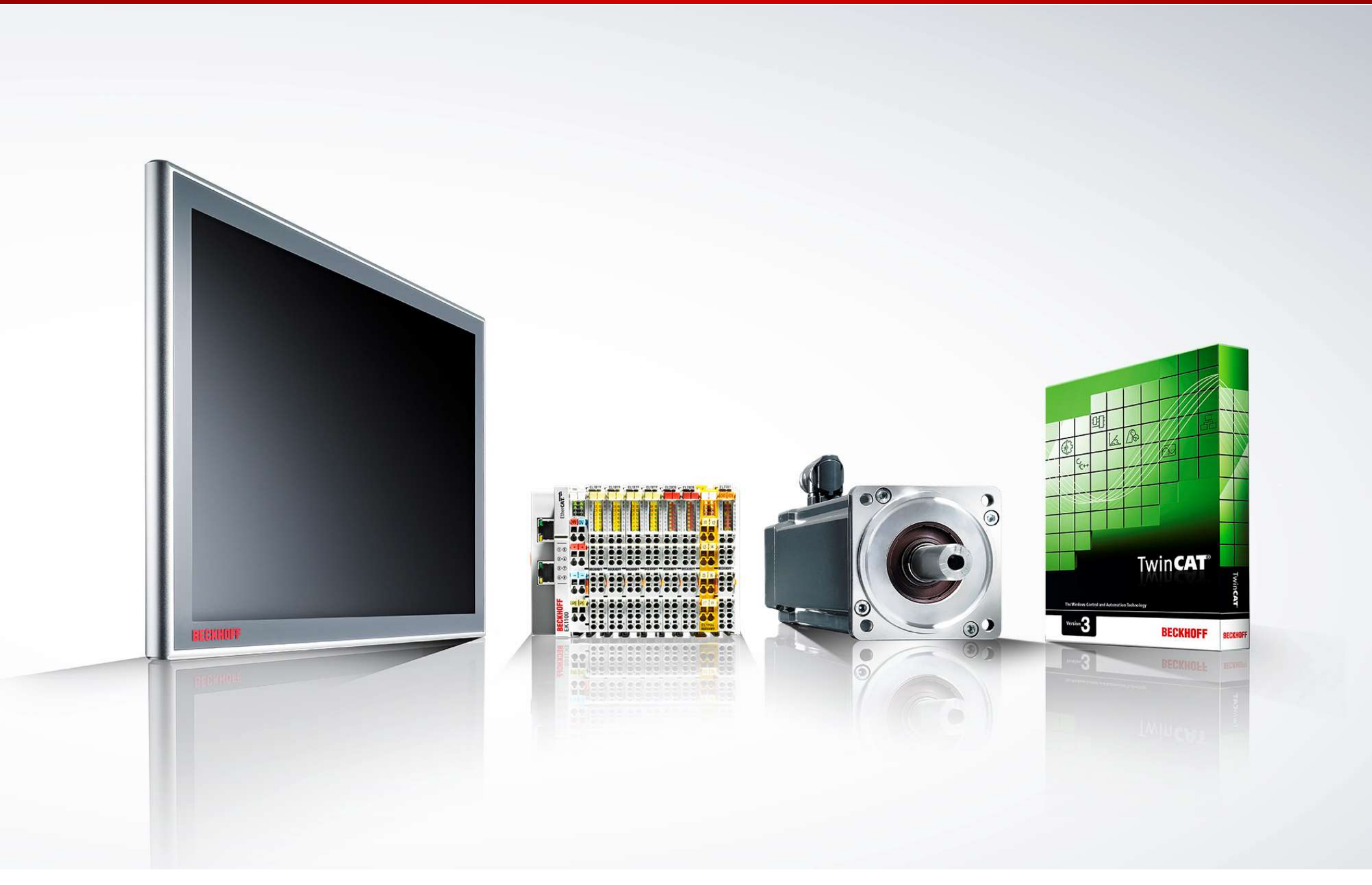
BECKHOFF


BECKHOFF
BECKHOFF
EK1100
TwinCAT
Version 3
BECKHOFF
BECKHOFF

Calling FBs with a CASE Instruction

Calling FBs with a CASE Instruction

BECKHOFF

- When calling Function Blocks from a CASE instruction it is important to remember that only one step is run per PLC cycle.

Calling FBs with a CASE Instruction

BECKHOFF

- In the below code step 0 will Initialize the timer
- Step 1 sets the input to TRUE and sets iStep to a value of 2
- Step 2 is monitoring the output of the timer, however the timer itself is never being called and will no longer update

```
CASE iStep OF

0:(*Init*);
    fbTON(IN:=FALSE);
1:(*Start TON*);
    fbTON(IN:=TRUE, PT:=T#1S);
    iStep := 2;
2:(*Wait*);
    IF fbTON.Q THEN
        xTimerComplete := TRUE;
    END_IF
END_CASE
```

Calling FBs with a CASE Instruction

BECKHOFF

- The function block instance along with the parentheses () must called in each step that uses the function block.

```
CASE iStep OF

0:(*Init*);
    fbTON(IN:=FALSE);
1:(*Start TON*);
    fbTON(IN:=TRUE, PT:=T#1S);
    iStep := 2;
2:(*Wait*);
    fbTON();
    IF fbTON.Q THEN
        xTimerComplete := TRUE;
    END_IF
END_CASE
```

Calling FBs with a CASE Instruction

BECKHOFF

- Another option is to call the function block from outside of the CASE instruction
- In this example the timer is called every PLC cycle
- The steps in the CASE instruction simply control a variable that is evaluated by the timer

```
CASE iStep OF

0:(*Init*);
    xStartTimer := FALSE;
1:(*Start TON*);
    xStartTimer := TRUE;
    iStep := 2;
2:(*Wait*);
    IF fbTON.Q THEN
        xTimerComplete := TRUE;
    END_IF
END_CASE

fbTON(IN:=xStartTimer, PT:=T#1S);
```

Calling FBs with a CASE Instruction

BECKHOFF

- Calling a Function Block outside of a CASE instruction is slightly less efficient.
- When the Function Block is outside of a CASE instruction, it is updated every PLC cycle.
- When the Function Block is only called from within a CASE instruction then it will only be updated when the step is active.
- Calling the Function Block from within a CASE instruction requires careful programming practices to ensure that the Function Block is updated properly.

Calling FBs with a CASE Instruction

BECKHOFF

- When calling a Function Block from within a CASE instruction it is important to call the FB in each step as needed.

Calling FBs with a CASE Instruction

BECKHOFF

- The Execute input of a Function Block from the Tc2_MC2.lib requires a rising edge for proper operation.

```
CASE iStep OF

0: (*Init*);
    fbMC_MoveRel(Axis:= Axis1,Execute:= FALSE);
10: (*Move*);
    fbMC_MoveRel(Axis:= Axis1,Execute:= FALSE);
    fbMC_MoveRel(Axis:= Axis1,
        Execute:= TRUE,
        Distance:= 100.0,
        Velocity:= 10.0);
    iStep := iStep + 1;
11: (*Wait Complete*);
    fbMC_MoveRel(Axis:= Axis1,Execute:= FALSE);
    IF NOT fbMC_MoveRel.Error THEN
        IF fbMC_MoveRel.Done THEN
            fbMC_MoveRel(Axis:= Axis1,Execute:= FALSE);
            iStep := 0;
        END_IF
    ELSE
        iErrStep := iStep;
        iStep := 999;
    END_IF
999: (*Error Step*);
    fbMC_MoveRel(Axis:= Axis1,Execute:= FALSE);
    iStep := 0;
ELSE
    iErrStep := iStep;
    iStep := 999;
END_CASE
```

Calling FBs with a CASE Instruction

BECKHOFF

- To ensure the rising edge is handled properly the FB is first called with a FALSE at the Execute and then again called with a TRUE at the Execute.
- iStep is immediately and unconditionally changed.

```
CASE iStep OF

0:(*Init*);
    fbMC_MoveRel(Axis:= Axis1,Execute:= FALSE);
10:(*Move*);
    fbMC_MoveRel(Axis:= Axis1,Execute:= FALSE);
    fbMC_MoveRel(Axis:= Axis1,
        Execute:= TRUE,
        Distance:= 100.0,
        Velocity:= 10.0);
    iStep := iStep + 1;
11:(*Wait Complete*);
    fbMC_MoveRel(Axis:= Axis1,Execute:= FALSE);
    IF NOT fbMC_MoveRel.Error THEN
        IF fbMC_MoveRel.Done THEN
            fbMC_MoveRel(Axis:= Axis1,Execute:= FALSE);
            iStep := 0;
        END_IF
    ELSE
        iErrStep := iStep;
        iStep := 999;
    END_IF
999:(*Error Step*);
    fbMC_MoveRel(Axis:= Axis1,Execute:= FALSE);
    iStep := 0;
ELSE
    iErrStep := iStep;
    iStep := 999;
END_CASE
```

Calling FBs with a CASE Instruction

BECKHOFF

- The FB is called again with the Execute set to FALSE, this calls the Function Block so that it will be updated in the step, and also resets the Execute
- The FB is then monitored for both an Error status and a Done status.

```
CASE iStep OF

0:(*Init*);
    fbMC_MoveRel(Axis:= Axis1,Execute:= FALSE);
10:(*Move*);
    fbMC_MoveRel(Axis:= Axis1,Execute:= FALSE);
    fbMC_MoveRel(Axis:= Axis1,
        Execute:= TRUE,
        Distance:= 100.0,
        Velocity:= 10.0);
    iStep := iStep + 1;
11:(*Wait Complete*);
    fbMC_MoveRel(Axis:= Axis1,Execute:= FALSE);
    IF NOT fbMC_MoveRel.Error THEN
        IF fbMC_MoveRel.Done THEN
            fbMC_MoveRel(Axis:= Axis1,Execute:= FALSE);
            iStep := 0;
        END_IF
    ELSE
        iErrStep := iStep;
        iStep := 999;
    END_IF
999:(*Error Step*);
    fbMC_MoveRel(Axis:= Axis1,Execute:= FALSE);
    iStep := 0;
ELSE
    iErrStep := iStep;
    iStep := 999;
END_CASE
```

Contact

BECKHOFF

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

Kontakt

BECKHOFF

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