

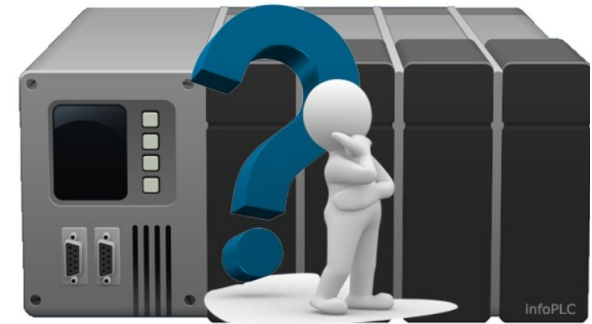
LENGUAJES DE PROGRAMACIÓN PLC's

Programa

Conjunto de instrucciones, órdenes y símbolos reconocibles por el PLC, a través de su unidad de programación, que le permiten ejecutar una secuencia de control deseada.

Lenguaje de programación

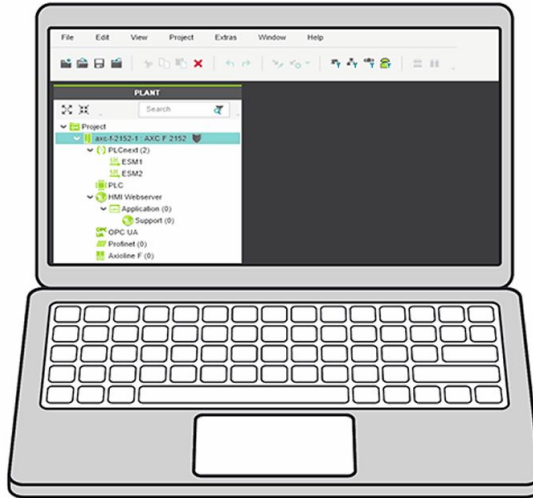
Permite al usuario ingresar un programa de control en la memoria del PLC, usando una sintaxis establecida.



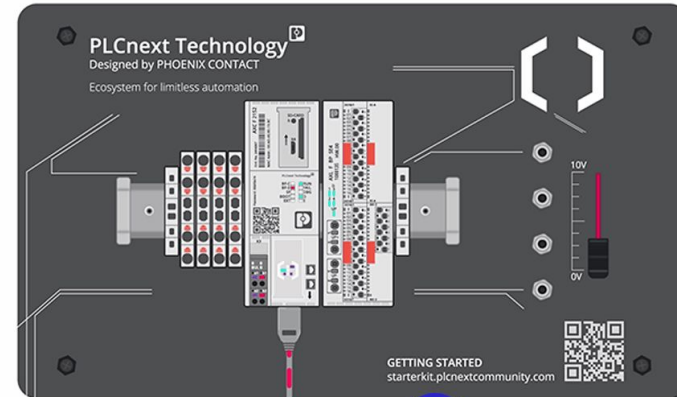
LENGUAJES DE PROGRAMACIÓN PLC's



 PLCnext Engineer



PLCnext Starterkit



Ethernet IP address:
192.168.1.1

15

REALPARS

SOFTWARE DE PROGRAMACIÓN PLC's



PLCnext Engineer

File Extras Help

Start Page

PHENIX CONTACT
INSPIRING INNOVATIONS

PLCnext Engineer 2022.0.1

(Build 5.0.546.0)

Seleccionar referencia PLC

Recent projects

Project templates



AXC F 1152 v00 / 2021.9.0



AXC F 1152 v00 / 2022.0.0

Need help?

Welcome to PLCnext Engineer
The Start Page welcomes you introducing PLCnext Engineer. The Start Page is shown every time you start closed when opening or creating a project.

Recent projects
The 'Recent projects' list shows the projects that you have recently opened in PLCnext Engineer. Click the

Project templates
The 'Project templates' list provides a list of predefined project templates to help you get started. Selecting a project with the desired controller type already added.

Further help
If you do not know what to do after a project is opened, you can find further help [here](#).

PLCnext Engineer - C:\Users\AndreitaE\AppData\Roaming\PHOENIX CONTACT\PLCnext Engineer\2022.0\NEWPROJECT1.pcwex

File Edit View Project Extras Window Help

PHOENIX CONTACT

PLANT

- Project
 - axc-f-1152-1 : AXC F 1152
 - PLCnext (1)
 - PLC
 - HMI Web Server
 - OPC UA
 - Profinet (0)
 - Axioline F (0)
 - PLCnext Components (0)

COMPONENTS

- Programming (300)
- PLCnext Components & Programs (0)
- Network (511)
- HMI (34)
- Libraries (1)

0 errors, 0 warnings 100%

IEC 61131-3

Comisión Electrónica Internacional



- Parte 1: información general Ed. 2.0 (2003)
- Parte 2: Especificaciones y ensayos de los equipos Ed 3.0 (2007) (hardware)
- **Parte 3: Lenguaje de programación Ed 2.0 (2003).**
- Parte 4: Guías de usuario Ed 2.0 (2004).
- Parte 5: Comunicaciones Ed. 1.0 (2000).
- Parte 6: Seguridad Funcional Ed. 1.0 (2012)
- Parte 7: Programación para control difuso (Fuzzy) Ed. 1.0 (2000)
- Parte 8: Directrices para la aplicación e implementación de lenguajes de programación Ed. 2.0 (2003)

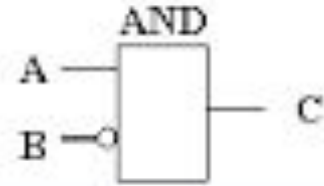
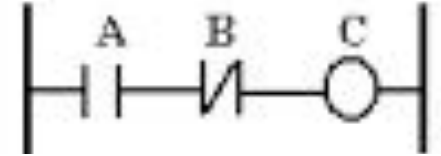


IEC 61131-3

Comisión Electrónica Internacional

1. Lenguajes Gráficos:

- Diagrama de escalera o Ladder (LD)
- Diagrama de bloques funcionales (FBD)



2. Lenguajes Escritos:

- Lista de instrucciones (IL)
- Texto Estructurado (ST)

LD A
ANDN B
ST C

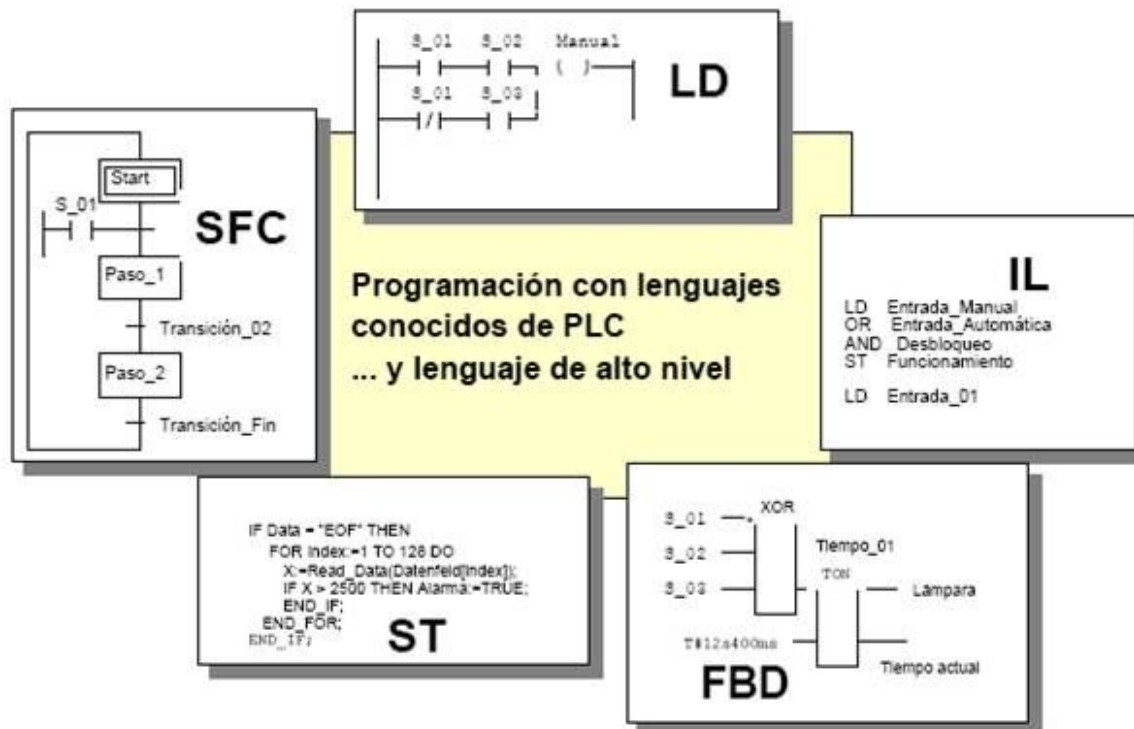
3. Sequential Function Chart (SFC)

Grafcet (modelado y programación)

$C = A \text{ AND NOT } B$

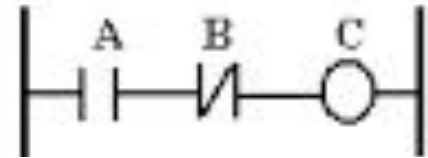
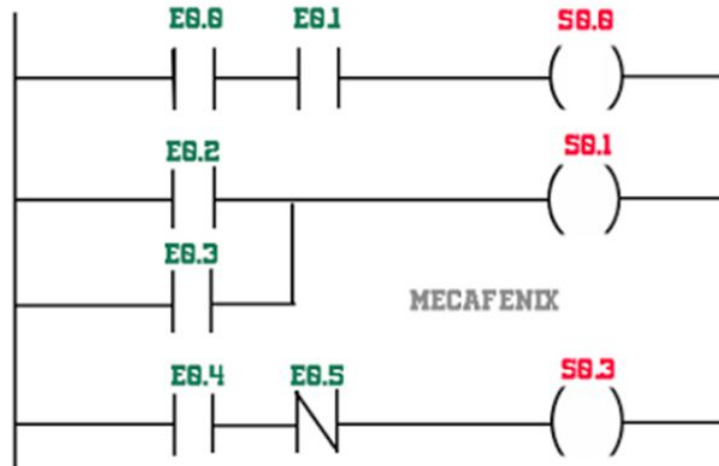
IEC 61131-3

Comisión Electrónica Internacional



Ladder

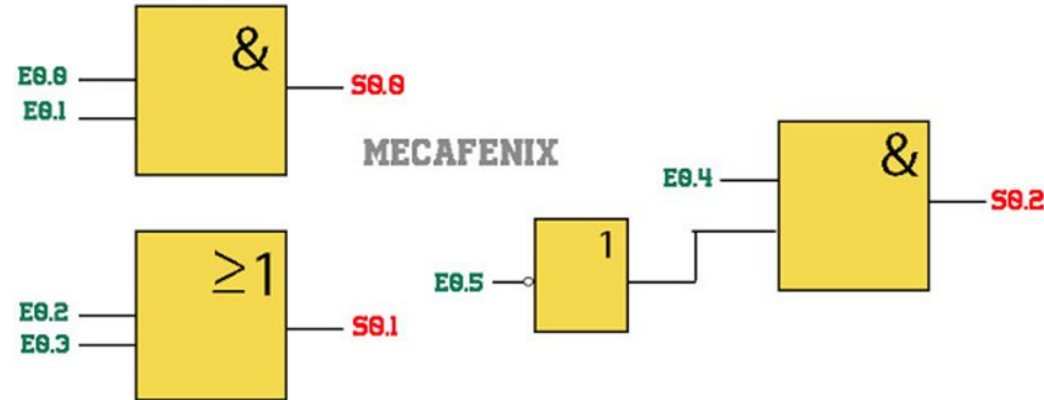
Lenguaje de programación inspirado en el concepto tradicional de la **lógica cableada**, Se basa en barras de alimentación, elementos de enlace, contactos, bobinas y bloques funcionales.



Bloques Funcionales FBD

Lenguaje de programación gráfico basado en **bloques con entradas y salidas** que pueden ser interconectados entre sí para formar una función lógica - aritmética, los tipos de datos dependen del bloque.

Ej: Bloque AND, Suma, Resta.



Lista de Instrucciones IL

Es un lenguaje tipo **ensamblador**. Usa la sintaxis del Álgebra de Boole para ingresar y explicar la lógica de control. Consiste en elaborar una lista de instrucciones o nemónicos, haciendo uso de operadores Booleanos (AND, OR, NOT, etc.) y otras instrucciones nemónicas, para implementar el circuito de control.

```

000 LD    %I0.1  Bp. inicio ciclo
    AND   %I0.0  Dp. presencia vehiculo
    AND   %M3     Bit autorización reloj calendario
    AND   %I0.5  Fc. alto rodillo
    AND   %I0.4  Fc. detrás pórtico
005 S     %M0     Memo inicio ciclo
    LD    %M2
    AND   %I0.5
    OR    %I0.2  Bp. parada ciclo
    R     %M0
010 LD    %M0
    ST    %Q0.0  Piloto ciclo
  
```

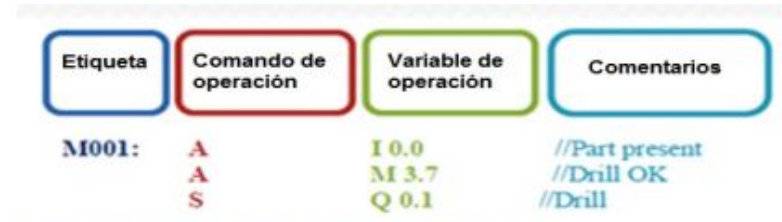


Imagen 7: Estructura de una orden en programación IL/STL.

Texto Estructurado

Es un lenguaje escrito, representa un modo de programación de **alto nivel**, algunos similares a pascal, C, C++.

```
IF ((E0.0 == TRUE) && (E0.1 == TRUE))  
{  
  S0.0 = TRUE;  
}  
ELSE S0.0 = FALSE;
```

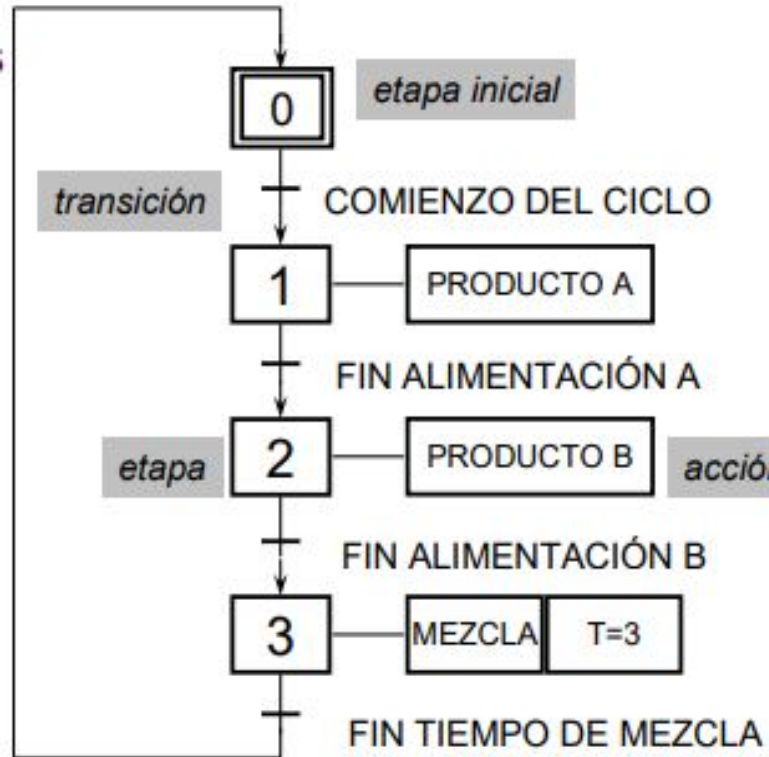
```
IF ((E0.2 == TRUE) || (E0.3 == TRUE))  
{  
  S0.1 = TRUE;  
}  
ELSE S0.1 = FALSE;
```

```
IF ((E0.4 == TRUE) && (E0.5 == FALSE))  
{  
  S0.2 = TRUE;  
}  
ELSE S0.2 = FALSE;
```

```
IF Dato = "EOF" THEN  
  FOR indice:=1 TO 128 DO  
    x:= Read_dato(Canal[indice]);  
    IF x>2500 THEN Alarma:=TRUE;  
    END_IF;  
  END_FOR;  
END_IF
```

SFC

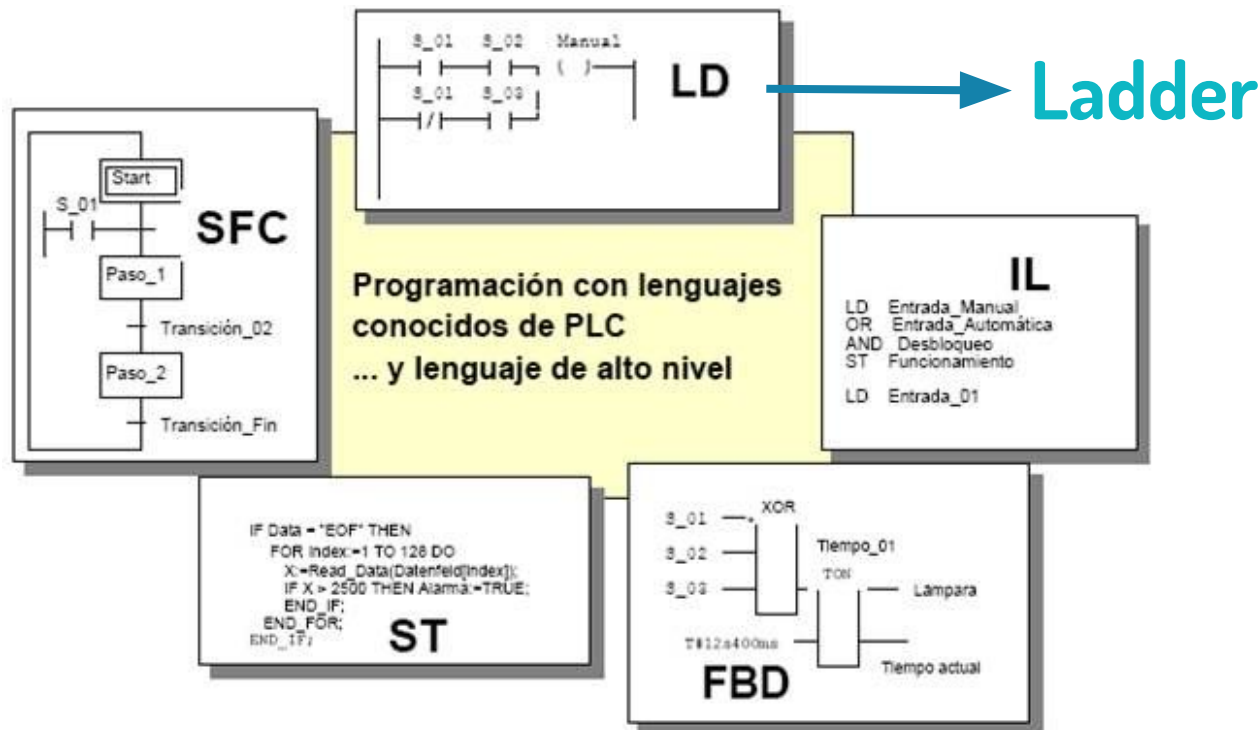
Sequential Function Chart



- Las etapas o estados implican acciones asociadas
- Las transiciones gobiernan los cambios de estado
- Las flechas indican la dirección de cambio
- Pueden darse esquemas menos lineales

IEC 61131-3

Comisión Electrónica Internacional



Ladder o escalera

- Lenguaje más usado para la programación de plc's
- Fue el primero con el que se empezó a programar, de allí su semejanza con diagramas eléctricos de escalera utilizados por los técnicos antes de la aparición del autómat.

