CCH0601称重模块

**1 技术参数以及外形尺寸**

|  |  |
| --- | --- |
| 称重信号 | 0-20mV输入范围；DC5V激励电压，可选1~4mV/V的传感器  每一路最多可连接4个350欧姆的传感器 |
| 工作温度 | -10-70℃ |
| 电源 | 24VDC |
| 重量 | 约0.1kg |
| 外形尺寸 | 101\*49\*25.8 |
| 功耗 | 1.5W |
| 温漂 | ±5ppm/℃ |
| 零飘 | ±0.1μV |
| 综合线性误差 | 0.005%F.S(满量程) |

**2接口定义**说明

1：L1-L3为LED指示灯， L1为闪烁，重量不稳,其他状态为稳定；L2快闪，232口为远程屏幕模式；L2慢闪，232口有接收无发送；L2常亮，232口有发送和接收；L3快闪，485口为远程屏幕模式；L3慢闪，485口有接收无发送；L3常亮，485口有发送和接收

2：R、T、G为232接口，可分别接电脑232口的3、2、5脚，A、B为485接口；

3：E-、E+分别对应传感器的激励-、激励+；

4：S1+、S1-、S2+、S2-、S3+、S3-、S4+、S4-、分别对应1-4号传感器的信号-、信号+；

5：DC+、DC-为仪表供电，需给仪表提供DC24V。

**3 通讯协议**

**通讯默认为19200波特率，偶校验，modbus rtu 协议**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能 | 数据类型 | 缺省(范围) | 描述 | 寄存器地址 |
| 第1路重量 | Long[读] |  | 写入0，为校零功能，写入非0，为校满功能。 | 1 |
| 第2路重量 | Long[读] |  |  | 3 |
| 第3路重量 | Long[读] |  |  | 5 |
| 第4路重量 | Long[读] |  |  | 7 |
| 第1路采样值 | Long[读] |  |  | 9 |
| 第2路采样值 | Long[读] |  |  | 11 |
| 第3路采样值 | Long[读] |  |  | 13 |
| 第4路采样值 | Long[读] |  |  | 15 |
| 第1路操作 | Long[读/写] |  | 写入0，为校零功能，写入非0，为校满功能。 | 17 |
| 第2路操作 | Long[读/写] |  | 写入0，为校零功能，写入非0，为校满功能。 | 19 |
| 第3路操作 | Long[读/写] |  | 写入0，为校零功能，写入非0，为校满功能。 | 21 |
| 第4路操作 | Long[读/写] |  | 写入0，为校零功能，写入非0，为校满功能。 | 23 |
| 地址 | Long[读/写] |  | 写入（寄存器地址-1） | 25 |
| 参数 | Long[读/写] |  | 可读写上面地址对应的值。该值无小数点。 | 27 |
| 第1路零点 | Long[读/写] |  |  | 51 |
| 第2路零点 | Long[读/写] |  |  | 53 |
| 第3路零点 | Long[读/写] |  |  | 55 |
| 第4路零点 | Long[读/写] |  |  | 57 |
| 第1路系数 | Long[读/写] |  | 采样值与重量的转换系数 | 59 |
| 第2路系数 | Long[读/写] |  | 采样值与重量的转换系数 | 61 |
| 第3路系数 | Long[读/写] |  | 采样值与重量的转换系数 | 63 |
| 第4路系数 | Long[读/写] |  | 采样值与重量的转换系数 | 65 |
| 小数点 | Long[读/写] | 2 (0-3) | 可由模块来存储小数点值，本身无意义 | 67 |
| 分度值 | Long[读/写] | 0(0-5) | 0:1 1:2 2:5 3:10 4:20 5:50 | 69 |
| 稳定检查范围 | Long[读/写] | 0.01(0-9999.99) | 在稳定检测时间内，显示值一直小于该范围，则认为稳定。 | 71 |
| 稳定检查时间 | Long[读/写] | 1.00(0.00-9.99)s | 73 |
| 自动置零范围 | Long[读/写] | 0.00(0-9999.99) | 在自动置零时间内，显示值一直小于该范围，且稳定，则置零。设置为0该功能无效。 | 75 |
| 自动置零时间 | Long[读/写] | 0.50(0.00-9.99)s | 77 |
| 滤波值 | Long[读/写] | 16(0-19) | 设置越大越稳定，但是响应时间变长 | 79 |
| 模块地址 | Long[读/写] | 1(0-128) |  | 81 |
| 单位 | Long[读/写] | 0(0-10) | 可由模块来存储单位值，本身无意义 | 93 |
| 功能 | Long[读/写] | 0(0-10) | 备用 | 95 |
| 第1路上限 | Long[读/写] | 10000(0-999999) | 第1路重量大于该值有指示 | 97 |
| 第2路上限 | Long[读/写] | 10000(0-999999) | 第2路重量大于该值有指示 | 99 |
| 第3路上限 | Long[读/写] | 10000(0-999999) | 第3路重量大于该值有指示 | 101 |
| 第4路上限 | Long[读/写] | 10000(0-999999) | 第4路重量大于该值有指示 | 103 |
| 第1路下限 | Long[读/写] | 1000(0-999999) | 第1路重量小于该值有指示 | 105 |
| 第2路下限 | Long[读/写] | 1000(0-999999) | 第2路重量小于该值有指示 | 107 |
| 第3路下限 | Long[读/写] | 1000(0-999999) | 第3路重量小于该值有指示 | 109 |
| 第4路下限 | Long[读/写] | 1000(0-999999) | 第4路重量小于该值有指示 | 111 |

注意:其他参数的通讯见参数表中的寄存器地址。

读为03命令；当数据类型为UShort，写为06命令，否则为10命令。

## 2.2 连续发送通讯协议

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 起始符 | 第1路符号[+/-] | 第1路数据 | ... | 第4路符号[+/-] | 第4路数据 | 小数点[0-3] | 异或校验 | 结束 |
| 0x02 | 0x2B/0X2D | 6个字节 | ... | 0x2B/0X2D | 6个字节 | 0x30-0x33 | 2个字节 | 0xFF |

1:数据采用ASCII码进行传递。例如显示为1234，则传递16进制30 30 31 32 33 34

2:异或校验为之前的除去起始符的所有数据进行异或运算，会得到一个字节的数据，然后把这个字节转换为两个ASCII码，例如，计算得到的校验为0x4A，其对饮的16进制ASCII为34 41。

本公司地址采用西门子系统地址描述规则，实际发送指令，指令为16进制，地址需要减1。

**主机对从机读数据操作**  
主机进行读1号寄存器32位的数据 操作，则报文是：   
    01             03            00 00           00 02          C4 0B    
 从机地址        功能号          数据地址      读取数据个数       CRC校验   
那么单片机接收到这串数据根据数据计算CRC校验判断数据是否正确，如果判断数据无误，则结果是：返回信息给主机，返回的信息也是有格式的：   
返回内容：     
    01         03            04             00 01 E2 40          E2 A3   
  从机地址   功能号     数据字节个数    四个字节数据    CRC校验

四个16进制数据转换为10进制，就是123456

**主机对从机写数据操作**

主机进行写1号寄存器32位的数据 操作

写入砝码重量123456，则报文是：   
    01          10       00 00         00 02       04   00 01 E2 40  EB 3F    
 从机地址    功能号   数据地址     寄存器数量  字节数    四个字节数据   CRC校验

校零操作，则报文是：   
    01          10       00 00         00 02       04   00 00 00 00  F3 AF    
 从机地址    功能号   数据地址     寄存器数量  字节数    四个字节数据   CRC校验

返回内容：

01          10       00 00         00 02         41 C8   
 从机地址    功能号   数据地址     寄存器数量     CRC校验

**Modbus RTU CRC校验码计算方法**

在CRC计算时只用8个数据位，起始位及停止位，如有奇偶校验位也包括奇偶校验位，都不参与CRC计算。

CRC计算方法是：

1、  加载一值为0XFFFF的16位寄存器，此寄存器为CRC寄存器。

2、  把第一个8位二进制数据（即通讯信息帧的第一个字节）与16位的CRC寄存器的相异或，异或的结果仍存放于该CRC寄存器中。

3、  把CRC寄存器的内容右移一位，用0填补最高位，并检测移出位是0还是1。

4、  如果移出位为零，则重复第三步（再次右移一位）；如果移出位为1，CRC寄存器与0XA001进行异或。

5、  重复步骤3和4，直到右移8次，这样整个8位数据全部进行了处理。

6、  重复步骤2和5，进行通讯信息帧下一个字节的处理。

7、  将该通讯信息帧所有字节按上述步骤计算完成后，得到的16位CRC寄存器的高、低字节进行交换

8、  最后得到的CRC寄存器内容即为：CRC校验码

代码：

///<summary>

/// 转换成CRC码

//modbus CRC16

publicvoid CRC16Calc(byte[] dataBuff, int dataLen)

{

int CRCResult = 0xFFFF;

if (dataLen < 2)

{

   return;

}

for (int i = 0; i < (dataLen - 2); i++)

{

    CRCResult = CRCResult ^ dataBuff[i];

for (int j = 0; j < 8; j++)

{

if ((CRCResult & 1) == 1)

CRCResult = (CRCResult >> 1) ^ 0xA001;

else CRCResult >>= 1;

}

}

dataBuff[dataLen - 1] =Convert.ToByte(CRCResult >> 8);

dataBuff[dataLen - 2] =Convert.ToByte(CRCResult & 0xff);

}