

# 目录

第一章 通用说明	3
1.1 简介	3
1.2 产品列表	3
1.3 正常的工作条件	4
第二章 CPU 模块介绍	5
2.1 概述	5
2.1.1 部件结构图	5
2.1.2 CPU 类型	5
2.2 各部分功能介绍	10
2.2.1 CPU 状态及指示灯	10
2.2.2 USB 编程口	12
2.2.3 串行通信口	12
2.2.4 以太网通信口	13
2.2.5 DIO 通道说明	14
2.2.6 高速脉冲计数和高速脉冲输出	14
2.2.7 边沿中断	15
2.2.8 数据保持和数据备份	15
2.2.9 实时时钟 (RTC)	16
2.2.10 后备电池	16
2.3 接线图	16
2.4 尺寸图	21
2.5 技术数据	21
第三章 软件功能及使用	25
3.1 概述	25
3.2 安装 USB 编程口的驱动程序	25

---

3.3 使用 Modbus TCP 协议与第三方设备通讯-----	31
3.3.1 Modbus 寄存器编号-----	31
3.4 高速计数器的使用-----	32
3.4.1 高速计数器工作模式和输入信号-----	32
3.4.2 控制寄存器和状态寄存器-----	33
3.4.3 预置值（PV 值）设定-----	35
3.4.4 “CV=PV” 中断编号-----	37
3.4.5 高速计数器的使用方法-----	38
3.5 高速脉冲输出功能的使用-----	40
3.5.1 高速脉冲输出指令-----	40
3.5.2 PLS 指令的使用-----	41
3.5.3 定位控制指令的使用-----	47

# 第一章 通用说明

## 1.1 简介

Kinco-K2 系列 PLC 属于小型一体化 PLC，是步科公司推出的经济单机型产品。

在保证功能丰富、高性能、高可靠性的前提下，K2 优化硬件设计以降低成本，更提供了 USB 编程口、Ethernet 接口、晶体管型 DIO 点（DI、DO 复用）、本体集成模拟量等更贴近用户需求的功能，是具有很高性价比的小型 PLC 产品。

## 1.2 产品列表

名称	订货号	功能描述
<b>CPU 模块</b>		
CPU205	K205-16DT	DC24V 供电，DI 6*DC24V，DIO 4*DC24V，DO 6*DC24V USB2.0 编程口，2*RS485，不可带扩展模块。
	K205-16DR	DC24V 供电，DI 6*DC24V，DIO 4*DC24V，DO 6*继电器 USB2.0 编程口，2*RS485，不可带扩展模块。
	K205EA-18DT	DC24V 供电，DI 8*DC24V，DO 8*DC24V,1*AI,1*A0 USB2.0 编程口，2*RS485，不可带扩展模块。
	K205EX-22DT	DC24V 供电，DI 8*DC24V，DIO 6*DC24V，DO 8*DC24V USB2.0 编程口，2*RS485，不可带扩展模块。
CPU204	K204ET-16DT	DC24V 供电，DI 8*DC24V，DO 6*DC24V,1*AI,1*A0 USB2.0 编程口，1*Ethernet，2*RS485，不可带扩展模块。
CPU209	K209M-56DT	DC24V 供电，DI 32*DC24V，DO 24*DC24V USB2.0 编程口，2*RS485，2*CAN，可带 KS 扩展模块
	K209EA-50DX	DC24V 供电，DI 22*DC24V，DO 8*DC24V+12*继电器,6*AI,2*A0 USB2.0 编程口，1*RS232,2*RS485，不可带扩展模块。

### 1.3 正常的工作条件

Kinco-K2 的设计符合 GB/T 15969.3-2007 (idt IEC61131-2: 2007) 标准和测试规范。

下表描述了 Kinco-K2 正常的工作条件。使用时用户必须保证不超出表中规定的 PLC 工作条件。

运输和存储		
气候条件	环境温度	-40℃ ~ +70℃ 。
	相对湿度	10% ~ 95%，无凝露。
	大气压	相当于 0 ~ 3000 米海拔高度。
机械条件	自由跌落	带运输包装，允许从 1 米高度 5 次跌落于水泥地面。
工作条件		
气候条件	环境温度	自然通风的开放式装置，环境气温 -10 ~ 55℃ 。
	相对湿度	10% ~ 95%，无凝露。
	大气压	海拔高度不超过 2000 米
	污染等级	适用于污染等级 2。
机械条件	正弦振动	5 < f < 8.4Hz ， 随机: 3.5mm 位移；连续: 1.75mm 位移。 8.4 < f < 150，随机: 1.0g 加速度；连续: 0.5g 加速度。
	冲击	半正弦波、15g、11ms，每轴向 6 次。
电磁兼容性 (EMC)	静电放电	空气放电 8KV，接触放电 4KV。性能等级 B。
	浪涌	交流电源 2KV CM，1KV DM；直流电源 0.5KV CM，0.5KV DM； IO 和通信口：1KVCM。 性能等级 B。
	快速瞬变脉冲群	电源耦合 2KV，5KHz；IO 及通信耦合 1KV，5KHz。 性能等级 B。
	电压跌落	交流系统，50Hz 时，电压 0%持续 1 周波，40%持续 10 周波， 75%持续 20 周波。 性能等级 A。
防护等级	防水防尘	IP20。

## 第二章 CPU 模块介绍

### 2.1 概述

#### 2.1.1 部件结构图



#### 2.1.2 CPU 类型

Kinco-K2 提供了多种规格的 CPU，供电电源均采用 DC 24V 的供电电压。

下表描述了各种类型 CPU 的主要技术参数。

参数	CPU205	CPU205EX	CPU205EA
<b>供电电源</b>			
额定供电电源	DC24V。备注：USB 口也可以直接供电供 CPU 运行。		
供电电压范围	DC20.4V—28.8V		
<b>I/O 及通信口</b>			
本体开关量	6*DI/6*DO/4*DIO	8*DI/8*DO/6*DIO	8*DI/8*DO
本体模拟量	--		1*AI/1*AO
扩展模块	不支持。		
编程口	USB2.0，采用 micro USB 接口形式。		
串行通信口	2 个 RS485 接口，分别为 PORT1、PORT2，通信速率最高 115.2kbps。 PORT1 支持编程协议、Modbus RTU 从站、自由通信； PORT2 支持 Modbus RTU 主站和从站、自由通信。		
高速计数器	4		
单相	HSC0、HSC1 最高计数频率：50KHz HSC2、HSC3 最高计数频率：20KHz		
双相	HSC0、HSC1 最高计数频率：50KHz HSC2、HSC3 最高计数频率：10KHz		
高速输出	3 通道 0 和 1 最高输出频率：50KHz（最高频率时要求负载电阻不大于 3K $\Omega$ ）。 通道 2 最高输出频率：10KHz		
边沿中断	4 路，I0.0-I0.3 可分别设置为上升沿或者下降沿中断。		
<b>存储区域</b>			
用户程序	最大 4K 条指令		
用户数据	M 区：1K 字节；V 区：4K 字节。		
数据备份	E <sup>2</sup> PROM，448 字节		
数据保持	4K 字节。锂电池，常温下 3 年		
<b>其它</b>			

定时器	共 256 个，其中 1ms 时基：4 10ms 时基：16 100ms 时基：236
定时中断	2 个，0.1ms 时基。
计数器	256 个
实时时钟	有，在 25℃ 时误差小于 5 分钟/月

参数	CPU209EA
<b>供电电源</b>	
额定供电电源	DC24V。备注：USB 口也可以直接供电供 CPU 运行。
供电电压范围	DC20.4V—28.8V
<b>I/O 及通信口</b>	
本体开关量	22*DI/20*DO
本体模拟量	6*AI/2*AO
扩展模块	不支持。
编程口	USB2.0，采用 micro USB 接口形式。
串行通信口	1 个 RS232 接口，命名为 PORT0，通信速率最高 115.2kbps。 2 个 RS485 接口，分别为 PORT1、PORT2，通信速率最高 115.2kbps。 PORT0 和 PORT1 支持编程协议、Modbus RTU 从站、自由通信； PORT2 支持 Modbus RTU 主站和从站、自由通信。
高速计数器	4
单相	HSC0、HSC1 最高计数频率：200KHz HSC2、HSC3 最高计数频率：20KHz
双相	HSC0、HSC1 最高计数频率：200KHz HSC2、HSC3 最高计数频率：10KHz
高速输出	3 通道 0 和 1 最高输出频率：200KHz（最高频率时要求负载电阻不大于 3KΩ）。 通道 2 最高输出频率：10KHz
边沿中断	4 路，I0.0-I0.3 可分别设置为上升沿或者下降沿中断。

存储区域	
用户程序	最大 4K 条指令
用户数据	M 区 1K 字节；V 区 4K 字节。
数据备份	E <sup>2</sup> PROM，448 字节
数据保持	4K 字节。锂电池，常温下 3 年
其它	
定时器	256 1ms 时基：4 10ms 时基：16 100ms 时基：236
定时中断	2 个，0.1ms 时基。
计数器	256 个
实时时钟	有，在 25℃ 时误差小于 5 分钟/月

参数	K204ET-16DT
供电电源	
额定供电电源	DC24V。备注：USB 口也可以直接供电供 CPU 运行。
供电电压范围	DC20.4V—28.8V
I/O 及通信口	
本体开关量	8*DI, 6*DO 晶体管
本体模拟量	1*AI, 1*AO
扩展模块	不支持
编程口	USB2.0, 采用 micro USB 接口形式。
以太网口	1 个，支持编程协议、Modbus TCP Server（即从站）
串行通信口	2 个 RS485 接口，分别为 PORT1、PORT2，通信速率最高 115.2kbps，支持编程协议、Modbus RTU 主站、Modbus RTU 从站、自由通信。
高速输入	4
单相	最高计数频率：200KHz
双相	最高计数频率：200KHz
高速输出	3，最高输出频率 200KHz（要求负载电阻不大于 3KΩ）。



边沿中断	4 路, I0.0-I0.3 可分别设置为上升沿或者下降沿中断。
<b>存储区域</b>	
用户程序	最大 4K 条指令
用户数据	M 区 1K 字节; V 区 4K 字节。
数据备份	E2PROM, 448 字节
数据保持	V 区 2K 字节 (VB0-VB2047)。锂电池, 常温下 3 年
<b>其它</b>	
定时器	256 1ms 时基: 4 10ms 时基: 16 100ms 时基: 236
定时中断	2 个, 0.1ms 时基。
计数器	256 个
实时时钟	有, 在 25°C 时误差小于 5 分钟/月

参数	K209M-56DT
<b>供电电源</b>	
额定供电电源	DC24V。备注: USB 口也可以直接供电供 CPU 运行。
供电电压范围	DC20.4V—28.8V
<b>I/O 及通信口</b>	
本体开关量	DI 32*DC24V, DO 24*DC24V
最大扩展模块	14 (KS 系列薄片式扩展模块, 注意外形有所区别)
编程口	USB2.0, 采用 micro USB 接口形式。
CAN 接口	2 个 CAN 接口 CAN1 支持扩展协议、CAN 自由通信功能 CAN2 支持 CANOpen 主站、Kinco 运动控制指令、MIoT 协议和 CAN 自由通信功能。

串行通信口	提供了 2 个 RS485 串行通信口，分别命名为 PORT1、PORT2，通信波特率最高为 115.2Kbps。 PORT1 口支持编程协议、Modbus RTU 主站、从站协议、Kinco PLC 互联协议和自由通信。 PORT2 口支持编程协议、Modbus RTU 主站、从站协议和自由通信。
高速输入 单相 双相	2 最高计数频率：200KHz 最高计数频率：200KHz
高速输出	4 3 路最高输出频率 200KHz，1 路最高输出频率 10KHz，（要求负载电阻不大于 3K $\Omega$ ）
<b>存储区域</b>	
用户程序	最大 8K 指令
用户数据	M 区 4K 字节；V 区 16K 字节
数据备份	E2PROM，V 区最后的 1K 字节，永久存储。
数据保持	全部 V 区。锂电池，常温下 3 年
<b>其它</b>	
定时器	256 个
定时中断	2 个，0.1ms 时基。
计数器	256 个
实时时钟	有，在 25℃时误差小于 5 分钟/月

## 2.2 各部分功能介绍

### 2.2.1 CPU 状态及指示灯

CPU 模块有两种状态：运行状态和停止状态。

在运行状态下，CPU 模块正常地循环执行主扫描任务和各种中断任务。在停止状态下，CPU 模块仅处理部分通信请求(包括来自于 KincoBuilder 编程软件的编程、调试等命令，以及作为 Modbus RTU 从站响应主站的通信命令)，同时将所有的输出点（DO、AO）立即输出为用户工程中【硬件配置】中定义的“停机输出”值。

### ➤ 改变 CPU 状态

正常情况下，用户可以通过如下方法来改变 CPU 状态：使用运行/停止开关；在 KincoBuilder 软件中执行【调试】->【启动...】或者【停止...】菜单命令。

这两种方法的组合操作结果如下表：

运行/停止开关位置	KincoBuilder 菜单命令	PLC 实际状态
RUN	启动	运行状态
	停止	停止状态
STOP	启动	停止状态
	停止	停止状态

但是在发生以下情形时，无论运行/停止开关在什么位置，PLC 都将进入停止状态：

- 1) PLC 运行出错（严重错误、致命错误）导致 PLC 停止；
- 2) 用户使用了 STOP 指令将 PLC 置 STOP；
- 3) 下载程序中途失败，PLC 会维持之前的 STOP 状态。

此时，用户可以将运行/停止开关拨到 RUN 位置，手动让 PLC 进入运行状态，但是建议用户首先排除引起 PLC 停止的原因，然后再让它运行。

### ➤ CPU 指示灯

Run、Stop、Comm、Err 这 4 个指示灯用于指示 CPU 当前的工作情况。

**【Run】**：若 CPU 正处于运行状态，则 RUN 灯点亮。

**【Stop】**：若 CPU 正处于停止状态，则 STOP 灯点亮。

**【Comm】**：任意一个通信口（PORT1/2）接收、发送时，Comm 灯都会闪烁。

**【Err.】**：若 CPU 检测到用户程序或者模块本身发生错误，则点亮 Err 灯。K2 将错误分为三个等级：致命错误、严重错误、一般错误。当 CPU 检测到错误时，会根据错误的等级采取不同的处理措施，同时点亮 Err 灯，并将具体的错误码根据发生的先后次序依次存储下来，以供用户读取并进行分析。

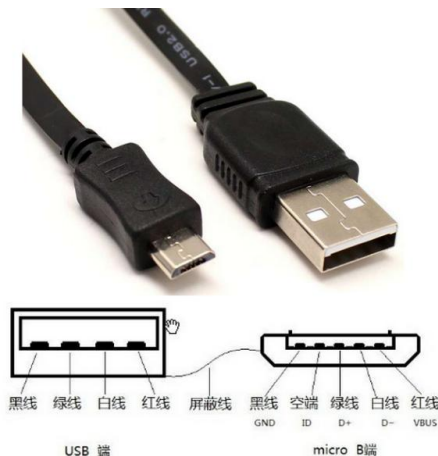
CPU209EA 将 PORT1 和 PORT2 的通信指示灯区分开了，提供了如下两个指示灯：

**【Port1】**：通信口 PORT1 的接收、发送，Comm 灯都会闪烁。

【Port2】:通信口 PORT2 的接收、发送，Comm 灯都会闪烁。

### 2.2.2 USB 编程口

K2 提供了一个 USB2.0 接口，这个接口只能作为编程口。它采用了 Micro USB 的接口形式，这种接口形式在智能手机上应用非常广泛，用户可以直接使用同样接口的手机数据线作为 K2 的编程电缆。但需要注意的是手机线，有些线缆只能用于充电，不能传输数据！



在电脑上，K2 编程口作为一个虚拟串口，首次使用时必须先安装驱动程序。当安装完成最新版本的 Kincobuilder 编程软件后，在安装目录下面的 \Drivers 目录下面存放着 K2 编程口针对各个版本 Windows 系统的驱动程序，目前支持 Windows XP、Windows 7 和 Windows 8 系统。当第一次使用编程数据线连接 K2 和电脑时，Windows 系统会自动检测到新硬件并提示安装驱动程序，此时用户根据自己的 Windows 版本选择相应目录下的驱动程序即可。

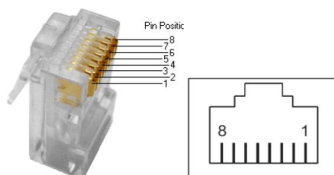
### 2.2.3 串行通信口

K2 系列 CPU 模块都提供了 2 个或者更多串行通信口，其中 RS232 口命名为 PORT0，RS485 接口分别命名为 PORT1、PORT2。串行通讯口的波特率最高为支持 115.2kbps。PORT0、PORT1 口既可以用作编程口，也支持 Modbus RTU 从站协议和自由通信。PORT2 口支持 Modbus RTU 主、从协议和自由通信。支持各种 CPU 模块的通信口数量及位置如下表：

模块类型	串行通讯口	备注
CPU205	2*RS485	PORT1、PORT2 均位于左上部接线端子上。
CPU205EX	2*RS485	PORT1、PORT2 均位于左下部 RJ45 接口中。

CPU205EA	2*RS485	PORT1 位于左上部接线端子上，PORT2 位于 RJ45 接口中。
CPU204ET	2*RS485	PORT1 位于左上部接线端子上，PORT2 位于左下部接线端子上。
CPU209M	2*RS485	PORT1、PORT2 均位于左上部接线端子上。
CPU209EA	2*RS485 1*RS232	PORT1/PORT2 均位于左上部接线端子上，PORT0 位于 RJ45 接口中。

RJ45 接口的管脚描述如下。



各通信口的接线图请参见章节的 [2.3 接线图](#)。

## 2.2.4 以太网通信口

部分 K2 产品提供了符合标准 IEEE802.3 规范的 Ethernet 接口。该接口支持编程协议，可以作为编程口使用，另外也支持 Modbus TCP Sever，即通常说的“从站”功能。

通信电缆采用直通电缆（直连线）或者交叉电缆（交叉线）均可。K2 的以太网接口提供了“自动协商”功能，当插入电缆后，K2 会跟通信对方自动进行协商以确定所用电缆类型。

用户可以通过 USB 口、串口或者以太网口自身来修改以太网接口的参数：在 KincoBuilder 软件中执行【工具】--【TCP/IP 参数配置】菜单命令，将弹出如下对话框，用户可以读取或者修改参数。



在局域网内，位于同一网段（即 IP 地址的前三段数字要相同，最后一段不同）的 PLC、PC 之间才能相互通信。

## 2.2.5 DIO 通道说明

部分晶体管型 DO 规格的 K2 产品提供了一定数量的 DIO 点。每个 DIO 通道均既可以用作 DI（源型），也可以用作 DO（源型），其信号形式为 DC24V。至于一个通道用作 DI 或者 DO，用户不需要作任何配置，只需根据实际需要接线使用，同时在程序中操作相应的映像区地址即可。相比于固定 IO 类型的 PLC，具有相同 IO 点数的 K2 模块显然能够适应更多的应用。

每个 DIO 通道在 CPU 的 I 区和 Q 区中均占有 1 位地址，即每个通道均有两个地址：DI 地址和 DO 地址。**在用户程序中需要注意：若某通道被实际用作 DI，则避免操作该通道的 DO 地址；同样地，若某通道被实际用作 DO，则要避免操作该通道的 DI 地址。**

例如，I/Q 0.6 是一个 DIO 通道，它占用了两个地址 I0.6 和 Q0.6。若用户需要将它用做 DI，那么直接将输入信号接入到该通道，同时在程序中使用 I0.6 即可，注意这种情况下避免操作 Q0.6。若用户需要将它用作 DO，那么将该通道接到相应的执行设备，同时在程序中对 Q0.6 赋值即可，注意这种情况下也避免读取 I0.6。

## 2.2.6 高速脉冲计数和高速脉冲输出

K2 提供了 4 路高速计数器，编号为 HSC0 至 HSC3。K209M 提供 2 路高速计数器，编号为 HSC0、HSC1。高速计数器支持多种模式，可以进行单相、双相（Up/Down）、AB 相（1 倍频和 4 倍频）等计数。各系列 CPU 最高计数频率如下

CPU 型号	单相		AB 相	
	HSC0 和 HSC1	HSC2 和 HSC3	HSC0 和 HSC1	HSC2 和 HSC3
K204ET	200K	200K	200K	200K
K205	50K	20K	50K	10K
K209EA	200K	20K	100K	10K
K209M	200K	/	200K	/

另外，K2 也提供了 3 路高速输出，所用通道分别为 Q0.0、Q0.1 和 Q0.4，都支持 PTO（脉冲串）和 PWM（脉宽调制）方式输出。各系列 CPU 最高输出频率如下（注意：Q0.0 和 Q0.1 最高输出频率时要求负载电阻不大于 3K $\Omega$ ）。

CPU 型号	Q0.0	Q0.1	Q0.4
K204ET	200K	200K	200K
K205	50K	50K	10K
K209EA	200K	200K	10K

K209M	200K	200K	200K
-------	------	------	------

注意：继电器输出型号的 CPU 模块（订货号最后位是“R”，比如 K205-16DR）不支持高速脉冲输出功能！

### 2.2.7 边沿中断

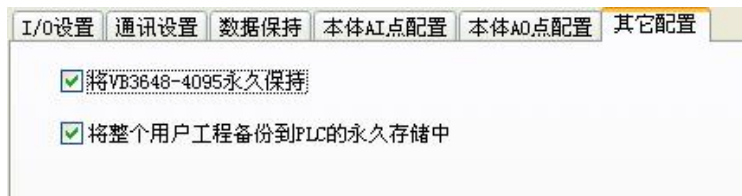
CPU 本体的输入点 I0.0—I0.3 支持边沿中断功能，可以利用输入信号的上升沿和下降沿产生中断。利用这一功能能够快速捕捉到输入的上升沿或下降沿，对一些脉冲宽度小于 CPU 扫描周期的输入信号实现快速响应。

### 2.2.8 数据保持和数据备份

数据保持是指在 CPU 模块 RAM 中的数据在断电后保持为断电瞬间的状态，并供 CPU 在下次上电的时候使用。CPU 模块内部均提供一个后备锂电池（不可充电，可更换）用于数据保持功能。在断电时，后备电池为 RAM 供电并保持 RAM 中的数据。用户需要使用 KincoBuilder 软件在用户工程的【PLC 硬件配置】中选择需保持的数据区类型（如 V 区、C 区等）及起止范围。常温下，电池典型寿命为 5 年，断电保持的时间累计不小于 3 年。

数据备份是指 CPU 模块在永久存储器中开辟一个区域，用于存放用户数据，该区域内的数据断电永久不会丢失，并供 CPU 在下次上电的时候使用。**CPU 模块提供了 E2PROM 存储器用于数据备份功能，由于 E2PROM 只有 100 万次的写入寿命，因此用户注意尽量避免永久备份那些变化频繁的数据！**

V 区中的最后 448 个字节（即 VB3648--VB4095）是数据备份区域，该区域中的数据会自动备份到永久存储器中。K2 默认的永久保持区域与 K5 保持一致，即为 VB3648--VB3902。若要使 VB3903--VB4095 也成为永久保持区域，则需要在【PLC 硬件配置】中进行配置，如不进行配置，则这个区域的数据不会自动备份。配置界面如下图：



## 2.2.9 实时时钟 (RTC)

CPU本体内都集成了实时时钟(RTC)，可提供实时的时间/日历表示。在第一次使用RTC时，用户需要通过在KincoBuilder中执行【PLC】->【调整CPU时钟...】菜单命令来设置时钟，之后就可以使用读写实时时钟的指令（READ\_RTC、SET\_RTC、RTC\_W、RTC\_R），实现与相关的控制应用。

CPU断电后，实时时钟依靠后备电池的供电来维持运行，常温下，电池典型寿命为5年,断电保持的时间累计不小于3年。

## 2.2.10 后备电池

K2允许使用特定规格的锂电池作为后备电池。当断电时，后备电池用于给实时时钟供电来维持时钟的运行，同时也给RAM供电来进行数据保持。

后备电池可以拆卸，在模块侧面上部是电池盒的位置，电池就安装在电池盒中。当后备电池耗完电后，用户可以打开电池盒自行更换一个新的电池。

电池为CR2032带连接器的3V锂电池，形状如下图，用户可以单独订购电池。（订货型号：MALAK CR2032-加线）。



## 2.3 接线图

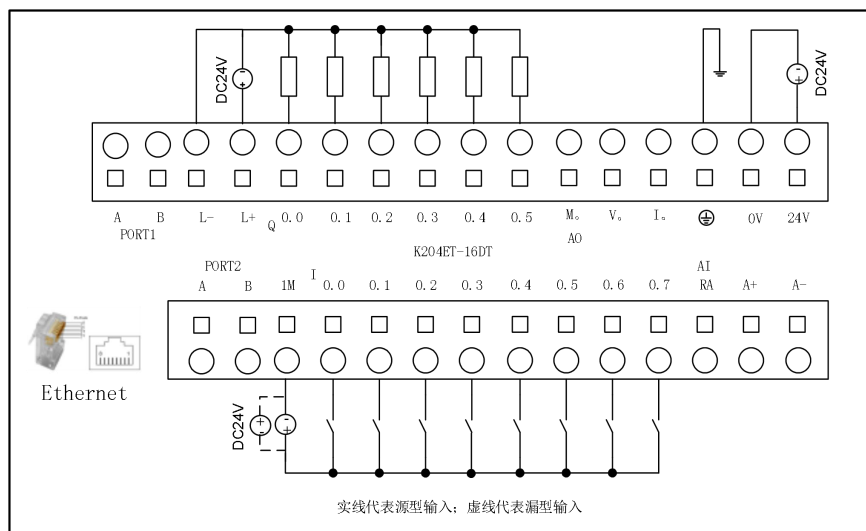


图 2-1 K204ET-16DT 接线图



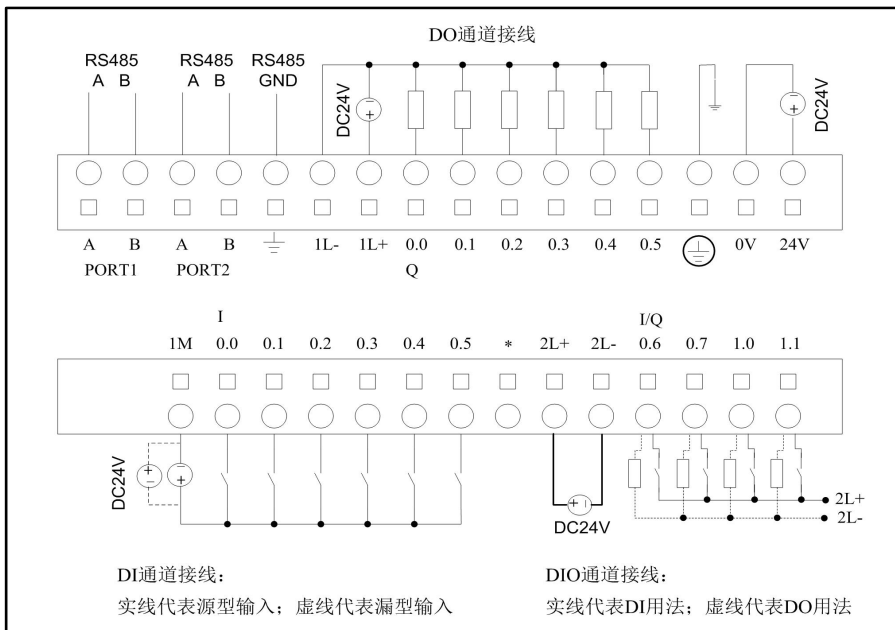


图 2-2 K205-16DT 接线图

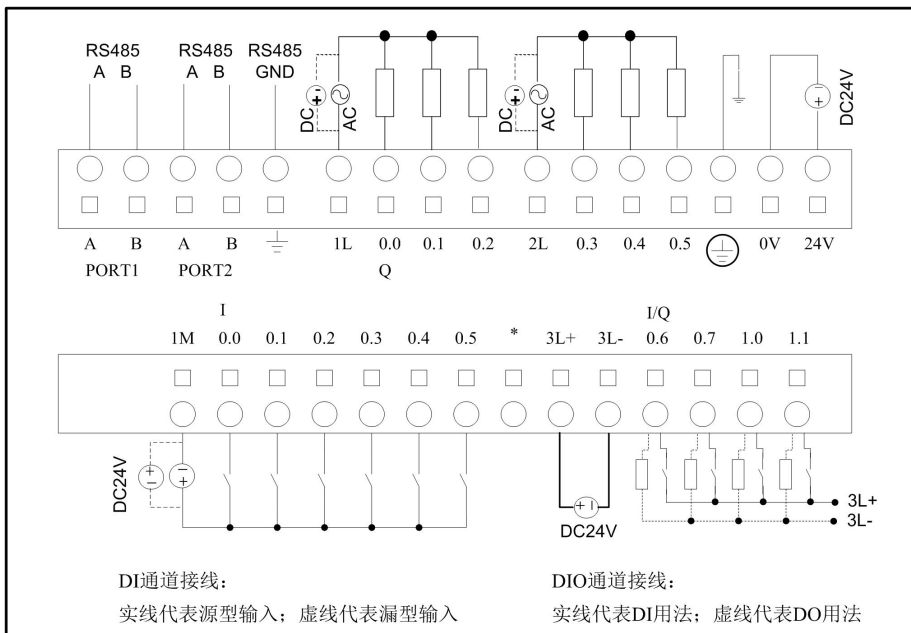


图 2-3 K205-16DR 接线图

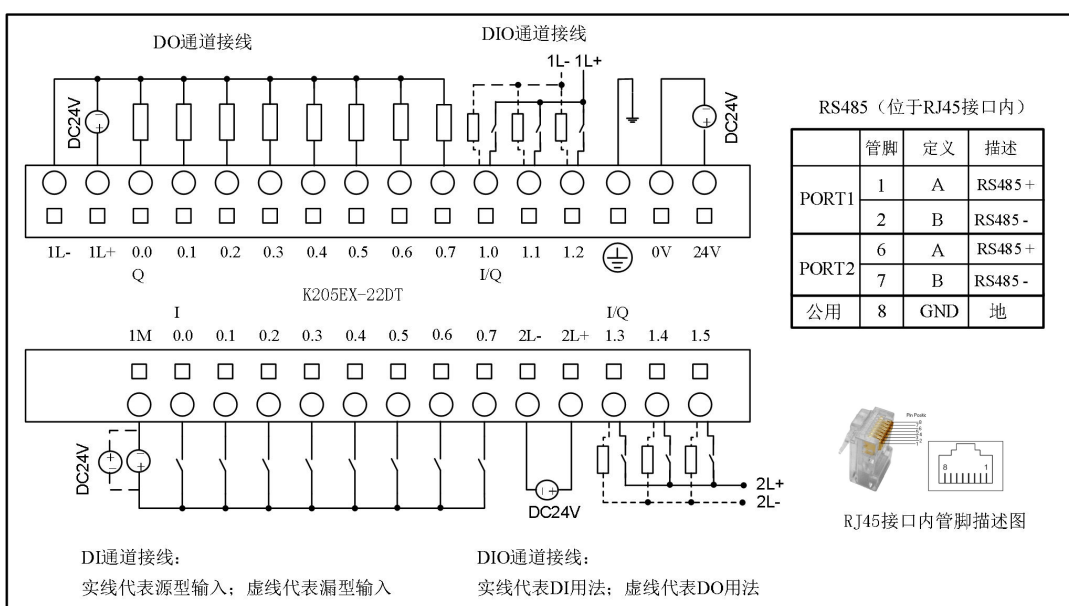


图 2-4 K205EX-22DT 接线图

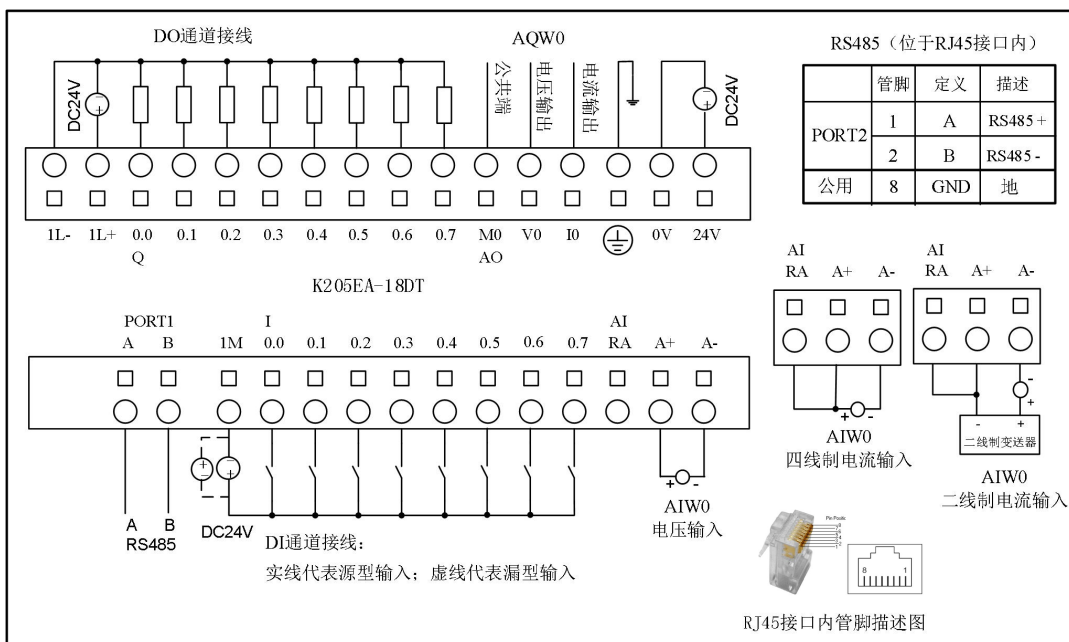


图 2-5 K205EA-18DT 接线图

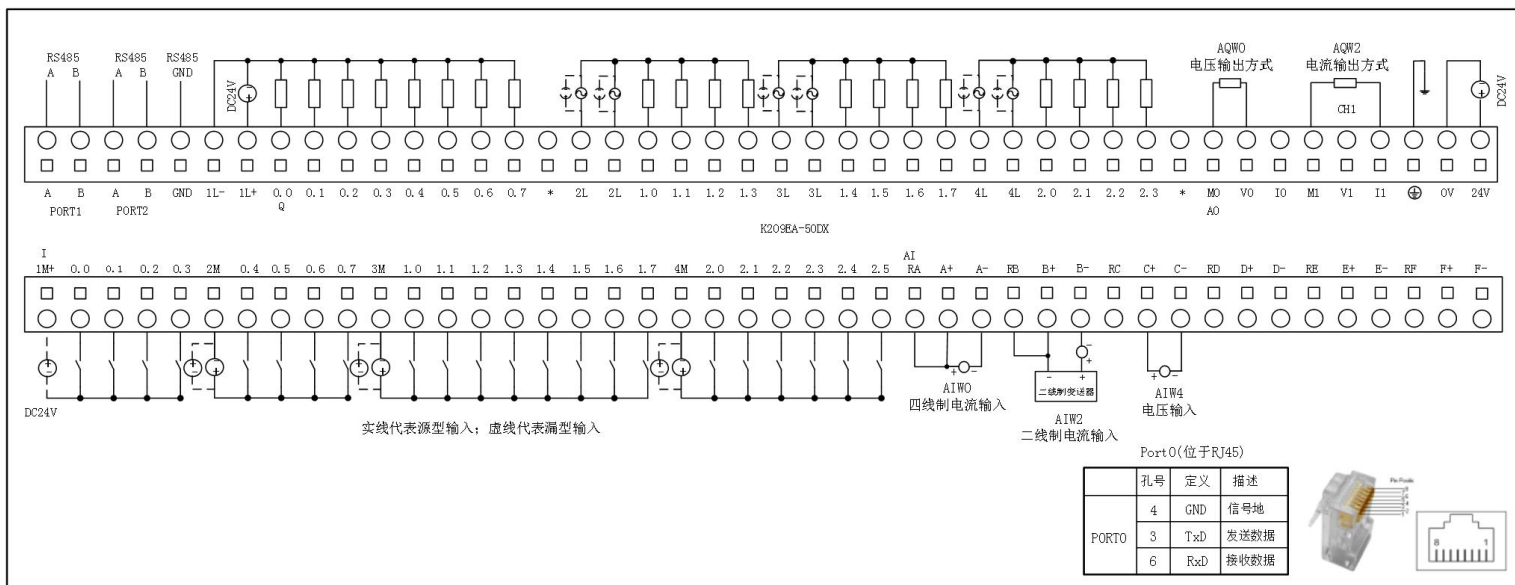


图 2-6 K209EA-50DX 接线图

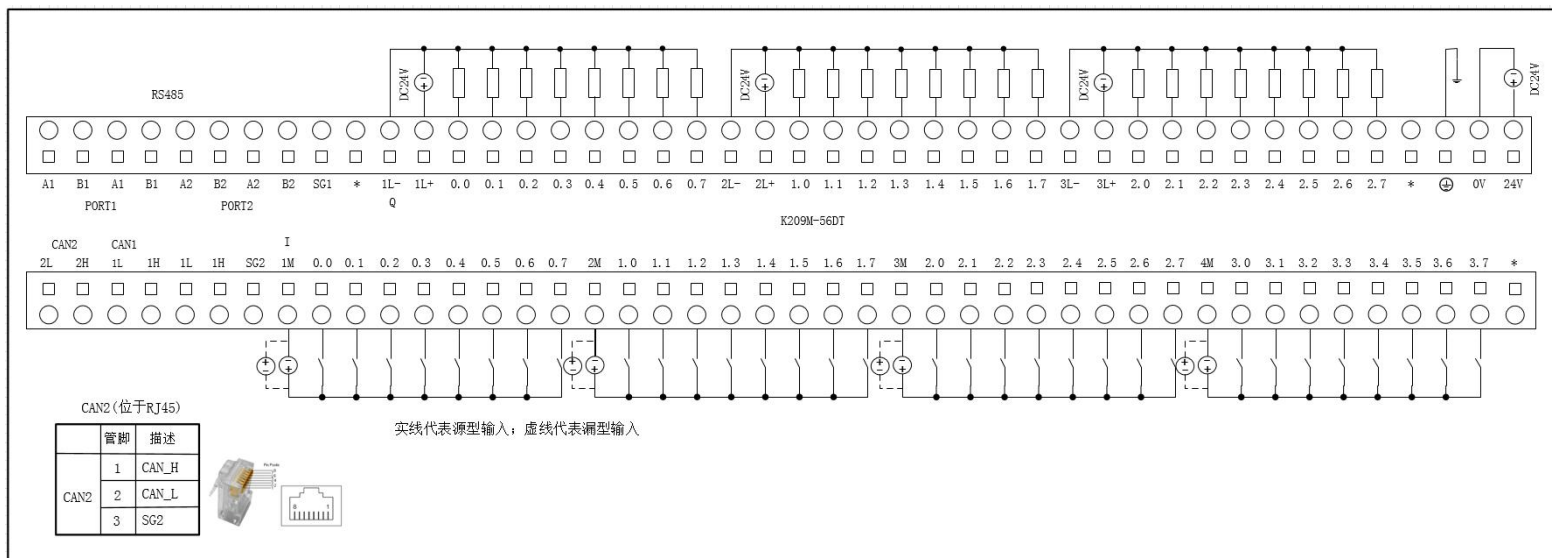
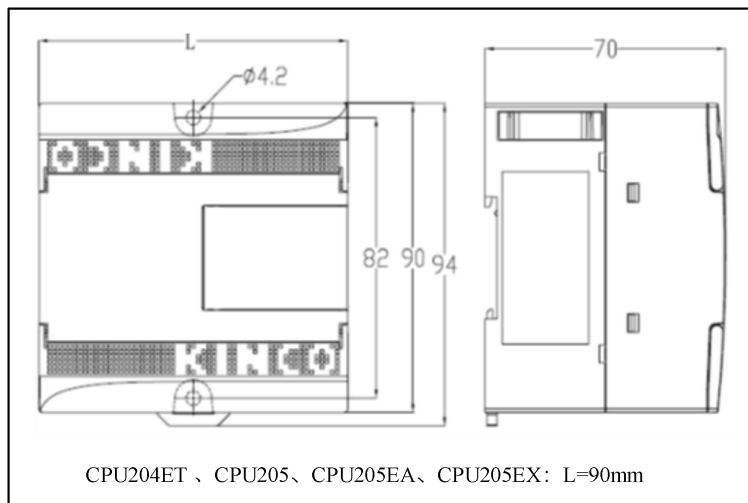
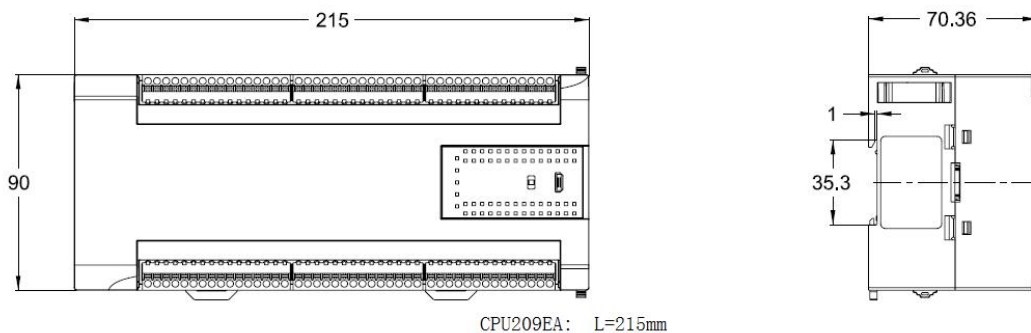


图 2-7 K209M-56DT 接线图

## 2.4 尺寸图



K205/204ET 外形尺寸图



K209 外形尺寸图

## 2.5 技术数据

## ► 本体 DI 通道参数

输入类型	源型/漏型
额定输入电压	DC24V, 最高允许 DC30V。
额定输入电流	3.5mA@24VDC

逻辑“0”最大输入电压	5V@0.7mA
逻辑“1”最小输入电压	11V@2.0mA
输入延迟时间 • 接通延时 • 断开延时	普通通道 15 $\mu$ S, 高速通道 10 $\mu$ S(50K),1.2 $\mu$ S(200K)  普通通道 60 $\mu$ S, 高速通道 6 $\mu$ S(50K),0.5 $\mu$ S(200K)
输入与内部逻辑电路的隔离 • 隔离方式 • 隔离电压	光电耦合器 500VAC/1 分钟
状态指示	各通道绿色 LED 输入信号“1”

➤ 本体 DO 通道参数（晶体管型）

输出类型	源型
额定输出电压	DC24V。允许范围: DC20.4V—DC28.8V（与供电电压一致）。
每通道输出电流	额定 200mA @24VDC
每通道瞬时冲击电流	1A,不超过 1S
输出漏电流	最大 0.5 $\mu$ A
输出阻抗	最大 0.2 $\Omega$
输出延迟时间 • 接通延时 • 断开延时	普通通道 15 $\mu$ S, 高速通道 10 $\mu$ S(50K),0.5 $\mu$ S(200K) 普通通道 35 $\mu$ S, 高速通道 6 $\mu$ S(50K),1 $\mu$ S(200K)
保护功能: • 电源接入极性保护 • 感性负载输出保护 • 短路保护 • 输出极性反向保护	无 有 有 有, 允许在输出端施加反极性信号不超过 10S。
输出与内部逻辑电路的隔离 • 隔离方式 • 隔离电压	光电耦合器 500VAC/1 分钟
状态指示	各通道绿色 LED 指示输出“1”

➤ 本体 DIO 通道参数（晶体管型）

输入/输出类型	源型
---------	----

额定输入电压	DC24V，最高允许 DC30V。
额定输入电流	3.5mA@24VDC
逻辑“0”最大输入电压	5V@0.7mA
逻辑“1”最小输入电压	11V@2.0mA
额定输出电压	DC24V。允许范围: DC20.4V—DC28.8V（与供电电压一致）。
每通道输出电流	额定 200mA @24VDC
每通道输出瞬时冲击电流	1A,不超过 1S
输出漏电流	最大 0.5 $\mu$ A
输出阻抗	最大 0.2 $\Omega$
输入延迟时间	
• 接通延时	15 $\mu$ S
• 断开延时	60 $\mu$ S
输出延迟时间	
• 接通延时	15 $\mu$ S
• 断开延时	35 $\mu$ S
保护功能:	
• 电源接入极性保护	无
• 感性负载输出保护	有
• 短路保护	有
• 输出极性反向保护	有，允许在输出端施加反极性信号不超过 10S。
输出与内部逻辑电路的隔离	
• 隔离方式	光电耦合器
• 隔离电压	500VAC/1 分钟
状态指示	各通道绿色 LED 指示输出“1”

➤ 本体 DO 通道参数（继电器型）

输出类型	继电器
允许负载电压	DC 30V/AC250V
允许负载电流	1A（DC 30V/AC250V）
每组输出电流	最大 5A
输出接通延迟时间	10ms（最大值）
输出断开延迟时间	5ms（最大值）
触点预期寿命	

<ul style="list-style-type: none"> <li>机械寿命（空载）</li> <li>电气寿命（额定负载）</li> </ul>	20, 000, 000 次（1200 次/分钟） 100, 000 次（6 次/分钟）
隔离特性 <ul style="list-style-type: none"> <li>隔离方式</li> <li>线圈与触点的隔离电压</li> </ul>	继电器 2000Vrms
状态指示	各通道绿色 LED 指示输出“1”

➤ 本体 AI 通道参数

信号形式	4-20mA、1-5V、0-20mA、0-10V
分辨率	12 位
测量精度	0.3% F.S.
转换速率（每通道）	约 200 次/秒
输入阻抗	电流模式：≤250 Ω 电压模式：>4M Ω
抗共模电压	（信号电压+共模电压）≤15V。

➤ 本体 AO 通道参数

信号形式	4-20mA、1-5V、0-20mA、0~10V
分辨率（含符号位）	12 位
输出信号精度	0.3% F.S.
转换速率（每通道）	约 200 次/秒
外部负载	电流模式：最大 500 Ω 电压模式：最小 1K Ω

➤ 本体 RTD 通道参数

电气参数	
信号形式	Pt100、Cu50、Pt1000、电阻
接线形式	两线制或者三线制
分辨率（含符号位）	24 位
测量精度	温度：±0.5℃； 电阻：±2Ω
转换速率（每通道）	约 1 次/秒
输入阻抗	>1MΩ



## 第三章 软件功能及使用

### 3.1 概述

Kinco K2 系列延续了 Kinco K5 等老产品的设计,同样采用 Kincobuilder 编程软件及相同的指令集。不同的地方主要是部分功能有所增强,详细请参考本手册的中列出的所有内容。对于绝大部分通用功能及指令请参考 Kincobuilder 编程软件【帮助】下【帮助主题】或下载 Kinco K5/K2 系列软件使用手册,下载链接:

[http://download.kinco.cn/D\\_UserManual/PLC/Kinco\\_K5K2PLC\\_Software\\_Manual\\_20151029.pdf](http://download.kinco.cn/D_UserManual/PLC/Kinco_K5K2PLC_Software_Manual_20151029.pdf)

### 3.2 安装 USB 编程口的驱动程序

Kinco-K2 提供了 USB 编程口,编程口在电脑上作为一个虚拟串口来使用,它的驱动程序存放在 Kincobuilder 编程软件安装目录下面的 \Drivers 目录下,目前支持 Windows XP、Windows 7 和 Windows 8 系统。如下图所示:



当用户第一次使用编程数据线连接 K2 和电脑时,Windows 系统会自动检测到新硬件并提示安装驱动程序,此时用户根据自己的 Windows 版本选择相应目录下的驱动程序即可。

#### ➤ 在 Windows 7 系统下,无法安装驱动程序怎么办?

##### 方法一:

这是因为使用的是精简过的 Win7 系统,系统缺少 mdmcpq.inf 和 usbser.sys 文件,从而导致了无法安装虚拟串口。

在操作系统的.\Win7\windows 目录下,存放着上述两个系统文件,用户将这两个文件按下面的说明复制到相应文件夹下,然后重新安装驱动程序,就可以解决:

- 将 mdmcpq.inf 复制到 C:\WINDOWS\INF 目录;
- 将 usbser.sys 复制到 C:\WINDOWS\SYSTEM32\DRIVERS\目录。

### 方法二:

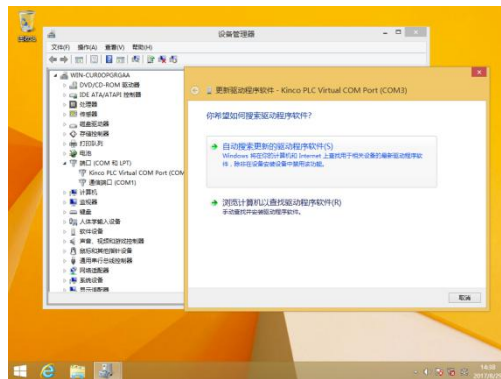
以上方法无法安装驱动程序时，原因是因为 WINDOWS 操作系统不是全功能安装版，缺少系统必要的文件，此时可使用第三方驱动程序安装软件进行自动搜索安装（如驱动精灵）

### 方法三:

以上方法都无法安装驱动程序时，win7 32bit 和 64bit 的用户可以发电脑系统盘，此路径下 C:\Windows\inf\setupapi.dev.log 的文件到技术支持邮箱 liuhl@kinco.cn。我们按照这个可以看到用户的 ghost win7 缺失什么系统文件，直接做个新补丁发回给用户，收到回复的邮件装上补丁即可安装驱动成功。

### ➤ 在 Windows 8 系统下，如何安装驱动程序？

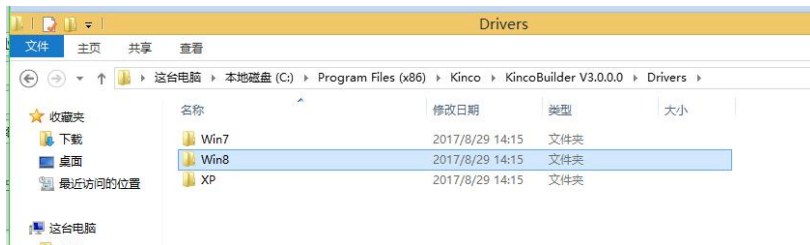
1、如果你可以联网，那么 win8 win10 使用自动搜索更新的驱动程序软件，最方便，可以自动安装好驱动

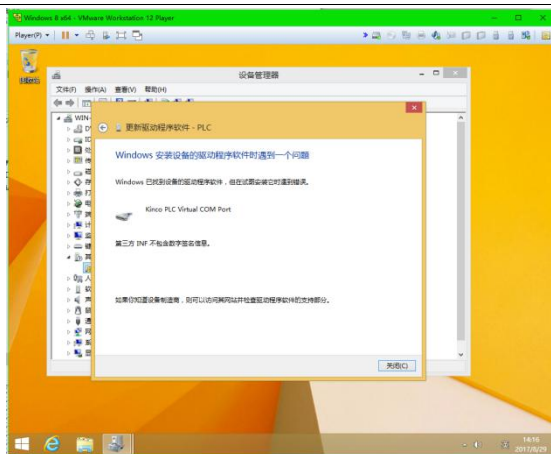


2、如果你没网络，请按照下图指示来安装。

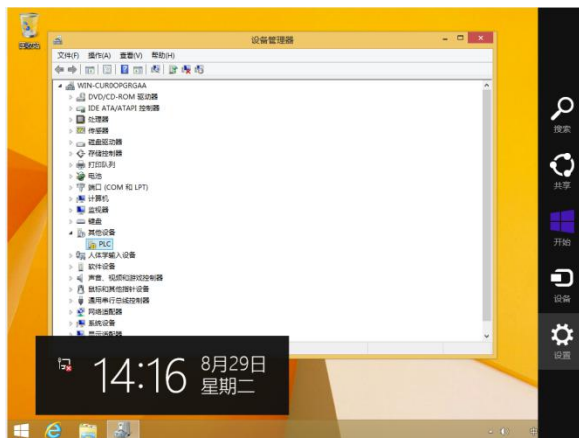
下面的截图是 win8 下高级启动的截图，win10 与 win8 类似，也是从设置里面找到高级启动然后选 7 禁止驱动程序强制签名

(1) 安装 PLC 驱动，按 windows 的提示，选择 win8 里面的驱动文件。出错的提示的样子

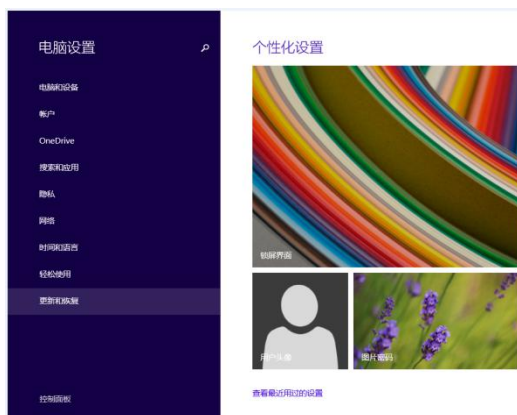




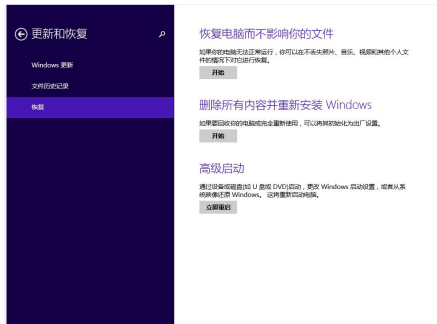
(2) 鼠标指向 win8 的右下角， 出来设置， 点设置



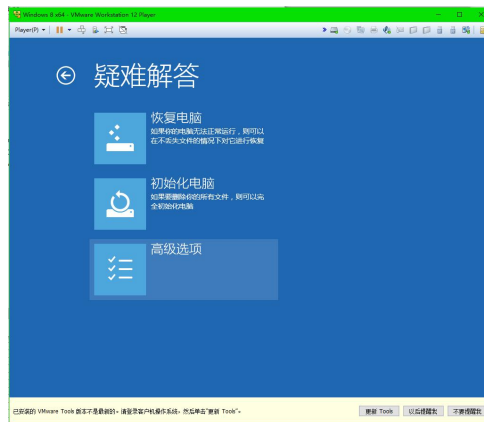
(3) 在设置里面，点更新和恢复



(4) 在里面点更新小项，然后点高级启动下面的立即启动。



(5) 点高级选项



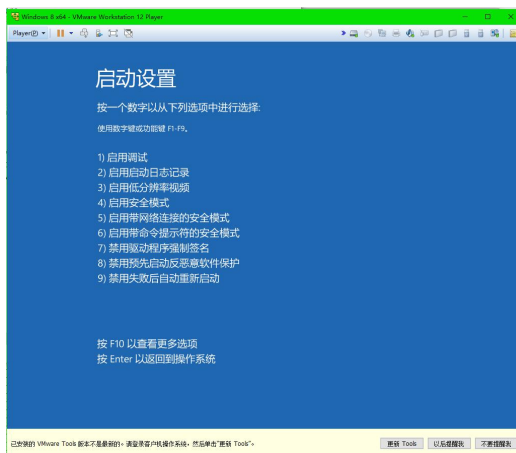
(6) 点启动设置



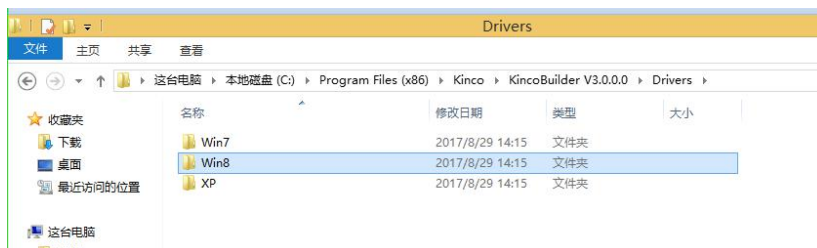
(7) 点重启

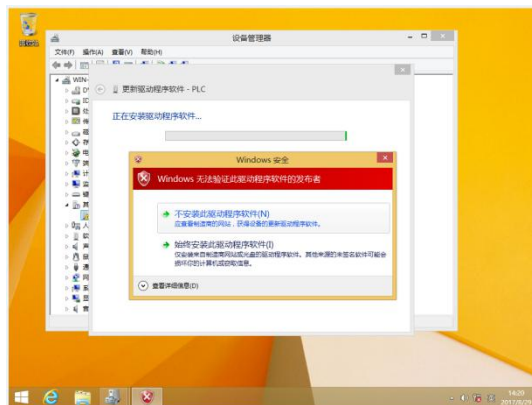


(8) 这是电脑重启后的界面，选 7 禁用驱动程序强制签名，电脑就开始启动

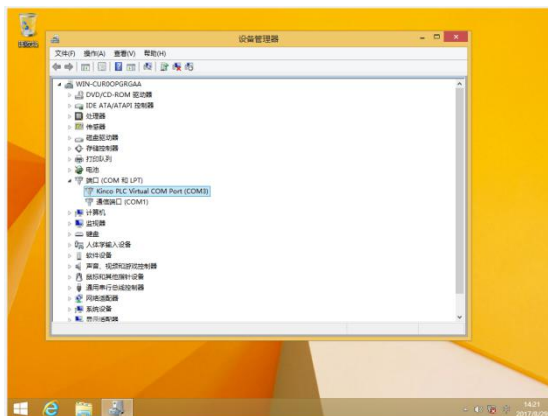
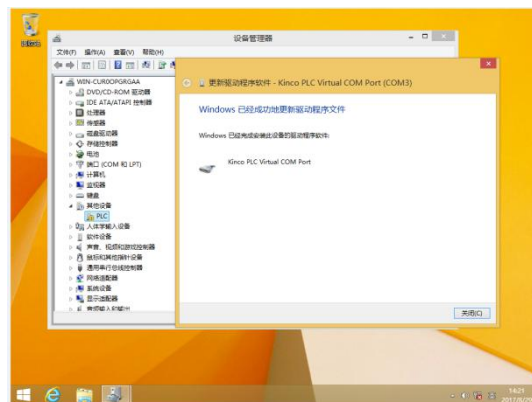


(9) 重新安装 PLC 驱动，按 windows 的提示，选择 win8 里面的驱动文件。  
会弹出如下提示，选始终安装此驱动程序软件





(10) 安装成功显示如下图



### 3.3 使用 Modbus TCP 协议与第三方设备通讯

Modbus TCP 主站可访问的内存区域分类如下：

类型	Modbus 功能码	对应的 PLC 内存区域
DO（开关量输出，0XXXX）	01, 05, 15	Q 区, M 区
DI（开关量输入，1XXXX）	02	I 区, M 区
AO（模拟量输出，4XXXX）	03, 06, 16	AQ 区, V 区
AI（模拟量输入，3XXXX）	04	AI 区, V 区
错误记录（16 位无符号整数）	03,04	PLC 错误记录区

一次命令最大访问的寄存器数量如下：

1. 读取位，一次最大读取 1600 个位(200 字节)。(01, 02 功能码)
2. 写入位，一次最大写入 800 个位。(15 功能码)
3. 读取字，一次最大读取 100 个字。(03, 04 功能码)
4. 写入字，一次最大写入 100 个字。(16 功能码)
5. 当操作的内存区小于上面的最大值时，只允许一次操作整个内存区。比如报文中读取 AI 区 90 个字，这是错误的，因为 AI 区最大 32 个字。

#### 3.3.1 Modbus 寄存器编号

由于各种规格的 CPU 的内存区域不同，所以允许访问的范围也有限制，其他规格的参考编程软件 KincoBuilder 里的帮助主题中的附录 A 使用 Modbus RTU 协议通讯。

若 Modbus TCP 主站的寄存器从 1 开始编号，那么将将下表中的寄存器号直接加 1 即可。

➤ 适用于 CPU204

内存区域	范围	类型	对应的 Modbus 寄存器号*
I	I0.0 --- I0.7	DI	0 --- 7
Q	Q0.0 --- Q0.5	DO	0 --- 5
M	M0.0 --- M1023.7	DI/DO	320 -- 8511
AI	AIW0 --- AIW0	AI	0---0
AQ	AQW0 --- AQW0	AO	0---0
V	VW0 --- VW4094	AI/AO	100 -- 2147

- 除以上内存区域支持 Modbus 访问外,K 系列 PLC 还提供了错误记录的内存区域供用户通过 Modbus 读取查看, 详见编程软件 KincoBuilder 里的帮助主题中的附录 D 错误诊断功能

### 3.4 高速计数器的使用

K2 提供了 4 路高速计数器, HSC0 --- HSC3。K209M 提供 2 路高速计数器, HSC0/HSC1。各系列 CPU 最高计数频率如下

CPU 型号	单相		AB 相	
	HSC0 和 HSC1	HSC2 和 HSC3	HSC0 和 HSC1	HSC2 和 HSC3
K204ET	200K	200K	200K	200K
K205	50K	20K	50K	10K
K209EA	200K	20K	100K	10K
K209M	200K	/	200K	/

高速计数器具有多种工作模式, 所有的高速计数器在相同的工作模式下均具有相同的功能。

所有高速计数器均允许指定最大 32 个预置值 (PV), 每个 PV 值均支持“计数值=预置值”中断。PV 值可以指定为相对值或者绝对值方式, 若选择为相对值方式, 那么“计数值=预置值”中断允许选择循环发生。

#### 3.4.1 高速计数器工作模式和输入信号

高速计数器的输入信号包括如下几种: 时钟 (即输入脉冲)、方向、启动和复位信号。

在不同的工作模式下, 所需要的输入信号也有所不同。下面各表详细描述了各个高速计数器所支持的工作模式及其输入信号的分配。

HSC 0				
模式	描述	I0.1	I0.0	I0.5
0	带内部方向控制的单相增/减计数器 方向控制位: SM37.3	时钟		
1			复位	
2			复位	启动
3	带外部方向控制的单相增/减计数器	时钟		方向
4			复位	方向



6	带增/减计数时钟输入的双相计数器	时钟（减）	时钟（增）	
9	A/B 相正交计数器	时钟 A 相	时钟 B 相	

HSC1					
模式	描述	I0.4	I0.6	I0.3	I0.2
0	带内部方向控制的单相增/减计数器 方向控制位：SM47.3			时钟	
1		复位			
2		复位	启动		
3	带外部方向控制的单相增/减计数器			时钟	方向
4		复位			方向
6	带增/减计数时钟输入的双相计数器			时钟（减）	时钟（增）
7		复位			
9	A/B 相正交计数器			时钟 A	时钟 B
10		复位			

HSC 2			
模式	描述	I0.4	I0.5
0	带内部方向控制的单相增/减计数器。方向控制位：SM57.3		时钟
9	A/B 相正交计数器	时钟 B 相	时钟 A 相

HSC 3			
模式	描述	I0.6	I0.7
0	带内部方向控制的单相增/减计数器。方向控制位：SM127.3		时钟
9	A/B 相正交计数器	时钟 B 相	时钟 A 相

### 3.4.2 控制寄存器和状态寄存器

#### ➤ 控制寄存器

在 SM 区中为每个高速计数器均提供了如下控制寄存器用于存放配置数据。其中，当前值用于修改计数器当前的计数值，若将当前值写入高速计数器，那么高速计数器就会立即从这个新数值开始计数。下表详细描述了这些寄存器。

HSC0	HSC1	HSC2	HSC3	描述
SM37.0	SM47.0	SM57.0	SM127.0	复位信号的有效电平：0=高电平；1=低电平
SM37.1	SM47.1	SM57.1	SM127.1	启动信号的有效电平：0=高电平；1=低电平
SM37.2	SM47.2	SM57.2	SM127.2	正交计数器速率：0=1x 速率；1=4x 速率*
SM37.3	SM47.3	SM57.3	SM127.3	计数方向：0=减计数；1=增计数。
SM37.4	SM47.4	SM57.4	SM127.4	是否向 HSC 中写入计数方向：0=否；1=是。
SM37.5	SM47.5	SM57.5	SM127.5	是否向 HSC 中写入新预置值：0=否；1=是。
SM37.6	SM47.6	SM57.6	SM127.6	是否向 HSC 中写入新当前值：0=否；1=是。
SM37.7	SM47.7	SM57.7	SM127.7	是否允许该高速计数器：0=禁止；1=允许。
HSC0	HSC1	HSC2	HSC3	描述
SMD38	SMD48	SMD58	SMD128	当前值
SMD42	SMD52	SMD62	SMD132	预置值

HSC0	HSC1	HSC2	HSC3	描述
SM141.0	SM151.0	SM161.0	SM171.0	是否使用多段预置值：0=否；1=是
SM141.1	SM151.1	SM161.1	SM171.1	预置值是相对值还是绝对值：0=绝对；1=相对
SM141.2	SM151.2	SM161.2	SM171.2	预置值比较（“CV=PV”）中断是否循环产生： 0=否；1=是。 注意：只有相对值方式才允许设定为循环产生。
SM141.3	SM151.3	SM161.3	SM171.3	保留
SM141.4	SM151.4	SM161.4	SM171.4	是否更新段数及预置值：0=否；1=是
SM141.5	SM151.5	SM161.5	SM171.5	是否复位中断变量：0=是；1=否
SM141.6	SM151.6	SM161.6	SM171.6	保留
SM141.7	SM151.7	SM161.7	SM171.7	保留
HSC0	HSC1	HSC2	HSC2	描述
SMW142	SMW152	SMW162	SMW172	预置值表的起始位置（用相对于 VB0 的字节偏移来表示），必须为奇数。

需要注意的是，控制字节中并非所有的控制位都适用于所有的工作模式。比如，“计数方向”和“是否向 HSC 中写入计数方向”这两个控制位就只用于模式 0、1 和 2（带内部方向控制的单相增/减计数器），若高速计数器所用的工作模式是采用外部的方向控制信号，那么这两个控制位就会被忽略。

控制字节、当前值和预置值上电后的缺省值均为 0。

#### ➤ 状态寄存器

每个高速计数器都在 SM 区中提供了状态寄存器用于指明高速计数器当前的状态信息。

HSC0	HSC1	HSC2	HSC3	描述
SM36.0	SM46.0	SM56.0	SM126.0	保留
SM36.1	SM46.1	SM56.1	SM126.1	保留
SM36.2	SM46.2	SM56.2	SM126.2	保留
SM36.3	SM46.3	SM56.3	SM126.3	多段 PV 值表设置是否有错误：0=否，1=是。
SM36.4	SM46.4	SM56.4	SM126.4	保留
SM36.5	SM46.5	SM56.5	SM126.5	当前计数方向：0=减；1=增。
SM36.6	SM46.6	SM56.6	SM126.6	当前计数值是否等于预置值：0=否；1=是。
SM36.7	SM46.7	SM56.7	SM126.7	当前计数值是否大于预置值：0=否；1=是。
HSC0	HSC1	HSC2	HSC3	描述
SMB140	SMB150	SMB160	SMB170	正在运行的 PV 值段序号（从 0 开始）。

### 3.4.3 预置值（PV 值）设定

K3 和 K5 中，高速计数器只允许设定一个 PV 值。

K2 允许每个高速计数器设定最多 32 个 PV 值，允许选择 PV 之间的关系是相对值或者绝对值，允许“CV=PV”中断循环产生。同时，K2 也兼容 K5 的单 PV 值设定方式。

下面将以 HSC0 为例来对 PV 值的功能和设定方法进行详细的描述。

#### ➤ 如何选择多段 PV 值

每个高速计数器的控制寄存器里都提供了一个控制位，用于选择是否使用多段预置值。

HSC0 的这个控制位是 SM141.0。

若 SM141.0 为 0，则表示采用单 PV 值方式，与 K5 的用法一致：SMD42 指定了新 PV 值，SM37.5 指明是否使用这个新的 PV 值。

若 SM141.0 为 1，则表示采用多段 PV 值方式，此时 SM37.5、SMD42 无效。各 PV 值存放于 PV 值表中（SMW142 为表起始地址），SM141.4 指明是否使用 PV 值表中的数据。若 SM141.4 为 1，则表示本次启动后，高速计数器时将采用 PV 值表中的数据。若 SM141.4 为 0，则表示本次启动后，高速计数器将采用上一次的 PV 值数据，而忽略 PV 值表中的数据。

#### ➤ 多段 PV 值表

若使用多段 PV 值，那么各 PV 值将采用 PV 值表中的数据。

每个高速计数器的控制寄存器里都提供一个控制字，用于存放 PV 值表的起始地址，表的起始地址必须为 V 区中的奇数地址，例如 301（表示 VB301）。

PV 值表的格式如下。

字节偏移 <sup>(1)</sup>	数据类型	描述
0	BYTE	PV 值个数
1	DINT	第 1 个 PV 值
5	DINT	第 2 个 PV 值
...	DINT	...

- (1) 所有偏移量均是相对于表起始位置的偏移字节数。
- (2) 当采用相对值方式时，PV 值的数学绝对值必须大于 1，否则 PLC 认为段数到此结束，并以此统计 PV 值个数（优先于个数设定值）；  
  
当采用绝对值方式时，相邻两个 PV 值之间的差值的数学绝对值必须大于 1，否则 PLC 认为段数到此结束，并以此统计 PV 值个数（优先于个数设定值）；
- (3) 用户设定 PV 值时需要注意：“CV=PV”中断必须依次产生。也就是说，当计数值达到第 1 个 PV 值并产生中断后，接下来 PLC 将会与第 2 个 PV 值进行比较，依次类推。
- (4) PV 值设置必须合理。以相对值为例，当增计数时，PV 值必须大于 0，否则该值对应的“CV=PV”中断也许永远不会产生；当减计数时，PV 值必须小于 0，否则该值对应的“CV=PV”中断也许永远不会产生。

#### ➤ 相对值和绝对值方式

每个高速计数器的控制寄存器里都提供了一个控制位，用于选择 PV 值是相对值或者绝对值方式。HSC0 的这个控制位是 SM141.1。

若 SM141.1 为 0，则表示 PV 值是绝对值方式。当计数值等于 PV 值时，将会产生相应的“CV=PV”中断。例如，若设定 3 个 PV 值，依次是 1000、2000、3000，那么计数值达到 1000 时，将产生第 1 个“CV=PV”中断；当计数值达到 2000 时，将产生第 2 个“CV=PV”中断；后面依次类推。

若 SM141.1 为 1，则表示 PV 值是相对值方式，若计数器以当前的计数值为基准，继续计数使得差值等于 PV 值时，将会产生相应的“CV=PV”中断。例如，若设定 3 个 PV 值，分别为 10、1000、1000，而且在高速计数器启动时的计数值是 100，那么当计数值分别达到 110、1110、2110 时，将分别产生“CV=PV”中断。

#### ➤ “CV=PV”中断循环产生。

只有 PV 值是相对值方式时，才允许设定为循环产生中断，否则无效。

若 SM141.0 为 0，则表示“CV=PV”中断只产生一次。当所有 PV 值对应的中断都发生完成后就会

停止。若要继续产生，则必须修改相应的寄存器值并再次调用 HSC 指令。

若 SM141.0 为 1，则表示“CV=PV”中断会循环产生。当最后一个 PV 值对应的中断发生完成时，PLC 将以当前的计数值为基准，与各个 PV 值再次相加，得到新的中断所需数值，然后继续与计数值进行比较，产生相应的“CV=PV”中断。这个过程会一直循环进行，永不停止。

例如，若设定 3 个 PV 值，分别为 10、1000、1000，而且在高速计数器启动时的计数值是 100，则各个中断每次产生所需数值如下：

当前计数值	中断次数	第 1 个值	第 2 个值	第 3 个值
100	第 1 次	110	1110	2110
2110	第 2 次	2120	3120	4120
4120	第 3 次	4130	5130	6130
...	第 n 次	...	...	...

### 3.4.4 “CV=PV” 中断编号

当采用单 PV 值方式时，那么高速计数器完全兼容 K5，即 HSC0 和 HSC1 的中断的编号与 K5 中的一致，且另外增加 HSC2 和 HSC3 的中断编号。

当采用多段 PV 值方式时，高速计数器为 32 个 PV 值均分配了一个新的中断编号，如下表。

高速计数器	中断事件号		描述
HSC0	单 PV 值方式	18	“CV=PV”中断
	多 PV 值方式	64	第 1 个 PV 值的“CV=PV”中断
		65	第 2 个 PV 值的“CV=PV”中断
		...	...（依次加 1）
		95	第 32 个 PV 值的“CV=PV”中断
HSC1	单 PV 值方式	15	“CV=PV”中断
	多 PV 值方式	96	第 1 个 PV 值的“CV=PV”中断
		97	第 2 个 PV 值的“CV=PV”中断
		...	...（依次加 1）
		127	第 32 个 PV 值的“CV=PV”中断
HSC2	单 PV 值方式	12	“CV=PV”中断
	多 PV 值方式	128	第 1 个 PV 值的“CV=PV”中断
		129	第 2 个 PV 值的“CV=PV”中断
		...	...（依次加 1）

		159	第 32 个 PV 值的 “CV=PV”中断
HSC3	单 PV 值方式	9	“CV=PV”中断
	多 PV 值方式	160	第 1 个 PV 值的 “CV=PV”中断
		161	第 2 个 PV 值的 “CV=PV”中断
		...	...（依次加 1）
		191	第 32 个 PV 值的 “CV=PV”中断

### 3.4.5 高速计数器的使用方法

#### ➤ 方法一：使用相关指令进行编程

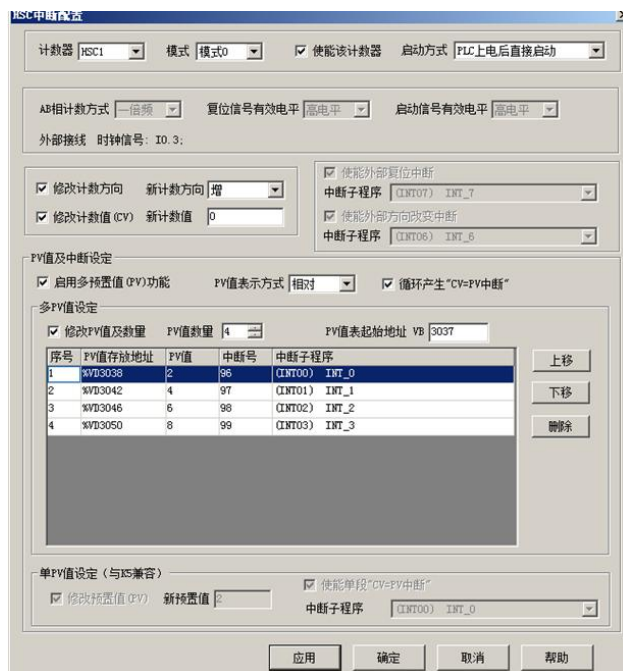
这种方法也是在 K3 和 K5 中使用的方法，K2 也支持这种方法。总体步骤如下：

- 1) 配置该高速计数器的控制字节，并指定当前值（也就是计数的起始值）和预置值；
- 2) 使用 HDEF 指令来定义一个高速计数器及其工作模式；
- 3) （可选）使用 ATCH 指令为高速计数器中断连接相应的中断服务程序；
- 4) 使用 HSC 指令来配置并启动高速计数器。

#### ➤ 方法二：使用 HSC 向导

在 K2 中，为高速计数器提供了配置向导。用户可以直接利用该向导对所有的高速计数器进行配置，无需再进行复杂的编程。向导如下图。

即使通过向导对 HSC 进行了配置，用户也可以在程序中按照“方法一”来随时对高速计数器进行参数修改、启动、停止。



HSC 向导的使用方法如下：

- 1) 在【计数器】中，选择将要使用的计数器。
- 2) 选中【使能该计数器】，然后将会允许进行后续的配置。
- 3) 在【模式】中，选择将要使用的计数器模式。
- 4) 在【启动方式】中，选择该高速计数器的启动方式。

启动方式有如下两种：

“**在程序中调用 HSC 指令**”：若选择这种方式，那么在用户程序中，通过调用 HSC 指令来启动该计数器。在调用 HSC 指令之前，无需再配置各寄存器和调用 HDEF 指令。

“**PLC 上电后直接启动**”：若选择这种方式，那么该高速计数器在 PLC 上电后就自动运行，无需调用任何指令。

- 5) 若要使用多段 PV 值方式，则选中【启用多预置值 (PV) 功能】，然后可以对 PV 值、数量、关联的中断子程序等进行配置。若选中【修改 PV 值及数量】，则可以调整【PV 值数量】中的数值，从而修改 PV 值个数。
- 6) 若要使用单 PV 值方式，则首先选中“单 PV 值设定 (与 K5 兼容)”中的【修改 PV 值】，然后可以修改 PV 值及关联的中断子程序。

7) 其它的配置项, 请参考前文的描述, 按实际需求进行配置。

### 3.5 高速脉冲输出功能的使用

Kinco-K2 提供了 3 路高速脉冲输出, 输出通道分别使用 Q0.0、Q0.1 和 Q0.4。各系列 CPU 最高输出频率如下 (注意: Q0.0 和 Q0.1 最高输出频率时要求负载电阻不大于  $3K\Omega$ )。

CPU 型号	Q0.0	Q0.1	Q0.4
K204ET	200K	200K	200K
K205	50K	50K	10K
K209EA	200K	200K	10K
K209M	200K	200K	200K

K209M 提供 4 路高速脉冲输出, 所用通道分别为 Q0.0、Q0.1 和 Q0.4、Q0.5, 都支持 PTO (脉冲串) 和 PWM (脉宽调制) 方式输出。其中, Q0.0 和 Q0.1、Q0.4 通道的最高输出频率可达 200KHz (要求负载电阻不大于  $3K\Omega$ ), Q0.5 通道的最高输出频率为 10KHz。

针对定位控制指令, Kinco-K2 系列为每路高速输出均指定了一个方向输出通道, 同时还在 SM 区中提供了一个方向使能控制位。如下表。

	Q0.0	Q0.1	Q0.4	Q0.5
方向输出通道	Q0.2	Q0.3	Q0.6	Q0.7
方向使能控制位	SM201.3	SM231.3	SM251.3	SM221.3

方向输出通道用于输出电机的方向控制信号, 正转时输出为 0, 反转时输出为 1。

方向使能控制位用来禁止或者允许使用相应的方向输出通道。方向使能控制位具有最高的优先级, 若设置为禁止, 那么定位控制指令执行时将不会输出方向控制信号, 相应的输出通道就可以作为普通的 DO 点使用。

#### 3.5.1 高速脉冲输出指令

K2 的指令集中提供了如下 3 种指令用于高速输出功能:

- 1) PLS 指令: 可以实现 PTO (单段或者多段) 和 PWM 输出功能。
- 2) 定位控制指令: 共计 5 条指令, 包括 PREL (相对运动)、PABS (绝对运动)、PHOME (回原点)、PJOG (点动)、PSTOP (急停) 指令, 用户能够很方便地实现简单的定位控制功能。**注意: 当使用定位控制指令时, 输出脉冲频率不能低于 125Hz!**
- 3) 跟随指令 PFLO\_F: 在输入参数中有输入频率 (F)、电子齿轮比 (NUME、DENOM)、脉冲个

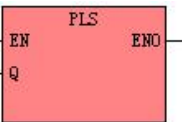


数(COUNT)等, 这些参数均可以使用变量。该指令的输出脉冲频率等于输入频率乘以电子齿轮比, 当输出脉冲个数达到 COUNT 个则停止输出并设置完成标志位。**注意: 当使用跟随指令时, 输出脉冲频率不能低于 30Hz!**

### 3.5.2 PLS 指令的使用

PLS 指令可以实现 PTO 或者 PWM 输出功能。

- PTO: Pulse Train Output, 脉冲串输出。
  - PWM: Pulse-Width Modulation, 脉宽调制。
- 指令及其操作数说明

	名称	指令格式	影响 CR 值	适用于
<b>LD</b>	PLS			<ul style="list-style-type: none"> <li>• K5</li> <li>• K2</li> </ul>
<b>IL</b>	PLS	PLS <i>Q</i>	U	

参数	输入/输出	数据类型	允许的内存区
<i>Q</i>	输入	INT	常量 (0、1 或者 2)

PLS 指令的作用是: 读取 SM 区中相应控制寄存器的值并配置高速脉冲输出的特性, 然后启动高速脉冲输出, 直到完成指定的脉冲输出功能。脉冲输出通道由参数 *Q* 指定, 0 表示使用 Q0.0 输出, 1 表示使用 Q0.1 输出, 2 表示使用 Q0.4 输出。

**注意:** 用户程序中, 仅在需要时执行一次 PLS 指令即可, 建议利用边沿指令的输出结果来调用 PLS 指令。若 PLS 的 EN 端一直保持为 1, 那么 PLS 指令将无法正常工作。

- **LD**  
若 EN 值为 1, 则 PLS 指令被执行。
- **IL**  
若 CR 值为 1, 则 PLS 指令被执行。该指令的执行不影响 CR 值。

#### 3.5.2.1 Kinco-K2 的高速脉冲输出功能

Kinco-K2 支持 3 路高速脉冲输出, 相应地就提供了 3 个 PTO/PWM 脉冲发生器用于产生 PTO/PWM

输出。其中，第一个脉冲发生器分配在 Q0.0，称为 PWM0 或者 PTO0；第二个分配在 Q0.1，称为 PWM1 或者 PTO1；第三个分配在 Q0.4，称为 PWM2 或者 PTO2。

PTO/PWM 发生器和 DO 映像寄存器共同使用内存地址 Q0.0、Q0.1 和 Q0.4。如果用户程序中调用了某个通道的高速输出指令，那么 PTO/PWM 发生器将控制输出通道，并禁止普通 DO 的输出。



*注意：若 Q0.0、Q0.1 或者 Q0.4 是继电器类型的，则避免使用高速脉冲输出功能！*

#### ➤ PWM

PWM 功能提供占空比可调的连续脉冲输出。用户可以控制输出的周期和脉宽。

周期和脉宽的单位可以选择微秒 ( $\mu\text{s}$ ) 或毫秒 (ms)，最大周期值为 65535。当脉宽大于等于周期时，占空比自动地被设为 100%，输出一直接通。当脉宽为 0 时，占空比为 0%，输出断开。

#### ➤ PTO

PTO 功能能够产生指定脉冲个数的脉冲串方波（50%占空比）。用户可以控制输出方波的周期和输出脉冲的个数。脉冲周期的单位是微秒 ( $\mu\text{s}$ ) 或者毫秒 (ms)，最大周期值为 65535。脉冲个数的范围是：2~4,294,967,295。如果指定脉冲数小于 2，则 Kinco-K2 将设置相应的错误标志位并禁止输出。

PTO 功能提供了单段操作和多段操作两种模式。

- **单段操作**

在单段操作模式下，每次执行 PLS 指令后仅会进行一次脉冲串输出。

- **多段操作**

在多段操作模式下，CPU 自动从 V 区的包络表中读出每个 PTO 段的设定值并依据设定值执行该段 PTO。

各段在包络表中的设置均占用 8 个字节，包括一个周期值（16 位无符号整数）、保留值（暂时未用到，16 位符号整数）和一个脉冲个数值（32 位无符号双整数）。也就是说，在同一段中，所有脉冲的输出频率是相同的。多段操作使用 PLS 指令来配置并启动。

包络表的起始位置存储在 SMW168（对应 PTO0）、SMW178（对应 PTO1）、SMW268（对应 PTO2）中，时基通过 SM67.3（对应 PTO0）、SM77.3（对应 PTO1）、SM87.3（对应 PTO1）设置，可以选择微秒或毫秒。包络表中的所有周期值必须使用同一个时基，并且在包络执行时不能改变。

包络表的格式如下表所示。

字节偏移 <sup>(1)</sup>	长度	段数	描述
---------------------	----	----	----

0	8 位		段数 (1 到 64)
1	16 位	第 1 段	初始周期 (2 到 65535 时基)
3	16 位		保留
5	32 位		脉冲个数 (1 到 4,294,967,295)
9	16 位	第 2 段	初始周期 (2 到 65535 时基)
11	16 位		保留
13	32 位		脉冲数 (1 到 4,294,967,295)
...		...	...

(1) 所有偏移量均是相对于包络表起始位置的偏移字节数。



注意: 包络表的起始位置必须为 V 区中的奇数地址, 如 VB3001。

### 3.5.2.2 PTO/PWM 寄存器

在 SM 区中为每个 PTO/PWM 发生器均提供了一些控制寄存器用于存放其配置数据。如下表。

Q0.0	Q0.1	Q0.4	Q0.5	描述
SM67.0	SM77.0	SM97.0	SM107.0	PTO/PWM 是否更新周期值: 0=否; 1=是
SM67.1	SM77.1	SM97.1	SM107.1	PWM 是否更新脉宽值: 0=否; 1=是
SM67.2	SM77.2	SM97.2	SM107.2	PTO 是否更新脉冲个数: 0=否; 1=是
SM67.3	SM77.3	SM97.3	SM107.3	PTO/PWM 时基: 0=1 $\mu$ s; 1=1ms
SM67.4	SM77.4	SM97.4	SM107.4	PWM 更新方法: 0=异步更新; 1=同步更新
SM67.5	SM77.5	SM97.5	SM107.5	PTO 操作方式: 0=单段操作; 1=多段操作
SM67.6	SM77.6	SM97.6	SM107.6	功能选择: 0=PTO; 1=PWM
SM67.7	SM77.7	SM97.7	SM107.7	PTO/PWM 允许或禁止此功能: 0=禁止; 1=允许
Q0.0	Q0.1	Q0.4	Q0.5	描述
SMW68	SMW78	SMW98	SMW108	PTO/PWM 周期值, 范围 2~65535
SMW70	SMW80	SMW100	SMW110	PWM 脉宽值, 范围 0~65535
SMD72	SMD82	SMD102	SMD112	PTO 脉冲个数, 范围 1~4,294,967,295
SMW168	SMW178	SMW218	SMW248	包络表的起始位置 (用相对于 VB0 的字节偏移来表

				示)，仅用于 PTO 多段操作。
--	--	--	--	------------------

所有控制字节、周期、脉冲数的缺省值都是 0。用户修改 PTO/PWM 波形的特性的方法是：首先设置相应的控制寄存器，如果是 PTO 多段操作，包络表也得先设置好，然后再执行 PLS 指令。

在 SM 区中也为每个 PTO/PWM 发生器均提供了一个状态字节，用户可以通过访问状态字节来了解 PTO/PWM 发生器的当前状态信息。如下表。

Q0.0	Q0.1	Q0.4	Q0.5	描述
SM66.0	SM76.0	SM96.0	SM106.0	保留
SM66.1	SM76.1	SM96.1	SM106.1	保留
SM66.2	SM76.2	SM96.2	SM106.2	保留
SM66.3	SM76.3	SM96.3	SM106.3	PWM 是否空闲：0=否；1=是
SM66.4	SM76.4	SM96.4	SM106.4	PTO 周期值、脉冲个数设置是否有错误：0=否；1=是 <b>注：周期值、脉冲个数必须大于 1。</b>
SM66.5	SM76.5	SM96.5	SM106.5	PTO 是否由于用户命令而终止：0=否；1=是
SM66.6	SM76.6	SM96.6	SM106.6	保留
SM66.7	SM76.7	SM96.7	SM106.7	PTO 是否空闲：0=忙；1=空闲

PTO 空闲位、PWM 空闲位指明了 PTO 输出、PWM 输出是否已经结束。

### 3.5.2.3 使用 PTO 功能

下面以 PTO0 为例来介绍如何编程使用 PTO 功能。

总体上，使用 PTO 包括两个步骤：设置相关的控制寄存器，初始化 PTO；执行 PLS 指令。

建议用户在工程中尽量编写单独的初始化子程序，这样可以使整个用户工程具有良好的结构。另外，若有可能的话，尽量在主程序中以 SM0.1 为条件来调用这个初始化子程序，这样该子程序将只在 CPU 上电后的首次扫描中调用并执行一次，可以减少 CPU 的扫描时间。

#### ➤ 执行 PTO（单段操作）

- 1) 根据期望的操作来设置控制字节 SMB67。

例如，SMB67 = B#16#85 表明了：

- 允许 PTO/PWM 功能；
- 选择使用 PTO 功能，单段操作；
- 时基选择为 1μS；
- 允许更新脉冲个数和周期值。

- 2) 将期望的周期值赋给 SMW68。

- 3) 将期望的脉冲个数赋给 SMD72。
- 4) (可选) 使用 *ATCH* 指令为“PTO0 完成”中断事件 (事件号 28) 连接一个中断服务程序以实现对该中断事件的快速响应。
- 5) 执行 *PLS* 指令来配置并启动 PTO0。

#### ➤ 改变 PTO 周期 (单段操作)

按照如下步骤来改变 PTO0 周期值:

- 1) 根据期望的操作来设置控制字节 SMB67:

例如,  $SMB67 = B\#16\#81$  表明了:

- 允许 PTO/PWM 功能;
- 选择使用 PTO 功能, 单段操作;
- 时基选择为  $1\mu S$ ;
- 允许更新周期值。

- 2) 将期望的周期值赋给 SMW68。

- 3) 执行 *PLS* 指令来配置并启动 PTO0, 具有新周期值的 PTO 就会立即接着启动。

#### ➤ 改变 PTO 脉冲个数 (单段操作)

按照如下步骤来改变 PTO0 输出的脉冲个数:

- 1) 根据期望的操作来设置控制字节 SMB67:

例如,  $SMB67 = B\#16\#84$  表明了:

- 允许 PTO/PWM 功能;
- 选择使用 PTO 功能, 单段操作;
- 时基选择为  $1\mu S$ ;
- 允许更新脉冲个数。

- 2) 将期望的脉冲个数赋给 SMD72。

- 3) 执行 *PLS* 指令来配置并启动 PTO0, 就会立即接着输出新指定个数的脉冲。

#### ➤ 执行 PTO (多段操作)

- 1) 根据期望的操作来设置控制字节 SMB67。

例如,  $SMB67 = B\#16\#A0$  表明了:

- 允许 PTO/PWM 功能;
- 选择使用 PTO 功能
- 选择多段操作;

- 时基选择为 1 $\mu$ S;
- 2) 将包络表的起始位置（奇数，表示包络表起始地址相对于 VB0 的字节偏移）赋给 SMW168。
- 3) 设置包络表中的相关数值。
- 4) （可选）使用 *ATCH* 指令为“PTO0 完成”中断事件（事件号 28）连接一个中断服务程序以实现对该中断事件的快速响应。
- 5) 执行 *PLS* 指令来配置并启动 PTO0。

#### 3.5.2.4 使用 PWM 功能

下面以 PWM0 为例来介绍如何编程使用 PWM 功能。

总体上，使用 PWM 包括两个步骤：设置相关的控制寄存器；执行 *PLS* 指令。

建议用户在工程中尽量编写单独的初始化子程序，这样可以使整个用户工程具有良好的结构。另外，若有可能的话，尽量在主程序中以 SM0.1 为条件来调用这个初始化子程序，这样该子程序将只在 CPU 上电后的首次扫描中调用并执行一次，可以减少 CPU 的扫描时间。

##### ➤ 使用 PWM

- 1) 根据期望的操作来设置控制字节 SMB67。

例如， $SMB67 = B\#16\#D3$  表明了：

- 允许 PTO/PWM 功能；
  - 选择使用 PWM 功能；
  - 选择使用同步更新方式；
  - 时基选择为 1 $\mu$ S；
  - 允许更新脉宽值和周期值。
- 2) 将期望的周期值赋给 SMW68。
  - 3) 将期望的脉宽值赋给 SMW70。
  - 4) 执行 *PLS* 指令来配置并启动 PWM0。

##### ➤ 改变脉宽

下面描述了如何改变 PWM0 的脉宽。

- 1) 根据期望的操作来设置控制字节 SMB67。

例如， $SMB67 = B\#16\#D2$  表明了：

- 允许 PTO/PWM 功能；
- 选择使用 PWM 功能；
- 选择使用同步更新方式；

- 时基选择为 1 $\mu$ S;
  - 允许更新脉宽值。
- 2) 将期望的脉宽值赋给 SMW70。
  - 3) 执行 PLS 指令来配置并启动 PWM0。

### 3.5.3 定位控制指令的使用

#### 3.5.3.1 如何修改定位控制指令的经过值

##### ➤ 控制寄存器和状态寄存器

针对定位控制指令，Kinco-K2 在 SM 区中为每路高速输出均分配了一个控制字节，在应用中用户需要注意设置该控制字节。另外，还分配了一个当前值（DINT 型）寄存器，用于存放当前已经输出的脉冲个数（正转时增加，反转时减少）。下表详细描述了这些寄存器。

Q0.0	Q0.1	Q0.4	Q0.5	描述
SMD212	SMD242	SMD262	SMD226	只读。当前值（正转时增加，反转时减少），表示当前已经输出的脉冲个数。
SMD208	SMD238	SDM258	SMD222	读写。 新当前值。与相应标志位配合，用于修改当前值。
Q0.0	Q0.1	Q0.4	Q0.5	描述
SM201.7	SM231.7	SM251.7	SM221.7	读写。急停标志位。若该位为 1，则表示处于急停状态，不执行任何定位控制指令。 当 PSTOP（急停）指令执行时，该位将自动置 1。用户需要使用程序将该位清 0。
SM201.6	SM231.6	SM251.6	SM221.6	读写。用于决定是否复位当前值 1 - 将当前值清零。 0 - 当前值保持不变。
SM201.5	SM231.5	SM251.5	SM221.5	保留
SM201.4	SM231.4	SM251.4	SM221.4	读写。用于决定是否修改当前值 1 - 将当前值清零。 0 - 当前值保持不变。
SM201.3	SM231.3	SM251.3	SM221.3	方向使能控制位。 • 禁止方向输出，方向通道作为普通 D0。 0 - 使能方向输出。

SM201.0	SM231.0	SM251.0	SM221.0	保留
~	~	~	~	
SM201.2	SM231.2	SM251.2	SM221.2	

### ➤ 如何修改当前值

3 路高速输出通道各有一个当前值寄存器，分别为 SMD212、SMD242 和 SMD262，其中存放着相应通道已经输出的脉冲个数。当前值寄存器是只读值，不允许在程序中直接进行修改。若需要修改当前值，则可以采取如下方法：

#### • 方法一

利用复位控制位来将当前值清除为 0。

3 个通道的复位控制位分别为 SM201.6、SM231.6 和 SM251.6。

4 个通道的复位控制位分别为 SM201.6、SM231.6、SM251.6 和 SM221.6。

只要复位控制位为 1，PLC 就会将相应的当前值寄存器清 0。因此复位控制位仅需要保持一个扫描周期即可发挥作用，使用时注意避免复位控制位长时间保持为 1，也尽量避免在运动过程中（包括 PHOME、PREL、PABS、PJOG、PFLO\_F 指令正在执行）来复位当前值，以免计数出现误差。

下面以通道 0 为例来说明如何复位当前值：

(\* Network 0 \*)

(\*以原点信号为基准，当运动到原点时，要求将当前值清 0.\*)

LD        %SM0.0

PHOME

0, %M0.0, %M0.1, %M0.2, %VW0, %VW2, %VW4, %VD6, %VW10, %M0.4, %M0.5, %MB1

(\* Network 1 \*)

(\*PHOME 指令完成后，利用 DONE 标志位将当前值清 0。\*)

LD        %M0.4

R\_TRIG

ST        %SM201.6

#### • 方法二

利用下述寄存器，可以将当前值设置为任意值。

Q0.0	Q0.1	Q0.4	Q0.5	描述
SMD208	SMD238	SDM258	SMD222	读写。新当前值。与相应标志位配合，用于修改当前值。



SM201.4	SM231.4	SM251.4	SM221.4	读写。用于决定是否修改当前值 1 - 将当前值清零。 0 - 当前值保持不变。
---------	---------	---------	---------	---

以通道 0 为例来说明使用方法：若 SM201.4 为 0，则保持当前值 SMD212 不变。若 SM201.4 为 1，则将 SMD208 中的值赋值给当前值 SMD212。注意尽量避免在运动过程中（包括 PHOME、PREL、PABS、PJOG、PFLO\_F 指令正在执行）来修改当前值寄存器，以免当前值计数出现误差。

下面的示例程序是以通道 0 为例，说明如何修改当前值：

(\* Network 0 \*)

(\*以原点信号为基准，当运动到原点时，要求将当前值设置为 100.\*)

LD        %SM0.0

PHOME

0, %M0.0, %M0.1, %M0.2, %VW0, %VW2, %VW4, %VD6, %VW10, %M0.4, %M0.5, %MB1

(\* Network 1 \*)

(\*PHOME 指令完成后，利用 DONE 标志位来修改当前值。\*)

LD        %M0.4

R\_TRIG

MOVE     DI#100, %SMD208

ST        %SM201.4

### 3.5.3.2 定位控制指令运行过程中是否可以改变最高输出频率？

PREL（相对运动）和 PABS（绝对运动）在脉冲输出过程中不会去改变最高输出频率。它们在启动时会读取当前最低频率、最高频率和加减速时间参数值，并根据这些值自动计算适当的加减速段数，然后启动脉冲输出。在脉冲输出过程中，PREL 和 PABS 不会再读取上述参数值，因此这些参数值的变化对于脉冲输出没有影响。

PJOG（点动）指令在执行过程中会实时读取输入频率参数（MAXF）值，并根据新的频率值来调整输出脉冲的频率。

PHOME（回原点）指令在运行到最高频率段之后、在回原点和原点信号产生之前，会实时读取最高频率参数（MAXF）值，并根据新的频率值自动计算加速或者减速段数，然后加速或者减速运行到新的频率值再维持匀速输出。