FBD功能库需求

功能表：(标准)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模块** | **名称** | **标识** | **编号** | **备注** |
| 比较 | 等于 | EQ | 100 | 比较所有输入是否相等，相等输出“1”，不等输出“0”。输入值得类型必须形同 |
|  | 大于或等于 | GE | 101 | 该功能检查连续输入的值是递减序列还是相等的。 |
|  | 大于 | GT | 102 | 该功能检查连续输入的值是否为递减序列。 |
|  | 小于或等于 | LE | 103 | 该功能检查连续输入的值是否为递减序列。 |
|  | 小于 | LT | 104 | 该功能检查连续输入的值是否为递增序列。 |
|  | 不等于 | NE | 105 | 该功能检查输入值是否不相等。 |
|  |  |  |  |  |
| 时间日期 | 将一段时间与日期相加 | ADD\_\*\*\*\_TIME | 200 | 函数用于向日期或时间中增加一段时间。 |
|  | 除法 | DIVTIME | 201 | 时间转换为秒后再除以一个整数 |
|  | 乘法 | MULTIME | 202 | 该功能将输入值相乘，并将结果分配给输出 |
|  | 时差 | SUB\_\*\*\*\_\*\*\* | 203 | 计算两个日期或时间之间的时间差 |
|  | 减法 | SUB\_\*\*\*\_TIME | 204 | 从日期中减去持续时间 |
|  |  |  |  |  |
| 逻辑 | AND功能 | AND | 300 | 该功能对输入处的位序列执行按位与运算，并将结果分配给输出 |
|  | 下降沿检测 | F\_TRIG | 301 | 此功能块用于检测下降沿 1 -> 0 |
|  | 下降沿检测 | FE | 302 | FE 功能用于检测与之关联的位从 1 到 0（下降沿）的跳变过程。 |
|  | 取反 | NOT | 303 | 该功能对输入位序列进行按位取反，并将结果分配给输出 |
|  | OR 功能 | OR | 304 | 该功能对输入处的位序列执行按位或运算，并在输出处返回结果。 |
|  | 检测上升沿 | R\_TRIG | 305 | 该功能块用于检测上升沿 0 -> 1 |
|  | 上升沿检测 | RE | 306 | RE 功能用于检测与之关联的位从 0 到 1（上升沿）的跳变过程。 |
|  | 置0 | RESET | 307 | RESET 功能用于将与之关联的位设置为 0 |
|  | 向左循环移位 | ROL | 308 | 此功能将 IN 输入处的位模式以循环方式向左移动 n 位 |
|  | 向右循环移位 | ROR | 309 | 此功能将 In 输入处的位模式以循环方式向右移动 n 位（输入 Number 处的值）。 |
|  | 双稳功能块，复位优先 | RS | 310 | 该功能块用作具有"复位优先"属性的 RS 存储器。 |
|  | 将位设置为 1 | SET | 311 | SET 功能用于将与之关联的位设置为 1。 |
|  | 向左移位 | SHL | 312 | 此功能将 IN 输入处的位模式左移 n 位。 |
|  | 向右位移 | SHR | 313 | 此功能将 IN 输入处的位模式右移 n 位（输入 N 处的值）。 |
|  | 双稳功能块，设置优先 | SR | 314 | 该功能块用作具有"设置优先"属性的 SR 存储器 |
|  | 检测所有跳变沿 | TRIGGER | 315 | 该功能块识别 CLK 输入中所有类型的跳变沿（1 -> 0 和 0 -> 1）。 |
|  | 异或功能 | XOR | 316 | 该功能对输入处的位序列执行按位异或运算，并在输出时返回结果 |
|  |  |  |  |  |
| 数学 | 绝对值计算 | ABS | 400 | 求绝对值 |
|  | 反余弦 | ACOS | 401 | ACOS 函数用于计算实数值的反余弦主值。其结果为以弧度表示的角度 |
|  | 加法 | ADD | 402 | 该功能将输入值相加，并将结果分配给输出 |
|  | 反正弦 | ASIN | 403 | ASIN 函数用于计算实数值的反正弦主值。其结果为以弧度表示的角度 |
|  | 反正切 | ATAN | 404 | ATAN 函数用于计算实数值的反正切主值。其结果为以弧度表示的角度 |
|  | 余弦 | COS | 405 | COS 函数用于计算角度的余弦值。 |
|  | 变量的递减 | DEC | 406 | DEC 功能将变量减小 1 |
|  | 除法 | DIV | 407 | 该功能将 Dividend 处的值除以 Divisor 输入处的值，并将结果分配给输出 |
|  | 除法和求模 | DIVMOD | 408 | 此过程将 Dividend 输入处的值除以 Divisor 输入处的值 |
|  | 自然指数 | EXP | 409 | EXP 函数用于计算实数值的自然指数。 |
|  | 幂 | EXPT\_REAL\_\*\*\*： | 410 | 用于计算一个值以另一个值为指数的幂 |
|  | 变量的递增 | INC | 411 | INC 功能将变量增加 1。 |
|  | 自然对数 | LN | 412 | LN 函数用于计算实数值的自然对数。 |
|  | 10 为底数的对数 | LOG | 413 | LOG 函数用于计算以 10 为底的实数值的对数。 |
|  | 模数 | MOD | 414 | 该功能将 Dividend 处的值除以 Divisor 输入处的值，并将模数分配给输出 |
|  | 赋值 | MOVE | 415 | 该功能将输入值分配给输出 |
|  | 乘法 | MUL | 416 | 该功能将输入值相乘，并将结果分配给输出。 |
|  | 取反 | NEG | 417 | 该功能对输入值取反 |
|  | 确定符号 | SIGN | 418 | 如果输入值 ≥ 0，则输出将变为"0"。如果输入值 < 0，则输出将变为"1"。 |
|  | 正弦 | SIN | 419 | SIN 函数用于计算角度的正弦值 |
|  | 减法 | SUB | 420 | 该功能从 Value1 输入处的值中减去 Value2 输入处的值，并将结果分配给输出 |
|  | 平方根 | SQRT\_\*\*\* | 421 | SQRT\_\*\*\* 函数用于计算变量的平方根。此函数也可使用其通用名称或下述的函数名称之一调用 |
|  | 正切 | TAN | 422 | TAN 函数用于计算角度的正切值 |
|  |  |  |  |  |
| 统计 | 求平均值 | AVE | 500 | 该过程计算加权输入值的平均值，并在输出处给出结果。 |
|  | 限制 | LIMIT | 501 | 如果输入值不小于最小值 (LowerLimit) 且不超过最大值 (UpperLimit)，则此功能将未更改的输入值 (Input) 传输给输出。如果输入值 (Input) 小于最小值 (LowerLimit)，则将最小值传输给输出。如果输入值 (Input) 超过了最大值 (UpperLimit)，则将最大值传输给输出。 |
|  | 带指示符的限制 | LIMIT\_IND | 502 | 如果输入值不小于最小值 (LimitMinimum) 且不超过最大值 (LimitMaximum)，则此过程将未更改的输入值 (Input) 传输给 (Output)。如果输入值 (Input) 小于最小值 (LimitMinimum)，则将最小值传输给输出。如果输入值 (Input) 超过了最大值 (LimitMaximum)，则将最大值传输给输出。  此外，如果低于最小值或超过最大值，则会指示说明。如果 (Input) 输入处的值小于 (LimitMinimum) 输入处的值，则 (MinimumViolation) 输出将变为"1"。如果 (Input) 输入处的值大于 (LimitMaximum) 输入处的值，则 (MaximumViolation) 输出将变为"1"。  （LimitMinimum、Input、LimitMaximum）输入值的数据类型必须与 (Output) 输出值的相同。 |
|  | 最大值功能 | MAX | 503 | 该功能将最大输入值分配给输出。 |
|  | 最小值功能 | MIN | 504 | 该功能将最小输入值分配给输出 |
|  | 多路复用器 | MUX | 505 | 此功能根据 K 输入的值，将相应的输入传输给输出 |
|  | 二进制选择 | SEL | 506 | 该功能用于两个输入值之间进行选择。 |
|  |  |  |  |  |
| 字符串 | 连接两个字符串 | CONCAT\_STR | 600 | CONCAT\_STR 功能用于连接两个字符串 |
|  | 删除子字符串 | DELETE\_INT | 601 | DELETE\_INT 功能用于从特定序号开始删除一定数量的字符。结果是产生另一个字符串。 |
|  | 比较两个字符串 | EQUAL\_STR | 602 | EQUAL\_STR 功能用于比较两个字符串。 |
|  | 查找子字符串 | FIND\_INT | 603 | FIND\_INT 功能用于搜索一个字符串在另一个字符串中的出现位置。 |
|  | 插入子字符串 | INSERT\_INT | 604 | INSERT\_INT 功能用于从给定序号开始将一个字符串插入另一个字符串。结果为字符串。 |
|  | 提取左侧的字符 | LEFT\_INT | 605 | LEFT\_INT 功能用于提取字符串最左侧特定数目的字符。结果是产生另一个字符串 |
|  | 字符串的长度 | LEN\_INT | 606 | LEN\_INT 功能用于计算字符串的字符数。 |
|  | 提取子字符串 | MID\_INT | 607 | MID\_INT 功能用于从特定序号开始提取子字符串。结果为字符串。 |
|  | 替换子字符串 | REPLACE\_INT | 608 | REPLACE\_INT 功能用于替换某个字符串中从特定序号开始的一个特定长度字符串。结果为字符串。 |
|  | 提取右侧的字符串 | RIGHT\_INT | 609 | RIGHT\_INT 功能用于提取字符串最右侧特定数目的字符。结果是产生另一个字符串 |
|  |  |  |  |  |
| 定时器计数器 | 减计数器 | CTD、CTD\_\*\*\* | 700 | 这些功能块用于递减计数。 LD 输入处的"1"信号导致将 PV 输入的值分配给 CV 输出。对于 CD 输入处每次从"0"到"1"的跳变，都会将 CV 的值减 1。 当 CV ≤ 0 时，Q 输出将变为"1"。 |
|  | 加计数器 | CTU、CTU\_\*\*\* | 701 | 这些功能块用于递增计数。 R 输入处的"1"信号会使值"0"赋予 CV 输出。对于 CU 输入处每次从"0"到"1"的跳变，都会将 CV 加 1。当 CV ≥ PV 时，Q 输出设置为"1"。 |
|  | 加/减计数器 | CTUD、CTUD\_\*\*\* | 702 | 这些功能块用于递增计数和递减计数。 R 输入处的"1"信号导致将值"0"赋予 CV 输出。LD 输入处的"1"信号导致将 PV 输入的值分配给 CV 输出。对于 CU 输入处每次从"0"到"1"的跳变，都将 CV 的值加 1。对于 CD 输入处从"0"到"1"的每次跳变，都将 CV 的值减 1。  如果在输入 R 和 LD 处同时存在"1"信号，则输入 R 优先。  当 CV ≥ PV 时，输出 QU 为"1"。  当 CV ≤ 0 时，QD 输出将变为"1"。 |
|  | 关闭延迟 | TOF | 703 | 该功能块用作关闭延迟。 首次调用该功能块时，ET 的初始状态为"0"。 可以将 EN 和 ENO 配置为附加参数。 |
|  | 接通延迟 | TON | 704 | 该功能块用作接通延迟。 首次调用该功能块时，ET 的初始状态为"0"。 可以将 EN 和 ENO 配置为附加参数。 |
|  | 脉冲 | TP | 705 | 该功能块用于生成一个具有已定义持续时间的脉冲。 首次调用该功能块时，ET 的初始状态为"0"。 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**应用程序结构**

1. **结构**

应用程序有四部分组成：主任务（MAST任务）、快速任务（FAST任务）、辅助任务、事件处理，这几个部分组成。

1. **任务之间优先级**

快速任务的优先级高于主任务，辅助任务为优先级最低的任务,事件处理的优先级高于其他所有的任务。

1. **主任务（MAST）**

主任务表示应用程序的主要任务。它是必需的，在缺省情况下创建。主任务 (MAST) 由段和子程序组成（。主任务的执行类型包括：循环（缺省选项）和或周期（1 至 255 毫秒）。

**3.1循环(缺省)**

循环执行将循环一个接一个串联在一起，各个循环之间没有等待时间。

**3.2周期**

在周期模式下，可以定义必须执行主任务的特定时间（周期）。如果执行时间短于这个时间，则在下一个循环之前将生成等待时间。如果执行时间超过这个时间，则控制系统将指示溢出。如果溢出太大，PLC 将停止。

1. **快速任务（FAST）**

快速任务用于持续时间较短和周期性处理的任务。快速任务用于持续时间较短和周期性处理的任务。快速任务是周期性执行的，周期由配置确定，介于 1 至 255 毫秒之间，不过，执行的程序必须较短，以避免低优先级任务溢出。

1. **辅助任务（AUX）**

辅助任务用于处理缓慢的任务，辅助任务 (AUX) 由段和子程序组成。辅助任务是周期性执行的，辅助任务是周期性执行的。

1. **事件处理**

事件处理用于缩短应用程序对以下事件的响应时间：来自输入/输出模块的事件、来自事件定时器的事件。这些处理任务的执行优先级高于所有其他任务。因此，它们适合处理与事件相关的需要很短响应时间的任务。事件处理任务是异步执行的。

1. **段**

段是自主编程实体，段按其在浏览器窗口（结构视图）中的编程顺序执行。执行条件可与主任务、快速任务和辅助任务中的一个或多个段相关联，但不能与事件处理任务中的段关联。

1. **子程序（SR段）**

子程序调用在段或其他子程序中执行，子程序不能调用自身（不递归）。

1. **保护模式**

保护模式包括，只读和不可读写。

如果段配置了保护（读或读/写），则在激活保护后，该段只读或不可读写状态。

如果段配置了保护但保护被禁用，则改段任可以读写。

1. **单任务**

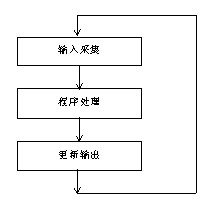


图1（单任务流程）

输入采集：将与任务相关的离散量和应用专用模块输入上的数据的状态写入存储器，这些值可通过强制设置值来修改。

程序处理：应用程序的执行，由用户写入。

更新输出：根据应用程序定义的状态，将输出位或字写入与任务相关的离散量或应用专用模块。

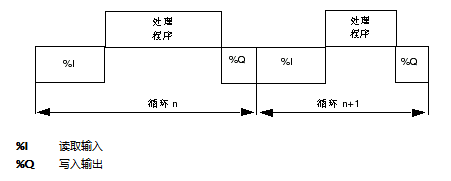


图2（循环执行）

此类操作由一个接一个的任务循环序列组成。更新输出之后，系统执行其特定处理，然后启动另一个任务循环，而不会暂停。

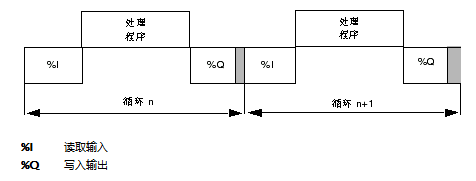


图3（周期执行）

在这种操作模式中，输入的采集、应用程序的处理和输出的更新都是周期执行的，定义的周期在 1 到 255 毫秒之间。 PLC 循环开始时，超时（其当前值初始化为定义的周期）开始倒计时。 PLC 循环必须在超时时间到期前完成并开始新循环。

如果周期尚未结束，处理器将通过执行内部处理完成其操作循环，直到周期束。

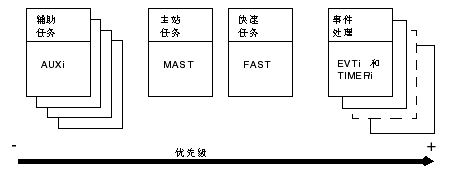
如果操作时间比分配给周期的时间长，周期溢出。然后继续并执行全部处理（但不能超过警戒时钟（自定义最大时间，如果警戒时钟溢出，则会声明应用程序出错，从而导致 PLC 立即停止）时间限制）。隐式写入当前循环的输出之后，开始下一次循环。

1. **多任务**

此类应用程序的任务结构如下：

|  |  |  |
| --- | --- | --- |
| **任务/处理** | **含义** | **说明** |
| 主站 | MAST | 始终存在，可以是循环或周期执行。 |
| 快速 | FAST | 可选，始终为周期执行。 |
| 辅助 | AUX 0 到 3 | 可选，始终为周期执行。 |
| 事件 | EVTi和TIMERi | 输入/输出模块发生事件时由系统调用，或由事件定时器触发。  这些类型的处理是可选的，可由需要在很短的响应时间内对输入/输出进行操作的应用程序使用。 |

下图显示了多任务结构中的任务及其优先级：



图四（优先级）

下表介绍优先任务的执行（此操作也在下面的图中说明）。

|  |  |
| --- | --- |
| **阶段** | **描述** |
| 1 | 事件发生或快速任务循环开始。 |
| 2 | 停止执行正在进行的较低优先级任务。 |
| 3 | 执行优先任务。 |
| 4 | 处理完优先任务后，继续执行被中断的任务。 |

下图介绍多任务处理的任务序列，该多任务处理具有一个循环主任务、一个周期为 20 毫秒的快速任务和事件处理。

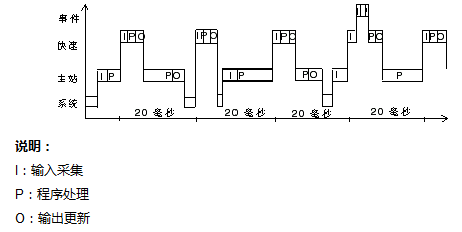


图5（执行顺序）

循环和周期操作

在多任务操作中，最高优先级任务将以周期模式使用，以便有足够的时间执行较低优先级任务。 因此，只有最低优先级的任务才应以循环模式使用。这样，如果主任务选择循环操作模式，则不能使用辅助任务。

1. **事件处理的管理**

下面的示意图介绍定义的 3 个优先级：

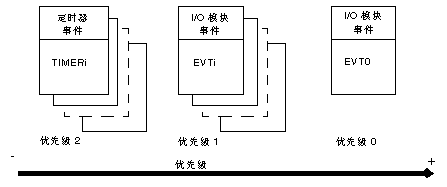


图6（优先级）

EVT0 事件处理是最高优先级处理。它可以自己中断其他类型的事件处理。

由输入/输出模块触发的 EVTi 事件处理（优先级 1）比由定时器触发的 TIMERi 事件处理（优先级 2）优先级高。

**1.比较功能：**

**功能1：EQ（等于）**

功能描述：该功能检查各个输入是否相等；即如果所有输入都相等，则输出变为"1"；否则输出仍然为"0"。所有输入值的数据类型必须是相同的。

公式： OUT = 1，如果 (IN1 = IN2) & (IN2 = IN3) & ..& (IN (n-1) = IN n)

**功能2：GE（大于等于）**

功能描述: 该功能检查连续输入的值是递减序列还是相等的, 所有输入值的数据类型必须相同。

公式: OUT = 1，如果 (IN1 ≥ IN2) & (IN2 ≥ IN3) & ..& (IN (n-1) ≥ IN n)

**功能3：GT（大于）**

功能描述: 该功能检查连续输入的值是否为递减序列, 所有输入值的数据类型必须相同。

公式: OUT = 1，如果 (IN1 > IN2) & (IN2 > IN3) & ..(IN (n-1) > IN n)

**功能4：LE(小于或等于)**

功能描述: 该功能检查连续输入的值是递增序列还是相等的, 所有输入值的数据类型必须相同。

OUT = 1，如果 (IN1 ≤ IN2) & (IN2 ≤ IN3) & ..& (IN (n-1) ≤ IN n)

**功能5：LT(小于)**

功能描述: 该功能检查连续输入的值是否为递增序列。

OUT = 1，如果 (IN1 < IN2) & (IN2 < IN3) & ..& (IN (n-1) < IN n)

**功能6：不等于**

功能描述: 该功能检查输入值是否不相等。

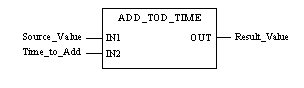
公式: OUT = 1，如果 IN1 < > IN2

**2.日期与时间功能：**

**功能1：将一段时间与日期相加（ADD\_\*\*\*\_TIME）**

功能描述：ADD\_\*\*\*\_TIME 函数用于向日期或时间中增加一段时间。

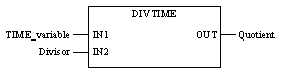
公式：ADD\_DT\_TIME（DateTime+Time）和ADD\_TOD\_TIME(TimeOfDay+Time)，



**功能2：除法（DIVTIME）**

功能描述：该功能将 TIME\_variable（数据类型为 TIME）输入处的值除以 Divisor 输入处的值，并将结果分配给输出。

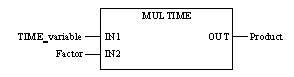
公式： OUT = （（ln1）/（ln2））



**功能3：乘法（MULTIME）**

功能描述：该功能将输入值相乘，并将结果分配给输出。

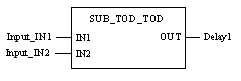
公式： OUT = IN1(TIME) x IN2(INT)



**功能4：计算两个日期或时间之间的时差（SUB\_\*\*\*\_\*\*\*）**

功能描述：SUB\_\*\*\*\_\*\*\* 函数用于计算两个日期或时间之间的时间差（绝对值）。

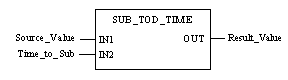
公式：SUB\_DATE\_DATE、SUB\_DT\_DT、SUB\_TOD\_TOD



**功能5：从日期中减去持续时间（SUB\_\*\*\*\_TIME）**

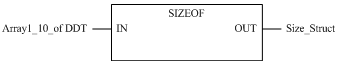
功能描述：SUB\_\*\*\*\_TIME 函数用于从日期或时间中减去一段时间。

公式：SUB\_DT\_TIME、SUB\_TOD\_TIME



**3.Extract功能：**

**功能1：SIZEOF 功能提供变量的存储器大小（以字节为单位）。**

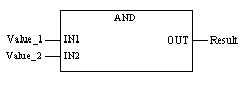


**4.逻辑功能：**

**功能1：AND 功能**

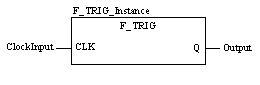
功能描述：该功能对输入处的位序列执行按位与运算，并将结果分配给输出。

公式： OUT = IN1 & IN2 & ...& INn



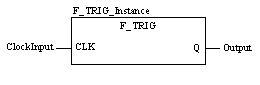
**功能2：F\_TRIG（下降沿检测）**

功能描述：此功能块用于检测下降沿 1 -> 0。如果在 CLK 输入处存在从"1"到"0"的跳变，则输出 Q 将变为"1"。从一个功能块执行到下一个功能块执行，输出将保持在"1"；输出随后返回为"0"。



**功能3：FE（下降沿检测）**

功能描述：FE 功能用于检测与之关联的位从 1 到 0（下降沿）的跳变过程。



**应用程序结构和执行**

**主任务：**

1. 主任务表示应用程序的主要任务，它是必须的，默认情况下创建。
2. 主任务由段（自主编程实体）和子程序组成
3. 主任务的执行类型可以为循环或周期

**快速任务：**

1. 快速任务由段和子程序组成
2. 快速任务是周期性执行的
3. 优先级高于主任务