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>

<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
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　40006	螺距 **(电机转一圈对应的距离)**	R/W（低 16 位在前）	0x0004、0x0005
40007	**机械零点信号(有效值 0--5)**	R/W	0x0006
40008	停止模式 **(0 缓慢停　　1 立即停)**	R/W	0x0007
40009	速度(单位：转/每分钟)	R/W	0x0008
40010　40011	距离**(即单次运行的距离)**	R/W（低 16 位在前）	0x0009、0x000a
40012	方向 (0 正向，1 反向)	R/W	0x000b
40013	单次运行到位反馈 **0 表示没到位　　1 表示运行到位**	R 只针对单次运行命令	0x000c
40014	备用		0x000d
40015	正限位信号(有效值 0--4)	R/W	0x000e
40016	反限位信号(有效值 0--4)	R/W	0x000f
40017	控制器 ID 号（485 设备号）	R/W	0x0010
40018	工程启动信号(有效值 0--4)	R/W	0x0011
40019	工程停止信号(有效值 0--4)	R/W	0x0012
40020 40021	系统工作次数计数	R/W（低 16 位在前）	0x0013、0x0014
40022	备用		0x0015
40023　40024	当前坐标显示 （低 16 位在前）	R,读这个寄存器会实时显示当前坐标	0x0016、0x0017
40025	工程号（只能为 1）	R/W	0x0018
40026	工程总步数(有效值 1--33)	R/W	0x0019
40027	本步启动口信号(有效值 0--4)	R/W	0x001a
40028	工程本步启动频率，单位 HZ	R/W	0x001b
40029	工程本步加减频率，单位 HZ	R/W	0x001c

40030	工程本步运行方向 (0 正向，1 反向)	R/W	0x001d
40031	工程本步运行速度 (单位：转/每分钟)	R/W	0x001e
40032　40033	工程本步运行距离	R/W（低 16 位在前）	0x001f、0x0020
40034	备用	R/W	0x0021
40035　40036	本步延时时间(单位：毫秒)	R/W（低 16 位在前）	0x0022、0x0023
40037	段循环起始步	R/W	0x0024
40038	段循环结束步	R/W	0x0025
40039	段循环次数	R/W	0x0026
40040	设定工程当前步号(有效值 1--33)	R/W	0x0027
40041	备用		0x0028
40042	工程实时步号显示	R（有效值 0—33）	0x0029
40043	第 1 路输入信号状态显示	R (1--ON　0--OFF)	0x002a
40044	第 2 路输入信号状态显示	R(1--ON　0--OFF)	0x002b
40045	第 3 路输入信号状态显示	R(1--ON　0--OFF)	0x002c
40046	第 4 路输入信号状态显示	R(1--ON　0--OFF)	0x002d
40047	备用	R/W	0x002e
40048	备用	R/W	0x002f
40049	电机运行状态显示	R(1--ON　0--OFF)	0x0030
40050	工程循环次数	R/W	0x0031
40051	本步停止口信号(有效值 0--4)	R/W	0x0032
40052	备用	R/W	0x0033
40053	备用	R/W	0x0034
40054	备用	R/W	0x0035
40055	备用	R/W	0x0036
40056	备用	R/W	0x0037
40057	备用	R/W	0x0038
40058	往返运行的次数	R/W	0x0039
40074	**波特率低 16 位**	R/W	0x0049
40075	**波特率高 16 位**	R/W	0x004a
40076	相对/绝对运行 **(0 相对　1 绝对)**	R/W 只针对单次运行命令	0x004b
40077	运行方式选择 0 位置　1 速度触发　2 速度点动	R/W (主要是针对正反启动信号控制的)	0x004c
40078	正转启动信号　(0 表示无设置) (1—4 对应 IN1—IN4 输入)	R/W 信号有效时，按上面选择的运行方式正转	0x004d
40079	反转启动信号　(0 表示无设置) (1—4 对应 IN1—IN4 输入)	R/W 信号有效时，按上面选择的运行方式反转	0x004e
40080	电机使能控制	R/W　0 使能　1 不使能	0x004f

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
00001	数据保存	断电保存所有参数	0x0000
00002	工程参数读取		0x0001
00003	工程参数清零		0x0002
00004	(工程)停止/急停		0x0003
00005	正转点动	置 ON，电机一直正转 置 OFF，电机停止	0x0004
00006	反转点动	置 ON，电机一直反转 置 OFF，电机停止	0x0005
00007	回数据零	电机运行到坐标零点	0x0006
00008	**单次运行(按 40009 的速度和 40010 40011 的距离运行 1 次)**	**可选择相对/绝对运行 2 种方式运行**	0x0007
00009	工程启动	按设定好的每一个步骤运行。直到所有步骤完毕或急停。	0x0008
00010	回机械零	电机一直反转，碰到机械零点信号停止。	0x0009
00011	坐标清零	将 40023 40024 的值设为 0	0x000a
00012	备用		0x000b
00013	备用		0x000c
00014	备用		0x000d
00015	备用		0x000e
00016	备用		0x000f
00017	备用		0x0010
00018	工程上一步	将 40040 的值减 1	0x0011
00019	工程下一步	将 40040 的值加 1	0x0012
00037	基本参数初始化		0x0024

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 00008)**

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 07	ff 00	3D FB

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 40010 对应低 16 位数据，40011 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 09	00 02	04	00 C8 00 00	B2 3B

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 09	00 02	91 CA

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 2a 00 01 A5 C2　　00 2a 是输入 1 状态地址

读电机当前坐标：01 03 00 16 00 02 25 CF　　00 16 是电机坐标地址

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 03 置 ON)：01 05 00 03 ff 00 7C 3A

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 04 置 ON)：01 05 00 04 ff 00 CD FB

(线圈地址 00 04 置 OFF)：01 05 00 04 00 00 8C 0B

电机反转命令(线圈地址 00 05 置 ON)：01 05 00 05 ff 00 9C 3B

(线圈地址 00 05 置 OFF)：01 05 00 05 00 00 DD CB

设置速度命令：01 06 00 08 00 32 89 DD　00 08 是地址，00 32 表示速度为 50

设置方向为正向：01 06 00 0b 00 00 F8 08　　00 0b 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 00 ff 00 8C 3A

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;
}
```