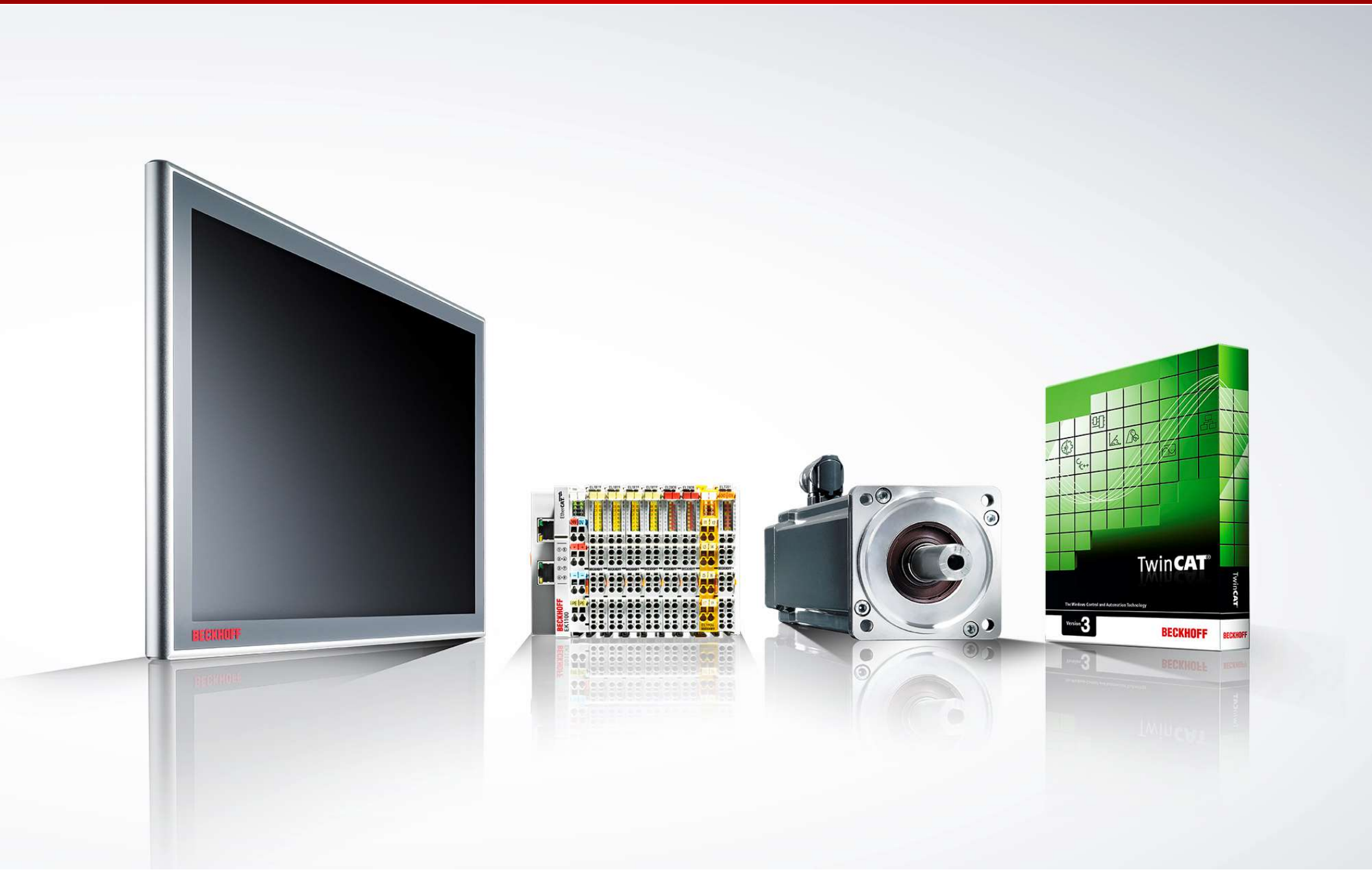
BECKHOFF

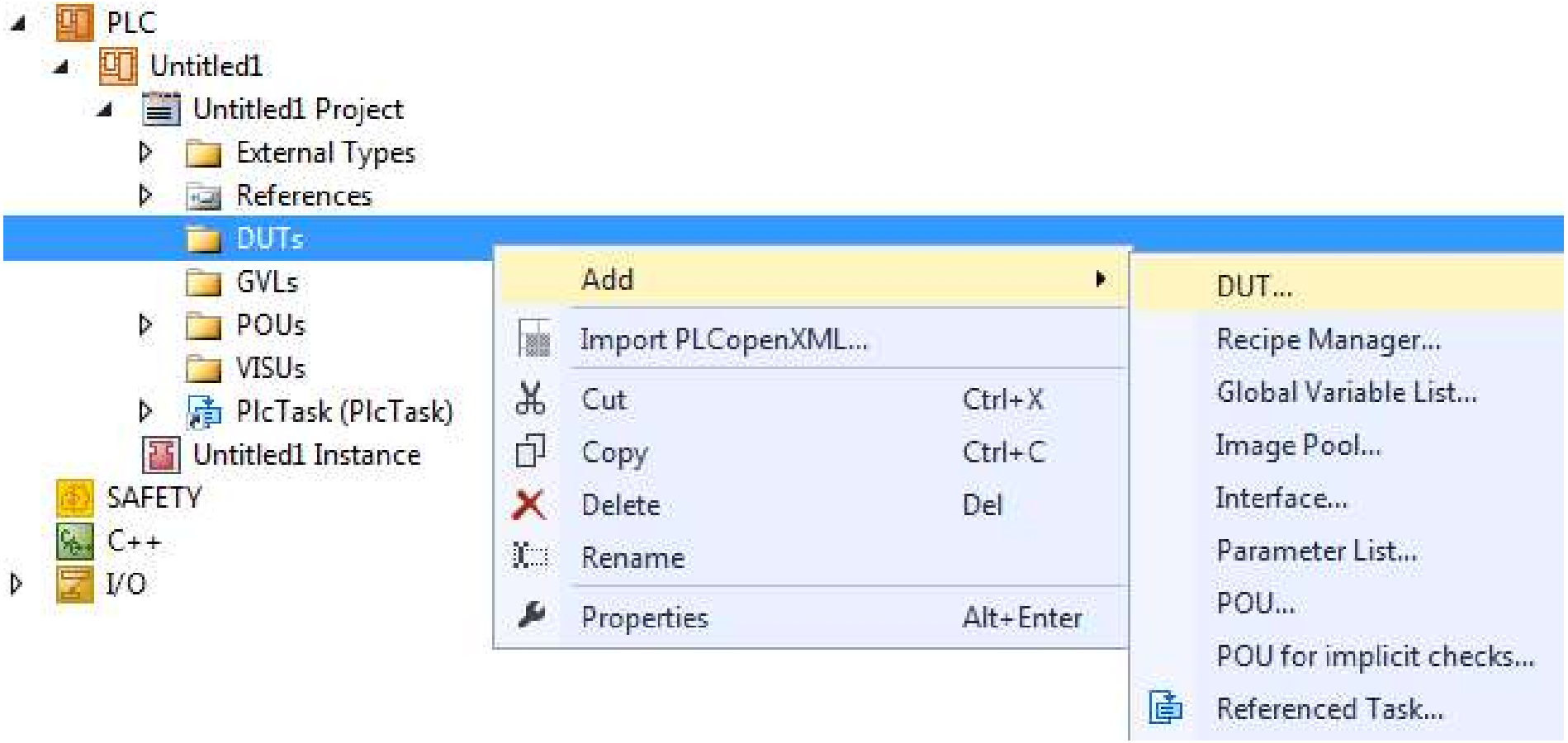

BECKHOFF
BECKHOFF
EK1100
TwinCAT
Version 3
BECKHOFF
BECKHOFF

Enumerations

- Enumerations can be used to assign a variable name to a number
- Enumerations can be used in two different ways; with or without being instantiated
- If the Enumeration is not instantiated then the Enumeration works similar to a list of constants
- If the Enumeration is instantiated then the instance of the enumeration will hold the variable of the current value of the Enumeration

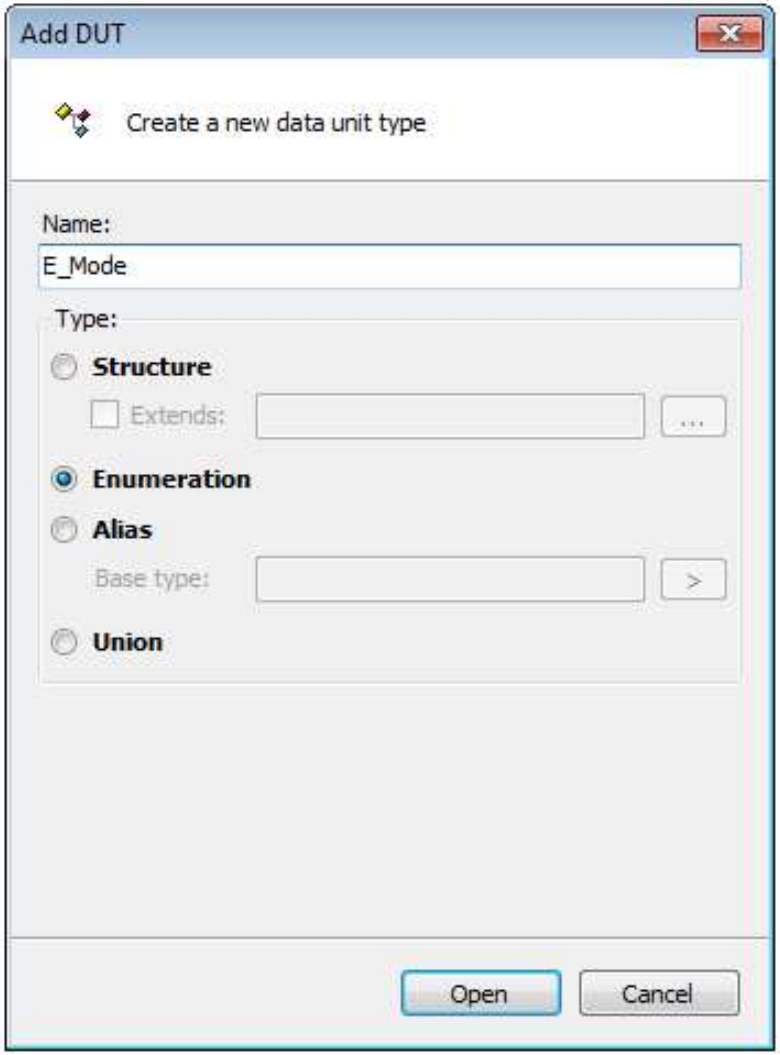
Enumerations

- To add an Enumeration, ‘Right-Click’ on the ‘DUTs’ folder
  - Then Select ‘Add’ -> ‘DUT’



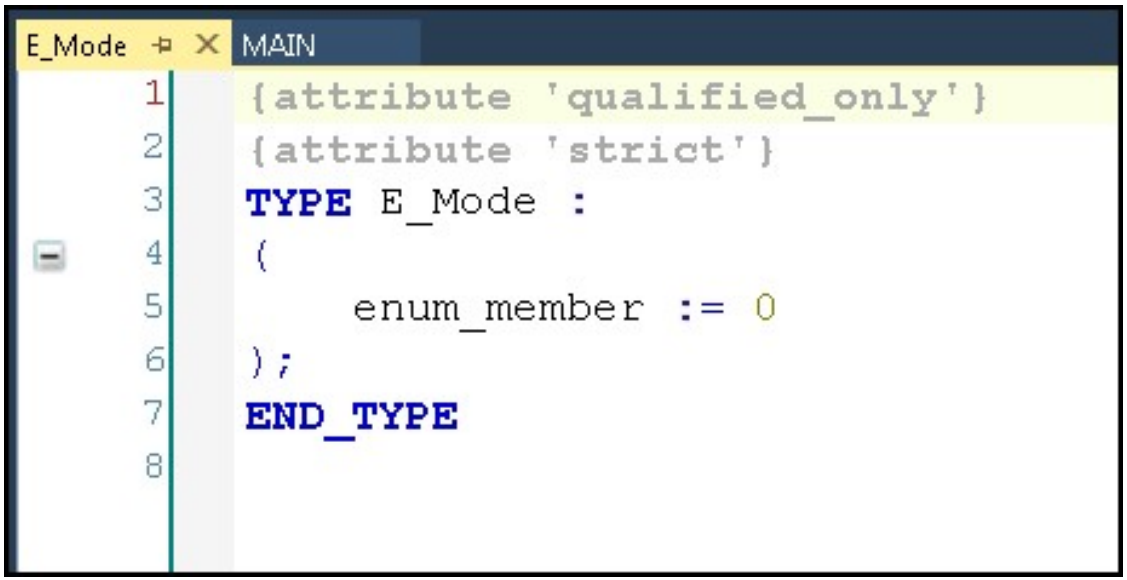
Enumerations

- Change the Name to E_Mode
- Select the Radio button for Enumeration
- Click the ‘Open’ button



Enumerations

- The basic framework of an Enumeration looks like the following



Enumerations

- When the Enumeration is defined the first variable in the list will be assigned a value of 0, the variables following will be assigned their values in ascending order
- Manual = 0
- Semi_Auto = 1
- Auto = 2

E_Mode* MAIN

```
1  {attribute 'qualified_only'}
2  {attribute 'strict'}
3  TYPE E_Mode :
4  (
5      Manual,
6      Semi_Auto,
7      Auto
8  );
9  END_TYPE
```

Enumerations

- If a variable in the list is explicitly assigned a value then the following variables will be incremented from this value
- Manual = 1
- Semi_Auto = 2
- Auto = 3

E_Mode* | MAIN

```
1  {attribute 'qualified_only'}
2  {attribute 'strict'}
3  TYPE E_Mode :
4  (
5      Manual := 1,
6      Semi_Auto,
7      Auto
8  );
9  END_TYPE
```

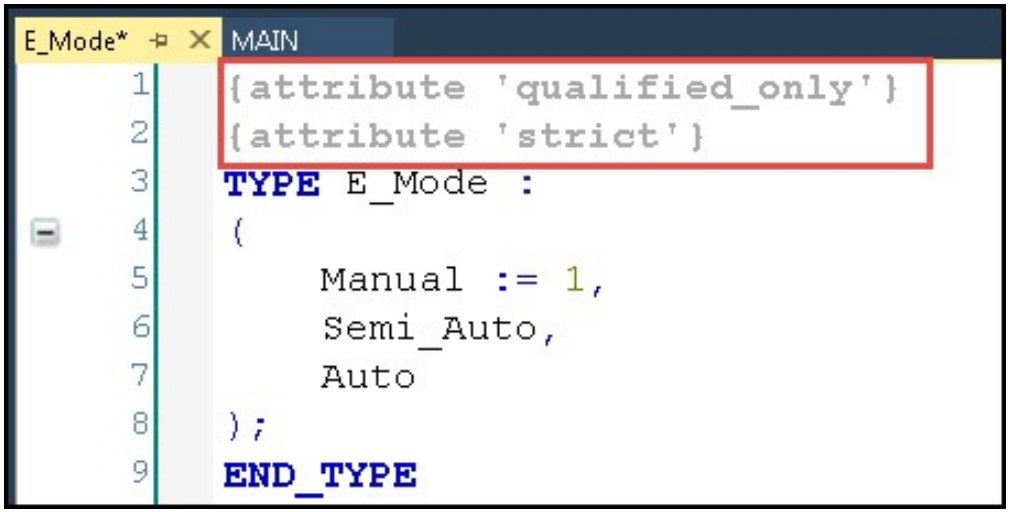
- Manual = 1
- Semi_Auto = 2
- Auto = 3
- Maintenance = 10
- Unknown = 11

E_Mode* | MAIN

```
1  {attribute 'qualified_only'}
2  {attribute 'strict'}
3  TYPE E_Mode :
4  (
5      Manual := 1,
6      Semi_Auto,
7      Auto,
8      Maintenance := 10,
9      Unknown
10 );
11 END_TYPE
```

Enumerations

- The gray texts included are called Pragmas
- The Pragmas influence the properties of the defined enumerations and affect how they are used within POUs
- If the programmer does not want the Pragmas, they can be commented out or simply deleted*



*The Pragmas are included by default and are recommended to add traceability; but not required

Enumerations

- The ‘qualified_only’ attribute Pragma requires the enumeration to be called by its Enumeration DUT name when being used within another POU

E_Mode* | MAIN

```
1  {attribute 'qualified_only'}
2  {attribute 'strict'}
3  TYPE E_Mode :
4  (
5      Manual := 1,
6      Semi_Auto,
7      Auto
8  );
9  END_TYPE
```

MAIN* | E_Mode*

```
1  PROGRAM MAIN
2  VAR
3  END_VAR
```

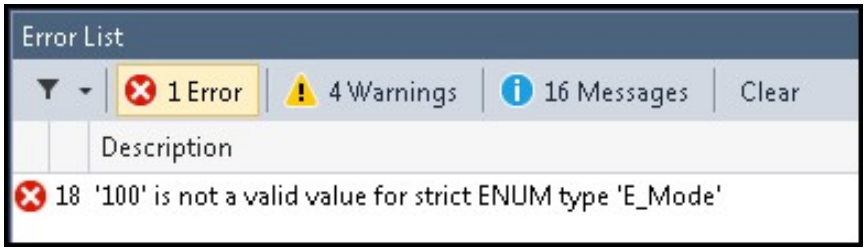
```
1  E_Mode.Manual;   //Valid Syntax
2
3  Manual;   //Invalid Syntax due to the Pragma
4
```

Enumerations

- The ‘strict’ attribute Pragma prevents alternate values from being assigned and any arithmetic operation with variables of the enumeration type

E_Mode* | MAIN

```
1  {attribute 'qualified_only'}
2  {attribute 'strict'}
3  TYPE E_Mode :
4  (
5      Manual := 1,
6      Semi_Auto,
7      Auto
8  );
9  END_TYPE
```



MAIN | E_Mode

```
1  PROGRAM MAIN
2  VAR
3  END_VAR
```

```
1  E_Mode.Manual := 100;   //Invalid Syntax
2
```

Enumerations

- Using Enumerations allows for easier understanding of the code when it is being read
- In this Case statement the Variables in the Enumeration are used to represent the number equivalent of iStep.
- As iStep changes in value the Case statement will change states
- iStep can be assigned a numerical value or an Enumeration variable
- The displayed online value of iStep1 will always be an INT value because iStep1 is declared as an INT

E_Mode | MAIN*

```
1   PROGRAM MAIN
2   VAR
3       iStep : INT;
4   END_VAR
```

```
1
2   CASE iStep OF
3
4   Manual:
5       // Place Code Here
6       iStep := 2;
7
8   Semi_Auto:
9       // Place Code Here
10      iStep := Auto;
11  Auto:
12      (* Place Code Here*);
13
14  END_CASE
15
```

Enumerations

- The displayed online value of iStep will always be an INT value because iStep is declared as an INT

```
1
2   CASE iStep 3 OF
3
4   Manual 1 :
5       // Place Code Here
6       iStep 3 := 2;
7
8   Semi_Auto 2 :
9       // Place Code Here
10      iStep 3 := Auto 3 ;
11  Auto 3 :
12      (* Place Code Here*);
13
14  END_CASE
15
```

Enumerations

- If an instance of the Enumeration is declared then the instance of the Enumeration holds the variable name of the value of the Enumeration

E_Mode | MAIN*

```
1  PROGRAM MAIN
2  VAR
3      eStep : E_Mode;
4  END_VAR
5
6
```

```
1
2  CASE eStep OF
3
4  Manual:
5      // Place Code Here
6      eStep := 2;
7
8  Semi_Auto:
9      // Place Code Here
10     eStep := Auto;
11 Auto:
12     (* Place Code Here*);
13
14 END_CASE
15
```

Enumerations

- If eStep is of type E_Mode then eStep holds either ‘Manual’, ‘Semi_Auto’, or ‘Auto’

E_Mode | MAIN [Online]*

TwinCAT_Device.Untitled1.MAIN

Expression	Type	Value	Prepared value
eStep	E_MODE	Auto	

```
1
2   CASE eStep [Auto] OF
3
4   Manual [1]:
5       // Place Code Here
6       eStep [Auto] := 2;
7
8   Semi_Auto [2]:
9       // Place Code Here
10      eStep [Auto] := Auto [3];
11  Auto [3]:
12      (* Place Code Here*);
13
14  END_CASE
15
```

Contact

BECKHOFF

Beckhoff Automation LLC
USA Headquarters
13130 Dakota Avenue
Savage, MN 55378
USA

Phone: 1 952 890 0000
Fax: 1 952 890 2888
E-Mail: beckhoff.usa@beckhoff.com
Web: www.beckhoffautomation.com

Beckhoff Automation GmbH & Co. KG
Global Headquarters
Huelshorstweg 20
33415 Verl
Germany

Phone: +49 5246 963-0
Fax: +49 5246 963-198
E-Mail: info@beckhoff.com
Web: www.beckhoff.com