汕头市美思固科技有限公司

美思固
无刷电机驱动器 CAN 通讯协议及寄存器
型号：MSSD-20EHA

说明书

驱控一体，省钱!省心!

目录

1. 概述 ····· 1

2. 通讯协议 ····· 1

2.1 通讯参数 ····· 1

2.2 CAN 帧格式 ····· 1

2.2.1 CAN 消息语法 ····· 1

2.2.2 通过 CAN 访问 MODBUSS 寄存器 ····· 3

3. 附录 ····· 4

3.1 使用 WINDOWS 自带的计算器进行十进制与十六进制转换 ····· 4

4. 免责声明 ····· 8

修订历史：

更新日期	修订作者	版本	主要修订摘要数
20240529	*Nana*	*V1.0*	*初版*

1. 概述

本说明适用于无刷电机驱动器；MSSD-20EHA。

2. 通讯协议

本驱动器采用 CAN 标准数据帧进行通讯（**暂不支持 CANopen 通讯协议**）。

本驱动器支持多节点通讯，支持最多 127 个节点 ID，支持通讯广播（节点 ID 为 0 视为广播帧），客户端可以是单片机、PLC 或 PC 机等。

2.1 通讯参数

默认通讯参数：

- 波特率固定为 500kbps。
- 节点 ID 固定为 1。

用户自定义通讯参数：

- 波特率可配置范围 10kbps~1Mbps。
- 节点 ID 可设置范围 1~127。

2.2 CAN 帧格式

本驱动器只支持 CAN 标准数据帧通讯，驱动器采用 11 位 CAN 标识符的第 0~6 位作为 CAN 节点 ID，第 7~10 位作为 CANOPEN 功能码（普通 CAN 通讯时 7~10 填 0）。驱动器不支持扩展帧和远程帧。

2.2.1 CAN 消息语法

- 主站/客户端发送消息格式：

表 2.1 主站/客户端发送消息格式

CAN 标识符	数据字节 0	数据字节 1-2	数据字节 3	数据字节 4-7
节点 ID	命令	索引号	子索引号	数据

CAN 标识符（共 11 位）：第 0~6 位为节点 ID（0x01~0x7F），每一个 CAN 设备的节点 ID 应唯一节。

数据字节 0：命令码（详见**表 2.2**）。

数据字节 1-2：对象字典索引号，低字节在前。

数据字节 3：对象字典子索引号。

数据字节 4-7：要写入的数据，低字节在前，对于每个无效字节，可用 0x00 填充（详见**表 2.2**）。

***注：**在 CAN 通讯模式下，无法访问数据长度大于 4 字节的对象。*

表 2.2 命令字节与数据字节 4-7 中有效数据字节对应关系

命令码	功能	数据字节 4-7 中有效字节（共 4 个字节）
0x21	写单字节	前 1 个字节

0x22	写双字节	前 2 个字节
0x23	写三字节	前 3 个字节
0x24	写四字节	均有效
0x40	读取数据	均无效

- 从站/服务器应答消息格式：

表 2.3 从站/服务器应答消息格式

CAN 标识符	数据字节 0	数据字节 1-2	数据字节 3	数据字节 4-7
节点 ID	状态码	索引号	子索引号	数据/传送结果代码

CAN 标识符（共 11 位）：第 0~6 位为节点 ID（0x01~0x7F），每一个 CAN 设备的节点 ID 应唯一节。

数据字节 0：状态码（详见**表 2.4**）。

数据字节 1-2：对象字典索引号，低字节在前。

数据字节 3：对象字典子索引号。

数据字节 4-7：读取的数据或传送结果代码，低字节在前（详见**表 2.4**）。

表 2.4 状态码与数据字节 4-7 中有效数据对应关系

状态码	状态码描述	数据字节 4-7 中有效字节（共 4 个字节）
0x41	数据长度为 1 字节	前 1 个字节
0x42	数据长度为 2 字节	前 2 个字节
0x43	数据长度为 3 字节	前 3 个字节
0x44	数据长度为 4 字节	均有效
0x60	传送成功	均无效
0x80	传送中断	4 个字节为中代码

表 2.5 传送中止码

中止码	功能描述
0x05030000	触发位无交替改变
0x05040000	SDO 协议超时
0x05040001	非法或未知的 Client/Server 命令字
0x05040002	无效的块大小（仅 Block Transfer 模式）
0x05040003	无效的序号（仅 Block Transfer 模式）

0x05040004	CRC 错误（仅 Block Transfer 模式）
0x05040005	内存溢出
0x06010000	对象不支持访问
0x06010001	试图读只写对象
0x06010002	试图写只读对象
0x06020000	对象字典中对象不存在
0x06040041	对象不能映射到 PDO
0x06040042	映射的对象的数目和长度超出 PDO 长度
0x06040043	一般性参数不兼容
0x06040047	一般性设备内部不兼容
0x06060000	硬件错误导致访问失败
0x06060010	数据类型不匹配，服务参数长度不匹配
0x06060012	数据类型不匹配，服务参数长度太大
0x06060013	数据类型不匹配，服务参数长度太短
0x06090011	子索引不存在
0x06090030	超出参数的值范围(写访问时)
0x06090031	写入参数数值太大
0x06090032	写入参数数值太小
0x06090036	最大值小于最小值
0x08000000	一般性错误
0x08000020	数据不能传送或保存到应用
0x08000021	由于本地控制导致数据不能传送或保存到应用
0x08000022	由于当前设备状态导致数据不能传送或保存到应用
0x08000023	对象字典动态产生错误或对象字典不存在

2.2.2 通过 CAN 访问 MODBUSS 寄存器

在 CAN 通讯方式下，可通过 Modbus 寄存器地址与对象字典索引号的转换来实现对 Modbus 寄存器的访问。Modbus 寄存器地址与对象字典索引号的转换关系如下：

表 2.6 Modbus 寄存器地址与对象字典索引号的转换关系

MODBUS 寄存器地址（REG_ADDR）	对象字典索引号（INDEX）	对象字典子索引号(SUB-INDEX)
0x0000~0x0FFF	0x4000 + REG_ADDR	0

3. 附录

3.1 使用 Windows 自带的计算器进行十进制与十六进制转换

➢ **使用 Windows XP 自带计算器进行十进制到十六进制的转换步骤如下：**

1) 打开系统自带的计算器，如**图 3.1** 所示。

图 3.1 Windows XP 自带的计算器

2) 选择“查看”–“科学型”菜单项，计算器界面将切换为如**图 3.2** 所示。

图 3.2 计算器界面切换科学型后的界面

3) 点击“十进制”按钮，输入要转换为十六进制的十进制数，我们以-100 为例进行说明，通过右边数字按钮首先输入 100，然后按“+/-”按扭输入负号，如**图 3.3** 所示。

图 3.3 在计算器里输入“-100”

4) 然后点击左边“十六进制”按钮，此时，我们先前输入的十进制数-100 被转换为__int64 类型的整数以十六进制进行显示。如果我们需以 long 型、short 型或 char 型整数的十六进制进行显示，可分别按右边的“双字”、“单字”和“字节”按钮进行显示，显示结果如**图 3.4** 所示。

图 3.4“-100”转换为 short 型并以十六进制显示

➢ **使用 Windows 7 自带计算器进行十进制到十六进制的转换步骤如下：**

1) 打开系统自带的计算器，如**图 3.5** 所示。

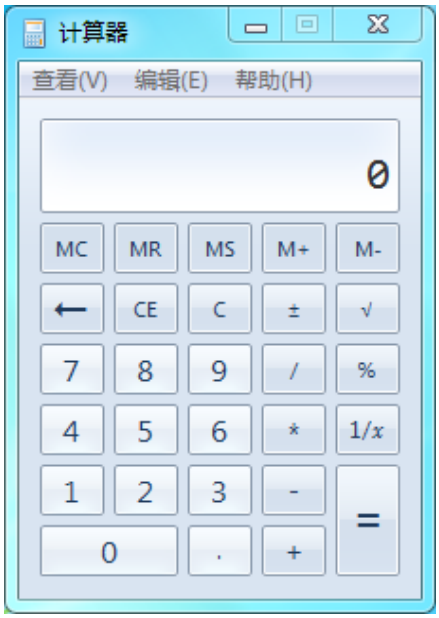

图 3.5Windows 7 自带的计算器

2) 选择“查看”–“程序员”菜单项，计算器界面将切换如**图 3.6** 所示。

图 3.6Windows 7 计算器程序员界面

3) 点击“十进制”按钮，输入要转换为十六进制的十进制数，我们以-100 为例进行说明，通过数字按钮首先输入 100，然后按“±”按扭输入负号，如**图 3.7** 所示。

图 3.7Windows 7 计算器程序员界面输入“-100”

4) 然后点击“十六进制”按钮，此时，我们先前输入的十进制数-100 被转换为__int64 类型的整数以十六进制进行显示。如果我们需以 long 型、short 型或 char 型整数的十六进制进行显示，可分别按左下方的“双字”、“字”和“字节”单选按钮进行显示，如**图 3.8** 所示。

图 3.8 “-100”转换为 short 型并以十六进制显示

4. 免责声明

本文档只是提供本产品的使用说明。本文档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止发言或其它方式授予任何知识产权许可。且本产品的销售和/或使用我们不作任何明示或暗示的担保，包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产权的侵权责任等，均不作担保。本产品为商业级产品，本产品并非设计用于医疗、救生或维生等用途。我们可能随时对产品规格及产品描述做出修改，恕不另行通知。