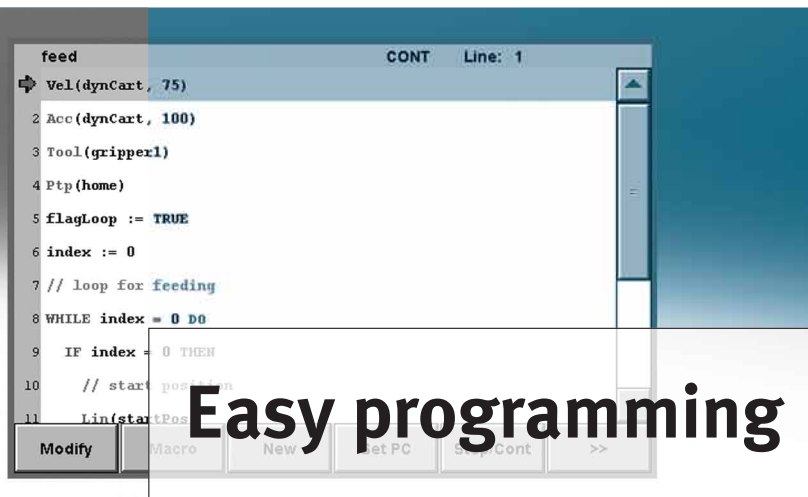


Festo Teach Language FTL

Programming handling systems



Easy programming

This is how you create motion diagrams for kinematics.

You don't need extensive training to use FTL (Festo Teach Language). FTL focuses completely on the basics, i.e. the application and provides the user with the benefit of short time to project realisation.

Know-how as a matter of course

The FTL language, which can be learned intuitively, contains macros, e.g. Lin for a Cartesian linear motion or Vel and Acc for setting the speed or acceleration. FTL programs are stored in text format and can therefore be easily edited and generated using existing PC-based programs

Project storage under control

FTL offers clear project structures for sorting applications, e.g. according to name and variants. The data with freely selectable

project and program names can easily be imported and exported using the Festo Configuration Tool (FCT) or a USB stick.

Processes optimised in no time at all

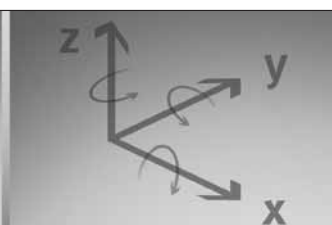
With the optional teach pendant CDSA it's simple to correct positions and to adopt them in the FTL program using the teach-in mechanism. Everything is immediately ready for use. Time-consuming processes, such as compiling, are no longer necessary.



Intuitive programming



Positions quickly corrected



6 degrees of freedom

129.4.PSI →

Product Short Information

Festo Teach Language FTL

Programming handling systems

Benefits

- Text format easy to create and edit
- Easy to understand, based on the English language
- Program generation in 6 degrees of freedom
- Immediately ready to use, e.g. no compiling, necessary
- Convenient FTL editor, imbedded in the Festo Configuration Tool (FCT)

Functions

- Different, freely definable variables
- Array programming
- Logical branches, e.g. IF..THEN..ELSE or loops, e.g. WHILE, LOOP
- Jumps in the program with GOTO or IF...THEN GOTO
- Motion commands for point-to-point as well as Cartesian linear and circular motions
- Influencing speed, acceleration and jerking, related to path and axis
- Functions that protect the mechanical components, e.g. skipping, different acceleration ramps
- Reference systems (zero point offsets) with 6 degrees of freedom
- Defining tool centre points (TCP)
- Subprograms and parallel programs
- Interface variables for higher-level systems, e.g. of a PLC
- Direct programming of peripheral signals, e.g. inputs/outputs
- Commands for switch actions on the path
- Tracking

Example of FTL program

Macros
Motion, dynamic response, robot macros

Variables
BOOL, integer, real, string

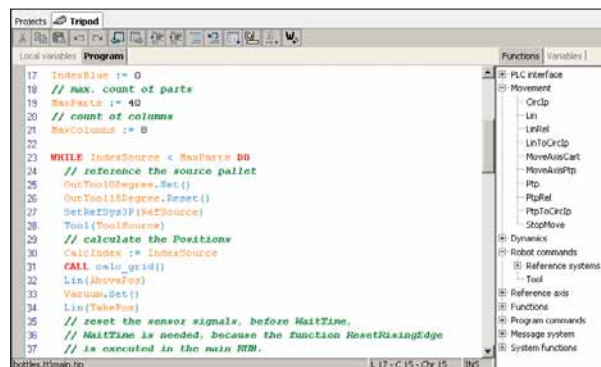
Logic commands
IF .. THEN .. ELSE
WHILE .. DO
LOOP

```

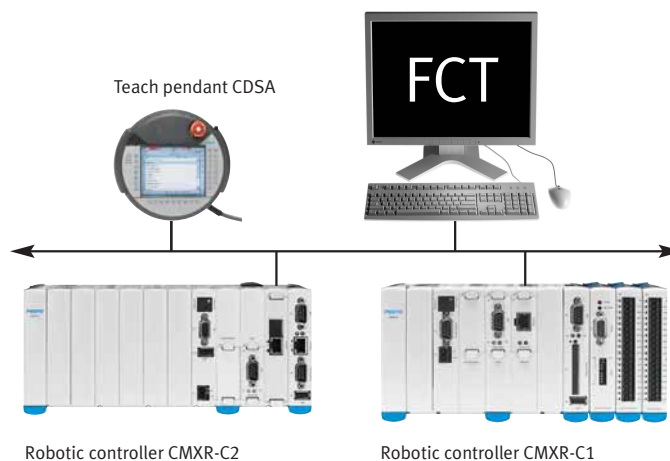
feed                                CONT   Line: 1
1 Vel(dynCart, 75)
2 Acc(dynCart, 100)
3 Tool(gripper1)
4 Ptp(home)
5 flagLoop := TRUE
6 index := 0
7 // loop for feeding
8 WHILE index = 0 DO
9   IF index = 0 THEN
10    // start position
11    Lin(startPos)

```

Convenient programming in the FTL editor



Components that support FTL programming



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