

# Festo Teach Language FTL

## Programación de sistemas de manipulación



### Así se crean programas de movimiento para cinemáticas:

El lenguaje de programación FTL (Festo Teach Language) no requiere gran formación en programación ya que se concentra en lo esencial: la aplicación. De este modo, se le facilitan las cosas al usuario y se consigue minimizar el tiempo de puesta en marcha.

### Nuestra experiencia a su disposición

El lenguaje FTL se aprende intuitivamente y utiliza macros, p. ej. “Lin” para definir un movimiento lineal cartesiano o “Vel” y “Acc” para definir la velocidad y la aceleración.

Los programas FTL se guardan en formato de texto y pueden editarse fácilmente con un sistema de programación existente.

### Estructuras claras para guardar los proyectos

Los proyectos en FTL tienen estructuras claras que permiten clasificar las aplicaciones p. ej.,

por nombre y por variantes. Los datos con nombres de proyecto y de programa, de libre configuración, se pueden importar y exportar simplemente con el FCT o con una memoria externa (lápiz de memoria USB).

### Optimización de proyectos en un abrir y cerrar de ojos

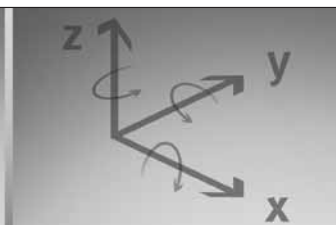
Con el terminal de mano CDSA es sencillísimo corregir posiciones e introducir las en el programa FTL mediante teach-in. Todo puede aplicarse en seguida, ya que se prescinde de procedimientos que exigen mucho tiempo, como p. ej., la compilación.



Programación intuitiva



Corrección rápida de posiciones



6 grados de movilidad

129.4.PSI →

Product Short Information

# Festo Teach Language FTL

## Programación de sistemas de manipulación

### Ventajas

- Formato de texto fácil de crear y de editar
- Fácil de entender; basado en instrucciones en inglés
- Creación de programas con 6 grados de libertad de movimiento
- Puesta en práctica inmediata, no es necesario ningún proceso de traducción (p. ej., compilación)
- Cómodo editor de FTL integrado en el Festo Configuration Tool (FCT)

### Ejemplo de programa en FTL

**Macros**  
Macros de movimiento, de dinámica, para robot

**Variables**  
BOOL, integer, real, string

**Órdenes lógicas**  
IF .. THEN .. ELSE  
WHILE .. DO  
LOOP

```

feed                                CONT   Line: 1
1 Vel(dynCart, 75)
2 Acc(dynCart, 100)
3 Tool(gripper1)
4 Ptp(home)
5 flagLoop := TRUE
6 index := 0
7 // loop for feeding
8 WHILE index = 0 DO
9   IF index = 0 THEN
10    // start position
11    Lin(startPos)
  
```

### Funciones

- Variables de libre definición
- Programación de matrices
- Bifurcaciones lógicas (p. ej., IF..THEN..ELSE) o bucles (p. ej., WHILE, LOOP)
- Saltos en el programa con GOTO o IF...THEN GOTO
- Instrucciones para movimientos punto a punto (PTP), cartesianos, lineales y circulares
- Modificación de la velocidad, aceleración, etc. según la trayectoria y el eje
- Funciones que conservan el sistema mecánico, p. ej., avance aproximado, diversas rampas de aceleración, etc.
- Sistemas de referencia (desplazamiento del punto cero) con 6 grados de libertad
- Definición de extremos de la herramienta (TCP)
- Subprogramas y programas paralelos
- Variables de interface a sistemas de nivel superior, p. ej. un PLC
- Programación directa de señales periféricas, p. ej., entradas/salidas
- Instrucciones de conmutación en la trayectoria
- Seguimiento de objetos (tracking)

### Programación cómoda en el editor de FTL

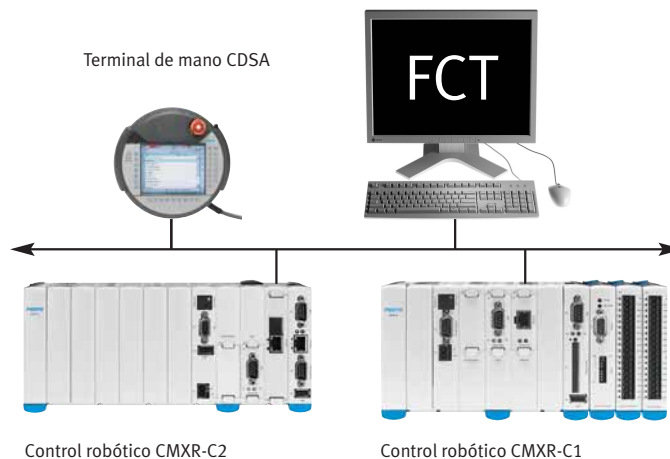
Local variables: Program

```

17 IndexBlue := 0
18 // MAX. count of parts
19 MaxParts := 40
20 // count of columns
21 MaxColumns := 8
22
23 WHILE IndexSource < MaxParts DO
24 // reference the source pallet
25 OutToolIDegree.Set()
26 OutToolIDegree.Reset()
27 SetRefDynaP(RefSource)
28 Tool(ToolSource)
29 // calculate the Positions
30 CalcIndex := IndexSource
31 CALL CalcIndex()
32 Lin(AbovePos)
33 Vacuum.Set()
34 Lin(TakePos)
35 // reset the sensor signals, before WaitTime
36 // WaitTime is needed, because the function ResetRisingEdge
37 // is executed in the main MB.
  
```

Functions: PLC interface, Movement, Circle, Lin, LinRel, LinToCircle, MoveAutoCart, MoveAxisPtp, Ptp, PtpRel, PtpToCircle, StopMove, Dynamic, Robot commands, Reference systems, Reference axis, Functions, Program commands, Message system, System Functions.

### Compatibilidad de la programación en FTL con todos los componentes



Festo AG & Co. KG

Ruiter Strasse 82  
73734 Esslingen  
www.festo.com  
Tel. +49 711 347-0  
Fax +49 711 347-2144  
service\_international@festo.com