# PLC Lab 4: Programming in Sequential Function Chart (SFC)

## Objectives

* Understand challenges and common mistakes made in programming sequential tasks in ladder programming
* Identify the strength of using SFC for sequential tasks
* Programming SFC directly in TIA Portal environment
* Programming SFC applications involving Timers and Counters

## Tasks

* Task 1: SFC Fundamentals
* Task 2: Identify common mistakes in programming sequential tasks
* Task 3: Programming SFC using ladder in a structured manner
* Task 4: Creating a GRAPH program in TIA Portal
* Task 5: Using Timers in GRAPH
* Task 6: Using Counters in GRAPH
* Task 7: Application using GRAPH with alternative branch (DIY)
* Task 8: Application using GRAPH – Mixing Process (DIY)

## Synopsis

The IEC 61131-3 programming language, Sequential Function Chart (SFC) is a “flow chart like” programming language that is allows ease of programming and tracking sequential task. It can be used as a planning tool, and can also be directly programmed in most modern PLC software environment.

## Equipment Required

* PLC training kit with control panel.
* Laptop/PC with internet access
* Laptop/PC installed with TIA Portal V15.1 and PLCSIM

## Reference and Self Study materials

* Lab 4 Videos Playlist   
  <https://www.youtube.com/playlist?list=PLo5IISMe0m5PYBIVYUvD2XwPnXPeSPs1s>
* TIA Portal help files (Access from TIA Portal)

## Task 1: SFC Fundamentals

1. You have come across SFC in the previous labs.

In Lab1, Task 13: you use a short 3-step SFC to program the conveyor kit. You learn to translate SFC sequences into Ladder Logic.

In Lab3: Task 6, it’s a 4-step SFC with counter involved. You programed the conveyor drink can packaging system.

Here, we shall do a quick revision. In SFC, a sequence is made up of “Steps”. Each step will carry out specific “Actions”. In order to move the next step, “Transition” conditions must be fulfilled. An example as follows. Fill in the blanks.

SensorA

INIT

S1

S2

Turn on Motor1

STARTB

S3

Turn On Motor2 and Motor1  
M

SensorB

S4

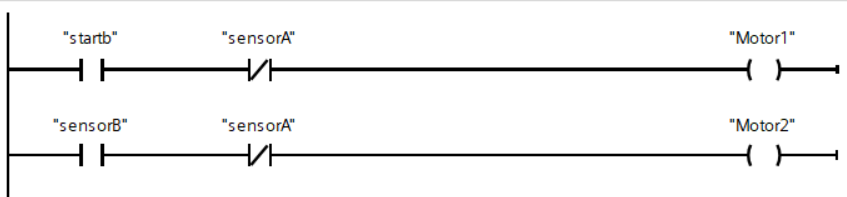
Turn off both Motors

‘1’ or TRUE

1. Important note

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Some of the demo videos are using %M for I/O as it can easily be modified in PLCSIM. However if you are using the physical PLC training kits in the lab, you should use the physical button and LEDs. Define as follows:   |  |  |  | | --- | --- | --- | | Tag | Type | Address | | StartB | Input | %I126.7 | | SensorA | Input | %I126.4 | | SensorB | Input | %I126.3 | | Motor1 | Output | %Q128.6 | | Motor2 | Output | %Q128.5 |   If you are using PLCSIM and wish to use %I instead of %M, it is also possible. Watch to learn the 2 methods to trigger %I using PLCSIM. <https://youtu.be/_081Qii0WyQ> |

## Task 2: Identify common mistakes in programming sequential tasks



1. Test the ladder above. Does it fulfil the SFC in Task 1?
2. Off all the inputs and put the system back to the first Step. Trigger Sensor B. Record your observations.

|  |
| --- |
|  |

1. Is this supposed to happen according to the SFC? If not, explain what should have happen

|  |
| --- |
|  |
|  |
|  |



If you need help to understand the previous question, watch this video <https://youtu.be/dUXua2DnHSU>

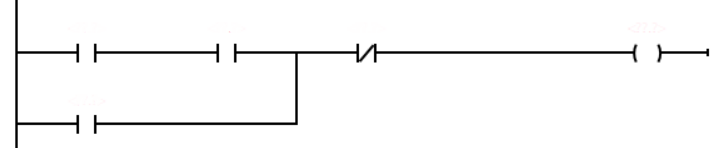
1. It is possible to make minor changes to the program above to make it work properly. However, for more complex sequences, programming without proper structure is not advisable as it’s prone to error and difficult to troubleshoot.

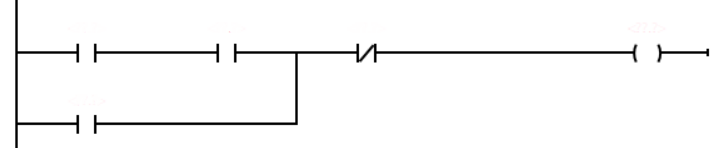
## Task 3: Programming SFC using ladder in a structured manner

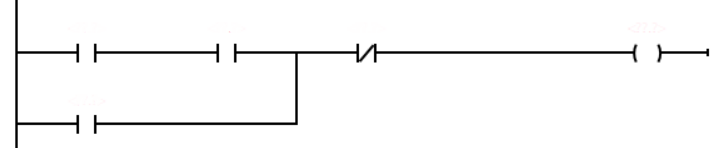


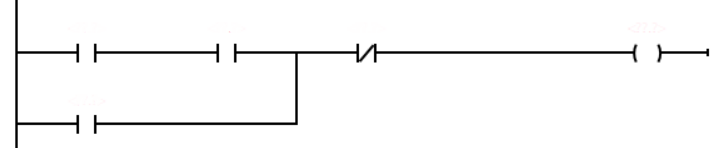
1. It is relatively easy to use a standard structure for sequential task such as Task 1 SFC.   
   Fill the blanks in the following template. Delete or add contacts as you deem necessary.  
   (If you need help, watch this video from Lab 3 Task 6: <https://youtu.be/uwuDephOiOs>)



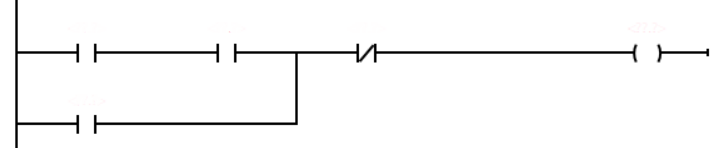




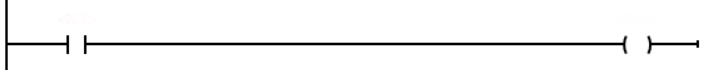
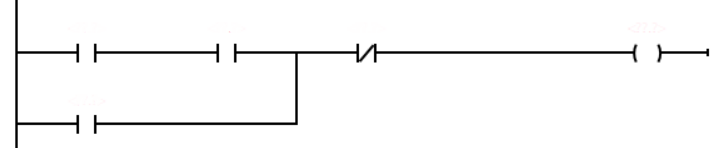


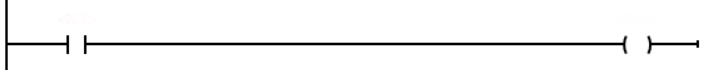












## Task 4: Creating a GRAPH program in TIA Portal

1. In TIA Portal, SFC is called GRAPH programming

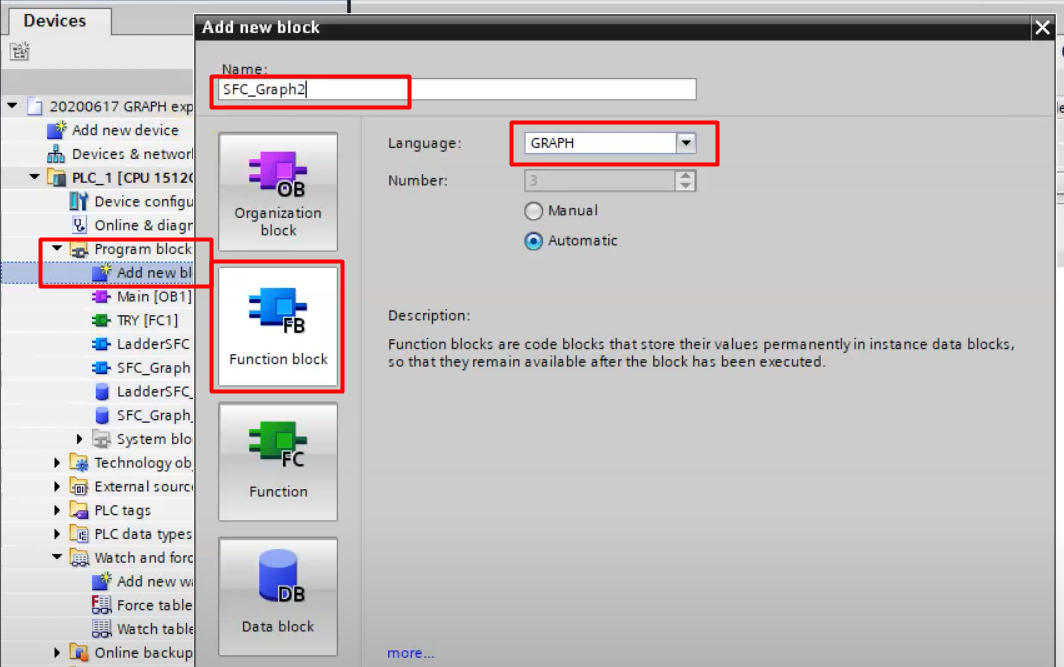




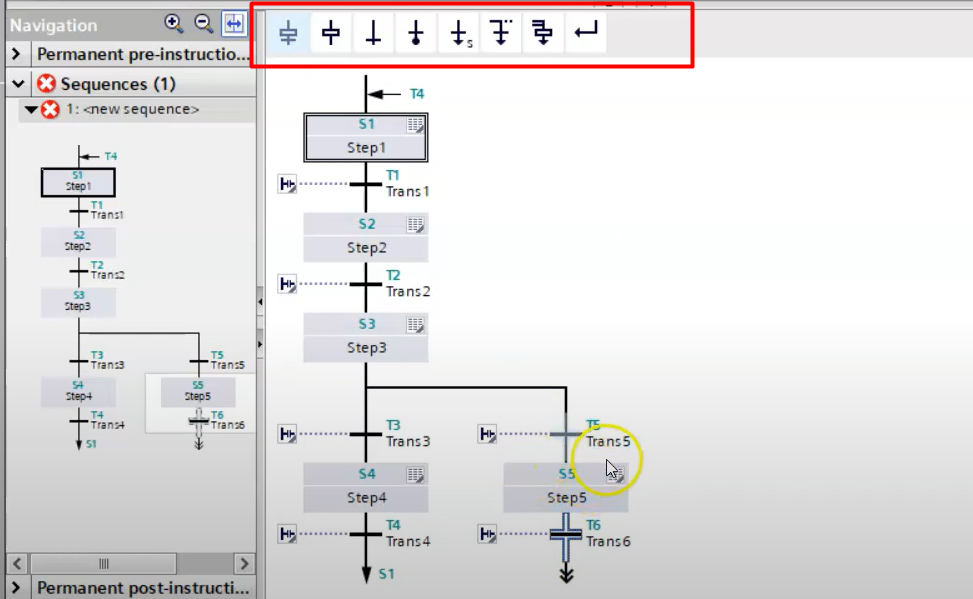
Watch this video to see how GRAHP works in TIA Portal <https://youtu.be/IWHR_T1_Tnw>

Watch this video to see how to create a GRAPH program <https://youtu.be/IBZM3ge1uTc>

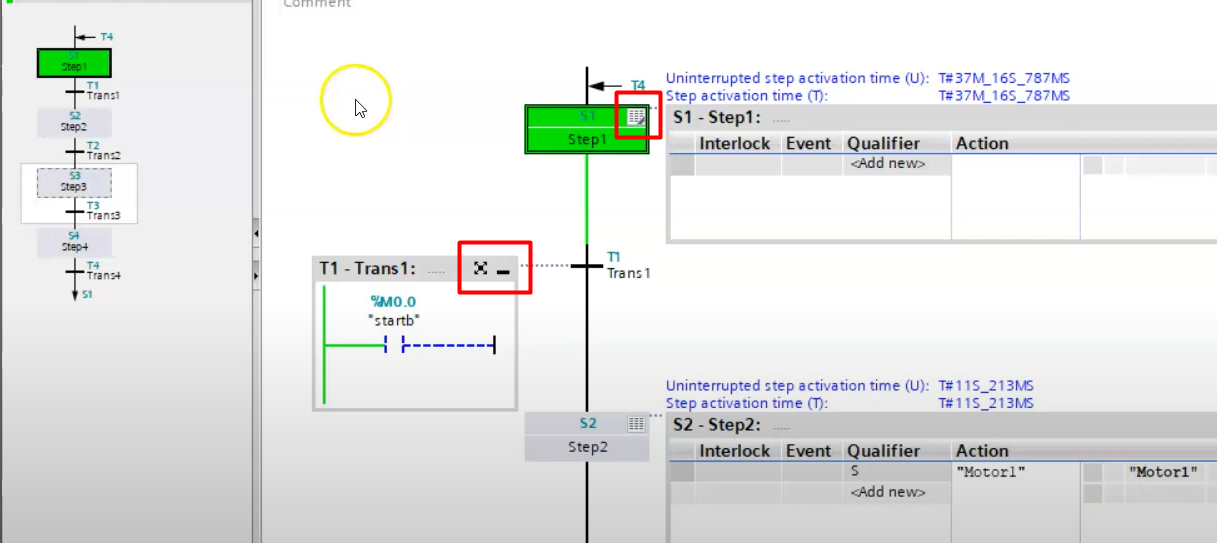
Only **Function Block (FB)** is able to use GRAPH programming language



Experiment with the instructions in the GRAPH programming environment



Expand the Step and Transition to program them.



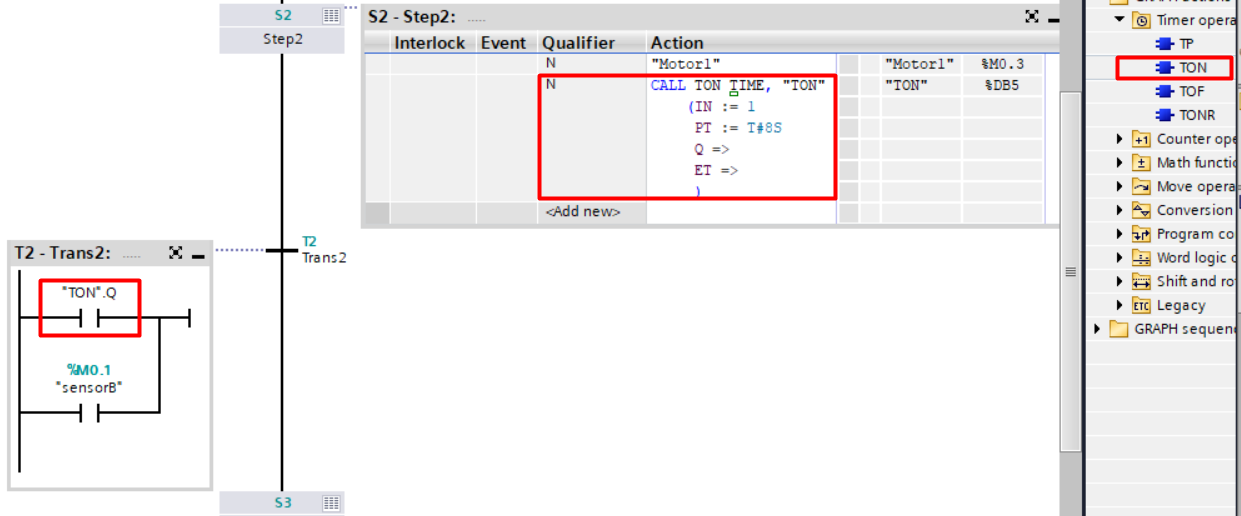
1. Create the GRAPH program to solve TASK 1.
2. Out of the many Qualifiers, there are three important ones, N, S, R.   
   State what each one of them does.

|  |  |
| --- | --- |
| Qualifier | Meaning |
| N |  |
| S |  |
| R |  |

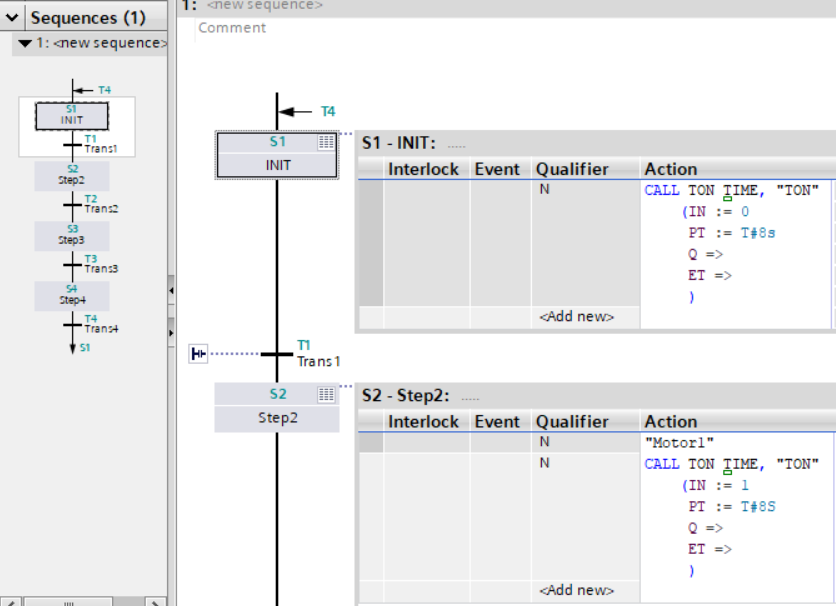
## Task 5: Using Timers in GRAPH

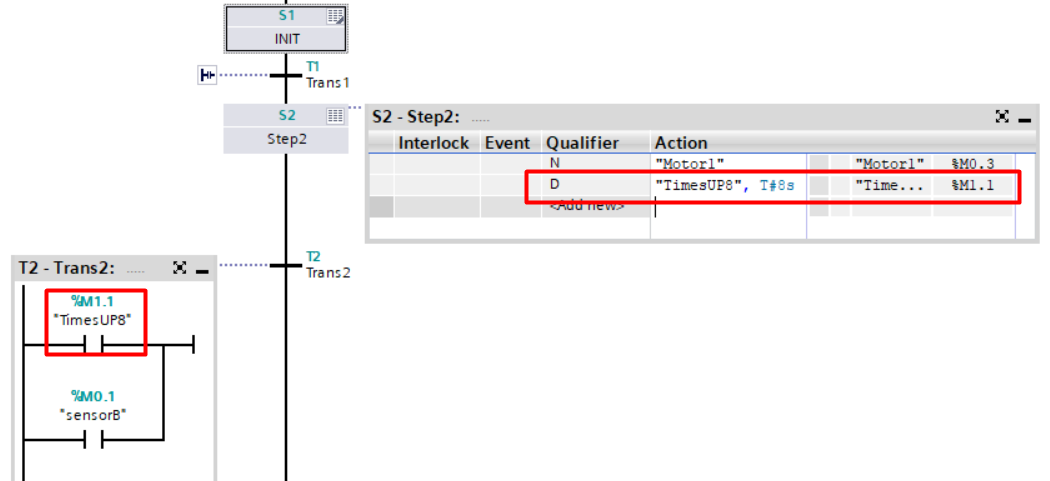
1. Watch this video to learn about using timers in GRAPH. <https://youtu.be/uVnXpm9Wies>

Click on the Step you wish to work on,   
drag the TON function into the Action.   
Initialize as you would normally use a timer.   
Use the Q contact to trigger the Transition



When using TON, we need to reset it, else TON.Q will only work once and will remain permanently on. To do this we can add a TON with the same timer name, and set IN = 0, either in the step immediately following the timer, or in the initial step as follows:



1. There are two ways to create timed steps in GRAPH. One is to use basic timer instructions   
   such as TON (as above), the other is to use the Qualifier D as below  
     
   
2. Follow the instruction in the video or otherwise, program the following. Test that it works

SensorA

INIT

S1

S2

Turn on Motor1

STARTB

S3

Turn On Motor2/Motor1  
M

**8 seconds** or **SensorB**

S4

Turn off both Motors

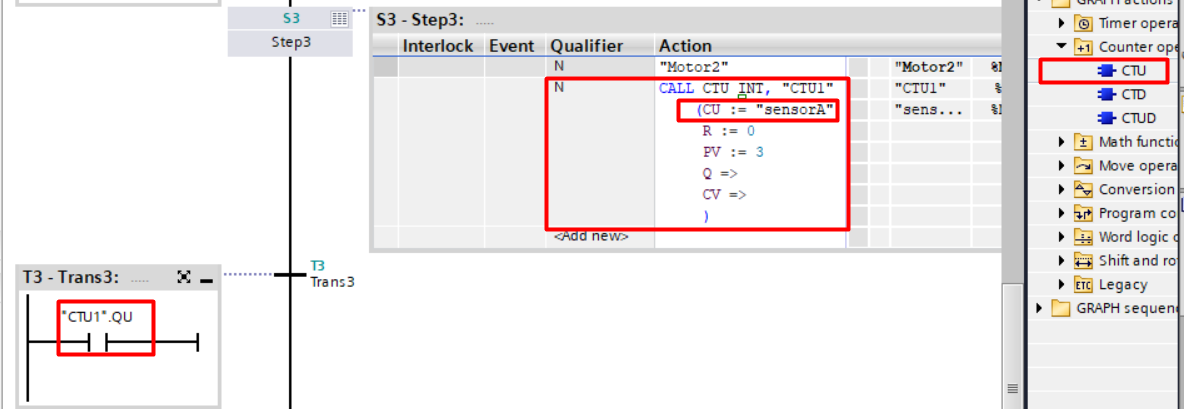
‘1’ or TRUE

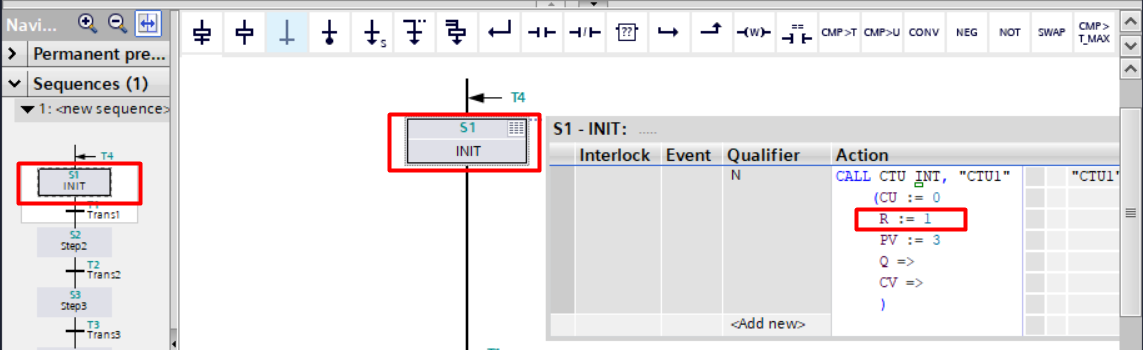
## Task 6: Using Counters in GRAPH

1. Watch this video to learn about using counters in GRAPH. <https://youtu.be/2rvpk0LBOjk>

Its very similar to using timer instructions

Click on the Step you wish to work on,   
drag the CTU function into the Action.   
Initialize as you would normally use a Counter.   
Use the Q contact to trigger the Transition



You would have notice that in the example above, we left the RESET input of the counter unconnected. The reason is that once the counter turns on, it will “leave” this Step. Thus it would be impossible to reset the counter in this Step. We could thus choose to reset it in appropriate steps, for example in the Step immediately following the current Step, or even in the Initial Step as follows  


1. Follow the instruction in the video or otherwise, program the following. Test that it works

SensorA **count 3 times**

INIT

S1

S2

Turn on Motor1

STARTB

S3

Turn On Motor2/Motor1  
M

**8 seconds** or **SensorB**

S4

Turn off both Motors

‘1’ or TRUE

## Task 7: Application using GRAPH with alternative branch (DIY)

Program the following SFC.

SensorB

INIT

S1

S2

Latch “Green\_Lamp” **%Q128.4**

STARTB

S3

On Motor1

SensorA

S4

Off both Motors, Reset Green\_Lamp

**‘1’**

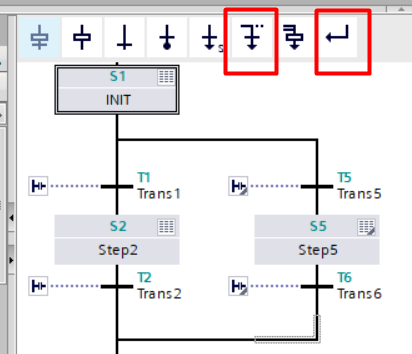
SensorA

S5

On Motor2

10 seconds

Make use of the branch instructions



## Task 8: Application using GRAPH – Mixing Process (DIY)

1. The figure below illustrates a system to be controlled by a PLC. A mixing process which specifies a final mix of 2 volumes of solvent to 1 volume of dyestuff is to be produced by operating V1 for twice the time duration of valve V2, thereby giving twice as much volume of solvent as for dyestuff. V2 is to be turned on for 5 s. After the solvent and dyestuff have been poured in, a stirrer motor M1 is activated for 7 s. V3, V4 and M2 is then operated to transfer the mixture to a heat exchanger for heating. The pump motor, M2 switches off when the low level detector, LD1 is triggered. V3 and V4 remains ON for a further 20s.

SOL 02

SOL 03

SOL 04

M1

**Solvent**

**Dyestuff**

pump

M2

Heat

Exchanger

V1

V2

V3

V4

Mixing Tank

LD1

SOL 01

Steam

Storage

Tank

LS1 High Limit

Sensor

Draw the SFC below and program it into the GRAPH in TIA Portal. Fill in the Tag table below with the appropriate Address. Test that the program functions accordingly.

|  |  |  |
| --- | --- | --- |
| Tag | Type | Address |
| StartB | Input |  |
| V1\_valve | Output |  |
| V2\_valve | Output |  |
| V3\_valve | Output |  |
| M1\_stirrer | Output |  |
| M2\_pump | Output |  |
| LD1\_low | Input |  |
| LS1\_high | Input |  |

INIT

S1

S2

STARTB

S3

S4

S5

S6