



中置电机测试通信协议规范

[填入文件编号]

编 制：周雄
审 核：
批 准：

武汉天腾动力科技有限公司

二〇二三年五月十一日



修改记录

修改日期	修改人	修改内容	版本
20230109	周雄	基于《测工标定一体机通信协议规范_V3r2_20211110》，针对大牙盘中置电机进行删减，保留助力功能测试，并增加接口功能测试和电机特性测试。	V1.0
20230511	周雄	增加上位机设置电机转速百分比指令，用于调整电机转速。	V1.1



中置电机测试设备通信协议规范

1 适用范围

本文件仅适用于 MOTINOVA 大牙盘中置电机生产测试设备的开发。

2 通信接口

- 1) CAN 总线
- 2) 波特率 125K、250K、500K、1M，要求提供设置界面可选，默认设置为 250K；
- 3) 帧类型：标准帧
- 4) 帧格式：数据帧

3 协议格式

表1 CAN 协议格式

帧头	帧模式	命令段 长度	命令字	数据段	校验位	帧尾
55 AA	读/写/上报	LENGTH	COMMAND	DATA	CRC	F0

其中：

- 1) 帧模式包含读指令 0x11，写指令 0x16，和上报指令 0x0C；
- 2) LENGTH 命令段总长度，占用 1 字节，有效值为 0x02~0xFF；
- 3) COMMAND 为命令字，占用 2 个字节，第 1 字节为命令字序号，第 2 字节为数据段长度；
- 4) DATA 为数据段，长度为 LENGTH - 2；
- 5) CRC 为校验位，占用 4 字节，由帧头开始，CAN_ID 插入到帧头和帧模式之间，计算到数据段最后一个字节，计算方法见附录 1，计算结果高字节在前，如：
CAN_ID 为 0x0715，数据帧为 55 AA 11 03 22 01 00 CRC1 CRC2 CRC3 CRC4 F0，
CRC 计算函数输入数据为 55 AA 07 15 11 03 22 01 00，计算结果依次由高到低写入 CRC1、CRC2、CRC3、CRC4；
- 6) 数据段发送时，采用小端模式；
- 7) 对于长度超过 8Bytes 的帧，采用 8+8+N 的方式，例如长度为 23Bytes 的帧，第一次发送前 8Bytes 数据，第二次发送中间 8Bytes 数据，第三次发送后 7Bytes 数据；



8) 每一包数据发送时需填充对应的 ID。

4 工作流程图

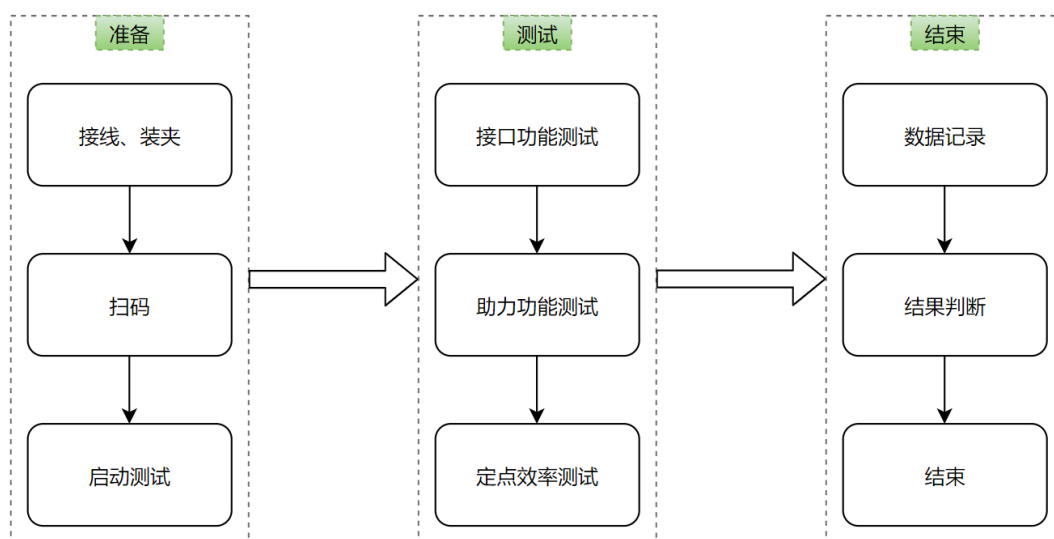


图1 测试简易流程图

上述流程中，中途如果存在异常，中断测试并显示警告信息。

5 协议内容

5.1 接口功能测试

5.1.1 测试目的

测试被测电机的外部接口功能是否正常，包括以下接口：

- 1) 电源接口：输入+、输入-、CAN 通信；
- 2) 仪表接口：输出 12V+、输出 12V-、CAN 通信、开机使能、WALK 信号
- 3) 车速传感器接口：输出 5V+、输出 5V-、输入霍尔信号；
- 4) 前灯接口：输出 6V/12V+、输出 6V/12V-；
- 5) 后灯接口：输出 6V/12V+、输出 6V/12V-；
- 6) 刹车转把接口：输出 5V+、输出 5V-、转把输入信号、刹车输入信号、Gear Sensor 输入信号。

5.1.2 测试步骤

5.1.2.1 步骤一（开机）

从电源接口的输入+和输入-接入 DC36V/48V 电源，仪表接口的开机使能与输入-短接 3s。

5.1.2.2 步骤二（读取电机信息）



发送指令[ID: 0x751, 帧模式: 0x11, 命令字: 0x1200, 数据: 空]查询电机的版本信息, 从接收到的电机版本信息[ID: 0x710, 帧模式: 0x0C, 命令字: 0x1240, 数据: 依次包含电机型号、电机序列号、硬件版本、软件版本共四条信息, 每条信息长度为 16 字符, 按照 ASCII 编码, 结束符为'. ', 无效填充 0x20]中提取电机型号、电机序列号、硬件版本、软件版本等信息, 进行保存。

5.1.2.3 步骤三（电源检测）

- 1) 检测仪表接口的输出 12V+、输出 12V-电压是否正常;
- 2) 检测车速传感器接口的输出 5V+、输出 5V-电压是否正常;
- 3) 检测刹车转把接口的输出 5V+、输出 5V-电压是否正常。

5.1.2.4 步骤四（前后灯检测）

- 1) 发送指令[ID: 0x751, 帧模式: 0x16, 命令字: 0x2802, 数据: 0x00 0xF1]指令打开前后灯;
- 2) 分别检测前灯接口和后灯接口输出电压是否正常(此处电压存在 6V 和 12V 两种配置, 根据配置表判断);
- 3) 发送指令[ID: 0x751, 帧模式: 0x16, 命令字: 0x2802, 数据: 0x00 0xF0]指令关闭前后灯;
- 4) 分别检测前灯接口和后灯接口输出电压是否为 0V。

5.1.2.5 步骤五（车速传感器检测）

- 1) 发送指令[ID: 0x751, 帧模式: 0x16, 命令字: 0x1901, 数据: 0x01]使电机进入配置模式, 此时电机会以 200ms 周期上传电机运行信息, 信息内容见表 2:

表2 电机运行信息

ID	帧模式	命令字	功能	数据段	数据大小 (Byte)	单位
0x710	0x0C	0x1020	电机运行信息	车速	2	km/h
				输出转速	2	rpm
				电功率	2	W, 需放大 2 倍显示
				母线电压	2	mV
				母线电流	2	mA
				踏频	1	rpm
				踩踏力矩	1	N.m
				踩踏方向	1	0: 正向 1: 反向



						2: 停止
				助力档位	1	0x00: OFF 0x01: ECO 0x02: NORM 0x03: SPORT 0x04: TURBO 0x05: 预留 0x22: WALK 0x33: SMART
				大灯开关状态	1	0xF0: 关 0xF1: 开
				剩余电量	1	%
				续航里程	2	km
				ODO 里程	2	km
				平均功耗	1	× 0.01Ah/km
				PCB 温度	1	-40×1℃
				绕组温度	1	-40×1℃
				MCU 温度	1	-40×1℃
				预留	1	-

- 2) 控制磁钢旋转，或从车速传感器接口输入霍尔信号，频率可设为固定值 2Hz，按照上表读取电机运行信息中的车速信息，判断是否异常。

5.1.2.6 步骤六（转把刹车功能检测）

- 1) 发送指令[ID: 0x751, 帧模式: 0x16, 命令字: 0x2802, 数据: 0x04 0xF0]指令打开电机助力功能；
- 2) 从刹车转把接口的转把输入信号输入模拟电压 1.5V，观察电机是否启动运转。
- 3) 使刹车转把接口的刹车输入信号与输出 5V-短路，观察电机是否停止运转；
- 4) 断开刹车转把接口的刹车输入信号与输出 5V-的连接，观察电机是否恢复运转；
- 5) 从刹车转把接口的 Gear Sensor 输入信号输入 200ms 低脉冲，观察电机是否停止 200ms 输出；
- 6) 断开刹车转把接口的转把输入信号输入模拟量的连接，观察电机是否停止运转。

5.1.3 数据记录要求

上述所有测试的结果需要与电机的型号、SN 以及测试时间关联，每个信号的测试结果需要单独标记。



5.2 助力功能测试

5.2.1 测试目的

测试被测电机在设定的工作模式（依次选择 ECO、NORM、SPORT、TURBO、SMART）和负载条件下，电机输出功率和助力比是否满足设定的要求。

5.2.2 测试步骤

5.2.2.1 步骤一（准备）

完成接口功能测试后，保持电机继续工作在配置模式，此时电机仍然按 200ms 周期上报运行信息。

5.2.2.2 步骤二（读取测试配置表）

按照测试配置表依次完成测试，配置表每一行代表一个测试状态，每个测试状态包含以下信息：

- 1) 被测电机工作模式：OFF、ECO、NORM、SPORT、TURBO、SMART；
- 2) 输入电机：转速、最大转矩；
- 3) 负载：转矩
- 4) 时间：保持时间、采集时间

测试配置表参考示例如下：

表3 测试配置表示例

序号	被测电机 工作模式	输入转速	输入最 大转矩	负载转矩	保持时间	采集时间
1	ECO	70	30	30	6	5
2	ECO	80	35	35	6	5
3	SPORT	90	40	40	6	5
4	TURBO	120	45	50	6	5
5	SMART	100	50	45	6	5
6	OFF	90	60	25	6	5
7	NORM	110	35	45	6	5
.....						

上表中每个测试状态可任意设置。

5.2.2.3 步骤三（加载测试）

读取测试配置表后，依次完成每个测试状态的测试，执行流程如下：

- 1) 发送指令[ID: 0x751, 帧模式: 0x16, 命令字: 0x1901, 数据: 0x01]使电机进入配置模式，此时电机将以 200ms 周期上传电机运行信息，信息内容见表 2。



- 2) 根据测试配置表信息，发送指令[ID: 0x751, 帧模式: 0x16, 命令字: 0x2802, 数据: 助力挡位 0x00]使电机进入助力模式，其中助力挡位含义为: 0x00-OFF, 0x01-ECO, 0x02-NORM, 0x03-SPORT, 0x04-TURBO, 0x33-SMART。助力模式设置成功后，表 3 运行信息中助力档位数值相应发生变化；
- 3) 输入电机按照测试配置表信息启动运转；
- 4) 输入转速稳定后，根据测试配置表设定负载；
- 5) 负载控制稳定后，在设定时刻读取传感器信息，并记录到表格中，记录表格参考模板如下：

表4 助力功能测试记录表格模板参考

负载大小	保持时间	采集时间	输入电压	输入电流	电功率	输入转速	输入转矩	输入功率	输出转速	输出转矩	输出功率	效率	助力比	电机测量力矩	电机测量踏频	判断结果
...	...	...	...	...	...	...	...	...	...	...	...	...	...			

- 6) 上表中判断结果需要对以下数值进行判断：输入电流、输出转速、效率、助力比、电机测量力矩、电机测量踏频等，判断依据可另行设定。

5.2.3 数据记录要求

测试配置表中每个测试状态的数据都需要按照表 4 进行记录与判断，记录数据时需包含被测试电机的型号、SN、测试时间等信息。

5.3 定点效率测试

5.3.1 测试目的

测试被测电机在设定负载条件下的输出特性，该测试状态下输入电机保持静止。

5.3.2 测试步骤

5.3.2.1 步骤一（准备）

完成助力功能测试后，保持电机继续工作在配置模式，此时电机仍然按 200ms 周期上报运行信息。

5.3.2.2 步骤二（读取测试配置表）

按照测试配置表依次完成测试，配置表每一行代表一个测试状态，每个测试状态包含以下信息：



- 1) 被测电机：转速
- 2) 负载：转矩
- 3) 时间：保持时间、采集时间

测试配置表参考示例如下：

表5 测试配置表示例

序号	输出转速	负载转矩	保持时间	采集时间
1	30	30	6	5
2	50	35	6	5
3	100	40	6	5
4	90	50	6	5
5	120	45	6	5
6	80	80	6	5
7	20	90	6	5
.....				

上表中每个测试状态可任意设置。

5.3.2.3 步骤三（加载测试）

读取测试配置表后，依次完成每个测试状态的测试，执行流程如下：

- 1) 发送指令[ID: 0x751, 帧模式: 0x16, 命令字: 0x2802, 数据: 0x22 0x00]使电机启动运转；
- 2) 发送指令[ID: 0x751, 帧模式: 0x16, 命令字: 0x2C01, 数据: 百分比]可调节电机转速，100%对应输出转速为 150rpm；
- 3) 电机输出转速稳定后，根据测试配置表设定负载；
- 4) 负载控制稳定后，在设定时刻读取传感器信息，并记录到表格中，记录表格参考模板如下：

表6 定点效率测试记录表格模板参考

负载大小	保持时间	采集时间	输入电压	输入电流	电功率	输出转速	输出转矩	输出功率	效率	判断结果
...	...	...	...	...	...	...	...	...	...	

- 5) 上表中判断结果需要对以下数值进行判断：输入电流、输出转速、效率等，判断依据可另行设定。



- 6) 完成测试配置表中所有测试后，发送指令[ID: 0x751, 帧模式: 0x16, 命令字: 0x2802, 数据: 0x00 0x00]使电机停止运转。

5.3.3 数据记录要求

测试配置表中每个测试状态的数据都需要按照表 6 进行记录与判断，记录数据时需包含被测试电机的型号、SN、测试时间等信息。



6 附录 1：CRC32 计算方法

6.1 CRC32 计算多项式表

```
uint32_t Crc32Table[ 256 ] =
```

```
{
```

```
    0x00000000, 0x04C11DB7, 0x09823B6E, 0x0D4326D9, 0x130476DC, 0x17C56B6B,
    0x1A864DB2, 0x1E475005, 0x2608EDB8, 0x22C9F00F, 0x2F8AD6D6, 0x2B4BCB61,
    0x350C9B64, 0x31CD86D3, 0x3C8EA00A, 0x384FBDBD, 0x4C11DB70, 0x48D0C6C7,
    0x4593E01E, 0x4152FDA9, 0x5F15ADAC, 0x5BD4B01B, 0x569796C2, 0x52568B75,
    0x6A1936C8, 0x6ED82B7F, 0x639B0DA6, 0x675A1011, 0x791D4014, 0x7DDC5DA3,
    0x709F7B7A, 0x745E66CD, 0x9823B6E0, 0x9CE2AB57, 0x91A18D8E, 0x95609039,
    0x8B27C03C, 0x8FE6DD8B, 0x82A5FB52, 0x8664E6E5, 0xBE2B5B58, 0xBAEA46EF,
    0xB7A96036, 0xB3687D81, 0xAD2F2D84, 0xA9EE3033, 0xA4AD16EA, 0xA06C0B5D,
    0xD4326D90, 0xD0F37027, 0xDDB056FE, 0xD9714B49, 0xC7361B4C, 0xC3F706FB,
    0xCEB42022, 0xCA753D95, 0xF23A8028, 0xF6FB9D9F, 0xFBB8BB46, 0xFF79A6F1,
    0xE13EF6F4, 0xE5FFEB43, 0xE8BCCD9A, 0xEC7DD02D, 0x34867077, 0x30476DC0,
    0x3D044B19, 0x39C556AE, 0x278206AB, 0x23431B1C, 0x2E003DC5, 0x2AC12072,
    0x128E9DCF, 0x164F8078, 0x1B0CA6A1, 0x1FCDBB16, 0x018AEB13, 0x054BF6A4,
    0x0808D07D, 0x0CC9CDCA, 0x7897AB07, 0x7C56B6B0, 0x71159069, 0x75D48DDE,
    0x6B93DDDB, 0x6F52C06C, 0x6211E6B5, 0x66D0FB02, 0x5E9F46BF, 0x5A5E5B08,
    0x571D7DD1, 0x53DC6066, 0x4D9B3063, 0x495A2DD4, 0x44190B0D, 0x40D816BA,
    0xACA5C697, 0xA864DB20, 0xA527FDF9, 0xA1E6E04E, 0xBFA1B04B, 0xBB60ADFC,
    0xB6238B25, 0xB2E29692, 0x8AAD2B2F, 0x8E6C3698, 0x832F1041, 0x87EE0DF6,
    0x99A95DF3, 0x9D684044, 0x902B669D, 0x94EA7B2A, 0xE0B41DE7, 0xE4750050,
    0xE9362689, 0xEDF73B3E, 0xF3B06B3B, 0xF771768C, 0xFA325055, 0xFE34DE2,
    0xC6BCF05F, 0xC27DEDE8, 0xCF3ECB31, 0xCBFFD686, 0xD5B88683, 0xD1799B34,
    0xDC3ABDED, 0xD8FBA05A, 0x690CE0EE, 0x6DCDFD59, 0x608EDB80, 0x644FC637,
    0x7A089632, 0x7EC98B85, 0x738AAD5C, 0x774BB0EB, 0x4F040D56, 0x4BC510E1,
    0x46863638, 0x42472B8F, 0x5C007B8A, 0x58C1663D, 0x558240E4, 0x51435D53,
    0x251D3B9E, 0x21DC2629, 0x2C9F00F0, 0x285E1D47, 0x36194D42, 0x32D850F5,
    0x3F9B762C, 0x3B5A6B9B, 0x0315D626, 0x07D4CB91, 0x0A97ED48, 0x0E56F0FF,
    0x1011A0FA, 0x14D0BD4D, 0x19939B94, 0x1D528623, 0xF12F560E, 0xF5EE4BB9,
    0xF8AD6D60, 0xFC6C70D7, 0xE22B20D2, 0xE6EA3D65, 0xEBA91BBC, 0xEF68060B,
    0xD727BBB6, 0xD3E6A601, 0xDEA580D8, 0xDA649D6F, 0xC423CD6A, 0xC0E2D0DD,
    0xCDA1F604, 0xC960EBB3, 0xBD3E8D7E, 0xB9FF90C9, 0xB4BCB610, 0xB07DABA7,
    0xAE3AFBA2, 0xAABFE615, 0xA7B8C0CC, 0xA379DD7B, 0x9B3660C6, 0x9FF77D71,
    0x92B45BA8, 0x9675461F, 0x8832161A, 0x8CF30BAD, 0x81B02D74, 0x857130C3,
    0x5D8A9099, 0x594B8D2E, 0x5408ABF7, 0x50C9B640, 0x4E8EE645, 0x4A4FFBF2,
```



0x470CDD2B, 0x43CDC09C, 0x7B827D21, 0x7F436096, 0x7200464F, 0x76C15BF8,
0x68860BFD, 0x6C47164A, 0x61043093, 0x65C52D24, 0x119B4BE9, 0x155A565E,
0x18197087, 0x1CD86D30, 0x029F3D35, 0x065E2082, 0x0B1D065B, 0x0FDC1BEC,
0x3793A651, 0x3352BBE6, 0x3E119D3F, 0x3AD08088, 0x2497D08D, 0x2056CD3A,
0x2D15EBE3, 0x29D4F654, 0xC5A92679, 0xC1683BCE, 0xCC2B1D17, 0xC8EA00A0,
0xD6AD50A5, 0xD26C4D12, 0xDF2F6BCB, 0xDBEE767C, 0xE3A1CBC1, 0xE760D676,
0xEA23F0AF, 0xEEE2ED18, 0xF0A5BD1D, 0xF464A0AA, 0xF9278673, 0xFDE69BC4,
0x89B8FD09, 0x8D79E0BE, 0x803AC667, 0x84FBDDBD, 0x9ABC8BD5, 0x9E7D9662,
0x933EB0BB, 0x97FFAD0C, 0xAFB010B1, 0xAB710D06, 0xA6322BDF, 0xA2F33668,
0xBCB4666D, 0xB8757BDA, 0xB5365D03, 0xB1F740B4

};

6.2 CRC32 计算方法

```
uint32_t CRC32_Calculate( uint8_t *pData, uint16_t Length )
{
    uint32_t nReg;
    uint32_t nTemp = 0;
    uint16_t i, n;

    nReg = 0xFFFFFFFF;
    for ( n = 0; n < Length; n++ )
    {
        nReg ^= (uint32_t) pData[ n ];
        for ( i = 0; i < 4; i++ )
        {
            nTemp = Crc32Table[ ( uint8_t )( ( nReg >> 24 ) & 0xFF ) ];
            nReg <<= 8;
            nReg ^= nTemp;
        }
    }
    return nReg;
}
```