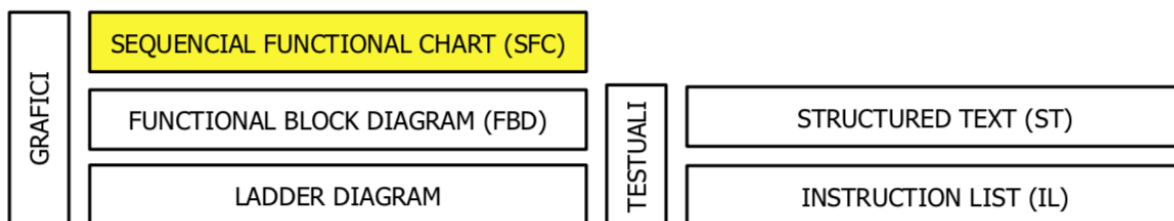




# SEQUENTIAL FUNCTIONAL CHART SFC

Ad oggi il linguaggio grafico SFC, rappresenta lo **STANDARD DE FACTO** per la PROGETTAZIONE e la DOCUMENTAZIONE dei PLC.

La Norma IEC 61131-3 definisce 5 linguaggi per i PLC:



- Il diagramma funzionale sequenziale o sequential function chart (SFC) è un [linguaggio di programmazione](#) grafico per i [PLC](#) introdotto e definito dallo standard internazionale [IEC 61131-3](#).
- Questo linguaggio si basa su Grafcet che a sua volta eredita le principali caratteristiche dal modello matematico di [Rete di Petri](#).
- L'SFC è utile anche come strumento di modellazione e analisi dell'evoluzione sequenziale di un sistema di automazione poiché permette di decomporre il ciclo macchina in una serie di azioni di controllo sequenziali.



# ELEMENTI BASE DELL'SFC

Gli elementi grafici di base che compongono l'SFC sono i seguenti:

- **Fase (Step o Passo):** rappresenta un singolo passo della sequenza di operazioni, è convenzionalmente indicata con un quadrato contenente un numero univoco che la identifica. Ad ogni fase possono essere associate delle azioni rappresentate da un rettangolo collegato alla fase contenente l'azione da eseguire. Ad ogni fase è associata una variabile booleana che ne rappresenta lo stato. Ogni fase può trovarsi in due soli stati: fase attiva o fase non attiva. Quando una fase è attiva vengono immediatamente eseguite tutte le azioni ad essa associate (ciascuna secondo le proprie caratteristiche temporali descritte da un qualificatore dell'azione).
- **Transizione:** riassume le condizioni per il passaggio da una fase ad un'altra, viene convenzionalmente indicata con una barretta orizzontale a cui è associato un identificatore univoco (T1, T2, T3...). Una transizione si dice attiva quando tutte le condizioni ad essa associate sono verificate nello stesso momento. Una transizione si dice superabile quando è attiva e tutte le fasi a monte sono contemporaneamente attive.
- **Arco orientato:** segmento con snodi ad angolo retto che collega fasi e transizioni.

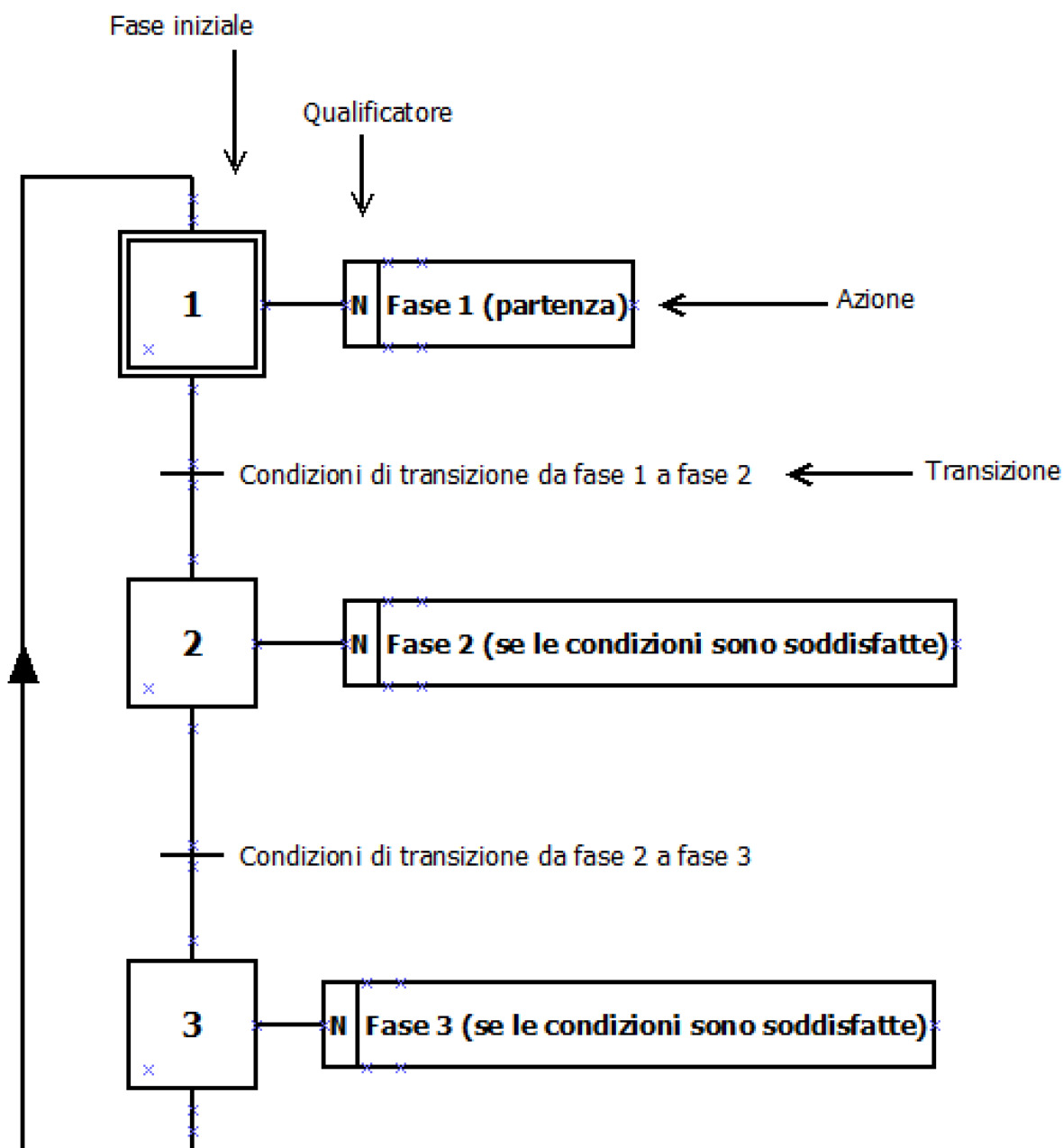


## Regole fondamentali per la costruzione grafica dell'SFC:

- tra due fasi è sempre presente almeno una transizione
- tra due transizioni è presente almeno una fase
- due fasi non possono essere collegate direttamente tra loro
- due transizioni non possono essere collegate direttamente tra loro
- quando una transizione è superabile, essa viene effettivamente superata (tutte le fasi a monte vengono disattivate e tutte le fasi a valle vengono attivate)
- se più transizioni sono superabili in un certo istante, vengono superate tutte contemporaneamente (modello sincrono)
- se la condizione associata alla transizione di uscita di una fase è già vera quando la fase viene attivata (fase instabile), le azioni ad essa associate **vanno comunque eseguite** prima della disattivazione della fase



## Struttura del linguaggio SFC e interpretazione:





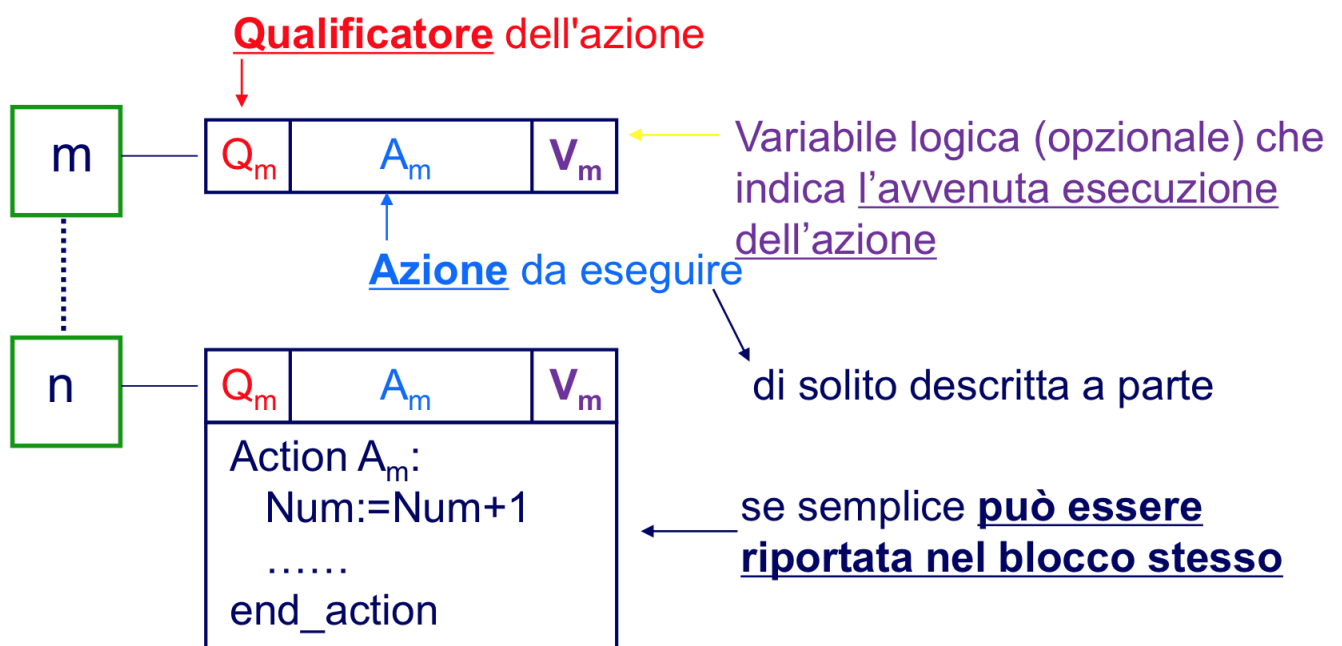
# AZIONI E QUALIFICATORI

Ad ogni azione è associato un qualificatore che ne specifica le proprietà, i **principali** qualificatori sono:

- **N** (non-stored, azione continua); L'azione inizia quando la fase è attiva e si disattiva quando la fase è disattivata
- **L** time-limited, azione limitata nel tempo (L'azione viene iniziata quando è attivato il passo. Continua finché il passo viene disattivato o quando è trascorso il tempo impostato)
- **D** delayed, ritardo di tempo (Quando il passo viene attivato parte un timer di ritardo. Se il passo è ancora attivo dopo il ritardo di tempo, l'azione inizia e continua finché non sarà disattivata)
- **P** pulse, azione impulsiva (eseguita una volta)
- **S** set, azione impostata
- **R** reset, azione arrestata
- **SD** stored and time-delayed, azione memorizzata e ritardata
- **DS** delayed and stored, azione ritardata e memorizzata
- **SL** stored and time-limited, azione memorizzata e limitata nel tempo

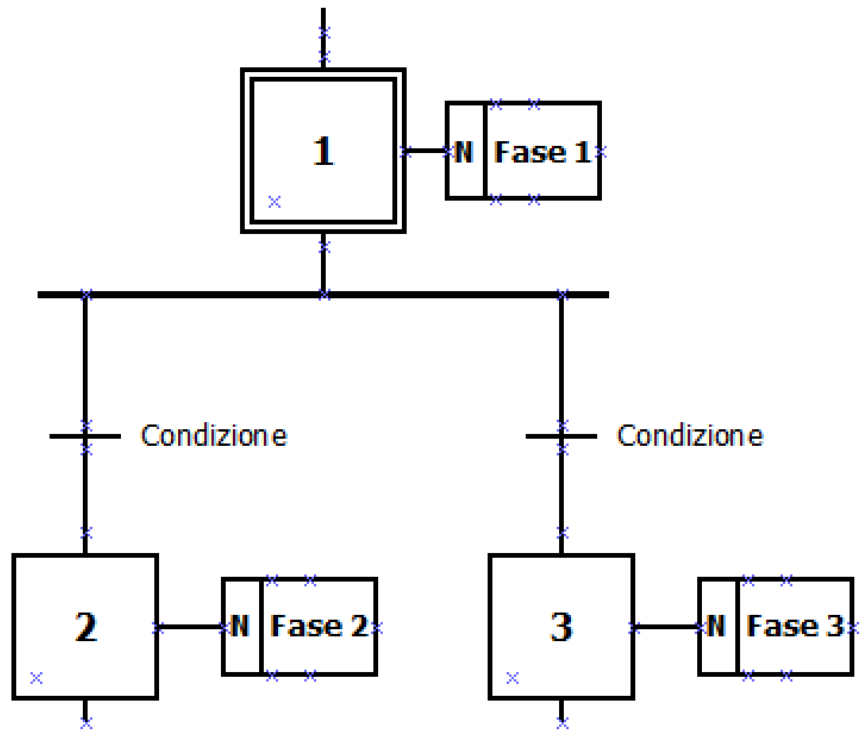
Per i qualificatori L, D, SD, DS e SL è necessario un valore di tempo nel formato di costante TIME. Immetterlo direttamente dopo il qualificatore, ad esempio L T#10s

## Azioni associate ad uno stato



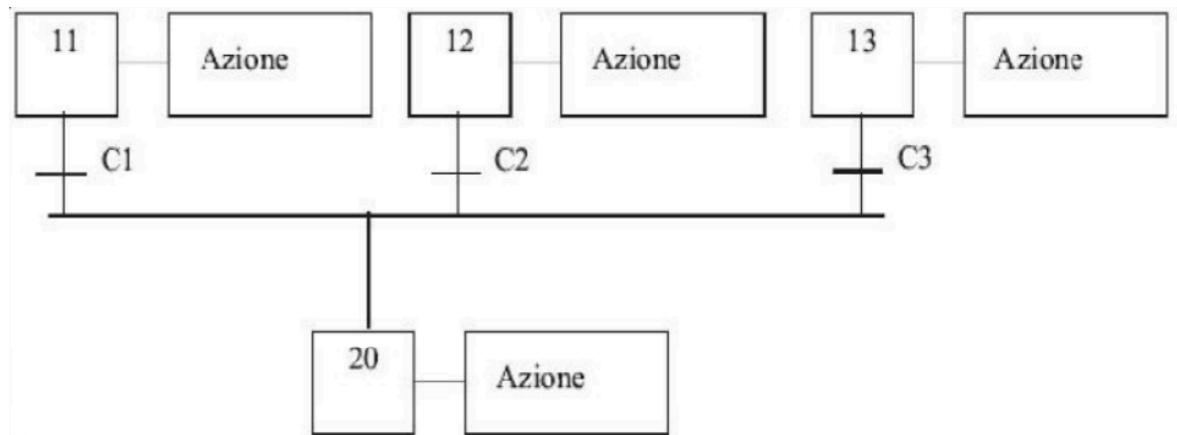


## SCELTA o DIVERGENZA tra due o più stati successivi





# CONVERGENZA

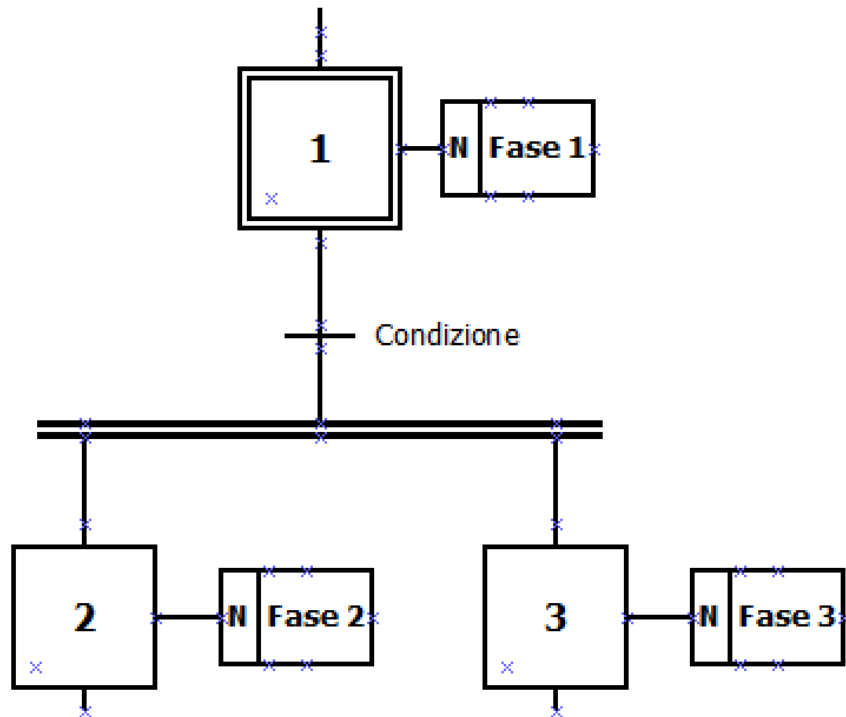


La STRUTTURA DI COLLEGAMENTO che vede TERMINARE PIÙ SEQUENZE in un MEDESIMO STATO attraverso DIFFERENTI TRANSIZIONI, è la CONVERGENZA.



# PARALLELISMO

La STRUTTURA DI COLLEGAMENTO che permette l'attivazione di SEQUENZE PARALLELE a partire da una singola TRANSIZIONE, è il **PARALLELISMO**:

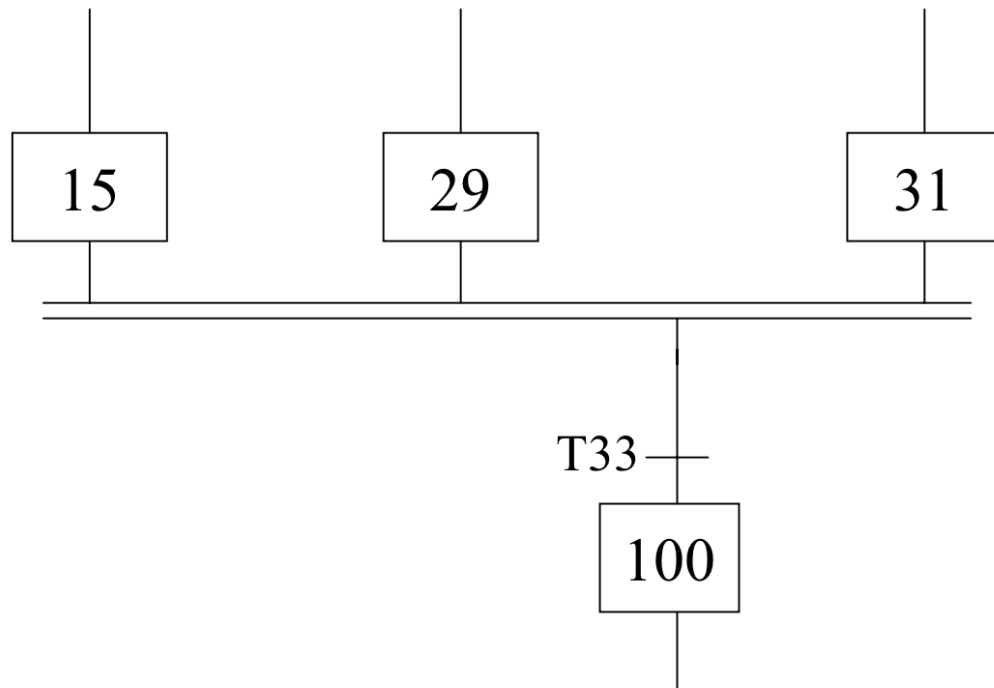


Quando la **TRANSIZIONE** viene ATTIVATA, TUTTI gli STATI A VALLE vengono ATTIVATI.



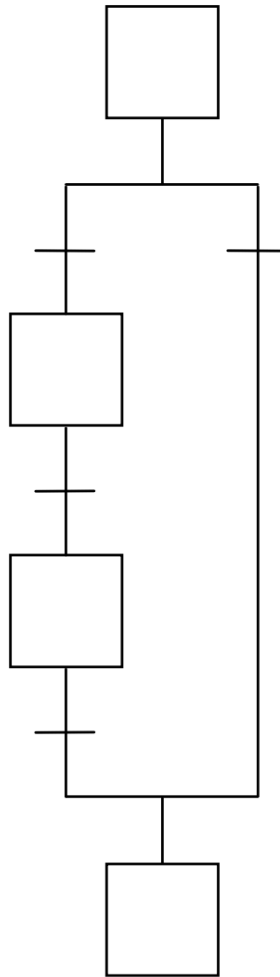


# SINCRONIZZAZIONE



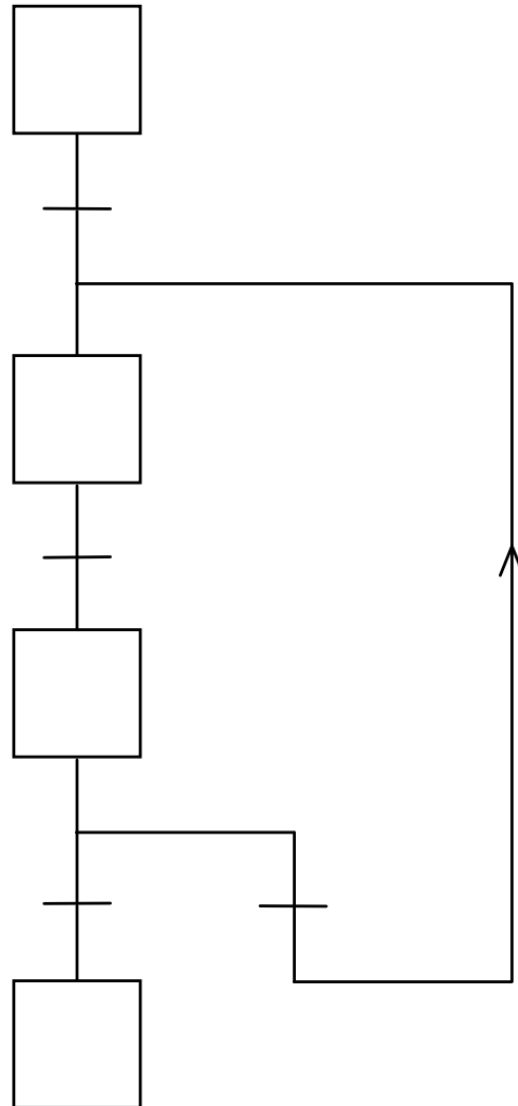


# SALTO DI SEQUENZA



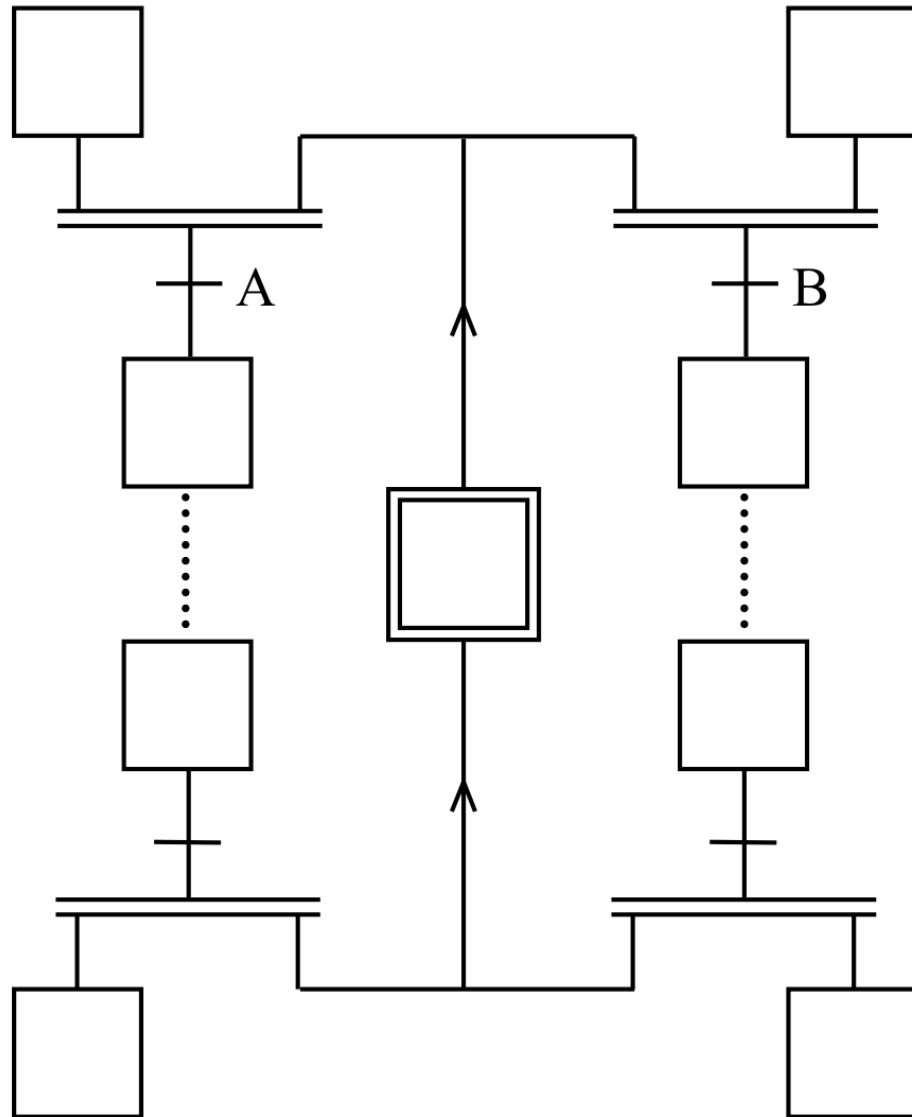


# RIPETIZIONE DI SEQUENZA





# SEMAFORO DI MUTUA ESCLUSIONE

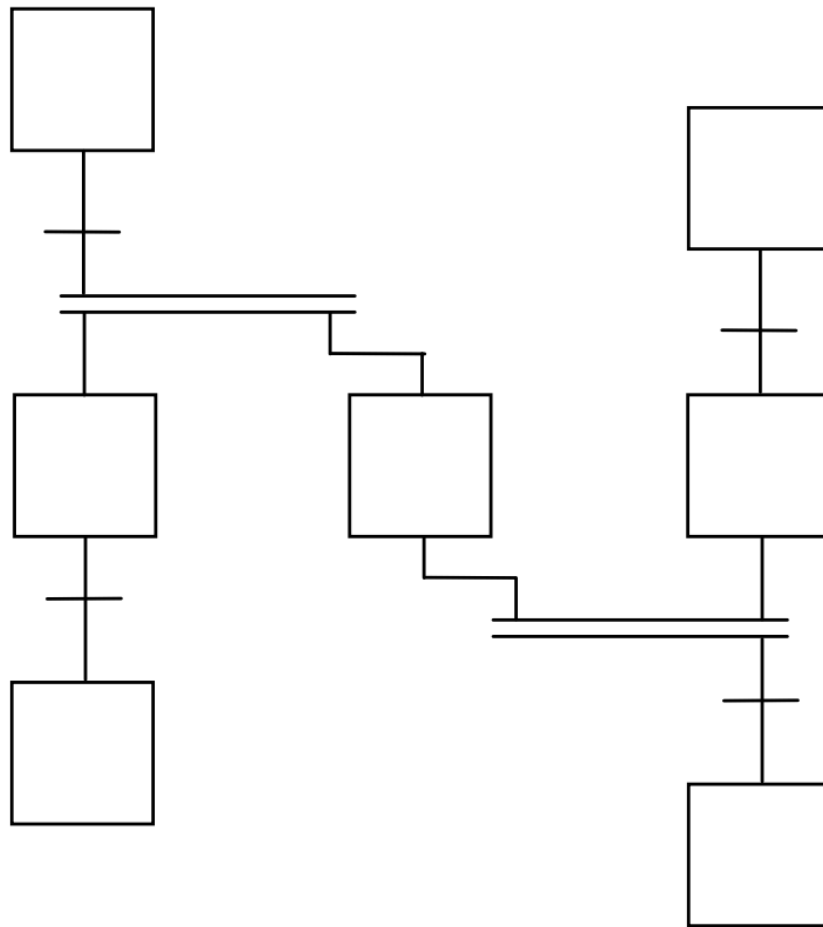


Le due condizioni, A e B, devono essere mutuamente esclusive.

La fase del semaforo deve essere inclusa tra le fasi iniziali (risorsa condivisa disponibile)



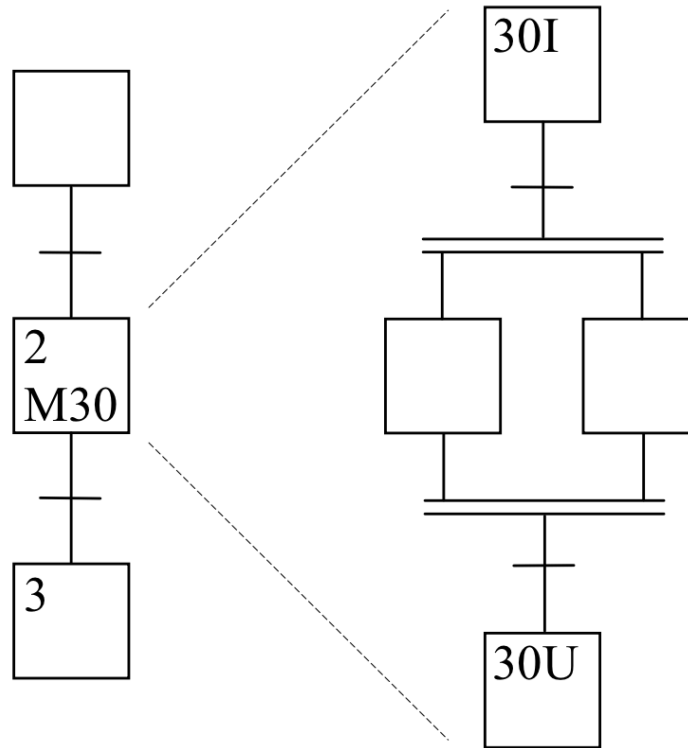
## SINCRONIZZAZIONE LOCALE



La sequenza di destra deve attendere la sequenza di sinistra.



## MACROFASI (o macrostati)



Le **macrofasi** (o azioni aggregate) sono azioni complesse costituite dalla combinazione di azioni semplici o aggregate e funzionano come i sotto-programmi. Quando si attiva la fase associata ad un'azione aggregate, essa comincia ad evolvere e termina quando la fase si disattiva.

Per consentire una rappresentazione più sintetica è possibile raggruppare pezzi di SFC in un **macrostato**:

All'ingresso nel macrostato, viene eseguito lo stato iniziale.

Viene eseguita la successione degli stati nel macrostato secondo le regole standard di SFC.

L'esecuzione del macrostato termina all'uscita dallo stato finale (si continua l'esecuzione dello schema SFC).

La transizione in uscita dal macrostato è abilitata solo se è stato raggiunto lo stato terminale dentro il macrostato.



# GERARCHIA PADRE/FIGLIO

Per un approccio strutturato alla programmazione si consiglia l'uso della gerarchia padre/figlio tra diagrammi SFC.

Il diagramma SFC figlio ha un proprio nome identificativo, una volta generato, evolve parallelamente e in maniera del tutto indipendente al padre.

In generale deve esistere un unico diagramma SFC principale (main) che genera e controlla SFC figli (avvia, mette in pausa, riavvia e termina).

## Comandi per le generazione e gestione dei diagrammi figli:

- **Gstart**(Nome-Figlio) per avviare il diagramma SFC Nome-figlio;
- **Gkill**(Nome-Figlio) per terminare l'evoluzione del diagramma SFC Nome-figlio;
- **Gfreeze**(Nome-Figlio) per sospendere l'evoluzione del diagramma SFC Nomefiglio;
- **GRST**(Nome-Figlio) per riavviare l'evoluzione del diagramma SFC Nome-Figlio precedentemente sospeso.

Tali comandi devono essere utilizzati come azioni caratterizzate dal **qualificatore P**, devono cioè essere implementati come azioni pulse ed eseguite solo una volta nel momento in cui lo step a cui sono associate diventa attivo.

## Differenze fondamentali tra la struttura gerarchica padre/figlio e un parallelismo:

- Il SFC padre può agire in qualsiasi istante sul SFC figlio terminandolo, sospendendolo o riavviandolo;
- Al contrario, due sequenze parallele attivate tramite un semplice parallelismo non sono assolutamente influenzabili: evolveranno infatti seguendo il loro diagramma SFC sino alla naturale conclusione del parallelismo (normalmente una sincronizzazione). La struttura gerarchica padre/figlio è particolarmente appropriata per gestione e supervisione.



## FLAG E TIMER DI FASE

Il flag di fase (stato attivo o inattivo di un passo) può essere rappresentato dal valore logico di un valore booleano.

Esso è rappresentato **\*\*\*.X**, dove **\*\*\*** è il nome della fase e **X** il flag.

Questa variabile booleana ha il valore 1 quando il passaggio corrispondente è attivo e 0 quando è inattivo.

Analogamente, il tempo trascorso, **\*\*\*.T**, esso parte all'attivazione della fase e viene congelato all'uscita di essa. (Tutti i timer di fase vengono azzerati nella fase iniziale e/o all'ingresso della fase stessa).

Quando un passo è disattivato, il valore del tempo trascorso del passo deve rimanere al valore che aveva al momento della disattivazione del passo.

Quando viene attivato un passo, il valore del tempo trascorso del passo deve essere resettato a **t#0s**.

### Regola base per il flag di stato della fase:

uno stato in un diagramma SFC deve rimanere attivo per almeno un periodo del funzionamento ciclico.

Le azioni associate a uno stato la cui transizione di uscita è caratterizzata da una condizione già verificata nel momento dell'attivazione, vengono quindi eseguite per almeno un ciclo di funzionamento.

Nel programma Ladder del PLC per ottemperare questa regola, bisogna appoggiare il flag di stato su altri rispettivi flag posizionati dopo la costruzione delle fasi. Si userà poi il flag d'appoggio per interrogare lo stato della fase.

Esempio:

----- First_scan -----	(Res all, Set P0)	Passo 0 di inizializzazione
----- P0aux ----- Tr1 -----	(Set P1, Res P0)	Passo 1
----- P1aux ----- Tr2 -----	(Set P2, Res P1)	Passo 2
----- P2aux ----- Tr3 -----	(Set P3, Res P2)	Passo 3
----- P3aux ----- Tr4 -----	(Set P0, Res P3)	Ritorno al passo 0
----- P0 -----	(P0aux)	
----- P1 -----	(P1aux)	
----- P2 -----	(P2aux)	
----- P3 -----	(P3aux)	





I timer di fase possono essere costruiti utilizzando una doppia word, con essa si può raggiungere un valore decimale di 4.294.967.295.

Tabella merker di clock/tempi

<b>Merker clock</b>	<b>Tempo max s</b>	<b>Tempo max h</b>	<b>Tempo max gg</b>
1ms	4294967	1193	49
10ms	42949672	11930	497
100ms	429496729	119304	4971

Esempio:

```
|-----|First_scan|------(Res all timer)      Passo 0 di inizializzazione
|-----|P1aux|---|ck|-----|inc|------(T1) Passo 1  (ck = merker di clock)
|-----|P2aux|---|ck|-----|inc|------(T2) Passo 1  (ck = merker di clock)
|-----|P3aux|---|ck|-----|inc|------(T3) Passo 1  (ck = merker di clock)
```



## Esempio strutturazione codice SFC

### Main:

avvio del processo (*Start*);  
inizializzazione del processo;  
arresto del processo (*stop*);  
gestione dell'emergenza in seguito al malfunzionamento di entrambi i nastri;

### ctrl-astro1:

gestione del controllo del nastro 1;  
diagnostica automatica dei guasti del nastro 1;

### ctrl-nastro2:

gestione del controllo del nastro 2;  
diagnostica automatica dei guasti del nastro 2;

### ctrl-Ventosa:

controllo del ribaltatore a ventosa;

### Buffer:

controllo della macchina di stampa con buffer di ingresso;

### Robot:

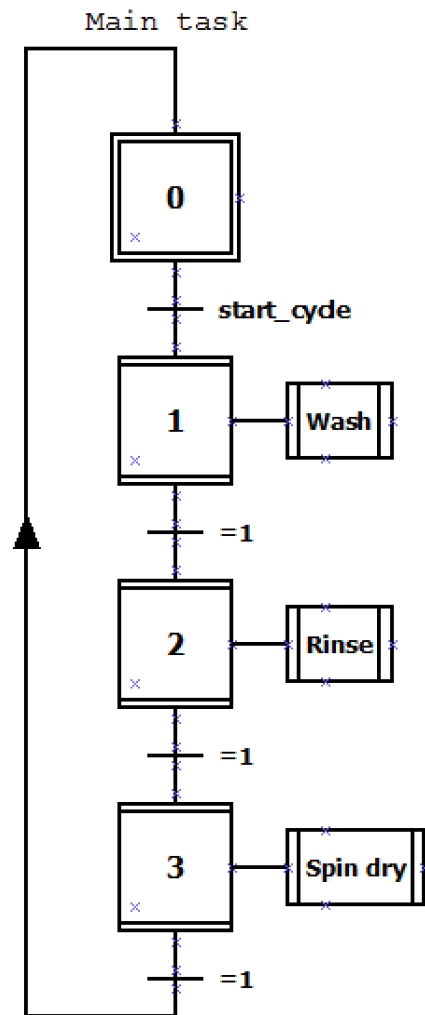
controllo dei due robot con sequenze mutuamente esclusive;  
controllo del nastro centrale;

### off\_filter:

filtraggio del segnale (*off*) e generazione del segnale (*stop*);

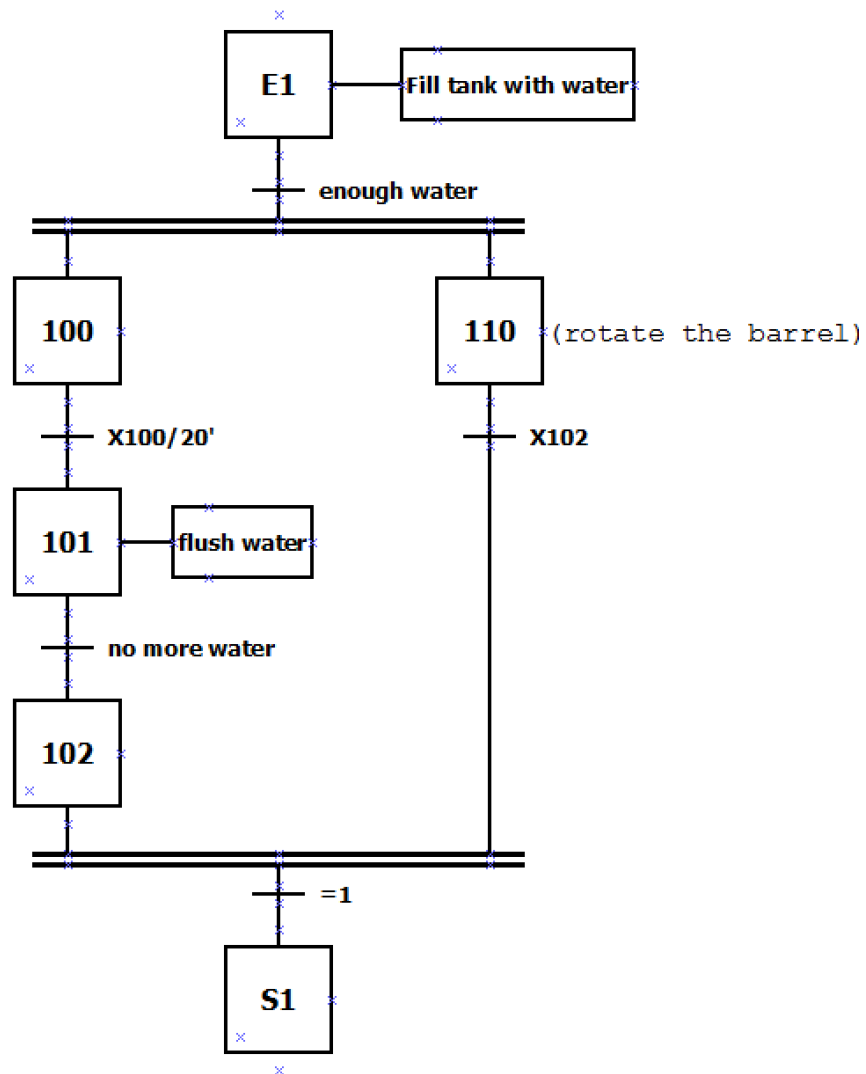


# ESEMPI



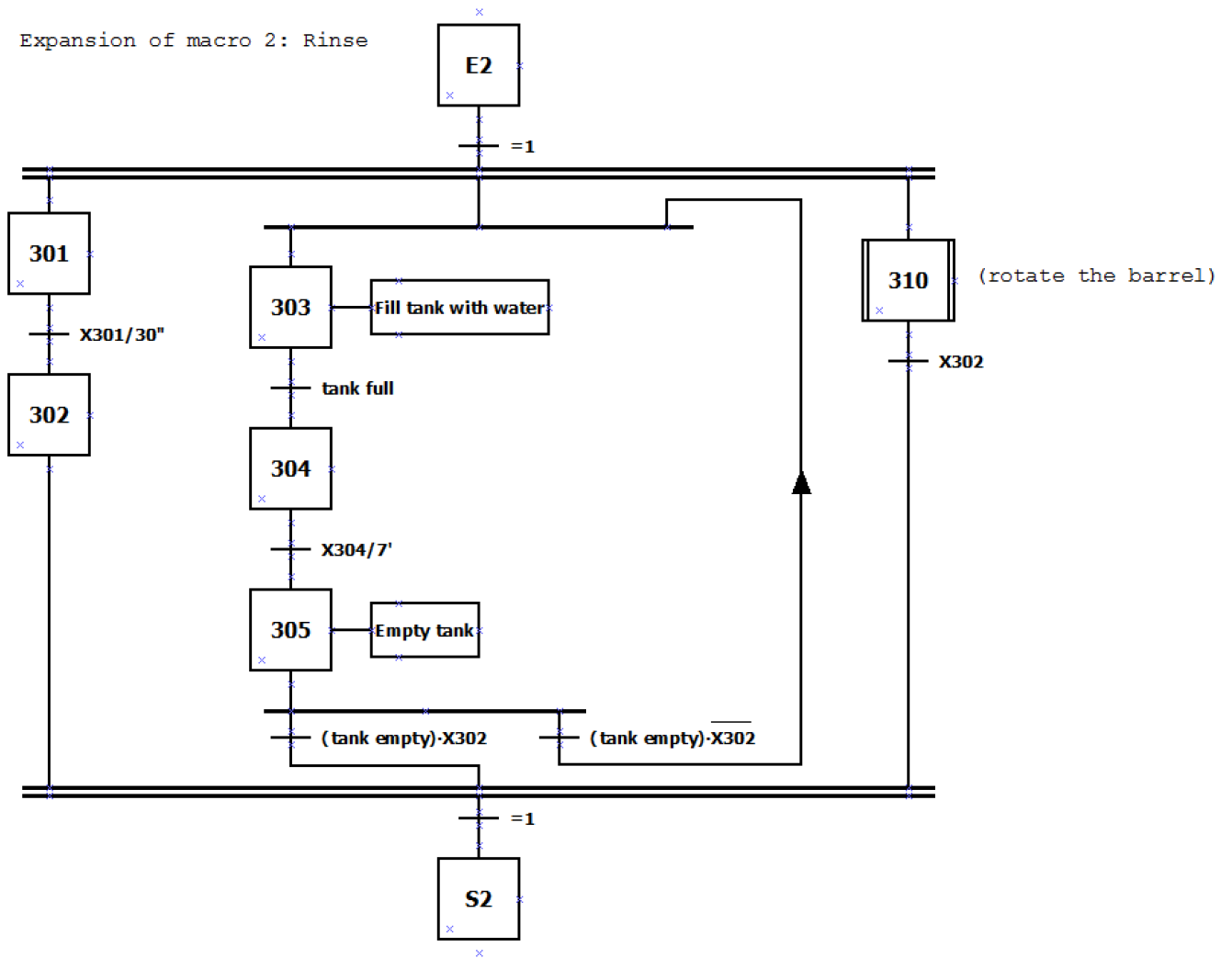


### Expansion of macro 1: Wash



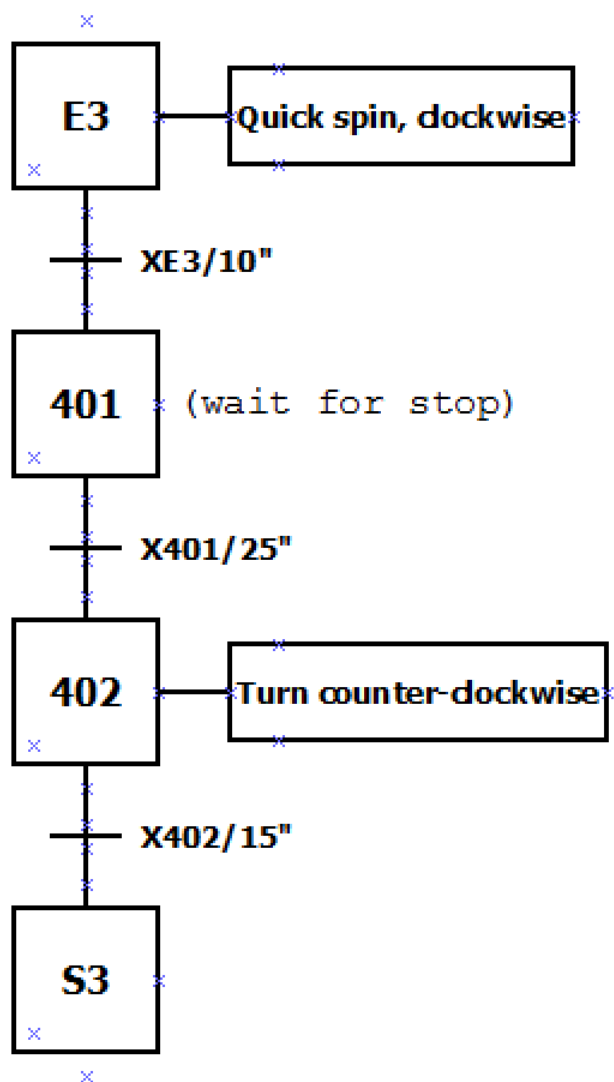


Expansion of macro 2: Rinse



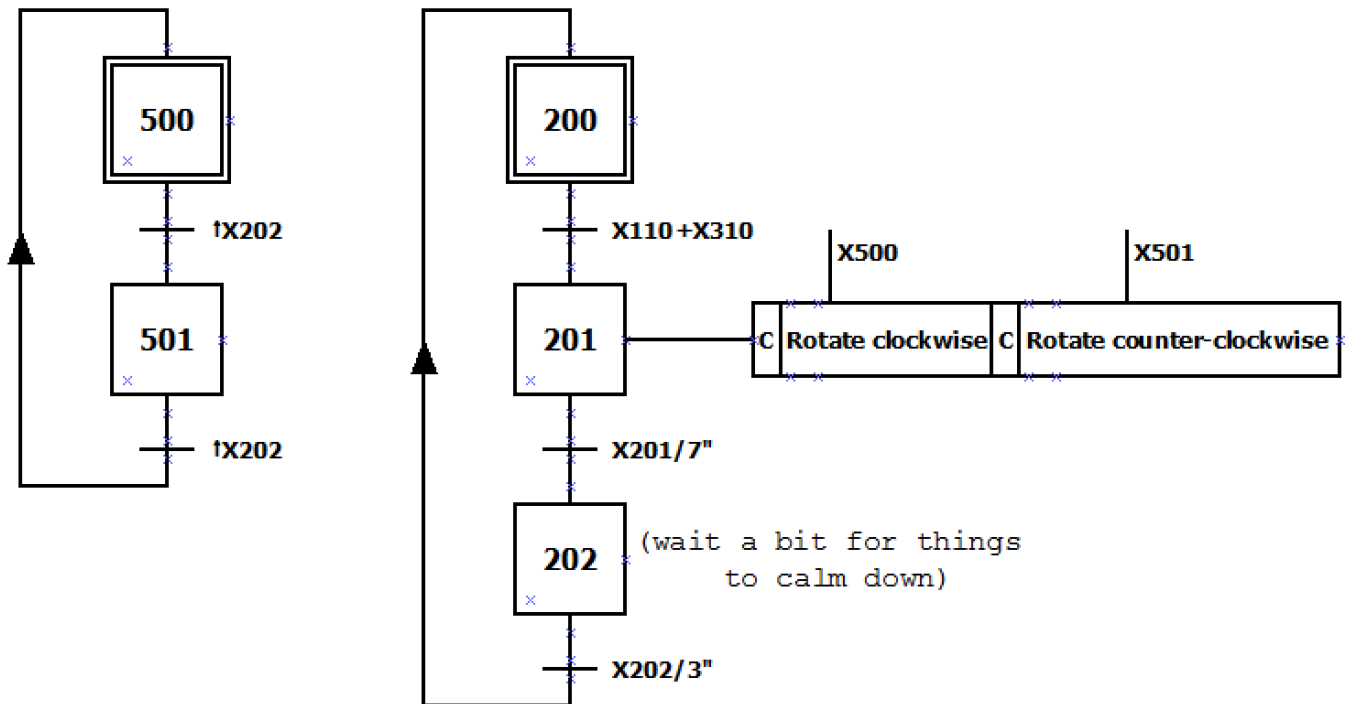


### Expansion of macro 3: Spin dry





Subprogram : Rotate the barrel  
(clockwise and counter-clockwise)





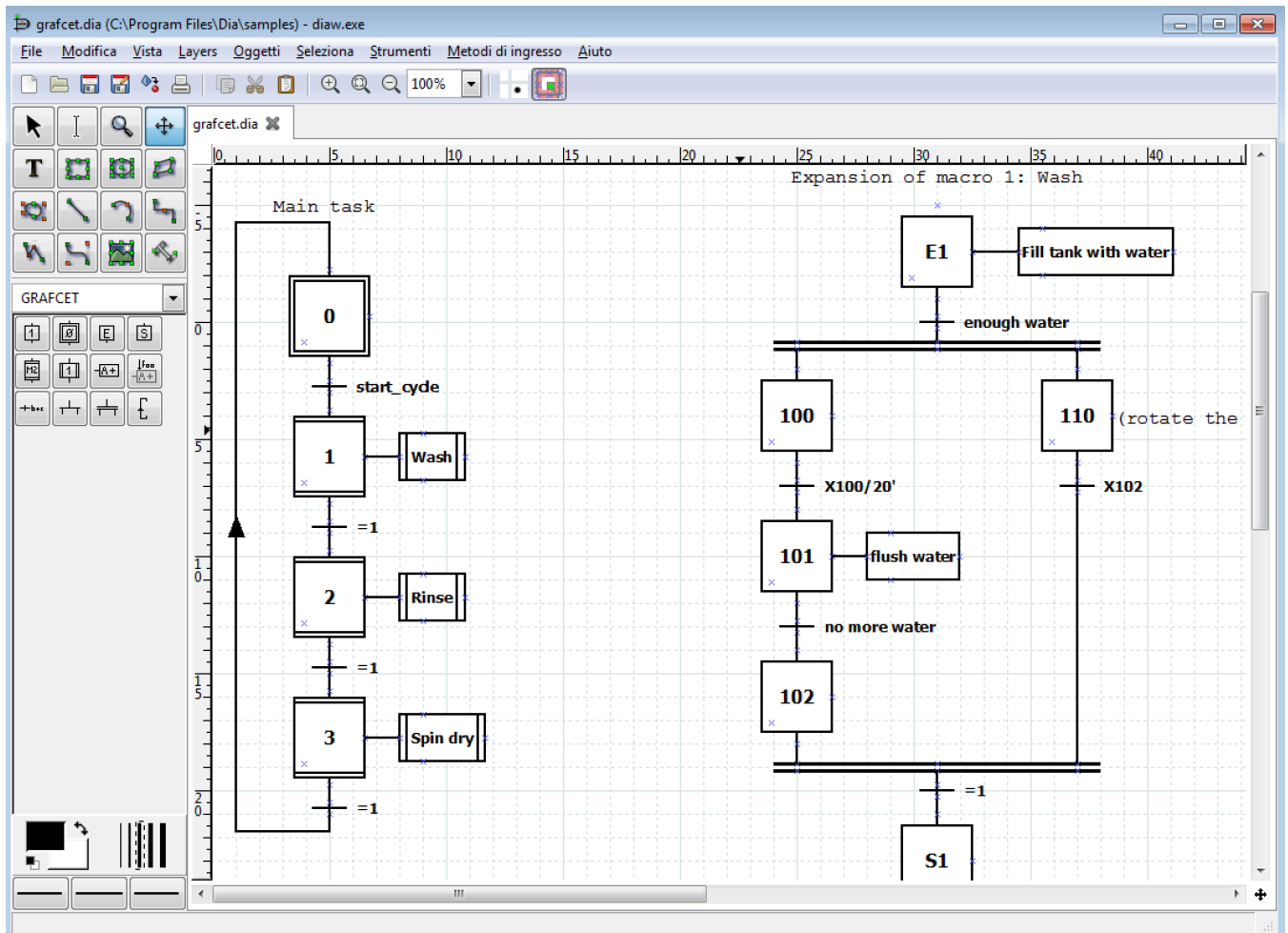
# IL PROGRAMMA DIA

**Dia** è un **software libero** per la creazione di **diagrammi**, sviluppato come parte del progetto **GNOME**. Ha un'architettura **modulare** che include varie **librerie** grafiche per disegnare vari tipi di diagrammi: **diagrammi di flusso**, **circuiti elettrici**, **circuiti booleani**, modelli **UML**, **diagrammi di rete**, **modelli E-R** ed altri schemi.

Dia può esportare i suoi diagrammi in PostScript, SVG, AutoCAD DXF ed altri formati immagini. Si possono creare script per Dia usando Python e interfacciarsi ad altri programmi.

Homepage Gnome: <https://wiki.gnome.org/Apps/Dia>



Homepage dia-installer.de <http://dia-installer.de/>










## DIA SETUP PAGINA

 Impostazioni della pagina 



**Formato della carta**



A4   
21 cm x 29,7 cm



**Orientazione**



☐  ☒ 

**Margini**



Superiore: 1,50 cm  





Inferiore: 1,50 cm  

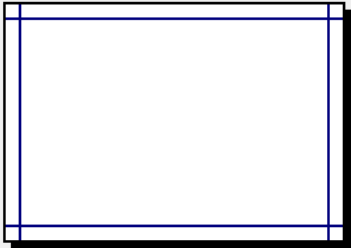
Sinistro: 1,50 cm  

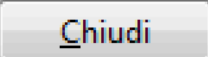

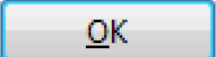
Destro: 1,50 cm  

**Ridimensiona**

☒ Scala: 36  

☐ Adatta a: 1   di 1  





# Indice

<b>ELEMENTI BASE DELL'SFC</b>	<b>2</b>
Regole fondamentali per la costruzione grafica dell'SFC:	3
Struttura del linguaggio SFC e interpretazione:	4
<b>AZIONI E QUALIFICATORI</b>	<b>5</b>
<b>SCELTA o DIVERGENZA</b> tra due o più stati successivi	6
<b>CONVERGENZA</b>	7
<b>PARALLELISMO</b>	8
<b>SINCRONIZZAZIONE</b>	9
<b>SALTO DI SEQUENZA</b>	10
<b>RIPETIZIONE DI SEQUENZA</b>	11
<b>SEMAFORO DI MUTUA ESCLUSIONE</b>	12
<b>SINCRONIZZAZIONE LOCALE</b>	13
<b>MACROFASI (o macrostati)</b>	14
<b>GERARCHIA PADRE/FIGLIO</b>	15
<b>FLAG E TIMER DI FASE</b>	16
<b>Esempio strutturazione codice SFC</b>	18
<b>ESEMPI</b>	19
<b>IL PROGRAMMA DIA</b>	24
DIA SETUP PAGINA	25
<b>Indice</b>	<b>26</b>