**磁力计校准板通信协议**

**1.协议说明**

通信协议针对不同的通信方式,采用TCP/IP(网络)协议

TCP/IP格式:

IP默认设置: IP:192.168.2.199 PORT:5000

MAC地址随IP地址改变而改变,局域网内不会重复

协议帧格式:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 报文头 | | | | 功能码 | 数据域 |
| 16-Bits(2Bytes) | 16-Bits(2Bytes) | 16-bits(2Bytes) | 8-bits(1Byte) | 8-bits | N\*8-bits  (N\*1Byte) |
| 事务标识  [0000H] | 协议标识  [0000H] | 长度  (单元标识及以后字段) | 单元标识  [01H] |

**2.寄存器表**

输入寄存器:

00H,01H:为软硬件版本号,其中硬件主版本号:A1,软件主版本号:01

02H,03H:保留

保持寄存器:

04H:保留

05H~07H:网络地址和端口号

08H:保留

09H:运行模式和软件ID寄存器

0AH~0FH:保留

与Bootloader共用部分:

输入寄存器:04H(功能码)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 地址 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 00H  (输入寄存器) | 硬件主版本号  (A:0x41) | | | | | | | | 硬件子版本号  (1:0x31) | | | | | | | |
| 01H  (输入寄存器) | 软件主版本号  (0:0x30) | | | | | | | | 软件子版本号  (1:0x31) | | | | | | | |
| 02H~03H  (输入寄存器) | Reserved | | | | | | | | Reserved | | | | | | | |

保持寄存器:03H,06H,10H(功能码)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 地址 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 04H  (保持寄存器) | Reserved | | | | | | | | Reserved | | | | | | | |
| 05H  (保持寄存器) | IP地址1  (192:0xC0) | | | | | | | | IP地址2  (168:0xA8) | | | | | | | |
| 06H  (保持寄存器) | IP地址3  (2:0x02) | | | | | | | | IP地址4  (199:0xC7) | | | | | | | |
| 07H  (保持寄存器) | 端口高字节  (0x13) | | | | | | | | 端口低字节  (0x88) | | | | | | | |
| 08H  (保持寄存器) | Reserved | | | | | | | | Reserved | | | | | | | |
| 09H  (保持寄存器) | 运行模式:  0x5A = Bootloader;0xA5 = APP | | | | | | | | 软件ID  (默认0x00,可做硬件区分) | | | | | | | |
| 0AH~0FH  (保持寄存器) | Reserved | | | | | | | | Reserved | | | | | | | |

APP独有:

输入寄存器:04H(功能码)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 地址 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 10H~1FH  (输入寄存器) | Reserved | | | | | | | | Reserved | | | | | | | |

保持寄存器:03H,06H,10H(功能码)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 地址 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 20H  (保持寄存器) | MCU\_CTRL\_DMM\_VOL  0 = 关闭,1 = 打开,其他 = 无效操作 | | | | | | | | Reserved | | | | | | | |
| 21H  (保持寄存器) | MCU\_CTRL\_DMM\_CUR  0 = 关闭,1 = 打开,其他 = 无效操作 | | | | | | | | Reserved | | | | | | | |
| 22H  (保持寄存器) | MCU\_RL\_CTRL\_RES1~MCU\_RL\_CTRL\_RES4  0 = 全部关闭;  1=RES1:大电流量程(1A及以上),FIMI;  2=RES2:小电流量程(40mA~400mA),FIMI;  3=RES3:小电流量程(1mA~40mA),FIMI;  4=RES4:小电流量程(μA),FVMI;  其他 = 无效操作 | | | | | | | | Reserved | | | | | | | |
| 23H  (保持寄存器) | 控制全部IO(测试用)  0 = 关闭,1 = 打开,其他 = 无效操作 | | | | | | | | Reserved | | | | | | | |
| 24H~2FH  (保持寄存器) | Reserved | | | | | | | | Reserved | | | | | | | |

**3.应用实例**

**3.1 保持寄存器操作**

**3.1.1 读取保持寄存器(0x03)**

读取一个或多个保持寄存器中的数据.

读取IP地址请求:00 00 00 00 00 06 01 03 00 05 00 02

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能码 | 起始寄存器高位 | 起始寄存器低位 | 寄存器数量高位 | 寄存器数量低位 |
| 03 | 00 | 05 | 00 | 02 |

从起始寄存器为0005H开始,读取2个寄存器(05,06)数据:

读取IP地址响应:00 00 00 00 00 07 01 03 04 C0 A8 02 C6

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能码 | 返回数据数量 | 数据1高位 | 数据1低位 | 数据2高位 | 数据2低位 |
| 03 | 04 | C0 | A8 | 02 | C6 |

其中数据1到数据2,对应寄存器0005H到0006H的值.

**3.1.2 写单个保持寄存器(0x06)**

将一个特定的数值写入一个单寄存器中.

打开MCU\_CTRL\_DMM\_VOL开关请求:00 00 00 00 00 06 01 06 00 20 01 00

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能码 | 起始寄存器高位 | 起始寄存器低位 | 数值高位 | 数值低位 |
| 06 | 00 | 20 | 01 | 00 |

向0020H寄存器写入0100H(打开MCU\_CTRL\_DMM\_VOL开关).

正确响应:00 00 00 00 00 06 01 06 00 20 01 00(把请求指令原样返回).

**3.1.2 写多个保持寄存器(0x10)**

将多个数值写入多个寄存器中.

同时打开MCU\_CTRL\_DMM\_CUR开关和MCU\_RL\_CTRL\_RES1开关请求:

00 00 00 00 00 0B 01 10 00 21 00 02 04 01 00 01 00

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 功能码 | 起始寄存器高位 | 起始寄存器低位 | 寄存器数量高位 | 寄存器数量低位 | 数据数量 | 数据1  高位 | 数据1  低位 | 数据2  高位 | 数据2低位 |
| 10 | 00 | 21 | 00 | 02 | 04 | 01 | 00 | 01 | 00 |

向0021H寄存器中写入0100H;向0022H寄存器中写入0100H.

本协议中为打开MCU\_CTRL\_DMM\_CUR开关和MCU\_RL\_CTRL\_RES1开关.

响应:00 00 00 00 00 06 01 10 00 21 00 02

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能码 | 起始寄存器高位 | 起始寄存器低位 | 寄存器数量高位 | 寄存器数量低位 |
| 10 | 00 | 21 | 00 | 02 |

**3.2 输入寄存器操作**

输入寄存器只有读取操作(功能码04H),与读取一个或多个保持寄存器(功能码03H)操作类似.

读取软硬件版本号请求:00 00 00 00 00 06 01 04 00 00 00 02

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能码 | 起始寄存器高位 | 起始寄存器低位 | 寄存器数量高位 | 寄存器数量低位 |
| 04 | 00 | 00 | 00 | 02 |

从起始寄存器为0000H开始,读取2个寄存器(00,01)数据:

响应:00 00 00 00 00 07 01 04 04 41 31 30 31

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能码 | 返回数据数量 | 数据1高位 | 数据1低位 | 数据2高位 | 数据2低位 |
| 04 | 04 | 41 | 31 | 30 | 31 |

其中数据1到数据2,对应寄存器0000H到0001H的值.

**3.3 逻辑线圈DO操作**

**3.3.1 读取线圈状态(0x01)(有问题,待修改)**

读取一个或多个线圈状态,线圈的起始地址为0000H,1~6个线圈的寻址地址分别为0000H

~0006H(DO1 = 0000H,DO2 = 0001H,以此类推).

读取所有开关状态请求:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能码 | 起始寄存器高位 | 起始寄存器低位 | 寄存器数量高位 | 寄存器数量低位 |
| 01 | 00 | 00 | 00 | 06 |

读取DO1到DO6的开关状态.

**3.3.2 强置单线圈(0x05)**

强置一个单线圈的类型,线圈的起始地址为0000H,1~6个线圈的寻址地址分别为0000H

~0006H(DO1 = 0000H,DO2 = 0001H,以此类推).FF00H值请求线圈处于ON状态,0000H值请求线圈处于OFF状态,其他值对线圈无效,不起作用.

打开MCU\_CTRL\_DMM\_VOL开关请求:00 00 00 00 00 06 01 05 00 00 FF 00

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能码 | 起始寄存器高位 | 起始寄存器低位 | 数值高位 | 数值低位 |
| 06 | 00 | 20 | 01 | 00 |

设置MCU\_CTRL\_DMM\_VOL开关为ON(置1)状态.

正确响应:正确响应:00 00 00 00 00 06 01 05 00 00 FF 00(把请求指令原样返回).

**3.3.3 强置多线圈(0x0F)**

强置多个线圈,线圈的起始地址为0000H,1~6个线圈的寻址地址分别为0000H~0006H

(DO1 = 0000H,DO2 = 0001H,以此类推).数据区的某位值为”1”表示请求的相应项圈状态为ON,位值为”0”,则为OFF状态.

同时打开MCU\_CTRL\_DMM\_CUR开关和MCU\_RL\_CTRL\_RES1开关请求:

00 00 00 00 00 08 01 0F 00 00 00 06 01 06

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 功能码 | 起始寄存器高位 | 起始寄存器低位 | 寄存器数量高位 | 寄存器数量低位 | 数据数量 | 数据1 |
| 10 | 00 | 00 | 00 | 06 | 01 | 06 |

强置DO1~DO6,其中DO6~DO4以及DO1为OFF(低电平),DO2~DO3为ON(高电平)

正确响应:00 00 00 00 00 06 01 0F 00 00 00 06

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能码 | 起始寄存器高位 | 起始寄存器低位 | 寄存器数量高位 | 寄存器数量低位 |
| 0F | 00 | 00 | 00 | 06 |

**4.错误码说明**

当请求指令或设备执行出错时,返回出错响应,格式如下:

|  |  |  |
| --- | --- | --- |
| 地址 | 错误功能码 | 错误码 |
| 8-bits(1Byte) | 8-bits(1Byte) | 8-bits(1Byte) |

其中

地址:从机地址;

错误功能码:将请求功能码高位置1(功能码 | 0x80);

错误码如下:

|  |  |  |  |
| --- | --- | --- | --- |
| 代码 | 名称 | 含义 | 举例 |
| 01 | 非法功能 | 设备中无该功能码的操作 | 使用01,02功能码操作系统寄存器 |
| 02 | 非法数据地址 | 操作地址超出设备设定范围 | 操作46H及以后的寄存器 |
| 03 | 非法数据值 | 设备无法响应该数据 | 波特率设置1111B |
| 04 | 从站设备故障 | 从站执行出错 | I2C读写失败 |

例:

同时打开MCU\_CTRL\_DMM\_CUR开关和MCU\_RL\_CTRL\_RES1开关请求:

00 00 00 00 00 08 01 10 00 00 00 06 01 06

返回:

00 00 00 00 00 03 01 90 03

错误码03:非法数据值,设备无法响应该数据