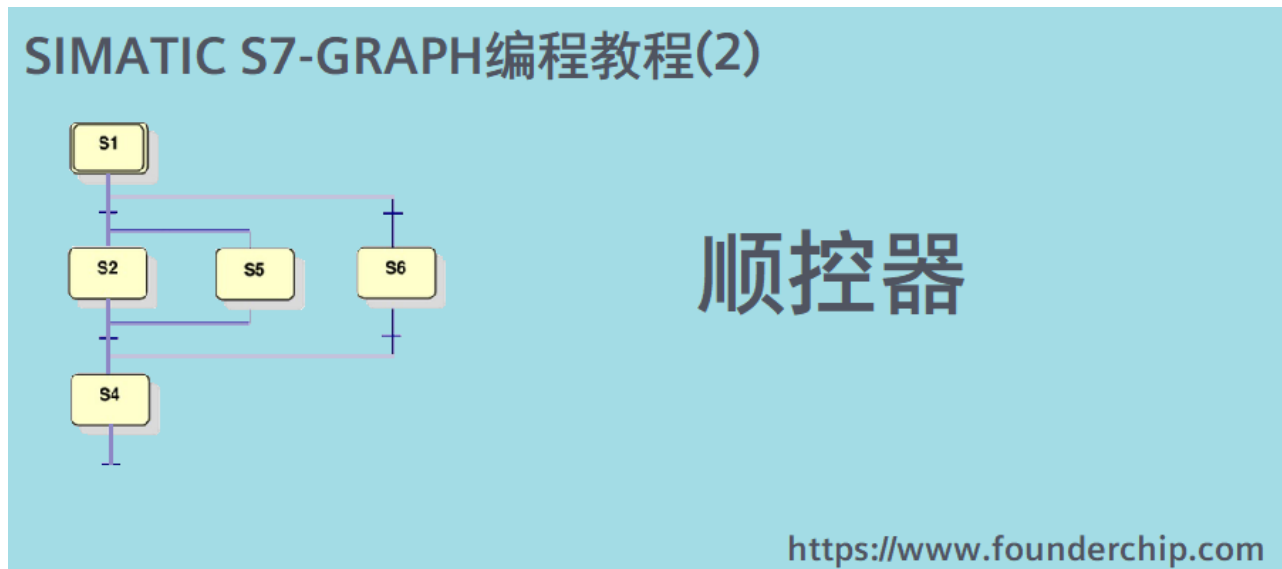


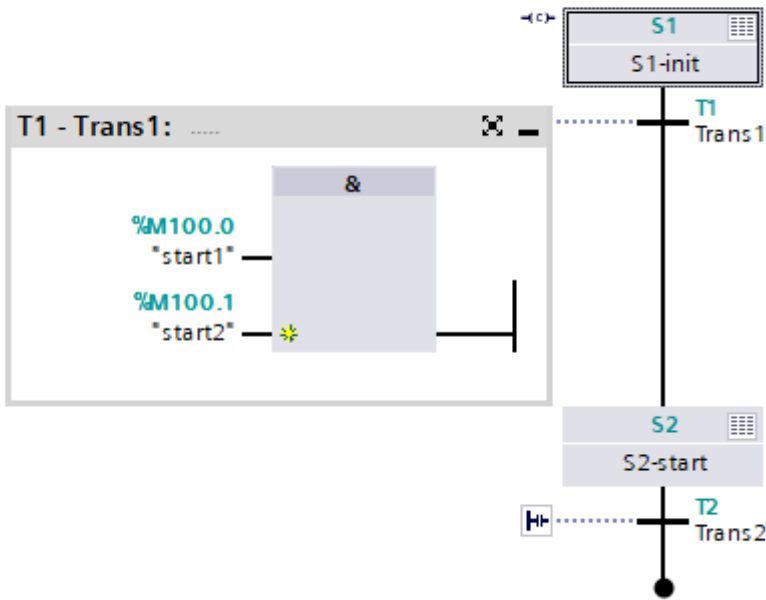
上一篇文章我们对S7-GRAPH语言有了基本的认识，今天这篇文章，我们来介绍下GRAPH语言中的重要内容——顺控器(sequencer)。



顺控器 (sequencer) 是由一系列的步 (step) 和跳转条件 (transition) 组成的代码。“步”是逻辑上人为划分的步骤，一个任务可以通过多个步来完成。“步”的概念可以参考我们日常生活中做事的步骤。比如那句经典台词“要把大象放冰箱，总共分几步？”你可以说需要三步或者其它你认为合理的答案，这取决于你的个人逻辑。GRAPH编程也是类似的，将某个任务划分成多少个步，取决于你的逻辑。步有两种状态：激活和未激活。顺控器当前正在执行的步处于激活状态，其它未执行的步处于未激活状态。在顺控器中，首次执行的步称为“初始步”。换句话说，在顺控器刚开始执行时，初始步处于激活状态，其它步处于非激活状态。步与步之间有跳转条件，当满足跳转条件 (transition) 时，从当前激活的步跳转到对应的步。此时，对应的步转变为激活状态，之前激活的步转变为非激活状态。比如下面图片的代码：

1: <新顺控器>

注释



最初步S1处于激活状态（初始步），当条件T1满足时，则跳转到步S2。此时步S2处于激活状态，而步S1则转变为非激活状态。

可以为每个步编辑相应的动作（action），也就是一些指令代码。当步处于激活状态时，其动作（指令）将被执行。比如下面图片的代码：

1: <新顺控器>

注释



当步S2处于激活状态时，它会执行两条指令：

- 1 N state
- 2 D startE, T#1s

①state是符号名，其地址为%M200.0。该指令（N）的意思是：只要当前步（S2）处于激活状态，则%M200.0的值就一直被置1。若当前步退出激活状态，则该值被置0；

②startE是符号名，其地址为%Q200.0。该指令（D）的意思是：当步（S2）处于激活状态后，延时1秒，将%Q200.0置1并保持；

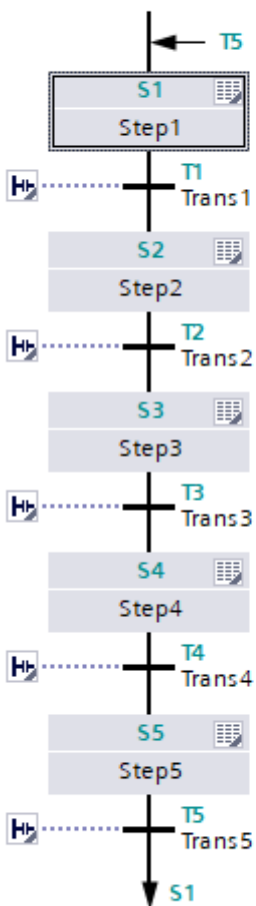
步的动作包括标准动作和依赖于事件的动作，内容较多，将在后续的文章中继续介绍。

至此，我们对顺控器已经有了初步的认识，下面来介绍下顺控器的结构。

顺控器有三种结构：线形结构、选择结构和并行结构。

①线形结构线形结构是最简单的顺控器结构，所有的步从上到下形成一条线（顺序执行），如下图所示：

1: <新顺控器>
注释

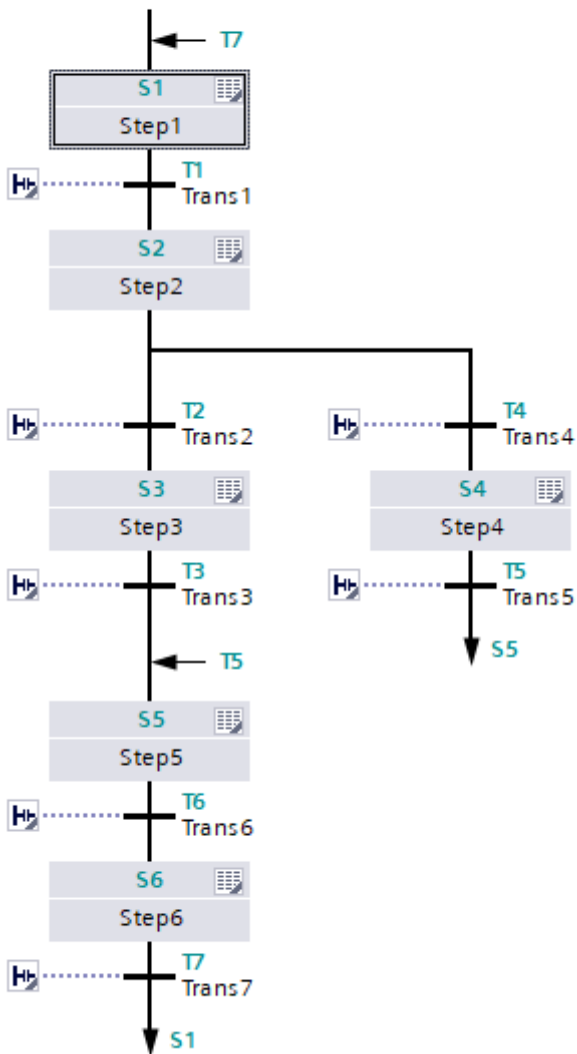


②选择结构（选择分支）

选择结构表达“或”逻辑关系，当条件满足时，顺控器会执行其中一条分支。比如下面的代码：

1: <新顺控器>

注释



当步S2激活后，若条件T2先满足，则跳转到步S3；若条件T4先满足，则跳转到步S4。

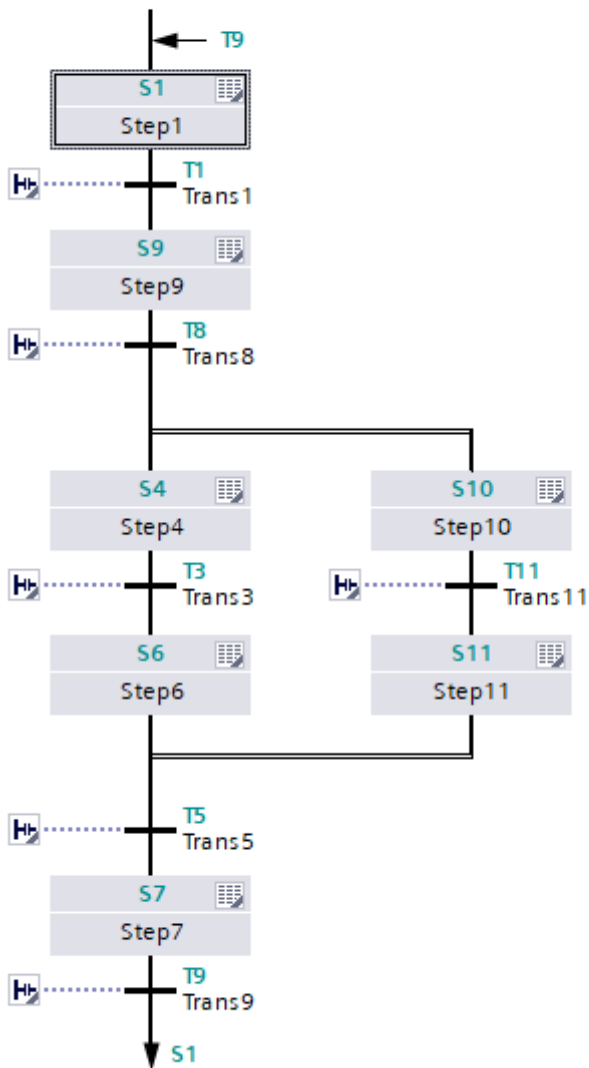
如果条件T2和T4同时满足了怎么办呢？

- 在自动或半自动模式下，左侧分支将会设置为较高优先级，因此会优先执行；
- 在手动模式下，完全取决于分支的优先级设置；

一个顺控器中最多可以有125个选择分支。

③并行结构（并行分支）

并行结构表达“与”逻辑关系，当条件满足时，顺控器会同时执行两个或多个分支。比如下面的代码：



当步S9处于激活状态时，若跳转条件T8满足，则两个并行分支同时执行，步S4和步S10同时处于激活状态。

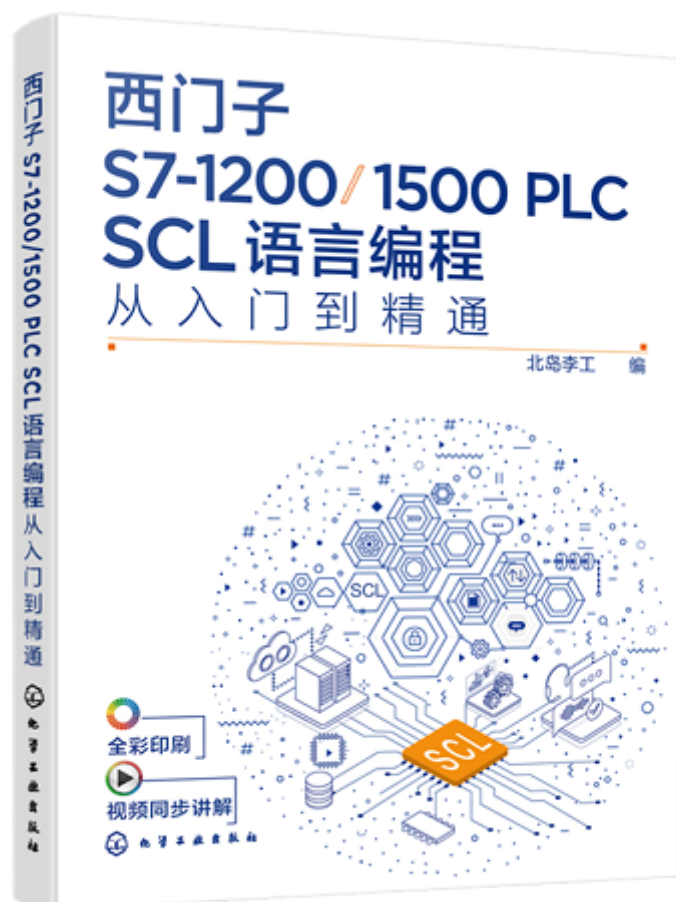
对于上图中，两个并行分支最后合并到步S7。只有当两个并行分支都执行完毕后，满足跳转条件T5，才能跳转到步S7。

注：并行分支不一定非要合并，也可以跳转到其它不同的步上。

从代码外观上看，并行分支是双线，而选择分支是单线，要注意区分。

好了，关于GRAPH语言教程的第二篇文章就先介绍到这里。

我的书《西门子S7-1200/1500 PLC SCL语言编程——从入门到精通》从硬件到软件，比较详细的介绍了SCL语言的编程，感兴趣的话可以扫描下面的二维码查看：



识别图中小
程序码购买