RyCAN伺服协议库入门指南

修改记录：

版本	修改日期	修改	修改人	备注
V1.00	20241017	初始版本	张礼富	
V1.01	20250403	增加波特率设置、主动上报、静默运运指令，软件重启等接口支持	张礼富	

目录

基本数据格式与数据类型

基础数据类型的定义

为了实现在不同的硬件平台上更好地兼容性，本协议库重定义了基础数据类型，库中api输入输出参数均使用重定义后的数据类型。

```
typedef char          s8_t;
typedef unsigned char   u8_t;
typedef unsigned short  u16_t;
typedef short         s16_t;
typedef unsigned long   u32_t;
typedef long          s32_t;
typedef unsigned long long  u64_t;
typedef long  long        s64_t;
```

API接口函数返回值及对应说明

针对API返回值，定义了枚举数据类型 enret_t，库中API接口函数的返回值，有指明参考枚举类型 enret_t 及对应解释的都可参看如下定义及解释。

```
typedef enum
{
    enServo_OK = 0,              // 0 电机响应函数操作成功 1
    enServo_TempratureHighW,     // 1 电机过温告警
    enServo_TempratureHighE,     // 2 电机过温保护
    enServo_VoltageLowE,         // 3 电机低压保护
    enServo_VoltageHighE,        // 4 电机过压保护
    enServo_CurrentOverE,        // 5 电机过流保护
    enServo_TorqueOverE,         // 6 电机力矩保护
    enServo_FuseE,               // 7 电机熔丝位错保护
    enServo_PwmE,                // 8 电机堵转保护
    enServo_DriveE,              // 9 驱动器异常保护
    enServo_HallE,               // 10 电机hall 错保护


    enServoStatus_Bottom,


    enServo_Fail = 250,          //  伺服未响应函数操作或响应超时
    enServo_ParamErr = 251,      //  伺服响应数据出错
    enServo_LibInitErr = 252,    //  伺服组群结构体读写函数未初始化或失败

    enCanFormatError = 253,      //  异步解析时can数据格式不符
    enCanMsgSentFail = 254,      //  can消息发送失败
    enLibHookApplyFailed = 255,  //  hook申请失败  255
} enret_t;
```

CAN数据帧定义

库重新定义了CAN数据帧格式，用户在使用协议库时需先将CAN消息封装成符合下述定义数据帧格式的数据再给协议库使用。

```
typedef struct
{
    u32_t ulId;     // id
    u8_t ucLen;     // 数据长度，取值0 - 8
    u8_t pucDat[64]; // 数据内容
} CanMsg_t;
```

CAN总线消息发送及回调接口类型定义

```
typedef s8_t(*BusWrite_t)(CanMsg_t stuMsg);
typedef void(*Callback_t)(CanMsg_t stuMsg, void * para);
```

伺服主动上报数据结构体定义与说明

```
#pragma pack(1)
```

```
typedef struct
{
    u64_t :8;             // 前面的CMD
    u64_t ucStatus:8;     // 故障状态，0 表示无故障，异常详情见 enret_t
    u64_t ub_P:12;        // 当前位置，0-4095 对应 0到满行程
    u64_t ub_V:12;        // 当前速度，-2048~2047 单位 0.001行程/s
    u64_t ub_I:12;        // 当前电流，-2048~2047 单位 0.01A
    u64_t ub_F:12;        // 当前位置，0-4095 对应手指压力传感器Adc原始值
}FingerServoInfo_t;



typedef struct
{
    u64_t :8;             // 前面的CMD
    u64_t usTp:16;        // 目标位置
    u64_t usTv:16;        // 目标速度
        u64_t usTc:16;        // 目标电流
    u64_t :  8;           // 保留
}FingerServoCmd_t;



typedef union
{
    FingerServoCmd_t  stuCmd;
    FingerServoInfo_t stuInfo;
    u8_t              pucDat[64];
}ServoData_t;
```

钩子数据类型定义与说明

```
typedef struct
{
    volatile u8_t ucEn;       // 使能开关
    volatile u8_t ucAlive;    // hook生命值，新添加Hook后，该值为255，
                              // 每收到一帧CAN，进行一次hook匹配，该值减1，减到0时hook自动由 使能变成失能状态
    CanMsg_t *pstuMsg;
    Callback_t funCbk;
} MsgHook_t;
```

监听数据类型定义与说明

```
typedef struct
{
    MsgHook_t stuListen;
    ServoData_t stuRet;
        u8_t  ucConfidence;   // 数据可信度，越大表示数据越可信，最大255，当为0的，表示数据不可信即收到好多can消息，但并未成功更新该数据
}MsgListen_t;
```

CAN总线数据类型定义与说明

```
typedef struct
```

```
{
    volatile u16_t *pusTicksMs; // 用户程序实现的ms计数器地址，周期是1000即 0~999，
    u16_t usTicksPeriod;        // ms记数器的周期
    u16_t usHookNum;
    u16_t usListenNum;
    MsgHook_t * pstuHook;       // hook 列表首地址
    MsgListen_t * pstuListen;   // listen 列表首地址
    BusWrite_t pfunWrite;       // CAN设备写接口地址
} RyCanServoBus_t;
```

快速使用说明

在使用本协议库之前，需在协议库头文件中根据当前所使用的实际平台设置Target_Platform和ExportType 两个宏的值，宏的设定标准可参考对应的注释说明。

```
// 0 - 目标平台是C语言嵌入式/linux平台（ .lib .o .a .so ... ）
// 1 - 目标平台是Window C++ DLL 平台 （ .dll ）
#ifndef Target_Platform
#define Target_Platform 0
#endif

// 0 - 编译库
// 1 - 引用库
#ifndef ExportType
#define ExportType 1
#endif
```

1、定义全局成员

```
// 定义1ms计数器周期
#define TICKS_PERIOD 1000
// 定义CAN总线成员,一般地独立的CAN总线需要一个RyCanServoBus_t成员与之对应，同时需要实现相应的write函数
RyCanServoBus_t stuServoCan;
// 定义1ms计数器，不同的RyCanServoBus_t成员可以共享用一个 usTicksMs 1ms计数器
volatile u16_t usTicksMs = 0;
```

2、实现硬件平台相关的CAN消息发送接口

```
// 实现CAN消息写接口函数
s8_t write(CanMsg_t stuMsg)
{
    // 硬件相关代码，由用户不同的硬件决定个体如何发送数据
    return sendToCan ( stuMsg.ulId,CAN_CONTROL_MODE,0xaa,1,stuMsg.pucDat,stuMsg.ucLen );
}
```

3、实现Hook或Listen回调函数（非必需）

```
// 实现Listen回调函数，非必需，如果不用，在
void CallBck(CanMsg_t stuMsg, void * para)
{
    u8_t id;
    u8_t buff[8] = {0x55,0x55,0x55,0x55,0x55,0x55,0x55,0x55};
    MsgListen_t *pstuListen = (MsgListen_t *)para;
    id = stuMsg.ulId;
```

```
    // 测试代码，打印一帧特殊数据
    sendToCan ( id,CAN_CONTROL_MODE,0xaa,1,buff,8 );

}
```

4、实现1ms计数器

```
// 用户需实现1ms计数器，如开启定时器，并在定时器（1ms）中断中实现1ms计数器
void SysTick_Handler(void)
{
    …

        // 在这时实现协议库的1ms定时器
        usTicksMs++;
        usTicksMs %= TICKS_PERIOD;

}
```

5、实现硬件平台相关的CAN数据接收与RyCanServoLibRcvMsg接口调用

```
// 在CAN消息接收中断中调用库提供的消息接收函数RyCanServoLibRcvMsg
void CAN1_RX1_IRQHandler(void)
{
        CanMsg_t stuMsg;
    CanRxMsg RxMessage;

    CAN_Receive(CAN1, CAN_FIFO1, &RxMessage);

    // 数据格式转换
    stuMsg.ulId = RxMessage.IDE ? RxMessage.ExtId : RxMessage.StdId;
    stuMsg.ucLen = RxMessage.DLC;
    memcpy(stuMsg.pucDat,RxMessage.Data,RxMessage.DLC);

    // 将接收到的CAN数据传递给 协议库
    RyCanServoLibRcvMsg(&stuServoCan, stuMsg);

}
```

6、主任务中库的使用

```
// 在主函数中实现用户功能代码
int main(void)
{
    …
    u8_t cfg = 0xff;

    // 复位stuServoCan内容
    memset( &stuServoCan ,0,sizeof(RyCanServoBus_t));

    // 指定最大支持的Hook数，一个总线至少需一个Hook，用户可以不指定，库内部最少会申请一个Hook
    stuServoCan.usHookNum = 16;

    // 申请并指定所需的Hook数据空间并初始化，可以直接用calloc，如果已指定stuServoCan.usHookNum，该步可由协议库自动完成
    stuServoCan.pstuHook = malloc(stuServoCan.usHookNum * sizeof(MsgHook_t));
    memset(stuServoCan.pstuHook, 0, stuServoCan.usHookNum * sizeof(MsgHook_t));

    // 指定最大支持的listen数,如果需要实现伺服主动上报功能，则需给定足够的listen，一个伺服需要一个listen
    stuServoCan.usListenNum = 16;

    // 申请并指定所需的listen数据空间，并初始化,可以直接用calloc
    stuServoCan.pstuListen = malloc(stuServoCan.usListenNum * sizeof(MsgListen_t));
    memset(stuServoCan.pstuListen, 0, stuServoCan.usListenNum * sizeof(MsgListen_t));
```

```
    // 初始化库，标红参数在上面已有声明
    RyCanServoBusInit(&stuServoCan, write, &usTicksMs, TICKS_PERIOD );



    // 针对伺服的一些配置，直接用ID实现广播配置，超时为0，表示不关心返回内容
    RyParam_SetProtectionCfg(&stuServoCan,0,&cfg,0);


    // Listen使用示例，为多个ID分别加Listen
       for( i=0;i<15;i++ )
       {
        #if 0

    // 这里要注意，每个Listen或Hook都要为分配一个CanMsg_t对象，如果对个Listen或Hook 用同一个CanMsg_t对象，
    // 效果等同于只有一个 Listen或Hook
        stuListenMsg[i].ulId = SERVO_BACK_ID(i+1);
        stuListenMsg[i].pucDat[0] = 0xaa;   // 直接监听运动指令应答
                #define CMD_GET_MOTOR_INFO            0xA0  // 读取电机信息
                #define CMD_SET_TARGET_POS            0xA1  // 单电机运动指令
                #define CMD_SET_MOTOR_PWM             0xA6  // 电机PWM运动
                #define CMD_SET_TARGET_CURRENT        0xA9  // 设置目标电流
                #define CMD_SET_TARGET_POS2           0xAA  // 设置目位置及过程最大电流 */

        ret = AddListen( &stuServoCan,stuListenMsg + i, CallBck0 );

        #else

    // 这里要注意，每个Listen或Hook都要为分配一个CanMsg_t对象，如果对个Listen或Hook 用同一个CanMsg_t对象，
    // 效果等同于只有一个 Listen或Hook
        stuListenMsg[i].ulId = SERVO_BACK_ID(i+1);
        stuListenMsg[i].pucDat[0] = 0xee;   // 监听主动上报数据
        ret = AddListen( &stuServoCan,stuListenMsg + i, CallBck0 );

        RyParam_SetUpateRate(&stuServoCan, i+1, 20, &sutServoData, 10);  // 开启20ms 主动上报
        RyParam_SetMotionMute(&stuServoCan, i+1, 1, 10);                 // 关闭运动指令主动应答

        #endif
       }

    // 伺服运行 力位混合控制，指定伺服ID 1，目标角度2048，目标速度1s满行程,运行中最大允许电流 70mA，回复指令接收最大等待时间10ms
    ServoData_t fdb;
    RyMotion_ServoMove_Mix( &stuServoCan, 1, 2048, 1000, 70, &fdb, 10 );


    While(1)
    {
        …
    }

}
```

7、常见问题与解答

1) 如何读取伺服的当前角度，速度，和电流以及工作状态信息?

a) 如果只是单次读取，可以使用 RyFunc_GetServoInfo(RyCanServoBus_t *pstuCan, u8_t ucId, ServoData_t *psutServoData, u16_t usTimeout); 接口，函数成功返回后将会在*psutServoData 中看到伺服返回的角度，速度以及电流，力等信息。

b) 如果有周期性调用控制接口（RyMotion_ServoMove_Speed、 RyMotion_ServoMove_Pwm 、RyMotion_CurrentMode、 RyMotion_ServoMove_Mix ） ，此时可以直接能通函数返回值的内容中查看到伺服返回信息， 也可以通过为伺服添加监听（通过AddListen接口），在对应的监听中的成员 stuRet 的值即是伺服的位

置、速度、电流、力及工作状态结果。

2) 添加hook 和 listen 需注意什么?

a) Ry can 伺服通信协议规定，伺服返回数据帧的id 等于伺服id 加 256，所以添加hook 和 listen 时，需设置ulid等于伺服Id加256（如 stuListenMsg[i].ulId = 50 + 256;）。

b) 一般地不建议用户自行加hook(需要很了解通信协议)

加listen是来监听伺服主动上报的数据时，除了ulid需要设置正确外，还需设置正确的监听命令为下面几个值:

```
#define CMD_GET_MOTOR_INFO          0xA0  // 读取电机信息
#define CMD_SET_TARGET_POS          0xA1  // 单电机运动指令
#define CMD_SET_MOTOR_PWM           0xA6  // 电机PWM运动
#define CMD_SET_TARGET_CURRENT      0xA9  // 设置目标电流
#define CMD_SET_TARGET_POS2         0xAA  // 设置目位置及过程最大电流
```

如（ stuListenMsg[i].pucDat[0] = 0xaa; ）。

详细接口说明

同步接口

```
//*****************************************************************************************************************************************
//*函数名称 - AddHook
//*函数作用 - 添加消息钩子
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*   *pstuMsg  - [i] 添加消息钩子的消息特征，ID和cmd，注，若用户使用异步操作方式，自行操作Hook, 给定ID = 实际伺服ID + 256
//*    funCallback    - [i] 添加消息钩子的回调函数
//* 返回值 - 成功 对应消息钩子的索引， 失败 -1
//*****************************************************************************************************************************************
Exportmode s16_t AddHook(RyCanServoBus_t* pstuCan, CanMsg_t* pstuMsg, Callback_t funCallback);


//*****************************************************************************************************************************************
//*函数名称 - DeleteHook
//*函数作用 - 删除消息钩子
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*   *pstuMsg  - [i] 删除消息钩子的消息特征，ID和cmd，注，若用户使用异步操作方式，自行操作Hook, 给定ID = 实际伺服ID + 256
//* 返回值 - 成功 0， 失败 -1
//*****************************************************************************************************************************************
Exportmode s16_t DeleteHook(RyCanServoBus_t* pstuCan, CanMsg_t* pstuMsg);

//*****************************************************************************************************************************************
//*函数名称 - AddListen
//*函数作用 - 添加消息监听
//*  参数：
```

```
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*   *pstuMsg  - [i] 添加消息监听的消息特征，ID和cmd，注，用户操作Listen,给定ID = 实际伺服ID + 256
//*    funCallback    - [i] 添加消息监听的回调函数
//* 返回值 - 成功 对应消息监听的索引， 失败 -1
//*****************************************************************************************************************************
Exportmode s16_t AddListen(RyCanServoBus_t* pstuCan, CanMsg_t* pstuMsg, Callback_t funCallback);


//*****************************************************************************************************************************
//*函数名称 - DeleteListen
//*函数作用 - 删除消息监听
//*  参数 :
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*   *pstuMsg  - [i]  删除消息监听的消息特征，ID和cmd，注，用户操作Listen,给定ID = 实际伺服ID + 256
//* 返回值 - 成功 0， 失败 -1
//*****************************************************************************************************************************
Exportmode s16_t DeleteListen(RyCanServoBus_t* pstuCan, CanMsg_t* pstuMsg);


//*****************************************************************************************************************************
//*函数名称 - GetServoUpdateInfo
//*函数作用 - 从监听的消息中得到伺服上报信息
//*  参数 :
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId      - [i] 指定要读取的信息对应的伺服id
//*    *pstuData - [o] 所读取伺服上报信息（同监听信息）地址
//* 返回值 - 成功 0 ， 失败 1 ， pstuCan参数为空 -1
//*****************************************************************************************************************************
Exportmode s8_t GetServoUpdateInfo(RyCanServoBus_t* pstuCan, u8_t ucId, MsgListen_t* pstuData);


//*****************************************************************************************************************************
//*函数名称 - RyCanServoLibRcvMsg
//*函数作用 - 协议库消息接收处理函数
//*  参数 :
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    stuMsg    - [i] 要处理的can消息
//* 返回值 - 成功 0 ， pstuCan参数为空 -1
//*****************************************************************************************************************************
Exportmode s8_t RyCanServoLibRcvMsg(RyCanServoBus_t* pstuCan, CanMsg_t stuMsg);


//*****************************************************************************************************************************
//*函数名称 - GetRyCanServoLibVersion
//*函数作用 - 获取协议库版本号
//*  参数 :
//*   pucVer[30] -  存储协议库版本号(ascii)字符串数组
//* 返回值 -  无
```

```
//*****************************************************************************************************************************************
Exportmode void GetRyCanServoLibVersion(u8_t pucVer[30]);

//*****************************************************************************************************************************************
//*函数名称 - RyCanServoBusInit
//*函数作用 - CAN总线对像初始化
//* 参数：
//*  *pstuCan  - [i] 伺服CAN总线对象地址
//*   funWrite  - [i] CAN总线消息发送函数地址
//*   pusTime   - [i] ms计数器地址
//*   pusPeriod - [i] ms计数器周期
//* 返回值 - 成功 0， 失败 -1
//*****************************************************************************************************************************************
Exportmode u8_t RyCanServoBusInit(RyCanServoBus_t* pstuCan, BusWrite_t funWrite, volatile u16_t* pusTime, u16_t usPeriod);



//*****************************************************************************************************************************************
//*函数名称 - RyCanServoBusDeInit
//*函数作用 - CAN总线对像销毁，释放占用内存，清除成员内容
//* 参数：
//*  *pstuCan  - [i] 伺服CAN总线对象地址
//* 返回值 - 无
//*****************************************************************************************************************************************
Exportmode void RyCanServoBusDeInit(RyCanServoBus_t* pstuCan);




//*****************************************************************************************************************************************
//*函数名称 - Transmit
//*函数作用 - 数据发送
//* 参数：
//*  *pstuCan  - [i] 伺服CAN总线对象地址
//*   pstuTxmsg - [i] 发送消息对象地址
//*   pstuRxmsg - [io] 接收消息对象地址
//*   *psHook   - [o] 消息接收钩子索引地址
//*   *pusT     - [o] 开始发送时间变量地址
//*   ucCmd     - [i] 发送数据指令码
//*   usTimeout - [i] 接收超时时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
u8_t Transmit(RyCanServoBus_t* pstuCan, CanMsg_t* pstuTxmsg, CanMsg_t* pstuRxmsg, s16_t* psHook, u16_t* pusT, u8_t ucCmd, u16_t  usTimeout);



//*****************************************************************************************************************************************
//*函数名称 - TransmitNone
```

```
//*函数作用 - 虚假数据发送数据，只是为了从总线接收相要数据
//* 参数：
//*  *pstuCan  - [i] 伺服CAN总线对象地址
//*   pstuTxmsg - [i] 发送消息对象地址
//*   pstuRxmsg - [io] 接收消息对象地址
//*   *psHook   - [o] 消息接收钩子索引地址
//*   *pusT     - [o] 开始发送时间变量地址
//*   ucCmd     - [i] 发送数据指令码
//*   usTimeout - [i] 接收超时时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
u8_t TransmitNone(RyCanServoBus_t* pstuCan, CanMsg_t* pstuTxmsg, CanMsg_t* pstuRxmsg, s16_t* psHook, u16_t* pusT, u8_t ucCmd, u16_t  usTimeout);



//*****************************************************************************************************************************
//*函数名称 - Receive
//*函数作用 - 接收CAN总线数据
//* 参数：
//*  *pstuCan  - [i] 伺服CAN总线对象地址
//*   pstuRxmsg - [i] 接收消息对象地址
//*   sHook     - [i] 消息接收钩子索引
//*   usT       - [i] 开始发送时间变量
//*   usTimeout - [i] 接收超时时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
u8_t Receive(RyCanServoBus_t* pstuCan, CanMsg_t* pstuRxmsg, s16_t sHook, u16_t usT, u16_t  usTimeout);



//*****************************************************************************************************************************
//*函数名称 - RyFunc_GetServoInfo
//*函数作用 - 得到伺服信息
//* 参数：
//*  *pstuCan     - [i] 伺服CAN总线对象地址
//*   *psutServoData - [o] 用于接收伺服信息的首地址 ， 参考 ServoData_t 数据描述
//*   usTimeout    - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t RyFunc_GetServoInfo(RyCanServoBus_t* pstuCan, u8_t ucId, ServoData_t* psutServoData, u16_t  usTimeout);



//*****************************************************************************************************************************
//*函数名称 - RyFunc_GetVersion
```

```
//*函数作用 - 读取版本信息
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId          - [i] 指定伺服ID，0-254 ，0表示广播，255保留不用
//*    *pucVer       - [o] 用于存储版本信息地址，
//*    usTimeout     - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t RyFunc_GetVersion(RyCanServoBus_t* pstuCan, u8_t ucId, u8_t pucVer[64], u16_t  usTimeout);



//*****************************************************************************************************************************
//*函数名称 - RyFunc_StartUpgrade
//*函数作用 - 开始升级请求
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId          - [i] 指定伺服ID，0-254 ，0表示广播，255保留不用
//*    ulDataLength   - [i] 升级数据长度单位 byte
//*    *usPsw        - [io] 升级密码，为防止误进入升级模式，升时需要进行密码验证！
//*                    主机首次使用该指令该值随便给，设备回复数据帧中
//*                    会主给出对应的正确的升级密码，主机要得到正确的升级密码后，
//*                    将正确确的升级密码通过该指令下发，设备便可进入升级模式。
//*   *pucFrameLen     - [io] 主机支持的单帧最大长度，返回主机及从机支持最大帧长度的最小值，后续升级操作需按返回值的帧长度进行操作
//*    *pucStatus     - [o] 设备执行指令状态，其内容含义：
//*                  0：进入密码不通过，些时在 *usPsw 中将返回正确密码
//*                  1：从app进入升级模式
//*                  2：boot进入升级模式
//*                  3：可供写入的内存不够，升级模式进入失败
//*                  4：数据擦除开始
//*                  5：数据擦除OK，可以进行升级数据传输
//*    usTimeout      - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t RyFunc_StartUpgrade(RyCanServoBus_t* pstuCan, u8_t ucId,  u32_t ulDataLength, u16_t* usPsw, u8_t* pucFrameLen, u8_t* pucStatus, u16_t
 usTimeout);



//*****************************************************************************************************************************
//*函数名称 - RyFunc_WriteUpgradeData
//*函数作用 - 写升级数据
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId          - [i] 指定伺服ID，0-254 ，0表示广播，255保留不用
//*    ulDataAddr     - [i] 升级数据地址，相对地址 0~0xFFFFFF，
//*    *pucData       - [i] 升级数据首地址
```

```
//*    ucDataLen      - [i] 当前帧数据长度，需小于等于 RyFunc_StartUpgrade 返回的 ( *pucFrameLen - 4 ) 值
//*    *ulNextAddr    - [o] 下一个要数据的数据地址
//*   *ulPreData      - [o] 接收到的最新的升级数据内容
//*    usTimeout      - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t RyFunc_WriteUpgradeData(RyCanServoBus_t* pstuCan, u8_t ucId, u32_t ulDataAddr, u8_t* pucData, u8_t ucDataLen, u32_t* ulNextAddr, u16_t
 usTimeout);


//*****************************************************************************************************************************
//*函数名称 - RyFunc_FinishUpgrade
//*函数作用 - 升级结束
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId          - [i] 指定伺服ID，0-254 ，0表示广播，255保留不用
//*    *pulCRCValue, - [io] 上升数据的CRC32校验内容，CRC32多项式为 0X4C11DB7，如果校验不成立，这里会返回伺服/设备计算的CRC32结果，
//*    *pucStatus,   - [o] 升级状态:
//*                      0: 升级成功，节点重起
//*                      1：CRC32校验失败
//*                      2：新固件硬件版本不匹配
//*    usTimeout      - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t RyFunc_FinishUpgrade(RyCanServoBus_t* pstuCan, u16_t ucId, u32_t* pulCRCValue, u32_t* pucStatus, u16_t  usTimeout);


//*****************************************************************************************************************************
//*函数名称 - RyFunc_SetSNCode
//*函数作用 - 设置伺服SN码
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId        - [i] 指定伺服ID，0-254 ，0表示广播，255保留不用
//*    pucBuff     - [i] 要写入的伺服SN码地址（标准内容是ASCII码,长度小于36字节）
//*    usTimeout      - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t RyFunc_SetSNCode(RyCanServoBus_t* pstuCan, u8_t ucId, u8_t pucBuff[40], u16_t  usTimeout);


//*****************************************************************************************************************************
//*函数名称 - RyFunc_GetSNCode
//*函数作用 - 读伺服SN码
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
```

```
//*    ucId          - [i] 指定伺服ID，0-254 ， 0表示广播，255保留不用
//*    pucBuff       - [o] 读取的伺服SN码首地址（标准内容是ASCII码,长度小于36字节）
//*    usTimeout     - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*************************************************************************************************************************
Exportmode u8_t RyFunc_GetSNCode(RyCanServoBus_t* pstuCan, u8_t ucId, u8_t pucBuff[40], u16_t  usTimeout);



//*************************************************************************************************************************
//*函数名称 - RyParam_SetTime
//*函数作用 - 设置设备时间
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId          - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*    ubTime        - [i] 要设置的时间对象，参看 DevTimeCmd_t 定义
//*    usTimeout     - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*************************************************************************************************************************
Exportmode u8_t RyParam_SetTime(RyCanServoBus_t* pstuCan, u8_t ucId, DevTimeCmd_t ubTime, u16_t  usTimeout);



//*************************************************************************************************************************
//*函数名称 - RyParam_SetID
//*函数作用 - 设置伺服ID
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId          - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*    ucNewId       - [i] 要修改的伺服ID值，1-254
//*    *pulUniqueCode - [io] 标识伺服唯一身份的码变量地址，变量值如果不知可以随便给一个（如给0），伺服在收到指令后的回复中会上报自身唯一码
//*    usTimeout     - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*************************************************************************************************************************
Exportmode u8_t RyParam_SetID(RyCanServoBus_t* pstuCan, u8_t ucId, u8_t ucNewId, u32_t* pulUniqueCode, u16_t  usTimeout);



//*************************************************************************************************************************
//*函数名称 - RyParam_Recover
//*函数作用 - 恢复出厂
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId          - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*    *pulUniqueCode - [io] 标识伺服唯一身份的码变量地址，变量值如果不知可以随便给一个（如给0），伺服在收到指令后的回复中会上报自身唯一码
//*    usTimeout     - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
```

```
//*****************************************************************************************************************************************
Exportmode u8_t RyParam_Recover(RyCanServoBus_t* pstuCan, u8_t ucId, u32_t* pulUniqueCode, u16_t  usTimeout);



//*****************************************************************************************************************************************
//*函数名称 - RyFunc_Reset
//*函数作用 - 设备重启
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId          - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*    usTimeout      - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
Exportmode u8_t RyFunc_Reset(RyCanServoBus_t* pstuCan, u8_t ucId, u16_t  usTimeout);



//*****************************************************************************************************************************************
//*函数名称 - RyParam_SetCanFDBaudRate
//*函数作用 - 设置CAN/CANFD总线波特率
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId          - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*    ulCtrlBuadRate - [i] CANFD 总线控制域波特率，或CAN 总线波特率，支持的波特率（*为默认）：
//*                      5000,
//*                      10000,
//*                      20000,
//*                      25000,
//*                      40000,
//*                      50000,
//*                      62500,
//*                      80000,
//*                      100000,
//*                      125000,
//*                      200000,
//*                      250000,
//*                      400000,
//*                      500000,
//*                      800000,
//*                      1000000, *
//*    ulDataBaudRate - [i] CANFD 总线数据域波特率, 如果是CAN 总线保持默认或与控制一致即可，支持波特率（*为默认）：
//*                       5000,
//*                       10000,
//*                       20000,
//*                       25000,
//*                       40000,
```

```
//*                     50000,
//*                     62500,
//*                     80000,
//*                     100000,
//*                     125000,
//*                     200000,
//*                     250000,
//*                     400000,
//*                     500000,
//*                     800000,
//*                     1000000,
//*                     2000000,
//*                     4000000,
//*                     5000000, *
//*    usTimeout      - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
Exportmode u8_t RyParam_SetCanFDBaudRate(RyCanServoBus_t* pstuCan, u8_t ucId, u32_t ulCtrlBuadRate, u32_t ulDataBaudRate, u16_t  usTimeout);



//*****************************************************************************************************************************************
//*函数名称 - RyParam_SetRS485BaudRate
//*函数作用 - 设置RS485总线波特率
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId          - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*    ulBuadRate     - [i] Rs485 总线波特率，常见的波特率（*为默认）：
//*                     1200
//*                     2400
//*                     4800
//*                     9600
//*                     19200
//*                     38400
//*                     57600
//*                     115200
//*                     230400
//*                     460800
//*                     921600
//*                     1000000
//*                     2000000
//*                     2500000
//*                     3000000
//*                     4000000
//*                     5000000 *
//*                     6000000
//*                     8000000
//*                     10000000
```

```
//*    usTimeout      - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t RyParam_SetRS485BaudRate(RyCanServoBus_t* pstuCan, u8_t ucId, u32_t ulBuadRate, u16_t  usTimeout);



//*****************************************************************************************************************************
//*函数名称 - RyParam_SetUpateRate
//*函数作用 - 设置核心数据主动上报频率，掉电不保存
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId          - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*    usTspan       - [i] 主动上报时间间隔，单位ms  0~1000, 0 表示不上报， 超过1000按1000上报
//*   *psutServoData  - [o] 用于接收伺服信息的首地址
//*    usTimeout      - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t RyParam_SetUpateRate(RyCanServoBus_t* pstuCan, u8_t ucId, u16_t usTspan, ServoData_t* psutServoData, u16_t  usTimeout);



//*****************************************************************************************************************************
//*函数名称 - RyParam_SetMotionMute
//*函数作用 - 设置运动指令静默，即开启或关闭运动指令应答，用于节省总线带宽，掉电不保存
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId          - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*    ucMute        - [i] 运动指令静默使用， 0 - 运动指令正常应签， 1 - 运动指令不应答
//*    usTimeout      - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t RyParam_SetMotionMute(RyCanServoBus_t* pstuCan, u8_t ucId, u8_t ucMute, u16_t  usTimeout);



//*****************************************************************************************************************************
//*函数名称 - RyParam_GetRigidity
//*函数作用 - 读取电机控制刚度值
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId          - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*    *pucRigidity    - [i] 用于接收伺服位置控制刚度信息
//*    usTimeout      - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t RyParam_GetRigidity(RyCanServoBus_t* pstuCan, u8_t ucId, u8_t* pucRigidity, u16_t  usTimeout);
```

```
//*****************************************************************************************************************************************
//*函数名称 - RyParam_SetRigidity
//*函数作用 - 设置电机控制刚度
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId          - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*    ucRigidity     - [i] 伺服位置控制刚度值
//*    usTimeout      - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
Exportmode u8_t RyParam_SetRigidity(RyCanServoBus_t* pstuCan, u8_t ucId, u8_t ucRigidity, u16_t  usTimeout);



//*****************************************************************************************************************************************
//*函数名称 - RyParam_SetPosition
//*函数作用 - 设置伺服位置
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId          - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*    usPosition     - [i] 目标位置值，0到4095对应 0~满行程
//*    *pusPosoff     - [o] 接收返回的内部位置补偿值，0到4095对应 0~满行程
//*    usTimeout      - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
Exportmode u8_t RyParam_SetPosition(RyCanServoBus_t* pstuCan, u8_t ucId, u16_t usPosition, u16_t* pusPosoff, u16_t  usTimeout);



//*****************************************************************************************************************************************
//*函数名称 - RyParam_ClearFault
//*函数作用 - 清除伺服故障
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId          - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*    usTimeout      - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
Exportmode u8_t RyParam_ClearFault(RyCanServoBus_t* pstuCan, u8_t ucId, u16_t  usTimeout);



//*****************************************************************************************************************************************
//*函数名称 - RyParam_GetProtectionCfg
```

```
//*函数作用 - 读取保护配置
//*  参数 :
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId          - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//* *pusProtectionCfg  - [o] 接收保护配置首地址，每1位对应一种保护，每一位代表一种保护
//*                        使能状态：0 - 不使能，1 - 使能
//*                        bit 0  -  电机温度保护
//*                        bit 1  -  电机电压保护
//*                        bit 2  -  电机过流保护
//*                        bit 3  -  电机力矩保护
//*                        bit 4  -  电机熔丝位错保护
//*                        bit 5  -  电机堵转保护
//*    usTimeout      - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t RyParam_GetProtectionCfg(RyCanServoBus_t* pstuCan, u8_t ucId, u16_t* pusProtectionCfg, u16_t  usTimeout);




//*****************************************************************************************************************************
//*函数名称 - RyParam_SetProtectionCfg
//*函数作用 - 设置保护配置
//*  参数 :
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId          - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*   *pusProtectionCfg  - [io] 保护配置值首地址，每1位对应一种保护，每一位代表一种保护
//*                        使能状态：0 - 不使能，1 - 使能
//*                        bit 0  -  电机温度保护
//*                        bit 1  -  电机电压保护
//*                        bit 2  -  电机过流保护
//*                        bit 3  -  电机力矩保护
//*                        bit 4  -  电机熔丝位错保护
//*                        bit 5  -  电机堵转保护
//*    usTimeout      - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t RyParam_SetProtectionCfg(RyCanServoBus_t* pstuCan, u8_t ucId, u16_t* pusProtectionCfg, u16_t  usTimeout);




//*****************************************************************************************************************************
//*函数名称 - RyParam_GetStroke
//*函数作用 - 读取伺服行程
//*  参数 :
//*   *pstuCan     - [i] 伺服CAN总线对象地址
```

```
//*    ucId      - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*   *pulStroke   - [o] 返回行程信息首地址，量纲为 电机原始位置传感器/编码器行程值
//* usTimeout     - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t RyParam_GetStroke(RyCanServoBus_t* pstuCan, u8_t ucId, u32_t* pulStroke,  u16_t  usTimeout);



//*****************************************************************************************************************************
//*函数名称 - RyParam_GetStroke
//*函数作用 - 设置伺服行程
//*  参数：
//*   *pstuCan     - [i] 伺服CAN总线对象地址
//*    ucId      - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*   ulStroke     - [i] 行程值，量纲为 电机原始位置传感器/编码器行程值
//* usTimeout     - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t RyParam_SetStroke(RyCanServoBus_t* pstuCan, u8_t ucId, u32_t ulStroke,  u16_t  usTimeout);



//*****************************************************************************************************************************
//*函数名称 - RyParam_GetStroke_H
//*函数作用 - 读取伺服行程上限
//*  参数：
//*   *pstuCan     - [i] 伺服CAN总线对象地址
//*    ucId      - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*   *pulStrokeH  - [o] 返回行程上限值首地址，量纲为 电机原始位置传感器/编码器行程值
//* usTimeout     - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t RyParam_GetStroke_H(RyCanServoBus_t* pstuCan, u8_t ucId, u32_t* pulStrokeH,  u16_t  usTimeout);



//*****************************************************************************************************************************
//*函数名称 - RyParam_SetStroke_H
//*函数作用 - 设置伺服行程上限
//*  参数：
//*   *pstuCan     - [i] 伺服CAN总线对象地址
//*    ucId      - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*   ulStrokeH    - [i] 行程值上限值，量纲为 电机原始位置传感器/编码器行程值
//* usTimeout     - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
```

```
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t RyParam_SetStroke_H(RyCanServoBus_t* pstuCan, u8_t ucId, u32_t ulStrokeH, u16_t usTimeout);



//*****************************************************************************************************************************
//*函数名称 - RyParam_GetStroke_L
//*函数作用 - 读取伺服行程下限值
//* 参数：
//*  *pstuCan    - [i] 伺服CAN总线对象地址
//*   ucId      - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*  *pulStrokeL - [o] 返回行程信下限值首地址，量纲为 电机原始位置传感器/编码器行程值
//* usTimeout    - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t RyParam_GetStroke_L(RyCanServoBus_t* pstuCan, u8_t ucId, u32_t* pulStrokeL, u16_t usTimeout);



//*****************************************************************************************************************************
//*函数名称 - RyParam_SetStroke_L
//*函数作用 - 设置伺服行程下限值
//* 参数：
//*  *pstuCan    - [i] 伺服CAN总线对象地址
//*   ucId      - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*  ulStrokeL   - [i] 行程值下限值，量纲为 电机原始位置传感器/编码器行程值
//* usTimeout    - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t RyParam_SetStroke_L(RyCanServoBus_t* pstuCan, u8_t ucId, u32_t ulStrokeL, u16_t usTimeout);



//*****************************************************************************************************************************
//*函数名称 - RyMotion_ServoMove_Speed
//*函数作用 - 指定速度运动
//* 参数：
//*  *pstuCan    - [i] 伺服CAN总线对象地址
//*   ucId       - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*   sTargetAngle  - [i] 伺服目标角度，0到4095 对应 0~满行程
//*   usRunSpeed    - [i] 伺服运行速度，0~65535 单位 0.001行程/s ， 实际值是 0~5000左右,取决于电大电机速度
//*  *psutServoData - [o] 用于接收伺服信息的首地址
//*   usTimeout    - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t RyMotion_ServoMove_Speed(RyCanServoBus_t* pstuCan, u8_t ucId, s16_t sTargetAngle, u16_t usRunSpeed, ServoData_t* psutServoData, u16_t
```

```
 usTimeout);


//*****************************************************************************************************************************************
//*函数名称 - RyMotion_ServoMove_Pwm
//*函数作用 - 指定PWM运动
//*  参数：
//*   *pstuCan     - [i] 伺服CAN总线对象地址
//*    ucId         - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*    sPwm         - [i] 电机运行PWM，-1000到1000对应正负满PWM
//*   *psutServoData - [o] 用于接收伺服信息的首地址 ,数据内容见 ServoData_t
//*    usTimeout    - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
Exportmode u8_t RyMotion_ServoMove_Pwm(RyCanServoBus_t* pstuCan, u8_t ucId, s16_t sPwm,  ServoData_t* psutServoData, u16_t  usTimeout);


//*****************************************************************************************************************************************
//*函数名称 - RyMotion_CurrentMode
//*函数作用 - 电流模式运动
//*  参数：
//*   *pstuCan  - [i] 伺服CAN总线对象地址
//*    ucId         - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*    sTargetCurrent - [i] 目标电流，-32768到32767 对应-32.768A到 32.767A ，实际可用范围 -3000 到 3000
//*   *psutServoData - [o] 用于接收伺服信息的首地址
//*    usTimeout    - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
Exportmode u8_t RyMotion_CurrentMode(RyCanServoBus_t* pstuCan, u8_t ucId, s16_t sTargetCurrent, ServoData_t* psutServoData, u16_t  usTimeout);


//*****************************************************************************************************************************************
//*函数名称 - RyMotion_ServoMove_Mix
//*函数作用 - 力位混合控制
//*  参数：
//*   *pstuCan     - [i] 伺服CAN总线对象地址
//*    ucId         - [i] 指定伺服ID，0-254，0表示广播，255保留不用
//*    sTargetAngle  - [i] 伺服目标角度，0到4095 对应 0~满行程
//*    usRunSpeed    - [i] 伺服运行速度，0~65535 单位 0.001行程/s ， 实际值是 0~5000左右,取决于电大电机速度
//*    sMaxCurrent   - [i] 运行中最大允许电流，-32768到32767 对应-32.768A到 32.767A ，实际可用范围 -1000 到 1000
//*   *psutServoData - [o] 用于接收伺服信息的首地址
//*    usTimeout    - [i] 接收超时时间,单位ms,0~65535,其中0表示异步解析返回值或不关心返回值，函数不等待，发完即回，其它表示最大等待返回时间
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
Exportmode u8_t RyMotion_ServoMove_Mix(RyCanServoBus_t* pstuCan, u8_t ucId, s16_t sTargetAngle, u16_t usRunSpeed, u16_t sMaxCurrent, ServoData_t*
psutServoData, u16_t  usTimeout);
```

异步解析接口

```
#if USE_ASYN_PRASE



//*****************************************************************************************************************************************
//*函数名称 - AsynParse_RyFunc_GetServoInfo
//*函数作用 - 异步解析 得到伺服信息
//*  参数：
//*     stuRxmsg      - [i] CAN接收消息
//*     *pusId        - [o] 消息对应的节点ID
//*    *psutServoData - [o] 返回的伺服数据
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
Exportmode u8_t AsynParse_RyFunc_GetServoInfo(CanMsg_t stuRxmsg, u16_t* pusId, ServoData_t* psutServoData);



//*****************************************************************************************************************************************
//*函数名称 - AsynParse_RyFunc_GetVersion
//*函数作用 - 异步解析 读取版本信息，异频模式下该接口可能会解析不完整，请使用同步模式
//*  参数：
//*     stuRxmsg      - [i] CAN接收消息
//*     *pusId        - [o] 消息对应的节点ID
//*     *pucVer       - [o] 用于接收版本信息内存地址，
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
Exportmode u8_t AsynParse_RyFunc_GetVersion(CanMsg_t stuRxmsg, u16_t* pusId, u8_t pucVer[64]);



//*****************************************************************************************************************************************
//*函数名称 - AsynParse_RyFunc_StartUpgrade
//*函数作用 - 异步解析 开始升级请求
//*  参数：
//*     stuRxmsg      - [i] CAN接收消息
//*     *pusId        - [o] 消息对应的节点ID
```

```
//*    *usPsw        - [io] 升级密码，为防止误进入升级模式，升时需要进行密码验证！
//*                  主机首次使用该指令该值随便给，设备回复数据帧中
//*                  会主给出对应的正确的升级密码，主机要得到正确的升级密码后，
//*                  将正确确的升级密码通过该指令下发，设备便可进入升级模式。
//*   *pucFrameLen    - [io] 主机支持的单帧最大长度，返回主机及从机支持最大帧长度的最小值，后续升级操作需按返回值的帧长度进行操作
//*    *pucStatus     - [o] 设备执行指令状态，其内容含义：
//*                0：进入密码不通过，些时在 *usPsw 中将返回正确密码
//*                1：从app进入升级模式
//*                2：boot进入升级模式
//*                3：可供写入的内存不够，升级模式进入失败
//*                4：数据擦除开始
//*                5：数据擦除OK，可以进行升级数据传输
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
Exportmode u8_t AsynParse_RyFunc_StartUpgrade(CanMsg_t stuRxmsg, u16_t* pusId, u16_t* usPsw, u8_t* pucFrameLen, u8_t* pucStatus);



//*****************************************************************************************************************************************
//*函数名称 - AsynParse_RyFunc_WriteUpgradeData
//*函数作用 - 异步解析 写升级数据
//*  参数：
//*    stuRxmsg      - [i] CAN接收消息
//*    *pusId        - [o] 消息对应的节点ID
//*    *ulNextAddr    - [o] 下一个要数据的数据地址
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
Exportmode u8_t AsynParse_RyFunc_WriteUpgradeData(CanMsg_t stuRxmsg, u16_t* pusId, u32_t* ulNextAddr);



//*****************************************************************************************************************************************
//*函数名称 - AsynParse_RyFunc_FinishUpgrade
//*函数作用 - 异步解析 升级结束
//*  参数：
//*    stuRxmsg      - [i] CAN接收消息
//*    *pusId        - [o] 消息对应的节点ID
//*    *pulCRCValue, - [io] 上升数据的CRC32校验内容，CRC32多项式为 0X4C11DB7，如果校验不成立，这里会返回伺服/设备计算的CRC32结果，
//*    *pucStatus,   - [o] 升级状态:
//*                  0: 升级成功，节点重起
//*                  1：CRC32校验失败
//*                  2：新固件硬件版本不匹配
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
Exportmode u8_t AsynParse_RyFunc_FinishUpgrade(CanMsg_t stuRxmsg, u16_t* pusId, u32_t* pulCRCValue, u8_t* pucStatus);
```

```
//*****************************************************************************************************************************************
//*函数名称 - AsynParse_RyFunc_SetSNCode
//*函数作用 - 异步解析 设置伺服SN码
//*  参数：
//*    stuRxmsg      - [i] CAN接收消息
//*    *pusId        - [o] 消息对应的节点ID
//* 返回值 - 参考枚举类型 enret_t 及对应解释
Exportmode u8_t AsynParse_RyFunc_SetSNCode(CanMsg_t stuRxmsg, u16_t* pusId);



//*****************************************************************************************************************************************
//*函数名称 - AsynParse_RyFunc_GetSNCode
//*函数作用 - 异步解析 读伺服SN码, 异步模式下该接口需要多次调用
//*  参数：
//*    stuRxmsg      - [i] CAN接收消息
//*    *pusId        - [o] 消息对应的节点ID
//*    *pucSN        - [o] 用于接收SN信息内存地址，
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
Exportmode u8_t AsynParse_RyFunc_GetSNCode(CanMsg_t stuRxmsg, u16_t* pusId, u8_t pucSN[64]);



//*****************************************************************************************************************************************
//*函数名称 - AsynParse_RyParam_SetTime
//*函数作用 - 异步解析 设置伺服SN码
//*  参数：
//*    stuRxmsg      - [i] CAN接收消息
//*    *pusId        - [o] 消息对应的节点ID
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
Exportmode u8_t AsynParse_RyParam_SetTime(CanMsg_t stuRxmsg, u16_t* pusId);



//*****************************************************************************************************************************************
//*函数名称 - AsynParse_RyParam_SetID
//*函数作用 - 异步解析 设置伺服ID
//*  参数：
//*    stuRxmsg      - [i] CAN接收消息
//*    *pusId        - [o] 消息对应的节点ID
//*    *ulPsw        - [o] 用于存储设备返回的密码首地址，
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
Exportmode u8_t AsynParse_RyParam_SetID(CanMsg_t stuRxmsg, u16_t* pusId, u32_t* ulPsw);
```

```
//*****************************************************************************************************************************************
//*函数名称 - AsynParse_RyParam_Recover
//*函数作用 - 异步解析 恢复出厂
//*  参数：
//*    stuRxmsg      - [i] CAN接收消息
//*    *pusId        - [o] 消息对应的节点ID
//*    *ulPsw        - [o] 用于存储设备返回的密码首地址，
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
Exportmode u8_t AsynParse_RyParam_Recover(CanMsg_t stuRxmsg, u16_t* pusId, u32_t* ulPsw);



//*****************************************************************************************************************************************
//*函数名称 - AsynParse_RyFunc_Reset
//*函数作用 - 异步解析  设备重启
//*  参数：
//*    stuRxmsg      - [i] CAN接收消息
//*    *pusId        - [o] 消息对应的节点ID
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
Exportmode u8_t AsynParse_RyFunc_Reset(CanMsg_t stuRxmsg, u16_t* pusId);

//*****************************************************************************************************************************************
//*函数名称 - AsynParse_RyParam_SetCanFDBaudRate
//*函数作用 - 异步解析  设置CAN/CANFD总线波特率
//*  参数：
//*    stuRxmsg      - [i] CAN接收消息
//*    *pusId        - [o] 消息对应的节点ID
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
Exportmode u8_t AsynParse_RyParam_SetCanFDBaudRate(CanMsg_t stuRxmsg, u16_t* pusId);

//*****************************************************************************************************************************************
//*函数名称 - AsynParse_RyParam_SetRS485BaudRate
//*函数作用 - 异步解析  设置RS485总线波特率
//*  参数：
//*    stuRxmsg      - [i] CAN接收消息
//*    *pusId        - [o] 消息对应的节点ID
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************************
Exportmode u8_t AsynParse_RyParam_SetRS485BaudRate(CanMsg_t stuRxmsg, u16_t* pusId);

//*****************************************************************************************************************************************
//*函数名称 - AsynParse_RyParam_SetUpateRate
```

```
//*函数作用 - 异步解析 设置核心数据主动上报频率，掉电不保存
//* 参数：
//*    stuRxmsg     - [i] CAN接收消息
//*    *pusId       - [o] 消息对应的节点ID
//*   *psutServoData - [o] 返回的伺服数据
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t AsynParse_RyParam_SetUpateRate(CanMsg_t stuRxmsg, u16_t* pusId, ServoData_t* psutServoData);


//*****************************************************************************************************************************
//*函数名称 - AsynParse_RyParam_SetMotionMute
//*函数作用 - 异步解析 设置运动指令静默，即开启或关闭运动指令应答，用于节省总线带宽，掉电不保存
//* 参数：
//*    stuRxmsg     - [i] CAN接收消息
//*    *pusId       - [o] 消息对应的节点ID
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t AsynParse_RyParam_SetMotionMute(CanMsg_t stuRxmsg, u16_t* pusId);


//*****************************************************************************************************************************
//*函数名称 - AsynParse_RyParam_GetRigidity
//*函数作用 - 异步解析 读取电机控制刚度值
//* 参数：
//*    stuRxmsg     - [i] CAN接收消息
//*    *pusId       - [o] 消息对应的节点ID
//*    *ucRigidity   - [o] 用于存储设备伺服控制刚度
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t AsynParse_RyParam_GetRigidity(CanMsg_t stuRxmsg, u16_t* pusId, u8_t* ucRigidity);


//*****************************************************************************************************************************
//*函数名称 - AsynParse_RyParam_SetRigidity
//*函数作用 - 异步解析 设置电机控制刚度
//* 参数：
//*    stuRxmsg     - [i] CAN接收消息
//*    *pusId       - [o] 消息对应的节点ID
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t AsynParse_RyParam_SetRigidity(CanMsg_t stuRxmsg, u16_t* pusId);


//*****************************************************************************************************************************
//*函数名称 - AsynParse_RyParam_SetPosition
```

```
//*函数作用 - 异步解析 设置伺服位置
//* 参数：
//*    stuRxmsg     - [i] CAN接收消息
//*    *pusId       - [o] 消息对应的节点ID
//*    *pusPosoff   - [o] 接收返回的内部位置补偿值，0到4095对应 0~满行程
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t AsynParse_RyParam_SetPosition(CanMsg_t stuRxmsg, u16_t* pusId, u16_t* pusPosoff);



//*****************************************************************************************************************************
//*函数名称 - AsynParse_RyParam_ClearFault
//*函数作用 - 异步解析 清除伺服故障
//* 参数：
//*    stuRxmsg     - [i] CAN接收消息
//*    *pusId       - [o] 消息对应的节点ID
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t AsynParse_RyParam_ClearFault(CanMsg_t stuRxmsg, u16_t* pusId);



//*****************************************************************************************************************************
//*函数名称 - AsynParse_RyParam_GetProtectionCfg
//*函数作用 - 异步解析 读取保护配置
//* 参数：
//*    stuRxmsg     - [i] CAN接收消息
//*    *pusId       - [o] 消息对应的节点ID
//* *pusProtectionCfg  - [o] 接收保护配置首地址，每1位对应一种保护，每一位代表一种保护
//*                      使能状态：0 - 不使能，1 - 使能
//*                      bit 0  -  电机温度保护
//*                      bit 1  -  电机电压保护
//*                      bit 2  -  电机过流保护
//*                      bit 3  -  电机力矩保护
//*                      bit 4  -  电机熔丝位错保护
//*                      bit 5  -  电机堵转保护
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t AsynParse_RyParam_GetProtectionCfg(CanMsg_t stuRxmsg, u16_t* pusId, u16_t* pusProtectionCfg);



//*****************************************************************************************************************************
//*函数名称 - AsynParse_RyParam_SetProtectionCfg
//*函数作用 - 异步解析 设置保护配置
//* 参数：
```

```
//*    stuRxmsg      - [i] CAN接收消息
//*    *pusId        - [o] 消息对应的节点ID
//* *pusProtectionCfg - [o] 接收保护配置首地址，每1位对应一种保护，每一位代表一种保护
//*                       使能状态：0 - 不使能，1 - 使能
//*                       bit 0  -  电机温度保护
//*                       bit 1  -  电机电压保护
//*                       bit 2  -  电机过流保护
//*                       bit 3  -  电机力矩保护
//*                       bit 4  -  电机熔丝位错保护
//*                       bit 5  -  电机堵转保护
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t AsynParse_RyParam_SetProtectionCfg(CanMsg_t stuRxmsg, u16_t* pusId, u16_t* pusProtectionCfg);



//*****************************************************************************************************************************
//*函数名称 - AsynParse_RyParam_GetStroke
//*函数作用 - 异步解析 读取伺服行程
//*  参数：
//*    stuRxmsg      - [i] CAN接收消息
//*    *pusId        - [o] 消息对应的节点ID
//*    *pulStroke    - [o] 返回行程信息首地址，量纲为 电机原始位置传感器/编码器行程值
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t AsynParse_RyParam_GetStroke(CanMsg_t stuRxmsg, u16_t* pusId,  u32_t* pulStroke);



//*****************************************************************************************************************************
//*函数名称 - AsynParse_RyParam_SetStroke
//*函数作用 - 异步解析 设置伺服行程
//*  参数：
//*    stuRxmsg      - [i] CAN接收消息
//*    *pusId        - [o] 消息对应的节点ID
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t AsynParse_RyParam_SetStroke(CanMsg_t stuRxmsg, u16_t* pusId);



//*****************************************************************************************************************************
//*函数名称 - AsynParse_RyParam_GetStroke_H
//*函数作用 - 异步解析 读取伺服行程上限
//*  参数：
//*    stuRxmsg      - [i] CAN接收消息
//*    *pusId        - [o] 消息对应的节点ID
```

```
//*   *pulStrokeH     - [o] 返回行程上限值首地址，量纲为 电机原始位置传感器/编码器行程值
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t AsynParse_RyParam_GetStroke_H(CanMsg_t stuRxmsg, u16_t* pusId,  u32_t* pulStrokeH);


//*****************************************************************************************************************************
//*函数名称 - AsynParse_RyParam_SetStroke_H
//*函数作用 - 异步解析 设置伺服行程上限
//*  参数：
//*     stuRxmsg      - [i] CAN接收消息
//*     *pusId        - [o] 消息对应的节点ID
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t AsynParse_RyParam_SetStroke_H(CanMsg_t stuRxmsg, u16_t* pusId);


//*****************************************************************************************************************************
//*函数名称 - AsynParse_RyParam_GetStroke_L
//*函数作用 - 异步解析 读取伺服行程下限值
//*  参数：
//*     stuRxmsg      - [i] CAN接收消息
//*     *pusId        - [o] 消息对应的节点ID
//*   *pulStrokeL     - [o] 返回行程信下限值首地址，量纲为 电机原始位置传感器/编码器行程值
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t AsynParse_RyParam_GetStroke_L(CanMsg_t stuRxmsg, u16_t* pusId,  u32_t* pulStrokeL);


//*****************************************************************************************************************************
//*函数名称 - AsynParse_RyParam_SetStroke_L
//*函数作用 - 异步解析 设置伺服行程下限值
//*  参数：
//*     stuRxmsg      - [i] CAN接收消息
//*     *pusId        - [o] 消息对应的节点ID
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t AsynParse_RyParam_SetStroke_L(CanMsg_t stuRxmsg, u16_t* pusId);


//*****************************************************************************************************************************
//*函数名称 - AsynParse_RyMotion_ServoMove_Speed
//*函数作用 - 异步解析 指定速度运动
//*  参数：
```

```
//*    stuRxmsg     - [i] CAN接收消息
//*    *pusId       - [o] 消息对应的节点ID
//*   *psutServoData - [o] 返回的伺服数据
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t AsynParse_RyMotion_ServoMove_Speed(CanMsg_t stuRxmsg, u16_t* pusId, ServoData_t* psutServoData);



//*****************************************************************************************************************************
//*函数名称 - AsynParse_RyMotion_ServoMove_Pwm
//*函数作用 - 异步解析 指定PWM运动
//*  参数：
//*    stuRxmsg     - [i] CAN接收消息
//*    *pusId       - [o] 消息对应的节点ID
//*   *psutServoData - [o] 返回的伺服数据
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t AsynParse_RyMotion_ServoMove_Pwm(CanMsg_t stuRxmsg, u16_t* pusId, ServoData_t* psutServoData);



//*****************************************************************************************************************************
//*函数名称 - AsynParse_RyMotion_CurrentMode
//*函数作用 - 异步解析 电流模式运动
//*  参数：
//*    stuRxmsg     - [i] CAN接收消息
//*    *pusId       - [o] 消息对应的节点ID
//*   *psutServoData - [o] 返回的伺服数据
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t AsynParse_RyMotion_CurrentMode(CanMsg_t stuRxmsg, u16_t* pusId, ServoData_t* psutServoData);



//*****************************************************************************************************************************
//*函数名称 - AsynParse_RyMotion_ServoMove_Mix
//*函数作用 - 异步解析 力位混合控制
//*  参数：
//*    stuRxmsg     - [i] CAN接收消息
//*    *pusId       - [o] 消息对应的节点ID
//*   *psutServoData - [o] 返回的伺服数据
//* 返回值 - 参考枚举类型 enret_t 及对应解释
//*****************************************************************************************************************************
Exportmode u8_t AsynParse_RyMotion_ServoMove_Mix(CanMsg_t stuRxmsg, u16_t* pusId, ServoData_t* psutServoData);
```

```
#endif
```