

北京和利时电机技术有限公司保留所有权力。

说明书的内容参照了相关法律基准和行业基准。使用产品时，如对本说明书提供的内容有疑问，请向购买产品的销售人员咨询，或致电客户服务热线，或致信本公司邮箱。

由于产品的不断更新升级，和利时电机保留在不事先通知的情况下，修改本手册中的产品和产品规格参数等文件的权利，提示客户请使用最新版本的说明书。

和利时电机具有本产品及其软件的专利权、版权和其它知识产权。未经许可，不得直接或者间接地复制、制造、加工、使用本产品及其相关部分。

和利时电机具有本说明书的著作权，未经许可，不得修改、复制说明书的全部或部分内容。

版本号 03/2016

目录

第一章 参数	- 3 -
1.1. 平台选择.....	- 3 -
1.2. 游戏参数.....	- 3 -
1.3. 六轴的位置信息.....	- 4 -
1.4. 体感参数.....	- 4 -
1.5. 平台的机械参数.....	- 5 -
1.6. UDP 端口号和采样时间	- 7 -
第二章 函数	- 8 -
2.1. 函数的返回值.....	- 8 -
2.2. 平台类型选择函数.....	- 8 -
2.3. 体感参数复位函数.....	- 8 -
2.4. 参数传递函数.....	- 8 -
2.5. 机械参数初始化函数.....	- 8 -
2.6. 打开 UDP 端口函数	- 8 -
2.7. 体感+反解算法函数	- 9 -
2.8. 平台复位函数函数.....	- 9 -
2.9. 平台到达中位高度函数.....	- 9 -
2.10. 平台到达零位函数.....	- 9 -
2.11. 关闭 UDP 端口函数	- 9 -
2.12. 函数调用顺序.....	- 9 -

第一章 参数

1.1. 平台选择

SIM_CUE_DLL.dll 动态链接库集合了六自由度、三加一旋转轴四自由度、三自由度三种平台的体感算法以及直接进行电动缸位置控制。可以通过调用函数来实现平台类型的选择平台类型和控制类型。

函数 `int __stdcall Choose_PlatformType(int PlatformType);` 可以设置平台的类型。

PlatformType 传入数值为 0x00 表示 3+1 轴平台，也可用于 3 轴平台；传入数值 0x01 表示 6 轴平台。

1.2. 游戏参数

游戏参数如下图所示

```
//游戏参数
typedef struct {
    //游戏中飞机实时的运动值
    float F32ReciveAlfaRad; /* 单位：弧度*/
    float F32ReciveBetaRad; /* 单位：弧度*/
    float F32ReciveGammaRad; /* 单位：弧度*/
    float F32ReciveXAcceG; /* 单位：重力加速度*/
    float F32ReciveYAcceG; /* 单位：重力加速度*/
    float F32ReciveZAcceG; /* 单位：重力加速度*/
}DOF6_GAME_PARA;
```

此六个参数为实时参数，是根据游戏中运动实体的实时姿态信息来设置。在游戏中有坐标系 XYZ，如下图所示。

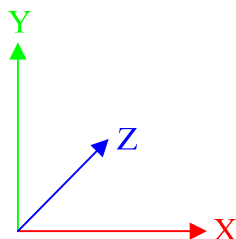


图 1 游戏坐标系

以赛车为例，车体绕 X 轴的旋转角度是 F32ReciveAlfaRad，绕 Y 轴的旋转角度是 F32ReciveBetaRad，绕 Z 轴的旋转角度是 F32ReciveGammaRad，沿着 X 轴的加速度是 F32ReciveXAcceG，绕着 Y 轴的加速度是 F32ReciveYAcceG，绕着 Z 轴的加速度是 F32ReciveZAcceG。车在运动的过程中，旋转角度和加速度是实时变化的，因此需要根据游戏的信息实时更新游戏参数。

注意：

当为六自由度平台时，F32ReciveAlfaRad、F32ReciveBetaRad、F32ReciveGammaRad、F32ReciveXAcceG、F32ReciveYAcceG、F32ReciveZAcceG 六个值都存在。

当为 3+1 轴平台时，只有 F32ReciveAlfaRad、F32ReciveBetaRad、F32ReciveGammaRad、F32ReciveYAcceG 四个值存在。

参数

当为 3 轴平台时，F32ReciveAlfaRad、F32ReciveGammaRad、F32ReciveYAcceG 三个值存在。

以上是因平台的机械结构导致平台缺少某个或某几个方向的旋转和平移。

1.3. 六轴的位置信息

经过体感算法和反解计算之后输出的六个轴的位置信息如下图。

```
//输出六个轴的位置信息
typedef struct {
    //六个缸的位置信息
    long int I32PlayXpos; //X轴位置信息
    long int I32PlayYpos; //Y轴位置信息
    long int I32PlayZpos; //Z轴位置信息
    long int I32PlayUpos; //U轴位置信息
    long int I32PlayVpos; //V轴位置信息
    long int I32PlayWpos; //W轴位置信息
}DOF6_POSITION_PARA;
```

经体感+反解算法函数运算会输出以上六个数值，依次对应电动缸的 1~6 号轴的位置信息。六个缸的位置信息是电机的脉冲数。

1.4. 体感参数

体感参数如下图。

```
// XYZ_LINE_ACC
typedef struct { float F32T1; /* 时间因子 T1 */
float F32T2; /* 时间因子 T2 */
float F32T3; /* 时间 T3 */
float F32T4; /* 时间因子 T4 */
float F32T5; /* 时间因子 T5 */
float F32T6; /* 时间因子 T6 */
float F32T7; /* 时间因子 T7 */
float F32T8; /* 时间因子 T8 */

float F32C1; /* 加速度阈值 1 */
float F32C2; /* 加速度阈值 2 */
float F32C3; /* 加速度阈值 3 */
float F32C4; /* 加速度阈值 4 */
float F32C5; /* 加速度阈值 5 */
float F32C6; /* 加速度阈值 6 */
float F32C7; /* 加速度阈值 7 */
float F32C8; /* 加速度阈值 8 */

float F32K1; /* 比例因子 1 */
float F32K2; /* 比例因子 2 */
float F32K3; /* 比例因子 3 */
float F32K4; /* 比例因子 4 */
float F32K5; /* 比例因子 5 */
float F32K6; /* 比例因子 6 */
float F32K7; /* 比例因子 7 */
float F32K8; /* 比例因子 8 */
} DOF6_CUE_PARA;
```

体感参数初始是可按默认值来赋值，待调试平台时根据平台的动态效果来调整这些参数。

1.5. 平台的机械参数

平台的机械参数如下图所示。

```
//平台的机械参数
typedef struct {

    //游戏中飞机的最大运动幅度
    float F32AlfaMaxRad; /* 单位: 弧度 */
    float F32BetaMaxRad; /* 单位: 弧度 */
    float F32GammaMaxRad; /* 单位: 弧度 */
    float F32XAcceMaxG; /* 单位: 重力加速度 */
    float F32YAcceMaxG; /* 单位: 重力加速度 */
    float F32ZAcceMaxG; /* 单位: 重力加速度 */

    //三自由度平台
    float F32TriangleHemMm; /* 三角形底边长度, 单位: 毫米 */
    float F32HeightOfTriangleHemMm; /* 三角形底边高, 单位: 毫米 */
    float F32HeightToTopMarkMm; /* 三角形顶点到座椅中心点的距离, 单位: 毫米 */

    //六自由度平台上下平面的尺寸
    float F32StaticShortMm; /* 下平台短边, 单位: 毫米 */
    float F32StaticDiameterMm; /* 下平台直径, 单位: 毫米 */
    float F32MovingShortMm; /* 上平台短边, 单位: 毫米 */
    float F32MovingDiameterMm; /* 上平台直径, 单位: 毫米 */

    //相对于初始坐标的旋转量
    float F32XAxisRotAngleDeg; /* 坐标旋转角度 */

    //平台机械结构决定其的运动范围
    float F32PlatformAlfaMaxDeg; /* Alfa最大运动范围, 单位: 度 */
    float F32PlatformBetaMaxDeg; /* Beta最大运动范围, 单位: 度 */
    float F32PlatformGammaMaxDeg; /* Gamma最大运动范围, 单位: 度 */
    float F32PlatformXMaxMm; /* X平移最大运动范围, 单位: 毫米 */
    float F32PlatformYMaxMm; /* Y平移最大运动范围, 单位: 毫米 */
    float F32PlatformZMaxMm; /* Z平移最大运动范围, 单位: 毫米 */

    //电动缸的参数
    float F32AccessDistanceUnitMm; /* 电动缸的行程, 单位: 毫米 */
    float F32LeadDistanceUnitMm; /* 电动缸的导程, 单位: 毫米 */
    float F32MinLongofCylinderMm; /* 电动缸的最小长度, 单位: 毫米 */

    //齿轮比
    float F32GearOne; /* 1号轴电子齿轮比 */
    float F32GearTwo; /* 2号轴电子齿轮比 */
    float F32GearThree; /* 3号轴电子齿轮比 */
    float F32GearFour; /* 4号轴电子齿轮比 */
    float F32GearFive; /* 5号轴电子齿轮比 */
    float F32GearSix; /* 6号轴电子齿轮比 */

}DOF6_MECH_PARA;
```

其中:

游戏中飞机的最大运动幅度是游戏中的实体如飞机或车在运动过程能绕着各轴旋转的最大角度和最大加速度。F32AlfaMaxRad 是绕着 X 轴旋转的最大角度, F32BetaMaxRad 是绕着 Y 轴旋转的最大角度, F32GammaMaxRad 是绕着 Z 轴旋转的最大角度; F32XAcceMaxG 是沿着 X 轴方向的加速度, F32YAcceMaxG 是沿着 Y 轴方向的加速度, F32ZAcceMaxG 是沿着 Z 轴方向的加速度, 其中加速度的最大值可以不进行设置。

三自由度平台的俯视图如下, 其三自由度平台参数标注如下图所示, 此参数只作用于 3+1 平台或三轴平台。

参数

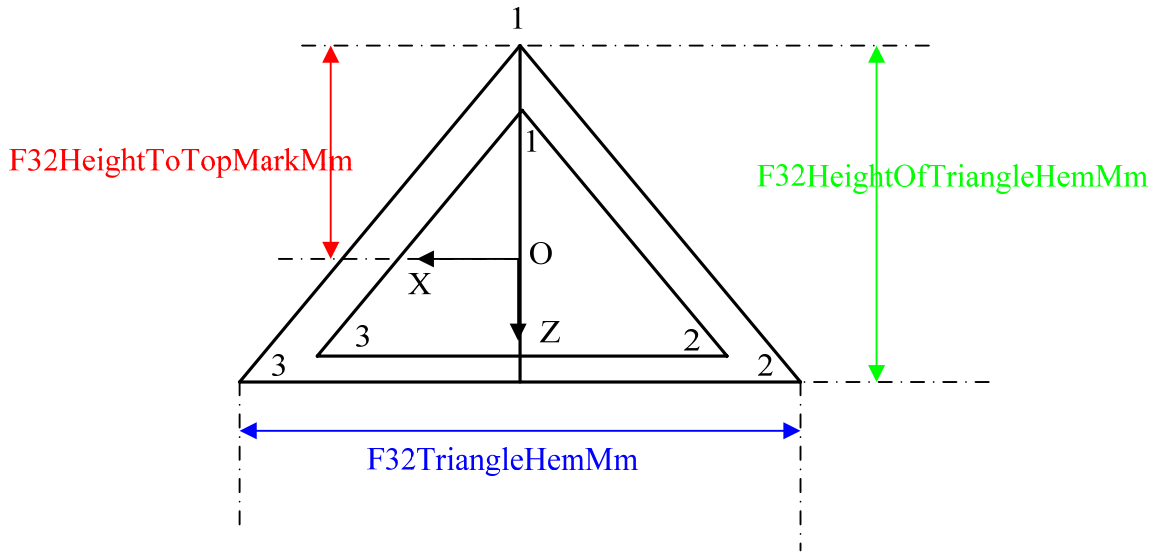


图 2 三自由度平台坐标系

六自由度平台的俯视图如下，其六自由度平台上下平面的尺寸参数标注如下图所示，此参数只作用于六自由度平台。

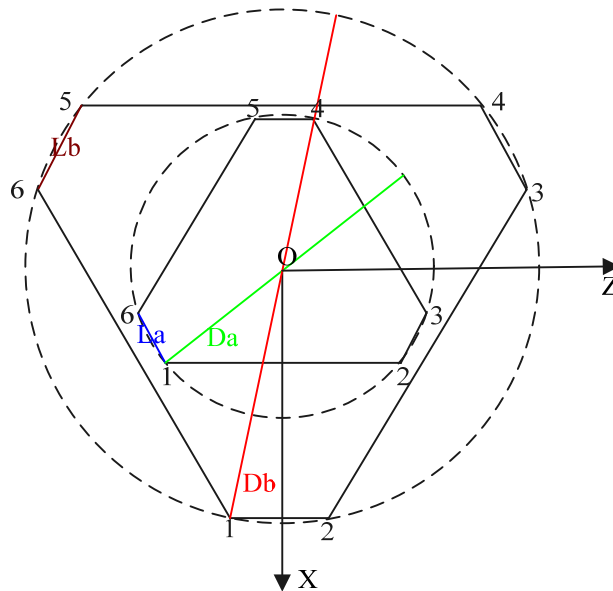


图 3 六自由度平台坐标系

如上图所示， L_b 是下平台短边 $F32StaticShortMm$ ， D_b 是下平台六个铰点的外接圆直径 $F32StaticDiameterMm$ ， L_a 是上平台短边 $F32MovingShortMm$ ， D_a 是上六个铰点的外接圆直径 $F32MovingDiameterMm$ 。

相对于初始坐标的旋转量只作用于六自由度平台。如图 3 所示，六自由度平台的初始坐标 X 轴正方向位于下平台 1、2 铰点的垂直平分线上，如需要将 X 轴正方向位于 3、4 铰点的垂直平分线上，则相当于将坐标系按逆时针方向旋转 120 度，即 $F32XaxisRotAngleDeg$ 赋值为 120。

平台机械结构决定其的运动范围，对于三自由度平台，如图 2 所示，平台绕着 X 轴旋转的最大角度是 `F32PlatformAlfaMaxDeg`，绕着 Z 轴旋转的最大角度是 `F32PlatformGammaMaxDeg`，沿着 Y 轴平移的距离是 `F32PlatformYMaxMm`。对于六自由度平台，如图 3 所示，平台绕着 X 轴旋转的最大角度是 `F32PlatformAlfaMaxDeg`，绕着 Y 轴旋转的最大角度是 `F32PlatformBetaMaxDeg`，绕着 Z 轴旋转的最大角度是 `F32PlatformGammaMaxDeg`，沿着 X 轴平移的距离是 `F32PlatformXMaxMm`，沿着 Y 轴平移的距离是 `F32PlatformYMaxMm`，沿着 Z 轴平移的距离是 `F32PlatformZMaxMm`。

电动缸的参数，`F32AccessDistanceUnitMm` 是电动缸的行程，`F32LeadDistanceUnitMm` 是电动缸的导程，`F32MinLongofCylinderMm` 电动缸最小长度是上平台落在最低高度时，上下平台两个相连的铰点之间的斜线长度。

齿轮比是电机和电动缸链接时的减速比，若无减速则为 1。

1.6. UDP 端口号和采样时间

UDP 端口号和采样时间如下图所示。

```
//UDP端口号和采样时间
typedef struct { float F32FlightSamplingPeriods; // 飞行采样时间，单位：秒*/
                short I16HostTxPort; /* 主机发送UDP端口 */
                short I16HostRxPort; /* 主机接收UDP端口 */
                short I16MboxTxPort; /* MBOX发送UDP端口 */
                short I16MboxRxPort; /* MBOX接收UDP端口 */

                short I16WhoAcceptCode ;
                short I16WhoReplyCode;
                int I16BaseDoutCode; /*特效输出
                long int I32PlayLine; /*校验输出

                }UDP_PORT_PARA;
```

`F32FlightSamplingPeriods` 是游戏数据更新的时间间隔，`I16HostTxPort`、`I16HostRxPort`、`I16MboxTxPort`、`I16MboxRxPort` 是 UDP 端口号，详细介绍可参考 MBOX 使用手册。`I16WhoAcceptCode`、`I16WhoReplyCode` 是谁接收，谁回复，详细介绍可参考 MBOX 使用手册。`I16BaseDoutCode` 为特效输出，`I32PlayLine` 为校验输出。

第二章 函数

2.1. 函数的返回值

所有函数都有返回值，返回值的定义如下：

0x20000000	函数调用成功
0x20000001	加密狗打开失败，函数调用失败
0x20000002	加密狗 ID 验证错误，函数调用失败
0x20000003	加密狗 HMAC 验证错误，函数调用失败
其他值	函数调用失败

表 1 函数返回值

2.2. 平台类型选择函数

```
int __stdcall Choose_PlatformType(int PlatformType);
```

选择要控制的平台类型，需要在程序运行初始时调用。

参数 PlatformType 为 0x00: 3+1 轴平台\3 轴平台；为 0x01: 6 轴平台。

函数的返回值是平台类型。

2.3. 体感参数复位函数

```
int __stdcall DOF6_Public_CueModule_Reset();
```

对体感参数进行复位，需要在程序运行初始时调用。

无参数，返回值如表 1 所示。

2.4. 参数传递函数

```
int __stdcall DOF6_Public_UserCueParaTranfer(DOF6_SYS_PARA *p);
```

将所有参数传递给动态链接库。

返回值如表 1 所示。

2.5. 机械参数初始化函数

```
int __stdcall DOF6_Public_MechModule_InitCa();
```

计算平台的坐标及中位高度。

无参数，返回值如表 1 所示。

2.6. 打开 UDP 端口函数

```
int __stdcall Public_OpenMboxUdpPort();
```


打开 UDP 端口。
无参数，返回值如表 1 所示。

2.7. 体感+反解算法函数

```
int __stdcall DOF6_Public_Cue2Inverse_Solution(DOF6_GAME_PARA *p, DOF6_POSITOPN_PARA *q, int UDPEnable);
```

通过 UDP 端口发送数据控制平台运动。

输入参数 DOF6_GAME_PARA *p 为游戏参数的实时数据，输出参数 DOF6_POSTION_PARA 输出六个缸的位置脉冲数，输入参数 UDPEnable 选择函数实现的功能，当 UDPEnable=1 时，函数输出六个缸的位置脉冲并送 UDP 数据控制平台运动；当 UDPEnable=0 时，函数仅输出六个缸的位置脉冲不发送 UDP 数据。函数的返回值如表 1 所示。

2.8. 平台复位函数函数

```
int __stdcall Public_ResetPlatform();
```

通过 UDP 端口发送数据控制平台复位。

无参数，返回值如表 1 所示。

2.9. 平台到达中位高度函数

```
int __stdcall LC_Public_GoMiddlePlatform(long int GoMiddleTimeMs);
```

通过 UDP 端口发送数据控制平台到达中位高度。

参数为平台到达中位高度的时间，单位是毫秒，返回值如表 1 所示。

2.10. 平台到达零位函数

```
int __stdcall Public_GoZeroPlatform(long int GoZeroTimeMs);
```

通过 UDP 端口发送数据控制平台到达零位。

参数为平台到达零点的时间，单位是毫秒，返回值如表 1 所示。

2.11. 关闭 UDP 端口函数

```
int __stdcall Public_CloseMboxUdpPort();
```

关闭 UDP 端口。

无参数，返回值如表 1 所示。

2.12. 函数调用顺序

- 1、程序运行初始调用 Choose_PlatformType(int PlatformType)函数选择控制平台的类型，然后调用 DOF6_Public_CueModule_Reset()函数，对体感参数进行初始化；
- 2、对 DOF6_SYS_PARA 结构体里的所有参数进行赋值，赋值过程中需要根据平台游戏的实际情况进行赋值；
- 3、调用 DOF6_Public_UserCueParaTranfer 进行结构参数传递；
- 4、调用 DOF6_Public_MechModule_InitCa()函数，对机械参数进行初始化；

函数

- 5、调用 `Public_OpenMboxUdpPort()`函数，打开 UDP 端口；
- 6、调用 `Public_GoMiddlePlatform(long int GoMiddleTimeMs)`函数，控制平台到达中位高度；
- 7、调用 `DOF6_Public_Cue2Inverse_Solution(DOF6_GAME_PARA *p, DOF6_POSITOPN_PARA *q, int UDPEnable)`函数，控制平台运动，此函数需要定时更新参数 `DOF6_GAME_PARA`，可将此函数放入定时器中，定时发送数据控制平台运动；
- 8、`Public_ResetPlatform()`平台复位函数、`Public_GoZeroPlatform(long int GoZeroTimeMs)`平台归零函数可以根据需要来调用；
- 9、程序关闭调用 `Public_CloseMboxUdpPort()`函数，关闭 UDP 端口。