

威灵中置电机测试设备通信协议

2022/01/11

目的

实现测试设备上位机和被测试电机之间数据双向通信，通过上位机可调节被测试电机的助力档位，并实时显示被测试电机的主要运行信息。

通信接口

CAN2.0A, 250Kbps

协议格式

参考《MOTINOVA中置驱动系统通信协议》中指定的协议格式内容，如下：

帧头	模式	命令段长度	命令字	数据段	CRC32	帧尾
0x55 0xAA	Mode	Len	Command	Data	CRC32	0xF0

其中，

- 1. Mode表示帧模式：0x11-读，0x16-写，0x0C-上报；
- 2. Len表示命令段和数据段总长度，占用1Byte；
- 3. Command表示命令字，占用2Bytes；
- 4. Data表示数据段，长度不固定，Len - 2表示数据段长度，采用小端模式传递数据；
- 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，计算结果高字节在前，如：CAN_ID为0x0710，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 10 11 03 22 01 00，计算结果依次由高到低写入CRC1、CRC2、CRC3、CRC4。

CRC32计算

为保证数据数据双向正确解析，数据发送和解析时统一按照下面计算方法。

```
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, 0xBF41B04B, 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, 0x9FFF77D71,
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, 0x84FBDBD0, 0x9ABC8BD5, 0x9E7D9662,
0x933EB0BB, 0x97FFAD0C, 0xAFB010B1, 0xAB710D06, 0xA6322BDF, 0xA2F33668,
0xBCB4666D, 0xB8757BDA, 0xB5365D03, 0xB1F740B4
};

```

```

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;
}

```

控制流程

1. 上位机点击**握手**按钮，电机返回应答信息，上位机判断和电机握手成功，使能对电机的控制和数据采集按钮；
2. 上位机点击**开始采集**按钮，电机开始定时发送运行信息，上位机接收到运行信息后执行显示实时数据、绘制曲线、存储数据等操作；
3. 上位机点击**停止采集**按钮，电机停止发送运行信息，上位机停止显示实时数据；
4. 测试过程中，可通过上位机调节电机的助力档位，有**0, 1, 2, 3, 4, Walk**共6种模式；

协议内容

1. 上位机发送握手指令

ID	帧头	模式	命令段长度	命令字	数据段	CRC32	帧尾
0x751	0x55 0xAA	0x10	0x02	0xF000	N/A	CRC32	0xF0

2. 电机返回应答指令

ID	帧头	模式	命令段长度	命令字	数据段	CRC32	帧尾
0x715	0x55 0xAA	0x0C	0x02	0xF000	N/A	CRC32	0xF0

3. 上位机发送开始采集指令

ID	帧头	模式	命令段长度	命令字	数据段	CRC32	帧尾
0x751	0x55 0xAA	0x16	0x03	0xF101	0x00	CRC32	0xF0

4. 上位机发送停止采集指令

ID	帧头	模式	命令段长度	命令字	数据段	CRC32	帧尾
0x751	0x55 0xAA	0x16	0x03	0xF101	0x01	CRC32	0xF0

5. 电机定时发送运行信息指令

ID	帧头	模式	命令段长度	命令字	数据段	CRC32	帧尾
0x715	0x55 0xAA	0x0C	0x14	0xF112	Data	CRC32	0xF0

其中Data定义如下：

byte1: 踩踏力矩，单位N.m

byte2: 踩踏方向，0-正向，1-反向，2-停止

byte3: 踏频，单位rpm

byte4: 助力档位，0、1、2、3、4、0x22-Walk byte5: PCB温度，偏移40，单位℃，例如0表示-40℃

byte6: 绕组温度，偏移40，单位℃，例如0表示-40℃

byte7-byte8: 母线电压，单位mV，低位在前

- byte9-byte10: 母线电流，单位mA，低位在前
- byte11-byte12: 电机转速，单位rpm，低位在前
- byte13-byte14: 车速，放大1倍，单位km/h，低位在前
- byte15-byte16: Iq，数据类型为int16
- byte17: 故障码，按位表示，最多8个故障类型
- byte18: 预留

6. 上位机发送调节电机助力档位指令

ID	帧头	模式	命令段长度	命令字	数据段	CRC32	帧尾
0x751	0x55 0xAA	0x16	0x04	0x2802	Data	CRC32	0xF0

其中Data定义如下：

- byte1: 助力档位，0、1、2、3、4、0x22-Walk
- byte2: 预留

7. 电机发送反馈指令

ID	帧头	模式	命令段长度	命令字	数据段	CRC32	帧尾
0x715	0x55 0xAA	0x0C	0x05	0xA903	"ACK"	CRC32	0xF0

---End---