_OK = 0,  
    NT_NO_BLUE_HOST = 10,  
    NT_ONE_HOST_AND_NO_CNN = 101,  
    NT_ONE_HOST_AND_CNN_LEFT = 102,  
    NT_ONE_HOST_AND_CNN_RIGHT = 103,  
    NT_TWO_HOST_AND_NO_CNN = 200,  
    NT_TWO_HOST_AND_CNN_LEFT = 201,  
    NT_TWO_HOST_AND_CNN_RIGHT = 202,  
} NT_ERR_CNN;
```

NT_QUATERNION

描述：四元数代表一个朝向。

```
typedef struct {  
    float w, x, y, z;  
} NT_QUATERNION;
```

NT_VECTOR

描述：三维向量。

```
typedef struct {  
    float x, y, z;  
} NT_VECTOR;
```

NT_FINGERS

描述：手指的弯曲度，标准化为 0-1 的范围。手指的弯曲程度在校准后按比例输出 0-1 的数据，接近 0 表示手指张开，接近 1 表示手指最大弯曲。

```
typedef struct {  
    float thumb, index, middle, ring, pinky;  
} NT_FINGERS;
```

NT_RT_EXCEPTIONS

描述：手套的异常的定义，枚举名称最后字段的 5 个二进制字符含义定义为按 bit 对应 5 个手指，从左至右 bit5-bit0 分别对应拇指、食指、中指、无名指、小指。

```
typedef enum {  
    NTEXP_FLX_BROKE_10000 = 0x10,    //弯曲传感器脱落, 拇指  
    NTEXP_FLX_BROKE_01000 = 0x08,    //弯曲传感器脱落, 食指  
    NTEXP_FLX_BROKE_00100 = 0x04,    //弯曲传感器脱落, 中指  
    NTEXP_FLX_BROKE_00010 = 0x02,    //弯曲传感器脱落, 无名指  
    NTEXP_FLX_BROKE_00001 = 0x01,    //弯曲传感器脱落, 小指  
    NTEXP_FLX_BROKE_11000 = 0x18,    //弯曲传感器脱落, 拇指、食指  
    NTEXP_FLX_BROKE_10100 = 0x14,    //弯曲传感器脱落, 拇指、中指  
    NTEXP_FLX_BROKE_10010 = 0x12,    //弯曲传感器脱落, 拇指、无名指  
    NTEXP_FLX_BROKE_10001 = 0x11,    //弯曲传感器脱落, 拇指、小指  
    NTEXP_FLX_BROKE_01100 = 0x0C,    //弯曲传感器脱落, 食指、中指  
    NTEXP_FLX_BROKE_01010 = 0x0A,    //弯曲传感器脱落, 食指、无名指  
    NTEXP_FLX_BROKE_01001 = 0x09,    //弯曲传感器脱落, 食指、小指  
    NTEXP_FLX_BROKE_00110 = 0x06,    //弯曲传感器脱落, 中指、无名指  
    NTEXP_FLX_BROKE_00101 = 0x05,    //弯曲传感器脱落, 中指、小指  
    NTEXP_FLX_BROKE_00011 = 0x03,    //弯曲传感器脱落, 无名指、小指  
    NTEXP_FLX_BROKE_11100 = 0x1C,    //弯曲传感器脱落, 拇指、食指、中指  
    NTEXP_FLX_BROKE_11010 = 0x1A,    //弯曲传感器脱落, 拇指、食指、无名指  
    NTEXP_FLX_BROKE_11001 = 0x19,    //弯曲传感器脱落, 拇指、食指、小指  
    NTEXP_FLX_BROKE_10110 = 0x16,    //弯曲传感器脱落, 拇指、中指、无名指  
    NTEXP_FLX_BROKE_10101 = 0x15,    //弯曲传感器脱落, 拇指、中指、小指  
    NTEXP_FLX_BROKE_10011 = 0x13,    //弯曲传感器脱落, 拇指、无名指、小指  
    NTEXP_FLX_BROKE_01110 = 0x0E,    //弯曲传感器脱落, 食指、中指、无名指
```

```
NTEXP_FLX_BROKE_01101 = 0x0D,    //弯曲传感器脱落, 食指、中指、小指
NTEXP_FLX_BROKE_01011 = 0x0B,    //弯曲传感器脱落, 食指、无名指、小指
NTEXP_FLX_BROKE_00111 = 0x07,    //弯曲传感器脱落, 中指、无名指、小指
NTEXP_FLX_BROKE_11110 = 0x1E,    //弯曲传感器脱落, 拇指、食指、中指、无名指
NTEXP_FLX_BROKE_10111 = 0x17,    //弯曲传感器脱落, 食指、中指、无名指、小指
NTEXP_FLX_BROKE_11011 = 0x1B,    //弯曲传感器脱落, 拇指、食指、无名指、小指
NTEXP_FLX_BROKE_11101 = 0x1D,    //弯曲传感器脱落, 拇指、食指、中指、小指
NTEXP_FLX_BROKE_01111 = 0x0F,    //弯曲传感器脱落, 食指、中指、无名指、小指
NTEXP_FLX_BROKE_11111 = 0x1F,    //弯曲传感器全部脱落, 拇指、食指、中指、无名指、小指
} NT_RT_EXCEPTIONS;
```

NT_STATE

描述: 动捕手套的状态, 表示一个手套。

```
typedef struct {
    int bConnectState; // 表示手套的连接状态
    int bSelfTest;     // 表示手套自检状态
    int eException;     // 对应 NT_RT_EXCEPTIONS 枚举
} NT_STATE;
```

NT_TYPE

描述: 定义请求数据的类型。

```
typedef enum {
    NT_QUATERNION_FLEX = 0, //四元数+弯曲数据
    NT_EULER_FLEX,         //四元数+弯曲数据
    NT_ACCELERATION,       //加速度数据
    NT_GYRO,                //陀螺仪数据
    NT_MAGNETIC             //磁力计数据
} NT_TYPE;
```

NT_DATA

描述: 动捕手套的数据。

```
typedef struct {
    unsigned int packetNumber;
    int eValueData; //对应 NT_TYPE 枚举类型
    NT_QUATERNION quaternion;
    NT_VECTOR euler;
    NT_FINGERS fingers;
    NT_VECTOR acc;
```

```
NT_VECTOR gyro;  
NT_VECTOR mag;  
} NT_DATA;
```

NT_MOTORS

描述：制定震动电机的枚举，可制定多个电机，以相同的模式进行震动。NTM_THUMB = 0, NTM_INDEX = 1, NTM_PALM = 2.....

```
typedef enum {  
    NTM_THUMB = 0,  
    NTM_INDEX,  
    NTM_PALM,  
    NTM_THUMB_INDEX,  
    NTM_THUMB_PALM,  
    NTM_INDEX_PALM,  
    NTM_ALL,  
} NT_MOTORS;
```

NT_VBT_MOD

描述：震动模式参数配置结构体。

```
typedef struct {  
    int eMotors;//需要震动的电机枚举，对应 NT_MOTORS  
    unsigned int strength;//振动强度值为电机 PWM 的占空比，范围 0-100;  
    unsigned int time;//振动时间 = 时间值 * 2ms，时间值范围 15-250  
} NT_VBT_MOD;
```

NT_HAND

描述：手的枚举，区分左右手。

```
typedef enum {  
    NT_LEFT,  
    NT_RIGHT,  
    NT_DOUBLE_HAND = 3,    //双手  
} NT_HAND;
```

NT_GESTURE

描述：校准手势枚举，区分握拳和张开五指。

```
typedef enum {
```



```
NT_NONE ,  
NT_ROCK,  
NT_PAPER,  
NT_ROCK_AND_PAPER,  
} NT_GESTURE;
```

NT_BCONNECT_STATE

描述：自检返回状态值枚举。

```
typedef enum {  
    NT_CONNECTED = 0,           //连接状态  
    NT_CONNECTING_FALSE = 1,    //未连接上状态  
    NT_CONNECTED_FALSE = 2,     //连接异常中断后的状态  
    NT_CONNECTED_TWO_TRUE = 3,  //连接一场中断后重新连接上的状态  
} NT_BCONNECT_STATE;
```

NT_CALIBRATE_STATE

描述：返回弯曲校准后状态值枚举。

```
typedef enum {  
    NT_CALIBRATE_TRUE = 0,      //成功  
    NT_CALIBRATE_FALSE = 1,    //失败  
} NT_CALIBRATE_STATE;
```

二、宏定义

```
#define NT_API __declspec(dllexport)
```

三、API

1. 概述

接口	说明
NT_ReadFlexConfig	打开配置文件，读取弯曲传感器的参数
NT_ConnectGlove	打开蓝牙模块，查询连接结果
NT_CalibrateFlex	对弯曲传感器校准运行参数
NT_WriteFlexConfig	生成配置文件，存储弯曲传感器的参数

NT_GetData	获取手套的运动数据
NT_GetState	获取设备的状态
NT_SetVibration	让动捕手套制定的震动电机震动
NT_Exit	释放动捕手套占用的系统资源，包括共享内存、关闭串口等
NT_GetCalibrateFlexState	获取弯曲校准后的状态等
NT_GetVersion	获取 SDK 版本号

2. API 的详细描述

NT_ReadFlexConfig

声明: NT_API NT_FILE_OPERATE NT_ReadFlexConfig(void);

描述: 打开配置文件, 读取弯曲传感器的参数。

参数: 无。

返回:

NT_FILE_OK	成功
NT_FILE_ERR	文件不存在
NT_NO_PERMISSION	没有文件操作权限

NT_WriteConfig

声明: NT_API NT_FILE_OPERATE NT_WriteConfig (void);

描述: 生成配置文件, 存储弯曲传感器的参数。

参数: 无。

返回:

NT_FILE_OK	成功
NT_FILE_ERR	文件打开失败
NT_NO_PERMISSION	没有文件操作权限

说明: 对于 Unity, 此接口隐藏。

NT_ConnectGlove

声明: NT_API NT_ERR_CNN NT_ConnectGlove(void);

描述: 打开蓝牙模块, 查询连接结果。可重复被调用。

参数: 无。

返回:

NT_TWO_CONN_OK	打开了两蓝牙模块, 且连接上双手的动捕手套
NT_NO_BLUE_HOST	没有扫描到蓝牙模块
NT_ONE_HOST_AND_NO_CONN	只打开了一个蓝牙模块, 未连接上动捕手套
NT_ONE_HOST_AND_CONN_LEFT	只打开了一个蓝牙模块, 连接上了左手的动捕手套
NT_ONE_HOST_AND_CONN_RIGHT	只打开了一个蓝牙模块, 连接上了右手的动捕手套
NT_TWO_HOST_AND_NO_CONN	打开了两蓝牙模块, 但都没连接上动捕手套
NT_TWO_HOST_AND_CONN_LEFT	打开了两蓝牙模块, 连接上左手的动捕手套
NT_TWO_HOST_AND_CONN_RIGHT	打开了两蓝牙模块, 连接上右手的动捕手套

NT_GetState

```
声明: NT_API NT_RET_STATE NT_GetState(NT_HAND hand, NT_TYPE dataType,  
NT_STATE* const gloveState);
```

描述: 获取设备的状态。

参数:

NT_HAND 指定左右手，
NT_TYPE 描述请求数据的类型，
NT_STATE 存放手套的状态信息。

返回:

NT_TRUE	获取状态成功
NT_FALSE	获取状态失败

NT_CalibrateFlex

```
声明: NT_API NT_RET_STATE NT_CalibrateFlex(NT_HAND, NT_GESTURE);
```

描述: 对弯曲传感器校准运行参数。

参数:

NT_HAND 指明是左手或是右手；
NT_GESTURE 指明校准时做出的动作。

返回:

NT_TRUE	成功
NT_FALSE	失败

说明: 对于 Unity，此接口隐藏。

NT_GetData

```
声明: NT_API NT_RET_STATE NT_GetData(NT_HAND hand, NT_TYPE type,  
NT_DATA* const gloveData);
```

描述: 获取手套的运动数据。

参数:

NT_HAND 指明是左手或是右手,

NT_DATA 是姿态数据和状态数据。

返回:

NT_NORMAL	成功
NT_CRC_FAIL	数据连续多次 CRC 校验失败
NT_DEV_DISCNN	有设备断开连接, 断开设备的数据均设置为零

NT_SetVibration

```
声明: NT_API NT_RET_STATE NT_SetVibration(NT_HAND, const NT_VBT_MOD * const  
mVibrationMode);
```

描述: 让动捕手套制定的震动电机震动。

参数: NT_HAND 指明是左手或是右手, NT_VBT_MOD 表明震动的模式。

返回:

NT_TRUE	成功
NT_FALSE	失败

NT_Exit

```
声明: NT_API NT_RET_STATE NT_Exit(void);
```

描述: 释放动捕手套占用的系统资源, 包括共享内存、关闭串口等。

参数: 无。

返回:

NT_TRUE	正常释放资源成功
NT_FALSE	资源释放失败

NT_GetCalibrateFlexState

```
声明: NT_API NT_CALIBRATE_STATE NT_GetCalibrateFlexState(NT_HAND hand,  
NT_GESTURE Gesture);
```

描述: 获取弯曲校准后的状态等。

参数: 无。

返回:

NT_CALIBRATE_TRUE	弯曲校准成功
NT_CALIBRATE_FALSE	弯曲校准失败

NT_GetVersion

声明: NT_API **DWORD** NT_GetVersion(void);

描述: 获取 SDK 版本号。

参数: 无。

返回:

返回双字型, 4字节, 高位word表示主版本号, 低位word表示子版本号。如“0200”表示v02.00。