文件名称：IEC.16.Frame向导功能设计说明书

文件编号：A11-B13-000446

项目名称：PLC系列AutoThink软件开发

项目编号：A1030A-3

物料编码：

版 本 号：A

文件密级：秘密

文件状态：CFC

受控标识：受控

|  |  |
| --- | --- |
| 拟制： | 蓝海滨 2011 年 9 月 26 日 |
| 审核： | 黄宁 2011 年 10 月 8 日 |

会签：

批准： 李宗杰 2011 年 10 月 8 日

**文件发放范围：**

**修订页**

| 序号 | 版本号 | 修订内容简述 | 拟制/日期 | 审核 | 批准 |
| --- | --- | --- | --- | --- | --- |
|  | A | 创建 | 蓝海滨/2011-9-26 |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**目 录**

[1 引言 1](#_Toc307408038)

[1.1 目的 1](#_Toc307408039)

[1.2 文档概述 1](#_Toc307408040)

[1.3 引用文档 1](#_Toc307408041)

[2 指令向导概述 1](#_Toc307408042)

[3 串口通讯指令向导 2](#_Toc307408043)

[3.1 Modbus从站通讯 3](#_Toc307408044)

[3.2 Modbus主站通讯 4](#_Toc307408045)

[3.3 自由口通讯 8](#_Toc307408046)

[3.3.1 只发送数据 9](#_Toc307408047)

[3.3.2 只接收数据 10](#_Toc307408048)

[3.3.3 发送与接收数据 13](#_Toc307408049)

[4 PID指令向导 13](#_Toc307408050)

[5 高速计数指令(HSC)向导 16](#_Toc307408051)

[6 脉冲输出串指令(PTO)向导 22](#_Toc307408052)

[7 指令向导总体设计 24](#_Toc307408053)

[7.1 指令向导主要数据结构 24](#_Toc307408054)

[7.2 指令向导与LD语言编辑区的接口 25](#_Toc307408055)

# 引言

## 目的

本文档在AutoThink组态软件框架需求规格说明书的基础上，对LD语言中的串口通讯指令向导、PID指令向导、高速计数指令(HSC)向导及脉冲输出串指令(PTO)向导等4种指令向导进行详细的阐述，以明确指令向导功能的需求，并作为其后续设计、开发及测试的依据。

本文预期的读者为：用户代表、开发人员、测试人员及手册编写人员。

## 文档概述

本文档主要针对LD语言中的串口通讯指令、PID指令、高速计数指令(HSC)及脉冲输出串指令(PTO)等4种指令的向导功能进行详细的叙述，主要包括指令向导的操作步骤、向导各步的显示界面以及需要说明的相关问题。

## 引用文档

1. 《HOLLIAS-C.IEC.SRS.02.框架需求规格说明书》
2. 《HOLLIAS-C.IEC.SRS.06.LD编辑器需求规格说明书》
3. 《新一代小型PLC RTS指令库函数接口说明书》

# 指令向导概述

当用户在LD语言中使用串口通讯指令、PID指令、高速计数指令(HSC)及脉冲输出串指令(PTO)等4种指令时，可以快速、方便地按照指令向导的提示步骤配置指令的各个参数值。用户在配置指令各个参数的过程中，在不熟悉或不参考指令手册的情况下，即可准确地知道各个参数的含义以及参数值的配置类型或范围。

当LD语言编辑区处于激活状态时，AutoThink工具菜单下的“串口通讯指令向导”、“PID指令向导”、“高速计数指令(HSC)向导”及“脉冲输出串指令(PTO)向导”等4项均为可用状态，否则其为禁用状态。

当点击AutoThink工具菜单下的指令向导项时，将弹出相应的指令向导的配置界面，用户即可根据提示依次配置当前指令的各个参数，直至点击“完成”。最后在LD语言编辑区的选中网络节点后续添加一个新的网络节点，并在该新的网络节点中生成相应指令的实例(此实例已经配置完成所有的输入参数，如果配置的输入参数为变量则依次弹出变量自动声明对话框声明变量的类型)，同时弹出变量自动声明对话框以便用户设置指令的实例名称等信息。如图2-1所示，当前选中“0001”网络节点，则在其后续添加一个新的网络节点“0002”，并在其中生成了数据发送指令COMM\_SEND的一个实例(此实例的所有输入参数均已经完成配置)。

图2-1 通过指令向导生成的指令实例

关于新添加网络节点的几点说明：

1. 空网络节点的定义：所谓空网络节点是指不包含任何LD 元件或跳转标签，但可包含注释的网络节点；
2. 当前选中网络节点为空网络节点时，则将新生成的指令实例直接添加到此空网络节点即可，而不必在其后继生成一个新的空网络节点；
3. 当前选中网络节点的后继网络节点为空时，则将新生成的指令实例直接添加到其后继空网络节点即可，而不必在其后继生成一个新的空网络节点；
4. 当前没有选中任何网络节点时，则首先在LD语言编辑区中查找最后一个非空网络节点，如果该非空网络节点存在后继网络节点(后继网络节点必为空网络节点)，则将新生成的指令实例直接添加到其后继网络节点中；否则在该非空网络节点的后继添加一个新的空网络节点，然后将新生成的指令实例添加到其中。

以下各节将针对串口通讯指令向导、PID指令向导、高速计数指令(HSC)向导及脉冲输出串指令(PTO)向导等4种指令向导的参数含义、配置界面等进行详细的叙述。

# 串口通讯指令向导

串口通讯指令主要包括Modbus从站通讯、Modbus主站通讯及自由口通讯等3种类型的应用，因此其指令向导的参数含义、配置界面等将分别根据用户选择的应用类型而不同。其中串口通讯指令向导的通讯口类型选择和应用类型选择的显示界面如图3-1及图3-2所示。



图3-1 串口通讯指令向导通讯口类型选择



图3-2 串口通讯指令向导应用类型选择

以下各小节分别针对用户选择的应用类型进行相应指令的参数配置步骤。

## Modbus从站通讯

当用户选择Modbus从站通讯应用时，只需要提示用户到硬件配置管理窗口的通讯参数信息属性页中设置Modbus从站地址的相关属性即可，而并不需要在LD语言编辑区中生成具体的实例。其信息提示界面如图3-3所示。



图3-3 Modbus从站通讯指令的信息提示界面

## Modbus主站通讯

Modbus主站通讯指令(MODBUS\_MASTER)主要用于实现串口Modbus协议的主站数据读写操作，当用户选择该应用时，则需要分五步来完成其所有参数的配置。其所有输入参数见表3-1所示。

表3-1 Modbus主站通讯指令的输入参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **数据类型** | **功能描述** | **参数值说明** | **默认值** |
| EN\_R | BOOL | 使能 | 0：无效  1：上升沿使能，且高电平有效 | 0 |
| Port | BYTE | 通讯口选择 | 0：CPU本体的PS/2  1：CPU本体端子通讯口（14点不支持）  2：扩展功能板通讯口（14点不支持）  3：第一个通讯扩展模块通讯口  4~N：第二个到第N个通讯扩展模块通讯口 | 0 |
| Slave | BYTE | 从站地址 | Modbus 从站地址，1-247 | 1 |
| RW | BOOL | 读/写选择 | 0：读取数据  1：写入数据 | 0 |
| DataAddress | DWORD | 从站存放数据的地址 | 详见表3-2 | 1 |
| DataLength | BYTE | 数据长度 | 1-100．对于开入/开出为所需要传输的总比特数。对于  模入/模出为所要传输的总通道数 | 1 |
| TBL | WORD | 主站存放数据的首字地址 | 诸如200，表示存放地址为%MW400开始的一段空间。  如果是读指令，读回的数据值存放到这个数据区中；  例如3号从站3050地址的数据为1000，则%MW400  存放1000。如果是写指令，要写出的数据值放到这个  数据区中；例如向3号从站的3050地址写入500，则  %MW400存放500。 | 0 |
| Timeout | WORD | 超时时间设置(ms) | 最小时间单位为50ms | 0 |
| INT\_EN | BOOL | 中断使能 | 0：禁止中断  1：中断使能  通讯口发送信息完成和接收信息完成各产生一个中断。中断列表中的通信口中断0和通信口中断1。 | 0 |

表3-2 DataAddress参数详细说明

|  |  |
| --- | --- |
| **参数说明** | |
| 最高位 | 低五位 |
| 0：开出 | Modbus原始地址 |
| 1：开入 |
| 3：模入 |
| 4：模出 |
| 从站的地址采用标准Modbus RTU驱动的地址填写方式，即采用六位数（十进制），第六位只能选择0、1、3、4，分别表示Modbus 不同的数据类型，低五位表示寄存器地址（地址范围0-65535）。  **例如：000001，表示Modbus地址为0001的开出点**  **403050，表示Modbus地址为3050的模出点** | |

下面分六步来详细叙述Modbus主站通讯指令的参数配置情况。

第一步提示用户选择Modbus主站通讯指令的读写属性(RW)，如图3-4所示。



图3-4 Modbus主站通讯指令的读写属性

第二步提示用户配置从站的数据存放地址(DataAddress)，如图3-5所示，其中Modbus数据类型下拉框中将有4个选项，它们分别是“0:开出”、“1:开入”、“3:模入”、“4:模出”，默认选项为“0:开出”。



图3-5 Modbus主站通讯指令的从站数据存放地址

第三步提示用户配置Modbus从站地址(Slave)及发送或者接收数据的长度(DataLength)，其单位为BYTE，如图3-6所示。



图3-6 Modbus主站通讯指令发送或者接收数据的长度

第四步提示用户配置Modbus主站存放数据的首地址(TBL)，如图3-7所示。



图3-7 Modbus主站存放数据的首地址

第五步提示用户配置超时时间(Timeout)，如果选择“禁止超时”，则其超时时间默认配置为0ms，否则将其配置为大于或等于50ms的时间值，如图3-8所示。



图3-8 超时时间

第六步提示用户配置中断使能标识(Interrupt\_EN)，最终完成Modbus主站通讯指令的所有参数配置，如图3-9所示。



图3-9 中断使能属性

## 自由口通讯

自由口通讯主要包括只发送数据、只接收数据、发送与接收数据等三个方面的应用，如图3-10所示。当用户选择该指令类型时，将分三种情形进行参数的配置。



图3-10 自由口通讯指令三个方面的应用

如果应用中存在接收数据指令，则需要在生成的接收数据指令实例前添加一个置反的触点，其触点名称为“接收数据指令名称.Q”，如图3-11所示。

图3-11 接收数据指令实例前需添加一个置反的触点

### 只发送数据

自由口通讯指令只应用于发送数据时，总共可分为两步完成其参数的配置，发送数据指令(COMM\_SEND)所有的输入参数如表3-3所示。

表3-3 发送数据指令的输入参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **数据类型** | **功能描述** | **参数值说明** | **默认值** |
| EN\_R | BOOL | 使能 | 0：无效  1：上升沿使能，且高电平有效 | 0 |
| Port | BYTE | 通讯口选择 | 0：CPU本体的PS/2  1：CPU本体端子通讯口（14点不支持）  2：扩展功能板通讯口（14点不支持）  3：第一个通讯扩展模块通讯口  4~N：第二个到第N个通讯扩展模块通讯口 | 0 |
| DataLength | BYTE | 发送字节数 | 发送的字节数，1-255个字节。 | 255 |
| TBL | WORD | 数据存放地址 | 指向M区数据存放的首字节地址。诸如200，表示存放地址为%MB200开始的一段空间。 | 0 |
| INT\_EN | BOOL | 中断使能输入 | 0：发送完数据后不产生中断  1：发送完数据后产生中断  中断程序名称通过中断配置界面配置 | 0 |

第一步配置其发送数据的存放地址(TBL)，如图3-12所示。



图3-12 发送数据的存放地址

第二步提示用户配置发送数据的字节数(DataLength)及中断使能输入(Interrupt\_EN)属性，最终完成发送数据指令的所有参数配置，如图3-13所示。



图3-13 发送数据的字节数和中断使能输入属性

### 只接收数据

自由口通讯指令只用于接收数据应用时，总共可分为三步完成其所有参数的配置，接收数据指令(COMM\_RECEIVE)的所有输入参数如表3-4所示。

表3-4 接收数据指令的输入参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **数据类型** | **功能描述** | **参数值说明** | **默认值** |
| EN\_R | BOOL | 使能 | 0：无效  1：上升沿使能，且高电平有效 | 0 |
| Port | BYTE | 通讯口选择 | 0：CPU本体的PS/2  1：CPU本体端子通讯口（14点不支持）  2：扩展功能板通讯口（14点不支持）  3：第一个通讯扩展模块通讯口  4~N：第二个到第N个通讯扩展模块通讯口 | 0 |
| DataLength | BYTE | 接收字节数 | 接收的字节数，0-255个字节，设为0是不定长接收模式 | 255 |
| TBL | WORD | 数据存放地址 | 指向M区数据存放的首字节地址，诸如200，表示存放地址为%MB200开始的一段空间。 | 0 |
| Timeout | WORD | 帧超时时间(ms) | 从启动接收过程开始计算，在规定的时间内没有再接收到完整的数据包，中止接收过程。最小值为50ms，最小时间单位为系统调度时间。 | 0 |
| StartCharacter | WORD | 开始字符 | 当启用开始字符时，只有接收到开始字符才表征一帧数据的开始，否则被丢弃 | 0 |
| EndCharacter | WORD | 结束字符 | 当启用结束字符时，只有接收到结束字符才表征一帧数据的结束，否则继续接收直到缓冲区满（255字节） | 0 |
| Mode | BYTE | 控制模式参数设置 | 详见表3-5 | 0 |
| INT\_EN | BOOL | 中断使能输入 | 0：接收完数据后不产生中断  1：接收完数据后产生中断  中断程序名称通过中断配置界面配置 | 0 |

表3-5 Mode参数详细说明

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Mode（BYTE）** | | | | | | |
| 7 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 未定义 | | | 开始、结束字符数据长度控制 | 结束字符控制 | 开始字符控制 | 超时控制 |
| **超时控制** | | | | | | |
| 0：禁止超时控制  1：启动超时控制 | | | | | | |
| **开始字符控制** | | | | | | |
| 0：禁止开始字符控制  1：启动开始字符控制 | | | | | | |
| **结束字符控制** | | | | | | |
| 0：禁止结束字符控制  1：启动结束字符控制 | | | | | | |
| **开始、结束字符数据长度控制** | | | | | | |
| 0：开始、结束字符为字  1：开始、结束字符为字节 | | | | | | |

第一步提示用户配置接收数据的存放地址(TBL)，如图3-14所示。



图3-14 接收数据的存放地址

第二步提示用户设置接收数据指令的控制模式，如图3-15所示。



图3-15 接收数据指令的控制模式

第三步提示用户配置接收数据的字节数(DataLength)、帧超时时间(Timeout)、开始字符(StartCharacter)、结束字符(EndCharacter)及中断使能输入(Interrupt\_EN)属性；如果第二步中选择了“禁止超时控制”、“禁止开始字符控制”、“禁止结束字符控制”等三项，则在第三步中将“帧超时时间”、“开始字符”、“结束字符”等编辑区置为灰色禁用状态，其值均默认为0。通过该步最终完成接收数据指令的所有参数配置，如图3-16所示。



图3-16 接收数据的字节数、帧超时时间、开始字符、结束字符及中断使能输入

### 发送与接收数据

自由口通讯指令同时应用于发送与接收数据时，其参数的配置步骤由3.3.1小节和3.3.2小节所述的步骤结合完成。

# PID指令向导

PID指令向导主要用来帮助用户配置PID指令的输入参数，总共分五个步骤完成，最终在LD语言编辑区中自动生成一个PID实例(该实例的所有输入参数均已经配置完成)，其中PID指令的所有输入参数如表4-1所示。

表4-1 PID指令的输入参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **数据类型** | **功能描述** | **参数值说明** | **默认值** |
| EN\_R | BOOL | 使能 | FALSE：无效  TRUE：上升沿设定、高电平有效 | FALSE |
| MVManual | REAL | 手动输入值 |  | 0 |
| Auto | BOOL | “自动”模式选择 | FALSE：手动  TRUE：自动 | FALSE |
| SP | REAL | 设定值 |  | 0 |
| PV | REAL | 过程值 |  | 0 |
| DerectAction | BOOL | “正作用”方式选择 | FALSE：反作用（EV=SP-PV）  TRUE：正作用（EV=PV-SP） | FALSE |
| TS | REAL | 运算周期（S） | Ts>0 并且应与调用该功能块的任务的间隔时间一致 | 0.05 |
| KP | REAL | 比例增益 |  | 1 |
| TI | REAL | 积分时间（S） | Ti>=0,如果积分时间为0，表示没有积分环节 | 1 |
| TD | REAL | 微分时间（S） | Td>=0,如果微分时间为0，表示没有微分环节 | 0 |
| DeadBand | REAL | 偏差死区限 | 1）DeadBand>=0  2）若DeadBand=0，表示没有死区  3）偏差绝对值大于该值时有效，否则偏差为0 | 0 |
| MVBias | REAL | 前馈控制量 |  | 0.0 |
| MVMax | REAL | 输出值上限 |  | +100 |
| MVMin | REAL | 输出值下限 |  | -100 |

第一步提示用户选择PID指令的“自动”模式(Auto)，如果选择“手动”，则需要设置“手动输入值”(MVManual)，否则“手动输入值”为不可编辑状态，如图4-1所示。



图4-1 PID指令的“自动”模式选择

第二步提示用户配置PID指令的设定值(SP)、过程值(PV)以及选择“正作用”方式，如图4-2所示。



图4-2 PID指令的设定值、过程值及“正作用”方式选择

第三步提示用户配置PID指令的积分时间(TI)和微分时间(TD)，如图4-3所示。



图4-3 PID指令的积分时间和微分时间

第四步提示用户配置PID指令的运算周期(TS)和偏差死区限(DeadBand)，如图4-4所示。



图4-4 PID指令的运算周期和偏差死区限

最后一步提示用户配置PID指令的比例增益(KP)、前馈控制量(MVBias)、输出值上限(MVMax)及输出值下限(MVMin)，最终完成PID指令所有输入参数的配置，如图4-5所示。



图4-5 PID指令的比例增益、前馈控制量及输出值上下限

# 高速计数指令(HSC)向导

高速计数指令(HSC)向导将帮助用户配置高速计数指令(HSC)的输入参数，总共分五个步骤完成，最终在LD语言编辑区中自动生成一个高速计数指令(HSC)实例(该实例的所有输入参数均已经配置完成)。针对不同的CPU型号其指令参数有所不同，其中CPU型号为LE5104、LE5107(本节仅针对CPU型号为LE5104、LE5107的高速计数指令(PTO)向导)对应的高速计数指令(HSC)的所有输入参数分别如表5-1、表5-3所示。

表5-1 LE5104对应高速计数指令(HSC)的输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **功能描述** | **参数值说明** |
| EN\_R | BOOL | 使能 | 0→1 上升沿使能，电平保持 |
| CLR | BOOL | 清零 | 0→1：清零  （可以通过设定Mode位选择变量或绑定的外部输入点\*） |
| UpDN | BOOL | 计数方向控制 | 0：向上计数  1：向下计数  (单相计数时可以通过设定Mode位选择变量或绑定的外部输入点控制\*) |
| In\_Select | BYTE | 输入点选择 | =1：I0.0为脉冲输入端，I0.2为清零端（可选\*），I0.4为计数方向控制（可选\*）  =2：I0.1为脉冲输入端，I0.3为清零端（可选\*），I0.5为计数方向控制（可选\*） |
| Mode | BYTE | 模式选择 | 见表5-2 |
| PV | DINT | 设定计数值 | -2147483648~2147483647 |
| SV | DINT | 触发后CV改变值 | -2147483648~2147483647 |
| INT\_EN | BOOL | 中断使能 | 0：计数到达后不中断  1：计数到达后中断 |

表5-2 Mode数据选择定义

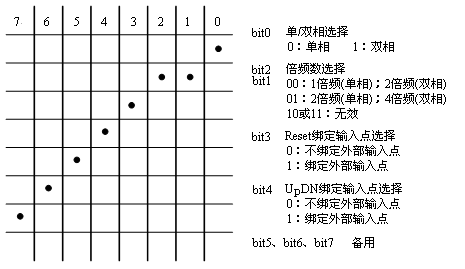


表5-3 LE5107对应高速计数指令(HSC)的输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **功能描述** | **参数值说明** |
| EN\_R | BOOL | 使能 | 0→1 上升沿使能，高电平保持 |
| CLR | BOOL | 清零 | 0→1：上升沿清零（可通过设定Mode参数绑定外部输入实现，绑定说明参阅表5-5） |
| UpDN | BOOL | 计数方向控制 | 可以选择变量或外部输入点(单相计数时) （绑定说明参阅表5-5）  0：向上计数  1：向下计数 |
| In\_Select | BYTE | 输入通道选择 | =1： 第1路单相计数 =2： 第2路单相计数 =3： 第3路单相计数 =4： 第4路单相计数 =21：第1路双相计数 =22：第2路双相计数 |
| Mode | BYTE | 模式选择 | 见表5-4 |
| PV | ARRAY[21]  OF DINT | 设定计数值 | -2147483648～2147483647  ARRAY[0]～ARRAY[20] |
| SV | DINT | 触发后CV改变值 | -2147483648～2147483647 |
| INT\_EN | BOOL | 中断使能 | 0：计数到达后不中断  1：计数到达后中断 |

表5-4 Mode数据选择定义



表5-5高速计数对应通道说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 高速计数通道 | 单相计数 | 对应方向通道  （单相计数且绑定时  有效） | 对应清零通道  （单相计数且绑定时  有效） | 双相  计数 |
| I0.0 | 第1路单相计数 | I0.4 | I0.6 | 第一路A相 |
| I0.1 | 第2路单相计数 | I0.5 | I0.7 | 第一路B相 |
| I0.2 | 第3路单相计数 | I1.0 | I1.2 | 第二路A相 |
| I0.3 | 第4路单相计数 | I1.1 | I1.3 | 第二路B相 |
| I1.4 | 第5路单相计数 | I2.0 | I2.4 | 第三路A相 |
| I1.5 | 第6路单相计数 | I2.1 | I2.5 | 第三路B相 |
| I1.6 | 第7路单相计数 | I2.2 | I2.6 | 第四路A相 |
| I1.7 | 第8路单相计数 | I2.3 | I2.7 | 第四路B相 |

第一步提示用户配置高速计数指令(HSC)的清零标识(CLR），如图5-1所示。



图5-1 清零标识

第二步提示用户选择高速计数指令(HSC)的模式(Mode)，模式数据选择定义根据CPU型号而定，其中CPU型号LE5104、LE5107选择高速计数指令(HSC)的模式分别如图5-2、图5-3所示。



图5-2 模式选择(CPU型号为LE5104)



图5-3 模式选择(CPU型号为LE5107)

第三步提示用户选择高速计数指令(HSC)的输入点(In\_Select)(当CPU型号为LE5104时)或输入通道(当CPU型号为LE5107时)，分别如图5-4、图5-5所示。



图5-4 输入点选择(CPU型号为LE5104)



图5-5 输入通道选择(CPU型号为LE5107)

第四步提示用户设定高速计数指令(HSC)的计数值(PV)及触发后CV的改变值(SV)，当CPU型号为LE5107时，则将“计数值(PV)”编辑区设置为禁用状态，并提示用户当在LD语言编辑区生成指令实例时在变量声明对话框中将PV的数据类型设置为“ARRAY[21] OF DINT”，如图5-6所示。



图5-6 计数值设定及触发后CV改变值

最后一步提示用户选择高速计数指令(HSC)的计数方向(UpDN)及中断使能(INT\_EN)属性，最终完成高速计数指令(HSC)所有输入参数的配置，如图5-7所示。



图5-7 计数方向选择与中断使能

# 脉冲输出串指令(PTO)向导

脉冲输出串指令(PTO)向导将帮助用户配置脉冲输出指令(PTO)的输入参数，总共分五个步骤完成，最终在LD语言编辑区中自动生成一个脉冲输出指令(PTO)实例(该实例的所有输入参数均已经配置完成)，其中CPU型号为LE5104(本节仅针对CPU型号为LE5104的脉冲输出串指令向导)的脉冲输出指令(PTO)的所有输入参数如表6-1所示。

表6-1 脉冲输出串指令(PTO)的输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **功能描述** | **参数值说明** |
| EN\_R | BOOL | 使能 | 0→1 上升沿使能，电平保持 |
| Axis | BYTE | 输出轴选择 | 1：轴1输出脉冲（包括方向轴）  2：轴2输出脉冲（包括方向轴） |
| Mode | BYTE | 动作模式 | 动作模式及输出逻辑关系  1：增量型PLS+SIGN (正向OFF/反向ON)  2：增量型PLS+SIGN (正向ON/反向OFF) |
| DIR\_CTRL | BOOL | 输出方向点是否关联的输入端配置 | 0：不关联  1：关联 |
| Duty | BOOL | 占空比 | 0：50%  1：25% |
| SEG\_Select | BYTE | 分段数选择 | SEG\_SELECT =0：10段  1：30段  2：60段  3：100段 |
| Target | DINT | 目标位置设定 | -2147483648~2147483647，最少脉冲数的绝对值不能小于20 |
| VMax | DWORD | 最高频率 | 10Hz~200,000Hz |
| ACCTime | WORD | 加减速时间 | 0~32767ms（0表示匀速） |
| INT\_EN | BOOL | 中断使能 | 0：脉冲输出完成后不产生中断  1：脉冲输出完成后产生中断 |

第一步提示用户选择脉冲输出串指令(PTO)的输出轴(Axis)与占空比(Duty)，如图6-1所示。



图6-1 输出轴与占空比

第二步提示用户选择脉冲输出串指令(PTO)的动作模式(Mode)及输出方向点与输入端配置的关联关系(DIR\_CTRL)，如图6-2所示。



图6-2 动作模式及输出方向点与输入端配置的关联关系

第三步提示用户选择脉冲输出串指令(PTO)的目标位置(Target)并选择分段数(SEG\_Select)，如图6-3所示。



图6-3 目标位置设定与分段数选择

最后一步提示用户配置脉冲输出串指令(PTO)的最高频率(VMax)、加减速时间(ACCTime)及中断使能(INT\_EN)属性，最终完成脉冲输出串指令(PTO)所有输入参数的配置，如图6-4所示。



图6-4 最高频率、加减速时间及中断使能

# 指令向导总体设计

## 指令向导主要数据结构

指令向导主要类之间的关系如图7-1所示，其中类CInstructionWizard作为各种指令向导类的基类，实现各种指令向导的公共功能，如存取指令的输入参数值。派生类CPIDInstructionWizard、CCommInstructionWizard、CHSCInstructionWizard、CPTOInstructionWizard等分别实现PID指令向导、串口通讯指令向导、高速计数指令(HSC)向导、脉冲输出串指令(PTO)指令向导功能。

类CWizardStepsContainer作为存取各种指令向导步的容器的基类，如果在某种指令向导中存在多种不同的跳转指令向导步，则需要将这些不同的跳转指令向导步进行分类，并分别存取在不同的指令向导步容器中，如串口通讯指令向导CCommInstructionWizard中可选择Modbus从站通讯、Modbus主站通讯、发送数据通讯、接收数据通讯等4种不同的跳转指令向导步，它们分别对应CModbusSecondaryContainer、CModbusMasterContainer、CSendDataContainer、CReceiveDataContainer等4种容器，在选择时可从相应的容器中动态添加这些指令向导步。



图7-1 指令向导主要类之间的关系

## 指令向导与LD语言编辑区的接口

类CInstructionWizardInfo存取各种通过指令向导配置完成的指令输入参数及指令向导的类型，在LD语言编辑区中根据当前指令的输入引脚名称从该结构中取得相应的值，并将其显示出来。

函数AddInstructionByWizard(CInstructionWizardInfo \* pInstructionInfo)作为指令向导与LD语言编辑区进行交互的接口。

—以下无正文