

## 1、PLC 寄存器地址（编号） 非工控屏应用的客户不需要了解这段

一般 PLC、信捷/昆仑通泰/威纶触摸屏，或文本显示器采用这样的方式表示数据地址。PLC 寄存器地址一般采用 **10 进制**描述，共有 5 位，其中**第一位数字表示寄存器类型**。第一位数字和寄存器类型的对应关系如下表所示。PLC 寄存器地址例如 40001、00001 等。

数据类型	PLC 寄存器地址（编号）	读功能码	写功能码	数据大小
数字量输出(线圈)	00001——09999	01H	05H, 0FH	位, 1bit
数字量输入(触点)	10001——19999	02H		位, 1bit
输入寄存器	30001——39999	04H		字, 16bit
保持寄存器	40001——49999	03H	06H, 10H	字, 16bit

## 2、协议地址（一般**十六进制**表示，一个地址占 2 个字节）

即本控制器内部对应的数据寄存器地址，例如 PLC 保持寄存器地址 40001 对应协议地址 0x0000，40002 对应协议地址 0x0001，40012 对应协议地址 0x000b，再如 PLC 线圈寄存器地址 00003 对应协议地址 0x0002，00013 对应协议地址 0x000c，保持寄存器地址和线圈寄存器地址对应我们控制器 2 块独立的地址上，不会有访问冲突。

**<PLC 寄存器地址减 1，然后再转换成十六进制数，就是协议地址>**

**<特别提醒： 协议地址，也就是我们控制器内部的地址。 >**

**4800、9600、19200、115200、38400 通信波特率可选择。**

## 3、默认通信参数：**9600 波特率 8 位数据位 1 位停止位 无校验**

（大端模式表示地址和数据项；而 CRC16 是低位在前，高位在后。）

（485 接口通信时，建议每帧数据响应时间不能低于 35ms）

#### 4、支持的功能码(十六进制表示): 不支持 01 02 0F

- 03: 读多个保持寄存器的内容(连续寄存器块)
- 05: 写单个线圈
- 06: 写单个保持寄存器
- 10: 写多个保持寄存器(连续寄存器块)

#### 5、保持数据寄存器: (用来存放和显示数据)

(1 个寄存器是 16 位的无符号数, 占 2 个字节)

PLC 或工控屏对应寄存器编号	定义(对应参数)	读/写	本控制器内部对应的数据寄存器地址(协议地址)
40001	步距角 (比如步距角是 1.8, 写的时候需要扩大 100 倍, 即设置为 180。读的时候缩小 100 倍)	R/W	0x0000
40002	细分 (驱动器上是多少细分, 就设为多少细分)	R/W	0x0001
40003	启动频率(单位: HZ)	R/W	0x0002
40004	加减频率(单位: HZ)	R/W	0x0003
40005	方向 (0 正向, 1 反向)	R/W	0x0004
40006	速度(单位: 转/每分钟)	R/W	0x0005
40007 40008	螺距 (电机转一圈对应的距离)	R/W (低 16 位在前)	0x0006、0x0007
40009 40010	距离(即单次运行的距离)	R/W (低 16 位在前)	0x0008、0x0009
40011	步距角 (比如步距角是 1.8, 写的时候需要扩大 100 倍, 即设置为 180。读的时候缩小 100 倍)	R/W	0x000a
40012	细分 (驱动器上是多少细分, 就设为多少细分)	R/W	0x000b
40013	启动频率(单位: HZ)	R/W	0x000c
40014	加减频率(单位: HZ)	R/W	0x000d
40015	方向 (0 正向, 1 反向)	R/W	0x000e
40016	速度(单位: 转/每分钟)	R/W	0x000f
40017 40018	螺距 (电机转一圈对应的距离)	R/W (低 16 位在前)	0x0010、0x0011
40019 40020	距离(即单次运行的距离)	R/W (低 16 位在前)	0x0012、0x0013
40021	步距角 (比如步距角是 1.8, 写的时候需要扩大 100 倍, 即设置为 180。读的时候缩小 100 倍)	R/W	0x0014
40022	细分 (驱动器上是多少细分, 就设为多少细分)	R/W	0x0015
40023	启动频率(单位: HZ)	R/W	0x0016
40024	加减频率(单位: HZ)	R/W	0x0017
40025	方向 (0 正向, 1 反向)	R/W	0x0018
40026	速度(单位: 转/每分钟)	R/W	0x0019
40027 40028	螺距 (电机转一圈对应的距离)	R/W (低 16 位在前)	0x001a、0x001b
40029 40030	距离(即单次运行的距离)	R/W (低 16 位在前)	0x001c、0x001d

40051	485 设备号	R/W	0x0032
40052 40053	485 通信波特率	R/W (低 16 位在前)	0x0033、0x0034
40081	X 轴正向限位信号	R/W	0x0050
40082	X 轴反向限位信号	R/W	0x0051
40083	X 轴机械零点信号	R/W	0x0052
40084	Y 轴正向限位信号	R/W	0x0053
40085	Y 轴反向限位信号	R/W	0x0054
40086	Y 轴机械零点信号	R/W	0x0055
40087	Z 轴正向限位信号	R/W	0x0056
40088	Z 轴反向限位信号	R/W	0x0057
40089	Z 轴机械零点信号	R/W	0x0058
40105	工程启动信号	R/W	0x0068
40106	工程停止信号	R/W	0x0069
40107	工程暂停信号	R/W	0x006a
40108 40109	X 轴当前坐标	R (低 16 位在前)	0x006b、0x006c
40110 40111	Y 轴当前坐标	R (低 16 位在前)	0x006d、0x006e
40112 40113	Z 轴当前坐标	R (低 16 位在前)	0x006f、0x0070
40124—40135	1—12 路输入状态	R	0x007b—0x0086
40150—40155	1—6 路输出状态	R	0x0095—0x009a
40156	ADC1 电压采集值	R	0x009b
40157	ADC2 电压采集值	R	0x009c
40158	ADC3 电压采集值	R	0x009d
40168	X 轴运行状态(0 静止 1 运行)	R	0x00a7
40169	Y 轴运行状态(0 静止 1 运行)	R	0x00a8
40170	Z 轴运行状态(0 静止 1 运行)	R	0x00a9
40176	DAC 输出调试电压值	R/W	0x00af
40177	第 1 个回机械零设定(0—3) 0 表示无设置, 1 表示 X 轴, 以此类推	R/W	0x00b0
40178	第 2 个回机械零设定(0—3)	R/W	0x00b1
40179	第 3 个回机械零设定(0—3)	R/W	0x00b2
	以下是工程相关的参数		
40257 40258	系统工作次数 (即工程运行次数计数)	R/W (低 16 位在前)	0x0100、0x0101
40259	工程运行状态显示	R	0x0102
40260	工程号	R/W	0x0103
40261	工程总步数	R/W	0x0104
40262	若输入	R/W	0x0105
40263	运行/停止	R/W	0x0106
40264	运行模式	R/W	0x0107
40265	坐标方式	R/W	0x0108
40266	方向	R/W	0x0109
40267	速度	R/W	0x010a
40268 40269	距离	R/W (低 16 位在前)	0x010b、0x010c

40270	若输入	R/W	0x010d
40271	运行/停止	R/W	0x010e
40272	运行模式	R/W	0x010f
40273	坐标方式	R/W	0x0110
40274	方向	R/W	0x0111
40275	速度	R/W	0x0112
40276 40277	距离	R/W (低 16 位在前)	0x0113、0x0114
40278	若输入	R/W	0x0115
40279	运行/停止	R/W	0x0116
40280	运行模式	R/W	0x0117
40281	坐标方式	R/W	0x0118
40282	方向	R/W	0x0119
40283	速度	R/W	0x011a
40284 40285	距离	R/W (低 16 位在前)	0x011b、0x011c
40351	若输入	R/W	0x015e
40326 40327	延时	R/W (低 16 位在前)	0x0145、0x0146
40352	第 1 路输出口	R/W	0x015f
40354	输出电平	R/W	0x0161
40355	若输入	R/W	0x0162
40331 40332	延时	R/W (低 16 位在前)	0x014a、0x014b
40356	第 2 路输出口	R/W	0x0163
40358	输出电平	R/W	0x0165
40359	若输入	R/W	0x0166
40336 40337	延时	R/W (低 16 位在前)	0x014f、0x0150
40360	第 3 路输出口	R/W	0x0167
40362	输出电平	R/W	0x0169
40386	本步是否 DA 输出使能	R/W	0x0181
40390	本步获得信号 Lx	R/W	0x0185
40371	本步输出电压值	R/W	0x0172
40372 40373	本步运行完延时时间	R/W (低 16 位在前)	0x0173、0x0174
40374	起始段	R/W	0x0175
40375	结束段	R/W	0x0176
40376 40377	起始段到结束段的循环次数	R/W (低 16 位在前)	0x0177、0x0178
40383	本工程是否循环	R/W	0x017e
40385	自动运行的下一个工程号	R/W	0x0180
40380	设定工程第几步	R/W	0x017b
40384	本步是否配置暂停	R/W	0x017f
40381	设定工程从第几步开始运行	R/W	0x017c

## 6、线圈输出寄存器 (用来控制操作)

线圈输出值表示请求的 ON/OFF 状态。十六进制值 0xFF00 请求线圈为 ON；十六进制值 0x0000 请求线圈为 OFF。其它所有值均为非法的，并且对线圈不起作用。

PLC 或工控屏对应寄存器编号	定义(对应参数)	说明	本控制器内部对应的线圈寄存器地址 (协议地址)
00001	X 轴单次运行		0x0000

00002	正转	置 ON, 电机一直正 置 OFF, 电机停止	0x0001
00003	反转	置 ON, 电机一直反 置 OFF, 电机停止	0x0002
00004	回机械零		0x0003
00005	急停		0x0004
00006	Y 轴单次运行		0x0005
00007	正转	置 ON, 电机一直正 置 OFF, 电机停止	0x0006
00008	反转	置 ON, 电机一直反 置 OFF, 电机停止	0x0007
00009	回机械零		0x0008
00010	急停		0x0009
00011	Z 轴单次运行		0x000a
00012	正转	置 ON, 电机一直正 置 OFF, 电机停止	0x000b
00013	反转	置 ON, 电机一直反 置 OFF, 电机停止	0x000c
00014	回机械零		0x000d
00015	急停		0x000e
00031	电机 1 坐标清零		0x001e
00032	电机 2 坐标清零		0x001f
00033	电机 3 坐标清零		0x0020
00034	电机 1 回数据零		0x0021
00035	电机 2 回数据零		0x0022
00036	电机 3 回数据零		0x0023
00046	按设定顺序回机械零		0x002d
00047	工程参数保存		0x002e
00048	工程参数读取		0x002f
00053—00064	输出 1—输出 6 开/关		0x0034—0x003F
00089	DAC 输出测试		0x0058
00090	工程上一步		0x0059
00091	工程下一步		0x005a
00092	工程参数删除		0x005b
00093	备用		0x005c
00094	工程另存		0x005d
00095	本步暂停配置		0x005e
00096	基本参数保存		0x005f
00097	基本参数清零		0x0060
00098	基本参数初始化		0x0061
00099	1—6 输出全开		0x0062
00100	1—6 输出全关		0x0063

## 7、通信实例说明

(1) 使用 03 功能码读取 2 个寄存器 40001H 40002H 中的数据内容。即步距角细分值

设备号/站号 (1 个字节)	功能码 (1 个字节)	数据起始地址 (2 个字节,高位在前)	读寄存器个数 (2 个字节,高位在前)	CRC 校验 (2 个字节,低位在前)
-------------------	----------------	------------------------	------------------------	------------------------

01	03	00 00	00 02	C4 0B
----	----	-------	-------	-------

回应信息格式： 回字节个数=5+2\*N N 为读的寄存器个数

设备号/站号 (1 个字节)	功能码 (1 个字节)	数据字节个数 (1 个字节)	回数据内容(高位在前) 40001 地址的数据 40002 地址的数据	CRC 校验
01	03	04	00h B4h 00h 08h	BBH D3H

(2) 写单个线圈 05 功能码 (比如：控制电机单次运行的命令。地址是 00001) 请求数据域中的常量说明请求的 ON/OFF 状态。十六进制值 FF 00 请求输出为 ON。十六进制值 00 00 请求输出为 OFF。其它所有值均是非法的，并且对输出不起作用

设备号/站号 (1 个字节)	功能码 (1 个字节)	线圈输出地址 (2 个字节,高位在前)	输出值 (2 个字节)	CRC 校验 (2 个字节)
01	05	00 00	ff 00	8C 3A

回信息格式： 和发送的数据一样。 回字节个数=8 个

(3) 写单个保持寄存器 06 功能码 (比如：设定细分值设为 4。地址是 40002)

设备号/站号 (1 个字节)	功能码 (1 个字节)	数据地址 (2 个字节,高位在前)	数据内容 (2 个字节)	CRC 校验 (2 个字节)
01	06	00 01	00 04	D9 C9

回信息格式： 和发送的数据一样。 回字节个数=8 个

(4) 写多个寄存器 10 功能码

(比如：设定运行距离的值为 200，等于十六进制 0x00c8。)

地址 40007 对应低 16 位数据，40008 对应高 16 位数据；

设备号/站号 (1 个字节)	功能码 (1 个字节)	数据起始地址 (2 个字节,高位在前)	寄存器个数 (2 个字节)	数据字节个数 (1 个字节)	数据内容 数据 1 数据 2,.....	CRC 校验 (2 个字节)
01	10	00 06	00 02	04	00 C8 00 00	A5 32

回应信息格式： 回字节个数=8 个

设备号/站号 (1 个字节)	功能码 (1 个字节)	数据起始地址 (2 个字节,高位在前)	寄存器个数 (2 个字节)	CRC 校验 (2 个字节)
01	10	00 06	00 02	A1 C9

**注意 1：** 修改 485 设备地址或者修改通信波特率的操作如下：

出厂默认设备号为 1。首先使用 06 功能码或 10 功能码写设备号，然后再使用 05 功能码发送 1 条数据保存命令，再断电重启。

**注意 2：** 读/写一个 32 位的参数(即占 2 个寄存器)时，低 16 位在前，

高 16 位在后。比如螺距、运行距离、当前坐标。

这几个参数也可以带 2 位小数。读取或写入时需要缩小/扩大 100

### (5) 直接举例

读输入 1 状态: 01 03 00 7B 00 01 F4 13      00 7B 是输入 1 状态地址

读电机当前坐标: 01 03 00 6B 00 02 B5 D7      00 6B 是电机 1 坐标地址

因为坐标是 32 位数, 占 2 个寄存器, 所以要读 2 个地址。

电机停止命令(线圈地址 00 04 置 ON): 01 05 00 04 ff 00 CD FB

线圈置 ON 和置 OFF, 可以实现类似按钮点动控制的操作。

置 ON 类似按钮按下 (启动电机), 置 OFF 类似按钮松开(停止电机)。

电机正转命令(线圈地址 00 01 置 ON): 01 05 00 01 ff 00 DD FA

(线圈地址 00 01 置 OFF): 01 05 00 01 00 00 9C 0A

电机反转命令(线圈地址 00 02 置 ON): 01 05 00 02 ff 00 2D FA

(线圈地址 00 02 置 OFF): 01 05 00 02 00 00 6C 0A

设置速度命令: 01 06 00 05 00 32 18 1E      00 05 是地址,00 32 表示速度为 50

设置方向为正向: 01 06 00 04 00 00 C8 0B      00 04 是地址, 00 00 表示正向

设置运行距离的命令, 请看上面第 4 点举例说明。

设置好速度和方向、距离之后, 再发一条单次运行的命令, 就可以对电机进行定位控制。

若需要设置的参数, 断电保存在控制器里面, 需要发一条数据保存命令。

01 05 00 5F ff 00 BC 28

## 8、CRC16 校验 计算方法 <C 语言>

```
unsigned int crc_chk(unsigned char* data,unsigned char length)
```

```
{  
    int j;  
    unsigned int reg_crc=0xffff;  
    while(length--)  
    {  
        reg_crc^=data++;  
        for(j=0;j<8;j++)  
        {  
            if(reg_crc&0x01)  
            {  
                reg_crc=(reg_crc>>1)^0xA001;  
            }  
            else  
            {  
                reg_crc=reg_crc>>1;  
            }  
        }  
    }  
    return reg_crc;  
}
```