

BECKHOFF



CASE Instruction

- A CASE instruction is used when making a decision based on an integer value
- Typically a CASE instruction is used to control the state of the machine, or of a sequential process

- In the below example 'iStep' is the variable that will be evaluated to determine which section of code to run

```
CASE iStep OF
```

```
0: (*Run this code when iStep = 0*);
```

```
1: (*Run this code when iStep = 1*);
```

```
2: (*Run this code when iStep = 2*);
```

```
ELSE (*Run this code when iStep = any other value*);
```

```
END_CASE
```

- It is important to understand that only one step is executed during a PLC cycle
- In Step 0 if xStep is TRUE the iStep will be incremented by a value of 1.
- Step1 will not run until the next PLC cycle

```
CASE iStep OF

0: (*Run this code when iStep = 0*);
    IF xStep THEN
        iStep := iStep + 1;
    END_IF
1: (*Run this code when iStep = 1*);

2: (*Run this code when iStep = 2*);

ELSE (*Run this code when iStep = any other value*);

END_CASE
```

CASE Instruction

- When the CASE instruction is reached the value of iStep is evaluated

```
1 CASE iStep OF
2
3 0: (*Run this code when iStep = 0*);
4     IF xStep THEN
5         iStep := iStep + 1;
6     END_IF
7 1: (*Run this code when iStep = 1*);
8     IF xStep THEN
9         iStep := iStep + 1;
10    END_IF
11 2: (*Run this code when iStep = 2*);
12     IF xStep THEN
13         iStep := iStep + 1;
14    END_IF
15 ELSE (*Run this code when iStep = any other value*);
16     IF xStep THEN
17         iStep := iStep + 1;
18    END_IF
19 END_CASE
```

CASE Instruction

BECKHOFF

- When iStep is equal to 0 this code will be executed

```
1 CASE iStep OF
2
3 0: (*Run this code when iStep = 0*);
4   IF xStep THEN
5     iStep := iStep + 1;
6   END_IF
7 1: (*Run this code when iStep = 1*);
8   IF xStep THEN
9     iStep := iStep + 1;
10  END_IF
11 2: (*Run this code when iStep = 2*);
12   IF xStep THEN
13     iStep := iStep + 1;
14   END_IF
15 ELSE (*Run this code when iStep = any other value*);
16   IF xStep THEN
17     iStep := iStep + 1;
18   END_IF
19 END_CASE
```

CASE Instruction

BECKHOFF

- When iStep is equal to 1 this code will be executed

```
1 CASE iStep OF
2
3 0: (*Run this code when iStep = 0*);
4   IF xStep THEN
5     iStep := iStep + 1;
6   END_IF
7 1: (*Run this code when iStep = 1*);
8   IF xStep THEN
9     iStep := iStep + 1;
10  END_IF
11 2: (*Run this code when iStep = 2*);
12   IF xStep THEN
13     iStep := iStep + 1;
14   END_IF
15 ELSE (*Run this code when iStep = any other value*);
16   IF xStep THEN
17     iStep := iStep + 1;
18   END_IF
19 END_CASE
```

CASE Instruction

BECKHOFF

- When iStep is equal to 2 this code will be executed

```
1 CASE iStep OF
2
3 0: (*Run this code when iStep = 0*);
4     IF xStep THEN
5         iStep := iStep + 1;
6     END_IF
7 1: (*Run this code when iStep = 1*);
8     IF xStep THEN
9         iStep := iStep + 1;
10    END_IF
11 2: (*Run this code when iStep = 2*);
12     IF xStep THEN
13         iStep := iStep + 1;
14     END_IF
15 ELSE (*Run this code when iStep = any other value*);
16     IF xStep THEN
17         iStep := iStep + 1;
18     END_IF
19 END_CASE
```

CASE Instruction

- When iStep is equal to any other value this code will be executed

```
1 CASE iStep OF
2
3 0: (*Run this code when iStep = 0*);
4     IF xStep THEN
5         iStep := iStep + 1;
6     END_IF
7 1: (*Run this code when iStep = 1*);
8     IF xStep THEN
9         iStep := iStep + 1;
10    END_IF
11 2: (*Run this code when iStep = 2*);
12    IF xStep THEN
13        iStep := iStep + 1;
14    END_IF
15 ELSE (*Run this code when iStep = any other value*);
16     IF xStep THEN
17         iStep := iStep + 1;
18     END_IF
19 END_CASE
```

CASE Instruction

- After the step is executed the code will continue after the END_CASE

```
1 CASE iStep OF
2
3 0: (*Run this code when iStep = 0*);
4     IF xStep THEN
5         iStep := iStep + 1;
6     END_IF
7 1: (*Run this code when iStep = 1*);
8     IF xStep THEN
9         iStep := iStep + 1;
10    END_IF
11 2: (*Run this code when iStep = 2*);
12    IF xStep THEN
13        iStep := iStep + 1;
14    END_IF
15 ELSE (*Run this code when iStep = any other value*);
16     IF xStep THEN
17         iStep := iStep + 1;
18     END_IF
19 END_CASE
```

- It is possible for steps to be combined
 - When step numbers are separated by a , each number will be evaluated individually
 - When step numbers are separated using .. The range of numbers will be evaluated
-
- Overlapping/Repeating step numbers will generate a compile error

CASE Instruction

BECKHOFF

- In the below example 0, 5, and 42 will have a common execution

```
CASE iStep OF
0,5,42: (*Run this code when iStep = 0*);
  IF xStep THEN
    iStep := iStep + 1;
  END_IF
1: (*Run this code when iStep = 1*);
  IF xStep THEN
    iStep := iStep + 1;
  END_IF
2..4: (*Run this code when iStep = 2*);
  IF xStep THEN
    iStep := iStep + 1;
  END_IF
ELSE (*Run this code when iStep = any other value*);
  IF xStep THEN
    iStep := iStep + 1;
  END_IF
END_CASE
```

CASE Instruction

BECKHOFF

- In the below example the values of 2,3 and 4 will have a common execution

```
CASE iStep OF  
  
0,5,42: (*Run this code when iStep = 0*);  
    IF xStep THEN  
        iStep := iStep + 1;  
    END_IF  
  
1: (*Run this code when iStep = 1*);  
    IF xStep THEN  
        iStep := iStep + 1;  
    END_IF  
  
2..4: (*Run this code when iStep = 2*);  
    IF xStep THEN  
        iStep := iStep + 1;  
    END_IF  
  
ELSE (*Run this code when iStep = any other value*);  
    IF xStep THEN  
        iStep := iStep + 1;  
    END_IF  
  
END_CASE
```

Beckhoff Automation GmbH & Co. KG

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

© Beckhoff Automation GmbH & Co. KG

All images are protected by copyright. The use and transfer to third parties is not permitted.

Beckhoff®, TwinCAT®, EtherCAT®, Safety over EtherCAT®, TwinSAFE®, XFC® and XTS® are registered trademarks of and licensed by Beckhoff Automation GmbH. Other designations used in this presentation may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

The information provided in this presentation contains merely general descriptions or characteristics of performance which in case of actual application do not always apply as described or which may change as a result of further development of the products. An obligation to provide the respective characteristics shall only exist if expressly agreed in the terms of contract.

Beckhoff Automation GmbH & Co. KG

Unternehmenszentrale
Hülshorstweg 20
33415 Verl
Deutschland

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

© Beckhoff Automation GmbH & Co. KG

Alle Bilder sind urheberrechtlich geschützt. Die Weitergabe und Nutzung durch Dritte ist nicht gestattet.

Beckhoff®, TwinCAT®, EtherCAT®, Safety over EtherCAT®, TwinSAFE®, XFC® und XTS® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH. Die Verwendung anderer in dieser Präsentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Kennzeichen führen.

Die Informationen in dieser Präsentation enthalten lediglich allgemeine Beschreibungen bzw. Leistungsmerkmale, welche im konkreten Anwendungsfall nicht immer in der beschriebenen Form zutreffen bzw. welche sich durch Weiterentwicklung der Produkte ändern können. Die gewünschten Leistungsmerkmale sind nur dann verbindlich, wenn sie bei Vertragsabschluss ausdrücklich vereinbart werden.