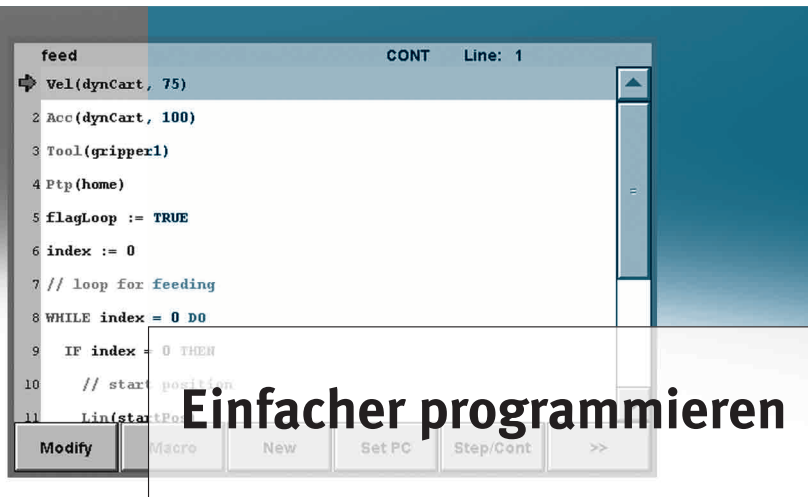


# Festo Teach Language FTL

## Programmieren von Handhabungssystemen



### Einfacher programmieren

#### So erstellt man Bewegungsprogramme für Kinematiken:

Die Programmiersprache FTL (Festo Teach Language) erfordert keine umfangreiche Programmierausbildung. Sie richtet den Fokus voll und ganz auf das Wesentliche: die Applikation. Im Sinne des Anwenders und nach der Prämisse „short time to application“.

#### Know-how ganz selbstverständlich

Die intuitiv erlernbare FTL Sprache beinhaltet Makros, z. B. Lin für eine kartesische Linearbewegung oder Vel und Acc zum Setzen der Geschwindigkeit bzw. Beschleunigung. Abgespeichert im Textformat, können FTL Programme ebenso leicht editiert wie über ein bestehendes Programmiersystem generiert werden.

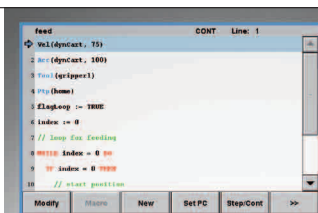
#### Projektanlage übersichtlich im Griff

FTL bietet klare Projektstrukturen, mit denen Applikationen,

z. B. nach Namen und Varianten, sortiert werden. Die Daten mit frei wählbaren Projekt- und Programmnamen lassen sich über FCT oder USB-Stick problemlos importieren und exportieren.

#### Prozesse im Handumdrehen optimiert

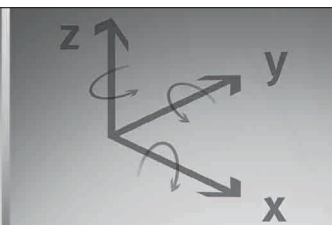
Mit dem optionalen Handbediengerät CDSA fällt es nicht schwer, Positionen zu korrigieren und per Teach-In-Mechanismus in das FTL-Programm zu übernehmen. Alles ist sofort startklar, denn zeitraubende Vorgänge, wie z. B. Kompilieren, entfallen.



Intuitive Programmierung



Positionen schnell korrigiert



6 Freiheitsgrade

129.4.PSI →

Product Short Information

# Festo Teach Language FTL

## Programmieren von Handhabungssystemen

### Vorteile

- Textuelles Format leicht zu bearbeiten und zu erzeugen
- Einfach verständlich, angelehnt an die englische Sprache
- Programmierstellung in 6 Freiheitsgraden
- Sofort startbar, kein Übersetzen, z. B. Kompilieren, notwendig
- Komfortabler FTL Editor, eingebettet in das Festo Configuration Tool (FCT)

### Beispiel FTL Programm

**Makros**  
Bewegung-, Dynamik-, Robot-Makros

**Variable**  
BOOL, integer, real, string

**Logikbefehle**  
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)

```

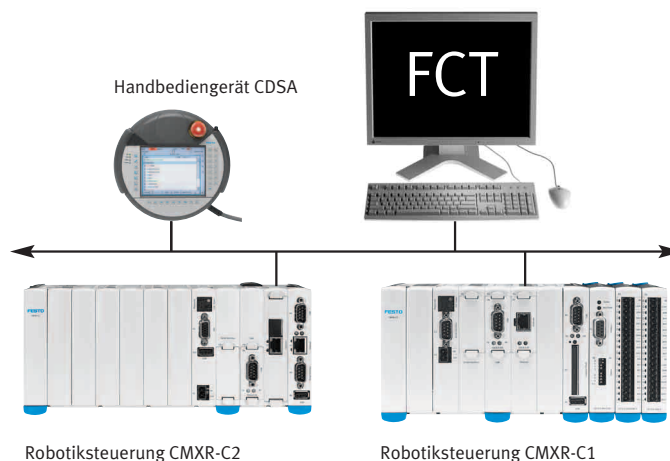
### Funktionen

- Unterschiedliche frei definierbare Variable
- Array-Programmierung
- Logische Verzweigungen, z. B. IF..THEN..ELSE oder Schleifen, z. B. WHILE, LOOP
- Sprünge im Programm mit GOTO oder IF...THEN GOTO
- Bewegungsbefehle für Point-To-Point- sowie kartesische Linear- und Zirkularbewegungen
- Beeinflussung von Geschwindigkeit, Beschleunigung und Ruck, bezogen auf Bahn und Achse
- Mechanikschonende Funktionen, z. B. Überschleifen, verschiedene Beschleunigungsrampen
- Referenzsysteme (Nullpunktverschiebungen) mit 6 Freiheitsgraden
- Definition von Werkzeugendpunkten (TCP)
- Unterprogramme und parallele Programme
- Schnittstellenvariablen zu übergeordneten Systemen, z. B. einer SPS
- Direkte Programmierung von Peripheriesignalen, z. B. Ein-/Ausgängen
- Befehle für Schaltaktionen auf der Bahn
- Objektverfolgung (Tracking)

### Komfortable Programmierung im FTL Editor

The screenshot shows the FTL Editor interface. On the left, there's a 'Local variables' section with a list of variables like 'IndexBlue := 0', 'MaxParts := 40', etc. The main area displays a program with comments and code lines. On the right, there's a 'Functions' list with categories like 'PLC Interface', 'Movement', 'Dynamics', etc.

### Unterstützung der FTL Programmierung durch die Komponenten



**Festo AG & Co. KG**  
 Postfach  
 73726 Esslingen  
 Rüter Straße 82  
 73734 Esslingen  
 Tel. +49 (0)711 347 0  
 Fax +49 (0)711 347 2628  
 E-mail: info\_de@festo.com