TRIO
MOTION | TECHNOLOGY

Doc No.:	**AN-285**
Version:	**1.0**
Date:	**27 September 2012**
Subject:	**Drive Homing for CT EtherCAT**

APPLICATION NOTE

1. Introduction

Many servo drives have built-in homing or datuming routines. When connected via a digital bus system, these routines can be used to home the axis. The use of the Emerson Control Techniques EtherCAT module for the Digitax and Unidrive SP is described in this application note.

2. System

The program described here requires a MC464 with P876 EtherCAT module connected to a Control Techniques SM-EtherCAT module on a suitable servo drive.

It is assumed that the MC464 firmware has brought the EtherCAT connection up to Operational State and that the drive is in Cyclic Sync Position mode.

3. The sequence

The homing sequence uses the DS402 Drive Controlled Homing defined for CanOpen-over-EtherCAT. For a full description of the actual motion sequence, see the Control Techniques SM EtherCAT Users Guide. The Homing modes used in this example are described on page 57 of issue 3. That is homing to the marker pulse. If other modes are required, then it is possible to put the required mode number in the program and run more-or-less the same sequence.

3.1. User Settings

Choose the homing method, speed and acceleration:

```
' Home counter-clockwise
ct_homing_method = 33
' Home clockwise
ct_homing_method = 34

ct_homing_speed = 20000
ct_homing_accel = 100000
```

(Note that these values are in encoder counts. They do not use the MC464's UNITS parameter)

3.2. Intialisation

The subroutine "init_homing" sets 4 CoE objects in the drive to prepare it for the homing operation.

3.2.1. Set the homing method in CoE object $6098:$00.

3.2.2. Set the homing speed #1 in object $6099:$01.

3.2.3. Set the homing speed #2 in object $6099:$02.

3.2.4. Set the homing Acceleration in object $609A:$00.

3.3. Run the homing sequence

The main homing sub-routine starts the drive controlled homing operation and waits for the homing complete flag in the DRIVE_STATUS, or for a timeout to occur in case the drive does not complete the homing.

Note that the homing motion is controlled by the drive itself. The MC464 does not profile any movement during this method of homing.

Set SERVO OFF

Save the original normal control mode from CoE object $6061:$00

Put the cyclic control word under control of the BASIC program

Set the cyclic control word to value 6 (to disable drive)

Set the control mode object $6060:$00 to 6 for homing mode.

Check that the homing mode was accepted by reading $6061:$00.

Set the cyclic control word to $07 then to $3F to start homing

Wait for the home finished bits in the cyclic status word OR for a timeout.

Put the original control mode into object $6060:$00.

Set the drive control word mode back to system control.

Finally the axis is set to the correct MPOS with DEFPOS(ENCODER/UNITS). This division by UNITS is so that it works OK when UNITS is set to other values than 1.0.

4. Program listing

This is the complete example program.

```
'
****************************************************************************
' CT EtherCAT Homing Test Program
' Date: 24 Sept 2012
'
' Tested using CT DigitaxST Drive, MC464 Firmware version 2.0196
'
'
  WDOG=ON

'Program data
  ct_vr = 100
  comms = 0
  ct_ax = 3
  orig_control_mode = 8
  max_timeout = 10000
```

```
'Homing parameters (choose one)
' Home to Z mark counter-clockwise
  ct_homing_method = 33
' Home to Z mark clockwise
'ct_homing_method = 34

  ct_homing_speed = 200000
  ct_homing_accel = 10000000

  drive_homed = FALSE

'begin

  GOSUB init_homing
  GOSUB home_axis

  PRINT#comms,"Program completed"
  PRINT#comms,""

  PRINT#comms,"MPOS: "; MPOS[2], "Encoder",ENCODER[0]


  STOP


'
****************************************************************************
'
init_homing:
  'Set homing method.
  IF NOT CO_WRITE_AXIS(ct_ax,$6098,0,5,-1,ct_homing_method) THEN
    PRINT#comms,"Err: failed to set homing method."
    STOP
  ENDIF

  'Set speed used when searching for switch.
  IF NOT CO_WRITE_AXIS(ct_ax,$6099,1,7,-1,ct_homing_speed) THEN
    PRINT#comms,"Err: failed to set homing speed(1)."
    STOP
  ENDIF

  'Set speed used when searching for zero mark.
  IF NOT CO_WRITE_AXIS(ct_ax,$6099,2,7,-1,ct_homing_speed) THEN
    PRINT#comms,"Err: failed to set homing speed(2)."
    STOP
  ENDIF

  'Set homing accel
  IF NOT CO_WRITE_AXIS(ct_ax,$609a,0,7,-1,ct_homing_accel) THEN
    PRINT#comms,"Err: failed to set homing accel."
    STOP
  ENDIF

  'Note that 'homing offset' $607c is also available

RETURN

'
****************************************************************************
'
home_axis:
  drive_homed = FALSE

  BASE(ct_ax)

  ' Ensure servo is off
```

```
SERVO = OFF

'Record original control mode
IF NOT CO_READ_AXIS(ct_ax,$6061,0,5,ct_vr) THEN
  PRINT#comms,"Err: failed to read control mode."
  STOP
ENDIF
orig_control_mode = VR(ct_vr)

'Put controlword under user control
DRIVE_CW_MODE = 1

'Set control word
DRIVE_CONTROLWORD = 6
WA(10)

'Set control mode
IF NOT CO_WRITE_AXIS(ct_ax,$6060,0,5,-1,6) THEN
  PRINT#comms,"Err: failed to set homing mode."
  STOP
ENDIF
' and verify
REPEAT
  CO_READ_AXIS(ct_ax,$6061,$00,5,ct_vr)
UNTIL VR(ct_vr)=6

'Enable Voltage
DRIVE_CONTROLWORD = 6
'Wait for 'ready to switch on' status
REPEAT
  VR(ct_vr) = DRIVE_STATUS
  WA(1)
UNTIL (VR(ct_vr).5 = 1) AND (VR(ct_vr).0 = 1)

'Enable 'switch on'
DRIVE_CONTROLWORD = 7
'Wait for 'switched on' status
REPEAT
  VR(ct_vr) = DRIVE_STATUS
  WA(1)
UNTIL (VR(ct_vr).5 = 1)AND (VR(ct_vr).1 = 1) AND (VR(ct_vr).0 = 1)
WA(100)

'Set 'Enable Operation' and start homing
IF verbose THEN PRINT#comms,"Start homing axis"
DRIVE_CONTROLWORD = $3f

'wait until drive homed
TICKS = 0
REPEAT
  VR(ct_vr) = DRIVE_STATUS
  WA(1)
UNTIL VR(ct_vr).12 = 1 OR TICKS < -max_timeout

IF VR(ct_vr).12 = 1 THEN
  TICKS = 0
  REPEAT
    VR(ct_vr) = DRIVE_STATUS
    WA(1)
  UNTIL VR(ct_vr).10 = 1 OR TICKS < -max_timeout
ENDIF

IF (DRIVE_STATUS AND $1400) = $1400 THEN
  PRINT#comms,"Drive homed successfully."
  drive_homed = TRUE
ELSE
```

```
        PRINT#comms,"Failed to home drive."
    ENDIF

    'Reset control mode back to original
    IF NOT CO_WRITE_AXIS(ct_ax,$6060,0,5,-1,orig_control_mode) THEN
      PRINT#comms,"Err: failed to reset control mode."
      STOP
    ENDIF
    ' and verify
    REPEAT
      CO_READ_AXIS(ct_ax,$6061,$00,5,ct_vr)
    UNTIL VR(ct_vr)=orig_control_mode

    ' Set the MPOS value to same as ENCODER
    DEFPOS(ENCODER/UNITS)
    WA(2)

    SERVO=ON
    WDOG=ON

    'Return controlword to firmware control
    DRIVE_CW_MODE = 0

RETURN
```