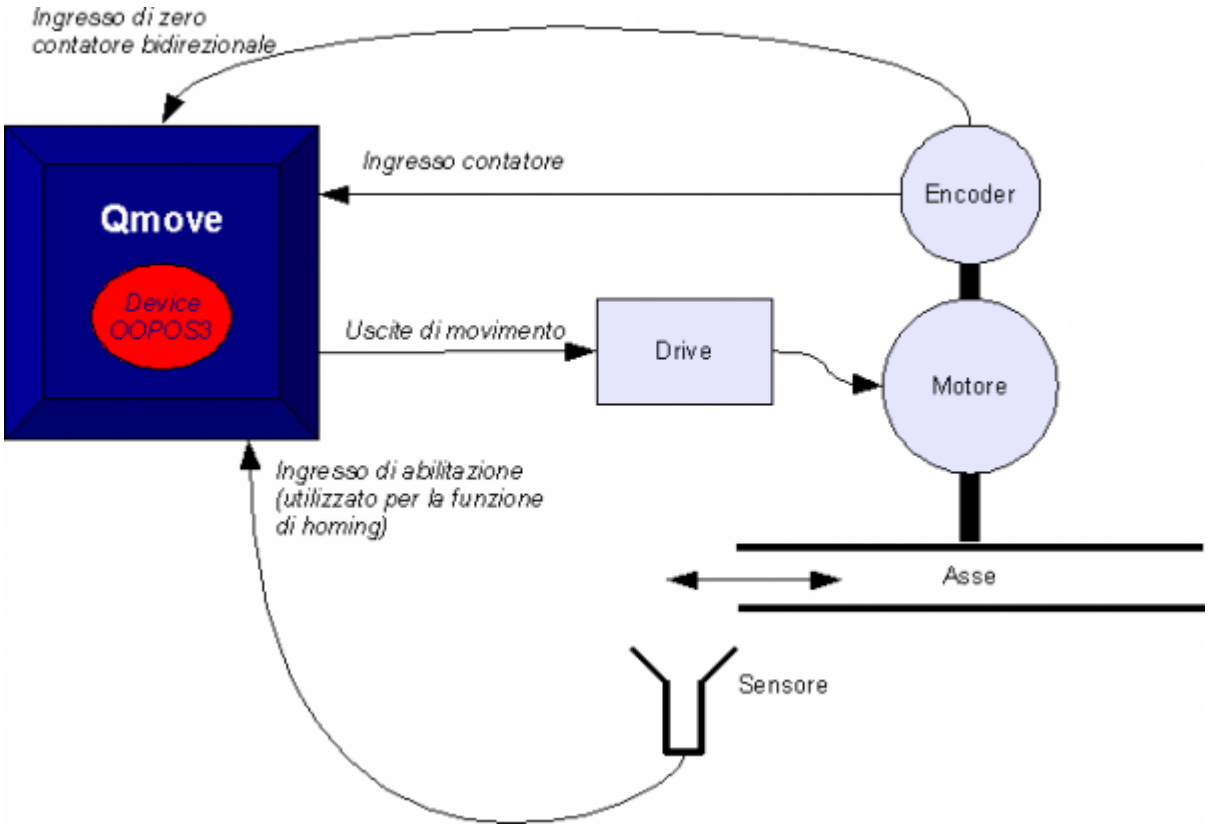
Table of Contents

AN009 - Example of using and calibrating the OOPOS3 device

In this section you want to describe the first operations that you will be doing the user in his first approach with the EANPOS device. You also want to provide a simple example where using the OOPOS3 device to implement a positioner.

We can divide the procedure into the following sections:

- device declaration in the configuration unit
- introduction of parameters to calibrate inputs and outputs
- application Development as required



Device declaration in the configuration unit

As has already been explained in the Description section of the OOPOS3 device, You must program correctly the configuration unit of the application. It is very important that the code portion where device is declared, here you will need to indicate the hardware resources to be used to ensure proper operation. It will be the task of the programmer to identify and choose the most appropriate inputs and outputs. For example, with the following line of code:

```
;---------------------------------
; Internal device declaration
;---------------------------------
INTDEVICE
Axis  OOPOS3  0004  2.CNT01 1 3.INP01 2.OUT01 2.OUT02 X.X X.X 2.OUT3  X.X
```

You define a OOPOS3 device with the name “axis” whose sampling time is 4 ms. Have been declared the following hardware resources: the input of the bidirectional counter, the digital input number for interruption dedicated to the encoder zero pulse, a digital input of enabling for the acquisition of the zero pulse. The movement outputs that have been declared are: 2.OUT01 as forward output, 2.OUT02 as backward output and 2.OUT03 as slow-down output.

An application that only has the device declaration within the configuration unit and a QCL unit that does not execute any operation (Except WAIT forced) It already allows you to do the first steps using the features of the device. After you download the application on the instrument and having it run, you will can already change the parameters, observe the states or give commands to the device via the monitor provided by Qview.
This is very convenient in the early stages of programming when you just want to check for some operations or when you're debugging.

Connecting the hardware

The OOPOS3 device requires some hardware resources such as an input counter for two-way transducer (CNTxx) and the digital

outputs.

Verifying how the Counter resource works

The following procedure is used to verify the operation of the counter input.

- Initialize the Axis device with the *INIT* command

```
INIT Axis
```

- Check *st_init* status activation

```
WAIT Axis:st_init
```

- Enter the values for *measure* and *pulse*

```
Axis:measure = 1000
Axis:pulse = 4000
```

- Reset the value of the *posit* parameter

```
Axis:posit = 0
```

- Move the Axis forward Running a complete turn to the encoder: verify that the value of the *Axis:posit* variable is positive and corresponds to the number of pulse/Rev of the encoder (with the value of *measure* and *pulse* above recommended)
- If the value of *Axis:posit* is negative, swap the encoder phases or give a *CNTREV* command
- If the value of *Axis:posit* remains at zero, Check the electrical connections or the *st_cntlock* flag

Verifica del funzionamento delle uscite digitali

Attention:
Before handling the axis, verify proper operation of emergency equipment and protection.

The following procedure is used to verify the operation of the digital outputs of forward, backward and slow-down moving the axis with manual device controls.
To continue, verify that the Axis device is initialized and with the *measure* and *pulse* value correct.

- Set the device software limits to the maximum value.
  Enter the 999999 value in the *maxpos* parameter and the -999999 value in the *minpos* parameter.

```
Axis:maxpos = 999999
Axis:minpos = -999999sse:posit = 0
```

- Give the *MANFFW* command to make turn the only forward output

```
MANFFW Axis
```

- To verify the correctly execution of the command, check that *st_still* = 0 and *st_movfwd* = 1

```
WAIT NOT Axis:st_still AND Axis:st_movfwd
```

- Check if the axis moves forward and that the count shown in *Asse:posit* increase, then stop the movement with the *STOP* command

```
STOP Axis
```

- If the forward output, corresponding for example to the 2.OUT01 resource, does not activate, check the electrical connection
- Give the MANFBW command To activate the backward output only

```
MANFBW Axis
```

- To verify that the command is running correctly, check that *st_still* = 0 and *st_movbwd* = 1

```
WAIT NOT Axis:st_still AND Axis:st_movbwd
```

- Check if the axle moves backward and that the count showed in *Axis:posit* decrements, quindi interrompere il movimento con il *STOP* comando.

```
STOP Axis
```

- If the backward output, for example the resource 2.OUT02, does not activate, check the electrical connection
- Give the MANSFW command to enable the forward outputs and slow-down

```
MANSFW Axis
```

- To verify that the command is running correctly, check that the *st_stil*l = 0, *st_movfwd* = 1 and *st_movslow* = 1

```
WAIT NOT Axis:st_still AND Axis:st_movfwd AND Axis:st_movslow
```

- Check if the axis moves forward at a speed lower than the previous and that the count showed in Axis:posit increase, then end the movement with the STOP command

```
STOP Axis
```

- If the slow-down output, for example the resource 2.OUT03, does not activate, check the electrical connection.

Correct parameterization of the device

After you correctly declare the hardware resources to use you must set some parameters based on the components that are attached to the Qmove product.

Measure and pulse introduction

Let us consider the case In which the bidirectional transducer is a digital encoder. Suppose that the encoder is directly keyed on a motor that must move the axis. You will need to set correctly the *measure* and *pulse*parameters of the device so that it can interpret the pulses arriving at QMove, the instrument will then calculate the position of the axis. The *measure* and *pulse* introduction Establishes a match between a space in a selected unit of measure and a certain number of pulses. In the event that the user already knows the path space in a encoder round Then you can proceed directly to the input of the values.
We clarify this concept with an example: If the encoder generates 1000 pulses turn and you know that the axis moves about 5 cm when the encoder execute a complete precisely round then you can enter the following values:

```
AxisX:measure = 50;
AxisX:pulse = 4000
```

The *measure* value introduced It also implies the choice of a unit of measure of millimeter to measure the positions, in the *pulse* parameter A value equal to the number of encoder pulses has been introduced multiplied x 4. You remember that the *measure/pulse* ratio must have a value between 0.00935 and 1. It is important to emphasize that the values just described are taken as a reference: it's not necessary to introduce the parameters taking as reference an encoder lap as you will describe below.

When the user does not know in advance the measurement parameters, you can still correct the calibration by following these steps:

- give the *INIT* command to device, check that the *st_init* state switch to 1
- through the “device monitor” of QView shown the *posit* parameter value on the PC
- set *measure* and *pulse* both to 1 value
- move the axis manually making it a move of an easily measurable position
- read the *posit* value
- now insert with the desired measurement unit the measured value in the *measure* parameter and the value of the *posit* parameter in the *pulse* parameter

The encoder resolution is now correctly set.

A further important task to do is set the *maxpos* and *minpos* parameters that respectively, define the maximum and minimum position to be reached by the axis.

Choosing the speed measurement unit

The unit of measurement of the instantaneous axis speed is chosen using the *unitvel* and *decpt* parameters. You can choose the speed time unit with the *unitvel* parameter: if this is 0 the speed is measured in Um/min, if is to 1 is measured in Um/s. The *decpt* parameter Instead it determines whether to measure the speed values in multiples of the fundamental unit of measure Um. For example, if the fundamental unit of measure Um=mm, and unitvel=1 You get the speed show in the vel in variable:

mm/s (with decpt = 0),
cm/s (with decpt = 1),
dm/s (with decpt = 2),
m/s (with decpt = 3).
Later, if necessary, you should to properly configure the display on the operator terminal to adjust the correct position of the decimal point.

Basic parameterizations

To operate the device OOPOS3 properly, you must enter some basic parameters.

- Determine the software limits to be introduced in the *maxpos* and *minpos* parameters

```
Axis:minpos = xxx (xxx = minimum value of the axis expressed in Um)
Axis:maxpos = yyy (yyy = maximum value of the axis expressed in Um)
```

- If you are using a two-speed system, set the required space for the axis to move from high speed to slow speed to the activation of the deceleration output; enter the data in the *slowpos*parameter

```
Axis:slowpos = zzz (zzz = slow-down value of the axis expressed in Um)
```

- Set the off time of the motion output when the axis enters the deceleration band so that the activation of the slow-down output does not cause electrical failure. Insert the data in the *slowdly* parameter.

```
Axis:slowdly = ttt (ttt = deceleration time expressed in s/100)
```

- Set the tolerance limits that you want to achieve during placement in the *tollp* and *tolln* parameters. As the first setting, introduce higher values to the required accuracy.

```
Axis:tollp = tpx (tpx = positive tolerance Value expressed in Um/10)
Axis:tolln = tnx (tpn = negative tolerance Value expressed in Um/10)
```

- Set the reverse time of axis in *tinv* parameter

```
Axis:tinv = inversion time expressed in s/100
```

- Consider a single inertia band throughout the axis, then set the *ninert* parameter to 1

```
Axis:ninert = 1
```

- Enable inertia recalculation When the positioning ends out tolerance: then set the *inertmode* parameter to 1

```
Axis:inertmode = 1
```

- Set the tolerance activation delay time, considering the time it takes the axle to decelerate until it stops; then set the *toldly* parameter

```
Axis:toldly = tdly (tdly = time delay activation tolerance expressed in ms)
```

Setting maxvel parameter

If you are not aware of the declared maximum motor speed, you must do so:

- enter in calibration mode (as described above)
- if the system allows it to supply to the drive the voltage and read the value of the *vel* parameter
- calculate the maximum speed with the proportion *vout* : 10 V = *vel* : *maxvel*

Now you can enter the maximum speed value in the *maxvel* parameter.

Handling

The procedures described here have allowed to complete the steps of defining the hardware resources necessary for the device, verification of electrical connections, setting the device's fundamental parameters.
Now it is possible to carry out a simple movement of the.

- Move the axis to a position so that it can make a certain space without meeting the maximum limit switches

- Reset the counter (*posit* = 0 parameter)
- Set the placement quota (*setpos* parameter)

```
Axis:setpos = Placement Quota (in Um Included between minpos and maxpos)
Start positioning (START command)
START Axis
```

- To interrupt the positioning use the STOP command

Development of an application that implements a positioner

In the previous section was explained what are the first steps to follow for an axis handling procedure. This example only uses a narrow spectrum of settable parameters of the device, In this section we insert a sample code, commented, from which the user can take a cue to develop an application.
The way the device should be declared is explained previously, Therefore, this section omits the configuration unit. See here.

```
;*******************************************************************************
; Configuration unit (are reported only the declarations of variables and
; the bus and device declarations are omitted)
;*******************************************************************************
;-------------------------------------------------------------------------------
; SYSTEM variables definition
;-------------------------------------------------------------------------------
SYSTEM
slQuotaPos L                    ;Variable for placement quota
;-------------------------------------------------------------------------------
; GLOBAL variables definition
;-------------------------------------------------------------------------------
GLOBAL
gfMovMan F                      ;Flag reporting manual movements in progress
gfMovAuto F                     ;Flag reporting automatic movements in progress
;-------------------------------------------------------------------------------
; INPUT variables definition
;-------------------------------------------------------------------------------
INPUT
ifAvMan F 2.INP01               ;Manual forward Input
ifInMan F 2.INP02               ;Manual backward input
ifStart F 2.INP03               ;START input axis
ifStop  F 2.INP04               ;STOP input axis
;-------------------------------------------------------------------------------
; OUTPUT variables definition
;-------------------------------------------------------------------------------
OUTPUT
ofToll F 2.OUT01                ;Axis Output in tolerance

;*******************************************************************************
; Unit gcl
;*******************************************************************************
;-------------------------------------------------------------------------------
; Device parameterization Operations
;-------------------------------------------------------------------------------
Axis:measure = 10000
Axis:pulse = 40000              ;how to calculate measure and pulse is explained in the appropriate section.*
Axis:maxvel = 100000            ;How to calculate Maxvel is explained in the appropriate section.*
Axis:maxpos = 999999            ;Maximum quota
Axis:minpos = -999999           ;Minimum quota
Axis:slowpos = 100              ;Slow-down quota
Axis:tollp = 10                 ;Positive tolerance
Axis:tolln = 50                 ;Negative tolerance
Axis:slowdly = 50               ;Slow-down time
Axis:tbrake = 30                ;Brake Intervention Time
Axis:overpos = 0                ;Over-quota Recovery play
Axis:bklashmode = 0             ;Mode Recovery play
Axis:ninert = 1                 ;Number of inertia bands
Axis:maxiner = 100              ;Maximum inertia after recalculation
Axis:inertmode = 1              ;Mode inertia recalculation
Axis:tool = 0                   ;Tool thickness
Axis:dobiner = 0                ;Mode forward/backward inertia
Axis:enstol = 0                 ;Enable START with axis in tolerance
Axis:unitvel = 0                ;Speed Time Unit (speed in Um/min)
Axis:decpt = 0                  ;Decimal digits in speed calculation
Axis:tinv = 50                  ;Axis inversion Time
Axis:toll = 5                   ;Tolerance (set in Um)
Axis:toldly = 10                ;Time Delay activation tolerance status
Axis:prsmode = 0                ;Homing function mode
Axis:prspos = 0                 ;Homing offset
Axis:prsdir = 0                 ;Direction of movement for home location search
Axis:prsvel = (20 * Axis:maxvel)/100  ;Search speed for activation digital input enabling (20% of the max.vel.)
Axis:sprsvel = (10 * Axis:maxvel)/100 ;Search speed of the home position (10% of the max.vel.)
Axis:breaktype = 0              ;Brake Intervention Logic
Axis:slowtype = 0               ;Slow-down calculation type
Axis:maxslow = 8                ;Slow-down Maximum calculated
Axis:minslow = 80               ;Slow-down Minimum calculated

INIT Axis                       ;Initializes the device
WAIT Axis:st_init               ;Wait for the device to be initialized
CNTUNLOCK Axis                  ;Unlock location capture
WAIT NOT Axis:st_cntlock        ;Wait for the capture of the location to be unlocked
CNTDIR Axis                     ;Sets the way of position acquisition
WAIT NOT Axis:st_cntrev         ;Wait for the way of position acquisition to be set
REGON Axis                      ;Enable adjustment
WAIT NOT Axis:st_regoff         ;Wait for the adjustment to be enabled

IF (slQuotaPos EQ 0)            ;If the axis placement quota is zero
  slQuotaPos = 2000             ;Set a placement quota
ENDIF

;-------------------------------------------------------------------------------
; Positioning operations
;-------------------------------------------------------------------------------
; ------------------------------- Used Variables -------------------------------
; slQuotaPos: Settable variable representing the axis placement dimension
; ---------------------------------- Used flags ----------------------------------
; gfMovMan:   Manual movement in progress
; gfMovAuto:  Automatic movement in progress
;-------------------------------------------------------------------------------

MAIN:
;-------------------------------------------------------------------------------
; Outputs management
;-------------------------------------------------------------------------------
ofToll = Axis:st_toll           ;Set the tolerance output as the tolerance status
```

```
;-------------------------------------------------------------------------------
; Automatic movement management
;-------------------------------------------------------------------------------
IF ifStart                          ;Waits for the START input
  IF NOT gfMovMan                   ;Check that there are no manual movements
    IF Axis:st_still                ;Check that the axis is stopped
      Axis:setpos = slQuotaPos      ;Set the placement quota
      START Axis                    ;Executes the start of the axis
      gfMovAuto = 1                 ;Report automatic movement in progress
    ENDIF
  ENDIF
ENDIF
IF ifStop                           ;Waits for the STOP input
  IF NOT Axis:st_still              ;Check that the axle is NOT stopped
    STOP Axis                       ;Executes the axis STOP
  ENDIF
ENDIF
IF gfMovAuto                        ;Check automatic motion reporting in progress
  IF Axis:st_still                  ;Check that the axle is stopped
    gfMovAuto = 0                   ;Reset Automatic Motion Status
  ENDIF
ENDIF

;-------------------------------------------------------------------------------
; Manual movement management (JOG)
;-------------------------------------------------------------------------------
IF ifAvMan                          ;Wait for manual motion input
  IF NOT (gfMovAuto OR gfMovMan)    ;Check that there are no automatic or manual movements in progress
    IF Axis:st_still                ;Check that the axle is stopped
      MANFFW Axis                   ;Forward axis in manual
      gfMovMan = 1                  ;Manual movement in progress reports
    ENDIF
  ENDIF
ENDIF
IF ifInMan                          ;Wait for manual motion input
  IF NOT (gfMovAuto OR gfMovMan)    ;Check that there are no automatic or manual movements in progress
    IF Axis:st_still                ;Check that the axle is stopped
      MANFBW Axis                   ;Forward axis in manual
      gfMovMan = 1                  ;Report Manual movement in progress
    ENDIF
  ENDIF
ENDIF
IF gfMovMan                         ;If the axis moves in manual
  IF NOT (ifAvMan OR ifInMan)       ;If the manual forward and backward inputs are off
    STOP Axis                       ;Stop the axis
    gfMovMan = 0                    ;Remove the manual moving axis signal
  ENDIF
ENDIF

;-------------------------------------------------------------------------------
; Final operations
;-------------------------------------------------------------------------------
WAIT 1
JUMP MAIN
END
```

How to calculate measure and pulse is explained in the appropriate section.