

1. 数据结构说明

1.1. 接口调用返回值类型

```
1  typedef int errno_t;
```

1.2. 关节位置数据类型

```
1  /**
2   * @brief 关节位置数据类型
3   */
4  typedef struct
5  {
6      double jPos[6];    /* 六个关节位置, 单位deg */
7  }JointPos;
```

1.3. 笛卡尔空间位置数据类型

```
1  /**
2   * @brief 笛卡尔空间位置数据类型
3   */
4  typedef struct
5  {
6      double x;    /* x轴坐标, 单位mm */
7      double y;    /* y轴坐标, 单位mm */
8      double z;    /* z轴坐标, 单位mm */
9  } DescTran;
```

1.4. 欧拉角姿态数据类型

```
1  /**
2   * @brief 欧拉角姿态数据类型
3   */
4  typedef struct
5  {
6      double rx;    /* 绕固定轴X旋转角度, 单位: deg */
7      double ry;    /* 绕固定轴Y旋转角度, 单位: deg */
8      double rz;    /* 绕固定轴Z旋转角度, 单位: deg */
9  } Rpy;
```

1.5. 笛卡尔空间位姿数据类型

```
1  /**
2  * @brief 笛卡尔空间位姿类型
3  */
4  typedef struct
5  {
6      DescTran tran;      /* 笛卡尔空间位置 */
7      Rpy rpy;           /* 笛卡尔空间姿态 */
8  } DescPose;
```

1.6. 扩展轴位置数据类型

```
1  /**
2  * @brief 扩展轴位置数据类型
3  */
4  typedef struct
5  {
6      double ePos[4];    /* 四个扩展轴位置, 单位mm */
7  } ExaxisPos;
```

1.7. 力矩传感器数据类型

```
1  /**
2  * @brief 力传感器的受力分量和力矩分量
3  */
4  typedef struct
5  {
6      double fx; /* 沿x轴受力分量, 单位N */
7      double fy; /* 沿y轴受力分量, 单位N */
8      double fz; /* 沿z轴受力分量, 单位N */
9      double tx; /* 绕x轴力矩分量, 单位Nm */
10     double ty; /* 绕y轴力矩分量, 单位Nm */
11     double tz; /* 绕z轴力矩分量, 单位Nm */
12 } ForceTorque;
```

1.8. 螺旋参数数据类型

```
1  /**
2  * @brief 螺旋参数数据类型
3  */
4  typedef struct
5  {
6      int    circle_num;          /* 螺旋圈数 */
7      float  circle_angle;       /* 螺旋倾角 */
8      float  rad_init;           /* 螺旋初始半径, 单位mm */
9      float  rad_add;            /* 半径增量 */
10     float  rotaxis_add;         /* 转轴方向增量 */
11     unsigned int rot_direction; /* 旋转方向, 0-顺时针, 1-逆时针 */
12 }SpiralParam;
```

1.9. 控制器状态反馈数据包

在 C++SDK-v2.1.4.0 版本发生变更.

```

1  /**
2   * @brief  控制器状态反馈数据包
3   */
4  typedef struct _ROBOT_STATE_PKG
5  {
6      uint16_t frame_head;      /* 帧头, 约定为0x5A5A */
7      uint8_t  frame_cnt;      /* 帧计数, 循环计数0-255 */
8      uint16_t data_len;      /* 数据内容的长度 */
9      uint8_t  program_state;  /* 程序运行状态, 1-停止; 2-运行; 3-暂停; */
10     uint8_t  robot_state;    /* 机器人运动状态, 1-停止; 2-运行; 3-暂停; 4-拖动 */
11     int      main_code;      /* 主故障码 */
12     int      sub_code;       /* 子故障码 */
13     uint8_t  robot_mode;     /* 机器人模式, 1-手动模式; 0-自动模式; */
14     double   jt_cur_pos[6];  /* 6个轴当前关节位置, 单位deg */
15     double   tl_cur_pos[6];  /* 工具当前位置
16                                tl_cur_pos[0], 沿x轴位置, 单位mm,
17                                tl_cur_pos[1], 沿y轴位置, 单位mm,
18                                tl_cur_pos[2], 沿z轴位置, 单位mm,
19                                tl_cur_pos[3], 绕固定轴X旋转角度, 单位deg
20                                tl_cur_pos[4], 绕固定轴y旋转角度, 单位deg
21                                tl_cur_pos[5], 绕固定轴z旋转角度, 单位deg */
22     double   flange_cur_pos[6]; /* 末端法兰当前位置
23                                flange_cur_pos[0], 沿x轴位置, 单位mm,
24                                flange_cur_pos[1], 沿y轴位置, 单位mm,
25                                flange_cur_pos[2], 沿z轴位置, 单位mm,
26                                flange_cur_pos[3], 绕固定轴X旋转角度, 单位deg
27                                flange_cur_pos[4], 绕固定轴y旋转角度, 单位deg
28                                flange_cur_pos[5], 绕固定轴z旋转角度, 单位deg */
29     double   actual_qd[6];    /* 当前6个关节速度, 单位deg/s */
30     double   actual_qdd[6];   /* 当前6个关节加速度, 单位deg/s^2 */
31     double   target_TCP_CmpSpeed[2]; /* target_TCP_CmpSpeed[0], TCP合成指令速度(位置), 单位mm/s
32                                target_TCP_CmpSpeed[1], TCP合成指令速度(姿态), 单位deg/s */
33     double   target_TCP_Speed[6]; /* TCP指令速度
34                                target_TCP_Speed[0], 沿x轴速度, 单位mm/s,
35                                target_TCP_Speed[1], 沿y轴速度, 单位mm/s,
36                                target_TCP_Speed[2], 沿z轴速度, 单位mm/s,
37                                target_TCP_Speed[3], 绕固定轴X旋转角速度, 单位deg/s
38                                target_TCP_Speed[4], 绕固定轴y旋转角速度, 单位deg/s
39                                target_TCP_Speed[5], 绕固定轴z旋转角速度, 单位deg/s */
40     double   actual_TCP_CmpSpeed[2]; /* actual_TCP_CmpSpeed[0], TCP合成实际速度(位置), 单位mm/s
41                                actual_TCP_CmpSpeed[1], TCP合成实际速度(姿态), 单位deg/s */
42     double   actual_TCP_Speed[6]; /* TCP实际速度
43                                actual_TCP_Speed[0], 沿x轴速度, 单位mm/s,
44                                actual_TCP_Speed[1], 沿y轴速度, 单位mm/s,
45                                actual_TCP_Speed[2], 沿z轴速度, 单位mm/s,
46                                actual_TCP_Speed[3], 绕固定轴X旋转角速度, 单位deg/s
47                                actual_TCP_Speed[4], 绕固定轴y旋转角速度, 单位deg/s
48                                actual_TCP_Speed[5], 绕固定轴z旋转角速度, 单位deg/s */
49     double   jt_cur_tor[6];   /* 6个轴当前扭矩, 单位N·m */
50     int      tool;           /* 应用的工具坐标系编号 */
51     int      user;           /* 应用的工件坐标系编号 */
52     uint8_t  cl_dgt_output_h; /* 控制箱数字量IO输出15-8 */
53     uint8_t  cl_dgt_output_l; /* 控制箱数字量IO输出7-0 */
54     uint8_t  tl_dgt_output_l; /* 工具数字量IO输出7-0, 仅bit0-bit1有效 */
55     uint8_t  cl_dgt_input_h;  /* 控制箱数字量IO输入15-8 */
56     uint8_t  cl_dgt_input_l;  /* 控制箱数字量IO输入7-0 */
57     uint8_t  tl_dgt_input_l;  /* 工具数字量IO输入7-0, 仅bit0-bit1有效 */
58     uint16_t cl_analog_input[2]; /* cl_analog_input[0], 控制箱模拟量
59                                cl_analog_input[1], 控制箱模拟量
60     uint16_t tl_analog_input; /* 工具模拟量输入 */
61     double   ft_sensor_raw_data[6]; /* 力矩传感器原始数据

```

```

62         ft_sensor_raw_data[0], 沿x轴力, 单位N
63         ft_sensor_raw_data[1], 沿y轴力, 单位N
64         ft_sensor_raw_data[2], 沿z轴力, 单位N
65         ft_sensor_raw_data[3], 沿x轴力矩, 单位Nm
66         ft_sensor_raw_data[4], 沿y轴力矩, 单位Nm
67         ft_sensor_raw_data[5], 沿z轴力矩, 单位Nm */
68     double    ft_sensor_data[6];    /* 力矩传感器数据,
69         ft_sensor_data[0], 沿x轴力, 单位N
70         ft_sensor_data[1], 沿y轴力, 单位N
71         ft_sensor_data[2], 沿z轴力, 单位N
72         ft_sensor_data[3], 沿x轴力矩, 单位Nm
73         ft_sensor_data[4], 沿y轴力矩, 单位Nm
74         ft_sensor_data[5], 沿z轴力矩, 单位Nm */
75     uint8_t    ft_sensor_active;    /* 力矩传感器激活状态, 0-复位, 1-激活 */
76     uint8_t    EmergencyStop;      /* 急停标志, 0-急停未按下, 1-急停按下 */
77     int         motion_done;        /* 运动到位信号, 1-到位, 0-未到位 */
78     uint8_t    gripper_motiondone; /* 夹爪运动完成信号, 1-完成, 0-未完成 */
79     int         mc_queue_len;       /* 运动指令队列长度 */
80     uint8_t    collisionState;      /* 碰撞检测, 1-碰撞, 0-无碰撞 */
81     int         trajectory_pnum;    /* 轨迹点编号 */
82     uint8_t    safety_stop0_state; /* 安全停止信号SI0 */
83     uint8_t    safety_stop1_state; /* 安全停止信号SI1 */
84     uint8_t    gripper_fault_id;    /* 错误夹爪号 */
85     uint16_t   gripper_fault;       /* 夹爪故障 */
86     uint16_t   gripper_active;      /* 夹爪激活状态 */
87     uint8_t    gripper_position;    /* 夹爪位置 */
88     int8_t     gripper_speed;       /* 夹爪速度 */
89     int8_t     gripper_current;     /* 夹爪电流 */
90     int         gripper_temp;       /* 夹爪温度 */
91     int         gripper_voltage;    /* 夹爪电压 */
92     robot_aux_state aux_state;
93     EXT_AXIS_STATUS extAxisStatus[4]; /* UDP扩展轴状态 */
94     uint16_t    extDIState[8];      /* 扩展DI输入
95     uint16_t    extD0State[8];      /* 扩展D0输出
96     uint16_t    extAISState[4];     /* 扩展AI输入
97     uint16_t    extA0State[4];     /* 扩展A0输出
98     int         rbtEnableState;     /* 机器人使能状态
99     double      jointDriverTorque[6]; /* 机器人关节驱动器扭矩
100    double      jointDriverTemperature[6]; /* 机器人关节驱动器温度
101    RobotTime    robotTime;         /* 机器人系统时间
102    int          softwareUpgradeState; /* 机器人软件升级状态
103    uint16_t     endLuaErrCode;      /* 末端LUA运行状态
104    uint16_t     cl_analog_output[2]; /* 控制箱模拟量输出
105    uint16_t     tl_analog_output;   /* 工具模拟量输出
106    float        gripperRotNum;      /* 旋转夹爪当前旋转圈数
107    uint8_t      gripperRotSpeed;    /* 旋转夹爪当前旋转速度百分比
108    uint8_t      gripperRotTorque;   /* 旋转夹爪当前旋转力矩百分比
109    WELDING_BREAKOFF_STATE weldingBreakOffState; /* 焊接中断状态
110    double       jt_tgt_tor[6];      /* 关节指令力矩
111    int          smartToolState;     /* SmartTool手柄按钮状态
112    float        wideVoltageCtrlBoxTemp; /* 宽电压控制箱温度
113    uint16_t     wideVoltageCtrlBoxFanCurrent; /* 宽电压控制箱风扇电流(mA)
114    uint16_t     check_sum;          /* 和校验 */
115 }ROBOT_STATE_PKG;

```